

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



IIC2115 - Programación como Herramienta para la Ingeniería

Introducción

Profesor: Hans Löbel

Todos tienen motivaciones distintas para escribir código

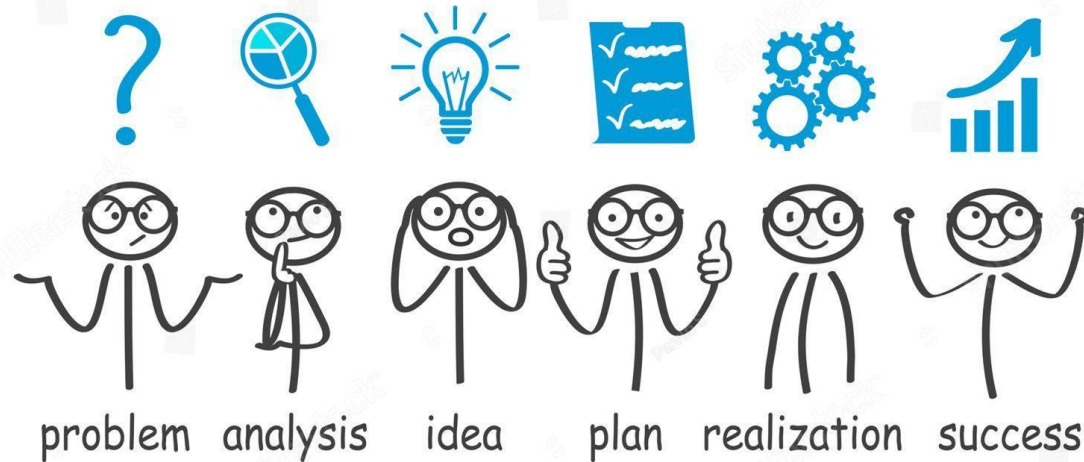


Algunos no tienen ninguna
motivación para hacer esto



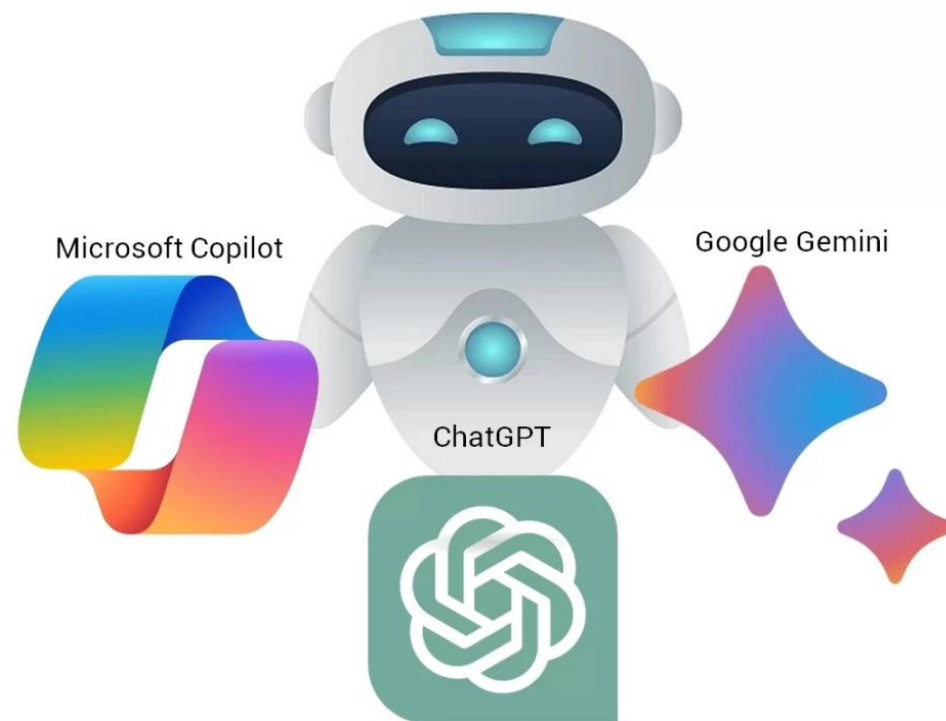
¿PARA QUÉ TENGO QUE TOMAR ESTE CURSO?

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, de manera sistemática y estructurada



El curso busca preparar y especializar a los estudiantes en el uso de la **programación** asistida por IA, 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 basados en evidencia, 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, EDD, uso de IA

Contenidos

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

- Capítulo 1: Fundamentos -> POO, EDD, uso de IA
- 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:

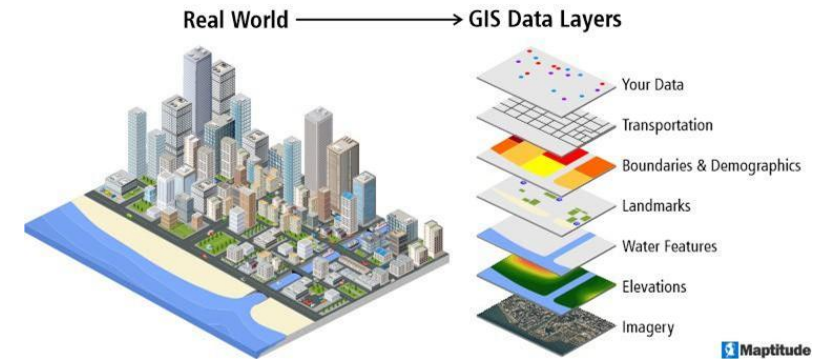
- Capítulo 1: Fundamentos -> POO, EDD, uso de IA
- Capítulo 2: Manejo y Análisis de Datos Tabulares
- Capítulo 3: Modelos Predictivos



Contenidos

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

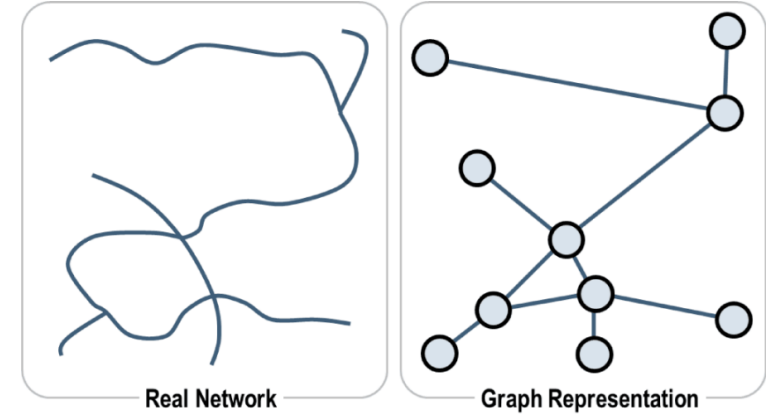
- Capítulo 1: Fundamentos -> POO, EDD, uso de IA
- Capítulo 2: Manejo y Análisis de Datos Tabulares
- Capítulo 3: Modelos Predictivos
- Capítulo 4: Análisis de Datos Geoespaciales



Contenidos

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

- Capítulo 1: Fundamentos -> POO, EDD, uso de IA
- 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:

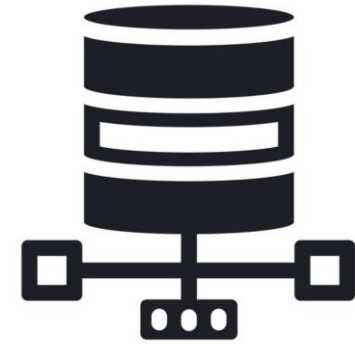
- Capítulo 1: Fundamentos -> POO, EDD, uso de IA
- Capítulo 2: Manejo y Análisis de Datos Tabulares
- Capítulo 3: Modelos Predictivos
- Capítulo 4: Análisis de Datos Geoespaciales
- Capítulo 5: Análisis de Datos Estructurados
- Capítulo 6: Extracción y Visualización de Datos



Contenidos

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

- Capítulo 1: Fundamentos -> POO, EDD, uso de IA
- 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, EDD, uso de IA
- 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

Cronograma

Laboratorio	Lunes	Martes	Miercoles	Jueves	Viernes	Sábado	Domingo	Mes
Introduccion	4	5	6	7	8	9	10	Agosto
L01	11	12	13	14	15	16	17	
	18	19	20	21	22	23	24	
L02	25	26	27	28	29	30	31	Septiembre
	1	2	3	4	5	6	7	
L03	8	9	10	11	12	13	14	
Receso	15	16	17	18	19	20	21	Octubre
L03	22	23	24	25	26	27	28	
	29	30	1	2	3	4	5	
L04	6	7	8	9	10	11	12	Noviembre
	13	14	15	16	17	18	19	
L05	20	21	22	23	24	25	26	
	27	28	29	30	31	1	2	Diciembre
L06	3	4	5	6	7	8	9	
	10	11	12	13	14	15	16	
L07	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
Cierre	1	2	3	4	5	6	7	
	8	9	10	11	12	13	14	
	15	16	17					

Metodología del curso

- 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** (salvo el Capítulo 3), mediante 3 tipos de sesiones:
 - Cátedra (primera semana): resumen, dudas, ejemplos -> ejercicios formativos de preparación + **ticket de salida**
 - Laboratorio (segunda semana): trabajo evaluado, primera parte durante la sesión, segunda para el resto de la semana
 - Ayudantía (ambas semanas): resolución de ejercicios y dudas
- Todas consideran la participación del cuerpo docente, entregando retroalimentación y contestando dudas.

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
- Mucho más detalle en el programa del curso.

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 es el *Syllabus* (<https://www.github.com/IIC2115/Syllabus>), donde se publicarán los apuntes del curso, *diapositivas* usadas en las sesiones de cátedra, ejercicios de ayudantías y rúbricas de corrección. Además, se encuentra disponible el material completo del curso, incluyendo ejercicios, desde el año 2017.

Fuera de las sesiones del curso, las dudas sobre los contenidos o aspectos administrativos de interés general deben realizarse **exclusivamente** a través de *issues* en el *Syllabus*, y no serán respondidas si se realizan por otro medio. Cualquier comunicación personal relativa a aspectos administrativos del curso (**no dudas de materia**) debe dirigirse al mail 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 8:20 a 17:20. Este horario será válido también en el sentido inverso, o sea, los estudiantes no recibirán correos o avisos por parte del cuerpo docente del curso fuera de este horario, salvo que sean temas urgentes.

Material del curso y desarrollo de código

- En este curso utilizaremos el formato Python Notebook (.ipynb) para todo: materia, entregas, etc.
- Python Notebook es un formato de archivo que describe documentos (*notebooks*) que contienen código, 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.
- Se recomienda el uso de la plataforma web Colab de Google (<https://colab.research.google.com>), o JupyterLab (<https://jupyter.org/>) de manera local.



Vamos a Colab...



¿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, es decir, no se permitirá el uso de archivos .py (o pegar todo el contenido de un .py en una única celda del notebook)



¿Cómo manejaremos los archivos del curso?

- Con el fin de mantener el orden y la consistencia, todos los archivos del curso (materia, entregas, etc.) se manejarán a través de un **sistema de control de versiones**.
- Un sistema de control de versiones es una herramienta que permite guardar el historial de los cambios en los archivos, de forma de, por ejemplo, volver atrás si algo sale mal.
- Un ejemplo muy popular de estos sistemas es **git**.
- **GitHub** Es una plataforma para alojar proyectos usando el sistema de control de versiones **git**.



GitHub



git → Sistema de control de versiones

GitHub → Plataforma para alojar repositorios de código

Colab o Jupyter → Plataformas 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, como Colab o Jupyter Lab
- La opción más simple para esto es mediante Colab, ya que no requiere descargar nada.
- Para ejecutarlos con Jupyter Lab, deberán primero descargar los archivos desde el Syllabus y luego abrirlos con esta plataforma.



git → Sistema de control de versiones

GitHub → Plataforma para alojar repositorios de código

Colab o Jupyter → Plataformas para escribir código (y más) en notebooks

- Para entregar las evaluaciones, cada uno tendrá un repositorio privado en GitHub (**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 a través de la terminal, se utilizan principalmente los siguientes comandos:

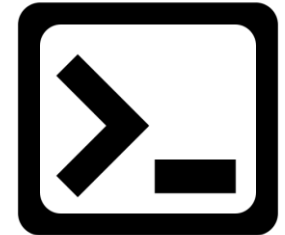
`git clone [link]`

`git add [path] o git add .`

`git commit -m "[msg]"`

`git push`

`git pull`



Pasemos ahora a lo central, ¿cómo enfrentaremos en este curso el uso de la programación como una herramienta?

Introducción

En este laboratorio, deberán desarrollar un sistema para la gestión, consulta de información y simulación en un sistema simple de transporte público urbano basado en buses, con el objetivo es aplicar conceptos de POO y estructuras de datos básicas para modelar las relaciones entre los distintos componentes.

El sistema deberá permitir construir la red y sus usuario de manera aleatoria o a partir de la información contenida en un archivo, consultar ciertos aspectos de la operación de la red y los servicios, y realizar simulaciones del funcionamiento del sistema.

Descripción del Sistema

El sistema de estará compuesto por los siguientes elementos principales:

Servicios de buses

Cada servicio de buses estará identificado por un nombre único (por ejemplo, "S1"). Un servicio de buses deberá pasar por un número predefinido de estaciones, que deben ser recorridas de manera secuencial sin repetir, teniendo un tiempo de viaje promedio para cada tramo entre estaciones, dependiendo del horario (bajo, punta mañana, valle, punta tarde). No es necesario que el recorrido del servicio sea el mismo para la idea y la vuelta. Debe considerar además la existencia de servicios especiales, que cambian su recorrido dependiendo del horario.

Estaciones

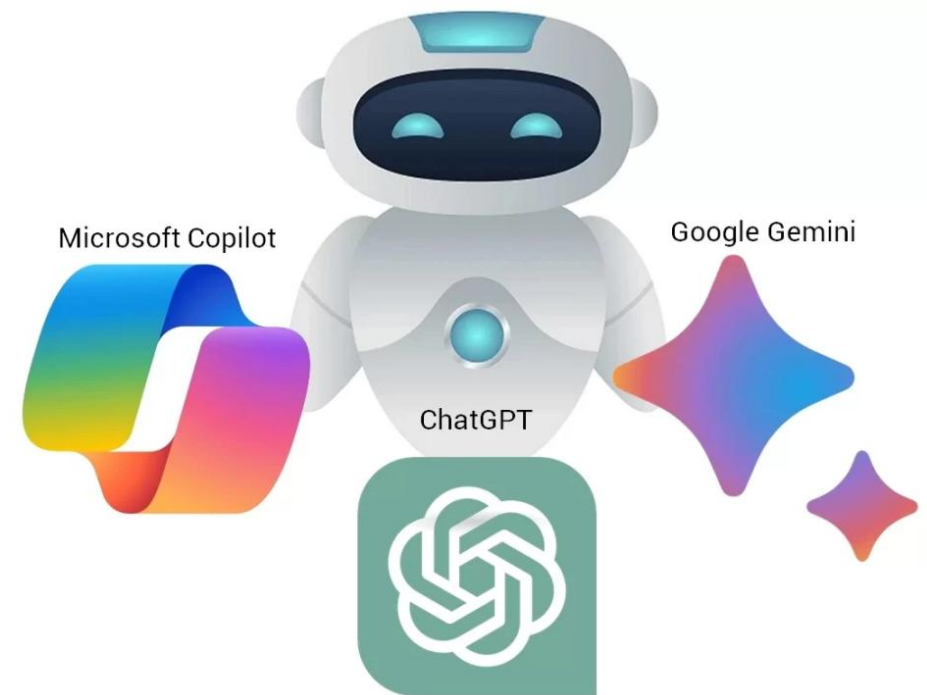
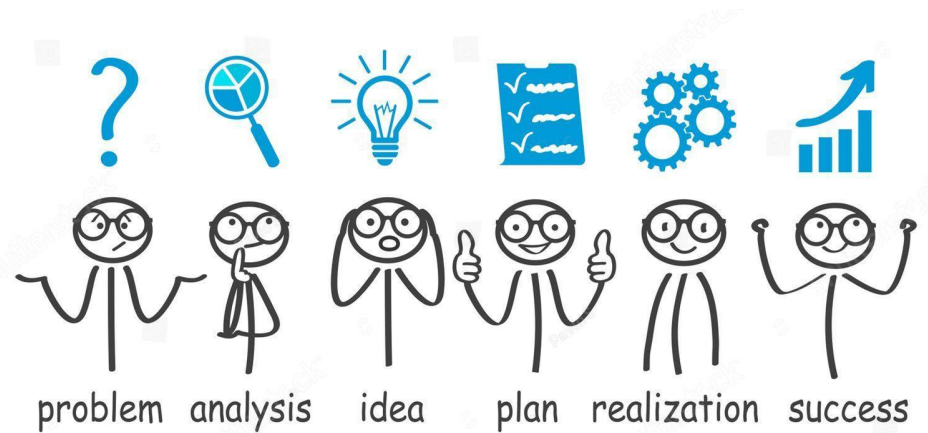
Las estaciones estarán identificadas por un nombre único (por ejemplo, "E1"). Cada estación podrá ser parte de múltiples recorridos de servicios de buses y deberá mantener un registro de los pasajeros que suben y bajan en ella. Existen estaciones de 2 tipos, básicas y con accesibilidad universal, donde estas últimas difieren de las básicas en que permiten que pasajeros con problemas de movilidad puedan subir o bajar de los buses.

Pasajeros

Cada pasajero tendrá un identificador único y estará asociado a una estación de origen, una de destino y un momento de inicio del viaje. Además, cada pasajero podrá tener un saldo en su tarjeta de transporte, que será necesario para realizar viajes. Los pasajeros pueden tener problemas de accesibilidad, lo que restringe las estaciones que pueden utilizar.

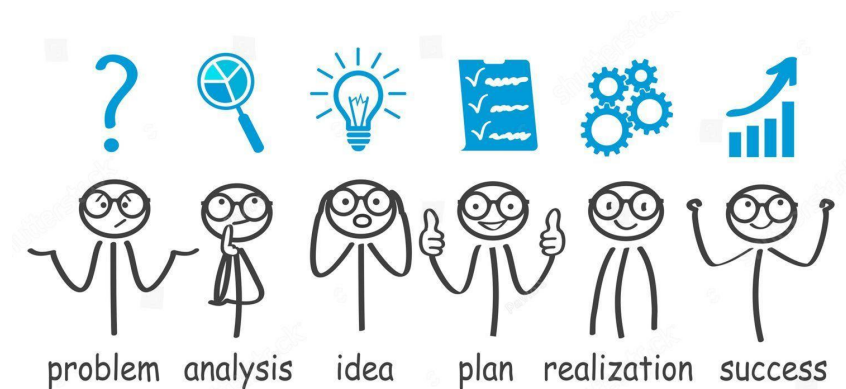
¿Cuál es el principal problema de alimentar directamente con esto a un asistente basado en IA ?

Para evitar estos problemas, utilizaremos un enfoque basado en el uso estructurado de asistencia basada en IA



Qué significa el uso estructurado de la IA

- Los detalles variarán en cada capítulo, pero se espera que la resolución involucre:
 - Descripción a alto nivel de la solución propuesta.
 - Subdivisión de la solución en módulos relevantes.
 - Prompts usados en cada módulo
 - Evidencia de la correctitud de cada módulo (relación con la materia, tests de código, etc).
- En resumidas cuentas, pueden usar IA para todo, pero deben entregar evidencia de colaboración con esta, no de tercerización de las labores.
- Esquema estructurado debe verse reflejado también en el archivo entregado, subdividiendo los módulos en celdas.



Qué es lo que NO se espera (ni se aceptará) del uso de IA en este curso

- Cosas del tipo: poner el pdf en el asistente inteligente y copiar lo que genera como salida.
- Link al chat con el asistente como evidencia de su correcto uso.
- Cualquier cosa que implique **no dar valor agregado** a lo realizado por el asistente.
- 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 disponible en el Syllabus.
- Durante y al terminar la sesión, subir el avance al repositorio privado.
- Responder el ticket de salida a más tardar hasta las 17:30.
- 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

Profesor: Hans Löbel