



IIC2115 - PROGRAMACIÓN COMO HERRAMIENTA PARA LA INGENIERÍA

– Programa de curso –

Profesor	: Hans Löbel (halobel@ing.puc.cl)
Sitio Web	: Siding y Syllabus en GitHub (github.com/IIC2115/Syllabus)
Clases	: lunes, módulos 4 y 5 (14:00 - 16:50) - Sala A6
Horario de atención	: agendar cita por email

1 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 (y serán) fundamentales para enfrentar de manera satisfactoria problemas de ingeniería, tanto en el aspecto profesional, como en el académico.

2 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 distintos lenguajes.

3 Contenido

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

Capítulo 0: Introducción y herramientas básicas

- Tipos de lenguaje de programación.
- Entornos de desarrollo.
- Sistemas de control de versiones.

Capítulo 1: Programación orientada a objetos (OOP)

- Clases.
- Agregación y composición.
- Herencia y herencia múltiple.
- Clases abstractas.
- Diagramas de clases.

Capítulo 2: Estructuras de datos

- Stacks y colas
- Diccionarios.
- Sets.
- Árboles, listas ligadas y grafos.

Capítulo 3: Programación funcional

- Funciones de Python.
- Funciones `lambda`.
- Decoradores.

Capítulo 4: Técnicas y algoritmos

- Recursión y Backtracking
- Dividir y conquistar.
- Ordenamiento y búsqueda en arreglos.
- Búsqueda en grafos.

Capítulo 5: 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 6: Análisis de datos en Python

- Manipulación y limpieza de datos.
- Visualización.
- Clasificación y regresión.

4 Metodología

El curso sigue una metodología de clase invertida (*flipped classroom*), donde los alumnos deben estudiar los contenidos de manera previa a la clase, para luego aplicarlos en ella mediante actividades prácticas de programación. El material de estudio se encontrará disponible en el sitio del curso y consiste en apuntes, manuales y libros, donde se describen detalladamente los tópicos. Se espera además que los alumnos utilicen otras fuentes para complementar y profundizar los contenidos.

El contenido del curso contempla una división en seis capítulos, donde cada uno es cubierto durante dos sesiones. El material de cada capítulo estará disponible una semana antes del inicio de esta. En la primera sesión de cada capítulo, se realizará un control escrito corto sobre la materia y una actividad no evaluada de preparación. Esta actividad puede realizarse individualmente o en parejas. La segunda sesión contempla la realización de una actividad práctica evaluada e individual, donde se deberán aplicar todos los contenidos del capítulo actual. Para ambos tipos de actividades, se espera que la entrega se realice **durante la sesión**.

Las sesiones de cátedra no considerarán en ningún caso la revisión de materia. Sólo se contemplará un período de alrededor de 30 minutos, para contestar dudas de materia y explicar lo que se espera de la actividad a realizar. Las actividades se realizarán con la asistencia del profesor y de los ayudantes, quienes estarán disponibles para contestar dudas y aclarar conceptos. La asistencia a clases es “voluntaria” y las inasistencias a las evaluaciones serán calificadas con nota 1.0.

5 Evaluaciones

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

- Actividades prácticas (40%): se realizarán 6 actividades prácticas evaluadas (cada dos sesiones), cuya calificación se basará en su completitud y la aplicación de los contenidos considerados para esa actividad. Las actividades prácticas serán individuales y la inasistencia a una actividad práctica evaluada (justificada o no) genera automáticamente nota 1.0 en esta. La nota final de las actividades prácticas (**A**) está dada por el promedio de las **cinco** mejores.
- Tareas (40%): se realizarán cuatro tareas individualmente o en parejas, sobre tópicos cubiertos en el material y actividades del curso, mezclando contenido de distintos capítulos. Se considerará un descuento por atraso de 1.0 puntos cada seis horas o fracción. Las fechas tentativas de las tareas se presentan a continuación, donde la fecha entre paréntesis indica la fecha de entrega:
 - T₁: 02/04 (20/04)
 - T₂: 23/04 (11/05)
 - T₃: 14/05 (01/06)
 - T₄: 04/06 (22/06)

La no entrega de una tarea genera automáticamente nota 1.0 en esta. La nota final de las tareas (**T**) se obtendrá promediando las **tres** mejores notas de estas.

- Evaluaciones escritas (10%): se realizarán seis evaluaciones escritas individuales a lo largo del semestre (una por capítulo), que evaluarán los tópicos cubiertos en el material del curso. La inasistencia a una evaluación escrita (justificada o no) genera automáticamente nota 1.0 en esta. La nota final de las evaluaciones escritas (**E**) se obtendrá promediando las **cinco** mejores notas.
- Participación (10%): las sesiones de clases que **no consideren actividades prácticas evaluadas**, tendrán incidencia en la nota del curso por concepto de participación. Esta se define como la presencia y participación en la actividad en la sala de clases, además de la entrega de lo realizado. Con respecto a esto último, entregas que no consideren un mínimo de avance (que será definido para cada actividad), no serán consideradas para la participación. La nota final de asistencia (**P**) se calculará usando el porcentaje de sesiones **con actividades prácticas no evaluadas** en la que se asistió, participó y entregó, utilizando una escala lineal entre 1.0 y 7.0. Se permitirá faltar a una sesión no evaluada, sin que esto afecte la nota de participación.

6 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:

$$\mathbf{F} = 0.4 \cdot \mathbf{A} + 0.4 \cdot \mathbf{T} + 0.1 \cdot \mathbf{E} + 0.1 \cdot \mathbf{P}$$

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

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

7 Retroalimentación y correcciones

Dada la naturaleza práctica de la metodología de curso, es fundamental la entrega de retroalimentación rápida en relación a lo realizado en las actividades, con el fin de contribuir de manera temprana al correcto aprendizaje de los contenidos. Tomando esto en consideración, cada una de las entregas realizadas tendrá retroalimentación, dependiendo del tipo de actividad realizada. Para las actividades no evaluadas, se entregará en un plazo no mayor a cuatro días, una retroalimentación simple, donde se indican las debilidades y fortalezas detectadas en la entrega, con el fin de guiar mejor preparación a la actividad evaluada. La única condición para obtener la retroalimentación, es que el código entregado corra y no presente errores.

Para las actividades evaluadas y las tareas, se entregará junto con la nota, una retroalimentación 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 **sólo** a través del *Syllabus* del curso.

8 Cronograma de actividades

Fecha	Actividades	Tópicos
05/03	Introducción al curso, uso de git, IDEs	
12/03	Control 1, Actividad no evaluada	Programación orientada a objetos
19/03	Actividad evaluada	Programación orientada a objetos
26/03	Control 2, Actividad no evaluada	Estructuras de datos
02/04	Actividad evaluada	Estructuras de datos
09/04	Control 3, Actividad no evaluada	Programación funcional
16/04	Actividad evaluada	Programación funcional
23/04	Control 4, Actividad no evaluada	Técnicas y algoritmos
07/05	Actividad evaluada	Técnicas y algoritmos
14/05	Control 5, Actividad no evaluada	Bases de datos y archivos
28/05	Actividad evaluada	Bases de datos y archivos
04/06	Control 6, Actividad no evaluada	Análisis y visualización de datos
11/06	Actividad evaluada	Análisis y visualización de datos
18/06	Por definir	

9 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.

10 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 (3ª edición).
- *Python Data Science Handbook*; VanderPlass; 2016.