

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



# IIC2115 - Programación como Herramienta para la Ingeniería

## Introducción

**Profesores:** Hans Löbel  
Pablo Seisdedos

Todos tienen motivaciones distintas para escribir código

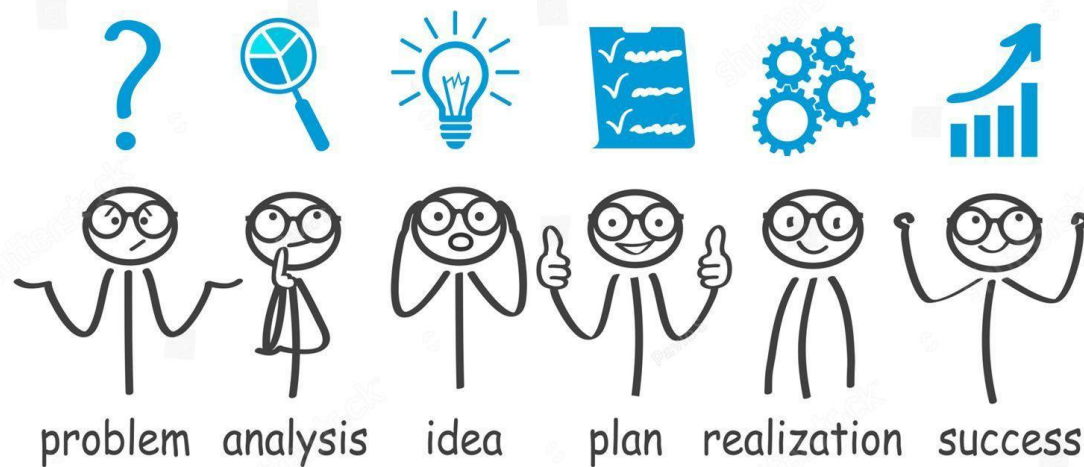


Algunos no tienen ninguna  
motivación para hacer esto



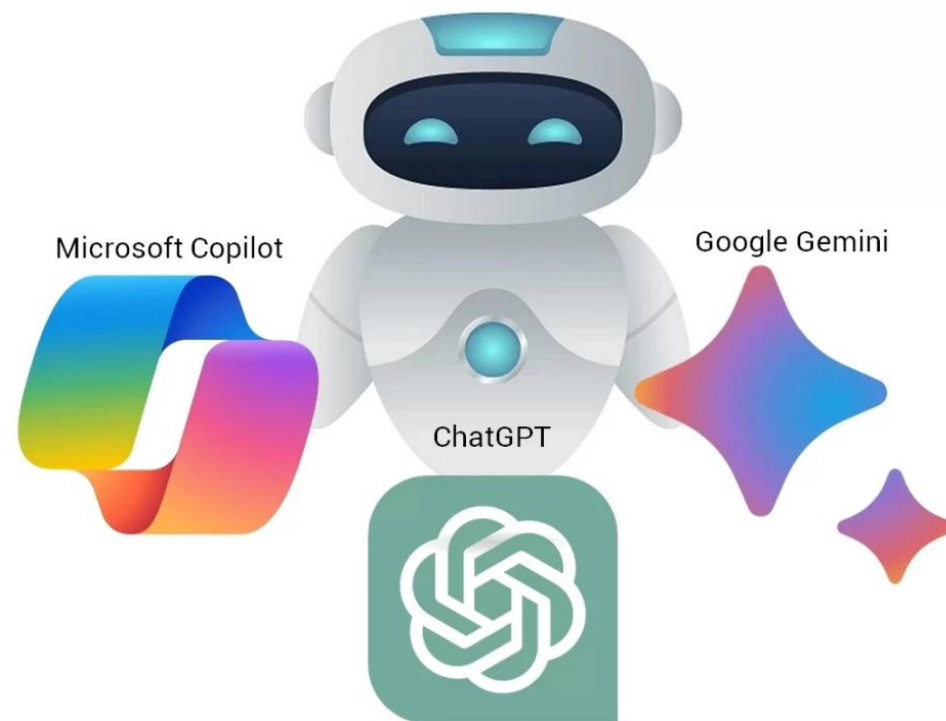
¿PARA QUÉ TENGO QUE TOMAR ESTE CURSO?

# Más que programar por programar, en este curso buscamos resolver problemas usando programación



La **programación** es en este contexto netamente una **herramienta**, que en la actualidad puede ser realizada de manera muy eficiente con la ayuda de **asistentes inteligentes**

Sí, en este curso queremos que usen  
ChatGPT u otros lo más posible  
(cumpliendo algunas reglas)



El curso busca preparar y especializar a los estudiantes en el uso de la **programación** como una **herramienta** para **solucionar problemas** avanzados, principalmente basados en datos

Desarrollar habilidades de trabajo **autónomo**, **pensamiento crítico** y toma de **decisiones** basada en supuestos razonables, para solucionar problemas

# Contenidos

El curso cubre múltiples temas divididos en 7 capítulos de contenido, con énfasis en el uso de datos:

- Capítulo 1: Fundamentos (POO y EDD)

# Contenidos

El curso cubre múltiples temas divididos en 7 capítulos de contenido, con énfasis en el uso de datos:

- Capítulo 1: Fundamentos (POO y EDD)
- Capítulo 2: Manejo y Análisis de Datos Tabulares





# Contenidos

El curso cubre múltiples temas divididos en 7 capítulos de contenido, con énfasis en el uso de datos:

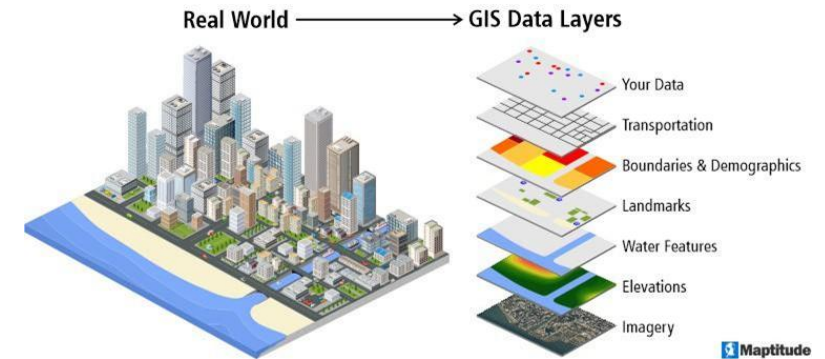
- Capítulo 1: Fundamentos (POO y EDD)
- Capítulo 2: Manejo y Análisis de Datos Tabulares
- Capítulo 3: Modelos Predictivos



# Contenidos

El curso cubre múltiples temas divididos en 7 capítulos de contenido, con énfasis en el uso de datos:

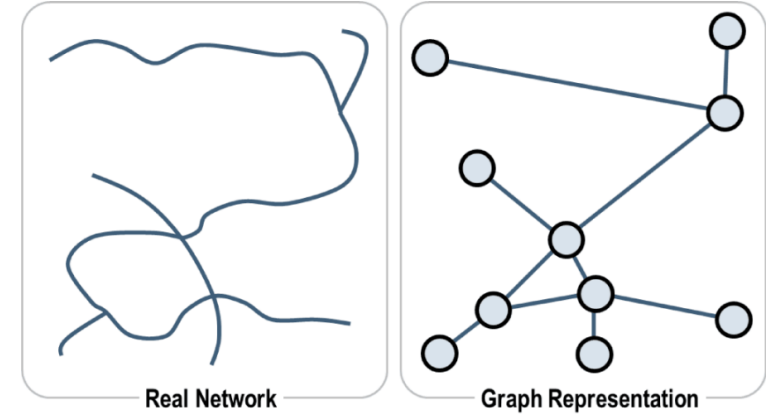
- Capítulo 1: Fundamentos (POO y EDD)
- Capítulo 2: Manejo y Análisis de Datos Tabulares
- Capítulo 3: Modelos Predictivos
- Capítulo 4: Análisis de Datos Geoespaciales



# Contenidos

El curso cubre múltiples temas divididos en 7 capítulos de contenido, con énfasis en el uso de datos:

- Capítulo 1: Fundamentos (POO y EDD)
- Capítulo 2: Manejo y Análisis de Datos Tabulares
- Capítulo 3: Modelos Predictivos
- Capítulo 4: Análisis de Datos Geoespaciales
- Capítulo 5: Análisis de Datos Estructurados



# Contenidos

El curso cubre múltiples temas divididos en 7 capítulos de contenido, con énfasis en el uso de datos:

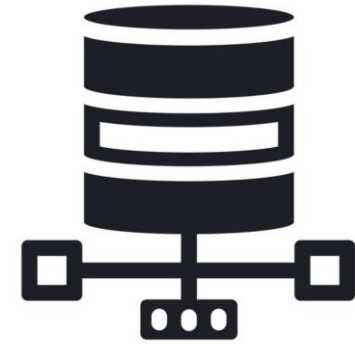
- Capítulo 1: Fundamentos (POO y EDD)
- Capítulo 2: Manejo y Análisis de Datos Tabulares
- Capítulo 3: Modelos Predictivos
- Capítulo 4: Análisis de Datos Geoespaciales
- Capítulo 5: Análisis de Datos Estructurados
- Capítulo 6: Extracción y Visualización de Datos



# Contenidos

El curso cubre múltiples temas divididos en 7 capítulos de contenido, con énfasis en el uso de datos:

- Capítulo 1: Fundamentos (POO y EDD)
- Capítulo 2: Manejo y Análisis de Datos Tabulares
- Capítulo 3: Modelos Predictivos
- Capítulo 4: Análisis de Datos Geoespaciales
- Capítulo 5: Análisis de Datos Estructurados
- Capítulo 6: Extracción y Visualización de Datos
- Capítulo 7: Manejo y Consultas sobre Datos Relacionales



RELATIONAL DATABASE



# Contenidos

El curso cubre múltiples temas divididos en 7 capítulos de contenido, con énfasis en el uso de datos:

- Capítulo 1: Fundamentos (POO y EDD)
- Capítulo 2: Manejo y Análisis de Datos Tabulares
- Capítulo 3: Modelos Predictivos
- Capítulo 4: Análisis de Datos Geoespaciales
- Capítulo 5: Análisis de Datos Estructurados
- Capítulo 6: Extracción y Visualización de Datos
- Capítulo 7: Manejo y Consultas sobre Datos Relacionales

Todo el desarrollo estará basado en Python y sus librerías, las cuales variarán dependiendo del tema

# Metodología

- El curso sigue una metodología de taller, donde **deben estudiar y practicar los contenidos de manera previa y durante las clases**, para **luego aplicarlos** en ejercicios prácticos.
- Cada capítulo se **desarrollará durante 2 semanas**, mediante 3 tipos de sesiones:
  - Cátedra: resumen, dudas, ejemplos -> ejercicio formativo (tutorial) + **ticket de salida**
  - Laboratorio: trabajo evaluado, primera parte durante la sesión, segunda para el resto de la semana
  - Ayudantía: resolución de ejercicios y dudas
- Todas consideran la participación del cuerpo docente, entregando retroalimentación y contestando dudas.

# Cronograma

#	Laboratorio	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo	Mes
1	Introducción	3	4	5	6	7	8	9	Marzo
2		10	11	12	13	14	15	16	
3	L01	17	18	19	20	21	22	23	
4		24	25	26	27	28	29	30	
5	L02	31	1	2	3	4	5	6	Abril
6		7	8	9	10	11	12	13	
7	L03	14	15	16	17	18	19	20	
8		21	22	23	24	25	26	27	
9	Receso	28	29	30	1	2	3	4	Mayo
10	L04	5	6	7	8	9	10	11	
11		12	13	14	15	16	17	18	
12	L05	19	20	21	22	23	24	25	
13		26	27	28	29	30	31	1	
14	L06	2	3	4	5	6	7	8	Junio
15		9	10	11	12	13	14	15	
16	L07	16	17	18	19	20	21	22	
17		23	24	25	26	27	28	29	
18	Cierre	30	1	2	3	4	5	6	Julio
19		7	8	9	10	11	12	13	
20		14	15	16	17	18	19	20	



# Esquema de evaluación

Laboratorios (L) → 80%

Promedio de los 6 mejores, se borra 1

Participación (P) → 20%

Proporción de sesiones de cátedra en que se participó,  
verificado con ticket de salida y entrega en repositorio

- Ausencias solo se pueden justificar con certificado de la DiPre

## Medios oficiales del curso

Todo lo relacionado con el curso se encuentra en el *Syllabus*

<http://github.com/IIC2115/Syllabus>

- Dudas de materia o administrativas se pueden plantear como *issues* (foro), que serán respondidas oportunamente por el cuerpo docente del curso.
- El *Syllabus* está montado sobre la plataforma GitHub, que usaremos de manera constante durante todo el semestre (más detalles en un rato).
- Solo avisos y notas en Canvas.

## Medios oficiales del curso

El sitio oficial del curso será el [Syllabus](#), donde se publicarán los apuntes del curso, *slides* usadas en cátedra, ejercicios de ayudantías y rúbricas de corrección. Además, se encuentra disponible el material completa del cursos, incluyendo ejercicios, de los últimos 7 años.

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@uc.cl](mailto:iic2115.ing@uc.cl). Situaciones urgentes, de mayor importancia o personales que requieran de alta privacidad deben realizarse directamente al mail del profesor.

Salvo correos urgentes al profesor, solo se responderán *issues* y correos de lunes a viernes de 8:30 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.

# Python Notebooks y Google Colab

- Python Notebook es un formato de archivo que describe documentos (*notebooks*) que contienen código fuente, ecuaciones, visualizaciones, texto explicativo, y más.
- Su interfaz se presenta típicamente como una página web, que nos permite, dentro de otras cosas, interactuar con código Python.
- En este curso utilizaremos la plataforma Colab de Google (<https://colab.research.google.com>), que permite el uso de notebooks sin la necesidad de instalar nada.



## ¿Por qué usar **notebooks**?

- El uso de notebooks facilita el desarrollo rápido y la claridad del código.
- Además, funciona en todos los sistemas operativos usados regularmente.
- Será uno de los estándares para este curso.



Vamos a Colab...



Unamos todo ahora: git, GitHub y Colab



git es un sistema distribuido de control de versiones de archivos de código, gratuito y open source, diseñado para manejar desde pequeños a enormes proyectos de software de forma rápida y eficiente.



Es una plataforma para alojar proyectos usando el sistema de control de versiones git.



git → Sistema de control de versiones

GitHub → Plataforma para alojar repositorios de código

Colab → Plataforma para escribir código (y más) en notebooks

- Syllabus del curso es un repositorio en GitHub, que contiene principalmente notebooks.
- Si bien el material puede ser revisado online, para ejecutarlo deben usar una plataforma para ejecutar notebooks.
- La opción más simple para esto es mediante Colab.
- Si no les gusta Colab, deberán primero descargar los archivos desde el Syllabus y luego abrirlos con la plataforma que prefieran.





git → Sistema de control de versiones

GitHub → Plataforma para alojar repositorios de código

Colab → Plataforma para escribir código (y más) en notebooks

- Para entregar las evaluaciones, cada uno tendrá un repositorio privado (independiente y distinto del Syllabus). La interacción con este puede ser con drag & drop (recomendado), o a través de la terminal (mayor flexibilidad). Recibirán en unos minutos un mail con las instrucciones para la creación.
- Interacción adecuada con estos repositorios es fundamental para el éxito en el curso.



- Para interactuar con los repositorios, y el Syllabus en particular, se utilizan principalmente los siguientes comandos a través de una terminal:

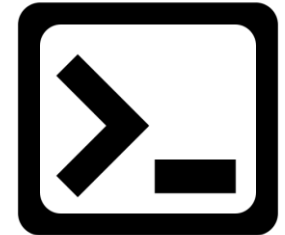
`git clone [link]`

`git add [path] o git add .`

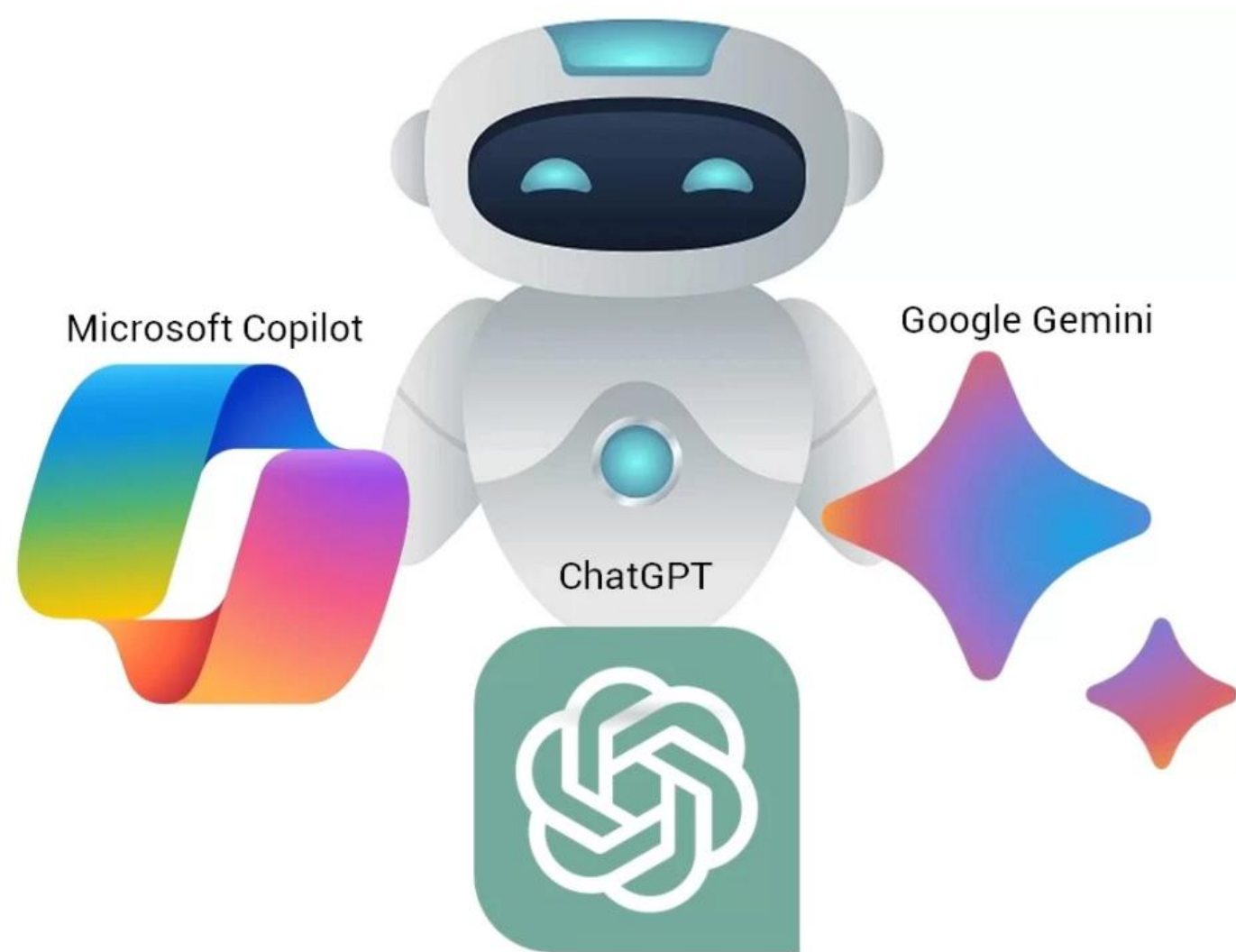
`git commit -m "[msg]"`

`git push`

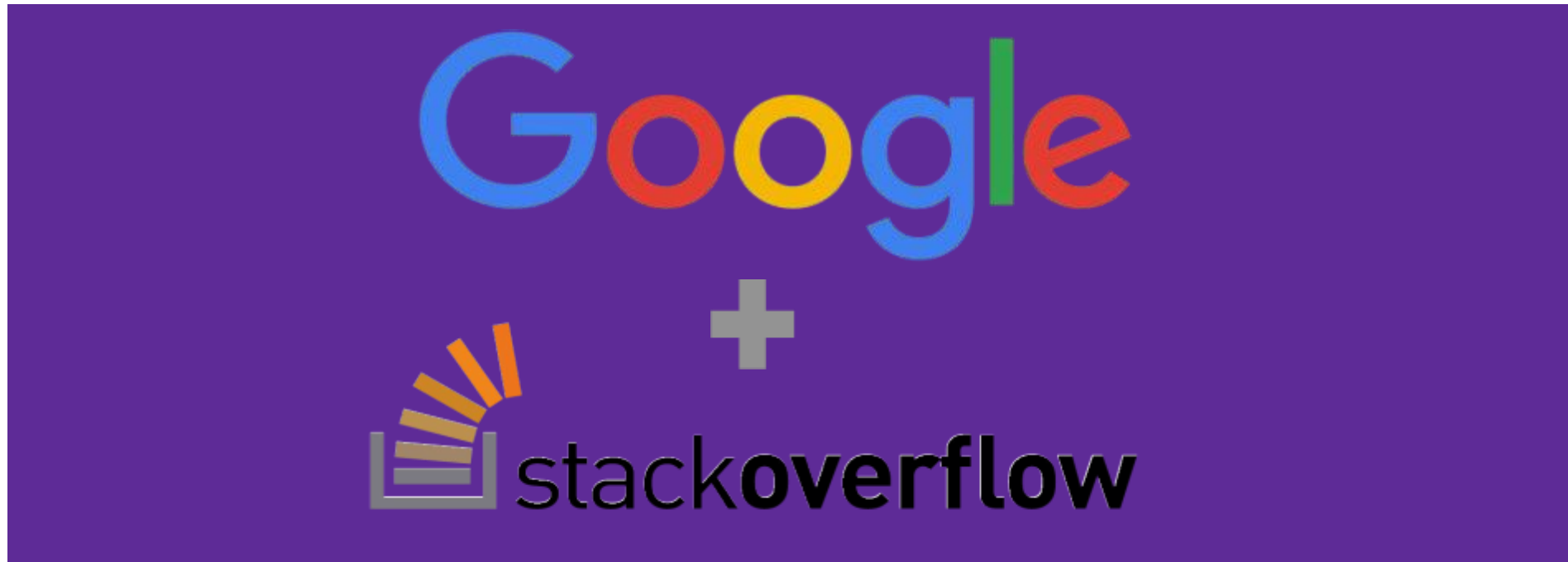
`git pull`



¿Cómo buscar soluciones a los problemas?



¿Cómo buscar soluciones a los problemas?



## MUY IMPORTANTE

Para el caso específico de los agentes inteligentes como ChatGPT, Gemini o Copilot, se recomienda su uso en todas las instancias del curso, siempre y cuando se indique explícitamente y al inicio de los archivos entregados, que sí se utilizó la ayuda de dicho asistentes. Adicionalmente, cada vez que se utilicen, se deberá indicar de manera clara su modo de uso (al menos el *prompt* y respuesta) y cómo las respuestas entregadas fueron modificadas sustancialmente. Se entiende como una modificación sustancial aquella que involucra el análisis crítico de la respuesta del asistente, y todas las modificaciones y mejoras que de este análisis se desprendan.

Revisar más detalles en el programa del curso

## MUY IMPORTANTE

*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.*

Revisar más detalles en el programa del curso

## Como sigue la sesión de hoy

- Revisar el programa del curso nuevamente.
- Trabajar en ejercicio/tutorial disponible en el Syllabus.
- Al terminar la sesión, subir el avance al repositorio privado.
- Luego de subir el avance, responder el ticket de salida (**el ticket de hoy también cuenta para la nota de participación**).

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



# IIC2115 - Programación como Herramienta para la Ingeniería

## Introducción

**Profesores:** Hans Löbel  
Pablo Seisdedos