

SILABO DEL CURSO

ANÁLISIS DE ALGORITMOS Y ESTRATEGIAS DE PROGRAMACIÓN

I. INFORMACIÓN GENERAL

Facultad	Ingeniería	Carrera profesional	Ingeniería de Sistemas Computacionales	Ciclo	5°	Créditos		3
Código de curso	COMP1303A	Requisitos	Estructuras de Datos	Horas	HT	HP	HL	PC
					2	0	2	0
Tipo de curso	Obligatorio	Modalidad del curso	A distancia sincrónico	Periodo lectivo	2025-1			
El curso aporta a la(s) competencia(s) general(es):		<ul style="list-style-type: none">Resolución de ProblemasPensamiento Creativo y Crítico						
El curso aporta a la(s) competencia(s) específica(s):		<ul style="list-style-type: none">Conocimientos de Ingeniería						
El curso desarrolla el componente:		<ul style="list-style-type: none">Experiencia Preprofesional						
ODS (número y nombre):		<ul style="list-style-type: none">No aplica						

II. SUMILLA

El curso es de naturaleza teórico - práctico y tiene como propósito brindar un conjunto de técnicas independientes del lenguaje, paradigma de programación y hardware que permitan afrontar con mayores posibilidades de éxito la solución de un problema computacional, comprendiendo el estudio de su complejidad y los recursos necesarios para un adecuado funcionamiento.

Los temas principales son: la complejidad algorítmica y principales estrategias de programación, Estrategias de programación y algoritmos en paralelo.

III. LOGRO DEL CURSO

Al finalizar el curso, el estudiante soluciona problemas computacionales, aplicando técnicas como recursión y backtracking, algoritmos voraces, divide y vencerás, fuerza bruta, programación dinámica, entre otras, utilizando lenguajes de programación como Python y C#, demostrando lógica, habilidad y legibilidad del código, que le permita implementar algoritmos óptimos en un proyecto de aplicación.

IV. METODOLOGÍA DE ENSEÑANZA APRENDIZAJE

Para alcanzar el logro de aprendizaje del curso y de las unidades, el docente integra métodos activos, estrategias y técnicas de manera reflexiva y crítica, buscando motivar, estimular y guiar el aprendizaje del estudiante.

Las estrategias y técnicas didácticas que se utilizan son: aprendizaje basado en casos en forma de ejercicios propuestos de manera individual y en equipo, de modo que analiza, prueba, valida y justifica las mejores técnicas computacionales y modelos a utilizar. Además, se emplean diferentes herramientas y software para el desarrollo de las propuestas.

El docente soporta su práctica pedagógica en un sistema multiplataforma y recursos multimedia que le permite el desarrollo de actividades sincrónicas y asincrónicas, así como la gestión de contenidos, videoconferencias y el uso de herramientas que permitan mejorar la experiencia formativa que promuevan el aprendizaje y el desarrollo de competencias generales y específicas en los estudiantes.

V. ORGANIZACIÓN DE UNIDADES DE APRENDIZAJE

HT=Horas de desarrollo teórico
HP= Horas de desarrollo práctico
HL= Horas de desarrollo práctico en laboratorio
PC= Horas de práctica de campo

UN	NOMBRE / LOGRO DE UNIDAD	SEM	SABERES ESENCIALES
I	La complejidad algorítmica y principales estrategias de programación. Al finalizar la unidad, el estudiante implementa programas utilizando las estrategias de programación fuerza bruta, algoritmos voraces, divide y vencerás, back tracking, utilizando el lenguaje Python, analizando su complejidad algorítmica.	1	Presentación del sílabo del curso y protocolo de seguridad de laboratorio. Introducción al lenguaje Python.
		2	Tipos estructurados, registros, cadena de caracteres en Python.
		3	Algoritmos de fuerza bruta. Algoritmos de Fuerza bruta. Definición y aplicaciones. Complejidad, funciones básicas, reglas de simplificación, reglas para el cálculo de complejidad, análisis asintótico, Notación O grande, omega y theta.
		4	Análisis de Algoritmos: Búsqueda, lineal, lineal acotada, búsqueda binaria iterativa, búsqueda binaria recursiva. Análisis algoritmos de Ordenamiento por intercambio, burbuja, inserción, selección, rápida, montículo, mezcla.
		5	Algoritmos Voraces. Cambio de moneda, factores primos, ruta entre dos nodos, árbol de recubrimiento mínimo. Problema de la mochila
		6	Evaluación T1
		7	Algoritmos recursivos
		8	Divide y conquista. Definición, forma general. Ejemplos: Ordenamientos, Búsqueda en conjuntos ordenados, multiplicación de enteros, mediana y moda de un vector, permutaciones, potencia, MCD.
		9	Backtracking. Problemas que pueden ser resueltos con backtracking, esquema general, ventajas, desventajas. Ejemplos diversos: Permutaciones, variaciones, Laberintos, Sudoku, Torres de Hanoi, 8 reinas, el salto del caballo, ruta crítica, el problema de la mochila.
		10	Evaluación T2
II	Estrategias de programación y algoritmos en paralelo Al finalizar la unidad, el estudiante implementa programas, utilizando las estrategias de programación dinámica, algoritmos probabilísticos y algoritmos en paralelo, analizando la complejidad algorítmica y presentando su proyecto final con aplicación de las estrategias adecuadas.	11	Programación dinámica Definición, forma general, condiciones, la programación dinámica como problema de optimización, el principio de optimalidad de Bellman. Problemas resueltos
		12	Algoritmos probabilistas o aleatorios. Tiempo esperado frente a tiempo promedio, algoritmos probabilistas numéricos. Algoritmos de Monte Carlo, algoritmo las vegas. Ejemplos: Integración, comprobación de primalidad, factorización de enteros grandes. Algoritmos paralelos. Desarrollo de casos.
		13	Evaluación T3
		14	Técnicas básicas: árbol binario completo, duplicación de punteros. Teoría de grafos. Evaluación de expresiones en paralelo.
		15	Redes de ordenación y fusión en paralelo. Algoritmos distribuidos
		16	Evaluación Final
		(-)	No aplica evaluación sustitutoria.

VI. SISTEMA DE EVALUACIÓN

EVALUACIÓN	PESOS	SEM	DESCRIPCIÓN DE LA EVALUACIÓN (Acción + Producto de la evidencia que debe presentar el estudiante)
T1 (a)	10%	6	Desarrollo de casos
T2 (a)	20%	10	Desarrollo de casos
T3 (a)	30%	13	Desarrollo de casos y componente Academia Cisco
Evaluación Final (a)	40%	16	Presentación y exposición de proyecto final (medición de competencias)

(a) Los calificativos deben ser publicados en el sistema de acuerdo con el Calendario Académico establecido para el presente Semestre.

VII. BIBLIOGRAFÍA BÁSICA

Nº	AUTOR	TÍTULO	AÑO	ENLACE URL
1	Rodriguez, Diego Fernando	Guía del programador competitivo	2021	https://digitalia.bibliotecaupn.elogim.com/a/130016

a) BIBLIOGRAFÍA COMPLEMENTARIA

Nº	AUTOR	TÍTULO	AÑO	ENLACE URL
1	Algar Díaz, M. J.; Fernández de Sevilla Vellón, M.	Introducción práctica a la programación con Python.	2019	https://elibro.bibliotecaupn.elogim.com/es/lc/upnorte/titulos/124259
2	Vegas Gertrudix, José M.	Java 17: programación avanzada	2021	https://elibro.bibliotecaupn.elogim.com/es/lc/upnorte/titulos/222668

VIII. INFORMACIÓN COMPLEMENTARIA

REFERENCIA	ENLACE URL
Biblioteca Virtual UPN	https://biblioteca.upn.edu.pe/
Una introducción a las matemáticas para el análisis y diseño de algoritmos	https://elibro.bibliotecaupn.elogim.com/es/lc/upnorte/titulos/35059
Criptografía sin secretos con Python	https://elibro.bibliotecaupn.elogim.com/es/lc/upnorte/titulos/106497
ACM UVA Online	http://uva.onlinejudge.org/
Cisco Networking Academy – Python Essentials 1	https://www.netacad.com/es