



IIC2115 - PROGRAMACIÓN COMO HERRAMIENTA PARA LA INGENIERÍA

– Programa de curso –

Profesores : Hans Lobel (halobel@uc.cl) y Pablo Seisdedos (pcseisdedos@uc.cl)
Sitio Web : Syllabus en GitHub (github.com/IIC2115/Syllabus) y Canvas (solo avisos)
Clases : lunes, módulos 5 y 6 (14:50 - 17:20) - Sala A5
Ayudantía : jueves, módulo 5 (14:50 - 16:00) - Sala AE203

Presentación del curso

La programación de herramientas de software especializadas para las distintas áreas de la ingeniería ya no es una novedad y se ha transformado en una constante, tanto en la industria, como en la academia. 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 solo para solucionar de manera más eficiente y efectiva problemas clásicos en ingeniería, sino que para desarrollar soluciones innovadoras a nuevos problemas, utilizando para ello grandes volúmenes de 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

Capítulo 2: Manejo y Análisis de Datos Tabulares

- Procesamiento, análisis y limpieza de datos con Pandas
- Visualización de datos con Matplotlib

Capítulo 3: Modelos Predictivos

- Fundamentos de Machine Learning
- Algoritmos y modelos
- Ejemplos de aplicación
- Uso de scikit-learn

Capítulo 4: Análisis de Datos Geoespaciales

- Sistemas de información geográfica (GIS)
- Representación y carga de datos geoespaciales con GeoPandas
- Operaciones y visualización de datos geoespaciales

Capítulo 5: Análisis de Datos Estructurados

- Representación y algoritmos en redes
- Carga y uso de redes con NetworkX
- Manejo de redes viales con OSMNX

Capítulo 6: Manejo de Datos

- Modelo relacional de datos
- Creación de bases de datos desde múltiples fuentes

Capítulo 7: Consultas sobre Grandes Volúmenes de Datos

- Consultas en SQL
- Consultas anidadas
- Parametrización de consultas

Metodología

El curso sigue una metodología de taller, donde los estudiantes deben estudiar, es decir, **leer y practicar**, los contenidos de manera previa y durante la clase, aplicándolos luego a través de ejercicios de programación formativos y sumativos. 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 siete capítulos de contenidos, donde cada uno tiene una duración de 2 semanas. Cada capítulo considera los siguientes tipos de sesiones:

- **Cátedra:** las sesiones de cátedra se realizan el primer lunes de cada capítulo y separan en dos partes. La primera consiste en un resumen de la materia del capítulo, incluyendo ejemplos prácticos. Posterior a esto, durante la segunda parte los estudiantes deberán desarrollar un ejercicio de programación formativo, basado en los tópicos y ejemplos discutidos al inicio de la sesión. Al finalizar la sesión, el cuerpo docente abrirá el espacio para hacer la entrega del avance realizado y contestar un *ticket de salida*, donde deberán reportar su avance e impresiones con respecto al avance logrado durante la sesión.
- **Laboratorio:** las sesiones de laboratorio se inician el segundo lunes de cada capítulo. En estas los estudiantes deberán desarrollar de manera individual un laboratorio de programación evaluado, enfocado en los tópicos del capítulo. Cada laboratorio consistirá en dos partes: la primera planificada para poder ser resuelta durante la sesión, con hora límite de entrega fijada para el final del segundo módulo, y la segunda planificada para ser resuelta durante el resto de la semana, enfocada en ejercicios que requieren un mayor tiempo de resolución, con fecha límite de entrega fijada para el domingo de esa semana.
- **Ayudantía:** consisten en la resolución por parte del cuerpo docente de ejercicios de programación o dudas, dependiendo de la etapa del capítulo en que se encuentre.

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.

Uso de recursos educacionales externos al curso

Se espera que los estudiantes utilicen otras fuentes para complementar y profundizar los contenidos del curso, además del material entregado por el cuerpo docente. Ejemplos de esto son los libros que se encuentran

indicados en la bibliografía o sitios de internet. Todas estas fuentes deben referenciarse de manera clara en el código entregado.

Para el caso específico de los agentes inteligentes como ChatGPT, Gemini o Copilot, se recomienda su uso en todas las instancias del curso, siempre y cuando se indique explícitamente y al inicio de los archivos entregados, que sí se utilizó la ayuda de dicho asistentes. Adicionalmente, cada vez que se utilicen, se deberá indicar de manera clara su modo de uso (al menos el *prompt* y respuesta) y cómo las las respuestas entregadas fueron modificadas sustancialmente. Se entiende como una modificación sustancial aquella que involucra el análisis crítico de la respuesta del asistente, y todas las modificaciones y mejoras que de este análisis se desprendan.

Evaluaciones

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

- Laboratorios (80%): cada capítulo considera el desarrollo individual de un laboratorio de programación dividido en dos partes con entregas diferidas, ambas con el mismo valor en la nota. La calificación de este se basará en su completitud y la aplicación de los contenidos involucrados. La nota final de los laboratorios (**L**) está dada por el promedio de los **6 mejores**.
- Participación (20%): al finalizar las sesiones de cátedra, los estudiantes deberán entregar el avance realizado a través de un ticket de salida y en sus repositorios privados, con el fin de validar su participación en la sesión. La nota final de participación (**P**) está dada por el promedio de los avances, donde la calificación es binaria (entregado = 7,0, no entregado o sin avance = 1,0).

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

- Laboratorios: dado que se elimina la peor nota obtenida en los laboratorios, no es obligatorio justificar la primera ausencia a estos. Sin embargo, en caso de ausencias dadas por motivos de salud, se recomienda igualmente gestionar un certificado a través de la Dirección de Pregrado.

En caso de más de una ausencia debidamente justificada ante la DiPre, se debe coordinar con el profesor del curso la solución más adecuada al caso.

- Participación: no se considerará la sesión para calcular la nota, contra certificado emitido por la Dirección de Pregrado.

Criterios de aprobación

Para aprobar el curso, las notas **L** y **P** deben ser mayores o iguales a 3,95. En caso de cumplir este criterio, la nota final del curso (**F**) se calcula de la siguiente manera:

$$\mathbf{F} = 0,8 \cdot \mathbf{L} + 0,2 \cdot \mathbf{P}$$

En caso de no satisfacer las restricciones, la nota final de reprobación ($\tilde{\mathbf{F}}$) será:

$$\tilde{\mathbf{F}} = \min(3,9, \mathbf{F})$$

Cronograma de actividades

#	Laboratorio	Lunes	Martes	Miercoles	Jueves	Viernes	Sadabo	Domingo	Mes
1	Introduccion	5	6	7	8		10	11	Agosto
2	L01	12	13	14	15	16	17	18	
3		19	20	21	22	23	24	25	
4	L02	26	27	28	29	30	31	1	Septiembre
5		2	3	4	5	6	7	8	
6	L03	9	10	11	12	13	14	15	
7	Receso	16	17	18	19	20	21	22	Octubre
8	L03	23	24	25	26	27	28	29	
9		30	1	2	3	4	5	6	
10	L04	7	8	9	10	11	12	13	Noviembre
11		14	15	16	17	18	19	20	
12	L05	21	22	23	24	25	26	27	
13		28	29	30	31	1	2	3	Diciembre
14	L06	4	5	6	7	8	9	10	
15		11	12	13	14	15	16	17	
16	L07	18	19	20	21	22	23	24	
17		25	26	27	28	29	30	1	
18	Cierre	2	3	4	5	6	7	8	
19		9	10	11	12	13	14	15	
20		16	17	18	19	20	21	22	

Retroalimentación y correcciones

Dada la naturaleza práctica del curso, es fundamental la entrega de retroalimentación rápida en relación a lo realizado en cátedra y laboratorios, con el fin de contribuir de manera temprana al correcto aprendizaje de los contenidos. Tomando esto en consideración, es fundamental que los estudiantes asistan a las cátedras y ayudantías.

Además de lo anterior, para cada evaluación se entregará, junto con la nota, una retroalimentación, 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 disponible 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, *slides* usadas en cátedra, ejercicios de ayudantías y rúbricas de corrección. Además, se encuentra disponible el material completa del cursos, incluyendo ejercicios, de los últimos 7 años.

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 del profesor.

Salvo correos urgentes al profesor, solo se responderán *issues* y correos de lunes a viernes de 8:30 a 18:30. Este horario será valido 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.

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.

Política de Integridad Académica

Los/as estudiantes de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los/as estudiantes que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada estudiante conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería.

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un/a estudiante para los efectos de la evaluación de un curso debe ser hecho **individualmente** por el/la estudiante, **sin apoyo en material de terceros**. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

En particular, si un/a estudiante copia un trabajo, o si a un/a estudiante se le prueba que compró o intentó comprar un trabajo, **obtendrá nota final 1.1 en el curso** y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral.

Por “copia” se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. En caso que corresponda a “copia” a otros estudiantes, la sanción anterior se aplicará a todos los involucrados. En todos los casos, se informará a la Dirección de Pregrado de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

También se entiende por copia extraer contenido sin modificarlo sustancialmente desde fuentes digitales como Wikipedia o mediante el uso de asistentes inteligentes como ChatGPT, Gemini o Copilot. Se entiende que una modificación sustancial involucra el análisis crítico de la información extraída y en consecuencia todas las modificaciones y mejoras que de este análisis se desprendan. Cualquiera sea el caso, el uso de fuentes bibliográficas, digitales o asistentes debe declararse de forma explícita, y debe indicarse cómo el/la estudiante mejoró la información extraída para cumplir con los objetivos de la actividad evaluativa.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, **siempre y cuando se incluya la referencia correspondiente**.

Lo anterior se entiende como complemento al Reglamento del Estudiante de la Pontificia Universidad Católica de Chile (<https://registrosacademicos.uc.cl/reglamentos/estudiantiles/>). Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.

Compromiso del Código de Honor

Este curso suscribe el Código de Honor establecido por la Universidad, el que es vinculante. Todo trabajo evaluado en este curso debe ser propio. En caso que exista colaboración permitida con otros/as estudiantes, el trabajo deberá referenciar y atribuir correctamente dicha contribución a quien corresponda. Como estudiante es un deber conocer el Código de Honor (<https://www.uc.cl/codigo-de-honor/>).