



IIC2115 - PROGRAMACIÓN COMO HERRAMIENTA PARA LA INGENIERÍA

– Programa –

Profesores	: Hans Lobel (halobel@uc.cl)
Ayudante Jefe	: Pablo Lourido (plourido@uc.cl)
Sitio Web	: Syllabus en GitHub (github.com/IIC2115/Syllabus) y Canvas (solo avisos y notas)
Clases	: lunes, módulos 5 y 6 (14:50 - 17:20) - Sala B12
Ayudantía	: jueves, módulo 5 (14:50 - 16:00) - Sala B12

Presentación del curso

El desarrollo de soluciones 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 y en todas las áreas de la ingeniería ha desplazado el foco hacia el almacenamiento, procesamiento y análisis de estos datos como tareas centrales. A esto se suma el crecimiento exponencial en el uso de herramientas de inteligencia artificial, que no solo han transformado la forma en que se resuelven problemas complejos, sino también cómo se programa: hoy es posible automatizar partes del proceso de desarrollo de software, generar código a partir de descripciones en lenguaje natural y construir modelos capaces de aprender y adaptarse a nuevas situaciones. En este contexto, el conocimiento y las habilidades relacionadas con la programación, el manejo de datos y la aplicación de técnicas de IA se han convertido 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 también para desarrollar soluciones innovadoras a nuevos desafíos, aprovechando grandes volúmenes de datos y, cada vez más, capacidades ofrecidas por la inteligencia artificial. Para alcanzar este objetivo, el curso se basa en el lenguaje Python y sus principales librerías aplicadas al manejo y análisis de datos provenientes de diversas fuentes y con distintos formatos, incorporando además nociones básicas sobre cómo los modelos de IA pueden complementar y potenciar el trabajo del ingeniero programador.

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, incluyendo enfoques asistidos por inteligencia artificial, para resolver problemas basados en datos, asociados a sus áreas de especialización.
- Proponer y desarrollar soluciones novedosas utilizando la programación, integrando técnicas tradicionales y herramientas de IA, tanto para problemas clásicos como para nuevos desafíos en ingeniería basados en datos.

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

- Utilizar entornos de programación modernos para Python y sus librerías, incluyendo asistentes de código y herramientas de IA que apoyan el desarrollo de software.
- 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, con apoyo de bibliotecas que integran funciones avanzadas de análisis automatizado.
- Procesar, analizar y visualizar datos geoespaciales y estructurados utilizando herramientas de programación.
- Manejar y modelar grandes volúmenes de datos, realizar consultas sobre estos, utilizando herramientas de programación que facilitan el análisis a escala.

Contenido

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

Capítulo 0: Introducción

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

Capítulo 1: Fundamentos

- Elementos fundamentales de programación orientada a objetos
- Estructuras de datos fundamentales

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 y Manejo de Redes

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

Capítulo 6: Extracción y Visualización de Datos

- Webscrapping y uso de APIs
- Visualización de información en dashboards
- Implementación de página web local

Capítulo 7: Manejo de Datos Relacionales y Consultas

- Modelo relacional de datos
- Creación de bases de datos desde múltiples fuentes
- Consultas en SQL: anidación y 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 (con excepción del Capítulo 3). Cada capítulo considera los siguientes tipos de sesiones:

- Cátedra: las sesiones de cátedra se realizan el primer lunes de cada capítulo, incluyendo el Capítulo 0, y separan en dos partes. La primera consiste en un resumen de la materia del capítulo, seguido de un ejemplo representativo del tipo de ejercicios a resolver en el capítulo. Posterior a esto, durante la segunda parte los estudiantes deberán desarrollar por cuenta propia 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 participación e impresiones con respecto al avance logrado durante la sesión.
- Laboratorio: las sesiones de laboratorio se inician el segundo lunes de cada capítulo, a partir del Capítulo 1. 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.

Evaluaciones

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

- Participación (20%): esta nota se calcula en base a los reportes de participación en cada sesión de cátedra. La calificación de cada uno es binaria: (ticket entregado **y** con avance en repositorio) \rightarrow 7, (ticket no entregado **o** sin avance sustantivo en repositorio) \rightarrow 1. La nota final de participación (**P**) está dada por el promedio de las **8 mejores**.
- Laboratorios (80%): ambas partes de los laboratorios cuentan con la misma ponderación en la nota final de la evaluación. 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**.

Ausencias

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

- Sesiones de Cátedra: en caso de existir más de una ausencia justificada ante la Dipre, e informada al cuerpo docente, la nota de Participación se calculará como el promedio simple de la participación en las sesiones donde sí se asistió, sin borrar la peor nota. Ausencias no justificadas ante la DiPre, no califican para realizar adecuaciones.
- Sesiones de Laboratorio: la ausencia a una sesión de Laboratorio genera de manera inmediata una calificación de 1,0 en ambas partes de este. En caso que esto ocurra, y sea posible, se recomienda fuertemente gestionar un certificado a través de la Dirección de Pregrado e informarlo al cuerpo docente del curso. En caso de más de una ausencia debidamente justificada ante la DiPre, se debe coordinar la situación con el profesor del curso.

Criterios de aprobación

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

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

En caso de no satisfacer las restricciones, la nota final de reprobación (**F̃**) será:

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

Cronograma de actividades

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

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 de contestar dudas o entregar retroalimentación a los estudiantes sobre sus avances en los ejercicios planteados para la sesión.

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, Copilot u otros, se recomienda fuertemente su uso en todas las instancias del curso, siguiendo las reglas específicas que se indicarán en clases. A modo general, siempre será necesario indicar de manera clara su modo de uso y cómo las respuestas entregadas fueron validadas y/o modificadas sustancialmente. Se entiende que esto último involucra el análisis crítico de la respuesta del asistente, y todas las modificaciones y mejoras que de este análisis se desprendan.

Medios oficiales del curso

El sitio oficial del curso es el *Syllabus* (<https://www.github.com/IIC2115/Syllabus>), donde se publicarán los apuntes del curso, *diapositivas* usadas en las sesiones de cátedra, ejercicios de ayudantías y rúbricas de corrección. Además, se encuentra disponible el material completo del curso, incluyendo ejercicios, desde el año 2017.

Fuera de las sesiones del curso, las dudas sobre los contenidos o aspectos administrativos de interés general deben realizarse **exclusivamente** a través de *issues* en el *Syllabus*, y no serán respondidas si se realizan por otro medio. Cualquier comunicación personal relativa a aspectos administrativos del curso (**no dudas de materia**) debe dirigirse al mail iic2115.ing@uc.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 8:20 a 17:20. Este horario será válido también en el sentido inverso, o sea, los estudiantes 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 (3ª edición).
- *Database Management Systems*; Ramakrishnan y Gehrke; 2002.
- *Python Data Science Handbook*; VanderPlass; 2016.
- *Learn AI-Assisted Python Programming*; Porter y Zingaro; 2024.

Política de integridad académica del curso

- **Código de Honor:** es indispensable que cada alumno tenga un comportamiento ético intachable. Por ello, este curso adscribe el Código de Honor establecido por la Universidad (<http://www.uc.cl/codigodehonor>) el cual es vinculante. Si el equipo docente sorprende o descubre evidencia de que uno o más alumnos realizan actitudes deshonestas y/o faltas al Código de Honor, entonces el profesor podrá calificar a los involucrados con nota 1,1 en el curso o evaluación relacionada, además de ejecutar todas las medidas disciplinarias pertinentes. Algunos ejemplos de conductas deshonestas son: falsificación de documentos, suplantación de identidad, uso de equipos prohibidos en evaluaciones, plagio y/o copia de trabajos y en evaluaciones online, uso indebido de inteligencia artificial, escribir mensajes indebidos en la prueba al corrector como, por ejemplo, solicitar nota o una corrección bondadosa, etc.
- **Copia y plagio:** todo trabajo evaluado en este curso debe ser propio. Se entiende por copia o plagio el “apropiarse de las ideas o palabras de otros y utilizarlas como propias sin dar crédito al autor original”. En caso de que exista colaboración permitida con otros estudiantes, el trabajo deberá referenciar y atribuir correctamente dicha contribución a quien corresponda. El cuerpo docente del curso se reserva el derecho a revisar cualquier documento con software de detección de plagio. Se recomienda revisar el siguiente recurso en Siding ¿ Pregrado ¿ Reglamentos ¿ Manual sobre Plagio.

- **Uso de herramientas de inteligencia artificial:** en este curso se permite el uso de herramientas de IA como apoyo en el desarrollo de las actividades del curso, siempre que sea de manera transparente y responsable. Cada estudiante debe revisar, validar y/o adaptar todo contenido generado por IA, asegurándose de comprenderlo completamente y de que cumple con los objetivos del curso y de la evaluación correspondiente. No se permite entregar respuestas generadas automáticamente sin una revisión crítica. Además, debe indicarse explícitamente cuándo y cómo se utilizó IA, por ejemplo: “Se utilizó ChatGPT para revisar la redacción del texto / generar ejemplos / asistir en la programación en Python”. El uso indebido de IA (como, por ejemplo, copiar sin comprender, no declarar su uso, entregar trabajos con “alucinaciones” de IA, falsear datos o el nivel de comprensión) será considerado una conducta deshonesta y una falta al Código de Honor.