



## Universidad Nacional de Ucayali (UNU)

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	:	Híbrido
2.7 Prerrequisitos	:	CS1D2. Estructuras Discretas II. (2 <sup>do</sup> Sem)

### 3. PROFESORES

Atención previa coordinación con el profesor

### 4. INTRODUCCIÓN AL 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.

### 5. OBJETIVOS

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

### 6. COMPETENCIAS

- 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 CS en el modelamiento y diseño de sistemas. (**Evaluar**)

### 7. TEMAS

<b>Unidad 1: Computabilidad y complejidad básica de autómatas (20)</b>	
<b>Competencias esperadas: a</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje</b>
<ul style="list-style-type: none"> <li>• Máquinas de estado finito.</li> <li>• Expresiones regulares.</li> <li>• Problema de la parada.</li> <li>• Gramáticas libres de contexto.</li> <li>• Introducción a las clases P y NP y al problema P vs. NP.</li> <li>• Introducción y ejemplos de problemas NP- Completos y a clases NP-Completos.</li> <li>• Máquinas de Turing, o un modelo formal equivalente de computación universal.</li> <li>• Máquinas de Turing no determinísticas.</li> <li>• Jerarquía de Chomsky.</li> <li>• La tesis de Church-Turing.</li> <li>• Computabilidad.</li> <li>• Teorema de Rice.</li> <li>• Ejemplos de funciones no computables.</li> <li>• Implicaciones de la no-computabilidad.</li> </ul>	<ul style="list-style-type: none"> <li>• Discute el concepto de máquina de estado finito [Evaluar]</li> <li>• Diseñe una máquina de estado finito determinista para aceptar un determinado lenguaje [Evaluar]</li> <li>• Genere una expresión regular para representar un lenguaje específico [Evaluar]</li> <li>• Explique porque el problema de la parada no tiene solución algorítmica [Evaluar]</li> <li>• Diseñe una gramática libre de contexto para representar un lenguaje especificado [Evaluar]</li> <li>• Defina las clases P y NP [Evaluar]</li> <li>• Explique el significado de NP-Complejidad [Evaluar]</li> <li>• Explique la tesis de Church-Turing y su importancia [Familiarizarse]</li> <li>• Explique el teorema de Rice y su importancia [Familiarizarse]</li> <li>• Da ejemplos de funciones no computables [Familiarizarse]</li> <li>• Demuestra que un problema es no computable al reducir un problema clásico no computable en base a él [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Mar10], [Lin11], [Sip12]	

<b>Unidad 2: Complejidad Computacional Avanzada (20)</b>	
<b>Competencias esperadas: a,b</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje</b>
<ul style="list-style-type: none"> <li>• Revisión de las clases P y NP; introducir espacio P y EXP.</li> <li>• Jerarquía polinomial.</li> <li>• NP completitud (Teorema de Cook).</li> <li>• Problemas NP completos clásicos.</li> <li>• Técnicas de reducción.</li> </ul>	<ul style="list-style-type: none"> <li>• Defina las clases P y NP (También aparece en AL / Automata Básico, Computabilidad y Complejidad) [Evaluar]</li> <li>• Defina la clase P-Space y su relación con la clase EXP [Evaluar]</li> <li>• Explique el significado de NP-Completo (También aparece en AL / Automata Básico, Computabilidad y Complejidad) [Evaluar]</li> <li>• Muestre ejemplos de problemas clásicos en NP - Completo [Evaluar]</li> <li>• Pruebe que un problema es NP- Completo reduciendo un problema conocido como NP-Completo [Evaluar]</li> </ul>
<b>Lecturas :</b> [Mar10], [Lin11], [Sip12], [HU13]	

Unidad 3: Teoría y Computabilidad Avanzada de Autómatas (20)	
Competencias esperadas: j	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Conjuntos y Lenguajes: <ul style="list-style-type: none"> <li>– Lenguajes Regulares.</li> <li>– Revisión de autómatas finitos determinísticos (Deterministic Finite Automata DFAs)</li> <li>– Autómata finito no determinístico (Nondeterministic Finite Automata NFAs)</li> <li>– Equivalencia de DFAs y NFAs.</li> <li>– Revisión de expresiones regulares; su equivalencia con autómatas finitos.</li> <li>– Propiedades de cierre.</li> <li>– Probando no-regularidad de lenguajes, a través del lema de bombeo (Pumping Lemma) o medios alternativos.</li> </ul> </li> <li>• Lenguajes libres de contexto: <ul style="list-style-type: none"> <li>– Autómatas de pila (Push-down automata PDAs)</li> <li>– Relación entre PDA y gramáticas libres de contexto.</li> <li>– Propiedades de los lenguajes libres de contexto.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Determina la ubicación de un lenguaje en la jerarquía de Chomsky (regular, libre de contexto, enumerable recursivamente) [Evaluar]</li> <li>• Convierte entre notaciones igualmente poderosas para un lenguaje, incluyendo entre estas AFDs, AFNDs, expresiones regulares, y entre AP y GLCs [Evaluar]</li> </ul>
Lecturas : [HU13], [Bro93]	

## 8. PLAN DE TRABAJO

### 8.1 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.

### 8.2 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.

### 8.3 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.

## 9. SISTEMA DE EVALUACIÓN

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Bro93] J. Glenn Brookshear. *Teoría de la Computación*. Addison Wesley Iberoamericana, 1993.
- [HU13] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. Pearson Education, 2013.
- [Lin11] Peter Linz. *An Introduction to Formal Languages and Automata*. 5th. Jones and Bartlett Learning, 2011.
- [Mar10] John Martin. *Introduction to Languages and the Theory of Computation*. 4th. McGraw-Hill, 2010.
- [Sip12] Michael Sipser. *Introduction to the Theory of Computation*. 3rd. Cengage Learning, 2012.