SOCIEDAD
PERUANA DE
COMPUTACIÓN

Sociedad Peruana de Computación (SPC)
Programa Profesional de
Ciencia de la Computación
Sílabo 2023-I

1. CURSO

CS211. Teoría de la Computación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Créditos	:	4
2.2 Horas de teoría	:	2 (Semanal)
2.3 Horas de práctica	:	2 (Semanal)
2.4 Duración del periodo	:	16 semanas
2.5 Condición	:	Obligatorio
2.6 Modalidad	:	■FaceToFace■
2.7 Prerrequisitos	:	CS1D2. Estructuras Discretas II. (2^{do} Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso hace enfasis en los lenguajes formales, modelos de computación y computabilidad, además de incluir fundamentos de la complejidad computacional y de los problemas NP completos.

5. OBJETIVOS

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

6. COMPETENCIAS

1) S.O. Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (**Evaluar**)

6) S.O. Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. . (**Evaluar**)

7. COMPETENCIAS ESPECÍFICAS

Nospecificoutcomes

8. TEMAS

Unidad 1: Autómatas y Lenguajes (20)	
Competencias esperadas: 1	
Temas	**Objetivos de Aprendizaje**
• 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. • Gramáticas libres de contexto. • 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] • 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] • Diseñe una gramática libre de contexto para representar un lenguaje especificado [Evaluar]
Lecturas : [Sip12], [HU08], [Bro93]	

Unidad 2: Teoría de la Computabilidad (20)	
Competencias esperadas: 1	
Temas	**Objetivos de Aprendizaje**
• Problema de la parada. • 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.	• Explique porque el problema de la parada no tiene solucion algorítmica [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
Lecturas : [Sip12], [Kel95]	

Unidad 3: Teoría de la Complejidad (20)	
Competencias esperadas: 6	
Temas	**Objetivos de Aprendizaje**
• 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 : [Sip12], [Kel95], [HU08]	

9. PLAN DE TRABAJO

9.1 Metodología

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.

9.2 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.

9.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

10. SISTEMA DE EVALUACIÓN

********* EVALUATION MISSING ********

11. BIBLIOGRAFÍA BÁSICA

[Bro93] J. Glenn Brookshear. *Teoría de la Computación*. Addison Wesley Iberoamericana, 1993.

[HU08] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. Pearson Educacion, 2008.

[Kel95] Dean Kelley. *Teoría de Autómatas y Lenguajes Formales*. Prentice Hall, 1995.

[Sip12] Michael Sipser. *Introduction to the Theory of Computation (third edition)*. Publisher: Cengage Learning, 2012.