



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE  
ESCUELA DE INGENIERIA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

**Diseño y Análisis de Algoritmos - IIC2283**  
**Programa de Curso**  
**2<sup>do</sup> semestre - 2019**

**Horario cátedra** : Lunes y Miércoles módulo 5 sala BC21  
**Horario ayudantía** : Viernes módulo 5 sala E16  
**Profesor** : Marcelo Arenas (marenas@ing.puc.cl)  
**Ayudantes** : Antonio Lopez (alopez7@uc.cl), Pablo Messina (pamessina@uc.cl),  
Martín Muñoz (mmunos@uc.cl)  
**URL** : <http://marenas.sitios.ing.uc.cl/iic2283-19>

## Objetivo

El objetivo del curso es introducir al alumno a las técnicas básicas y algunas técnicas avanzadas tanto para el diseño como para el análisis de la complejidad computacional de un algoritmo. Se dará especial énfasis a la comprensión del modelo computacional sobre el cual se diseña y analiza un algoritmo, y al uso de estructuras de datos adecuadas para su implementación. Además, para cada una de las técnicas mostradas se estudiará algunos algoritmos que permiten entender su potencial, poniendo énfasis en la variedad e importancia de las áreas donde estos algoritmos son utilizados.

## Metodología

El curso se basa en clases expositivas de 80 mins. cada una. Se realizará un promedio de 2 clases semanales. Además de las clases expositivas, durante el semestre se realizará una ayudantía semanal, con el objetivo de reforzar o guiar el trabajo de los alumnos.

## Evaluación

La evaluación del curso estará basada solo en tareas, en las cuales los alumnos programaran en Python algoritmos para resolver distintos tipos de problemas, y donde se utilizará las técnicas aprendidas en el curso. Se realizará al menos cuatro tareas, y la nota final del curso será el promedio de ellas.

## Contenido

1. Introducción

- a) El proceso de diseño y análisis de un algoritmo
- b) Algunos ejemplos: ordenación y multiplicación de números enteros
2. Análisis de la eficiencia de un algoritmo
  - a) Notaciones asintóticas
  - b) Ecuaciones de recurrencia: cambio de variables, inducción constructiva, teorema maestro
  - c) Análisis de la complejidad de un algoritmo en el peor caso
3. Técnicas fundamentales de diseño de algoritmos
  - a) Dividir para conquistar
  - b) Programación dinámica
  - c) Algoritmos codiciosos
4. Transformaciones de dominio
  - a) Representación, evaluación e interpolación de polinomios y la transformada rápida de Fourier
5. Algoritmos aleatorizados
  - a) Algoritmos de Monte Carlo: igualdad de polinomios
  - b) Algoritmos de Las Vegas: cálculo de la mediana de una lista
  - c) Hashing universal y hashing perfecto
6. Algoritmos en teoría de números
  - a) Aritmética modular
  - b) Algoritmos básicos: exponenciación rápida, cálculo del máximo común divisor, el algoritmo de Euclides extendido y el cálculo del inverso modular
  - c) Un algoritmo de Monte Carlo para la verificación de primalidad
  - d) Un algoritmo de Las Vegas para el cálculo de raíces modulares
7. Técnicas para demostrar cotas inferiores: mejor estrategia del adversario, árboles de decisión y reducciones
8. Análisis de la eficiencia de un algoritmo más allá del peor caso
  - a) Análisis en el caso promedio: Quicksort
  - b) Análisis amortizado de algoritmos

## Bibliografía

1. Transparencias de clases.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest y Clifford Stein. *Introduction to Algorithms*, 3ª edición. MIT Press, 2009.
3. Gilles Brassard y Paul Bratley. *Algorithmics: Theory and Practice*, 1ª edición. Prentice Hall, 1988.
4. Rajeev Motwani y Prabhakar Raghavan. *Randomized Algorithms*, 1ª edición, 1995.
5. Jon Kleinberg y Éva Tardos. *Algorithm Design*, 1ª edición. Pearson, 2005.
6. Michael Mitzenmacher y Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.