

**Análisis y diseño de algoritmos
Programa del curso
2026-1**

Profesores: Camilo Rocha y Carlos Ramírez

Correos: camilo.rocha@javerianacali.edu.co y carlosalbertoramirez@javerianacali.edu.co

Página web: <http://www.camilorocha.info/teaching/ada/2026-1>

Texto guía: *Diseño y análisis de Algoritmos* por C. Rocha (v0.0)

Textos adicionales:

- *Introduction to Algorithms (Fourth Edition)* por T. Cormen, C. Leiserson, R. Rivest, C. Stein.
- *Algorithm Design* por J. Kleinberg, É. Tardos.
- *Algorithms, Etc.* por J. Erickson (primera edición, junio 2019)

Lenguaje de programación: Python 3.7

Descripción

Esta asignatura aborda técnicas para diseñar y analizar algoritmos. Las técnicas incluyen dividir y conquistar, programación dinámica, programación voraz/avara y reintento. A los algoritmos diseñados se les aplican los dos tipos de análisis usuales en algoritmia: corrección (i.e., el algoritmo hace lo que debe hacer) y eficiencia (i.e., el algoritmo hace buen uso de los recursos disponibles). Se estudia la existencia de problemas decidibles e indecidibles, y también problemas intratables en una introducción a la teoría de la NP-completitud.

En esta asignatura se: (i) apropián y aplican conocimientos de ciencias de la computación para resolver algorítmicamente problemas computacionales; (ii) diseñan, implementan y optimizan algoritmos con base en técnicas de diseño algorítmico; (iii) comunican efectivamente decisiones de diseño e implementación de algoritmos; y (iv) usan técnicas formales de demostración y la notación asintótica para razonar acerca de algoritmos, al igual que herramientas para desarrollar experimentos computacionales.

Código de honor

Como miembro de la comunidad académica de la Pontificia Universidad Javeriana Cali, los valores éticos y la integridad son tan importantes como la excelencia académica. En este curso se espera que los estudiantes se comporten ética y honestamente, con los más altos niveles de integridad escolar. En particular, se asume que cada estudiante adopta el siguiente *código de honor*:

Como miembro de la comunidad académica de la Pontificia Universidad Javeriana Cali me comprometo a seguir los más altos estándares de integridad académica.

Integridad académica se refiere a ser honesto, dar crédito a quien lo merece y respetar el trabajo de los demás. Por eso es importante evitar plagiar, engañar, ‘hacer trampa’, etc. En particular, el acto de entregar un programa de computador ajeno como propio constituye un acto de plagio; cambiar el nombre de las variables, agregar o eliminar comentarios y reorganizar comandos no cambia el hecho de que se está copiando el programa de alguien más. Para más detalles consultar el *Reglamento de Estudiantes*, Sección VI.

Objetivos de aprendizaje

Al finalizar satisfactoriamente el curso, el estudiante estará en capacidad de:

OA1 Apropiar técnicas algorítmicas para solucionar problemas computacionales a través de la participación activa en el diseño y análisis de algoritmos para identificar soluciones, correctas y eficientes, a problemas de decisión, búsqueda, ordenamiento y optimización.

- 1.1** Apropiar conceptos y conocimientos de matemáticas y estadística para resolver problemas.
- 1.3** Apropiar conceptos y conocimientos de computación para resolver problemas.
- 1.4** Seleccionar y aplicar conceptos de ingeniería y ciencias naturales para resolver problemas propios de la disciplina.

OA2 Diseñar, implementar y optimizar algoritmos a partir de técnicas de diseño como dividir y conquistar, programación dinámica, algoritmos voraces reintento, para resolver problemas computacionales en contextos específicos y diversos.

- 2.3** Optimizar procesos y artefactos de acuerdo con métricas y criterios de la ingeniería y las ciencias naturales.
- 2.4** Evaluar algoritmos considerando requerimientos.

OA3 Comunicar efectivamente decisiones de diseño e implementación, demostraciones de correctitud y análisis de eficiencia de algoritmos, a través de la elaboración de documentos técnicos y sustentaciones orales ante compañeros y profesores.

- 3.2** Comunicar ideas efectivamente a diferentes audiencias, usando lenguajes, estilos y estrategias adecuadas.
- 3.4** Elaborar e interpretar material técnico en inglés.

OA6 Usar técnicas formales de demostración y la notación asintótica para asegurar la correctitud y eficiencia de los algoritmos diseñados e implementados, al igual que experimentos computacionales apoyados en herramientas de simulación modernas y apropiadas para asegurar la confiabilidad de los algoritmos implementados.

- 6.2** Usar herramientas avanzadas de especificación, análisis, modelado, diseño, optimización, síntesis y simulación computacional.
- 6.4** Verificar rigurosamente el desempeño y/o correctitud de sistemas computacionales.

Estrategias didácticas

Esta es una asignatura orientada por problemas algorítmicos que se encuentran en la literatura y en arenas de programación públicas. Consta de clases semanales, con los profesores y los estudiantes participando activamente, acompañadas de trabajo individual.

- Durante las sesiones de clase, laboratorio y trabajo individual se implementarán y analizarán algoritmos implementados en un lenguaje de programación.
- Para el trabajo individual se usará software de simulación que permita retroalimentar automáticamente al estudiante en relación con los algoritmos que implementa para las tareas y exámenes parciales.
- Los problemas, en su gran mayoría, se formularán en inglés.
- Eventualmente, se realizarán comprobaciones de lectura.

En particular, se usarán las siguientes estrategias didácticas con los respectivos recursos:

Clase magistral: aula de clases, proyector, pizarra, bibliografía, problemas algorítmicos, internet.

Aprendizaje basado en problemas: computador, problemas algorítmicos, casos de prueba, herramientas de comparación de archivos de texto, línea de comandos

Aprendizaje basado en simulaciones: servidor con software de juzgamiento, problemas algorítmicos, casos de prueba, internet.

Actividades de evaluación

El curso tendrá exámenes parciales escritos y de implementación. Las tareas son parte integral del proceso de aprendizaje en el curso. Las tareas son individuales e incluyen problemas conceptuales y de implementación.

	<i>Frecuencia</i>	<i>Liberación</i>	<i>Entrega</i>
<i>Tareas</i>	bi-semanal	lunes	2do domingo (conceptuales e implementación)

Cada estudiante desarrollará un proyecto de implementación con entregables durante las últimas semanas del semestre.

Actividades de evaluación y porcentajes

	<i>Parcial 1</i>	<i>Parcial 2</i>	<i>Examen final</i>	<i>Tareas</i>	<i>Proyecto</i>
<i>Porcentaje</i>	20	20	20	20	20

Actividades de evaluación por indicador de desempeño

	<i>1.1</i>	<i>1.3</i>	<i>1.4</i>	<i>2.3</i>	<i>2.4</i>	<i>3.2</i>	<i>3.4</i>	<i>6.2</i>	<i>6.4</i>
<i>Parcial 1</i>	10	15	10	20	15		5	5	20
<i>Parcial 2</i>	10	15	10	20	15		5	5	20
<i>Examen final</i>	10	20	15		15		10		30
<i>Proyecto</i>			20	20	20	20		20	
<i>Tareas</i>		20	10	20	15		10	10	15

Cronograma

SESIÓN	INICIO	TEMA	LECTURAS	ENTREGAS
1	01/29	Preliminares	0.1-0.4	
2	01/30	Análisis asintótico y Teorema Maestro	1.1-4	
3	02/05	Propiedades de la notación asintótica y el Teorema Maestro	2.3-4	
4	02/06	Dividir y conquistar	2.1	
5	02/12	Ordenamiento de arreglos	2.2-4	
6	02/13	Búsqueda binaria	2.5	
7	02/19	Bisección		Tarea 1
8	02/20	Programación dinámica; números de Fibonacci	3.1-2	
9	02/26	Una metodología; suma máxima de un subarreglo	3.3-4	
10	02/27	El problema del morral	3.5	
11	03/05	El problema del agente viajero	3.6	Tarea 2
12	03/06	Repaso		
13	03/12	PARCIAL 1*		
14	03/13	Algoritmos voraces; agendamiento de actividades	4.1	
15	03/19	Cubrimiento de intervalos	4.2	
16	03/20	Tiempo en el sistema	4.3	
17	03/26	Caminos en grafos	4.4	
18	03/27	Árboles de cubrimiento	4.5	Tarea 3
19	04/09	Reintentos; el problema de las 8 reinas	[Erickson 2.1]	
20	04/10	Sudoku		
21	04/16	Segmentación de texto	[Erickson 2.5]	
22	04/17	Secuencias crecientes maximales	[Erickson 2.6-7]	Tarea 4
23	04/23	PARCIAL 2*		
24	04/24	Teoría de números computacional	[Cormen 31.1-2]	
25	04/30	Teoría de números computacional		
26	05/01	Un marco universal basado en lenguajes; aceptación vs. decisión	5.2-3	
27	05/07	Algoritmos de decisión computables e indecidibilidad	5.4	
28	05/08	Las clases P y NP; P vs NP	6.5-6	Tarea 5
29	05/14	Reducciones polinomiales; NP-completitud		
30	05/15	SAT es NP-completo	[Cormen 34.4]	
31	05/21	Clique y Vertex-Cover son NP-completos	[Cormen 34.5]	
32	05/22	El agente viajero es NP-completo – Cierre	[Cormen 34.5]	
		EXAMEN FINAL*		

* Las fechas de los exámenes parciales y del examen final son susceptibles de cambios y se ajustarán de acuerdo con la programación de semanas de exámenes que establezca la Universidad.