



## Universidad Nacional del Altiplano (UNA)

Escuela Profesional de  
Ciencia de la Computación  
Sílabo 2024-II

### 1. CURSO

CS211. Teoría de la Computación (Obligatorio)

### 2. INFORMACIÓN GENERAL

2.1 Curso	:	CS211. Teoría de la Computación
2.2 Semestre	:	4 <sup>to</sup> Semestre.
2.3 Créditos	:	4
2.4 horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS1D2. Estructuras Discretas II. (2 <sup>do</sup> Sem) 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. RESULTADOS DEL ESTUDIANTE

) ()

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

### 7. TEMAS

Unidad 1: Autómatas y Lenguajes (20)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<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>• Gramáticas libres de contexto.</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> <li>• Discute el concepto de máquina de estado finito [Evaluar]</li> <li>• Diseña 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>• Diseña una gramática libre de contexto para representar un lenguaje especificado [Evaluar]</li> </ul>
Lecturas : [Sipser12], [Hopcroft08], [Brookshear93]	

Unidad 2: Teoría de la Computabilidad (20)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Problema de la parada.</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>• Explique porque el problema de la parada no tiene solución algorítmica [Evaluar]</li> <li>• Defina las clases P y NP [Evaluar]</li> <li>• Explique el significado de NP-Complejidad [Evaluar]</li> <li>• Explica la tesis de Church-Turing y su importancia [Familiarizarse]</li> <li>• Explica 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</li> </ul>
Lecturas : [Sipser12], [Kelley95]	

Unidad 3: Teoría de la Complejidad (20)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<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 la clase P-Space y su relación con la clase EXP [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>
Lecturas : [Sipser12], [Kelley95], [Hopcroft08]	

## 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**