



IIC2115 - PROGRAMACIÓN COMO HERRAMIENTA PARA LA INGENIERÍA

– Programa de curso –

Profesores	: Francisco Garrido (fgv@ing.puc.cl) y Hans Löbel (halobel@ing.puc.cl)
Ayudante Jefe	: Pablo Seisdedos (pcseisdedos@uc.cl)
Sitio Web	: Siding y Syllabus en GitHub (github.com/IIC2115/Syllabus)
Clases	: Lunes, módulos 4 y 5 (14:00 - 16:50)
Ayudantía	: Jueves, módulo 5 (15:30 - 16:50)

Presentación del curso

Durante los últimos años, el uso y desarrollo de software especializado en las distintas especialidades de la ingeniería se ha transformado en una constante, ya sea por lo complejo de las tareas a realizar, o por la gran cantidad de datos que es necesario analizar. Es por esto que el conocimiento y las habilidades relacionadas con la programación se han transformado no sólo en una ventaja, sino en una necesidad para los profesionales de la ingeniería.

El propósito de este curso es que el alumno se familiarice con la programación como una básica y poderosa herramienta, no sólo para solucionar de manera más eficiente y efectiva problemas clásicos en ingeniería, sino que además para desarrollar soluciones innovadoras a nuevos problemas. Para alcanzar este objetivo, el curso cubre una amplia variedad de tópicos, incluyendo lenguajes y herramientas que son fundamentales para enfrentar de manera satisfactoria problemas de ingeniería, tanto en el aspecto profesional, como en el académico.

Objetivos de aprendizaje

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

- Evaluar y utilizar de manera efectiva distintos lenguajes y herramientas de programación para resolver problemas asociados a sus áreas de especialización, en base a los requerimientos de estos.
- Proponer y desarrollar soluciones novedosas utilizando la programación, ya sea para problemas tradicionales o para nuevos problemas en ingeniería.

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

- Utilizar herramientas modernas para el desarrollo de software.
- Modelar problemas utilizando técnicas de programación orientada a objetos.
- Crear soluciones a problemas, utilizando estructuras y técnicas avanzadas de programación.
- Modelar datos y sus relaciones, y realizar consultas sobre estos, mediante distintos modelos y lenguajes.
- Analizar, visualizar y presentar datos utilizando herramientas de programación.
- Manejar, procesar y visualizar datos de sistemas de información geográfica (SIG) mediante lenguajes de programación.

Contenido

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

Capítulo 1: Elementos básicos

- Python, Jupyter Notebook y Google Colab
- Sistema de control de versiones: git y GitHub
- Programación orientada a objetos: clases, herencia, composición
- Estructuras de dato básicas: tuplas, listas, diccionarios

Capítulo 2: Estructuras de datos y algoritmos

- Stacks, colas, sets
- Árboles, listas ligadas, grafos

- Recursión, backtracking, dividir y conquistar
- Ordenamiento y búsqueda en arreglos
- Búsqueda en grafos

Capítulo 3: Uso de bases de datos y archivos

- Manejo de archivos
- Modelo relacional de datos
- Consultas sobre datos usando SQL
- Uso de SQL en Python

Capítulo 4: Análisis de datos en Python

- Librerías: pandas, scikit-learn, numpy, matplotlib
- Manipulación y limpieza de datos
- Visualización de datos
- Aprendizaje de máquina

Capítulo 5: Tópicos avanzados

- Manejo de GIS en Python
- NetworkX
- Web scrapping y expresiones regulares

Metodología

El curso sigue una metodología de clase invertida (*flipped classroom*), donde los alumnos deben estudiar y practicar los contenidos de manera previa a la clase, para luego aplicarlos en ella mediante actividades prácticas y laboratorios de programación. Tanto actividades como laboratorios son de desarrollo individual y son acumulativos en cuanto a contenidos a lo largo del semestre.

La dinámica del curso se desarrolla en el marco de cinco capítulos de contenidos, donde el primero tiene carácter formativo. Cada uno tiene una duración aproximada de tres semanas (con excepción del primero). Los capítulos considera 5 tipos de sesiones:

- Cátedra: consisten inicialmente en la consolidación de los contenidos cubiertos por el material del capítulo, mediante una exposición por parte de los profesores del curso. Posteriormente, se desarrollan ejercicios prácticos relacionados con los tópicos cubiertos en el capítulo.
- Lectura de enunciado: consisten en la lectura y resolución de dudas por parte de los profesores sobre enunciado del laboratorio.
- Actividades prácticas: consisten en la resolución individual por parte de los alumnos de un problema de programación centrado en los tópicos del capítulo.
- Trabajo en laboratorio: consisten en el desarrollo del laboratorio práctico individual del capítulo.
- Ayudantías: consisten en el desarrollo y resolución por parte de los ayudantes de un conjunto de ejercicios prácticos centrados en los tópicos del capítulo.

Las sesiones de cátedra, lectura de enunciado y ayudantía son realizadas a través de Zoom. Las sesiones de actividades prácticas y de trabajo en laboratorio consideran el trabajo en una sala virtual en la plataforma Discord, con la presencia de integrantes del cuerpo docente con el fin de resolver sus consultas.

Dada la naturaleza práctica del curso, es fundamental que los alumnos asistan a las sesiones, de manera que puedan recibir ayuda para solucionar problemas y certificar el avance realizado. **Se espera además que los alumnos 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. Con excepción del laboratorio del primer capítulo, que es de carácter formativo, la calificación de los laboratorios se basará en su completitud y la aplicación de los contenidos involucrados. Para la entrega se utilizará la plataforma GitHub y la fecha límite será las 23:59 del día indicado en el enunciado.

En relación a las entregas atrasadas, 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. Tareas con atrasos superiores a 6 horas serán calificadas con nota 1.0.

La nota final de los laboratorios (**L**) está dada por el promedio de los 4 últimos laboratorios.

- Actividades prácticas (30%): en este tipo de evaluación, cada alumno deberán resolver durante la sesión, un problema que tiene directa relación con los contenidos del capítulo. Con excepción de la actividad práctica del primer capítulo, que es de carácter formativo, la calificación de las actividades se basará en su completitud y la aplicación de los contenidos involucrados. La nota final de las actividades de prácticas (**P**) está dada por el promedio de las 4 últimas actividades.

Exigencias de aprobación

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

$$\mathbf{F} = 0.7 \cdot \mathbf{L} + 0.3 \cdot \mathbf{P}$$

En caso contrario, la nota final de reprobación ($\tilde{\mathbf{F}}$) será:

$$\tilde{\mathbf{F}} = \min(3.9, \mathbf{F})$$

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, con el fin de contribuir de manera temprana al correcto aprendizaje de los contenidos. Tomando esto en consideración, cada uno de los laboratorios tendrá retroalimentación, que se entregará junto con la nota. Consistirá en 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.

Cronograma de actividades

Capítulo	Fecha	Actividades
Capítulo 1	10/08	Cátedra capítulo 1
	11/08	Publicación L01
	13/08	Lectura enunciado L01
	17/08	Sesión de trabajo L01
	20/08	Actividad práctica P01
	23/08	Entrega L01
Capítulo 2	24/08	Cátedra capítulo 2
	25/08	Publicación L02
	27/08	Lectura enunciado L02
	31/08	Sesión de trabajo L02
	03/09	Ayudantía capítulo 2
	07/09	Sesión de trabajo L02
	10/09	Sesión de trabajo L02
	14/09	Actividad práctica P02
	16/09	Entrega L02
Capítulo 3	28/09	Cátedra capítulo 3
	29/09	Publicación L03
	01/10	Lectura enunciado L03
	05/10	Sesión de trabajo L03
	08/10	Ayudantía capítulo 3
	15/10	Actividad práctica P03
	18/10	Entrega L03

Capítulo	Fecha	Actividades
Capítulo 4	19/10	Cátedra capítulo 4
	20/10	Publicación L04
	22/10	Lectura enunciado L04
	26/10	Sesión de trabajo L04
	29/10	Ayudantía capítulo 4
	02/11	Sesión de trabajo L04
	05/11	Actividad práctica P04
Capítulo 5	08/11	Entrega L04
	09/11	Cátedra capítulo 5
	10/11	Publicación L05
	12/11	Lectura enunciado L05
	16/11	Sesión de trabajo L05
	19/11	Ayudantía capítulo 5
	23/11	Sesión de trabajo L05
	26/11	Actividad práctica P05
	29/11	Entrega L05

Medios oficiales del curso

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

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 los profesores.

Salvo correos urgentes a los profesores, solo se responderán issues y correos de lunes a viernes de 8:00 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.

Política de Integridad Académica

Los alumnos 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 alumnos 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 alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (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.
- *Database Management Systems*; Ramakrishnan y Gehrke; 2002.
- *LaTeX Beginner's Guide*; Kottwitz; 2011.
- *Introduction to Algorithms*; Cormen, Leiserson, Rivest y Stein; 2009 (3^a edición).
- *Python Data Science Handbook*; VanderPlass; 2016.