



## PROGRAMA DE ASIGNATURA<sup>1</sup>

NOMBRE ASIGNATURA: **Computación Científica con Python**

Código: **m31**

### Identificación general

Docente responsable	Pablo HUIJSE	Docentes colaboradores	
Correo electrónico	<a href="mailto:phuijse@inf.uach.cl">phuijse@inf.uach.cl</a>	Correo electrónico	
Horario y sala de clases	Por definir (4 horas pedagógicas a la semana, 2 teóricas y 2 prácticas).		
Año y semestre	2019 – Primer semestre		

Antecedentes de la asignatura, según proyecto curricular de la carrera									
Unidad Académica	Instituto de Informática		Carrera	Ingeniería civil en Informática			Semestre en plan de estudios		Semestre V
Asignaturas- requisito (con código)	<b>Métodos numéricos para ingeniería (BAIN087)</b> <b>Estadística y probabilidades para ingeniería (BAIN091)</b>						Créditos SCT-Chile		4
Horas cronológicas semestre	Teóricas presenciales	25.5	Prácticas presenciales	25.5	Trabajo Autónomo	51	Total		102
Ciclo formativo	Bachillerato		Licenciatura	X	Profesional				
Área de formación	Especialidad		General		Vinculante-prof esional		Optativa		X
Descripción de la asignatura	La asignatura de "Computación científica con Python" introduce al estudiante conceptos básicos de ciencia de datos preparándolo para resolver problemas científicos aplicados usando las herramientas libres de computación numérica que ofrece el lenguaje de programación Python 3. En particular el estudiante aprenderá a leer, manipular y preprocesar datos, crear visualizaciones a partir de datos, resolver problemas numéricos de álgebra lineal y optimización, extraer información mediante inferencia estadística básica y entrenar modelos sencillos para hacer regresión, clasificación y predicción.								

**Aporte de la asignatura al Perfil de Egreso, según proyecto curricular de la carrera**

Competencias	Nivel de dominio que alcanza la competencia en la asignatura							
-Específicas:	Básico		Medio		Superior		Avanzado	
<b>Nº4- Aplicar principios de la ciencias de la computación, para el manejo de la información y conocimiento</b> d.1- Emplea principios, teorías, modelos y técnicas de ciencias de la computación para manejar información.				X				
<b>Nº5- Desarrollar soluciones robustas y eficientes que manejan información y conocimiento, considerando un enfoque sistémico e integrando teoría y práctica</b> d.1- Diseña programas computacionales considerando un enfoque sistémico e integrando teoría y práctica				X				
-Genéricas:	Básico		Medio		Superior		Avanzado	
<b>Nº3- Manifestar una actitud innovadora, emprendedora y de adaptación al cambio en contextos globales y locales del ejercicio de la Ingeniería Civil en Informática.</b> d.1- Examina escenarios que ejemplifican acciones asociadas a la innovación, emprendimiento y cambio, en el contexto de las experiencias formativas que la UACH ofrece a los estudiantes				X				

-Sello:	Básico		Medio		Superior		Avanzado	
<b>Nº4- Evidenciar habilidades para trabajar en forma autónoma, en el contexto formativo del desarrollo personal y profesional del estudiante con sello UACH</b>  d.1- Planifica el tiempo disponible en función de las diversas actividades de tipo académico y personales que debe realizar, en el contexto de las experiencias formativas de la UACH.				X				

Programación por Unidades de Aprendizaje					
Unidades de Aprendizaje	Resultados de aprendizaje (El estudiante es capaz de...)	Estrategias de enseñanza y aprendizaje	Estrategias de evaluación de los aprendizajes y ponderación	Horas presenciales	Horas de trabajo autónomo
<b>Unidad 1: Ambiente de desarrollo y manejo de datos</b> <i>(10 clases: 18/03-19/04)</i> <ul style="list-style-type: none"> <li>● Ambiente de desarrollo interactivo basado en IPython y Jupyter con librerías numpy, pandas, scipy y matplotlib</li> <li>● Repaso de python (Slicing, Listas, diccionarios, generadores, sets, clases)</li> <li>● IPython magics</li> <li>● Debugging</li> <li>● Arreglos y operaciones vectoriales con Numpy</li> <li>● Lectura y manipulación de datos usando Numpy y Pandas</li> <li>● Serialización</li> <li>● Visualización de datos usando matplotlib</li> <li>● Jupyter widgets</li> <li>● Manteniendo un repositorio en github</li> </ul>	<ul style="list-style-type: none"> <li>● Montar un ambiente de desarrollo basado en Python 3 enfocado en ciencia de datos</li> <li>● Manejar con soltura el ambiente de programación interactiva Jupyter</li> <li>● Leer datos guardados en distintos formatos y ser capaz de manipularlos y visualizarlos</li> </ul>	<b>Clases expositivas con actividades prácticas formativas:</b>  Se usa un Jupyter notebook donde se presentan los contenidos de la unidad. Esto se combina con actividades prácticas donde se resuelven problemas de forma interactiva y guiada	<b>Evaluación parcial: Tarea 1 (33%)</b>  Los estudiantes desarrollan y entregan un trabajo en formato Jupyter notebook en donde contestan preguntas teóricas y resuelven problemas prácticos sobre fundamentos de Python, manipulación de datos, visualización de datos y otros contenidos vistos en la unidad 1	15 horas	15 horas

Unidades de Aprendizaje	Resultados de aprendizaje (El estudiante es capaz de...)	Estrategias de enseñanza y aprendizaje	Estrategias de evaluación de los aprendizajes y ponderación	Horas presenciales	Horas de trabajo autónomo
<b>Unidad 2: Computación científica con Python</b> <i>(12 clases: 22/04 – 31/05)</i> <ul style="list-style-type: none"> <li>Algebra lineal con Numpy</li> <li>Regresión lineal y polinomial</li> <li>Gradiente descendente</li> <li>Regresión lineal online</li> <li>Neurona artificial y regresión logística</li> <li>Optimización numérica con Scipy</li> <li>Cálculo de valores y vectores propios con Numpy/Scipy</li> <li>Análisis de componentes principales (PCA)</li> <li>Variables aleatorias, distribuciones e inferencia estadística básica</li> <li>Ajuste de modelo estadístico con Scipy</li> </ul>	<ul style="list-style-type: none"> <li>Factorizar matrices y resolver sistemas de ecuaciones lineales (Regresión lineal)</li> <li>Ajustar un modelo de clasificación usando gradiente descendente (regresión logística)</li> <li>Resolver el problema de valores propios y ajustar un modelo lineal de variable latente (PCA)</li> <li>Generar números aleatorios a partir de distribuciones y ajustar modelos estadísticos</li> </ul>	<b>Clases expositivas con actividades prácticas formativas:</b>  Se usa un Jupyter notebook donde se presentan los contenidos de la unidad. Esto se combina con actividades prácticas donde se resuelven problemas de forma interactiva y guiada	<b>Evaluación parcial: Tarea 2 (34%)</b>  Los estudiantes desarrollan y entregan un trabajo en formato Jupyter notebook en donde contestan preguntas teóricas y resuelven problemas prácticos sobre sistemas de ecuaciones lineales, optimización, ajuste estadístico de modelos y otros contenidos vistos en la unidad 2	18 horas	18 horas

Unidades de Aprendizaje	Resultados de aprendizaje (El estudiante es capaz de...)	Estrategias de enseñanza y aprendizaje	Estrategias de evaluación de los aprendizajes y ponderación	Horas presenciales	Horas de trabajo autónomo
<b>Unidad 3: Procesamiento de señales con Python y aceleración de cómputos</b> (12 clases: 03/06 – 12/07) <ul style="list-style-type: none"> <li>• Introducción al análisis de señales y series de tiempo con Scipy</li> <li>• Transformada de Fourier y espectrograma con Scipy</li> <li>• Filtrado de señales (audio e imágenes) con Scipy</li> <li>• Filtros adaptivos para limpieza y predicción de señales</li> <li>• Profiling de código Python</li> <li>• Mapeo de memoria con Numpy</li> <li>• Computación paralela con multiprocessing</li> <li>• Aceleración de rutinas de cómputo intensivo con Cython</li> </ul>	<ul style="list-style-type: none"> <li>• Analizar señales y series de tiempo usando técnicas estadísticas y transformada de Fourier</li> <li>• Suavizar, mejorar, y ecualizar señales usando filtros</li> <li>• Predecir series de tiempo usando filtros</li> <li>• Detectar cuellos de botella en el desempeño de un programa</li> <li>• Acelerar código Python aprovechando arquitecturas multi-núcleo y/o usando C/C++</li> </ul>	<b>Clases expositivas con actividades prácticas formativas:</b>  Se usa un Jupyter notebook donde se presentan los contenidos de la unidad. Esto se combina con actividades prácticas donde se resuelven problemas de forma interactiva y guiada	<b>Evaluación parcial: Tarea 3 (33%)</b>  Los estudiantes desarrollan y entregan un trabajo en formato Jupyter notebook en donde contestan preguntas teóricas y resuelven problemas prácticos sobre análisis de señales, transformada de Fourier, filtros adaptivos, optimización de código para cómputo intensivo y otros contenidos vistos en la unidad 3	12 horas	12 horas

### Requisitos de aprobación

- Cada unidad será evaluada con una tarea a resolverse en grupos de máximo 2 estudiantes
- Los estudiantes tendrán una semana para completar y entregar la tarea
- La entrega de la tarea se hace mediante correo electrónico a [phuijse@inf.uach.cl](mailto:phuijse@inf.uach.cl)
- Luego de cumplida la fecha de entrega se descontará un punto por día de atraso
- El promedio de las evaluaciones parciales se calcula como:  $NF = 0.33 T1 + 0.34 T2 + 0.33 T3$
- El estudiante aprueba el curso si  $NF \geq 4.0$

### Recursos de aprendizaje

#### Bibliografía:

- Jake VanderPlas, "Python Data Science Handbook", O' Reilly, 2016
  - Disponible libremente en: <https://jakevdp.github.io/PythonDataScienceHandbook/>
- Cyrille Rossant, "IPython Cookbook", 2ed, Packt, 2018
  - Disponible libremente en: <https://ipython-books.github.io/>
- Robert Johansson, "Numerical Python", 2ed, Apress, 2018
  - Disponible en Springer: <https://link.springer.com/book/10.1007%2F978-1-4842-4246-9>

#### Otros:

- Medio de comunicación: <https://escueladeinformatica.slack.com>
- Software y librerías
  - Lenguaje: [Python 3.7](#)
  - Ambiente: [IPython](#), [Jupyter](#)
  - Cómputo científico: [Numpy](#), [Scipy](#), [Pandas](#)
  - Visualización: [Matplotlib](#), [bokeh](#)



