SOCIEDAD
PERUANA DE
COMPUTACIÓN

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

1. CURSO

CS311. Programación Competitiva (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	:	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. COMPETENCIAS

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

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

7. COMPETENCIAS ESPECÍFICAS

Nospecificoutcomes

8. TEMAS

Unidad 1: Introducción (20)	
Competencias esperadas: 1	
Temas	**Objetivos de Aprendizaje**
• Introducción a la Programación competitiva • Modelo computacional • Complejidad algoritmica • Problémas sobre búsqueda y ordenamiento • Recursión y recurrencia • Estrategia divide y conquista	• 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 : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 2: Estructuras de datos (20)	
Competencias esperadas: 1	
Temas	**Objetivos de Aprendizaje**
• Problemas sobre arrays y strings • Problemas sobre listas enlazadas • Problemas sobre pilas, colas • Problemas sobre arboles • Problemas sobre Hash tables • Problemas sobre Heaps	• 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 : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 3: Paradigmas de diseño (20)	
Competencias esperadas: 1	
Temas	**Objetivos de Aprendizaje**
• Fuerza bruta • Divide y conquista • Backtracking • Greedy • Programación Dinamica	• 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 : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 4: Gráfos (20)	
Competencias esperadas: 1	
Temas	**Objetivos de Aprendizaje**
• Recorrido de gráfos • Aplicaciones y problemas sobre gráfos • Camino mas corto • Redes y flujos	• 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 : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 5: Tópicos avanzados (20)	
Competencias esperadas: 1	
Temas	**Objetivos de Aprendizaje**
• Teoria de números • Probabilidad y combinaciones • Algoritmos para manejos de strings (tries, string hashing, z-algorithm) • Geometria y sweep line algorithms, segment trees	• 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 : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 6: Problemas de dominio especifico (20)	
Competencias esperadas: 1	
Temas	**Objetivos de Aprendizaje**
• Latencia y rendimiento • Paralelismo • Redes • Almacenamiento • Alta disponibilidad • Caching • Proxies • Equilibradores de carga • Almacenamiento clave-valo • Replicar y compartir • Elección del líder • Limitación de la tasa • Registro y monitoreo	• Aprender a diseñar sistemas para diferentes problemas de dominio especifico aplicando conocimiento sobre redes, computación distribuida, alta disponibilidad, almacenamiento y arquitectura de sistemas. [Usar]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

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

[ALP12] A. Aziz, T.H. Lee, and A. Prakash. *Elements of Programming Interviews: The Insiders' Guide*. ElementsOfProgrammingInterviews.com, 2012. ISBN: 9781479274833. URL: `https://books.google.com.pe/books?id=y6FLBQAAQBAJ`.

[Cor+09] T. H. Cormen et al. *Introduction to Algorithms*. MIT Press, 2009.

[Hal13] Steven Halim. *Competitive Programming*. 3 rd. Lulu, 2013.

[Kul19] Alexander S. Kulikov. *Learning Algorithms Through Programming and Puzzle Solving*. Active Learning Technologies, 2019.

[Laa17] Antti Laaksonen. *Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*. Stringer, 2017.

[Mig03] Steve Skiena Miguel A. Revilla. *Programming Challenges: The Programming Contest Training Manual*. Springer, May 2003. ISBN: 978-0387001630.