



## IIC2115 - PROGRAMACIÓN COMO HERRAMIENTA PARA LA INGENIERÍA

### – Programa de curso –

<b>Profesor</b>	: Hans Löbel ( <a href="mailto:halobel@ing.puc.cl">halobel@ing.puc.cl</a> ) y Francisco Garrido ( <a href="mailto:fgv@ing.puc.cl">fgv@ing.puc.cl</a> )
<b>Sitio Web</b>	: Siding y Syllabus en GitHub ( <a href="https://github.com/IIC2115/Syllabus">github.com/IIC2115/Syllabus</a> )
<b>Clases</b>	: lunes, módulos 4 y 5 (14:00 - 16:50) - Sala A7
<b>Ayudantía</b>	: jueves, módulo 5 (15:30 - 16:50) - Sala B13
<b>Horario de atención</b>	: 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 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**

- Sistema de control de versiones: git y GitHub
- Python, Jupyter Notebook y Google Colab
- Manejo de errores y debugging

#### **Capítulo 1: Programación orientada a objetos (OOP)**

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

#### **Capítulo 2: Estructuras de datos**

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

#### **Capítulo 3: Técnicas y algoritmos**

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

#### **Capítulo 4: 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 5: Análisis de datos en Python

- Manipulación y limpieza de datos
- Visualización con herramientas externas
- Clasificación y regresión
- Programación funcional

## Capítulo 6: Tópicos avanzados I

- Manejo de GIS en python
- Simulación.
- Web services y web scrapping
- Manejo de datos espaciales.

# 4 Metodología

El curso sigue una metodología de clase invertida (*flipped classroom*), donde los alumnos deben estudiar y manejar los contenidos de manera previa a la clase, para luego aplicarlos en ella mediante actividades prácticas de programación. Estas actividades son individuales, tienen una duración de 3 a 4 días (a excepción de un laboratorio final) y son acumulativas en cuanto a contenidos. Todas las actividades serán explicadas y discutidas al inicio de cada clase y al final de esta deben realizar una entrega parcial evaluada de su trabajo. Dicho esto, es fundamental que los alumnos asistan a las sesiones para recibir recomendaciones e indicaciones del cuerpo docente. De este modo puedan solucionar problemas y certificar el avance realizado.

El contenido del curso contempla una división en siete capítulos (uno introductorio y seis de contenidos), donde cada uno es cubierto durante dos o tres sesiones (con excepción del primero). El material de estudio para cada tópico se encontrará disponible en el sitio del curso aproximadamente una semana antes del inicio de la sesión correspondiente (con excepción del capítulo introductorio). **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.**

Las sesiones de cátedra no considerarán en ningún caso la revisión de materia, con excepción de las clases introductorias. Sólo se considera una revisión inicial de la actividad a realizar durante la sesión. Los laboratorios se realizarán con la presencia de los profesores y ayudantes, quienes estarán disponibles para contestar dudas y aclarar conceptos. La asistencia a clases es “voluntaria”, pero existe una fracción de la nota del curso que se asigna por concepto de trabajo en clases (asistencia y avance parcial). La no asistencia a una clase, implica inmediatamente nota 1.0 en el avance parcial sin excepción.

# 5 Evaluaciones

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

- Laboratorios prácticos (70%): se realizarán X laboratorios prácticos evaluados, cuya calificación se basará en su completitud y la aplicación de los contenidos involucrados. Los laboratorios se realizarán de manera individual. Para la entrega se utilizará la plataforma GitHub y la fecha límite será las 23:59 del día indicado en el enunciado. La no entrega de un laboratorio será calificada con nota 1.0, mientras que por atraso se descontará 1.0 punto cada 4 horas, o fracción. La nota final de los laboratorios prácticos (**L**) está dada por el promedio de las X-2 mejores notas. El laboratorio final que contempla los contenidos de todo el curso, tendrá nota coeficiente 2.

- Asistencia y participación(30%): las sesiones de clases tendrán incidencia en la nota del curso por concepto de asistencia y participación. En relación a la asistencia, esta se contabilizará en algún momento del segundo módulo de clases, ya sea con nota 1.0 ó 7.0. La participación se evaluará en base al avance parcial realizado durante la sesión, y podrá tener notas 1.0, 4.0 ó 7.0. Por cada laboratorio, la nota de asistencia y participación se calculará como el promedio simple de ambas componentes, si y solo si, la asistencia es distinta de 1.0, de lo contrario la nota en este apartado será de 1.0. La nota final de asistencia y participación (**A**) se calculará como el promedio de ls X-2 mejores.

## 6 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:

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

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 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 **sólo** a través del formulario indicado en el sitio del curso.

## 8 Cronograma de actividades

Fecha	Actividades	Tópicos
11/03	Introducción al curso	
18/08	Lab. 1	Programación orientada a objetos
25/08	Lab. 2	Programación orientada a objetos
01/04	Lab. 3	Estructuras de datos
08/04	Lab. 4	Estructuras de datos
15/04	Lab. 5	Técnicas y algoritmos
22/04	Lab. 6	Técnicas y algoritmos
29/04	Lab. 7	Bases de datos y archivos
06/05	Lab. 8	Bases de datos y archivos
13/05	Lab. 9	Análisis y visualización de datos
27/05	Lab. 10	Análisis y visualización de datos
03/06	Lab. 11	Tópicos avanzados
10/06	Lab. 12	Tópicos avanzados
17/06	Lab. 13	Tópicos avanzados

## 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.