Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación


# IIC2115 - PROGRAMACIÓN COMO HERRAMIENTA PARA LA INGENIERÍA

## – Programa de curso –

| | | |
|---|---|---|
| **Profesor** | : | Hans Löbel (`halobel@ing.puc.cl`) y Francisco Garrido (`fogarri1@uc.cl`) |
| **Sitio Web** | : | Siding y Syllabus en GitHub (`github.com/IIC2115/Syllabus`) |
| **Clases** | : | jueves, módulos 4 y 5 (14:00 - 16:50) - Sala B18 |
| **Ayudantía** | : | martes, módulo 4 (14:00 - 15:20) - Sala E10 |
| **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 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**

- Python y Jupyter Notebook
- Sistema de control de versiones: git y GitHub.
- Manejo de errores y debugging.

**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.

**Capítulo 7: Tópicos avanzado**

- Simulación.
- Web services.
- 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 laboratorios prácticos de programación. Estos laboratorios tienen una duración de 2 semanas, se realizan en parejas asignadas aleatoriamente, y son acumulativos en cuanto a contenido. Durante este lapso, los alumnos deberán asistir a las sesiones para recibir recomendaciones e indicaciones del cuerpo docente, para solucionar problemas, y para certificar el avance realizado.

El contenido del curso contempla una división en ocho capítulos (uno introductorio y siete de contenidos), donde cada uno es cubierto durante dos sesiones (con excepción del primero y el último). 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**.

Antes de iniciar un nuevo tópico, se realizará un control escrito, que evalúa el conocimiento de la materia y el trabajo realizado en el laboratorio, por lo que ambos integrantes de cada grupo tienen que conocer en detalle todo lo realizado.

Las sesiones de cátedra no considerarán en ningún caso la revisión de materia, con excepción de las clases introductorias. Los laboratorios se realizarán con la asistencia 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. Las inasistencias a los controles serán calificadas con nota 1.0.

# 5 Evaluaciones

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

- Laboratorio prácticos (55%): se realizarán 7 laboratorios prácticos evaluados, cuya calificación se basará en su completitud y la aplicación de los contenidos involucrados. Los laboratorios se realizarán en parejas aleatorias. Para la entrega se utilizará la plataforma GitHub y la fecha limité será las 23:59

del día anterior a comenzar un nuevo laboratorio. La no entrega de un laboratorio será calificada con nota 1.0, mientras que por atraso se descontará 1.0 puntos cada 4 horas, o fracción. La nota final de los laboratorios prácticos (**L**) está dada por el promedio de estos.

- Controles escritos (35%): se realizarán seis controles escritos individuales a lo largo del semestre (uno por capítulo, no hay para el capítulo final), que evaluarán los tópicos cubiertos en el material del curso y los laboratorios. La inasistencia a un control (justificada o no) genera automáticamente nota 1.0 en este. La nota final de los controles (**C**) se obtendrá como el promedio simple de estos..
- Asistencia (10%): las sesiones de clases tendrán incidencia en la nota del curso por concepto de participación. Esta se define como la presencia y participación en el laboratorio en la sala de clases, además de la certificación de lo realizado durante la semana (en caso que corresponda). La nota final de asistencia (**A**) se calculará usando el porcentaje de sesiones en la que se asistió y participó, utilizando una escala lineal entre 1.0 y 7.0. Se permitirá faltar a una sesión, sin que esto afecte la nota de asistencia.

## 6 Exigencias de aprobación

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

$$\mathbf{F} = 0.55 \cdot \mathbf{L} + 0.35 \cdot \mathbf{C} + 0.1 \cdot \mathbf{A}$$

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

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

## 7 Retroalimentación y recorrecciones

Dada la naturaleza práctica de la metodología de 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 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 recorrección **sólo** a través del *Syllabus* del curso.

## 8 Cronograma de actividades

| Fecha | Actividades | Tópicos |
|---|---|---|
| 07/08 | Introducción al curso | |
| 09/08 | Uso de herramientas básicas | Jupyter Notebook, git, GitHub y Syllabus, errores |
| 16/08 | Inicio Lab. 1 | Programación orientada a objetos |
| 23/08 | Continuación Lab. 1 | Programación orientada a objetos |
| 30/08 | Control 1, Inicio Lab. 2 | Estructuras de datos |
| 06/09 | Continuación Lab. 2 | Estructuras de datos |
| 13/09 | Control 2, Inicio Lab. 3 | Programación funcional |
| 20/09 | Continuación Lab. 3 | Programación funcional |
| 27/09 | Control 3, Inicio Lab. 4 | Técnicas y algoritmos |
| 04/10 | Continuación Lab. 4 | Técnicas y algoritmos |
| 11/10 | Control 4, Inicio Lab. 5 | Bases de datos y archivos |
| 18/10 | Continuación Lab. 5 | Bases de datos y archivos |
| 25/10 | Control 5, Inicio Lab. 6 | Análisis y visualización de datos |
| 08/11 | Continuación Lab. 6 | Análisis y visualización de datos |
| 15/11 | Control 6, Inicio Lab. 7 | Tópicos avanzados |
| 22/11 | Continuación Lab. 7 | 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.