



Ingeniería Industrial

FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Auxiliar 1

IN3501 - Tecnologías de Información y Comunicaciones para la Gestión

Auxiliares

José Canto - jose.canto@ug.uchile.cl

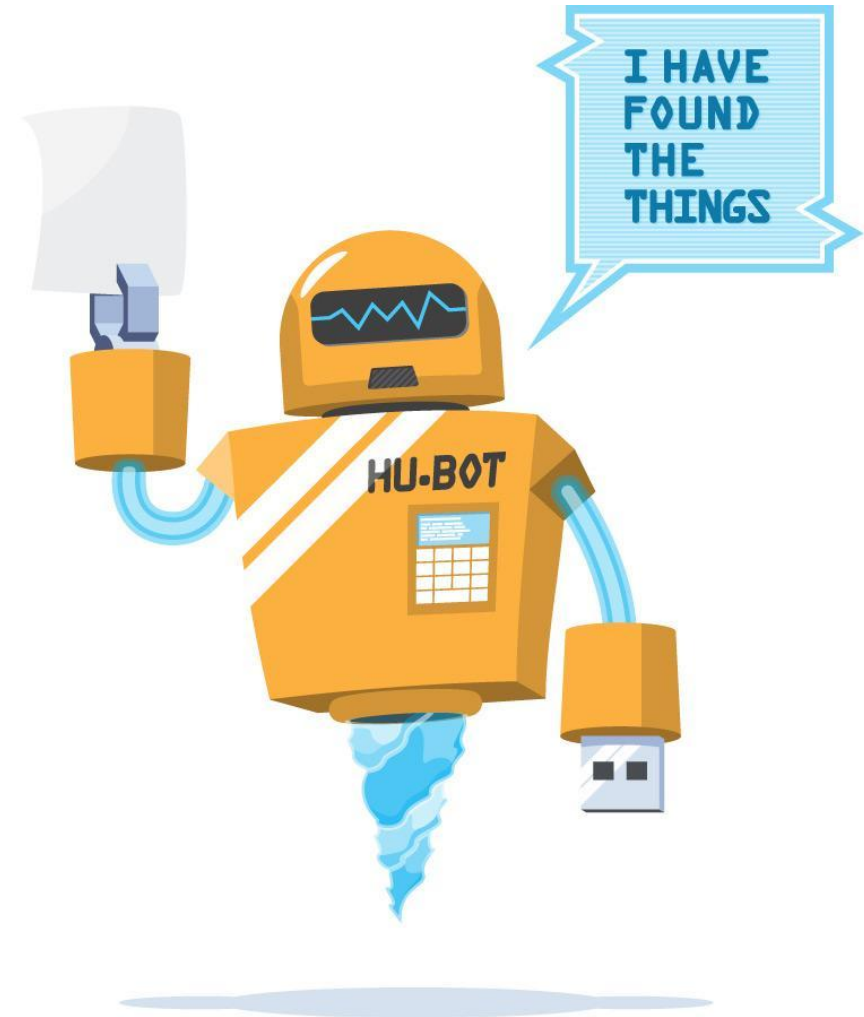
Javiera Ovalle - javiera.ovallet@gmail.com

Jorge Pinto - jorgepintoriveros@gmail.com



Agenda

- Presentación
- Git - Github
- Entorno virtual
- Proyecto de Django
- Desafío 0



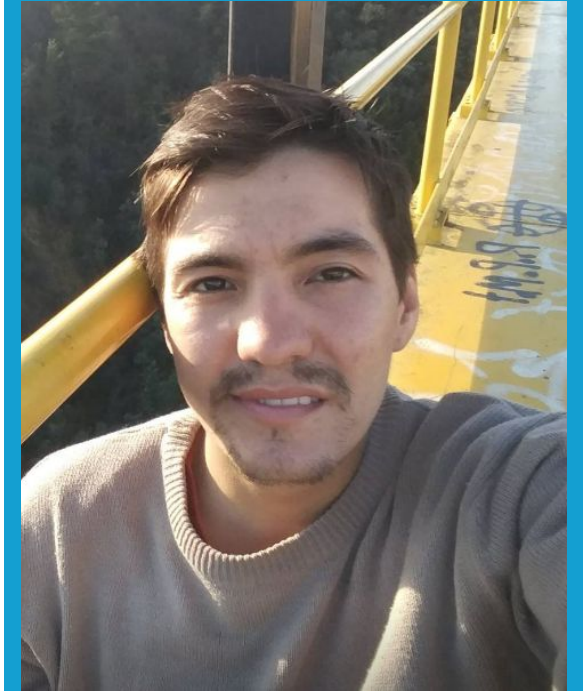
¿Quiénes son los auxiliares? The 3 J's



José Canto
Virgo
jose.canto@ug.uchile.cl



Javiera Ovalle
Géminis
Javiera.ovallet@gmail.com



Jorge Pinto
Virgo
Jorgepintoriveros@gmail.com

Descripción del curso

- **Desarrollo del curso en dos modalidades semi-independientes**
 - Cátedras (teórico)
 - Auxiliares (práctico)
- **Desafíos**
 - 8 desafíos + 1 desafío recuperativo
 - Plazo de un día para entregarlo, **entrega oficial únicamente vía U-Cursos**
 - El uso de github es **obligatorio**
- **Proyecto semestral**
 - Se divide en 4 tareas acumulativas
 - Consiste en desarrollar una página web para una pyme/microempresa/emprendimiento
 - Se programará en Django – Python y HTML

Git / Github



- Software de control de versiones que permite, a través de GitHub, el trabajo colaborativo no presencial.
- Permite copiar la última versión del trabajo en distintos computadores y actualizar el contenido desde la consola.

Para cada auxiliar o tarea se les entregará un link de Github, este link creará un **repositorio** en el cual ustedes podrán realizar su trabajo individual o grupal dependiendo del caso

Git / Github - ¿Cómo funciona?



Para algún proyecto en específico:

Ustedes poseen un **repositorio local** que corresponde a una carpeta dentro de su PC, esta carpeta se puede enlazar con un **repositorio online**, para que más personas tengan acceso.

La idea principal es que ustedes realizan cambios en su repositorio local y actualicen los cambios al repositorio online.

Git / Github - Comandos útiles

Para crear un repositorio: deben ir a alguna dirección de su computador (~~idealmente no en el escritorio~~) y usar el siguiente comando:

```
$ git init
```

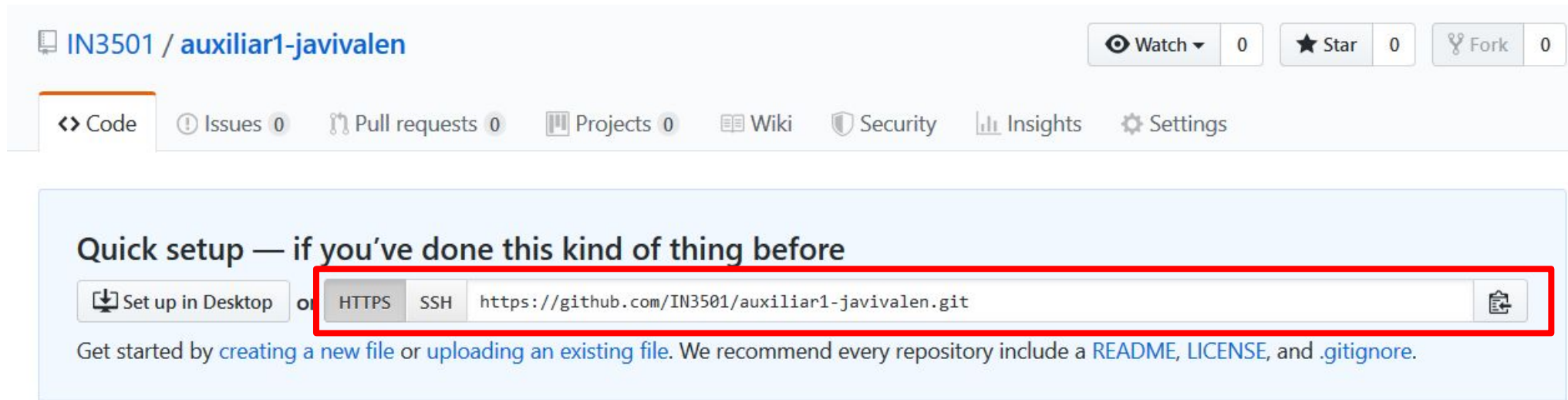
Para enlazar con un repositorio no vacío: deben ir a alguna dirección de su computador (~~nuevamente, no en su escritorio~~) y usar el siguiente comando:

```
$ git clone url_del_repositorio.git
```

Git / Github - Comandos útiles

Para enlazar con un repositorio no vacío: deben ir a alguna dirección de su computador (~~nuevamente, no en su escritorio~~) y usar el siguiente comando:

```
$ git clone url_del_repositorio.git
```



IN3501 / auxiliar1-javivalen

Watch 0 Star 0 Fork 0

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

Quick setup — if you've done this kind of thing before

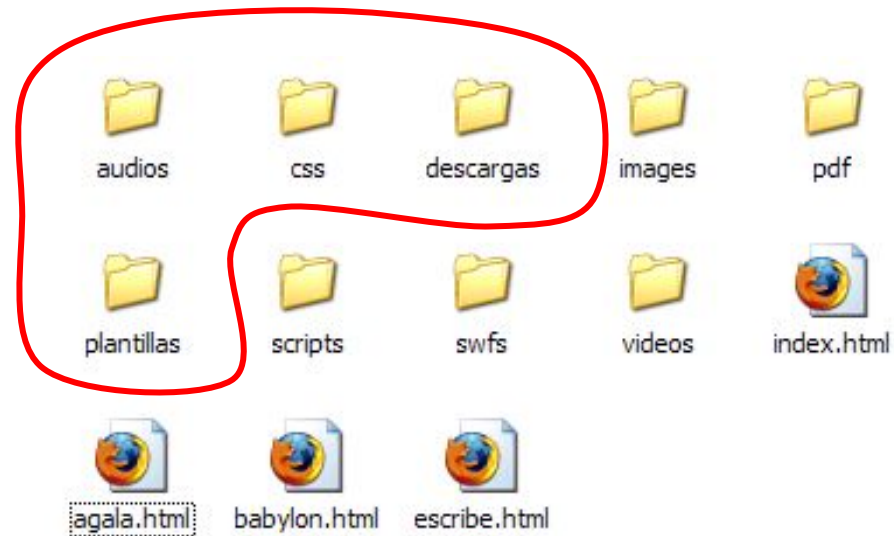
Set up in Desktop or HTTPS SSH https://github.com/IN3501/auxiliar1-javivalen.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

Git / Github - ¿Qué es un commit?

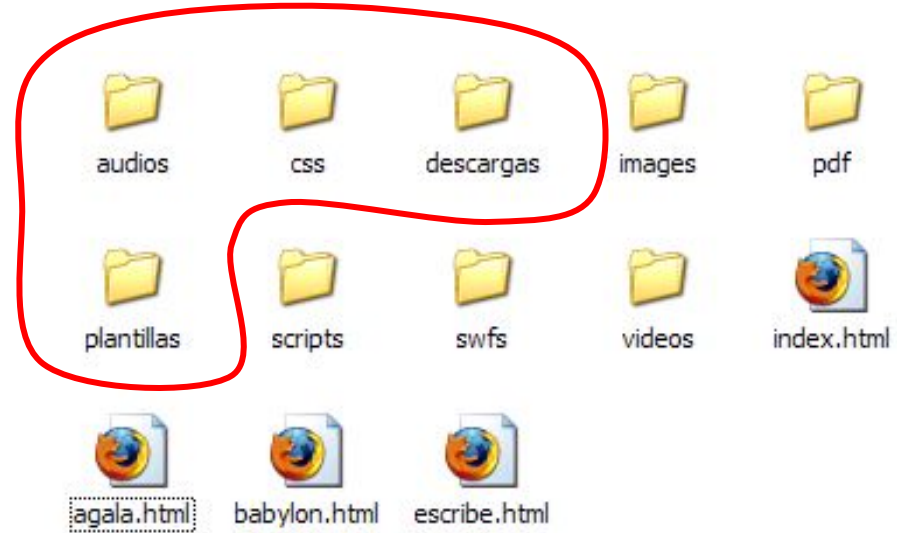
Un **commit** corresponde a un conjunto de cambios que yo realice en mi repositorio, los archivos que yo modifiko debo **agregarlos al commit**. En la siguiente imagen tengo un directorio con muchos archivos, pero no quiero agregarlos todos al commit, entonces **seleccionaré** los que deseo:

```
$ git add audios  
$ git add css  
$ git add descargas  
$ git add plantillas
```



Git / Github - Mi primer commit

```
$ git add audios  
$ git add css  
$ git add descargas  
$ git add plantillas
```



Luego de agregados los archivos, debo realizar un commit. Cada commit va acompañado de un mensaje que indica que es lo que hemos hecho en el commit

```
$ git commit -m "mensaje del commit"
```

Git / Github - Subir o rescatar cambios

Para subir los cambios que hemos hecho (que se ven reflejados a través de un commit), debemos hacer **push**:

```
$ git push
```

Para traer cambios que están en el repositorio online, pero no en el nuestro debemos hacer **pull**:

```
$ git pull
```

Github - ¿Cómo lo usamos?

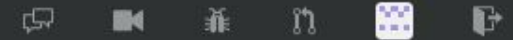
Deben ingresar al [link de esta auxiliar](#) y realizar los siguientes pasos:

1. Online: Aceptar la tarea
2. Online: Acceder al repositorio creado
3. CMD: Crear entorno virtual
4. CMD: Iniciar el entorno virtual
5. CMD: Iniciar git
6. CMD: Clonar el repositorio a su computador con git
7. CMD: Instalar requirements.txt
8. CMD: Crear proyecto Django
9. CMD: Agregar los cambios a github (git add - commit - push)
10. CMD: Correr el servidor

Github - Aceptar la tarea

GitHub Classroom

GitHub Education



IN3501 - Otoño 2019

@IN3501



Accept the **Auxiliar 2** assignment

Accepting this assignment will give you access to the **auxiliar-2-jpintoriv** repository in the [@IN3501](#) organization on GitHub.

Accept this assignment

Github - Se crea el repositorio

IN3501-Primavera2019

IN3501



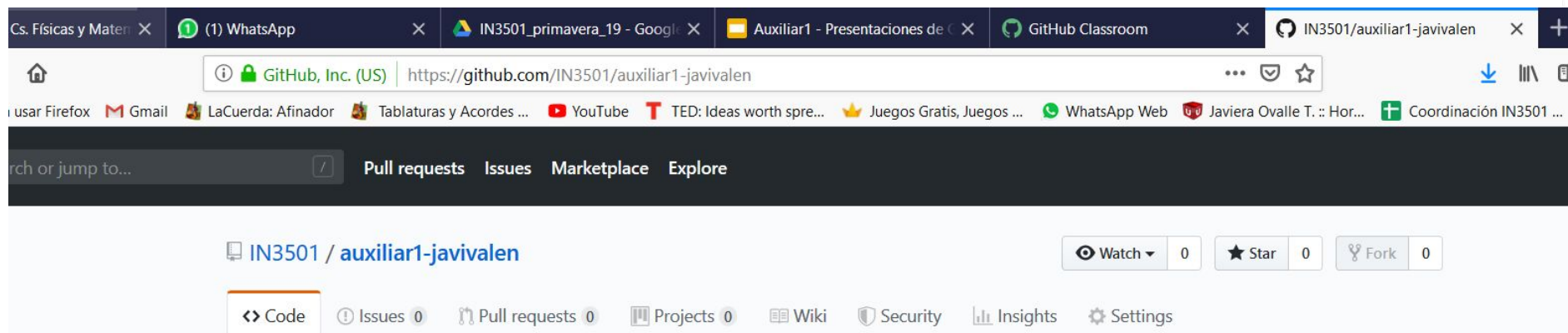
Accepted the **Auxiliar1** assignment

You are ready to go!

You may receive an invitation to join @IN3501 via email invitation on your behalf. No further action is necessary.

Your assignment has been created here: <https://github.com/IN3501/auxiliar1-javivalen>

Github - Se crea el repositorio



Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/IN3501/auxiliar1-javivalen.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# auxiliar1-javivalen" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/IN3501/auxiliar1-javivalen.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/IN3501/auxiliar1-javivalen.git
git push -u origin master
```

Crear entorno virtual

Vamos a crear un entorno virtual **DENTRO** de la carpeta donde clonamos nuestro repositorio: (se recomienda usar el nombre env o venv)

```
$ virtualenv nombre_entorno_virtual
```

```
$ python -m venv nombre_entorno_virtual
```

La idea del entorno virtual es que dentro de él instalamos los elementos de python que queremos ocupar (en este caso usaremos Django)

Iniciar entorno virtual

Windows:

```
$ nombre_entorno_virtual\Scripts\activate
```

Mac/Ubuntu:

```
$ source nombre_entorno_virtual/bin/activate
```

Iniciar repositorio

Vayan a alguna dirección de su computador (~~idealmente no en el escritorio~~) y clonar el repositorio. **IMPORTANTE:** como es un repositorio vacío, no sirve el comando “git clone” deben usar los siguientes comandos:

Para iniciar un repositorio local:

```
$ git init
```

Para enlazar su repositorio local con el de github vacío

```
$ git remote add origin url de su repositorio
```


Subir requirements.txt

Agregar archivos al commit:

```
$ git add requirements.txt
```

Realizar commit:

```
$ git commit -m "agregar requirements.txt"
```

Subir al repositorio:

- En caso de que sea su primer push:

```
$ git push -u origin master
```

- En caso de que ya hayan hecho un push

```
$ git push
```

Acceder a mi cuenta github



```
(entorno11) C:\Users\javie\OneDrive\Escritorio\FCFM\Semestre 12\TICs\Auxiliar1>git add requirements.txt
```

```
(entorno11) C:\Users\javie\OneDrive\Escritorio\FCFM\Semestre 12\TICs\Auxiliar1>git commit -m "requirements"
```

```
*** Please tell me who you are.
```

Run

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

to set your account's default identity.

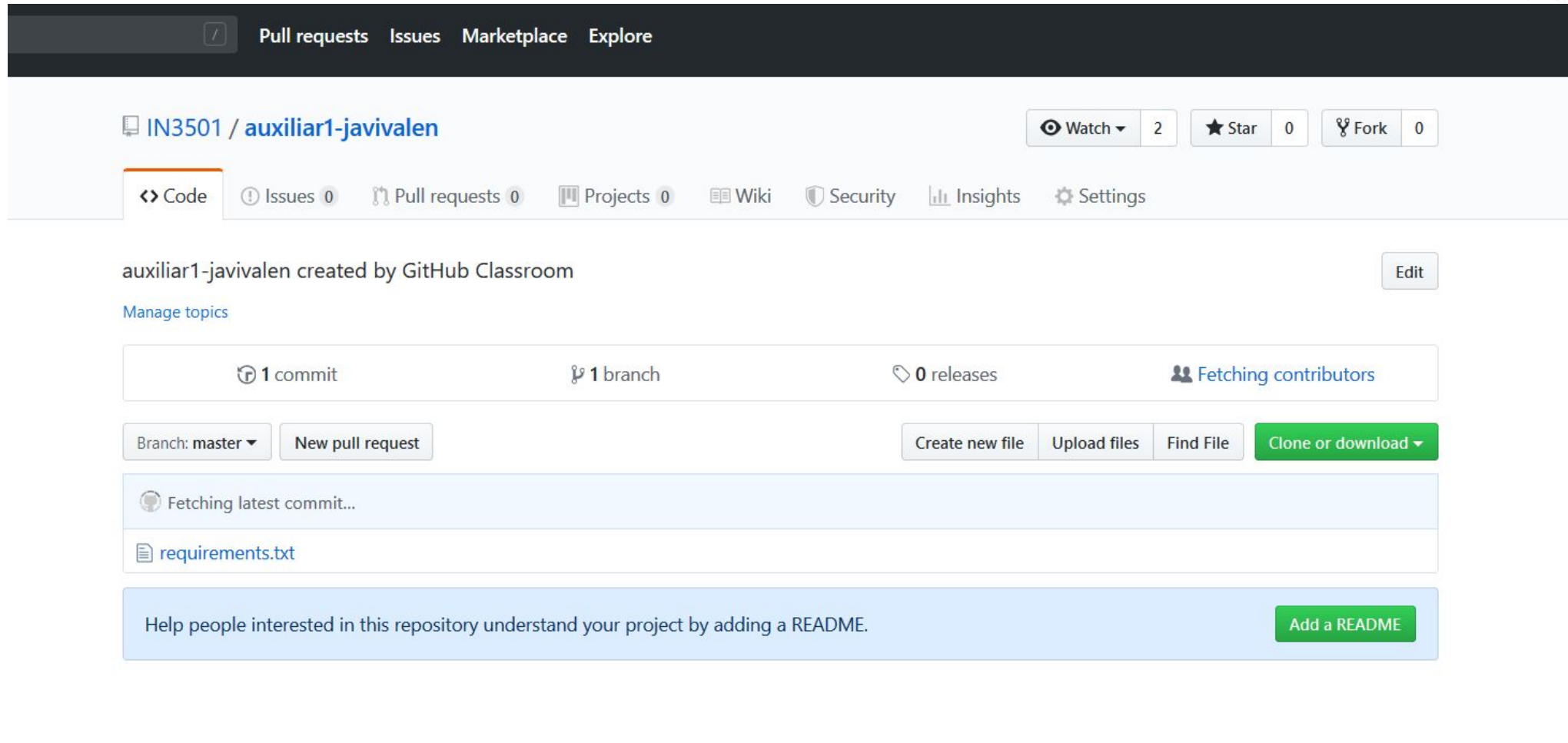
Omit --global to set the identity only in this repository.

```
fatal: unable to auto-detect email address (got 'Javiera@JAVIERA.(none)')
```

Linkearemos la cuenta github al PC

```
git config user.email "tu_correo"
git config user.name "tu_usuario_git"
```

Commit realizado - actualización online



The screenshot shows the GitHub interface for a repository named 'auxiliar1-javivalen' under the user 'IN3501'. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. The repository header shows 2 watches, 0 stars, and 0 forks. Below the header, tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Security, Insights, and Settings are visible. The repository description states it was created by GitHub Classroom, with an 'Edit' button. A summary bar indicates 1 commit, 1 branch, and 0 releases, with a 'Fetching contributors' button. Action buttons include 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find File', and 'Clone or download'. A status bar shows 'Fetching latest commit...' and a file named 'requirements.txt'. A blue banner at the bottom encourages adding a README, with an 'Add a README' button.

Pull requests Issues Marketplace Explore

IN3501 / auxiliar1-javivalen

Watch 2 Star 0 Fork 0

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

auxiliar1-javivalen created by GitHub Classroom [Edit](#)

[Manage topics](#)

1 commit 1 branch 0 releases [Fetching contributors](#)

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

Fetching latest commit...

requirements.txt

Help people interested in this repository understand your project by adding a README. [Add a README](#)



Pero... ¿qué es requirements.txt?

El archivo **requirements.txt** es **MUY** utilizado para guardar los elementos que se deben instalar en el entorno virtual para que los proyectos funcionen.

Opción 1: Crearlo a partir de nuestro entorno virtual:

```
$ pip freeze > requirements.txt
```

Opción 2: En caso de clonar un proyecto que cuenta con requirements.txt, **deben** instalar en su entorno virtual las dependencias que necesita este proyecto

```
$ pip install -r requirements.txt
```

Crear proyecto de Django

Para crear un proyecto de Django, se usa el siguiente comando:

```
$ django-admin startproject nombre_proyecto
```

luego de creado, debemos ingresar al proyecto a través de la consola:

```
$ cd nombre_proyecto
```


Correr proyecto en Django

Cuando estemos dentro del proyecto, para “hacer correr” el servidor de django en nuestro computador:

```
$ python manage.py runserver
```

Este comando hace que el proceso de django corra en el puerto 8000 de nuestro computador, esto quiere decir que para ingresar a nuestro proyecto de django a través del computador debemos ingresar a las direcciones:

<http://localhost:8000> o <http://127.0.0.1:8000>

Subir proyecto al repositorio

Para subir el proyecto debemos volver a la carpeta que contiene el proyecto (podemos cerrar el proyecto o abrir una nueva terminal)

Para “subir” un directorio:

```
$ cd ..
```

add, commit y push:

```
$ git add nombre_proyecto
```

```
$ git commit -m “proyecto de django”
```

```
$ git push
```

DESAFÍO 0

Este desafío es un ejemplo de los que vienen a lo largo del semestre. Es extremadamente importante que suba lo solicitado a U-Cursos - Desafío 0

Su entrega debe contener:

- Pantallazo de instalación de:
 - Python (pantallazo de la versión en CMD)
 - Django (pantallazo del cohete de Django)
 - Workbench (pantallazo de la página de inicio del programa)
 - Git (pantallazo del programa en el escritorio)
- Debe iniciar su repositorio en el link de la auxiliar y tener un mínimo de 2 commits:
 - Uno para el requirements.txt
 - Otro para el proyecto de Django
- Pantallazo de los commits realizados (como la diapositiva 21)
- Link del repositorio GitHub

SUBIR ANTES DEL MIÉRCOLES 7 DE AGOSTO A LAS 23:59



Ingeniería Industrial

FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Auxiliar 1

IN3501 - Tecnologías de Información y Comunicaciones para la Gestión

Auxiliares

José Canto - jose.canto@ug.uchile.cl

Javiera Ovalle - javiera.ovallet@gmail.com

Jorge Pinto - jorgepintoriveros@gmail.com

