



## IIC2115 - PROGRAMACIÓN COMO HERRAMIENTA PARA LA INGENIERÍA

### – Programa de curso –

<b>Profesor</b>	: Hans Lobel ( <a href="mailto:halobel@ing.puc.cl">halobel@ing.puc.cl</a> )
<b>Ayudante Jefe</b>	: Nicolás Alvarado ( <a href="mailto:nfalvarado@mat.uc.cl">nfalvarado@mat.uc.cl</a> )
<b>Sitio Web</b>	: Syllabus en GitHub ( <a href="https://github.com/IIC2115/Syllabus">github.com/IIC2115/Syllabus</a> ) y Canvas (solo avisos)
<b>Clases</b>	: lunes, módulos 5 y 6 (14:50 - 17:20) - Sala BC24
<b>Ayudantía</b>	: jueves, módulo 5 (14:50 - 16:00) - Sala R6
<b>Horario de atención</b>	: jueves, módulo 6 (16:10 - 17:20)

### 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, 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 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
- Representación y algoritmos en redes
- Carga y uso de redes de transporte

### **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 grandes volúmenes de datos**

- Web scraping
- Modelo relacional de datos
- 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 a través de actividades de programación formativas y sumativas. 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 seis capítulos de contenidos, donde cada uno tiene una duración de 2 o 3 semanas. Cada capítulo considera los siguientes tipos de sesiones:

- Cátedra: este tipo de sesión se separa en dos partes. La primera consiste en un breve resumen, sesión de preguntas y respuestas dirigida por el cuerpo docente y centrada en los contenidos planificados para el capítulo (que deben ser previamente estudiados por los estudiantes), y el desarrollo de un breve ejercicio de ejemplo. Posterior a esto, durante la segunda parte los estudiantes deberán desarrollar una actividad formativa de programación, basada en los tópicos y ejemplos discutidos al inicio de la sesión. Al finalizar, el cuerpo docente realizará una consolidación del contenido y abrirá el espacio para contestar un *ticket de salida*, donde deberán reportar su avance e impresiones con respecto al avance logrado durante la sesión.
- Actividad práctica: cada capítulo considera una sesión para el desarrollo individual de una actividad práctica de programación. La sesión comienza con la presentación y aclaración de dudas sobre el enunciado de la actividad, para luego pasar el trabajo individual en esta durante el resto de la sesión. Cada actividad será planificada para poder ser resuelta por los estudiantes durante la sesión, con la hora límite de entrega fijado para el final del segundo módulo.
- Ayudantías: consisten en la resolución por parte del cuerpo docente de los ejercicios de programación desarrollados durante la última sesión de cátedra o de actividad práctica realizada, 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, además del material del curso, tales como 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.

**Con respecto al uso de asistentes inteligentes, como ChatGPT o Bard, estos deben tratarse como cualquier otro recurso externo, es decir, debe referenciarse de manera exhaustiva su uso (fecha de consulta, *prompt* utilizado, resultado obtenido, entre otros).**

## Evaluaciones

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

- Actividades prácticas (50%): en este tipo de evaluación, cada alumno deberá resolver de manera individual durante la sesión correspondiente (excepto A1), un problema que tiene directa relación con

los contenidos del capítulo. La calificación de la actividad práctica de esta se basará en la completitud de lo entregado y la aplicación de los contenidos involucrados. En caso de atraso en la entrega, se aplicará un descuento de 5 décimas por cada 10 minutos de atraso. La nota final de las actividades de prácticas (**A**) está dada por el promedio de las 5 mejores notas obtenidas.

- Tareas (50%): se realizarán 3 tareas individuales, todas de igual valor, sobre tópicos cubiertos en el material y actividades del curso, mezclando contenido de distintos capítulos. La calificación de las tareas se basará en su completitud y la aplicación de los contenidos involucrados. En caso de atraso en la entrega, se aplicará un descuento de 5 décimas por cada hora de atraso. La nota final de las tareas (**T**) está dada por el promedio de notas obtenidas en estas.

Dado que se elimina la peor nota obtenida en las actividades prácticas, no es necesario justificar las ausencias a estas. Por el contrario, en caso de ausencias extendidas (más de 5 días hábiles) y debidamente justificadas ante la DiPre, se debe coordinar con el cuerpo docente del curso las condiciones de recuperación de la evaluación correspondiente, sea esta tarea o actividad práctica.

## Exigencias de aprobación

Para aprobar el curso, las notas **A** y **T** 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:

$$\mathbf{F} = 0,5 \cdot \mathbf{A} + 0,5 \cdot \mathbf{T} + \mathbf{B},$$

donde **A** es un bono de 2 décimas que se puede obtener si se contestan todos los *tickets de salida* durante el semestre.

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

#	Capítulo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo	Mes
1	1	7	8	9	10   C0-C1	11	12	13	Agosto
2		14	15	16	17   AyC1	18	19	20	
3		21   A1	22	23	24   Entrega A1	25	26	27	
4	2	28   C2	29	30	31   AyC2	1	2	3	Septiembre
5		4   A2	5	6	7   AyA2	8	9	10	
6	3	11   C3	12	13	14   AyC3	15	16	17	
7		18	19	20   T1	21   AyC3	22	23	24	Octubre
8		25   A3	26	27	28   AyA3	29	30	1	
9	Receso	2	3	4	5	6	7	8	
10	4	9	10	11	12   AyT1	13	14	15   Entrega T1	Noviembre
11		16   C4	17	18	19   AyC4	20	21	22	
12		23   A4	24   T2	25	26   AyA4	27	28	29	
13	5	30   C5	31	1	2   AyC5	3	4	5	Diciembre
14		6   A5	7	8	9   AyA5	10	11	12	
15	6	13   C6	14	15	16   AyC6	17	18	19   Entrega T2	
16		20   C6 - T3	21	22	23   AyC6	24	25	26	Diciembre
17		27   A6	28	29	30   AyA6	1	2	3	
18	Cierre Notas	4	5	6	7	8	9	10   Entrega T3	
19		11	12	13	14	15	16	17	
20		18	19   Cierre notas	20	21	22	23	24	

## 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, es fundamental que los estudiantes asistan a las ayudantías, donde se resolverán los ejercicios formativos y las actividades prácticas.

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.

## 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 8:30 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.

## 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.

## 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 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/>).