

Técnicas y Prácticas de Programación

(Laboratorio de Programación)

Luis Garreta

luis.garreta@javerianacali.edu.co

Ingeniería de Sistemas y Computación
Pontificia Universidad Javeriana – Cali

26 de julio de 2017

Información Básica

- ▶ Créditos: 3
- ▶ Horas de trabajo acompañado: 5 / semana (3 horas clase, 2 horas taller)
- ▶ Horas de trabajo independiente: 4 / semana
- ▶ Prerequisitos:
- ▶ Tipo de curso: Núcleo de Formación Fundamental.

Descripción del Curso

- ▶ Se abordan los principales conceptos y técnicas que se aplican a los algoritmos
- ▶ Las estrategias de validación y de solución de problemas
- ▶ Los principales conceptos de control de errores y definición de casos de pruebas
- ▶ Los conceptos fundamentales de construcción de programas.

El curso desarrollará unos sólidos conceptos teóricos acompañados de prácticas semanales de los temas vistos en clase.

Objetivos (Parte I)

- ▶ Conocer los diferentes comportamientos de los algoritmos:
 - ▶ Realizar análisis asintóticos de límites de complejidad superior y esperados.
- ▶ Establecer la clases de complejidad de un algoritmo:
 - ▶ Constante, logarítmica, lineal, cuadrática o exponencial
- ▶ Utilizar diferentes estrategias para la construcción de algoritmos:
 - ▶ Divide y vencerás
 - ▶ Algoritmos numéricos simples,
 - ▶ Algoritmos de búsqueda secuenciales y binarios, entre otros.
- ▶ Conocer los conceptos de:
 - ▶ Validación de entrada,
 - ▶ Verificación de programas,
 - ▶ Computación simultánea múltiple,
 - ▶ Objetivos de paralelismos vs concurrencia
 - ▶ Comparación informal de la eficiencia de un algoritmo.

Objetivos (Parte II)

- ▶ Definir y utilizar estructuras básicas de datos:
 - ▶ Arreglos y las cadenas de caracteres.
- ▶ Identificar los diferentes tipos de errores en un programa:
 - ▶ sintaxis, lógica, en tiempo de ejecución
- ▶ Conocer el concepto de una especificación y la programación defensiva :
 - ▶ Codificación segura, manejo de excepciones
- ▶ Mejorar la calidad de los programas a través de:
 - ▶ La utilización de las revisiones de código y la generación casos de pruebas.
- ▶ Identificar el papel y el uso de contratos
 - ▶ incluyendo pre- y post-condiciones.
- ▶ Conocer los principios/paradigmas de diseño :
 - ▶ Diseño estructurado, análisis y diseño orientado a objetos, diseño orientado a eventos, diseño a nivel de componentes
- ▶ Aplicar en los procesos de codificación diferentes técnicas:
 - ▶ Prácticas de codificación defensiva, prácticas de código seguro y estándares de codificación.

Competencias técnicas específicas que se desarrollan

- Lenguaje de programación C++ (intermedio).

Contenido

- ▶ Capítulo 1: Introducción y Conceptos de algoritmos (15 + 8 horas)
- ▶ Capítulo 2: Estrategias algorítmicas (20 + 16 horas)
- ▶ Capítulo 3: Validación de algoritmos (5+4 horas)
- ▶ Capítulo 4: Estrategias de Solución de problemas y estructuras básicas de datos (15 + 12 horas)
- ▶ Capítulo 5: Comprensión y correctitud de programas (20 + 16 horas)
- ▶ Capítulo 6: Principios de desarrollo de sistemas (10 + 8 horas)

Reglas del Curso

- ▶ **Asistencia:**
 - ▶ Obligatoria
- ▶ **Calificación:**

Instrumento	Porcentaje
Tareas y Talleres	15 %
Exposición	10 %
Parcial 1	20 %
Parcial 2	25 %
Exámen Final	30 %

Bibliografía

1. **Cormen, Leiserson, Rivest, and Stein. Introduction to Algorithms, 3rd Edition (MIT Press). 2009**
2. S. Dasgupta, C. H. Papadimitriou, and U. Vazirani. Algorithms. McGraw-Hill Science/Engineering/Math, 2006.
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to algorithms. MIT Press, Cambridge, MA, USA, 2001.
4. S. S. Skiena. The algorithm design manual. Springer-Verlag New York, Inc., New York, NY, USA, 1998.
5. **Weiss, Mark Allen. Data structures and algorithm analysis in C++ / Mark Allen Weiss, Florida International University. — Fourth edition.**
6. Roger S. Pressman. Ingeniería de Software, un enfoque práctico. 7ed. McGraw Hill. México, 2010.
7. Ian Sommerville. Ingeniería de Software. 9ed. Pearson Educación S.A. México, 2011.