



# Universidad Nacional de Ingeniería (UNI)

Escuela Profesional de

Ciberseguridad

Sílabo 2024-II

## 1. CURSO

CS311. Programación Competitiva (Obligatorio)

## 2. INFORMACIÓN GENERAL

2.1 Curso	:	CS311. Programación Competitiva
2.2 Semestre	:	6 <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	:	CS212. Análisis y Diseño de Algoritmos. (5 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La Programación Competitiva combina retos de solucionar problemas con el añadido de poder competir con otras personas. Enseña a los participantes a pensar más rápido y desarrollar habilidades para resolver problemas, que son de gran demanda en la industria. Este curso enseñará la resolución de problemas algorítmicos de manera rápida combinando la teoría de algoritmos y estructuras de datos con la práctica la solución de los problemas.

## 5. OBJETIVOS

- Que el alumno utilice técnicas de estructuras de datos y algoritmos complejos.
- Que el alumno aplique los conceptos aprendidos para la aplicación sobre un problema real.
- Que el alumno investigue la posibilidad de crear un nuevo algoritmo y/o técnica nueva para resolver un problema real.

## 6. RESULTADOS DEL ESTUDIANTE

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)
- 6) Aplicar principios y prácticas de seguridad para mantener las operaciones en presencia de riesgos y amenazas. (Usage)

## 7. TEMAS

Unidad 1: Introducción (20 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Introducción a la Programación competitiva</li> <li>• Modelo computacional</li> <li>• Complejidad algorítmica</li> <li>• Problemas sobre búsqueda y ordenamiento</li> <li>• Recursión y recurrencia</li> <li>• Estrategia divide y conquista</li> </ul>	<ul style="list-style-type: none"> <li>• Reconocer y sabes como usar los recursos del modelo de computación RAM (Random Access Machine). [Usar]</li> <li>• Determinar el tiempo y espacio de complejidad de algoritmos. [Usar]</li> <li>• Determinar relaciones de recurrencia para algoritmos recursivos.[Usar]</li> <li>• Resolver problemas de búsqueda y ordenamiento.[Usar]</li> <li>• Aprender a seleccionar los algoritmos adecuados para problemas de tipo divide y conquista.[Usar]</li> <li>• Diseñar nuevos algoritmos para la resolución de problemas.[Usar]</li> </ul>
Lecturas : [Cormen2009], [Steven09], [Kulikov09], [SkienaRevilla:PC:2003], [Laaksonen17], [aziz2012elements]	

Unidad 2: Estructuras de datos (20 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Problemas sobre arrays y strings</li> <li>• Problemas sobre listas enlazadas</li> <li>• Problemas sobre pilas, colas</li> <li>• Problemas sobre arboles</li> <li>• Problemas sobre Hash tables</li> <li>• Problemas sobre Heaps</li> </ul>	<ul style="list-style-type: none"> <li>• Reconocer las distintas estructuras de datos sus complejidades usos y restricciones. [Usar]</li> <li>• Identificar el tipo de estructura de datos adecuado a la resolución del problema. [Usar]</li> <li>• Reconocer tipos de problemas asociado a operaciones sobre estructuras de datos como búsqueda, inserción, eliminación y actualización.[Usar]</li> </ul>
Lecturas : [Cormen2009], [Steven09], [Kulikov09], [SkienaRevilla:PC:2003], [Laaksonen17], [aziz2012elements]	

Unidad 3: Paradigmas de diseño (20 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Fuerza bruta</li> <li>• Divide y conquista</li> <li>• Backtracking</li> <li>• Greedy</li> <li>• Programación Dinamica</li> </ul>	<ul style="list-style-type: none"> <li>• Aprender los distintos paradigmas de resolución de problemas.[Usar]</li> <li>• Aprender a seleccionar los algoritmos adecuados para distintos problemas según el tipo de paradigma.[Usar]</li> </ul>
Lecturas : [Cormen2009], [Steven09], [Kulikov09], [SkienaRevilla:PC:2003], [Laaksonen17], [aziz2012elements]	

Unidad 4: Gráfos (20 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Recorrido de gráfos</li> <li>• Aplicaciones y problemas sobre gráfos</li> <li>• Camino mas corto</li> <li>• Redes y flujos</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar problemas clasificados como problemas de grafos. [Usar]</li> <li>• Aprender a seleccionar los algoritmos adecuados para problemas de grafos (recorrido, MST, camino mas costo, redes y flujos) y conocer sus soluciones eficientes. [Usar]</li> </ul>
Lecturas : [Cormen2009], [Steven09], [Kulikov09], [SkienaRevilla:PC:2003], [Laaksonen17], [aziz2012elements]	

Unidad 5: Tópicos avanzados (20 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Teoria de números</li> <li>• Probabilidad y combinaciones</li> <li>• Algoritmos para manejos de strings (tries, string hashing, z-algorithm)</li> <li>• Geometria y sweep line algorithms, segment trees</li> </ul>	<ul style="list-style-type: none"> <li>• Aprender a elegir los algoritmos adecuados para problemas sobre teoria de números y matemáticas ya que son importantes en programación competitiva. [Usar]</li> <li>• Aprender a seleccionar los algoritmos adecuados para problemas sobre probabilidades y combinaciones, manejos de strings y geometría computacional. [Usar]</li> </ul>
Lecturas : [Cormen2009], [Steven09], [Kulikov09], [SkienaRevilla:PC:2003], [Laaksonen17], [aziz2012elements]	

Unidad 6: Problemas de dominio específico (20 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Latencia y rendimiento</li> <li>• Paralelismo</li> <li>• Redes</li> <li>• Almacenamiento</li> <li>• Alta disponibilidad</li> <li>• Caching</li> <li>• Proxies</li> <li>• Equilibradores de carga</li> <li>• Almacenamiento clave-valor</li> <li>• Replicar y compartir</li> <li>• Elección del líder</li> <li>• Limitación de la tasa</li> <li>• Registro y monitoreo</li> </ul>	<ul style="list-style-type: none"> <li>• Aprender a diseñar sistemas para diferentes problemas de dominio específico aplicando conocimiento sobre redes, computación distribuida, alta disponibilidad, almacenamiento y arquitectura de sistemas. [Usar]</li> </ul>
Lecturas : [Cormen2009], [Steven09], [Kulikov09], [SkienaRevilla:PC:2003], [Laaksonen17], [aziz2012elements]	

## **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**