



IIC2115 - PROGRAMACIÓN COMO HERRAMIENTA PARA LA INGENIERÍA

– Programa de curso –

Profesor	: 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: Extracción, almacenamiento y consulta de datos

- Manejo de archivos
- Web scrapping y expresiones regulares
- Modelo relacional de datos
- Consultas sobre datos usando SQL
- Uso de SQL en Python

Capítulo 4: Análisis de datos tabulares

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

Capítulo 5: Análisis de datos geoespaciales

- Manejo de GIS en Python
- Manejo de redes en Python

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 talleres, actividades prácticas, y laboratorios de programación. Todos estos elementos 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 cada uno tiene una duración de tres semanas. Cada capítulo considera 4 tipos de sesiones, descritas a continuación:

- Resúmenes de materia: consisten en la consolidación de los contenidos cubiertos por el material del capítulo, que han sido previamente estudiados por los alumnos, mediante una exposición por parte

del profesor del curso. Estas sesiones se realizan durante las dos primeras semanas de cada capítulo, durante el primer módulo de los días lunes.

- Talleres: consisten en la resolución individual por parte de los alumnos de un problema de programación centrado en algunos tópicos del capítulo. Estas sesiones se realizan durante las dos primeras semanas de cada capítulo, durante el segundo módulo de los días lunes.
- Ayudantías: consisten en el desarrollo y resolución por parte de los ayudantes del taller realizado el lunes inmediatamente anterior, además de la resolución de dudas de los contenidos del curso. Estas sesiones se realizan los dos primeros jueves de cada capítulo.
- Actividad práctica: consiste en la resolución individual por parte de los alumnos de problemas de programación centrados en todos los tópicos del capítulo. Esta sesión se realiza únicamente durante la tercera semana de cada capítulo, durante ambos módulos del día lunes.

Las sesiones de resumen de materia y ayudantía son realizadas a través de Zoom. Las sesiones destinadas a los talleres y actividades prácticas consideran el trabajo en una sala virtual en la plataforma Discord, con la presencia de integrantes del cuerpo docente con el fin de resolver las 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 validar 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 tres tipos, cada una con su correspondiente nota final promedio:

- Actividades prácticas (50%): cada capítulo considera el desarrollo individual de una actividad práctica de programación. La calificación de esta se basará en su completitud y la aplicación de los contenidos involucrados. Cada actividad será planificada para poder ser resuelta en aproximadamente 120 minutos. Para la entrega se utilizará la plataforma GitHub y la hora límite será las 18:30 del día en que se realiza la actividad, con el fin de evitar problemas de último minuto asociados a conectividad u otros. Actividades entregadas pasada la hora límite no serán corregidas. La nota final de las actividades prácticas (**A**) está dada por el promedio de estas.
- Laboratorios (40%): el curso considera el desarrollo individual de dos laboratorios de programación, que se centrarán en los tópicos de los capítulos 2 y 3 (L1), y 4 y 5 (L2). En ambos casos, los

tópicos cubiertos son acumulativos, es decir, lo cubierto en los capítulos anteriores a los del foco de cada laboratorio también podrán ser incluidos. 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. Laboratorios con atrasos superiores a 6 horas serán calificadas con nota 1.0. Tienen hasta 12 horas después de la hora oficial de entrega de los laboratorios para llenar el formulario disponible en el Syllabus, indicando los datos del *commit* atrasado que desean que sea revisado. En caso que no se llene el formulario dentro del plazo se revisará el último *commit* (de la carpeta LX correspondiente) realizado dentro del plazo de entrega.

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

- Participación (10%): durante las sesiones de taller, los alumnos deberán realizar un cierto nivel de avance, con el fin de validar el aprendizaje de los tópicos del capítulo cubiertos hasta el momento. Los detalles del avance para cada taller serán especificados en el enunciado y corresponderán a elementos básicos de la materia. Para la entrega se utilizará la plataforma GitHub y la hora límite será las 16:50 del día en que se realiza el taller. Talleres entregados pasada la hora límite no serán corregidos. La evaluación de este ítem será binaria, es decir, se obtendrá un 7,0 si se cumple el avance y un 1,0 si no. La nota final de participación (**P**) está dada por el promedio de los 7 mejores talleres.

Exigencias de aprobación

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

$$\mathbf{F} = 0.5 \cdot \mathbf{A} + 0.4 \cdot \mathbf{L} + 0.1 \cdot \mathbf{P}$$

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

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

Cronograma de actividades

Capítulo	Fecha	Actividades
Capítulo 1	22/03	Resumen capítulo 1a Taller T1a
	25/03	Ayudantía T1a
	29/03	Resumen capítulo 1b Taller T1b
	05/04	Ayudantía T1b
	05/04 - 08/04	Actividad práctica A1
Capítulo 2	12/04	Resumen capítulo 2a Taller T2a
	15/04	Ayudantía T2a
	19/04	Resumen capítulo 2b Taller T2b
	22/04	Ayudantía T2b
	26/04	Actividad práctica A2
	29/04	Ayudantía L1
Capítulo 3	03/05	Resumen capítulo 3a Taller T3a
	06/05	Ayudantía T3a
	17/05	Resumen capítulo 3b Taller T3b
	20/05	Ayudantía T3b
	24/05	Actividad práctica A3

Capítulo	Fecha	Actividades
Capítulo 4	12/04	Resumen capítulo 4a Taller T4a
	15/04	Ayudantía T4a
	19/04	Resumen capítulo 4b Taller T4b
	22/04	Ayudantía T4b
	26/04	Actividad práctica A4
	29/04	Ayudantía L2
Capítulo 5	03/05	Resumen capítulo 5a Taller T5a
	06/05	Ayudantía T5a
	17/05	Resumen capítulo 5b Taller T5b
	20/05	Ayudantía T5b
	24/05	Actividad práctica A5

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 las evaluaciones 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.

Medios oficiales del curso

El sitio oficial del curso será el *Syllabus*, donde se publicarán los apuntes del curso, las *slides* usadas en cátedra y los grabaciones de estas. También serán publicados aquí los ejercicios y grabaciones 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 del profesor.

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 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.

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.