



Libro de Sílabos

Carrera Profesional de Ingeniería de
Software

- 2021-II -

Lima: 7 de julio de 2021

Equipo de Trabajo

Ernesto Cuadros-Vargas (Editor)

Presidente de la Sociedad Peruana de Computación (SPC) 2001-2007, 2009
Miembro del *Steering Committee de ACM/IEEE-CS Computing Curricula for
Computer Science (CS2013)*

Miembro del *Steering Committee de ACM/IEEE-CS Computing Curricula
2020 (CS2020)*

email: ecuadros@spc.org.pe

<http://socios.spc.org.pe/ecuadros>

Índice general

Primer Semestre	5
1.1. SFW52098. Estructuras Discretas I	5
1.2. SFW52099. Introducción a la Ciencia de la Computación	10
1.3. MAC41010. Fundamentos de Cálculo	17
1.4. COM42008. Lenguaje	20
1.5. ENG42003. English I	22
1.6. PSI53001. Desarrollo Humano	25
Segundo Semestre	29
2.1. SFW52096. Estructuras Discretas II	29
2.2. SFW52097. Programación Orientada a Objetos I	34
2.3. MAC41001. Cálculo de una Variable	42
2.4. COM42009. Lenguaje II	46
2.5. ENG42005. Inglés II	48
Tercer Semestre	51
3.1. SFW52094. Desarrollo Basado en Plataformas	51
3.2. SFW52095. Programación Orientada a Objetos II	57
3.3. MAC41002. Cálculo de varias Variables	68
3.4. ENG42007. English III	72
3.5. EGI61009. Arquitectura de Computadores	76
3.6. GES51001. Administración para los negocios	84
Cuarto Semestre	86
4.1. SFW52091. Gerenciamiento de Datos I	86
4.2. SFW52092. Teoría de la Computación	93
4.3. SFW52093. Algoritmos y Estructuras de Datos	98
4.4. ENG42009. English IV	101
4.5. EST41005. Estadística Descriptiva y Probabilidades	104
4.6. IIS52024. Sistemas Operativos	107
4.7. ECO51002. Economía General	116
Quinto Semestre	118
5.1. SFW52087. Compiladores	118
5.2. SFW52088. Ingeniería de Software I	123
5.3. SFW52089. Análisis y Diseño de Algoritmos	128
5.4. SFW52090. Gerenciamiento de Datos II	134
5.5. ENG42012. Inglés V	140

5.6.	FIS41001. Física General	143
5.7.	MAR52008. Marketing	146
Sexto Semestre		148
6.1.	SFW52082. Sistemas Inteligentes	148
6.2.	SFW52083. Sistemas de Infomación	157
6.3.	SFW52084. Estructuras de Datos Avanzadas	160
6.4.	SFW52085. Programación Competitiva	164
6.5.	SFW52086. Ingeniería de Software II	169
6.6.	ENG42032. English VI: RREL	175
6.7.	FIN52004. Finanzas Empresariales	178
Séptimo Semestre		180
7.1.	SFW52077. Ingeniería de Software III	180
7.2.	SFW52078. Experiencia de Usuario (UX)	186
7.3.	SFW52081. Metodología de la Investigación en Computación	193
7.4.	SFW52079. Aprendizaje Automático	196
7.5.	SFW52080. Computación Gráfica	205
7.6.	MAC41014. Álgebra Lineal	212
7.7.	IIS53004. Redes y Telecomunicaciones	216
7.8.	BSC51001. Biología	221
Octavo Semestre		223
8.1.	SFW62069. Formación de Empresas de Base Tecnológica I	223
8.2.	SFW62070. Computación en la Sociedad	228
8.3.	SFW62071. Computación Paralela y Distribuida	238
8.4.	SFW62073. Seguridad en Computación	246
8.5.	SFW62074. Proyecto de Final de Carrera I	257
8.6.	SFW62075. Matemática aplicada a la computación	260
8.7.	SFW62072. Visión Computacional	262
8.8.	SFW62076. Biología Computacional	264
Noveno Semestre		266
9.1.	SFW62064. Proyecto de Final de Carrera II	266
9.2.	SFW62065. Bioinformática	269
9.3.	SFW62068. Formación de Empresas de Base Tecnológica II	274
9.4.	SFW62066. Tópicos en Ingeniería de Software	278
9.5.	SFW62067. Tópicos en Computación Gráfica	283
9.6.	IIS64000. Big Data y Analítica de Datos	285
9.7.	INV51055. Fundamentos del Metodo Científico	288
Décimo Semestre		290
10.1.	SFW62058. Proyecto de Final de Carrera III	290
10.2.	SFW62059. Computación Evolutiva	292
10.3.	SFW62060. Formación de Empresas de Base Tecnológica III	298
10.4.	SFW62061. Cloud Computing	301
10.5.	SFW62062. Internet de las Cosas	307
10.6.	IIS64016. Robótica	315
10.7.	ETM41007. Ética y Ciudadanía	319



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52098. Estructuras Discretas I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 4
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: 2 (Semanal)
2.4 Horas autónomas	: 128 (horas)
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Obligatorio
2.7 Modalidad	: Presencial
2.8 Prerrequisitos	: Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Las estructuras discretas proporcionan los fundamentos teóricos necesarios para la computación. Estos fundamentos no sólo son útiles para desarrollar la computación desde un punto de vista teórico como sucede En el curso de la teoría computacional, pero también es útil para la práctica de la informática; En particular en aplicaciones tales como verificación, Criptografía, métodos formales, etc.

5. OBJETIVOS

- Aplicar correctamente conceptos de matemáticas finitas (conjuntos, relaciones, funciones) para representar datos de problemas reales.
- Modelar situaciones reales descritas en lenguaje natural, usando lógica proposicional y lógica predicada.
- Determinar las propiedades abstractas de las relaciones binarias.
- Elegir el método de demostración más apropiado para determinar la veracidad de una propuesta y construir argumentos matemáticos correctos.
- Interpretar soluciones matemáticas a un problema y determinar su fiabilidad, ventajas y desventajas.
- Expresar el funcionamiento de un circuito electrónico simple usando álgebra booleana.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Familiarizarse**)
- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Familiarizarse**)

7. TEMAS

Unidad 1: Funciones, relaciones y conjuntos (22 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Conjuntos:<ul style="list-style-type: none">– Diagramas de Venn– Unión, intersección, complemento– Producto Cartesiano– Potencia de conjuntos– Cardinalidad de Conjuntos finitos• Relaciones:<ul style="list-style-type: none">– Reflexividad, simetria, transitividad– Relaciones de equivalencia– Relación de orden parcial y conjuntos parcialmente ordenados– Elementos extremos de un conjunto parcialmente ordenado• Funciones:<ul style="list-style-type: none">– Suryecciones, inyecciones, biyecciones– Inversas– Composición	<ul style="list-style-type: none">• Explicar con ejemplos la terminología básica de funciones, relaciones y conjuntos [Evaluar]• Realizar las operaciones asociadas con conjuntos, funciones y relaciones [Evaluar]• Relacionar ejemplos prácticos para conjuntos funciones o modelos de relación apropiados e interpretar la asociación de operaciones y terminología en contexto [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Gri03], [Ros07], [Vel06]	

Unidad 2: Lógica básica (14 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Lógica proposicional. • Conectores lógicos. • Tablas de verdad. • Forma normal (conjuntiva y disyuntiva) • Validación de fórmula bien formada. • Reglas de inferencia proposicional (conceptos de modus ponens y modus tollens) • Logica de predicados: <ul style="list-style-type: none"> – Cuantificación universal y existencial • Limitaciones de la lógica proposicional y de predicados (ej. problemas de expresividad) 	<ul style="list-style-type: none"> • Convertir declaraciones lógicas desde el lenguaje informal a expresiones de lógica proposicional y de predicados [Usar] • Aplicar métodos formales de simbolismo proposicional y lógica de predicados, como el cálculo de la validez de formulas y cálculo de formas normales [Usar] • Usar reglas de inferencia para construir demostraciones en lógica proposicional y de predicados [Usar] • Describir como la lógica simbólica puede ser usada para modelar situaciones o aplicaciones de la vida real, incluidos aquellos planteados en el contexto computacional como análisis de software (ejm. programas correctores), consulta de base de datos y algoritmos [Familiarizarse] • Aplicar demostraciones de lógica formal y/o informal, pero rigurosa, razonamiento lógico para problemas reales, como la predicción del comportamiento de software o solución de problemas tales como rompecabezas [Usar] • Describir las fortalezas y limitaciones de la lógica proposicional y de predicados [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Ros07], [Gri03], [Vel06]	

Unidad 3: Técnicas de demostración (14 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Nociones de implicancia, equivalencia, conversión, inversa, contrapositivo, negación, y contradicción • Estructura de pruebas matemáticas. • Demostración directa. • Refutar por contraejemplo. • Demostración por contradicción. • Inducción sobre números naturales. • Inducción estructural. • Inducción leve y fuerte (Ej. Primer y Segundo principio de la inducción) • Definiciones matemáticas recursivas. • Conjuntos bien ordenados. 	<ul style="list-style-type: none"> • Identificar la técnica de demostración utilizada en una demostración dada [Evaluar] • Describir la estructura básica de cada técnica de demostración (demostración directa, demostración por contradicción e inducción) descritas en esta unidad [Usar] • Aplicar las técnicas de demostración (demostración directa, demostración por contradicción e inducción) correctamente en la construcción de un argumento solido [Usar] • Determine que tipo de demostración es la mejor para un problema dado [Evaluar] • Explicar el paralelismo entre ideas matemáticas y/o inducción estructural para la recursión y definir estructuras recursivamente [Familiarizarse] • Explicar la relación entre inducción fuerte y débil y dar ejemplos del apropiado uso de cada uno [Evaluar] • Enunciar el principio del buen-orden y su relación con la inducción matemática [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Ros07], [Sch12], [Vel06]	

Unidad 4: Representación de Datos (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Representaciones numéricas: signo magnitud, punto flotante. • Representaciones de otros objetos: conjuntos, relaciones, funciones 	<ul style="list-style-type: none"> • Conocer las formas de representación numérica como signo magnitud y punto flotante. [Evaluar]. • Llevar a cabo operaciones aritméticas utilizando las distintas formas de representación. [Evaluar]. • Conocer el estándar de punto flotante IEEE-754 [Familiarizarse].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Ros07], [Gri03], [Vel06]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.
- [Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 7 ed. Mc Graw Hill, 2007.
- [Sch12] Edward R. Scheinerman. *Mathematics: A Discrete Introduction*. 3 ed. Brooks Cole, 2012.
- [Vel06] Daniel J. Velleman. *How to Prove It: A Structured Approach*. Ed. by Cambridge University Pres. 2nd. 2006. ISBN: 978-0521675994.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52099. Introducción a la Ciencia de la Computación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 3
2.2 Horas de teoría	: 1 (Semanal)
2.3 Horas de práctica	: 2 (Semanal)
2.4 Horas autónomas	: 96 (horas)
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Obligatorio
2.7 Modalidad	: Presencial
2.8 Prerrequisitos	: Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este es el primer curso en la secuencia de los cursos introductorios a la Ciencia de la Computación. En este curso se pretende cubrir los conceptos señalados por la Computing Curricula IEEE-CS/ACM 2013. La programación es uno de los pilares de la Ciencia de la Computación; cualquier profesional del Área, necesitará programar para concretizar sus modelos y propuestas. Este curso introduce a los participantes en los conceptos fundamentales de este arte. Los tópicos incluyen tipos de datos, estructuras de control, funciones, listas, recursividad y la mecánica de la ejecución, prueba y depuración.

5. OBJETIVOS

- Introducir los conceptos fundamentales de programación.
- Desarrollar su capacidad de abstracción utilizar un lenguaje de programación.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Familiarizarse**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Familiarizarse**)
- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (**Familiarizarse**)

7. TEMAS

Unidad 1: Historia (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Pre-historia – El mundo antes de 1946. • Historia del hardware, software, redes. • Pioneros de la Computación. • Historia de Internet. 	<ul style="list-style-type: none"> • Identificar importantes tendencias en la historia del campo de la computación [Familiarizarse] • Identificar las contribuciones de varios pioneros en el campo de la computación [Familiarizarse] • Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse] • Comparar la vida diaria antes y después de la llegada de los ordenadores personales y el Internet [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [BB19], [Gut13], [Zel10]	

Unidad 2: Sistemas de tipos básicos (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tipos como conjunto de valores junto con un conjunto de operaciones. <ul style="list-style-type: none"> – Tipos primitivos (p.e. números, booleanos) – Composición de tipos contruídos de otros tipos (p.e., registros, uniones, arreglos, listas, funciones, referencias) • Asociación de tipos de variables, argumentos, resultados y campos. • Tipo de seguridad y los errores causados por el uso de valores de manera incompatible dadas sus tipos previstos. 	<ul style="list-style-type: none"> • Tanto para tipo primitivo y un tipo compuesto, describir de manera informal los valores que tiene dicho tipo [Familiarizarse] • Para un lenguaje con sistema de tipos estático, describir las operaciones que están prohibidas de forma estática, como pasar el tipo incorrecto de valor a una función o método [Familiarizarse] • Describir ejemplos de errores de programa detectadas por un sistema de tipos [Familiarizarse] • Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar] • Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar] • Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Gut13], [Zel10]	

Unidad 3: Conceptos Fundamentales de Programación (9 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Sintaxis y semántica básica de un lenguaje de alto nivel.• Variables y tipos de datos primitivos (ej., numeros, caracteres, booleanos)• Expresiones y asignaciones.• Operaciones básicas I/O incluyendo archivos I/O.• Estructuras de control condicional e iterativas.• Paso de funciones y parámetros.• Concepto de recursividad.	<ul style="list-style-type: none">• Analiza y explica el comportamiento de programas simples que involucran estructuras fundamentales de programación variables, expresiones, asignaciones, E/S, estructuras de control, funciones, paso de parámetros, y recursividad [Evaluar]• Identifica y describe el uso de tipos de datos primitivos [Familiarizarse]• Escribe programas que usan tipos de datos primitivos [Usar]• Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar]• Diseña, implementa, prueba, y depura un programa que usa cada una de las siguientes estructuras de datos fundamentales: cálculos básicos, E/S simple, condicional estándar y estructuras iterativas, definición de funciones, y paso de parámetros [Usar]• Escribe un programa que usa E/S de archivos para brindar persistencia a través de ejecuciones múltiples [Usar]• Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Familiarizarse]• Describe el concepto de recursividad y da ejemplos de su uso [Evaluar]• Identifica el caso base y el caso general de un problema basado en recursividad [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Gut13], [Zel10]	

Unidad 4: Análisis Básico (2 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Diferencias entre el mejor, el esperado y el peor caso de un algoritmo.• Definición formal de la Notación Big O.• Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial.• Uso de la notación Big O.• Análisis de algoritmos iterativos y recursivos.	<ul style="list-style-type: none">• Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Familiarizarse]• En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Familiarizarse]• Indique la definición formal de Big O [Familiarizarse]• Use la notación formal de la Big O para dar límites superiores asintóticos en la complejidad de tiempo y espacio de los algoritmos [Usar]• Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Gut13], [Zel10]	

Unidad 5: Algoritmos y Estructuras de Datos fundamentales (8 horas)

Competencias esperadas:

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo. • Algoritmos de búsqueda secuencial y binaria. • Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción) • Algoritmos de ordenamiento con peor caso o caso promedio en $O(N \lg N)$ (Quicksort, Heapsort, Mergesort) • Tablas Hash, incluyendo estrategias para evitar y resolver colisiones. • Árboles de búsqueda binaria: <ul style="list-style-type: none"> – Operaciones comunes en árboles de búsqueda binaria como seleccionar el mínimo, máximo, insertar, eliminar, recorrido en árboles. • Grafos y algoritmos en grafos: <ul style="list-style-type: none"> – Representación de grafos (ej., lista de adyacencia, matriz de adyacencia) – Recorrido en profundidad y amplitud • Montículos (Heaps) • Grafos y algoritmos en grafos: <ul style="list-style-type: none"> – Problema de corte máximo y mínimo – Búsqueda local • Búsqueda de patrones y algoritmos de cadenas/texto (ej. búsqueda de subcadena, búsqueda de expresiones regulares, algoritmos de subsecuencia común más larga) 	<ul style="list-style-type: none"> • Implementar algoritmos numéricos básicos [Usar] • Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar] • Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Usar] • Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Familiarizarse] • Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Familiarizarse] • Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse] • Explicar como el balanceamiento del arbol afecta la eficiencia de varias operaciones de un arbol de búsqueda binaria [Familiarizarse] • Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Usar] • Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto en específico [Evaluar] • Describir la propiedad del heap y el uso de heaps como una implementación de colas de prioridad [Familiarizarse] • Resolver problemas usando algoritmos de grafos, incluyendo camino más corto de una sola fuente y camino más corto de todos los pares, y como mínimo un algoritmo de arbol de expansion minima [Usar] • Trazar y/o implementar un algoritmo de comparación de string [Usar]
<p>Aprendizaje autónomo</p> <ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
<p>Lecturas : [Gut13], [Zel10]</p>	

Unidad 6: Algoritmos y Diseño (9 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Conceptos y propiedades de los algoritmos <ul style="list-style-type: none"> – Comparación informal de la eficiencia de los algoritmos (ej., conteo de operaciones) • Rol de los algoritmos en el proceso de solución de problemas • Estrategias de solución de problemas <ul style="list-style-type: none"> – Funciones matemáticas iterativas y recursivas – Recorrido iterativo y recursivo en estructura de datos – Estrategias Divide y Conquistar • Conceptos y principios fundamentales de diseño <ul style="list-style-type: none"> – Abstracción – Descomposición de Program – Encapsulamiento y camuflaje de información – Separación de comportamiento y aplicación 	<ul style="list-style-type: none"> • Discute la importancia de los algoritmos en el proceso de solución de un problema [Familiarizarse] • Discute como un problema puede ser resuelto por múltiples algoritmos, cada uno con propiedades diferentes [Familiarizarse] • Crea algoritmos para resolver problemas simples [Usar] • Usa un lenguaje de programación para implementar, probar, y depurar algoritmos para resolver problemas simples [Usar] • Implementa, prueba, y depura funciones recursivas simples y sus procedimientos [Usar] • Determina si una solución iterativa o recursiva es la más apropiada para un problema [Evaluar] • Implementa un algoritmo de divide y vencerás para resolver un problema [Usar] • Aplica técnicas de descomposición para dividir un programa en partes más pequeñas [Usar] • Identifica los componentes de datos y el comportamiento de múltiples tipos de datos abstractos [Usar] • Implementa un tipo de dato abstracto coherente, con la menor pérdida de acoplamiento entre componentes y comportamientos [Usar] • Identifica las fortalezas y las debilidades relativas entre múltiples diseños e implementaciones de un problema [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Gut13], [Zel10]	

Unidad 7: Métodos de Desarrollo (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Entornos modernos de programación: <ul style="list-style-type: none"> – Búsqueda de código. – Programación usando librería de componentes y sus APIs. 	<ul style="list-style-type: none"> • Construir y depurar programas que utilizan las bibliotecas estándar disponibles con un lenguaje de programación elegido [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Gut13], [Zel10]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [BB19] J. Glenn Brookshear and Dennis Brylow. *Computer Science: An Overview*. Ed. by PEARSON. Global Edition. Pearson, 2019. ISBN: 1292263423. URL: <http://www.pearsonhighered.com/brookshear>.
- [Gut13] John V Guttag. . *Introduction To Computation And Programming Using Python*. MIT Press, 2013.
- [Zel10] John Zelle. *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates Inc, 2010.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

MAC41010. Fundamentos de Cálculo (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 4
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: -
2.4 Horas autónomas	: 128 (horas)
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Obligatorio
2.7 Modalidad	: Presencial
2.8 Prerrequisitos	: Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El curso desarrolla los conocimientos y capacidades para reconocer, evaluar y aplicar los efectos de los fenómenos físicos relacionados a la mecánica en el campo de la ingeniería. En la industria en general, el control de los procesos, el funcionamiento de las máquinas, su mantenimiento, etc., siempre están regidas por algún tipo de manifestación física. Debido a eso, es importante para el estudiante entender los fundamentos de los fenómenos físicos, las leyes que los rigen, su manifestación y la forma de detectarlos. El presente curso permitirá al estudiante comprender e identificar los fenómenos físicos relacionados a la mecánica con el fin de que puedan controlar sus efectos sobre algún proceso técnico.

5. OBJETIVOS

- Capacidad de aplicar los conocimientos de ciencias.
- Capacidad de diseñar y llevar a cabo experimentos.
- Capacidad de aplicar conocimientos de computación y de matemáticas.
- Capacidad de desarrollar principios de investigación con nivel internacional.

6. COMPETENCIAS

CG1) Comunicación integral. (**Evaluar**)

CG3) Investigación. (**Evaluar**)

CG4) Gestiona recursos. (**Evaluar**)

CG5) Desarrollo humano. (**Evaluar**)

7. TEMAS

Unidad 1: Trabajo, Energía y Potencia (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Definición de trabajo y la relación entre trabajo neto y energía cinética. Potencia y Eficiencia 	<ul style="list-style-type: none"> Determinar las variables que afectan la oposición a la traslación y la oposición a la rotación (momento de inercia) y calcular la energía cinética de traslación y rotación. Calcular el trabajo de una fuerza, aplicar el Teorema de trabajo neto y energía sobre un sistema de la vida real, y determinar la potencia y eficiencia.
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [Hug13], [Hew07]	

Unidad 2: Cinemática (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Sistemas de referencia espacial y temporal. Velocidad media, aceleración media, lineal y angular. Vectores de posición, velocidad y aceleración, lineal y angular. Relación entre cinemática lineal y angular. 	<ul style="list-style-type: none"> Entender los conceptos de cinemática sistema de referencia espacial y temporal, y trayectoria y determinar la posición, velocidad y aceleración lineal y angular, según un contexto físico o de gráficos. Descomponer la aceleración lineal, en función a un sistema de coordenadas, para poder describir la posición y en aceleración radial y tangencial. Determina la posición, velocidad y aceleración, usando el cálculo diferencial e integral.
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [Hug13], [Hew07]	

Unidad 3: Las tres leyes de Newton (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Las 3 leyes de Newton y su aplicación en partículas. Momento de una fuerza. Rotación de un cuerpo rígido. 	<ul style="list-style-type: none"> Plantear las ecuaciones de rotación y traslación para un sólido y aplicar las leyes de Newton. Analizar las características de la fuerza de fricción. Calcular la fuerza radial neta y la fuerza centrípeta neta. Calcular el centro de masa y analizar la relación entre las variables de fuerza neta, tiempo y cambio de velocidad.
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [Hug13], [Hew07]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[Hew07] Paul Hewitt. *Física conceptual*. 10th. Pearson Educación, 2007.

[Hug13] Roger A. Freedman Hugh D. Young. *Física universitaria*. 13th. Pearson, 2013.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

COM42008. Lenguaje (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 4 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Write justification for this course here ...

5. OBJETIVOS

- Write your first goal here.
- Write your second goal here.
- Just in case you need more goals write them here

6. COMPETENCIAS

CG1) Comunicación integral. (Evaluar)

7. TEMAS

Unidad 1: title for the unit goes here (0 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Topic1• Topic2• Topic3	<ul style="list-style-type: none">• Learning outcome1 [Levelforthislearningoutcome].• i.e Apply computing in complex problems [Usar].• Create a search engine [Evaluar].• Study data structures [Familiarizarse].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Bibitem1], [Bibitem2]	

Unidad 2: another unit goes here (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Topic1 	<ul style="list-style-type: none"> • Learning outcome xyz [Levelforthislearningoutcome].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Bibitem3], [Bibitem1]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

ENG42003. English I (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El aprendizaje del idioma inglés en los estudiantes de nivel superior se ha convertido en elemento necesario para su desarrollo académico, personal y profesional.

La mayor cantidad de literatura académica de los diversos campos del saber es redactada en inglés, lo cual garantiza que las personas que cuenten con el dominio del idioma puedan siempre estar actualizados.

Asimismo, el conocimiento de este idioma permite tener perspectivas sociales y culturales más amplias. En ese sentido, para un efectivo aprendizaje de la lengua es necesario el desarrollo de las cuatro (4) habilidades; escuchar, hablar, leer y escribir considerando los lineamientos del Marco Común Europeo de Referencia de Lengua - MCERL.

5. OBJETIVOS

- Comunicarse e intercambiar información de manera limitada en situaciones predecibles de forma general.
- Alcanzar el nivel B1 según el MCERL.
- Manejar terminología propia del área de estudios.

6. COMPETENCIAS

CG2) Comunicación bilingüe. (Evaluar)

7. TEMAS

Unidad 1: Hello everybody (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Greetings and farewells.• To be (affirmative, question and negatives).• Connector (addition – contrast).	<ul style="list-style-type: none">• Greeting people• Introduce oneself and others
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [De 02]	

Unidad 2: Meeting people (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • To be (affirmative, question and negatives) • Questions (what / where / when- to be) • Possessive adjectives • Connector (addition – contrast) 	<ul style="list-style-type: none"> • Exchange basic personal information
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 3: What are you doing? (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Present continuous (affirmative, question and negatives). • Questions (what / where / when). • Connector (addition – contrast). 	<ul style="list-style-type: none"> • Talk about ongoing activities.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 4: My clothes (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Personal pronouns. • Possessive pronouns. • Common adjectives. • Demonstrative adjectives. • Connector (addition – contrast). 	<ul style="list-style-type: none"> • Describe clothes • Ask and answer about possession
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 5: Smaller or Bigger? (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Comparatives. • Intensifiers - very basic. • Superlatives. • Connector (addition – contrast). 	<ul style="list-style-type: none"> • Compare clothing items
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 6: Can you speak English? (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Design by the teacher • Connector (addition – contrast). 	<ul style="list-style-type: none"> • Get familiar with technical expressions of the field.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[De 02] Consejo De Europa. “Marco común europeo de referencia para las lenguas”. In: *Strasburgo: Consejo de Europa, Ministerio de Educación, Cultura y Deporte/Instituto Cervantes* (2002).



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de

Ingeniería de Software

Sílabo 2021-II

1. CURSO

PSI53001. Desarrollo Humano (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 4
2.2 Horas de teoría	: 4 (Semanal)
2.3 Horas de práctica	: -
2.4 Horas autónomas	: 128 (horas)
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Obligatorio
2.7 Modalidad	: Presencial
2.8 Prerrequisitos	: Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El ingreso a la universidad es un momento de nuevos desafíos y decisiones en la vida de una persona. En ese sentido, la Universidad Católica San Pablo busca, mediante el presente espacio, escuchar y acoger al joven ingresante con sus inquietudes y anhelos personales, presentar la identidad y misión de la universidad como su “alma mater”, señalando los principales desafíos que el futuro profesional enfrentará en el mundo actual y orientando a nuestros jóvenes estudiantes, a través de diversos principios, medios y otros recursos, con el fin de que puedan formarse integralmente y desplegarse plenamente en la fascinante aventura de la vida universitaria. Su realización como buen profesional depende de una buena formación personal y cultural que le brinde horizontes amplios, que sustenten y proyecten su conocimiento y quehacer técnicos e intelectuales y que le permitan contribuir siendo agentes de cambio cultural y social.

5. OBJETIVOS

- Que el alumno canalice sus inquietudes y anhelos a través del encuentro y descubrimiento de sí mismo, que le brinden espacios de análisis y reflexión personales para asumir posturas bien fundamentadas hacia los valores e ideales de su entorno. Mediante su inserción en la vida universitaria, logrará una disposición de apertura a su propio mundo interior y a su misión en el mundo, cuestionando su cosmovisión y a sí mismo para obtener un conocimiento y crecimiento personales que permitan su despliegue integral y profesional.

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: (24 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Introducción al curso: presentación y dinámicas. • Sentido de la Vida, búsqueda de propósito y vocación profesional. • Obstáculos para el autoconocimiento: el ruido, la falta de comunicación, la mentira existencial, máscaras. • Ofertas Intramundanas: Hedonismo, Relativismo, Consumismo, Individualismo, Inmanentismo • Las consecuencias: la falta de interioridad, masificación y el desarraigo, soledad • Los vicios capitales como plasmación en lo personal 	<ul style="list-style-type: none"> • Identificar y caracterizar la propia cosmovisión y los criterios personales predominantes en sí mismos acerca del propósito y sentido de la vida y la felicidad. [Usar] • Crear un vínculo de confianza con el docente del curso para lograr apertura a nuevas perspectivas[Usar].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [San08], [Ril41], [Mar95], [Fra91]	

Unidad 2: (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Quién soy yo, las preguntas fundamentales • El hombre como unidad. • El hombre: nostalgia de infinito. • La libertad como elemento fundamental en las elecciones personales: la experiencia del mal. • Análisis del Amor y la Amistad. • Aceptación y Reconciliación personal. • Llamados a ser personas: la vivencia de la virtud según un modelo concreto. 	<ul style="list-style-type: none"> • Reconocer la importancia de iniciar un proceso de autoconocimiento [Usar]. • Identificar las manifestaciones que evidencian la unidad de la persona humana y su anhelo de trascendencia [Usar]. • Contrastar los modelos de amor y libertad ofertados por la cultura actual con los propuestos en el curso [Usar]. • Distinguir los criterios que conducen a una recta valoración personal [Usar].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Gua94], [Fro59], [Fig02], [Pie07]	

Unidad 3: (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Origen y propósito de la Universidad: breve reseña histórica. • La identidad católica de la UCSP: comunidad académica, búsqueda de la verdad, la formación integral y la evangelización de la cultura. • Proyecto final. 	<ul style="list-style-type: none"> • Conocer e identificar a la UCSP dentro del contexto histórico de las universidades [Usar]. • Reconocer a su universidad como un ámbito de despliegue y espacio para crear cultura [Usar]. • Afirmar, desde su vocación profesional, la necesidad de transformar el mundo que le toca vivir [Usar].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Jua01], [Gua12], [07]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [07] *Identidad y Misión 10 Años Comprometidos con el Futuro Mensajes por el X Aniversario de la UCSP*. Letrera Gráfica S.A.C, 2007.
- [Fig02] Luis Fernando Figari. *Nostalgia de Infinito*. Fondo Editorial, 2002.
- [Fra91] Viktor Frankl. *El hombre en búsqueda de sentido*. UCSP:616.891 F80. Editorial Herder, 1991.
- [Fro59] Erich Fromm. *El arte de amar*. Ediciones Paidós Iberica, 1959.
- [Gua12] Romano Guardini. *Tres escritos sobre la Universidad*. EUNSA, 2012.
- [Gua94] Romano Guardini. *La aceptación de sí mismo*. Lumen, 1994.
- [Jua01] Juan Pablo II. "Constitución Apostólica sobre las Universidades Católicas *Ex Corde Ecclesiae*". In: *Teológica Limense* (2001).
- [Mar95] Julian Marias. *La Felicidad Humana*. Alianza, 1995.
- [Pie07] Josef Pieper. *Las Virtudes Fundamentales*. Ediciones Rialp, Madrid, 2007.

- [Ril41] Rainer María Rilke. *Los cuadernos de Malte Lauridis Brigge*. Lossada, 1941.
- [San08] Alfredo Sanz. *El Hombre Moderno*. Ediciones Gladius, 2008.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52096. Estructuras Discretas II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52098. Estructuras Discretas I. (1 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Para entender las técnicas computacionales avanzadas, los estudiantes deberán tener un fuerte conocimiento de las diversas estructuras discretas, estructuras que serán implementadas y usadas en laboratorio en el lenguaje de programación.

5. OBJETIVOS

- Que el alumno sea capaz de modelar problemas de ciencia de la computación usando grafos y árboles relacionados con estructuras de datos.
- Que el alumno aplique eficientemente estrategias de recorrido para poder buscar datos de una manera óptima.
- Que el alumno utilice las diversas técnicas de conteo para resolver problemas computacionales.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Familiarizarse**)
- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Familiarizarse**)

7. TEMAS

Unidad 1: Lógica Digital y Representación de Datos (10 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Retículo: Tipos y propiedades.• Álgebras booleanas.• Funciones y expresiones booleanas.• Representación de las funciones booleanas: Disjuntiva normal y conjuntiva normal.• Puertas Lógicas.• Minimización del Circuito.	<ul style="list-style-type: none">• Explicar la importancia del álgebra booleana como una unificación de la teoría de conjuntos y la lógica proposicional [Evaluar].• Explicar las estructuras algebraicas del retículo y sus tipos [Evaluar].• Explicar la relación entre el retículo y el conjunto de ordenadas y el uso prudente para demostrar que un conjunto es un retículo [Evaluar].• Explicar las propiedades que satisfacen un álgebra booleana [Evaluar].• Demostrar si una terna formada por un conjunto y dos operaciones internas es o no Álgebra booleana [Evaluar].• Encuentra las formas canónicas de una función booleana [Evaluar].• Representar una función booleana como un circuito booleano usando puertas lógicas [Evaluar].• Minimizar una función booleana [Evaluar].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Ros07], [Gri03]	

Unidad 2: Fundamentos de conteo (40 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Técnicas de Conteo: <ul style="list-style-type: none"> – Conteo y cardinalidad de un conjunto – Regla de la suma y producto – Principio de inclusión-exclusión – Progresión geométrica y aritmética • Principio de las casillas. • Permutaciones y combinaciones: <ul style="list-style-type: none"> – Definiciones básicas – Identidad de Pascal – Teorema del binomio • Resolviendo relaciones de recurrencia: <ul style="list-style-type: none"> – Un ejemplo de una relación de recurrencia simple, como los números de Fibonacci – Otras ejemplos, mostrando una variedad de soluciones • Aritmetica modular basica 	<ul style="list-style-type: none"> • Aplicar argumentos de conteo, incluyendo las reglas del producto y de la suma, principio de inclusión-exclusión y progresiones aritméticas/geométricas [Familiarizarse] • Aplicar el principio de las casillas en el contexto de una demostración formal [Familiarizarse] • Calcular permutaciones y combinaciones en un conjunto, e interpreta su significado en el contexto de una aplicación en particular [Familiarizarse] • Mapear aplicaciones del mundo real a formalismos de conteo adecuados, como el determinar el número de formas de acomodar a un conjunto de personas alrededor de una mesa, sujeto a restricciones en la disposición de los asientos, o en el número de maneras de determinar ciertas manos en juegos de cartas (ejm. una casa llena) [Familiarizarse] • Resolver una variedad de relaciones de recurrencia básicas [Familiarizarse] • Analizar un problema para determinar las relaciones de recurrencia implícitas [Familiarizarse] • Realizar cálculos que involucran aritmética modular [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Gri03]	

Unidad 3: Árboles y Grafos (40 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Árboles. <ul style="list-style-type: none"> – Propiedades – Estrategias de recorrido • Grafos no dirigidos • Grafos dirigidos • Grafos ponderados • Árboles de expansión/bosques. • Isomorfismo en grafos. 	<ul style="list-style-type: none"> • Ilustrar mediante ejemplos la terminología básica de teoría de grafos, y de alguna de las propiedades y casos especiales de cada tipo de grafos/árboles [Familiarizarse] • Demostrar diversos métodos de recorrer árboles y grafos, incluyendo recorridos pre, post e inorden de árboles [Familiarizarse] • Modelar una variedad de problemas del mundo real en ciencia de la computación usando formas adecuadas de grafos y árboles, como son la representación de una topología de red o la organización jerárquica de un sistema de archivos [Familiarizarse] • Demostrar como los conceptos de grafos y árboles aparecen en estructuras de datos, algoritmos, técnicas de prueba (inducción estructurada), y conteos [Familiarizarse] • Explicar como construir un árbol de expansión de un grafo [Familiarizarse] • Determinar si dos grafos son isomorfos [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Joh99]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.
- [Joh99] Richard Johnsonbaugh. *Matemáticas Discretas*. Prentice Hall, México, 1999.
- [Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 7 ed. Mc Graw Hill, 2007.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52097. Programación Orientada a Objetos I (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : 2 (Semanal)
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : SFW52099. Introducción a la Ciencia de la Computación. (1^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este es el segundo curso en la secuencia de los cursos introductorios a la Ciencia de la Computación. El curso introducirá a los participantes en los diversos temas del área de computación como: algoritmos, estructuras de datos, ingeniería del software, etc.

5. OBJETIVOS

- Introducir al alumno a los fundamentos del paradigma de orientación a objetos, permitiendo asimilar los conceptos necesarios para desarrollar sistemas de información.

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Familiarizarse**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Familiarizarse**)

7. TEMAS

Unidad 1: Visión General de los Lenguajes de Programación (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Breve revisión de los paradigmas de programación.• Comparación entre programación funcional y programación imperativa.• Historia de los lenguajes de programación.	<ul style="list-style-type: none">• Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse]
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Str13], [Dei17]	

Unidad 2: Sistemas de tipos básicos (2 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Tipos como conjunto de valores junto con un conjunto de operaciones.<ul style="list-style-type: none">– Tipos primitivos (p.e. números, booleanos)– Composición de tipos construídos de otros tipos (p.e., registros, uniones, arreglos, listas, funciones, referencias)• Declaración de modelos (enlace, visibilidad, alcance y tiempo de vida).• Vista general del chequeo de tipos.	<ul style="list-style-type: none">• Tanto para tipo primitivo y un tipo compuesto, describir de manera informal los valores que tiene dicho tipo [Familiarizarse]• Para un lenguaje con sistema de tipos estático, describir las operaciones que están prohibidas de forma estática, como pasar el tipo incorrecto de valor a una función o método [Familiarizarse]• Describir ejemplos de errores de programa detectadas por un sistema de tipos [Familiarizarse]• Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar]• Dar un ejemplo de un programa que no verifique tipos en un lenguaje particular y sin embargo no tenga error cuando es ejecutado [Familiarizarse]• Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar]• Explicar como las reglas de tipificación definen el conjunto de operaciones que legales para un tipo [Familiarizarse]• Escribir las reglas de tipo que rigen el uso de un particular tipo compuesto [Usar]• Explicar por qué indecidibilidad requiere sistemas de tipo para conservadoramente aproximar el comportamiento de un programa [Familiarizarse]• Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar]• Discutir las diferencias entre, genéricos (<i>generics</i>), subtipo y sobrecarga [Familiarizarse]• Explicar múltiples beneficios y limitaciones de tipificación estática en escritura, mantenimiento y depuración de un software [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Str13], [Dei17]	

Unidad 3: Conceptos Fundamentales de Programación (6 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Sintaxis y semántica básica de un lenguaje de alto nivel.• Variables y tipos de datos primitivos (ej., numeros, caracteres, booleanos)• Expresiones y asignaciones.• Operaciones básicas I/O incluyendo archivos I/O.• Estructuras de control condicional e iterativas.• Paso de funciones y parámetros.	<ul style="list-style-type: none">• Analiza y explica el comportamiento de programas simples que involucran estructuras fundamentales de programación variables, expresiones, asignaciones, E/S, estructuras de control, funciones, paso de parámetros, y recursividad [Evaluar]• Identifica y describe el uso de tipos de datos primitivos [Familiarizarse]• Escribe programas que usan tipos de datos primitivos [Usar]• Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar]• Diseña, implementa, prueba, y depura un programa que usa cada una de las siguientes estructuras de datos fundamentales: cálculos básicos, E/S simple, condicional estándar y estructuras iterativas, definición de funciones, y paso de parámetros [Usar]• Escribe un programa que usa E/S de archivos para brindar persistencia a través de ejecuciones múltiples [Usar]• Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Evaluar]• Describe el concepto de recursividad y da ejemplos de su uso [Familiarizarse]• Identifica el caso base y el caso general de un problema basado en recursividad [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Str13], [Dei17]	

Unidad 4: Programación orientada a objetos (10 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Diseño orientado a objetos: <ul style="list-style-type: none"> – Descomposición en objetos que almacenan estados y poseen comportamiento – Diseño basado en jerarquía de clases para modelamiento • Lenguajes orientados a objetos para la encapsulación: <ul style="list-style-type: none"> – privacidad y la visibilidad de miembros de la clase – Interfaces revelan único método de firmas – clases base abstractas • Definición de las categorías, campos, métodos y constructores. • Las subclasses, herencia y método de alteración temporal. • Subtipificación: <ul style="list-style-type: none"> – Polimorfismo artículo Subtipo; upcasts implícitos en lenguajes con tipos. – Noción de reemplazo de comportamiento: los subtipos de actuar como supertipos. – Relación entre subtipos y la herencia. • Uso de colección de clases, iteradores, y otros componentes de la librería estándar. • Asignación dinámica: definición de método de llamada. 	<ul style="list-style-type: none"> • Diseñar e implementar una clase [Usar] • Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclasses [Usar] • Razonar correctamente sobre el flujo de control en un programa mediante el envío dinámico [Usar] • Comparar y contrastar (1) el enfoque procedurador/funcional- definiendo una función por cada operación con el cuerdo de la función proporcionando un caso por cada variación de dato - y (2) el enfoque orientado a objetos - definiendo una clase por cada variación de dato con la definición de la clase proporcionando un método por cada operación. Entender ambos enfoques como una definición de variaciones y operaciones de una matriz [Evaluar] • Explicar la relación entre la herencia orientada a objetos (codigo compartido y <i>overriding</i>) y subtipificación (la idea de un subtipo es ser utilizable en un contexto en el que espera al supertipo) [Familiarizarse] • Usar mecanismos de encapsulación orientada a objetos, tal como interfaces y miembros privados [Usar] • Definir y usar iteradores y otras operaciones sobre agregaciones, incluyendo operaciones que tienen funciones como argumentos, en múltiples lenguajes de programación, seleccionar la forma mas natural por cada lenguaje [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Str13], [Dei17]	

Unidad 5: Algoritmos y Diseño (3 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Estrategias de solución de problemas <ul style="list-style-type: none"> – Funciones matemáticas iterativas y recursivas – Recorrido iterativo y recursivo en estructura de datos – Estrategias Divide y Conquistar • Rol de los algoritmos en el proceso de solución de problemas • Estrategias de solución de problemas <ul style="list-style-type: none"> – Funciones matemáticas iterativas y recursivas – Recorrido iterativo y recursivo en estructura de datos – Estrategias Divide y Conquistar • Conceptos y principios fundamentales de diseño <ul style="list-style-type: none"> – Abstracción – Descomposición de Program – Encapsulamiento y camuflaje de información – Separación de comportamiento y aplicación 	<ul style="list-style-type: none"> • Discute la importancia de los algoritmos en el proceso de solución de un problema [Familiarizarse] • Discute como un problema puede ser resuelto por múltiples algoritmos, cada uno con propiedades diferentes [Familiarizarse] • Crea algoritmos para resolver problemas simples [Usar] • Usa un lenguaje de programación para implementar, probar, y depurar algoritmos para resolver problemas simples [Usar] • Implementa, prueba, y depura funciones recursivas simples y sus procedimientos [Usar] • Determina si una solución iterativa o recursiva es la más apropiada para un problema [Evaluar] • Implementa un algoritmo de divide y vencerás para resolver un problema [Usar] • Aplica técnicas de descomposición para dividir un programa en partes más pequeñas [Usar] • Identifica los componentes de datos y el comportamiento de múltiples tipos de datos abstractos [Usar] • Implementa un tipo de dato abstracto coherente, con la menor pérdida de acoplamiento entre componentes y comportamientos [Usar] • Identifica las fortalezas y las debilidades relativas entre múltiples diseños e implementaciones de un problema [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Str13], [Dei17]	

Unidad 6: Estrategias Algorítmicas (3 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmos de fuerza bruta. • Algoritmos voraces. • Divide y vencerás. • Backtracking recursivo. • Programación Dinámica. 	<ul style="list-style-type: none"> • Para cada una de las estrategias (fuerza bruta, algoritmo goloso, divide y vencerás, recursividad en reversa y programación dinámica), identifica un ejemplo práctico en el cual se pueda aplicar [Familiarizarse] • Utiliza un enfoque voraz para resolver un problema específico y determina si la regla escogida lo guía a una solución óptima [Evaluar] • Usa un algoritmo de divide-y-vencerás para resolver un determinado problema [Usar] • Usa recursividad en reversa a fin de resolver un problema como en el caso de recorrer un laberinto [Usar] • Usa programación dinámica para resolver un problema determinado [Usar] • Determina el enfoque algorítmico adecuado para un problema [Evaluar] • Describe varios métodos basados en heurísticas para resolver problemas [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Str13], [Dei17]	

Unidad 7: Análisis Básico (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Diferencias entre el mejor, el esperado y el peor caso de un algoritmo. 	<ul style="list-style-type: none"> • Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Str13], [Dei17]	

Unidad 8: Algoritmos y Estructuras de Datos fundamentales (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo. • Algoritmos de búsqueda secuencial y binaria. • Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción) • Algoritmos de ordenamiento con peor caso o caso promedio en $O(N \lg N)$ (Quicksort, Heapsort, Mergesort) 	<ul style="list-style-type: none"> • Implementar algoritmos numéricos básicos [Usar] • Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar] • Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Usar] • Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Familiarizarse] • Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse] • Explicar como el balanceamiento del arbol afecta la eficiencia de varias operaciones de un arbol de búsqueda binaria [Familiarizarse] • Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto en específico [Evaluar] • Trazar y/o implementar un algoritmo de comparación de string [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Str13], [Dei17]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[Dei17] Deitel & Deitel. *C++17 - The Complete Guide*. 10th. Pearson, 2017. ISBN: 978-0201734843.

[Str13] Bjarne Stroustrup. *The C++ Programming Language*. 4th. Addison-Wesley, 2013. ISBN: 978-0-321-56384-2.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

MAC41001. Cálculo de una Variable (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : MAC41010. Fundamentos de Cálculo. (1^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso es importante en la medida que proporciona modelos de optimización útiles para la toma de decisiones en negocios.

5. OBJETIVOS

- Reconocer, modelar, resolver, implementar e interpretar modelos de optimización no lineales y estocásticos en problemas reales.

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Programación no lineal (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Funciones convexas y cóncavas.• Soluciones de PNL con una variable.• Maximización y minimización no restringida con varias variables.• Métodos de ascenso no escalonado.• Condiciones de Kuhn-Tucker.	<ul style="list-style-type: none">• resolver problemas de optimización no lineal.
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Taha2004], [Hillier2006]	

Unidad 2: Toma de decisiones bajo incertidumbre (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Teoría de la utilidad. • Árboles de decisión. • Toma de decisión con objetivos múltiples. 	<ul style="list-style-type: none"> • Presentar los fundamentos de la teoría de decisiones en condiciones de incertidumbre.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Wayne2005], [Hillier2002IO]	

Unidad 3: Teoría de juegos (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Juegos de suma cero. • Juegos de suma no constante para dos personas. • Introducción a la teoría de juegos para n personas. 	<ul style="list-style-type: none"> • Resolver problemas de toma de decisiones donde dos o más personas que deciden se enfrentan a un conflicto de intereses.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Taha2004], [Hillier2006]	

Unidad 4: Cadenas de Markov (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Procesos estocásticos. • Cadenas de Markov. • Clasificación de los estados de una cadena de Markov. • Propiedades a largo de las cadenas de Markov. 	<ul style="list-style-type: none"> • Dar elementos de toma de decisiones para variables aleatorias que varían el tiempo.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Wayne2005], [Hillier2002IO]	

Unidad 5: Teoría de colas (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Estructura básicas de los modelos de colas. • Ejemplos de sistemas de colas reales. • Proceso de nacimiento y muerte. • Modelos de colas basados en el proceso de nacimiento y muerte. • Modelos de colas con distribuciones no exponenciales. • Modelos de colas con disciplina de prioridades. 	<ul style="list-style-type: none"> • Responder preguntas relacionadas a modelos de espera.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Taha2004], [Hillier2006]	

Unidad 6: Simulación (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tipos comunes de aplicaciones. • Generación de números aleatorios. • Estudio de casos. • Simulación con hojas de cálculos. • Técnicas de reducción de varianza. • Método regenerativo de análisis estadístico. 	<ul style="list-style-type: none"> • Poder hacer un análisis de riesgo ante situaciones de alta complejidad.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Taha2004], [Hillier2006]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

COM42009. Lenguaje II (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 4 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : COM42008. Lenguaje. (1^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Write justification for this course here ...

5. OBJETIVOS

- Write your first goal here.
- Write your second goal here.
- Just in case you need more goals write them here

6. COMPETENCIAS

CG1) Comunicación integral. (Evaluar)

7. TEMAS

Unidad 1: title for the unit goes here (0 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Topic1• Topic2• Topic3	<ul style="list-style-type: none">• Learning outcome1 [Levelforthislearningoutcome].• i.e Apply computing in complex problems [Usar].• Create a search engine [Evaluar].• Study data structures [Familiarizarse].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Bibitem1], [Bibitem2]	

Unidad 2: another unit goes here (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Topic1 	<ul style="list-style-type: none"> • Learning outcome xyz [Levelforthislearningoutcome].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Bibitem3], [Bibitem1]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

ENG42005. Inglés II (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : ENG42003. English I. (1^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

The learning of the English language in higher level students has become a necessary element for their academic, personal and professional development. Most of the academic literature from the various fields of knowledge is written in English, which ensures that people with a command of the language can always be up-to-date.

Likewise, knowledge of this language allows broader social and cultural perspectives. In this sense, for effective language learning, the development of the four (4) skills is necessary; listen, speak, read and write considering the guidelines of the Common European Framework of Reference for Languages - CEFR.

5. OBJETIVOS

- Comunicarse e intercambiar información de manera limitada en situaciones predecibles de forma general.
- Alcanzar el nivel B1 según el MCERL.
- Manejar terminología propia del área de estudios.

6. COMPETENCIAS

CG2) Comunicación bilingüe. (Evaluar)

7. TEMAS

Unidad 1: What time is the concert? (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Present simple – verb to be• Prepositions of place and time, including in/on/at• Connector (addition – contrast)	<ul style="list-style-type: none">• Talk about time and dates
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [De 02]	

Unidad 2: My day (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Present simple – action verbs. • Adverbs of frequency. • Questions (what / where / when / who– Present Simple). • Connector (addition – contrast). 	<ul style="list-style-type: none"> • Discuss one’s daily routine and habits • Express frequency
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 3: My favorite food (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • There is/are. • Common uncountable nouns. • Articles – with countable and uncountable nouns. • Verb + ing: like/hate/love. • Connector (addition – contrast). 	<ul style="list-style-type: none"> • Express likes and dislikes • Identify food items and drinks
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 4: Let’s cook (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • There is/are. • How much/how many and very. • Articles – with countable and uncountable nouns. • Connector (addition – contrast). 	<ul style="list-style-type: none"> • Give instructions for a recipe. • Talk about quantity.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 5: My childhood (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Past simple (negative). • Questions (what / where / when / who). • Connector (addition – contrast). 	<ul style="list-style-type: none"> • Talk about one’s childhood. • Describe a past event.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 6: Can you speak English? (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Design by the teacher. • Connector (addition – contrast). 	<ul style="list-style-type: none"> • Get familiar with technical expressions of the field.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[De 02] Consejo De Europa. “Marco común europeo de referencia para las lenguas”. In: *Strasburgo: Consejo de Europa, Ministerio de Educación, Cultura y Deporte/Instituto Cervantes* (2002).



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52094. Desarrollo Basado en Plataformas (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	1 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	96 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52097. Programación Orientada a Objetos I. (2 ^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El mundo ha cambiado debido al uso de la web y tecnologías relacionadas, el acceso rápido, oportuno y personalizado de la información, a través de la tecnología web, ubicuo y pervasiva; han cambiado la forma de ¿cómo hacemos las cosas?, ¿cómo pensamos? y ¿cómo la industria se desarrolla?.

Las tecnologías web, ubicuo y pervasivo se basan en el desarrollo de servicios web, aplicaciones web y aplicaciones móviles, las cuales son necesarias entender la arquitectura, el diseño, y la implementación de servicios web, aplicaciones web y aplicaciones móviles.

5. OBJETIVOS

- Que el alumno sea capaz de diseño e implementación de servicios, aplicaciones web utilizando herramientas y lenguajes como HTML, CSS, JavaScript (incluyendo AJAX), back-end scripting y una base de datos, a un nivel intermedio.
- Que el alumno sea capaz de desarrollar aplicaciones móviles, administrar servidores web en sistemas basados en UNIX y aplicar técnicas de seguridad en la web a un nivel intermedio.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Familiarizarse**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Familiarizarse**)
- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (**Familiarizarse**)

7. TEMAS

Unidad 1: Introducción (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión general de plataformas (ejemplo, Web, Mobil, Juegos, Industrial) • Programación a través de APIs específicos. • Visión general de lenguajes de plataforma (ejemplo, Objective C, HTML5) • Programación bajo restricciones de plataforma. 	<ul style="list-style-type: none"> • Describir cómo el desarrollo basado en plataforma difiere de la programación de propósito general [Familiarizarse] • Listar las características de lenguajes de plataforma [Familiarizarse] • Escribir y ejecutar un programa simple basado en plataforma [Familiarizarse] • Listar las ventajas y desventajas de la programación con restricciones de plataforma [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Gro09], [ADC13], [TC15]	

Unidad 2: Plataformas web (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Lenguajes de programación web (e.g., HTML5, Javascript, PHP, CSS) • Restricciones de las plataformas web: Client-Server, Stateless-Stateful, Caché, Uniform Interface, Layered System, Code on Demand, ReST. • Restricción de plataformas web. • Software como servicio. • Estándares web. 	<ul style="list-style-type: none"> • Diseñar e implementar una aplicación web sencilla [Familiarizarse] • Describir las limitaciones que la web pone a los desarrolladores [Familiarizarse] • Comparar y contrastar la programación web con la programación de propósito general [Familiarizarse] • Describir las diferencias entre software como un servicio y productos de software tradicionales [Familiarizarse] • Discutir cómo los estándares de web impactan el desarrollo de software [Familiarizarse] • Revisar una aplicación web existente con un estándar web actual [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Fie00]	

Unidad 3: Desarrollo de servicios y aplicaciones web (25 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Describir, identificar y depurar problemas relacionados con el desarrollo de aplicaciones web.• Diseño y desarrollo de aplicaciones web interactivas usando HTML5 y Python.• Utilice MySQL para la gestión de datos y manipular MySQL con Python.• Diseño y desarrollo de aplicaciones web asíncronos utilizando técnicas Ajax.• Uso del lado del cliente dinámico lenguaje de script Javascript y del lado del servidor lenguaje de scripting python con Ajax.• Aplicar las tecnologías XML / JSON para la gestión de datos.• Utilizar los servicios, APIs Web, Ajax y aplicar los patrones de diseño para el desarrollo de aplicaciones web.	<ul style="list-style-type: none">• Del lado del servidor lenguaje de scripting python: variables, tipos de datos, operaciones, cadenas, funciones, sentencias de control, matrices, archivos y el acceso a directorios, mantener el estado. [Usar]• Enfoque de programación web usando python incrustado. [Usar]• El acceso y la manipulación de MySQL. [Usar]• El enfoque de desarrollo de aplicaciones web Ajax. [Usar]• DOM y CSS utilizan en JavaScript. [Usar]• Tecnologías de actualización de contenido asíncrono. [Usar]• Objetos XMLHttpRequest utilizar para comunicarse entre clientes y servidores. [Usar]• XML y JSON. [Usar]• XSLT y XPath como mecanismos para transformar documentos XML. [Usar]• Servicios web y APIs (especialmente Google Maps). [Usar]• Marcos Ajax para el desarrollo de aplicaciones web contemporánea. [Usar]• Los patrones de diseño utilizados en aplicaciones web. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [FR11]	

Unidad 4: Plataformas móviles (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Lenguajes de Programación para Móviles. • Principios de diseño: Segregación de Interfaces, Responsabilidad Única, Separación de Responsabilidades, Inversión de Dependencias. • Desafíos con movilidad y comunicación inalámbrica. • Aplicaciones Location-aware. • Rendimiento / Compensación de Potencia. • Restricciones de las Plataformas Móviles. • Tecnologías Emergentes. 	<ul style="list-style-type: none"> • Diseñar e implementar una aplicación móvil para una plataforma móvil dada [Familiarizarse] • Discutir las limitaciones que las plataformas móviles ponen a los desarrolladores [Familiarizarse] • Discutir los principios de diseño que guían la construcción de aplicaciones móviles [Familiarizarse] • Discutir el rendimiento vs pérdida de potencia [Familiarizarse] • Compare y contraste la programación móvil con la programación de propósito general [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Mar17]	

Unidad 5: Aplicaciones Móviles para dispositivos Android (25 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • The Android Platform • The Android Development Environment • Application Fundamentals • The Activity Class • The Intent Class • Permissions • The Fragment Class • User Interface Classes • User Notifications • The BroadcastReceiver Class • Threads, AsyncTask & Handlers • Alarms • Networking (http class) • Multi-touch & Gestures • Sensors • Location & Maps 	<ul style="list-style-type: none"> • Los estudiantes identifican software necesario y lo instalan en sus ordenadores personales. Los estudiantes realizan varias tareas para familiarizarse con la plataforma Android y Ambiente para el Desarrollo. [Usar] • Los estudiantes construyen aplicaciones que trazan los métodos de devolución de llamada de ciclo de vida emitidas por la plataforma Android y que demuestran el comportamiento de Android cuando los cambios de configuración de dispositivos (por ejemplo, cuando el dispositivo se mueve de vertical a horizontal y viceversa). [Usar] • Los estudiantes construyen aplicaciones que requieren iniciar múltiples actividades a través de ambos métodos estándar y personalizados. [Usar] • Los estudiantes construyen aplicaciones que requieren permisos estándar y personalizados. [Usar] • Los estudiantes construyen una aplicación que utiliza una única base de código, sino que crea diferentes interfaces de usuario dependiendo del tamaño de la pantalla de un dispositivo. [Usar] • Los estudiantes construyen un gestor de listas de tareas pendientes utilizando los elementos de la interfaz de usuario discutidos en clase. La aplicación permite a los usuarios crear nuevos elementos y para mostrarlos en un ListView. [Usar] • Los estudiantes construyen una aplicación que utiliza la información de ubicación para recoger latitud, longitud de los lugares que visitan. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [ADC13], [TC15]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [ADC13] J. Annuzzi, L. Darcey, and S. Conder. *Introduction to Android Application Development: Android Essentials*. Developer's Library. Pearson Education, 2013. ISBN: 9780133477337.
- [Fie00] Roy Thomas Fielding. "Fielding dissertation: Chapter 5: Representational state transfer (rest)". In: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (2000).
- [FR11] Eric Freeman and Elisabeth Robson. *Head first HTML5 programming: building web apps with JavaScript*. "O'Reilly Media, Inc.", 2011.
- [Gro09] R. Grove. *Web Based Application Development*. Jones & Bartlett Learning, 2009. ISBN: 9780763759407.
- [Mar17] Robert C Martin. *Clean architecture: a craftsman's guide to software structure and design*. Prentice Hall Press, 2017.
- [TC15] Trish and Richard Cornez. *Android Programming Concepts*. Jones and Bartlett Publishers, 2015. ISBN: 9781284070705.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52095. Programación Orientada a Objetos II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52097. Programación Orientada a Objetos I. (2 ^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este es el tercer curso en la secuencia de los cursos introductorios a la informática. En este curso se pretende cubrir los conceptos señalados por la Computing Curricula IEEE(c)-ACM 2001, bajo el enfoque funcional-first. El paradigma orientado a objetos nos permite combatir la complejidad haciendo modelos a partir de abstracciones de los elementos del problema y utilizando técnicas como encapsulamiento, modularidad, polimorfismo y herencia. El dominio de estos temas permitirá que los participantes puedan dar soluciones computacionales a problemas de diseño sencillos del mundo real.

5. OBJETIVOS

- Introducir al alumno a los fundamentos del paradigma de orientación a objetos, permitiendo asimilar los conceptos necesarios para desarrollar un sistema de información

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Familiarizarse**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Familiarizarse**)

7. TEMAS

Unidad 1: Conceptos Fundamentales de Programación (5 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Sintaxis y semántica básica de un lenguaje de alto nivel.• Variables y tipos de datos primitivos (ej., numeros, caracteres, booleanos)• Expresiones y asignaciones.• Operaciones básicas I/O incluyendo archivos I/O.• Estructuras de control condicional e iterativas.• Paso de funciones y parámetros.• Concepto de recursividad.	<ul style="list-style-type: none">• Analiza y explica el comportamiento de programas simples que involucran estructuras fundamentales de programación variables, expresiones, asignaciones, E/S, estructuras de control, funciones, paso de parámetros, y recursividad [Usar]• Identifica y describe el uso de tipos de datos primitivos [Usar]• Escribe programas que usan tipos de datos primitivos [Usar]• Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar]• Diseña, implementa, prueba, y depura un programa que usa cada una de las siguientes estructuras de datos fundamentales: cálculos básicos, E/S simple, condicional estándar y estructuras iterativas, definición de funciones, y paso de parámetros [Usar]• Escribe un programa que usa E/S de archivos para brindar persistencia a través de ejecuciones múltiples [Usar]• Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Usar]• Describe el concepto de recursividad y da ejemplos de su uso [Usar]• Identifica el caso base y el caso general de un problema basado en recursividad [Usar]
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [stroustrup2013], [Van02], [LE13]	

Unidad 2: Programación orientada a objetos (7 horas)

Competencias esperadas:

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Diseño orientado a objetos: <ul style="list-style-type: none"> – Descomposición en objetos que almacenan estados y poseen comportamiento – Diseño basado en jerarquía de clases para modelamiento • Definición de las categorías, campos, métodos y constructores. • Las subclases, herencia y método de alteración temporal. • Asignación dinámica: definición de método de llamada. • Subtipificación: <ul style="list-style-type: none"> – Polimorfismo artículo Subtipo; upcasts implícitos en lenguajes con tipos. – Noción de reemplazo de comportamiento: los subtipos de actuar como supertipos. – Relación entre subtipos y la herencia. • Lenguajes orientados a objetos para la encapsulación: <ul style="list-style-type: none"> – privacidad y la visibilidad de miembros de la clase – Interfaces revelan único método de firmas – clases base abstractas • Uso de colección de clases, iteradores, y otros componentes de la librería estándar. 	<ul style="list-style-type: none"> • Diseñar e implementar una clase [Usar] • Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Usar] • Razonar correctamente sobre el flujo de control en un programa mediante el envío dinámico [Usar] • Comparar y contrastar (1) el enfoque proceduracional/funcional- definiendo una función por cada operación con el uso de la función proporcionando un caso por cada variación de dato - y (2) el enfoque orientado a objetos - definiendo una clase por cada variación de dato con la definición de la clase proporcionando un método por cada operación. Entender ambos enfoques como una definición de variaciones y operaciones de una matriz [Usar] • Explicar la relación entre la herencia orientada a objetos (código compartido y <i>overriding</i>) y subtipificación (la idea de un subtipo es ser utilizable en un contexto en el que espera al supertipo) [Usar] • Usar mecanismos de encapsulación orientada a objetos, tal como interfaces y miembros privados [Usar] • Definir y usar iteradores y otras operaciones sobre agregaciones, incluyendo operaciones que tienen funciones como argumentos, en múltiples lenguajes de programación, seleccionar la forma más natural por cada lenguaje [Usar]
<p>Aprendizaje autónomo</p> <ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
<p>Lecturas : [Str13]</p>	

Unidad 3: Algoritmos y Diseño (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Conceptos y propiedades de los algoritmos <ul style="list-style-type: none"> – Comparación informal de la eficiencia de los algoritmos (ej., conteo de operaciones) • Rol de los algoritmos en el proceso de solución de problemas • Estrategias de solución de problemas <ul style="list-style-type: none"> – Funciones matemáticas iterativas y recursivas – Recorrido iterativo y recursivo en estructura de datos – Estrategias Divide y Conquistar • Conceptos y principios fundamentales de diseño <ul style="list-style-type: none"> – Abstracción – Descomposición de Program – Encapsulamiento y camuflaje de información – Separación de comportamiento y aplicación 	<ul style="list-style-type: none"> • Discute la importancia de los algoritmos en el proceso de solución de un problema [Usar] • Discute como un problema puede ser resuelto por múltiples algoritmos, cada uno con propiedades diferentes [Usar] • Crea algoritmos para resolver problemas simples [Usar] • Usa un lenguaje de programación para implementar, probar, y depurar algoritmos para resolver problemas simples [Usar] • Implementa, prueba, y depura funciones recursivas simples y sus procedimientos [Usar] • Determina si una solución iterativa o recursiva es la más apropiada para un problema [Usar] • Implementa un algoritmo de divide y vencerás para resolver un problema [Usar] • Aplica técnicas de descomposición para dividir un programa en partes más pequeñas [Usar] • Identifica los componentes de datos y el comportamiento de múltiples tipos de datos abstractos [Usar] • Implementa un tipo de dato abstracto coherente, con la menor pérdida de acoplamiento entre componentes y comportamientos [Usar] • Identifica las fortalezas y las debilidades relativas entre múltiples diseños e implementaciones de un problema [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [stroustrup2013], [Weert16], [LE13]	

Unidad 4: Análisis Básico (3 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Diferencias entre el mejor, el esperado y el peor caso de un algoritmo. • Análisis asintótico de complejidad de cotas superior y esperada. • Definición formal de la Notación Big O. • Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial. • Medidas empíricas de desempeño. • Compensación entre espacio y tiempo en los algoritmos. • Uso de la notación Big O. • Notación Little o, Big omega y Big theta. • Relaciones recurrentes. • Análisis de algoritmos iterativos y recursivos. • Teorema Maestro y Árboles Recursivos. 	<ul style="list-style-type: none"> • Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Usar] • En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Usar] • Determine informalmente el tiempo y el espacio de complejidad de diferentes algoritmos [Usar] • Indique la definición formal de Big O [Usar] • Lista y contraste de clases estándares de complejidad [Usar] • Realizar estudios empíricos para validar una hipótesis sobre runtime stemming desde un análisis matemático Ejecute algoritmos con entrada de varios tamaños y compare el desempeño [Usar] • Da ejemplos que ilustran las compensaciones entre espacio y tiempo que se dan en los algoritmos [Usar] • Use la notación formal de la Big O para dar límites superiores asintóticos en la complejidad de tiempo y espacio de los algoritmos [Usar] • Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Usar] • Explicar el uso de la notación theta grande, omega grande y o pequeña para describir la cantidad de trabajo hecho por un algoritmo [Usar] • Usar relaciones recurrentes para determinar el tiempo de complejidad de algoritmos recursivamente definidos [Usar] • Resuelve relaciones de recurrencia básicas, por ejemplo. usando alguna forma del Teorema Maestro [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Str13]	

Unidad 5: Sistemas de tipos básicos (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tipos como conjunto de valores junto con un conjunto de operaciones. <ul style="list-style-type: none"> – Tipos primitivos (p.e. números, booleanos) – Composición de tipos contruídos de otros tipos (p.e., registros, uniones, arreglos, listas, funciones, referencias) • Asociación de tipos de variables, argumentos, resultados y campos. • Tipo de seguridad y los errores causados por el uso de valores de manera incompatible dadas sus tipos previstos. • Metas y limitaciones de tipos estáticos <ul style="list-style-type: none"> – Eliminación de algunas clases de errores sin ejecutar el programa – Indecisión significa que un análisis estatico puede aproximar el comportamiento de un programa • Tipos genéricos (polimorfismo paramétrico) <ul style="list-style-type: none"> – Definición – Uso de librerías genéricas tales como colecciones. – Comparación con polimorfismo ad-hoc y polimorfismo de subtipos • Beneficios complementarios de tipos estáticos y dinámicos: <ul style="list-style-type: none"> – Errores tempranos vs. errores tardíos/evitados. – Refuerzo invariante durante el desarrollo y mantenimiento del código vs. decisiones pospuestas de tipos durante la la creación de prototipos y permitir convenientemente la codificación flexible de patrones tales como colecciones heterogéneas. – Evitar el mal uso del código vs. permitir más reuso de código. – Detectar programas incompletos vs. permitir que programas incompletos se ejecuten 	<ul style="list-style-type: none"> • Tanto para tipo primitivo y un tipo compuesto, describir de manera informal los valores que tiene dicho tipo [Usar] • Para un lenguaje con sistema de tipos estático, describir las operaciones que están prohibidas de forma estática, como pasar el tipo incorrecto de valor a una función o método [Usar] • Describir ejemplos de errores de programa detectadas por un sistema de tipos [Usar] • Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar] • Dar un ejemplo de un programa que no verifique tipos en un lenguaje particular y sin embargo no tenga error cuando es ejecutado [Usar] • Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar] • Explicar como las reglas de tipificación definen el conjunto de operaciones que legales para un tipo [Usar] • Escribir las reglas de tipo que rigen el uso de un particular tipo compuesto [Usar] • Explicar por qué indecidibilidad requiere sistemas de tipo para conservadoramente aproximar el comportamiento de un programa [Usar] • Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar] • Discutir las diferencias entre, genéricos (<i>generics</i>), subtipo y sobrecarga [Usar] • Explicar múltiples beneficios y limitaciones de tipificación estática en escritura, mantenimiento y depuración de un software [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Str13]	

Unidad 6: Algoritmos y Estructuras de Datos fundamentales (3 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo. • Algoritmos de búsqueda secuencial y binaria. • Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción) • Algoritmos de ordenamiento con peor caso o caso promedio en $O(N \lg N)$ (Quicksort, Heapsort, Mergesort) • Tablas Hash, incluyendo estrategias para evitar y resolver colisiones. • Árboles de búsqueda binaria: <ul style="list-style-type: none"> – Operaciones comunes en árboles de búsqueda binaria como seleccionar el mínimo, máximo, insertar, eliminar, recorrido en árboles. • Grafos y algoritmos en grafos: <ul style="list-style-type: none"> – Representación de grafos (ej., lista de adyacencia, matriz de adyacencia) – Recorrido en profundidad y amplitud • Montículos (Heaps) • Grafos y algoritmos en grafos: <ul style="list-style-type: none"> – Problema de corte máximo y mínimo – Búsqueda local • Búsqueda de patrones y algoritmos de cadenas/texto (ej. búsqueda de subcadena, búsqueda de expresiones regulares, algoritmos de subsecuencia común más larga) 	<ul style="list-style-type: none"> • Implementar algoritmos numéricos básicos [Usar] • Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Usar] • Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Usar] • Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Usar] • Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Usar] • Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Usar] • Explicar como el balanceamiento del arbol afecta la eficiencia de varias operaciones de un arbol de búsqueda binaria [Usar] • Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Usar] • Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto en específico [Usar] • Describir la propiedad del heap y el uso de heaps como una implementación de colas de prioridad [Usar] • Resolver problemas usando algoritmos de grafos, incluyendo camino más corto de una sola fuente y camino más corto de todos los pares, y como mínimo un algoritmo de arbol de expansion minima [Usar] • Trazar y/o implementar un algoritmo de comparación de string [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [stroustrup2013], [PA18]	

Unidad 7: Programación reactiva y dirigida por eventos (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Eventos y controladores de eventos. • Usos canónicos como interfaces gráficas de usuario, dispositivos móviles, robots, servidores. • Uso de frameworks reactivos. <ul style="list-style-type: none"> – Definición de controladores/oyentes (handles/listeners) de eventos. – Bucle principal de eventos no controlado por el escritor controlador de eventos (event-handler-writer) • Eventos y eventos del programa generados externamente generada. • La separación de modelo, vista y controlador. 	<ul style="list-style-type: none"> • Escribir manejadores de eventos para su uso en sistemas reactivos tales como GUIs [Usar] • Explicar porque el estilo de programación manejada por eventos es natural en dominios donde el programa reacciona a eventos externos [Usar] • Describir un sistema interactivo en términos de un modelo, una vista y un controlador [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [stroustrup2013], [Wil11]	

Unidad 8: Árboles y Grafos (7 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Árboles. <ul style="list-style-type: none"> – Propiedades – Estrategias de recorrido • Grafos no dirigidos • Grafos dirigidos • Grafos ponderados • Árboles de expansión/bosques. • Isomorfismo en grafos. 	<ul style="list-style-type: none"> • Ilustrar mediante ejemplos la terminología básica de teoría de grafos, y de alguna de las propiedades y casos especiales de cada tipo de grafos/árboles [Usar] • Demostrar diversos métodos de recorrer árboles y grafos, incluyendo recorridos pre, post e inorden de árboles [Usar] • Modelar una variedad de problemas del mundo real en ciencia de la computación usando formas adecuadas de grafos y árboles, como son la representación de una topología de red o la organización jerárquica de un sistema de archivos [Usar] • Demostrar como los conceptos de grafos y árboles aparecen en estructuras de datos, algoritmos, técnicas de prueba (inducción estructurada), y conteos [Usar] • Explicar como construir un árbol de expansión de un grafo [Usar] • Determinar si dos grafos son isomorfos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Nak13]	

Unidad 9: Diseño de Software (6 horas)**Competencias esperadas:****Temas****Objetivos de Aprendizaje**

- Principios de diseño del sistema: niveles de abstracción (diseño arquitectónico y el diseño detallado), separación de intereses, ocultamiento de información, de acoplamiento y de cohesión, de reutilización de estructuras estándar.
- Diseño de paradigmas tales como diseño estructurado (descomposición funcional de arriba hacia abajo), el análisis orientado a objetos y diseño, orientado a eventos de diseño, diseño de nivel de componente, centrado datos estructurada, orientada a aspectos, orientado a la función, orientado al servicio.
- Modelos estructurales y de comportamiento de los diseños de software.
- Diseño de patrones.
- Relaciones entre los requisitos y diseños: La transformación de modelos, el diseño de los contratos, invariantes.
- Conceptos de arquitectura de software y arquitecturas estándar (por ejemplo, cliente-servidor, n-capas, transforman centrados, tubos y filtros).
- El uso de componentes de diseño: selección de componentes, diseño, adaptación y componentes de ensamblaje, componentes y patrones, componentes y objetos (por ejemplo, construir una GUI usando un standard widget set)
- Diseños de refactorización utilizando patrones de diseño
- Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos.
- Medición y análisis de la calidad de un diseño.
- Compensaciones entre diferentes aspectos de la calidad.
- Aplicaciones en frameworks.
- Middleware: El paradigma de la orientación a objetos con middleware, requerimientos para correr y clasificar objetos, monitores de procesamiento de transacciones y el sistema de flujo de trabajo.
- Principales diseños de seguridad y codificación (cross-reference IAS/Principles of secure design).
 - Principio de privilegios mínimos
 - Principio de falla segura por defecto
 - Principio de aceptabilidad psicológica

- Formular los principios de diseño, incluyendo la separación de problemas, ocultación de información, acoplamiento y cohesión, y la encapsulación [Usar]
- Usar un paradigma de diseño para diseñar un sistema de software básico y explicar cómo los principios de diseño del sistema se han aplicado en este diseño [Usar]
- Construir modelos del diseño de un sistema de software simple los cuales son apropiado para el paradigma utilizado para diseñarlo [Usar]
- En el contexto de un paradigma de diseño simple, describir uno o más patrones de diseño que podrían ser aplicables al diseño de un sistema de software simple [Usar]
- Para un sistema simple adecuado para una situación dada, discutir y seleccionar un paradigma de diseño apropiado [Usar]
- Crear modelos apropiados para la estructura y el comportamiento de los productos de software desde la especificaciones de requisitos [Usar]
- Explicar las relaciones entre los requisitos para un producto de software y su diseño, utilizando los modelos apropiados [Usar]
- Para el diseño de un sistema de software simple dentro del contexto de un único paradigma de diseño, describir la arquitectura de software de ese sistema [Usar]
- Dado un diseño de alto nivel, identificar la arquitectura de software mediante la diferenciación entre las arquitecturas comunes de software, tales como 3 capas (*3-tier*), *pipe-and-filter*, y cliente-servidor [Usar]
- Investigar el impacto de la selección arquitecturas de software en el diseño de un sistema simple [Usar]
- Aplicar ejemplos simples de patrones en un diseño de software [Usar]
- Describir una manera de refactorar y discutir cuando esto debe ser aplicado [Usar]
- Seleccionar componentes adecuados para el uso en un diseño de un producto de software [Usar]
- Explicar cómo los componentes deben ser adaptados para ser usados en el diseño de un producto de software [Usar]
- Diseñar un contrato para un típico componente de software pequeño para el uso de un dado sistema [Usar]
- Discutir y seleccionar la arquitectura de software adecuada para un sistema de software simple para un dado escenario [Usar]

Unidad 10: Ingeniería de Requisitos (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Al describir los requisitos funcionales utilizando, por ejemplo, los casos de uso o historias de los usuarios. • Propiedades de requisitos, incluyendo la consistencia, validez, integridad y viabilidad. • Requisitos de software elicitación. • Descripción de datos del sistema utilizando, por ejemplo, los diagramas de clases o diagramas entidad-relación. • Requisitos no funcionales y su relación con la calidad del software. • Evaluación y uso de especificaciones de requisitos. • Requisitos de las técnicas de modelado de análisis. • La aceptabilidad de las consideraciones de certeza/incertidumbre sobre el comportamiento del software/sistema. • Prototipos. • Conceptos básicos de la especificación formal de requisitos. • Especificación de requisitos. • Validación de requisitos. • Rastreo de requisitos. 	<ul style="list-style-type: none"> • Enumerar los componentes clave de un caso de uso o una descripción similar de algún comportamiento que es requerido para un sistema [Usar] • Describir cómo el proceso de ingeniería de requisitos apoya la obtención y validación de los requisitos de comportamiento [Usar] • Interpretar un modelo de requisitos dada por un sistema de software simple [Usar] • Describir los retos fundamentales y técnicas comunes que se utilizan para la obtención de requisitos [Usar] • Enumerar los componentes clave de un modelo de datos (por ejemplo, diagramas de clases o diagramas ER) [Usar] • Identificar los requisitos funcionales y no funcionales en una especificación de requisitos dada por un sistema de software [Usar] • Realizar una revisión de un conjunto de requisitos de software para determinar la calidad de los requisitos con respecto a las características de los buenos requisitos [Usar] • Aplicar elementos clave y métodos comunes para la obtención y el análisis para producir un conjunto de requisitos de software para un sistema de software de tamaño medio [Usar] • Comparar los métodos ágiles y el dirigido por planes para la especificación y validación de requisitos y describir los beneficios y riesgos asociados con cada uno [Usar] • Usar un método común, no formal para modelar y especificar los requisitos para un sistema de software de tamaño medio [Usar] • Traducir al lenguaje natural una especificación de requisitos de software (por ejemplo, un contrato de componentes de software) escrito en un lenguaje de especificación formal [Usar] • Crear un prototipo de un sistema de software para reducir el riesgo en los requisitos [Usar] • Diferenciar entre el rastreo (<i>tracing</i>) hacia adelante y hacia atrás y explicar su papel en el proceso de validación de requisitos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Str13]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [LE13] Stanley B. Lippman and Barbara E.Moo. *C++ Primer*. 5th. O'Reilly, 2013. ISBN: 9780133053043.
- [Nak13] S. Nakariakov. *The Boost C++ Libraries: Generic Programming*. CreateSpace Independent Publishing Platform, 2013.
- [PA18] Praseed Pai and Peter Abraham. *C++ Reactive Programming*. 1st. Packt, 2018.
- [Str13] B Stroustrup. *The C++ Programming Language, 4th edition*. Addison-Wesley, 2013.
- [Van02] David Vandervoorde. *C++ Templates: The Complete Guide*. 1st. Addison-Wesley, 2002. ISBN: 978-0134448237.
- [Wil11] Anthony Williams. *C++ Concurrency in Action*. 1st. Manning, 2011.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

MAC41002. Cálculo de varias Variables (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : MAC41001. Cálculo de una Variable. (2^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso introduce los primeros conceptos del álgebra lineal, así como los métodos numéricos con un énfasis en la resolución de problemas con el paquete de software libre de código abierto Scilab. La teoría matemática se limita a los fundamentos, mientras que la aplicación efectiva para la resolución de problemas es privilegiada. En cada tópico, se enseña unos cuantos métodos de relevancia para la ingeniería. Los conocimientos sobre estos métodos prepara a los estudiantes para la búsqueda de alternativas más avanzadas, si se lo requiere.

5. OBJETIVOS

- Capacidad para aplicar los conocimientos sobre Matemáticas.
- Capacidad para aplicar los conocimientos sobre Ingeniería .
- Capacidad para aplicar los conocimientos, técnicas, habilidades y herramientas modernas de la ingeniería moderna para la práctica de la ingeniería.

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Introducción (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Importancia del álgebra lineal y métodos numéricos. Ejemplos.	<ul style="list-style-type: none">• Ser capaz de entender los conceptos básicos y la importancia de Álgebra Lineal y Métodos Numéricos.
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [AR14], [CC15]	

Unidad 2: Álgebra lineal (14 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Álgebra matricial elemental y determinantes.• Espacio nulo y soluciones exactas de sistemas de ecuaciones lineales $Ax = b$:<ul style="list-style-type: none">– Sistemas tridiagonal y triangular y eliminación gaussiana con y sin giro.– Factorización LU y algoritmo Crout.• Conceptos básicos sobre valores propios y vectores propios<ul style="list-style-type: none">– Polinomios característicos.– Multiplicaciones algebraicas y geométricas.• Estimación de mínimos cuadrados.• Transformaciones lineales.	<ul style="list-style-type: none">• Comprender los conceptos básicos del Álgebra Lineal.• Resolver problemas de transformaciones lineales.
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [AR14], [CC15]	

Unidad 3: Métodos Numéricos (22 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Fundamentos de soluciones de sistemas de ecuaciones lineales $Ax = b$: métodos de Jacobi y Gauss Seidel • Aplicación de factorizaciones de matriz a la solución de sistemas lineales (descomposición de valores singulares, QR, Cholesky) Cálculo numérico del espacio nulo, rango y número de condición • Conclusión de la raíz: <ul style="list-style-type: none"> – Bisección. – Iteración de punto fijo. – Métodos de Newton-Raphson. • Fundamentos de la interpolación: <ul style="list-style-type: none"> – Interpolaciones polinomiales de Newton y Lagrange. – Interpolación de spline. • Fundamentos de la diferenciación numérica y la aproximación de Taylor. • Aspectos básicos de la integración numérica: <ul style="list-style-type: none"> – Trapecio, punto medio y regla de Simpson – Cuadratura gaussiana • Conceptos básicos sobre las soluciones numéricas a las EDOs: <ul style="list-style-type: none"> – Diferencias finitas; Métodos de Euler y Runge-Kutta – Convertir ODEs de orden superior en un sistema de ODEs de bajo orden. – Métodos de Runge-Kutta para sistemas de ecuaciones – Método simple.XYZ • Breve introducción a las técnicas de optimización: visión general sobre la programación lineal, sistemas lineales acotados, programación cuadrática, descenso gradiente. 	<ul style="list-style-type: none"> • Comprender los conceptos básicos de los métodos numéricos. • Aplicar los métodos más frecuentes para la resolución de problemas matemáticos. • Implementación y aplicación de algoritmos numéricos para la solución de problemas matemáticos utilizando el paquete computacional Scilab open-source. • Aplicación de Scilab para la solución de problemas matemáticos y para trazar graficas.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [AR14], [CC15]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje

activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[AR14] H. Anton and C. Rorres. *Elementary Linear Algebra, Applications Version*. 11th. Wiley, 2014.

[CC15] S.C. Chapra and R.P. Canale. *Numerical Methods for Engineers*, 7th. Vol. 1. McGraw-Hill, 2015.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

ENG42007. English III (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : ENG42005. Inglés II. (2^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El aprendizaje del idioma inglés en los estudiantes de nivel superior se ha convertido en elemento necesario para su desarrollo académico, personal y profesional. La mayor cantidad de literatura académica de los diversos campos del saber es redactada en inglés, lo cual garantiza que las personas que cuenten con el dominio del idioma puedan siempre estar actualizados.

Asimismo, el conocimiento de este idioma permite tener perspectivas sociales y culturales más amplias. En ese sentido, para un efectivo aprendizaje de la lengua es necesario el desarrollo de las cuatro (4) habilidades: escuchar, hablar, leer y escribir considerando los lineamientos del Marco Común Europeo de Referencia de Lengua - MCERL.

5. OBJETIVOS

- Comunicarse e intercambiar información de manera limitada en situaciones predecibles de forma general.
- Alcanzar el nivel B1 según el MCERL.
- Manejar terminología propia del área de estudios.

6. COMPETENCIAS

CG2) Comunicación bilingüe. (Evaluar)

7. TEMAS

Unidad 1: Be polite (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Imperatives (+/-).• Modals – can/could – have to – should.• Connector (effect).	<ul style="list-style-type: none">• Make polite requests.
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [De 02]	

Unidad 2: My neighborhood (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Adverbial phrases (place – word order). • Prepositional phrases (place, time and movement). • Connector (effect). 	<ul style="list-style-type: none"> • Give directions. • Describe places and location.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 3: The best in town! (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Comparative. • Definite article. • Intensifiers (too + enough). • Superlative. • Connector (effect). 	<ul style="list-style-type: none"> • Compare places.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 4: What can you do? (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Modal verb “can”. • Action verbs. • Phrasal verbs – common. • Connector (effect). 	<ul style="list-style-type: none"> • Talk about abilities
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 5: What a day! (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Present simple – action verbs. • Phrasal verbs – common. • Adverbial and prepositional phrases (time, place and frequency). • Connector (effect). 	<ul style="list-style-type: none"> • Discuss one's daily routine and habits. • Express frequency on everyday activities.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 6: Can you speak English? (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Design by the teacher. • Connector (effect). 	<ul style="list-style-type: none"> • Get familiar with technical expressions of the field.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[De 02] Consejo De Europa. “Marco común europeo de referencia para las lenguas”. In: *Strasburgo: Consejo de Europa, Ministerio de Educación, Cultura y Deporte/Instituto Cervantes* (2002).



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

EGI61009. Arquitectura de Computadores (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	<ul style="list-style-type: none">• SFW52096. Estructuras Discretas II. (2^{do} Sem)• ENG42005. Inglés II. (2^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Un profesional en Ciencia de la Computación debe tener un conocimiento sólido de la organización y los principios de diseño de diversos sistemas de computación, al comprender las limitaciones de los sistemas modernos serán capaces de proponer nuevos paradigmas en la próxima generación. Este curso enseña los fundamentos y principios de la arquitectura de computadoras. Esta clase incluye diseño de lógica digital, conceptos básicos de arquitectura de computadora y diseño de procesador (*Instruction Set Architecture*, microarquitectura, ejecución fuera de orden, predicción de *branches*), paradigmas de ejecución (superescalar, flujo de datos, VLIW, SIMD, GPU, sistólica, multiproceso) y organización del sistema de memoria.

5. OBJETIVOS

- Proporcionar un primer enfoque en Arquitectura de Computadoras.
- Estudiar el diseño y la evolución de las arquitecturas de computador, que llevaron a las implementaciones de los sistemas modernos.
- Proporcionar un estudio profundo del hardware y su relación con la ejecución del software.
- Implementar un microprocesador simple usando el lenguaje Verilog.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Familiarizarse**)

7. TEMAS

Unidad 1: Lógica digital y sistemas digitales (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Revisión e historia de la Arquitectura de Computadores. • Lógica combinacional y secuencial/<i>field programmable gate arrays</i> como bloque fundamental de construcción lógico combinacional secuencial. • Modelos de representación(abstracción) • Herramientas de diseño asistidas por computadora que procesan hardware y representaciones arquitecturales. • Registrar transferencia notación / Hardware language descriptivo (Verilog/VHDL) • Restriccion física (Retrasos de Entrada, fan-in, fan-out, energía/potencia) 	<ul style="list-style-type: none"> • Describir el avance de la tecnología de dispositivos, desde los tubos de vacío hasta VLSI, desde las arquitecturas mainframe a las arquitecturas en escala warehouse [Familiarizarse] • Comprender que la tendencia de las arquitecturas modernas de computadores es hacia núcleos múltiples y que el paralelismo es inherente en todos los sistemas de hardware [Usar] • Explicar las implicancias de los límites de potencia para mejoras adicionales en el rendimiento de los procesadores y también en el aprovechamiento del paralelismo [Usar] • Relacionar las varias representaciones equivalentes de la funcionalidad de un computador, incluyendo expresiones y puertas lógicas, y ser capaces de utilizar expresiones matemáticas para describir las funciones de circuitos combinacionales y secuenciales sencillos [Familiarizarse] • Diseñar los componentes básicos de construcción de un computador: unidad aritmético lógica (a nivel de puertas lógicas), unidad central de procesamiento (a nivel de registros de transferencia), memoria (a nivel de registros de transferencia) [Usar] • Usar herramientas CAD para capturar, sistetizar, y simular bloques de construcción (como ALUs, registros, movimiento entre registros) de un computador simple [Familiarizarse] • Evaluar el comportamiento de un diagrama de tiempos y funcional de un procesador simple implementado a nivel de circuitos lógicos [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 2: Representación de datos a nivel máquina (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Bits, Bytes y Words. • Representación de datos numérica y bases numéricas. • Sistemas de punto flotante y punto fijo. • Representaciones con signo y complemento a 2. • Representación de información no numérica (códigos de caracteres, información gráfica) • Representación de registros y arreglos. 	<ul style="list-style-type: none"> • Explicar porqué en computación todo es datos, inclusive las instrucciones [Evaluar] • Explicar las razones de usar formatos alternativos para representar datos numéricos [Familiarizarse] • Describir cómo los enteros negativos se almacenan con representaciones de bit de signo y complemento a 2 [Usar] • Explicar cómo las representaciones de tamaño fijo afectan en la exactitud y la precisión [Usar] • Describir la representación interna de datos no numéricos como caracteres, cadenas, registros y arreglos [Usar] • Convertir datos numéricos de un formato a otro [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 3: Organización de la Máquina a Nivel Ensamblador (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Organización Básica de la Máquina de Von Neumann. • Unidad de Control. • <i>Instruction sets</i> y tipos (manipulación de información, control, I/O) • Assembler y Programación en Lenguaje de Máquina. • Formato de instrucciones. • Modos de direccionamiento. • Llamada a subrutinas y mecanismos de retorno. • I/O e Interrupciones. • Montículo (Heap) vs. Estático vs. Pila vs. Segmentos de código. 	<ul style="list-style-type: none"> • Explicar la organización de la maquina clásica de von Neumann y sus principales unidades funcionales [Familiarizarse] • Describir cómo se ejecuta una instrucción en una máquina de von Neumann con extensión para hebras, sincronización multiproceso y ejecución SIMD (máquina vectorial) [Familiarizarse] • Describir el paralelismo a nivel de instrucciones y sus peligros, y cómo es esto tratado en pipelines de proceso típicos [Familiarizarse] • Resumir cómo se representan las instrucciones, tanto a nivel de máquina bajo el contexto de un ensamblador simbólico [Familiarizarse] • Demostrar cómo se mapean los patrones de lenguajes de alto nivel en notaciones en lenguaje ensamblador o en código máquina [Usar] • Explicar los diferentes formatos de instrucciones, así como el direccionamiento por instrucción, y comparar formatos de tamaño fijo y variable [Usar] • Explicar como las llamadas a subrutinas son manejadas a nivel de ensamblador [Usar] • Explicar los conceptos básicos de interrupciones y operaciones de entrada y salida (I/O) [Familiarizarse] • Escribir segmentos de programa simples en lenguaje ensamblador [Usar] • Ilustrar cómo los bloques constructores fundamentales en lenguajes de alto nivel son implementados a nivel de lenguaje máquina [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 4: Organización funcional (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Implementación de <i>datapath</i>, incluyendo un <i>pipeline</i> de instrucciones, detección de <i>hazards</i> y la resolución. • Control de unidades: Microprogramada. • Instrucción (Pipelining) • Introducción al paralelismo al nivel de instrucción (PNI) 	<ul style="list-style-type: none"> • Comparar implementaciones alternativas de ruta de datos [Evaluar] • Discutir el concepto de puntos de control y la generación de señales de control usando implementaciones a nivel de circuito o microprogramadas [Familiarizarse] • Explicar el paralelismo a nivel de instrucciones básicas usando pipelining y los mayores riesgos que pueden ocurrir [Usar] • Diseñar e implementar un procesador completo, incluyendo ruta de datos y control [Usar] • Calcular la cantidad promedio de ciclos por instrucción de una implementación con procesador y sistema de memoria determinados [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 5: Organización y Arquitectura del Sistema de Memoria (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Sistemas de Almacenamiento y su Tecnología. • Jerarquía de Memoria: importancia de la localización temporal y espacial. • Organización y Operaciones de la Memoria Principal. • Latencia, ciclos de tiempo, ancho de banda e <i>inter-leading</i>. • Memorias caché (Mapeo de direcciones, Tamaño de bloques, Reemplazo y Políticas de almacenamiento) • Multiprocesador coherencia cache / Usando el sistema de memoria para las operaciones de sincronización de memoria / atómica inter-core. • Memoria virtual (tabla de página, TLB) • Manejo de Errores y confiabilidad. • Error de codificación, compresión de datos y la integridad de datos. 	<ul style="list-style-type: none"> • Identificar las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarizarse] • Explique el efecto de latencia de memoria en tiempo de ejecución [Familiarizarse] • Describir como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el atraso efectivo en la memoria [Usar] • Describir los principios de la administración de memoria [Usar] • Explique el funcionamiento de un sistema con gestión de memoria virtual [Usar] • Calcule el tiempo de acceso promedio a memoria bajo varias configuraciones de caché y memoria y para diversas combinaciones de instrucciones y referencias a datos [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 6: Interfaz y comunicación (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Fundamentos de I/O: Handshaking, Bbuffering, I/O programadas, interrupciones dirigidas de I/O. • Interrumpir estructuras: interrumpir reconocimiento, vectorizado y priorizado. • Almacenamiento externo, organización física y discos. • Buses: Protocolos de bus, arbitraje, acceso directo a memoria (DMA). • Introducción a Redes: comunicación de redes como otra capa de acceso remoto. • Soporte Multimedia. • Arquitecturas RAID. 	<ul style="list-style-type: none"> • Explicar como las interrupciones son aplicadas para implementar control de entrada-salida y transferencia de datos [Familiarizarse] • Identificar diversos tipos de buses en un sistema computacional [Familiarizarse] • Describir el acceso a datos desde una unidad de disco magnético [Usar] • Comparar organizaciones de red conocidas como organizaciones en bus/Ethernet, en anillo y organizaciones conmutadas versus ruteadas [Evaluar] • Identificar las interfaces entre capas necesarios para el acceso y presentación multimedia, desde la captura de la imagen en almacenamiento remoto, a través del transporte por una red de comunicaciones, hasta la puesta en la memoria local y la presentación final en una pantalla gráfica [Familiarizarse] • Describir las ventajas y limitaciones de las arquitecturas RAID [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 7: Multiprocesamiento y arquitecturas alternativas (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • <i>Power Law.</i> • Ejemplos de <i>sets</i> de instrucciones y arquitecturas SIMD y MIMD. • Redes de interconexión (Hypercube, Shuffle-exchange, Mesh, Crossbar) • Sistemas de memoria de multiprocesador compartido y consistencia de memoria. • Coherencia de cache multiprocesador. 	<ul style="list-style-type: none"> • Discutir el concepto de procesamiento paralelo mas allá del clásico modelo de von Neumann [Evaluar] • Describir diferentes arquitecturas paralelas como SIMD y MIMD [Familiarizarse] • Explicar el concepto de redes de interconexión y mostrar diferentes enfoques [Usar] • Discutir los principales cuidados en los sistemas de multiprocesamiento presentes con respecto a la gestión de memoria y describir como son tratados [Familiarizarse] • Describir las diferencias entre conectores electricos en paralelo backplane, interconexión memoria procesador y memoria remota via red, sus implicaciones para la latencia de acceso y el impacto en el rendimiento de un programa [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unidad 8: Mejoras de rendimiento (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Arquitectura superescalar. • Predicción de ramificación, Ejecución especulativa, Ejecución fuera de orden. • Prefetching. • Procesadores vectoriales y GPU's • Soporte de hardware para multiprocesamiento. • Escalabilidad. • Arquitecturas alternativas, como VLIW / EPIC y aceleradores y otros tipos de procesadores de propósito especial. 	<ul style="list-style-type: none"> • Describir las arquitecturas superescalares y sus ventajas [Familiarizarse] • Explicar el concepto de predicción de bifurcaciones y su utilidad [Usar] • Caracterizar los costos y beneficios de la precarga prefetching [Evaluar] • Explicar la ejecución especulativa e identifique las condiciones que la justifican [Evaluar] • Discutir las ventajas de rendimiento ofrecida en una arquitectura de multihebras junto con los factores que hacen difícil dar el máximo beneficio de estas [Evaluar] • Describir la importancia de la escalabilidad en el rendimiento [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[HH12] David Harris and Sarah Harris. *Digital Design and Computer Architecture*. 2nd. Morgan Kaufmann, 2012. ISBN: 978-0123944245.

- [HP06] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. 4th. San Mateo, CA: Morgan Kaufman, 2006.
- [JAs07] Peter J. Ashenden. *Digital Design (Verilog): An Embedded Systems Approach Using Verilog*. Morgan Kaufmann, 2007. ISBN: 978-0123695277.
- [Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers*. New York: Oxford Univ. Press, 2005. ISBN: ISBN 0-19-515455-X.
- [PCh06] Pong P. Chu. *RTL Hardware Design Using VHDL*. 1st. Wiley-Interscience, 2006.
- [PH04] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 3rd ed. San Mateo, CA: Morgan Kaufman, 2004.
- [PP05] Yale N. Patt and Sanjay J. Patel. *Introduction to Computing Systems*. 2nd. McGraw Hill, 2005.
- [Sta10] William Stalings. *Computer Organization and Architecture: Designing for Performance*. 8th. Upper Saddle River, NJ: Prentice Hall, 2010.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

GES51001. Administración para los negocios (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 4 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Write justification for this course here ...

5. OBJETIVOS

- Write your first goal here.
- Write your second goal here.
- Just in case you need more goals write them here

6. COMPETENCIAS

CG4) Gestiona recursos. (Evaluar)

7. TEMAS

Unidad 1: title for the unit goes here (0 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Topic1• Topic2• Topic3	<ul style="list-style-type: none">• Learning outcome1 [Levelforthislearningoutcome].• i.e Apply computing in complex problems [Usar].• Create a search engine [Evaluar].• Study data structures [Familiarizarse].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Bibitem1], [Bibitem2]	

Unidad 2: another unit goes here (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Topic1 	<ul style="list-style-type: none"> • Learning outcome xyz [Levelforthislearningoutcome].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Bibitem3], [Bibitem1]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52091. Gerenciamiento de Datos I (Obligatorio)

2. INFORMACIÓN GENERAL

- | | | |
|--------------------------|---|---|
| 2.1 Créditos | : | 2 |
| 2.2 Horas de teoría | : | - |
| 2.3 Horas de práctica | : | 2 (Semanal) |
| 2.4 Horas autónomas | : | 64 (horas) |
| 2.5 Duración del periodo | : | 16 semanas |
| 2.6 Condición | : | Obligatorio |
| 2.7 Modalidad | : | Presencial |
| 2.8 Prerrequisitos | : | <ul style="list-style-type: none">• SFW52097. Programación Orientada a Objetos I. (2^{do} Sem)• SFW52096. Estructuras Discretas II. (2^{do} Sem) |

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La gestión de la información (IM) juega un rol principal en casi todas las áreas donde los computadores son usados. Esta área incluye la captura, digitalización, representación, organización, transformación y presentación de información; algoritmos para mejorar la eficiencia y efectividad del acceso y actualización de información almacenada, modelamiento de datos y abstracción, y técnicas de almacenamiento de archivos físicos.

Este también abarca la seguridad de la información, privacidad, integridad y protección en un ambiente compartido. Los estudiantes necesitan ser capaces de desarrollar modelos de datos conceptuales y físicos, determinar que métodos de (IM) y técnicas son apropiados para un problema dado, y ser capaces de seleccionar e implementar una apropiada solución de IM que refleje todas las restricciones aplicables, incluyendo escalabilidad y usabilidad.

5. OBJETIVOS

- Que el alumno aprenda a representar información en una base de datos priorizando la eficiencia en la recuperación de la misma
- Que el alumno aprenda los conceptos fundamentales de gestión de bases de datos. Esto incluye aspectos de diseño de bases de datos, lenguajes de bases de datos y realización de bases de datos
- Discutir el modelo de bases de datos con base en el álgebra relacional, cálculo relacional y en el estudio de sentencias SQL.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Familiarizarse**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Familiarizarse**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Familiarizarse**)

7. TEMAS

Unidad 1: Sistemas de Bases de Datos (14 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Enfoque y Evolución de Sistemas de Bases de Datos. • Componentes del Sistema de Bases de Datos. • Diseño de las funciones principales de un DBMS. • Arquitectura de base de datos e independencia de datos. • Uso de un lenguaje de consulta declarativa. • Sistemas de apoyo a contenido estructurado y / o corriente. • Enfoques para la gestión de grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce). 	<ul style="list-style-type: none"> • Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Usar] • Describe los diseños más comunes para los componentes base de sistemas de bases de datos incluyendo el optimizador de consultas, ejecutor de consultas, administrador de almacenamiento, métodos de acceso y procesador de transacciones [Usar] • Cita las metas básicas, funciones y modelos de un sistema de bases de datos [Usar] • Describe los componentes de un sistema de bases de datos y da ejemplos de su uso [Usar] • Identifica las funciones principales de un SGBD y describe sus roles en un sistema de bases de datos [Usar] • Explica los conceptos de independencia de datos y su importancia en un sistema de bases de datos [Usar] • Usa un lenguaje de consulta declarativo para recoger información de una base de datos [Usar] • Describe las capacidades que las bases de datos brindan al apoyar estructuras y/o la secuencia de flujo de datos, ejm. texto [Usar] • Describe los enfoques principales para almacenar y procesar largos volúmenes de datos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [RC04], [EN04], [RG03], [ER15], [CJ11], [KS02]	

Unidad 2: Modelado de datos (14 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Modelado de datos• Modelos conceptuales (e.g., entidad-relación, diagramas UML)• Modelos de hoja de cálculo• Modelos Relacionales.• Modelos orientados a objetos.• Modelos de datos semi-estructurados (expresados usando DTD o XML Schema, por ejemplo)	<ul style="list-style-type: none">• Compare y contrasta modelos apropiados de datos, incluyendo estructuras sus estructuras internas, para diversos tipos de datos [Usar]• Describe los conceptos en notación de modelos (ejm. Diagramas Entidad-Relación o UML) y cómo deben de ser usados [Usar]• Define la terminología fundamental a ser usada en un modelo relacional de datos [Usar]• Describe los principios básicos del modelo relacional de datos [Usar]• Aplica los conceptos de modelado y la notación de un modelo relacional de datos [Usar]• Describe los conceptos principales del modelado OO como son identidad de objetos, constructores de tipos, encapsulación, herencia, polimorfismo, y versiones [Usar]• Describe las diferencias entre modelos de datos relacionales y semi-estructurados [Usar]• Da una semi estructura equivalente (ejm. en DTD o Esquema XML) para un esquema relacional dado [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [SW04], [EN04], [KS02]	

Unidad 3: Indexación (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • El impacto de índices en el rendimiento de consultas. • La estructura básica de un índice. • Mantener un buffer de datos en memoria. • Creando índices con SQL. • Indexando texto. • Indexando la web (e.g., web crawling) 	<ul style="list-style-type: none"> • Generar un archivo índice para una colección de recursos [Usar] • Explicar la función de un índice invertido en la localización de un documento en una colección [Usar] • Explicar cómo rechazar y detener palabras que afectan a la indexación [Usar] • Identificar los índices adecuados para determinado el esquema relacional y el conjunto de consultas [Usar] • Estimar el tiempo para recuperar información, cuando son usados los índices comparado con cuando no son usados [Usar] • Describir los desafíos claves en el rastreo web, por ejemplo, la detección de documentos duplicados, la determinación de la frontera de rastreo [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [WM01], [RG03], [ER15], [CJ11], [KS02]	

Unidad 4: Bases de Datos Relacionales (14 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Mapeo de esquemas conceptuales a esquemas relacionales. • Entidad y integridad referencial. • Algebra relacional y calculo relacional. • Diseño de bases de datos relacionales. • Dependencia funcional. • Descomposición de un esquema. • Llaves candidatas, SuperLlaves y cierre de un conjunto de atributos. • Formas Normales (BCNF) • Dependencias multi-valoradas (4NF) • Uniendo dependencias (PJNF, 5NF) • Teoría de la representación. 	<ul style="list-style-type: none"> • Prepara un esquema relacional de un modelo conceptual desarrollado usando el modelo entidad-relación [Usar] • Explica y demuestra los conceptos de restricciones de integridad de la entidad e integridad referencial (incluyendo la definición del concepto de clave foránea) [Usar] • Demuestra el uso de las operaciones de álgebra relacional de la teoría matemática de conjuntos (unión, intersección, diferencia, y producto Cartesiano) y de las operaciones de álgebra relacional desarrolladas específicamente para las bases de datos relacionales (selección (restringida), proyección, unión y división) [Usar] • Escribe consultas en álgebra relacional [Usar] • Escribe consultas en cálculo relacional de tuplas [Usar] • Determina la dependencia funcional entre dos o más atributos que son subconjunto de una relación [Usar] • Conecta restricciones expresadas como clave primaria y foránea, con dependencias funcionales [Usar] • Calcula la cerradura de un conjunto de atributos dado dependencias funcionales [Usar] • Determina si un conjunto de atributos forma una superclave y/o una clave candidata de una relación dada dependencias funcionales [Usar] • Evalua una descomposición propuesta, a fin de determinar si tiene una unión sin pérdidas o preservación de dependencias [Usar] • Describe las propiedades de la FNBC, FNUP (forma normal unión de proyecto), 5FN [Usar] • Explica el impacto de la normalización en la eficacia de las operaciones de una base de datos especialmente en la optimización de consultas [Usar] • Describe que es una dependencia de multi valor y cual es el tipo de restricciones que especifica [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [WM01], [RG03], [ER15], [CJ11], [KS02]	

Unidad 5: Lenguajes de Consulta (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión general de lenguajes de base de datos. • SQL (definición de datos, formulacion de consultas, sublenguaje update, restricciones, integridad) • Selecciones • Proyecciones • Select-project-join • Agregaciones y agrupaciones. • Subconsultas. • Entornos QBE de cuarta generación. • Diferentes maneras de invocar las consultas no procedimentales en lenguajes convencionales. • Introducción a otros lenguajes importantes de consulta (por ejemplo, XPATH, SPARQL) • Procedimientos almacenados. 	<ul style="list-style-type: none"> • Crear un esquema relacional de bases de datos en SQL que incorpora restricciones clave y restricciones de integridad de entidad e integridad referencial [Usar] • Usar SQL para crear tablas y devuelve (SELECT) la información de una base de datos [Usar] • Evaluar un conjunto de estrategias de procesamiento de consultas y selecciona la estrategia óptima [Usar] • Crear una consulta no-procedimental al llenar plantillas de relaciones para construir un ejemplo del resultado de una consulta requerida [Usar] • Adicionar consultas orientadas a objetos en un lenguaje stand-alone como C++ o Java (ejm. SELECT ColMethod() FROM Objeto) [Usar] • Escribe un procedimiento almacenado que trata con parámetros y con algo de flujo de control de tal forma que tenga funcionalidad [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Die01], [EN04], [Cel05], [KS02]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.
- [CJ11] Date C.J. *SQL and Relational Theory: How to Write Accurate SQL Code*. O'Reilly Media, 2011.
- [Die01] Suzanne W Dietrich. *Understanding Relational Database Query Languages, First Edition*. Prentice Hall, 2001.
- [EN04] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.
- [ER15] Jim Webber Emil Eifrem and Ian Robinson. *Graph Databases*. 2nd. O'Reilly Media, 2015.
- [KS02] Henry F. Korth and Abraham Silberschatz. *Fundamentos de Base de Datos*. McGraw-Hill, 2002.
- [RC04] Peter Rob and Carlos Coronel. *Database Systems: Design, Implementation and Management, Sixth Edition*. Morgan Kaufmann, 2004.
- [RG03] Raghuram Ramakrishnan and Johannes Gehrke. *Database Management Systems*. 3rd. McGraw-Hill, 2003.
- [SW04] Graeme Simsion and Graham Witt. *Data Modeling Essentials, Third Edition*. Morgan Kaufmann, 2004.
- [WM01] Mark Whitehorn and Bill Marklyn. *Inside Relational Databases, Second Edition*. Springer, 2001.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52092. Teoría de la Computación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	1 (Semanal)
2.3 Horas de práctica	:	4 (Semanal)
2.4 Horas autónomas	:	96 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	<ul style="list-style-type: none">• SFW52096. Estructuras Discretas II. (2^{do} Sem)• ENG42005. Inglés II. (2^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso hace énfasis en los lenguajes formales, modelos de computación y computabilidad, además de incluir fundamentos de la complejidad computacional y de los problemas NP completos.

5. OBJETIVOS

- Que el alumno aprenda los conceptos fundamentales de la teoría de lenguajes formales.

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Familiarizarse**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Familiarizarse**)
- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (**Familiarizarse**)

7. TEMAS

Unidad 1: Computabilidad y complejidad básica de autómatas (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Máquinas de estado finito. • Expresiones regulares. • Problema de la parada. • Gramáticas libres de contexto. • Introducción a las clases P y NP y al problema P vs. NP. • Introducción y ejemplos de problemas NP- Completos y a clases NP-Completos. • Máquinas de Turing, o un modelo formal equivalente de computación universal. • Máquinas de Turing no determinísticas. • Jerarquía de Chomsky. • La tesis de Church-Turing. • Computabilidad. • Teorema de Rice. • Ejemplos de funciones no computables. • Implicaciones de la no-computabilidad. 	<ul style="list-style-type: none"> • Discute el concepto de máquina de estado finito [Evaluar] • Diseñe una máquina de estado finito determinista para aceptar un determinado lenguaje [Evaluar] • Genere una expresión regular para representar un lenguaje específico [Evaluar] • Explique porque el problema de la parada no tiene solución algorítmica [Evaluar] • Diseñe una gramática libre de contexto para representar un lenguaje especificado [Evaluar] • Defina las clases P y NP [Evaluar] • Explique el significado de NP-Complejidad [Evaluar] • Explica la tesis de Church-Turing y su importancia [Familiarizarse] • Explica el teorema de Rice y su importancia [Familiarizarse] • Da ejemplos de funciones no computables [Familiarizarse] • Demuestra que un problema es no computable al reducir un problema clásico no computable en base a él [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Mar10], [Lin11], [Sip12]	

Unidad 2: Complejidad Computacional Avanzada (20 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Revisión de las clases P y NP; introducir espacio P y EXP.• Jerarquía polinomial.• NP completitud (Teorema de Cook).• Problemas NP completos clásicos.• Técnicas de reducción.	<ul style="list-style-type: none">• Defina las clases P y NP (También aparece en AL / Automata Básico, Computabilidad y Complejidad) [Evaluar]• Defina la clase P-Space y su relación con la clase EXP [Evaluar]• Explique el significado de NP-Completo (También aparece en AL / Automata Básico, Computabilidad y Complejidad) [Evaluar]• Muestre ejemplos de problemas clásicos en NP - Completo [Evaluar]• Pruebe que un problema es NP- Completo reduciendo un problema conocido como NP-Completo [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Mar10], [Lin11], [Sip12], [HU13]	

Unidad 3: Teoría y Computabilidad Avanzada de Autómatas (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Conjuntos y Lenguajes: <ul style="list-style-type: none"> – Lenguajes Regulares. – Revisión de autómatas finitos determinísticos (Deterministic Finite Automata DFAs) – Autómata finito no determinístico (Nondeterministic Finite Automata NFAs) – Equivalencia de DFAs y NFAs. – Revisión de expresiones regulares; su equivalencia con autómatas finitos. – Propiedades de cierre. – Probando no-regularidad de lenguajes, a través del lema de bombeo (Pumping Lemma) o medios alternativos. • Lenguajes libres de contexto: <ul style="list-style-type: none"> – Autómatas de pila (Push-down automata (PDAs) – Relación entre PDA y gramáticas libres de contexto. – Propiedades de los lenguajes libres de contexto. 	<ul style="list-style-type: none"> • Determina la ubicación de un lenguaje en la jerarquía de Chomsky (regular, libre de contexto, enumerable recursivamente) [Evaluar] • Convierte entre notaciones igualmente poderosas para un lenguaje, incluyendo entre estas AFDs, AFNDs, expresiones regulares, y entre AP y GLCs [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HU13], [Bro93]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Bro93] J. Glenn Brookshear. *Teoría de la Computación*. Addison Wesley Iberoamericana, 1993.
- [HU13] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. Pearson Education, 2013.
- [Lin11] Peter Linz. *An Introduction to Formal Languages and Automata*. 5th. Jones and Bartlett Learning, 2011.
- [Mar10] John Martin. *Introduction to Languages and the Theory of Computation*. 4th. McGraw-Hill, 2010.
- [Sip12] Michael Sipser. *Introduction to the Theory of Computation*. 3rd. Cengage Learning, 2012.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52093. Algoritmos y Estructuras de Datos (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	2
2.2 Horas de teoría	:	-
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	64 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52095. Programación Orientada a Objetos II. (3 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El fundamento teórico de todas las ramas de la informática descansa sobre los algoritmos y estructuras de datos, este curso brindará a los participantes una introducción a estos temas, formando así una base que servirá para los siguientes cursos en la carrera.

5. OBJETIVOS

- Hacer que el alumno entienda la importancia de los algoritmos para la solución de problemas.
- Introducir al alumno hacia el campo de la aplicación de las estructuras de datos.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Familiarizarse**)
- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Familiarizarse**)

7. TEMAS

Unidad 1: Grafos (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Concepto de Grafos. • Grafos Dirigidos y Grafos no Dirigidos. • Utilización de los Grafos. • Medida de la Eficiencia. En tiempo y espacio. • Matrices de Adyacencia. • Matrices de Adyacencia etiquetada. • Listas de Adyacencia. • Implementación de Grafos usando Matrices de Adyacencia. • Implementación de Grafos usando Listas de Adyacencia. • Inserción, Búsqueda y Eliminación de nodos y aristas. • Algoritmos de búsqueda en grafos. 	<ul style="list-style-type: none"> • Adquirir destreza para realizar una implementación correcta. [Usar] • Desarrollar los conocimientos para decidir cuando es mejor usar una técnica de implementación que otra. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cor+09], [Fag+14], [Knu97], [Knu98]	

Unidad 2: Matrices Esparzas (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Conceptos Iniciales. • Matrices poco densas • Medida de la Eficiencia en Tiempo y en Espacio • Creación de la matriz esparza estática vs Dinámicas. • Métodos de inserción, búsqueda y eliminación 	<ul style="list-style-type: none"> • Comprender el uso y implementación de matrices esparzas. [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cor+09], [Fag+14], [Knu97], [Knu98]	

Unidad 3: Árboles Equilibrados (16 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Árboles AVL. • Medida de la Eficiencia. • Rotaciones Simples y Compuestas • Inserción, Eliminación y Búsqueda. • Árboles B , B+ B* y Patricia. 	<ul style="list-style-type: none"> • Comprender las funciones básicas de estas estructuras complejas con el fin de adquirir la capacidad para su implementación. [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cor+09], [Fag+14], [Knu97], [Knu98]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Cor+09] Thomas H. Cormen et al. *Introduction to Algorithms*. Third Edition. ISBN: 978-0-262-53305-8. MIT Press, 2009.
- [Fag+14] José Fager et al. *Estructura de datos*. First Edition. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIN), 2014.
- [Knu97] Donald E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. 3rd. Addison-Wesley Professional, 1997.
- [Knu98] Donald E. Knuth. *The art of computer programming, volume 3: Sorting and searching*. 2nd. Addison-Wesley Professional, 1998.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

ENG42009. English IV (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : ENG42007. English III. (3^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El aprendizaje del idioma inglés en los estudiantes de nivel superior se ha convertido en elemento necesario para su desarrollo académico, personal y profesional. La mayor cantidad de literatura académica de los diversos campos del saber es redactada en inglés, lo cual garantiza que las personas que cuenten con el dominio del idioma puedan siempre estar actualizados.

Asimismo, el conocimiento de este idioma permite tener perspectivas sociales y culturales más amplias. En ese sentido, para un efectivo aprendizaje de la lengua es necesario el desarrollo de las cuatro (4) habilidades; escuchar, hablar, leer y escribir considerando los lineamientos del Marco Común Europeo de Referencia de Lengua - MCERL.

5. OBJETIVOS

- Comunicarse e intercambiar información de manera limitada en situaciones predecibles de forma general.
- Alcanzar el nivel B1 según el MCERL.
- Manejar terminología propia del área de estudios.

6. COMPETENCIAS

CG2) Comunicación bilingüe. (Evaluar)

7. TEMAS

Unidad 1: Talking about facts (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Zero conditional.• Verb + ing /infinitive: like/ want-would like.• Connector (cause - effect).	<ul style="list-style-type: none">• Talking about facts.
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [De 02]	

Unidad 2: What might happen? (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • 1st conditional. • Connector (effect). 	<ul style="list-style-type: none"> • Talking about possibilities.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 3: What are you planning to do? (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Present continuous for future. • Going to • Phrasal verbs – common. • Connector (effect) 	<ul style="list-style-type: none"> • Talk about plans.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 4: Fortune teller (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Future Time (will and going to). • I'd like. • Connector (effect). 	<ul style="list-style-type: none"> • Make predictions.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 5: What was happening? (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Past continuous – including negative. • Questions (what / where / when / who) • Past Continuous. • Connector (effect). 	<ul style="list-style-type: none"> • Describe an ongoing past event.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 6: Can you speak English? (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Design by the teacher. • Connector (effect). 	<ul style="list-style-type: none"> • Get familiar with technical expressions of the field.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.				
Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.				
	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[De 02] Consejo De Europa. "Marco común europeo de referencia para las lenguas". In: *Strasburgo: Consejo de Europa, Ministerio de Educación, Cultura y Deporte/Instituto Cervantes* (2002).



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

EST41005. Estadística Descriptiva y Probabilidades (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 4 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : MAC41010. Fundamentos de Cálculo. (1^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Provee de una introducción a la teoría de las probabilidades e inferencia estadística con aplicaciones, necesarias en el análisis de datos, diseño de modelos aleatorios y toma de decisiones.

5. OBJETIVOS

- Capacidad para diseñar y conducir experimentos, así como usar tecnología como para analizar e interpretar datos.
- Capacidad para identificar, formular y resolver problemas reales.

6. COMPETENCIAS

CG1) Comunicación integral. (Evaluar)

CG3) Investigación. (Evaluar)

7. TEMAS

Unidad 1: Tipo de variable (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Tipo de variable: Continua, discreta.	<ul style="list-style-type: none">• Clasificar las variables relevantes identificadas según su tipo: continuo (intervalo y razón), categórico (nominal, ordinario, dicotómico).• Identificar las variables relevantes de un sistema utilizando un enfoque de proceso.
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [MRo14], [Men14]	

Unidad 2: Estadísticas descriptiva (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tendencia Central (Media, mediana, modo) • Dispersión (Rango, desviación estándar, cuartil) • Gráficos: histograma, boxplot, etc. ∴ Capacidad de comunicación. 	<ul style="list-style-type: none"> • Utilizar medidas de tendencia central y medidas de dispersión para describir los datos recopilados. • Utilizar gráficos para comunicar las características de los datos recopilados.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [MRo14], [Men14]	

Unidad 3: Estadística inferencial (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Determinación del tamaño de la muestra • Intervalo de confianza • Tipo I y error del tipo II • Tipo de distribución • Prueba de hipótesis (t-student, medias, proporciones y ANOVA) • Relaciones entre variables: correlación, regresión. 	<ul style="list-style-type: none"> • Proponer preguntas e hipótesis de interés. • Analizar los datos recopilados utilizando diferentes herramientas estadísticas para responder preguntas de interés. • Dibujar conclusiones basadas en el análisis realizado.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [MRo14], [Men14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Men14] Beaver Mendenhall. *Introducción a la probabilidad y estadística*. 13th. Cengage Learning, 2014.
- [MRo14] Sheldon M.Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. 5th. Academic Press, 2014.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

IIS52024. Sistemas Operativos (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	4 (Semanal)
2.3 Horas de práctica	:	-
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	EGI61009. Arquitectura de Computadores. (3 ^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Un sistema operativo (SO) gestiona los recursos computacionales para completar la ejecución de múltiples aplicaciones y sus procesos asociados. Este curso enseña el diseño de sistemas operativos modernos; e introduce sus conceptos fundamentales que cubren la ejecución multi-programa, *scheduling*, gerencia de memoria, sistemas de archivos y seguridad. Además, el curso incluye actividades de programación en un *sistema operativo mínimo* para resolver problemas y ampliar su funcionalidad. Tenga en cuenta que estas actividades requieren mucho tiempo para completarse. Sin embargo, trabajar en ellos proporciona un valioso aprendizaje sobre los sistemas operativos.

5. OBJETIVOS

- Estudiar el diseño de sistemas operativos modernos.
- Proveer una experiencia práctica al diseñar e implementar un sistema operativo mínimo.

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Familiarizarse**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Familiarizarse**)

7. TEMAS

Unidad 1: Visión general de Sistemas Operativos (3 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Papel y el propósito del sistema operativo. • Funcionalidad de un sistema operativo típico. • Los mecanismos de apoyo modelos cliente-servidor. • Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad) • Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas. 	<ul style="list-style-type: none"> • Explicar los objetivos y funciones de un sistema operativo moderno [Familiarizarse] • Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Evaluar] • Describir las funciones de un sistema operativo contemporáneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarizarse] • Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarizarse] • Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 2: Principios de Sistemas Operativos (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Estructuración de Sistemas Operativos (monolítico, capas, modular, los modelos micro-kernel) • Abstracciones, procesos y recursos. • Los conceptos de interfaces de programa de aplicación (API) • La evolución de las técnicas de hardware / software y las necesidades de aplicación • Organización de dispositivos. • Interrupciones: métodos e implementaciones. • Concepto de estado de usuario / sistema y la protección, la transición al modo kernel. 	<ul style="list-style-type: none"> • Explicar el concepto de una capa lógica [Familiarizarse] • Explicar los beneficios de construir capas abstractas en forma jerárquica [Familiarizarse] • Describir el valor de la API y <i>middleware</i> [Familiarizarse] • Describir como los recursos computacionales son usados por aplicaciones de software y administradas por el software del sistema [Familiarizarse] • Contrastar el modo <i>kernel</i> y modo usuario en un sistema operativo [Evaluar] • Discutir las ventajas y desventajas del uso de procesamiento con interrupciones [Familiarizarse] • Explicar el uso de una lista de dispositivos y el controlador de colas de entrada y salida [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 3: Concurrencia (9 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Diagramas de estado. • Estructuras (lista preparada, bloques de control de procesos, y así sucesivamente) • <i>Dispatching</i> y cambio de contexto. • El papel de las interrupciones. • Gestionar el acceso a los objetos del sistema operativo de forma atómica. • La implementación de primitivas de sincronización. • Problemas de multiprocesador (spin-locks, reentrada) 	<ul style="list-style-type: none"> • Describir la necesidad de concurrencia en el marco de un sistema operativo [Familiarizarse] • Demostrar los potenciales problemas de tiempo de ejecución derivados de la operación simultánea de muchas tareas diferentes [Usar] • Resumir el rango de mecanismos que pueden ser usados a nivel del sistema operativo para realizar sistemas concurrentes y describir los beneficios de cada uno [Familiarizarse] • Explicar los diferentes estados por los que una tarea debe pasar y las estructuras de datos necesarias para el manejo de varias tareas [Familiarizarse] • Resumir las técnicas para lograr sincronización en un sistema operativo (por ejemplo, describir como implementar semáforos usando primitivas del sistema operativo.) [Familiarizarse] • Describir las razones para usar interrupciones, <i>dispatching</i>, y cambio de contexto para soportar concurrencia en un sistema operativo [Familiarizarse] • Crear diagramas de estado y transición para los problemas de dominios simples [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 4: Planificación y despacho (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • <i>Scheduling preemptive</i> y <i>non-preemptive</i>. • <i>Scheduling</i> y políticas. • Procesos y subprocesos. • Plazos y cuestiones en tiempo real. 	<ul style="list-style-type: none"> • Comparar y contrastar los algoritmos comunes que se utilizan tanto para <i>scheduling preemptive</i> y <i>pre-emptive</i> de tareas en los sistemas operativos, como la comparación de prioridad, el rendimiento, y los esquemas de distribución equitativa [Evaluar] • Describir las relaciones entre los algoritmos de <i>scheduling</i> y dominios de aplicación [Familiarizarse] • Discutir los tipos de <i>scheduling</i> en procesadores en de corto, mediano, largo plazo y I/O [Familiarizarse] • Describir las diferencias entre procesos y <i>threads</i> [Familiarizarse] • Comparar y contrastar enfoques estáticos y dinámicos para <i>scheduling</i> en tiempo real [Evaluar] • Discutir sobre la necesidad de <i>preemption</i> y <i>deadline scheduling</i> [Familiarizarse] • Identificar formas en que la lógica expresada en algoritmos de planificación son de aplicación a otros ámbitos, tales como I/O del disco, la programación de disco de red, programación de proyectos y problemas más allá de la computación [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 5: Manejo de memoria (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Revisión de la memoria física y hardware de gestión de memoria. • Conjuntos de trabajo y thrashing. • El almacenamiento en caché 	<ul style="list-style-type: none"> • Explicar la jerarquía de la memoria y <i>tradeoffs</i> de costo-rendimiento [Familiarizarse] • Resumir los principios de memoria virtual tal como se aplica para el almacenamiento en cache y paginación [Familiarizarse] • Evaluar las ventajas y desventajas en términos del tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y la velocidad del procesador [Evaluar] • Describir las diferentes formas de asignar memoria a las tareas, citando las ventajas relativas de cada uno [Familiarizarse] • Describir el motivo y el uso de memoria caché (rendimiento y proximidad, dimensión diferente de como los caches complican el aislamiento y abstracción en VM) [Familiarizarse] • Estudiar los conceptos de <i>thrashing</i>, tanto en términos de las razones por las que se produce y las técnicas usadas para el reconocimiento y manejo del problema [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 6: Seguridad y protección (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión general de la seguridad del sistema . • Política / mecanismo de separación. • Métodos de seguridad y dispositivos. • Protección, control de acceso y autenticación. • Las copias de seguridad. 	<ul style="list-style-type: none"> • Explicar la necesidad para la protección y seguridad en un sistema operativo [Familiarizarse] • Resumir las características y limitaciones de un sistema operativo usado para proporcionar protección y seguridad [Familiarizarse] • Explicar el mecanismo disponible en un OS para controlar los accesos a los recursos [Familiarizarse] • Realizar tareas de administración de sistemas sencillas de acuerdo a una política de seguridad, por ejemplo la creación de cuentas, el establecimiento de permisos, aplicación de parches y organización de backups regulares [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 7: Máquinas virtuales (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red) • Paginación y la memoria virtual. • Sistemas de archivos virtuales. • Los Hypervisores. • Virtualización portátil; emulación vs aislamiento. • Costo de la virtualización. 	<ul style="list-style-type: none"> • Explicar el concepto de memoria virtual y la forma cómo se realiza en hardware y software [Familiarizarse] • Diferenciar emulación y el aislamiento [Familiarizarse] • Evaluar virtualización de compensaciones [Evaluar] • Discutir sobre hipervisores y la necesidad para ellos en conjunto con diferentes tipos de hipervisores [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 8: Manejo de dispositivos (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Características de los dispositivos serie y paralelo. • Haciendo de abstracción de dispositivos. • Estrategias de buffering. • Acceso directo a memoria. • La recuperación de fallos. 	<ul style="list-style-type: none"> • Explique la diferencia clave entre dispositivos seriales y paralelos e identificar las condiciones en las cuales cada uno es apropiado [Familiarizarse] • Identificar la relación entre el hardware físico y los dispositivos virtuales mantenidos por el sistema operativo [Familiarizarse] • Explicar <i>buffering</i> y describir las estrategias para su aplicación [Familiarizarse] • Diferenciar los mecanismos utilizados en la interconexión de un rango de dispositivos (incluyendo dispositivos portátiles, redes, multimedia) a un ordenador y explicar las implicaciones de éstas para el diseño de un sistema operativo [Familiarizarse] • Describir las ventajas y desventajas de acceso directo a memoria y discutir las circunstancias en cuales se justifica su uso [Familiarizarse] • Identificar los requerimientos para recuperación de errores [Familiarizarse] • Implementar un controlador de dispositivo simple para una gama de posibles equipos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 9: Sistema de archivos (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Archivos: los datos, metadatos, operaciones, organización, amortiguadores, secuenciales, no secuencial. • Directorios: contenido y estructura. • Los sistemas de archivos: partición, montar / desmontar sistemas de archivos virtuales. • Técnicas estándar de implementación . • Archivos asignados en memoria. • Sistemas de archivos de propósito especial. • Naming, búsqueda, acceso, copias de seguridad. • La bitacora y los sistemas de archivos estructurados (log) 	<ul style="list-style-type: none"> • Describir las decisiones que deben tomarse en el diseño de sistemas de archivos [Familiarizarse] • Comparar y contrastar los diferentes enfoques para la organización de archivos, el reconocimiento de las fortalezas y debilidades de cada uno. [Evaluar] • Resumir cómo el desarrollo de hardware ha dado lugar a cambios en las prioridades para el diseño y la gestión de sistemas de archivos [Familiarizarse] • Resumir el uso de diarios y como los sistemas de archivos de registro estructurado mejora la tolerancia a fallos [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 10: Sistemas <i>embedded</i> y de tiempo real (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Proceso y programación de tareas. • Los requisitos de gestión de memoria / disco en un entorno en tiempo real. • Los fracasos, los riesgos y la recuperación. • Preocupaciones especiales en sistemas de tiempo real. 	<ul style="list-style-type: none"> • Describir que hace a un sistema un sistema en tiempo real [Familiarizarse] • Explicar la presencia y describir las características de latencia en sistemas de tiempo real [Familiarizarse] • Resumir los problemas especiales que los sistemas en tiempo real presentan, incluyendo el riesgo, y cómo se tratan estos problemas [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 11: Tolerancia a fallas (3 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Conceptos fundamentales: sistemas fiables y disponibles. • Redundancia espacial y temporal. • Los métodos utilizados para implementar la tolerancia a fallos. • Los ejemplos de los mecanismos del sistema operativo para la detección, recuperación, reinicio para implementar la tolerancia a fallos, el uso de estas técnicas para los servicios propios del sistema operativo. 	<ul style="list-style-type: none"> • Explicar la importancia de los términos tolerancia a fallos, fiabilidad y disponibilidad [Familiarizarse] • Explicar en términos generales la gama de métodos para implementar la tolerancia a fallos en un sistema operativo [Familiarizarse] • Explicar cómo un sistema operativo puede continuar funcionando después de que ocurra una falla [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unidad 12: Evaluación del desempeño de sistemas (3 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • ¿Por qué el rendimiento del sistema debe ser evaluado? • ¿Qué se va a evaluar? • Sistemas de políticas de rendimiento, por ejemplo, el almacenamiento en caché, de paginación, la programación, la gestión de memoria, y la seguridad. • Modelos de evaluación: analítica, simulación, o de implementación específico determinista. • Cómo recoger los datos de evaluación (perfiles y mecanismos de localización) 	<ul style="list-style-type: none"> • Describir las medidas de rendimiento utilizados para determinar cómo el sistema funciona [Familiarizarse] • Explicar los principales modelos de evaluación utilizados para evaluar un sistema [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [AD14] Thomas Anderson and Michael Dahlin. *Operating Systems: Principles and Practice*. 2nd. Recursive Books, 2014. ISBN: 978-0985673529.
- [Avi12] Greg Gagne Avi Silberschatz Peter Baer Galvin. *Operating System Concepts, 9/E*. John Wiley & Sons, Inc., 2012. ISBN: 978-1-118-06333-0.
- [Sta05] William Stallings. *Operating Systems: Internals and Design Principles, 5/E*. Prentice Hall, 2005. ISBN: 0-13-147954-7.
- [Tan01] Andrew S. Tanenbaum. *Modern Operating Systems, 4/E*. Prentice Hall, 2001. ISBN: 0-13-031358-0.
- [Tan06] Andrew S. Tanenbaum. *Operating Systems Design and Implementation, 3/E*. Prentice Hall, 2006. ISBN: 0-13-142938-8.



1. CURSO

ECO51002. Economía General (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
2.2 Horas de teoría : 4 (Semanal)
2.3 Horas de práctica : -
2.4 Horas autónomas : 128 (horas)
2.5 Duración del periodo : 16 semanas
2.6 Condición : Obligatorio
2.7 Modalidad : Presencial
2.8 Prerrequisitos : Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El objetivo del curso es que el estudiante comprenda la interrelación que existe entre los sistemas políticos y económicos de un país o región. El hilo conductor de este curso será el libro "Why Nations Fail: The Origins of Power, Prosperity, and Poverty" de Acemoglu-Robinson. El aprendizaje del curso debe ser una interpretación informada de distintas dinámicas sociales en las que se organiza y reparte poder, sea de carácter simbólico, económico y/o político. Este curso debe trabajar la capacidad del estudiante de utilizar conceptos más complejos y desarrollar interpretaciones más elaboradas de la realidad.

5. OBJETIVOS

- Capacidad de interpretar información.
- Capacidad para formular alternativas de solución.
- Capacidad de comprender textos

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Culturas de Gobernanza y Distribución de Poder (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• ¿Cómo se relaciona la economía con la política?.• El rol de las Instituciones.• Análisis de casos.	<ul style="list-style-type: none">• Desarrollo del interés por conocer sobre temas actuales en la sociedad peruana y el mundo.
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Lar15]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[Lar15] Lessig, Larry. *Our democracy no longer represents the people. Here's how we fix it*. Youtube. 2015. URL: <https://www.youtube.com/watch?v=PJy8vTu66tE>.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52087. Compiladores (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	2
2.2 Horas de teoría	:	-
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	64 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52092. Teoría de la Computación. (4 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Que el alumno conozca y comprenda los conceptos y principios fundamentales de la teoría de compilación para realizar la construcción de un compilador

5. OBJETIVOS

- Conocer las técnicas básicas empleadas durante el proceso de generación intermedio, optimización y generación de código.
- Aprender a implementar pequeños compiladores.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinares en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Usar**)

7. TEMAS

Unidad 1: Representación de programas (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Programas que tienen otros programas como entrada tales como interpretes, compiladores, revisores de tipos y generadores de documentación. • Árboles de sintaxis abstracta, para contrastar la sintaxis correcta. • Estructuras de datos que representan código para ejecución, traducción o transmisión. • Compilación en tiempo just-in time y re-compilación dinámica. • Otras características comunes de las máquinas virtuales, tales como carga de clases, hilos y seguridad. 	<ul style="list-style-type: none"> • Explicar como programas que procesan otros programas tratan a los otros programas como su entrada de datos [Familiarizarse] • Describir un árbol de sintaxis abstracto para un lenguaje pequeño [Familiarizarse] • Describir los beneficios de tener representaciones de programas que no sean cadenas de código fuente [Familiarizarse] • Escribir un programa para procesar alguna representación de código para algún propósito, tales como un interprete, una expresión optimizada, o un generador de documentación [Familiarizarse] • Explicar el uso de metadatos en las representaciones de tiempo de ejecución de objetos y registros de activación, tales como los punteros de la clase, las longitudes de arreglos, direcciones de retorno, y punteros de <i>frame</i> [Familiarizarse] • Discutir las ventajas, desventajas y dificultades del término (<i>just-in-time</i>) y recompilación automática [Familiarizarse] • Identificar los servicios proporcionados por los sistemas de tiempo de ejecución en lenguajes modernos [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Lou04b]	

Unidad 2: Traducción y ejecución de lenguajes (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Interpretación vs. compilación a código nativo vs. compilación de representación portable intermedia. • Pipeline de traducción de lenguajes: análisis, revisión opcional de tipos, traducción, enlazamiento, ejecución: <ul style="list-style-type: none"> – Ejecución como código nativo o con una máquina virtual – Alternativas como carga dinámica y codificación dinámica de código (o “just-in-time”) • Representación en tiempo de ejecución de construcción del lenguaje núcleo tales como objetos (tablas de métodos) y funciones de primera clase (cerradas) • Ejecución en tiempo real de asignación de memoria: pila de llamadas, montículo, datos estáticos: <ul style="list-style-type: none"> – Implementación de bucles, recursividad y llamadas de cola • Gestión de memoria: <ul style="list-style-type: none"> – Gestión manual de memoria: asignación, limpieza y reuso de la pila de memoria – Gestión automática de memoria: recolección de datos no utilizados (<i>garbage collection</i>) como una técnica automática usando la noción de accesibilidad 	<ul style="list-style-type: none"> • Distinguir una definición de un lenguaje de una implementación particular de un lenguaje (compilador vs interprete, tiempo de ejecución de la representación de los objetos de datos, etc) [Evaluar] • Distinguir sintaxis y parseo de la semántica y la evaluación [Evaluar] • Bosqueje una representación de bajo nivel de tiempo de ejecución de construcciones del lenguaje base, tales como objetos o cierres (<i>closures</i>) [Evaluar] • Explicar cómo las implementaciones de los lenguajes de programación típicamente organizan la memoria en datos globales, texto, <i>heap</i>, y secciones de pila y cómo las características tales como recursión y administración de memoria son mapeados a este modelo de memoria [Evaluar] • Identificar y corregir las pérdidas de memoria y punteros desreferenciados [Evaluar] • Discutir los beneficios y limitaciones de la recolección de basura (<i>garbage collection</i>), incluyendo la noción de accesibilidad [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Aho+11], [Lou04a], [TS98], [App02]	

Unidad 3: Análisis de sintaxis (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Exploración (análisis léxico) usando expresiones regulares. • Estrategias de análisis incluyendo técnicas de arriba a abajo (top-down) (p.e. descenso recursivo, análisis temprano o LL) y de abajo a arriba (bottom-up) (ej. ‘llamadas hacia atrás - bracktracking, o LR); rol de las gramáticas libres de contexto. • Generación de exploradores (scanners) y analizadores a partir de especificaciones declarativas. 	<ul style="list-style-type: none"> • Usar gramáticas formales para especificar la sintaxis de los lenguajes [Evaluar] • Usar herramientas declarativas para generar parseadores y escáneres [Evaluar] • Identificar las características clave en las definiciones de sintaxis: ambigüedad, asociatividad, precedencia [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Aho+11], [Lou04a], [TS98], [App02]	

Unidad 4: Análisis semántico de compiladores (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Representaciones de programas de alto nivel tales como árboles de sintaxis abstractas. • Alcance y resolución de vínculos. • Revisión de tipos. • Especificaciones declarativas tales como gramáticas atribuidas. 	<ul style="list-style-type: none"> • Implementar analizadores sensibles al contexto y estáticos a nivel de fuente, tales como, verificadores de tipos o resolvedores de identificadores para identificar las ocurrencias de vínculo [Evaluar] • Describir analizadores semánticos usando una gramática con atributos [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Aho+11], [Lou04a], [TS98], [App02]	

Unidad 5: Generación de código (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Llamadas a procedimientos y métodos en envío. • Compilación separada; vinculación. • Selección de instrucciones. • Calendarización de instrucciones. • Asignación de registros. • Optimización por rendija (peephole) 	<ul style="list-style-type: none"> • Identificar todos los pasos esenciales para convertir automáticamente código fuente en código ensamblador o otros lenguajes de bajo nivel [Evaluar] • Generar código de bajo nivel para llamadas a funciones en lenguajes modernos [Evaluar] • Discutir por qué la compilación separada requiere convenciones de llamadas uniformes [Evaluar] • Discutir por qué la compilación separada limita la optimización debido a efectos de llamadas desconocidas [Evaluar] • Discutir oportunidades para optimización introducida por la traducción y enfoques para alcanzar la optimización, tales como la selección de la instrucción, planificación de instrucción, asignación de registros y optimización de tipo mirilla (<i>peephole optimization</i>) [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Aho+11], [Lou04a], [TS98], [App02]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Aho+11] Alfred Aho et al. *Compilers Principles Techniques And Tools*. 2nd. ISBN:10-970-26-1133-4. Pearson, 2011.
- [App02] A. W. Appel. *Modern compiler implementation in Java*. 2.a edición. Cambridge University Press, 2002.
- [Lou04a] Kenneth C. Louden. *Compiler Construction: Principles and Practice*. Thomson, 2004.
- [Lou04b] Kenneth C. Louden. *Lenguajes de Programacion*. Thomson, 2004.
- [TS98] Bernard Teufel and Stephanie Schmidt. *Fundamentos de Compiladores*. Addison Wesley Iberoamericana, 1998.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52088. Ingeniería de Software I (Obligatorio)

2. INFORMACIÓN GENERAL

- | | | |
|--------------------------|---|---|
| 2.1 Créditos | : | 3 |
| 2.2 Horas de teoría | : | 1 (Semanal) |
| 2.3 Horas de práctica | : | 2 (Semanal) |
| 2.4 Horas autónomas | : | 96 (horas) |
| 2.5 Duración del periodo | : | 16 semanas |
| 2.6 Condición | : | Obligatorio |
| 2.7 Modalidad | : | Presencial |
| 2.8 Prerrequisitos | : | <ul style="list-style-type: none">• SFW52095. Programación Orientada a Objetos II. (3^{er} Sem)• SFW52091. Gerenciamiento de Datos I. (4^{to} Sem) |

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La tarea de desarrollar software, excepto para aplicaciones sumamente simples, exige la ejecución de un proceso de desarrollo bien definido. Los profesionales de esta área requieren un alto grado de conocimiento de los diferentes modelos e proceso de desarrollo, para que sean capaces de elegir el más idóneo para cada proyecto de desarrollo. Por otro lado, el desarrollo de sistemas de mediana y gran escala requiere del uso de bibliotecas de patrones y componentes y del dominio de técnicas relacionadas al diseño basado en componentes.

5. OBJETIVOS

- Brindar al alumno un marco teórico y práctico para el desarrollo de software bajo estándares de calidad.
- Familiarizar al alumno con los procesos de modelamiento y construcción de software a través del uso de herramientas CASE.
- Los alumnos debe ser capaces de seleccionar Arquitecturas y Plataformas tecnológicas ad-hoc a los escenarios de implementación.
- Aplicar el modelamiento basado en componentes y fin de asegurar variables como calidad, costo y *time-to-market* en los procesos de desarrollo.
- Brindar a los alumnos mejores prácticas para la verificación y validación del software.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Usar**)

7. TEMAS

Unidad 1: Ingeniería de Requisitos (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Al describir los requisitos funcionales utilizando, por ejemplo, los casos de uso o historias de los usuarios. • Propiedades de requisitos, incluyendo la consistencia, validez, integridad y viabilidad. • Requisitos de software elicitación. • Descripción de datos del sistema utilizando, por ejemplo, los diagramas de clases o diagramas entidad-relación. • Requisitos no funcionales y su relación con la calidad del software. • Evaluación y uso de especificaciones de requisitos. • Requisitos de las técnicas de modelado de análisis. • La aceptabilidad de las consideraciones de certeza/incertidumbre sobre el comportamiento del software/sistema. • Prototipos. • Conceptos básicos de la especificación formal de requisitos. • Especificación de requisitos. • Validación de requisitos. • Rastreo de requisitos. 	<ul style="list-style-type: none"> • Enumerar los componentes clave de un caso de uso o una descripción similar de algún comportamiento que es requerido para un sistema [Evaluar] • Describir cómo el proceso de ingeniería de requisitos apoya la obtención y validación de los requisitos de comportamiento [Evaluar] • Interpretar un modelo de requisitos dada por un sistema de software simple [Evaluar] • Describir los retos fundamentales y técnicas comunes que se utilizan para la obtención de requisitos [Evaluar] • Enumerar los componentes clave de un modelo de datos (por ejemplo, diagramas de clases o diagramas ER) [Evaluar] • Identificar los requisitos funcionales y no funcionales en una especificación de requisitos dada por un sistema de software [Evaluar] • Realizar una revisión de un conjunto de requisitos de software para determinar la calidad de los requisitos con respecto a las características de los buenos requisitos [Evaluar] • Aplicar elementos clave y métodos comunes para la obtención y el análisis para producir un conjunto de requisitos de software para un sistema de software de tamaño medio [Evaluar] • Comparar los métodos ágiles y el dirigido por planes para la especificación y validación de requisitos y describir los beneficios y riesgos asociados con cada uno [Evaluar] • Usar un método común, no formal para modelar y especificar los requisitos para un sistema de software de tamaño medio [Evaluar] • Traducir al lenguaje natural una especificación de requisitos de software (por ejemplo, un contrato de componentes de software) escrito en un lenguaje de especificación formal [Evaluar] • Crear un prototipo de un sistema de software para reducir el riesgo en los requisitos [Evaluar] • Diferenciar entre el rastreo (<i>tracing</i>) hacia adelante y hacia atrás y explicar su papel en el proceso de validación de requisitos [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [ES14], [HF03]	

Unidad 2: Diseño de Software (18 horas)

Competencias esperadas:

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> ● Principios de diseño del sistema: niveles de abstracción (diseño arquitectónico y el diseño detallado), separación de intereses, ocultamiento de información, de acoplamiento y de cohesión, de reutilización de estructuras estándar. ● Diseño de paradigmas tales como diseño estructurado (descomposición funcional de arriba hacia abajo), el análisis orientado a objetos y diseño, orientado a eventos de diseño, diseño de nivel de componente, centrado datos estructurada, orientada a aspectos, orientado a la función, orientado al servicio. ● Modelos estructurales y de comportamiento de los diseños de software. ● Diseño de patrones. ● Relaciones entre los requisitos y diseños: La transformación de modelos, el diseño de los contratos, invariantes. ● Conceptos de arquitectura de software y arquitecturas estándar (por ejemplo, cliente-servidor, n-capas, transforman centrados, tubos y filtros). ● El uso de componentes de diseño: selección de componentes, diseño, adaptación y componentes de ensamblaje, componentes y patrones, componentes y objetos (por ejemplo, construir una GUI usando un standard widget set) ● Diseños de refactorización utilizando patrones de diseño ● Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos. ● Medición y análisis de la calidad de un diseño. ● Compensaciones entre diferentes aspectos de la calidad. ● Aplicaciones en frameworks. ● Middleware: El paradigma de la orientación a objetos con middleware, requerimientos para correr y clasificar objetos, monitores de procesamiento de transacciones y el sistema de flujo de trabajo. ● Principales diseños de seguridad y codificación (cross-reference IAS/Principles of secure design). <ul style="list-style-type: none"> – Principio de privilegios mínimos – Principio de falla segura por defecto – Principio de aceptabilidad psicológica 	<ul style="list-style-type: none"> ● Formular los principios de diseño, incluyendo la separación de problemas, ocultación de información, acoplamiento y cohesión, y la encapsulación [Familiarizarse] ● Usar un paradigma de diseño para diseñar un sistema de software básico y explicar cómo los principios de diseño del sistema se han aplicado en este diseño [Usar] ● Construir modelos del diseño de un sistema de software simple los cuales son apropiado para el paradigma utilizado para diseñarlo [Usar] ● En el contexto de un paradigma de diseño simple, describir uno o más patrones de diseño que podrían ser aplicables al diseño de un sistema de software simple [Familiarizarse] ● Para un sistema simple adecuado para una situación dada, discutir y seleccionar un paradigma de diseño apropiado [Usar] ● Crear modelos apropiados para la estructura y el comportamiento de los productos de software desde la especificaciones de requisitos [Usar] ● Explicar las relaciones entre los requisitos para un producto de software y su diseño, utilizando los modelos apropiados [Evaluar] ● Para el diseño de un sistema de software simple dentro del contexto de un único paradigma de diseño, describir la arquitectura de software de ese sistema [Familiarizarse] ● Dado un diseño de alto nivel, identificar la arquitectura de software mediante la diferenciación entre las arquitecturas comunes de software, tales como 3 capas (<i>3-tier</i>), <i>pipe-and-filter</i>, y cliente-servidor [Familiarizarse] ● Investigar el impacto de la selección arquitecturas de software en el diseño de un sistema simple [Evaluar] ● Aplicar ejemplos simples de patrones en un diseño de software [Usar] ● Describir una manera de refactorar y discutir cuando esto debe ser aplicado [Familiarizarse] ● Seleccionar componentes adecuados para el uso en un diseño de un producto de software [Usar] ● Explicar cómo los componentes deben ser adaptados para ser usados en el diseño de un producto de software [Familiarizarse] ● Diseñar un contrato para un típico componente de software pequeño para el uso de un dado sistema, [Usar] ● Discutir y seleccionar la arquitectura de software

Unidad 3: Construcción de Software (24 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> ● Prácticas de codificación: técnicas, idiomas/patrones, mecanismos para construcción de programas de calidad: <ul style="list-style-type: none"> – Prácticas de codificación defensiva – Prácticas de codificación segura – Utilizando mecanismos de manejo de excepciones para hacer el programa más robusto, tolerante a fallas ● Normas de codificación. ● Estrategias de integración. ● Desarrollando contexto: "campo verde" frente a la base de código existente : <ul style="list-style-type: none"> – Análisis de cambio impacto – Cambio de actualización ● Los problemas de seguridad potenciales en los programas : <ul style="list-style-type: none"> – Buffer y otros tipos de desbordamientos – Condiciones elemento Race – Inicialización incorrecta, incluyendo la elección de los privilegios – Entrada Comprobación – Suponiendo éxito y corrección – La validación de las hipótesis 	<ul style="list-style-type: none"> ● Describir técnicas, lenguajes de codificación y mecanismos de implementación para conseguir las propiedades deseadas, tales como la confiabilidad, la eficiencia y la robustez [Evaluar] ● Construir código robusto utilizando los mecanismos de manejo de excepciones [Evaluar] ● Describir la codificación segura y prácticas de codificación de defensa [Evaluar] ● Seleccionar y utilizar un estándar de codificación definido en un pequeño proyecto de software [Evaluar] ● Comparar y contrastar las estrategias de integración incluyendo: de arriba hacia abajo (<i>top-down</i>), de abajo hacia arriba (<i>bottom-up</i>), y la integración Sándwich [Evaluar] ● Describir el proceso de analizar e implementar los cambios a la base de código desarrollado para un proyecto específico [Evaluar] ● Describir el proceso de analizar e implementar los cambios a una gran base de código existente [Evaluar] ● Reescribir un programa sencillo para eliminar vulnerabilidades comunes, tales como desbordamientos de búffer, desbordamientos de enteros y condiciones de carrera [Evaluar] ● Escribir un componente de software que realiza alguna tarea no trivial y es resistente a errores en la entrada y en tiempo de ejecución [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> ● Desarrollo de ejercicios prácticos 	
Lecturas : [ES14], [HF03]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [ES14] Bert Bates Eric Freeman Elisabeth Robson and Kathy Sierra. *Head First Design Patterns*. 2nd. O'Reilly Media, Inc, July 2014.
- [HF03] Brian Lyons Hans-Erik Eriksson Magnus Penker and Davis Fado. *UML 2 Toolkit*. 2nd. Wiley, Oct. 2003.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52089. Análisis y Diseño de Algoritmos (Obligatorio)

2. INFORMACIÓN GENERAL

- | | | |
|--------------------------|---|--|
| 2.1 Créditos | : | 2 |
| 2.2 Horas de teoría | : | - |
| 2.3 Horas de práctica | : | 2 (Semanal) |
| 2.4 Horas autónomas | : | 64 (horas) |
| 2.5 Duración del periodo | : | 16 semanas |
| 2.6 Condición | : | Obligatorio |
| 2.7 Modalidad | : | Presencial |
| 2.8 Prerrequisitos | : | <ul style="list-style-type: none">• SFW52093. Algoritmos y Estructuras de Datos. (4^{to} Sem)• SFW52092. Teoría de la Computación. (4^{to} Sem) |

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Un algoritmo es, esencialmente, un conjunto bien definido de reglas o instrucciones que permitan resolver un problema computacional. El estudio teórico del desempeño de los algoritmos y los recursos utilizados por estos, generalmente tiempo y espacio, nos permite evaluar si un algoritmo es adecuado para un resolver un problema específico, compararlo con otros algoritmos para el mismo problema o incluso delimitar la frontera entre lo viable y lo imposible.

Esta materia es tan importante que incluso Donald E. Knuth definió a Ciencia de la Computación como el estudio de algoritmos.

En este curso serán presentadas las técnicas más comunes utilizadas en el análisis y diseño de algoritmos eficientes, con el propósito de aprender los principios fundamentales del diseño, implementación y análisis de algoritmos para la solución de problemas computacionales.

5. OBJETIVOS

- Desarrollar la capacidad para evaluar la complejidad y calidad de algoritmos propuestos para un determinado problema.
- Estudiar los algoritmos más representativos, introductorios de las clases más importantes de problemas tratados en computación.
- Desarrollar la capacidad de resolución de problemas algorítmicos utilizando los principios fundamentales de diseño de algoritmos aprendidos.
- Ser capaz de responder a las siguientes preguntas cuando le sea presentado un nuevo algoritmo: ¿Cuán buen desempeño tiene?, ¿Existe una mejor forma de resolver el problema?

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Usar**)

7. TEMAS

Unidad 1: Análisis Básico (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Diferencias entre el mejor, el esperado y el peor caso de un algoritmo. • Análisis asintótico de complejidad de cotas superior y esperada. • Definición formal de la Notación Big O. • Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial. • Uso de la notación Big O. • Relaciones recurrentes. • Análisis de algoritmos iterativos y recursivos. • Teorema Maestro y Árboles Recursivos. 	<ul style="list-style-type: none"> • Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Evaluar] • En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Evaluar] • Determine informalmente el tiempo y el espacio de complejidad de diferentes algoritmos [Evaluar] • Indique la definición formal de Big O [Evaluar] • Lista y contraste de clases estándares de complejidad [Evaluar] • Use la notación formal de la Big O para dar límites superiores asintóticos en la complejidad de tiempo y espacio de los algoritmos [Evaluar] • Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Evaluar] • Explicar el uso de la notación theta grande, omega grande y o pequeña para describir la cantidad de trabajo hecho por un algoritmo [Evaluar] • Usar relaciones recurrentes para determinar el tiempo de complejidad de algoritmos recursivamente definidos [Evaluar] • Resuelve relaciones de recurrencia básicas, por ejemplo. usando alguna forma del Teorema Maestro [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KT05], [DPV06], [RS09], [SF13], [Knu97]	

Unidad 2: Estrategias Algorítmicas (30 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmos de fuerza bruta. • Algoritmos voraces. • Divide y vencerás. • Programación Dinámica. 	<ul style="list-style-type: none"> • Para cada una de las estrategias (fuerza bruta, algoritmo goloso, divide y vencerás, recursividad en reversa y programación dinámica), identifica un ejemplo práctico en el cual se pueda aplicar [Evaluar] • Utiliza un enfoque voraz para resolver un problema específico y determina si la regla escogida lo guía a una solución óptima [Evaluar] • Usa un algoritmo de divide-y-vencerás para resolver un determinado problema [Evaluar] • Usa programación dinámica para resolver un problema determinado [Evaluar] • Determina el enfoque algorítmico adecuado para un problema [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KT05], [DPV06], [RS09], [Als99]	

Unidad 3: Algoritmos y Estructuras de Datos fundamentales (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo. • Algoritmos de búsqueda secuencial y binaria. • Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción) • Algoritmos de ordenamiento con peor caso o caso promedio en $O(N \lg N)$ (Quicksort, Heapsort, Mergesort) • Grafos y algoritmos en grafos: <ul style="list-style-type: none"> – Representación de grafos (ej., lista de adyacencia, matriz de adyacencia) – Recorrido en profundidad y amplitud • Montículos (Heaps) • Grafos y algoritmos en grafos: <ul style="list-style-type: none"> – Problema de corte máximo y mínimo – Búsqueda local 	<ul style="list-style-type: none"> • Implementar algoritmos numéricos básicos [Evaluar] • Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar] • Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Evaluar] • Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Usar] • Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse] • Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Evaluar] • Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto en específico [Evaluar] • Describir la propiedad del heap y el uso de heaps como una implementación de colas de prioridad [Evaluar] • Resolver problemas usando algoritmos de grafos, incluyendo camino más corto de una sola fuente y camino más corto de todos los pares, y como mínimo un algoritmo de arbol de expansion minima [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KT05], [DPV06], [RS09], [SW11], [GT09]	

Unidad 4: Computabilidad y complejidad básica de autómatas (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Introducción a las clases P y NP y al problema P vs. NP. • Introducción y ejemplos de problemas NP- Completos y a clases NP-Completos. 	<ul style="list-style-type: none"> • Defina las clases P y NP [Familiarizarse] • Explique el significado de NP-Complejidad [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KT05], [DPV06], [RS09]	

Unidad 5: Estructuras de Datos Avanzadas y Análisis de Algoritmos (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Grafos (ej. Ordenamiento Topológico, encontrando componentes fuertemente conectados) • Algoritmos Teórico-Numéricos (Aritmética Modular, Prueba del Número Primo, Factorización Entera) • Algoritmos aleatorios. • Análisis amortizado. • Análisis Probabilístico. 	<ul style="list-style-type: none"> • Entender el mapeamiento de problemas del mundo real a soluciones algorítmicas (ejemplo, problemas de grafos, programas lineales,etc) [Familiarizarse] • Seleccionar y aplicar técnicas de algoritmos avanzadas (ejemplo, randomización, aproximación) para resolver problemas reales [Usar] • Seleccionar y aplicar técnicas avanzadas de análisis (ejemplo, amortizado, probabilístico,etc) para algoritmos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KT05], [DPV06], [RS09], [Tar83], [Raw92]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Als99] H. Alsuwaiyel. *Algorithms: Design Techniques and Analysis*. World Scientific, 1999. ISBN: 9789810237400.
- [DPV06] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill Education, 2006. ISBN: 9780073523408.
- [GT09] Michael T. Goodrich and Roberto Tamassia. *Algorithm Design: Foundations, Analysis and Internet Examples*. 2nd. John Wiley & Sons, Inc., 2009. ISBN: 0470088540, 9780470088548.
- [Knu97] D.E. Knuth. *The Art of Computer Programming: Fundamental algorithms Vol 1*. Third Edition. Addison-Wesley, 1997. ISBN: 9780201896831. URL: <http://www-cs-faculty.stanford/~knuth/taocp.html>.
- [KT05] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN: 0321295358.
- [Raw92] G.J.E. Rawlins. *Compared to What?: An Introduction to the Analysis of Algorithms*. Computer Science Press, 1992. ISBN: 9780716782438.
- [RS09] Thomas H. Cormen; Charles E. Leiserson ; Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009. ISBN: 0262033844.
- [SF13] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms*. Pearson Education, 2013. ISBN: 9780133373486.
- [SW11] R. Sedgewick and K. Wayne. *Algorithms*. Pearson Education, 2011. ISBN: 9780132762564.
- [Tar83] Robert Endre Tarjan. *Data Structures and Network Algorithms*. Society for Industrial and Applied Mathematics, 1983. ISBN: 0-89871-187-8.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52090. Gerenciamiento de Datos II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	2
2.2 Horas de teoría	:	-
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	64 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52091. Gerenciamiento de Datos I. (4 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La Gestión de la Información (*IM-Information Management*) juega un rol principal en casi todas las áreas donde los computadores son usados. Esta área incluye la captura, digitalización, representación, organización, transformación y presentación de información; algoritmos para mejorar la eficiencia y efectividad del acceso y actualización de información almacenada, modelamiento de datos y abstracción, y técnicas de almacenamiento de archivos físicos.

Este también abarca la seguridad de la información, privacidad, integridad y protección en un ambiente compartido. Los estudiantes necesitan ser capaces de desarrollar modelos de datos conceptuales y físicos, determinar que métodos de IM y técnicas son apropiados para un problema dado, y ser capaces de seleccionar e implementar una apropiada solución de IM que refleje todas las restricciones aplicables, incluyendo escalabilidad y usabilidad.

5. OBJETIVOS

- Hacer que el alumno entienda las diferentes aplicaciones que tienen las bases de datos, en las diversas áreas de conocimiento.
- Mostrar las formas adecuadas de almacenamiento de información basada en sus diversos enfoques y su posterior recuperación de información.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Usar**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Usar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Usar**)

7. TEMAS

Unidad 1: Diseño Físico de Bases de Datos (10 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Almacenamiento y estructura de archivos.• Archivos indexados.• Archivos Hash.• Archivos de Firma.• Árboles B.• Archivos con índice denso.• Archivos con registros de tamaño variable.• Eficiencia y Afinación de Bases de Datos.	<ul style="list-style-type: none">• Explica los conceptos de registro, tipos de registro, y archivos, así como las diversas técnicas para colocar registros de archivos en un disco [Usar]• Da ejemplos de la aplicación de índices primario, secundario y de agrupamiento [Usar]• Distingue entre un índice no denso y uno denso [Usar]• Implementa índices de multinivel dinámicos usando árboles-B [Usar]• Explica la teoría y la aplicación de técnicas de hash internas y externas [Usar]• Usa técnicas de hasp para facilitar la expansión de archivos dinámicos [Usar]• Describe las relaciones entre hashing, compresión, y búsquedas eficientes en bases de datos [Usar]• Evalúa el costo y beneficio de diversos esquemas de hashing [Usar]• Explica como el diseño físico de una base de datos afecta la eficiencia de las transacciones en ésta [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Bur04], [Cel05]	

Unidad 2: Procesamiento de Transacciones (12 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Transacciones.• Fallo y recuperación.• Control concurrente.• Interacción de gestión de transacciones con el almacenamiento, especialmente en almacenamiento.	<ul style="list-style-type: none">• Crear una transacción mediante la incorporación de SQL en un programa de aplicación [Usar]• Explicar el concepto de confirmaciones implícitas [Usar]• Describir los problemas específicos para la ejecución de una transacción eficiente [Usar]• Explicar cuando y por qué se necesita un <i>rollback</i>, y cómo registrar todo asegura un <i>rollback</i> adecuado [Usar]• Explicar el efecto de diferentes niveles de aislamiento sobre los mecanismos de control de concurrencia [Usar]• Elegir el nivel de aislamiento adecuado para la aplicación de un protocolo de transacción especificado [Usar]• Identificar los límites apropiados de la transacción en programas de aplicación [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Phi97], [Ram04]	

Unidad 3: Almacenamiento y Recuperación de Información (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Documentos, publicación electrónica, markup, y lenguajes markup. • Tries, archivos invertidos, Árboles PAT, archivos de firma, indexación. • Análisis Morfológico, stemming, frases, stop lists. • Distribuciones de frecuencia de términos, incertidumbre, fuzificación (fuzzyness), ponderación. • Espacio vectorial, probabilidad, lógica, y modelos avanzados. • Necesidad de Información , Relevancia, evaluación, efectividad. • Thesauri, ontologías, clasificación y categorización, metadata. • Información bibliográfica, bibliometría, citasiones. • Enrutamiento y filtrado. • Búsqueda multimedia. • Información de resumen y visualización. • Búsqueda por facetas (por ejemplo, el uso de citas, palabras clave, esquemas de clasificación). • Librerías digitales. • Digitalización, almacenamiento, intercambio, objetos digitales, composición y paquetes. • Metadata y catalogación. • Nombramiento, repositorios, archivos • Archivamiento y preservación, integrdad • Espacios (Conceptual, geográfico, 2/3D, Realidad virtual) • Arquitecturas (agentes, autobuses, envolturas / mediadores), de interoperabilidad. • Servicios (búsqueda, de unión, de navegación, y así sucesivamente). • Gestión de derechos de propiedad intelectual, la privacidad y la protección (marcas de agua). 	<ul style="list-style-type: none"> • Explica los conceptos básicos de almacenamiento y recuperación de la información [Usar] • Describe que temas son específicos para una recuperación de la información eficiente [Usar] • Da aplicaciones de estrategias alternativas de búsqueda y explica porqué una estrategia en particular es apropiada para una aplicación [Usar] • Diseña e implementa un sistema de almacenamiento y recuperación de la información o librería digital de tamaño pequeño a mediano [Usar] • Describe algunas de las soluciones técnicas a los problemas relacionados al archivamiento y preservación de la información en una librería digital [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pet98], [Ram04]	

Unidad 4: Bases de Datos Distribuidas (36 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • DBMS Distribuidas <ul style="list-style-type: none"> – Almacenamiento de datos distribuido – Procesamiento de consultas distribuido – Modelo de transacciones distribuidas – Soluciones homogéneas y heterogéneas – Bases de datos distribuidas cliente-servidor • Parallel DBMS <ul style="list-style-type: none"> – Arquitecturas paralelas DBMS: memoria compartida, disco compartido, nada compartido; – Aceleración y ampliación, por ejemplo, el uso del modelo de procesamiento MapReduce – Replicación de información y modelos de consistencia débil 	<ul style="list-style-type: none"> • Explicar las técnicas usadas para la fragmentación de datos, replicación, y la asignación durante el proceso de diseño de base de datos distribuida [Usar] • Evaluar estrategias simples para la ejecución de una consulta distribuida para seleccionar una estrategia que minimice la cantidad de transferencia de datos [Usar] • Explicar como el protocolo de dos fases de <i>commit</i> es usado para resolver problemas de transacciones que acceden a bases de datos almacenadas en múltiples nodos [Usar] • Describir el control concurrente distribuido basados en técnicas de copia distinguidos y el método de votación. [Usar] • Describir los tres niveles del software en el modelo cliente servidor [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [M T99]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Bur04] Donald K. Burleson. *Physical Database Design Using Oracle*. CRC Press, 2004.
 [Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

- [M T99] Patrick Valduriez M. Tamer Ozsü. *Principles of Distributed Database Systems, Second Edition*. Prentice Hall, 1999.
- [Pet98] Julita Vassileva Peter Brusilovsky Alfred Kobsa. *Adaptive Hypertext and Hypermedia, First Edition*. Springer, 1998.
- [Phi97] Eric Newcomer Philip A. Bernstein. *Principles of Transaction Processing, First Edition*. Morgan Kaufmann, 1997.
- [Ram04] Shamkant B. Navathe Ramez Elmasri. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

ENG42012. Inglés V (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	ENG42009. English IV. (4 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El aprendizaje del idioma inglés en los estudiantes de nivel superior se ha convertido en elemento necesario para su desarrollo académico, personal y profesional. La mayor cantidad de literatura académica de los diversos campos del saber es redactada en inglés, lo cual garantiza que las personas que cuenten con el dominio del idioma puedan siempre estar actualizados.

Asimismo, el conocimiento de este idioma permite tener perspectivas sociales y culturales más amplias. En ese sentido, para un efectivo aprendizaje de la lengua es necesario el desarrollo de las cuatro (4) habilidades: escuchar, hablar, leer y escribir considerando los lineamientos del Marco Común Europeo de Referencia de Lengua - MCERL.

5. OBJETIVOS

- Comunicarse e intercambiar información de manera limitada en situaciones predecibles de forma general.
- Alcanzar el nivel B1 según el MCERL.
- Manejar terminología propia del área de estudios.

6. COMPETENCIAS

CG2) Comunicación bilingüe. (**Evaluar**)

7. TEMAS

Unidad 1: I had an accident (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Past continuous. • Past simple. • Information questions in the past. • Connector (effect). 	<ul style="list-style-type: none"> • Describe an accident.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 2: I might be right (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Adverbs Modals - must/can't deduction. • Modals – might, may, will, probably. • Connector (effect). 	<ul style="list-style-type: none"> • Talk about deductions, probabilities.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 3: Call of duty (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Modals: must/have to. • Connector (effect). 	<ul style="list-style-type: none"> • Talk about obligations, possibilities.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 4: My favorite artist (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Simple passive. • Connector (effect). 	<ul style="list-style-type: none"> • Describe art. • Express preferences.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 5: I have been there (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Present perfect. • Present perfect/past simple. • Adverbs. 	<ul style="list-style-type: none"> • Talk about experiences.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 6: Can you speak English? (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Design by the teacher. • Connector (effect). 	<ul style="list-style-type: none"> • Get familiar with technical expressions of the field..
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[De 02] Consejo De Europa. "Marco común europeo de referencia para las lenguas". In: *Strasburgo: Consejo de Europa, Ministerio de Educación, Cultura y Deporte/Instituto Cervantes* (2002).



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

FIS41001. Física General (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : 2 (Semanal)
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : MAC41010. Fundamentos de Cálculo. (1^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Un aspecto muy importante en el nivel universitario lo constituye el cálculo diferencial, aspecto que constituye la piedra angular de las posteriores asignaturas de matemáticas así como de la utilidad de la matemática en la solución de problemas aplicados a la ciencia y la tecnología. Cualquier profesional con rango universitario debe por lo tanto tener conocimiento amplio de esta asignatura, pues se convertirá en su punto de partida para los intereses de su desarrollo profesional; así también será soporte para no tener dificultades en las asignaturas de matemática y física de toda la carrera.

5. OBJETIVOS

- Asimilar y manejar los conceptos de función, sucesión y relacionarlos con los de límites y continuidad.
- Describir, analizar, diseñar y formular modelos continuos que dependan de una variable.
- Conocer y manejar la propiedades del cálculo diferencial y aplicarlas a la resolución de problemas.

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Atomic structure• Periodic Table and electronic properties	<ul style="list-style-type: none">• Comprender la importancia del sistema de los números reales (construcción), manipular los axiomas algebraicos y de orden [Evaluar].• Comprender el concepto de función. Manejar dominios, operaciones, gráficas, inversas [Evaluar].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Leithold2000], [Stewart], [ThomasFinney]	

Unidad 2: (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Stoichiometry • Chemical bonding and formation of molecules and materials. 	<ul style="list-style-type: none"> • Entender el concepto de sucesión y su importancia [Evaluar]. • Conocer los principales tipos de sucesiones, manejar sus propiedades [Evaluar]. • Manejar y calcular límites de sucesiones [Evaluar].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Leithold2000], [ThomasFinney]	

Unidad 3: (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Límites • Continuidad • Aplicaciones de funciones continuas. Teorema del valor intermedio 	<ul style="list-style-type: none"> • Comprender el concepto de límite. calcular límites [Evaluar]. • Analizar la continuidad de una función [Evaluar]. • Aplicar el teorema del valor intermedio [Evaluar].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Leithold2000], [Leithold2000], [Stewart]	

Unidad 4: (22 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Definición. reglas de derivación • Incrementos y diferenciales • Regla de la cadena. Derivación implícita 	<ul style="list-style-type: none"> • Comprender el concepto de derivada e interpretarlo [Evaluar]. • Manipular las reglas de derivación [Evaluar].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Leithold2000], [ThomasFinney], [Stewart]	

Unidad 5: (22 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Definición. reglas de derivación Incrementos y diferenciales Regla de la cadena. Derivación implícita 	<ul style="list-style-type: none"> Comprender el concepto de derivada e interpretarlo [Evaluar]. Manipular las reglas de derivación [Evaluar].
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [Leithold2000], [ThomasFinney], [Stewart]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

MAR52008. Marketing (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 4 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : 40cr

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Write justification for this course here ...

5. OBJETIVOS

- Write your first goal here.
- Write your second goal here.
- Just in case you need more goals write them here

6. COMPETENCIAS

CG4) Gestiona recursos. (Evaluar)

7. TEMAS

Unidad 1: title for the unit goes here (0 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Topic1• Topic2• Topic3	<ul style="list-style-type: none">• Learning outcome1 [Levelforthislearningoutcome].• i.e Apply computing in complex problems [Usar].• Create a search engine [Evaluar].• Study data structures [Familiarizarse].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Bibitem1], [Bibitem2]	

Unidad 2: another unit goes here (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Topic1 	<ul style="list-style-type: none"> • Learning outcome xyz [Levelforthislearningoutcome].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Bibitem3], [Bibitem1]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52082. Sistemas Inteligentes (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	2
2.2 Horas de teoría	:	-
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	64 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	EST41005. Estadística Descriptiva y Probabilidades. (4 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La investigación en Inteligencia Artificial ha conducido al desarrollo de numerosas técnicas relevantes, dirigidas a la automatización de la inteligencia humana, dando una visión panorámica de diferentes algoritmos que simulan los diferentes aspectos del comportamiento y la inteligencia del ser humano.

5. OBJETIVOS

- Evaluar las posibilidades de simulación de la inteligencia, para lo cual se estudiarán las técnicas de modelización del conocimiento.
- Construir una noción de inteligencia que soporte después las tareas de su simulación.

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Usar**)
- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (**Usar**)

7. TEMAS

Unidad 1: Cuestiones fundamentales (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial. • ¿Qué es comportamiento inteligente? <ul style="list-style-type: none"> – El Test de Turing – Razonamiento Racional versus No Racional • Características del Problema: <ul style="list-style-type: none"> – Observable completamente versus observable parcialmente – Individual versus multi-agente – Determinístico versus estocástico – Estático versus dinámico – Discreto versus continuo • Naturaleza de agentes: <ul style="list-style-type: none"> – Autónomo versus semi-autónomo – Reflexivo, basado en objetivos, y basado en utilidad – La importancia en percepción e interacciones con el entorno • Cuestiones filosóficas y éticas. 	<ul style="list-style-type: none"> • Describir el test de Turing y el experimento pensado "cuarto chino" (<i>Chinese Room</i>) [Usar] • Determinando las características de un problema dado que sistemas inteligentes deberían resolver [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 06], [Pon+14]	

Unidad 2: Agentes (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Definición de Agentes • Arquitectura de agentes (Ej. reactivo, en capa, cognitivo) • Teoría de agentes • Racionalidad, teoría de juegos: <ul style="list-style-type: none"> – Agentes de decisión teórica – Procesos de decisión de Markov (MDP) • Agentes de Software, asistentes personales, y acceso a información: <ul style="list-style-type: none"> – Agentes colaborativos – Agentes de recolección de información – Agentes creíbles (carácter sintético, modelamiento de emociones en agentes) • Agentes de aprendizaje • Sistemas Multi-agente <ul style="list-style-type: none"> – Agentes Colaborativos – Equipos de Agentes – Agentes Competitivos (ej., subastas, votaciones) – Sistemas de enjambre y modelos biológicamente inspirados 	<ul style="list-style-type: none"> • Lista las características que definen un agente inteligente [Usar] • Describe y contrasta las arquitecturas de agente estándares [Usar] • Describe las aplicaciones de teoría de agentes para dominios como agentes de software, asistentes personales, y agentes creíbles [Usar] • Describe los paradigmas primarios usados por agentes de aprendizaje [Usar] • Demuestra mediante ejemplos adecuados como los sistemas multi-agente soportan interacción entre agentes [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Nil01], [RN03], [Pon+14]	

Unidad 3: Estrategias de búsquedas básicas (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Espacios de Problemas (estados, metas y operadores), solución de problemas mediante búsqueda. • Factored representation (factoring state hacia variables) • Uninformed search (breadth-first, depth-first, depth-first with iterative deepening) • Heurísticas y búsqueda informada (hill-climbing, generic best-first, A*) • El espacio y el tiempo de la eficiencia de búsqueda. • Dos jugadores juegos (introducción a la búsqueda minimax). • Satisfacción de restricciones (backtracking y métodos de búsqueda local). 	<ul style="list-style-type: none"> • Formula el espacio eficiente de un problema para un caso expresado en lenguaje natural (ejm. Inglés) en términos de estados de inicio y final, así como sus operadores [Usar] • Describe el rol de las heurísticas y describe los intercambios entre completitud, óptimo, complejidad de tiempo, y complejidad de espacio [Usar] • Describe el problema de la explosión combinatoria del espacio de búsqueda y sus consecuencias [Usar] • Compara y contrasta tópicos de búsqueda básica con temas jugabilidad de juegos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Nil01], [Pon+14]	

Unidad 4: Búsqueda Avanzada (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Búsqueda estocástica: <ul style="list-style-type: none"> – Simulated annealing – Algoritmos genéticos – Búsqueda de árbol Monte-Carlo • Construcción de árboles de búsqueda, espacio de búsqueda dinámico, explosión combinatoria del espacio de búsqueda. • Implementación de búsqueda A *, búsqueda en haz. • Búsqueda Minimax, poda alfa-beta. • Búsqueda Expectimax (MDP-Solving) y los nodos de azar. 	<ul style="list-style-type: none"> • Diseñar e implementar una solución a un problema con algoritmo genético [Usar] • Diseñar e implementar un esquema de recocido simulado (<i>simulated annealing</i>) para evitar mínimos locales en un problema [Usar] • Diseñar e implementar una búsqueda A* y búsqueda en haz (<i>beam search</i>) para solucionar un problema [Usar] • Aplicar búsqueda minimax con poda alfa-beta para simplificar el espacio de búsqueda en un juego con dos jugadores [Usar] • Comparar y contrastar los algoritmos genéticos con técnicas clásicas de búsqueda [Usar] • Comparar y contrastar la aplicabilidad de varias heurísticas de búsqueda, para un determinado problema [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Gol89], [Nil01], [RN03], [Pon+14]	

Unidad 5: Razonamiento Bajo Incertidumbre (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Revisión de Probabilidad Básica • Variables aleatorias y distribuciones de probabilidad: <ul style="list-style-type: none"> – Axiomas de probabilidad – Inferencia probabilística – Regla de Bayes • Independencia Condicional • Representaciones del conocimiento: <ul style="list-style-type: none"> – Redes bayesianas <ul style="list-style-type: none"> * Inferencia exacta y su complejidad * Métodos de Muestreo aleatorio (Monte Carlo) (p.e. Muestreo de Gibbs) – Redes Markov – Modelos de probabilidad relacional – Modelos ocultos de Markov 	<ul style="list-style-type: none"> • Aplicar la regla de Bayes para determinar el cumplimiento de una hipótesis [Usar] • Explicar cómo al tener independencia condicional permite una gran eficiencia en sistemas probabilísticos [Usar] • Identificar ejemplos de representación de conocimiento para razonamiento bajo incertidumbre [Usar] • Indicar la complejidad de la inferencia exacta. Identificar métodos para inferencia aproximada [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KF09], [RN03]	

Unidad 6: Aprendizaje Automático Básico (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Definición y ejemplos de la extensa variedad de tareas de aprendizaje de máquina, incluida la clasificación. • Aprendizaje inductivo • Aprendizaje simple basado en estadísticas, como el clasificador ingenuo de Bayes, árboles de decisión. • El problema exceso de ajuste. • Medición clasificada con exactitud. 	<ul style="list-style-type: none"> • Listar las diferencias entre los tres principales tipos de aprendizaje: supervisado, no supervisado y por refuerzo [Usar] • Identificar ejemplos de tareas de clasificación, considerando las características de entrada disponibles y las salidas a ser predecidas [Usar] • Explicar la diferencia entre aprendizaje inductivo y deductivo [Usar] • Describir el sobre ajuste (<i>overfitting</i>) en el contexto de un problema [Usar] • Aplicar un algoritmo de aprendizaje estadístico simple como el Clasificador Naive Bayesiano e un problema de clasificación y medirla precisión del clasificador [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Mit98], [RN03], [Pon+14]	

Unidad 7: Aprendizaje de máquina avanzado (20 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Definición y ejemplos de una amplia variedad de tareas de aprendizaje de máquina• Aprendizaje general basado en estadística, estimación de parámetros (máxima probabilidad)• Programación lógica inductiva (<i>Inductive logic programming ILP</i>)• Aprendizaje supervisado<ul style="list-style-type: none">– Aprendizaje basado en árboles de decisión– Aprendizaje basado en redes neuronales– Aprendizaje basado en máquinas de soporte vectorial (<i>Support vector machines SVMs</i>)• Aprendizaje y <i>clustering</i> no supervisado<ul style="list-style-type: none">– EM– K-means– Mapas auto-organizados• Aprendizaje semi-supervisado.• Aprendizaje de modelos gráficos• Evaluación del desempeño (tal como cross-validation, area bajo la curva ROC)• Aplicación de algoritmos Machine Learning para Minería de datos.	<ul style="list-style-type: none">• Explica las diferencias entre los tres estilos de aprendizaje: supervisado, por refuerzo y no supervisado [Usar]• Implementa algoritmos simples para el aprendizaje supervisado, aprendizaje por refuerzo, y aprendizaje no supervisado [Usar]• Determina cuál de los tres estilos de aprendizaje es el apropiado para el dominio de un problema en particular [Usar]• Compara y contrasta cada una de las siguientes técnicas, dando ejemplo de cuando una estrategia es la mejor: árboles de decisión, redes neuronales, y redes bayesianas [Usar]• Evalúa el rendimiento de un sistema de aprendizaje simple en un conjunto de datos reales [Usar]• Describe el estado del arte en la teoría del aprendizaje, incluyendo sus logros y limitantes [Usar]• Explica el problema del sobreajuste, conjuntamente con técnicas para determinar y manejar el problema [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [RN03], [KF09], [Mur12]	

Unidad 8: Procesamiento del Lenguaje Natural (12 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Gramaticas determinísticas y estocásticas• Algoritmos de parseo<ul style="list-style-type: none">– Gramáticas libres de contexto (CFGs) y cuadros de parseo (e.g. Cocke-Younger-Kasami CYK)– CFGs probabilísticos y ponderados CYK• Representación del significado / Semántica<ul style="list-style-type: none">– Representación de conocimiento basado en lógica– Roles semánticos– Representaciones temporales– Creencias, deseos e intenciones• Metodos basados en el corpus• N-gramas y Modelos ocultos de Markov (HMMs)• Suavizado y back-off• Ejemplos de uso: POS etiquetado y morfología• Recuperación de la información:<ul style="list-style-type: none">– Modelo de espacio vectorial<ul style="list-style-type: none">* TF & IDF– Precision y cobertura• Extracción de información• Traducción de lenguaje• Clasificación y categorización de texto:<ul style="list-style-type: none">– Modelo de bolsa de palabras	<ul style="list-style-type: none">• Define y contrasta gramáticas de tipo estocásticas y determinísticas, dando ejemplos y demostrando como adecuar cada una de ellas [Usar]• Simula, aplica, o implementa algoritmos clásicos y estocásticos para el parseo de un lenguaje natural [Usar]• Identifica los retos de la representación del significado [Usar]• Lista las ventajas de usar corpus estándares. Identifica ejemplos de corpus actuales para una variedad de tareas de PLN [Usar]• Identifica técnicas para la recuperación de la información, traducción de lenguajes, y clasificación de textos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Nil01], [RN03], [Pon+14]	

Unidad 9: Visión y percepción por computador (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión Computacional <ul style="list-style-type: none"> – Adquisición de imágenes, representación, procesamiento y propiedades – Representación de formas, reconocimiento y segmentación de objetos – Análisis de movimiento • Modularidad en reconocimiento. • Enfoques de reconocimiento de patrones <ul style="list-style-type: none"> – Algoritmos de clasificación y medidas de calidad de la clasificación. – Técnicas estadísticas. 	<ul style="list-style-type: none"> • Resumir la importancia del reconocimiento de imágenes y objetos en Inteligencia Artificial (AI) e indicar varias aplicaciones significativas de esta tecnología [Usar] • Listar al menos tres aproximaciones de segmentación de imágenes, tales como algoritmos de límites (thresholding), basado en el borde y basado en regiones, junto con sus características definitorias, fortalezas y debilidades [Usar] • Implementar reconocimiento de objetos en 2d basados en la representación del contorno y/o regiones basadas en formas [Usar] • Proporcionar al menos dos ejemplos de transformación de una fuente de datos de un dominio sensorial a otro, ejemplo, datos táctiles interpretados como imágenes en 2d de una sola banda [Usar] • Implementar un algoritmo para la extracción de características en información real, ejemplo, un detector de bordes o esquinas para imágenes o vectores de coeficientes de Fourier describiendo una pequeña porción de señal de audio [Usar] • Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Usar] • Evaluar el desempeño de la función de extracción subyacente, en relación con al menos una aproximación alternativa posible (ya sea implementado o no) en su contribución a la tarea de clasificación (8) anterior [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Nil01], [RN03], [Pon+14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. CRC Press, 2006.
- [Gol89] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN: 0262013193.
- [Mit98] M. Mitchell. *An introduction to genetic algorithms*. The MIT press, 1998.
- [Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 0262018020.
- [Nil01] Nils Nilsson. *Inteligencia Artificial: Una nueva visión*. McGraw-Hill, 2001.
- [Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.
- [RN03] Stuart Russell and Peter Norvig. *Inteligencia Artificial: Un enfoque moderno*. Prentice Hall, 2003.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52083. Sistemas de Infomación (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 3
- 2.2 Horas de teoría : 1 (Semanal)
- 2.3 Horas de práctica : 2 (Semanal)
- 2.4 Horas autónomas : 96 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : SFW52088. Ingeniería de Software I. (5^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Analizar técnicas para la correcta implementación de Sistemas de Información escalables, robustos, confiables y eficientes en las organizaciones.

5. OBJETIVOS

- Implementar de forma correcta (escalables, robustos, confiables y eficientes) Sistemas de Información en las organizaciones.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Usar**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Usar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Usar**)

7. TEMAS

Unidad 1: Introducción (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Introducción a la gestión de la información• Software para gestión de información.• Tecnología para gestión de información.	<ul style="list-style-type: none">• Aplicar correctamente la tecnología para la gestión de la información [Evaluar]
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Som17], [PM15], [LL17]	

Unidad 2: Estrategia (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Estrategia para gestión de información • Estrategia para gestión conocimiento • Estrategia para sistema de información. 	<ul style="list-style-type: none"> • Aplicar y evaluar correctamente estrategias de gestión [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Som17], [PM15]	

Unidad 3: Implementación (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Gestión de desarrollo de sistemas de información. • Gestión del cambio • Arquitectura de Información 	<ul style="list-style-type: none"> • Aplicar y evaluar correctamente estrategias de implementación [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Som17], [PM15]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[LL17] Kenneth C. Laudon and Jane P. Laudon. *Management Information Systems: Managing the Digital Firm*. 15th. Pearson, Mar. 2017.

- [PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.
- [Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52084. Estructuras de Datos Avanzadas (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52089. Análisis y Diseño de Algoritmos. (5 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Los algoritmos y estructuras de datos son una parte fundamental de la ciencia de la computación que nos permiten organizar la información de una manera más eficiente, por lo que es importante para todo profesional del área tener una sólida formación en este aspecto.

En el curso de estructuras de datos avanzadas nuestro objetivo es que el alumno conozca y analice estructuras complejas, como los Métodos de Acceso Multidimensional, Métodos de Acceso Espacio-Temporal y Métodos de Acceso Métrico, etc.

5. OBJETIVOS

- Que el alumno entienda, diseñe, implemente, aplique y proponga estructuras de datos innovadoras para solucionar problemas relacionados al tratamiento de datos multidimensionales, recuperación de información por similitud, motores de búsqueda y otros problemas computacionales.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Usar**)
- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Usar**)

7. TEMAS

Unidad 1: Técnicas Básicas de Implementación de Estructuras de Datos (16 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Programación estructurada • Programación Orientada a Objetos • Tipos Abstractos de Datos • Independencia del lenguaje de programación del usuario de la estructura • Independencia de Plataforma • Control de concurrencia • Protección de Datos • Niveles de encapsulamiento (struct, class, namespace, etc) 	<ul style="list-style-type: none"> • Que el alumno entienda las diferencias básicas que involucran las distintas técnicas de implementación de estructuras de datos[Usar] • Que el alumno analice las ventajas y desventajas de cada una de las técnicas existentes[Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cua+04], [Knu07a], [Knu07b], [Gam+94], [Bjö18], [Dav18]	

Unidad 2: Métodos de Acceso Multidimensionales (16 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Métodos de Acceso para datos puntuales • Métodos de Acceso para datos no puntuales • Problemas relacionados con el aumento de dimensión 	<ul style="list-style-type: none"> • Que el alumno entienda conozca e implemente algunos Métodos de Acceso para datos multidimensionales y espacio temporales[Usar] • Que el alumno entienda el potencial de estos Métodos de Acceso en el futuro de las bases de datos comerciales[Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Sam06], [Gü98]	

Unidad 3: Métodos de Acceso Métrico (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Métodos de Acceso Métrico para distancias discretas • Métodos de Acceso Métrico para distancias continuas 	<ul style="list-style-type: none"> • Que el alumno entienda conozca e implemente algunos métodos de acceso métrico[Usar] • Que el alumno entienda la importancia de estos Métodos de Acceso para la Recuperación de Información por Similitud[Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Sam06], [Chá+01], [Tra+00], [Zez+07]	

Unidad 4: Métodos de Acceso Aproximados (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Space Filling Curves • Locality Sensitive Hashing 	<ul style="list-style-type: none"> • Que el alumno entienda conozca e implemente algunos métodos de acceso aproximados[Usar] • Que el alumno entienda la importancia de estos Métodos de Acceso para la Recuperación de Información por Similitud en entornos donde la Escalabilidad sea una factor muy importante[Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Sam06], [PI06], [Zez+07]	

Unidad 5: Seminarios (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Métodos de Acceso Espacio Temporal • Estructuras de Datos con programación genérica 	<ul style="list-style-type: none"> • Que el alumno pueda discutir sobre los últimos avances en métodos de acceso para distintos dominios de conocimiento[Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Sam06], [Chá+01]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Bjö18] Stefan Björnander. *C++17 By Example: Practical projects to get you up and running with C++17*. Packt Publishing, Feb. 2018.
- [Chá+01] E. Chávez et al. “Proximity Searching in Metric Spaces”. In: *ACM Computing Surveys* 33.3 (Sept. 2001), pp. 273–321.
- [Cua+04] Ernesto Cuadros-Vargas et al. “Implementing data structures: An incremental approach”. <http://socios.spc.org.pe/ecuadros/cursos/pdfs/>. 2004.
- [Dav18] Doug Gregor David Vandevoorde Nicolai M. Josuttis. *C++ Templates: The Complete Guide*. Addison-Wesley Professional, Sept. 2018. URL: <http://informit.com/aw>.
- [Gam+94] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Computing Series. ISBN-10: 0201633612. Addison-Wesley Professional, Nov. 1994.
- [Gü98] Volker Gaede and Oliver ünther. “Multidimensional Access Methods”. In: *ACM Computing Surveys* 30.2 (1998), pp. 170–231.
- [Knu07a] Donald Ervin Knuth. *The Art of Computer Programming, Fundamental Algorithms*. 3rd. Vol. I. 0-201-89683-4. Addison-Wesley, Feb. 2007.
- [Knu07b] Donald Ervin Knuth. *The Art of Computer Programming, Sorting and Searching*. 2nd. Vol. II. 0-201-89685-0. Addison-Wesley, Feb. 2007.
- [PI06] Trevor Darrell PGregory Shakhnarovich and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. 1st. ISBN 0-262-19547-X. MIT Press, Mar. 2006.
- [Sam06] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Illustrated. Elsevier/Morgan Kaufmann, Aug. 2006. ISBN: 9780123694461. URL: <http://books.google.com.pe/books?id=v0-NRRKHG84C>.
- [Tra+00] C. Traina Jr et al. “Slim-Trees: High Performance Metric Trees Minimizing Overlap between Nodes”. In: *Advances in Database Technology - EDBT 2000, 6th International Conference on Extending Database Technology*. Vol. 1777. Lecture Notes in Computer Science. Konstanz, Germany: Springer, Mar. 2000, pp. 51–65.
- [Zez+07] Pavel Zezula et al. *Similarity Search: The Metric Space Approach*. 1st. ISBN-10: 0387291466. Springer, Nov. 2007.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52085. Programación Competitiva (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 2
2.2 Horas de teoría	: -
2.3 Horas de práctica	: 2 (Semanal)
2.4 Horas autónomas	: 64 (horas)
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Obligatorio
2.7 Modalidad	: Presencial
2.8 Prerrequisitos	: SFW52089. Análisis y Diseño de Algoritmos. (5 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La Programación Competitiva combina retos de solucionar problemas con el añadido de poder competir con otras personas. Enseña a los participantes a pensar más rápido y desarrollar habilidades para resolver problemas, que son de gran demanda en la industria. Este curso enseñará la resolución de problemas algorítmicos de manera rápida combinando la teoría de algoritmos y estructuras de datos con la práctica la solución de los problemas.

5. OBJETIVOS

- Que el alumno utilice técnicas de estructuras de datos y algoritmos complejos.
- Que el alumno aplique los conceptos aprendidos para la aplicación sobre un problema real.
- Que el alumno investigue la posibilidad de crear un nuevo algoritmo y/o técnica nueva para resolver un problema real.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinares en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Usar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Usar**)

7. TEMAS

Unidad 1: Introducción (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Introducción a la Programación competitiva • Modelo computacional • Complejidad algorítmica • Problemas sobre búsqueda y ordenamiento • Recursión y recurrencia • Estrategia divide y conquista 	<ul style="list-style-type: none"> • Reconocer y saber como usar los recursos del modelo de computación RAM (Random Access Machine). [Usar] • Determinar el tiempo y espacio de complejidad de algoritmos. [Usar] • Determinar relaciones de recurrencia para algoritmos recursivos.[Usar] • Resolver problemas de búsqueda y ordenamiento.[Usar] • Aprender a seleccionar los algoritmos adecuados para problemas de tipo divide y conquista.[Usar] • Diseñar nuevos algoritmos para la resolución de problemas.[Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 2: Estructuras de datos (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Problemas sobre arrays y strings • Problemas sobre listas enlazadas • Problemas sobre pilas, colas • Problemas sobre arboles • Problemas sobre Hash tables • Problemas sobre Heaps 	<ul style="list-style-type: none"> • Reconocer las distintas estructuras de datos sus complejidades usos y restricciones. [Usar] • Identificar el tipo de estructura de datos adecuado a la resolución del problema. [Usar] • Reconocer tipos de problemas asociado a operaciones sobre estructuras de datos como búsqueda, inserción, eliminación y actualización.[Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 3: Paradigmas de diseño (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Fuerza bruta • Divide y conquista • Backtracking • Greedy • Programación Dinamica 	<ul style="list-style-type: none"> • Aprender los distintos paradigmas de resolución de problemas.[Usar] • Aprender a seleccionar los algoritmos adecuados para distintos problemas según el tipo de paradigma.[Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 4: Gráfos (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Recorrido de gráfos • Aplicaciones y problemas sobre gráfos • Camino mas corto • Redes y flujos 	<ul style="list-style-type: none"> • Identificar problemas clasificados como problemas de grafos. [Usar] • Aprender a seleccionar los algoritmos adecuados para problemas de grafos (recorrido, MST, camino mas costo, redes y flujos) y conocer sus soluciones eficientes. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 5: Tópicos avanzados (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Teoria de números • Probabilidad y combinaciones • Algoritmos para manejos de strings (tries, string hashing, z-algorithm) • Geometria y sweep line algorithms, segment trees 	<ul style="list-style-type: none"> • Aprender a elegir los algoritmos adecuados para problemas sobre teoria de números y matemáticas ya que son importantes en programación competitiva. [Usar] • Aprender a seleccionar los algoritmos adecuados para problemas sobre probabilidades y combinaciones, manejos de strings y geometría computacional. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 6: Problemas de dominio específico (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Latencia y rendimiento • Paralelismo • Redes • Almacenamiento • Alta disponibilidad • Caching • Proxies • Equilibradores de carga • Almacenamiento clave-valor • Replicar y compartir • Elección del líder • Limitación de la tasa • Registro y monitoreo 	<ul style="list-style-type: none"> • Aprender a diseñar sistemas para diferentes problemas de dominio específico aplicando conocimiento sobre redes, computación distribuida, alta disponibilidad, almacenamiento y arquitectura de sistemas. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [ALP12] A. Aziz, T.H. Lee, and A. Prakash. *Elements of Programming Interviews: The Insiders' Guide*. ElementsOfProgrammingInterviews.com, 2012. ISBN: 9781479274833. URL: <https://books.google.com.pe/books?id=y6FLBQAAQBAJ>.
- [Cor+09] T. H. Cormen et al. *Introduction to Algorithms*. MIT Press, 2009.
- [Hal13] Steven Halim. *Competitive Programming*. 3 rd. Lulu, 2013.
- [Kul19] Alexander S. Kulikov. *Learning Algorithms Through Programming and Puzzle Solving*. Active Learning Technologies, 2019.
- [Laa17] Antti Laaksonen. *Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*. Stringer, 2017.
- [Mig03] Steve Skiena Miguel A. Revilla. *Programming Challenges: The Programming Contest Training Manual*. Springer, May 2003. ISBN: 978-0387001630.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52086. Ingeniería de Software II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	1 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	96 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52088. Ingeniería de Software I. (5 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Los tópicos de este curso extienden las ideas del diseño y desarrollo de software desde la secuencia de introducción a la programación para abarcar los problemas encontrados en proyectos de gran escala. Es una visión más amplia y completa de la Ingeniería de Software apreciada desde un punto de vista de Proyectos.

5. OBJETIVOS

- Capacitar a los alumnos para formar parte y definir equipos de desarrollo de software que afronten problemas de envergadura real.
- Familiarizar a los alumnos con el proceso de administración de un proyecto de software de tal manera que sea capaz de crear, mejorar y utilizar herramientas y métricas que le permitan realizar la estimación y seguimiento de un proyecto de software.
- Crear, evaluar e implementar un plan de prueba para segmentos de código de tamaño medio , Distinguir entre los diferentes tipos de pruebas , sentar las bases para crear, mejorar los procedimientos de prueba y las herramientas utilizadas con ese propósito.
- Seleccionar con justificación un apropiado conjunto de herramientas para soportar el desarrollo de un rango de productos de software.
- Crear, mejorar y utilizar los patrones existentes para el mantenimiento de software . Dar a conocer las características y patrones de diseño para la reutilización de software.
- Identificar y discutir diferentes sistemas especializados , crear , mejorar y utilizar los patrones especializados para el diseño , implementación , mantenimiento y prueba de sistemas especializados

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Usar**)

7. TEMAS

Unidad 1: Herramientas y Entornos (12 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Administración de configuración de software y control de versiones.• Administración de despliegues.• Análisis de requerimientos y herramientas para modelado del diseño.• Herramientas de <i>testing</i> incluyendo herramientas de análisis estático y dinámico.• Entornos de programación que automatizan el proceso de construcción de partes de programa (ejem., construcciones automatizadas)<ul style="list-style-type: none">– Integración continua.• Mecanismos y conceptos de herramientas de integración.	<ul style="list-style-type: none">• Administración de configuración de software y control de versiones. [Usar]• Administración de despliegues. [Usar]• Análisis de requerimientos y herramientas para modelado del diseño. [Usar]• Herramientas de <i>testing</i> incluyendo herramientas de análisis estático y dinámico. [Usar]• Entornos de programación que automatizan el proceso de construcción de partes de programa (ejem., construcciones automatizadas)<ul style="list-style-type: none">– Integración continua.[Usar]• Mecanismos y conceptos de herramientas de integración. [Usar]
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unidad 2: Verificación y Validación de Software (12 horas)

Competencias esperadas:

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Verificación y validación de conceptos. • Inspecciones, revisiones, auditorias. • Tipos de pruebas, incluyendo la interfaz humano computador, usabilidad, confiabilidad, seguridad, desempeño para la especificación. • Fundamentos de testeo: <ul style="list-style-type: none"> – Pruebas de Unit, integración, validación y de Sistema – Creación de plan de pruebas y generación de casos de test – Técnicas de test de caja negra y caja blanca – Test de regresión y automatización de pruebas • Seguimiento de defectos. • Limitaciones de testeo en dominios particulares, tales como sistemas paralelos o críticos en cuanto a seguridad. • Enfoques estáticos y enfoques dinámicos para la verificación. • Desarrollo basado en pruebas. • Plan de Validación, documentación para validación. • Pruebas Orientadas a Objetos, Sistema de Pruebas. • Verificación y validación de artefactos no codificados (documentación, archivos de ayuda, materiales de entrenamiento) • Logeo fallido, error crítico y apoyo técnico para dichas actividades. • Estimación fallida y terminación de las pruebas que incluye la envios por defecto. 	<ul style="list-style-type: none"> • Distinguir entre la validación y verificación del programa [Usar] • Describir el papel que las herramientas pueden desempeñar en la validación de software [Usar] • Realizar, como parte de una actividad de equipo, una inspección de un segmento de código de tamaño medio [Usar] • Describir y distinguir entre diferentes tipos y niveles de pruebas (unitaria, integración, sistemas y aceptación) [Usar] • Describir técnicas para identificar casos de prueba representativos para integración, regresión y pruebas del sistema [Usar] • Crear y documentar un conjunto de pruebas para un segmento de código de mediano tamaño [Usar] • Describir cómo seleccionar buenas pruebas de regresión y automatizarlas [Usar] • Utilizar una herramienta de seguimiento de defectos para manejar defectos de software en un pequeño proyecto de software [Usar] • Discutir las limitaciones de las pruebas en un dominio particular [Usar] • Evaluar un banco de pruebas (<i>a test suite</i>) para un segmento de código de tamaño medio [Usar] • Comparar los enfoques estáticos y dinámicos para la verificación [Usar] • Identificar los principios fundamentales de los métodos de desarrollo basado en pruebas y explicar el papel de las pruebas automatizadas en estos métodos [Usar] • Discutir los temas relacionados con las pruebas de software orientado a objetos [Usar] • Describir las técnicas para la verificación y validación de los artefactos de no código [Usar] • Describir los enfoques para la estimación de fallos [Usar] • Estimar el número de fallos en una pequeña aplicación de software basada en la densidad de defectos y siembra de errores [Usar] • Realizar una inspección o revisión del de código fuente de un software para un proyecto de software de tamaño pequeño o mediano [Usar]

Aprendizaje autónomo

- Desarrollo de ejercicios prácticos

Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]

Unidad 3: Evolución de Software (12 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">● Desarrollo de Software en el contexto de código grande pre existente<ul style="list-style-type: none">– Cambios de software– Preocupaciones y ubicación de preocupaciones– <i>Refactoring</i>● Evolución de Software.● Características de Software mantenible.● Sistemas de Reingeniería.● Reuso de Software.<ul style="list-style-type: none">– Segmentos de código– Bibliotecas y <i>frameworks</i>– Componentes– Líneas de Producto	<ul style="list-style-type: none">● Identificar los problemas principales asociados con la evolución del software y explicar su impacto en el ciclo de vida del software [Usar]● Estimar el impacto del cambio de requerimientos en productos existentes de tamaño medio [Usar]● Usar refactorización en el proceso de modificación de un componente de software [Usar]● Estudiar los desafíos de mejorar sistemas en un entorno cambiante [Usar]● Perfilar los procesos de pruebas de regresión y su rol en el manejo de versiones [Usar]● Estudiar las ventajas y desventajas de diferentes tipos de niveles de confiabilidad [Usar]
Aprendizaje autónomo <ul style="list-style-type: none">● Desarrollo de ejercicios prácticos	
Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unidad 4: Gestión de Proyectos de Software (12 horas)

Competencias esperadas:

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> ● La participación del equipo: <ul style="list-style-type: none"> – Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo – Roles y responsabilidades en un equipo de software – Equipo de resolución de conflictos – Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura) ● Estimación de esfuerzo (a nivel personal) ● Riesgo. <ul style="list-style-type: none"> – El papel del riesgo en el ciclo de vida – Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de ● Gestión de equipos: <ul style="list-style-type: none"> – Organización de equipo y la toma de decisiones – Roles de identificación y asignación – Individual y el desempeño del equipo de evaluación ● Gestión de proyectos: <ul style="list-style-type: none"> – Programación y seguimiento de elementos – Herramientas de gestión de proyectos – Análisis de Costo/Beneficio ● Software de medición y técnicas de estimación. ● Aseguramiento de la calidad del software y el rol de las mediciones. ● Riesgo. <ul style="list-style-type: none"> – Identificación de riesgos y gestión. – Análisis riesgo y evaluación. – La tolerancia al riesgo (por ejemplo, riesgo adverso, riesgo neutral, la búsqueda de riesgo) – Planificación de Riesgo ● En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas. 	<ul style="list-style-type: none"> ● Discutir los comportamientos comunes que contribuyen al buen funcionamiento de un equipo [Usar] ● Crear y seguir un programa para una reunión del equipo [Usar] ● Identificar y justificar las funciones necesarias en un equipo de desarrollo de software [Usar] ● Entender las fuentes, obstáculos y beneficios potenciales de un conflicto de equipo [Usar] ● Aplicar una estrategia de resolución de conflictos en un ambiente de equipo [Usar] ● Utilizar un método ad hoc para estimar el esfuerzo de desarrollo del software (ejemplo, tiempo) y comparar con el esfuerzo actual requerido [Usar] ● Listar varios ejemplos de los riesgos del software [Usar] ● Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Usar] ● Describir las diferentes categorías de riesgo en los sistemas de software [Usar] ● Demostrar a través de la colaboración de proyectos de equipo los elementos centrales de la construcción de equipos y gestión de equipos [Usar]
<p>Aprendizaje autónomo</p> <ul style="list-style-type: none"> ● Desarrollo de ejercicios prácticos 	
<p>Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]</p>	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Amb01] Vincenzo Ambriola. *Software Process Technology*. Springer, July 2001.
- [Blu92] Bruce I. Blum. *Software Engineering: A Holistic View*. 7th. Oxford University Press US, May 1992.
- [Con00] R. Conradi. *Software Process Technology*. Springer, Mar. 2000.
- [Key04] Jessica Keyes. *Software Configuration Management*. CRC Press, Feb. 2004.
- [Mon96] Carlo Montangero. *Software Process Technology*. Springer, Sept. 1996.
- [Oqu03] Flavio Oquendo. *Software Process Technology*. Springer, Sept. 2003.
- [Pre04] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. 6th. McGraw-Hill, Mar. 2004.
- [PS01] John W. Priest and Jose M. Sanchez. *Product Development and Design for Manufacturing*. Marcel Dekker, Jan. 2001.
- [Sch04] Stephen R Schach. *Object-Oriented and Classical Software Engineering*. McGraw-Hill, Jan. 2004.
- [WA02] Daniel R. Windle and L. Rene Abreo. *Software Requirements Using the Unified Process*. Prentice Hall, Aug. 2002.
- [WK00] Yingxu Wang and Graham King. *Software Engineering Processes: Principles and Applications*. CRC Press, Apr. 2000.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

ENG42032. English VI: RREL (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 3
- 2.2 Horas de teoría : -
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 96 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : ENG42012. Inglés V. (5^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El aprendizaje del idioma inglés en los estudiantes de nivel superior se ha convertido en elemento necesario para su desarrollo académico, personal y profesional. La mayor cantidad de literatura académica de los diversos campos del saber es redactada en inglés, lo cual garantiza que las personas que cuenten con el dominio del idioma puedan siempre estar actualizados.

Asimismo, el conocimiento de este idioma permite tener perspectivas sociales y culturales más amplias. En ese sentido, para un efectivo aprendizaje de la lengua es necesario el desarrollo de las cuatro (4) habilidades: escuchar, hablar, leer y escribir considerando los lineamientos del Marco Común Europeo de Referencia de Lengua - MCERL.

5. OBJETIVOS

- Communicate and exchange information in a limited way in generally predictable situations.
- Reach level B1 according to the CEFR.
- Manage terminology specific to the study area.

6. COMPETENCIAS

CG2) Comunicación bilingüe. (Evaluar)

7. TEMAS

Unidad 1: My greatest adventure (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Past perfect.• Past perfect / simple past.• Connector (cause - effect).	<ul style="list-style-type: none">• Describe actions in the past.
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [De 02]	

Unidad 2: Building our future (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Future continuous. • Past perfect continuous. • Connector (cause - effect). 	<ul style="list-style-type: none"> • Talk about future.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 3: If I were you... (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Conditionals, 2nd and 3rd. • Connector (cause - effect). 	<ul style="list-style-type: none"> • Talk about imaginary situations.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 4: I should have not done it! (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Modals – should have/might have/etc. • Connector (cause - effect) 	<ul style="list-style-type: none"> • Talk about regrets.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 5: Breaking News! (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Reported speech. • Tag questions. • Connector (cause - effect). 	<ul style="list-style-type: none"> • Report news. • Check information.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

Unidad 6: Can you speak English? (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Design by the teacher. 	<ul style="list-style-type: none"> • Get familiar with technical expressions of the field.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 02]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[De 02] Consejo De Europa. "Marco común europeo de referencia para las lenguas". In: *Strasburgo: Consejo de Europa, Ministerio de Educación, Cultura y Deporte/Instituto Cervantes* (2002).



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

FIN52004. Finanzas Empresariales (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Write justification for this course here ...

5. OBJETIVOS

- Write your first goal here.
- Write your second goal here.
- Just in case you need more goals write them here

6. COMPETENCIAS

CG4) Gestiona recursos. (Evaluar)

7. TEMAS

Unidad 1: title for the unit goes here (0 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Topic1• Topic2• Topic3	<ul style="list-style-type: none">• Learning outcome1 [Levelforthislearningoutcome].• i.e Apply computing in complex problems [Usar].• Create a search engine [Evaluar].• Study data structures [Familiarizarse].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Bibitem1], [Bibitem2]	

Unidad 2: another unit goes here (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Topic1 	<ul style="list-style-type: none"> • Learning outcome xyz [Levelforthislearningoutcome].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Bibitem3], [Bibitem1]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52077. Ingeniería de Software III (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Horas autónomas	:	96 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52086. Ingeniería de Software II. (6 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El desarrollo de software requiere del uso de mejores prácticas de desarrollo, gestión de proyectos de TI, manejo de equipos y uso eficiente y racional de frameworks de aseguramiento de la calidad, estos elementos son pieza clave y transversal durante todo el proceso productivo.

La construcción de software contempla la implementación y uso de procesos, métodos, modelos y herramientas que permitan lograr la realización de los atributos de calidad de un producto.

5. OBJETIVOS

- Comprender y poner en práctica los conceptos fundamentales sobre la gestión de proyectos y manejo de equipos de software.
- Comprender los fundamentos de la gestión de proyectos, incluyendo su definición, alcance, y la necesidad de gestión de proyectos en la organización moderna.
- Los alumnos deben comprender los conceptos fundamentales de CMMI, PSP, TSP para que sean adoptados en los proyectos de software.
- Describir y comprender los modelos de aseguramiento de la calidad como marco clave para el éxito de los proyectos de TI.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Usar**)

7. TEMAS

Unidad 1: Evolución de Software (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Desarrollo de Software en el contexto de código grande pre existente <ul style="list-style-type: none"> – Cambios de software – Preocupaciones y ubicación de preocupaciones – <i>Refactoring</i> • Evolución de Software. • Características de Software mantenible. • Sistemas de Reingeniería. • Reuso de Software. <ul style="list-style-type: none"> – Segmentos de código – Bibliotecas y <i>frameworks</i> – Componentes – Líneas de Producto 	<ul style="list-style-type: none"> • Identificar los problemas principales asociados con la evolución del software y explicar su impacto en el ciclo de vida del software [Familiarizarse] • Estimar el impacto del cambio de requerimientos en productos existentes de tamaño medio [Usar] • Usar refactorización en el proceso de modificación de un componente de software [Usar] • Estudiar los desafíos de mejorar sistemas en un entorno cambiante [Familiarizarse] • Perfilar los procesos de pruebas de regresión y su rol en el manejo de versiones [Familiarizarse] • Estudiar las ventajas y desventajas de diferentes tipos de niveles de confiabilidad [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [PM15], [Som17]	

Unidad 2: Gestión de Proyectos de Software (10 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • La participación del equipo: <ul style="list-style-type: none"> – Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo – Roles y responsabilidades en un equipo de software – Equipo de resolución de conflictos – Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura) • Estimación de esfuerzo (a nivel personal) • Riesgo. <ul style="list-style-type: none"> – El papel del riesgo en el ciclo de vida – Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de • Gestión de equipos: <ul style="list-style-type: none"> – Organización de equipo y la toma de decisiones – Roles de identificación y asignación – Individual y el desempeño del equipo de evaluación • Gestión de proyectos: <ul style="list-style-type: none"> – Programación y seguimiento de elementos – Herramientas de gestión de proyectos – Análisis de Costo/Beneficio 	<ul style="list-style-type: none"> • Discutir los comportamientos comunes que contribuyen al buen funcionamiento de un equipo [Familiarizarse] • Crear y seguir un programa para una reunión del equipo [Usar] • Identificar y justificar las funciones necesarias en un equipo de desarrollo de software [Usar] • Entender las fuentes, obstáculos y beneficios potenciales de un conflicto de equipo [Usar] • Aplicar una estrategia de resolución de conflictos en un ambiente de equipo [Usar] • Utilizar un método ad hoc para estimar el esfuerzo de desarrollo del software (ejemplo, tiempo) y comparar con el esfuerzo actual requerido [Usar] • Listar varios ejemplos de los riesgos del software [Familiarizarse] • Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Familiarizarse] • Describir las diferentes categorías de riesgo en los sistemas de software [Familiarizarse] • Demostrar a través de la colaboración de proyectos de equipo los elementos centrales de la construcción de equipos y gestión de equipos [Usar] • Describir como la elección de modelos de procesos afectan la estructura organizacional de equipos y procesos de toma de decisiones [Familiarizarse] • Crear un equipo mediante la identificación de los roles apropiados y la asignación de funciones a los miembros del equipo [Usar] • Evaluar y retroalimentar a los equipos e individuos sobre su desempeño en un ambiente de equipo [Usar] • Usando un software particular procesar, describir los aspectos de un proyecto que necesita ser planeado y monitoreado, (ejemplo, estimar el tamaño y esfuerzo, un horario, reasignación de recursos, control de configuración, gestión de cambios, identificación de riesgos en un proyecto y gestión) [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [PM15], [Som17]	

Unidad 3: Gestión de Proyectos de Software (8 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Software de medición y técnicas de estimación.• Aseguramiento de la calidad del software y el rol de las mediciones.• Riesgo.<ul style="list-style-type: none">– Identificación de riesgos y gestión.– Análisis riesgo y evaluación.– La tolerancia al riesgo (por ejemplo, riesgo adverso, riesgo neutral, la búsqueda de riesgo)– Planificación de Riesgo• En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas.	<ul style="list-style-type: none">• Realizar el seguimiento del progreso de alguna etapa de un proyecto que utiliza métricas de proyectos apropiados [Usar]• Comparar las técnicas simples de tamaño de software y estimación de costos [Usar]• Usar una herramienta de gestión de proyectos para ayudar en la asignación y rastreo de tareas en un proyecto de desarrollo de software [Usar]• Describir el impacto de la tolerancia de riesgos en el proceso de desarrollo de software [Evaluar]• Identificar riesgos y describir enfoques para manejar riesgos (evitar, aceptar, transferir, mitigar) y caracterizar fortalezas y defectos para cada uno [Familiarizarse]• Explicar cómo el riesgo afecta las decisiones en el proceso de desarrollo de software [Usar]• Identificar los riesgos de seguridad para un sistema de software [Usar]• Demostrar un enfoque sistemático para la tarea de identificar los peligros y riesgos en una situación particular [Usar]• Aplicar los principios básicos del manejo de riesgos en una variedad de escenarios simples incluyendo una situación de seguridad [Usar]• Dirigir un análisis de costo/beneficio para el enfoque de mitigación de riesgos [Usar]• Identificar y analizar alguno de los riesgos para un sistema entero que surgen de aspectos distintos del software [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [PM15], [Som17]	

Unidad 4: Procesos de Software (12 horas)**Competencias esperadas:****Temas**

- Consideraciones a nivel de sistemas, ejem., la interacción del software con su entorno.
- Introducción a modelos del proceso de software (e.g., cascada, incremental, ágil):
 - Actividades con ciclos de vida de software.
- Programación a gran escala versus programación individual.
- Evaluación de modelos de proceso de software.
- Conceptos de calidad de software.
- Mejoramiento de procesos.
- Modelos de madurez de procesos de software.
- Mediciones del proceso de software.

Objetivos de Aprendizaje

- Describa cómo el software puede interactuar y participar en varios sistemas, incluyendo la gestión de información, integración, control de procesos y sistemas de comunicaciones [Usar]
- Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Usar]
- Describir las diferentes prácticas que son componentes clave de los diversos modelos de procesos [Usar]
- Diferenciar entre las fases de desarrollo de software [Usar]
- Describir cómo la programación en grandes equipos difiere de esfuerzos individuales con respecto a la comprensión de una gran base de código, lectura de código, comprensión de las construcciones, y comprensión de contexto de cambios [Usar]
- Explicar el concepto de ciclo de vida del software y proporcionar un ejemplo que ilustra sus fases incluyendo los entregables que se producen [Usar]
- Comparar varios modelos comunes de procesos con respecto a su valor para el desarrollo de las clases particulares de sistemas de software, teniendo en cuenta diferentes aspectos tales como, estabilidad de los requisitos, tamaño y características no funcionales [Usar]
- Definir la calidad del software y describir el papel de las actividades de aseguramiento de la calidad en el proceso de software [Usar]
- Describir el objetivo y similitudes fundamentales entre los enfoques de mejora de procesos [Usar]
- Comparar varios modelos de mejora de procesos, tales como CMM, CMMI, CQI, *Plan-Do-Check-Act*, o ISO9000 [Usar]
- Evaluar un esfuerzo de desarrollo y recomendar cambios potenciales al participar en la mejora de procesos (usando un modelo como PSP) o involucración en una retrospectiva de un proyecto [Usar]
- Explicar el papel de los modelos de madurez de procesos en la mejora de procesos [Usar]
- Describir varias métricas de procesos para la evaluación y el control de un proyecto [Usar]
- Usar las medidas en proyecto para describir el estado actual de un proyecto [Usar]

Aprendizaje autónomo

- Desarrollo de ejercicios prácticos

Lecturas : [PM15], [Som17]

Unidad 5: Estándares ISO/IEC (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • ISO 9001:2001. • ISO 9000-3. • ISO/IEC 9126. • ISO/IEC 12207. • ISO/IEC 15939. • ISO/IEC 14598. • ISO/IEC 15504-SPICE. • IT Mark. • SCRUM. • SQuaRE. • CISQ. 	<ul style="list-style-type: none"> • Aprender y aplicar correctamente normas y estándares internacionales. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Som17], [PM15]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.

[Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52078. Experiencia de Usuario (UX) (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 2
2.2 Horas de teoría	: -
2.3 Horas de práctica	: 2 (Semanal)
2.4 Horas autónomas	: 64 (horas)
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Obligatorio
2.7 Modalidad	: Presencial
2.8 Prerrequisitos	: SFW52083. Sistemas de Infomación. (6 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El lenguaje ha sido una de las creaciones más significativas de la humanidad. Desde el lenguaje corporal y gestual, pasando por la comunicación verbal y escrita, hasta códigos simbólicos icónicos y otros, ha posibilitado interacciones complejas entre los seres humanos y facilitado considerablemente la comunicación de información. Con la invención de dispositivos automáticos y semiautomáticos, entre los que se cuentan las computadoras, la necesidad de lenguajes o interfaces para poder interactuar con ellos, ha cobrado gran importancia.

La usabilidad del software, aunada a la satisfacción del usuario y su incremento de productividad, depende de la eficacia de la Interfaz Usuario-Computador. Tanto es así, que a menudo la interfaz es el factor más importante en el éxito o el fracaso de cualquier sistema computacional. El diseño e implementación de adecuadas Interfaces Humano-Computador, que además de cumplir los requisitos técnicos y la lógica transaccional de la aplicación, considere las sutiles implicaciones psicológicas, culturales y estéticas de los usuarios, consume buena parte del ciclo de vida de un proyecto software, y requiere habilidades especializadas, tanto para la construcción de las mismas, como para la realización de pruebas de usabilidad.

5. OBJETIVOS

- Conocer y aplicar criterios de usabilidad y accesibilidad al diseño y construcción de interfaces humano-computador, buscando siempre que la tecnología se adapte a las personas y no las personas a la tecnología.
- Que el alumno tenga una visión centrada en la experiencia de usuario al aplicar apropiados enfoques conceptuales y tecnológicos.
- Entender como la tecnologica emergente hace posible nuevos estilos de interacción.
- Determinar los requerimientos básicos a nivel de interfaces, hardware y software para la construcción de ambientes inmersivos.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Usar**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Usar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Usar**)

- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (Usar)

7. TEMAS

Unidad 1: Fundamentos (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Contextos para IHC (cualquiera relacionado con una interfaz de usuario, p.e., página web, aplicaciones de negocios, aplicaciones móviles y juegos) Heurística de usabilidad y los principios de pruebas de usabilidad. Procesos para desarrollo centrado en usuarios, p.e., enfoque inicial en usuarios, pruebas empíricas, diseño iterativo. Principios del buen diseño y buenos diseñadores; ventajas y desventajas de ingeniería. Diferentes medidas para evaluación, p.e., utilidad, eficiencia, facilidad de aprendizaje, satisfacción de usuario. 	<ul style="list-style-type: none"> Discutir por qué el desarrollo de software centrado en el hombre es importante [Familiarizarse] Define un proceso de diseño centrado en el usuario que de forma explícita considere el hecho que un usuario no es como un desarrollador o como sus conocimientos [Familiarizarse] Resumir los preceptos básicos de la interacción psicológica y social [Familiarizarse] Desarrollar y usar un vocabulario conceptual para analizar la interacción humana con el software: disponibilidad, modelo conceptual, retroalimentación, y demás [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [Dix+04], [Sto+05], [RS11]	

Unidad 2: Factores Humanos (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Modelos cognoscitivos que informan diseño de interacciones, p.e., atención, percepción y reconocimiento, movimiento, memoria, golfos de expectativa y ejecución. Capacidades físicas que informan diseño de interacción, p.e. percepción del color, ergonomía. Accesibilidad, p.e., interfaces para poblaciones con diferentes habilidades (p.e., invidentes, discapacitados) Interfaces para grupos de población de diferentes edades (p.e., niños, mayores de 80) 	<ul style="list-style-type: none"> Crear y dirigir una simple prueba de usabilidad para una aplicación existente de software [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [Dix+04], [Sto+05], [RS11], [Mat11], [Nor04]	

Unidad 3: Diseño y Testing centrados en el usuario (16 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Enfoque y características del proceso de diseño. • Requerimientos de funcionalidad y usabilidad. • Técnicas de recolección de requerimientos, ej. entrevistas, encuestas, etnografía e investigación contextual. • Técnicas y herramientas para el análisis y presentación de requerimientos ej. reportes, personas. • Análisis de tareas, incluidos los aspectos cualitativos de la generación de modelos de análisis de tareas. • Consideración de IHC como una disciplina de diseño: <ul style="list-style-type: none"> – Sketching – Diseño participativo – Sketching – Diseño participativo • Técnicas de creación de prototipos y herramientas, ej. bosquejos, <i>storyboards</i>, prototipos de baja fidelidad, esquemas de página. • Prototipos de baja fidelidad (papel) • Técnicas de evaluación cuantitativa ej. evaluación Keystroke-level. • Evaluación sin usuarios, usando ambas técnicas cualitativas y cuantitativas. Ej. Revisión estructurada, GOMS, análisis basado en expertos, heurísticas, lineamientos y estándar. • Evaluación con usuarios. Ej. Observación, Método de pensamiento en voz alta, entrevistas, encuestas, experimentación. • Desafíos para la evaluación efectiva, por ejemplo, toma de muestras, la generalización. • Reportar los resultados de las evaluaciones. • Internacionalización, diseño para usuarios de otras culturas, intercultural. 	<ul style="list-style-type: none"> • Llevar a cabo una evaluación cuantitativa y discutir / informar sobre los resultados [Familiarizarse] • Para un grupo de usuarios determinado, realizar y documentar un análisis de sus necesidades [Familiarizarse] • Discutir al menos un standard nacional o internacional de diseño de interfaz de usuario [Familiarizarse] • Explicar cómo el diseño centrado en el usuario complementa a otros modelos de proceso software [Familiarizarse] • Utilizar <i>lo-fi</i> (baja fidelidad) técnicas de prototipado para recopilar y reportar, las respuestas del usuario [Usar] • Elegir los métodos adecuados para apoyar el desarrollo de una específica interfaz de usuario [Evaluar] • Utilizar una variedad de técnicas para evaluar una interfaz de usuario dada [Evaluar] • Comparar las limitaciones y beneficios de los diferentes métodos de evaluación [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Dix+04], [Sto+05], [RS11], [Mat11], [Bux07]	

Unidad 4: Diseño de Interacción (8 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Principios de interfaces gráficas de usuario (GUIs)• Elementos de diseño visual (disposición, color, fuentes, etiquetado)• Manejo de fallas humanas/sistema.• Estándares de interfaz de usuario.• Presentación de información: navegación, representación, manipulación.• Técnicas de animación de interfaz (ej. grafo de escena)• Clases Widget y bibliotecas.• Internacionalización, diseño para usuarios de otras culturas, intercultural.• Elección de estilos de interacción y técnicas de interacción.	<ul style="list-style-type: none">• Crear una aplicación simple, junto con la ayuda y la documentación, que soporta una interfaz gráfica de usuario [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Dix+04], [Sto+05], [RS11], [Joh10], [Mat11], [LS06]	

Unidad 5: Nuevas Tecnologías Interactivas (8 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Elección de estilos de interacción y técnicas de interacción. • Enfoques para el diseño, implementación y evaluación de la interacción sin mouse <ul style="list-style-type: none"> – Interfaces táctiles y multitáctiles. – Interfaces compartidas, incorporadas y grandes – Nuevas modalidades de entrada (tales como datos de sensores y localización) – Nuevas ventanas, por ejemplo, iPhone, Android – Reconocimiento de voz y procesamiento del lenguaje natural – Interfaces utilizables y tangibles – Interacción persuasiva y emoción – Tecnologías de interacción ubicuas y contextuales (UbiComp) – Inferencia bayesiana (por ejemplo, texto predictivo, orientación guiada) – Visualización e interacción de ambiente / periféricos • Salida: <ul style="list-style-type: none"> – Sonido – Visualización estereoscópica – Forzar la simulación de retroalimentación, dispositivos hápticos • Arquitectura de Sistemas: <ul style="list-style-type: none"> – Motores de Juego – Realidad Aumentada móvil – Simuladores de vuelo – CAVEs – Imágenes médicas 	<ul style="list-style-type: none"> • Describe cuando son adecuadas las interfaces sin uso de ratón [Familiarizarse] • Comprende las posibilidades de interacción que van más allá de las interfaces de ratón y puntero [Familiarizarse] • Discute las ventajas (y desventajas) de las interfaces no basadas en ratón [Usar] • Describir el modelo óptico realizado por un sistema de gráficos por computadora para sintetizar una visión estereoscópica [Familiarizarse] • Describir los principios de las diferentes tecnologías de seguimiento de espectador [Familiarizarse] • Determinar los requerimientos básicos en interfaz, software, hardware, y configuraciones de software de un sistema VR para una aplicación específica [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Dix+04], [Sto+05], [RS11], [WW11], [Mat11]	

Unidad 6: Colaboración y Comunicación (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • La comunicación asíncrona en grupo, por ejemplo, el correo electrónico, foros, redes sociales. • Medios de comunicación social, informática social, y el análisis de redes sociales. • Colaboración en línea, espacios "inteligentes" y aspectos de coordinación social de tecnologías de flujo de trabajo. • Comunidades en línea. • Personajes de Software y agentes inteligentes, mundos virtuales y avatares. • Psicología Social 	<ul style="list-style-type: none"> • Describir la diferencia entre la comunicación sincrónica y asíncrona [Familiarizarse] • Comparar los problemas de IHC en la interacción individual con la interacción del grupo [Familiarizarse] • Discuta varias problemas de interés social planteados por el software colaborativo [Usar] • Discutir los problemas de IHC en software que personifica la intención humana [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Dix+04], [Sto+05], [RS11]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Bux07] Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., 2007.
- [Dix+04] Alan Dix et al. *Human-computer Interaction*. 3 ed. Prentice-Hall, Inc, 2004.
- [Joh10] Jeff Johnson. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. 3 ed. Morgan Kaufmann Publishers Inc., 2010.

- [LS06] M. Leavitt and B. Shneiderman. *Research-Based Web Design & Usability Guidelines*. Health and Human Services Dept, 2006.
- [Mat11] Lukas Mathis. *Designed for Use: Create Usable Interfaces for Applications and the Web*. Pragmatic Bookshelf, 2011.
- [Nor04] Donald A. Norman. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Book, 2004.
- [RS11] Y. Rogers and J Sharp H. & Preece. *Interaction Design: Beyond Human-Computer Interaction*. 3 ed. John Wiley and Sons Ltd, 2011.
- [Sto+05] D. Stone et al. *User Interface Design and Evaluation*. Morgan Kaufmann Series in Interactive Technologies, 2005.
- [WW11] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc, 2011.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52081. Metodología de la Investigación en Computación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 3
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: -
2.4 Horas autónomas	: 96 (horas)
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Obligatorio
2.7 Modalidad	: Presencial
2.8 Prerrequisitos	: SFW52092. Teoría de la Computación. (4 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso tiene por objetivo que el alumno aprenda a realizar una investigación de carácter científico en el área de computación. Los docentes del curso determinarán un área de estudio para cada alumno, y se le hará entrega de bibliografía para analizar y a partir de la misma, y de fuentes bibliográficas adicionales (investigadas por el alumno), el alumno deberá ser capaz de construir un artículo del tipo survey del tema asignado.

5. OBJETIVOS

- Que el alumno aprenda como se inicia una investigación científica en el área de computación.
- Que el alumno conozca las principales fuentes para obtener bibliografía relevante para trabajos de investigación en el área de computación: Researchindex, IEEE-CS¹, ACM².
- Que el alumno sea capaz de analizar las propuestas existentes sobre un determinado tópico y relacionarlos de forma coherente en una revisión bibliográfica.
- Que el alumno pueda redactar documentos técnicos en computación utilizando L^AT_EX.
- Que el alumno sea capaz de reproducir los resultados ya existentes en un determinado tópico a través de la experimentación.
- Los entregables de este curso son:

Avance parcial: Dominio del tema del artículo y bibliografía preliminar en formato de artículo L^AT_EX.

Final: Entendimiento del artículo del tipo survey, documento concluido donde se contenga, opcionalmente, los resultados experimentales de la(s) técnica(s) estudiada(s).

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Usar**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Usar**)

¹<http://www.computer.org>

²<http://www.acm.org>

5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (Usar)

CG3) Investigación. (Evaluar)

7. TEMAS

Unidad 1: Iniciación científica en el área de computación (60 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Búsqueda bibliográfica en computación. Redacción de artículos técnicos en computación. 	<ul style="list-style-type: none"> Aprender a hacer una investigación correcta en el área de computación[Usar] Conocer las fuentes de bibliografía adecuada para esta área[Usar] Saber redactar un documento de acorde con las características que las conferencias de esta área exigen[Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [IEE08], [Ass08], [Cit08]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[Ass08] Association for Computing Machinery. *Digital Library*. <http://portal.acm.org/dl.cfm>. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Library*. <http://citeseer.ist.psu.edu>. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray*. <http://www.computer.org/publications/dlib>. IEEE-Computer Society, 2008.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52079. Aprendizaje Automático (Electivo)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52082. Sistemas Inteligentes. (6 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La investigación en Inteligencia Artificial ha conducido al desarrollo de numerosas técnicas relevantes, dirigidas a la automatización de la inteligencia humana, dando una visión panorámica de diferentes algoritmos que simulan los diferentes aspectos del comportamiento y la inteligencia del ser humano.

5. OBJETIVOS

- Evaluar las posibilidades de simulación de la inteligencia, para lo cual se estudiarán las técnicas de modelización del conocimiento.
- Construir una noción de inteligencia que soporte después las tareas de su simulación.

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Cuestiones fundamentales (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial. • ¿Qué es comportamiento inteligente? <ul style="list-style-type: none"> – El Test de Turing – Razonamiento Racional versus No Racional • Características del Problema: <ul style="list-style-type: none"> – Observable completamente versus observable parcialmente – Individual versus multi-agente – Determinístico versus estocástico – Estático versus dinámico – Discreto versus continuo • Naturaleza de agentes: <ul style="list-style-type: none"> – Autónomo versus semi-autónomo – Reflexivo, basado en objetivos, y basado en utilidad – La importancia en percepción e interacciones con el entorno • Cuestiones filosóficas y éticas. 	<ul style="list-style-type: none"> • Describir el test de Turing y el experimento pensado "cuarto chino" (<i>Chinese Room</i>) [Usar] • Determinando las características de un problema dado que sistemas inteligentes deberían resolver [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [De 06], [Pon+14]	

Unidad 2: Agentes (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Definición de Agentes • Arquitectura de agentes (Ej. reactivo, en capa, cognitivo) • Teoría de agentes • Racionalidad, teoría de juegos: <ul style="list-style-type: none"> – Agentes de decisión teórica – Procesos de decisión de Markov (MDP) • Agentes de Software, asistentes personales, y acceso a información: <ul style="list-style-type: none"> – Agentes colaborativos – Agentes de recolección de información – Agentes creíbles (carácter sintético, modelamiento de emociones en agentes) • Agentes de aprendizaje • Sistemas Multi-agente <ul style="list-style-type: none"> – Agentes Colaborativos – Equipos de Agentes – Agentes Competitivos (ej., subastas, votaciones) – Sistemas de enjambre y modelos biológicamente inspirados 	<ul style="list-style-type: none"> • Lista las características que definen un agente inteligente [Usar] • Describe y contrasta las arquitecturas de agente estándares [Usar] • Describe las aplicaciones de teoría de agentes para dominios como agentes de software, asistentes personales, y agentes creíbles [Usar] • Describe los paradigmas primarios usados por agentes de aprendizaje [Usar] • Demuestra mediante ejemplos adecuados como los sistemas multi-agente soportan interacción entre agentes [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Nil01], [RN03], [Pon+14]	

Unidad 3: Estrategias de búsquedas básicas (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Espacios de Problemas (estados, metas y operadores), solución de problemas mediante búsqueda. • Factored representation (factoring state hacia variables) • Uninformed search (breadth-first, depth-first, depth-first with iterative deepening) • Heurísticas y búsqueda informada (hill-climbing, generic best-first, A*) • El espacio y el tiempo de la eficiencia de búsqueda. • Dos jugadores juegos (introducción a la búsqueda minimax). • Satisfacción de restricciones (backtracking y métodos de búsqueda local). 	<ul style="list-style-type: none"> • Formula el espacio eficiente de un problema para un caso expresado en lenguaje natural (ejm. Inglés) en términos de estados de inicio y final, así como sus operadores [Usar] • Describe el rol de las heurísticas y describe los intercambios entre completitud, óptimo, complejidad de tiempo, y complejidad de espacio [Usar] • Describe el problema de la explosión combinatoria del espacio de búsqueda y sus consecuencias [Usar] • Compara y contrasta tópicos de búsqueda básica con temas jugabilidad de juegos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Nil01], [Pon+14]	

Unidad 4: Búsqueda Avanzada (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Búsqueda estocástica: <ul style="list-style-type: none"> – Simulated annealing – Algoritmos genéticos – Búsqueda de árbol Monte-Carlo • Construcción de árboles de búsqueda, espacio de búsqueda dinámico, explosión combinatoria del espacio de búsqueda. • Implementación de búsqueda A *, búsqueda en haz. • Búsqueda Minimax, poda alfa-beta. • Búsqueda Expectimax (MDP-Solving) y los nodos de azar. 	<ul style="list-style-type: none"> • Diseñar e implementar una solución a un problema con algoritmo genético [Usar] • Diseñar e implementar un esquema de recocido simulado (<i>simulated annealing</i>) para evitar mínimos locales en un problema [Usar] • Diseñar e implementar una búsqueda A* y búsqueda en haz (<i>beam search</i>) para solucionar un problema [Usar] • Aplicar búsqueda minimax con poda alfa-beta para simplificar el espacio de búsqueda en un juego con dos jugadores [Usar] • Comparar y contrastar los algoritmos genéticos con técnicas clásicas de búsqueda [Usar] • Comparar y contrastar la aplicabilidad de varias heurísticas de búsqueda, para un determinado problema [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Gol89], [Nil01], [RN03], [Pon+14]	

Unidad 5: Razonamiento Bajo Incertidumbre (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Revisión de Probabilidad Básica • Variables aleatorias y distribuciones de probabilidad: <ul style="list-style-type: none"> – Axiomas de probabilidad – Inferencia probabilística – Regla de Bayes • Independencia Condicional • Representaciones del conocimiento: <ul style="list-style-type: none"> – Redes bayesianas <ul style="list-style-type: none"> * Inferencia exacta y su complejidad * Métodos de Muestreo aleatorio (Monte Carlo) (p.e. Muestreo de Gibbs) – Redes Markov – Modelos de probabilidad relacional – Modelos ocultos de Markov 	<ul style="list-style-type: none"> • Aplicar la regla de Bayes para determinar el cumplimiento de una hipótesis [Usar] • Explicar cómo al tener independencia condicional permite una gran eficiencia en sistemas probabilísticos [Usar] • Identificar ejemplos de representación de conocimiento para razonamiento bajo incertidumbre [Usar] • Indicar la complejidad de la inferencia exacta. Identificar métodos para inferencia aproximada [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KF09], [RN03]	

Unidad 6: Aprendizaje Automático Básico (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Definición y ejemplos de la extensa variedad de tareas de aprendizaje de máquina, incluida la clasificación. • Aprendizaje inductivo • Aprendizaje simple basado en estadísticas, como el clasificador ingenuo de Bayes, árboles de decisión. • El problema exceso de ajuste. • Medición clasificada con exactitud. 	<ul style="list-style-type: none"> • Listar las diferencias entre los tres principales tipos de aprendizaje: supervisado, no supervisado y por refuerzo [Usar] • Identificar ejemplos de tareas de clasificación, considerando las características de entrada disponibles y las salidas a ser predecidas [Usar] • Explicar la diferencia entre aprendizaje inductivo y deductivo [Usar] • Describir el sobre ajuste (<i>overfitting</i>) en el contexto de un problema [Usar] • Aplicar un algoritmo de aprendizaje estadístico simple como el Clasificador Naive Bayesiano e un problema de clasificación y medirla precisión del clasificador [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Mit98], [RN03], [Pon+14]	

Unidad 7: Aprendizaje de máquina avanzado (20 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Definición y ejemplos de una amplia variedad de tareas de aprendizaje de máquina • Aprendizaje general basado en estadística, estimación de parámetros (máxima probabilidad) • Programación lógica inductiva (<i>Inductive logic programming ILP</i>) • Aprendizaje supervisado <ul style="list-style-type: none"> – Aprendizaje basado en árboles de decisión – Aprendizaje basado en redes neuronales – Aprendizaje basado en máquinas de soporte vectorial (<i>Support vector machines SVMs</i>) • Aprendizaje y <i>clustering</i> no supervisado <ul style="list-style-type: none"> – EM – K-means – Mapas auto-organizados • Aprendizaje semi-supervisado. • Aprendizaje de modelos gráficos • Evaluación del desempeño (tal como cross-validation, area bajo la curva ROC) • Aplicación de algoritmos Machine Learning para Minería de datos. 	<ul style="list-style-type: none"> • Explica las diferencias entre los tres estilos de aprendizaje: supervisado, por refuerzo y no supervisado [Usar] • Implementa algoritmos simples para el aprendizaje supervisado, aprendizaje por refuerzo, y aprendizaje no supervisado [Usar] • Determina cuál de los tres estilos de aprendizaje es el apropiado para el dominio de un problema en particular [Usar] • Compara y contrasta cada una de las siguientes técnicas, dando ejemplo de cuando una estrategia es la mejor: árboles de decisión, redes neuronales, y redes bayesianas [Usar] • Evalúa el rendimiento de un sistema de aprendizaje simple en un conjunto de datos reales [Usar] • Describe el estado del arte en la teoría del aprendizaje, incluyendo sus logros y limitantes [Usar] • Explica el problema del sobreajuste, conjuntamente con técnicas para determinar y manejar el problema [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [RN03], [KF09], [Mur12]	

Unidad 8: Procesamiento del Lenguaje Natural (12 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Gramaticas determinísticas y estocásticas • Algoritmos de parseo <ul style="list-style-type: none"> – Gramáticas libres de contexto (CFGs) y cuadros de parseo (e.g. Cocke-Younger-Kasami CYK) – CFGs probabilísticos y ponderados CYK • Representación del significado / Semántica <ul style="list-style-type: none"> – Representación de conocimiento basado en lógica – Roles semánticos – Representaciones temporales – Creencias, deseos e intenciones • Metodos basados en el corpus • N-gramas y Modelos ocultos de Markov (HMMs) • Suavizado y back-off • Ejemplos de uso: POS etiquetado y morfología • Recuperación de la información: <ul style="list-style-type: none"> – Modelo de espacio vectorial <ul style="list-style-type: none"> * TF & IDF – Precision y cobertura • Extracción de información • Traducción de lenguaje • Clasificación y categorización de texto: <ul style="list-style-type: none"> – Modelo de bolsa de palabras 	<ul style="list-style-type: none"> • Define y contrasta gramáticas de tipo estocásticas y determinísticas, dando ejemplos y demostrando como adecuar cada una de ellas [Usar] • Simula, aplica, o implementa algoritmos clásicos y estocásticos para el parseo de un lenguaje natural [Usar] • Identifica los retos de la representación del significado [Usar] • Lista las ventajas de usar corpus estándares. Identifica ejemplos de corpus actuales para una variedad de tareas de PLN [Usar] • Identifica técnicas para la recuperación de la información, traducción de lenguajes, y clasificación de textos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Nil01], [RN03], [Pon+14]	

Unidad 9: Visión y percepción por computador (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión Computacional <ul style="list-style-type: none"> – Adquisición de imágenes, representación, procesamiento y propiedades – Representación de formas, reconocimiento y segmentación de objetos – Análisis de movimiento • Modularidad en reconocimiento. • Enfoques de reconocimiento de patrones <ul style="list-style-type: none"> – Algoritmos de clasificación y medidas de calidad de la clasificación. – Técnicas estadísticas. 	<ul style="list-style-type: none"> • Resumir la importancia del reconocimiento de imágenes y objetos en Inteligencia Artificial (AI) e indicar varias aplicaciones significativas de esta tecnología [Usar] • Listar al menos tres aproximaciones de segmentación de imágenes, tales como algoritmos de límites (thresholding), basado en el borde y basado en regiones, junto con sus características definitorias, fortalezas y debilidades [Usar] • Implementar reconocimiento de objetos en 2d basados en la representación del contorno y/o regiones basadas en formas [Usar] • Proporcionar al menos dos ejemplos de transformación de una fuente de datos de un dominio sensorial a otro, ejemplo, datos táctiles interpretados como imágenes en 2d de una sola banda [Usar] • Implementar un algoritmo para la extracción de características en información real, ejemplo, un detector de bordes o esquinas para imágenes o vectores de coeficientes de Fourier describiendo una pequeña porción de señal de audio [Usar] • Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Usar] • Evaluar el desempeño de la función de extracción subyacente, en relación con al menos una aproximación alternativa posible (ya sea implementado o no) en su contribución a la tarea de clasificación (8) anterior [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Nil01], [RN03], [Pon+14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. CRC Press, 2006.
- [Gol89] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN: 0262013193.
- [Mit98] M. Mitchell. *An introduction to genetic algorithms*. The MIT press, 1998.
- [Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 0262018020.
- [Nil01] Nils Nilsson. *Inteligencia Artificial: Una nueva visión*. McGraw-Hill, 2001.
- [Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.
- [RN03] Stuart Russell and Peter Norvig. *Inteligencia Artificial: Un enfoque moderno*. Prentice Hall, 2003.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW52080. Computación Gráfica (Electivo)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52084. Estructuras de Datos Avanzadas. (6 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Ofrece una introducción para el área de Computación Gráfica, la cual es una parte importante dentro de Ciencias de la Computación. El proposito de este curso es investigar los principios, técnicas y herramientas fundamentales para esta área.

5. OBJETIVOS

- Acercar al alumno a conceptos y técnicas usados en aplicaciones gráficas 3-D complejas.
- Dar al alumno las herramientas necesarias para determinar que software gráfico y que plataforma son los más adecuados para desarrollar una aplicación específica.

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Conceptos Fundamentales (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Aplicaciones multimedia, incluyendo interfaces de usuario, edición de audio y vídeo, motores de juego, cad, visualización, realidad virtual. • Soluciones de compensación entre el almacenamiento de datos y los datos re-computing es personalizado por vectores y raster en representaciones de imágenes. • Modelos de color sustractivo Aditivo y (CMYK y RGB) y por qué estos proporcionan una gama de colores. • Animación como una secuencia de imágenes fijas. 	<ul style="list-style-type: none"> • Explicar en términos generales cómo las señales analógicas pueden ser representadas por muestras discretas, por ejemplo, cómo las imágenes pueden ser representadas por píxeles [Familiarizarse] • Describir modelos de color y su uso en los dispositivos de visualización de gráficos [Familiarizarse] • Describir las ventajas y desventajas entre el almacenamiento de información vs almacenar suficiente información para reproducir la información, como en la diferencia entre el vector y la representación de la trama [Familiarizarse] • Describir los procesos básico de la producción de movimiento continuo a partir de una secuencia de cuadros discretos (algunas veces llamado it flicker fusion) [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HB90]	

Unidad 2: Rendering Básico (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Renderizado en la naturaleza, por ejemplo, la emisión y dispersión de la luz y su relación con la integración numérica. • Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización) • Radiometría básica, triángulos similares y modelos de proyecciones • Afinamiento y Transformaciones de Sistemas de coordenadas • <i>Ray tracing</i> • Visibilidad y oclusión, incluyendo soluciones a este problema, como el almacenamiento en búfer de profundidad, algoritmo del pintor, y el trazado de rayos. • Rasterización triangular simple. • Renderización con una API basada en shader. • Aplicación de la representación de estructuras de datos espaciales. • Muestreo y anti-aliasing. • Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización) 	<ul style="list-style-type: none"> • Discutir el problema de transporte de la luz y su relación con la integración numérica, es decir, se emite luz, dispersa alrededor de la escena, y es medida por el ojo [Familiarizarse] • Describir la tubería básica gráficos y cómo el factor de representación va hacia adelante y atrás en esta [Familiarizarse] • Crear un programa para visualizar modelos 3D de imágenes gráficas simples [Usar] • Obtener puntos en 2-dimensiones y 3-dimensiones por aplicación de transformaciones afin [Usar] • Aplicar sistema de coordenadas de 3-dimensiones y los cambios necesarios para extender las operaciones de transformación 2D para manejar las transformaciones en 3D [Usar] • Contrastar la renderización hacia adelante <i>forward</i> y hacia atrás <i>backward</i> [Evaluar] • Explicar el concepto y las aplicaciones de mapeo de texturas, muestreo y el <i>anti-aliasing</i> [Familiarizarse] • Explicar la dualidad de rastreo de rayos/rasterización para el problema de visibilidad [Familiarizarse] • Implementar un sencillo renderizador en tiempo real utilizando una API de rasterización (por ejemplo, OpenGL) utilizando buffers de vértices y <i>shaders</i> [Usar] • Calcular las necesidades de espacio en base a la resolución y codificación de color [Evaluar] • Calcular los requisitos de tiempo sobre la base de las frecuencias de actualización, técnicas de rasterización [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [MS16]	

Unidad 3: Programación de Sistemas Interactivos (2 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">● Manejo de eventos e interacción de usuario.● Enfoques para el diseño, implementación y evaluación de la interacción sin mouse<ul style="list-style-type: none">– Interfaces táctiles y multitáctiles.– Interfaces compartidas, incorporadas y grandes– Nuevas modalidades de entrada (tales como datos de sensores y localización)– Nuevas ventanas, por ejemplo, iPhone, Android– Reconocimiento de voz y procesamiento del lenguaje natural– Interfaces utilizables y tangibles– Interacción persuasiva y emoción– Tecnologías de interacción ubicuas y contextuales (UbiComp)– Inferencia bayesiana (por ejemplo, texto predictivo, orientación guiada)– Visualización e interacción de ambiente / periféricos	<ul style="list-style-type: none">● Discute las ventajas (y desventajas) de las interfaces no basadas en ratón [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none">● Desarrollo de ejercicios prácticos	
Lecturas : [MS16]	

Unidad 4: Modelado Geométrico (15 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Operaciones geométricas básicas como cálculo de intersección y pruebas de proximidad.• Volúmenes, voxels y representaciones basadas en puntos.• Curvas polinomiales y Superficies paramétricas.• Representación implícita de curvas y superficies.• Técnicas de aproximación, tales como curvas polinómicas, curvas Bezier, curvas spline y superficies, y base racional no uniforme (NURB) espinas, y el método de ajuste de nivel.• Técnicas de superficie de representación incluyendo teselación, la representación de malla, carenado malla, y las técnicas de generación de mallas, como la triangulación de Delaunay, marchando cubos.• Técnicas de subdivisión espacial.• Modelos procedimentales como fractales, modelamiento generativo y sistemas L.• Modelos deformables de forma libre y elásticamente deformables.• Subdivisión de superficies.• Modelado multiresolución.• Reconstrucción.• Representación de Geometría Sólida Constructiva (GSC)	<ul style="list-style-type: none">• Representar curvas y superficies utilizando formas tanto implícitas y paramétricas [Usar]• Crear modelos poliédrico simples por teselación de superficies [Usar]• Generar una representación de malla de una superficie implícita [Usar]• Generar una malla de un conjunto de puntos adquiridos por un scanner laser [Usar]• Construct modelos de geometría sólida constructiva a partir de simples primitivas, tales como cubos y superficies cuádricas [Usar]• Contrastar métodos de modelización con respecto a espacio y tiempo de complejidad y calidad de imagen [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [MS16]	

Unidad 5: Renderizado Avanzado (6 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Tiempo (desenfoque de movimiento), la posición del objetivo (enfoque), y la frecuencia continua (color) y su impacto en la representación. • Mapeo de Sombras. • Selectiva de oclusión. • Dispersión de la Superficie. • Renderizado no fotorealístico. • Arquitectura del GPU. • Sistemas visuales humanos incluida la adaptación a la luz, la sensibilidad al ruido, y la fusión de parpadeo. 	<ul style="list-style-type: none"> • Demostrar como un algoritmo calcula una solución a la ecuación de renderización [Evaluar] • Demostrar las propiedades de un algoritmo de renderización, por ejemplo, completo, consistente, e imparcial [Evaluar] • Implementar un algoritmo no trivial de sombreado (por ejemplo, sombreado caricaturizado (<i>toon shading</i>), mapas de sombras en cascada (<i>cascaded shadow maps</i>)) bajo una API de rasterización [Usar] • Discutir como una técnica artística particular puede ser implementada en un renderizador [Familiarizarse] • Explicar como reconocer las técnicas gráficas usadas para crear una imagen en particular [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [MS16]	

Unidad 6: Animación por computadora (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Cinemática directa e inversa. • Detección de colisiones y respuesta. • Animación procedimental empleando ruido, reglas (boids/crowds) y sistemas de partículas. • Algoritmos Skinning. • Movimientos basado en la física, incluyendo la dinámica del cuerpo rígido, sistemas de partículas físicas, redes de masa-muelle de tela y la carne y el pelo. • Animación de Cuadros Principales • Splines • Estructuras de datos para rotaciones, como cuaterniones. • Animación de Cámara. • Captura de Movimiento. 	<ul style="list-style-type: none"> • Calcular la localización y orientación de partes de un modelo usando un enfoque de cinemática hacia delante [Usar] • Implementar el método de interpolación <i>spline</i> para producir las posiciones y orientaciones en medio [Usar] • Implementar algoritmos para el modelamiento físico de partículas dinámicas usando simplemente la mecánica de Newton, por ejemplo Witkin & Kass, serpientes y gusanos, Euler simpléctica, Stormer/Verlet, o métodos de punto medio de Euler [Usar] • Discutir las ideas básicas detrás de algunos métodos para dinámica de fluidos para el modelamiento de trayectorias balísticas, por ejemplo salpicaduras, polvo, fuego, o humo [Familiarizarse] • Usar el software de animación común para construir formas orgánicas simples usando <i>metaball</i> y el esqueleto [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [MS16]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[HB90] Donald Hearn and Pauline Baker. *Computer Graphics in C*. Prentice Hall, 1990.

[MS16] Steve Marschner and Peter Shirley. *Fundamentals of Computer Graphics*. Fourth Edition. CRC Press, 2016. ISBN: ISBN-10: 1482229390.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

MAC41014. Álgebra Lineal (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 4 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : MAC41001. Cálculo de una Variable. (2^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso introduce los primeros conceptos del álgebra lineal, así como los métodos numéricos con un énfasis en la resolución de problemas con el paquete de software libre de código abierto Scilab. La teoría matemática se limita a los fundamentos, mientras que la aplicación efectiva para la resolución de problemas es privilegiada. En cada tópico, se enseña unos cuantos métodos de relevancia para la ingeniería. Los conocimientos sobre estos métodos prepara a los estudiantes para la búsqueda de alternativas más avanzadas, si se lo requiere.

5. OBJETIVOS

- Capacidad para aplicar los conocimientos sobre Matemáticas.
- Capacidad para aplicar los conocimientos sobre Ingeniería .
- Capacidad para aplicar los conocimientos, técnicas, habilidades y herramientas modernas de la ingeniería moderna para la práctica de la ingeniería.

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Introducción (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Importancia del álgebra lineal y métodos numéricos. Ejemplos.	<ul style="list-style-type: none">• Ser capaz de entender los conceptos básicos y la importancia de Álgebra Lineal y Métodos Numéricos.
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [AR14], [CC15]	

Unidad 2: Álgebra lineal (14 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Álgebra matricial elemental y determinantes.• Espacio nulo y soluciones exactas de sistemas de ecuaciones lineales $Ax = b$:<ul style="list-style-type: none">– Sistemas tridiagonal y triangular y eliminación gaussiana con y sin giro.– Factorización LU y algoritmo Crout.• Conceptos básicos sobre valores propios y vectores propios<ul style="list-style-type: none">– Polinomios característicos.– Multiplicaciones algebraicas y geométricas.• Estimación de mínimos cuadrados.• Transformaciones lineales.	<ul style="list-style-type: none">• Comprender los conceptos básicos del Álgebra Lineal.• Resolver problemas de transformaciones lineales.
Aprendizaje autónomo <ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [AR14], [CC15]	

Unidad 3: Métodos Numéricos (22 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Fundamentos de soluciones de sistemas de ecuaciones lineales $Ax = b$: métodos de Jacobi y Gauss Seidel • Aplicación de factorizaciones de matriz a la solución de sistemas lineales (descomposición de valores singulares, QR, Cholesky) Cálculo numérico del espacio nulo, rango y número de condición • Conclusión de la raíz: <ul style="list-style-type: none"> – Bisección. – Iteración de punto fijo. – Métodos de Newton-Raphson. • Fundamentos de la interpolación: <ul style="list-style-type: none"> – Interpolaciones polinomiales de Newton y Lagrange. – Interpolación de spline. • Fundamentos de la diferenciación numérica y la aproximación de Taylor. • Aspectos básicos de la integración numérica: <ul style="list-style-type: none"> – Trapecio, punto medio y regla de Simpson – Cuadratura gaussiana • Conceptos básicos sobre las soluciones numéricas a las EDOs: <ul style="list-style-type: none"> – Diferencias finitas; Métodos de Euler y Runge-Kutta – Convertir ODEs de orden superior en un sistema de ODEs de bajo orden. – Métodos de Runge-Kutta para sistemas de ecuaciones – Método simple.XYZ • Breve introducción a las técnicas de optimización: visión general sobre la programación lineal, sistemas lineales acotados, programación cuadrática, descenso gradiente. 	<ul style="list-style-type: none"> • Comprender los conceptos básicos de los métodos numéricos. • Aplicar los métodos más frecuentes para la resolución de problemas matemáticos. • Implementación y aplicación de algoritmos numéricos para la solución de problemas matemáticos utilizando el paquete computacional Scilab open-source. • Aplicación de Scilab para la solución de problemas matemáticos y para trazar graficas.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [AR14], [CC15]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje

activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[AR14] H. Anton and C. Rorres. *Elementary Linear Algebra, Applications Version*. 11th. Wiley, 2014.

[CC15] S.C. Chapra and R.P. Canale. *Numerical Methods for Engineers*, 7th. Vol. 1. McGraw-Hill, 2015.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

IIS53004. Redes y Telecomunicaciones (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	IIS52024. Sistemas Operativos. (4 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El siempre creciente desarrollo de las tecnologías de comunicación y la información hace que exista una marcada tendencia a establecer más redes de computadores que permitan una mejor gestión de la información.

En este segundo curso se brindará a los participantes una introducción a los problemas que conlleva la comunicación entre computadores, a través del estudio e implementación de protocolos de comunicación como TCP/IP y la implementación de software sobre estos protocolos.

5. OBJETIVOS

- Que el alumno implemente y/o modifique un protocolo de comunicación de datos.
- Que el alumno domine las técnicas de transmisión de datos utilizadas por los protocolos de red existentes.
- Que el alumno conozca las últimas tendencias en redes que se están aplicando en el Internet.

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Usar**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Usar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Usar**)
- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (**Usar**)

7. TEMAS

Unidad 1: Introducción (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Organización de la Internet (proveedores de servicios de Internet, proveedores de contenido, etc) • Técnicas de Switching (por ejemplo, de circuitos, de paquetes) • Piezas físicas de una red, incluidos hosts, routers, switches, ISPs, inalámbrico, LAN, punto de acceso y firewalls. • Principios de capas (encapsulación, multiplexación) • Roles de las diferentes capas (aplicación, transporte, red, enlace de datos, física) 	<ul style="list-style-type: none"> • Articular la organización de la Internet [Familiarizarse] • Listar y definir la terminología de red apropiada [Familiarizarse] • Describir la estructura en capas de una arquitectura típica en red [Familiarizarse] • Identificar los diferentes tipos de complejidad en una red (bordes, núcleo, etc.) [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KR13]	

Unidad 2: Aplicaciones en red (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Esquemas de denominación y dirección (DNS, direcciones IP, identificadores de recursos uniformes, etc) • Las aplicaciones distribuidas (cliente / servidor, peer-to-peer, nube, etc) • HTTP como protocolo de capa de aplicación . • Multiplexación con TCP y UDP • API de Socket 	<ul style="list-style-type: none"> • Listar las diferencias y las relaciones entre los nombres y direcciones en una red [Familiarizarse] • Definir los principios detrás de esquemas de denominación y ubicación del recurso [Familiarizarse] • Implementar una aplicación simple cliente-servidor basada en <i>sockets</i> [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KR13]	

Unidad 3: Entrega confiable de datos (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Control de errores (técnicas de retransmisión, temporizadores) • El control de flujo (agradecimientos, ventana deslizante) • Problemas de rendimiento (pipelining) • TCP 	<ul style="list-style-type: none"> • Describir el funcionamiento de los protocolos de entrega fiables [Familiarizarse] • Listar los factores que afectan al rendimiento de los protocolos de entrega fiables [Familiarizarse] • Diseñar e implementar un protocolo confiable simple [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KR13]	

Unidad 4: Ruteo y reenvío (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Enrutamiento vs reenvío . • Enrutamiento estático . • Protocolo de Internet (IP) • Problemas de escalabilidad (direccionamiento jerárquico) 	<ul style="list-style-type: none"> • Describir la organización de la capa de red [Familiarizarse] • Describir cómo los paquetes se envían en una red IP [Familiarizarse] • Listar las ventajas de escalabilidad de direccionamiento jerárquico [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KR13]	

Unidad 5: Redes de área local (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Problemas de Acceso Múltiple. • Enfoques comunes a Acceso múltiple (exponencial backoff, multiplexación por división de tiempo, etc) • Redes de área local . • Ethernet . • Switching . 	<ul style="list-style-type: none"> • Describir como los paquetes son enviados en una red Ethernet [Familiarizarse] • Describir las relaciones entre IP y Ethernet [Familiarizarse] • Describir las etapas usadas en un enfoque común para el problema de múltiples accesos [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KR13]	

Unidad 6: Asignación de recursos (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Necesidad de asignación de recursos . • Asignación fija (TDM, FDM, WDM) versus la asignación dinámica . • De extremo a extremo frente a las red de enfoque asistida . • Justicia. • Principios del control de congestión. • Enfoques para la congestión (por ejemplo, redes de distribución de contenidos) 	<ul style="list-style-type: none"> • Describir como los recursos pueden ser almacenados en la red [Familiarizarse] • Describir los problemas de congestión en una red grande [Familiarizarse] • Comparar y contrastar las técnicas de almacenamiento estático y dinámico [Familiarizarse] • Comparar y contrastar los enfoques actuales de la congestión [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KR13]	

Unidad 7: Celulares (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Principios de redes celulares. • Redes 802.11 • Problemas en el apoyo a los nodos móviles (agente local) 	<ul style="list-style-type: none"> • Describir la organización de una red inalámbrica [Familiarizarse] • Describir como las redes inalámbricas soportan usuarios móviles [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KR13], [Cha16]	

Unidad 8: Redes sociales (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Panorama de las redes sociales. • Ejemplo plataformas de redes sociales. • Estructura de los grafos de redes sociales. • Análisis de redes sociales. 	<ul style="list-style-type: none"> • Discutir los principios fundamentales (como pertenencia, confianza) de una red social [Familiarizarse] • Describir como redes sociales existentes operan [Familiarizarse] • Construir un grafo de una red social a partir de datos de la red [Usar] • Analizar una red social para determinar quienes son las personas importantes [Usar] • Evaluar una determinada interpretación de una pregunta de red social con los datos asociados [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KR13], [Kad11]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Cha16] Paresh Chayapathi Rajendra; Syed F. Hassan; Shah. *Network Functions Virtualization (NFV) with a Touch of SDN*. Addison-Wesley Professional; 1 edition, 2016. ISBN: 978-0134463056.
- [Kad11] Charles Kadushin. *Understanding Social Networks: Theories, Concepts, And Findings*. Oxford University Press, Usa; 1 edition, 2011. ISBN: 978-0195379471.
- [KR13] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-down Approach*. 7th. Always learning. Pearson, 2013. ISBN: 978-0133594140.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

BSC51001. Biología (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : 2 (Semanal)
- 2.4 Horas autónomas : 112 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Write justification for this course here ...

5. OBJETIVOS

- Write your first goal here.
- Write your second goal here.
- Just in case you need more goals write them here

6. COMPETENCIAS

CG4) Gestiona recursos. (Evaluar)

7. TEMAS

Unidad 1: title for the unit goes here (0 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Topic1• Topic2• Topic3	<ul style="list-style-type: none">• Learning outcome1 [Levelforthislearningoutcome].• i.e Apply computing in complex problems [Usar].• Create a search engine [Evaluar].• Study data structures [Familiarizarse].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Bibitem1], [Bibitem2]	

Unidad 2: another unit goes here (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Topic1 	<ul style="list-style-type: none"> • Learning outcome xyz [Levelforthislearningoutcome].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Bibitem3], [Bibitem1]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62069. Formación de Empresas de Base Tecnológica I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	2
2.2 Horas de teoría	:	1 (Semanal)
2.3 Horas de práctica	:	-
2.4 Horas autónomas	:	64 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	ENG42032. English VI: RREL. (6 ^{to} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este es el primer curso dentro del área de formación de empresas de base tecnológica, tiene como objetivo dotar al futuro profesional de conocimientos, actitudes y aptitudes que le permitan elaborar un plan de negocio para una empresa de base tecnológica. El curso está dividido en las siguientes unidades: Introducción, Creatividad, De la idea a la oportunidad, el modelo Canvas, Customer Development y Lean Startup, Aspectos Legales y Marketing, Finanzas de la empresa y Presentación.

Se busca aprovechar el potencial creativo e innovador y el esfuerzo de los alumnos en la creación de nuevas empresas.

5. OBJETIVOS

- Que el alumno conozca como elaborar un plan de negocio para dar inicio a una empresa de base tecnológica.
- Que el alumno sea capaz de realizar, usando modelos de negocio, la concepción y presentación de una propuesta de negocio.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Evaluar**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Evaluar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Evaluar**)

7. TEMAS

Unidad 1: (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Emprendedor, emprendedurismo e innovación tecnológica • Modelos de negocio • Formación de equipos 	<ul style="list-style-type: none"> • Identificar características de los emprendedores [Familiarizarse] • Introducir modelos de negocio [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [BDN10], [OP10], [Gar+14]	

Unidad 2: (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión • Misión • La Propuesta de valor • Creatividad e invención • Tipos y fuentes de innovación • Estrategia y Tecnología • Escala y ámbito 	<ul style="list-style-type: none"> • Plantear correctamente la vision y misión de empresa [Usar] • Caracterizar una propuesta de valor innovadora [Evaluar] • Identificar los diversos tipos y fuentes de innovación [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [BDN10], [BD12], [Gar+14]	

Unidad 3: (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Estrategia de la Empresa • Barreras • Ventaja competitiva sostenible • Alianzas • Aprendizaje organizacional • Desarrollo y diseño de productos 	<ul style="list-style-type: none"> • Conocer estrategias empresariales [Familiarizarse] • Caracterizar barreras y ventajas competitivas [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [BDN10], [OP10], [Rie11], [Gar+14]	

Unidad 4: (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Creación de un nuevo negocio • El plan de negocio • Canvas • Elementos del Canvas 	<ul style="list-style-type: none"> • Conocer los elementos del modelo Canvas [Usar] • Elaborar un plan de negocio basado en el modelo Canvas [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [OP10], [BD12], [Gar+14]	

Unidad 5: (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Aceleración versus incubación • Customer Development • Lean Startup 	<ul style="list-style-type: none"> • Conocer y aplicar el modelo Customer Development [Usar] • Conocer y aplicar el modelo Lean Startup [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [BD12], [Rie11], [Gar+14]	

Unidad 6: (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Aspectos Legales y tributarios para la constitución de la empresa • Propiedad intelectual • Patentes • Copyrights y marca registrada • Objetivos de marketing y segmentos de mercado • Investigación de mercado y búsqueda de clientes 	<ul style="list-style-type: none"> • Conocer los aspectos legales necesarios para la formación de una empresa tecnológica [Familiarizarse] • Identificar segmentos de mercado y objetivos de marketing [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [BDN10], [Rie11], [Con96], [Rep97], [Gar+14]	

Unidad 7: (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Modelo de costos • Modelo de utilidades • Precio • Plan financiero • Formas de financiamiento • Fuentes de capital • Capital de riesgo 	<ul style="list-style-type: none"> • Definir un modelo de costos y utilidades [Evaluar] • Conocer las diversas fuentes de financiamiento [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [BDN10], [BD12], [Gar+14]	

Unidad 8: (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • The Elevator Pitch • Presentación • Negociación 	<ul style="list-style-type: none"> • Conocer las diversas formas de presentar propuestas de negocio [Familiarizarse] • Realizar la presentación de una propuesta de negocio [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [BDN10], [BD12], [Gar+14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [BD12] Steve Blank and Bob Dorf. *The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company*. K and S Ranch, 2012.
- [BDN10] Thomas Byers, Richard Dorf, and Andrew Nelson. *Technology Ventures: From Idea to Enterprise*. McGraw-Hill Science, 2010.
- [Con96] Congreso de la Republica del Perú. *Decreto Legislativo N° 823. Ley de la Propiedad Industrial*. El Peruano, 1996.
- [Gar+14] René Garzozzi-Pincay et al. *Planes de Negocios para Emprendedores*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.
- [OP10] Alexander Osterwalder and Yves Pigneur. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley, 2010.
- [Rep97] Congreso de la Republica del Peru. *Ley N° 26887. Ley General de Sociedades*. El Peruano, 1997.
- [Rie11] Eric Ries. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, 2011.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62070. Computación en la Sociedad (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	2
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Horas autónomas	:	64 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Ofrece una visión amplia de los aspectos éticos y profesionales relacionados con la computación. Los tópicos que se incluyen abarcan los aspectos éticos, sociales y políticos. Las dimensiones morales de la computación. Los métodos y herramientas de análisis. Administración de los recursos computacionales. Seguridad y control de los sistemas computacionales. Responsabilidades profesionales y éticas. Propiedad intelectual.

5. OBJETIVOS

- Hacer que el alumno entienda la importancia del cuidado y la ética en la transferencia y uso de la información.
- Inculcar en el alumno que las tendencias de mejoramiento de la tecnología, no debe ser llevada a degradar la moral de la sociedad.

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Evaluar**)
- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (**Evaluar**)

7. TEMAS

Unidad 1: Historia (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Pre-historia – El mundo antes de 1946. • Historia del hardware, software, redes. • Pioneros de la Computación. • Historia de Internet. 	<ul style="list-style-type: none"> • Identificar importantes tendencias en la historia del campo de la computación [Familiarizarse] • Identificar las contribuciones de varios pioneros en el campo de la computación [Familiarizarse] • Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse] • Comparar la vida diaria antes y después de la llegada de los ordenadores personales y el Internet [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [LL04], [McL00]	

Unidad 2: Contexto Social (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Implicancias sociales de la computación en un mundo conectado en red. • Impacto de los medios sociales en el individualismo, colectivismo y en la cultura. • Crecimiento y control de la Internet • A menudo se refiere como la brecha digital, las diferencias en el acceso a los recursos de la tecnología digital y sus ramificaciones resultantes para el género, la clase, la etnia, la geografía, y/o los países subdesarrollados. • Los problemas de accesibilidad, incluyendo los requisitos legales. • Computación consciente del contexto. 	<ul style="list-style-type: none"> • Describir las formas positivas y negativas en las que la tecnología computacional (redes, computación móvil, <i>cloud computing</i>) altera los modos de interacción social en el plano personal [Familiarizarse] • Identificar los supuestos y valores incorporados en el hardware y el software de diseño de los desarrolladores, especialmente lo que se refiere a la facilidad de uso para diversas poblaciones incluyendo minorías poblaciones y los discapacitados [Usar] • Interpretar el contexto social de un determinado diseño y su aplicación [Evaluar] • Evaluar la eficacia de un diseño y aplicación dada a partir de datos empíricos [Familiarizarse] • Resumir las implicaciones de los medios sociales en el individualismo frente al colectivismo y la cultura [Familiarizarse] • Discuta cómo el acceso a Internet sirve como una fuerza liberadora para las personas que viven bajo las formas opresivas de gobierno; explicar la utilización los límites al acceso a Internet como herramientas de represión política y social [Familiarizarse] • Analizar los pros y los contras de la dependencia de la computación en la implementación de la democracia (por ejemplo, prestación de servicios sociales, votación electrónica) [Familiarizarse] • Describir el impacto de la escasa representación de las diversas poblaciones en la profesión (por ejemplo, la cultura de la industria, la diversidad de productos) [Usar] • Explicar las consecuencias de la sensibilidad al contexto en los sistemas de computación ubicua [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [LL04], [McL00]	

Unidad 3: Herramientas de Análisis (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Argumentación ética. • Teorías éticas y toma de decisiones. • Suposiciones morales y valores. 	<ul style="list-style-type: none"> • Evaluar las posiciones de las partes interesadas en una situación dada [Familiarizarse] • Analizar errores lógicos básicos en una discusión [Usar] • Analizar un argumento para identificar premisas y la conclusión [Familiarizarse] • Ilustrar el uso de ejemplo y analogía en el argumento ético [Familiarizarse] • Evaluar compensaciones éticos / sociales en las decisiones técnicas [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [LL04], [McL00]	

Unidad 4: Ética Profesional (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Community values and the laws by which we live. • La naturaleza del profesionalismo incluido el cuidado, la atención y la disciplina, la responsabilidad fiduciaria y mentoría. • Mantenerse al día como profesional de computación en términos de familiaridad, herramientas, habilidades, marco legal y profesional, así como la capacidad de autoevaluarse y avances en el campo de la computación. • La certificación profesional, códigos de ética, conducta y práctica, como la ACM / IEEE-CS, SE, AITP, IFIP y las sociedades internacionales. • Rendición de cuentas, la responsabilidad y la confiabilidad (por ejemplo, la corrección de software, fiabilidad y seguridad, así como la confidencialidad ética de los profesionales de seguridad cibernética) • El papel del profesional de de computación en las políticas públicas. • Mantenimiento de la conciencia en relación a las consecuencias. • Disidencia ética y la denuncia de irregularidades. • La relación entre la cultura regional y dilemas éticos. • Tratar con el acoso y la discriminación. • Formas de credenciamiento profesional. • Políticas de uso aceptable para la computación en el lugar de trabajo. • Ergonomía y entornos de trabajo computacionales saludables. • Consideraciones a tiempos de entrega de mercado vs estándares de calidad profesional. 	<ul style="list-style-type: none"> • Identificar los problemas éticos que se plantean en el desarrollo de software y determinar cómo abordarlos técnica y éticamente [Usar] • Explicar la responsabilidad ética de velar por la corrección de software, confiabilidad y seguridad [Evaluar] • Describir los mecanismos que normalmente existen para que profesional se mantenga al día [Familiarizarse] • Describir las fortalezas y debilidades de códigos profesionales relevantes como expresiones de profesionalismo y guías para la toma de decisiones [Familiarizarse] • Analizar un problema mundial de computación, observando el papel de los profesionales y funcionarios del gobierno en el manejo de este problema [Familiarizarse] • Evaluar los códigos de ética profesional de la ACM, la Sociedad de Computación de la IEEE, y otras organizaciones [Familiarizarse] • Describir las formas en que los profesionales pueden contribuir a las políticas públicas [Familiarizarse] • Describir las consecuencias de la conducta profesional inadecuada [Usar] • Identificar las etapas progresivas en un incidente de denuncia de irregularidades [Usar] • Identificar ejemplos de cómo interactúa la cultura regional con dilemas éticos [Familiarizarse] • Investigar las formas de acoso, discriminación y formas de ayuda [Usar] • Examine las diversas formas de acreditación de profesionales [Usar] • Explicar la relación entre la ergonomía en los ambientes y la salud de las personas de computación [Usar] • Desarrollar un uso del computador/política de uso aceptable con medidas coercitivas [Familiarizarse] • Describir los problemas asociados con la presión de la industrias para centrarse en el tiempo de comercialización en comparación con la aplicación de normas de calidad profesional [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unidad 5: Propiedad Intelectual (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Fundamentos filosóficos de propiedad intelectual. • Derechos de propiedad intelectual. • Propiedad intelectual digital intangible (IDIP). • Fundamentos legales para protección de la propiedad intelectual. • Gestión de derechos digitales. • Copyrights, patentes, secretos de comercio, marcas registradas. • Plagiarismo. • Fundamentos del movimiento Open Source. • Piratería de Software. 	<ul style="list-style-type: none"> • Discute las bases filosóficas de la propiedad intelectual [Evaluar] • Discute la racionalidad de la protección legal de la propiedad intelectual [Familiarizarse] • Describe la legislación orientada a los delitos de derechos de autor digitales [Evaluar] • Critica la legislación orientada a los delitos digitales de derechos de autor [Familiarizarse] • Identifica ejemplos contemporáneos de propiedad intelectual digital intangible [Evaluar] • Justifica el uso de material con derechos de autor [Evaluar] [Familiarizarse] • Evalúa los asuntos éticos inherentes a diversos mecanismos de detección de plagio [Familiarizarse] • Interpreta el intento y la implementación de licencias de software [Familiarizarse] • Discute asuntos que involucran la seguridad de patentes en software [Familiarizarse] • Caracteriza y contrasta los conceptos de derechos de autor, patentes y de marcas comerciales [Familiarizarse] • Identifica los objetivos del movimiento de software libre [Evaluar] • Identifica la naturaleza global de la piratería de software [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unidad 6: Privacidad y Libertades Civiles (4 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Fundamentos filosóficos de derechos de privacidad.• Fundamentos legales de protección de privacidad.• Implicaciones de privacidad de recopilación de datos generalizada de bases de datos transaccionales, almacenes de datos, sistemas de vigilancia y la computación en la nube.• Ramificaciones de privacidad diferencial.• Soluciones basadas en la tecnología para la protección de la privacidad.• Legislación de privacidad en áreas de práctica.• Libertades civiles y diferencias culturales.• Libertad de expresión y sus limitaciones.	<ul style="list-style-type: none">• Discute las bases filosóficas para la protección legal de la privacidad personal [Familiarizarse]• Evalúa soluciones para amenazas a la privacidad en bases de datos transaccionales y almacenes de datos [Familiarizarse]• Describe los roles de la recolección de datos en la implementación de sistemas de vigilancia intrusiva (ejm. RFID, reconocimiento de rostro, cobro electrónico, computación móvil) [Familiarizarse]• Describe las ramificaciones de la privacidad diferenciada [Familiarizarse]• Investiga el impacto de soluciones tecnológicas a los problemas de privacidad [Familiarizarse]• Critica la intención, el valor potencial y la implementación de las diversas formas de legislación en privacidad [Familiarizarse]• Identifica estrategias que permitan la apropiada libertad de expresión [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unidad 7: Políticas de seguridad, Leyes y crímenes computacionales (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Ejemplos de delitos informáticos y reparación legal para delincuentes informáticos. • Ingeniería social, robo de identidad y recuperación. • Tópicos relacionados al uso de acceso indebido y las infracciones y materia de seguridad. • Motivaciones y ramificaciones del ciberterrorismo y el hacking criminal, cracking. • Efectos de malware, como virus, worms y Trojan horses. • Estrategias de prevención de Crimen. • Políticas de Seguridad. 	<ul style="list-style-type: none"> • Listar ejemplos clásicos de delitos informáticos y incidentes de ingeniería social con impacto social [Familiarizarse] • Identificar leyes que se aplican a delitos informáticos [Familiarizarse] • Describir la motivación y ramificaciones de ciberterrorismo y hackeo criminal [Familiarizarse] • Examinar los problemas éticos y legales relacionados con el mal uso de accesos y diversas violaciones en la seguridad [Familiarizarse] • Discutir el rol del profesional en seguridad y los problemas que están envueltos [Familiarizarse] • Investigar medidas que puedan ser consideradas por personas y organizaciones incluyendo al gobierno para prevenir o mitigar efectos indeseables de los delitos informáticos y robo de identidad [Familiarizarse] • Escribir una política de seguridad de una empresa, la cual incluye procedimientos para administrar contraseñas y monitorizar a los empleados [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unidad 8: Economía de la Computación (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Monopolio y sus implicaciones económicas. • Efecto del suministro de mano de obra calificada y la demanda sobre la calidad de los productos de computación. • Estrategias de precio en el dominio de la computación. • El fenómeno del desarrollo de software outsourcing y off-shoring; impactos en el empleo y la economía. • Consecuencias de la globalización para la profesión de Ciencias de la Computación. • Diferencias en acceso a recursos de computación y el posible efecto de los mismos. • Analisis costo/beneficio de trabajos con consideraciones para manufactura, hardware, software e implicaciones de ingeniería. • Costo estimado versus costo actual in relacion al costo total. • Emprendimiento: perspectivas y entrampamientos. • Efectos de red o economías de escala del lado de la demanda. • El uso de la ingeniería económica para hacer frente a las finanzas. 	<ul style="list-style-type: none"> • Resumir los fundamentos para los esfuerzos antimonopolio [Familiarizarse] • Identificar diversas maneras en que la industria de la tecnología de la información está afectada por la escasez de la oferta de trabajo [Familiarizarse] • Identificar la evolución de la estrategia de precios para el cálculo de los bienes y servicios [Familiarizarse] • Discutir los beneficios, los inconvenientes y las implicaciones de <i>off-shoring</i> y <i>outsourcing</i> [Familiarizarse] • Investigar y defender maneras de tratar las limitaciones en el acceso a la computación. [Usar] • Describir los beneficios económicos de efectos de la red [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Edi09a] Datamation Ediciones, ed. *Revista Datamation MC Ediciones*. 2009.
- [Edi09b] Datamation Ediciones, ed. *Understanding the Digital Economy*. 2009.
- [Edi10] Datamation Ediciones, ed. *Financial Times Mastering Information Management*. 2010.
- [LL04] Kenneth C. Laudon and Jane P. Laudon. *Sistemas de Información Gerencial*. Prentice Hall, 2004.
- [McL00] Raymond McLeod Jr. *Sistemas de Información Gerencial*. Prentice Hall, 2000.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62071. Computación Paralela y Distribuída (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	-
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	<ul style="list-style-type: none">• SFW52084. Estructuras de Datos Avanzadas. (6^{to} Sem)• EGI61009. Arquitectura de Computadores. (3^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La última década ha traído un crecimiento explosivo en computación con multiprocesadores, incluyendo los procesadores de varios núcleos y centros de datos distribuidos. Como resultado, la computación paralela y distribuida se ha convertido de ser un tema ampliamente electivo para ser uno de los principales componentes en la malla estudios en ciencia de la computación de pregrado. Tanto la computación paralela como la distribuida implica la ejecución simultánea de múltiples procesos, cuyas operaciones tienen el potencial para intercalarse de manera compleja. La computación paralela y distribuida construye sobre cimientos en muchas áreas, incluyendo la comprensión de los conceptos fundamentales de los sistemas, tales como: concurrencia y ejecución en paralelo, consistencia en el estado/manipulación de la memoria, y latencia. La comunicación y la coordinación entre los procesos tiene sus cimientos en el paso de mensajes y modelos de memoria compartida de la computación y conceptos algorítmicos como atomicidad, el consenso y espera condicional. El logro de aceleración en la práctica requiere una comprensión de algoritmos paralelos, estrategias para la descomposición problema, arquitectura de sistemas, estrategias de implementación y análisis de rendimiento. Los sistemas distribuidos destacan los problemas de la seguridad y tolerancia a fallos, hacen hincapié en el mantenimiento del estado replicado e introducen problemas adicionales en el campo de las redes de computadoras.

5. OBJETIVOS

- Que el alumno sea capaz de crear aplicaciones paralelas de mediana complejidad aprovechando eficientemente máquinas con múltiples núcleos.
- Que el alumno sea capaz de comparar aplicaciones secuenciales y paralelas.
- Que el alumno sea capaz de convertir, cuando la situación lo amerite, aplicaciones secuenciales a paralelas de forma eficiente.

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Evaluar**)
- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (**Evaluar**)

7. TEMAS

Unidad 1: Fundamentos de paralelismo (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Procesamiento Simultáneo Múltiple. • Metas del Paralelismo (ej. rendimiento) frente a Concurrencia (ej. control de acceso a recursos compartidos) • Paralelismo, comunicación, y coordinación: <ul style="list-style-type: none"> – Paralelismo, comunicación, y coordinación – Necesidad de Sincronización • Errores de Programación ausentes en programación secuencial: <ul style="list-style-type: none"> – Tipos de Datos (lectura/escritura simultánea o escritura/escritura compartida) – Tipos de Nivel más alto (interleavings violating program intention, no determinismo no deseado) – Falta de vida/progreso (deadlock, starvation) 	<ul style="list-style-type: none"> • Distinguir el uso de recursos computacionales para una respuesta mas rápida para administrar el acceso eficiente a un recurso compartido [Familiarizarse] • Distinguir múltiples estructuras de programación suficientes para la sincronización que pueden ser interimplementables pero tienen ventajas complementarias [Familiarizarse] • Distinguir datos de carrera (<i>data races</i>) a partir de carreras de mas alto nivel [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pac11], [Mat14], [quinnz], [Geo10]	

Unidad 2: Arquitecturas paralelas (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Procesadores mutlinúcleo. • Memoria compartida vs memoria distribuida. • Multiprocesamiento simétrico. • SIMD, procesamiento de vectores. • GPU, coprocesamiento. • Taxonomía de Flynn. • Soporte a nivel de instrucciones para programación paralela. <ul style="list-style-type: none"> – Instrucciones atómicas como Compare/Set (Comparar / Establecer) • Problemas de Memoria: <ul style="list-style-type: none"> – Caches multiprocesador y coherencia de cache – Acceso a Memoria no uniforme (NUMA) • Topologías. <ul style="list-style-type: none"> – Interconexiones – Clusters – Compartir recursos (p.e., buses e interconexiones) 	<ul style="list-style-type: none"> • Explicar las diferencias entre memoria distribuida y memoria compartida [Evaluar] • Describir la arquitectura SMP y observar sus principales características [Evaluar] • Distinguir los tipos de tareas que son adecuadas para máquinas SIMD [Usar] • Describir las ventajas y limitaciones de GPUs vs CPUs [Usar] • Explicar las características de cada clasificación en la taxonomía de Flynn [Usar] • Describir los desafíos para mantener la coherencia de la caché [Familiarizarse] • Describir los desafíos clave del desempeño en diferentes memorias y topologías de sistemas distribuidos [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pac11], [KH13], [SK10], [Geo10]	

Unidad 3: Descomposición en paralelo (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Necesidad de Comunicación y coordinación/sincronización. • Independencia y Particionamiento. • Conocimiento Básico del Concepto de Descomposición Paralela. • Descomposición basada en tareas: <ul style="list-style-type: none"> – Implementación de estrategias como hebras • Descomposición de Información Paralela <ul style="list-style-type: none"> – Estrategias como SIMD y MapReduce • Actores y Procesos Reactivos (solicitud de gestores) 	<ul style="list-style-type: none"> • Explicar por qué la sincronización es necesaria en un programa paralelo específico [Usar] • Identificar oportunidades para particionar un programa serial en módulos paralelos independientes [Familiarizarse] • Escribir un algoritmo paralelo correcto y escalable [Usar] • Paralelizar un algoritmo mediante la aplicación de descomposición basada en tareas [Usar] • Paralelizar un algoritmo mediante la aplicación de descomposición de datos en paralelo [Usar] • Escribir un programa usando actores y/o procesos reactivos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pac11], [Mat14], [Qui03], [Geo10]	

Unidad 4: Comunicación y coordinación (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Memoria Compartida. • La consistencia, y su papel en los lenguaje de programación garantías para los programas de carrera libre. • Pasos de Mensaje: <ul style="list-style-type: none"> – Mensajes Punto a Punto versus multicast (o basados en eventos) – Estilos para enviar y recibir mensajes Blocking vs non-blocking – Buffering de mensajes • Atomicidad: <ul style="list-style-type: none"> – Especificar y probar atomicidad y requerimientos de seguridad – Granularidad de accesos atómicos y actualizaciones, y uso de estructuras como secciones críticas o transacciones para describirlas – Exclusión mutua usando bloques, semáforos, monitores o estructuras relacionadas <ul style="list-style-type: none"> * Potencial para fallas y bloqueos (<i>deadlock</i>) (causas, condiciones, prevención) – Composición <ul style="list-style-type: none"> * Componiendo acciones atómicas granulares más grandes usando sincronización * Transacciones, incluyendo enfoques optimistas y conservadores • Consensos: <ul style="list-style-type: none"> – (Cíclicos) barreras, contadores y estructuras relacionadas • Acciones condicionales: <ul style="list-style-type: none"> – Espera condicional (p.e., empleando variables de condición) 	<ul style="list-style-type: none"> • Usar exclusión mutua para evitar una condición de carrera [Usar] • Dar un ejemplo de una ordenación de accesos entre actividades concurrentes (por ejemplo, un programa con condición de carrera) que no son secuencialmente consistentes [Familiarizarse] • Dar un ejemplo de un escenario en el que el bloqueo de mensajes enviados pueden dar <i>deadlock</i> [Usar] • Explicar cuándo y por qué mensajes de multidifusión (<i>multicast</i>) o basado en eventos puede ser preferible a otras alternativas [Familiarizarse] • Escribir un programa que termine correctamente cuando todo el conjunto de procesos concurrentes hayan sido completados [Usar] • Dar un ejemplo de un escenario en el que un intento optimista de actualización puede nunca completarse [Familiarizarse] • Usar semaforos o variables de condición para bloquear hebras hasta una necesaria precondition de mantenga [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pac11], [Mat14], [Qui03], [Geo10]	

Unidad 5: Análisis y programación de algoritmos paralelos (18 horas)

Competencias esperadas:

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Caminos críticos, el trabajo y la duración y la relación con la ley de Amdahl. • Aceleración y escalabilidad. • Naturalmente (vergonzosamente) algoritmos paralelos. • Patrones Algoritmicos paralelos (divide-y-conquista, map/reduce, amos-trabajadores, otros) <ul style="list-style-type: none"> – Algoritmos específicos (p.e., MergeSort paralelo) • Algoritmos de grafos paralelo (por ejemplo, la ruta más corta en paralelo, árbol de expansión paralela) • Cálculos de matriz paralelas. • Productor-consumidor y algoritmos paralelos segmentados. • Ejemplos de algoritmos paralelos no-escalables. 	<ul style="list-style-type: none"> • Definir: camino crítico, trabajo y <i>span</i> [Familiarizarse] • Calcular el trabajo y el <i>span</i> y determinar el camino crítico con respecto a un diagrama de ejecución paralela. [Usar] • Definir <i>speed-up</i> y explicar la noción de escalabilidad de un algoritmo en este sentido [Familiarizarse] • Identificar tareas independientes en un programa que debe ser paralelizado [Usar] • Representar características de una carga de trabajo que permita o evite que sea naturalmente paralelizable [Familiarizarse] • Implementar un algoritmo dividir y conquistar paralelo (y/o algoritmo de un grafo) y medir empíricamente su desempeño relativo a su analogo secuencial [Usar] • Descomponer un problema (por ejemplo, contar el número de ocurrencias de una palabra en un documento) via operaciones <i>map</i> y <i>reduce</i> [Usar] • Proporcionar un ejemplo de un problema que se corresponda con el paradigma productor-consumidor [Usar] • Dar ejemplos de problemas donde el uso de <i>pipelining</i> sería un medio eficaz para la paralelización [Usar] • Implementar un algoritmo de matriz paralela [Usar] • Identificar los problemas que surgen en los algoritmos del tipo productor-consumidor y los mecanismos que pueden utilizarse para superar dichos problemas [Usar]
<p>Aprendizaje autónomo</p>	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos <p>Lecturas : [Mat14], [Qui03], [Geo10]</p>	

Unidad 6: Desempeño en paralelo (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Equilibrio de carga. • La medición del desempeño. • Programación y contención. • Evaluación de la comunicación de arriba. • Gestión de datos: <ul style="list-style-type: none"> – Costos de comunicación no uniforme debidos a proximidad – Efectos de Cache (p.e., false sharing) – Manteniendo localidad espacial • Consumo de energía y gestión. 	<ul style="list-style-type: none"> • Detectar y corregir un desbalanceo de carga [Usar] • Calcular las implicaciones de la ley de Amdahl para un algoritmo paralelo particular [Usar] • Describir como la distribución/disposición de datos puede afectar a los costos de comunicación de un algoritmo [Familiarizarse] • Detectar y corregir una instancia de uso compartido falso (<i>false sharing</i>) [Usar] • Explicar el impacto de la planificación en el desempeño paralelo [Familiarizarse] • Explicar el impacto en el desempeño de la localidad de datos [Familiarizarse] • Explicar el impacto y los puntos de equilibrio relacionados al uso de energía en el desempeño paralelo [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pac11], [Mat14], [KH13], [SK10], [Geo10]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[Geo10] Gerhard Wellein Georg Hager. *Introduction to High Performance Computing for Scientists and Engineers (Chapman & Hall/CRC Computational Science)*. Ed. by CRC Press. 1st. 2010. ISBN: 978-1439811924.

- [KH13] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. 2nd. Morgan Kaufmann, 2013. ISBN: 978-0-12-415992-1.
- [Mat14] Norm Matloff. *Programming on Parallel Machines*. University of California, Davis, 2014. URL: <http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf>.
- [Pac11] Peter S. Pacheco. *An Introduction to Parallel Programming*. 1st. Morgan Kaufmann, 2011. ISBN: 978-0-12-374260-5.
- [Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. 1st. McGraw-Hill Education Group, 2003. ISBN: 0071232656.
- [SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. Addison-Wesley Professional, 2010. ISBN: 0131387685, 9780131387683.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62073. Seguridad en Computación (Obligatorio)

2. INFORMACIÓN GENERAL

- | | | |
|--------------------------|---|---|
| 2.1 Créditos | : | 3 |
| 2.2 Horas de teoría | : | 1 (Semanal) |
| 2.3 Horas de práctica | : | 2 (Semanal) |
| 2.4 Horas autónomas | : | 96 (horas) |
| 2.5 Duración del periodo | : | 16 semanas |
| 2.6 Condición | : | Obligatorio |
| 2.7 Modalidad | : | Presencial |
| 2.8 Prerrequisitos | : | <ul style="list-style-type: none">• IIS53004. Redes y Telecomunicaciones. (7^{mo} Sem)• ENG42032. English VI: RREL. (6^{to} Sem) |

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Hoy en día la información es uno de los activos más preciados en cualquier organización. Este curso está orientado a poder brindar al alumno los elementos de seguridad orientados a proteger la información de la organización y principalmente poder prevenir los posibles problemas relacionados con este rubro. Esta materia involucra el desarrollo de una actitud preventiva por parte del alumno en todas las áreas relacionadas al desarrollo de software.

5. OBJETIVOS

- Discutir a un nivel intermedio avanzado los fundamentos de la Seguridad Informática.
- Brindar los diferentes aspectos que presenta el código malicioso.
- Que el alumno conozca los conceptos de criptografía y seguridad en redes de computadoras.
- Discutir y analizar junto con el alumno los aspectos de la Seguridad en Internet.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Evaluar**)
- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (**Evaluar**)

7. TEMAS

Unidad 1: Fundamentos y Conceptos en Seguridad (25 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • CIA (Confidencialidad, Integridad, Disponibilidad) • Conceptos de riesgo, amenazas, vulnerabilidades, y los tipos de ataque . • Autenticación y autorización, control de acceso (vs. obligatoria discrecional) • Concepto de la confianza y la honradez . • Ética (revelación responsable) 	<ul style="list-style-type: none"> • Analizar las ventajas y desventajas de equilibrar las propiedades clave de seguridad(Confidenciabilidad, Integridad, Disponibilidad) [Familiarizarse] • Describir los conceptos de riesgo, amenazas, vulnerabilidades y vectores de ataque(incluyendo el hecho de que no existe tal cosa como la seguridad perfecta) [Familiarizarse] • Explicar los conceptos de autenticación, autorización, control de acceso [Familiarizarse] • Explicar el concepto de confianza y confiabilidad [Familiarizarse] • Reconocer de que hay problemas éticos más importantes que considerar en seguridad computacional, incluyendo problemas éticos asociados a arreglar o no arreglar vulnerabilidades y revelar o no revelar vulnerabilidades [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [WL14]	

Unidad 2: Principios de Diseño Seguro (25 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Menor privilegio y aislamiento.• Valores predeterminados a prueba de fallos.• Diseño abierto.• La seguridad de extremo a extremo.• La defensa en profundidad (por ejemplo, la programación defensiva, defensa en capas)• Diseño de seguridad.• Las tensiones entre la seguridad y otros objetivos de diseño.• Mediación completa.• El uso de componentes de seguridad vetados.• Economía del mecanismo (la reducción de la base informática de confianza, minimizar la superficie de ataque)• Seguridad utilizable.• Componibilidad de seguridad.• Prevención, detección y disuasión.	<ul style="list-style-type: none">• Describir el principio de privilegios mínimos y el aislamiento que se aplican al diseño del sistema [Familiarizarse]• Resumir el principio de prueba de fallos y negar por defecto [Familiarizarse]• Discutir las implicaciones de depender de diseño abierto o secreto de diseño para la seguridad [Familiarizarse]• Explicar los objetivos de seguridad de datos de extremo a extremo [Familiarizarse]• Discutir los beneficios de tener múltiples capas de defensas [Familiarizarse]• Por cada etapa en el ciclo de vida de un producto, describir que consideraciones de seguridad deberían ser evaluadas [Familiarizarse]• Describir el costo y ventajas y desventajas asociadas con el diseño de seguridad de un producto. [Familiarizarse]• Describir el concepto de mediación y el principio de mediación completa [Familiarizarse]• Conocer los componentes estándar para las operaciones de seguridad, en lugar de reinventar las operaciones fundamentales [Familiarizarse]• Explicar el concepto de computación confiable incluyendo base informática confiable y de la superficie de ataque y el principio de minimización de base informática confiable [Familiarizarse]• Discutir la importancia de la usabilidad en el diseño de mecanismos de seguridad [Familiarizarse]• Describir problemas de seguridad que surgen en los límites entre varios componentes [Familiarizarse]• Identificar los diferentes roles de mecanismos de prevención y mecanismos de eliminación/disuasión [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [WL14]	

Unidad 3: Programación Defensiva (25 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Validación de datos de entrada y sanitización• Elección del lenguaje de programación y lenguajes con tipos de datos seguro.• Ejemplos de validación de entrada de datos y sanitización de errores.<ul style="list-style-type: none">– Desbordamiento de búfer– Errores enteros– Inyección SQL– Vulnerabilidad XSS• Las condiciones de carrera.• Manejo correcto de las excepciones y comportamientos inesperados.• Uso correcto de los componentes de terceros.• Desplegar eficazmente las actualizaciones de seguridad.• Información de control de flujo.• Generando correctamente el azar con fines de seguridad.• Mecanismos para la detección y mitigación de datos de entrada y errores de sanitización.• Fuzzing• El análisis estático y análisis dinámico.• Programa de verificación.• Soporte del sistema operativo (por ejemplo, la asignación al azar del espacio de direcciones, canarios)• El soporte de hardware (por ejemplo, el DEP, TPM)	<ul style="list-style-type: none">• Explicar por que la validación de entrada y desinfección de datos es necesario en el frente del control contencioso del canal de entrada [Usar]• Explicar por que uno debería escoger para desallorar un programa en un lenguaje tipo seguro como Java, en contraste con un lenguaje de programación no seguro como C/C++ [Usar]• Clasificar los errores de validación de entrada común, y escribir correctamente el código de validación de entrada [Usar]• Demostrar el uso de un lenguaje de programación de alto nivel cómo prevenir una condición de competencia que ocurran y cómo manejar una excepción [Usar]• Demostrar la identificación y el manejo elegante de las condiciones de error [Familiarizarse]• Explique los riesgos de mal uso de las interfaces con código de terceros y cómo utilizar correctamente el código de terceros [Familiarizarse]• Discutir la necesidad de actualizar el software para corregir las vulnerabilidades de seguridad y la gestión del ciclo de vida de la corrección [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [WL14]	

Unidad 4: Ataques y Amenazas (25 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Atacante metas, capacidades y motivaciones (como economía sumergida, el espionaje digital, la guerra cibernética, las amenazas internas, hacktivismo, las amenazas persistentes avanzadas)• Los ejemplos de malware (por ejemplo, virus, gusanos, spyware, botnets, troyanos o rootkits)• Denegación de Servicio (DoS) y Denegación de Servicio Distribuida (DDoS)• Ingeniería social (por ejemplo, pescando)• Los ataques a la privacidad y el anonimato .• El malware / comunicaciones no deseadas, tales como canales encubiertos y esteganografía.	<ul style="list-style-type: none">• Describir tipos de ataques similares en contra de un sistema en particular [Familiarizarse]• Discutir los limitantes de las medidas en contra del malware (ejm. detección basada en firmas, detección de comportamiento) [Familiarizarse]• Identificar las instancias de los ataques de ingeniería social y de los ataques de negación de servicios [Familiarizarse]• Discutir como los ataques de negación de servicios puede ser identificados y reducido [Familiarizarse]• Describir los riesgos de la privacidad y del anonimato en aplicaciones comunmente usadas [Familiarizarse]• Discutir los conceptos de conversión de canales y otros procedimientos de filtrado de datos [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [WL14]	

Unidad 5: Seguridad de Red (25 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Red de amenazas y tipos de ataques específicos (por ejemplo, la denegación de servicio, spoofing, olfateando y la redirección del tráfico, el hombre en el medio, ataques integridad de los mensajes, los ataques de enrutamiento, y el análisis de tráfico)• El uso de cifrado de datos y seguridad de la red .• Arquitecturas para redes seguras (por ejemplo, los canales seguros, los protocolos de enrutamiento seguro, DNS seguro, VPN, protocolos de comunicación anónimos, aislamiento)• Los mecanismos de defensa y contramedidas (por ejemplo, monitoreo de red, detección de intrusos, firewalls, suplantación de identidad y protección DoS, honeypots, seguimientos)• Seguridad para redes inalámbricas, celulares .• Otras redes no cableadas (por ejemplo, ad hoc, sensor, y redes vehiculares)• Resistencia a la censura.• Gestión de la seguridad operativa de la red (por ejemplo, control de acceso a la red configure)	<ul style="list-style-type: none">• Describir las diferentes categorías de amenazas y ataques en redes [Familiarizarse]• Describir las arquitecturas de criptografía de clave pública y privada y cómo las ICP brindan apoyo a la seguridad en redes [Familiarizarse]• Describir ventajas y limitaciones de las tecnologías de seguridad en cada capa de una torre de red [Familiarizarse]• Identificar los adecuados mecanismos de defensa y sus limitaciones dada una amenaza de red [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [WL14]	

Unidad 6: Criptografía (25 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Terminología básica de criptografía cubriendo las nociones relacionadas con los diferentes socios (comunicación), canal seguro / inseguro, los atacantes y sus capacidades, cifrado, descifrado, llaves y sus características, firmas. • Tipos de cifrado (por ejemplo, cifrado César, cifrado affine), junto con los métodos de ataque típicas como el análisis de frecuencia. • Apoyo a la infraestructura de clave pública para la firma digital y el cifrado y sus desafíos. • Criptografía de clave simétrica: <ul style="list-style-type: none"> – El secreto perfecto y el cojín de una sola vez – Modos de funcionamiento para la seguridad semántica y encriptación autenticada (por ejemplo, cifrar-entonces-MAC, OCB, GCM) – Integridad de los mensajes (por ejemplo, CMAC, HMAC) • La criptografía de clave pública: <ul style="list-style-type: none"> – Permutación de trampa, por ejemplo, RSA – Cifrado de clave pública, por ejemplo, el cifrado RSA, cifrado El Gamal – Las firmas digitales – Infraestructura de clave pública (PKI) y certificados – Supuestos de dureza, por ejemplo, Diffie-Hellman, factoring entero • Protocolos de intercambio de claves autenticadas, por ejemplo, TLS . • Primitivas criptográficas: <ul style="list-style-type: none"> – generadores pseudo-aleatorios y cifrados de flujo – cifrados de bloque (permutaciones pseudo-aleatorios), por ejemplo, AES – funciones de pseudo-aleatorios – funciones de hash, por ejemplo, SHA2, resistencia colisión – códigos de autenticación de mensaje – funciones derivaciones clave 	<ul style="list-style-type: none"> • Describir el propósito de la Criptografía y listar formas en las cuales es usada en comunicación de datos [Familiarizarse] • Definir los siguientes términos: Cifrado, Criptoanálisis, Algoritmo Criptográfico, y Criptología y describe dos métodos básicos (cifrados) para transformar texto plano en un texto cifrado [Familiarizarse] • Discutir la importancia de los números primos en criptografía y explicar su uso en algoritmos criptográficos [Familiarizarse] • Ilustrar como medir la entropía y como generar aleatoriedad criptográfica [Usar] • Usa primitivas de clave pública y sus aplicaciones [Usar] • Explicar como los protocolos de intercambio de claves trabajan y como es que pueden fallar [Familiarizarse] • Discutir protocolos criptográficos y sus propiedades [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [WL14]	

Unidad 7: Seguridad en la Web (25 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">● Modelo de seguridad Web<ul style="list-style-type: none">– Modelo de seguridad del navegador incluida la política de mismo origen– Los límites de confianza de cliente-servidor, por ejemplo, no pueden depender de la ejecución segura en el cliente● Gestión de sesiones, la autenticación:<ul style="list-style-type: none">– Single Sign-On– HTTPS y certificados● Vulnerabilidades de las aplicaciones y defensas :<ul style="list-style-type: none">– Inyección SQL– XSS– CSRF● Seguridad del lado del cliente :<ul style="list-style-type: none">– Política de seguridad Cookies– Extensiones de seguridad HTTP, por ejemplo HSTS– Plugins, extensiones y aplicaciones web– Seguimiento de los usuarios Web● Herramientas de seguridad del lado del servidor, por ejemplo, los cortafuegos de aplicación Web (WAFS) y fuzzers	<ul style="list-style-type: none">● Describe el modelo de seguridad de los navegadores incluyendo las políticas del mismo origen y modelos de amenazas en seguridad web [Familiarizarse]● Discutir los conceptos de sesiones web, canales de comunicación seguros tales como Seguridad en la Capa de Transporte(<i>TLS</i>) y la importancia de certificados de seguridad, autenticación incluyendo inicio de sesión único, como OAuth y Lenguaje de Marcado para Confirmaciones de Seguridad(<i>SAML</i>) [Familiarizarse]● Investigar los tipos comunes de vulnerabilidades y ataques en las aplicaciones web, y defensas contra ellos [Familiarizarse]● Utilice las funciones de seguridad del lado del cliente [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">● Desarrollo de ejercicios prácticos	
Lecturas : [WL14]	

Unidad 8: Seguridad de plataformas (25 horas)**Competencias esperadas:**

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Integridad de código y firma de código.• Arranque seguro, arranque medido, y la raíz de confianza.• Testimonio.• TPM y coprocesadores seguros.• Las amenazas de seguridad de los periféricos, por ejemplo, DMA, IOMMU.• Ataques físicos: troyanos de hardware, sondas de memoria, ataques de arranque en frío.• Seguridad de dispositivos integrados, por ejemplo, dispositivos médicos, automóviles.• Ruta confiable.	<ul style="list-style-type: none">• Explica el concepto de integridad de código y firma de códigos, así como el alcance al cual se aplica [Familiarizarse]• Discute los conceptos del origen de la confidencialidad y el de los procesos de arranque y carga segura [Familiarizarse]• Describe los mecanismos de arresto remoto de la integridad de un sistema [Familiarizarse]• Resume las metas y las primitivas claves de los modelos de plataforma confiable (TPM) [Familiarizarse]• Identifica las amenazas de conectar periféricos en un dispositivo [Familiarizarse]• Identifica ataques físicos y sus medidas de control [Familiarizarse]• Identifica ataques en plataformas con hardware que no son del tipo PC [Familiarizarse]• Discute los conceptos y la importancia de ruta confiable [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [WL14]	

Unidad 9: Investigación digital (Digital Forensics) (25 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Principios básicos y metodologías de análisis digital forense. • Diseñar sistemas con necesidades forenses en mente. • Reglas de Evidencia - conceptos generales y las diferencias entre las jurisdicciones y la Cadena de Custodia. • Búsqueda y captura de comprobación: requisitos legales y de procedimiento. • Métodos y normas de evidencia digital. • Las técnicas y los estándares para la conservación de los datos. • Cuestiones legales y reportes incluyendo el trabajo como perito. • Investigación digital de los sistema de archivos. • Los forenses de aplicación. • Investigación digital en la web. • Investigación digital en redes. • Investigación digital en dispositivos móviles. • Ataques al computador/red/sistema. • Detección e investigación de ataque. • Contra investigación digital. 	<ul style="list-style-type: none"> • Describe qué es una investigación digital, las fuentes de evidencia digital, y los límites de técnicas forenses [Familiarizarse] • Explica como diseñar software de apoyo a técnicas forenses [Familiarizarse] • Describe los requisitos legales para usar datos recuperados [Familiarizarse] • Describe el proceso de recolección de evidencia desde el tiempo en que se identifico el requisito hasta la colocación de los datos [Familiarizarse] • Describe como se realiza la recolección de datos y el adecuado almacenamiento de los datos originales y de la copia forense [Familiarizarse] • Realiza recolección de datos en un disco duro [Usar] • Describe la responsabilidad y obligación de una persona mientras testifica como un examinador forense [Familiarizarse] • Recupera datos basados en un determinado término de búsqueda en una imagen del sistema [Usar] • Reconstruye el historial de una aplicación a partir de los artefactos de la aplicación [Familiarizarse] • Reconstruye el historial de navegación web de los artefactos web [Familiarizarse] • Captura e interpreta el tráfico de red [Familiarizarse] • Discute los retos asociados con técnicas forenses de dispositivos móviles [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [WL14]	

Unidad 10: Seguridad en Ingeniería de Software (25 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • La construcción de la seguridad en el ciclo de vida de desarrollo de software. • Principios y patrones de diseño seguros. • Especificaciones de software seguros y requisitos. • Prácticas de desarrollo de software de seguros. • Asegure probar el proceso de las pruebas de que se cumplan los requisitos de seguridad (incluyendo análisis estático y dinámico) 	<ul style="list-style-type: none"> • Describir los requisitos para la integración de la seguridad en el SDL [Familiarizarse] • Aplicar los conceptos de los principios de diseño para mecanismos de protección, los principios para seguridad de software (Viega and McGraw) y los principios de diseño de seguridad (Morrie Gasser) en un proyecto de desarrollo de software [Familiarizarse] • Desarrollar especificaciones para un esfuerzo de desarrollo de software que especifica completamente los requisitos funcionales y se identifican las rutas de ejecución esperadas [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [WL14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[WL14] Stallings. W and Brown. L. *Computer Security: Principles and Practice*. Pearson Education, Limited, 2014. ISBN: 9780133773927.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62074. Proyecto de Final de Carrera I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Horas autónomas	:	96 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52081. Metodología de la Investigación en Computación. (7 ^{mo} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso tiene por objetivo que el alumno pueda realizar un estudio del estado del arte de un que el alumno ha elegido como tema para su tesis.

5. OBJETIVOS

- Que el alumno realice una investigación inicial en un tema específico realizando el estudio del estado del arte del tema elegido.
- Que el alumno muestre dominio en el tema de la línea de investigación elegida.
- Que el alumno elija un docente que domine el de investigación elegida como asesor.
- Los entregables de este curso son:

Avance parcial: Bibliografía sólida y avance de un Reporte Técnico.

Final: Reporte Técnico con experimentos preliminares comparativos que demuestren que el alumno ya conoce las técnicas existentes en el área de su proyecto y elegir a un docente que domine el área de su proyecto como asesor de su proyecto.

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Evaluar**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Evaluar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Evaluar**)

7. TEMAS

Unidad 1: Levantamiento del estado del arte (60 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Realizar un estudio profundo del estado del arte en un determinado t3pico del 3rea de Computaci3n. Redacci3n de art3culos t3cnicos en computaci3n. 	<ul style="list-style-type: none"> Hacer un levantamiento bibliogr3fico del estado del arte del tema escogido (esto significa muy probablemente 1 o 2 cap3tulos de marco te3rico adem3s de la introducci3n que es el cap3tulo I de la tesis) [Usar] Redactar un documento en latex en formato articulo (<i>paper</i>) con mayor calidad que en Proyecto I (dominar tablas, figuras, ecuaciones, 3ndices, bibtex, referencias cruzadas, citas, pstricks) [Usar] Tratar de hacer las presentaciones utilizando prosper [Usar] Mostrar experimentos b3sicos [Usar] Elegir un asesor que domine el 3rea de investigaci3n realizada [Usar]
Aprendizaje aut3nomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios pr3cticos 	
Lecturas : [IEE08], [Ass08], [Cit08]	

8. PLAN DE TRABAJO

8.1 Metodolog3a

Se fomenta la participaci3n individual y en equipo para exponer sus ideas, motiv3ndolos con puntos adicionales en las diferentes etapas de la evaluaci3n del curso.

8.2 Sesiones Te3ricas

Las sesiones de teor3a se llevan a cabo en clases magistrales donde se realizar3n actividades que propicien un aprendizaje activo, con din3micas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Pr3cticas

Las sesiones pr3cticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos pr3cticos mediante planteamiento de problemas, la resoluci3n de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACI3N

Cada uno de los rubros del esquema de evaluaci3n y la nota final del curso son redondeados a n3meros enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluaci3n permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluaci3n Permanente' mantendr3n su c3lculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluaci3n Continua	70%			
Pr3ctica Calificada	70%			
Pr3ctica Calificada ₁		Se elimina la pr3ctica con la menor nota	4	No
Pr3ctica Calificada ₂		Se elimina la pr3ctica con la menor nota	8	No
Pr3ctica Calificada ₃		Se elimina la pr3ctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAF3A B3SICA

[Ass08] Association for Computing Machinery. *Digital Library*. <http://portal.acm.org/dl.cfm>. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Library*. <http://citeseer.ist.psu.edu>. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray*. <http://www.computer.org/publications/dlib>. IEEE-Computer Society, 2008.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62075. Matemática aplicada a la computación (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 2
- 2.2 Horas de teoría : 1 (Semanal)
- 2.3 Horas de práctica : 1 (Semanal)
- 2.4 Horas autónomas : 64 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : MAC41002. Cálculo de varias Variables. (3^{er} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Write justification for this course here ...

5. OBJETIVOS

- Write your first goal here.
- Write your second goal here.
- Just in case you need more goals write them here

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Evaluar**)

7. TEMAS

Unidad 1: title for the unit goes here (0 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Topic1• Topic2• Topic3	<ul style="list-style-type: none">• Learning outcome1 [Levelforthislearningoutcome].• i.e Apply computing in complex problems [Usar].• Create a search engine [Evaluar].• Study data structures [Familiarizarse].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Bibitem1], [Bibitem2]	

Unidad 2: another unit goes here (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Topic1 	<ul style="list-style-type: none"> • Learning outcome xyz [Levelforthislearningoutcome].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Bibitem3], [Bibitem1]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62072. Visión Computacional (Electivo)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 4
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : 2 (Semanal)
- 2.4 Horas autónomas : 128 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Electivo
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : SFW52079. Aprendizaje Automático. (7^{mo} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La investigación en Inteligencia Artificial ha conducido al desarrollo de numerosas tónicas relevantes, dirigidas a la automatización de la inteligencia humana, dando una visión panorámica de diferentes algoritmos que simulan los diferentes aspectos del comportamiento y la inteligencia del ser humano.

5. OBJETIVOS

- Evaluar las posibilidades de simulación de la inteligencia, para lo cual se estudiarán las técnicas de modelización del conocimiento.

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Cuestiones fundamentales (2 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• ...• ...• ...	<ul style="list-style-type: none">• ... [Usar]• ... [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [De 06], [Pon+14]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. CRC Press, 2006.
- [Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62076. Biología Computacional (Electivo)

2. INFORMACIÓN GENERAL

- | | | |
|--------------------------|---|--|
| 2.1 Créditos | : | 4 |
| 2.2 Horas de teoría | : | 2 (Semanal) |
| 2.3 Horas de práctica | : | 2 (Semanal) |
| 2.4 Horas autónomas | : | 128 (horas) |
| 2.5 Duración del periodo | : | 16 semanas |
| 2.6 Condición | : | Electivo |
| 2.7 Modalidad | : | Presencial |
| 2.8 Prerrequisitos | : | <ul style="list-style-type: none">• SFW52089. Análisis y Diseño de Algoritmos. (5^{to} Sem)• BSC51001. Biología. (7^{mo} Sem) |

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El uso de métodos computacionales en las ciencias biológicas se ha convertido en una de las herramientas claves para el campo de la biología molecular, siendo parte fundamental en las investigaciones de esta área.

En Biología Molecular, existen diversas aplicaciones que involucran tanto al ADN, al análisis de proteínas o al secuenciamiento del genoma humano, que dependen de métodos computacionales. Muchos de estos problemas son realmente complejos y tratan con grandes conjuntos de datos.

Este curso puede ser aprovechado para ver casos de uso concretos de varias áreas de conocimiento de Ciencia de la Computación como: Lenguajes de Programación (PL), Algoritmos y Complejidad (AL), Probabilidades y Estadística, Manejo de Información (IM), Sistemas Inteligentes (IS).

5. OBJETIVOS

- Que el alumno tenga un conocimiento sólido de los problemas biológicos moleculares que desafían a la computación.
- Que el alumno sea capaz de abstraer la esencia de los diversos problemas biológicos para plantear soluciones usando sus conocimientos de Ciencia de la Computación.

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Introducción a la Biología Molecular (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • empty • empty • empty 	<ul style="list-style-type: none"> • empty [Familiarizarse] • empty [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [CB00], [SM97]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction*. 279 pages. John Wiley & Sons Ltd., 2000.

[SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology*. Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62064. Proyecto de Final de Carrera II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Horas autónomas	:	96 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW62074. Proyecto de Final de Carrera I. (8 ^{vo} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso tiene por objetivo que el alumno concluya su proyecto de tesis.

5. OBJETIVOS

- Que el alumno este en la capacidad de presentar formalmente su proyecto de tesis con el marco teórico y levantamiento bibliográfico completo.
- Que el alumno domine el estado del arte de su área de investigación.
- Los entregables de este curso son:

Avance parcial: Avance del plan de tesis incluyendo motivación y contexto, definición del problema, objetivos, cronograma de actividades hasta el proyecto final de tesis y el estado del arte del tema abordado.

Final: Plan de tesis completo y Avance de la Tesis incluyendo los capítulos de marco teórico, trabajos relacionados y resultados (formales o estadísticos) preliminares orientados a su tema de tesis.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Evaluar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Evaluar**)
- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (**Evaluar**)

7. TEMAS

Unidad 1: Proyecto de Tesis (30 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Proyecto de Tesis. 	<ul style="list-style-type: none"> • Descripción del formato utilizado por la Universidad para el plan de tesis [Evaluar] • Concluir el plan del proyecto de tesis[Evaluar] • Presentar el estado del arte del tema de tesis (50%)[Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [IEE08], [Ass08], [Cit08]	

Unidad 2: Avance de Tesis (30 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Avance de Tesis. 	<ul style="list-style-type: none"> • Descripción del formato utilizado por la Universidad para la tesis[Evaluar] • Concluir el capítulo del Marco Teórico de la Tesis[Evaluar] • Concluir el capítulo de Trabajos Relacionados (35%)[Evaluar] • Planear, desarrollar y presentar resultados (formales o estadísticos) de experimentos orientados a su tema de tesis (35%)[Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [IEE08], [Ass08], [Cit08]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Ass08] Association for Computing Machinery. *Digital Libray*. <http://portal.acm.org/dl.cfm>. Association for Computing Machinery, 2008.
- [Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. <http://citeseer.ist.psu.edu>. College of Information Sciences and Technology, Penn State University, 2008.
- [IEE08] IEEE-Computer Society. *Digital Libray*. <http://www.computer.org/publications/dlib>. IEEE-Computer Society, 2008.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62065. Bioinformática (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	1 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	BSC51001. Biología. (7 ^{mo} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El uso de métodos computacionales en las ciencias biológicas se ha convertido en una de las herramientas claves para el campo de la biología molecular, siendo parte fundamental en las investigaciones de esta área.

En Biología Molecular, existen diversas aplicaciones que involucran tanto al ADN, al análisis de proteínas o al secuenciamiento del genoma humano, que dependen de métodos computacionales. Muchos de estos problemas son realmente complejos y tratan con grandes conjuntos de datos.

Este curso puede ser aprovechado para ver casos de uso concretos de varias áreas de conocimiento de Ciencia de la Computación como: Lenguajes de Programación (PL), Algoritmos y Complejidad (AL), Probabilidades y Estadística, Manejo de Información (IM), Sistemas Inteligentes (IS).

5. OBJETIVOS

- Que el alumno tenga un conocimiento sólido de los problemas biológicos moleculares que desafían a la computación.
- Que el alumno sea capaz de abstraer la esencia de los diversos problemas biológicos para plantear soluciones usando sus conocimientos de Ciencia de la Computación

6. COMPETENCIAS

- 2) Aplica tópicos de investigación, metodologías, técnicas y mejores prácticas de la Ingeniería de Software para la construcción de soluciones en base al diseño, desarrollo, pruebas, implementación, documentación y mejora continua del Software.. (**Evaluar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Evaluar**)

7. TEMAS

Unidad 1: Introducción a la Biología Molecular (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Revisión de la química orgánica: moléculas y macromoléculas, azúcares, ácidos nucleicos, nucleótidos, ARN, ADN, proteínas, aminoácidos y niveles de estructura en las proteínas. • El dogma de la vida: del ADN a las proteínas, transcripción, traducción, síntesis de proteínas • Estudio del genoma: Mapas y secuencias, técnicas específicas 	<ul style="list-style-type: none"> • Lograr un conocimiento general de los tópicos más importantes en Biología Molecular. [Familiarizarse] • Entender que los problemas biológicos son un desafío al mundo computacional. [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [CB00], [SM97]	

Unidad 2: Comparación de Secuencias (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Secuencias de nucleótidos y secuencias de aminoácidos. • Alineamiento de secuencias, el problema de alineamiento por pares, búsqueda exhaustiva, Programación dinámica, alineamiento global, alineamiento local, penalización por gaps • Comparación de múltiples secuencias: suma de pares, análisis de complejidad por programación dinámica, heurísticas de alineamiento, algoritmo estrella, algoritmos de alineamiento progresivo. 	<ul style="list-style-type: none"> • Entender y solucionar el problema de alineamiento de un par de secuencias. [Usar] • Comprender y solucionar el problema de alineamiento de múltiples secuencias. [Usar] • Conocer los diversos algoritmos de alineamiento de secuencias existentes en la literatura. [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [CB00], [SM97], [Pev00]	

Unidad 3: Árboles Filogenéticos (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Filogenia: Introducción y relaciones filogenéticas. • Árboles Filogenéticos: definición, tipo de árboles, problema de búsqueda y reconstrucción de árboles • Métodos de Reconstrucción: métodos por parsimonia, métodos por distancia, métodos por máxima verosimilitud, confianza de los árboles reconstruidos 	<ul style="list-style-type: none"> • Comprender el concepto de filogenia, árboles filogenéticos y la diferencia metodológica entre biología y biología molecular. [Familiarizarse] • Comprender el problema de reconstrucción de árboles filogenéticos, conocer y aplicar los principales algoritmos para reconstrucción de árboles filogenéticos. [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [CB00], [SM97], [Pev00]	

Unidad 4: Ensamblaje de Secuencias de ADN (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Fundamento biológico: caso ideal, dificultades, métodos alternativos para secuenciamiento de ADN • Modelos formales de ensamblaje: <i>Shortest Common Superstring, Reconstruction, Multicontig</i> • Algoritmos para ensamblaje de secuencias: representación de overlaps, caminos para crear <i>superstrings</i>, algoritmo voraz, grafos acíclicos. • Heurísticas para ensamblaje: búsqueda de sobreposiciones, ordenación de fragmentos, alineamientos y consenso. 	<ul style="list-style-type: none"> • Comprender el desafío computacional que ofrece el problema de Ensamblaje de Secuencias. [Familiarizarse] • Entender el principio de modelo formal para ensamblaje. [Evaluar] • Conocer las principales heurísticas para el problema de ensamblaje de secuencias ADN [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [SM97], [Alu06]	

Unidad 5: Estructuras secundarias y terciarias (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Estructuras moleculares: primaria, secundaria, terciaria, cuaternaria. • Predicción de estructuras secundarias de ARN: modelo formal, energía de pares, estructuras con bases independientes, solución con Programación Dinámica, estructuras con bucles. • <i>Protein folding</i>: Estructuras en proteínas, problema de <i>protein folding</i>. • <i>Protein Threading</i>: Definiciones, Algoritmo <i>Branch & Bound</i>, <i>Branch & Bound</i> para <i>protein threading</i>. • <i>Structural Alignment</i>: definiciones, algoritmo DALI 	<ul style="list-style-type: none"> • Conocer las estructuras protéicas y la necesidad de métodos computacionales para la predicción de la geometría. [Familiarizarse] • Conocer los algoritmos de solución de problemas de predicción de estructuras secundarias ARN, y de estructuras en proteínas. [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [SM97], [CB00], [Alu06]	

Unidad 6: Modelos Probabilísticos en Biología Molecular (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Probabilidad: Variables aleatorias, Cadenas de Markov, Algoritmo de Metropoli-Hasting, Campos Aleatorios de Markov y Muestreador de Gibbs, Máxima Verosimilitud. • Modelos Ocultos de Markov (HMM), estimación de parámetros, algoritmo de Viterbi y método Baul-Welch, Aplicación en alineamientos de pares y múltiples, en detección de Motifs en proteínas, en ADN eucariótico, en familias de secuencias. • Filogenia Probabilística: Modelos probabilísticos de evolución, verosimilitud de alineamientos, verosimilitud para inferencia, comparación de métodos probabilísticos y no probabilísticos 	<ul style="list-style-type: none"> • Revisar conceptos de Modelos Probabilísticos y comprender su importancia en Biología Molecular Computacional. [Evaluar] • Conocer y aplicar Modelos Ocultos de Markov para varios análisis en Biología Molecular. [Usar] • Conocer la aplicación de modelos probabilísticos en Filogenia y compararlos con modelos no probabilísticos [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Dur+98], [CB00], [Alu06], [Kro+94]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Alu06] Srinivas Aluru, ed. *Handbook of Computational Molecular Biology*. Computer and Information Science Series. Boca Raton, FL: Chapman & Hall, CRC, 2006.
- [CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction*. 279 pages. John Wiley & Sons Ltd., 2000.
- [Dur+98] R. Durbin et al. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998, p. 357. ISBN: 9780521629713.

- [Kro+94] Anders Krogh et al. “Hidden Markov Models in Computational Biology, Applications to Protein Modeling”. In: *J Molecular Biology* 235 (1994), pp. 1501–1531.
- [Pev00] Pavel A. Pevzner. *Computational Molecular Biology: an Algorithmic Approach*. Cambridge, Massachusetts: The MIT Press, 2000.
- [SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology*. Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296. ISBN: 978-0-534-95262-4.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62068. Formación de Empresas de Base Tecnológica II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Horas autónomas	:	96 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW62069. Formación de Empresas de Base Tecnológica I. (8 ^{vo} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso tiene como objetivo dotar al futuro profesional de conocimientos, actitudes y aptitudes que le permitan formar su propia empresa de desarrollo de software y/o consultoría en informática. El curso está dividido en tres unidades: Valorización de Proyectos, Marketing de Servicios y Negociaciones. En la primera unidad se busca que el alumno pueda analizar y tomar decisiones en relación a la viabilidad de un proyecto y/o negocio.

En la segunda unidad se busca preparar al alumno para que este pueda llevar a cabo un plan de marketing satisfactorio del bien o servicio que su empresa pueda ofrecer al mercado. La tercera unidad busca desarrollar la capacidad negociadora de los participantes a través del entrenamiento vivencial y práctico y de los conocimientos teóricos que le permitan cerrar contrataciones donde tanto el cliente como el proveedor resulten ganadores. Consideramos estos temas sumamente críticos en las etapas de lanzamiento, consolidación y eventual relanzamiento de una empresa de base tecnológica.

5. OBJETIVOS

- Que el alumno comprenda y aplique la terminología y conceptos fundamentales de ingeniería económica que le permitan valorizar un proyecto para tomar la mejor decisión económica.
- Que el alumno adquiera las bases para formar su propia empresa de base tecnológica.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Evaluar**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Evaluar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Evaluar**)

7. TEMAS

Unidad 1: (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Introducción • Proceso de toma de decisiones • El valor del dinero en el tiempo • Tasa de interés y tasa de rendimiento • Interés simple e interés compuesto • Identificación de costos • Flujo de Caja Neto • Tasa de Retorno de Inversión (TIR) • Valor Presente Neto (VPN) • Valorización de Proyectos 	<ul style="list-style-type: none"> • Permitir al alumno tomar decisiones sobre como invertir mejor los fondos disponibles, fundamentadas en el análisis de los factores tanto económicos como no económicos que determinen la viabilidad de un emprendimiento. [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [BT06]	

Unidad 2: (30 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Introducción • Importancia del marketing en las empresas de servicios • El Proceso estratégico. • El Plan de Marketing • Marketing estratégico y marketing operativo • Segmentación, targeting y posicionamiento de servicios en mercados competitivos • Ciclo de vida del producto • Aspectos a considerar en la fijación de precios en servicios • El rol de la publicidad, las ventas y otras formas de comunicación • El comportamiento del consumidor en servicios • Fundamentos de marketing de servicios • Creación del modelo de servicio • Gestión de la calidad de servicio 	<ul style="list-style-type: none"> • Brindar las herramientas al alumno para que pueda identificar, analizar y aprovechar las oportunidades de marketing que generan valor en un emprendimiento. [Usar] • Lograr que el alumno conozca, entienda e identifique criterios, habilidades, métodos y procedimientos que permitan una adecuada formulación de estrategias de marketing en sectores y medios específicos como lo es una empresa de base tecnológica. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [KK06], [LW09]	

Unidad 3: (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Introducción. ¿Qué es una negociación? • Teoría de las necesidades de la negociación • La proceso de la negociación • Estilos de negociación • Teoría de juegos • El método Harvard de negociación 	<ul style="list-style-type: none"> • Conocer los puntos clave en el proceso de negociación. [Usar] • Establecer una metodología de negociación eficaz. [Usar] • Desarrollar destrezas y habilidades que permitan llevar a cabo una negociación exitosa. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [FUP96], [MM06]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [BT06] Leland Blank and Anthony Tarkin. *Ingeniería Económica*. McGraw Hill, México D.F., México, 2006.
- [FUP96] Roger Fisher, William Ury, and Bruce Patton. *Si... ¿de acuerdo! Cómo negociar sin ceder*. Norma, Barcelona, 1996.
- [KK06] Philip Kotler and Kevin L. Keller. *Dirección de Marketing*. Prentice Hall, México, 2006.
- [LW09] Christopher Lovelock and Jochen Wirtz. *Marketing de servicios. Personal, tecnología y estrategia*. Prentice Hall, México, 2009.
- [MM06] Fernando de Manuel Dasí and Rafael Martínez-Vilanova Martínez. *Técnicas de Negociación. Un método práctico*. Esic, Madrid, 2006.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62066. Tópicos en Ingeniería de Software (Electivo)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52077. Ingeniería de Software III. (7 ^{mo} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El desarrollo de software requiere del uso de mejores prácticas de desarrollo, gestión de proyectos de TI, manejo de equipos y uso eficiente y racional de frameworks de aseguramiento de la calidad y de Gobierno de Portfolios, estos elementos son pieza clave y transversal para el éxito del proceso productivo.

Este curso explora el diseño, selección, implementación y gestión de soluciones TI en las Organizaciones. El foco está en las aplicaciones y la infraestructura y su aplicación en el negocio.

5. OBJETIVOS

- Entender una variedad de frameworks para el análisis de arquitectura empresarial y la toma de decisiones
- Utilizar técnicas para la evaluación y gestión del riesgo en el portfolio de la empresa
- Evaluar y planificar la integración de tecnologías emergentes
- Entender el papel y el potencial de las TI para apoyar la gestión de procesos empresariales
- Entender los diferentes enfoques para modelar y mejorar los procesos de negocio
- Describir y comprender modelos de aseguramiento de la calidad como marco clave para el éxito de los proyectos de TI.
- Comprender y aplicar el framework de IT Governance como elemento clave para la gestión del portfolio de aplicaciones Empresariales

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Diseño de Software (18 horas)**Competencias esperadas:****Temas****Objetivos de Aprendizaje**

- Principios de diseño del sistema: niveles de abstracción (diseño arquitectónico y el diseño detallado), separación de intereses, ocultamiento de información, de acoplamiento y de cohesión, de reutilización de estructuras estándar.
- Diseño de paradigmas tales como diseño estructurado (descomposición funcional de arriba hacia abajo), el análisis orientado a objetos y diseño, orientado a eventos de diseño, diseño de nivel de componente, centrado datos estructurada, orientada a aspectos, orientado a la función, orientado al servicio.
- Modelos estructurales y de comportamiento de los diseños de software.
- Diseño de patrones.
- Relaciones entre los requisitos y diseños: La transformación de modelos, el diseño de los contratos, invariantes.
- Conceptos de arquitectura de software y arquitecturas estándar (por ejemplo, cliente-servidor, n-capas, transforman centrados, tubos y filtros).
- El uso de componentes de diseño: selección de componentes, diseño, adaptación y componentes de ensamblaje, componentes y patrones, componentes y objetos (por ejemplo, construir una GUI usando un estándar widget set)
- Diseños de refactorización utilizando patrones de diseño
- Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos.
- Medición y análisis de la calidad de un diseño.
- Compensaciones entre diferentes aspectos de la calidad.
- Aplicaciones en frameworks.
- Middleware: El paradigma de la orientación a objetos con middleware, requerimientos para correr y clasificar objetos, monitores de procesamiento de transacciones y el sistema de flujo de trabajo.
- Principales diseños de seguridad y codificación (cross-reference IAS/Principles of secure design).
 - Principio de privilegios mínimos
 - Principio de falla segura por defecto
 - Principio de aceptabilidad psicológica

- Formular los principios de diseño, incluyendo la separación de problemas, ocultación de información, acoplamiento y cohesión, y la encapsulación [Usar]
- Usar un paradigma de diseño para diseñar un sistema de software básico y explicar cómo los principios de diseño del sistema se han aplicado en este diseño [Usar]
- Construir modelos del diseño de un sistema de software simple los cuales son apropiado para el paradigma utilizado para diseñarlo [Usar]
- En el contexto de un paradigma de diseño simple, describir uno o más patrones de diseño que podrían ser aplicables al diseño de un sistema de software simple [Usar]
- Para un sistema simple adecuado para una situación dada, discutir y seleccionar un paradigma de diseño apropiado [Usar]
- Crear modelos apropiados para la estructura y el comportamiento de los productos de software desde la especificaciones de requisitos [Usar]
- Explicar las relaciones entre los requisitos para un producto de software y su diseño, utilizando los modelos apropiados [Usar]
- Para el diseño de un sistema de software simple dentro del contexto de un único paradigma de diseño, describir la arquitectura de software de ese sistema [Usar]
- Dado un diseño de alto nivel, identificar la arquitectura de software mediante la diferenciación entre las arquitecturas comunes de software, tales como 3 capas (*3-tier*), *pipe-and-filter*, y cliente-servidor [Usar]
- Investigar el impacto de la selección arquitecturas de software en el diseño de un sistema simple [Usar]
- Aplicar ejemplos simples de patrones en un diseño de software [Usar]
- Describir una manera de refactorar y discutir cuando esto debe ser aplicado [Usar]
- Seleccionar componentes adecuados para el uso en un diseño de un producto de software [Usar]
- Explicar cómo los componentes deben ser adaptados para ser usados en el diseño de un producto de software [Usar]
- Diseñar un contrato para un típico componente de software pequeño para el uso de un dado sistema [Usar]
- Discutir y seleccionar la arquitectura de software adecuada para un sistema de software simple para un dado escenario [Usar]

Unidad 2: Gestión de Proyectos de Software (14 horas)

Competencias esperadas:

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> ● La participación del equipo: <ul style="list-style-type: none"> – Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo – Roles y responsabilidades en un equipo de software – Equipo de resolución de conflictos – Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura) ● Estimación de esfuerzo (a nivel personal) ● Riesgo. <ul style="list-style-type: none"> – El papel del riesgo en el ciclo de vida – Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de ● Gestión de equipos: <ul style="list-style-type: none"> – Organización de equipo y la toma de decisiones – Roles de identificación y asignación – Individual y el desempeño del equipo de evaluación ● Gestión de proyectos: <ul style="list-style-type: none"> – Programación y seguimiento de elementos – Herramientas de gestión de proyectos – Análisis de Costo/Beneficio ● Software de medición y técnicas de estimación. ● Aseguramiento de la calidad del software y el rol de las mediciones. ● Riesgo. <ul style="list-style-type: none"> – El papel del riesgo en el ciclo de vida – Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de ● En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas. 	<ul style="list-style-type: none"> ● Discutir los comportamientos comunes que contribuyen al buen funcionamiento de un equipo [Usar] ● Crear y seguir un programa para una reunión del equipo [Usar] ● Identificar y justificar las funciones necesarias en un equipo de desarrollo de software [Usar] ● Entender las fuentes, obstáculos y beneficios potenciales de un conflicto de equipo [Usar] ● Aplicar una estrategia de resolución de conflictos en un ambiente de equipo [Usar] ● Utilizar un método ad hoc para estimar el esfuerzo de desarrollo del software (ejemplo, tiempo) y comparar con el esfuerzo actual requerido [Usar] ● Listar varios ejemplos de los riesgos del software [Usar] ● Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Usar] ● Describir las diferentes categorías de riesgo en los sistemas de software [Usar] ● Demostrar a través de la colaboración de proyectos de equipo los elementos centrales de la construcción de equipos y gestión de equipos [Usar] ● Describir como la elección de modelos de procesos afectan la estructura organizacional de equipos y procesos de toma de decisiones [Usar] ● Crear un equipo mediante la identificación de los roles apropiados y la asignación de funciones a los miembros del equipo [Usar] ● Evaluar y retroalimentar a los equipos e individuos sobre su desempeño en un ambiente de equipo [Usar] ● Usando un software particular procesar, describir los aspectos de un proyecto que necesita ser planeado y monitoreado, (ejemplo, estimar el tamaño y esfuerzo, un horario, reasignación de recursos, control de configuración, gestión de cambios, identificación de riesgos en un proyecto y gestión) [Usar] ● Realizar el seguimiento del progreso de alguna etapa de un proyecto que utiliza métricas de proyectos apropiados [Usar] ● Comparar las técnicas simples de tamaño de software y estimación de costos [Usar] ● Usar una herramienta de gestión de proyectos para ayudar en la asignación y rastreo de tareas en un proyecto de desarrollo de software [Usar] ● Describir el impacto de la tolerancia de riesgos en el proceso de desarrollo de software [Usar]

Unidad 3: (14 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Administración del servicio como práctica. • Ciclo de vida del servicio. • Definiciones y conceptos genéricos. • Modelos y principios claves. • Procesos. • Tecnología y arquitectura. • Competencia y entrenamiento. 	<ul style="list-style-type: none"> • Utilizar y aplicar correctamente ITIL en el proceso de software. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Som17], [PM15]	

Unidad 4: (14 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Fundamentos e Introducción. • Frameworks de Control y IT Governance. 	<ul style="list-style-type: none"> • Utilizar y aplicar correctamente COBIT en el proceso de software. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Som17], [PM15]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.
- [Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62067. Tópicos en Computación Gráfica (Electivo)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW52080. Computación Gráfica. (7 ^{mo} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

En este curso se puede profundizar en alguno de los tópicos mencionados en el área de Computación Gráfica (*Graphics and Visual Computing* - GV).

Éste curso está destinado a realizar algún curso avanzado sugerido por la curricula de la ACM/IEEE. [Hug+13; HB90]

5. OBJETIVOS

- Que el alumno utilice técnicas de computación gráfica más sofisticadas que involucren estructuras de datos y algoritmos complejos.
- Que el alumno aplique los conceptos aprendidos para crear una aplicación sobre un problema real.
- Que el alumno investigue la posibilidad de crear un nuevo algoritmo y/o técnica nueva para resolver un problema real.

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Tópicos Avanzados en Computación Gráfica (0 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • CS355. Advanced Computer Graphics • CS356. Computer animation • CS313. Geometric Algorithms • CS357. Visualización • CS358. Virtual reality 	<ul style="list-style-type: none"> • Tópicos Avanzados en Computación Gráfica
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [MS16]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[HB90] Donald Hearn and Pauline Baker. *Computer Graphics in C*. Prentice Hall, 1990.

[Hug+13] John F. Hughes et al. *Computer Graphics - Principles and Practice 3rd Edition*. Addison-Wesley, 2013.

[MS16] Steve Marschner and Peter Shirley. *Fundamentals of Computer Graphics*. Fourth Edition. CRC Press, 2016. ISBN: ISBN-10: 1482229390.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

IIS64000. Big Data y Analítica de Datos (Obligatorio)

2. INFORMACIÓN GENERAL

- | | | |
|--------------------------|---|---|
| 2.1 Créditos | : | 4 |
| 2.2 Horas de teoría | : | - |
| 2.3 Horas de práctica | : | 2 (Semanal) |
| 2.4 Horas autónomas | : | 128 (horas) |
| 2.5 Duración del periodo | : | 16 semanas |
| 2.6 Condición | : | Obligatorio |
| 2.7 Modalidad | : | Presencial |
| 2.8 Prerrequisitos | : | <ul style="list-style-type: none">• SFW52090. Gerenciamiento de Datos II. (5^{to} Sem)• SFW62071. Computación Paralela y Distribuída. (8^{vo} Sem) |

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

En la actualidad conocer enfoques escalables para procesar y almacenar grande volúmenes de información (terabytes, petabytes e inclusive exabytes) es fundamental en cursos de ciencia de la computación. Cada día, cada hora, cada minuto se genera gran cantidad de información la cual necesitará ser procesada, almacenada, analizada.

5. OBJETIVOS

- Que el alumno sea capaz de crear aplicaciones paralelas para procesar grandes volúmenes de información.
- Que el alumno sea capaz de comparar las alternativas para el procesamiento de big data.
- Que el alumno sea capaz de proponer arquitecturas para una aplicación escalable.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Evaluar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Evaluar**)

7. TEMAS

Unidad 1: Introducción a Big Data (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión global sobre Cloud Computing • Visión global sobre Sistema de Archivos Distribuidos • Visión global sobre el modelo de programación MapReduce 	<ul style="list-style-type: none"> • Explicar el concepto de Cloud Computing desde el punto de vista de Big Data[Familiarizarse] • Explicar el concepto de los Sistema de Archivos Distribuidos [Familiarizarse] • Explicar el concepto del modelo de programación MapReduce[Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cou+11]	

Unidad 2: Hadoop (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión global de Hadoop. • Historia. • Estructura de Hadoop. • HDFS, Hadoop Distributed File System. • Modelo de Programación MapReduce 	<ul style="list-style-type: none"> • Entender y explicar la suite de Hadoop. [Familiarizarse] • Implementar soluciones usando el modelo de programación MapReduce. [Usar] • Entender la forma como se guardan los datos en el HDFS. [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HDF11], [BVS13]	

Unidad 3: Procesamiento de Grafos en larga escala (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Pregel: A System for Large-scale Graph Processing. • Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud. • Apache Giraph is an iterative graph processing system built for high scalability. 	<ul style="list-style-type: none"> • Entender y explicar la arquitectura del proyecto Pregel. [Familiarizarse] • Entender la arquitectura del proyecto GraphLab. [Familiarizarse] • Entender la arquitectura del proyecto Giraph. [Familiarizarse] • Implementar soluciones usando Pregel, GraphLab o Giraph. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Low+12], [Mal+10], [Bal+08]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.				
Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.				
	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Bal+08] Shumeet Baluja et al. "Video Suggestion and Discovery for Youtube: Taking Random Walks Through the View Graph". In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. Beijing, China: ACM, 2008, pp. 895–904. ISBN: 978-1-60558-085-2. DOI: 10.1145/1367497.1367618. URL: <http://doi.acm.org/10.1145/1367497.1367618>.
- [BVS13] Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi. *Mastering Cloud Computing: Foundations and Applications Programming*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN: 9780124095397, 9780124114548.
- [Cou+11] George Coulouris et al. *Distributed Systems: Concepts and Design*. 5th. USA: Addison-Wesley Publishing Company, 2011. ISBN: 0132143011, 9780132143011.
- [HDF11] Kai Hwang, Jack Dongarra, and Geoffrey C. Fox. *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123858801, 9780123858801.
- [Low+12] Yucheng Low et al. "Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud". In: *Proc. VLDB Endow.* 5.8 (Apr. 2012), pp. 716–727. ISSN: 2150-8097. DOI: 10.14778/2212351.2212354. URL: <http://dx.doi.org/10.14778/2212351.2212354>.
- [Mal+10] Grzegorz Malewicz et al. "Pregel: A System for Large-scale Graph Processing". In: *ACM SIGMOD Record*. SIGMOD '10 (2010), pp. 135–146. DOI: 10.1145/1807167.1807184. URL: <http://doi.acm.org/10.1145/1807167.1807184>.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

INV51055. Fundamentos del Metodo Científico (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 2
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 64 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Write justification for this course here ...

5. OBJETIVOS

- Write your first goal here.
- Write your second goal here.
- Just in case you need more goals write them here

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: title for the unit goes here (0 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Topic1• Topic2• Topic3	<ul style="list-style-type: none">• Learning outcome1 [Levelforthislearningoutcome].• i.e Apply computing in complex problems [Usar].• Create a search engine [Evaluar].• Study data structures [Familiarizarse].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Bibitem1], [Bibitem2]	

Unidad 2: another unit goes here (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Topic1 	<ul style="list-style-type: none"> • Learning outcome xyz [Levelforthislearningoutcome].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Bibitem3], [Bibitem1]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62058. Proyecto de Final de Carrera III (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	3
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	-
2.4 Horas autónomas	:	96 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW62064. Proyecto de Final de Carrera II. (9 ^{no} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso tiene por objetivo que el alumno logre finalizar adecuadamente su borrador de tesis.

5. OBJETIVOS

- Que el alumno complete este curso con su tesis elaborada en calidad suficiente como para una inmediata sustentación.
- Que el alumno presente formalmente el borrador de tesis ante las autoridades de la facultad.
- Los entregables de este curso son:

Parcial: Avance del proyecto de tesis incluyendo en el documento: introducción, marco teórico, estado del arte, propuesta, análisis y/o experimentos y bibliografía sólida.

Final: Documento de tesis completo y listo para sustentar en un plazo no mayor de quince días.

6. COMPETENCIAS

- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Evaluar**)
- 5) Valora la necesidad del desarrollo profesional permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos. (**Evaluar**)

7. TEMAS

Unidad 1: Escritura del Borrador del trabajo de final de carrera (tesis) (60 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Redacción y corrección del trabajo de final de carrera 	<ul style="list-style-type: none"> Parte experimental concluida (si fuese adecuado al proyecto) [Evaluar] Verificar que el documento cumpla con el formato de tesis de la carrera [Evaluar] Entrega del borrador de tesis finalizado y considerado listo para una sustentación pública del mismo (requisito de aprobación) [Evaluar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [IEE08], [Ass08], [Cit08]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Ass08] Association for Computing Machinery. *Digital Library*. <http://portal.acm.org/dl.cfm>. Association for Computing Machinery, 2008.
- [Cit08] CiteSeer.IST. *Scientific Literature Digital Library*. <http://citeseer.ist.psu.edu>. College of Information Sciences and Technology, Penn State University, 2008.
- [IEE08] IEEE-Computer Society. *Digital Library*. <http://www.computer.org/publications/dlib>. IEEE-Computer Society, 2008.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62059. Computación Evolutiva (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 4
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: 2 (Semanal)
2.4 Horas autónomas	: 128 (horas)
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Obligatorio
2.7 Modalidad	: Presencial
2.8 Prerrequisitos	: SFW52079. Aprendizaje Automático. (7 ^{mo} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La Computación Evolutiva comprende un conjunto de metodologías de búsqueda y optimización cuya base primordial es el Paradigma Neodarwiniano que agrupa la Herencia Genética (Mendel), el Seleccionismo (Weismann) y la Evolución de las Especies (Darwin) que, cuando llevadas a implementaciones computacionales, ofrecen una herramienta poderosa de optimización global para una determinada función objetivo. Son bastante robustos cuando se supone la existencia de muchos óptimos locales. De esta forma, estos algoritmos pueden aplicarse en diversos problemas de optimización.

5. OBJETIVOS

- Que el alumno sea capaz de entender y aplicar el Paradigma Neodarwiniano para solucionar problemas complejos de optimización.
- Entendimiento a detalle del principio, fundamentos teóricos, funcionamiento, implementación, interpretación de resultados y operación de los algoritmos de la Computación Evolutiva más populares y utilizados por la comunidad científica y profesional.
- Conocimiento del estado del arte en Computación Evolutiva
- Capacidad de tratar un problema real de optimización utilizando Computación Evolutiva

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Introducción a la Optimización (4 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Definiciones de Optimización: principio de estabilidad, optimización global. Optimización Clásica: Definición del problema de optimización, concepto de convexidad, optimización numérica y combinatoria. Técnicas de optimización clásica: optimización lineal, algoritmo simplex, optimización no lineal, algoritmos <i>steepest descent</i>, <i>conjugate gradient</i>, algoritmos de búsqueda, programación dinámica, Heurísticas: definición, <i>Tabu search</i>, <i>Hill Climbing</i>, <i>Simulated Annealing</i>, <i>Evolutionary Algorithms</i> 	<ul style="list-style-type: none"> Entender los principios básicos de la optimización Entender e implementar algoritmos básicos de Optimización aplicados a problemas <i>benchmark</i>. Entender la necesidad de uso de heurísticas
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [Wei09], [RBK12]	

Unidad 2: Computación Evolutiva: Conceptos básicos (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Computación Evolutiva: definiciones Ideas precursoras: El origen de las ideas, L'Eclerc, Lamarck, Darwin, Weismann, Mendel, Baldwin, Paradigma Neodarwiniano Conceptos básicos de Computación Evolutiva: genes, cromosomas, individuos, población. Paradigmas de la Computación Evolutiva: Programación Evolutiva, Estrategias Evolutivas, Algoritmos Genéticos, <i>Learning Classifier Systems</i>, Programación Genética. 	<ul style="list-style-type: none"> Entender los principios básicos que rigen la computación evolutiva Conocer el contexto en que surgió la computación evolutiva.
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [RBK12], [Wei09], [Fog95], [koza98], [Mit04], [Mic96]	

Unidad 3: Algoritmo Genético Canónico (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmo Genético: definición, componentes. • Algoritmo Genético Canónico: procedimiento elemental, ciclo de un AG, representación (codificación binaria, real a binario, decodificación binario a real), inicialización de la población, evaluación y aptitud, selección (proporcional, torneo), operadores genéticos (cruces, mutaciones), el dilema <i>exploiting-exploring</i>, ajustes en la aptitud, ajustes en la selección. • Monitoreo de un AG, curva <i>best-so-far</i>, <i>online</i>, <i>offline</i>, • Convergencia • Teoría de <i>Schemata</i>: Máscaras, esquemas, definiciones y propiedades, <i>Schemata theorem</i>: impacto de la selección, cruce de 1 punto y mutación, teorema fundamental de los algoritmos genéticos, hipótesis de los bloques constructores. 	<ul style="list-style-type: none"> • Entender los algoritmos genéticos tradicionales. • Analizar y evaluar ventajas y desventajas del modelo genético tradicional. • Implementar un ejemplo de algoritmo genético tradicional y analizar su comportamiento.
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [RBK12], [Hol75], [Gol89], [Mit04], [Mic96]	

Unidad 4: Algoritmos Evolutivos en Optimización Numérica (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Problemas con restricciones: definiciones, espacios válido e inválido. • Tratamiento de las restricciones: Penalización, reparación, uso de codificadores, operadores especializados. • Uso de codificación real: binario vs real algoritmo evolutivo con codificación real. • Modelo GENOCOP, tratamiento de restricciones lineales, inicialización, operadores, inicialización, modelo GENOCOP III para restricciones no lineales: reparación de individuos 	<ul style="list-style-type: none"> • Comprensión de las formas de tratar problemas de optimización con restricciones. • Entender y analizar los algoritmos evolutivos con codificación real. • Evaluar la aplicación de computación evolutiva en problemas de optimización numérica
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [RBK12], [Mic96], [Mic00], [SC00]	

Unidad 5: Algoritmos Evolutivos en Optimización Combinatoria (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Espacios discretos y finitos • Algoritmos Evolutivos discretos: definición, modelo discreto generalizado • Algoritmos Evolutivos de orden: representación de soluciones, operadores de orden: cruces, mutaciones • Aplicaciones: <i>Quadratic assignment Problem</i> (QAP), <i>Travelling Salesman Problem</i> (TSP) • Problemas de Planificación: variables típicas, características, representación, codificadores, evaluación de una planificación. 	<ul style="list-style-type: none"> • Comprender e identificar el uso de Computación Evolutiva en problemas de optimización combinatoria • Evaluar la aplicación de computación evolutiva en problemas reales discretos
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [RBK12], [Mit04], [Cru03]	

Unidad 6: Paralelización y Multi objetivos (8 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Algoritmos Evolutivos en Paralelo (PEA): arquitecturas de paralelización, arquitecturas <i>master-slave</i>, <i>coarse-grained</i>, <i>fine-grained</i> e híbridas • Análisis de la ejecución de una implementación <i>master-slave</i>. • Optimización de Múltiples Objetivos: Definición formal, criterio de Pareto, Algoritmos Evolutivos Multi Objetivos (MOEA) sin uso de Pareto, MOEA con uso de Pareto: MOGA, NSGA, NPGA, NPGA2, PESA, SPEA, SPEA-II, Algoritmo Microgenético. • Métricas de desempeño (MOEA), investigación futura 	<ul style="list-style-type: none"> • Comprender y analizar la capacidad de paralelización de los modelos evolutivos • Analizar la aplicabilidad de Computación Evolutiva en problemas de múltiples objetivos • Implementación de modelos paralelos y multiobjetivo
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [RBK12], [Can00], [Coe07]	

Unidad 7: Algoritmos Genéticos Avanzados (16 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> Algoritmos Evolutivos Híbridos (HEA): Por qué hibridizar?, formas de hibridización, búsqueda local y aprendizaje. Programación Genética (GP): definición, representación, ciclo de la GP, Algoritmos Culturales (CA): Evolución Cultural, componentes, procedimiento, espacio de creencia, operadores culturales. Coevolución: características (CoEv), modelo competitivo, modelo cooperativo. Evolución Diferencial (DE): inicialización, operaciones, selección, DE vs. GA, variantes de DE, <i>Dy-namic DE</i> Algoritmos Evolutivos con Inspiración Cuántica (QIEA): Computación cuántica, algoritmos con inspiración cuántica, QIEA-B, QIEA-R 	<ul style="list-style-type: none"> Reconocer y analizar la necesidad de usar Algoritmos Evolutivos más avanzados Implementación de modelos avanzados de computación evolutiva
Aprendizaje autónomo	
<ul style="list-style-type: none"> Desarrollo de ejercicios prácticos 	
Lecturas : [RBK12], [ELM+06], [Koz92], [Reynolds94], [SP95], [Cru07]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Can00] Erick Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 2000. ISBN: 0792372212.
- [Coe07] Carlos A. Coello Coello. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. 2nd Edition. Springer, Sept. 2007.
- [Cru03] André Vargas Abs da Cruz. “Otimização de planejamento com restrições de precedência usando algoritmos genéticos e co-evolução cooperativa”. MA thesis. Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Feb. 2003. URL: http://www2.dbd.puc-rio.br/pergamum/biblioteca/php/mostrateses.php?open=1&arqtese=5000066121_03_Indice.html.
- [Cru07] André Abs da Cruz. “Algoritmos Evolutivos com Inspiração Quântica para Problemas com Representação Numérica”. (In Portuguese). PhD thesis. Rio de Janeiro, Brasil: Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Mar. 2007.
- [ELM+06] Tarek A. El-Mihoub et al. “Hybrid Genetic Algorithms: A Review”. In: *Engineering Letters* 13.2 (Aug. 2006). ISSN: 1816-0948. URL: www.engineeringletters.com/issues_v13/issue.../EL_13_2_11.pdf.
- [Fog95] David B. Fogel. *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*. New York: The Institute of Electrical and Electronic Engineers, 1995.
- [Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Massachusetts: Addison-Wesley Publishing Co., 1989.
- [Hol75] John Henry Holland. *Adaptation in Natural and Artificial Systems*. first. Ann Arbor, Michigan: University of Michigan Press, 1975.
- [Koz92] John R. Koza. *Genetic Programming. On the Programming of Computers by Means of Natural Selection*. Cambridge, Massachusetts: The MIT Press, 1992.
- [Mic00] Zbigniew Michalewicz. “Introduction to constraint-handling techniques, Decoders, Repair algorithms, Constraint-preserving operators”. In: *Evolutionary Computation 2, Advanced Algorithms and Operators* (2000), pp. 38–40, 49–55, 56–61, 62–68.
- [Mic96] Zibgniew Michalewicz. *Genetic Algorithms+Data Structures = Evolution Programs*. Springer-Verlag, 1996.
- [Mit04] Melanie Mitchell. *An Introduction to Genetic Algorithms: Complex Adaptive Systems*. The MIT Press, 2004.
- [RBK12] Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok, eds. *Handbook of Natural Computing*. 1st. Springer Publishing Company, Incorporated, 2012. ISBN: 3540929096, 9783540929093.
- [SC00] Alice E. Smith and David W. Coit. “Penalty functions”. In: *Evolutionary Computation 2, Advanced Algorithms and Operators* (2000), pp. 41–48.
- [SP95] Rainer Storn and Kenneth Price. *Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*. Tech. rep. TR-95-012. Berkeley, California: International Computer Science Institute, Mar. 1995.
- [Wei09] Thomas Weise. *Global Optimization Algorithms - Theory and Application*. <http://www.it-weise.de>. 2009.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62060. Formación de Empresas de Base Tecnológica III (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	: 3
2.2 Horas de teoría	: 2 (Semanal)
2.3 Horas de práctica	: -
2.4 Horas autónomas	: 96 (horas)
2.5 Duración del periodo	: 16 semanas
2.6 Condición	: Obligatorio
2.7 Modalidad	: Presencial
2.8 Prerrequisitos	: SFW62068. Formación de Empresas de Base Tecnológica II. (9 ^{no} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso dentro del área formación de empresas de base tecnológica, pretende abordar todos los procesos y buenas prácticas en la gestión de proyectos recomendadas por el *Project Management Institute* (PMI) contenidas en el *Project Management Body of Knowledge 2012* (PMBOK) aplicado en particular a proyectos de base tecnológica como pueden ser la construcción, desarrollo, integración e implementación de soluciones de software de aplicación.

El futuro profesional que pretenda incursionar con una empresa de software en el competitivo mercado globalizado, debe necesariamente conocer las habilidades duras y practicar las habilidades blandas que se consideran en el PMBOK. Todos los contratos de suministro de bienes tangibles (Hardware) o intangibles (Software) así como los servicios de consultoría deben ser manejados como pequeños proyectos.

Creemos de suma importancia impartir los fundamentos y experiencias asociadas a la dirección de proyectos a los futuros profesionales, debemos considerar que en la actualidad las empresas cliente (nacionales o internacionales) que demandan soluciones exigen a las empresas de consultoría se lleve a cabo los proyectos de sistemas de información y tecnología de información con los estándares del PMI, cada vez mas resulta ser una condición de exigibilidad para poder ganar licitaciones y firmar contratos de suministro de soluciones de tecnología, asimismo se exige que el jefe del proyecto, adicionalmente a su formación y experiencia para llevar a buen puerto el proyecto sea un PMP.

5. OBJETIVOS

- Que el alumno domine los conceptos relacionados a la gestión de proyectos informáticos.
- Proporcionar al alumno las técnicas y herramientas que le permitan gestionar con éxito proyectos de diversas magnitudes.
- Que el alumno construya su plan de negocios orientado a conseguir un inversionista internacional que pueda impulsar y proyectar a la empresa a un ámbito internacional.

6. COMPETENCIAS

- 1) Evalúa las necesidades del cliente y del entorno e identificar los requisitos de software para generar soluciones integrales e innovadoras optimizando los recursos tecnológicos, de capital humano, costo y tiempo. (**Evaluar**)
- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Evaluar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Evaluar**)

7. TEMAS

Unidad 1: Marco Conceptual de la Dirección de Proyectos (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Introducción • Finalidad de la guía del PMBOK, ¿Qué es un proyecto?, ¿Qué es la dirección de proyectos?, La estructura de la guía del PMBOK, Áreas de experiencia, contexto de la dirección de proyectos • Ciclo de Vida del Proyecto y Organización • Ciclo de vida del proyecto, interesados en el proyecto, influencias de la organización 	<ul style="list-style-type: none"> • Conocer el marco conceptual en el que se desarrollan los proyectos. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pro12], [Rit09]	

Unidad 2: Norma para la dirección de un proyecto (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Procesos de Dirección de Proyectos para un Proyecto • Procesos de dirección de proyectos, grupos de procesos de dirección de proyectos, interacciones entre procesos, correspondencia de los procesos de dirección de proyectos 	<ul style="list-style-type: none"> • Conocer los estándares de gestión de proyectos aplicado a proyectos. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pro12], [Rit09]	

Unidad 3: Áreas de conocimiento de la dirección de proyectos (60 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Introducción • Gestión de la Integración del Proyecto • Gestión del Alcance del Proyecto • Gestión del Tiempo del Proyecto • Gestión de los Costes del Proyecto • Gestión de la Calidad del Proyecto • Gestión de los Recursos Humanos del Proyecto • Gestión de las Comunicaciones del Proyecto • Gestión de los Riesgos del Proyecto • Gestión de las Adquisiciones del Proyecto 	<ul style="list-style-type: none"> • Entender la naturaleza de la gerencia de proyectos y su importancia para lograr el éxito en los proyectos. [Evaluar] • Adquirir el conocimiento necesario para gestionar proyectos de manera exitosa en terminos de: Tiempo, Costos, Alcance, Riesgos, Calidad, RRHH, Procura, Comunicaciones e Integración. [Usar] • Valorar la importancia de una buena Gerencia de Proyectos. [Evaluar] • Demostrar competencias para la realización de presentaciones efectivas. [Usar] • Desarrollar habilidades para gestionar equipos de trabajo multidisciplinarios. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pro12], [Rit09]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[Pro12] PMI Project Management Institute. *PMBOK Guide, 5th Edition*. Project Management Institute, 2012.

[Rit09] PMP Rita Mulcahy. *PMP Exam Prep - 6th Edition*. RMC Publications, 2009.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de

Ingeniería de Software

Sílabo 2021-II

1. CURSO

SFW62061. Cloud Computing (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	2
2.2 Horas de teoría	:	-
2.3 Horas de práctica	:	4 (Semanal)
2.4 Horas autónomas	:	64 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	IIS64000. Big Data y Analítica de Datos. (9 ^{no} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Para entender las técnicas computacionales avanzadas, los estudiantes deberán tener un fuerte conocimiento de las diversas estructuras discretas, estructuras que serán implementadas y usadas en laboratorio en el lenguaje de programación.

5. OBJETIVOS

- Que el alumno sea capaz de modelar problemas de ciencia de la computación usando grafos y árboles relacionados con estructuras de datos.
- Que el alumno aplicar eficientemente estrategias de recorrido para poder buscar datos de una manera óptima.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Evaluar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Evaluar**)

7. TEMAS

Unidad 1: Sistemas distribuidos (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Fallos: <ul style="list-style-type: none"> – Fallos basados en red (incluyendo particiones) y fallos basados en nodos – Impacto en garantías a nivel de sistema (p.e., disponibilidad) • Envío de mensajes distribuido: <ul style="list-style-type: none"> – Conversión y transmisión de datos – Sockets – Secuenciamiento de mensajes – Almacenando <i>Buffering</i>, reenviando y desechando mensajes • Compensaciones de diseño para Sistemas Distribuidos: <ul style="list-style-type: none"> – Latencia versus rendimiento – Consistencia, disponibilidad, tolerancia de particiones • Diseño de Servicio Distribuido: <ul style="list-style-type: none"> – Protocolos y servicios Stateful versus stateless – Diseños de Sesión (basados en la conexión) – Diseños reactivos (provocados por E/S) y diseños de múltiples hilos • Algoritmos de Distribución de Núcleos: <ul style="list-style-type: none"> – Elección, descubrimiento 	<ul style="list-style-type: none"> • Distinguir las fallas de red de otros tipos de fallas [Familiarizarse] • Explicar por qué estructuras de sincronización como cerraduras simples (<i>locks</i>) no son útiles en la presencia de fallas distribuidas [Familiarizarse] • Escribir un programa que realiza cualquier proceso de <i>marshalling</i> requerido y la conversión en unidades de mensajes, tales como paquetes, para comunicar datos importantes entre dos <i>hosts</i> [Usar] • Medir el rendimiento observado y la latencia de la respuesta a través de los <i>hosts</i> en una red dada [Usar] • Explicar por qué un sistema distribuido no puede ser simultáneamente Consistente (<i>Consistent</i>), Disponible (<i>Available</i>) y Tolerante a fallas (<i>Partition tolerant</i>). [Familiarizarse] • Implementar un servidor sencillo - por ejemplo, un servicio de corrección ortográfica [Usar] • Explicar las ventajas y desventajas entre: <i>overhead</i>, escalabilidad y tolerancia a fallas entre escoger un diseño sin estado (<i>stateless</i>) y un diseño con estado (<i>stateful</i>) para un determinado servicio [Familiarizarse] • Describir los desafíos en la escalabilidad, asociados con un servicio creciente para soportar muchos clientes, así como los asociados con un servicio que tendrá transitoriamente muchos clientes [Familiarizarse] • Dar ejemplos de problemas donde algoritmos de consenso son requeridos, por ejemplo, la elección de líder [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Cou+11]	

Unidad 2: Cloud Computing (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión global de <i>Cloud Computing</i>. • Historia. • Visión global de las tecnologías que envuelve. • Beneficios, riesgos y aspectos económicos. • Servicios en la nube. <ul style="list-style-type: none"> – Infraestructura como servicio <ul style="list-style-type: none"> * Elasticidad de recursos * APIs de la Plataforma – Software como servicio – Seguridad – Administración del Costo • Computación a Escala de Internet: <ul style="list-style-type: none"> – Particionamiento de Tareas – Acceso a datos – Clusters, grids y mallas 	<ul style="list-style-type: none"> • Explicar el concepto de <i>Cloud Computing</i>. [Familiarizarse] • Listar algunas tecnologías relacionadas con <i>Cloud Computing</i>. [Familiarizarse] • Explicar las estrategias para sincronizar una vista común de datos compartidos a través de una colección de dispositivos [Familiarizarse] • Discutir las ventajas y desventajas del paradigma de <i>Cloud Computing</i>. [Familiarizarse] • Expresar los beneficios económicos así como las características y riesgos del paradigma de Cloud para negocios y proveedores de cloud. [Familiarizarse] • Diferenciar entre los modelos de servicio. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HDF11], [BVS13]	

Unidad 3: Centros de Procesamiento de Datos (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión global de un centro de procesamiento de datos. • Consideraciones en el diseño. • Comparación de actuales grandes centros de procesamiento de datos. 	<ul style="list-style-type: none"> • Describir la evolución de los <i>Data Centers</i>. [Familiarizarse] • Esbozar la arquitectura de un data center en detalle. [Familiarizarse] • Indicar consideraciones de diseño y discutir su impacto. [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HDF11], [BVS13]	

Unidad 4: Cloud Computing (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Virtualización. <ul style="list-style-type: none"> – Gestión de recursos compartidos – Migración de procesos • Seguridad, recursos y aislamiento de fallas. • Almacenamiento como servicio. • Elasticidad. • Xen y VMware. • Amazon EC2. 	<ul style="list-style-type: none"> • Virtualización. <ul style="list-style-type: none"> – Gestión de recursos compartidos – Migración de procesos . [Familiarizarse] • Explicar las ventajas y desventajas de usar una infraestructura virtualizada. [Familiarizarse] • Identificar las razones por qué la virtualización está llegando a ser enormemente útil, especialmente en la cloud. [Familiarizarse] • Explicar diferentes tipos de aislamiento como falla, recursos y seguridad proporcionados por la virtualización y utilizado por la nube. [Familiarizarse] • Explicar la complejidad que puede tener el administrar en términos de niveles de abstracción y interfaces bien definidas y su aplicabilidad para la virtualización en la nube. [Familiarizarse] • Definir virtualización y identificar diferentes tipos de máquinas virtuales. [Familiarizarse] • Identificar condiciones de virtualización de CPU, reconocer la diferencia entre <i>full virtualization</i> y <i>paravirtualization</i>, explicar emulación como mayor técnica para virtualización del CPU y examinar planificación virtual del CPU en Xen. [Familiarizarse] • Esbozar la diferencia entre la clásica memoria virtual del SO y la virtualización de memoria. Explicar los múltiples niveles de mapeamiento de páginas en oposición a la virtualización de la memoria. Definir memoria <i>over-commitment</i> e ilustrar sobre VMware <i>memory ballooning</i> como técnica de reclamo para sistemas virtualizados con memoria <i>over-committed</i>. [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HDF11], [BVS13]	

Unidad 5: Cloud Computing (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Almacenamiento de datos en la nube: <ul style="list-style-type: none"> – Acceso compartido a data stores de consistencia débil – Sincronización de datos – Particionamiento de datos – Sistemas de Archivos Distribuidos – Replicación • Visión global sobre tecnologías de almacenamiento. • Conceptos fundamentales sobre almacenamiento en la cloud. • Amazon S3 y EBS. • Sistema de archivos distribuidos. • Sistema de bases de datos NoSQL. 	<ul style="list-style-type: none"> • Describir la organización general de datos y almacenamiento. [Familiarizarse] • Identificar los problemas de escalabilidad y administración de la big data. Discutir varias abstracciones en almacenamiento. [Familiarizarse] • Comparar y contrastar diferentes tipos de sistema de archivos. • Comparar y contrastar el Sistema de Archivos Distribuido de Hadoop (HDFS) y el Sistema de Archivos Paralelo Virtual (PVFS). [Usar] • Comparar y contrastar diferentes tipos de bases de datos. Discutir las ventajas y desventajas sobre las bases de datos NoSQL. [Usar] • Discutir los conceptos de almacenamiento en la cloud. [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HDF11], [BVS13]	

Unidad 6: Modelos de Programación (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión global de los modelos de programación basados en <i>cloud computing</i>. • Modelo de Programación MapReduce. • Modelo de programación para aplicaciones basadas en Grafos. 	<ul style="list-style-type: none"> • Explicar los aspectos fundamentales de los modelos de programación paralela y distribuida. [Familiarizarse] • Diferencias entre los modelos de programación: MapReduce, Pregel, GraphLab y Giraph. [Usar] • Explicar los principales conceptos en el modelo de programación MapReduce. [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [HDF11], [BVS13], [Low+12], [Mal+10], [Bal+08]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

- [Bal+08] Shumeet Baluja et al. "Video Suggestion and Discovery for Youtube: Taking Random Walks Through the View Graph". In: *Proceedings of the 17th International Conference on World Wide Web. WWW '08*. Beijing, China: ACM, 2008, pp. 895–904. ISBN: 978-1-60558-085-2. DOI: 10.1145/1367497.1367618. URL: <http://doi.acm.org/10.1145/1367497.1367618>.
- [BVS13] Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi. *Mastering Cloud Computing: Foundations and Applications Programming*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN: 9780124095397, 9780124114548.
- [Cou+11] George Coulouris et al. *Distributed Systems: Concepts and Design*. 5th. USA: Addison-Wesley Publishing Company, 2011. ISBN: 0132143011, 9780132143011.
- [HDF11] Kai Hwang, Jack Dongarra, and Geoffrey C. Fox. *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123858801, 9780123858801.
- [Low+12] Yucheng Low et al. "Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud". In: *Proc. VLDB Endow*. 5.8 (Apr. 2012), pp. 716–727. ISSN: 2150-8097. DOI: 10.14778/2212351.2212354. URL: <http://dx.doi.org/10.14778/2212351.2212354>.
- [Mal+10] Grzegorz Malewicz et al. "Pregel: A System for Large-scale Graph Processing". In: *Proc. ACM SIGMOD. SIGMOD '10 (2010)*, pp. 135–146. DOI: 10.1145/1807167.1807184. URL: <http://doi.acm.org/10.1145/1807167.1807184>.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

SFW62062. Internet de las Cosas (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	2
2.2 Horas de teoría	:	-
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	64 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW62071. Computación Paralela y Distribuida. (8 ^{vo} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La última década ha traído un crecimiento explosivo en computación con multiprocesadores, incluyendo los procesadores de varios núcleos y centros de datos distribuidos. Como resultado, la computación paralela y distribuida se ha convertido de ser un tema ampliamente electivo para ser uno de los principales componentes en la malla estudios en ciencia de la computación de pregrado. Tanto la computación paralela como la distribuida implica la ejecución simultánea de múltiples procesos en diferentes dispositivos que cambian de posición.

5. OBJETIVOS

- Que el alumno sea capaz de crear aplicaciones paralelas de mediana complejidad aprovechando eficientemente distintos dispositivos móviles.

6. COMPETENCIAS

- 3) Justifica un desempeño individual, como parte de equipos de trabajo o como líder de proyectos de grupos multidisciplinarios en entornos globales con el fin de asegurar la calidad de software, aplicando normas, configuraciones, regulaciones y métricas. (**Evaluar**)
- 4) Diseña soluciones de Software de acuerdo a los estándares y políticas de seguridad de la información en uno o varios dominios de aplicación siendo socialmente responsables y demostrando ética profesional. (**Evaluar**)

7. TEMAS

Unidad 1: Fundamentos de paralelismo (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Procesamiento Simultáneo Múltiple. • Metas del Paralelismo (ej. rendimiento) frente a Concurrencia (ej. control de acceso a recursos compartidos) • Paralelismo, comunicación, y coordinación: <ul style="list-style-type: none"> – Paralelismo, comunicación, y coordinación – Necesidad de Sincronización • Errores de Programación ausentes en programación secuencial: <ul style="list-style-type: none"> – Tipos de Datos (lectura/escritura simultánea o escritura/escritura compartida) – Tipos de Nivel más alto (interleavings violating program intention, no determinismo no deseado) – Falta de vida/progreso (deadlock, starvation) 	<ul style="list-style-type: none"> • Distinguir el uso de recursos computacionales para una respuesta mas rápida para administrar el acceso eficiente a un recurso compartido [Familiarizarse] • Distinguir múltiples estructuras de programación suficientes para la sincronización que pueden ser interimplementables pero tienen ventajas complementarias [Familiarizarse] • Distinguir datos de carrera (<i>data races</i>) a partir de carreras de mas alto nivel [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pac11], [Mat14], [Qui03]	

Unidad 2: Arquitecturas paralelas (12 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Procesadores mutlinúcleo. • Memoria compartida vs memoria distribuida. • Multiprocesamiento simétrico. • SIMD, procesamiento de vectores. • GPU, coprocesamiento. • Taxonomía de Flynn. • Soporte a nivel de instrucciones para programación paralela. <ul style="list-style-type: none"> – Instrucciones atómicas como Compare/Set (Comparar / Establecer) • Problemas de Memoria: <ul style="list-style-type: none"> – Caches multiprocesador y coherencia de cache – Acceso a Memoria no uniforme (NUMA) • Topologías. <ul style="list-style-type: none"> – Interconecciones – Clusters – Compartir recursos (p.e., buses e interconexiones) 	<ul style="list-style-type: none"> • Explicar las diferencias entre memoria distribuida y memoria compartida [Evaluar] • Describir la arquitectura SMP y observar sus principales características [Evaluar] • Distinguir los tipos de tareas que son adecuadas para máquinas SIMD [Usar] • Describir las ventajas y limitaciones de GPUs vs CPUs [Usar] • Explicar las características de cada clasificación en la taxonomía de Flynn [Usar] • Describir los desafíos para mantener la coherencia de la caché [Familiarizarse] • Describir los desafíos clave del desempeño en diferentes memorias y topologías de sistemas distribuidos [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pac11], [KH13], [SK10]	

Unidad 3: Descomposición en paralelo (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Necesidad de Comunicación y coordinación/sincronización. • Independencia y Particionamiento. • Conocimiento Básico del Concepto de Descomposición Paralela. • Descomposición basada en tareas: <ul style="list-style-type: none"> – Implementación de estrategias como hebras • Descomposición de Información Paralela <ul style="list-style-type: none"> – Estrategias como SIMD y MapReduce • Actores y Procesos Reactivos (solicitud de gestores) 	<ul style="list-style-type: none"> • Explicar por qué la sincronización es necesaria en un programa paralelo específico [Usar] • Identificar oportunidades para particionar un programa serial en módulos paralelos independientes [Familiarizarse] • Escribir un algoritmo paralelo correcto y escalable [Usar] • Paralelizar un algoritmo mediante la aplicación de descomposición basada en tareas [Usar] • Paralelizar un algoritmo mediante la aplicación de descomposición de datos en paralelo [Usar] • Escribir un programa usando actores y/o procesos reactivos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pac11], [Mat14], [Qui03]	

Unidad 4: Comunicación y coordinación (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Memoria Compartida. • La consistencia, y su papel en los lenguaje de programación garantías para los programas de carrera libre. • Pasos de Mensaje: <ul style="list-style-type: none"> – Mensajes Punto a Punto versus multicast (o basados en eventos) – Estilos para enviar y recibir mensajes Blocking vs non-blocking – Buffering de mensajes • Atomicidad: <ul style="list-style-type: none"> – Especificar y probar atomicidad y requerimientos de seguridad – Granularidad de accesos atómicos y actualizaciones, y uso de estructuras como secciones críticas o transacciones para describirlas – Exclusión mutua usando bloques, semáforos, monitores o estructuras relacionadas <ul style="list-style-type: none"> * Potencial para fallas y bloqueos (<i>deadlock</i>) (causas, condiciones, prevención) – Composición <ul style="list-style-type: none"> * Componiendo acciones atómicas granulares más grandes usando sincronización * Transacciones, incluyendo enfoques optimistas y conservadores • Consensos: <ul style="list-style-type: none"> – (Cíclicos) barreras, contadores y estructuras relacionadas • Acciones condicionales: <ul style="list-style-type: none"> – Espera condicional (p.e., empleando variables de condición) 	<ul style="list-style-type: none"> • Usar exclusión mutua para evitar una condición de carrera [Usar] • Dar un ejemplo de una ordenación de accesos entre actividades concurrentes (por ejemplo, un programa con condición de carrera) que no son secuencialmente consistentes [Familiarizarse] • Dar un ejemplo de un escenario en el que el bloqueo de mensajes enviados pueden dar <i>deadlock</i> [Usar] • Explicar cuándo y por qué mensajes de multidifusión (<i>multicast</i>) o basado en eventos puede ser preferible a otras alternativas [Familiarizarse] • Escribir un programa que termine correctamente cuando todo el conjunto de procesos concurrentes hayan sido completados [Usar] • Dar un ejemplo de un escenario en el que un intento optimista de actualización puede nunca completarse [Familiarizarse] • Usar semaforos o variables de condición para bloquear hebras hasta una necesaria precondition de mantenga [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pac11], [Mat14], [Qui03]	

Unidad 5: Análisis y programación de algoritmos paralelos (18 horas)

Competencias esperadas:

Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Caminos críticos, el trabajo y la duración y la relación con la ley de Amdahl. • Aceleración y escalabilidad. • Naturalmente (vergonzosamente) algoritmos paralelos. • Patrones Algoritmicos paralelos (divide-y-conquista, map/reduce, amos-trabajadores, otros) <ul style="list-style-type: none"> – Algoritmos específicos (p.e., MergeSort paralelo) • Algoritmos de grafos paralelo (por ejemplo, la ruta más corta en paralelo, árbol de expansión paralela) • Cálculos de matriz paralelas. • Productor-consumidor y algoritmos paralelos segmentados. • Ejemplos de algoritmos paralelos no-escalables. 	<ul style="list-style-type: none"> • Definir: camino crítico, trabajo y <i>span</i> [Familiarizarse] • Calcular el trabajo y el <i>span</i> y determinar el camino crítico con respecto a un diagrama de ejecución paralela. [Usar] • Definir <i>speed-up</i> y explicar la noción de escalabilidad de un algoritmo en este sentido [Familiarizarse] • Identificar tareas independientes en un programa que debe ser paralelizado [Usar] • Representar características de una carga de trabajo que permita o evite que sea naturalmente paralelizable [Familiarizarse] • Implementar un algoritmo dividir y conquistar paralelo (y/o algoritmo de un grafo) y medir empíricamente su desempeño relativo a su analogo secuencial [Usar] • Descomponer un problema (por ejemplo, contar el número de ocurrencias de una palabra en un documento) via operaciones <i>map</i> y <i>reduce</i> [Usar] • Proporcionar un ejemplo de un problema que se corresponda con el paradigma productor-consumidor [Usar] • Dar ejemplos de problemas donde el uso de <i>pipelining</i> sería un medio eficaz para la paralelización [Usar] • Implementar un algoritmo de matriz paralela [Usar] • Identificar los problemas que surgen en los algoritmos del tipo productor-consumidor y los mecanismos que pueden utilizarse para superar dichos problemas [Usar]
<p>Aprendizaje autónomo</p>	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
<p>Lecturas : [Mat14], [Qui03]</p>	

Unidad 6: Desempeño en paralelo (18 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Equilibrio de carga. • La medición del desempeño. • Programación y contención. • Evaluación de la comunicación de arriba. • Gestión de datos: <ul style="list-style-type: none"> – Costos de comunicación no uniforme debidos a proximidad – Efectos de Cache (p.e., false sharing) – Manteniendo localidad espacial • Consumo de energía y gestión. 	<ul style="list-style-type: none"> • Detectar y corregir un desbalanceo de carga [Usar] • Calcular las implicaciones de la ley de Amdahl para un algoritmo paralelo particular [Usar] • Describir como la distribución/disposición de datos puede afectar a los costos de comunicación de un algoritmo [Familiarizarse] • Detectar y corregir una instancia de uso compartido falso (<i>false sharing</i>) [Usar] • Explicar el impacto de la planificación en el desempeño paralelo [Familiarizarse] • Explicar el impacto en el desempeño de la localidad de datos [Familiarizarse] • Explicar el impacto y los puntos de equilibrio relacionados al uso de energía en el desempeño paralelo [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Pac11], [Mat14], [KH13], [SK10]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA

[KH13] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. 2nd. Morgan Kaufmann, 2013. ISBN: 978-0-12-415992-1.

- [Mat14] Norm Matloff. *Programming on Parallel Machines*. University of California, Davis, 2014. URL: <http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf>.
- [Pac11] Peter S. Pacheco. *An Introduction to Parallel Programming*. 1st. Morgan Kaufmann, 2011. ISBN: 978-0-12-374260-5.
- [Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. 1st. McGraw-Hill Education Group, 2003. ISBN: 0071232656.
- [SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. Addison-Wesley Professional, 2010. ISBN: 0131387685, 9780131387683.



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

IIS64016. Robótica (Electivo)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Horas autónomas	:	128 (horas)
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad	:	Presencial
2.8 Prerrequisitos	:	SFW62072. Visión Computacional. (8 ^{vo} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Que el alumno conozca y comprenda los conceptos y principios fundamentales de control, planificación de caminos y definición de estrategias en robótica móvil así como conceptos de percepción robótica de forma que entienda el potencial de los sistemas robóticos actuales

5. OBJETIVOS

- Sintetizar el potencial y las limitaciones del estado del arte de los sistemas robóticos actuales.
- Implementar algoritmos de planeamiento de movimientos simples.
- Explicar las incertezas asociadas con sensores y la forma de tratarlas.
- Diseñar una arquitectura de control simple
- Describir varias estrategias de navegación
- Entender el rol y las aplicaciones de la percepción robótica
- Describir la importancia del reconocimiento de imágenes y objetos en sistemas inteligentes
- Delinear las principales técnicas de reconocimiento de objetos
- Describir las diferentes características de las tecnologías usadas en percepción

6. COMPETENCIAS

Nooutcomes

7. TEMAS

Unidad 1: Robótica (5 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Vision general: problemas y progreso <ul style="list-style-type: none"> – Estado del arte de los sistemas robóticos, incluyendo sus sensores y una visión general de su procesamiento – Arquitecturas de control robótico, ejem., deliverado vs. control reactivo y vehiculos Braitenberg – Modelando el mundo y modelos de mundo – Incertidumbre inherente en detección y control • Configuración de espacio y mapas de entorno. 	<ul style="list-style-type: none"> • Listar capacidades y limitaciones de sistemas del estado del arte en robótica de hoy , incluyendo sus sensores y el procesamiento del sensor crucial que informa a esos sistemas [Familiarizarse] • Integrar sensores, actuadores y software en un robot diseñado para emprender alguna tarea [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Siegwart04], [Trun05], [Stone00]	

Unidad 2: Robótica (15 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Interpretando datos del sensor con incertidumbre. • Localización y mapeo. 	<ul style="list-style-type: none"> • Programar un robot para llevar a cabo tareas simples usando arquitecturas de control deliverativo, reactivo y/o híbrido [Usar] • Implementar algoritmos de planificación de movimientos fundamentales dentro del espacio de configuración de un robot [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Siegwart04], [Trun05]	

Unidad 3: Robótica (20 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Navegación y control. • Planeando el movimiento. 	<ul style="list-style-type: none"> • Caracterizar las incertidumbres asociadas con sensores y actuadores de robot comunes; articular estrategias para mitigar esas incertidumbres. [Usar] • Listar las diferencias entre representaciones de los robot de su entorno externo, incluyendo sus fortalezas y defectos [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Siegwart04]	

Unidad 4: Visión y percepción por computador (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Visión Computacional <ul style="list-style-type: none"> – Adquisición de imágenes, representación, procesamiento y propiedades – Representación de formas, reconocimiento y segmentación de objetos – Análisis de movimiento • Modularidad en reconocimiento. 	<ul style="list-style-type: none"> • Resumir la importancia del reconocimiento de imágenes y objetos en Inteligencia Artificial (AI) e indicar varias aplicaciones significativas de esta tecnología [Usar] • Implementar reconocimiento de objetos en 2d basados en la representación del contorno y/o regiones basadas en formas [Usar]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Sonka07], [Gonzales07]	

Unidad 5: Robótica (10 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Coordinación multi-robots. 	<ul style="list-style-type: none"> • Comparar y contrastar al menos tres estrategias para la navegación de robots dentro de entornos conocidos y/o no conocidos, incluyendo sus fortalezas y defectos [Familiarizarse] • Describir al menos una aproximación para la coordinación de acciones y detección de varios robots para realizar una simple tarea [Familiarizarse]
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Stone00]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA



Universidad San Ignacio de Loyola (USIL)

Carrera Profesional de
Ingeniería de Software
Sílabo 2021-II

1. CURSO

ETM41007. Ética y Ciudadanía (Obligatorio)

2. INFORMACIÓN GENERAL

- 2.1 Créditos : 3
- 2.2 Horas de teoría : 2 (Semanal)
- 2.3 Horas de práctica : -
- 2.4 Horas autónomas : 96 (horas)
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad : Presencial
- 2.8 Prerrequisitos : Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Write justification for this course here ...

5. OBJETIVOS

- Write your first goal here.
- Write your second goal here.
- Just in case you need more goals write them here

6. COMPETENCIAS

CG5) Desarrollo humano. (Evaluar)

7. TEMAS

Unidad 1: title for the unit goes here (0 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none">• Topic1• Topic2• Topic3	<ul style="list-style-type: none">• Learning outcome1 [Levelforthislearningoutcome].• i.e Apply computing in complex problems [Usar].• Create a search engine [Evaluar].• Study data structures [Familiarizarse].
Aprendizaje autónomo	
<ul style="list-style-type: none">• Desarrollo de ejercicios prácticos	
Lecturas : [Bibitem1], [Bibitem2]	

Unidad 2: another unit goes here (1 horas)	
Competencias esperadas:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> • Topic1 	<ul style="list-style-type: none"> • Learning outcome xyz [Levelforthislearningoutcome].
Aprendizaje autónomo	
<ul style="list-style-type: none"> • Desarrollo de ejercicios prácticos 	
Lecturas : [Bibitem3], [Bibitem1]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Cada uno de los rubros del esquema de evaluación y la nota final del curso son redondeados a números enteros. La nota final del curso es el promedio ponderado de los rubros correspondientes: evaluación permanente, examen parcial y examen final.

Los promedios calculados componentes del rubro 'Evaluación Permanente' mantendrán su cálculo con 2 decimales.

	%	Observaciones	Semana	Rezagable
Evaluación Continua	70%			
Práctica Calificada	70%			
Práctica Calificada ₁		Se elimina la práctica con la menor nota	4	No
Práctica Calificada ₂		Se elimina la práctica con la menor nota	8	No
Práctica Calificada ₃		Se elimina la práctica con la menor nota	12	No
Proyecto	30%		15	
Examen final	30%			

10. BIBLIOGRAFÍA BÁSICA