

MÉTODOS COMPUTACIONALES

NOMBRE DEL CURSO: Métodos Computacionales
CÓDIGO DEL CURSO: FISI 2028 (Magistral) FISI 2029 (Laboratorio)
UNIDAD ACADÉMICA: Departamento de Física
PERIODO ACADÉMICO: 201920
HORARIO (MAGISTRAL): Ma y Ju, 12:30 a 13:50.
HORARIO (LABORATORIO - SECCIÓN 1): Ma 15:30 - 16:50
HORARIO (LABORATORIO - SECCIÓN 2): Mie 12:30 - 13:50

NOMBRE PROFESOR MAGISTRAL: Juan Pablo Mallarino Robayo
CORREO ELECTRÓNICO: jp.mallarino50@uniandes.edu.co
HORARIO Y LUGAR DE ATENCIÓN: Miercoles 9:00 a 12:00, Oficina I 105

NOMBRE PROFESOR LABORATORIO: Felipe Leonardo Gómez Cortés
CORREO ELECTRÓNICO: fl.gomez10@uniandes.edu.co
HORARIO DE ATENCIÓN: a disposición del profesor.

I Introducción

Los métodos computacionales son un aspecto inseparable de cualquier área de trabajo en ciencia e ingeniería. Esto se debe a la facilidad de acceso a computadoras programables y al aumento exponencial en su capacidad de procesamiento. Este curso busca guiar a los estudiantes en el desarrollo de sus *habilidades computacionales* para obtener información sobre la realidad que los datos generados por computador 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 matemática de una pregunta sobre la realidad y su descripción numérica/computacional. El objetivo principal es mostrar diferentes formas de usar un computador para formular esa pregunta. La parte de carpintería presenta algunas técnicas y prácticas necesarias para 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:

- implementar algoritmos sencillos para la resolución de ecuaciones diferenciales y para el análisis estadístico exploratorio de datos,
- manejar la sintaxis básica de dos lenguajes de programación modernos de computación numérica: uno de alto nivel y otro de bajo nivel,

- desarrollar un esquema sencillo para obtener resultados computacionales reproducibles.

IV Contenido por semanas

Semana 1. Presentación del curso. Binder. Repaso Unix. Repaso Python.

- Temas: Presentación del curso. Binder. Comandos básicos Unix. Sintaxis básica de Python.
- Lecturas preparatorias: Los siguientes videos de Herramientas Computacionales
 - Introducción a Unix: primera parte
 - Introducción a Unix: segunda parte
 - Introducción a Python: primera parte
 - Python: listas y strings.
 - Intruducción a Python: segunda parte

Semana 2. Repaso Python.

- Temas: Funciones de Python. Numpy y Matplotlib.
- Lecturas preparatorias: Los siguientes videos de Herramientas Computacionales
 - Python: Funciones, tipos de variables y recursividad.
 - Numpy con IPython
 - `import matplotlib as plt`

Semana 3

- Temas: Conceptos básicos de probabilidad y estadística. Procesos aleatorios.
- Lecturas preparatorias: Capítulo 5 (Monte Carlo Simulations) del libro de Landau.

Semana 4

- Temas: Métodos de Monte Carlo y Distribuciones de Probabilidad
- Lecturas preparatorias: Capítulo 1 (The Basics) del libro de Sivia&Skilling. Capítulo 1 (Monte Carlo Methods) del libro de Krauth.

Semana 5

- Temas: Estimación bayesiana de parámetros.
- Lecturas preparatorias: Capítulo 2 (Parameter Estimation I) y Capítulo 3 (Parameter Estimation II) del libro de Sivia&Skilling.

Semana 6

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

Semana 7

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

Semana 8

- Temas: Solución de sistemas de ecuaciones lineales. Ajustes por mínimos cuadrados.
- Lecturas preparatorias: Capítulo 8 (Matrix Equation Solutions) del libro de Landau. Capítulo 1 (Fitting Functions to data) del libro de Hutchinson.

Semana 9

- Temas: Autovalores y autovectores. Análisis de Componentes Principales.

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

Semana 10

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

Semana 11

- Temas: Git. Github. Unit Tests.
- Videos de la serie **Version Control with Git** y **Python Testing** en Software Carpentry.

Semana 12

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

Semana 13

- Temas: C++. If/while, Funciones, arreglos, Pointers, input/output. Makefiles
- Lecturas preparatorias: C++ Tutorial. Videos de Software Carpentry sobre Makefiles.

Semana 14

- Temas: Ecuaciones diferenciales ordinarias.
- Lecturas preparatorias: Capítulo 2 del libro de Hutchinson. Capítulo 9 (Ordinary Differential Equations) del libro de Landau.

Semana 15

- Temas: Ecuaciones diferenciales parciales (Ecuación de Difusión).
- Lecturas preparatorias: Capítulo 4 (PDEs) del libro de Hutchinson. Capítulo 5 (Diffusion) del libro de Hutchinson. Capítulo 17 (PDEs) del libro de Landau.

Semana 16

- Temas: Ecuaciones diferenciales parciales (Ecuación de Poisson, Ecuación de Onda).
- Lecturas preparatorias: Capítulo 6 (Elliptic problems) del libro de Hutchinson. Capítulo 18 (PDEs) del libro de Landau.

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/jpmallarino/FISI2028-202010>.

<https://github.com/jpmallarino/FISI2029-202010>.

VI Criterios de evaluación

La Magistral y el Laboratorio se califican por separado. Las componentes que reciben calificación en la Magistral (en paréntesis su contribución a la nota definitiva) son las siguientes:

- Asistencia (10%). Cada asistencia a clase cuenta como una nota de 5.0 y una falta como 0.0. El promedio de esas notas será la nota de asistencia. Si hay **seis** o más fallas no justificadas durante todo el semestre esta nota es cero (0.0).
- Ejercicios (6% cada uno). En cada clase hay un ejercicio para entregar. Cada ejercicio tiene dos partes. La primera se publica al menos un día antes de la clase y debe resolverse por fuera de la magistral. La segunda se publica y se resuelve durante la magistral. Durante el semestre el profesor elegirá a su discreción diez (10) de estos ejercicios para ser calificados.

- Examen Final (30 %). Tendrá una componente escrita y otra de programación.

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

- Asistencia (30 %) Similar a la Magistral. Si hay **tres** o más fallas no justificadas durante todo el semestre esta nota es cero (0.0).
- Ejercicios (70 %). En cada clase hay un ejercicio para entregar. Se desarrollan y entregan en clase. Durante el semestre el profesor elegirá a su discreción cinco (5) de estos ejercicios para ser calificados. El promedio de esas notas será la nota de ejercicios. Si se dejaron de entregar **tres** o más ejercicios (sin justificación) durante todo el semestre esta nota es cero (0.0).

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 b se calcula así: $4,0 < x \leq 4,4 \rightarrow b = 0,1$; $4,4 < x \leq 4,8 \rightarrow b = 0,2$; $4,8 < x \leq 5,0 \rightarrow b = 0,3$.

Todos los exámenes y ejercicios son **individuales**. Si en las entregas se detecta que el trabajo no fue individual (esto incluye colaboración con personas no inscritas en el curso, i.e. a través de “monitorías”) se llevará el caso a comité disciplinario y la nota del curso queda como Pendiente Disciplinario hasta que el comité tome alguna decisión.

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>
- *A Student's Guide to Numerical Methods*. I. H. Hutchinson. Cambridge, 2015.
- *Data Analysis: A Bayesian Tutorial*. D. S. Sivia, J. Skilling. Second Edition, Oxford Science Publications. 2006.
- *Software Carpentry: Python Testing* <http://kathyhuff.github.io/python-testing/>
- *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*, Gutttag, J. V. The MIT Press. 2013.
- <http://xkcd.com/>