

MÉTODOS COMPUTACIONALES

Programa desarrollado por: Jaime E. Forero Romero

NOMBRE DEL CURSO: Métodos Computacionales en ciencias

CÓDIGO DEL CURSO: FISI 2028 (Magistral) FISI 2029 (Laboratorio)

UNIDAD ACADÉMICA: Departamento de Física

PERIODO ACADÉMICO: 201820

HORARIO (MAGISTRAL - SECCIÓN 1): Ma y Ju, 6:30 a 7:50.

HORARIO (MAGISTRAL - SECCIÓN 2): Mie y Vie, 12:30 a 13:50.

HORARIO (LABORATORIO - SECCIONES 1 Y 3): Ma 15:30 - 16:50

HORARIO (LABORATORIO - SECCIÓN 2): Mie 14:00 - 15:20

NOMBRE PROFESOR MAGISTRAL (SECCIÓN 1): Veronica Arias Callejas

CORREO ELECTRÓNICO: v.arias@uniandes.edu.co

HORARIO DE ATENCIÓN: con cita previa.

NOMBRE PROFESOR MAGISTRAL (SECCIÓN 2): Manu Forero Shelton

CORREO ELECTRÓNICO: anforero@uniandes.edu.co

HORARIO Y LUGAR DE ATENCIÓN: con cita previa

I Introducción

Los métodos computacionales son un aspecto inseparable de cualquier área de trabajo en ciencia e ingeniería. Esto se debe en gran parte a la disminución en costos y complejidad de uso de las computadoras programables, unido al aumento exponencial en su capacidad de procesamiento. Este curso busca guiar a los estudiantes en el desarrollo de sus *habilidades computacionales* 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: métodos numéricos y *carpintería* de software. La parte de métodos numéricos busca ilustrar el paso entre la formulación de una pregunta sobre la realidad en términos matemáticos y su descripción numérica/computacional, para mostrar posibles formas de escribir esa pregunta con software. La parte de carpintería busca presentar algunas técnicas y prácticas necesarias (no suficientes) para poder obtener resultados computacionales reproducibles.

II Objetivos

El objetivo principal del curso es presentar algoritmos y técnicas básicas para:

- Resolver ecuaciones diferenciales ordinarias y parciales
- Analizar y describir datos con técnicas estadísticas basadas en métodos Monte Carlo,
- 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 y otro de bajo nivel,
- Implementar algoritmos sencillos para la resolución de ecuaciones diferenciales y para el análisis estadístico exploratorio de datos,
- Desarrollar un esquema sencillo para obtener resultados computacionales reproducibles.

IV Contenido por semanas

Semana 1

- Temas: Presentación del curso. Unix. Consola. Comandos básicos. Editores de texto. Git. Github. Intérprete de Python. Variables. Aritmética. Listas. Diccionarios.
- Lecturas preparatorias. Videos `Introducción a Unix: primera parte`, `Introducción a Unix: segunda parte`, `Introducción a Python: primera parte`, `Python: listas y strings`. Videos de la serie `Version Control with Git` en Software Carpentry

Semana 2

- Temas: If/while/break/continue. Ciclos. Funciones. Variables globales y locales. Arreglos (numpy). Lectura y escritura de archivos (numpy). Gráficas y visualización (matplotlib), Objetos. Unit Tests
- Lecturas preparatorias. Videos `Intruducción a Python: segunda parte`, `Programación Orientada a Objetos en Python`, `Python: Funciones, tipos de variables y recursividad.`, `Numpy con IPython`, `import matplotlib as plt` (los dos últimos videos usan Ipython, no es necesario que uds. lo hagan asi.).

Semana 3

- Temas: Integrales. Interpolación.
- Lecturas preparatorias: Capítulo 6 (Integration) del libro de Landau.

Semana 4

- Temas: Derivadas. Raíces de ecuaciones.
- Lecturas preparatorias: Capítulos 7.I (Numerical Differentiation) y 7.II (Trial-and-Error Searching) del libro de Landau.

Semana 5

- Temas: Solución de sistemas de ecuaciones lineales.
- Lecturas preparatorias: Capítulo 8 (Matrix Equation Solutions) del libro de Landau.

Semana 6

- Temas: Autovalores y autovectores. Principal Component Analysis.
- Lecturas preparatorias: Secciones 6.3.1 (Principal Component Regression) y 10.2 (Principal Component Analysis) del libro ISL.

Semana 7

- Temas: Transformadas de Fourier
- Lecturas preparatorias: Capítulo 10 (Fourier Analysis) del libro de Landau.

Semana 8

- **PARCIAL**
- Temas: C++. Introducción, sintaxis, compilar/ejecutar, variables, ciclos.
- Lecturas preparatorias: C++ Tutorial

Semana 9

- Temas: Makefile, C++. If/while, Funciones, arreglos, Pointers, input/output
- Lecturas preparatorias: C++ Tutorial y Videos de la serie `Automation and Make` en Software Carpentry.

Semana 10

- Temas: Ecuaciones diferenciales ordinarias.

- Lecturas preparatorias: Capítulo 9 (ODEs) del libro de Landau.

Semana 11

- Temas: Ecuaciones diferenciales parciales.
- Lecturas preparatorias: Capítulo 17 (PDEs) del libro de Landau.

Semana 12

- Temas: Ecuaciones diferenciales parciales.
- Lecturas preparatorias: Capítulo 17 (PDEs) del libro de Landau.

Semana 13

- Temas: Conceptos básicos de probabilidad y estadística. Procesos aleatorios.
- Lecturas preparatorias: Capítulo 1 (The Basics) del libro de Sivia&Skilling.

Semana 14

- Temas: Métodos de Monte Carlo.
- Lecturas preparatorias: Capítulo 1 (Monte Carlo Methods) del libro de Krauth.

Semana 15

- Temas: Métodos de Monte Carlo.
- Lecturas preparatorias: Capítulo 1 (Monte Carlo Methods) del libro de Krauth.

Primera o segunda semana de finales

- **Examen FINAL**

V Metodología

Las *habilidades computacionales* se desarrollan trabajando activamente. Por esto en las sesiones magistrales, luego de presentar un resumen de los conceptos teóricos, se hará énfasis en la práctica y experimentación. Para que esto funcione es necesario que los estudiantes estudien el tema correspondiente **antes de cada clase** siguiendo las lecturas preparatorias recomendadas. En el *Laboratorio de Métodos Computacionales* habrá más tiempo para practicar lo visto en la clase magistral, hacer ejercicios y aclarar dudas.

La Magistral y el Laboratorio cuentan con repositorios en GitHub:

<https://github.com/ComputoCienciasUniandes/MetodosComputacionales>.

<https://github.com/ComputoCienciasUniandes/MetodosComputacionalesLaboratorio>.

VI Criterios de evaluación

Al comienzo del semestre se hará un examen (corto, sin nota) para diagnosticar el conocimiento general que ya tienen los estudiantes sobre los temas del curso.

La Magistral y el Laboratorio se califican por separado. Las componentes que reciben **calificación en la Magistral** son las siguientes:

- Cuatro talleres para resolver por fuera del horario de clase. Cada taller tiene un valor del 10% de la nota definitiva. Nota: deben asegurarse de que los códigos de estos talleres compilen y/o corran sin problema en los computadores de computufísica (Y-110-B).
- Ejercicios para resolver y entregar en clase. Se elegirán al azar 4 de estos ejercicios para ser calificados. El promedio de esas 4 notas tiene un valor del 25% de la nota definitiva. Esta contribución a la nota definitiva será de cero (0.0) si se dejaron de entregar **seis** o más de estos ejercicios.
- Un examen parcial (con una componente escrita y otra de programación) con un valor del 15% de la nota definitiva.
- Un examen final (con una componente escrita y otra de programación) con un valor del 20% de la nota definitiva.

Las componentes que reciben calificación en el **Laboratorio** son las siguientes:

- Ejercicios cortos para resolver y entregar en clase. Se elegirán al azar 6 de esos ejercicios para ser calificados. Cada ejercicio corresponde a un 15% de la nota definitiva del Laboratorio. **Esta contribución a la nota definitiva será de cero (0.0) si se dejaron de entregar tres o más de estos ejercicios.**
- Asistencia a clase. Cada falta a clase recibe una nota de 0.0 y cada asistencia recibe una nota de 5.0. El promedio de estas notas corresponde **al 10% de la nota definitiva.**

De acuerdo a la nota definitiva en el Laboratorio habrá **un bono** en la nota definitiva de la Magistral. Siendo x la nota de Laboratorio, el bono correspondiente se calcula así: $4.0 < x \leq 4.4 \rightarrow 0.1$, $4.4 < x \leq 4.8 \rightarrow 0.2$, $4.8 < x \leq 5.0 \rightarrow 0.4$.

NOTA IMPORTANTE: Todos los exámenes, talleres y ejercicios tanto del laboratorio como de la clase magistral serán **individuales**. Esto quiere decir lo siguiente:

- Los trabajos en clase y tareas los deben hacer sin ninguna ayuda externa (esto incluye colaboración con personas no inscritas en el curso, amigos, monitorías, etc.)
- No deben mostrar su código ni mirar códigos de compañeros o antiguos estudiantes de la clase.
- Copias (así les cambien el nombre a las variables o modifiquen la estructura) de cualquier código (de internet, de amigos, etc...) están completamente prohibidas.
- Como regla general, sus códigos usted los debe haber escrito solo y debe estar en capacidad de rehacerlos sin mirar ningún otro código. Esto se evaluará en los parciales y el final. **Si tiene alguna duda de si algo se puede o no hacer, pregunte. Y siempre indique con un comentario en su código si siente que debe citar el trabajo de otros.**
- Nota: para evitarse problemas, asegúrese siempre de borrar sus códigos de los computadores de las salas de biocómputo o computofísica.

Si en las entregas se detecta que el trabajo no fue individual (esto incluye colaboración con personas no inscritas en el curso), entonces **la nota de todos los talleres y ejercicios quedará automáticamente en cero** y se llevará el caso a comité disciplinario.

Las soluciones de los talleres se reciben a más tardar el lunes a las 11PM de la semana de entrega correspondiente. Todas las entregas de talleres y ejercicios se harán a través de SICUA. **No se aceptará ninguna tarea por fuera de esa plataforma**, a menos que ocurra una falla en los servidores de SICUA que afecte a **todos** los estudiantes del curso.

VII Bibliografía

Bibliografía principal:

- *A survey of Computational Physics - Enlarged Python Book*. R. H. Landau, M. J. Páez, C. C. Bordeianu. WILEY. 2012. <https://psrc.aapt.org/items/detail.cfm?ID=11578>
- *Data Analysis: A Bayesian Tutorial*. D. S. Sivia, J. Skilling. Second Edition, Oxford Science Publications. 2006.
- *Statistical Mechanics: Algorithms and Computations*. W. Krauth, Oxford Univ. Press.
- *An Introduction to Statistical Learning*. G. James, D. Witten, T. Hastie, R. Tibshirani, Springer. <http://www-bcf.usc.edu/~gareth/ISL/>
- *C++ programming for the absolute beginner*. M. Lee & D. Henkemans, Second Edition, Cengage Learning, 2009.
- *The C programming language*. B. Kernighan & D. Ritchie, Second Edition, Prentice Hall.

- Videos del curso Herramientas Computacionales que muestran los fundamentos de Unix y Python https://www.youtube.com/playlist?list=PLHQtzvthdVM_MGC9dPFKe4hPAwBd_7RJ3
- Software Carpentry: <http://software-carpentry.org/>
- C++ Tutorial: <https://www.tutorialspoint.com/cplusplus/>

Bibliografía secundaria:

- *Elements of Scientific Computing* Tveito A., Langtangen H.P., Nielsen B.F., Cai X. Springer. 2010.
- *Introduction to Computation and Programming Using Python*, Guttag, J. V. The MIT Press. 2013.
- <http://xkcd.com/>