

PROGRAMA DE ASIGNATURA

I. IDENTIFICACIÓN DE LA ASIGNATURA

Sigla	ICI 2245
Nombre Asignatura	Análisis y diseño de algoritmos
Créditos	4
Duración	108 horas pedagógicas
Semestre	4
Requisitos	-
Horas Teóricas	4
Horas Ayudantía	2
Horas Laboratorio	-
Horas Taller	-
Horas de Estudio Personal	6
Área curricular a la que pertenece la asignatura	Ingeniería Aplicada: Desarrollo de Aplicaciones Informáticas
N° y año Decreto Programa de Estudio	-
Carácter de la asignatura	Obligatorio
N.º máximo de estudiantes	50

II. DESCRIPCIÓN Y CONTEXTUALIZACIÓN DE LA ASIGNATURA EN EL CURRÍCULO

Asignatura de carácter obligatorio, correspondiente al área de la Computación. Esta asignatura presenta aspectos teóricos y prácticos para el análisis de algoritmos, así como las técnicas utilizadas para diseñar algoritmos eficientes. El propósito general de esta asignatura es preparar al estudiante para analizar, diseñar e implementar algoritmos que permitan solucionar de manera eficiente problemas de la disciplina y comprender las diferentes clases de complejidad computacional.

Progreso de la(s) las competencias a las cuales aporta la asignatura:

CD1 Integra conocimientos de ciencias básicas y ciencias de la ingeniería para identificar, analizar y resolver problemas de la especialidad.

CP2 Lidera unidades del área informática en la organización aplicando conocimientos administrativos y de gestión para mejorar la eficiencia de los procesos organizacionales.

CF4 Usa las tecnologías de la información y comunicación como herramienta del desarrollo académico y profesional.

CF5 Demuestra capacidad científica; de análisis, abstracción, síntesis y reflexión crítica con el objetivo de resolver problemas, construir conocimiento y desarrollar autoaprendizaje, tanto a nivel individual como en el trabajo en equipos interdisciplinarios.

III. RESULTADOS DE APRENDIZAJE

Al final de la Asignatura, el estudiante será capaz de:

- Evaluar algoritmos considerando criterios de eficiencia
- Utilizar técnicas para el diseño e implementación de algoritmos eficientes
- Comprender las diferentes clases de complejidad computacional y sus problemas asociados

IV. CONTENIDOS o UNIDADES DE APRENDIZAJE

1. Introducción

- 1.1. Algoritmia elemental
- 1.2. Por qué buscar eficiencia
- 1.3. Notación asintótica

2. Análisis de algoritmos

- 2.1. Algoritmos de búsqueda, ordenamiento, sobre árboles y grafos
- 2.2. Análisis caso medio, análisis amortizado
- 2.3. Resolución de recurrencias
- 2.4. Estructuras de datos eficientes

3. Técnicas para el diseño de algoritmos

- 3.1. Fuerza bruta
- 3.2. Dividir y conquistar
- 3.3. Algoritmos voraces
- 3.4. Programación dinámica
- 3.5. Backtracking
- 3.6. Branch and Bound

4. Complejidad computacional

- 4.1. Algoritmos y complejidad
- 4.2. P/NP
- 4.3. NP completitud

V. ACTIVIDADES DE APRENDIZAJE

- Trabajo individual y en equipo
- Aprendizaje basado en problemas
- Clases expositivas

VI. EVALUACIÓN DE LOS RESULTADOS DE APRENDIZAJE

✓ **Instancias de Evaluación:**

- 6 evaluaciones formativas (tareas) del trabajo desarrollado clase a clase
- 3 evaluaciones sumativas intermedias (pruebas de cátedra)

✓ **Condiciones de Aprobación:**

- Nota Final: pruebas (60%) + tareas (40%)

Las condiciones de aprobación son las que están establecidas en el reglamento de la Escuela

VII. BIBLIOGRAFÍA Y OTROS RECURSOS PARA EL APRENDIZAJE

✓ **Bibliografía Obligatoria:**

- G. Brassard, P. Bratley. **Fundamentos de Algoritmia**, Prentice Hall, 1997
- T. Cormen et al. **Introduction to algorithms**, MIT, 3ra edición.

✓ **Otros materiales de apoyo:**

- Proyector digital (data show)
- Computador

Académico responsable de la elaboración del programa: Wenceslao Palma

Fecha de elaboración del programa: 22/02/2020

Académico responsable de la modificación del programa: Wenceslao Palma

Fecha de modificación del programa: 22/05/2020