



IIC2115 - PROGRAMACIÓN COMO HERRAMIENTA PARA LA INGENIERÍA

– Programa de curso –

Profesor	: Francesca Lucchini (flucchini@uc.cl)
Ayudante Jefe	: Nicolás Alvarado (mamagna@uc.cl)
Sitio Web	: Syllabus en GitHub (github.com/IIC2115/Syllabus) y Canvas (solo avisos)
Clases	: lunes, módulos 4 y 5 (14:00 - 16:50) - Sala BC24
Ayudantía	: jueves, módulo 5 (15:30 - 16:50) - Sala BC24
Horario de atención	: agendar cita por email

Presentación del curso

La programación de herramientas de software especializadas en las distintas áreas de la ingeniería ya no es una novedad y se ha transformado en una constante. Si bien en un comienzo el objetivo principal de este desarrollo fue la optimización y la resolución de problemas numéricos complejos, la reciente explosión en el uso de datos masivos de diversos tipos en todas las áreas de la ingeniería hacen que el almacenamiento, procesamiento y análisis de estos sean las principales tareas para las cuales las herramientas de programación son utilizadas. Así, el conocimiento y las habilidades relacionadas con la programación y su uso para el manejo de datos, se han transformado no solo en una ventaja, sino en una necesidad para los profesionales de la ingeniería.

El propósito de este curso es que los estudiantes se familiaricen con la programación como una herramienta fundamental y poderosa, ya no sólo para solucionar de manera más eficiente y efectiva problemas clásicos en ingeniería, sino que para desarrollar soluciones innovadoras a nuevos problemas basados en datos. Para

alcanzar este objetivo, el curso se basa en el lenguaje Python y sus principales librerías aplicadas para el manejo de datos provenientes de diversas fuentes y con distintos formatos.

Objetivos de aprendizaje

A nivel general, al finalizar el curso, los estudiantes serán capaces de:

- Evaluar y utilizar de manera efectiva distintas técnicas y herramientas de programación para resolver problemas basados en datos, asociados a sus áreas de especialización.
- Proponer y desarrollar soluciones novedosas utilizando la programación, tanto para problemas tradicionales, como para nuevos problemas en ingeniería basados en datos.

A nivel particular, al finalizar el curso, los estudiantes serán capaces de:

- Utilizar entornos de programación modernos para Python y sus librerías.
- Modelar y solucionar problemas utilizando técnicas de programación orientada a objetos y estructuras de datos.
- Procesar, analizar y visualizar datos tabulares utilizando herramientas de programación.
- Procesar, analizar y visualizar datos geoespaciales y estructurados utilizando herramientas de programación.
- Manejar y modelar grandes volúmenes de datos, y realizar consultas sobre estos, utilizando herramientas de programación.

Contenido

A continuación se presenta un desglose detallado de los contenidos del curso:

Capítulo 0: introducción

- Fundamentos de Python
- Entornos de desarrollo modernos para Python: Jupyter Notebook en Google Colab
- Sistema de control de versiones: git y GitHub

Capítulo 1: fundamentos

- Elementos fundamentales de programación orientada a objetos
- Estructuras de datos
- Técnicas y algoritmos

Capítulo 2: análisis de datos tabulares

- Procesamiento, análisis y limpieza de datos
- Visualización de datos
- Aprendizaje de máquina

Capítulo 3: análisis de datos geoespaciales y estructurados

- Representación y carga de datos geoespaciales
- Operaciones y visualización de datos geoespaciales
- Representación y algoritmos en redes
- Carga y uso de redes de transporte

Capítulo 4: manejo de grandes volúmenes de datos

- Modelo relacional de datos
- Web scraping
- Consultas sobre datos

Metodología

El curso sigue una metodología de clase invertida (*flipped classroom*), donde los estudiantes deben estudiar, es decir, **leer y practicar**, los contenidos de manera previa a la clase, para luego aplicarlos en ella ejercicios prácticos y laboratorios. En todas estas instancias, el cuerpo docente del curso estará en la sala para guiar activamente a los estudiantes y contestar sus dudas.

La dinámica del curso se desarrolla en el marco de cuatro capítulos de contenidos, donde cada uno tiene una duración de cuatro semanas. Cada capítulo considera los siguientes tipos de sesiones:

- Cátedras: consisten en un breve resumen y sesión de preguntas y respuestas dirigida por la profesora y centrada en los contenidos planificados para la semana (que deben ser previamente estudiados por los estudiantes). Posterior a esto, los estudiantes deberán contestar un breve control de los contenidos,

para luego resolver problemas de programación centrados en estos tópicos. Finalmente, la profesora realizará una consolidación del contenido, revisando además el ejercicio previamente planteado. Estas sesiones se realizan durante los dos primeros lunes de cada capítulo.

- Laboratorios: consiste en la resolución **individual** por parte de los estudiantes de problemas de programación centrados en todos los tópicos del capítulo. Estas sesiones se inician el tercer lunes de cada capítulo y terminan el cuarto, es decir, el laboratorio tendrá un plazo de desarrollo de una semana. Para cada laboratorio se considera además un espacio de lectura y explicación del enunciado.
- Ayudantías: consisten en el desarrollo y resolución por parte de los ayudantes de ejercicios de programación, además de la resolución de dudas de los contenidos del curso.

Dada la naturaleza práctica del curso, es fundamental que los estudiantes asistan a las sesiones, de manera que puedan recibir ayuda para solucionar problemas y validar el avance realizado. **Se espera además que los estudiantes utilicen otras fuentes para complementar y profundizar los contenidos, tales como los libros que se encuentran indicados en la bibliografía o sitios de internet.**

Evaluaciones

Las evaluaciones se dividen en dos tipos, cada una con su correspondiente nota final promedio:

- Laboratorios (70%): cada capítulo considera el desarrollo individual de un laboratorio de programación, que tendrá una duración de una semana. La calificación de esta se basará en su completitud y la aplicación de los contenidos involucrados. Para la entrega se utilizará la plataforma GitHub.

Las entregas tendrán como hora límite las 23:59 del día indicado en el enunciado. En relación a los atrasos, se aplicará un descuento lineal hasta llegar a un máximo de 6 horas, en base a la siguiente fórmula:

$$\text{Descuento} = 0.5 + 0.2 \times \frac{t}{60}$$

Donde t es la cantidad de minutos de atraso del commit a corregir. Laboratorios con atrasos superiores a 6 horas serán calificadas con nota 1.0. Tienen hasta 12 horas después de la hora oficial de entrega de los laboratorios para llenar el formulario disponible en el Syllabus, indicando los datos del *commit* atrasado que desean que sea revisado. En caso que no se llene el formulario dentro del plazo se revisará el último *commit* (de la carpeta LX correspondiente) realizado dentro del plazo de entrega.

La nota final de los laboratorios (**L**) está dada por el promedio de estos.

- Controles (15%): durante las sesiones de cátedra y antes de empezar los ejercicios, los estudiantes deberán responder un breve control (6 preguntas) sobre la materia. El control se realizará mediante Google Forms, en formato de selección múltiple. Se dará 20 minutos para contestar.

La nota final de los controles (**C**) está dada por el promedio de estos.

- Participación (15%): durante las sesiones de cátedra, los estudiantes deberán entregar el avance realizado, con el fin de validar su participación en la sesión. Los detalles del avance para cada instancia serán especificados en cátedra y corresponderán a elementos básicos de la materia. Para la entrega se utilizará la plataforma GitHub y la hora límite será las 16:50 del día en que se realiza la cátedra. Avances entregados pasada la hora límite no serán considerados. La nota final de participación (**P**) está dada por el promedio de los avances, donde la calificación es binaria (entregado=7,0, no entregado = 1,0).

En caso de ausencias justificadas a las sesiones, se utilizarán las siguientes reglas para adecuar las notas:

- Laboratorios: “cupón” de extensión por 4 días, válido 1 vez por semestre, contra certificado emitido por la Dirección de Pregrado (solo Covid u otro evento de fuerza mayor).
- Participación: no se considerará la sesión para calcular la nota, contra certificado emitido por la Dirección de Pregrado.

En base a todo lo descrito anteriormente, la nota final del curso (**N**) se calcula de la siguiente manera:

$$\mathbf{N} = 0.7 \cdot \mathbf{L} + 0.15 \cdot \mathbf{C} + 0.15 \cdot \mathbf{P}$$

Cronograma de actividades

Capítulo	Fecha	Actividades
Capítulo 0	08/08	Introducción al curso
Capítulo 1	18/08	Cátedra Capítulo 1a
	22/08	Cátedra Capítulo 1b
	25/08	Ayudantía Capítulo 1b
	29/08	Inicio L1
	01/09	Ayudantía L1
	08/09	Entrega L1
Capítulo 2	12/09	Cátedra Capítulo 2a
	15/09	Ayudantía Capítulo 2a
	22/09	Cátedra Capítulo 2b
	26/09	Inicio L2
	29/09	Ayudantía L2
	06/10	Entrega L2

Capítulo 3	13/10	Cátedra Capítulo 3a
	17/10	Cátedra Capítulo 3b
	20/10	Ayudantía Capítulo 3b
	24/10	Inicio L3
	27/10	Ayudantía L3
	03/11	Entrega L3
Capítulo 4	07/11	Cátedra Capítulo 4a
	10/11	Ayudantía Capítulo 4a
	14/11	Cátedra Capítulo 4b
	17/11	Ayudantía Capítulo 4b
	21/11	Inicio L4
	24/11	Ayudantía L4
	01/12	Entrega L4

Retroalimentación y correcciones

Dada la naturaleza práctica de la metodología del curso, es fundamental la entrega de retroalimentación rápida en relación a lo realizado en los laboratorios y cátedras, con el fin de contribuir de manera temprana al correcto aprendizaje de los contenidos. Tomando esto en consideración, Los laboratorios tendrán retroalimentación, que se entregará junto con la nota, en la forma de una descripción detallada, donde se indicarán todos los elementos que fueron relevantes para la corrección, además de la asignación de puntaje por cada uno de estos. En caso de no quedar conforme con la nota obtenida y/o la retroalimentación, se debe realizar una solicitud de corrección **solo** a través del formulario indicado en el sitio del curso.

En relación a las cátedras, los miembros del cuerpo docente recorrerán la sala de manera continua, con el fin contestar dudas o entregar retroalimentación a los estudiantes sobre sus avances en los ejercicios planteados para la sesión.

Medios oficiales del curso

El sitio oficial del curso será el *Syllabus*, donde se publicarán los apuntes del curso y las *slides* usadas en cátedra. También serán publicados aquí los ejercicios y pautas de las evaluaciones.

Las dudas sobre los contenidos del curso o aspectos administrativos de interés general deben realizarse **exclusivamente** a través de *issues* en el *Syllabus*. Cualquier comunicación personal relativa a aspectos administrativos del curso (**no dudas de materia**) debe dirigirse al mail iic2115@ing.puc.cl. Situaciones urgentes, de mayor importancia o personales que requieran de alta privacidad deben realizarse directamente al mail de la profesor.

Salvo correos urgentes al profesor, solo se responderán *issues* y correos de lunes a viernes de 8:00 a 18:30. Este horario será válido también en el sentido inverso, o sea, uds. no recibirán correos o avisos por parte del cuerpo docente del curso fuera de este horario, salvo que sean temas urgentes.

Política de Integridad Académica

Los estudiantes de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los estudiantes que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: todo trabajo presentado por un estudiante (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el estudiante (grupo), sin apoyo en material de terceros. Si un estudiante (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de

Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.

Bibliografía

- Apuntes del curso disponibles en el sitio.
- *Advanced Computer Programming in Python*; Pichara y Pieringer; 2017.
- *Introduction to Algorithms*; Cormen, Leiserson, Rivest y Stein; 2009 (3^a edición).
- *Database Management Systems*; Ramakrishnan y Gehrke; 2002.
- *Python Data Science Handbook*; VanderPlass; 2016.