

FISI 2028, 2029 Métodos Computacionales (Magistral y Laboratorio) Semestre 201620**Magistral** - Martes y Jueves 7:00 - 8:30 am**Laboratorio** - Miércoles 1:00 - 2:30 pm

Salón Q508

Profesor: Jaime Forero, email: **je.forero**

Oficina, Monitores: Por definir

Horario de Atención: Con cita previa

Objetivo

El curso tiene como objetivo principal desarrollar los fundamentos de una buena *actitud computacional*. Esta buena *actitud* corresponde a un conjunto de habilidades 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, geológicos, biológicos, financieros o industriales, entre otros. Estas habilidades incluyen el aprendizaje de al menos dos lenguajes de programación, la familiarización con diferentes métodos numéricos, la implementación de técnicas de desarrollo de software y la práctica de una actitud escéptica ante resultados computacionales.

Metodología

Esa *actitud computacional* se desarrolla trabajando. 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**.

El programa del curso tiene dos componentes diferenciadas. La parte de métodos de cómputo numérico 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 prácticas necesarias para el desarrollo de software de calidad.

Esta materia se ve al mismo tiempo que el *Laboratorio de Métodos Computacionales*. El objetivo del Laboratorio es tener más tiempo para practicar lo visto en clase, hacer ejercicios y aclarar dudas.

Lenguajes

Se usarán principalmente dos lenguajes de programación: C y Python. Eventualmente utilizaremos notebooks de Jupyter. No se aceptarán tareas en Matlab, Mathematica, R, C++ o cualquier otro lenguaje de programación que no esté en la lista mencionada antes.

Evaluación - Laboratorio

En las sesiones de Laboratorio se planteará un total de cinco ejercicios cortos para desarrollar y entregar en clase. Cada ejercicio corresponde a un 20 % de la nota final. El repositorio de GitHub del Laboratorio es:

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

Todas las entregas 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.

Evaluación - Clase Magistral

Se darán cinco talleres y un bono para resolver y calificar, cada uno con un valor del 20 % de la nota definitiva. **Para poder entregar el bono es necesario responder un examen escrito (corto) sobre los temas teóricos vistos durante el semestre.** Al comienzo del semestre se hará un examen (sin nota) para diagnosticar el conocimiento general que ya tienen los estudiantes sobre los temas del curso.

Todos los talleres serán **individuales**. Si en las entregas individuales 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 quedará automáticamente **en cero**.

Todas las entregas 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.

De acuerdo a la nota definitiva en Laboratorio habrá **otro bono** en la nota definitiva de la Clase Magistral. Siendo x la nota de Laboratorio, el bono correspondiente se calcula así: $4,0 < x \leq 4,4 \rightarrow 0,1$, $4,4 < x \leq 4,8 \rightarrow 0,2$, $4,8 < x \leq 5,0 \rightarrow 0,4$.

El curso cuenta con un repositorio en GitHub:

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

El material se encuentra distribuido en las siguientes carpetas.

- `hands_on/`: Ejemplos para hacer en clase.
- `homework/`: Enunciados y calificaciones de las tareas.
- `notas/`: Notas de clase.
- `syllabus/`: Programa del curso.

Programa

Semana	Teoría	Carpintería	Taller (Mag.)	Ejercicio (Lab.)
1 (18.1)		Linux / Consola / Editores de texto	Exámen diagnóstico (sin nota)	
2 (25.1)		Python básico		#1
3 (1.2)		Python (objetos, numpy, matplotlib), Jupyter	#1	
4 (8.2)	Operaciones matriciales, sistemas de ecuaciones lineales, mínimos cuadrados			
5 (15.2)	Autovalores, autovectores, PCA y tensores			# 2
6 (22.2)	Interpolación, extrapolación y búsqueda de raíces		#2	
7 (29.2)	Transformada de Fourier			
8 (7.3)	Derivación e integración			#3
9 (21.3)	Semana de trabajo individual		# 3	
10 (14.3)		Git, GitHub		
11 (28.3)		C básico		#4
14 (18.4)		Makefiles y Testing	#4	
12 (4.4)	Ecuaciones diferenciales ordinarias			
13 (11.4)	Ecuaciones diferenciales parciales			#5
15 (25.4)	Estadística frecuentista. Estadística Bayesiana.		#5	
16 (2.5)	Markov Chain Monte Carlo		Exámen y Bono	

Lecturas/Actividades requeridas para cada semana.

1. A
2. B

Referencias Bibliográficas

- *Elements of Scientific Computing* Tveito A., Langtangen H.P., Nielsen B.F., Cai X. Springer. 2010.
- *A survey of Computational Physics* . R. H. Landau, M. J. Páez, C. C. Bordeianu. Princeton Univ. Press. 2006
- *Statistical Mechanics: Algorithms and Computations*. W. Krauth, Oxford Univ. Press.
- *Introduction to Computation and Programming Using Python*, Guttag, J. V. The MIT Press. 2013.

- *The C programming language*. B. Kernighan & D. Ritchie, Second Edition, Prentice Hall.
- <http://software-carpentry.org/>
- <http://people.ds.cam.ac.uk/nmm1/Fortran/index.html>
- <http://xkcd.com/>
- <https://www.khanacademy.org>