



## UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

### DEPARTAMENTO DE INFORMÁTICA

#### Identificación

Talle Libre: Programación Científica		
Créditos UTFSM: 1	Asignaturas Prerrequisitos: INF134 Estructura de Datos + MAT022 Matemáticas II	
Hrs. Cat. Sem.: 1:30 (hora reloj)	Horario: Miércoles 5-6 (Conversable)	
Cant. Sesiones: 10	Fecha Inicio: 14/03/2017	Fecha Término: 30/05/2017

#### Descripción

Este curso tiene por objetivo que el alumno aprenda a utilizar las herramientas que provee el lenguaje de programación Python, para su utilización en la resolución de problemas en la computación científica y data science.

#### Requisitos de entrada

- Manejo y conocimiento de los lenguajes Python y C.
- Conocimiento básico de Álgebra Lineal.
- (Recomendable) Conocimiento básico de Ecuaciones Diferenciales.

#### Competencias Específicas del Perfil de Egreso a las que contribuye (máximo 2, ver Anexo 1)

- Analizar problemas susceptibles de ser resueltos computacionalmente, diseñar algoritmos y programar las soluciones utilizando las herramientas adecuadas en cuanto a lenguaje de programación y estructuras de datos.

#### Competencias Transversales del Perfil de Egreso a las que contribuye (máximo 2, ver Anexo 2)

- Actuar con autonomía, flexibilidad, iniciativa, y pensamiento crítico al enfrentar problemáticas de la profesión.

**Objetivos** (Resultados del aprendizaje): Al aprobar la asignatura, el estudiante será capaz de:

1. Comprender el funcionamiento interno de Python en cuanto al uso y manipulación de arreglo numéricos.
2. Desarrollar código optimizado para la resolución de problemas científicos.
3. Analizar y detectar secciones críticas del código (cuellos de botellas) por medio de profiling.
4. Crear gráficos que permitan entender y visualizar fenómenos de distinto tipo.
5. Crear soluciones eficientes a problemas científicos reales.
6. Hacer interfaces en Python de bibliotecas en C (Wrapping).

## Metodología de enseñanza y de aprendizaje

Clases expositivas complementadas con:

- Ejemplos y demostraciones interactivas (tiempo real) en Jupyter Notebooks, donde se discutirán códigos asociados a las diversas temáticas del curso.
- Desafíos/Actividades individuales y grupales, donde se incentivará la aplicación de lo aprendido.
- Material teórico online para su consulta fuera de la clase, con las demostraciones vistas en clases, así como con ejemplos extra que no se alcancen a cubrir.

## Bibliografía sugerida

1. Learning IPython for Interactive Computing and Data Visualization, second edition. Cyrille Rossant.
2. IPython Interactive Computing and Visualization Cookbook. Cyrille Rossant.

## Evaluación

Nota Final = Asistencia \* 20% + Actividades en Clases \* 50% + Mini Proyecto Final \* 30%

## Programación semestre

Sesión Nº	Nombre	Tipo Actividad
1	Introducción a Anaconda, Ipython, Jupyter Notebook y Git.	Clase práctica para conocer y configurar las herramientas utilizadas en el curso.
2	Computación Numérica (1): NumPy + NumExpr.	Estudio de los mecanismos internos de Python/NumPy, y su utilización práctica.
3	Computación Numérica (2): NumPy + Manipulación de array de gran tamaño.	Estudio de problemas de gran escala y herramientas para tratarlos: Matrices Sparse, HDF5, Memory Mapping, entre otras.
4	Visualización: Matplotlib + Ipython Widgets.	Visualización interactiva de datos.
5	Acelerando Python con Numba: Just In Time Compilation.	Exposición de la compilación en tiempo real y aplicaciones.
6	Acelerando Python con Cython: Escribiendo C en Python.	Desarrollo de código C en Python.
7	Profiling de Código y Optimización.	Detección de códigos críticos y su optimización.
8	Resolviendo Problemas Numéricos:	Utilizar las herramientas de

	SciPy	SciPy para resolver problemas numéricos.
9	Modulos de C a Python: Wrapping. <sup>1</sup>	Estudio de caso real: Crear un interfaz en Python de C.
10	Paralelización en Python: Threads. <sup>2</sup>	Exposición básica de Threads y cómo utilizarlos en Python.

Elaborado por	Martín Villanueva	Observaciones:
Aprobado por		
Fecha Aprobación		

1 Contenido será visto de forma opcional, sólo si alcanza el tiempo.

2 Contenido será visto de forma opcional, sólo si alcanza el tiempo.