Escuela de Datos Vivos

Curso Data Engineer: Creando un pipeline de datos

SQL WF, Python, Git hub, comandos shell- Clase 2

Agenda

- SQL WF
- Comandos shell
- GitHub
- Python

SQL WF

Windows function vs Group by

JOB_TITLE	SALARY
ANALYST	3100
ANALYST	2900
ANALYST	3250
SALES	1700
SALES	2500
SALES	4100
SALES	1600
SALES	2200
ENGINEER	3500
ENGINEER	3100
ENGINEER	4100

GROUP BY

JOB_TITLE	AVG_SALARY
ANALYST	3083.33333
ENGINEER	3566.66667
SALES	2420

Window Function

JOB_TITLE	SALARY	AVG_SALARY
ANALYST	3100	3083.333333
ANALYST	2900	3083.333333
ANALYST	3250	3083.333333
SALES	1700	2420
SALES	2500	2420
SALES	4100	2420
SALES	1600	2420
SALES	2200	2420
ENGINEER	3500	3566.666667
ENGINEER	3100	3566.666667
ENGINEER	4100	3566.666667

Avg

```
SELECT o.Customer_Id,

od.Unit_Price,

AVG(od.Unit_Price) OVER (PARTITION BY o.Customer_Id) AS
“AvgUnitPrice”

FROM Orders o

INNER JOIN Order_details od ON o.order_Id = od.order_id;
```

Sum

```
SELECT Customer_Id,
od.unit_price,
SUM(Unit_Price) OVER (PARTITION BY Customer_Id ) AS "SumSales"
FROM Orders o
INNER JOIN order_details od ON o.order_id = od.order_id;
```

Rank

```
SELECT Customer_Id,

Unit_Price,

RANK() OVER (PARTITION BY Customer_Id ORDER BY unit_price DESC) AS
"Unit_Rank"

FROM Orders

INNER JOIN order_details

ON orders.order_id = Order_Details.Order_Id
```

Row Number

```
SELECT Customer_Id,

Unit_Price,

ROW_NUMBER() OVER (PARTITION BY Customer_Id ORDER BY unit_price
DESC) AS "Unit_R_number"

FROM Orders

INNER JOIN order_details

ON orders.order_id = Order_Details.Order_Id
```

LAG

```
SELECT od.customer_id,

o.order_date,

od.unit_price,

LAG(od.unit_price) OVER (PARTITION BY od.customer_id ORDER BY
o.order_date) AS "LAG"

FROM orders o

INNER JOIN order_details od ON o.order_id = od.order_id
```

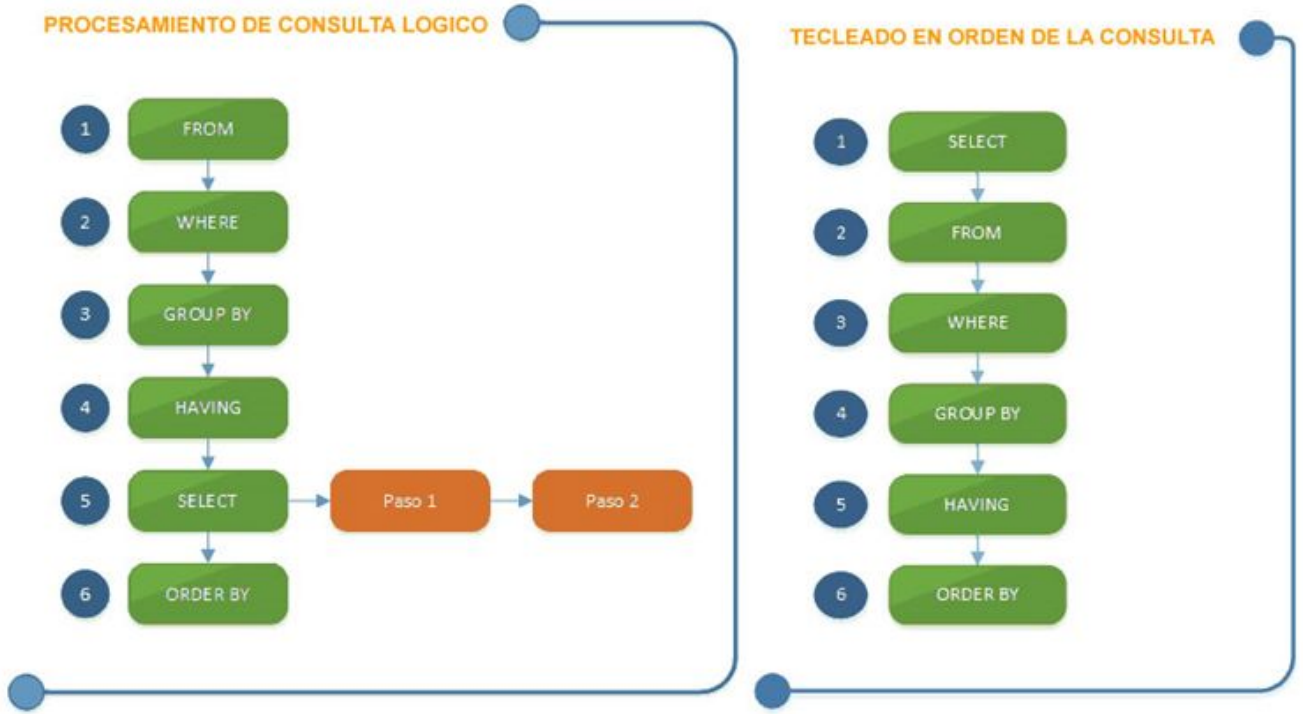
LEAD

```
SELECT od.customer_id,

o.order_date,

od.unit_price,

LEAD(od.unit_price) OVER (PARTITION BY od.customer_id ORDER BY
o.order_date) AS "LEAD"

FROM orders o

INNER JOIN order_details od ON o.order_id = od.order_id
```

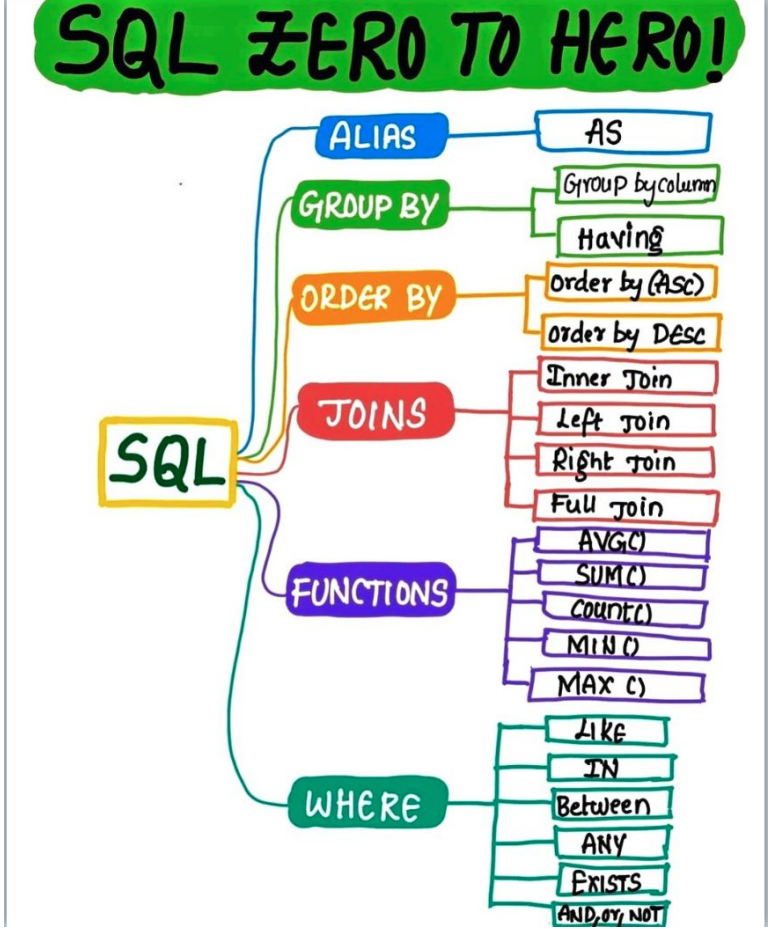
SQL - Orden


TECLEADO EN ORDEN DE LA CONSULTA
1
SELECT
2
FROM
3
WHERE
4
GROUP BY
5
HAVING
6
ORDER BY

SQL - Orden


PROCESAMIENTO DE CONSULTA LOGICO
1
FROM
2
WHERE
3
GROUP BY
4
HAVING
5
SELECT
Paso 1
Paso 2
6
ORDER BY
TECLEADO EN ORDEN DE LA CONSULTA
1
SELECT
2
FROM
3
WHERE
4
GROUP BY
5
HAVING
6
ORDER BY

Conceptos SQL



Comandos Shell

Listar directorios

ls **ls -a** (archivos ocultos) **ls -l** (lista) **ls -R** (sub directorios)

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents
$ ls -l
total 0
lrwxrwxrwx 1 fede 197609 20 Jan 21 15:37 'My Music' -> /c/Users/f_pin/Music/
lrwxrwxrwx 1 fede 197609 23 Jan 21 15:37 'My Pictures' -> /c/Users/f_pin/Pictures
lrwxrwxrwx 1 fede 197609 21 Jan 21 15:37 'My Videos' -> /c/Users/f_pin/Videos/
drwxr-xr-x 1 fede 197609  0 Mar 11 11:38  Repo/
```

Ingresar a directorios y ubicación

cd (ingreso a un directorio)

pwd (me indica en qué directorio estoy)

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents
$ cd Repo/

fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ pwd
/c/Users/f_pin/Documents/Repo
```

Crear/Borrar directorios

mkdir <nombre del directorio> (crea un directorio nuevo)
rmdir <nombre del directorio> (borra un directorio)

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents
$ mkdir Repo
```

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents
$ rmdir Repo
```

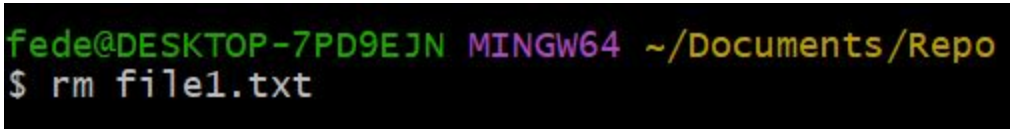
Crear archivos

cat > <nombre de archivo> (crea un archivo nuevo)
cat <nombre de archivo> (muestra el contenido del archivo)

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ cat > archivo.txt
agrego nuevo contenido al archivo y para finalizar escribir ctrl + d

fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ cat archivo.txt
agrego nuevo contenido al archivo y para finalizar escribir ctrl + d
```

Borrar archivos

rm <nombre de archivo> (Borra un archivo)



Copiar o mover archivos

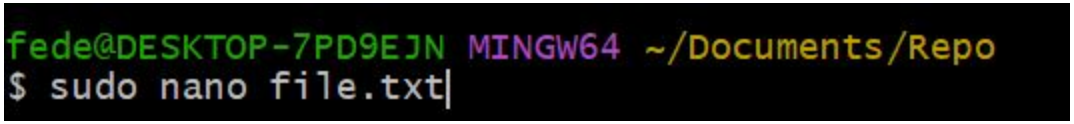
cp <nombre de archivo> <nuevo archivo> (copia un archivo a uno nuevo)
mv <nombre de archivo> <nuevo archivo> (mueve el archivo de un lugar a otro) - También sirve para renombrar un archivo.

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ cp archivo.txt /c/Users/f_pin/Documents/
```

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ mv archivo.txt archivo2.txt

fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ ls
archivo2.txt  homework-0/
```

Elevar permisos

sudo <comando> (Permite ejecutar comandos con privilegios de Root)



NOTA: puede que no esté instalado y para poder ejecutar el comando debemos que tener permisos de sudoers

Modificar permisos

Modo octal

```
hadoop@5dc251dd43fb:~$ ls -l
total 404
drwxrwxr-x 1 hadoop hadoop   4096 Mar 11 13:45 airflow
-rw-rw-r-- 1 hadoop hadoop     16 Mar 11 13:44 archivo.txt
```

Posición de la cifra de la clase de usuario	Significado
1	Corresponde a la clase de usuario “propietario” (*user*).
2	Corresponde a la clase de usuario “grupo” (*group*).
3	Corresponde a la clase de usuario “otros” (*other*).

Modificar permisos

Modo octal

Valor para derechos de acceso	Significado
4	Leer
2	Escribir
1	Ejecutar
0	Sin permisos

Valor	Derecho(s) de acceso
0	Ninguno
1	Solo ejecutar
2	Solo escribir
3	Escribir / ejecutar
4	Solo leer
5	Leer / ejecutar
6	Leer / escribir
7	Todos los permisos

Modificar permisos

chmod <permisos> <archivo> (Modifica los permisos de un archivo o directorio)

```
hadoop@5dc251dd43fb:~$ ls -l
total 404
drwxrwxr-x 1 hadoop hadoop   4096 Mar 11 13:45 airflow
-rw-rw-r-- 1 hadoop hadoop     16 Mar 11 13:44 archivo.txt
```

```
hadoop@5dc251dd43fb:~$ chmod 644 archivo.txt
```

```
hadoop@5dc251dd43fb:~$ ls -l
total 404
drwxrwxr-x 1 hadoop hadoop   4096 Mar 11 13:45 airflow
-rw-r--r-- 1 hadoop hadoop     16 Mar 11 13:44 archivo.txt
```

Modificar permisos

Modo simbólico o carácter

Carácter para permisos	Significado
r	Permiso de lectura (*read*); también llamado bit R
w	Permiso de escritura (*write*); también llamado bit W
x	Permiso de ejecución (*execute*); también llamado bit X
Carácter para clases de usuarios	**Significado**
u	*user*, propietario
g	*group*, grupo
o	*other*, otros
a	*all*, todas las clases

Modificar permisos

Modo simbólico o carácter

+	Con el operador “+” se asignan más derechos de archivo a una clase de usuario. Solo se sobrescriben los derechos afectados.
-	El operador “-“retira derechos de acceso a una clase de usuario.
=	Si los permisos de archivo de una clase de usuario se han de renovar, sin importar qué derechos tuvo antes, se usa el operador “=”.

Modificar permisos

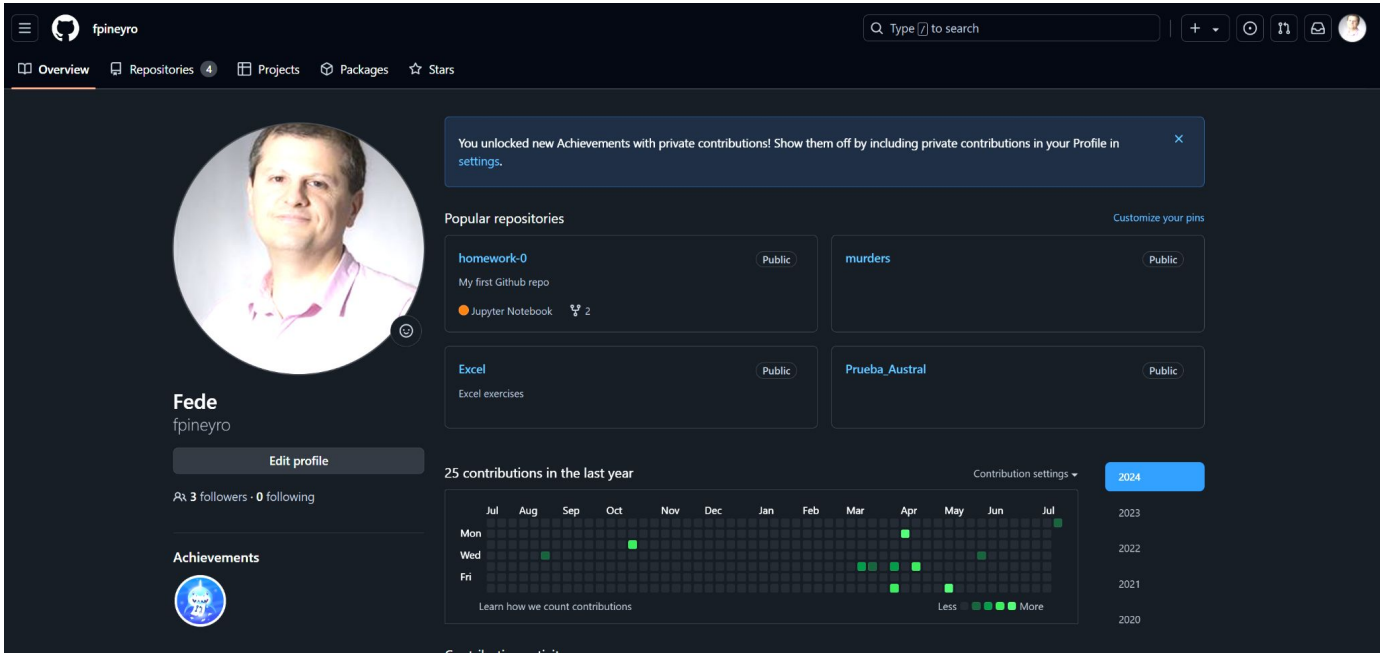
chmod <permisos> <archivo> (Modifica los permisos de un archivo o directorio)

```
hadoop@5dc251dd43fb:~$ ls -l
total 404
drwxrwxr-x 1 hadoop hadoop   4096 Mar 11 14:05 airflow
-rw-r--r-- 1 hadoop hadoop     16 Mar 11 13:44 archivo.txt
```

```
hadoop@5dc251dd43fb:~$ chmod ugo+rw archivo.txt
hadoop@5dc251dd43fb:~$ ls -l
total 404
drwxrwxr-x 1 hadoop hadoop   4096 Mar 11 14:05 airflow
-rw-rw-rw- 1 hadoop hadoop     16 Mar 11 13:44 archivo.txt
```

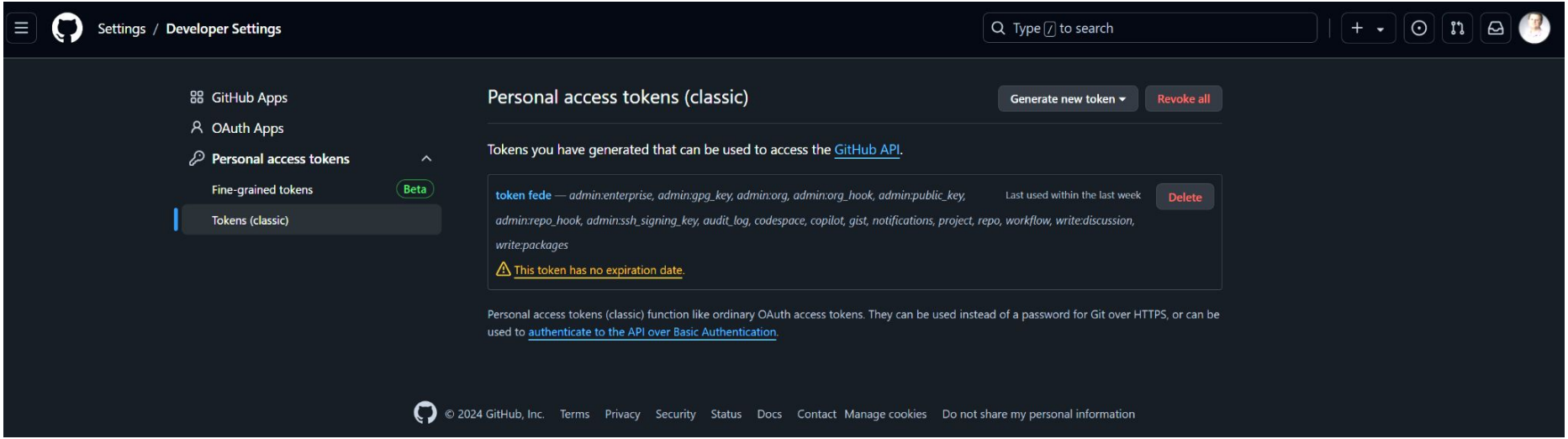
GitHub

Crear una cuenta en github



Crear un token

Settings -> developer settings -> personal access token -> classic token



Instalación

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

Installing on macOS

There are several ways to install Git on a Mac. The easiest is probably to install the Xcode Command Line Tools. On Mavericks (10.9) or above you can do this simply by trying to run `git` from the Terminal the very first time.

```
$ git --version
```

If you don't have it installed already, it will prompt you to install it.

If you want a more up to date version, you can also install it via a binary installer. A macOS Git installer is maintained and available for download at the Git website, at https://git-scm.com/download/mac.

1.5 Getting Started - Installing Git

Installing Git

Before you start using Git, you have to make it available on your computer. Even if it's already installed, it's probably a good idea to update to the latest version. You can either install it as a package or via another installer, or download the source code and compile it yourself.

Note	This book was written using Git version 2. Since Git is quite excellent at preserving backwards compatibility, any recent version should work just fine. Though most of the commands we use should work even in ancient versions of Git, some of them might not or might act slightly differently.

Installing on Linux

If you want to install the basic Git tools on Linux via a binary installer, you can generally do so through the package management tool that comes with your distribution. If you're on Fedora (or any closely-related RPM-based distribution, such as RHEL or CentOS), you can use `dnf`:

```
$ sudo dnf install git-all
```

If you're on a Debian-based distribution, such as Ubuntu, try `apt`:

```
$ sudo apt install git-all
```

For more options, there are instructions for installing on several different Unix distributions on the Git website, at https://git-scm.com/download/linux.

Instalación

sudo apt-get install git

```
fpineyro@BOOK-Fede:~$ sudo apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates git-man libbrotli1 libcbor0.8 libcurl3-gnutls liberror-perl libexpat1 libfido2-1 libgdbm-compat4
  libgdbm6 libldap-2.5-0 libldap-common libnghttp2-14 libperl5.36 libpsl5 librtmp1 libsasl2-2 libsasl2-modules
  libsasl2-modules-db libssh2-1 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxext6 libxmuu1 openssh-client
  openssl patch perl perl-modules-5.36 publicsuffix xauth
Suggested packages:
  gettext-base git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki
  git-svn gdbm-l10n libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap
  libsasl2-modules-otp libsasl2-modules-sql keychain libpam-ssh monkeysphere ssh-askpass ed diffutils-doc perl-doc
  libterm-readline-gnu-perl | libterm-readline-perl-perl make libtap-harness-archive-perl
The following NEW packages will be installed:
```

Verificación

git - -version

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ git --version
git version 2.39.2.windows.1
```

Configuración

git config - -global user.name <tu usuario>

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ git config --global user.name fpineyro
```

git config - -global user.email <tu correo>

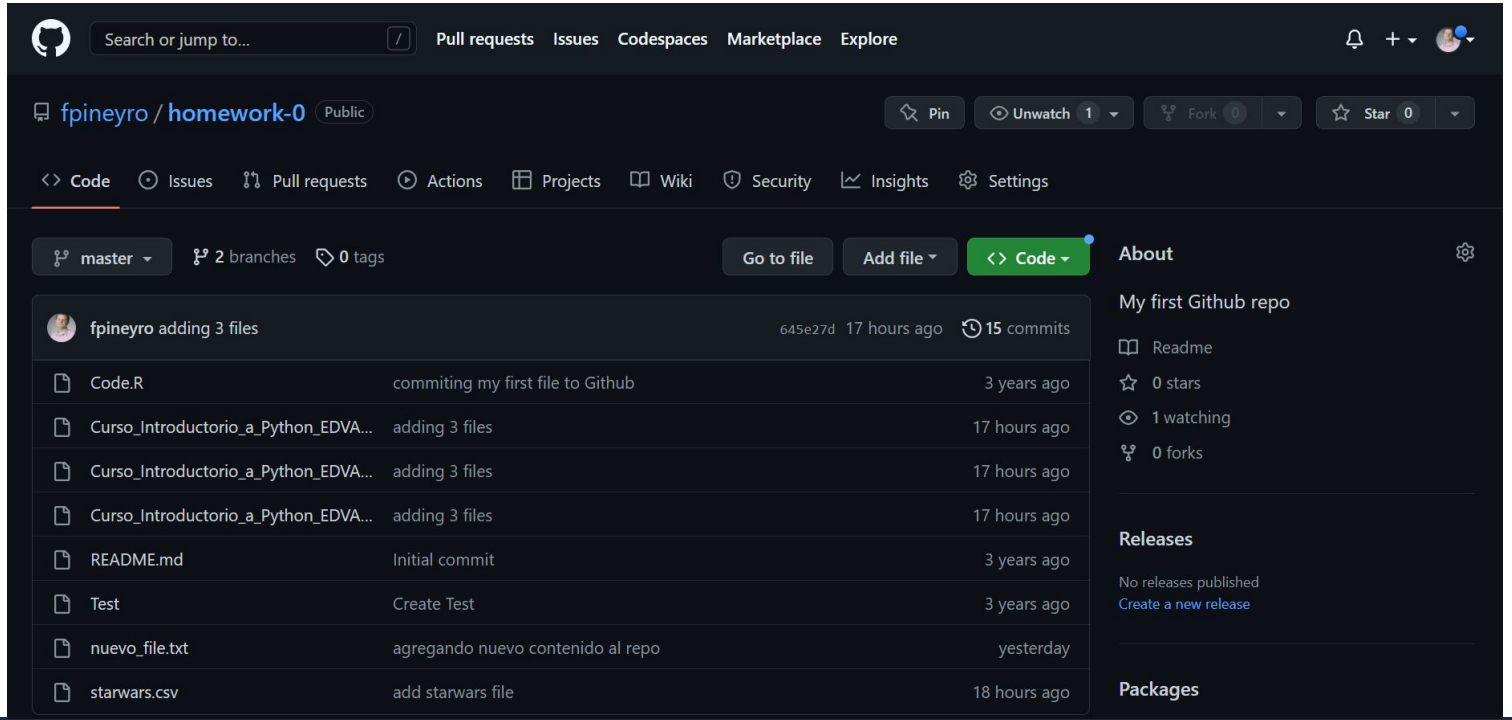
```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ git config --global user.email fpineyro@gmail.com
```

Configuración

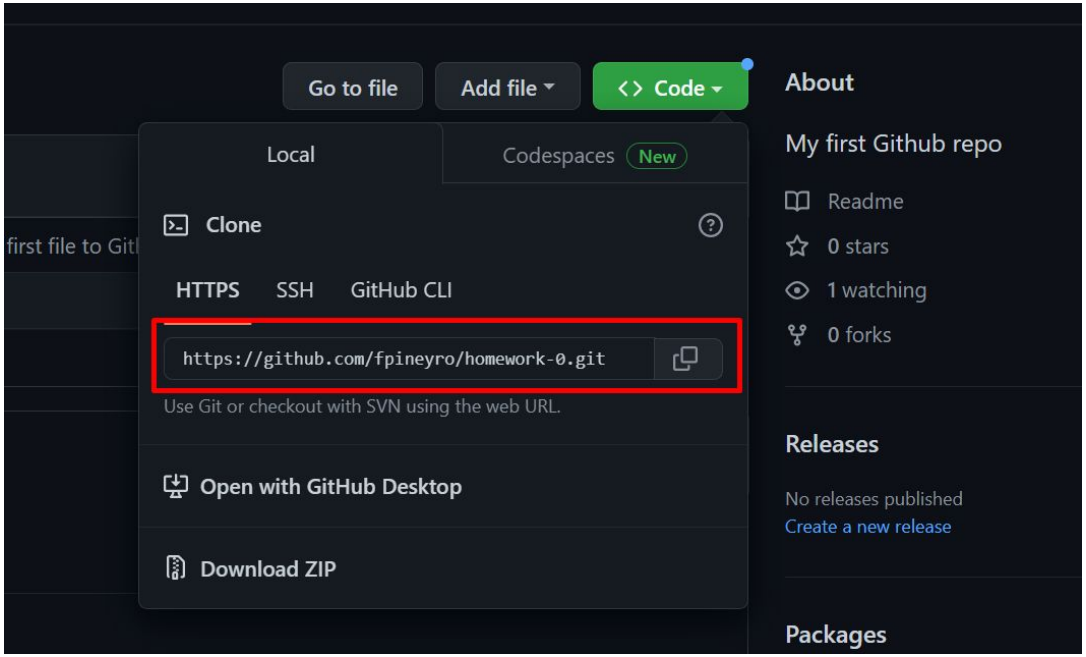
git config - -global - -list

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ git config --global --list
user.name=fpineyro
user.email=fpineyro@gmail.com
```

Repo con el cual trabajarás



Repo con el cual trabajarás



Clonar el repo

git clone https://github.com/fpineyro/homework-0.git

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ git clone https://github.com/fpineyro/homework-0.git
Cloning into 'homework-0'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 1), reused 2 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), done.
Resolving deltas: 100% (1/1), done.
```

Ingresar al repo clonado

cd <nombre del repo clonado>

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo
$ cd homework-0/

fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo/homework-0 (master)
$ ls
Code.R                                  Curso_Introductorio_a_Python_EDVAI_2.ipynb  README.md  nuevo_file.txt
Curso_Introductorio_a_Python_EDVAI_1.ipynb  Curso_Introductorio_a_Python_EDVAI_3.ipynb  Test       starwars.csv
```

Agregar contenido

cat > nuevo_file.txt

agregamos el contenido y para dejar de agregar info al file, escribimos ctrl + D

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo/homework-0 (master)
$ cat > nuevo_file.txt
agregamos nuevo codigo
para dejar de agregar texto escribimos ctrl + D
```

Agregamos el nuevo contenido

git add <archivo>

git add <directorio>

git add .

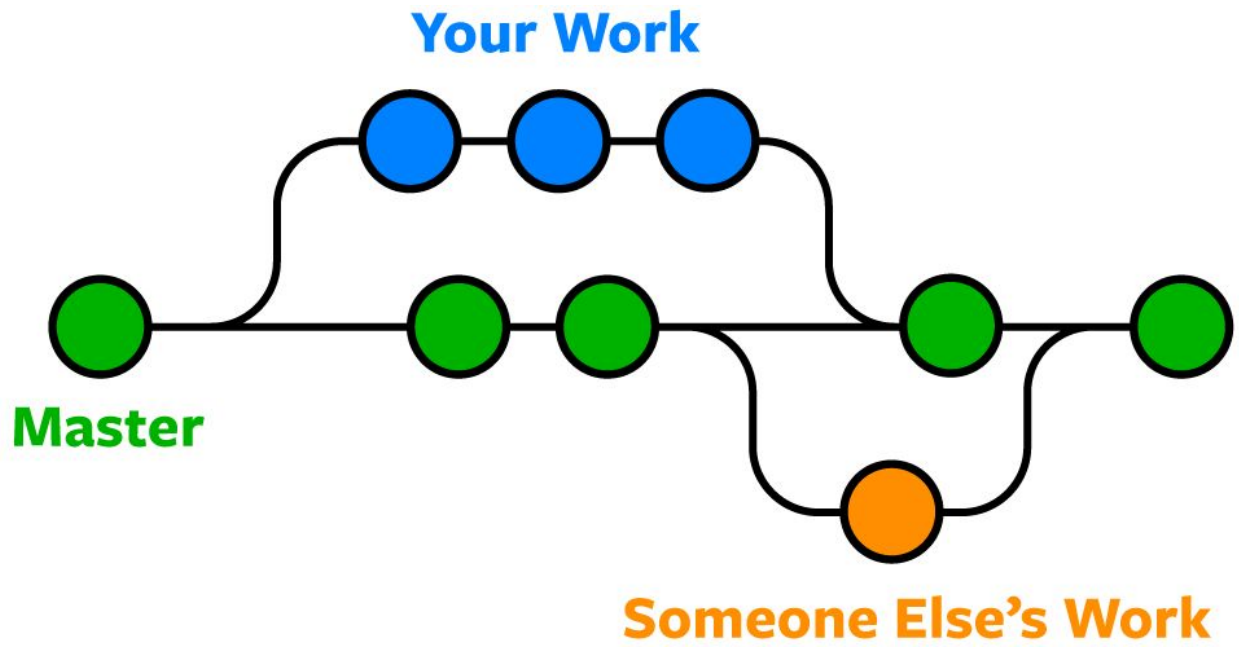
```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo/homework-0 (master)
$ git add .
warning: in the working copy of 'nuevo_file.txt', LF will be replaced by CRLF the next time Git touches it
```

Vemos el status

git status

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo/homework-0 (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    1.txt
        new file:   nuevo_file.txt
```

Branches



Hacemos el commit

git commit -m "comentario que indique que estamos haciendo"

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo/homework-0 (master)
$ git commit -m "agregando nuevo contenido al repo"
[master 0abe0bb] agregando nuevo contenido al repo
 2 files changed, 2 insertions(+), 1 deletion(-)
 delete mode 100644 1.txt
 create mode 100644 nuevo_file.txt
```

Hacemos el git push

git push origin HEAD:master

git push origin master

```
fede@DESKTOP-7PD9EJN MINGW64 ~/Documents/Repo/homework-0 (master)
$ git push origin HEAD:master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 554 bytes | 184.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/fpineyro/homework-0.git
   2080fda..8594b23  HEAD -> master
```

Autenticación

Username: <tu usuario>

Password: pegar tu token (no la pass de Github)

```
fpineyro@BOOK-Fede:~/repositorio/homework-0$ git push origin master
Username for 'https://github.com': fpineyro
Password for 'https://fpineyro@github.com':
```

Ejercitación

Ejercicios

- SQL WF
- Git Hub
- Python

Practica Python

Bajamos los archivos:

Curso_Introductorio_a_Python_EDVAI_1.ipynb

Curso_Introductorio_a_Python_EDVAI_2.ipynb

Curso_Introductorio_a_Python_EDVAI_3.ipynb

De:

https://github.com/fpineyro/homework-0

Practica Python