

**FISI 2028, 2029 Métodos Computacionales y Laboratorio** Semestre 2016-I**Magistral** - Martes y Jueves 7:00 - 8:30 am**Laboratorio** - Miércoles 1:00 - 2:30 pm

Salón Q508

Profesor: Germán Chaparro, email: **g.chaparro**

Oficina, Monitores: Por definir

Horario de Atención: Con cita previa

## Objetivo

El curso tiene como objetivo principal desarrollar en los estudiantes una adecuada *actitud computacional*, con la capacidad de discernir sobre los métodos y técnicas para solucionar cualquier problema computacional y entender sus limitaciones.

El foco de la clase será esta *actitud computacional* que corresponde al conjunto de habilidades para trabajar con computadores en generar y procesar datos para obtener intuición a partir de ellos. Estos datos pueden corresponder a mediciones o simulaciones sobre sistemas físicos, biológicos, financieros o industriales, entre otros.

## Metodología

Esa *actitud computacional* se desarrolla trabajando. Las sesiones de **Métodos Computacionales** serán, sobre todo enfocadas en la exploración, práctica y experimentación. Para que esto funcione es necesario que los estudiantes lean y exploren el tema correspondiente antes de cada clase.

El programa del curso tiene dos componentes diferenciados. La parte de métodos de cómputo numérico y la parte de *carpintería* de software. La parte de métodos numéricos ilustra como pasar de la matemática a la computación numérica, al igual que como se implementan algunos algoritmos en la práctica. La parte de carpintería de software aumentará la familiaridad con la producción científica de software de los asistentes.

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, desarrollando ejercicios y aclarando dudas.

## Software

Se usarán principalmente: notebooks de IPython complementados con FORTRAN 90/95 y C. También se aceptan tareas en los siguientes lenguajes de programación: Python y C++. No se aceptarán tareas en Matlab, Mathematica, R o cualquier otro lenguaje de programación que no este en la lista mencionada antes.

## Evaluación - Laboratorio

En las sesiones de Laboratorio se dejarán talleres cada 15 días. El repositorio de GitHub del Laboratorio es

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

La nota del Laboratorio será el promedio de las notas de los talleres entregados.

## Evaluación - Clase Magistral

Hay 7 talleres para entregar. **No habrá parciales ni examen final.** Todos los talleres serán **individuales**. Si en las entregas individuales se detecta que hubo trabajo en grupos entonces la nota de todos los talleres quedará automáticamente en cero (**0.0**).

Las entregas para los últimos 3 talleres se harán en dos tiempos: una primera entrega donde se muestre explícitamente un borrador del código con comentarios, luego la entrega definitiva con el código completo. La primera entrega es una condición necesaria para aceptar la segunda. Solamente la segunda entrega recibe una nota.

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

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
1 (18.1)		Linux / Consola / Editores de texto	
2 (25.1)		Python básico	# 1 (15 %)
3 (1.2)		Python (objetos, numpy, matplotlib), IPython	
4 (8.2)	Operaciones matriciales, sistemas de ecuaciones lineales, mínimos cuadrados		# 2 (15 %)
5 (15.2)	Autovalores, autovectores, PCA y tensores		
6 (22.2)	Interpolación, extrapolación y búsqueda de raíces		# 3 (15 %)
7 (29.2)	Transformada de Fourier		
8 (7.3)	Derivación e integración		# 4 (15 %)
9 (14.3)		Git, GitHub	
10 (21.3)	<b>Semana de trabajo individual</b>		
11 (28.3)		Lenguajes compilados, makefiles	#5 (10 %)
12 (4.4)	Ecuaciones diferenciales ordinarias (1er orden)		
13 (11.4)	Ecuaciones diferenciales ordinarias (2do orden)		#6 (15 %)
14 (18.4)	Ecuaciones diferenciales parciales		
15 (25.4)	Markov Chain Monte Carlo		#7 (15 %)
16 (2.5)	Cómputo paralelo		

## 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*, Gutttag, 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>