

PLANTILLA BÁSICA PARA EL DISEÑO CURRICULAR DE PROGRAMA

1. IDENTIFICACIÓN:

NOMBRE ESCUELA:	Escuela de Ingeniería
NOMBRE DEPARTAMENTO:	Departamento de Informática y Sistemas
NOMBRE DEL PROGRAMA:	Ingeniería de sistemas
NOMBRE ASIGNATURA EN ESPAÑOL:	Estructuras de Datos y Algoritmos 2
NOMBRE ASIGNATURA EN INGLÉS:	Design of Algorithms
MATERIA PREREQUISITO:	Estructuras de Datos y Algoritmos 1
CÓDIGO:	ST0247
SEMESTRE DE UBICACIÓN:	III
INTENSIDAD HORARIA SEMANAL:	3 horas
INTENSIDAD HORARIA SEMESTRAL:	48 horas
CRÉDITOS:	Generado por el sistema
CARACTERÍSTICAS:	Generados por el sistema
	Generada por el sistema

2. JUSTIFICACIÓN DEL CURSO:

El Ingeniero de Sistemas egresado de Eafit debe estar en capacidad de hacer el análisis, diseño e implementación de sistemas intensivos en Software, y, por esa razón, es imperativo que sea capaz de diseñar un algoritmo para resolver eficientemente un problema y hacer su implementación.

3. PROPÓSITO U OBJETIVO GENERAL DEL PROGRAMA ACADÉMICO:

Resolver un problema donde se requiere diseñar estructuras de datos y algoritmos, y argumentar los criterios de diseño.

3.1. COMPETENCIAS GENÉRICAS:

1. Realizar búsqueda de literatura impresa y electrónica (CDIO 2.2.2)
2. Capacidad de trabajar en distintos tipos de equipos (CDIO 3.1.5)
3. Capacidad de comunicación escrita efectiva (CDIO 3.2.3)
4. Capacidad de comunicación por medios Multimedia (CDIO 3.2.4)
5. Capacidad de comunicación por presentaciones orales (CDIO 3.2.6)

3.2 COMPETENCIAS ESPECÍFICAS:

1. Diseña algoritmos de fuerza bruta, backtracking, voraces, dividir para conquistar y programación dinámica para solucionar problemas

3.3 RESULTADOS DE APRENDIZAJE

Competencia específica:

Diseña algoritmos de fuerza bruta, backtracking, voraces, dividir para conquistar y programación dinámica para solucionar problemas

Resultados de aprendizaje de la competencia:

1. Explica el problema que va a resolver
2. Plantea alternativas de solución al problema
3. Explica los principios que se están aplicando en el algoritmo de solución del problema
4. Explica el algoritmo que soluciona el problema conceptualmente
5. Implementa correctamente el algoritmo que soluciona el problema

4. DESCRIPCION DE LOS CONTENIDOS: TEMAS Y SUBTEMAS:

1. Algoritmos de fuerza bruta
2. Algoritmos de backtracking
3. Algoritmos codiciosos
4. Algoritmos de dividir para conquistar
5. Algoritmos de programación dinámica

5. ESTRATEGIAS METODOLÓGICAS Y DIDÁCTICAS:

Se sugiere al maestro orientar clases magistrales donde se exponen los temas del curso, tareas donde se trabaja en el diseño de algoritmos para resolver eficientemente problemas y se implementan en un lenguaje de programación. Adicionalmente, hacer énfasis en la práctica final de semestre donde se resuelve un problema, que se implementa en un lenguaje de programación, y orientarlos en la escritura de un reporte técnico sobre la práctica final. No se

recomienda dedicar mucho tiempo a la fundamentación matemática necesaria para calcular la complejidad computacional (por ejemplo, sumatorias, teorema maestro y resolución de ecuaciones de recurrencia); en su lugar, dedicar ese tiempo a la interpretación de los cálculos de complejidad y hacer uso de herramientas computacionales para resolver las ecuaciones y sumatorias como Wolfram Alpha.

5.1. Metodología docente y estimación de volumen de trabajo del estudiante

5.2. Temporalización o cronograma

1. Algoritmos de fuerza bruta (6h)
2. Algoritmos de backtracking (9h)
3. Algoritmos codiciosos (9h)
4. Algoritmos de dividir para conquistar (6h)
5. Algoritmos de programación dinámica (9h)
6. Evaluaciones parciales y del proyecto (9h)
 - 6.1 Parcial 1 (3h)
 - 6.2 Parcial 2 (3h)
 - 6.3 Sustentación del proyecto (3h)

6. RECURSOS:

Se sugiere al maestro utilizar las salas de cómputo disponibles en la universidad para realizar ejercicios de programación que ayuden a afianzar los conceptos teóricos y a desarrollar habilidades en el diseño e implementación de algoritmos.

6.1. Locativos:

6.2. Tecnológicos:

6.3. Didácticos:

6.4. Bibliográficos

7. CRITERIOS Y POLÍTICAS DE SEGUIMIENTO Y EVALUACIÓN ACADÉMICA:

Criterios de Evaluación

Se recomienda al maestro realizar un seguimiento periódico y sumativo a través de tareas que evalúen la capacidad de diseñar algoritmos para solucionar problemas. Adicionalmente, se propone realizar un proyecto donde deba solucionarse un problema del mundo real, que no tenga

una única respuesta, y que implique diseñar la mejor solución para el problema teniendo en cuenta varios criterios. Se sugiere realizar 2 evaluaciones parciales donde se evalúe, principalmente, el diseño de algoritmos. Se proponen los siguientes porcentajes para las evaluaciones: Parcial 1 20%, Parcial 2 25%, Proyecto 30%, Tareas 25%.

8. BIBLIOGRAFIA GENERAL:

R.C.T Lee et al., Introducción al análisis y diseño de algoritmos, 2005.

Robert Sedgewick, Algorithms (Deluxe edition), 2011.

Anany Levitin, Introduction to the design and analysis of algorithms, 2011

Skiena, The algorithm design manual (2nd edition), 2010

Thomas Cormen, Introduction to Algorithms (3rd edition), 2009

9. NOMBRE DEL PROFESOR COORDINADOR DE MATERIA Y NOMBRE DE PROFESORES DE LA MATERIA QUE PARTICIPARON EN LA ELABORACIÓN.

Coordinador: Mauricio Toro Bermúdez

Participantes: Juan G. Lalinde, Juan F. Cardona, Paola Vallejo, Edwin Montoya, Juan C. Montoya

10. REQUISITOS DEL PROCESOS DE ASEGURAMIENTO DE LA CALIDAD

- 10.1. Versión número: _____
- 10.2. Fecha elaboración: _____
- 10.3. Fecha actualización: _____
- 10.4. Responsable: _____
- 10.5. Aprobación: _____

Jefe del Programa

Jefe del Departamento