



Compendio de Silabos

Escuela Profesional de Ciencia de la
Computación

– 2018-II–

Lima: 4 de enero de 2019

Equipo de trabajo

Comisión de Evaluación Curricular

Oscar Benito Pacheco

Docente de la E.A.P. de Computación Científica, UNMSM, Lima
email: *obenitop@unmsm.edu.pe*

Hugo Vega Huerta

Docente de la E.A.P. de Ingeniería de Sistemas, UNMSM, Lima
email: *hugovegahuerta@hotmail.com*

Silverio Bustos

Docente de la E.A.P. de Ingeniería de Sistemas, UNMSM, Lima
email: *silverio.bustos@gmail.com*

Luis Rivera Escriba

Docente de la Universidade Estadual do Norte Fluminense, Brasil
email: *rivera@uenf.br*

Ulises Román Concha

Docente de la E.A.P. de Ingeniería de Sistemas, UNMSM, Lima
email: *nromanc@yahoo.es*

Cristian Loli Prudencio

Docente de la E.A.P. de Ingeniería de Sistemas, UNMSM, Lima
email: *cristianhz@hotmail.com*

Además, han colaborado con este esfuerzo la Sociedad Peruana de Computación a quienes dejamos público nuestro agradecimiento.

Índice general

Primer Semestre	5
1.1. CS1100. Introducción a la Ciencia de la Computación	5
1.2. CS1D01. Estructuras Discretas I	12
1.3. GH0005. Laboratorio de Comunicación I	17
1.4. EG0003. Matemática I	19
1.5. EG0004. Desafíos Globales	24
Segundo Semestre	26
2.1. CS1102. Programación Orientada a Objetos I	26
2.2. CS1D02. Estructuras Discretas II	35
2.3. ME0019. Física I	39
2.4. GH0006. Laboratorio de Comunicación II	44
2.5. GH0007. Introducción al Desarrollo de Empresas	46
2.6. GH1002. Arte y Tecnología	49
2.7. GH1101. Inglés I	52
2.8. EG0005. Matemática II	56
Tercer Semestre	60
3.1. CS1103. Programación Orientada a Objetos II	60
3.2. CS2201. Arquitectura de Computadores	72
3.3. CS2B01. Desarrollo Basado en Plataformas	80
3.4. EN0021. Física II	85
3.5. GH0008. Gestión de Empresas	90
3.6. GH1102. Inglés II	92
3.7. EG0006. Matemática III	96
Cuarto Semestre	100
4.1. CS2100. Algoritmos y Estructuras de Datos	100
4.2. CS2101. Teoría de la Computación	103
4.3. CS2701. Bases de Datos I	107
4.4. IN0054. Estadística y Probabilidades	113
4.5. GH2101. Inglés III	116
Quinto Semestre	120
5.1. CS2102. Análisis y Diseño de Algoritmos	120
5.2. CS2702. Bases de Datos II	126
5.3. CS2901. Ingeniería de Software I	131
5.4. CS2S01. Sistemas Operativos	136

5.5.	CS3402. Compiladores	145
5.6.	GH0010. Ética y Tecnología	150
5.7.	ID104. Inglés IV	153
Sexto Semestre		157
6.1.	CS2301. Redes y Comunicaciones	157
6.2.	CS3101. Programación Competitiva	162
6.3.	CS3102. Estructuras de Datos Avanzadas	164
6.4.	CS3903. Sistemas de Infomación	168
Séptimo Semestre		171
7.1.	CS2H01. Interacción Humano Computador	171
7.2.	CS341. Lenguajes de Programación	179
7.3.	CS3P01. Computación Paralela y Distribuída	186
7.4.	CS2501. Computación Gráfica	193
7.5.	CS2601. Inteligencia Artificial	200
7.6.	CS2902. Ingeniería de Software II	210
Octavo Semestre		216
8.1.	CS4002. Proyecto de Final de Carrera I	216
8.2.	ET201. Formación de Empresas de Base Tecnológica I	219
Noveno Semestre		224
9.1.	CS3700. Big Data	224
9.2.	CS3I01. Seguridad en Computación	227
9.3.	CS4003. Proyecto de Final de Carrera II	238
9.4.	CS3501. Tópicos en Computación Gráfica	241
9.5.	CS3602. Robótica	243
9.6.	CS3901. Ingeniería de Software III	247
9.7.	BI0021. Bioinformática y Bioestadística	254
9.8.	ET301. Formación de Empresas de Base Tecnológica II	259
Décimo Semestre		263
10.1.	CS3P02. Cloud Computing	263
10.2.	CS3P03. Internet de las Cosas	269
10.3.	CS4004. Proyecto de Final de Carrera III	276



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS1100. Introducción a la Ciencia de la Computación (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Bro11] J. Glenn Brookshear. *Computer Science: An Overview*. Addison-Wesley, 2011.
- [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.

6. Información del curso

- (a) **Breve descripción del 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 introducción a los participantes en los conceptos fundamentales de este arte. Lo tópicos incluyen tipos de datos, estructuras de control, funciones, listas, recursividad y la mecánica de la ejecución, prueba y depuración.
- (b) **Prerrequisitos:** Ninguno
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

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

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome b**

10. Lista de temas a estudiar en el curso

1. Historia

2. Sistemas de tipos básicos
3. Conceptos Fundamentales de Programación
4. Análisis Básico
5. Algoritmos y Estructuras de Datos fundamentales
6. Algoritmos y Diseño
7. Métodos de Desarrollo

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Historia (5)	
Competencias esperadas: C4	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Bro11], [Gut13], [Zel10]	

Unidad 2: Sistemas de tipos básicos (2)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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. numeros, booleanos) – Composición de tipos construidos 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.
Lecturas : [Gut13], [Zel10]	

Unidad 3: Conceptos Fundamentales de Programación (9)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Sintaxis y semántica básica de un lenguaje de alto nivel. • Variables y tipos de datos primitivos (ej., números, 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.
Lecturas : [Gut13], [Zel10]	

Unidad 4: Análisis Básico (2)	
Competencias esperadas: C1,C5	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Gut13], [Zel10]	

Unidad 5: Algoritmos y Estructuras de Datos fundamentales (8)	
Competencias esperadas: C1,C2,C5	
Objetivos de Aprendizaje	Tópicos
<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] 	<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"> – Algoritmos de la ruta más corta (algoritmos de Dijkstra y Floyd) – Árbol de expansión mínima (algoritmos de Prim y Kruskal) • Búsqueda de patrones y algoritmos de cadenas/texto (ej. búsqueda de subcadena, búsqueda de expresiones regulares, algoritmos de subsecuencia común más larga)
Lecturas : [Gut13], [Zel10]	

Unidad 6: Algoritmos y Diseño (9)	
Competencias esperadas: C1,C2,C5	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Gut13], [Zel10]	

Unidad 7: Métodos de Desarrollo (1)	
Competencias esperadas: C2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Construir y depurar programas que utilizan las bibliotecas estándar disponibles con un lenguaje de programación elegido [Familiarizarse] 	<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.
Lecturas : [Gut13], [Zel10]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS1D01. Estructuras Discretas I (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Epp10] Susanna S. Epp. *Discrete Mathematics with Applications*. 4 ed. Brooks Cole, 2010.
- [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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** Ninguno
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Usar**)

9. Competencias (IEEE)

C1. La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**

C20. Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome j**

10. Lista de temas a estudiar en el curso

1. Funciones, relaciones y conjuntos
2. Lógica básica
3. Técnicas de demostración
4. Representación de Datos

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Funciones, relaciones y conjuntos (22)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Gri03], [Ros07]	

Unidad 2: Lógica básica (14)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [Ros07], [Gri03]	

Unidad 3: Técnicas de demostración (14)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Ros07], [Epp10], [Sch12]	

Unidad 4: Representación de Datos (10)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<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]. 	<ul style="list-style-type: none"> • Representaciones numéricas: signo magnitud, punto flotante. • Representaciones de otros objetos: conjuntos, relaciones, funciones
Lecturas : [Ros07], [Gri03]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** GH0005. Laboratorio de Comunicación I (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 2 HT; 2 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[D93] Cassany. D. *La cocina de la Escritura*. Barcelona, España, Anagrama, 1993.

6. Información del curso

- (a) **Breve descripción del curso** A través de este curso, el alumno mejorará y fortalecerá sus capacidades para comunicarse tanto a nivel oral como escrito en un contexto académico. Para ello, el alumno se ejercitará en la composición de textos, tomando en cuenta las exigencias propias de un lenguaje formal académico: características de la redacción académica (reglas de puntuación, ortografía, competencia léxico gramatical, normativa) y empleo correcto de la información. A su vez, el curso promueve una lectura comprensiva que no se limita al nivel descriptivo, sino que abarca también lo conceptual y metafórico, pues solo de ese modo el estudiante desarrollará su capacidad crítica y analítica. El estudiante afrontará lecturas académicas y de divulgación científica que le permitirán distinguir los objetivos planteados en los distintos tipos de textos, y reconocer al texto oral y escrito como una unidad coherente y cohesionada en cuanto a forma y contenido. Alcanzados estos objetivos, el estudiante comprenderá que las habilidades comunicativas orales y escritas son competencias centrales de su vida universitaria y, posteriormente, de su vida profesional.
- (b) **Prerrequisitos:** Ninguno
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Con este curso el estudiante desarrolla y fortalece sus habilidades comunicativas orales y escritas en el marco de un contexto académico. Además, comprende conceptual y metafóricamente textos expositivos, e identifica los objetivos, jerarquía de las ideas y estructura de dichos textos. Al finalizar el curso, el estudiante es capaz de producir textos expositivos descriptivos e informativos. Así mismo, desarrolla su capacidad de apertura y tolerancia hacia la diversidad de puntos de vista gracias al continuo trabajo grupal, autoevaluaciones y evaluaciones de pares que enfrentará a lo largo del ciclo en el curso.

8. Contribución a los resultados (*Outcomes*)

- e) Entender correctamente las implicancias profesionales, éticas, legales, de seguridad y sociales de la profesión. (**Usar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- n) Aplicar conocimientos de humanidades en su labor profesional. (**Usar**)

9. Competencias (IEEE)

C17. Capacidad para expresarse en los medios de comunicación orales y escritos como se espera de un graduado.⇒ **Outcome f,h,n**

C20. Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome f,n**

C24. Comprender la necesidad de la formación permanente y la mejora de habilidades y capacidades.⇒ **Outcome f,h**

10. Lista de temas a estudiar en el curso

1. Laboratorio de Comunicación I

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Laboratorio de Comunicación I (12)	
Competencias esperadas: 4	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Auto-evaluación: el estudiante es capaz de reconocer sus propias fortalezas y deficiencias al formular críticas constructivas sobre su propio trabajo.	<ul style="list-style-type: none">• Características de Escritura Académica.• Estrategias de Lectura.• Estructura del texto.• Estructura de párrafos.• Características del párrafo.• Texto argumentativo Vs. Texto expositivo.• Proceso de Redacción.• Citas:función y tipos -Bibliografía.• Aproximación a características de la exposición oral.• Conferencia :características exposición formal.• Redacción de texto completo con citas.
Lecturas : [D93]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** EG0003. Matemática I (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 4 HT; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. **Bibliografía**

6. **Información del curso**

- (a) **Breve descripción del curso** El curso tiene como objetivo desarrollar en los estudiantes las habilidades para manejar modelos en ciencia e ingeniería relacionados con habilidades de cálculo diferencial simple. En el curso se estudian y aplican conceptos relacionados con el cálculo de Límites, derivados e integrales de funciones reales y vectoriales de variables reales únicas que se utilizarán como base y apoyo al estudio de nuevos contenidos y materias. También busca lograr capacidades de razonamiento y aplicabilidad para interactuar con problemas del mundo real proporcionando una base matemática para actividades de desarrollo.
- (b) **Prerrequisitos:** Ninguno
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. **Objetivos específicos del curso.**

- Aplicar los conceptos de números complejos y funciones para resolver problemas relacionados con la ciencia.
- Aplicar conceptos matemáticos y técnicas de cálculo diferencial de una variable para resolver situaciones problemáticas de la ciencia
- Calcular las expresiones matemáticas de las integrales indefinidas con exactitud, orden y claridad en el tratamiento de los datos.

8. **Contribución a los resultados (*Outcomes*)**

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Evaluar**)

9. **Competencias (IEEE)**

- C1. La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C20. Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome j**
- C24. Comprender la necesidad de la formación permanente y la mejora de habilidades y capacidades.⇒ **Outcome j**

10. **Lista de temas a estudiar en el curso**

1. Números complejos
2. Funciones de una sola variable
3. Límites y derivadas
4. Integrales

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Números complejos (20)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Definir y operar con números complejos, calculando su forma polar y exponencial [Evaluar]. • Utilizar el teorema de Moivre para simplificar los cálculos de complejos[Evaluar]. 	<ul style="list-style-type: none"> • Operaciones con números complejos • Teorema de Moivre
Lecturas : [Stewart], [RonLarson]	

Unidad 2: Funciones de una sola variable (10)	
Competencias esperadas: C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Definir una función de una sola variable y entender y ser capaz de determinar su dominio y rango. [Evaluar]. • Reconocer diferentes tipos específicos de funciones y crear diagramas de dispersión y seleccionar un modelo apropiado. [Evaluar]. • Comprender cómo un cambio en la base afecta a la gráfica de exponenciales y funciones logarítmicas. [Evaluar]. • Reconoce y construye funciones trigonométricas. [Evaluar]. • Aplicar reglas para transformar funciones. [Evaluar]. • Ser capaz de resolver problemas de aplicaciones simples como regresión y ajuste de curvas. [Evaluar]. 	<ul style="list-style-type: none"> • Dominio y rango. • Tipos de funciones. • Gráfico de exponenciales y funciones logarítmicas. • Funciones trigonométricas. • Aplicar reglas para transformar funciones. • Problemas de aplicaciones usando Excel, modelando crecimiento bacteriano, escala logarítmica, etc.
Lecturas : [Stewart], [RonLarson]	

Unidad 3: Límites y derivadas (20)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Entender el concepto de límites y calcular los límites de la gráfica de una función. [Evaluar]. • Encontrar límites usando las leyes de límites y la simplificación algebraica. [Evaluar]. • Encontrar asíntotas verticales y horizontales. [Evaluar]. • Calcular y estimar derivados. [Evaluar]. • Interpretar la derivada como una tasa de cambio. [Evaluar]. • Encontrar los derivados de la función básica y compuesta [Evaluar]. • Aproximación de funciones usando conceptos de derivadas y calculo de errores relativos [Evaluar]. • Encontrar los números críticos , los valores máximos y mínimos absolutos y locales para la función continua. [Evaluar]. • Aplicar Teorema de L'Hospital para calcular algunos límites. [Evaluar]. • Entender los problemas de optimización, encontrar la función a ser optimizada y resolver[Evaluar]. • Ser capaz de resolver problemas de aplicaciones simples.. [Evaluar]. 	<ul style="list-style-type: none"> • Límites • Derivadas • Conceptos sobre Derivadas y calcular errores relativos. • El Teorema de L'Hospital • Problemas de aplicaciones tales como velocidad, crecimiento exponencial y decaimiento, acumulación de grava creciente, optimización de una lata, etc.
Lecturas : [Stewart], [RonLarson]	

Unidad 4: Integrales (22)	
Competencias esperadas: C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Resolver correctamente el área de estimación usando los rectángulos izquierdo y derecho del punto final y del punto medio.[Evaluar]. • Utilizar el teorema fundamental para encontrar derivados de funciones de evaluar integrales definidas e indefinidas mediante sustitución.[Evaluar]. • Utilizar diferentes técnicas para integrar funciones. [Evaluar]. • Aplicar integrales a las áreas encontradas.[Evaluar]. • Calcular volúmenes de sólidos obtenidos girando una región limitada alrededor del eje x o del eje y. [Evaluar]. • Calcular el volumen de sólidos obtenidos al girar una región limitada alrededor del eje x o del eje y, considerando cascarones cilíndricos.[Evaluar]. • Calcula el valor promedio de una función. [Evaluar]. • Calcular el trabajo realizado por una fuerza y calcule el centro de masa para una placa plana en el plano.[Evaluar]. • Definir curvas paramétricas y funciones vectoriales encontrando relaciones entre ellas. [Evaluar]. • Aplicar integrales para calcular la longitud de las curvas descritas por las funciones vectoriales.[Evaluar]. • Ser capaz de resolver problemas de aplicaciones simples tales como tráfico en un servicio de Internet, consumo de combustible, tomografía: volumen del cerebro, bomba de agua, masa en espesante, superformula, volumen en máquina de Wankel, longitud de hélice de molécula de ADN, etc.[Evaluar]. 	<ul style="list-style-type: none"> • Estrategia para la integración. • Técnica para integrar funciones. • Herramientas adicionales para encontrar integrales • Problemas de aplicaciones.
Lecturas : [Stewart], [RonLarson]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** EG0004. Desafíos Globales (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 2 HT; 2 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [E15] Upton. E. *Intuición, acción, creación: Graphic Design Thinking*. México:Editorial Gustavo Gili, 2015.
- [R12] Curedale. R. *Design methods 1: 200 ways to apply design thinking*. EE.UU Design Community College Inc, 2012.

6. Información del curso

- (a) **Breve descripción del curso** Durante las sesiones plenarias, se realizarán clases magistrales relacionadas a la metodología de Design Thinking así como su uso e importancia en los procesos de creación . Así mismo, durante estas sesiones tendremos ponencias sobre emprendimientos y startups relacionados a la ingeniería o tecnología. Durante las sesiones de laboratorio, los alumnos forman equipos que mantienen durante el ciclo. Con la guía del profesor y a través de la metodología del Design Thinking desarrollada en las plenarias, los alumnos deberán plantear soluciones innovadoras a problemas reales inspirados en los Global Challenges de las Naciones Unidas. Los alumnos contarán con una Bitácora Digital que será revisada constantemente por los docentes a cargo. En ella se encontrarán los avances, procesos y referentes del proyecto grupal. El curso culmina con las presentaciones de las propuestas planteadas por los grupos.
- (b) **Prerrequisitos:** Ninguno
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Capacidad de diseñar y llevar a cabo experimentos
- Capacidad de analizar información
- Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (Nivel 1)
- Capacidad de trabajo en equipo
- Capacidad de liderar un equipo
- Capacidad de comunicación oral (Nivel 1)
- Capacidad de comunicación escrita (Nivel 1)
- Comprende el impacto de las soluciones de la ingeniería en un contexto global, económico, ambiental y de la sociedad.

8. Contribución a los resultados (*Outcomes*)

- n) Aplicar conocimientos de humanidades en su labor profesional. (**Usar**)

ñ) Comprender que la formación de un buen profesional no se desliga ni se opone sino mas bien contribuye al auténtico crecimiento personal. Esto requiere de la asimilación de valores sólidos, horizontes espirituales amplios y una visión profunda del entorno cultural. (**Usar**)

9. Competencias (IEEE)

C20. Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome n,ñ**

10. Lista de temas a estudiar en el curso

1. Desafíos Globales

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Desafíos Globales (12)	
Competences esperadas: 4	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Flexibilidad y Adaptabilidad: Los alumnos aprenden a trabajar en equipo, en un ambiente flexible, variable y de constantes retos. 	<ul style="list-style-type: none"> • Pasos de DT. • Técnica y usos del Brainstorm. • Conocimiento del usuario, empatía y uso de arquetipos. • Tipos de Investigación, diferencias y usos. • Estrategias de recolección de Insights. • Métodos de Ideación. • Introducción al Prototipado. • Introducción a la Experiencia de Usuario. • Estrategias de Testeo e Iteración • Usos del Storytelling
Lecturas : [R12], [E15]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS1102. Programación Orientada a Objetos I (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [PH13] Deitel. P.J and Deitel. H.M. *C++ How to Program (Early Objects Version)*. Deitel, How to Program. Prentice Hall, 2013. ISBN: 9780133378719. URL: <http://books.google.com.pe/books?id=XIZJNQEACAAJ>.
- [Str13] Bjarne Stroustrup. *The C++ Programming Language*. 4th. Addison-Wesley, 2013. ISBN: 978-0-321-56384-2.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS1100. Introducción a la Ciencia de la Computación. (1^{er} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

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

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C5.** Capacidad para implementar algoritmos y estructuras de datos en el software.⇒ **Outcome b**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome b**
- CS3.** Analizar el grado en que un sistema basado en el ordenador cumple con los criterios definidos para su uso actual y futuro desarrollo.⇒ **Outcome i**

10. Lista de temas a estudiar en el curso

1. Visión General de los Lenguajes de Programación
2. Máquinas virtuales
3. Sistemas de tipos básicos
4. Conceptos Fundamentales de Programación
5. Programación orientada a objetos
6. Algoritmos y Diseño
7. Estrategias Algorítmicas
8. Análisis Básico
9. Algoritmos y Estructuras de Datos fundamentales

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Visión General de los Lenguajes de Programación (1)	
Competences esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse] 	<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.
Lecturas : [Str13], [PH13]	

Unidad 2: Máquinas virtuales (1)	
Competencias esperadas: C2	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • El concepto de máquina virtual. • Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red) . • Lenguajes intermedios.
Lecturas : [Str13], [PH13]	

Unidad 3: Sistemas de tipos básicos (2)	
Competencias esperadas: C1,C2,CS1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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. numeros, booleanos) – Composición de tipos construidos 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.
Lecturas : [Str13], [PH13]	

Unidad 4: Conceptos Fundamentales de Programación (6)	
Competencias esperadas: C1,C2,CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Sintaxis y semántica básica de un lenguaje de alto nivel. • Variables y tipos de datos primitivos (ej., números, caracteres, booleanos) • Expresiones y asignaciones. • Operaciones básicas I/O incluyendo archivos I/O. • Estructuras de control condicional e iterativas. • Paso de funciones y parámetros.
Lecturas : [Str13], [PH13]	

Unidad 5: Programación orientada a objetos (10)	
Competencias esperadas: C2,C24,CS1,CS2	
Objetivos de Aprendizaje	Tópicos
<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 procedurar/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] 	<ul style="list-style-type: none"> • Diseño orientado a objetos: <ul style="list-style-type: none"> – Descomposicion en objetos que almacenan estados y poseen comportamiento – Diseño basado en jerarquia 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 subclases, 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 coleccion de clases, iteradores, y otros componentes de la libreria estandar. • Asignación dinámica: definición de método de llamada.
Lecturas : [Str13], [PH13]	

Unidad 6: Algoritmos y Diseño (3)	
Competencias esperadas: C1,C2,C23	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Str13], [PH13]	

Unidad 7: Estrategias Algorítmicas (3)	
Competencias esperadas: C1,C2,C24,CS1	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Algoritmos de fuerza bruta. • Algoritmos voraces. • Divide y vencerás. • Bactraking recursivo. • Programación Dinámica.
Lecturas : [Str13], [PH13]	

Unidad 8: Análisis Básico (2)	
Competencias esperadas: C1,C2,C24,CS1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Familiarizarse] 	<ul style="list-style-type: none"> • Diferencias entre el mejor, el esperado y el peor caso de un algoritmo.
Lecturas : [Str13], [PH13]	

Unidad 9: Algoritmos y Estructuras de Datos fundamentales (6)	
Competencias esperadas: C1,C2,C24,CS1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [Str13], [PH13]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS1D02. Estructuras Discretas II (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.
- [Gri97] R. Grimaldi. *Matemáticas Discretas y Combinatoria*. Addison Wesley Iberoamericana, 1997.
- [Joh99] Richard Johnsonbaugh. *Matemáticas Discretas*. Prentice Hall, México, 1999.
- [Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 7 ed. Mc Graw Hill, 2007.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS1D01. Estructuras Discretas I. (1^{er} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Familiarizarse**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
 - i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Familiarizarse**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C5.** Capacidad para implementar algoritmos y estructuras de datos en el software.⇒ **Outcome b**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome i**

10. Lista de temas a estudiar en el curso

1. Lógica Digital y Representación de Datos
2. Fundamentos de conteo
3. Árboles y Grafos

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Lógica Digital y Representación de Datos (10)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<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 lógica[Evaluar]. • Minimizar una función booleana [Evaluar]. 	<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.
Lecturas : [Ros07], [Gri03]	

Unidad 2: Fundamentos de conteo (40)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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 – Otros ejemplos, mostrando una variedad de soluciones • Aritmetica modular basica
Lecturas : [Gri97]	

Unidad 3: Árboles y Grafos (40)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Joh99]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** ME0019. Física I (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 4 HT; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[AF95] Marcelo Alonso and Edward Finn. *Física*. Addison Wesley Iberoamericana, 1995. ISBN: 0-201-62565-2.

[SB02] Raymond Serway and Robert Beichner. *Física, para Ciencias e Ingenierias*. Mc Graw Hill, 2002. ISBN: 970-10-3581-X.

6. Información del curso

- (a) **Breve descripción del curso** Este curso es útil en esta carrera para que el alumno aprenda a mostrar un alto grado de dominio de las leyes del movimiento de la Física General.
- (b) **Prerrequisitos:** Ninguno
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Capacitar y presentar al estudiante los principios básicos de la Física como ciencia natural abarcando sus tópicos más importantes y su relación con los problemas cotidianos.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C20.** Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome i,j**

10. Lista de temas a estudiar en el curso

1. FI1. Introducción
2. FI2. Movimiento de partículas en una dimensión

3. FI3. Movimiento de partículas en dos y tres dimensiones
4. FI4. Leyes del movimiento
5. FI5. Trabajo y Energía
6. FI6. Momento lineal
7. FI7. Rotación de cuerpos rígidos
8. FI8. Dinámica del movimiento de rotación

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: FI1. Introducción (4)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Entender y trabajar con las magnitudes físicas del SI. • Abstracter de la naturaleza los conceptos físicos rigurosos y representarlos en modelos vectoriales. • Entender y aplicar los conceptos vectoriales a problemas físicos reales. 	<ul style="list-style-type: none"> • La investigación científica. El método científico. • Concepto de Química. La Química en la actualidad. • Materia. Clasificación y propiedades físicas, químicas, intensivas y extensivas. • Modelo idealizado. • Magnitudes físicas. • Propiedades de los vectores. • Componentes de un vector y vectores unitarios. • Producto de vectores. • Ejercicios y problemas.
Lecturas : [SB02], [AF95]	

Unidad 2: FI2. Movimiento de partículas en una dimensión (2)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Describir matemáticamente el movimiento mecánico de una partícula unidimensional como un cuerpo de dimensiones despreciables. • Conocer y aplicar conceptos de magnitudes cinemáticas. • Describir el comportamiento de movimiento de partículas, teórica y gráficamente. • Conocer representaciones vectoriales de estos movimientos unidimensionales. • Resolver problemas. 	<ul style="list-style-type: none"> • Desplazamiento, velocidad y rapidez. • Velocidad instantánea. • Aceleración media e instantánea. • Movimiento con aceleración constante. • Caída libre de los cuerpos. • Ejercicios y problemas.
Lecturas : [SB02], [AF95]	

Unidad 3: FI3. Movimiento de partículas en dos y tres dimensiones (4)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Describir matemáticamente el movimiento mecánico de una partícula en dos y tres dimensiones como un cuerpo de dimensiones despreciables. • Conocer y aplicar conceptos de magnitudes cinemáticas vectoriales en dos y tres dimensiones. • Describir el comportamiento de movimiento de partículas teórica y gráficamente en dos y tres dimensiones. • Conocer y aplicar conceptos del movimiento circular. • Resolver problemas. 	<ul style="list-style-type: none"> • Desplazamiento y velocidad. • El vector aceleración. • Movimiento parabólico. • Movimiento circular. • Componentes tangencial y radial de la aceleración. • Ejercicios y problemas
Lecturas : [SB02], [AF95]	

Unidad 4: FI4. Leyes del movimiento (6)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Conocer los conceptos de fuerza. • Conocer las interacciones mas importantes de la naturaleza y representarlos en un diagrama de cuerpo libre. • Conocer los conceptos de equilibrio estático. • Conocer y aplicar las leyes del movimiento y caracterizarlos vectorialmente. • Conocer y aplicar las leyes de Newton. • Resolver problemas. 	<ul style="list-style-type: none"> • Fuerza e interacciones. • Primera ley de Newton. • Masa inercial. • Segunda ley de Newton. • Peso. • Diagramas de cuerpo libre. • Tercera Ley de newton. • Fuerzas de fricción. • Dinámica del movimiento circular. • Ejercicios y problemas.
Lecturas : [SB02], [AF95]	

Unidad 5: FI5. Trabajo y Energía (4)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Establecer los conceptos de energía física. (Física clásica) • Conocer algunas formas de energía. • Establecer la relación entre trabajo y energía. • Conocer y aplicar los conceptos de conservación de energía. • Resolver problemas. 	<ul style="list-style-type: none"> • Trabajo realizado por una fuerza constante. • Trabajo realizado por fuerzas variables. • Trabajo y energía cinética. • Potencia. • Energía potencial gravitatoria. • Energía potencial elástica. • Fuerzas conservativas y no conservativas. • Principios de conservación de la energía. • Ejercicios y problemas.
Lecturas : [SB02], [AF95]	

Unidad 6: FI6. Momento lineal (3)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Establecer los conceptos de momento lineal. • Conocer los conceptos de conservación del momento lineal. • Conocer el movimiento de un sistema de partículas. • Resolver problemas. 	<ul style="list-style-type: none"> • Momento lineal. • Conservación del momento lineal. • Centro de masa y de gravedad. • Movimiento de un sistema de partículas. • Ejercicios y problemas.
Lecturas : [SB02], [AF95]	

Unidad 7: FI7. Rotación de cuerpos rígidos (4)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Conocer los conceptos básicos de cuerpo rígido. • Conocer y aplicar conceptos de rotación de cuerpo rígido. • Conocer conceptos de torsión. • Aplicar conceptos de energía al movimiento de rotación. • Resolver problemas. 	<ul style="list-style-type: none"> • Velocidad y aceleraciones angulares. • Rotación con aceleración angular constante. • Relación entre cinemática lineal y angular. • Energía en el movimiento de rotación. • Momento de torsión. • Relación entre momento de torsión y aceleración angular. • Ejercicios y problemas.
Lecturas : [SB02], [AF95]	

Unidad 8: FI8. Dinámica del movimiento de rotación (3)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Conocer conceptos básicos de dinámica de rotación. • Conocer y aplicar conceptos de torsión. • Entender el momento angular y su conservación. • Resolver problemas. 	<ul style="list-style-type: none"> • Momento de torsión y aceleración angular de un cuerpo rígido. • Rotación de un cuerpo rígido sobre un eje móvil. • Trabajo y potencia en el movimiento de rotación. • Momento angular. • Conservación del momento angular. • Ejercicios y problemas.
Lecturas : [SB02], [AF95]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** GH0006. Laboratorio de Comunicación II (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 2 HT; 2 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[D08] Cassany. D. *Prácticas letradas contemporáneas*. DF, México, Ríos de tinta, 2008.

6. Información del curso

- (a) **Breve descripción del curso** Este laboratorio está orientado a consolidar las habilidades comunicativas del estudiante, tanto a nivel oral como escrito en el marco de la disciplina que se estudia. En particular, el estudiante fortalecerá sus capacidades expositivas al ejercitarse en toda la primera parte del curso en la escritura de un tipo de texto que desarrollará a lo largo de su carrera como ingeniero: los informes de laboratorio. Reflexionará sobre la situación retórica que enfrenta al escribir este tipo de texto: quién será su lector, cuál es la intención comunicativa de ese texto y el tema sobre el que está escribiendo. En una segunda parte, el curso se presenta como un espacio de discusión sobre el discurso argumentativo y de lectura crítica de textos argumentativos, para que el alumno reflexione, conozca y emplee las herramientas comunicativas para producir textos argumentativos formales. En este sentido, el curso se orienta hacia la producción permanente de textos escritos y orales, por lo que el alumno participará no solo en foros de discusión sino que se espera que sea capaz de debatir con sus compañeros sobre un tema propuesto por el profesor. En suma, el curso busca consolidar las competencias de lectura, análisis y elaboración de textos escritos y orales, tanto expositivos como argumentativos.
- (b) **Prerrequisitos:** GH0005. Laboratorio de Comunicación I. (1^{er} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Desarrollar habilidades que les permitan a los estudiantes mejorar sus capacidades comunicativas, tanto orales como escritas.
- Comprender y producir textos expositivos en los que informen sobre la aplicación del conocimiento teórico en un experimento o contexto diferente.
- Comprender y producir textos argumentativos orales y escritos.
- Se capaz de debatir empleando argumentos sólidos.
- Emplear adecuadamente y reflexivamente la información obtenida en diferentes fuentes.
- Mostrar apertura y respeto para escuchar la diversidad de opiniones o puntos de vista de los compañeros de clase.

8. Contribución a los resultados (*Outcomes*)

- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

9. Competencias (IEEE)

C17. Capacidad para expresarse en los medios de comunicación orales y escritos como se espera de un graduado.⇒ **Outcome i**

C20. Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome f**

C24. Comprender la necesidad de la formación permanente y la mejora de habilidades y capacidades.⇒ **Outcome f**

10. Lista de temas a estudiar en el curso

1. Laboratorio de Comunicación II

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Laboratorio de Comunicación II (12)	
Competences esperadas: C17	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Manejar adecuadamente el sistema citado y de referencias bibliográficas, y reconocer la importancia de su uso.	<ul style="list-style-type: none">• ¿Qué es un informe de laboratorio ?• Desarrollo del Laboratorio y aplicaciones.• Resultados de Laboratorio y aplicaciones.• Introducción y conclusiones.• Citado, referencias parentéticas y construcción de bibliografía.• Preparación para la exposición oral.• Presentación de un texto Argumentativo: textos formales y no formales.• Citado, referencias (formato APA)
Lecturas : [D08]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** GH0007. Introducción al Desarrollo de Empresas (Obligatorio)
2. **Créditos:** 2
3. **Horas de Teoría y Laboratorio:** 1 HT; 2 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[AY10] Osterwalder. A and Pigneur. Y. *Business Model Generation.* ., 2010.

6. Información del curso

- (a) **Breve descripción del curso** Este curso tiene como objetivo proporcionar a los estudiantes una experiencia práctica de la vida real en los primeros pasos dentro de un ciclo de vida de negocios, a través del cual una idea se transforma en un modelo de negocio formal. Es el primero de un conjunto de tres cursos diseñados para acompañar a los estudiantes a medida que transforman una idea en un negocio o negocio prospectivo, desde la idea hasta la revisión de la estrategia empresarial actual.
- (b) **Prerrequisitos:** GH0005. Laboratorio de Comunicación I. (1^{er} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Capacidad de Análisis de la información.
- Interpretación de información y resultados.
- Capacidad de Trabajo en equipo.
- Ética.
- Comunicación oral.
- Comunicación escrita.
- Comunicación gráfica.
- Entender la necesidad de aprender de forma continua.

8. Contribución a los resultados (*Outcomes*)

- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- e) Entender correctamente las implicancias profesionales, éticas, legales, de seguridad y sociales de la profesión. (**Usar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)
- n) Aplicar conocimientos de humanidades en su labor profesional. (**Usar**)
- o) Mejorar las condiciones de la sociedad poniendo la tecnología al servicio del ser humano. (**Usar**)

9. Competencias (IEEE)

C10. Comprensión del impacto en las personas, las organizaciones y la sociedad de la implementación de soluciones tecnológicas e intervenciones.⇒ **Outcome d,n,o**

C17. Capacidad para expresarse en los medios de comunicación orales y escritos como se espera de un graduado.⇒ **Outcome f**

C18. Capacidad para participar de forma activa y coordinada en un equipo.⇒ **Outcome e**

10. Lista de temas a estudiar en el curso

1. Introducción al Desarrollo de Empresas.

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Introducción al Desarrollo de Empresas. (12)	
Competencias esperadas: 4	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Transformar una idea inicial de negocio o un proceso de innovación en un modelo de negocio factible. 	<ul style="list-style-type: none"> • El ciclo de vida empresarial: desde la idea hasta la revisión de su estrategia. • El proceso de ideación y la visión del cliente. • Cómo construir y mantener equipos eficaces? • Running LEAN: lo básico. • Diseño de un modelo de negocio: herramientas de diseño y Canvas. • Generación de Modelos de Negocio: Modelo de Negocio Canvas (Osterwalder). • Venture Engineering: utilizando las habilidades de la informática para construir un modelo de negocio efectivo. • Herramientas de investigación de mercado primario y nichos de mercado. • La Importancia del Capital: Humano, Financiero e Intelectual • Técnicas de monetización y financiamiento. • Comunicación eficaz: crear una presentación de un modelo de negocio de impacto.
Lecturas : [AY10]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** GH1002. Arte y Tecnología (Obligatorio)
2. **Créditos:** 1
3. **Horas de Teoría y Laboratorio:** 2 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [J12] Maeda J. *Processing: A Programming Handbook for Visual Designers and Artists*. Cambridge: The MIT Press, 2012.
- [S02] Wilson. S. *Intersections of Art, Science and Technology*. Cambridge: The MIT Press, 2002.

6. Información del curso

- (a) **Breve descripción del curso** El curso busca dar una visión global, histórica y crítica de las transformaciones y sinergias del arte contemporáneo. Donde los alumnos se aproximan a dos componentes del arte y diseño contemporáneo: las prácticas interdisciplinarias y a los puntos de contacto entre las artes y los procesos tecnológicos y de ingeniería.
- (b) **Prerrequisitos:** EG0004. Desafíos Globales. (1^{er} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Desarrollar la capacidad de analizar información .
- Desarrollar la capacidad de interpretar información.
- Desarrollar la Capacidad de trabajo en equipo.
- Desarrollar la Capacidad de comunicación oral.
- Reconocer la necesidad del aprendizaje permanente.

8. Contribución a los resultados (*Outcomes*)

- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- e) Entender correctamente las implicancias profesionales, éticas, legales, de seguridad y sociales de la profesión. (**Usar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)
- n) Aplicar conocimientos de humanidades en su labor profesional. (**Usar**)
- o) Mejorar las condiciones de la sociedad poniendo la tecnología al servicio del ser humano. (**Usar**)

9. Competencias (IEEE)

- C10.** Comprensión del impacto en las personas, las organizaciones y la sociedad de la implementación de soluciones tecnológicas e intervenciones.⇒ **Outcome d,n,o**

C17. Capacidad para expresarse en los medios de comunicación orales y escritos como se espera de un graduado.⇒ **Outcome f**

C18. Capacidad para participar de forma activa y coordinada en un equipo.⇒ **Outcome f**

C21. Comprender el aspecto profesional, legal, seguridad, asuntos políticos, humanistas, ambientales, culturales y éticos.⇒ **Outcome e**

10. Lista de temas a estudiar en el curso

1. Arte y Tecnología
2. Arte Digita
3. Prototipado, análisis y creación

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Arte y Tecnología (12)	
Competencias esperadas: 4	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Impulsar el interés por conocer sobre temas actuales de la sociedad peruana y del mundo.	<ul style="list-style-type: none">• ¿Qué es el arte y para qué sirve?• El discurso artístico: identidad, territorio, política y sociedad.
Lecturas : [S02]	

Unidad 2: Arte Digita (24)	
Competencias esperadas: 3	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Desarrollo de habilidades como: creatividad, pensamiento crítico, observación y síntesis.	<ul style="list-style-type: none">• Arte Generativo.• Net Art.• Realidad Virtual.
Lecturas : [J12]	

Unidad 3: Prototipado, análisis y creación (24)	
Competencias esperadas: 3	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Los alumnos entienden la importancia y efectividad del trabajo en equipo tanto en la vida académica como profesional. Durante el semestre los estudiantes realizan actividades grupales e individuales cuyo objetivo común es la generación de un proyecto que vincule conceptos de arte, tecnología e ingeniería. 	<ul style="list-style-type: none"> • Fabricación Digital. • Intervención: Acción y espacio público. • Presentación: Montaje, portafolio.
Lecturas : [S02]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** GH1101. Inglés I (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 10 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.
[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.
[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

6. Información del curso

- (a) **Breve descripción del curso** Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida de las personas. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde éste no sea hablado. En las carreras relacionadas con los servicios al turista el Inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento, como parte de su formación integral.
- (b) **Prerrequisitos:** Ninguno
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Conocer el idioma Inglés y su estructura gramatical.
- Identificar situaciones y emplear diálogos relacionados a ellas.

8. Contribución a los resultados (*Outcomes*)

- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

9. Competencias (IEEE)

C25. Capacidad para comunicarse en un segundo idioma.⇒ **Outcome f**

10. Lista de temas a estudiar en el curso

1. Hello everybody
2. Meeting people
3. The world of work
4. Take it easy

5. Where do you live ?
6. Can you speak English?
7. Then and now

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Hello everybody (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la primera unidad, cada uno de los alumnos, comprendiendo la gramática del tiempo presente es capaz de expresar una mayor cantidad de expresiones de tiempo y además usar oraciones con el verbo To Be para expresar situación y estado. • Que el alumno sea capaz de analizar y expresar ideas acerca de fechas y números en orden. 	<ul style="list-style-type: none"> • Verbo To Be. • Oraciones Afirmativas, Negativas y Preguntas. • Expresiones Numéricas. • Objetos y Países. • Expresiones para saludar y hacer presentaciones.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 2: Meeting people (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la segunda unidad, los alumnos habiendo identificado la forma de expresar pedidos y hacer ofrecimientos en restaurantes los utilizan en situaciones varias. Explica y aplica vocabulario de comidas y bebidas. 	<ul style="list-style-type: none"> • Adjetivos Posesivos. • Expresiones para averiguar precios. • Expresiones de Posesión. • Vocabulario de Familia, Comidas y Bebidas. • Pedidos formales. • Cartas informales.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 3: The world of work (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la tercera unidad, los alumnos habiendo reconocido las características del presente simple, lo utiliza para hacer descripciones de diversos tipos. Describen personas y lugares y dan indicaciones de dirección. Expresa la hora. 	<ul style="list-style-type: none"> • Tiempo Presente Simple. Auxiliares. • Oraciones Afirmativas, Negativas y Preguntas. • Verbos comunes y Ocupaciones. • Indicaciones para expresar la hora.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 4: Take it easy (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la cuarta unidad, los alumnos habiendo identificado la idea de expresar ideas de acciones de tiempo libre en Presente Simple y Continuo. Expresan ideas de estaciones y actividades relacionadas. 	<ul style="list-style-type: none"> • Presente Simple 2. • Oraciones Afirmativas, Negativas y Preguntas. • Uso de Verbos de entretenimiento. • Tiempo Libre. • Las estaciones del año. • Expresiones de actividades sociales.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 5: Where do you live ? (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la quinta unidad, los alumnos, a partir de la comprensión del tiempo presente continuo, elaborarán oraciones utilizando ideas de ubicación y de lugar. Asimilarán además la necesidad de expresar objetos de uso común. Adquirirán vocabulario para describir las partes de una casa usan expresiones para pedir indicaciones de dirección. 	<ul style="list-style-type: none"> • Uso There is/There are. • Oraciones con Preposiciones. • Expresiones de Cantidad. • Vocabulario de aviones y lugares. • Expresiones de indicaciones de dirección.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 6: Can you speak English? (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la sexta unidad, los alumnos habiendo conocido los fundamentos del uso de auxiliares de modo, crearan oraciones aplicadas al contexto adecuado. Enfatizan la diferencia entre idiomas y nacionalidades. Describen sentimientos. Utilizan expresiones en el teléfono. 	<ul style="list-style-type: none"> • Can/cant. • Pasado del verbo To Be. Uso de Could. • Vocabulario de países e idiomas. • Expresiones para el uso del teléfono. • Redacción de cartas formales. • Lecturas.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 7: Then and now (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la séptima unidad, los alumnos habiendo conocido los fundamentos de la estructuración del Pasado Simple experimentan la necesidad de poder expresar este tipo de tiempo en acciones. Realizarán prácticas en contextos adecuados. Enfatizan la diferencia entre verbos irregulares y regulares. Describen acciones con verbos varios. Utilizan expresiones para describir el clima. 	<ul style="list-style-type: none"> • Pasado Simple. • Expresiones de tiempo pasado. • Vocabulario verbos regulares e irregulares. • Expresiones para describir el clima. • Redacción de párrafos descriptivos. • Ocasiones Especiales.
Lecturas : [SJ02], [Cam06], [Mac99]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** EG0005. Matemática II (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 4 HT; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[Ste12] James Stewart. *Calculus*. 7th. CENGAGE Learning, 2012.

[Zil13] Dennis G. Zill. *Differential equations with Boundary value problems*. 8th. CENGAGE Learning, 2013.

6. Información del curso

- (a) **Breve descripción del curso** El curso desarrolla en los estudiantes las habilidades para manejar modelos de habilidades de ingeniería y ciencia. En la primera parte Del curso un estudio de las funciones de varias variables, derivadas parciales, integrales múltiples y una Introducción a campos vectoriales. Luego el estudiante utilizará los conceptos básicos de cálculo para modelar y resolver ecuaciones diferenciales ordinarias utilizando técnicas como las transformadas de Laplace y las series de Fourier.
- (b) **Prerrequisitos:** EG0003. Matemática I. (1^{er} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Aplicar reglas de derivación y diferenciación parcial en funciones de varias variables.
- Aplicar técnicas para el cálculo de integrales múltiples.
- Comprender y utilizar los conceptos de cálculo vectorial.
- Comprender la importancia de las series.
- Identificar y resolver ecuaciones diferenciales de primer orden y sus aplicaciones en problemas químicos y físicos.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Evaluar**)

9. Competencias (IEEE)

- C1. La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C20. Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome j**

10. Lista de temas a estudiar en el curso

1. Multi-Variable Function Differential
2. Multi-Variable function Integral
3. Series
4. Ordinary Differential Equations

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Multi-Variable Function Differential (24)	
Competences esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Comprender el concepto de funciones multi-variables. • Dominar el concepto y método de cálculo de la derivada direccional y gradiente de la guía. • Dominar el método de cálculo de la derivada parcial de primer orden y de segundo orden de las funciones compuestas. • Dominar línea tangente, plano normal a línea de curva y plano tangente, línea normal a un plano de curva. Saber calcular sus ecuaciones. • Dominar el método de cálculo de las derivadas parciales para funciones implícitas. • Entender línea tangente, plano normal a línea de curva y plano tangente, línea normal a un plano de curva. Saber calcular sus ecuaciones. • Aprender el concepto de valor extremo y valor extremo condicional de funciones multi-variables; Saber para averiguar el valor extremo de la función binaria. • Ser capaz de resolver problemas de aplicaciones simples. 	<ul style="list-style-type: none"> • Concepto de funciones multi-variables. • Derivados Direccionales • Línea tangente, plano normal a línea de curva y plano tangente, línea normal a un plano de curva. Conocer para calcular sus ecuaciones. • Concepto de valor extremo y valor extremo condicional de funciones multi-variables. • Problemas de aplicación tales como modelización de la producción total de un sistema económico, velocidad del sonido a través del océano, optimización del espesante, etc.
Lecturas : [Ste12], [Zil13]	

Unidad 2: Multi-Variable function Integral (12)	
Competences esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Entender la integral doble, integral triple, y entender la naturaleza de la integral múltiple. • Dominar el método de cálculo de la integral doble (coordenadas cartesianas, coordenadas polares), la integral triple (coordenadas cartesianas, coordenadas cilíndricas, coordenadas esféricas). • Entender el concepto de línea Integral, sus propiedades y relaciones. • Saber calcular la integral de línea. • Dominar el cálculo de la rotación, la divergencia y Laplaciano. 	<ul style="list-style-type: none"> • Integral doble, integral triple y naturaleza de la integral múltiple. • Método de doble integral • Línea integral • La Divergencia, Rotación y Laplaciano
Lecturas : [Ste12], [Zil13]	

Unidad 3: Series (24)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Dominio del cálculo si la serie es convergente, y si es convergente, encontrar la suma de la serie tratando de encontrar el radio de convergencia y el intervalo de convergencia de una serie de potencia. • Representa una función como una serie de potencias y encuentra la serie de Taylor y MacLaurin para estimar los valores de las funciones con la precisión deseada. • Entender los conceptos de funciones ortogonales y la expansión de una función dada f para encontrar su serie de Fourier. 	<ul style="list-style-type: none"> • Serie convergente. • Serie Taylor y MacLaurin. • Funciones ortogonales.
Lecturas : [Ste12], [Zil13]	

Unidad 4: Ordinary Differential Equations (30)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Comprender ecuaciones diferenciales, soluciones, orden, solución general, condiciones iniciales y soluciones especiales, etc. • Dominar el método de cálculo para las variables ecuación separable y ecuaciones lineales de primer orden. Conocido para resolver la ecuación homogénea y las ecuaciones de Bernoulli (Bernoulli); Entender la sustitución de la variable para resolver la ecuación. • Diminio para resolver ecuaciones diferenciales totales. • Ser capaz de utilizar el método de orden reducido para resolver ecuaciones. • Comprender la estructura de la ecuación diferencial lineal de segundo orden. • Dominio del cálculo para las ecuaciones diferenciales lineales homogéneas de coeficiente constante; Y comprender el método de cálculo para las ecuaciones diferenciales lineales homogéneas de orden superior. • Saber aplicar el método de cálculo de ecuaciones diferenciales para resolver problemas simples de aplicación geométrica y física. • Resolver correctamente ciertos tipos de ecuaciones diferenciales utilizando transformadas de Laplace. 	<ul style="list-style-type: none"> • Concepto de ecuaciones diferenciales • Métodos para resolver ecuaciones diferenciales • Métodos para resolver las ecuaciones diferenciales lineales de segundo orden • Ecuaciones diferenciales ordinarias lineales de orden superior • Problemas de aplicaciones con las transformaciones de Laplace
Lecturas : [Ste12], [Zil13]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS1103. Programación Orientada a Objetos II (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 3 HT; 2 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Nak13] S. Nakariakov. *The Boost C++ Libraries: Generic Programming*. CreateSpace Independent Publishing Platform, 2013.
- [Str13] B Stroustrup. *The C++ Programming Language, 4th edition*. Addison-Wesley, 2013.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS1102. Programación Orientada a Objetos I. (2^{do} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

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

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C3.** Una comprensión intelectual de, y el aprecio por el papel central de los algoritmos y estructuras de datos.⇒ **Outcome a**
- C18.** Capacidad para participar de forma activa y coordinada en un equipo.⇒ **Outcome d**
- CS1.** Modelar y diseñar sistemas de computadora de una manera que se demuestre comprensión del balance entre las opciones de diseño.⇒ **Outcome a**

CS2. Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome b**

10. Lista de temas a estudiar en el curso

1. Conceptos Fundamentales de Programación
2. Algoritmos y Diseño
3. Programación reactiva y dirigida por eventos
4. Análisis Básico
5. Algoritmos y Estructuras de Datos fundamentales
6. Sistemas de tipos básicos
7. Programación orientada a objetos
8. Árboles y Grafos
9. Diseño de Software
10. Ingeniería de Requisitos

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Conceptos Fundamentales de Programación (5)	
Competencias esperadas: C1,C18	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Sintaxis y semántica básica de un lenguaje de alto nivel. • Variables y tipos de datos primitivos (ej., números, 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.
Lecturas : [Str13]	

Unidad 2: Algoritmos y Diseño (5)	
Competencias esperadas: C3,C18	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Str13]	

Unidad 3: Programación reactiva y dirigida por eventos (2)	
Competencias esperadas: C1,C18	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Str13]	

Unidad 4: Análisis Básico (3)	
Competencias esperadas: CS2,C18	
Objetivos de Aprendizaje	Tópicos
<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 simples 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] 	<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. • Algunas versiones del Teorema Maestro.
Lecturas : [Str13]	

Unidad 5: Algoritmos y Estructuras de Datos fundamentales (3)	
Competencias esperadas: C3,C18	
Objetivos de Aprendizaje	Tópicos
<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] 	<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"> – Algoritmos de la ruta más corta (algoritmos de Dijkstra y Floyd) – Árbol de expansión mínima (algoritmos de Prim y Kruskal) • Búsqueda de patrones y algoritmos de cadenas/texto (ej. búsqueda de subcadena, búsqueda de expresiones regulares, algoritmos de subsecuencia común más larga)
Lecturas : [Str13]	

Unidad 6: Sistemas de tipos básicos (5)	
Competencias esperadas: C1,C18	
Objetivos de Aprendizaje	Tópicos
<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] 	<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. numeros, booleanos) – Composición de tipos construidos 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 creacion 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
Lecturas : [Str13]	

Unidad 7: Programación orientada a objetos (7)	
Competencias esperadas: C1,C18	
Objetivos de Aprendizaje	Tópicos
<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 procedurar/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 [Usar] • 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) [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 mas natural por cada lenguaje [Usar] 	<ul style="list-style-type: none"> • Diseño orientado a objetos: <ul style="list-style-type: none"> – Descomposicion en objetos que almacenan estados y poseen comportamiento – Diseño basado en jerarquia 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 coleccion de clases, iteradores, y otros componentes de la libreria estandar.
Lecturas : [Str13]	

Unidad 8: Árboles y Grafos (7)	
Competencias esperadas: C3,C18	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Nak13]	

Unidad 9: Diseño de Software (6)	
Competencias esperadas: CS1,C18	
Objetivos de Aprendizaje	Tópicos
<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 [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 (<i>3-tier</i>), <i>pipe-and-filter</i>, 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] 	<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: seleccion de componentes,diseño,adaptacion y componentes de ensamblaje, componentes y patrones, componentes y objetos(por ejemplo,construir una GUI usando un standar 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. • Aaplicaciones en frameworks. • Middleware: El paradigma de la orientacion 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

Unidad 10: Ingeniería de Requisitos (1)	
Competencias esperadas: CS1,C18	
Objetivos de Aprendizaje	Tópicos
<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] 	<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 elicitation. • 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.
Lecturas : [Str13]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2201. Arquitectura de Computadores (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 2 HT; 2 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Den05] Peter J. Denning. "The locality principle". In: *Commun. ACM* 48.7 (July 2005), pp. 19–24. ISSN: 0001-0782. DOI: 10.1145/1070838.1070856. URL: <http://doi.acm.org/10.1145/1070838.1070856>.
- [Don06] J. Dongarra. "Trends in high performance computing: a historical overview and examination of future developments". In: *Circuits and Devices Magazine, IEEE* 22.1 (2006), pp. 22–27. ISSN: 8755-3996. DOI: 10.1109/MCD.2006.1598076.
- [EA05] Hesham El-Rewini and Mostafa Abd-El-Barr. *Advanced Computer Architecture and Parallel Processing*. Hoboken, NJ: John Wiley & Sons, 2005. ISBN: 0-471-46740-5.
- [HP06] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. 4th. San Mateo, CA: Morgan Kaufman, 2006.
- [Joh91] M. Johnson. *Superscalar microprocessor design*. Prentice Hall series in innovative technology. Prentice Hall, 1991. ISBN: 9780138756345.
- [Par02] Behrooz Parhami. *Introduction to parallel processing: algorithms and architectures*. Plenum series in computer science. Plenum Press, 2002. ISBN: 9780306459702.
- [Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers*. New York: Oxford Univ. Press, 2005. ISBN: ISBN 0-19-515455-X.
- [PH04] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 3rd ed. San Mateo, CA: Morgan Kaufman, 2004.
- [Sta10] William Stalings. *Computer Organization and Architecture: Designing for Performance*. 8th. Upper Saddle River, NJ: Prentice Hall, 2010.

6. Información del curso

- (a) **Breve descripción del curso** Es necesario que el profesional en Ciencia de la Computación tenga sólido conocimiento de la organización y funcionamiento de los diversos sistema de cómputo actuales en los cuales gira se instala el entorno de programación. Con ello también sabrá establecer los alcances y límites de las aplicaciones que se desarrollen de acuerdo a la plataforma siendo usada.

Se tratarán los siguientes temas: componentes de lógica digital básicos en un sistema de computación, diseño de conjuntos de instrucciones, microarquitectura del procesador y ejecución en *pipelining*, organización de la memoria: caché y memoria virtual, protección y compartición, sistema I/O e interrupciones, arquitecturas super escalares y ejecución fuera de orden, computadoras vectoriales, arquitecturas para *multithreading*, multiprocesadores simétricos, modelo de memoria y sincronización, sistemas integrados y computadores en paralelo.

- (b) **Prerrequisitos:** CS1D02. Estructuras Discretas II. (2^{do} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Este curso tiene como propósito ofrecer al estudiante una base sólida de la evolución de las arquitecturas de computadores y los factores que influenciaron en el diseño de los elementos de *hardware* y *software* en sistemas de computación actuales.
- Garantizar la comprensión de cómo es el *hardware* en sí y cómo interactúan *hardware* y *software* en un sistema de cómputo actual.
- Tratar los siguientes temas: componentes de lógica digital básicos en un sistema de computación, diseño de conjuntos de instrucciones, microarquitectura del procesador y ejecución en *pipelining*, organización de la memoria: caché y memoria virtual, protección y compartición, sistema I/O e interrupciones, arquitecturas super escalares y ejecución fuera de orden, computadoras vectoriales, arquitecturas para *multithreading*, multiprocesadores simétricos, modelo de memoria y sincronización, sistemas integrados y computadores en paralelo.

8. Contribución a los resultados (*Outcomes*)

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)

9. Competencias (IEEE)

- C4. Una comprensión del hardware de la computadora desde la perspectiva del software, por ejemplo, el uso del procesador, memoria, unidades de disco, pantalla, etc.⇒ **Outcome i**
- C8. Entendimiento de lo que las tecnologías actuales pueden y no pueden lograr.⇒ **Outcome b,i,g**
- C9. Comprensión de las limitaciones de la computación, incluyendo la diferencia entre lo que la computación es inherentemente incapaz de hacer frente a lo que puede lograrse a través de un futuro de ciencia y tecnología.⇒ **Outcome b,g**

10. Lista de temas a estudiar en el curso

1. Lógica digital y sistemas digitales
2. Representación de datos a nivel máquina
3. Organización de la Máquina a Nivel Ensamblador
4. Organización funcional
5. Organización y Arquitectura del Sistema de Memoria
6. Interfaz y comunicación
7. Multiprocesamiento y arquitecturas alternativas
8. Mejoras de rendimiento

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:**12. Contenido**

Unidad 1: Lógica digital y sistemas digitales (18)	
Competencias esperadas: C8	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Describir el avance paulatino de los componentes de la tecnología de computación, 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, sistematizar, 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] 	<ul style="list-style-type: none"> • Revisión e historia de la Arquitectura de Computadores. • Lógica combinacional vs. secuencial/Arreglos de puertas de campo programables como bloque fundamental de construcción lógico combinacional-secuencial. • Múltiples representaciones / Capas de interpretación (El hardware es solo otra capa) • Herramientas de diseño asistidas por computadora que procesan hardware y representaciones arquitecturales. • Registrar transferencia notación / Hardware language descriptivo (Verilog/VHDL) • Restricción física (Retrasos de Entrada, fan-in, fan-out, energía/poder)
Lecturas : [Par05], [PH04]	

Unidad 2: Representación de datos a nivel máquina (8)	
Competencias esperadas: C9	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Bits, Bytes y Words. • Representacion 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.
Lecturas : [Par05], [Sta10]	

Unidad 3: Organización de la Máquina a Nivel Ensamblador (8)	
Competencias esperadas: C4,CS3	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Organización Básica de la Máquina de Von Neumann. • Unidad de Control. • Paquetes de instrucciones y tipos (manipulación de información, control, I/O) • Assembler / 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.
Lecturas : [Par05], [PH04], [HP06]	

Unidad 4: Organización funcional (8)	
Competencias esperadas: C9	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Implementación de rutas de datos simples, incluyendo la canalización de instrucciones, detección de riesgos y la resolución. • Control de unidades: Realización Cableada vs Realización Microprogramada. • Instrucción (Pipelining) • Introducción al paralelismo al nivel de instrucción (PNI)
Lecturas : [Par05], [HP06]	

Unidad 5: Organización y Arquitectura del Sistema de Memoria (8)	
Competencias esperadas: CS3	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Identifique las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarizarse] • Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse] • Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Usar] • Describa 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] 	<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 intercalación. • 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.
Lecturas : [Par05], [PH04], [Den05]	

Unidad 6: Interfaz y comunicación (8)	
Competencias esperadas: C4,C9,CS3	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Par05], [Sta10]	

Unidad 7: Multiprocesamiento y arquitecturas alternativas (8)	
Competencias esperadas: C9	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Ley potencial. • Ejemplos de juego 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.
Lecturas : [Par05], [Par02], [EA05]	

Unidad 8: Mejoras de rendimiento (8)	
Competencias esperadas: C8,C9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Par05], [Par02], [PH04], [Don06], [Joh91]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2B01. Desarrollo Basado en Plataformas (Obligatorio)
2. **Créditos:** 2
3. **Horas de Teoría y Laboratorio:** 1 HT; 2 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.

- (b) **Prerrequisitos:** CS1102. Programación Orientada a Objetos I. (2^{do} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. (**Usar**)

i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)

9. Competencias (IEEE)

C1. La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome c,d,i**

C6. Capacidad para diseñar y poner en práctica las unidades estructurales mayores que utilizan algoritmos y estructuras de datos y las interfaces a través del cual estas unidades se comunican.⇒ **Outcome c,d,i**

CS8. Aplicar los principios de la interacción persona-ordenador para la evaluación y la construcción de una amplia gama de materiales, incluyendo interfaces de usuario, páginas web, sistemas multimedia y sistemas móviles.⇒ **Outcome g**

10. Lista de temas a estudiar en el curso

1. Introducción
2. Plataformas web
3. Desarrollo de servicios y aplicaciones web
4. Plataformas móviles
5. Aplicaciones Móviles para dispositivos Android

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Introducción (5)	
Competencias esperadas: CS8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Gro09], [ADC13]	

Unidad 2: Plataformas web (5)	
Competencias esperadas: C1,C6	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Fie00]	

Unidad 3: Desarrollo de servicios y aplicaciones web (25)	
Competencias esperadas: C1,C6	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [FR11]	

Unidad 4: Plataformas móviles (5)	
Competencias esperadas: C1,C6	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Mar17]	

Unidad 5: Aplicaciones Móviles para dispositivos Android (25)	
Competencias esperadas: C1,C6	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [ADC13]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** EN0021. Física II (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 4 HT; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [EL98] Robert Eisberg and Lawrence Lerner. *Física: Fundamentos y Aplicaciones*. Vol. 1. Mc Graw Hill, 1998.
- [RH98] Robert Resnick and David Halliday. *Física para Estudiantes de Ciencias e Ingeniería*. John Wiley, 1998.
- [Sea98] Francis Sears. *Física Universitaria*. Addison Wesley-Longman, 1998.

6. Información del curso

- (a) **Breve descripción del curso** Mostrar un alto grado de dominio de las leyes del movimiento ondulatorio, la naturaleza de los fluidos y la termodinámica. Utilizando adecuadamente los conceptos de movimiento ondulatorio, de fluidos y de termodinámica en la resolución de problemas de la vida cotidiana. Poseer capacidad y habilidad en la interpretación de los fenómenos ondulatorios, de fluidos y termodinámicos, que contribuyan en la elaboración de soluciones eficientes y útiles en diferentes áreas de la ciencia de la computación.
- (b) **Prerrequisitos:** ME0019. Física I. (2^{do} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Que el alumno aprenda y domine los principios fluidos estáticos y en movimiento.
- Que el alumno aprenda y domine los principios del MAS, particularmente del movimiento ondulatorio.
- Que el alumno aprenda y domine los principios de Termodinámica.
- Que el alumno aprenda a aplicar principios de la Física de fluidos, ondas y termodinámica para desarrollar modelos computacionales.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**

C20. Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome i,j**

10. Lista de temas a estudiar en el curso

1. FI1. Elasticidad
2. FI2. Fluidos
3. FI3. Movimiento Periódico
4. FI4. Ondas
5. FI5. Temperatura y Teoría Cinética
6. FI6. Calor y primera Ley de la Termodinámica
7. FI7. Máquinas térmicas, entropía y la segunda ley de la Termodinámica

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: FI1. Elasticidad (4)	
Competences esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Entender y caracterizar los procesos de elasticidad• Resolver problemas	<ul style="list-style-type: none">• Esfuerzo y deformación unitaria• Módulo de Young• Módulo y Coeficiente de Poisson• Módulo de Rigidez• Módulo y Coeficiente de compresibilidad
Lecturas : [Sea98], [EL98]	

Unidad 2: FI2. Fluidos (8)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar, analizar y caracterizar la presión de fluidos • Entender, caracterizar y aplicar el principio de Arquímedes • Entender, caracterizar y aplicar el principio de Bernoulli • Explicar, analizar y caracterizar la tensión superficial y capilaridad 	<ul style="list-style-type: none"> • Densidad y peso específico • Presión en los fluidos. Presión atmosférica y presión manométrica • Principio de Pascal. Medición de la presión: manómetro y barómetro • Flotabilidad y Principio de Arquímedes • Fluidos en movimiento: flujo y ecuación de continuidad • Ecuación de Bernoulli. Aplicaciones del principio de Bernoulli: teorema de Torricelli, el tubo ventura • Tensión superficial y capilaridad
Lecturas :	

Unidad 3: FI3. Movimiento Periódico (8)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar, analizar y caracterizar el movimiento oscilatorio a partir del MAS. • Resolver problemas. 	<ul style="list-style-type: none"> • Introducción. Constante elástica de un resorte • Movimiento armónico simple. Energía en el oscilador armónico simple • Círculo de referencia: el período y la naturaleza senoidal del movimiento armónico simple • Péndulo simple. • Movimiento armónico amortiguado. • Oscilaciones forzadas: resonancia.
Lecturas :	

Unidad 4: FI4. Ondas (8)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar, encontrar y caracterizar mediante problemas de la vida cotidiana el movimiento ondulatorio, así como, la reflexión y transmisión de ondas en el espacio • Resolver problemas 	<ul style="list-style-type: none"> • Movimiento ondulatorio. Tipos de ondas. Ondas viajeras unidimensionales • Superposición e interferencia de ondas • Velocidad de las ondas en una cuerda tensa. Reflexión y transmisión de ondas • Ondas senoidales. Energía transmitida por ondas senoidales en cuerdas • Ondas estacionarias en una cuerda. Ondas sonoras. Velocidad de las ondas sonoras • Ondas sonoras periódicas. Intensidad de ondas sonoras periódicas • Fuentes de sonido: cuerdas vibratorias y columnas vibratorias de aire • Efecto Doppler
Lecturas :	

Unidad 5: FI5. Temperatura y Teoría Cinética (12)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar, analizar y caracterizar el concepto de Temperatura y la dilatación térmica de sólidos y líquidos • Entender la ley del gas ideal y los procesos isotérmicos y adiabáticos para un gas ideal • Entender la ley cero de la Termodinámica • Resolver problemas 	<ul style="list-style-type: none"> • Átomos. Temperatura. Termómetros y escalas de temperatura • Dilatación térmica de sólidos y líquidos. Coeficientes de dilatación lineal, superficial y cúbico • Leyes de los gases y la temperatura absoluta. La ley del gas ideal en términos moleculares: número de Avogadro • Teoría cinética e interpretación molecular de la temperatura. Distribución de velocidades moleculares • Procesos isotérmicos y adiabáticos para un gas ideal. La equipartición de la energía • Termodinámica. Tipos de sistemas que estudia la Termodinámica • Ley cero de la Termodinámica • El termómetro de gas a volumen constante y la escala Kelvin • Punto triple del agua
Lecturas :	

Unidad 6: FI6. Calor y primera Ley de la Termodinámica (8)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Entender el concepto de calor y de energía interna de un gas ideal • Explicar, analizar y caracterizar la primera ley de la Termodinámica • Resolver problemas 	<ul style="list-style-type: none"> • Calor como transferencia de energía • Capacidad calorífica y calor específico • Energía interna de un gas ideal • Calor específico de un gas ideal • Cambios de fase. Calor latente de fusión y de vaporización • Calorimetría. Trabajo y calor en procesos termodinámicos • La primera ley de la Termodinámica • Algunas aplicaciones de la primera ley de la Termodinámica • Transmisión del calor por conducción, convección y radiación
Lecturas : [EL98], [RH98]	

Unidad 7: FI7. Máquinas térmicas, entropía y la segunda ley de la Termodinámica (8)	
Competencias esperadas: C1,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar, analizar y caracterizar la primera ley de la Termodinámica • Explicar, analizar y caracterizar la máquina de Carnot • Resolver problemas 	<ul style="list-style-type: none"> • Máquinas térmicas y la segunda ley de la Termodinámica • Procesos reversibles e irreversibles. La máquina de Carnot • Escala de temperatura absoluta. Refrigeradores • Entropía. Cambios de entropía en procesos irreversibles
Lecturas : [EL98], [RH98]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** GH0008. Gestión de Empresas (Obligatorio)
2. **Créditos:** 2
3. **Horas de Teoría y Laboratorio:** 1 HT; 2 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[A12] Maurya. A. *Running lean: Iterate from plan A to a plan that works*. Sebastopol, 2012.

[PF03] Kotler. P and Trias de Bes. F. *Marketing Lateral*. Madrid, Person Prentice Hill., 2003.

6. Información del curso

- (a) **Breve descripción del curso** Este curso tiene como objetivo proporcionar a los estudiantes con las herramientas necesarias para ir un paso más allá de la idea inicial y modelo de negocio. Aprenderán los primeros pasos hacia la conceptualización de una empresa y la construcción de su equipo. También explorarán los fundamentos de la creación de un plan de negocios eficaz. Es el segundo de un conjunto de tres cursos diseñados para acompañar a los estudiantes a medida que transforman una idea en un negocio o emprendimiento, desde la ideación, hasta la revisión de la estrategia empresarial actual
- (b) **Prerrequisitos:** GH0007. Introducción al Desarrollo de Empresas. (2^{do} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Entendimiento de los conceptos básicos del proceso de planificación de negocios y su papel dentro del ciclo de vida empresarial

8. Contribución a los resultados (*Outcomes*)

- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)
- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Usar**)
- n) Aplicar conocimientos de humanidades en su labor profesional. (**Usar**)

9. Competencias (IEEE)

- C17.** Capacidad para expresarse en los medios de comunicación orales y escritos como se espera de un graduado. ⇒ **Outcome f,h,n**
- C20.** Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad. ⇒ **Outcome f,n**
- C24.** Comprender la necesidad de la formación permanente y la mejora de habilidades y capacidades. ⇒ **Outcome f,h**

10. Lista de temas a estudiar en el curso

1. Gestión de Empresas

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Gestión de Empresas (16)	
Competencias esperadas: C17,C20	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">Entendimiento de la importancia de una planificación eficaz y cómo contribuye al lanzamiento y éxito de una empresa.	<ul style="list-style-type: none">El ciclo de vida empresarial:¿Por qué necesito un plan de negocios?Diferencias entre el modelo de negocio y la planificación empresarialLa importancia de un equipo bien estructuradoAnálisis ambiental y principales herramientas de investigación de mercadoPlanificación Estratégica: Por qué es necesario y cómo se haceLa importancia del capital: humano, financiero e intelectualCómo construir un plan de operacionesLos fundamentos del marketing: definir estrategia de marketingProyecciones financieras: costos y ventasAsuntos legalesNegocios Responsables: lo básico
Lecturas : [A12], [PF03]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** GH1102. Inglés II (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 10 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.
[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.
[SJ02] Liz Soars and John. *American Headway N 1 Student Book*. Editorial Oxford, 2002.

6. Información del curso

- (a) **Breve descripción del curso** Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida de las personas. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde éste no sea hablado. En las carreras relacionadas con los servicios al turista el Inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento, como parte de su formación integral.
- (b) **Prerrequisitos:** GH1101. Inglés I. (2^{do} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Desarrollar la capacidad de hablar fluidamente el idioma.
- Incrementar el vocabulario y el manejo de expresiones simples.

8. Contribución a los resultados (*Outcomes*)

- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

9. Competencias (IEEE)

C25. Capacidad para comunicarse en un segundo idioma. ⇒ **Outcome f**

10. Lista de temas a estudiar en el curso

1. How long ago?
2. Food you like!
3. The world of work
4. Looking good!

5. Life is an adventure!
6. You're pretty smart!
7. Have you ever?

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: How long ago? (0)	
Competences esperadas: 2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la octava unidad, cada uno de los alumnos, comprendiendo la gramática del tiempo pasado es capaz de expresar una mayor cantidad de expresiones de tiempo y además usar preposiciones para describir lugares y tiempos variados. Además es capaz de analizar y expresar ideas acerca de fechas y números en orden. 	<ul style="list-style-type: none"> • Pasado Simple • Oraciones Negativas con ago. • Conjunciones • Expresiones de Tiempo en pasado • Relaciones y símbolos fonéticos • Expresiones para dar la fecha
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 2: Food you like! (0)	
Competences esperadas: 2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la novena unidad, los alumnos habiendo identificado la forma de expresar pedidos y hacer ofrecimientos, los utilizan en situaciones varias. Expresar situaciones y estados relacionados con cantidades. Explica y aplica vocabulario de comidas y bebidas. 	<ul style="list-style-type: none"> • Sustantivos Contables y No Contables • Expresiones con Would like y I'd like • Cuantificadores • Comidas alrededor del mundo • Pedidos formales • Cartas formales
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 3: The world of work (0)	
Competences esperadas: 2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la décima unidad, los alumnos habiendo reconocido las características de los adjetivos, utilizan éstos para hacer comparaciones de diversos tipos. Describen personas y lugares y dan indicaciones de dirección. Utilizarán conjunciones para unir ideas tipo. 	<ul style="list-style-type: none"> • Adjetivos • Oraciones con Adjetivos Comparativos. • Oraciones con Adjetivos Superlativos • Ciudades y el campo • Indicaciones de dirección
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 4: Looking good! (0)	
Competences esperadas: 2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la décimo primera unidad, los alumnos habiendo identificado la idea de expresar ideas de acciones que suceden en el momento o que se relacionan a cualquier tiempo estructuran oraciones en Presente Progresivo. Expresan ideas de posesión con respecto a la ropa y los colores. 	<ul style="list-style-type: none"> • Presente Continuo • Oraciones Afirmativas, Negativas y Preguntas • Uso de Whose • Pronombres Posesivos • Ropa y colores • Expresiones a usar en tiendas de ropa • Símbolos fonéticos.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 5: Life is an adventure! (0)	
Competences esperadas: 2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la décimo segunda unidad, los alumnos, a partir de la comprensión del tiempo futuro, elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar infinitivos de propósito. Adquirirán vocabulario para describir el clima. Se presentará expresiones para hacer y pedir sugerencias. 	<ul style="list-style-type: none"> • Uso de going to • Oraciones en Tiempo Futuro • Expresiones de Cantidad. • Verbos de acción • Vocabulario del clima • Expresiones de Sugerencia • Escribir una postal
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 6: You're pretty smart! (0)	
Competences esperadas: 2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la décimo tercera unidad, los alumnos habiendo conocido los fundamentos de la estructuración de preguntas diversas, realizarán trabajos aplicativos en contextos adecuados. Enfatizan la diferencia entre adjetivos y adverbios. Describen sentimientos. Utilizan expresiones para coger un tren. Asumen la idea se sufijos y prefijos. 	<ul style="list-style-type: none"> • Formas de Preguntas • Adverbios y Adjetivos • Vocabulario descripción de sentimientos • Expresiones para viajes en tren • Redacción de historias cortas • Lecturas.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 7: Have you ever? (0)	
Competences esperadas: 2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la décimo cuarta unidad, los alumnos habiendo conocido los fundamentos de la estructuración del Presente Perfecto experimentan la necesidad de poder expresar este tipo de tiempo en acciones. Realizarán prácticas en contextos adecuados. Enfatizan la diferencia entre pasado simple y presente perfecto. Describen acciones con never, ever y yet. Utilizan expresiones para utilizar en un aeropuerto. 	<ul style="list-style-type: none"> • Presente Perfecto • Expresiones con never, ever y yet • Vocabulario verbos en Participio pasado • Expresiones para viajes en avión • Redacción de cartas de agradecimiento • Lecturas
Lecturas : [SJ02], [Cam06], [Mac99]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** EG0006. Matemática III (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 4 HT; 1 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** EG0005. Matemática II. (2^{do} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Evaluar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C20.** Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome j**
- C24.** Comprender la necesidad de la formación permanente y la mejora de habilidades y capacidades.⇒ **Outcome j**

10. Lista de temas a estudiar en el curso

1. Introducción
2. Álgebra lineal
3. Métodos Numéricos

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Introducción (18)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Ser capaz de entender los conceptos básicos y la importancia de Álgebra Lineal y Métodos Numéricos.	<ul style="list-style-type: none">• Importancia del álgebra lineal y métodos numéricos. Ejemplos.
Lecturas : [AR14], [CC15]	

Unidad 2: Álgebra lineal (14)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Comprender los conceptos básicos del Álgebra Lineal. • Resolver problemas de transformaciones lineales. 	<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.
Lecturas : [AR14], [CC15]	

Unidad 3: Métodos Numéricos (22)	
Competencias esperadas: C24	
Objetivos de Aprendizaje	Tópicos
<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. 	<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.
Lecturas : [AR14], [CC15]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2100. Algoritmos y Estructuras de Datos (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS1103. Programación Orientada a Objetos II. (3^{er} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome b**
- C5.** Capacidad para implementar algoritmos y estructuras de datos en el software.⇒ **Outcome c**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome b**

10. Lista de temas a estudiar en el curso

1. Grafos
2. Matrices Esparzas
3. Arboles Equilibrados

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Grafos (12)	
Competencias esperadas: C1,C2,C5	
Objetivos de Aprendizaje	Tópicos
<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]	<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.
Lecturas : [Cor+09], [Fag+14]	

Unidad 2: Matrices Esparzas (8)	
Competencias esperadas: C1,C2,C5	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Comprender el uso y implementación de matrices esparzas.[Evaluar] 	<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
Lecturas : [Cor+09], [Fag+14]	

Unidad 3: Árboles Equilibrados (16)	
Competencias esperadas: C2,C5,C6	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Cor+09], [Fag+14]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2101. Teoría de la Computación (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Bro93] J. Glenn Brookshear. *Teoría de la Computación*. Addison Wesley Iberoamericana, 1993.
- [HU93] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. CECSA, 1993.
- [Kel95] Dean Kelley. *Teoría de Autómatas y Lenguajes Formales*. Prentice Hall, 1995.
- [Kol97] Ross Kolman Busby. *Estructuras de Matemáticas Discretas para la Computación*. Prentice Hall, 1997.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS1D02. Estructuras Discretas II. (2^{do} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

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

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Evaluar**)

9. Competencias (IEEE)

- C8. Entendimiento de lo que las tecnologías actuales pueden y no pueden lograr. ⇒ **Outcome a**
- C9. Comprensión de las limitaciones de la computación, incluyendo la diferencia entre lo que la computación es inherentemente incapaz de hacer frente a lo que puede lograrse a través de un futuro de ciencia y tecnología. ⇒ **Outcome b,j**

10. Lista de temas a estudiar en el curso

1. Computabilidad y complejidad básica de autómatas

2. Complejidad Computacional Avanzada
3. Teoría y Computabilidad Avanzada de Autómatas

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Computabilidad y complejidad básica de autómatas (20)	
Competencias esperadas: C9	
Objetivos de Aprendizaje	Tópicos
<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] • Explique la tesis de Church-Turing y su importancia [Familiarizarse] • Explique 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] 	<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.
Lecturas : [Kol97], [Kel95]	

Unidad 2: Complejidad Computacional Avanzada (20)	
Competencias esperadas: C8,C9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Kel95], [HU93]	

Unidad 3: Teoría y Computabilidad Avanzada de Autómatas (20)	
Competencias esperadas: C8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [HU93], [Bro93]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2701. Bases de Datos I (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.
- [Dat05] C.J. Date. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Elsevier, 2005.
- [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.
- [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.
- [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.

6. Información del curso

- (a) **Breve descripción del 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.

(b) **Prerrequisitos:**

- CS1102. Programación Orientada a Objetos I. (2^{do} Sem)
- CS1D02. Estructuras Discretas II. (2^{do} Sem)

(c) **Tipo de Curso:** Obligatorio

(d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Evaluar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome b**
- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome d**
- C7.** Ser capaz de aplicar los principios y tecnologías de ingeniería de software para asegurar que las implementaciones de software son robustos, fiables y apropiados para su público objetivo.⇒ **Outcome i**
- CS4.** Implementar la teoría apropiada, prácticas y herramientas para la especificación, diseño, implementación y mantenimiento, así como la evaluación de los sistemas basados en computadoras.⇒ **Outcome i**
- CS5.** Especificar, diseñar e implementar sistemas basados en computadoras.⇒ **Outcome j**

10. Lista de temas a estudiar en el curso

1. Sistemas de Bases de Datos
2. Modelado de datos
3. Indexación
4. Bases de Datos Relacionales
5. Lenguajes de Consulta

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Sistemas de Bases de Datos (14)	
Competencias esperadas: C1,C7,CS4,CS5	
Objetivos de Aprendizaje	Tópicos
<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] 	<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).
Lecturas : [RC04], [EN04], [Dat05], [KS02]	

Unidad 2: Modelado de datos (14)	
Competencias esperadas: C1,C2,C7,CS4,CS5	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [SW04], [EN04], [KS02]	

Unidad 3: Indexación (4)	
Competencias esperadas: CS4,CS5	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [WM01], [Dat05], [KS02]	

Unidad 4: Bases de Datos Relacionales (14)	
Competencias esperadas: 5	
Objetivos de Aprendizaje	Tópicos
<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] • Evalúa 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] 	<ul style="list-style-type: none"> • Mapeo de esquemas conceptuales a esquemas relacionales. • Entidad y integridad referencial. • Álgebra relacional y cálculo 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.
Lecturas : [WM01], [Dat05], [KS02]	

Unidad 5: Lenguajes de Consulta (12)	
Competencias esperadas: C1,CS4,CS5	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Die01], [EN04], [Cel05], [KS02]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** IN0054. Estadística y Probabilidades (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 3 HT; 2 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** EG0003. Matemática I. (1^{er} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

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

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Evaluar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome j**

10. Lista de temas a estudiar en el curso

1. Tipo de variable
2. Estadísticas descriptiva
3. Estadística inferencial

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Tipo de variable (6)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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.	<ul style="list-style-type: none">• Tipo de variable: Continua, discreta.
Lecturas : [MRo14], [Men14]	

Unidad 2: Estadísticas descriptiva (6)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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.	<ul style="list-style-type: none">• Tendencia Central (Media, mediana, modo)• Dispersión (Rango, desviación estándar, cuartil)• Gráficos: histograma, boxplot, etc. ∴ Capacidad de comunicación.
Lecturas : [MRo14], [Men14]	

Unidad 3: Estadística inferencial (6)	
Competences esperadas: CS2	
Objetivos de Aprendizaje	Tópicos
<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. 	<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.
Lecturas : [MRo14], [Men14]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** GH2101. Inglés III (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 10 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.
[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.
[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

6. Información del curso

- (a) **Breve descripción del curso** Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde este no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.
- (b) **Prerrequisitos:** GH1102. Inglés II. (3^{er} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Formar en el alumno de capacidad de comprender y retener una conversación.
- Brindar técnicas de ilación de ideas.

8. Contribución a los resultados (*Outcomes*)

- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

9. Competencias (IEEE)

C25. Capacidad para comunicarse en un segundo idioma.⇒ **Outcome f**

10. Lista de temas a estudiar en el curso

1. Getting to know you!
2. The way we live!
3. It all went wrong!
4. Lets go shopping!
5. What do you want to do?

6. The best in the world!

7. Fame!

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Getting to know you! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">Al terminar la primera unidad, cada uno de los alumnos, comprendiendo la gramática de los tiempos presente, pasado y futuro es capaz de expresar una mayor cantidad de acciones en forma de oraciones. Además es capaz de expresar ideas en forma de preguntas. Asume la idea de palabras con más de un significado. Utiliza expresiones sociales en situaciones de entretenimiento.	<ul style="list-style-type: none">Tiempos Presente, Pasado y Futuro.Oraciones Interrogativas con Wh-.Palabras con más de un significado.Partes de la oración.Expresiones para tiempo libre.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 2: The way we live! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">Al terminar la segunda unidad, los alumnos habiendo identificado la forma de expresar presente reconocen la diferencia entre las formas del mismo y las aplican adecuadamente. Describen países acuciosamente. Asumen expresiones para demostrar interés. Utilizan conectores para unir ideas varias.	<ul style="list-style-type: none">Tiempo Presente Simple.Tiempo Presente Continuo.Colocaciones.Vocabulario de países del mundo.Expresiones de enojo.Conectores.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 3: It all went wrong! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> Al terminar la tercera unidad, los alumnos habiendo reconocido las características de los tiempos en pasado los utilizan adecuadamente. Utilizan prefijos y sufijos para crear y reconocer nuevas palabras. Describen tiempo en forma amplia. Utilizarán conjunciones para unir ideas tipo. 	<ul style="list-style-type: none"> Tiempo Pasado Simple. Tiempo Pasado Continuo. Verbos Irregulares. Expresiones de Tiempo. Conectores de tiempo.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 4: Lets go shopping! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> Al terminar la cuarta primera unidad, los alumnos habiendo identificado la idea de cantidad expresan diversas situaciones que la involucran. Reconocen y aplican artículos a sustantivos. Asumen la idea de ir de compras con la ayuda de expresiones. Expresan precios e ideas de dinero. Llenan formatos varios. Expresan actitudes. 	<ul style="list-style-type: none"> Expresiones de Cantidad Indefinida Oraciones Afirmativas, Negativas y Preguntas Uso de Artículos Precios de productos Llenado de formatos y encuestas Expresiones para ir de compras
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 5: What do you want to do? (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> Al finalizar la quinta unidad, los alumnos, a partir de la comprensión de la idea de patrones verbales elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar intenciones futuras. Adquirirán vocabulario para describir sentimientos. Se presentará expresiones para describir planes y ambiciones. 	<ul style="list-style-type: none"> Patrones Verbales I. Intenciones Futuras. Verbos de Percepción. Vocabulario de sentimientos. Expresiones de Planes y Ambiciones.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 6: The best in the world! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la sexta unidad, los alumnos habiendo conocido los fundamentos del uso de adjetivos, estructuran oraciones con diversas formas de los mismos en contextos adecuados. Enfatizan la diferencia entre tipos de ciudades y pueblos y estilos de vida. Utilizan expresiones indicación de direcciones. 	<ul style="list-style-type: none"> • Whats it like?. • Adjetivos. • Comparativos y Superlativos. • Sinónimos y Antónimos. • Indicaciones de Dirección. • Lecturas.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 7: Fame! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la séptima unidad, los alumnos habiendo conocido los fundamentos de la estructuración del tiempo presente perfecto y lo diferencian del pasado simple. Enfatizan la diferencia entre formas de adjetivos. Describen ideas de la música. Utilizan expresiones para dar respuestas cortas. Asumen la idea de dar explicaciones extra de los elementos de una oración. 	<ul style="list-style-type: none"> • Presente Perfecto y Pasado Simple • Expresiones for, ever, since • Adverbios • Expresiones que vienen en pares • Respuestas cortas • Celebridades
Lecturas : [SJ02], [Cam06], [Mac99]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2102. Análisis y Diseño de Algoritmos (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.

- (b) **Prerrequisitos:** CS2100. Algoritmos y Estructuras de Datos. (4^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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?

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome b**
- C3.** Una comprensión intelectual de, y el aprecio por el papel central de los algoritmos y estructuras de datos.⇒ **Outcome b**
- C5.** Capacidad para implementar algoritmos y estructuras de datos en el software.⇒ **Outcome a**
- C6.** Capacidad para diseñar y poner en práctica las unidades estructurales mayores que utilizan algoritmos y estructuras de datos y las interfaces a través del cual estas unidades se comunican.⇒ **Outcome a**
- C9.** Comprensión de las limitaciones de la computación, incluyendo la diferencia entre lo que la computación es inherentemente incapaz de hacer frente a lo que puede lograrse a través de un futuro de ciencia y tecnología.⇒ **Outcome a**

10. Lista de temas a estudiar en el curso

1. Análisis Básico
2. Estrategias Algorítmicas
3. Algoritmos y Estructuras de Datos fundamentales
4. Computabilidad y complejidad básica de autómatas
5. Estructuras de Datos Avanzadas y Análisis de Algoritmos

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:**12. Contenido**

Unidad 1: Análisis Básico (10)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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 simples 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] 	<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. • Algunas versiones del Teorema Maestro.
Lecturas : [KT05], [DPV06], [RS09], [SF13], [Knu97]	

Unidad 2: Estrategias Algorítmicas (30)	
Competencias esperadas: C2	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Algoritmos de fuerza bruta. • Algoritmos voraces. • Divide y vencerás. • Programación Dinámica.
Lecturas : [KT05], [DPV06], [RS09], [Als99]	

Unidad 3: Algoritmos y Estructuras de Datos fundamentales (10)	
Competencias esperadas: C6	
Objetivos de Aprendizaje	Tópicos
<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 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 árbol de expansión mínima [Evaluar] 	<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"> – Algoritmos de la ruta más corta (algoritmos de Dijkstra y Floyd) – Árbol de expansión mínima (algoritmos de Prim y Kruskal)
Lecturas : [KT05], [DPV06], [RS09], [SW11], [GT09]	

Unidad 4: Computabilidad y complejidad básica de autómatas (2)	
Competencias esperadas: C9	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Define las clases P y NP [Familiarizarse] • Explique el significado de NP-Complejidad [Familiarizarse] 	<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-Complejos y a clases NP-Complejos.
Lecturas : [KT05], [DPV06], [RS09]	

Unidad 5: Estructuras de Datos Avanzadas y Análisis de Algoritmos (8)	
Competencias esperadas: C16	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [KT05], [DPV06], [RS09], [Tar83], [Raw92]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2702. Bases de Datos II (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 1 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Bur04] Donald K. Bursleson. *Physical Database Design Using Oracle*. CRC Press, 2004.
- [Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.
- [Dat05] C.J. Date. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. 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.

6. Información del curso

- (a) **Breve descripción del 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.

- (b) **Prerrequisitos:** CS2701. Bases de Datos I. (4^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)

j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Usar**)

k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. (**Usar**)

9. Competencias (IEEE)

C1. La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome b**

C7. Ser capaz de aplicar los principios y tecnologías de ingeniería de software para asegurar que las implementaciones de software son robustos, fiables y apropiados para su público objetivo.⇒ **Outcome j**

CS4. Implementar la teoría apropiada, prácticas y herramientas para la especificación, diseño, implementación y mantenimiento, así como la evaluación de los sistemas basados en computadoras.⇒ **Outcome k**

10. Lista de temas a estudiar en el curso

1. Diseño Físico de Bases de Datos
2. Procesamiento de Transacciones
3. Almacenamiento y Recuperación de Información
4. Bases de Datos Distribuidas

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Diseño Físico de Bases de Datos (10)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Bur04], [Dat05], [Cel05]	

Unidad 2: Procesamiento de Transacciones (12)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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 porqué 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] • Elejir 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] 	<ul style="list-style-type: none"> • Transacciones. • Fallo y recuperación. • Control concurrente. • Interacción de gestión de transacciones con el almacenamiento, especialmente en almacenamiento.
Lecturas : [Phi97], [Ram04]	

Unidad 3: Almacenamiento y Recuperación de Información (10)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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, citas. • 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).
Lecturas : [Pet98], [Ram04]	

Unidad 4: Bases de Datos Distribuidas (36)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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 minimise 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] 	<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
Lecturas : [M T99], [Dat05]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2901. Ingeniería de Software I (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Lar08] Craig Larman. *Applying UML and Patterns*. Prentice Hall, 2008.
- [Pre05] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. 6th. McGraw-Hill, Mar. 2005.
- [Som08] Ian Sommerville. *Software Engineering*. 7th. ISBN: 0321210263. Addison Wesley, May 2008.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:**
- CS1103. Programación Orientada a Objetos II. (3^{er} Sem)
 - CS2701. Bases de Datos I. (4^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)
- k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. (**Usar**)

9. Competencias (IEEE)

C7. Ser capaz de aplicar los principios y tecnologías de ingeniería de software para asegurar que las implementaciones de software son robustos, fiables y apropiados para su público objetivo.⇒ **Outcome i**

C12. Entender las implicaciones de ciclo de vida para el desarrollo de todos los aspectos de los sistemas informáticos (incluyendo software, hardware, y la interfaz de la computadora humana).⇒ **Outcome i**

CS1. Modelar y diseñar sistemas de computadora de una manera que se demuestre comprensión del balance entre las opciones de diseño.⇒ **Outcome k**

CS2. Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome i**

CS4. Implementar la teoría apropiada, prácticas y herramientas para la especificación, diseño, implementación y mantenimiento, así como la evaluación de los sistemas basados en computadoras.⇒ **Outcome i,k**

CS10. Implementar efectivamente las herramientas que se utilizan para la construcción y la documentación de software, con especial énfasis en la comprensión de todo el proceso involucrado en el uso de computadoras para resolver problemas prácticos. Esto debe incluir herramientas para el control de software, incluyendo el control de versiones y gestión de la configuración.⇒ **Outcome i,k**

10. Lista de temas a estudiar en el curso

1. Ingeniería de Requisitos
2. Diseño de Software
3. Construcción de Software

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Ingeniería de Requisitos (18)	
Competencias esperadas: C7, C11, CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<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 elicitation. • 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.
Lecturas : [Pre05], [Som08], [Lar08]	

Unidad 2: Diseño de Software (18)	
Competencias esperadas: C5, C7, C8, CS10	
Objetivos de Aprendizaje	Tópicos
<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 	<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: seleccion de componentes, diseño, adaptación y componentes de ensamblaje, componentes y patrones, componentes y objetos (por ejemplo, construir una GUI usando un standar 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

Unidad 3: Construcción de Software (24)	
Competencias esperadas: C4, C5, C7, C8, CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Prácticas de codificación: técnicas, idiomias/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
Lecturas : [Pre05], [Som08], [Lar08]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2S01. Sistemas Operativos (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Avi12] Greg Gagne Avi Silberschatz Peter Baer Galvin. *Operating System Concepts, 9/E*. John Wiley & Sons, Inc., 2012. ISBN: 978-1-118-06333-0.
- [Mat99] Luis Mateu. *Apuntes de Sistemas Operativos*. Universidad de Chile, 1999.
- [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, 2/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.

6. Información del curso

- (a) **Breve descripción del curso** Un Sistema Operativo es un programa que actúa como intermediario entre el usuario y la máquina.

El propósito de un sistema operativo es proveer un ambiente en que el usuario pueda ejecutar sus aplicaciones.

En este curso se estudiará el diseño del núcleo de los sistemas operativos. Además el curso contempla actividades prácticas en donde se resolverán problemas de concurrencia y se modificará el funcionamiento de un pseudo Sistema Operativo.

- (b) **Prerrequisitos:** CS2201. Arquitectura de Computadores. (3^{er} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Conocer los elementos básicos del diseño de los sistemas operativos.

8. Contribución a los resultados (*Outcomes*)

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome b**
- C6.** Capacidad para diseñar y poner en práctica las unidades estructurales mayores que utilizan algoritmos y estructuras de datos y las interfaces a través del cual estas unidades se comunican.⇒ **Outcome b**

CS8. Aplicar los principios de la interacción persona-ordenador para la evaluación y la construcción de una amplia gama de materiales, incluyendo interfaces de usuario, páginas web, sistemas multimedia y sistemas móviles. ⇒ **Outcome b**

10. Lista de temas a estudiar en el curso

1. Visión general de Sistemas Operativos
2. Principios de Sistemas Operativos
3. Concurrencia
4. Planificación y despacho
5. Manejo de memoria
6. Seguridad y protección
7. Máquinas virtuales
8. Manejo de dispositivos
9. Sistema de archivos
10. Sistemas empujados y de tiempo real
11. Tolerancia a fallas
12. Evaluación del desempeño de sistemas

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Visión general de Sistemas Operativos (3)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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, dispositivos de mano. • Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad) • Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 2: Principios de Sistemas Operativos (6)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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 interrumpido [Familiarizarse] • Explicar el uso de una lista de dispositivos y el controlador de colas de entrada y salida [Familiarizarse] 	<ul style="list-style-type: none"> • Métodos de estructuración (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 usuario de estado / sistema y la protección, la transición al modo kernel.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 3: Concurrency (9)	
Competences expected: C6	
Objectives of Learning	Topics
<ul style="list-style-type: none"> • Describe the need for concurrency in the framework of an operating system [Familiarizarse] • Demonstrate the potential problems of time of execution derived from the simultaneous operation of many different tasks [Usar] • Summarize the range of mechanisms that can be used at the level of the operating system to perform concurrent systems and describe the benefits of each one [Familiarizarse] • Explain the different states by which a task must pass and the data structures necessary for the management of various tasks [Familiarizarse] • Summarize the techniques for synchronization in an operating system (for example, describe how to implement semaphores using primitives of the operating system.) [Familiarizarse] • Describe the reasons for using interruptors, dispatching, and context switching to support concurrency in an operating system [Familiarizarse] • Create state and transition diagrams for simple problem domains [Usar] 	<ul style="list-style-type: none"> • State diagrams. • Structures (prepared list, control blocks of processes, and so on successively) • Dispatching and context switching. • The role of interrupts. • Manage access to objects of the operating system atomically. • The implementation of synchronization primitives. • Multiprocessor questions (spin-locks, reentry)
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 4: Planificación y despacho (6)	
Competencias esperadas: CS8	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Comparar y contrastar los algoritmos comunes que se utilizan tanto para un programa preferente y no preferente de las 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 planificación y dominios de aplicación [Familiarizarse] • Discutir los tipos de planeamiento de procesos <i>scheduling</i> de corto, a mediano, a largo plazo y I/O [Familiarizarse] • Describir las diferencias entre procesos y hebras [Familiarizarse] • Comparar y contrastar enfoques estáticos y dinámicos para <i>scheduling</i> en tiempo real [Evaluar] • Hablar sobre la necesidad de tiempos límites de <i>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] 	<ul style="list-style-type: none"> • Planificación preventiva y no preferente. • Planificadores y políticas. • Procesos y subprocesos. • Plazos y cuestiones en tiempo real.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 5: Manejo de memoria (6)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar la jerarquía de la memoria y costo-rendimiento de intercambio [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] • Defiende 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] 	<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
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 6: Seguridad y protección (6)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Articular 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] 	<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.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 7: Máquinas virtuales (6)	
Competencias esperadas: CS8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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 Hypervisor. • Virtualización portátil; emulación vs aislamiento. • Costo de la virtualización.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 8: Manejo de dispositivos (6)	
Competencias esperadas: C6	
Objetivos de Aprendizaje	Tópicos
<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] • Explique <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] 	<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.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 9: Sistema de archivos (6)	
Competencias esperadas: CS8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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 sistemas de archivos / desmontar, 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)
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 10: Sistemas empujados y de tiempo real (6)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 11: Tolerancia a fallas (3)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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, reinicie para implementar la tolerancia a fallos, el uso de estas técnicas para los servicios propios del sistema operativo.
Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]	

Unidad 12: Evaluación del desempeño de sistemas (3)**Competencias esperadas: C1****Objetivos de Aprendizaje**

- 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]

Tópicos

- ¿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)

Lecturas : [Avi12], [Sta05], [Tan06], [Tan01], [Mat99]



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3402. Compiladores (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Aho+08] Alfred Aho et al. *Compiladores. Principios, técnicas y herramientas*. 2nd. ISBN:10-970-26-1133-4. Addison Wesley, 2008.
- [Aho90] Alfred Aho. *Compiladores Principios, técnicas y herramientas*. Addison Wesley, 1990.
- [ALe96] Karen A.Lemone. *Fundamentos de Compiladores*. CECSA-Mexico, 1996.
- [App02] A. W. Appel. *Modern compiler implementation in Java*. 2.a edición. Cambridge University Press, 2002.
- [Lou04a] Kenneth C. Louden. *Construcción de Compiladores Principios y Practica*. Thomson, 2004.
- [Lou04b] Kenneth C. Louden. *Lenguajes de Programacion*. Thomson, 2004.
- [PV98] Terrence W. Pratt and Marvin V.Zelkowitz. *Lenguajes de Programacion Diseño e Implementacion*. Prentice-Hall Hispanoamericana S.A., 1998.
- [TS98] Bernard Teufel and Stephanie Schmidt. *Fundamentos de Compiladores*. Addison Wesley Iberoamericana, 1998.

6. Información del curso

- (a) **Breve descripción del curso** Que el alumno conozca y comprenda los conceptos y principios fundamentales de la teoría de compilación para realizar la construcción de un compilador
- (b) **Prerrequisitos:** CS2101. Teoría de la Computación. (4^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

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

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Evaluar**)

9. Competencias (IEEE)

C8. Entendimiento de lo que las tecnologías actuales pueden y no pueden lograr. ⇒ **Outcome a**

C9. Comprensión de las limitaciones de la computación, incluyendo la diferencia entre lo que la computación es inherentemente incapaz de hacer frente a lo que puede lograrse a través de un futuro de ciencia y tecnología. ⇒ **Outcome b,j**

10. Lista de temas a estudiar en el curso

1. Representación de programas
2. Traducción y ejecución de lenguajes
3. Análisis de sintaxis
4. Análisis semántico de compiladores
5. Generación de código

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Representación de programas (5)	
Competencias esperadas: C9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Lou04b], [PV98]	

Unidad 2: Traducción y ejecución de lenguajes (10)	
Competencias esperadas: C8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Aho+08], [Aho90], [Lou04a], [TS98], [ALe96], [App02]	

Unidad 3: Análisis de sintaxis (10)	
Competencias esperadas: C8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Aho+08], [Aho90], [Lou04a], [TS98], [ALe96], [App02]	

Unidad 4: Análisis semántico de compiladores (15)	
Competencias esperadas: C8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Aho+08], [Aho90], [Lou04a], [TS98], [ALe96], [App02]	

Unidad 5: Generación de código (20)	
Competencias esperadas: C8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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 (<i>peephole</i>)
Lecturas : [Aho+08], [Aho90], [Lou04a], [TS98], [ALe96], [App02]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** GH0010. Ética y Tecnología (Obligatorio)
2. **Créditos:** 2
3. **Horas de Teoría y Laboratorio:** 1 HT; 2 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[Alo06] García. Alonso. *Ética o Filosofía moral*. México, Editorial Trillas, 2006.

[Mar05] Alvarado. Martín. *Ética*. México, Editorial Trillas, 2005.

6. Información del curso

- (a) **Breve descripción del curso** Este curso busca proporcionar a los y las estudiantes ciertos marcos referenciales con los cuales analizar las disyuntivas que se pueden presentar en su ejercicio profesional. El curso pone en práctica constante el razonamiento crítico y responsable de los y las estudiantes, siendo esta una competencia fundamental para los procesos de toma de decisión que asumiremos como profesionales y ciudadanos.
- (b) **Prerrequisitos:** GH0006. Laboratorio de Comunicación II. (2^{do} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Introducir a los estudiantes al pensamiento crítico y ético aplicado a su campo profesional.
- Desarrollar la competencia de mirar un fenómeno desde varias disciplinas y perspectivas genera en la persona empatía y respeto a la diversidad de opinión.
- Capacidad de trabajo en equipo.
- Capacidad para identificar problemas.
- Capacidad de comunicación oral.
- Tiene interés por conocer sobre temas actuales de la sociedad peruana y del mundo.
- Capacidad de comunicación escrita.

8. Contribución a los resultados (Outcomes)

- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- e) Entender correctamente las implicancias profesionales, éticas, legales, de seguridad y sociales de la profesión. (**Usar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)
- n) Aplicar conocimientos de humanidades en su labor profesional. (**Usar**)
- o) Mejorar las condiciones de la sociedad poniendo la tecnología al servicio del ser humano. (**Usar**)

9. Competencias (IEEE)

- C10.** Comprensión del impacto en las personas, las organizaciones y la sociedad de la implementación de soluciones tecnológicas e intervenciones.⇒ **Outcome d,n,o**
- C17.** Capacidad para expresarse en los medios de comunicación orales y escritos como se espera de un graduado.⇒ **Outcome f**
- C18.** Capacidad para participar de forma activa y coordinada en un equipo.⇒ **Outcome f**
- C21.** Comprender el aspecto profesional, legal, seguridad, asuntos políticos, humanistas, ambientales, culturales y éticos.⇒ **Outcome e**

10. Lista de temas a estudiar en el curso

1. Ética, ciencia y tecnología.
2. Responsabilidad en la ciencia e ingeniería
3. Ciudadanía y ejercicio de la justicia en la era digital

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Ética, ciencia y tecnología. (12)	
Competences esperadas: C10	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Fortalecer en el estudiante la capacidad de pensar interdisciplinariamente..	<ul style="list-style-type: none">• Definición y alcance de la ética Pensamiento crítico / argumentación ética.• Ciencia y Tecnología , ¿Son las ingenierías y la tecnología cuestiones objetivas?• Tecnología: concepto y límites.• Importancia de la ética en las ciencias e ingeniería .
Lecturas : [Alo06]	

Unidad 2: Responsabilidad en la ciencia e ingeniería (24)	
Competences esperadas: C17,C21	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Comprender las responsabilidades profesionales y éticas. 	<ul style="list-style-type: none"> • Alcance del concepto Responsabilidad en la ciencia (Imperative of Responsibility) • Introducción al tema Responsabilidad / libertad
Lecturas : [Mar05]	

Unidad 3: Ciudadanía y ejercicio de la justicia en la era digital (30)	
Competences esperadas: C17,C21	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Comprende el impacto de las soluciones de la ingeniería en un contexto global, económico, ambiental y de la sociedad. 	<ul style="list-style-type: none"> • Introducción al tema de ciudadanía en la era digital • Tecnología, nuevos activismos y ciudadanía
Lecturas : [Mar05]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** ID104. Inglés IV (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 10 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.
[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.
[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

6. Información del curso

- (a) **Breve descripción del curso** Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde este no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.
- (b) **Prerrequisitos:** GH2101. Inglés III. (4^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Incrementar el nivel de conversación en diferentes temas, en los alumnos. Así como la capacidad de escribir y leer documentación de todo tipo.
- Llevar al alumno a una expresión más intensa en el dominio del idioma.

8. Contribución a los resultados (*Outcomes*)

- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

9. Competencias (IEEE)

C25. Capacidad para comunicarse en un segundo idioma. ⇒ **Outcome f**

10. Lista de temas a estudiar en el curso

1. Do and don't!
2. Going places!
3. Scared to death!
4. Things that changed the world!

5. Dreams and reality!
6. Making a living!
7. All you need is love!

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Do and don't! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la octava unidad, cada una de las alumnas, comprendiendo la gramática de los auxiliares should y must es capaz de expresar una mayor cantidad de acciones en forma obligación y sugerencia. Además es capaz de expresar ideas describiendo ocupaciones. Asume la necesidad de escribir cartas formales. 	<ul style="list-style-type: none"> • Auxiliares de Modo should, must y have got to. • Oraciones afirmativas, negativas e interrogativas con modals. • Términos para cartas formales. • Partes de las respuestas cortas. • Expresiones para ocupaciones.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 2: Going places! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la novena unidad, los alumnos habiendo identificado la forma de expresar presente reconocen la diferencia entre las formas de futuro y las aplican adecuadamente. Describen condiciones acuciosamente. Asumen expresiones para demostrar ubicación de lugar. Utilizan expresiones de tiempo y conectores para unir ideas varias. 	<ul style="list-style-type: none"> • Tiempo Presente Simple y Futuro con Will • Primer Condicional • Colocaciones • Vocabulario de preposiciones de lugar y de tiempo • Expresiones de conexión de ideas
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 3: Scared to death! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la décimo unidad los alumnos serán capaces de reconocer y utilizar patrones tiempos en pasado los utilizan adecuadamente. Utilizan expresiones de exclamación. Y describen sentimientos. Utilizarán conjunciones para unir ideas tipo. 	<ul style="list-style-type: none"> • Patrones de Verbos Infinitivos y gerundios • What + Infinitivo • Something + infinitive • Expresiones de sentimientos • Exclamaciones de sorpresa
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 4: Things that changed the world! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al terminar la décimo primera primera unidad, los alumnos habiendo identificado la idea de acciones pasivas describen acciones adecuadamente en diversas situaciones que la involucran. Reconocen y aplican participios. Asumen la idea de respetar signos y señales públicas. Expresan ideas de hábitos . Hacen resúmenes. 	<ul style="list-style-type: none"> • Voz Pasiva • Oraciones Afirmativas, Negativas y Preguntas • Uso de Participios, verbos y sustantivos que van unidos • Señales. Signos y notas • Resúmenes • Expresiones para indicar prohibición
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 5: Dreams and reality! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la décimo segunda unidad, los alumnos, a partir de la comprensión de la idea de Condicionales y de expresar posibilidad elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de frases verbales (verbos de 2 palabras). Adquirirán vocabulario para describir expresiones sociales. 	<ul style="list-style-type: none"> • Segundo Condicional • Auxiliar de Modo Might • Verbos de Frase • Vocabulario de expresiones sociales • Adverbios • Expresiones para dar consejo
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 6: Making a living! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la décimo tercera unidad estructuran oraciones con acciones que incluyen presente y pasado en contextos adecuados. Enfatizan la diferencia entre tipos de ocupaciones. Utilizan expresiones adecuadas para conversaciones telefónicas 	<ul style="list-style-type: none"> • Present Perfect Continuous • Present Continuous • Ocupaciones • Formación de palabras • Adverbios • Expresiones de uso en el teléfono.
Lecturas : [SJ02], [Cam06], [Mac99]	

Unidad 7: All you need is love! (0)	
Competences esperadas: C25	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Al finalizar la décimo cuarta unidad, los alumnos habiendo conocido los fundamentos de la estructuración del tiempo pasado perfecto, lo diferencian del pasado simple. Enfatizan la diferencia entre palabras en contextos diferentes. Describen ideas de despedidas. Utilizan expresiones para escribir historias de amor. Asumen la idea de dar y hacer entrevistas. 	<ul style="list-style-type: none"> • Pasado Perfecto y Pasado Simple • Expresiones de Reporte • Expresiones de palabras en contextos diferentes • Despedidas cortas y formales • Historias de amor
Lecturas : [SJ02], [Cam06], [Mac99]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2301. Redes y Comunicaciones (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 1 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[KR13] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-down Approach*. Always learning. Pearson, 2013. ISBN: 9780132856201.

6. Información del curso

- (a) **Breve descripción del 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.

- (b) **Prerrequisitos:** CS2S01. Sistemas Operativos. (5^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Familiarizarse**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)

9. Competencias (IEEE)

- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome c**
- C6.** Capacidad para diseñar y poner en práctica las unidades estructurales mayores que utilizan algoritmos y estructuras de datos y las interfaces a través del cual estas unidades se comunican.⇒ **Outcome c,b**
- C7.** Ser capaz de aplicar los principios y tecnologías de ingeniería de software para asegurar que las implementaciones de software son robustos, fiables y apropiados para su público objetivo.⇒ **Outcome c**

CS2. Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome b**

CS5. Especificar, diseñar e implementar sistemas basados en computadoras.⇒ **Outcome c**

CS6. Evaluar los sistemas en términos de atributos de calidad en general y las posibles ventajas y desventajas que se presentan en el problema dado.⇒ **Outcome c**

CS12. Operar equipos de computación y software eficaz de dichos sistemas.⇒ **Outcome i**

10. Lista de temas a estudiar en el curso

1. Introducción
2. Aplicaciones en red
3. Entrega confiable de datos
4. Ruteo y reenvío
5. Redes de área local
6. Asignación de recursos
7. Celulares
8. Redes sociales

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Introducción (5)	
Competencias esperadas: C1,CS8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [KR13]	

Unidad 2: Aplicaciones en red (5)	
Competencias esperadas: CS2,CS5	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [KR13]	

Unidad 3: Entrega confiable de datos (10)	
Competencias esperadas: C6,CS2,CS5	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [KR13]	

Unidad 4: Ruteo y reenvío (12)	
Competencias esperadas: CS2,CS5	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Enrutamiento vs reenvío . • Enrutamiento estático . • Protocolo de Internet (IP) • Problemas de escalabilidad (direccionamiento jerárquico)
Lecturas : [KR13]	

Unidad 5: Redes de área local (10)	
Competencias esperadas: C1,C7	
Objetivos de Aprendizaje	Tópicos
<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 relaciones entre IP y Ethernet [Familiarizarse] • Describir las etapas usadas en un enfoque común para el problema de múltiples accesos [Familiarizarse] 	<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 .
Lecturas : [KR13]	

Unidad 6: Asignación de recursos (12)	
Competencias esperadas: C6,CS5,CS12	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [KR13]	

Unidad 7: Celulares (5)	
Competencias esperadas: C1,C7	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Describir la organización de una red inalámbrica [Familiarizarse] • Describir como las redes inalámbricas soportan usuarios móviles [Familiarizarse] 	<ul style="list-style-type: none"> • Principios de redes celulares. • Redes 802.11 • Problemas en el apoyo a los nodos móviles (agente local)
Lecturas : [KR13]	

Unidad 8: Redes sociales (5)	
Competencias esperadas: C1,CS2,CS8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [KR13]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3101. Programación Competitiva (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[Cor+09] T. H. Cormen et al. *Introduction to Algorithms*. MIT Press, 2009.

6. Información del curso

- (a) **Breve descripción del curso** La Programación Competitiva combina retos de solucionar problemas con la diversión de 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.
- (b) **Prerrequisitos:** CS2102. Análisis y Diseño de Algoritmos. (5^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a,b**
- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome a,b**
- C24.** Comprender la necesidad de la formación permanente y la mejora de habilidades y capacidades.⇒ **Outcome h**

10. Lista de temas a estudiar en el curso

1. Primera Unidad

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Primera Unidad (20)	
Competences esperadas: C24,C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Aprender a seleccionar los algoritmos adecuados para un problema dado, integrando múltiples algoritmos para la solución de un problema complejo. [Usar] • Diseñar nuevos algoritmos para la resolución de problemas del mundo real.[Usar] 	<ul style="list-style-type: none"> • Estructura de datos • Programación dinámica • Algoritmos basados en grafos • Geometría computacional • Algoritmos de ordenamiento
Lecturas : [Cor+09]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3102. Estructuras de Datos Avanzadas (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.
- [Gam+94] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Computing Series. ISBN-10: 0201633612. Addison-Wesley Professional, Nov. 1994.
- [GG98] Volker Gaede and Oliver Günther. "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, 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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS2102. Análisis y Diseño de Algoritmos. (5^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Familiarizarse**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Familiarizarse**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C20.** Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome b**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome c**

10. Lista de temas a estudiar en el curso

1. Técnicas Básicas de Implementación de Estructuras de Datos
2. Métodos de Acceso Multidimensionales
3. Métodos de Acceso Métrico
4. Métodos de Acceso Aproximados
5. Seminarios

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Técnicas Básicas de Implementación de Estructuras de Datos (16)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

Unidad 2: Métodos de Acceso Multidimensionales (16)	
Competencias esperadas: C20	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Sam06], [GG98]	

Unidad 3: Métodos de Acceso Métrico (20)	
Competencias esperadas: C24	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Métodos de Acceso Métrico para distancias discretas • Métodos de Acceso Métrico para distancias continuas
Lecturas : [Sam06], [Chá+01], [Tra+00], [Zez+07]	

Unidad 4: Métodos de Acceso Aproximados (20)	
Competences esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Space Filling Curves • Locality Sensitive Hashing
Lecturas : [PI06], [Zez+07], [Sam06]	

Unidad 5: Seminarios (8)	
Competences esperadas: C20	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Métodos de Acceso Espacio Temporal • Estructuras de Datos con programación genérica
Lecturas : [Sam06], [Chá+01]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3903. Sistemas de Infomación (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [PM14] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2014.
- [Som10] Ian Sommerville. *Software Engineering*. 9th. Addison-Wesley, Mar. 2010.

6. Información del curso

- (a) **Breve descripción del curso** Analizar técnicas para la correcta implementación de Sistemas de Información escalables, robustos, confiables y eficientes en las organizaciones.
- (b) **Prerrequisitos:** CS2901. Ingeniería de Software I. (5^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

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

8. Contribución a los resultados (*Outcomes*)

- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. (**Evaluar**)

9. Competencias (IEEE)

- C7.** Ser capaz de aplicar los principios y tecnologías de ingeniería de software para asegurar que las implementaciones de software son robustos, fiables y apropiados para su público objetivo.⇒ **Outcome c**
- C8.** Entendimiento de lo que las tecnologías actuales pueden y no pueden lograr.⇒ **Outcome c**
- C16.** Capacidad para identificar temas avanzados de computación y de la comprensión de las fronteras de la disciplina.⇒ **Outcome k**
- CS4.** Implementar la teoría apropiada, prácticas y herramientas para la especificación, diseño, implementación y mantenimiento, así como la evaluación de los sistemas basados en computadoras.⇒ **Outcome k**

CS6. Evaluar los sistemas en términos de atributos de calidad en general y las posibles ventajas y desventajas que se presentan en el problema dado.⇒ **Outcome i**

CS10. Implementar efectivamente las herramientas que se utilizan para la construcción y la documentación de software, con especial énfasis en la comprensión de todo el proceso involucrado en el uso de computadoras para resolver problemas prácticos. Esto debe incluir herramientas para el control de software, incluyendo el control de versiones y gestión de la configuración.⇒ **Outcome k**

10. Lista de temas a estudiar en el curso

1. Introducción
2. Estrategia
3. Implementación

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Introducción (15)	
Competences esperadas: C7,C8	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Aplicar correctamente la tecnología para la gestión de la información [Evaluar]	<ul style="list-style-type: none">• Introducción a la gestión de la información• Software para gestión de información.• Tecnología para gestión de información.
Lecturas : [Som10], [PM14]	

Unidad 2: Estrategia (15)	
Competences esperadas: C16, CS4	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Aplicar y evaluar correctamente estrategias de gestión [Evaluar]	<ul style="list-style-type: none">• Estrategia para gestión de información• Estrategia para gestión conocimiento• Estrategia para sistema de información.
Lecturas : [Som10], [PM14]	

Unidad 3: Implementación (15)	
Competencias esperadas: CS4, CS6, CS10	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Aplicar y evaluar correctamente estrategias de implementación [Evaluar] 	<ul style="list-style-type: none"> • Gestión de desarrollo de sistemas de información. • Gestión del cambio • Arquitectura de Información
Lecturas : [Som10], [PM14]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2H01. Interacción Humano Computador (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 1 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS3903. Sistemas de Información. (6^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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 tecnología emergente hace posible nuevos estilos de interacción.
- Determinar los requerimientos básicos a nivel de interfaces, hardware y software para la construcción de ambientes inmersivos.

8. Contribución a los resultados (*Outcomes*)

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Familiarizarse**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Evaluar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- o) Mejorar las condiciones de la sociedad poniendo la tecnología al servicio del ser humano. (**Familiarizarse**)

9. Competencias (IEEE)

- CS8.** Aplicar los principios de la interacción persona-ordenador para la evaluación y la construcción de una amplia gama de materiales, incluyendo interfaces de usuario, páginas web, sistemas multimedia y sistemas móviles.⇒ **Outcome b**
- C7.** Ser capaz de aplicar los principios y tecnologías de ingeniería de software para asegurar que las implementaciones de software son robustos, fiables y apropiados para su público objetivo.⇒ **Outcome c**
- C9.** Comprensión de las limitaciones de la computación, incluyendo la diferencia entre lo que la computación es inherentemente incapaz de hacer frente a lo que puede lograrse a través de un futuro de ciencia y tecnología.⇒ **Outcome o**
- C15.** Entendimiento del concepto esencial del proceso, ya que se relaciona con la actividad profesional sobre todo la relación entre la calidad del producto y el despliegue de los procesos humanos apropiados durante el desarrollo de productos.⇒ **Outcome d**
- CS10.** Implementar efectivamente las herramientas que se utilizan para la construcción y la documentación de software, con especial énfasis en la comprensión de todo el proceso involucrado en el uso de computadoras para resolver problemas prácticos. Esto debe incluir herramientas para el control de software, incluyendo el control de versiones y gestión de la configuración.⇒ **Outcome d**

10. Lista de temas a estudiar en el curso

1. Fundamentos
2. Factores Humanos
3. Diseño y Testing centrados en el usuario
4. Diseño de Interacción
5. Nuevas Tecnologías Interactivas
6. Colaboración y Comunicación

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Fundamentos (8)	
Competencias esperadas: CS8	
Objetivos de Aprendizaje	Tópicos
<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]	<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.
Lecturas : [Dix+04], [Sto+05], [RS11]	

Unidad 2: Factores Humanos (8)	
Competences esperadas: CS8	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Crear y dirigir una simple prueba de usabilidad para una aplicación existente de software [Familiarizarse] 	<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)
Lecturas : [Dix+04], [Sto+05], [RS11], [Mat11], [Nor04]	

Unidad 3: Diseño y Testing centrados en el usuario (16)	
Competencias esperadas: C7, CS8, CS10	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Dix+04], [Sto+05], [RS11], [Mat11], [Bux07]	

Unidad 4: Diseño de Interacción (8)	
Competencias esperadas: CS8, CS15	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Dix+04], [Sto+05], [RS11], [Joh10], [Mat11], [LS06]	

Unidad 5: Nuevas Tecnologías Interactivas (8)	
Competencias esperadas: C9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Dix+04], [Sto+05], [RS11], [WW11], [Mat11]	

Unidad 6: Colaboración y Comunicación (8)	
Competencias esperadas: CS8, CS9	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Describir la diferencia entre la comunicación sincrónica y asincrónica [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] 	<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
Lecturas : [Dix+04], [Sto+05], [RS11]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS341. Lenguajes de Programación (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 2 HP; 2 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [RH04] Peter Van Roy and Seif Haridi. *Concepts, Techniques, and Models of Computer Programming*. Cambridge, MA, USA: MIT Press, 2004. ISBN: 0262220695.
- [Seb12] Robert W. Sebesta. *Concepts of Programming Languages*. 10th. USA: Addison-Wesley Publishing Company, 2012. ISBN: 0131395319.
- [Web10] Adam Brooks Webber. *Modern Programming Languages: A Practical Introduction*. 2nd. Franklin, Beedle and Associates, Inc, 2010. ISBN: 978-1-59028-250-2.

6. Información del curso

- (a) **Breve descripción del curso** Los lenguajes de programación son el medio a través del cual los programadores describen con precisión los conceptos, formulan algoritmos y representan sus soluciones. Un científico de la computación trabajará con diferentes lenguajes, por separado o en conjunto. Los científicos de la computación deben entender los modelos de programación de los diferentes lenguajes, tomar decisiones de diseño basados en el lenguaje de programación y sus conceptos. El profesional a menudo necesitará aprender nuevos lenguajes y construcciones de programación y debe entender los fundamentos de como las características del lenguaje de programación están definidas, compuestas e implementadas. El uso eficaz de los lenguajes de programación y la apreciación de sus limitaciones, también requiere un conocimiento básico de traducción de lenguajes de programación y su análisis de ambientes estáticos y dinámicos, así como los componentes de tiempo de ejecución tales como la gestión de memoria, entre otros detalles de relevancia.
- (b) **Prerrequisitos:** CS2101. Teoría de la Computación. (4^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Capacitar a los estudiantes para entender los lenguajes de programación desde diferentes tipos de vista, según el modelo subyacente, los componentes fundamentales presentes en todo lenguaje de programación y como objetos formales dotados de una estructura y un significado según diversos enfoques.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Usar**)

9. Competencias (IEEE)

- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome a**
- CS1.** Modelar y diseñar sistemas de computadora de una manera que se demuestre comprensión del balance entre las opciones de diseño.⇒ **Outcome b**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome i**
- CS3.** Analizar el grado en que un sistema basado en el ordenador cumple con los criterios definidos para su uso actual y futuro desarrollo.⇒ **Outcome j**
- CS6.** Evaluar los sistemas en términos de atributos de calidad en general y las posibles ventajas y desventajas que se presentan en el problema dado.⇒ **Outcome j**

10. Lista de temas a estudiar en el curso

- 1.
2. Pragmática de lenguajes
3. Sistemas de tipos
4. Programación orientada a objetos
5. Programación funcional
6. Programación reactiva y dirigida por eventos
7. Programación lógica

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: (18)	
Competencias esperadas: C2, CS1, CS2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Reconocer el desarrollo histórico de los lenguajes de programación. [Familiarizarse] • Identificar los paradigmas que agrupan a la mayoría de lenguajes de programación existentes hoy en día. [Familiarizarse] • 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] • Escribir un programa para procesar alguna representación de código para algún propósito, tales como un interprete, una expresión optimizada, o un generador de documentación [Usar] • 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) [Familiarizarse] • Reconocer como funciona un programa a nivel de computador. [Familiarizarse] 	<ul style="list-style-type: none"> • Historia de los Lenguajes de Programación • Programas que tienen otros programas como entrada tales como interpretes, compiladores, revisores de tipos y generadores de documentación. • Estructuras de datos que representan código para ejecución, traducción o transmisión. • Estructura de un programa: Léxico, Sintáctico y Semántico • BNF • Interpretación vs. compilación a código nativo vs. compilación de representación portable intermedia. [Familiarizarse]
Lecturas : [Seb12], [Web10]	

Unidad 2: Pragmática de lenguajes (12)	
Competencias esperadas: C2, CS1, CS2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Discute el rol de conceptos como ortogonalidad y el buen criterio de selección en el diseño de lenguajes [Usar] • Utiliza criterios objetivos y nítidos para evaluar las decisiones en el diseño de un lenguaje [Usar] • Da un ejemplo de un programa cuyo resultado puede diferir dado diversas reglas de orden de evaluación, precedencia, o asociatividad [Usar] • Muestra el uso de evaluación con retraso, como en el caso de abstracciones definidas y controladas por el usuario [Familiarizarse] • Discute la necesidad de permitir llamadas a librerías externas y del sistema y las consecuencias de su implementación en un lenguaje [Familiarizarse] 	<ul style="list-style-type: none"> • Principios de diseño de lenguaje tales como la ortogonalidad. • Orden de evaluación, precedencia y asociatividad. • Evaluación tardía vs. evaluación temprana. • Definiendo controles y constructos de iteración. • Llamadas externas y sistema de librerías.
Lecturas : [Seb12], [Web10], [RH04]	

Unidad 3: Sistemas de tipos (18)	
Competencias esperadas: C2, CS1, CS2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Definir un sistema de tipo de forma precisa y en su composición [Usar] • Para varias construcciones de tipo fundamental, identificar los valores que describen y las invariantes que hacen que se cumplan [Familiarizarse] • Precisar las invariantes preservadas por un sistema de tipos seguro (<i>sound type system</i>) [Familiarizarse] • Demostrar la seguridad de tipos para un lenguaje simple en términos de conservación y progreso teoremas [Usar] • Implementar un algoritmo de inferencia de tipos basado en la unificación para un lenguaje básico [Usar] • Explicar cómo la sobrecarga estática y algoritmos de resolución asociados influyen el comportamiento dinámico de los programas [Familiarizarse] 	<ul style="list-style-type: none"> • Constructores de tipo composicional, como tipos de producto (para agregados), tipos de suma (para uniones), tipos de función, tipos cuantificados y tipos recursivos. • Comprobación de tipos. • Seguridad de tipos como preservación más progreso. • Inferencia de tipos. • Sobrecarga estática.
Lecturas : [Seb12], [Web10], [RH04]	

Unidad 4: Programación orientada a objetos (12)	
Competencias esperadas: CS2, CS3, CS6	
Objetivos de Aprendizaje	Tópicos
<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 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) [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 mas natural por cada lenguaje [Usar] 	<ul style="list-style-type: none"> • Diseño orientado a objetos: <ul style="list-style-type: none"> – Descomposicion en objetos que almacenan estados y poseen comportamiento – Diseño basado en jerarquia 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 coleccion de clases, iteradores, y otros componentes de la libreria estandar.
Lecturas : [Seb12], [Web10], [RH04]	

Unidad 5: Programación funcional (18)	
Competencias esperadas: CS2, CS3, CS6	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Escribir algoritmos básicos que eviten asignación a un estado mutable o considerar igualdad de referencia [Usar] • Escribir funciones útiles que puedan tomar y retornar otras funciones [Usar] • Comparar y contrastar (1) el enfoque procedurar/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] • Razonar correctamente sobre variables y el ámbito léxico en un programa usando funciones de cierre (<i>function closures</i>) [Usar] • Usar mecanismos de encapsulamiento funcional, tal como <i>closures</i> e interfaces modulares [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] 	<ul style="list-style-type: none"> • El efecto de la programación libre: <ul style="list-style-type: none"> – Llamadas a función que no tiene efecto secundarios, para facilitar el razonamiento composicional – Variables inmutables, prevencion de cambios no esperados en los datos del programa por otro código. – Datos que pueden ser subnombrados o copiados libremente sin introducir efectos no deseados del cambio • Procesamiento de estructuras de datos (p.e. arboles) a través de fuciones con casos para cada variación de los datos. <ul style="list-style-type: none"> – Constructores asociados al lenguaje tales como uniones discriminadas y reconocimiento de patrones sobre ellos. – Funciones definidas sobre datos compuestos en términos de funciones aplicadas a las piezas constituidas. • Funciones de primera clase (obtener, retornar y funciones de almacenamiento) • Cierres de función (funciones que usan variables en entornos léxicos cerrados) <ul style="list-style-type: none"> – Significado y definicion básicos - creacion de cierres en tiempo de ejecución mediante la captura del entorno. – Idiomas canónicos: llamadas de retorno, argumentos de iteradores, código reusable mediante argumentos de función – Uso del cierre para encapsular datos en su entorno – Evaluación y aplicación parcial • Definición de las operaciones de orden superior en los agregados, especialmente en mapa, reducir / doblar, y el filtro.
Lecturas : [Seb12], [Web10], [RH04]	

Unidad 6: Programación reactiva y dirigida por eventos (12)	
Competencias esperadas: CS2, CS3, CS6	
Objetivos de Aprendizaje	Tópicos
<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 [Familiarizarse] • Describir un sistema interactivo en términos de un modelo, una vista y un controlador [Familiarizarse] 	<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.
Lecturas : [Seb12]	

Unidad 7: Programación lógica (12)	
Competencias esperadas: CS2, CS3, CS6	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Usa un lenguaje lógico para implementar un algoritmo convencional [Usar] • Usa un lenguaje lógico para implementar un algoritmo empleando búsqueda implícita usando cláusulas, relaciones, y cortes [Usar] 	<ul style="list-style-type: none"> • Representación causal de estructura de datos y algoritmos. • Unificación. • Backtracking y búsqueda. • Cuts.
Lecturas : [Seb12], [Web10], [RH04]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3P01. Computación Paralela y Distribuida (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.

(b) **Prerrequisitos:**

- CS2102. Análisis y Diseño de Algoritmos. (5^{to} Sem)
- CS2301. Redes y Comunicaciones. (6^{to} Sem)

(c) **Tipo de Curso:** Obligatorio

(d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. (**Usar**)

9. Competencias (IEEE)

- C3.** Una comprensión intelectual de, y el aprecio por el papel central de los algoritmos y estructuras de datos.⇒ **Outcome b,g**
- C4.** Una comprensión del hardware de la computadora desde la perspectiva del software, por ejemplo, el uso del procesador, memoria, unidades de disco, pantalla, etc.⇒ **Outcome b,c**
- CS1.** Modelar y diseñar sistemas de computadora de una manera que se demuestre comprensión del balance entre las opciones de diseño.⇒ **Outcome a**
- CS4.** Implementar la teoría apropiada, prácticas y herramientas para la especificación, diseño, implementación y mantenimiento, así como la evaluación de los sistemas basados en computadoras.⇒ **Outcome a**

10. Lista de temas a estudiar en el curso

1. Fundamentos de paralelismo
2. Arquitecturas paralelas
3. Descomposición en paralelo
4. Comunicación y coordinación
5. Análisis y programación de algoritmos paralelos
6. Desempeño en paralelo

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas

es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Fundamentos de paralelismo (18)	
Competences esperadas: C2	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [Pac11], [Mat14], [Qui03]	

Unidad 2: Arquitecturas paralelas (12)	
Competencias esperadas: C4	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [Pac11], [KH13], [SK10]	

Unidad 3: Descomposición en paralelo (18)	
Competencias esperadas: C16	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [Pac11], [Mat14], [Qui03]	

Unidad 4: Comunicación y coordinación (18)	
Competencias esperadas: C16	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Usar exclusión mútua 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] 	<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"> – (Ciclicos) barreras, contadores y estructuras relacionadas • Acciones condicionales: <ul style="list-style-type: none"> – Espera condicional (p.e., empleando variables de condición)
Lecturas : [Pac11], [Mat14], [Qui03]	

Unidad 5: Análisis y programación de algoritmos paralelos (18)	
Competencias esperadas: CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Mat14], [Qui03]	

Unidad 6: Desempeño en paralelo (18)	
Competencias esperadas: CS3	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Pac11], [Mat14], [KH13], [SK10]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2501. Computación Gráfica (Electivo)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS3102. Estructuras de Datos Avanzadas. (6^{to} Sem)
- (c) **Tipo de Curso:** Electivo
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome b**
- C5.** Capacidad para implementar algoritmos y estructuras de datos en el software.⇒ **Outcome b**

C4. Una comprensión del hardware de la computadora desde la perspectiva del software, por ejemplo, el uso del procesador, memoria, unidades de disco, pantalla, etc.⇒ **Outcome i**

C8. Entendimiento de lo que las tecnologías actuales pueden y no pueden lograr.⇒ **Outcome i**

10. Lista de temas a estudiar en el curso

1. Conceptos Fundamentales
2. Rendering Básico
3. Programación de Sistemas Interactivos
4. Modelado Geométrico
5. Renderizado Avanzado
6. Animación por computadora

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Conceptos Fundamentales (6)	
Competencias esperadas: C1,C2	
Objetivos de Aprendizaje	Tópicos
<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 básicos de la producción de movimiento continuo a partir de una secuencia de cuadros discretos (algunas veces llamado it flicker fusion) [Familiarizarse] 	<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.
Lecturas : [HB90]	

Unidad 2: Rendering Básico (12)	
Competencias esperadas: C1,C4	
Objetivos de Aprendizaje	Tópicos
<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 afín [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] 	<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)
Lecturas : [MS16]	

Unidad 3: Programación de Sistemas Interactivos (2)	
Competencias esperadas: C8	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Discute las ventajas (y desventajas) de las interfaces no basadas en ratón [Evaluar] 	<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
Lecturas : [MS16]	

Unidad 4: Modelado Geométrico (15)	
Competencias esperadas: C1,C5	
Objetivos de Aprendizaje	Tópicos
<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 cuadradas [Usar] • Contrastar métodos de modelización con respecto a espacio y tiempo de complejidad y calidad de imagen [Evaluar] 	<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) espigas, 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)
Lecturas : [MS16]	

Unidad 5: Renderizado Avanzado (6)	
Competencias esperadas: C1,C4	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [MS16]	

Unidad 6: Animación por computadora (4)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [MS16]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2601. Inteligencia Artificial (Electivo)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.
- [Hay99] Simon Haykin. *Neural networks: A Comprehensive Foundation*. Prentice Hall, 1999.
- [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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** IN0054. Estadística y Probabilidades. (4^{to} Sem)
- (c) **Tipo de Curso:** Electivo
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Familiarizarse**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**

C20. Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome a**

CS2. Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome j**

10. Lista de temas a estudiar en el curso

1. Cuestiones fundamentales
2. Estrategias de búsquedas básicas
3. Raciocinio y representación básica de conocimiento
4. Búsqueda Avanzada
5. Representación Avanzada y Razonamiento
6. Agentes
7. Procesamiento del Lenguaje Natural
8. Aprendizaje Automático Básico
9. Robótica
10. Visión y percepción por computador

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Cuestiones fundamentales (2)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [De 06], [Pon+14]	

Unidad 2: Estrategias de búsquedas básicas (4)	
Competencias esperadas: C20	
Objetivos de Aprendizaje	Tópicos
<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] • Selecciona e implementa un apropiado algoritmo de búsqueda no informado para un problema, y describe sus complejidades de tiempo y espacio [Usar] • Selecciona e implementa un apropiado algoritmo de búsqueda informado para un problema al definir la función heurística de evaluación necesaria [Usar] • Evalúa si una heurística dada para un determinado problema es admisible/puede garantizar una solución óptima [Usar] • Formula un problema en particular en lenguaje natural (ejm. Inglés) como un problema de satisfacción de restricciones y lo implementa usando un algoritmo de retroceso cronológico o una búsqueda estocástica local [Usar] • Compara y contrasta tópicos de búsqueda básica con temas jugabilidad de juegos [Usar] 	<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).
Lecturas : [Nil01], [Pon+14]	

Unidad 3: Raciocinio y representación básica de conocimiento (6)	
Competencias esperadas: C24	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Traducir una sentencia en lenguaje natural (Por ejemplo español) en una declaración lógica de predicados [Usar] • Convertir una declaración lógica en forma de cláusula [Usar] • Aplicar resolución a un conjunto de declaraciones lógicas para responder una consulta [Usar] • Hacer una inferencia probabilística para un problema real usando el teorema de Bayes para determinar la probabilidad que se cumpla una hipótesis [Usar] 	<ul style="list-style-type: none"> • Revisión de la lógica proposicional y de predicados • Resolución y demostración de teoremas (sólo la lógica proposicional). • Encadenamiento hacia adelante, encadenamiento hacia atrás. • Examen de razonamiento probabilístico, el teorema de Bayes.
Lecturas : [Nil01], [RN03], [Pon+14]	

Unidad 4: Búsqueda Avanzada (4)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Construcción de árboles de búsqueda, espacio de búsqueda dinámico, explosión combinatoria del espacio de búsqueda. • Búsqueda estocástica: <ul style="list-style-type: none"> – Simulated annealing – Algoritmos genéticos – Búsqueda de árbol Monte-Carlo • Implementación de búsqueda A *, búsqueda en haz. • Búsqueda Minimax, poda alfa-beta. • Búsqueda Expectimax (MDP-Solving) y los nodos de azar.
Lecturas : [Gol89], [Nil01], [RN03], [Pon+14]	

Unidad 5: Representación Avanzada y Razonamiento (6)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Comparar y contrastar los modelos más usados para la representación del conocimiento estructurado, destacando sus puntos fuertes y débiles [Usar] • Identificar los componentes de razonamiento no monótono y su utilidad como mecanismo de representación de los sistemas de confianza [Usar] • Comparar y contrastar las técnicas básicas para la representación de la incertidumbre [Usar] • Comparar y contrastar las técnicas básicas para la representación cualitativa [Usar] • Aplicar cálculo de situaciones y eventos a problemas de acción y cambios [Usar] • Explicar la diferencia entre razonamiento temporal y espacial, y cómo se relacionan entre sí. [Usar] • Explicar la diferencia entre técnicas de razonamiento basado en modelos, basado en casos y basados en reglas [Usar] • Definir el concepto de un sistema planificación y cómo se diferencia de las técnicas de búsqueda clásicas [Usar] 	<ul style="list-style-type: none"> • Problemas de Representación del Conocimiento: <ul style="list-style-type: none"> – Lógica de Descripción – Ingeniería de Ontología • Razonamiento no monotónico (p.e., lógica no clásica, razonamiento por defecto) • Argumentación • El razonamiento sobre la acción y el cambio (por ejemplo, la situación y cálculo de eventos). • Razonamiento temporal y espacial. • Sistemas Expertos basados en reglas. • Redes semánticas. • Razonamiento basado en modelos y razonamiento basado en casos.
Lecturas : [Nil01], [RN03], [Pon+14]	

Unidad 6: Agentes (6)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Nil01], [RN03], [Pon+14]	

Unidad 7: Procesamiento del Lenguaje Natural (4)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Nil01], [RN03], [Pon+14]	

Unidad 8: Aprendizaje Automático Básico (10)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Hay99], [Nil01], [RN03], [Pon+14]	

Unidad 9: Robótica (6)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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 [Usar] • Integrar sensores, actuadores y software en un robot diseñado para emprender alguna tarea [Usar] • 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] • 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] • 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 [Usar] • Describir al menos una aproximación para la coordinación de acciones y detección de varios robots para realizar una simple tarea [Usar] 	<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. • Interpretando datos del sensor con incertidumbre. • Localización y mapeo. • Navegación y control. • Planeando el movimiento. • Coordinación multi-robots.
Lecturas : [Nil01], [RN03], [Pon+14]	

Unidad 10: Visión y percepción por computador (6)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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] • Distinguir las metas de reconocimiento de sonido, palabras y del habla e identificar como la señal de audio bruto sera manejada diferentemente en cada uno de esos casos. [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 que combina características en percepciones de más alto nivel, p.e., un contorno o poligono a partir de primitivas visuales o fonemas de una señal de audio [Usar] • Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalua 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] • Describir por lo menos tres enfoques de clasificación, sus pre requisitos para aplicabilidad, fortalezas y deficiencias [Usar] 	<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 • Audio y reconocimiento de dictado. • 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.
Lecturas : [Nil01], [RN03], [Pon+14]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS2902. Ingeniería de Software II (Electivo)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS2901. Ingeniería de Software I. (5^{to} Sem)
- (c) **Tipo de Curso:** Electivo
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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

8. Contribución a los resultados (*Outcomes*)

- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)

9. Competencias (IEEE)

- C7.** Ser capaz de aplicar los principios y tecnologías de ingeniería de software para asegurar que las implementaciones de software son robustos, fiables y apropiados para su público objetivo.⇒ **Outcome c**
- C11.** Entendimiento del concepto del ciclo de vida, incluyendo la importancia de sus fases (planificación, desarrollo, implementación y evolución).⇒ **Outcome c**
- C12.** Entender las implicaciones de ciclo de vida para el desarrollo de todos los aspectos de los sistemas informáticos (incluyendo software, hardware, y la interfaz de la computadora humana).⇒ **Outcome c,i**
- C18.** Capacidad para participar de forma activa y coordinada en un equipo.⇒ **Outcome f**
- CS1.** Modelar y diseñar sistemas de computadora de una manera que se demuestre comprensión del balance entre las opciones de diseño.⇒ **Outcome c**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome c**
- CS4.** Implementar la teoría apropiada, prácticas y herramientas para la especificación, diseño, implementación y mantenimiento, así como la evaluación de los sistemas basados en computadoras.⇒ **Outcome c,i**
- CS5.** Especificar, diseñar e implementar sistemas basados en computadoras.⇒ **Outcome c,i**
- CS10.** Implementar efectivamente las herramientas que se utilizan para la construcción y la documentación de software, con especial énfasis en la comprensión de todo el proceso involucrado en el uso de computadoras para resolver problemas prácticos. Esto debe incluir herramientas para el control de software, incluyendo el control de versiones y gestión de la configuración.⇒ **Outcome i**

10. Lista de temas a estudiar en el curso

1. Herramientas y Entornos
2. Verificación y Validación de Software
3. Evolución de Software
4. Gestión de Proyectos de Software

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Herramientas y Entornos (12)	
Competencias esperadas: C1	
Objetivos de Aprendizaje	Tópicos
<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]	<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.
Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unidad 2: Verificación y Validación de Software (12)	
Competencias esperadas: C20	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Verificación y validación de conceptos. • Inspecciones, revisiones, auditorías. • 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 envíos por defecto.
Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unidad 3: Evolución de Software (12)	
Competencias esperadas: C20	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unidad 4: Gestión de Proyectos de Software (12)	
Competencias esperadas: C24	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS4002. Proyecto de Final de Carrera I (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 3 HT; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS2102. Análisis y Diseño de Algoritmos. (5^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)

- e) Entender correctamente las implicancias profesionales, éticas, legales, de seguridad y sociales de la profesión. (**Evaluar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)
- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)
- l) Desarrollar principios investigación en el área de computación con niveles de competitividad internacional. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a,b,c**
- C20.** Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome e,f,g**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome h,i,l**

10. Lista de temas a estudiar en el curso

1. Levantamiento del estado del arte

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Levantamiento del estado del arte (60)	
Competencias esperadas: C1,C20,CS2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Hacer un levantamiento bibliográfico del estado del arte del tema escogido (esto significa muy probablemente 1 o 2 capítulos de marco teórico además de la introducción que es el capítulo 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, índices, bibtex, referencias cruzadas, citas, pstricks) [Usar] • Tratar de hacer las presentaciones utilizando prosper [Usar] • Mostrar experimentos básicos [Usar] • Elegir un asesor que domine el área de investigación realizada [Usar] 	<ul style="list-style-type: none"> • Realizar un estudio profundo del estado del arte en un determinado tópico del área de Computación. • Redacción de artículos técnicos en computación.
Lecturas : [IEE08], [Ass08], [Cit08]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** ET201. Formación de Empresas de Base Tecnológica I (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 2 HT; 2 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.

- (b) **Prerrequisitos:** Ninguno
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)

- f) Comunicarse efectivamente con audiencias diversas. (**Evaluar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- k) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. (**Evaluar**)
- m) Transformar sus conocimientos del área de Ciencia de la Computación en emprendimientos tecnológicos. (**Evaluar**)

9. Competencias (IEEE)

- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome d**
- C10.** Comprensión del impacto en las personas, las organizaciones y la sociedad de la implementación de soluciones tecnológicas e intervenciones.⇒ **Outcome f**
- C17.** Capacidad para expresarse en los medios de comunicación orales y escritos como se espera de un graduado.⇒ **Outcome f**
- C18.** Capacidad para participar de forma activa y coordinada en un equipo.⇒ **Outcome i**
- C19.** Capacidad para identificar eficazmente los objetivos y las prioridades de su trabajo / área / proyecto con indicación de la acción, el tiempo y los recursos necesarios.⇒ **Outcome i**
- C20.** Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome k**
- C23.** Capacidad para emprender, completar, y presentar un proyecto final.⇒ **Outcome k**
- CS5.** Especificar, diseñar e implementar sistemas basados en computadoras.⇒ **Outcome m**

10. Lista de temas a estudiar en el curso

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:**12. Contenido**

Unidad 1: (5)	
Competences esperadas: C2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Identificar características de los emprendedores [Familiarizarse] • Introducir modelos de negocio [Familiarizarse] 	<ul style="list-style-type: none"> • Emprendedor, emprendedurismo e innovación tecnológica • Modelos de negocio • Formación de equipos
Lecturas : [BDN10], [OP10], [Gar+14]	

Unidad 2: (5)	
Competences esperadas: C10	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [BDN10], [BD12], [Gar+14]	

Unidad 3: (5)	
Competences esperadas: C17	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Conocer estrategias empresariales [Familiarizarse] • Caracterizar barreras y ventajas competitivas [Familiarizarse] 	<ul style="list-style-type: none"> • Estrategia de la Empresa • Barreras • Ventaja competitiva sostenible • Alianzas • Aprendizaje organizacional • Desarrollo y diseño de productos
Lecturas : [BDN10], [OP10], [Rie11], [Gar+14]	

Unidad 4: (20)	
Competencias esperadas: C18	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Conocer los elementos del modelo Canvas [Usar] • Elaborar un plan de negocio basado en el modelo Canvas [Usar] 	<ul style="list-style-type: none"> • Creación de un nuevo negocio • El plan de negocio • Canvas • Elementos del Canvas
Lecturas : [OP10], [BD12], [Gar+14]	

Unidad 5: (20)	
Competencias esperadas: C19	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Conocer y aplicar el modelo Customer Development [Usar] • Conocer y aplicar el modelo Lean Startup [Usar] 	<ul style="list-style-type: none"> • Aceleración versus incubación • Customer Development • Lean Startup
Lecturas : [BD12], [Rie11], [Gar+14]	

Unidad 6: (5)	
Competencias esperadas: C20	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [BDN10], [Rie11], [Con96], [Rep97], [Gar+14]	

Unidad 7: (5)	
Competencias esperadas: C23	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Definir un modelo de costos y utilidades [Evaluar] • Conocer las diversas fuentes de financiamiento [Familiarizarse] 	<ul style="list-style-type: none"> • Modelo de costos • Modelo de utilidades • Precio • Plan financiero • Formas de financiamiento • Fuentes de capital • Capital de riesgo
Lecturas : [BDN10], [BD12], [Gar+14]	

Unidad 8: (5)	
Competencias esperadas: CS5	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • The Elevator Pitch • Presentación • Negociación
Lecturas : [BDN10], [BD12], [Gar+14]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3700. Big Data (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 1 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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>.

6. Información del curso

(a) **Breve descripción del 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 necesita ser procesada, almacenada, analizada.

(b) **Prerrequisitos:**

- CS2702. Bases de Datos II. (5^{to} Sem)
- CS3P01. Computación Paralela y Distribuida. (7^{mo} Sem)

(c) **Tipo de Curso:** Obligatorio

(d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Usar**)

9. Competencias (IEEE)

- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome a,b**
- C16.** Capacidad para identificar temas avanzados de computación y de la comprensión de las fronteras de la disciplina.⇒ **Outcome i**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome i,b**
- CS3.** Analizar el grado en que un sistema basado en el ordenador cumple con los criterios definidos para su uso actual y futuro desarrollo.⇒ **Outcome j**
- CS6.** Evaluar los sistemas en términos de atributos de calidad en general y las posibles ventajas y desventajas que se presentan en el problema dado.⇒ **Outcome j**

10. Lista de temas a estudiar en el curso

1. Introducción a Big Data
2. Hadoop
3. Procesamiento de Grafos en larga escala

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Introducción a Big Data (15)	
Competencias esperadas: C2, C4	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Cou+11]	

Unidad 2: Hadoop (15)	
Competencias esperadas: C2, C4	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Visión global de Hadoop. • Historia. • Estructura de Hadoop. • HDFS, Hadoop Distributed File System. • Modelo de Programación MapReduce
Lecturas : [HDF11], [BVS13]	

Unidad 3: Procesamiento de Grafos en larga escala (10)	
Competencias esperadas: C16	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Low+12], [Mal+10], [Bal+08]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3I01. Seguridad en Computación (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 1 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

[WL14] Stallings. W and Brown. L. *Computer Security: Principles and Practice*. Pearson Education, Limited, 2014. ISBN: 9780133773927.

6. Información del curso

- (a) **Breve descripción del 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 prever 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.
- (b) **Prerrequisitos:** CS2301. Redes y Comunicaciones. (6^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- e) Entender correctamente las implicancias profesionales, éticas, legales, de seguridad y sociales de la profesión. (**Usar**)
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. (**Evaluar**)
- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)

9. Competencias (IEEE)

- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional. ⇒ **Outcome a**

- C8.** Entendimiento de lo que las tecnologías actuales pueden y no pueden lograr.⇒ **Outcome g**
- C9.** Comprensión de las limitaciones de la computación, incluyendo la diferencia entre lo que la computación es inherentemente incapaz de hacer frente a lo que puede lograrse a través de un futuro de ciencia y tecnología.⇒ **Outcome g,a**
- C21.** Comprender el aspecto profesional, legal, seguridad, asuntos políticos, humanistas, ambientales, culturales y éticos.⇒ **Outcome e**
- C22.** Capacidad para demostrar las actitudes y prioridades que honrar, proteger y mejorar la estatura y la reputación ética de la profesión.⇒ **Outcome e,h**
- CS7.** Aplicar los principios de una gestión eficaz de la información, organización de la información, y las habilidades de recuperación de información a la información de diversos tipos, incluyendo texto, imágenes, sonido y vídeo. Esto debe incluir la gestión de los problemas de seguridad.⇒ **Outcome i,h**
- CS9.** Identificar los riesgos (y esto incluye cualquier seguridad o los aspectos de seguridad) que pueden estar involucrados en la operación de equipo de cómputo dentro de un contexto dado.⇒ **Outcome b**
- CS11.** Ser consciente de la existencia de software a disposición del público y la comprensión del potencial de los proyectos de código abierto.⇒ **Outcome b,g**

10. Lista de temas a estudiar en el curso

1. Fundamentos y Conceptos en Seguridad
2. Principios de Diseño Seguro
3. Programación Defensiva
4. Ataques y Amenazas
5. Seguridad de Red
6. Criptografía
7. Seguridad en la Web
8. Seguridad de plataformas
9. Investigación digital (Digital Forensics)
10. Seguridad en Ingeniería de Software

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Fundamentos y Conceptos en Seguridad (25)	
Competencias esperadas: C2,C8	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [WL14]	

Unidad 2: Principios de Diseño Seguro (25)	
Competencias esperadas: C,9C21,C22	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [WL14]	

Unidad 3: Programación Defensiva (25)	
Competencias esperadas: CS6,CS7,CS9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [WL14]	

Unidad 4: Ataques y Amenazas (25)	
Competencias esperadas: CS6,CS7,CS9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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, perscando) • Los ataques a la privacidad y el anonimato . • El malware / comunicaciones no deseadas, tales como canales encubiertos y esteganografía.
Lecturas : [WL14]	

Unidad 5: Seguridad de Red (25)	
Competencias esperadas: CS6,CS7,CS9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [WL14]	

Unidad 6: Criptografía (25)	
Competencias esperadas: CS6,CS7,CS9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [WL14]	

Unidad 7: Seguridad en la Web (25)	
Competencias esperadas: C8,C9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [WL14]	

Unidad 8: Seguridad de plataformas (25)	
Competencias esperadas: CS6,CS7,CS9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [WL14]	

Unidad 9: Investigación digital (Digital Forensics) (25)	
Competencias esperadas: C8,C9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [WL14]	

Unidad 10: Seguridad en Ingeniería de Software (25)	
Competencias esperadas: C21,C22	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [WL14]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS4003. Proyecto de Final de Carrera II (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del curso** Este curso tiene por objetivo que el alumno concluya su proyecto de tesis.
- (b) **Prerrequisitos:** CS4002. Proyecto de Final de Carrera I. (8^{vo} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Evaluar**)
- e) Entender correctamente las implicancias profesionales, éticas, legales, de seguridad y sociales de la profesión. (**Evaluar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Evaluar**)

- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Evaluar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)
- l) Desarrollar principios investigación en el área de computación con niveles de competitividad internacional. (**Evaluar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a,b,c**
- C20.** Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome e,f,g**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome h,i,l**

10. Lista de temas a estudiar en el curso

1. Proyecto de Tesis
2. Avance de Tesis

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Proyecto de Tesis (30)	
Competences esperadas: C1,C20,CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Proyecto de Tesis.
Lecturas : [IEE08], [Ass08], [Cit08]	

Unidad 2: Avance de Tesis (30)	
Competencias esperadas: C1,C20,CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Avance de Tesis.
Lecturas : [IEE08], [Ass08], [Cit08]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3501. Tópicos en Computación Gráfica (Electivo)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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]
- (b) **Prerrequisitos:** CS2501. Computación Gráfica. (7^{mo} Sem)
- (c) **Tipo de Curso:** Electivo
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- l) Desarrollar principios investigación en el área de computación con niveles de competitividad internacional. (**Usar**)
- m) Transformar sus conocimientos del área de Ciencia de la Computación en emprendimientos tecnológicos. (**Usar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a,b**

C20. Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome l,m**

10. Lista de temas a estudiar en el curso

1. Topics Avanzados en Computación Gráfica

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Topics Avanzados en Computación Gráfica (0)	
Competences esperadas: 4	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none">• Tópicos Avanzados en Computación Gráfica	<ul style="list-style-type: none">• CS355. Advanced Computer Graphics• CS356. Computer animation• CS313. Geometric Algorithms• CS357. Visualización• CS358. Virtual reality
Lecturas : [MS16]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3602. Robótica (Electivo)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [MVR07] Sonka. M, Hlavac. V, and Boile. R. *Image Processing, Analysis and Machine Vision*. Cengage-Engineering, 2007.
- [RR07] Gonzales. R C and Woods. R E. *Digital Image Processing*. Prentice Hall, 2007. ISBN: 013168728X,978013168728B.
- [SN04] R. Siegwart and I. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. The MIT Press., 2004. ISBN: 0-262-19502-X.
- [Sto00] Peter Stone. *Layered Learning in Multiagent Systems*. Intelligent Robots and Autonomous Agents. The MIT Press, 2000. ISBN: 9780262194389.
- [SWD05] Thrun. S, Burgard. W, and Fox. D. *Probabilistic Robotics*. Intelligent Robots and Autonomous Agents. The MIT Press, 2005.

6. Información del curso

- (a) **Breve descripción del 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
- (b) **Prerrequisitos:** CS2601. Inteligencia Artificial. (7^{mo} Sem)
- (c) **Tipo de Curso:** Electivo
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- e) Entender correctamente las implicancias profesionales, éticas, legales, de seguridad y sociales de la profesión. (**Usar**)
- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Familiarizarse**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)

9. Competencias (IEEE)

- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional. ⇒ **Outcome a,b**
- C8.** Entendimiento de lo que las tecnologías actuales pueden y no pueden lograr. ⇒ **Outcome e**
- C23.** Capacidad para emprender, completar, y presentar un proyecto final. ⇒ **Outcome b,i,h**
- CS1.** Modelar y diseñar sistemas de computadora de una manera que se demuestre comprensión del balance entre las opciones de diseño. ⇒ **Outcome b**

10. Lista de temas a estudiar en el curso

1. Robótica
2. Robótica
3. Robótica
4. Visión y percepción por computador
5. Robótica

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Robótica (5)	
Competencias esperadas: CS12	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [SN04], [SWD05], [Sto00]	

Unidad 2: Robótica (15)	
Competencias esperadas: C2,C23	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Interpretando datos del sensor con incertidumbre. • Localización y mapeo.
Lecturas : [SN04], [SWD05]	

Unidad 3: Robótica (20)	
Competencias esperadas: CS1	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Navegación y control. • Planeando el movimiento.
Lecturas : [SN04]	

Unidad 4: Visión y percepción por computador (10)	
Competencias esperadas: C2,CS1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [MVR07], [RR07]	

Unidad 5: Robótica (10)	
Competencias esperadas: C23,CS1	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Coordinación multi-robots.
Lecturas : [Sto00]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3901. Ingeniería de Software III (Electivo)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [PM14] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2014.
- [Som10] Ian Sommerville. *Software Engineering*. 9th. Addison-Wesley, Mar. 2010.

6. Información del curso

- (a) **Breve descripción del 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 elemento 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.
- (b) **Prerrequisitos:** CS2902. Ingeniería de Software II. (7^{mo} Sem)
- (c) **Tipo de Curso:** Electivo
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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 éxitos de los proyectos de TI.

8. Contribución a los resultados (*Outcomes*)

- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Evaluar**)

- m) Transformar sus conocimientos del área de Ciencia de la Computación en emprendimientos tecnológicos. (**Evaluar**)
- o) Mejorar las condiciones de la sociedad poniendo la tecnología al servicio del ser humano. (**Usar**)

9. Competencias (IEEE)

- C7. Ser capaz de aplicar los principios y tecnologías de ingeniería de software para asegurar que las implementaciones de software son robustos, fiables y apropiados para su público objetivo.⇒ **Outcome c**
- C11. Entendimiento del concepto del ciclo de vida, incluyendo la importancia de sus fases (planificación, desarrollo, implementación y evolución).⇒ **Outcome i,k**
- C12. Entender las implicaciones de ciclo de vida para el desarrollo de todos los aspectos de los sistemas informáticos (incluyendo software, hardware, y la interfaz de la computadora humana).⇒ **Outcome j,m**
- C13. Comprender la relación entre la calidad y la gestión del ciclo de vida.⇒ **Outcome c,i,m**
- C18. Capacidad para participar de forma activa y coordinada en un equipo.⇒ **Outcome d**
- C19. Capacidad para identificar eficazmente los objetivos y las prioridades de su trabajo / área / proyecto con indicación de la acción, el tiempo y los recursos necesarios.⇒ **Outcome j**
- CS6. Evaluar los sistemas en términos de atributos de calidad en general y las posibles ventajas y desventajas que se presentan en el problema dado.⇒ **Outcome c,i,m**
- CS7. Aplicar los principios de una gestión eficaz de la información, organización de la información, y las habilidades de recuperación de información a la información de diversos tipos, incluyendo texto, imágenes, sonido y vídeo. Esto debe incluir la gestión de los problemas de seguridad.⇒ **Outcome d,i,o**
- CS9. Identificar los riesgos (y esto incluye cualquier seguridad o los aspectos de seguridad) que pueden estar involucrados en la operación de equipo de cómputo dentro de un contexto dado.⇒ **Outcome c,d,m**

10. Lista de temas a estudiar en el curso

1. Evolución de Software
2. Gestión de Proyectos de Software
3. Gestión de Proyectos de Software
4. Procesos de Software
5. Estándares ISO/IEC

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Evolución de Software (12)	
Competencias esperadas: C7, C11, C12, CS6	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [PM14], [Som10]	

Unidad 2: Gestión de Proyectos de Software (10)	
Competencias esperadas: C18, C19, CS7, CS9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [PM14], [Som10]	

Unidad 3: Gestión de Proyectos de Software (8)	
Competencias esperadas: C18, C19, CS7, CS9	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [PM14], [Som10]	

Unidad 4: Procesos de Software (12)	
Competencias esperadas: C7, C13, C19, CS6, CS7	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • 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, <i>Plan-Do-Check-Act</i>, 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] 	<ul style="list-style-type: none"> • 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): <ul style="list-style-type: none"> – 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.
Lecturas : [PM14], [Som10]	

Unidad 5: Estándares ISO/IEC (6)	
Competencias esperadas: C7, C13, C19, CS6, CS7	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Aprender y aplciar correctamente normas y estandares internacionales. [Usar] 	<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.
Lecturas : [Som10], [PM14]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** BI0021. Bioinformática y Bioestadística (Obligatorio)
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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).
- (b) **Prerrequisitos:** CS2102. Análisis y Diseño de Algoritmos. (5^{to} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- l) Desarrollar principios investigación en el área de computación con niveles de competitividad internacional. (**Evaluar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a,b**
- C3.** Una comprensión intelectual de, y el aprecio por el papel central de los algoritmos y estructuras de datos.⇒ **Outcome b,l**
- C5.** Capacidad para implementar algoritmos y estructuras de datos en el software.⇒ **Outcome a,b**

10. Lista de temas a estudiar en el curso

1. Introducción a la Biología Molecular
2. Comparación de Secuencias
3. Árboles Filogenéticos
4. Ensamblaje de Secuencias de ADN
5. Estructuras secundarias y terciarias
6. Modelos Probabilísticos en Biología Molecular

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Introducción a la Biología Molecular (4)	
Competencias esperadas: CS1	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [CB00], [SM97]	

Unidad 2: Comparación de Secuencias (4)	
Competencias esperadas: CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [CB00], [SM97], [Pev00]	

Unidad 3: Árboles Filogenéticos (4)	
Competencias esperadas: CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [CB00], [SM97], [Pev00]	

Unidad 4: Ensamblaje de Secuencias de ADN (4)	
Competencias esperadas: CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Fundamento biológico: caso ideal, dificultades, métodos alternativos para secuenciación 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.
Lecturas : [SM97], [Alu06]	

Unidad 5: Estructuras secundarias y terciarias (4)	
Competencias esperadas: CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<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, Branch & Bound</i> para <i>protein threading</i>. • <i>Structural Alignment</i>: definiciones, algoritmo DALI
Lecturas : [SM97], [CB00], [Alu06]	

Unidad 6: Modelos Probabilísticos en Biología Molecular (4)	
Competencias esperadas: CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<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 Baum-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
Lecturas : [Dur+98], [CB00], [Alu06], [Kro+94]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** ET301. Formación de Empresas de Base Tecnológica II (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 2 HT; 2 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.

- (b) **Prerrequisitos:** ET201. Formación de Empresas de Base Tecnológica I. (8^{vo} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (Outcomes)

- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Usar**)

m) Transformar sus conocimientos del área de Ciencia de la Computación en emprendimientos tecnológicos. (**Evaluar**)

9. Competencias (IEEE)

C17. Capacidad para expresarse en los medios de comunicación orales y escritos como se espera de un graduado.⇒ **Outcome f**

C18. Capacidad para participar de forma activa y coordinada en un equipo.⇒ **Outcome d**

C19. Capacidad para identificar eficazmente los objetivos y las prioridades de su trabajo / área / proyecto con indicación de la acción, el tiempo y los recursos necesarios.⇒ **Outcome m**

C20. Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome m**

C21. Comprender el aspecto profesional, legal, seguridad, asuntos políticos, humanistas, ambientales, culturales y éticos.⇒ **Outcome m**

C22. Capacidad para demostrar las actitudes y prioridades que honrar, proteger y mejorar la estatura y la reputación ética de la profesión.⇒ **Outcome m**

C23. Capacidad para emprender, completar, y presentar un proyecto final.⇒ **Outcome m**

C24. Comprender la necesidad de la formación permanente y la mejora de habilidades y capacidades.⇒ **Outcome m**

10. Lista de temas a estudiar en el curso

- 1.
- 2.
- 3.

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: (20)	
Competencias esperadas: C19	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [BT06]	

Unidad 2: (30)	
Competencias esperadas: C20	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [KK06], [LW09]	

Unidad 3: (10)	
Competencias esperadas: C18	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [FUP96], [MM06]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3P02. Cloud Computing (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 1 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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>.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS3700. Big Data. (9^{no} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- g) Analizar el impacto local y global de la computación sobre los individuos, organizaciones y sociedad. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Usar**)

9. Competencias (IEEE)

- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome a**
- C4.** Una comprensión del hardware de la computadora desde la perspectiva del software, por ejemplo, el uso del procesador, memoria, unidades de disco, pantalla, etc.⇒ **Outcome b**
- C16.** Capacidad para identificar temas avanzados de computación y de la comprensión de las fronteras de la disciplina.⇒ **Outcome g,i**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome i**
- CS3.** Analizar el grado en que un sistema basado en el ordenador cumple con los criterios definidos para su uso actual y futuro desarrollo.⇒ **Outcome j**
- CS6.** Evaluar los sistemas en términos de atributos de calidad en general y las posibles ventajas y desventajas que se presentan en el problema dado.⇒ **Outcome g,j**

10. Lista de temas a estudiar en el curso

1. Sistemas distribuidos
2. Cloud Computing
3. Centros de Procesamiento de Datos
4. Cloud Computing
5. Cloud Computing
6. Modelos de Programación

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Sistemas distribuidos (15)	
Competencias esperadas: C2, C4	
Objetivos de Aprendizaje	Tópicos
<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] 	<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
Lecturas : [Cou+11]	

Unidad 2: Cloud Computing (15)	
Competencias esperadas: C2, C4	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Explicar el concepto de Cloud Computing. [Familiarizarse] • Listar algunas tecnologías relacionadas con Cloud Computing. [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 Cloud Computing. [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] 	<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
Lecturas : [HDF11], [BVS13]	

Unidad 3: Centros de Procesamiento de Datos (10)	
Competencias esperadas: C16	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Describir la evolución de los Data Centers. [Familiarizarse] • Esbozar la arquitectura de un data center en detalle. [Familiarizarse] • Indicar consideraciones de diseño y discutir su impacto. [Familiarizarse] 	<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.
Lecturas : [HDF11], [BVS13]	

Unidad 4: Cloud Computing (20)	
Competencias esperadas: CS2, CS3	
Objetivos de Aprendizaje	Tópicos
<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 cloud. [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 cloud. [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] 	<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.
Lecturas : [HDF11], [BVS13]	

Unidad 5: Cloud Computing (12)	
Competencias esperadas: CS2, CS3	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [HDF11], [BVS13]	

Unidad 6: Modelos de Programación (12)	
Competencias esperadas: CS6	
Objetivos de Aprendizaje	Tópicos
<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] 	<ul style="list-style-type: none"> • Visión global de los modelos de programación basados en cloud computing. • Modelo de Programación MapReduce. • Modelo de programación para aplicaciones basadas en Grafos.
Lecturas : [HDF11], [BVS13], [Low+12], [Mal+10], [Bal+08]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS3P03. Internet de las Cosas (Obligatorio)
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 1 HT; 4 HL; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del 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.
- (b) **Prerrequisitos:** CS3P01. Computación Paralela y Distribuída. (7^{mo} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- Que el alumno sea capaz de crear aplicaciones paralelas de mediana complejidad aprovechando eficientemente distintos dispositivos móviles.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Usar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Usar**)

9. Competencias (IEEE)

- C2.** Capacidad para tener una perspectiva crítica y creativa para identificar y resolver problemas utilizando el pensamiento computacional.⇒ **Outcome a**
- C4.** Una comprensión del hardware de la computadora desde la perspectiva del software, por ejemplo, el uso del procesador, memoria, unidades de disco, pantalla, etc.⇒ **Outcome b**
- C16.** Capacidad para identificar temas avanzados de computación y de la comprensión de las fronteras de la disciplina.⇒ **Outcome i**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome i**
- CS3.** Analizar el grado en que un sistema basado en el ordenador cumple con los criterios definidos para su uso actual y futuro desarrollo.⇒ **Outcome j**
- CS6.** Evaluar los sistemas en términos de atributos de calidad en general y las posibles ventajas y desventajas que se presentan en el problema dado.⇒ **Outcome j**

10. Lista de temas a estudiar en el curso

1. Fundamentos de paralelismo
2. Arquitecturas paralelas
3. Descomposición en paralelo
4. Comunicación y coordinación
5. Análisis y programación de algoritmos paralelos
6. Desempeño en paralelo

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Fundamentos de paralelismo (18)	
Competencias esperadas: C2	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [Pac11], [Mat14], [Qui03]	

Unidad 2: Arquitecturas paralelas (12)	
Competencias esperadas: C4	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [Pac11], [KH13], [SK10]	

Unidad 3: Descomposición en paralelo (18)	
Competencias esperadas: C16	
Objetivos de Aprendizaje	Tópicos
<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] 	<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)
Lecturas : [Pac11], [Mat14], [Qui03]	

Unidad 4: Comunicación y coordinación (18)	
Competencias esperadas: C16	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Usar exclusión mútua 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] 	<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"> – (Ciclicos) barreras, contadores y estructuras relacionadas • Acciones condicionales: <ul style="list-style-type: none"> – Espera condicional (p.e., empleando variables de condición)
Lecturas : [Pac11], [Mat14], [Qui03]	

Unidad 5: Análisis y programación de algoritmos paralelos (18)	
Competencias esperadas: CS2	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Mat14], [Qui03]	

Unidad 6: Desempeño en paralelo (18)	
Competencias esperadas: CS3	
Objetivos de Aprendizaje	Tópicos
<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] 	<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.
Lecturas : [Pac11], [Mat14], [KH13], [SK10]	



Universidad Nacional Mayor de San Marcos
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2018-II

1. **Código del curso y nombre:** CS4004. Proyecto de Final de Carrera III (Obligatorio)
2. **Créditos:** 6
3. **Horas de Teoría y Laboratorio:** 2 HT; 8 HP; (15 semanas)
4. **Docente(s)**

Atención previa coordinación con el profesor

5. Bibliografía

- [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.

6. Información del curso

- (a) **Breve descripción del curso** Este curso tiene por objetivo que el alumno logre finalizar adecuadamente su borrador de tesis.
- (b) **Prerrequisitos:** CS4003. Proyecto de Final de Carrera II. (9^{no} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. Objetivos específicos del curso.

- 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.

8. Contribución a los resultados (*Outcomes*)

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- c) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. (**Evaluar**)
- e) Entender correctamente las implicancias profesionales, éticas, legales, de seguridad y sociales de la profesión. (**Evaluar**)
- f) Comunicarse efectivamente con audiencias diversas. (**Evaluar**)

- h) Incorporarse a un proceso de aprendizaje profesional continuo. (**Evaluar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)
- l) Desarrollar principios investigación en el área de computación con niveles de competitividad internacional. (**Evaluar**)

9. Competencias (IEEE)

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a,b,c**
- C20.** Posibilidad de conectar la teoría y las habilidades aprendidas en la academia a los acontecimientos del mundo real que explican su pertinencia y utilidad.⇒ **Outcome e,f,g**
- CS2.** Identificar y analizar los criterios y especificaciones apropiadas a los problemas específicos, y planificar estrategias para su solución.⇒ **Outcome h,i,l**

10. Lista de temas a estudiar en el curso

1. Escritura del Borrador del trabajo de final de carrera (tesis)

11. Metodología y Evaluación

Metodología:

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.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:

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.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

12. Contenido

Unidad 1: Escritura del Borrador del trabajo de final de carrera (tesis) (60)	
Competences esperadas: C1,C20,CS2	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Parte experimental concluída (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] 	<ul style="list-style-type: none"> • Redacción y correccion del trabajo de final de carrera
Lecturas : [IEE08], [Ass08], [Cit08]	