

Árboles y Grafos Programa del Curso 2026-2

Profesores: Carlos Alberto Ramírez Restrepo y Carlos Delgado

Correo: carlosalbertoramirez@javerianacali.edu.co, carlos.delgado@javerianacali.edu.co

Página web: <https://www.carlosramirez.info/teaching/agra/2026-2>

Texto guía: *Introduction to Algorithms (Third Edition)* por T. Cormen, C. Leiserson, R. Rivest, C. Stein.

Textos adicionales:

- *Diseño y Análisis de Algoritmos* por C. Rocha (v0.0)
- *Algorithm Design* por J. Kleinberg, É. Tardos.
- *Algorithms, Etc.* por J. Erickson (primera edición, junio 2019)
- *Graph Theory and Complex Networks: An Introduction* por M. van Steen (primera edición, junio 2010)
- *Competitive Programming 4: The Lower Bound of Programming Contests in the 2020s* por S. Halim, F. Halim, S. Effendy (libro 1, julio 2020)

Lenguaje de programación: Python 3.7

Descripción

En esta asignatura se exploran las propiedades matemáticas de árboles y grafos como: (i) elementos computacionales generales para representar situaciones prácticas y con alto contenido algorítmico, y (ii) sus posibles realizaciones como estructuras de datos y utilización en algoritmos eficientes. Sobre los árboles y grafos se estudian problemas y algoritmos clásicos de búsqueda y optimización. También se presenta una introducción a redes complejas desde una perspectiva algorítmica y sus aplicaciones para analizar redes modernas como redes sociales, redes biológicas y *blockchain*.

En esta asignatura se: (i) apropian y aplican conocimientos de ciencias de la computación para resolver 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) se analizan algoritmos a través de demostraciones, y la interpretación de resultados experimentales de ejecuciones y simulaciones.

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 conocimientos de computación a través de la participación activa en diseño y análisis de algoritmos para identificar soluciones, correctas y eficientes, a problemas computacionales usando las estructuras árboles y grafos, y algoritmos asociados.

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 para resolver problemas de ingeniería en contextos específicos y diversos mediante el uso de árboles y grafos, y considerando impactos sociales y medioambientales.

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.

2.5 Modelar y analizar sistemas computacionales, empleando métodos modernos de análisis en ingeniería.

2.6 Descomponer sistemas en subsistemas de menor complejidad que interactúan entre sí, empleando para ello representaciones adecuadas.

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.

Los objetivos anteriores aplicados a los conocimientos impartidos en el curso le permitirán al estudiante:

1. Conocer las representaciones y operaciones principales de los árboles y grafos.

2. Implementar y evaluar algoritmos de búsqueda, rutas más cortas, cadenas y pattern matching.

3. Implementar, identificando sus ventajas, estructuras de datos arborescentes y jerárquicas como montones, conjuntos disyuntos y árboles de segmentos.

4. Modelar problemas reales con grafos (e.g., redes sociales, optimización industrial).

5. Conocer técnicas de programación paralela para solucionar problemas con estructuras de datos arborescentes y jerárquicas.

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.

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.

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

Porcentaje por actividad

	<i>Parcial 1</i>	<i>Parcial 2</i>	<i>Examen final</i>	<i>Tareas</i>	<i>Proyecto</i>
<i>Porcentaje</i>	25	25	25	12.5	12.5

Porcentaje por indicador de desempeño y actividad

	<i>1.1</i>	<i>1.3</i>	<i>1.4</i>	<i>2.3</i>	<i>2.4</i>	<i>2.5</i>	<i>2.6</i>	<i>3.2</i>	<i>3.4</i>	<i>6.2</i>	<i>6.4</i>
<i>Parcial 1</i>	5	15	5	15	10	15	10		10	5	10
<i>Parcial 2</i>	5	15	5	15	10	15	10		10	5	10
<i>Examen final</i>	10	25		15	15	20				5	10
<i>Proyecto</i>			10	15	20	10	10	25		10	
<i>Tareas</i>		15	10	10	10	10		10	15	10	10

Cronograma

SEMANA	INICIO	TEMA	LECTURAS	ENTREGAS
1	07/27-29	Repaso de estructuras de datos lineales y orden asintótico	[Rocha 1.4]	Tarea 1
2	07/31	Propiedades de la función O, Invariantes de Ciclo		
3	08/03-05	Invariantes de ciclo, Demostraciones O		
4	08/07	Ejemplos Invariantes de Ciclo		
5	08/10-12	Dividir y Conquistar	20.2 20.3 20.4	Tarea 2
6	08/14	Dividir y Conquistar		
7	08/17-19	Búsqueda Binaria		
8	08/21	Bisección		
9	08/24-26	Introducción Grafos: Generalidades y Representaciones		Tarea 3
10	08/28	Recorridos sobre grafos: DFS y BFS		
11	08/31 09/02	Grafos Implícitos		
12	09/04	REPASO PARCIAL		
	09/05	PARCIAL 1*		
13	09/07-09	Propiedades Recorridos DFS-BFS		
14	09/11	Orden Topológico		
	09/14-09/18	SEMANA DE REFLEXIÓN		
15	09/21-23	Conceptos Conectividad Grafos, Componentes Conextos		
16	09/25	Componentes Fuertemente Conexas		
17	09/28-30	Puntos de Articulación, Puentes		
18	10/02	Componentes Biconexos		
19	10/05-07	Caminos más cortos		
20	10/09	Caminos más cortos: BFS		
21	10/12-14	Caminos más cortos: Algoritmo de Bellman-Ford-Moore		
22	10/16	Caminos más cortos: Algoritmo de Dijkstra		
23	10/19-21	Caminos más cortos: Algoritmo de Floyd Warshall		
24	10/23	Ejemplos problemas		
	10/24	PARCIAL 2*		
25	10/26-28	Generalidades Árboles		
26	10/30	Díametro, Radio y Centro de Árboles		
27	11/02-04	Díametro, Radio y Centro de Árboles		
28	11/06	Árboles de Segmentos		
29	11/09-11	Árboles de Segmentos		
30	11/13	Estructuras Union-Find		
31	11/16-18	Árboles de Cubrimiento Mínimo		
32	11/20	Árboles de Fenwick		
33	11/23-25	Arreglos de Sufijos		
34	11/27	Ejemplos		
	11/28	EXAMEN FINAL*		
	11/30-12/02	Sustentación Proyecto	Entrega 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.