Universidad Católica San Pablo
Escuela Profesional de
Ciencia de la Computación
SILABO

CS211. Teoría de la Computación (Obligatorio)

2017-II

1. DATOS GENERALES

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS211. Teoría de la Computación
1.3 SEMESTRE ACADÉMICO	:	4^{to} Semestre.
1.4 PREREQUISITO(S)	:	CS1D2. Estructuras Discretas II. (2^{do} Sem)
1.5 CARÁCTER	:	Obligatorio
1.6 HORAS	:	2 HT; 2 HP; 2 HL;
1.7 CRÉDITOS	:	4

2. DOCENTE

Dra. Marcela Quispe Cruz

- Dr. Ciencia de la Computación, Pontificia Universidad Católica de Rio de Janeiro, Brasil, 2014.
- Mag. Ciencia de la Computación, Universidad de Pernambuco, Brasil, 2009.

3. FUNDAMENTACIÓN DEL CURSO

Este curso hace fasis 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. Computabilidad y complejidad básica de autómatas2. Complejidad Computacional Avanzada3. Teoría y Computabilidad Avanzada de Autómatas

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. (**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**)

7. COMPETENCIAS ESPECÍFICAS DE COMPUTACIÓN

Esta disciplina contribuye a la formación de las siguientes competencias del área de computación (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**

8. CONTENIDOS

UNIDAD 1: Computabilidad y complejidad básica de autómatas(20)

Competencias: C9

CONTENIDO	OBJETIVO GENERAL
▪ 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.	▪ 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 solucion algorítmica[Evaluar] ▪ Diseñe una gramática libre de contexto para representar un lenguaje especificado[Evaluar] ▪ Define las clases P y NP[Evaluar] ▪ Explique el significado de NP-Completitud[Evaluar] ▪ Explica la tesis de Church-Turing y su importancia[Familiarizarse] ▪ Explica el teorema de Rice y su importancia[Familiarizarse] ▪ Da ejemplos de funciones no computables[Familiarizarse] ▪ Demuestra que un problema es no computable al reducir un problema clásico no computable en base a él[Familiarizarse]

Lecturas: [Kolman, 1997], [Kelley, 1995]

UNIDAD 2: Complejidad Computacional Avanzada(20)	
Competencias: C8,C9	
CONTENIDO	**OBJETIVO GENERAL**
▪ Revisión de las clases P y NP; introducir spacio P y EXP. ▪ Jerarquía polimonial. ▪ NP completitud (Teorema de Cook). ▪ Problemas NP completos clásicos. ▪ Técnicas de reducción.	▪ Define las clases P y NP (También aparece en AL / Automata Básico, Computalidad y Complejidad)[Evaluar] ▪ Define la clase P-Space y su relación con la clase EXP[Evaluar] ▪ Explique el significado de NP-Completo (También aparece en AL / Automata Básico, Computalidad 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]
Lecturas: [Kelley, 1995], [Hopcroft and Ullman, 1993]	

UNIDAD 3: Teoría y Computabilidad Avanzada de Autómatas(20)	
Competencias: C8	
CONTENIDO	**OBJETIVO GENERAL**
▪ Conjuntos y Lenguajes: • 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: • Autómatas de pila (Push-down automata (PDAs) • Relación entre PDA y gramáticas libres de contexto. • Propiedades de los lenguajes libres de contexto.	▪ 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]
Lecturas: [Hopcroft and Ullman, 1993], [Brookshear, 1993]	

9. 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.

10. 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.