

Universidad Nacional de San Agustín  
VICE RECTORADO ACADÉMICO  
SILABO

CODIGO DEL CURSO: CS100

<b>1 Datos Generales</b>	<b>FACULTAD :</b> Ingeniería de Producción y Servicios						
	<b>DEPARTAMENTO :</b> Ingeniería de Sistemas e Informática			<b>ESCUELA :</b> Ciencia de la Computación			
	<b>PROFESOR :</b>						
	<b>TÍTULO :</b>						
	<b>ASIGNATURA :</b> Introducción a la Ciencia de la Computación						
	<b>PREREQUISITO:</b> Ninguno		<b>CREDITOS:</b> 3		<b>Año:</b> 2010-1		<b>Total Horas:</b> 2 HT;
				<b>Sem:</b> 2 <sup>do</sup> Semestre.		2 HT    2 HP	
<b>Horario</b>		Lun	Mar	Mie	Jue	Vie	Sáb
<b>Total Semanal</b>							
<b>Aula</b>							

**2 Exposición de Motivos**      La Ciencia de la Computación es un campo de estudio enorme con muchas especialidades y aplicaciones. Este curso brindará a sus participantes, una visión panorámica de la informática y mostrará sus campos más relevantes: Algoritmos, Estructuras de Datos, Sistemas Operativos, Bases de Datos, etc.

**2 Objetivo**      ■ Brindar un panorama del área del conocimiento que es cubierta en la ciencia de la computación.

	<b>Objetivos Específicos</b>	<b>Contenidos</b>
<b>3 Contenido Temático</b> 3 DS/Lógica Básica.(2 horas)	<ul style="list-style-type: none"> <li>■ Aplicar métodos formales de lógica simbólica proposicional y de predicados.</li> <li>■ Describir como las herramientas formales de lógica simbólica son utilizadas para modelar algoritmos en situaciones reales.</li> <li>■ Usar demostraciones lógico-formales y razonamiento lógico para solucionar problemas tales como rompecabezas (<i>puzzles</i>).</li> <li>■ Describir la importancia y limitaciones de la lógica de predicados.</li> </ul>	<ul style="list-style-type: none"> <li>■ Lógica proposicional</li> <li>■ Conectivos lógicos</li> <li>■ Tablas de verdad</li> <li>■ Formas normales (conjuntiva y disyuntiva).</li> <li>■ Validación.</li> <li>■ Lógica de predicados</li> <li>■ Cuantificación universal y existencial.</li> <li>■ <i>Modus ponens</i> y <i>modus tollens</i></li> <li>■ Limitaciones de la lógica de predicados.</li> </ul>
		[1]

**3 PF/Construcciones fundamentales.(2 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Analizar y explicar el comportamiento de programas simples involucrando las estructuras de programación fundamental cubiertas por esta unidad.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Sintaxis básica y semántica de lenguaje de más alto nivel.</li> <li>▪ Variables, tipos, expresiones y declaraciones.</li> <li>▪ Descomposición estructural de programas.</li> </ul> <p style="text-align: right;">[1]</p>

**3 PF/Algoritmos y Resolución de Problemas.(3 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Discutir la importancia de los algoritmos en el proceso de solución de problemas.</li> <li>▪ Identificar las propiedades necesarias de un buen algoritmo.</li> <li>▪ Crear algoritmos para resolver problemas simples.</li> <li>▪ Usar pseudocódigo o un lenguaje de programación para implementar, probar y depurar algoritmos para resolver problemas simples.</li> <li>▪ Describir estrategias útiles para depuración.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Estrategias para la resolución de problemas.</li> <li>▪ El rol de los algoritmos en el proceso de solución de problemas.</li> <li>▪ Estrategias de diseño de algoritmos.</li> <li>▪ Estrategias de implementación de algoritmos.</li> <li>▪ El Concepto y tipos de algoritmos.</li> </ul> <p style="text-align: right;">[1]</p>

	Objetivos Específicos	Contenidos
3 PF/Estructuras de Datos.(2 horas)	<ul style="list-style-type: none"> <li>▪ Describir la representación de datos numéricos y de caracteres</li> <li>▪ Entender como la precisión y el redondeo puede afectar los cálculos numéricos.</li> <li>▪ Discutir la representación y uso de tipos de datos primitivos y estructuras de datos incorporadas en el lenguaje.</li> <li>▪ Describir aplicaciones comunes para cada estructura de datos en la lista de temas.</li> <li>▪ Implementar estructuras de datos definidas por el usuario en un lenguaje de alto nivel.</li> <li>▪ Comparar implementaciones alternativas de estructuras de datos considerando su desempeño.</li> <li>▪ Escribir programas que usan cada una de las siguientes estructuras de datos: arreglos, registros, cadenas, listas enlazadas, pilas, colas y tablas de <i>hash</i>.</li> <li>▪ Comparar y contrastar los costos y beneficios de las implementaciones dinámicas y estáticas de las estructuras de datos.</li> <li>▪ Escoger la estructura de datos apropiada para modelar un problema dado.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Representación de datos numérico</li> <li>▪ Rango, precisión y errores de redondeo.</li> <li>▪ Arreglos.</li> <li>▪ Registros.</li> <li>▪ Cadenas y procesamiento de cadenas.</li> <li>▪ Representación de caracteres.</li> <li>▪ Administración del almacenamiento en tiempo de ejecución.</li> <li>▪ Punteros y referencias.</li> <li>▪ Estructuras enlazadas.</li> <li>▪ Estrategias de implementación para pilas, colas y tablas <i>hash</i>.</li> <li>▪ Estrategias de implementación para grafos y árboles.</li> <li>▪ Estrategias para escoger la estructura de datos correcta.</li> </ul> <p>[1]</p>

	Objetivos Específicos	Contenidos
3 AL/Análisis Básico de Algoritmos.(2 horas)	<ul style="list-style-type: none"> <li>▪ Determinar la complejidad de tiempo y espacio de algoritmos simples.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Identificar la diferencia de comportamiento entre el mejor y peor caso.</li> </ul> <p>[1]</p>

	Objetivos Específicos	Contenidos
3 AL/Estrategias Algorítmicas.(2 horas)	<ul style="list-style-type: none"> <li>▪ Describir las desventajas de los algoritmos de fuerza bruta.</li> <li>▪ Implementar un algoritmo de divide y vencerás para solucionar apropiadamente un problema.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Algoritmos de fuerza bruta (<i>force</i>).</li> <li>▪ Divide y vencerás.</li> </ul> <p>[1]</p>

**3 AR/Lógica Digital y Representación de Datos.(2 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"><li>▪ Diseñar un circuito simple usando los bloques de construcción fundamentales.</li><li>▪ Aprender el efecto de las operaciones AND, OR, NOT y XOR en datos binarios.</li><li>▪ Entender como números, texto, imágenes y sonido pueden ser representados en forma digital y discutir las limitaciones en cada representación.</li><li>▪ Entender los errores debido a los efectos de redondeo y como su propagación afecta la precisión de cálculos encadenados.</li><li>▪ Aprender como los datos pueden ser comprimidos para reducir los requerimientos de almacenamiento incluyendo el concepto de pérdida de información debido a la compresión.</li></ul>	<ul style="list-style-type: none"><li>▪ Introducción (compuertas lógicas).</li><li>▪ Expresiones booleanas.</li><li>▪ Representación de números.</li><li>▪ Aritmética de números binarios.</li><li>▪ Rango, precisión y error en la representación de números.</li><li>▪ Representación de texto e imágenes.</li><li>▪ Compresión de datos.</li></ul> <p>[1]</p>

3 AR/Arquitectura y Organización de Computadores.(2 horas)

Objetivos Específicos	Contenido
<ul style="list-style-type: none"> <li>▪ Describir el progreso de las computadoras desde los tubos de vacío hasta la <i>Very Large Scale Integration</i> (VLSI).</li> <li>▪ Appreciar el conjunto de instrucciones de la arquitectura, <i>Industry Standard Architecture</i> (ISA) y la naturaleza de instrucciones a nivel de máquina en términos de su funcionalidad y uso de recursos (registros y memoria).</li> <li>▪ Entender la relación entre el conjunto de instrucciones de arquitectura, microarquitectura y arquitectura del sistema así como sus roles en el desarrollo de la computadora.</li> <li>▪ Prestar atención a las varias clases de instrucciones: movimiento de datos, aritmética, lógica y control de flujo.</li> <li>▪ Appreciar la diferencia entre ISAs registro-a-memoria e ISAs de carga/almacenamiento.</li> <li>▪ Appreciar cómo las operaciones condicionales están implementadas a nivel de máquina.</li> <li>▪ Entender la forma en la cual se ejecuta el llamado y retorno de subrutinas.</li> <li>▪ Appreciar cómo la falta de recursos en Proveedores de Servicios de Internet (<i>Internet Service Providers-ISP</i>) tiene un impacto en los lenguajes de alto nivel y en el diseño de compiladores.</li> <li>▪ Entender cómo, a nivel de lenguaje ensamblador, los parámetros son pasados a las subrutinas y cómo se crea y accede un ambiente de trabajo local.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Visión de las cosas</li> <li>▪ Introducción a la arquitectura de máquina.</li> <li>▪ Arquitectura de instrucciones</li> <li>▪ Estructura de la computadora y carga</li> <li>▪ Seguridad de la computadora</li> <li>▪ Estructura de la máquina</li> <li>▪ Limitaciones de la máquina</li> <li>▪ Soporte de programas</li> </ul> <p>[1]</p>

**3 AR/Arquitectura de Memoria.(2 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Identificar los tipos principales de la tecnología de memoria.</li> <li>▪ Apreciar la necesidad de estándares de almacenamiento para mecanismos complejos de almacenamiento de datos tales como un DVD.</li> <li>▪ Entender porque la jerarquía de memorias es necesaria para reducir la latencia efectiva de la memoria.</li> <li>▪ Apreciar que la mayoría de datos en el bus de memoria de debe a a tráfico de recarga en la memoria cache.</li> <li>▪ Describir las varias formas de organizar la memoria cache y apreciar el punto de equilibrio entre costo y desempeño para cada configuración.</li> <li>▪ Apreciar la necesidad de la coherencia de la memoria cache en sistemas de múltiples procesadores.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Sistemas de almacenamiento tecnología (semiconductores néticos).</li> <li>▪ Estándares de almacenamiento (CD-ROM, DVD, Blue-Ray)</li> <li>▪ Jerarquía de memoria, latencia y rendimiento (<i>throughput</i>).</li> <li>▪ Memorias cache, principios de operación, políticas de recarga de cache multinivel.</li> </ul> <p>[1]</p>

**3 AR/Interfases y Estrategias de I/O.(2 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Apreciar la necesidad de comunicaciones <i>open-loop</i> y <i>closed-loop</i> y el uso de buffer para el control de flujo de datos.</li> <li>▪ Explicar como las interrupciones son utilizadas para implementar controles de I/O y transferencia de datos.</li> <li>▪ Identificar varios tipos de buses en un sistema de computadoras y entender como los dispositivos compiten y ganan el acceso al bus.</li> <li>▪ Prestar atención al progreso de la tecnología de buses y entender las características y el desempeño de un conjunto de buses modernos (seriales y paralelos).</li> </ul>	<ul style="list-style-type: none"> <li>▪ Fundamentos de entrada y salida y protocolos de inicio de transmisión (<i>handshaking</i>) y <i>buffering</i>.</li> <li>▪ Mecanismos de interrupción de vector y con priorización de interrupción.</li> <li>▪ Buses: protocolos de arbitraje, Acceso directo (DMA).</li> <li>▪ Buses modernos: <i>Peripheral Interconnect Express</i>, USB, <i>Hypertransport</i>.</li> </ul> <p>[1]</p>

**3 OS/Visión General de los Sistemas Operativos.(2 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Explicar los objetivos y funciones de los sistemas operativos modernos.</li> <li>▪ Describir como los sistemas operativos han evolucionado en el tiempo desde sistemas primitivos <i>batch</i> a sofisticados sistemas multiusuarios.</li> <li>▪ Analizar las ventajas y desventajas inherentes en el diseño de sistemas operativos.</li> <li>▪ Describir las funciones de un sistema operativo contemporáneo con respecto a la conveniencia, eficiencia y habilidad para evolucionar.</li> <li>▪ Discutir sistemas operativos de tipos distribuido, para redes y cliente-servidor y como ellos difieren de un sistema operativo para un único usuario.</li> <li>▪ Identificar las amenazas potenciales a sistemas operativos y el diseño de características de seguridad para resguardarlos.</li> <li>▪ Describir como los temas tales como el software de código abierto y el incremento del uso de Internet están influyendo el diseño de sistemas operativos.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Rol y propósitos operativos.</li> <li>▪ Historia de sistemas operativos.</li> <li>▪ Funcionalidad típica.</li> <li>▪ Mecanismos cliente-servidor <i>held</i>.</li> <li>▪ Asuntos de seguridad, flexibilidad, confiabilidad, compatibilidad, etc.</li> <li>▪ Influencias de la multimedia.</li> </ul> <p>[1]</p>

**3 NC/Introducción.(2 horas)**

Objetivos Específicos	Contenidos	Horas
<ul style="list-style-type: none"> <li>▪ Discutir la evolución de las primeras redes y de la Internet.</li> <li>▪ Demostrar la habilidad para usar efectivamente un conjunto de aplicaciones de red incluyendo e-mail, telnet, FTP, wikis, navegadores web, cursos en línea y mensajería instantánea.</li> <li>▪ Explicar la estructura por capas jerárquica de una arquitectura de red típica.</li> <li>▪ Describir las tecnologías emergentes en el área de la computación centrada en redes, evaluar sus actuales capacidades, limitaciones y su potencial a corto plazo.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Historia de las redes y de la Internet.</li> <li>▪ Arquitecturas de redes.</li> <li>▪ Especializaciones dentro de la computación centrada en redes.</li> <li>▪ Redes y protocolos.</li> <li>▪ Sistemas Multimedia en redes.</li> <li>▪ Computación distribuida.</li> <li>▪ Paradigmas cliente/servidor y <i>Peer-to-Peer</i>.</li> <li>▪ Computación móvil e inalámbrica.</li> </ul> <p>[1]</p>	

**3 PL/Visión General de los Lenguajes de Programación.(2 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Listar la evolución de los lenguajes de programación identificando como es que su historia nos ha conducido a los paradigmas actuales.</li> <li>▪ Identificar al menos una característica distintiva para cada uno de los paradigmas de programación cubiertos en esta unidad.</li> <li>▪ Evaluar las ventajas y desventajas entre los diferentes paradigmas, considerando temas tales como: eficiencia de espacio, eficiencia en el tiempo (para ambas partes computadora y programador), seguridad y el poder de las expresiones.</li> <li>▪ Distinguir entre la programación a menor y mayor escala.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Hi...</li> <li>ma</li> <li>▪ Br...</li> <li>pr</li> <li>▪ Le...</li> <li>▪ Le...</li> <li>▪ Le...</li> <li>mi</li> <li>▪ Le...</li> <li>Lo</li> <li>me</li> </ul> <p>[1]</p>

**3 PL/Máquinas Virtuales.(2 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Describir la importancia y poder de la abstracción en el contexto de máquinas virtuales.</li> <li>▪ Explicar los beneficios de los lenguajes intermedios en el proceso de compilación.</li> <li>▪ Evaluar las ventajas y desventajas entre desempeño vs. portabilidad.</li> <li>▪ Explicar como los programas ejecutables pueden violar la seguridad de sistema computacional accediendo a archivos de disco y memoria.</li> </ul>	<ul style="list-style-type: none"> <li>▪ El concepto de máquina virtual.</li> <li>▪ Jerarquías de las máquinas virtuales.</li> <li>▪ Lenguajes intermedios.</li> <li>▪ Temas de seguridad relacionados a ejecutar código sobre una máquina externa.</li> </ul> <p>[1]</p>



3 PL/Programación Orientada a Objetos.(1 horas)

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Justificar la filosofía de diseño orientado a objetos y los conceptos de encapsulación, abstracción, herencia y polimorfismo.</li> <li>▪ Diseñar, implementar, probar y depurar programas simples en un lenguaje de programación orientado a objetos.</li> <li>▪ Describir como los mecanismos de clases soportan encapsulación y ocultamiento de la información.</li> <li>▪ Diseñar, implementar y probar la implementación de la relación es-un <i>IsKindOf</i> entre objetos usando jerarquía de clases y herencia.</li> <li>▪ Comparar y contrastar las nociones de sobrecarga y sobreescritura de métodos en un lenguaje de programación.</li> <li>▪ Explicar la relación entre la estructura estática de una clase y la estructura dinámica de las instancias de dicha clases.</li> <li>▪ Describir como los iteradores acceden a los elementos de un contenedor.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Diseño orientado a objetos.</li> <li>▪ Encapsulación y ocultamiento de la información.</li> <li>▪ Separación de preocupaciones y implementación.</li> <li>▪ Clases y subclases.</li> <li>▪ Herencia (sobreescripción dinámica).</li> <li>▪ Polimorfismo (polimorfismo de tipo vs. herencia).</li> <li>▪ Jerarquías de clases.</li> <li>▪ Clases de tipo colección y de iteración.</li> <li>▪ Representaciones internas de los datos y tablas de métodos.</li> </ul> <p>[1]</p>

**3 PL/Sistemas de Traducción del Lenguaje.(1 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Describir los pasos y algoritmos usados por traductores lenguajes.</li> <li>▪ Reconocer los modelos formales subyacentes tales como los autómatas finitos, autómatas de pila y su conexión con la definición del lenguaje a través de expresiones regulares y gramáticas.</li> <li>▪ Discutir la efectividad de la optimización.</li> <li>▪ Explicar el impacto de la facilidad de la compilación separada y la existencia de librerías de programas en el proceso de compilación.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Aplicación de las técnicas en analizadores.</li> <li>▪ Análisis sintáctico (concreta y abstracta, abstracta).</li> <li>▪ Aplicación de las técnicas de contexto en compiladores por tablas o recursión.</li> <li>▪ Administración de recursos.</li> <li>▪ Generación de código de máquina de un árbol de parseo.</li> <li>▪ Operaciones especiales de selección: selección de registros, asignación de registros.</li> <li>▪ Técnicas de optimización.</li> <li>▪ El uso de herramientas de desarrollo en el proceso de compilación y sus ventajas de éste.</li> <li>▪ Librerías de programación separada.</li> <li>▪ Construcción de compiladores dirigidos por la sintaxis.</li> </ul> <p>[1]</p>

**3 HC/Fundamentos de la Interacción Hombre-Computador (HCI)(2 horas)**

Objetivos Específicos
<ul style="list-style-type: none"> <li>▪ Discutir las razones por las cuales es importante el desarrollo de software centrado en el usuario.</li> <li>▪ Explicar porqué los modelos humanos individuales y los modelos sociales son importantes a la hora de diseñar la Interacción Humano-Computador.</li> </ul>

	<b>Objetivos Específicos</b>	<b>Contenidos</b>
<b>3 IS/Tópicos Fundamentales en Sistemas Inteligentes.(2 horas)</b>	<ul style="list-style-type: none"> <li>▪ Describir la prueba de Turing y el experimento de pensamiento del “Cuarto Chino”.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Historia</li> <li>▪ Cuestiones</li> <li>▪ La programación</li> </ul>
	<b>Objetivos Específicos</b>	<b>Contenidos</b>
<b>3 IM/Sistemas de Base de Datos.(2 horas)</b>	<ul style="list-style-type: none"> <li>▪ Explicar las características que distinguen a las bases de datos de los métodos tradicionales de programación con archivos de datos.</li> <li>▪ Citar el objetivo, funciones, modelos, componentes, aplicaciones y el impacto social de los sistemas de bases de datos.</li> <li>▪ Describir los componentes de un sistema de base de datos y dar ejemplos de su uso.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Historia y motivación de los sistemas de base de datos.</li> <li>▪ Componentes de los sistemas de base de datos.</li> </ul>
	<b>Objetivos Específicos</b>	<b>Contenidos</b>
<b>3 IM/Modelamiento de Datos.(2 horas)</b>	<ul style="list-style-type: none"> <li>▪ Describir los conceptos de modelado y la notación del modelo entidad-relación y UML, incluyendo su uso en modelamiento de datos.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Modelamiento de datos.</li> <li>▪ Modelos conceptuales (incluyendo entidad-relación y UML).</li> </ul>
	<b>Objetivos Específicos</b>	<b>Contenidos</b>
<b>3 IM/Base de Datos Relacionales.(2 horas)</b>	<ul style="list-style-type: none"> <li>▪ Demostrar consultas en el álgebra relacional.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Álgebra relacional y cálculo relacional.</li> </ul>

**3 SP/Contexto Social de la Computación.(4 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Interpretar el contexto social de una implementación particular.</li> <li>▪ Identificar suposiciones y valores insertados en un diseño particular incluyendo aquellos de naturaleza cultural.</li> <li>▪ Evaluar una implementación particular a través del uso de datos empíricos.</li> <li>▪ Describir las formas positivas o negativas en las cuales la computación altera los modos de interacción entre las personas.</li> <li>▪ Explicar por qué el acceso a redes de computadores y computadoras es restringido en algunos países.</li> <li>▪ Indicar el rol de los temas culturales para el trabajo en equipo.</li> <li>▪ Analizar el rol y riesgos de la introducción de la computación en políticas públicas y gobierno: por ejemplo voto electrónico.</li> <li>▪ Articular el impacto del deficit de profesionales en computación.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Introducción a las sociales de la computación.</li> <li>▪ Implicaciones sociales de comunicación.</li> <li>▪ Crecimiento, control Internet.</li> <li>▪ Temas relacionados</li> <li>▪ Asuntos culturales</li> <li>▪ Temas internacionales</li> <li>▪ Accesibilidad: baja de minorías, mujeres, discapacidad en computación.</li> <li>▪ Asuntos de política ejemplo: voto electrónico</li> </ul> <p>[1]</p>

**3 SP/Propiedad Intelectual.(2 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Distinguir entre patentes, <i>copyright</i> y protección de secretos del negocio.</li> <li>▪ Discutir el fondo legal del <i>copyright</i> en las leyes nacionales e internacionales.</li> <li>▪ Explicar como las leyes de patentes y el <i>copyright</i> pueden variar internacionalmente.</li> <li>▪ Delinear el desarrollo histórico de las patentes de software.</li> <li>▪ Discutir las consecuencias de la piratería de software sobre los desarrolladores de software y el rol de las organizaciones de soporte relevante.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Fundamentos de la propiedad intelectual.</li> <li>▪ <i>Copyrights</i>, patentes y secretos de negocio.</li> <li>▪ Piratería de software.</li> <li>▪ Patentes de software.</li> <li>▪ Asuntos transnacionales concernientes a la propiedad intelectual.</li> </ul> <p>[1]</p>

	<b>Objetivos Específicos</b>	<b>Contenidos</b>	
<b>3 SE/Diseño de Software.(2 horas)</b>	<ul style="list-style-type: none"> <li>Discutir las propiedades del buen diseño de software incluyendo la naturaleza y el rol de la documentación asociada.</li> </ul>	<ul style="list-style-type: none"> <li>Conceptos y principios fundamentales de diseño.</li> </ul>	[1]
	<b>Objetivos Específicos</b>	<b>Contenidos</b>	<b>Horas</b>
<b>3 SE/Usando APIs.(2 horas)</b>	<ul style="list-style-type: none"> <li>Explicar el valor de las interfaces para programación de aplicaciones (APIs) en el desarrollo de software.</li> </ul>	<ul style="list-style-type: none"> <li>Programación usando API.</li> </ul>	[1]
	<b>Objetivos Específicos</b>	<b>Contenidos</b>	
<b>3 SE/Herramientas y Entornos de Software.(2 horas)</b>	<ul style="list-style-type: none"> <li>Analizar y evaluar un conjunto de herramientas en una área dada del desarrollo de software (ej: administración, modelamiento o pruebas).</li> </ul>	<ul style="list-style-type: none"> <li>Entornos de programación.</li> </ul>	[1]
	<b>Objetivos Específicos</b>	<b>Contenidos</b>	
<b>3 SE/Procesos de Software.(2 horas)</b>	<ul style="list-style-type: none"> <li>Explicar el ciclo de vida del software y sus fases incluyendo las entregas que son producidas (dar un ejemplo concreto).</li> </ul>	<ul style="list-style-type: none"> <li>Ciclo de vida del software y modelos de procesos.</li> </ul>	[1]

#### 4 Actividades

- Asignaciones
- Controles de Lectura
- Exposiciones

#### 5 Recursos Materiales

- Apuntes del curso
- Libro(s) de la bibliografía

#### 6 Metodología

- Clase Magistral.
- Taller didáctico.
- Social Constructivismo.
- Prácticas personales y en grupo.

#### 7 Evaluación

La nota final (*NF*) se obtiene de la siguiente manera:

**NE** Nota de Exámenes 60 %, esta nota se divide en

- Exámen Parcial 40 %
- Examen Final 60 %

**NT** Nota de Trabajos e Intervención en clase 40 %

$$NF = 0,6 * NE + 0,4 * NT$$

## Referencias

- [1] J. G. Brookshear. *Computer Science: An Overview*. Addison-Wesley, 11th edition, January 2012. ISBN 10: 0-13-256903-5.

---

Docente del curso