

Temario de Algoritmos Computacionales

Diego Alberto Barceló Nieves
Facultad de Ciencias
Universidad Nacional Autónoma de México

El siguiente temario está basado en el [temario oficial del curso](#). Al elaborarlo, asumimos que l@s estudiantes del curso no tienen experiencia previa con programación, pero sí tienen bases teóricas sólidas de álgebra lineal y cálculo diferencial e integral de una variable, así como nociones básicas de ecuaciones diferenciales ordinarias, pues esto último será necesario en el módulo 3. Para los módulos 1, 2 y 3 utilizaremos el lenguaje de programación Julia. La duración aproximada de cada módulo se indica en paréntesis.

0. Introducción a la programación (2 semanas)

1. ¿Qué es un programa? (Paradigma imperativo de la programación)
2. ¿Cómo se ejecuta un programa? (Lenguaje de programación, código fuente y sintáxis, comentarios y mensajes de error)
3. Licencias: legalidad y ética. (Software privativo, software de código abierto y software libre)
4. ¿Cómo creo un programa? (Editor de texto y terminal virtual, REPLs e IDEs)
5. ¿Cómo aprendo a programar? (Manuales, documentación y foros de preguntas)
6. Herramientas útiles para hacer programación. (Jupyter y Pluto, Git y GitHub/GitLab)

1. Estructura básica de la programación (3 semanas)

1. Algoritmos y diagramas de flujo.
2. Conceptos fundamentales. (Operaciones aritméticas, asociatividad y precedencia, tipos de datos, arreglos, variables y constantes, funciones, manejo de memoria, recursividad)
3. Estructura lógica. (Operaciones lógicas, condicionales y ciclos)
4. Sistemas numéricos de punto flotante y error numérico. (Épsilon de máquina y propagación de errores)
5. Métodos numéricos. (Estabilidad y convergencia).
6. Estructura de la programación modular. (Bibliotecas)

2. Representaciones visuales (2 semanas)

1. Gráfica de funciones, visualización de datos y animación con Plots.
2. Manipulación de imágenes digitales con JuliaImages.

3. Cómputo científico: construcción de pseudocódigo e implementación en código (8 semanas)

1. Solución de sistemas lineales de ecuaciones algebraicas. (Método de eliminación Gaussiana)
2. Aproximación de raíces. (Método de Newton)
3. Solución de ecuaciones diferenciales ordinarias. (Método de Euler)
4. Caminante aleatorio.

4. Introducción a otros lenguajes de programación (1 semana)

1. GNU Octave.
2. R.