

ASIGNATURA	:	COMPUTACIÓN CIENTÍFICA I
DEPARTAMENTO	:	MATEMÁTICAS
SEMESTRE	:	III SEMESTRE
PRERREQUISITO	:	NINGUNO
No. CRÉDITOS	:	4 CREDITOS
INTENSIDAD HORARIA	:	4 SEMANALES
CÓDIGO	:	MATE5010

PERFIL DOCENTE:

Catedrático

Profesional en matemáticas, física, estadística o ciencias exactas; con estudios de Especialización y/o Maestría en el campo matemático. Con mínimo 2 años de experiencia docente en cursos de pregrado de las carreras de matemáticas, estadística y/o física.

Medio Tiempo

Profesional en matemáticas, física, estadística o ciencias exactas; con estudios de Especialización y/o Maestría en el campo matemático. Con mínimo 2 años de experiencia docente en cursos de pregrado de matemáticas, estadística y/o física.

Tiempo Completo

Profesional en matemáticas, física, estadística o ciencias exactas; con estudios de Especialización y/o Maestría en el campo matemático. Con mínimo 5 años de experiencia docente en cursos de pregrado de matemáticas, estadística y/o física.

JUSTIFICACIÓN DE LA UNIDAD DE ESTUDIO

Los algoritmos y la programación aportan a los estudiantes de Matemáticas los conocimientos, así como las competencias y habilidades de análisis y síntesis, para plantear la solución de problemas, utilizando algoritmos y las estructuras de un lenguaje de programación.

1. COMPETENCIAS DE LA UNIDAD DE ESTUDIO

El estudiante es capaz de plantear, programar y explicar soluciones a diferentes problemas que tengan soluciones algorítmicas relacionado con problemas en las matemáticas como, teoría de conjuntos, teoría de números y matemáticas discretas.

El estudiante analiza problemas de complejidad computacional de los algoritmos estableciendo la eficiencia o ineficiencia de estos. Maneja criterios formales para entender como funcionan los algoritmos.

2. CONTENIDO PROGRAMÁTICO

Corte 1:

1. Introducción a los algoritmos
 - a. ¿Qué es un algoritmo?
 - b. Ejemplos de algoritmos en la vida cotidiana.
 - c. Ejemplos de algoritmos en matemáticas.
2. Introducción a la programación
 - a. Introducción a Python.
 - b. Estructura de Datos.
 - c. Tipos de datos.
 - d. Operadores aritméticos y lógicos.
 - e. Funciones.
 - f. Control de flujo (condicionales y bucles).
3. Programación orientada a objetos.
 - a. Conceptos de programación orientada a objetos.
 - b. Clases y objetos en Python.
 - c. Herencia y polimorfismo.
4. Complejidad de los Algoritmos.
 - a. Notación O-grande.
 - b. Análisis de la complejidad de los algoritmos.
 - c. Algoritmos eficientes y algoritmos ineficientes.
5. Algoritmos recursivos en Python
 - a. Definición de algoritmos recursivos
 - b. Implementación de algoritmos recursivos en Python
 - c. Ejemplos de algoritmos recursivos (factorial, números de Fibonacci, torres de Hanoi)
6. Algoritmos de ordenamiento.
 - a. Algoritmo de burbuja
 - b. Algoritmo de selección
 - c. Algoritmo de inserción
 - d. Algoritmos de ordenamiento más avanzados (quicksort, mergesort)
7. Algoritmos de búsquedas
8. Simulaciones
9. Proyecto final
 - a. Aplicación práctica de los conceptos aprendidos en el curso en un proyecto final

3. SOPORTE BIBLIOGRAFICO

- *Python Fácil*, Arnaldo Pérez Castaño, Editorial Alfaomega.
 - *Learning Python*, Mark Lutz. Editorial O'Reilly. Quinta edición.
 - *Fundamentals of Python: First Programs*, Kenneth Lambert. Editorial Cengage Learning.
-

4. ESTRATEGIA METODOLÓGICA

CÁTEDRA
EJERCICIOS PRÁCTICOS

5. SISTEMA DE EVALUACIÓN

PRUEBA ACADÉMICA	PORCENTAJE DE LA CALIFICACIÓN DEFINITIVA
Primer examen parcial	30
Segundo examen parcial	30
Proyecto final	40