

Programación Avanzada

IIC2233 2024-1

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Dante Pinto - Francisca Cattan



Agenda

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

Motivación



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



Vamos por partes

- Pedir y mostrar datos de perfil de usuario.
- Revisar historial de viajes.
- Definir camino entre dos direcciones.
- Interfaz gráfica.
- Comunicación con conductor.

x

Edit Account



First Name

Cristian



Last Name

Ruz



Phone Number

Verified



Email



x

Your Trips

Past

Upcoming

Family

9/16/18, 22:43

\$2,287

Peugeot 301

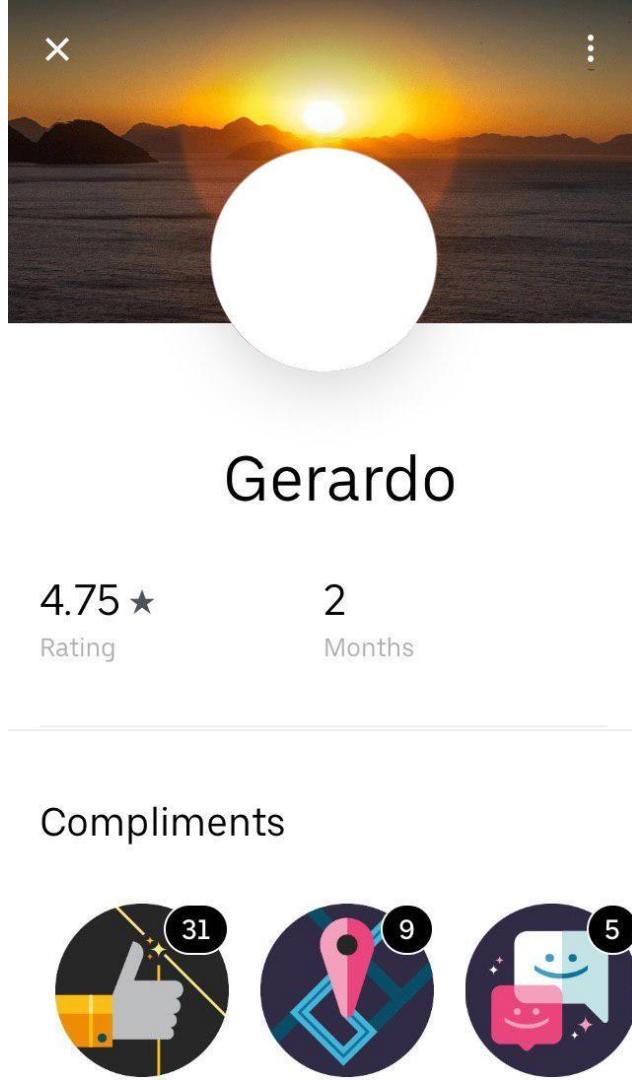
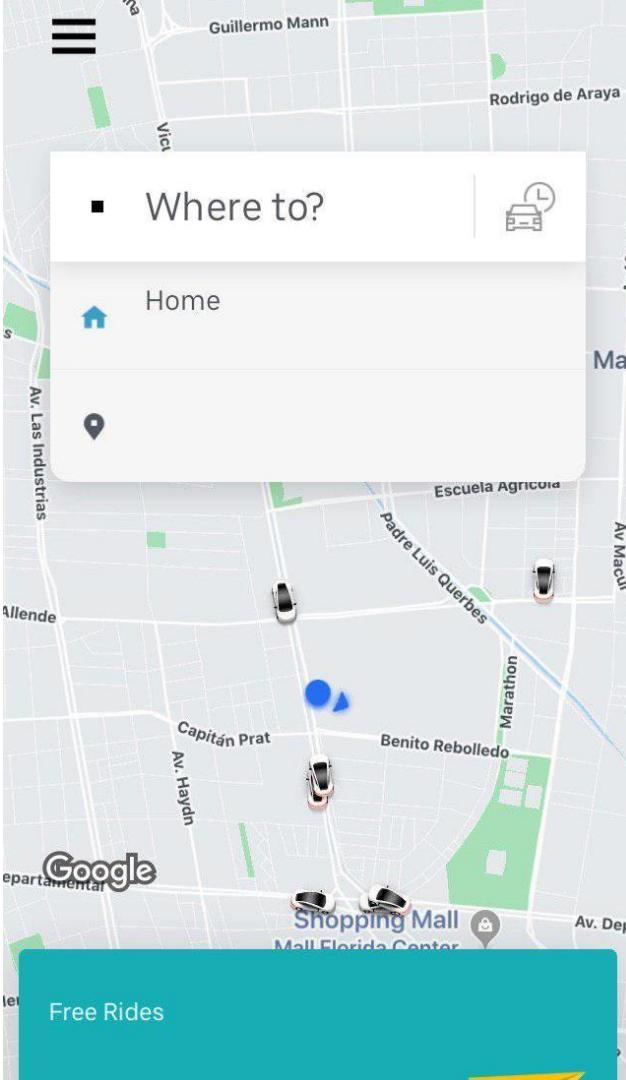


9/1/18, 19:30

\$6,083

Toyota Corolla





Equipo docente



Profesores



Sección 1
Hernán



Sección 2
Daniela



Sección 3
~~Francisca~~
gatochico



Sección 4
Dante



Sección 5
~~Francisca~~
Paqui



Ayudantes jefes



Referencia: "Star Wars: episodio II - el ataque de los clones"



Ayudantes jefes verdaderos

Cris
Coordinador
General

Pía
Jefa
Tareas

Nati
Coordinadora
Bienestar

Gery
Coordinadora
Tareas

Cata
Coordinadora
Docencia





Cuerpo de ayudantes (52 😱)



Programa y metodología

Plataformas

<https://cursos.canvas.uc.cl>

Canvas

Envío de anuncios

<https://github.com/IIC2233/Syllabus>

Syllabus del curso

Material de cátedra y enunciado evaluaciones
Foro donde se responderán todas las dudas

<https://github.com/IIC2233/Contenidos>

Contenidos

Material para estudio autónomo

<https://github.com/IIC2233/Syllabus/wiki>

Wiki del curso

Guía de instalación, reglas del foro, etc.

Contenidos y metodología



Contenidos del curso

Fundamentos de programación

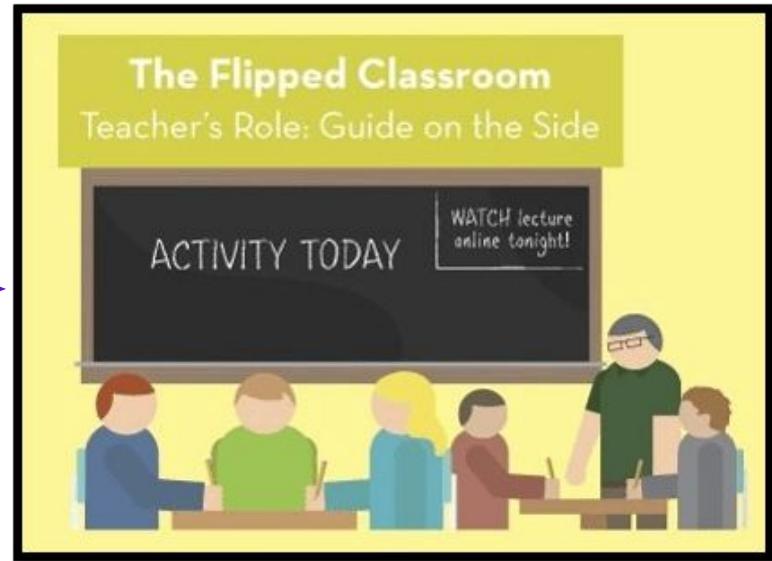
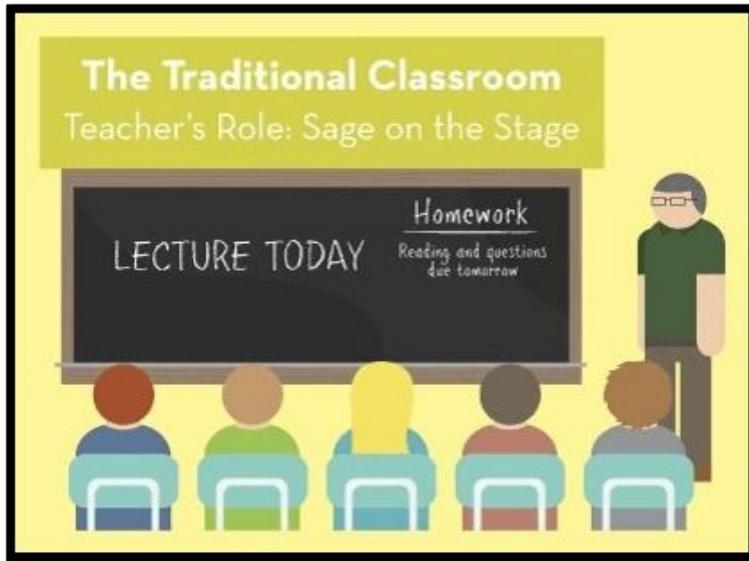
- Estructuras de datos *básicas*
- Programación orientada a objetos
- Iterables y programación funcional
- Manejo de excepciones

Herramientas de programación

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



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**:
 - Ejercicios por cuenta propia.
 - **Actividades** o **Experiencias** en el horario de cátedra.
 - **Tareas** a su propio ritmo.
 - **Evaluaciones escritas**.

Actividades

Objetivo:

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



Actividades

- Tienen por objetivo **aplicar y evaluar** el aprendizaje de cada estudiante en un contenido esencial.
- Serán corregidas **automáticamente** y se proveerá de un *reporte* a **cada estudiante** con la respuesta de cada caso testeado automáticamente.
- Se proveerá de un **conjunto de test junto al enunciado** para que puedan probar su actividad en tiempo real.
- Hay en total **5** actividades durante el semestre.
- Se asignará hasta un máximo de **4 puntos por actividad**.
- Solo se responderán dudas de **forma presencial** en el **horario de cátedra**.



Actividades

Jueves

Se abre sesión de **repaso** de los contenidos de la semana, para resolver dudas con el profesor.



14:50



Finaliza la **cátedra**, se da por terminado el espacio para resolver dudas.



17:20

Estudiantes realizan **actividad** para aplicar los contenidos y resuelven dudas con ayudantes y profesor en la sala.

Prox. Lunes

20:00



Finaliza el plazo para entregar la actividad.

Experiencias

Objetivo:
Poner en práctica
contenidos estudiados en
un espacio no evaluado.



Experiencias

- 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.
- Hay **4** en total durante el semestre.
- No tiene incidencia en el cálculo de nota del curso.

Tareas

Objetivo:

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



Tareas

- 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**: 7 días hábiles de extensión, corrección automática y manual.
 - o **T₂**: 11 días hábiles de extensión, corrección manual.
 - o **T₃**: 10 días hábiles de extensión, corrección automática.
 - o **T₄**: 10 días hábiles de extensión, corrección manual.



Tareas

- Tienen por objetivo **evaluar** el aprendizaje individual de cada estudiante mediante la resolución de problemas complejos.
 - El desarrollo de las tareas es **100% individual**.
 - No está autorizado el apoyo y/o conversación de este tipo de evaluación 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**.



Tareas

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 eliminar días de atraso

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

Para ambos casos deberán **llenar un formulario** indicando que desean optar por una entrega atrasada y si usarán 0, 1 o 2 cupones para dicha entrega.



Tareas

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 cupon = 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 día → Opta a la nota máxima 7.0.



Tareas

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

- 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** y **Examen**.



Evaluaciones escritas

Importante

- La ausencia a una de estas evaluaciones implica obtener nota mínima (1.0) en dicha evaluación.
- Solo se aceptarán justificaciones aceptadas por la unidad académica ante la ausencia de una evaluación.

Recomendaciones

- Revisar el material del semestre pasado. Están ambos enunciados, su solución y unos compilados con más preguntas para estudiar.
- 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		Publicación Actividad o realización Experiencia			
Entrega Actividad						

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



Nota del curso (NC) y nota final (NF)

- **AC**: Promedio actividades:

$$\mathbf{AC} = \text{MIN}[(\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3 + \mathbf{A}_4 + \mathbf{A}_5) / 12 \times 6 + 1, 7]$$

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

$$\mathbf{T} = (1 \times \mathbf{T}_1 + 2 \times \mathbf{T}_2 + 2 \times \mathbf{T}_3 + 3 \times \mathbf{T}_4) / 8$$

- **EE**: Promedio evaluaciones escritas:

$$\mathbf{EE} = (\mathbf{Midterm} + \mathbf{Examen}) / 2$$



Nota del curso (NC) y nota final (NF)

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

$$\mathbf{NC} = \mathbf{AC} \times 10\% + \mathbf{T} \times 60\% + \mathbf{EE} \times 30\%$$

$$\mathbf{NF} = \begin{cases} \min(\mathbf{NC}; 7.0) & \text{si } \mathbf{T} \geq 3.95, \mathbf{EE} \geq 3.50 \text{ y } \mathbf{NC} \geq 3.95 \\ \min(\mathbf{NC}; 3.9) & \text{e.o.c.} \end{cases}$$



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.**
- Si el código es relacionado al curso, 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 **individuales**, a menos de que se indique lo contrario.
- Compartir código para una evaluación o conversar respecto al desarrollo de una evaluación individual se considera como una **falta a la ética.**



Generadores de código

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

Investigación realizada este año en la Universidad de Texas A&M donde se encuestaron a profesores y más de 400 estudiantes.

¿Cómo crees que [competencia] se afectará?

Porcentaje de estudiantes que cree que la siguiente competencia se afectará “algo negativamente” o “bastante negativamente”:

- 43% pensamiento crítico
- 40% resolución de problemas
- 33% motivación intrínseca
- 32% compromiso (engagement)
- 31% trabajo en equipo
- 31% autoeficacia

150 posibles casos de falta a la ética detectados en los últimos semestres.

108 casos confirmados.



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 y definir la sanción correspondiente.

Break

Vayan al baño y estírense un poco

...Volvemos con más temas del curso en unos instantes...



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 3.50 y **NC** \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, **correos de profesores:**

Sección 1

Hernán Valdivieso
hvaldivieso@uc.cl

Sección 2

Daniela Concha
daconcha@uc.cl

Sección 3

Francisca Ibarra
faibarra1@uc.cl

Sección 4

Dante Pinto
drpinto@uc.cl

Sección 5

Francisca Cattan
fpcattan@uc.cl



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.** 



Aoyo en catedra y ayudantía

Ayudantes encargados de realizar las ayudantías y acompañarlos en las actividades

Coordinadora de Docencia

Catalina Miranda

Sección 1

Julián García

Sección 2

Clemente Campos

Sección 3

Diego Toledo

Sección 4

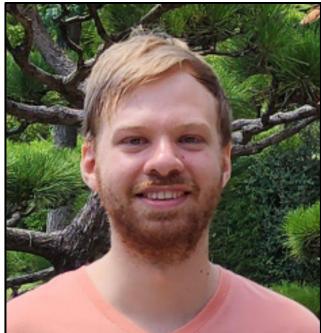
Julio Huerta

Sección 5

Carlos Olguín



Apoyo en cátedra y ayudantía



Sección 1
Julián



Sección 2
Clemente



Sección 3
Diego



Sección 4
Julio



Sección 5
Carlos



Apoyo bienestar

Solicitudes de apoyo ante problemas personales

Coordinadora de Bienestar

Nati Correal

bienestar.iic2233@gmail.com

Sección 1

Teresita López

bienestar.iic2233+1@gmail.com

Sección 4

Maria José Millán

bienestar.iic2233+4@gmail.com

Sección 2

Patricio Hinostroza

bienestar.iic2233+2@gmail.com

Sección 5

Natalia Correal

bienestar.iic2233+5@gmail.com

Sección 3

Amanda Sandoval

bienestar.iic2233+3@gmail.com



Apoyo bienestar



Sección 1
Tere



Sección 2
Pato



Sección 3
Amanda



Sección 4
Mari



Sección 5
Nati

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.

Herramientas del curso



<https://google.com/>

<https://stackoverflow.com/>



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

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



Python

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

Este semestre **ocuparemos la versión 3.11.x. (x>=7)**

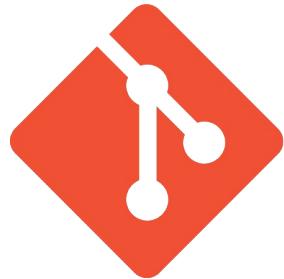


<https://www.jupyter.org/>

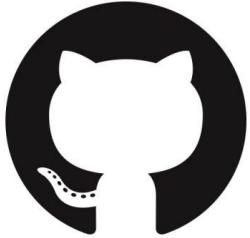


Jupyter Notebook

- Es una aplicación web que permite crear documentos interactivos con código, gráficos y texto explicativo.
- 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 +



GitHub

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

<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>

Ruta de Aprendizaje

Enlace



Tarea para la casa

1. Crear cuenta en GitHub.
2. Responder formulario: <https://forms.gle/YvYDBoBzjvjEGrF96>
3. Instalar *software obligatorio* (Python, git, Jupyter Notebook y una IDE).
4. Estudiar los primeros contenidos:
 - Semana 0: Repaso contenidos de Introducción a la programación y complementarios (ya está subido).
 - Semana 1: Entorno de trabajo (se sube mañana).

✨ (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 2024-1

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Dante Pinto - Francisca Cattan