

## MÉTODOS COMPUTACIONALES AVANZADOS

Jaime E. Forero Romero

NOMBRE DEL CURSO: Métodos Computacionales  
CÓDIGO DEL CURSO: FISI 2028 (Magistral) FISI 2809 (Laboratorio)  
UNIDAD ACADÉMICA: Departamento de Física  
PERIODO ACADÉMICO: 201620  
HORARIO (MAGISTRAL): Mi 12:30 a 14:00 y Vi, 15:30 a 17:00  
HORARIO (LABORATORIO): Ju

---

NOMBRE PROFESOR(A) PRINCIPAL: Jaime E. Forero Romero  
CORREO ELECTRÓNICO: [je.forero@uniandes.edu.co](mailto:je.forero@uniandes.edu.co)  
HORARIO Y LUGAR DE ATENCIÓN: Mi 14:00 a 16:00, Oficina Ip208

NOMBRE PROFESOR(A) COMPLEMENTARIO(A):  
CORREO ELECTRÓNICO: [@uniandes.edu.co](mailto:@uniandes.edu.co)

---

### I Introducción

Los métodos computacionales fundamentan el trabajo en todas las áreas técnicas y científicas, ya se consideren principalmente experimentales o teóricas. Esto se debe en gran parte a que la capacidad de utilizar computadoras programables ha disminuido en costo monetario y en complejidad.

Este curso busca guiar a los estudiantes en el desarrollo de sus *habilidades computacionales* para trabajar con computadores en generar y procesar datos para obtener información sobre la realidad que esos datos pretenden describir. Estos datos pueden ser mediciones o simulaciones de sistemas físicos, químicos, geológicos, biológicos, financieros o industriales, entre otros.

El programa del curso tiene dos componentes diferenciadas. La parte de métodos numéricos y la parte de *carpintería* de software. La parte de métodos numéricos ilustra como pasar de alguna pregunta sobre la realidad a un formalismo matemático general, luego a una descripción numérica y de ahí a su implementación en términos de software. La parte de carpintería busca presentar algunas técnicas mínimas para que los resultados del último paso sea confiables y permitan obtener resultados reproducibles.

### II Objetivos

Los objetivos principales del curso son presentar algoritmos y técnicas básicas para:

- resolver ecuaciones diferenciales ordinarias, parciales y estocásticas.
- analizar y describir datos con técnicas estadísticas.
- desarrollar esquemas reproducibles para el análisis de datos científicos.

### III Competencias a desarrollar

Al finalizar el curso, se espera que el estudiante esté en capacidad de:

- Manejar dos lenguajes de programación modernos de computación numérica: uno de alto nivel (Python) y otro de bajo nivel (C).
- Implementar algoritmos sencillos para la resolución de ecuaciones diferenciales y para el análisis estadístico exploratorio de datos.

- Desarrollar un esquema computacional reproducible para simular sistemas y analizar datos.

## IV Contenido por semanas

**Semana 1.** Unix. Repositorios. Makefiles. C. Python. Norma IEEE para aritmética de punto flotante.

**Semana 2.** Integración de ecuaciones diferenciales ordinarias. Método de Euler (explícito e implícito), Métodos de Runge-Kutta, paso de tiempo adaptativo, leapfrog, integradores simplécticos.

**Semana 3.** Ecuaciones diferenciales parciales. Diferentes esquemas de solución. Mecánica de fluidos. Leyes de conservación hiperbólicas. Riemann Solvers. Método de Godunov.

**Semana 4.** Generadores de números aleatorios. Integración. Cadenas de Markov.

**Semana 5.** Ecuaciones diferenciales estocásticas. Convergencia. Estabilidad.

**Semana 6.** Termodinámica de sistemas simples. Estimación de parámetros. **Entrega Taller 1.**

**Semana 7.** Aprendizaje Supervisado. Árboles de Decisión, Clasificación, Ranking Regresión, Redes Neuronales.

**Semana 8.** Clustering: k-means, Jerárquico y Maximización del valor esperado.

**Semana 9.** Reducción de Dimensionalidad y Predicción Estructurada: Principal Component Analysis (PCA), Multi-dimensional Scaling (MDS), Linear Discriminant Analysis (LDA); Modelos Gráficos, Redes Complejas y Análisis de datos Topológico.

**Semana 10. Receso.**

**Semana 11.** Procesamiento Natural de Lenguaje (Naïve Bayes) y Aprendizaje Profundo (Deep Learning).

**Semana 12.** Fundamentos de programación en paralelo. Taxonomía de arquitecturas para cómputo en paralelo. Máquinas para cómputo en paralelo en Uniandes. **Entrega Taller 2.**

**Semana 13.** Fundamentos de programación en paralelo: OpenMP

**Semana 14.** Fundamentos de programación en paralelo: MPI

**Semana 15.** Fundamentos de programación en paralelo: MPI

**Semana 16.** Cómo desplegar máquinas de cómputo en La Nube. Ejemplos de aplicación en Amazon Web Services y Docker. **Entrega Taller 3 al final de la primera semana de finales.**

## V Metodología

Cada clase tendrá una corta presentación teórica (30 minutos aproximadamente) para pasar a practicar todos los conceptos directamente en la computadora/cluster a través de ejercicios de práctica (50 minutos aproximadamente).

## VI Criterios de evaluación

El curso tendrá tres entregas de trabajos, cada una con un valor del  $(100/3)\%$  de la nota definitiva. Los temas de las entregas serán los siguientes:

1. Ecuaciones diferenciales (ordinarias, parciales, estocásticas).
2. Machine Learning.
3. Cómputo en paralelo (OpenMP, MPI).

## VII Bibliografía

Bibliografía principal:

- R. L. Burden, J. D. Faires. *Numerical analysis*, 2011. (Biblioteca General - 519.4 B862 2011)
- A. Tveito, H.P. Langtangem, B.F. Nielsen., *Elements of Scientific Computing*, 2010. (Biblioteca General, Recurso Electrónico 510. )
- O. Maimon and L. Rokach, *The Data Mining and Knowledge Discovery Handbook*, 2010. (Biblioteca General, Recurso Electrónico 006.312)
- M. Snir, *MPI : the complete reference*, 1996. (Biblioteca General, 004.35 M637)

- J. Sanders, E. Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*, 2010. (Biblioteca General - 005.275 S152)

Bibliografía complementaria:

- D. Conway and J. M. White. *Machine learning for hackers*, 2012.
- S.Bird. *Natural Language Processing with Python*, 2009.
- Theano Development. *Deep Learning Tutorial* <http://deeplearning.net/tutorial/>