

# ***Programación Avanzada***

## **IIC2233 2019-2**

Cristian Ruz - Fernando Florenzano – Antonio Ossa – Vicente Domínguez



# Agenda

- Motivación
- Equipo docente
- Programa y metodología del curso
- Recomendaciones
- Herramientas del curso
- Actividad de Git y 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.



## Edit Account



First Name

Fernando



Last Name

Florenzano



Phone Number

Verified >

Email



## Your Trips

Past

Upcoming

Family

9/16/18, 22:43

\$2,287

Peugeot 301



9/1/18, 19:30


\$6,083

Toyota Corolla



×

Edit Account



First Name

Fernando

>

Last Name

Florenzano

>

Phone Number

Verified >

Email

>

```
class Usuario:
```

```
def __init__(self, f, l, p, e):
    self.first_name = f
    self.last_name = l
    self.phone_number = p
    self.email = e
    self.trips = []
    self.picture = None
```

```
def add_trip(self, t):
    self.trips.append(t)
```

```
def edit_picture(self, p)
    self.picture = p
```

```
u = Usuario("Fernando", "Florenzano", "123", "f@uc.cl")
v1 = Viaje("9/16/18, 22:43", "2287", A, B, driver1)
v2 = Viaje("9/1/18, 19:30", "6083", C, D, driver2)
u.add_trip(v1)
u.add_trip(v2)
```

×

Your Trips

Past


Upcoming

Family

9/16/18, 22:43

\$2,287

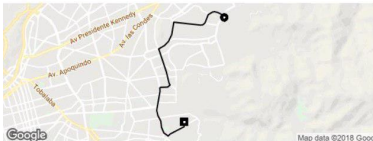
Peugeot 301



9/1/18, 19:30

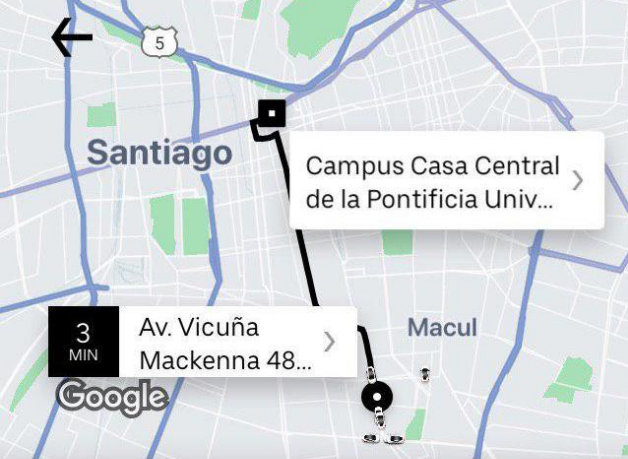
\$6,083

Toyota Corolla



```
class Viaje:
```

```
def __init__(self, d, p, s, e, dr):
    self.date = d
    self.price = p
    self.start = s
    self.end = e
    self.driver = dr
```



## Popular

Affordable, everyday rides



UberX

CLP4,664 ⓘ



Black

CLP9,829

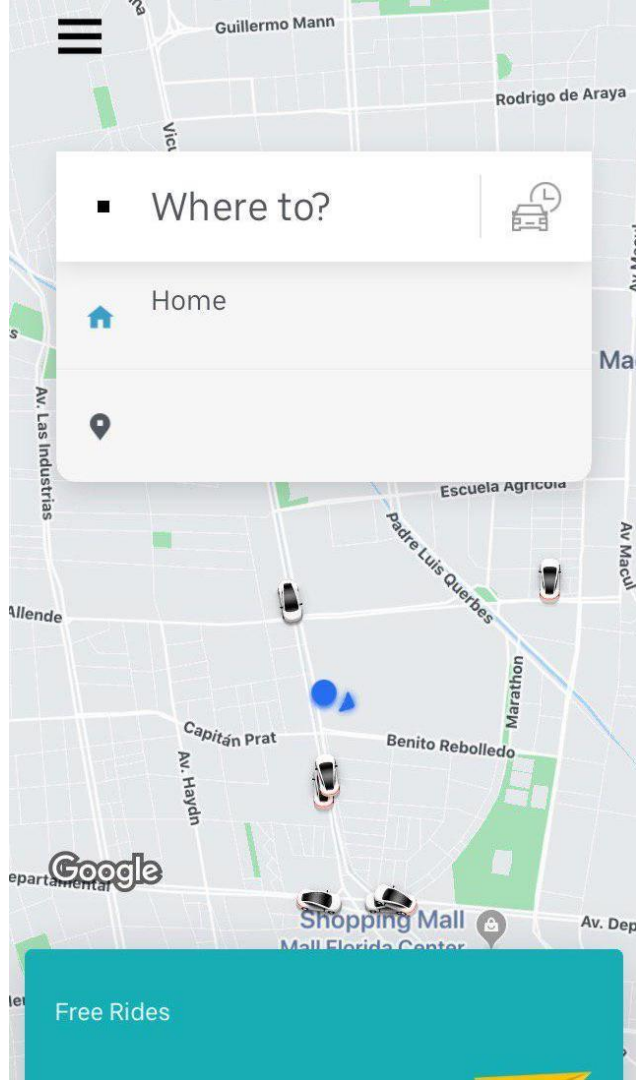


\*\*\*\*



1-4

CONFIRM UBERX



Where to?



Home



Gerardo

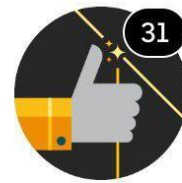
4.75 ★

Rating

2

Months

Compliments





# ¿Qué podemos hacer?

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



**Equipo docente**

# Profesores



Sección 1  
Cristian



Sección 2  
Fernando



Sección 3  
Antonio



Sección 4  
Vicente

# Ayudantes jefes

Área Docencia  
Dante



Área Tareas  
Daniela

Coordinador  
Enzo

# Cuerpo de ayudantes



# Cuerpo de ayudantes

## Full Tareas

Juan Schuwirth (@jeschuwirth)  
Diego Collao (@DCCollao)  
Francisca Ibarra (@Gatochico)  
José Manuel Wielandt (@jmwielandt)  
Rodrigo Alonso (@rialonsom)  
Catalina Álvarez (@cfalvarez2)  
Tomas Cantergiani (@tomcan127)  
Antonio Jara Rozas (@aljara97)  
Rosita Muñoz (@RositaMunoz)  
Jorge Ignacio Toro (@jitorcas)  
Ana Maria Undurraga (@Aundurraga)  
Francisco Urzúa (@furzuadr)  
Ariela Valenzuela (@ariela1)  
Jessica Hormazabal (@Arcoirisky)

## Híbridos Tareas

Raúl Andrés Álvarez (@KnowYourselfes)  
Juan Echavarri (@Jcechavarri)  
Juan Romero (@jaromero6)  
Maria Jose Varela (@Mjvarela)  
Daniela Andrea Poblete (@Dnpoblete)  
Josefina Fernández (@jfernandez7)  
Javiera Irrarrázabal (@jirarrazabal)  
Tomás Andrés González (@tomasgv)  
Roberto Vidal Brunet (@robertov1)  
Bartolome Peirano (@bpeirano1)  
Susana Figueroa (@sufigueroa)

## Híbridos Docencia

Roberto Negrin (@roberto009)  
Mario Reinike (@marioreinike)  
Ian Basly Stuardo (@igbasly)  
Javiera Ignacia Ochoa (@javieraocchoa)  
Nicolás Orellana (@Nhorellana)  
Maximiliano Agustín Narea (@manarea)  
Nicolás Benítez Romero (@nabenitez)

## Full Docencia

Benjamín Martínez (@bimartinez)  
Juan José Aguillón (@jjaguillon)  
Matías Oportus (@matiopts)  
Cauã Terra Santiago Paz (@csantiagopaz)  
Brian Murtagh Fontanet (@bmurtagh01)  
Pablo Hernán Araneda (@StroveLight)  
Joaquín Couyoumdjian (@JoacoCoyu)

# Programa y metodología

# La “fama” del curso

- Consume mucho tiempo
- Muchos contenidos
- Consume mucho tiempo
- Difícil
- Consume mucho tiempo



# La “fama” del curso

- Consume mucho tiempo
- Muchos contenidos
- Consume mucho tiempo
- Difícil
- Consume mucho tiempo
- Se aprende mucho
- Prepara para el futuro
- Se aprende mucho
- Otorga autonomía
- Se aprende mucho

# La nueva versión 2019-2

Durante diciembre 2018 y enero 2019 se reformularon los contenidos y competencias que el curso quiere que sus alumnos aprendan.

Con nuestra experiencia del semestre pasado, hicimos algunos ajustes para que su aprendizaje fuera más efectivo.

Y este es el resultado...

**<https://iic2233.github.io>**

**Página del curso**

*“A lo largo del este curso, el **alumno** desarrollará técnicas para diseñar, implementar, ejecutar y evaluar herramientas de software que resuelven problemas algorítmicos a partir de especificaciones detalladas.”*

# Contenidos del curso

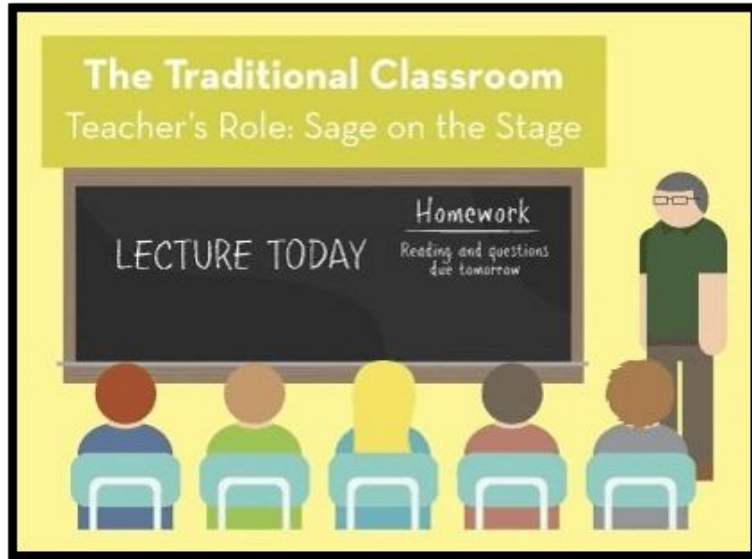
## Fundamentos de programación

- Estructuras de datos *built-ins*
- Programación orientada a objetos
- Iterables y funciones de orden superior
- Manejo de excepciones
- Estructuras de datos con nodos

## Herramientas y patrones de programación

- *Threading*
- Interfaces gráficas
- I/O y serialización
- *Networking*

# Metodología: *Flipped Classroom*



# ¿Cómo se refleja en el curso?

- Los contenidos y evaluaciones son preparados (y corregidos) por el equipo docente
- Los alumnos aprenden **haciendo**:

# ¿Cómo se refleja en el curso?

- Los contenidos y evaluaciones son preparados (y corregidos) por el equipo docente
- Los alumnos aprenden:
  - haciendo **actividades** en clase
  - haciendo **tareas** en casa
  - y nada más.



# ¿Cómo se refleja en el curso?

- Los contenidos y evaluaciones son preparados (y corregidos) por el equipo docente
- Los alumnos aprenden:
  - haciendo **actividades** en clase
  - haciendo **tareas** en casa
  - y nada más.
- La planificación se asegura que cada semana hay **a lo más una** evaluación con nota.

# Actividades

## **Objetivo:**

poner en práctica un  
contenido estudiado  
mediante una actividad  
de programación breve

---

# Actividades

Lu	Ma	Mi	Ju	Vi	Sa	Do

# Actividades

Lu	Ma	Mi	Ju	Vi	Sa	Do

**Viernes:** Los profesores **ya habrán subido el material** de la semana para que los alumnos **estudien** y resuelvan dudas en el [foro del curso](#).

# Actividades

Lu	Ma	Mi	Ju	Vi	Sa	Do

**Martes:** Habrá una ayudantía para **reforzar** los contenidos y para resolver dudas de forma presencial con los ayudantes.

# Actividades

Lu	Ma	Mi	Ju	Vi	Sa	Do

**Miércoles:** Se subirán **puestos asignados** de la sala al [Syllabus](#), para la clase del siguiente día.

# Actividades

Lu	Ma	Mi	Ju	Vi	Sa	Do

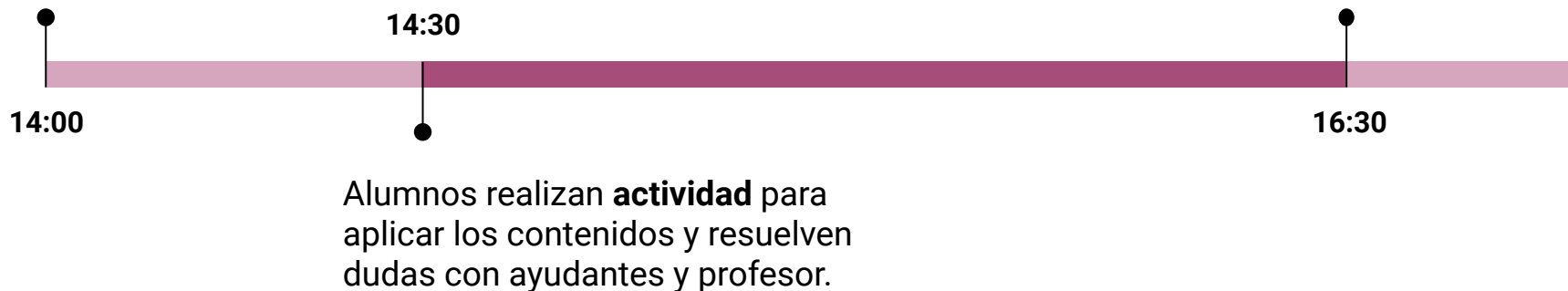
**Jueves:** ¡Día de actividad!

# Actividades

**Jueves:** ¡Día de actividad!

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

Las actividades realizadas son **entregadas** y el profesor realizará un **cierre** discutiendo solución y aprendizajes.





# Actividades

- Serán de los **contenidos de la semana**.
- Los ayudantes **pasarán lista**. Si están marcados como ausentes, su actividad no será corregida.
- Tendrán hasta las **16:30** para entregar su actividad.
- Hay dos tipos de actividades: **formativas y sumativas**.

# Actividades formativas

- Tienen por objetivo conseguir que el estudiante **practique** el contenido de la semana.
- **No** son parte del cálculo de nota del curso.
- Hay **6** en total durante el semestre.
- Serán corregidas **de forma colectiva** y se proveerá un *feedback* **general** detallando los resultados generales del curso.

# Actividades sumativas

- Tienen por objetivo **evaluar** el aprendizaje individual de cada estudiante en un contenido esencial.
- Obtendrán una **nota** por su resultado.
- Hay **4** en total durante el semestre, y se considerarán en el promedio solo las 3 mejores.
- Serán corregidas **individualmente** y se proveerá un *feedback* a **cada** alumno.

# Tareas

## **Objetivo:**

aplicar los contenidos  
estudiados resolviendo  
un problema complejo

---

# Tareas

Lu	Ma	Mi	Ju	Vi	Sa	Do

# Tareas

Lu	Ma	Mi	Ju	Vi	Sa	Do

**Sábado:** el enunciado de una tarea se **publica**.

# Tareas

Lu	Ma	Mi	Ju	Vi	Sa	Do

**Sábado (subsiguiente):** **termina** el plazo para realizar la tarea.

# Tareas

Lu	Ma	Mi	Ju	Vi	Sa	Do

**Domingo (después de fin de plazo):** se permite hasta **24 horas** de atraso, pero con descuento.



# Tareas

Lu	Ma	Mi	Ju	Vi	Sa	Do

**Lunes (después de fin de plazo):** se permite hasta **48 horas** para entregar el README, sin descuento.

# Tareas

Lu	Ma	Mi	Ju	Vi	Sa	Do

¿Y qué ocurre en las ayudantías y clase?

# Tareas

Lu	Ma	Mi	Ju	Vi	Sa	Do

**Clases:** se realizarán **Actividades Formativas** de contenido del curso

# Tareas

Lu	Ma	Mi	Ju	Vi	Sa	Do

**Ayudantías:** se realizarán **salas de ayuda** con los ayudantes o se revisarán **contenidos de la semana**.

# Participación

## **Objetivo:**

Incentivar el aprendizaje y premiar la constancia de trabajo

---

# Actividades formativas: auto-evaluación

- Tienen por objetivo conseguir que el alumno **reflexione** sobre su rendimiento durante la actividad formativa.
- Tras terminar una actividad formativa, una **rúbrica** será publicada.
- El alumno **puede** enviar su auto-evaluación siguiendo tal rúbrica.
- Por **cada** auto-evaluación **enviada** (de una actividad formativa **realizada**) el alumno podrá ganar **dos décimas**, si es que cumple con ciertos requisitos.
- Para obtener las décimas se pide: asistencia, entregar un mínimo de avance, y responder la autoevaluación.
- El **total** de décimas ganadas se sumarán a la nota de la **mejor actividad sumativa** del alumno.

# Tareas: entrega de avance

- Tienen por objetivo incentivar la **realización anticipada** de las tareas y poder **entregar un *feedback*** rápido a los alumnos de sus avances.
- Tras la publicación del enunciado de una tarea, tendrán un plazo de unos días para enviar un **avance de su tarea**.
- Los avances entregados serán corregidos de forma **colectiva** y un *feedback general* será publicado.
- El enviar un avance **satisfactorio** agrega dos décimas a la nota de la tarea.
- El contenido del avance, plazo de envío y condiciones de satisfacción se detallarán en cada tarea.

# Fechas de evaluaciones

**<https://iic2233.github.io/calendario/>**

**(La primera tarea se debería publicar el sábado pero es posible que la subamos el viernes, y tendrán hasta el próximo sábado)**



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

- **AC**: Promedio de las tres mejores actividades sumativas.
- **T**: Promedio ponderado de tareas:  $(T_0 + 2 \times T_1 + 4 \times T_2 + 5 \times T_3) / 12$

$$NC = 0,4 \times AC + 0,6 \times T$$

$$NF = \begin{cases} NC & \text{si } AC \geq 3,95 \text{ y } T \geq 3,95 \\ \min(NC; 3,9) & \text{e.o.c.} \end{cases}$$

# Aprobación

$$\mathbf{NF} = \begin{cases} \mathbf{NC} & \text{si } \mathbf{AC} \geq 3,95 \text{ y } \mathbf{T} \geq 3,95 \\ \min(\mathbf{NC}; 3,9) & \text{e.o.c.} \end{cases}$$

El alumno **aprobará** si **NF** es mayor o igual a **3,95**.

# Evaluación de última instancia

En el caso extremo que **NC** sea mayor a 3,95, pero **AC** o **T** se encuentren entre **3,50 y 3,94**, el alumno tendrá opción de apelar y rendir una **última evaluación oral** para aprobar el curso.

Esta evaluación se realizará el **9 de diciembre** y busca identificar si el alumno logró desarrollar las competencias del curso. Será interrogado en varios contenidos del curso.

El aprobar esta evaluación permitirá que su nota final se calcule como: **NF = NC**

# Detalles sobre notas

- La **inasistencia** a una actividad se interpreta como no realizada, tanto para **formativas como sumativas**.
- No hay Actividad Recuperativa.
- **No** habrá reemplazos de notas de **tareas** ni de **actividades**.
- Todas las notas del curso se calculan con **dos decimales**, excepto **NF** que se calcula con **un decimal**.

# Correcciones y recorreciones

- Las notas de una evaluación se publicarán a **más tardar 15 días hábiles** después de haberse realizado.
- Tras la publicación de una nota, tendrán **una semana** para enviar a corregir mediante un formulario a publicar.
- La recorreción es una instancia de aprendizaje para **reflexionar sobre su desempeño**, no una oportunidad de subir nota.
- Se esperan solicitudes **bien fundamentadas y poco ambiguas, que ayude a entender** a la persona que le revise su recorreción.

# Recorrecciones extremas

- De no estar de acuerdo con la respuesta de una corrección, se puede solicitar una revisión por parte de los profesores a través de un formulario.
- Puede que tome tiempo, ya que implica la revisión completa de la evaluación.
- Es la última instancia de apelación.

# 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, pero **debe indicarse 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 **plagio**.
- Todas las evaluaciones del curso se consideran **individuales**, a menos de que se indique lo contrario.
- Luego, compartir código para una evaluación se considera copia.



# Consultas

- Consultas administrativas: **iic2233@ing.uc.cl**
- 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 o ayudantes**
  - Cristian: [cruz@ing.puc.cl](mailto:cruz@ing.puc.cl)
  - Fernando: [faflorenzano@ing.puc.cl](mailto:faflorenzano@ing.puc.cl)
  - Antonio: [aaossa@ing.puc.cl](mailto:aaossa@ing.puc.cl)
  - Vicente: [vidominguez@uc.cl](mailto:vidominguez@uc.cl)

# Recomendaciones

# Recomendaciones

- Leer el enunciado apenas lo entreguen, para empezar a programar lo antes posible.
- Buscar más en Google.
- Estudiar e interactuar con el material de clases.
- Ir a las ayudantías.
- Estudiar el ramo en serio desde el principio.
- Ser estratégico con las tareas.
- Dedicarle tiempo a otros ramos.
- Dormir

# 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** (en el buen sentido).

# Herramientas del curso



<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 3.7.x.**

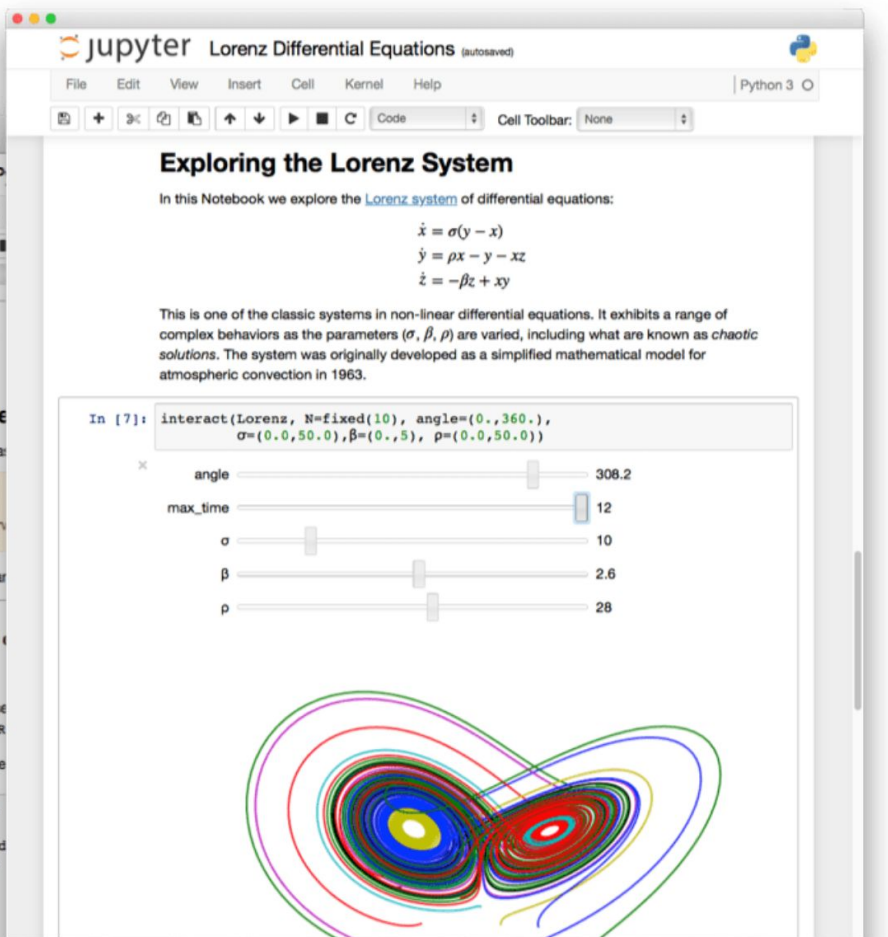
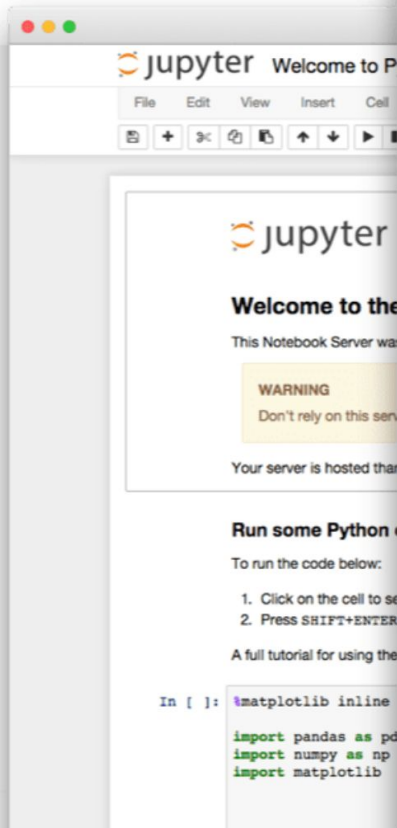


<https://www.jupyter.org/>



# 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.**
- Instalación: <https://jupyter.org/install.html>



Google +  stackoverflow

<https://google.com/>

<https://stackoverflow.com/>

# ¿Cómo buscar soluciones?

**¡En ingles!**

python [versión] [librería] [duda]



¿Cómo imprimir una cola con Python?



Python 3.6 collections print queue



# ¿Cómo buscar soluciones?

¡En ingles!

python [versión] [error]



`NameError: name "variable" is not defined`



`NameError: name * is not defined`





python3.5 NameError: name \* is not defined



Todos

Imágenes

Maps

Imágenes

Noticias

Más

Preferencias

Herramientas

Google encontró 95,800 resultados (0.50 segundos)

[In Python3.5:NameError: name 'image\\_to\\_string' is not defined](#)

[https://stackoverflow.com/.../in-python3-5nameerror-name-image...](https://stackoverflow.com/.../in-python3-5nameerror-name-image-...) ▼ Traducir esta página

11 jun. 2017 - Please post your source code so we can look over the code and get more details. Also your error is caused by a variable declaration without a ...

[oop - Python3 NameError: name 'method' is not defined - Stack Overflow](#)

[https://stackoverflow.com/.../python3-nameerror-name-method-is...](https://stackoverflow.com/.../python3-nameerror-name-method-is-...) ▼ Traducir esta página

18 mar. 2016 - consider you have the function defined in the global scope: def recursive(x): if (x>5): print (x) recursive(x - 1). you would simply call this with ...

[input\(\) error - NameError: name '...' is not defined - Stack Overflow](#)

[https://stackoverflow.com/.../input-error-nameerror-name-is-not...](https://stackoverflow.com/.../input-error-nameerror-name-is-not-...) ▼ Traducir esta página

14 ene. 2014 - input\_variable = input ("Enter your name:") print ("your name is" + input\_variable) ... input ("Enter your name:") File "<string>", line 1, in <module> NameError: name 'dude' is not defined ... I did what Kevin said and it is version 2.7.5! ... If you are using Python 3.x, raw\_input has been renamed to input .

[python NameError: name 'file' is not defined in python 3.5 - Stack ...](#)

[https://stackoverflow.com/.../python-nameerror-name-file-is-not...](https://stackoverflow.com/.../python-nameerror-name-file-is-not-...) ▼ Traducir esta página

26 nov. 2015 - Traceback (most recent call last): File "c:\python3.5\lib\runpy.py", line .... python 3.x from ... Q: python NameError: name 'file' is not defined But ...

[python 3.x - NameError: name 'value' is not defined - Stack Overflow](#)

[https://stackoverflow.com/.../python3-x-nameerror-name-value-is-not-defined...](https://stackoverflow.com/.../python3-x-nameerror-name-value-is-not-defined-...) ▼ Traducir esta página

3 abr. 2014 - NameError: name 'value' is not defined ... A variable defined in a function isn't visible outside the function. ... answered Apr 5 '14 at 2:39.

[NameError: global name 'unicode' is not defined - in Python 3 - Stack ...](#)

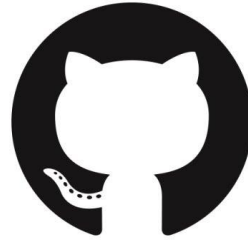
[https://stackoverflow.com/.../nameerror-global-name-unicode-is...](https://stackoverflow.com/.../nameerror-global-name-unicode-is-...) ▼ Traducir esta página

9 nov. 2013 - Python 3 renamed the unicode type to str , the old str type has been replaced by bytes . if isinstance(unicode or str, str): text = unicode or str ...



**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** gigantes.

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

# GitHub

- GitHub es una plataforma para alojar proyectos, usando el sistema de control de versiones Git.
- En abril de 2019, GitHub tenía 36 millones de usuarios y 100 millones de repositorios albergados.<sup>1</sup>
- Provee una interfaz web que permite visualizar y administrar proyectos controlados con Git.

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

***Break***

# Lo primero es lo primero

En este curso se aprende a programar programando, así que aprenderemos a usar `git` utilizandolo

- `git` nos permite mover archivos entre nuestro computador y GitHub
- ¡Aprendamos a usar el repositorio que hicieron los ayudantes!

---

# Antes de partir... ¿tenemos git?

Escribir en la consola (Git Bash cuenta) lo siguiente...

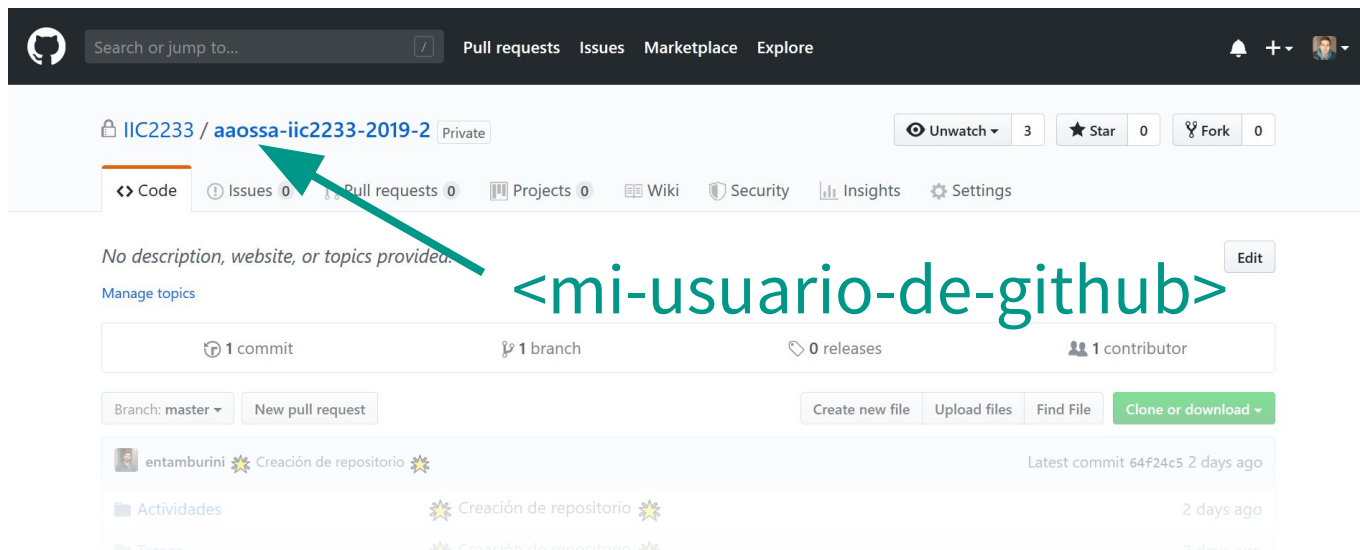
git

Si la consola les responde con la documentación del uso de git (algo como...  
usage: git [ - -version] ...), significa que tienen git instalado. Si no lo tienen  
instalado, las instrucciones están en [la wiki de la página oficial del curso](#).

# Antes de partir... ¿tenemos un repositorio?

[https://github.com/IIC2233/&ltmi-usuario-de-github>-iic2233-2019-2](https://github.com/IIC2233/<mi-usuario-de-github>-iic2233-2019-2)

Si respondieron el form que les enviamos en el primer anuncio, debería haberles llegado un **correo de GitHub**, con una invitación a un repositorio en la organización del curso en GitHub. Si no lo han hecho... ¡vayan a Siding y haganlo pronto!



# Llevando el repositorio a nuestro PC

Los ayudantes ya nos dieron nuestro repositorio. Si queremos bajar ese repositorio, hacer cambios y subirlos, tenemos que crear una copia local. Es hora de **clonar**.



# Llevando el repositorio a nuestro PC

Los ayudantes ya nos dieron nuestro repositorio. Si queremos bajar ese repositorio, hacer cambios y subirlos, tenemos que crear una copia local. Es hora de **clonar**.

```
git clone url_a_clonar
```

¡NO HAGAN NADA AÚN! Nos falta algo...

¿De dónde obtengo la URL para clonar mi repositorio?



# Llevando el repositorio a nuestro PC

The screenshot shows the GitHub interface for a repository named 'IIC2233 / aaossa-iic2233-2019-2'. The repository is private and has 3 watchers, 0 stars, and 0 forks. The 'Code' tab is selected, showing options for cloning or downloading. A green arrow points from the text 'url\_a\_clonar' to the URL 'https://github.com/IIC2233/aaossa-iic2233' in the 'Clone with HTTPS' dropdown menu.

Search or jump to... Pull requests Issues Marketplace Explore

IIC2233 / aaossa-iic2233-2019-2 Private

Unwatch 3 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/IIC2233/aaossa-iic2233

Open in Desktop Download ZIP

2 days ago

entamburini Creación de repositorio

Actividades Creación de repositorio

Tareas Creación de repositorio

.gitignore Creación de repositorio

README.md Creación de repositorio

README.md

# Llevando el repositorio a nuestro PC

Recuerden estar en la carpeta en la que quieren mantener el repositorio, ya que se creará una carpeta con los contenidos.

```
git clone url_a_clonar
```

Repositorio  
en GitHub



Repositorio  
local

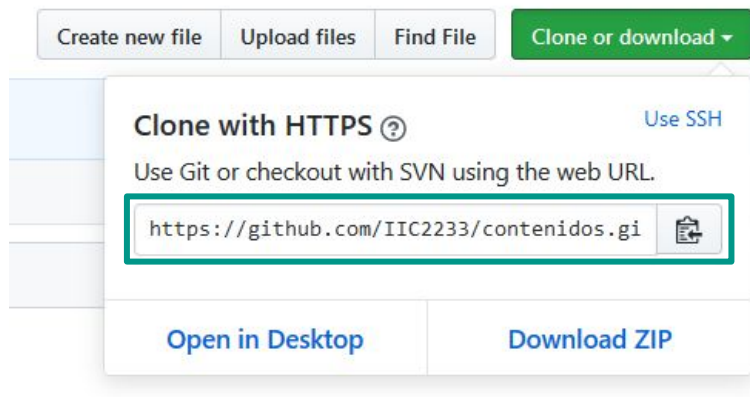
# Repositorios del curso

Ahora que saben clonar, aprovechemos de descargar los repositorios del curso: Syllabus (donde se suben enunciados) y contenidos (donde se sube material del curso).

```
git clone url_a_clonar
```

```
git clone https://github.com/IIC2233/syllabus.git
```

```
git clone https://github.com/IIC2233/contenidos.git
```



Syllabus



contenidos



aaossa-iic2233  
-2019-2

# Movernos entre carpetas

Dado que estamos trabajando en la consola, es necesario conocer al menos este comando, que nos permite cambiar de carpeta o directorio (*change directory*)

```
cd directorio_destino
```

Para ingresar a una carpeta basta con que escribamos su nombre en `directorio_destino`. Si queremos salir de esa carpeta, subiendo un nivel, tenemos que usar *punto-punto*:

```
cd ..
```

# Trabajemos con nuestro repositorio

Para poder ocupar los comandos de `git`, debemos estar dentro de un repositorio clonado, por lo que debemos movernos a la carpeta correspondiente.

```
git clone url_a_clonar  
cd <usuario>-iic2233-2019-2
```



Syllabus



contenidos



aaossa-iic2233  
-2019-2

# Desde ahora... `git status`

Usen `git status` muy seguido. Antes y después de hacer algo, ocupen `git status`. Esto les permitirá saber qué están haciendo, si les faltó un paso, y de hecho les sugerirá comandos si es que les falta algo por hacer.

`git status`

```
Antonio@Dell-XPS MINGW64 ~/Desktop/aaossa-iic2233-2019-2 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

# Editemos algo... README.md

Es un archivo para ser leído antes de la ejecución de cualquier código. En su repositorio hay un README para ustedes y **en cada tarea ustedes deben hacer otro README** para que los ayudantes puedan revisar de mejor forma su entrega.

```
README.md
1  # Repositorio de {Tu usuario de GitHub}
2
3  Este es el repositorio que utilizarás dur
4
5  **Asegúrate de seguir la estructura de la
6
7  Todo el proceso de recolección de tareas
8
9  ## Datos del alumno
10
11  | Nombre | Mail UC |
12  | :--: | :--: |
13  | {Tu nombre} | {Tu correo UC} |
14
```

```
README.md
1  # Repositorio de aaossa
2
3  Este es el repositorio que utilizarás dur
4
5  **Asegúrate de seguir la estructura de la
6
7  Todo el proceso de recolección de tareas
8
9  ## Datos del alumno
10
11  | Nombre | Mail UC |
12  | :--: | :--: |
13  | Antonio Ossa | aaossa@uc.cl |
14
```

# Hicimos un cambio... git status

Antes de hacer cualquier cosa... git status.

```
Antonio@Dell-XPS MINGW64 ~/Desktop/aaossa-iic2233-2019-2 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```



# Seleccionar cambios... git add (y status)

Hagámosle caso a `git status`, usemos `git add`. Con esto, definimos la lista de cambios que queremos declarar.

```
Antonio@Dell-XPS MINGW64 ~/Desktop/aaossa-iic2233-2019-2 (master)
$ git add README.md
```

```
Antonio@Dell-XPS MINGW64 ~/Desktop/aaossa-iic2233-2019-2 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

```
Changes to be committed:
  (use "git reset HEAD <file> ..." to unstage)
```

```
    modified:   README.md
```

# Declarar cambios... `git commit` (y `status`)

Para declarar los cambios que hemos realizado, y ponerles un mensaje describiendo los cambios, utilizamos `git commit -m "Mensaje"`.

```
Antonio@Dell-XPS MINGW64 ~/Desktop/aaossa-iic2233-2019-2 (master)
$ git commit -m "Agregué mis datos"
[master 319a840] Agregué mis datos
1 file changed, 2 insertions(+), 2 deletions(-)
```

```
Antonio@Dell-XPS MINGW64 ~/Desktop/aaossa-iic2233-2019-2 (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
```

```
nothing to commit, working tree clean
```

Los mensajes son **muy importantes**. Son una ayuda a ustedes en el futuro.

Revisen esta [guía de estilo](#).

# Subir cambios... git push (y status)

Para dejar en GitHub los cambios que hemos declarado, debemos hacer los que nos dice `git status` y hacer `git push`.

```
Antonio@Dell-XPS MINGW64 ~/Desktop/aaossa-iic2233-2019-2 (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 339 bytes | 169.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/IIC2233/aaossa-iic2233-2019-2.git
    64f24c5..319a840  master -> master
```

# Subir cambios... `git push` (y `status`)

Ahora, si hacemos `git status` nuevamente, nos damos cuenta de que tenemos lo mismo que GitHub y no tenemos cambios pendientes por declarar. Recuerden usar `git status` muy seguido.

```
Antonio@Dell-XPS MINGW64 ~/Desktop/aaossa-iic2233-2019-2 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

# El alumno/a que ocupa git status



**Siempre** hagan  
*commit y push*  
de su trabajo.

- Cada vez que avancen en algo importante de su actividad o tarea.
- Si llevan programando más de media hora.
- Cada vez que paren de programar para dedicarse a otra cosa.

---



**De verdad:**  
**Siempre hagan**  
***commit y push***  
**de su trabajo.**

- Tener su trabajo en GitHub es una copia de seguridad.
  - *Shit happens:*
    - Accidentes con líquidos.
    - Robos en Deportes.
    - Fallas de *hardware* o *software*.
    - Cortes de internet.
    - Echar a perder la tarea.
    - Y muchas otras cosas.
-



# Actividad en clase 0 (ACoo)

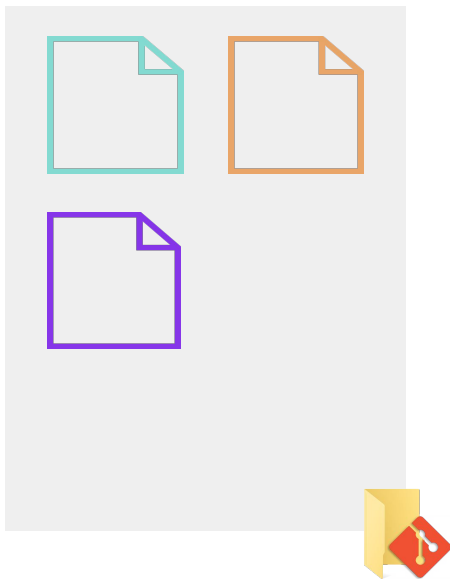
# ACoo

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

1. En el syllabus, vayan a la carpeta “Actividades” y descarguen el enunciado de la actividad (AC00).
2. Trabajen en ella hasta las 16:30.

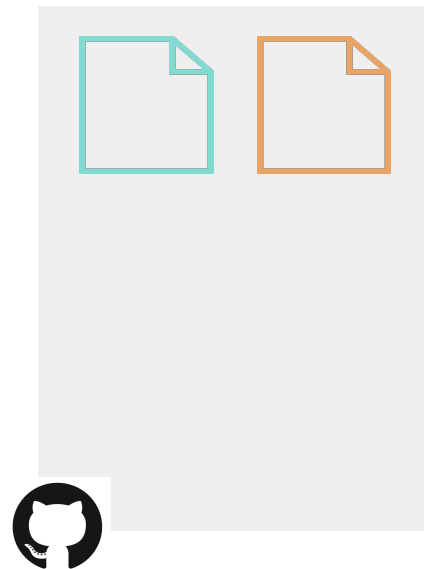
# ¿Qué acabamos de hacer?

Mi computador



```
git add README.md
```

GitHub

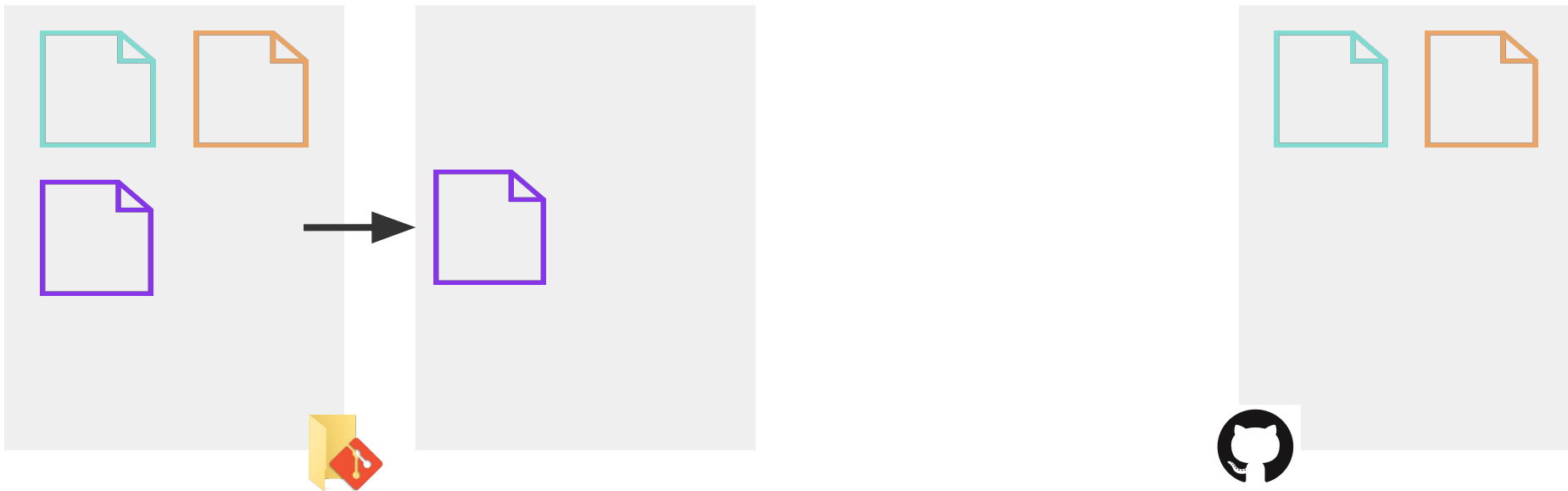


# ¿Qué acabamos de hacer?

Mi computador

Lista de cambios  
(*Staging area*)

GitHub



# ¿Qué acabamos de hacer?

Mi computador

Lista de cambios  
(*Staging area*)

GitHub

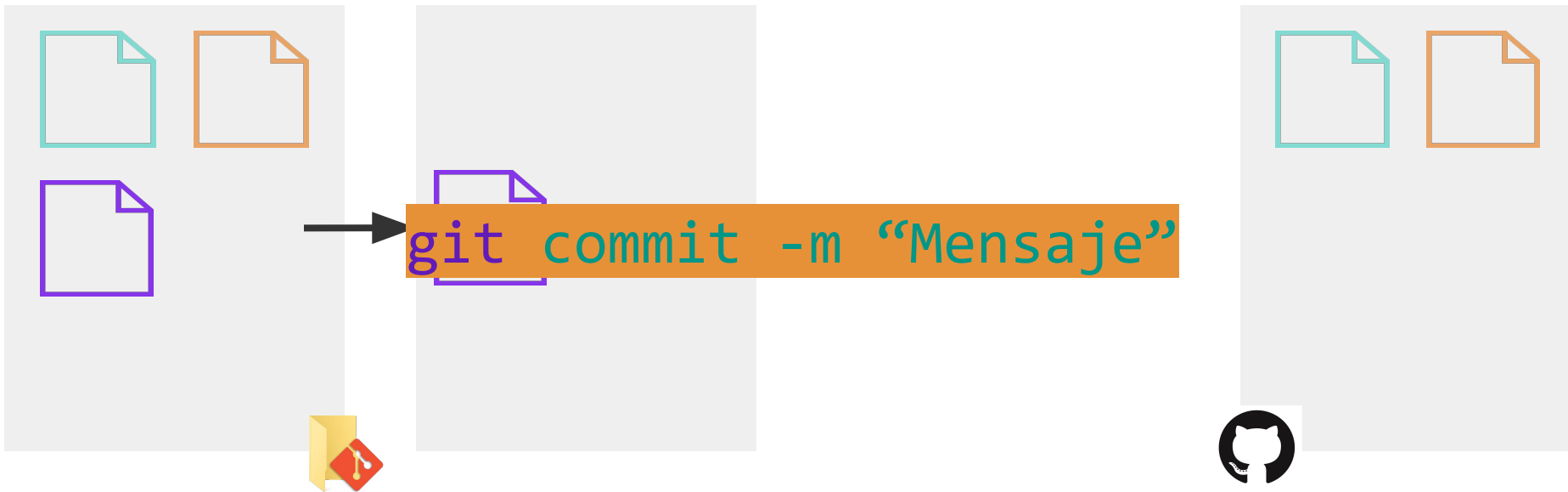


# ¿Qué acabamos de hacer?

Mi computador

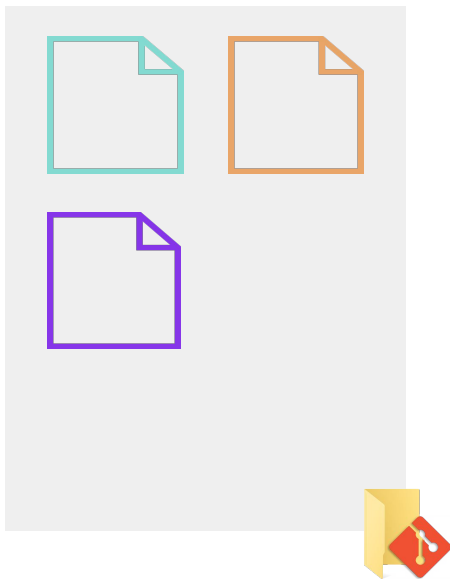
Lista de cambios  
(*Staging area*)

GitHub

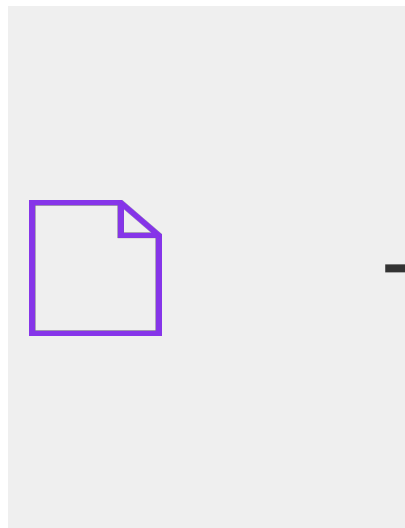


# ¿Qué acabamos de hacer?

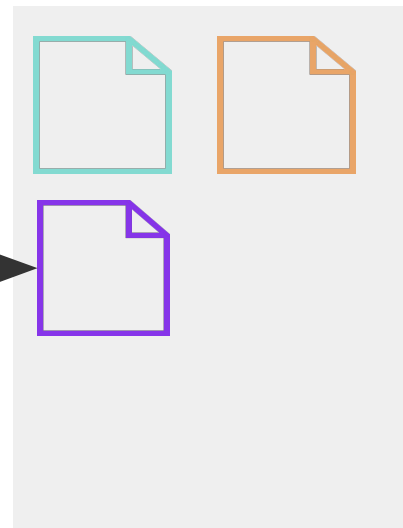
Mi computador



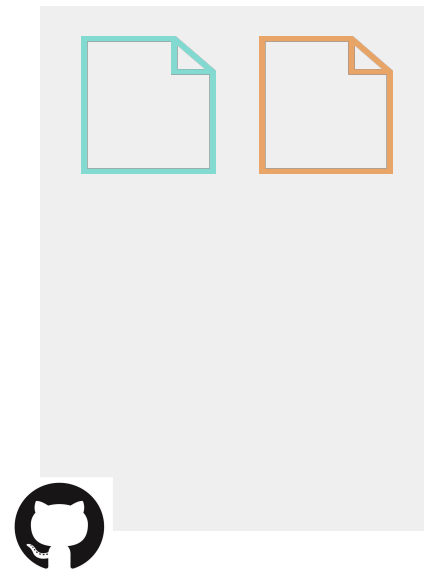
Lista de cambios  
(*Staging area*)



Repositorio local



GitHub



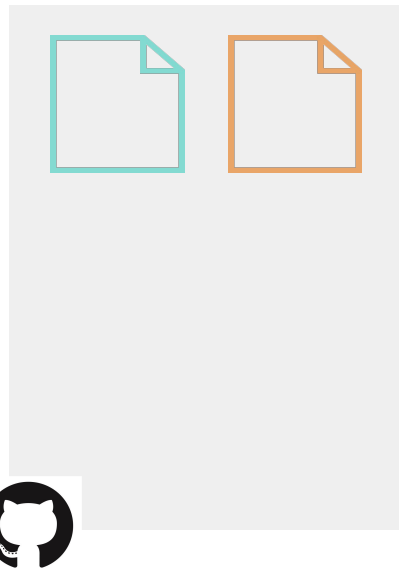
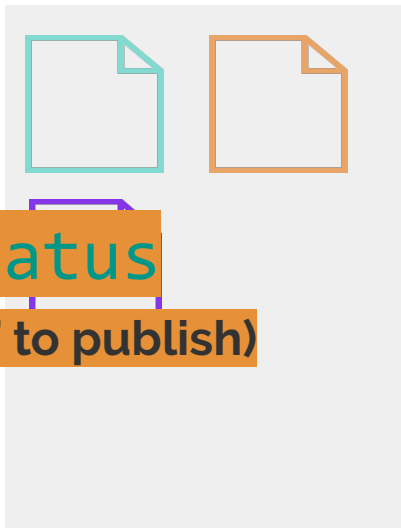
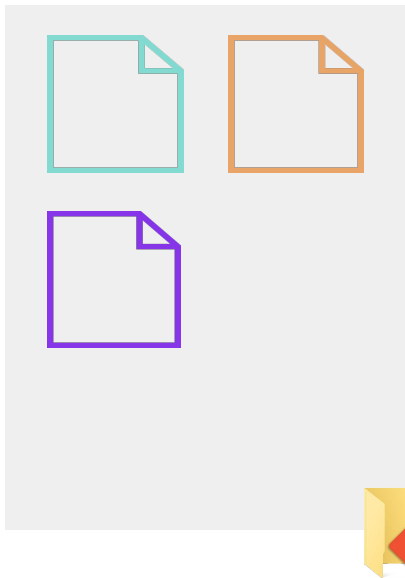
# ¿Qué acabamos de hacer?

Mi computador

Lista de cambios  
(*Staging area*)

Repositorio local

GitHub

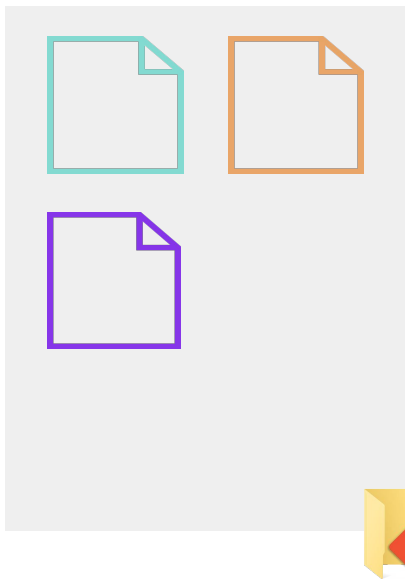


`git status`  
(use "git push" to publish)



# ¿Qué acabamos de hacer?

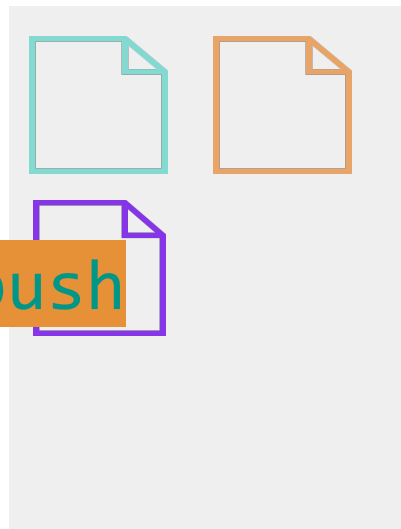
Mi computador



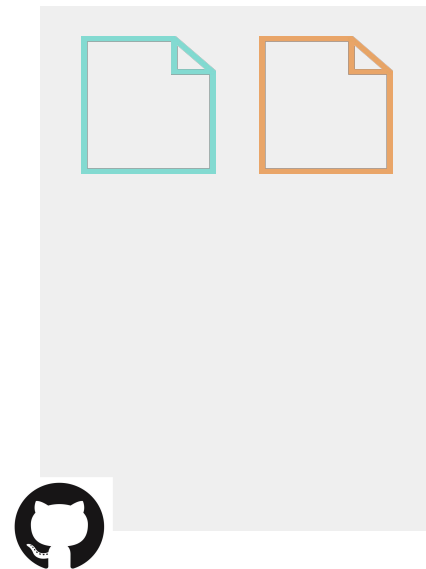
Lista de cambios  
(*Staging area*)



Repositorio local

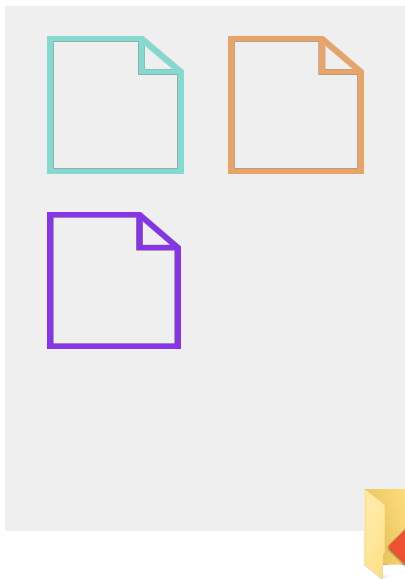


GitHub

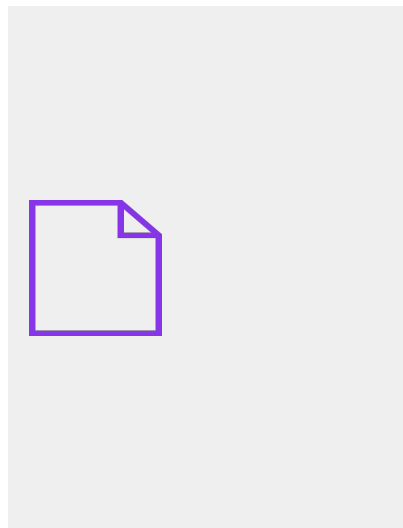


# ¿Qué acabamos de hacer?

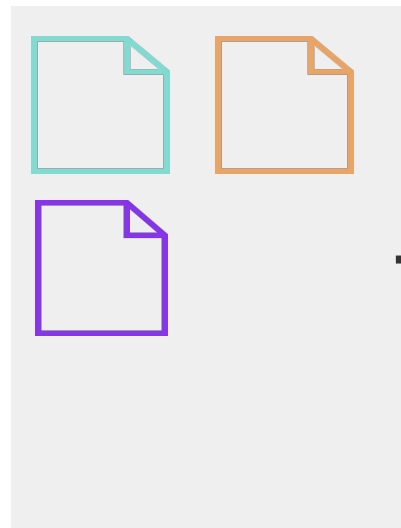
Mi computador



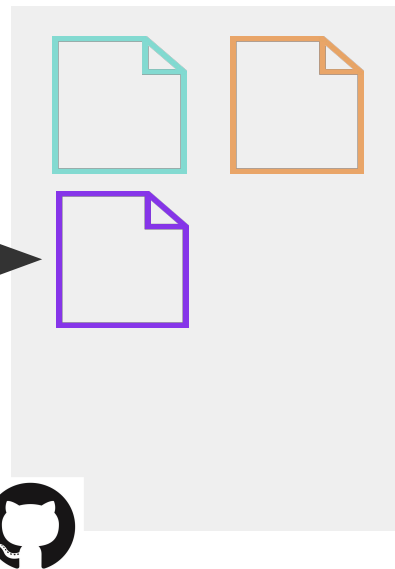
Lista de cambios  
(*Staging area*)



Repositorio local

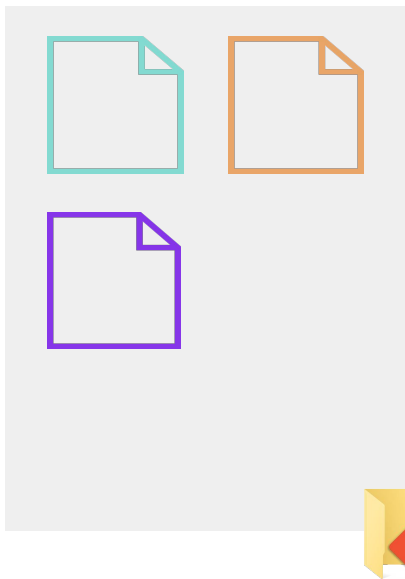


GitHub  
(Repositorio remoto)

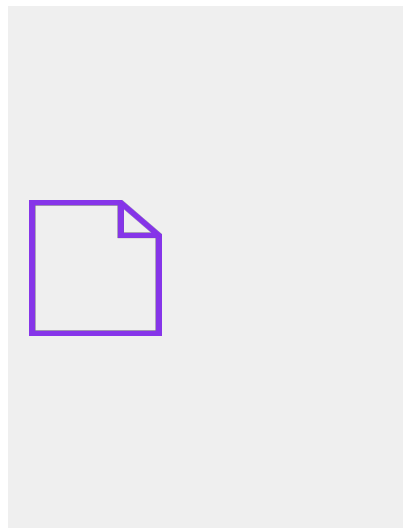


# ¿Qué acabamos de hacer?

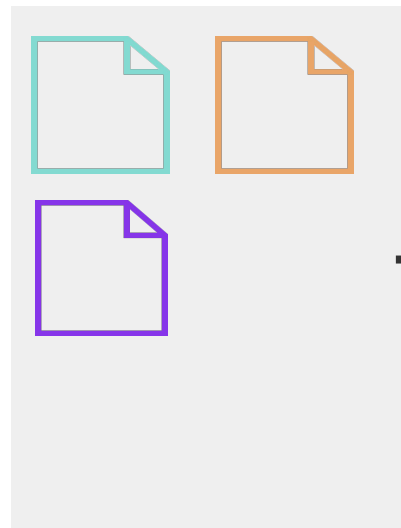
Mi computador



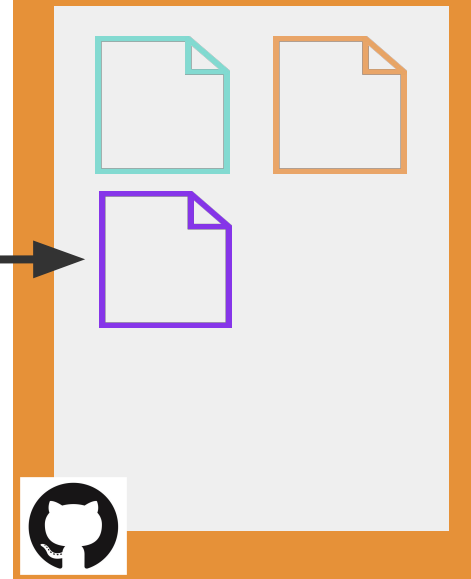
Lista de cambios  
(*Staging area*)



Repositorio local



GitHub  
(Repositorio remoto)

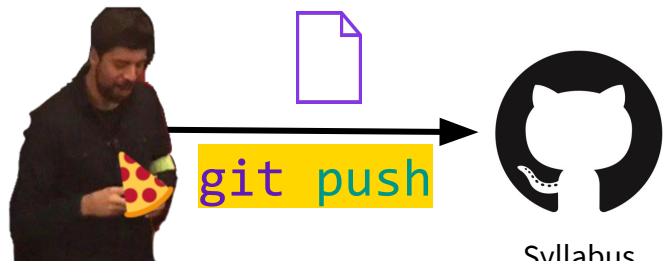




**El alumno/a que aprendió  
cómo funciona git**

# El flujo de una AC normal

14:30



Syllabus

`git pull`



Syllabus



`git add file`

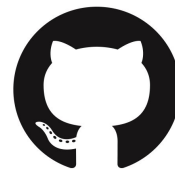
`git commit -m "avance 1"`

`git add file`

`git commit -m "avance 2"`

La nube  
Tu PC

16:30



`git pull`



aaossa-iic2233-2019-2



`git push`



aaossa-iic2233-2019-2



# Próximas semanas

- Mañana se publicarán contenidos mínimos para la realización de sus tareas.
- Y el sábado, a más tardar, se publicará el enunciado de la T00.
- Próximo martes tendrán ayudantía dónde se resolverán dudas de la T00, dudas generales y tal vez se discutan consejos para su realización. Los ayudantes les tienen una sorpresa para entonces.
- El próximo jueves es feriado, así que no nos vemos hasta dos semanas más.
- El próximo viernes se publica el material para la AC01, que se realiza el 22 de agosto.