

FISI 2028 Métodos Computacionales Cod. 13834

Semestre 2015-I

Martes y Miércoles 15:30 - 16:50

Salón LL204

Profesor: Sebastián Pérez Saaibi, email: **spsaaibi**

Oficina: I120

Horario de Atención: calendly.com/spsaaibi

Monitores:

(Magistral) María Camila Remolina, email: **mc.remolina197**

(Magistral) Juan Felipe Mendez Valderrama, email: **jf.mendez53**

(Laboratorio) Juan Nicolás Garavito, email: **jn.garavito57**

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 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 lleguen a clase después de haber leído sobre el tema correspondiente.

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.

Software

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

Evaluación

Hay 8 talleres para entregar. Los primeros 4 valen el 10% cada uno, los siguientes 4 valen el 15% cada uno. **No habrá parciales ni examen final.** Los primeros cuatro talleres serán

individuales. Los últimos cuatro talleres serán en **parejas**. Si en las entregas individuales se detecta que hubo trabajo en grupos entonces la nota de todos los talleres individuales quedará automáticamente en cero (**0.0**).

Las entregas para los últimos 4 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.

Esta materia se ve al mismo tiempo que el *Laboratorio de Métodos Computacionales*. El objetivo del Laboratorio es tener más tiempo para practicar todo lo visto en clase. De acuerdo a la nota definitiva en Laboratorio habrá un bono en la nota definitiva de este curso. Siendo x la nota de Laboratorio, el bono correspondiente se calcula así: $4,0 < x \leq 4,4 \rightarrow 0,2$, $4,4 < x \leq 4,8 \rightarrow 0,3$, $4,8 < x \leq 5,0 \rightarrow 0,5$.

El curso cuenta con un repositorio en github: <https://github.com/forero/ComputationalMethods>. 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 (1.20-21)	Algoritmos	Consola/ SublimeText/ R (gráficas)	#1 (shell scripts)
2 (1.27-28)		Consola/SublimeText - C	
3 (2.3-4)		C	#2 (C)
4 (2.10-11)		Python, R	
5 (2.17-18)		IPython Notebook (gráficas) / Github Individual / Rmd Reports	#3 (Python)
6 (2.24-25)	Solución de sistemas lineales de ecuaciones		
7 (3.3-4)	Interpolación	Diseño de programas / Github Colaborando	#4 (Matrices, Git, C)
8 (3.10-11)	Análisis de Fourier - FFT	IPython / Numpy	
9 (3.17-18)	Integración y derivación numérica	IPython / Numpy	#5 (Fourier, Git, Python)
10 (3.24-25)	Ecuaciones diferenciales ordinarias (1er orden)		#6 (EDO1, Derivadas, Python, R, Git)
11 (3.31-4.1)	Semana de trabajo individual		
12 (4.7-8)	Ecuaciones diferenciales ordinarias (2do orden)	Makefile	
13 (4.14-15)	Ecuaciones diferenciales parciales		#7 (EDO2, EDP, Python, R, Git)
14 (4.21-22)	Métodos Monte Carlo		
15 (4.28-29)	MCMC para hacer fits		#8 (MCMC, Python, R)
16 (5.5-6)			

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://xkcd.com/>
- <https://www.khanacademy.org>