

## Programa del curso

**2021-19**

Nombre del curso:	Diseño y Análisis de algoritmos
Course Name:	
Créditos:	3
Profesor:	Nelson Andrés Sánchez Otálora
Horario:	08 de Junio al 31 de Julio
	M,I,J,V - 8 AM a 9:15 AM
Versión PDF	<a href="#">Click Aquí</a>

### Descripción

Presentar conceptos básicos del diseño y análisis de algoritmos. Al finalizar el curso, el estudiante debe ser capaz de aplicar técnicas de desarrollo de algoritmos como dividir y conquistar, programación dinámica, utilizar diversos algoritmos de búsqueda y analizar su complejidad en tiempo y en espacio. - **Semántica**

- Operacional
- Axiomática

Semántica operacional de un lenguaje de comandos guardados (GCL) definida sobre una máquina abstracta simple. Semántica axiomática presentada como un modelo teórico del cual es instancia la semántica operacional. - **Complejidad** Se elige estimar el desempeño de algoritmos basado en estimaciones de peor caso. Se suponen conocimientos de cálculo (uso de notación  $O$ ). Se suponen conocimientos de solución de ecuaciones de recurrencia (lineales o no -lineales reducibles a éstas mediante cambios de dominios). - **Derivación de algoritmos**

- Algoritmos sin ciclos
- Algoritmos con ciclos
- Algoritmos recurrentes

Presentación y práctica de una metodología de desarrollo de programas basada en especificaciones formales (pre- poscondiciones). - **Estrategias generales**

- Divide y vencerás
- Programación dinámica

Presentación de estrategias generales aplicables en situaciones conocidas. - **Ruta**

## óptima en grafos

- Algoritmos clásicos de optimización para ruteo óptimo en grafos

Ejemplo de problemas solucionables de manera óptima y conocida. - **Búsqueda en grafos**

- Algoritmos A, A\*

Solución genérica de problemas de búsqueda. Nivel introductorio. - **Intratabilidad**

- P : NP

Conceptos fundamentales. Nivel introductorio.

## Objetivos

### - Modelar

- Especificar programas
- Conocer límites de la algorítmica

### - Solucionar problemas de programación

- Diseñar Algoritmos
- Implementar algoritmos
- Documentar

### - Razonar Formalmente

- Analizar algoritmos
- Verificar programas

### - Trabajar en grupo

## Categorías de habilidades y objetivos pedagógicos

Objetivo pedagógico	Metas específicas	%
O4	Especificar programas	7%
O5	Conocer límites de la algorítmica	7%
O6	Analizar algoritmos	25%
O8	Diseñar Algoritmos	25%
O9	Derivar programas	11%
O10	Verificar programas	6%
O11	Diseñar Algoritmos	6%

O12	Implementar algoritmos	6%
O14	Documentar soluciones de problemas de programación	6%

## Objetivos pedagógicos transversales

Objetivo Pedagógico	Metas específicas
OT2	Analizar algoritmos
OT4	Trabajar en grupo
OT8	Autoaprender desarrollando

[tab title="Enlaces de interés"]

## Enlaces de interés

Las siguientes referencias son parte de [\[Boh2992\]](#):

- [1 Complejidad](#)
- [2 Ecuaciones de recurrencia](#)
- [3 Métodos generales de solución de problemas](#)
- [4 Programación dinámica](#)
- [5 Ruta más corta en grafos](#)
- [7 Búsqueda en grafos](#)
- [8 Intratabilidad](#)

## Evaluación

- Exámenes parciales : 70%

Parcial 1 : 25%  
Parcial 2 : 25%  
Parcial 3 : 20%

- Proyecto final : 15%
- Tareas y quizzes : 15%

Los porcentajes pueden variar, pero no por razones individuales, sino por conveniencia de todo el grupo. También puede haber bonificaciones especiales posteriores a la calificación grupal (de nuevo, dependientes de la conveniencia de todo el grupo); en estos casos puede haber notas que excedan la nota máxima de la escala. **Política de aproximación de notas finales**

- Durante el curso, las evaluaciones se califican con notas enteras, en la escala

0..100. Al final se calcula un promedio ponderado TOT, de acuerdo con las notas obtenidas en las diferentes evaluaciones. TOT es un número real, donde  $0 \leq TOT \leq 100$ .

- Se calcula el promedio ponderado PP tal que  $0 \leq PP \leq 5$ , un número real, como  $PP = TOT/20$ .
- La nota definitiva es un valor entre 1.5 a 5.0, en intervalos de 0.5. La forma en que se define este valor se explica a continuación:
- $ND = \min(5.0, \max(1.5, \text{entero}(2*PP+0.5)/2))$  donde  $\text{entero}(x)$  es el mayor entero menor o igual a  $x$ . Para múltiplos de 0.5 se aproxima al entero superior.
- Para aprobar el curso es indispensable lograr una nota definitiva ND, de 3.0 o superior.

## Bibliografía

**Texto guía:** [Cor2009] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Introduction to algorithms, MIT Press, 3a. ed., 2009. **Literatura recomendada:**

- [Boh2992] Bohórquez, J., Cardoso, R., Análisis de algoritmos, Tercera versión preliminar, Universidad de los Andes, Departamento de Sistemas y Computación, Enero 1992.
- [Bra1996] Brassard, G, Bratley, T., Fundamentos de algoritmia, Prentice-Hall, 1997.
- [Car1993] Cardoso, R., Verificación y desarrollo de programas, Ediciones Ecoé-Uniandes, 1993.
- [Coh2990] Cohen, E. Programming in the 1990s. Springer Verlag, 1990.
- [Gri1981] Gries, D., The science of programming, Springer Verlag, 2a. impresión, 1983.
- [Gri1993] Gries, D., A logical approach to discrete Math, Springer Verlag, 1993.
- [Hor1978] Horowitz, E., Sahni, S., Fundamental Algorithms, Computer Science Press, 1978.
- [Kal1991] Kaldewajj, A., Programming: the derivation of algorithms, Prentice-Hall, 1990.
- [Par1995] Parberry, I., Problems on algorithms, Prentice-Hall, 1995.