

**Universidad Católica San Pablo**  
**Facultad de Ingeniería y Computación**  
**Escuela Profesional de**  
**Ciencia de la Computación**  
**SILABO**



**CS102O. Objetos y Abstracción de Datos (Obligatorio)**

2016-2

**1. DATOS GENERALES**

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS102O. Objetos y Abstracción de Datos
1.3 SEMESTRE ACADÉMICO	:	3 <sup>er</sup> Semestre.
1.4 PREREQUISITO(S)	:	CS101O. Introducción a la Programación Orientada a Objetos. (2 <sup>do</sup> Sem)
1.5 CARÁCTER	:	Obligatorio
1.6 HORAS	:	2 HT; 2 HP; 2 HL;
1.7 CRÉDITOS	:	4

**2. DOCENTE**

Mg. Gustavo Delgado Ugarte

- Mag. Ingeniería del Software, Escuela Universitaria de Ingeniería Industrial, Informática y Sistemas - UTA, Chile, 2009.
- Prof. Ingeniero de Sistemas, UCSM, Perú, 2006.

**3. FUNDAMENTACIÓN 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 *functional-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.

**4. SUMILLA**

1. DS/Gráfos y Árboles.2. PF/Construcciones fundamentales.3. PF/Algoritmos y Resolución de Problemas.4. PF/Programación Orientada a Eventos.5. AL/Análisis Básico de Algoritmos.6. AL/Algoritmos Fundamentales.7. PL/Declaración y Tipos.8. PL/Mecanismos de Abstracción.9. PL/Programación Orientada a Objetos.10. SE/Diseño de Software.11. SE/Usando APIs.12. SE/Especificación de Requerimientos.

**5. OBJETIVO GENERAL**

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

## 6. CONTRIBUCIÓN A LA FORMACIÓN PROFESIONAL Y FORMACIÓN GENERAL

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- ) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. [**Nivel Bloom: 3**]
- ) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. [**Nivel Bloom: 3**]
- ) Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas. [**Nivel Bloom: 3**]
- ) Trabajar efectivamente en equipos para cumplir con un objetivo común. [**Nivel Bloom: 3**]
- ) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. [**Nivel Bloom: 3**]
- ) Aplicar los principios de desarrollo y diseño en la construcción de sistemas de software de complejidad variable. [**Nivel Bloom: 3**]

## 7. CONTENIDOS

### UNIDAD 1: DS/Gráfos y Árboles.(7 horas)

**Nivel Bloom: 3**

#### OBJETIVO GENERAL

- Ilustrar con ejemplos la terminología básica de teoría de grafos y algunas de las propiedades y casos especiales de cada una.
- Mostrar diferentes métodos de recorrido en árboles y grafos.
- Modelar problemas en Ciencias de la Computación usando grafos y árboles.
- Relacionar grafos y árboles con estructura de datos, algoritmos y conteo.

#### CONTENIDO

- Árboles.
- Grafos no dirigidos.
- Grafos dirigidos.
- Árboles de expansión.
- Estrategias de recorrido.

**Lecturas:** [Nakariakov, 2013]

<b>UNIDAD 2: PF/Construcciones fundamentales.(5 horas)</b>	
<b>Nivel Bloom: 3</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
<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> <li>▪ Modificar y extender programas cortos que usan condicionales estándar, estructuras de control iterativas y funciones.</li> <li>▪ Diseñar, implementar, probar y depurar un programa que use cada una de las siguientes estructuras fundamentales de programación: cálculos básicos, entrada y salida simple, estructuras estándar condicionales e iterativas y definición de funciones.</li> <li>▪ Escoger la estructura apropiada condicional e iterativa para una estructura de programación dada.</li> <li>▪ Aplicar técnicas de descomposición estructurada o funcional para dividir un programa en pequeñas partes.</li> <li>▪ Describir los mecanismos de paso de parámetros.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Sintaxis básica y semántica de un lenguaje de más alto nivel.</li> <li>▪ Variables, tipos, expresiones y asignaciones.</li> <li>▪ Entrada y salida simple.</li> <li>▪ Estructuras de control condicionales e iterativas.</li> <li>▪ Funciones y paso de parámetros.</li> <li>▪ Descomposición estructurada.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013]	

<b>UNIDAD 3: PF/Algoritmos y Resolución de Problemas.(5 horas)</b>	
<b>Nivel Bloom: 4</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
<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 solución de problemas.</li> <li>▪ El rol de los algoritmos en el proceso de solución de problemas.</li> <li>▪ Estrategias de implementación para algoritmos.</li> <li>▪ Estrategias de depuración.</li> <li>▪ El Concepto y propiedades de algoritmos.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013]	

<b>UNIDAD 4: PF/Programación Orientada a Eventos.(2 horas)</b>	
<b>Nivel Bloom: 3</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
<ul style="list-style-type: none"> <li>▪ Explicar la diferencia entre programación orientada a eventos y programación por línea de comandos.</li> <li>▪ Diseñar, codificar, probar y depurar programas de manejo de eventos simples que respondan a eventos del usuario.</li> <li>▪ Desarrollar código que responda a las condiciones de excepción lanzadas durante la ejecución.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Métodos para la manipulación de eventos.</li> <li>▪ Propagación de eventos.</li> <li>▪ Manejo de excepciones.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013]	

<b>UNIDAD 5: AL/Análisis Básico de Algoritmos.(3 horas)</b>	
<b>Nivel Bloom: 2</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
<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>▪ Análisis asintótico de límites en los casos promedio y superior.</li> <li>▪ Identificar la diferencias entre el comportamiento entre el mejor, mediano y peor caso.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013]	

UNIDAD 6: AL/Algoritmos Fundamentales.(3 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> <li>▪ Implementar los algoritmos cuadráticos más comunes y los algoritmos de ordenamiento <math>O(N\log N)</math>.</li> <li>▪ Diseñar e implementar una función de (<i>hash</i>) apropiada para una aplicación.</li> <li>▪ Diseñar e implementar un algoritmo de resolución de colisiones para tablas de <i>hash</i>.</li> <li>▪ Discutir la eficiencia computacional de los principales algoritmos de ordenamiento, búsqueda y (<i>hashing</i>).</li> <li>▪ Discutir otros factores, además de la eficiencia computacional, que influyen en la elección de los algoritmos, tales como tiempo de programación, mantenimiento y el uso de patrones específicos de aplicación en los datos de entrada.</li> <li>▪ Resolver problemas usando los algoritmos de grafos fundamentales, incluyendo búsqueda por amplitud y profundidad; caminos más cortos con uno y múltiples orígenes, cerradura transitiva, ordenamiento topológico y al menos un algoritmo de árbol de expansión mínima.</li> <li>▪ Demostrar las siguientes capacidades: evaluar algoritmos, seleccionar una opción de un rango posible, proveer una justificación para tal elección e implementar el algoritmo..</li> </ul>	<ul style="list-style-type: none"> <li>▪ Algoritmos numéricos simples.</li> <li>▪ Búsqueda secuencial y binaria.</li> <li>▪ Algoritmos cuadráticos de ordenamiento (selección, inserción).</li> <li>▪ Algoritmos de tipo <math>O(N\log N)</math> (Quicksort, heapsort, mergesort).</li> <li>▪ Tablas de (<i>hash</i>) incluyendo estrategias de solución para las colisiones.</li> <li>▪ Árboles de búsqueda binaria.</li> <li>▪ Representación de grafos (Listas y Matrices de adyacencia).</li> <li>▪ Recorridos por amplitud y profundidad.</li> <li>▪ El algoritmo del camino más corto (algoritmos de Dijkstra y Floyd).</li> <li>▪ Cerradura transitiva (algoritmo de Floyd).</li> <li>▪ Árbol de expansión mínima (algoritmos de Kruskal y Prim).</li> <li>▪ Ordenamiento Topológico.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013]	

<b>UNIDAD 7: PL/Declaración y Tipos.(2 horas)</b>	
<b>Nivel Bloom: 3</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
<ul style="list-style-type: none"> <li>▪ Explicar el valor de los modelos de declaración, especialmente con respecto a la programación en mayor escala.</li> <li>▪ Identificar y describir las propiedades de una variable, tales como su: dirección asociada, valor, ámbito, persistencia y tamaño.</li> <li>▪ Discutir la incompatibilidad de tipos.</li> <li>▪ Demostrar las diferentes formas de enlace, visibilidad, ámbito y manejo del tiempo de vida.</li> <li>▪ Defender la importancia de los tipos y el chequeo de tipos para brindar abstracción y seguridad.</li> <li>▪ Evaluar las ventajas y desventajas en el manejo del tiempo de vida (conteo por referencia vs. recolección de basura).</li> </ul>	<ul style="list-style-type: none"> <li>▪ La concepción de tipos como un conjunto de valores unidos a un conjunto de operaciones.</li> <li>▪ Declaración de modelos (enlace, visibilidad, alcance y tiempo de vida).</li> <li>▪ Vista general del chequeo de tipos.</li> <li>▪ Recolección de basura.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013]	

<b>UNIDAD 8: PL/Mecanismos de Abstracción.(5 horas)</b>	
<b>Nivel Bloom: 3</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
	<ul style="list-style-type: none"> <li>▪ Procedimientos, funciones e iteradores como mecanismos de abstracción.</li> <li>▪ Mecanismos de parametrización (referencia vs. valor).</li> <li>▪ Registros de activación y administración de almacenamiento.</li> <li>▪ Tipos de parámetros y tipos parametrizados.</li> <li>▪ Módulos en lenguajes de programación.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013]	

UNIDAD 9: PL/Programación Orientada a Objetos.(7 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<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 sobrescritura 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 comportamiento e implementación.</li> <li>▪ Clases y subclasses.</li> <li>▪ Herencia (sobrescritura, despacho dinámico).</li> <li>▪ Polimorfismo (polimorfismo de subtipo vs. herencia).</li> <li>▪ Jerarquías de clases.</li> <li>▪ Clases de tipo colección y protocolos de iteración.</li> <li>▪ Representaciones internas de objetos y tablas de métodos.</li> </ul>
Lecturas: [Stroustrup, 2013]	
UNIDAD 10: SE/Diseño de Software.(5 horas)	
Nivel Bloom: 2	
OBJETIVO GENERAL	CONTENIDO
<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> <li>▪ Conducir una revisión de diseño de software con material de código abierto utilizando lineamientos apropiados.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Conceptos y principios fundamentales de diseño.</li> <li>▪ El rol y uso de contratos.</li> <li>▪ Patrones de diseño.</li> </ul>
Lecturas: [Stroustrup, 2013]	

<b>UNIDAD 11: SE/Usando APIs.(1 horas)</b>	
<b>Nivel Bloom: 2</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</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> <li>▪ Usar navegadores de clases y herramientas relacionadas durante el desarrollo de aplicaciones usando APIs.</li> <li>▪ Diseñar, implementar, probar y depurar programas que usan paquetes API de larga escala.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Programación usando API.</li> <li>▪ Diseño de API.</li> <li>▪ Navegadores de clases (<i>Class browsers</i>) y herramientas relacionadas.</li> <li>▪ Depuración en el entorno API.</li> <li>▪ Introducción a la computación basada en componentes.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013]	

<b>UNIDAD 12: SE/Especificación de Requerimientos.(1 horas)</b>	
<b>Nivel Bloom: 2</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
<ul style="list-style-type: none"> <li>▪ Discutir los retos de mantener software heredado.</li> <li>▪ Usar un método común, no formal para modelar y especificar (en la forma de un documento de especificación de requerimientos) los requerimientos para un sistema de software de tamaño medio.</li> <li>▪ Traducir en lenguaje natural una especificación de requerimientos de software escrita en un lenguaje de especificación formal comunmente usado.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Técnicas de modelamiento del análisis de requerimientos.</li> <li>▪ Prototipeo.</li> <li>▪ Conceptos básicos de técnicas de especificación formal.</li> </ul>
<b>Lecturas:</b> [Stroustrup, 2013]	

<b>8. METODOLOGÍA</b>
<p>El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.</p> <p>El profesor del curso presentará demostraciones para fundamentar clases teóricas.</p> <p>El profesor y los alumnos realizarán prácticas</p> <p>Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.</p>

<b>9. EVALUACIONES</b>
<p><b>Evaluación Permanente 1 : 20 %</b></p> <p><b>Examen Parcial : 30 %</b></p> <p><b>Evaluación Permanente 2 : 20 %</b></p> <p><b>Examen Final : 30 %</b></p>



## Referencias

[Nakariakov, 2013] Nakariakov, S. (2013). *The Boost C++ Libraries: Generic Programming*. CreateSpace Independent Publishing Platform.

[Stroustrup, 2013] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley, 4th edition.