



Universidad Nacional del Altiplano (UNA)

Escuela Profesional de
Ciencia de la Computación
Sílabo 2024-II

1. CURSO

CS311. Programación Competitiva (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS311. Programación Competitiva
2.2 Semestre	:	6 ^{to} 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 ^{to} Sem) CS212. Análisis y Diseño de Algoritmos. (5 ^{to} 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

) ()

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

7. TEMAS

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

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

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

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

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

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