



Curso Introducción a la programación en Python

## Syllabus

### Equipo Docente

#### Valeria Herskovic

Profesora asociada del Departamento de Ciencias de la Computación de la Pontificia Universidad Católica de Chile. Su investigación está enfocada en sistemas de información, sistemas colaborativos, dispositivos móviles, interfaces humano-computador y redes sociales.

#### Cristian Ruz

Profesor Asistente Adjunto en el Departamento de Ciencias de la Computación de la Escuela de Ingeniería, Pontificia Universidad Católica de Chile. Posee un doctorado en informática de la Université de Nice-Sophia Antipolis luego de haber desarrollado su investigación en el centro INRIA de Francia. Es Ingeniero Civil de Computación y Magister en Ciencias de la Ingeniería de la Pontificia Universidad Católica de Chile. Su tema de investigación es el área de sistemas distribuidos y computación paralela en sistema de cómputo de alto rendimiento (HPC). Ha dictado numerosas veces cursos introductorios de programación, así como también cursos de técnicas avanzadas de programación, sistemas operativos, compiladores y redes de computadores.

#### Jorge Muñoz Gama

Ingeniero Informático y Master en Computación por la Universitat Politècnica de Catalunya, recibió el grado de Doctor en Computación con distinción Cum Laude y

mención internacional en la misma universidad. Nacido en Barcelona, actualmente se desempeña como profesor del Departamento de Ciencia de la Computación UC, realizando docencia e investigación en las áreas de Process Mining y Process Oriented Data Science. Autor del libro 'Conformance Checking and Diagnosis in Process Mining', en 2015 recibió el premio IEEE Best Dissertation on Process Mining.

## Descripción del curso

En este curso los estudiantes revisarán los conceptos fundamentales para el desarrollo de algoritmos y su programación en computadores, abordando conceptos que permitirán el desarrollo de programas más complejos utilizando el lenguaje de programación Python. Para esto, el curso cuenta con variadas actividades formativas (clases, videos y talleres) y evaluaciones que buscan medir tanto el manejo teórico como práctico de los contenidos (cuestionarios, tareas y proyecto final).

Al finalizar el curso, los estudiantes serán capaces de desarrollar sus propios programas informáticos, utilizando estructuras de datos y tipos de datos propios, para implementar algoritmos más complejos y representar problemas interesantes a través de la programación.

## Resultados de Aprendizaje

- Aplicar el concepto y operaciones de tipos de datos secuenciales y no secuenciales para construir soluciones computacionales.
- Crear tipos de datos propios mediante los conceptos de clases, objetos, atributos y métodos, para desarrollar programas sencillos.
- Construir modelos mediante interacción entre objetos y herencia, para desarrollar programas de mediana complejidad.
- Utilizar el concepto de recursión para resolver problemas computacionales.

El curso desarrolla sus contenidos y actividades a través de una modalidad online, que combina diferentes recursos de aprendizaje según cada tipo de contenido, por ejemplo: videos, imágenes, esquemas, textos, audios, preguntas formativas, foros y otros. Todos estos recursos se encuentran integrados en una plataforma virtual a la que accede el alumno y desde la cual también obtendrá información de apoyo (materiales anexos, bibliografía, evaluaciones, mensajería, información sobre calificaciones y otros), además, desde la misma plataforma podrá acceder a las clases sincrónicas (online streaming) contempladas en el programa del curso. Específicamente, se trabaja con:

- Sesiones expositivas
- Discusión de casos.
- Lecturas
- Análisis de casos

\* El curso contendrá además actividades formativas (sin calificación), las que tienen como propósito la preparación para las evaluaciones calificadas.

## Estructura del curso

El curso está estructurado de la siguiente forma:

### **Módulo 1: Estructuras de datos secuenciales**

- Listas
- Tuplas
- Stacks
- Colas

### **Módulo 2: Estructuras de datos no-secuenciales**

- Sets

- Dictionarios.

### **Módulo 3: Programación Orientada a Objetos**

- Construyendo nuestros propios tipos de datos.
- Clases, objetos, atributos y métodos.
- Interacción entre Objetos
- Herencia
- Overriding de métodos

### **Módulo 4: Recursión**

- Funciones recursivas
- Ejemplos de problemas recursivos: pintar mapas, fractales, etc.

El requisito académico se cumple realizando todos los test (evaluaciones) del curso. El alumno solamente podrá aprobar el curso si aprueba todos los test. Los test se aprueban si se obtiene el 80% de las respuestas correctas. El promedio final del curso será el promedio de la nota final de cada módulo.

Actividad	Evaluación
Cuestionarios sumativos	36% nota final
Tareas	45% nota final
Trabajo Final	19% nota final

## Plataforma e Información General

**Sigla:** INF3103

**Créditos:**5

**Horas:** 90 horas de dedicación total (24 directas y 66 indirectas)

**Requisitos:** No tiene

**Restricciones:** (Programa MDS) o (Programa MANE)

**Carácter:** Mínimo

**TIPO:** Cátedra

**Calificación:** Estándar

**Palabras clave:** Programación, análisis de datos, computación

**Nivel Formativo:** Magíster

## Bibliografía

### Bibliografía mínima

- Python software foundation, Python v3 Documentation, <http://docs.python.org/3/>.

### Bibliografía complementaria

- V. Ceder. The quick python book. Manning Publications Co., 2010.
- B. Downey. Think Python: How to think like a computer scientist. Green Tea Press, 2013 Version web: <http://www.greenteapress.com/thinkpython/>
- J. M. Zelle. Python programming: An introduction to computer science. Franklin, Beedle & Associates, Inc., 2nd edition, 2010.

### Bibliografía sugerida

- Py-Libre, Apunte interactivo para el curso Introducción a la Programación, <http://runest.ing.puc.cl>