

Programación Avanzada

IIC2233 2025-1

Pablo Araneda - Daniela Concha - Francisca Ibarra - Lucas Van Sint Jan - Francisca Cattan



Agenda

- Motivación
- Equipo docente
- Programa y metodología del curso
- Recomendaciones
- Herramientas del curso
- GitHub y Git
- El curso en GitHub

Motivación



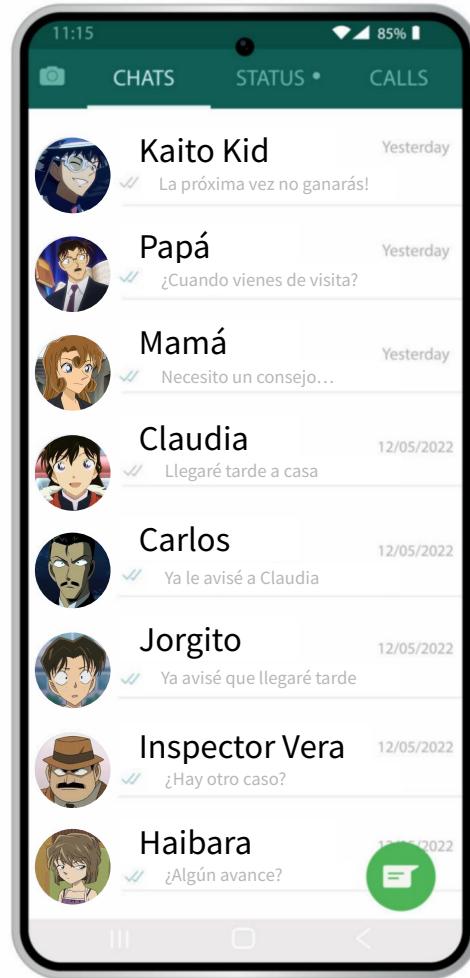
WhatsApp

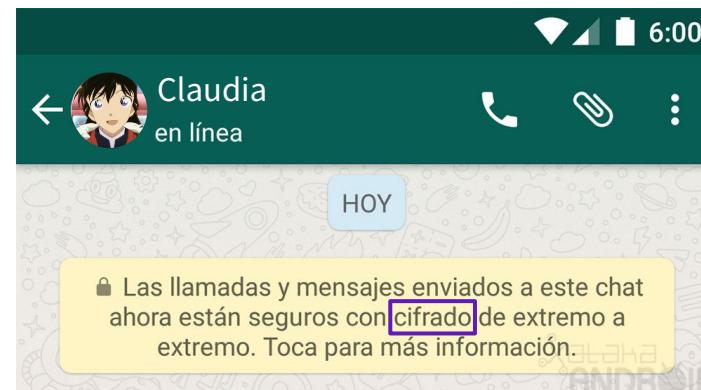
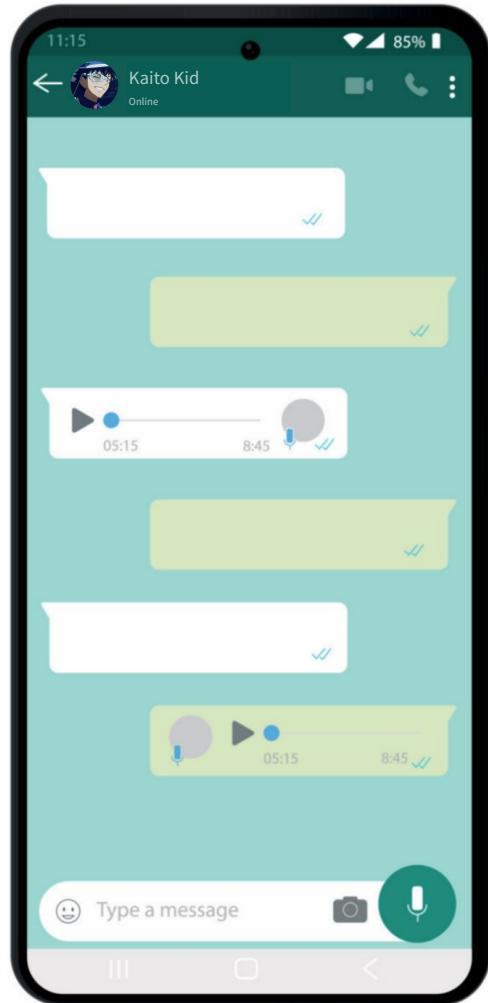
¿Qué implicaría **programar** la aplicación
WhatsApp usando lo que sabemos de
Introducción a la Programación?



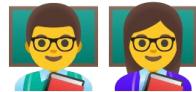
Vamos por partes

- Disponer de un perfil con tu nombre, estado, telefono y foto.
- Mostrar todos los grupos con sus chats.
- Interfaz gráfica.
- Comunicación 1 a 1 o grupal.
- Envío encriptado de información. Ya sea mensajes, archivos, fotos, videos.





Equipo docente



Profesores



Sección 1
Pablo



Sección 2
Daniela



Sección 3
~~Francisca~~
gatochico



Sección 4
Lucas



Sección 5
~~Francisca~~
Paqui



Ayudantes jefes



Referencia: "The Boys"



Ayudantes jefes verdaderos

Cris
Coordinador
General

Claudio
Coordinadora
de Tareas

Cata
Coordinadora de
Automatización



Amanda
Coordinadora
Bienestar

Julio
Coordinador de
Docencia



Cuerpo de ayudantes (muchos...)



Apoyo en cátedra y ayudantía

Ayudantes encargados de realizar las ayudantías y acompañarlos en las actividades



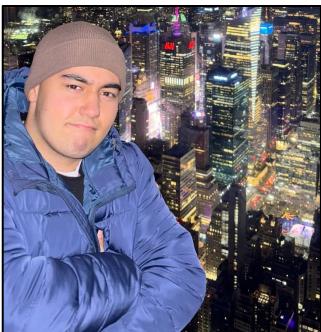
Julio



Francisca



Diego



Agustín



Carlos



Apoyo bienestar

Solicitudes de apoyo ante problemas personales (Correos en la [Wiki del curso](#)).



Sección 1
Tere



Sección 2
Maxi



Sección 3
Amanda



Sección 4
Augusta



Sección 5
Melany

Programa y metodología

Plataformas



- Envío de anuncios
- Publicación de notas y *feedback* correcciones
- Desarrollo de Controles de Salida
- **¡NO SE ENTREGAN LAS
EVALUACIONES POR ACA!**

GitHub



Syllabus



Contenidos



Wiki



Repo.
Privado

- Syllabus:
 - Material de cátedra y enunciado evaluaciones
 - Foro donde se responderán todas las dudas
- Contenidos:
 - Material para estudio autónomo
- Wiki:
 - Guía de instalación, reglas del foro, etc.
- Repositorio Privado:
 - Entrega de evaluaciones

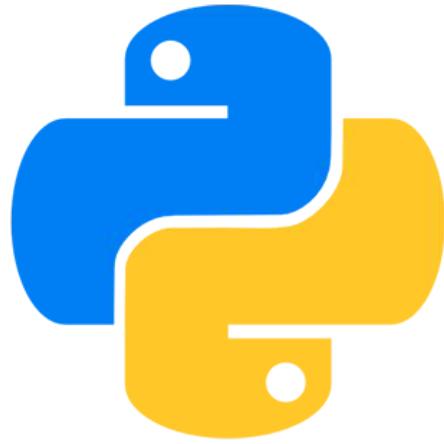
Herramientas del curso



<https://google.com/>



<https://stackoverflow.com/>



PYTHON

<https://www.python.org/>
<https://zen-of-python.info/>

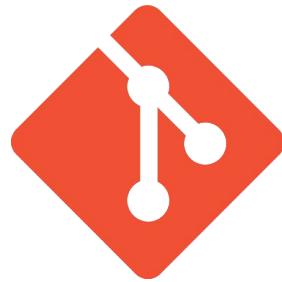
- Es el lenguaje de programación que utilizaremos en el curso para aprender los contenidos.
- Es de alto nivel, de propósito general y sumamente popular.

**Ocuparemos la versión
3.11.x. ($x \geq 7$)**



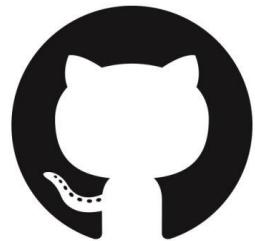
<https://www.jupyter.org/>

- Aplicación web que permite crear documentos de código interactivos.
- Es el formato de los apuntes del curso.
- Se recomienda bajar los apuntes e interactuar con el código, no solo leerlo desde la página.
- **NO DEBEN USARLO PARA
PROGRAMAR SUS ACTIVIDADES
NI TAREAS.**



git

<https://git-scm.com/>



GitHub

<https://github.com/>



Git

Git es un sistema distribuido de **control de versión**, gratuito y *open source*, diseñado para manejar de pequeños a enormes **proyectos** de forma rápida y eficiente.

- Permite revisar distintas versiones en cualquier momento.
- Permite controlar los cambios que se aplican sobre un proyecto.
- Permite programar en paralelo y luego juntar los resultados.
- Permite tener copias de apoyo de programas.
- Permite un trabajo en equipo mucho más fluido.

Se usa **masivamente en la vida real**,
incluso por **empresas** grandes.

Será el **medio de entrega oficial**
de las evaluaciones del curso.



- GitHub es una plataforma para alojar proyectos, usando el sistema de control de versiones Git.
- En febrero de 2024, GitHub tenía más de 100 millones de usuarios y 420 millones de repositorios albergados.¹
- Provee una interfaz web que permite visualizar y administrar proyectos controlados con Git.

1: <https://github.com/about>

Contenidos y metodología



Contenidos del curso

Fundamentos de programación

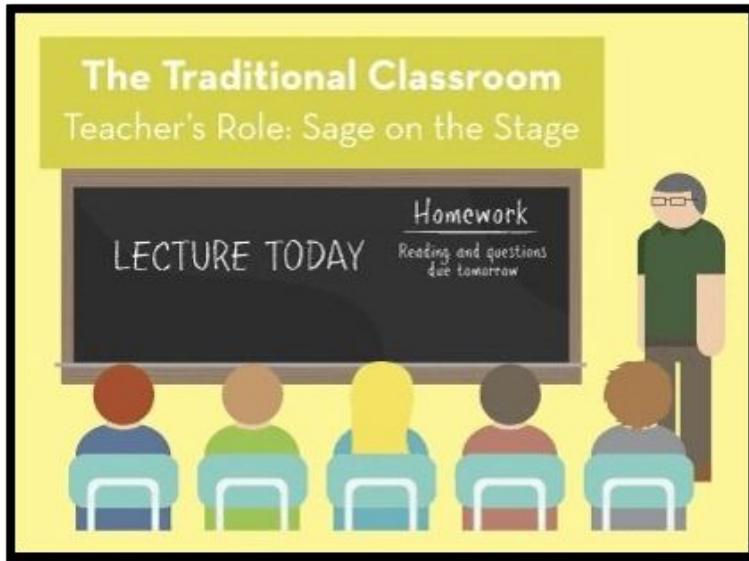
- Estructuras de datos *básicas*
- Programación orientada a objetos
- Iterables
- Programación funcional
- Manejo de errores

Herramientas de programación

- Técnicas básicas de mantenimiento de código
- *Threading*
- Interfaces gráficas
- I/O
- *Networking*
- Aplicaciones: *Web Services, API, expresiones regulares y más*



Metodología: *Flipped Classroom*





¿Cómo se refleja en el curso?

- Los contenidos y evaluaciones son preparados (y corregidos) por el equipo docente.
- Los estudiantes aprenden **haciendo**:
 - **Actividades** o **Experiencias (Control de Salida)** en el horario de cátedra.
 - **Tareas** a su propio ritmo.
 - **Evaluaciones Escritas**.

Break

Vayan al baño y estírense un poco

...Volvemos en unos instantes...

Actividades

Objetivo:

Poner en práctica un contenido estudiado mediante una actividad de programación breve.



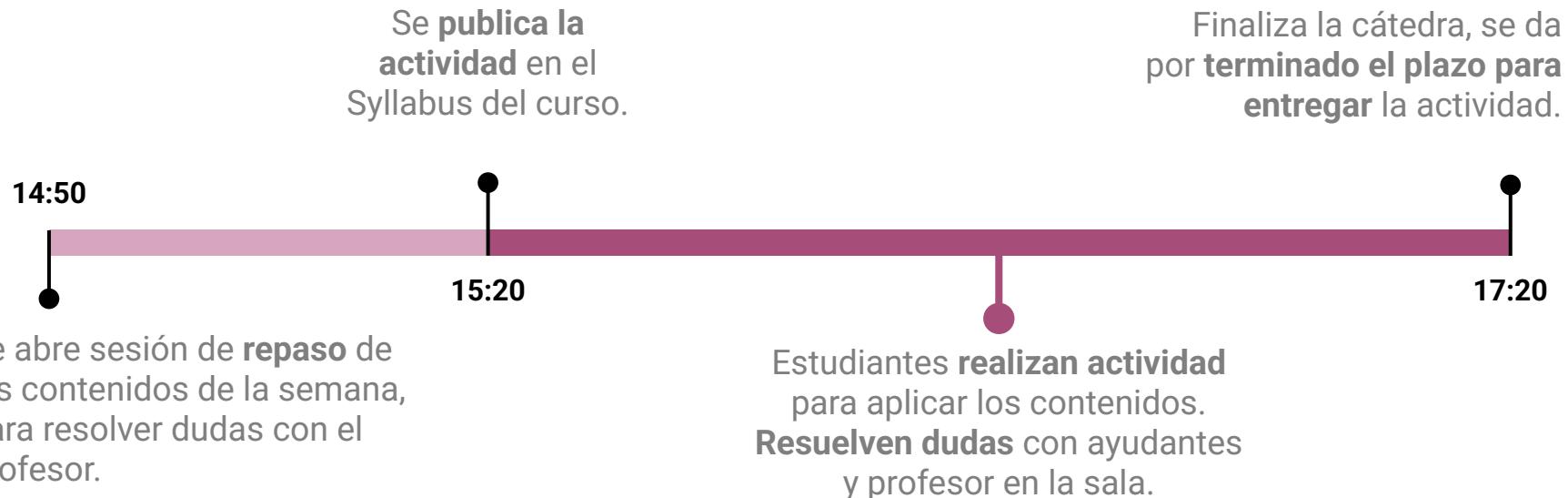
Actividades (AC)

- Tienen por objetivo **aplicar y evaluar** el aprendizaje de cada estudiante en un contenido esencial.
- Serán corregidas **automáticamente** y se proveerá eventualmente un *reporte* a **cada estudiante** con la respuesta de cada caso testeado automáticamente.
- Se proveerá de un **conjunto de tests junto al enunciado** para que puedan probar su actividad en tiempo real.
- Hay en total **9 actividades con asistencia** durante el semestre.
- Se asignará hasta un máximo de **4 puntos por actividad**.
- Habrá una Actividad 0, que no tendrá puntaje.
- **No se realizarán Actividades recuperativas.**



Actividades (AC)

Jueves correspondiente al contenido



Experiencias (Control de Salida)

Objetivo:
Poner en práctica un contenido estudiado mediante la resolución de un problema en conjunto.



Experiencias (CS)

- Tienen por objetivo conseguir que el estudiante **practique** el contenido de la semana.
- Durante la clase, se solucionará un **problema en conjunto** al profesor y el resto de los estudiantes.
- Se aplicará un **Control de Salida (CS)** durante los últimos 15 minutos de la clase. Estos serán realizados en Canvas.
- Hay **3 controles de salida con asistencia** durante el semestre.
- Se asignará hasta un máximo de **4 puntos por control**.
- **No se realizarán Controles de Salida recuperativos.**

Tareas

Objetivo:

Aplicar los contenidos estudiados resolviendo un problema complejo y de mayor extensión.



Tareas (T)

- Tienen por objetivo **evaluar** el aprendizaje **individual** de cada estudiante mediante la resolución de problemas complejos.
- La dificultad, extensión y formato de corrección variará según cada tarea.
- Las fechas están en el [programa](#).
- Hay en total **4** tareas durante el semestre:
 - o **T₁**: 8 días hábiles de extensión, corrección automática y manual.
 - o **T₂**: 10 días hábiles de extensión, corrección manual.
 - o **T₃**: 10 días hábiles de extensión, corrección automática y manual.
 - o **T₄**: 10 días hábiles de extensión, corrección manual.



Tareas (T)

Política de atraso

- Hasta 1 día de atraso
(00:01 a 24:00 hrs de atraso)
Nota máxima 6.0
- Hasta 2 días de atraso
(24:01 a 48:00 hrs de atraso)
Nota máxima 4.0
- Después de las 48 hrs,
no se aceptarán entregas atrasadas.

Cupones para disminuir penalización

- Durante el semestre cada estudiante **dispondrá de 2 cupones**.
- Cada cupón permite eliminar 1 día de penalización de nota en la tarea recién entregada.

Para ambos casos (usar o no cupón) deberán **llenar un formulario** indicando que desean optar por una entrega atrasada y si usarán 0, 1 o 2 cupones para dicha entrega.



Tareas (T)

Ejemplos política de atraso y uso de cupones

1) Entregó 1 minuto tarde.

Se considera como 1 día de atraso. Nota máxima 6.0.

Usa 1 cupón = Elimina el atraso de 1 día → Opta a la nota máxima 7.0.

2) Entregó 47 horas tarde.

Se considera como 2 días de atraso. Nota máxima: 4.0

Usa 1 cupón = Elimina el atraso de 1 día → Opta a la nota máxima 6.0.

Usa 2 cupones = Elimina el atraso de 2 días → Opta a la nota máxima 7.0.



Tareas (T)

Ejemplos política de atraso y uso de cupones

- 3) Entregó después de 48 horas de atraso.

Se considera como evaluación no entregada.

No se acepta esa entrega aunque se tengan 2 cupones.

IMPORTANTE

1. Los cupones no extienden la entrega, solo eliminan la penalización por entregar atrasado.
2. La política de atraso solo aplica a la entrega original de cada evaluación.

Evaluaciones Escritas

Objetivo:
Comprobar que interiorizaron correctamente los contenidos básicos del curso.



Evaluaciones Escritas (EE)

- Tienen por objetivo **evaluar** el aprendizaje individual de cada estudiante en los contenidos básicos del curso:
 - Aplicar los contenidos en diversos problemas.
 - Identificar posibles errores de código y proponer mejoras.
- Serán desarrolladas con lápiz y papel y podrán contener preguntas de alternativas y/o de desarrollo.
- Hay en total **2** evaluaciones escritas durante el semestre:

- **Midterm:** 11 de abril a las 17:30
- **Examen:** 01 de julio a las 13:30



Evaluaciones Escritas (EE)

Importante

- La ausencia a una de estas evaluaciones implica obtener nota mínima (1.0) en dicha evaluación.
- Sólo puede justificarse la ausencia a una (1) evaluación escrita, y sólo se aceptarán justificaciones aceptadas por la unidad académica.

Recomendaciones

- Revisar el material de los semestres pasados ([2024-2](#), [2024-1](#) o [2023-2](#)).
- Vayan estudiando semana a semana los contenidos.



Fechas de evaluaciones

Ruta de Aprendizaje
(están todas las fechas)



¿Cómo sería una semana en Avanzada?

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
				Publicación Batch de contenidos*		
	Ayudantía de repaso		Repaso, realización Actividad o Experiencia (Control de Salida)			

* Los Batch de contenidos se publican los viernes en ciertas fechas específicas.



Nota presentación (NP) y nota final (NF)

- **C**: Nota de cátedra:

$$C = \text{MIN} [(AC + CS) / 36 \times 6 + 1; 7]$$

- **T**: Promedio ponderado de tareas:

$$T = (1 \times T_1 + 2 \times T_2 + 2 \times T_3 + 3 \times T_4) / 8$$

- **EE**: Promedio evaluaciones escritas:

$$EE = (4 \times \text{Midterm} + 6 \times \text{Examen}) / 10$$



Nota presentación (NP) y nota final (NF)

- **NP:** Suma del Promedio ponderado de Tareas y la Bonificación de Actividades:

$$NP = T \times 50\% + EE \times 30\% + C \times 20\%$$

$$NF = \begin{cases} \min(NP; 7.0) & \text{si } T \geq 3.95, EE \geq 3.95 \text{ y } NP \geq 3.95 \\ \min(NP; 3.9) & \text{e.o.c.} \end{cases}$$

!! Importante

Todas las evaluaciones (**AC**, **CS**, **T** y **EE**) tienen por objetivo **evaluar** el aprendizaje **individual** de cada estudiante mediante la resolución de problemas.

- El desarrollo de todas las evaluaciones es **100% individual**.
- No está autorizado el apoyo y/o conversación de este tipo de evaluaciones con personas que no sean del equipo docente.
- En caso de **detectar faltas a la integridad académica** producto de un apoyo que no sea del equipo docente, **todos los involucrados serán sancionados**.



Integridad académica

*“Cualquier situación de copia en alguna evaluación tendrá como **sanción un 1,1 final en el curso**. Esto sin perjuicio de sanciones posteriores que estén de acuerdo a la Política de Integridad Académica de la Escuela de Ingeniería y de la Universidad, que sean aplicables para el caso.”*

También aplica la política de integridad académica del Departamento de Ciencia de la Computación (DCC), disponible como anexo en el programa del curso.



Integridad académica

- Está permitido el uso de código encontrado en internet u otra fuente de información similar, **siempre y cuando fue publicado antes de la liberación del enunciado. Siempre deben indicar la fuente de dicho código.**
- El origen de dicho código **puede ser ajeno al curso, pero debe ser citado.**
- Si el código es relacionado al curso 2025-1, solo se permite si es material del equipo docente como: material del curso o ayudantías.
- El no seguir estas reglas, se considera como una **falta a la ética.**
- Todas las evaluaciones del curso se consideran estrictamente **individuales**, a menos de que se indique lo contrario.
- Compartir código para una evaluación o conversar respecto a cómo resolver una evaluación individual se considera como una **falta a la ética.**



Integridad académica

- Está permitido el **uso de código encontrado en internet** u otra fuente de información similar, **siempre y cuando haya sido publicado antes de la liberación del enunciado. Siempre deben indicar la fuente de dicho código.**
 - Su uso debe ser producto de un proceso de investigación, integración y adaptación.
 - Deben ser capaces de **explicar cada línea del código** extraído de internet. Además de poder entender y explicar **qué sucede cuando se modifica y/o elimina** una o más líneas de dicho código.
 - Estudiante que **no cumpla esto o no cite apropiadamente**, será penalizado según la política de integridad académica del curso.



Generadores de código: Explicación



El uso de herramientas generadoras de código se encuentra estrictamente prohibido y se considera como una **falta a la ética**.

¿Por qué?

- Frecuentemente entregan información errónea.
- Es difícil replicar sus respuestas.
- Si bien ustedes deciden el *prompt*, el código entregado no es de su autoría.
- No entrega las fuentes de dónde basa su respuesta.

Afecta negativamente el aprendizaje y el logro de los objetivos del curso.



Generadores de código: Sanciones



En caso de que el equipo docente **verifique el uso de una herramienta generadora de código** en algunas de las evaluaciones del curso, se aplicarán la siguiente sanción:



1.1 en promedio del ramo



En caso de que el estudiante realice un arrepentimiento temprano, esto será considerado al momento de definir la sanción.

185 posibles casos de falta
a la ética detectados y
125 casos confirmados
en los últimos
semestres.



Arrepentimiento temprano

¿Qué es?

Este es un espacio para que los estudiantes puedan confesar cualquier falta a la integridad académica realizada durante la evaluación.

¿Por qué es importante?

Implica que están siendo honestos y responsables con su proceso de aprendizaje.

¿Cómo hacerlo?

Una vez terminada una evaluación, tendrán una semana para contactarnos.

¿Hay algún beneficio?

Aquellos que opten por esta vía, se tendrá especial consideración al momento de evaluar el caso.



Encuesta de Carga Académica (ECA)

Este curso habilitará la **Encuesta de Carga Académica** durante el semestre para medir el tiempo invertido en los diferentes ejes del curso: estudio autónomo, tareas, entre otros.

Si para el final del semestre:

1. El estudiante contesta 10 ECA o más.
2. Cumple todos los requisitos de aprobación (**T** \geq 3.95, **EE** \geq 3.95 y **NP** \geq 3.95).

Se asignará 1 décima a la Nota Final (**NF**) del curso.



Consultas en el curso

- Consultas administrativas: coordinacion.iic2233@gmail.com
- Consultas sobre los contenidos del curso, enunciados y pautas:
foro del curso, alojado en <https://github.com/IIC2233/syllabus/issues>
- Consultas personales: **correo al profesor** (correos en la [Wiki del curso](#))
- Problemas personales: **correo al ayudante de bienestar** (correos en la [Wiki del curso](#))



No usar canvas para enviar preguntas. Use las *issues* o el correo.

 **No usar canvas para enviar preguntas.** 

Use las *issues* o el correo.

 **No usar canvas para enviar preguntas.** 

Use las *issues* o el correo.

 **No usar canvas para enviar preguntas.** 

Use las *issues* o el correo.

 **No usar canvas para enviar preguntas.** 

Use las *issues* o el correo.

 **No usar canvas para enviar preguntas.** 

Use las *issues* o el correo.

 **No usar canvas para enviar preguntas.** 

Use las *issues* o el correo.

 **No usar canvas para enviar preguntas.** 

Recomendaciones



Recomendaciones

¡Aprovechen!

Aprovechen el **curso** para aprender lo más posible en herramientas que les servirán en el futuro.

Aprovechen a sus **profesores**, sus **ayudantes** y a sus **compañeros**.



Recomendaciones

¡Pidan apoyo si lo necesitan!

Es posible evaluar flexibilidades en las evaluaciones para casos de fuerza mayor contactando al ayudante de bienestar asignado a su sección.

OJO: La carga académica de cada estudiante **no es un motivo** de fuerza mayor.

Si les es más cómodo, también pueden contactar con algún profesor del curso, en caso de requerir apoyo de cualquier tipo.



Recomendaciones

- Leer el enunciado apenas lo entreguen, para empezar a programar lo antes posible.
- Buscar en Google (pueden usarlo en actividades y en tareas).
- Estudiar e interactuar con el material de clases y otros materiales.
- Asistir a las ayudantías.
- Estudiar el ramo en serio desde el principio.
- Ser estratégico con las tareas.
- Dedicarle tiempo a otros ramos.
- Dormir.



Tarea para la casa

1. Crear cuenta en GitHub.
2. Responder formulario: <https://forms.gle/43RgrXVxuXD6Hvw68>
3. Instalar *software obligatorio* (Python, git, Jupyter Notebook y una IDE).
4. Estudiar los primeros contenidos:
 - Semana 1: Repaso contenidos de Introducción a la programación y complementarios (ya está subido).
 - Semana 2: Entorno de trabajo (ya está subido).

✨ (Bonus) Grupos de Telegram ✨

1. El DCC cuenta con diferentes grupos para sus cursos, para comunicar eventos, proyectos e incluso oportunidades de empleo, por lo que queremos invitarlos a que se sumen a esta comunidad.
2. Aquí además podrán encontrar un grupo de Telegram específico de este ramo para que puedan resolver sus dudas y comunicarse entre ustedes.



<https://ccc.ing.puc.cl/>

Programación Avanzada

IIC2233 2025-1

Pablo Araneda - Daniela Concha - Francisca Ibarra - Lucas Van Sint Jan - Francisca Cattan