

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



**CS211T. Teoría de la Computación (Obligatorio)**

2010-1

**1. DATOS GENERALES**

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS211T. Teoría de la Computación
1.3 SEMESTRE ACADÉMICO	:	4 <sup>to</sup> Semestre.
1.4 PREREQUISITO(S)	:	CS106. Estructuras Discretas II. (2 <sup>do</sup> Sem)
1.5 CARÁCTER	:	Obligatorio
1.6 HORAS	:	2 HT; 2 HP; 2 HL;
1.7 CRÉDITOS	:	4

**2. DOCENTE**

**3. FUNDAMENTACIÓN 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.

**4. SUMILLA**

1. AL/Computabilidad Básica.2. AL/Clases de Complejidad P y NP.3. AL/Teoría de Automatas.

**5. OBJETIVO GENERAL**

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

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

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

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. [Nivel Bloom: 4]
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. [Nivel Bloom: 4]
- 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. [Nivel Bloom: 4]

**7. CONTENIDOS**

<b>UNIDAD 1: AL/Computabilidad Básica.(20 horas)</b>	
<b>Nivel Bloom: 4</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
<ul style="list-style-type: none"> <li>▪ Discutir el concepto de máquinas de estado finito.</li> <li>▪ Explicar las gramáticas libres de contexto.</li> <li>▪ Diseñar una máquina de estados finitos determinística para aceptar un lenguaje específico.</li> <li>▪ Explicar cómo algunos problemas no tienen solución algorítmica.</li> <li>▪ Proveer ejemplos que ilustren el concepto de no-computabilidad.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Máquinas de estado finito.</li> <li>▪ Gramáticas libres del contexto.</li> <li>▪ Problemas tratables e intratables.</li> <li>▪ Funciones no computables.</li> <li>▪ El problema de la parada.</li> <li>▪ Implicaciones de la no-computabilidad.</li> </ul>
<b>Lecturas:</b> [Kolman, 1997], [Kelley, 1995]	

<b>UNIDAD 2: AL/Clases de Complejidad P y NP.(20 horas)</b>	
<b>Nivel Bloom: 5</b>	
<b>OBJETIVO GENERAL</b>	<b>CONTENIDO</b>
<ul style="list-style-type: none"> <li>▪ Definir las clases P y NP.</li> <li>▪ Explicar el significado de la NP-Complejidad.</li> <li>▪ Probar que un problema es NP-completo reduciendo un problema NP-Completo clásico conocido a éste.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Definición de las clases P y NP.</li> <li>▪ NP-complejidad (El teorema de Cook).</li> <li>▪ Problemas NP-completos estándares.</li> <li>▪ Técnicas de reducción.</li> </ul>
<b>Lecturas:</b> [Kelley, 1995], [Hopcroft and Ullman, 1993]	

UNIDAD 3: AL/Teoría de Autómatas.(20 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> <li>▪ Determinar la localización de un lenguaje en la jerarquía de Chomsky (conjuntos regulares, libres del contexto, sensibles al contexto y lenguajes enumerables recursivos).</li> <li>▪ Probar que un lenguaje se encuentra en una clase específica y que este no se encuentra en la siguiente clase inferior.</li> <li>▪ Conversiones entre notaciones potentes equivalentes para un lenguaje, incluyendo conversiones entre DFAs, NFAs y expresiones regulares así como entre PDAs y CFGs.</li> <li>▪ Explicar al menos un algoritmo de de análisis de arriba hacia abajo (<i>parsing top-down</i>) o de análisis de abajo hacia arriba (<i>bottom-up</i>).</li> <li>▪ Explicar la tesis de Church-Turing y su importancia.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Autómatas finitos determinísticos (DFAs).</li> <li>▪ Autómatas finitos no determinísticos (NFAs).</li> <li>▪ Equivalencias entre los DFAs y NFAs.</li> <li>▪ Expresiones regulares.</li> <li>▪ El teorema del bombeo (<i>pumping</i>) para expresiones regulares.</li> <li>▪ Autómatas de pila (PDAs).</li> <li>▪ Relación entre los PDAs y las gramáticas libres del contexto.</li> <li>▪ Propiedades de las gramáticas libres del contexto.</li> <li>▪ Máquinas de Turing.</li> <li>▪ Máquinas de Turing no determinísticas.</li> <li>▪ Conjuntos y lenguajes.</li> <li>▪ La jerarquía de Chomsky.</li> <li>▪ La tesis de Church-Turing.</li> </ul>
<b>Lecturas:</b> [Hopcroft and Ullman, 1993], [Brookshear, 1993]	

## 8. METODOLOGÍA

El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.

El profesor del curso presentará demostraciones para fundamentar clases teóricas.

El profesor y los alumnos realizarán prácticas

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.

## 9. EVALUACIONES

**Evaluación Permanente 1** : 20 %

**Examen Parcial** : 30 %

**Evaluación Permanente 2** : 20 %

**Examen Final** : 30 %

## Referencias

[Brookshear, 1993] Brookshear, J. G. (1993). *Teoría de la Computación*. Addison Wesley Iberoamericana.

- [Hopcroft and Ullman, 1993] Hopcroft, J. E. and Ullman, J. D. (1993). *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. CECSA.
- [Kelley, 1995] Kelley, D. (1995). *Teoría de Autómatas y Lenguajes Formales*. Prentice Hall.
- [Kolman, 1997] Kolman, Busby, R. (1997). *Estructuras de Matemáticas Discretas para la Computación*. Prentice Hall.