

Actividad Evaluable

Módulo	Servicios de red
Nombre y Apellidos:	DAVID RECIO RAMIREZ
Nombre y Apellidos:	DOMENICO ROSAS
Nombre y Apellidos:	MIGUEL ANGEL DIAZ
Estas soluciones deben ser entregadas en PDF.	

FECHA DE ENTREGA: --/--/21

Objetivos

El objetivo de esta actividad individual es la de repasar, asentar y adquirir un mayor conocimiento de lo impartido en clase.

El archivo debe nombrarse como: **nombre_apellido1_apellido2.PDF**

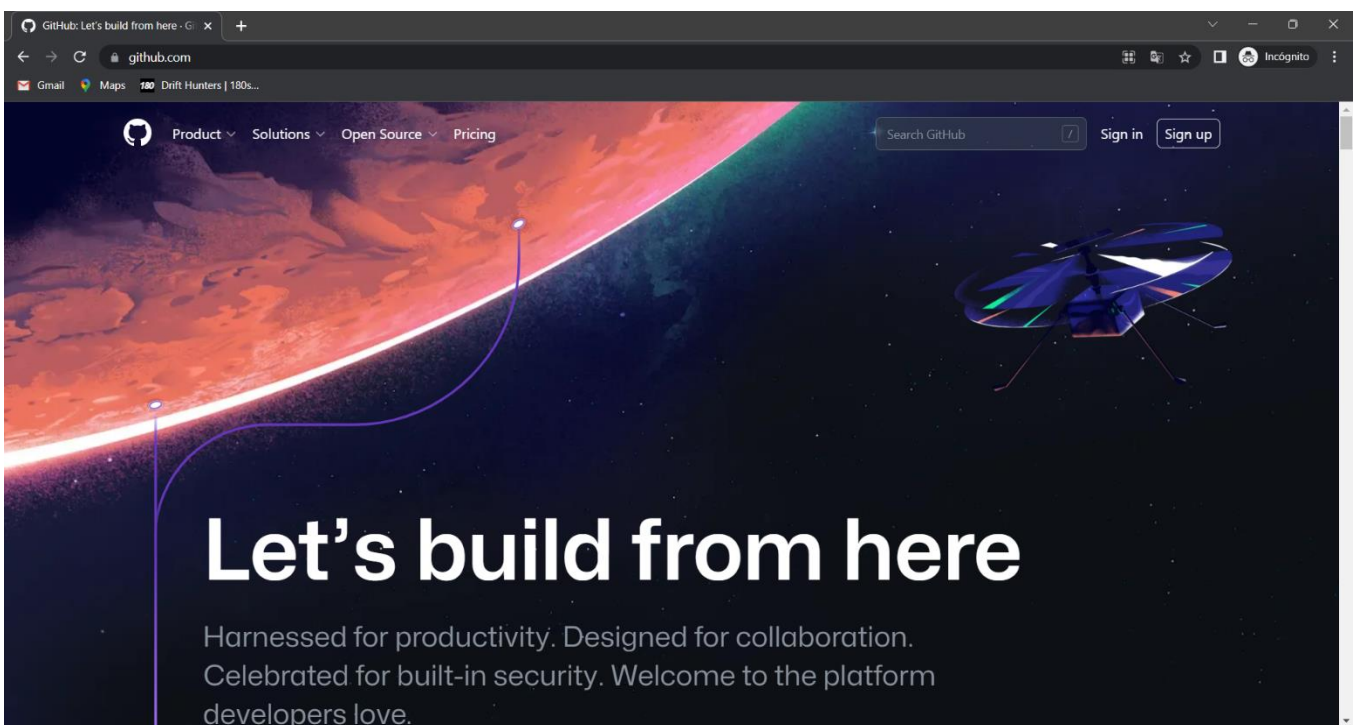
Ej: Belen_Vargas_Sabater.pdf

Cuestiones

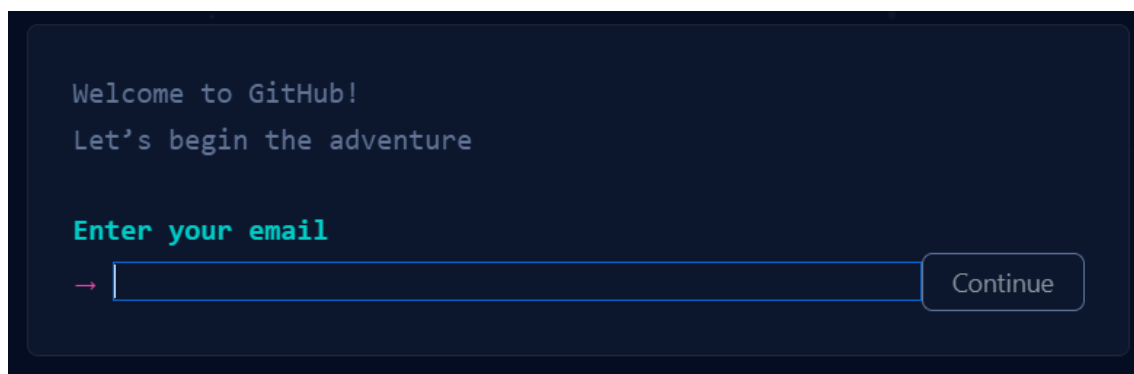
EJERCICIO 1: **Práctica:** Realizar una guía práctica a modo de tutorial, sobre el uso de GitHub, algunos de los elementos que debe cubrir son:

- Creación de repositorio y conexión con repositorio personal de GitHub mediante terminal con ejemplos.

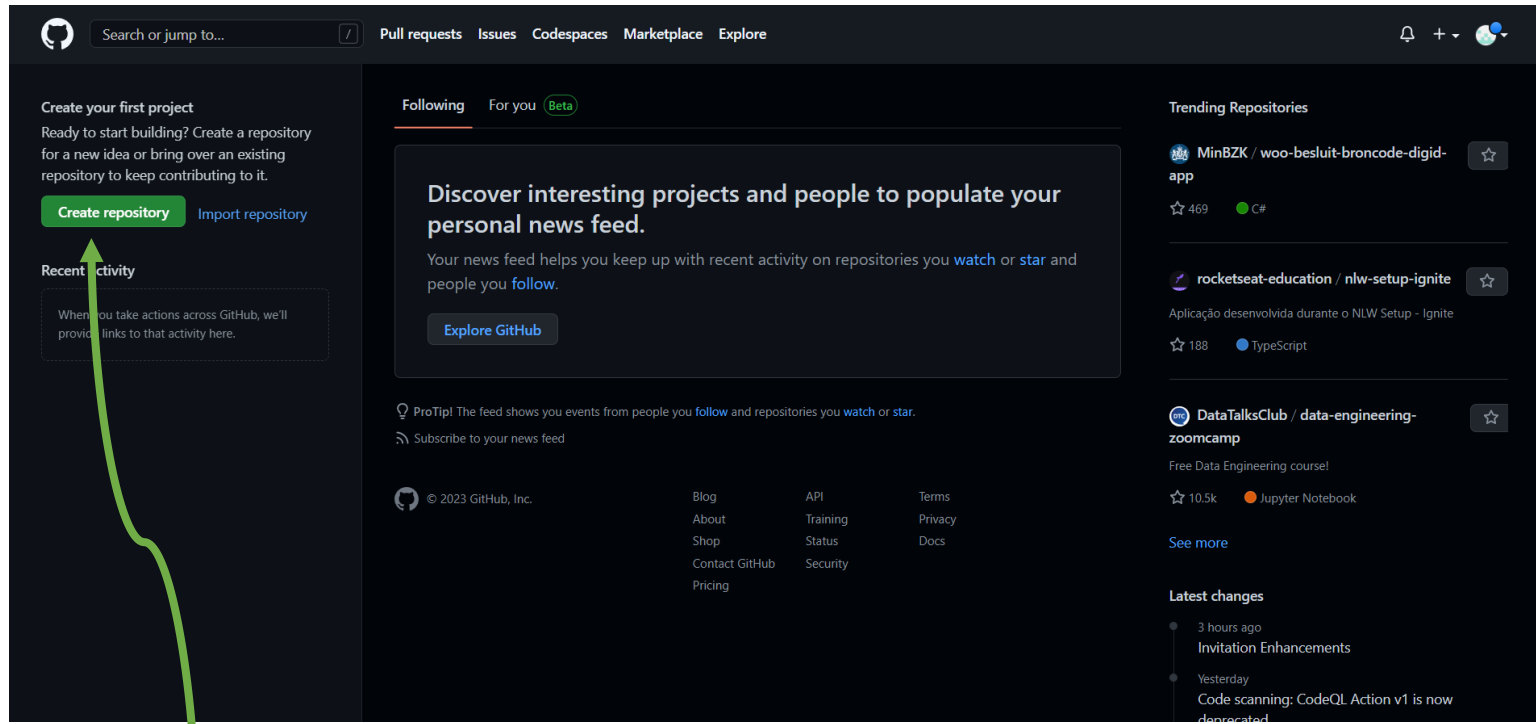
-Lo primero que tenemos que hacer para crear un repositorio es dirigirnos a la web de GitHub → <https://github.com>



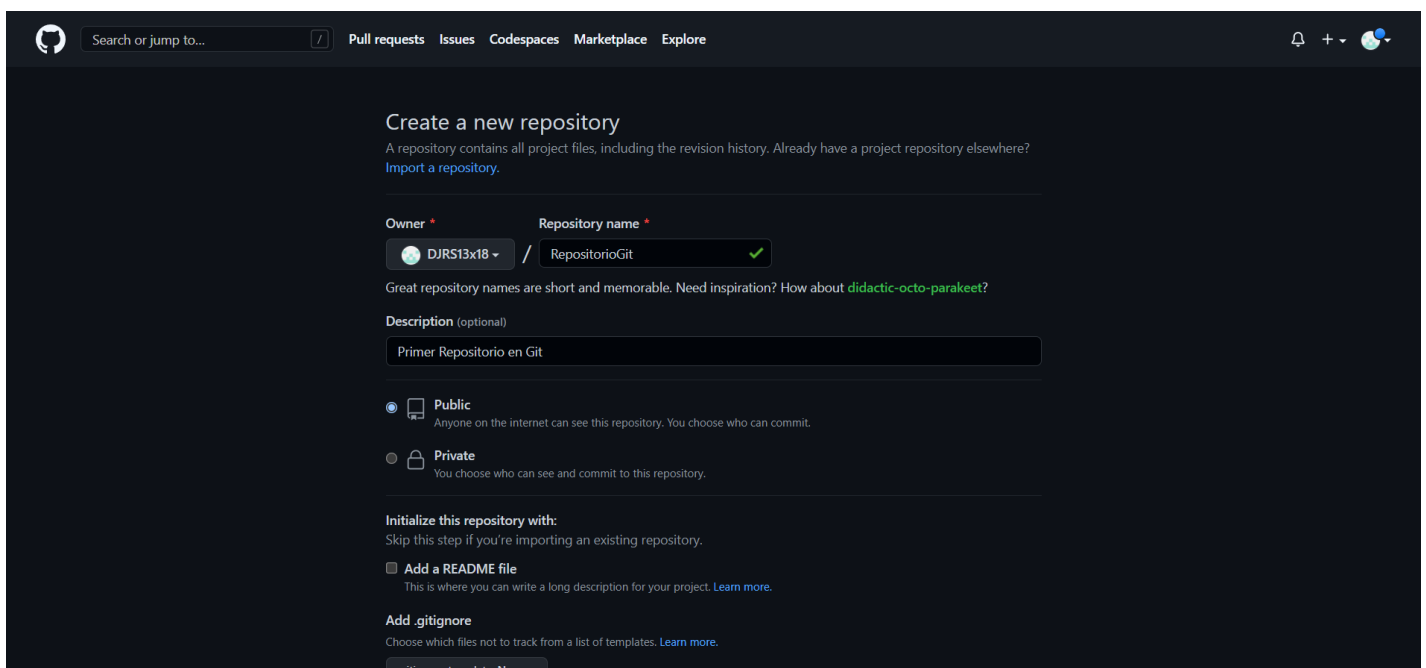
Una vez dentro de la página de GitHub para crear el repositorio necesitaremos una cuenta, para ello haremos click en el botón **Sign up**. Allí dentro nos encontraremos con una interfaz bastante intuitiva y llamativa. Nos pedirán algunos datos para poder gestionar el alta en la página.



Introducimos los datos que nos solicita la página, creamos la cuenta y la verificamos con el código que enviarán al email que se ha usado para crear la cuenta. Una vez que hayamos hecho todo esto la página nos dejará en este punto.



A partir de aquí para crear el repositorio solo tendremos que hacer click en **Create repository**. Nos redireccionará a una pestaña donde crearemos nuestro repositorio, tenemos que colocarle un nombre y una descripción, dependiendo de la finalidad para la que lo vayamos a utilizar.



Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None** ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

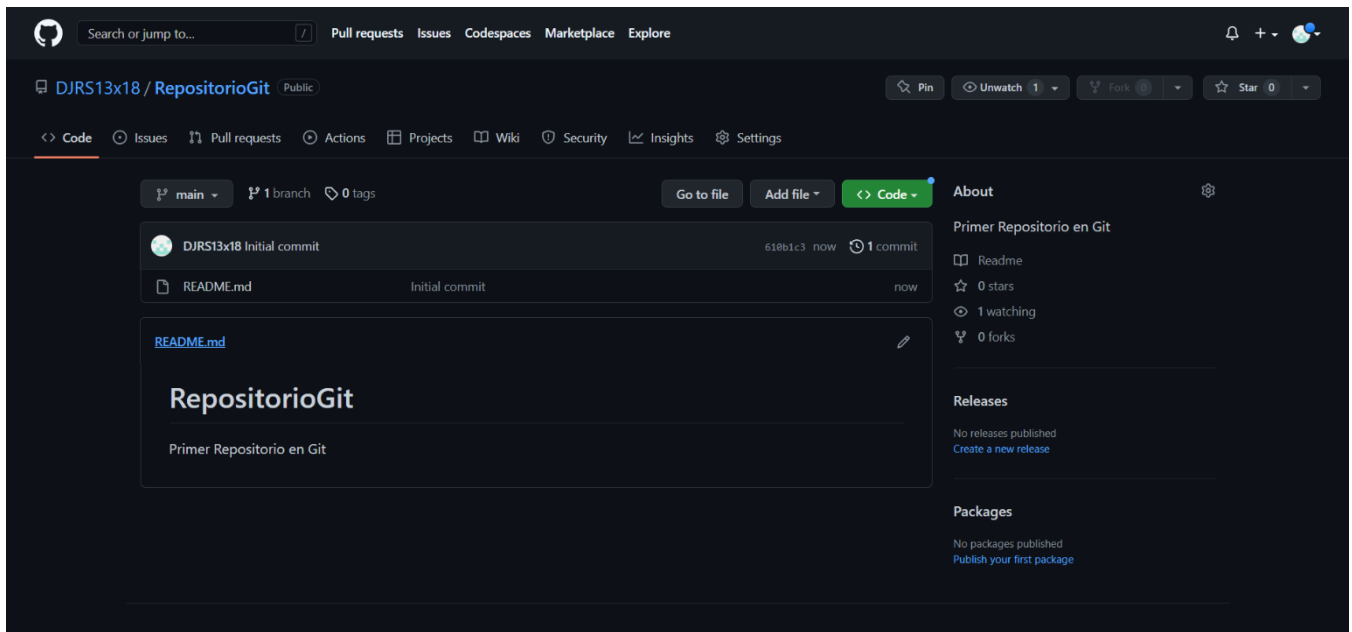
License: **None** ▾

This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.


Create repository

Luego de añadir un nombre y una descripción a nuestro repositorio, hacemos click en [Add a README file](#) este archivo es para escribir una descripción del proyecto dentro del repositorio. y hacemos click en [Create repository](#).

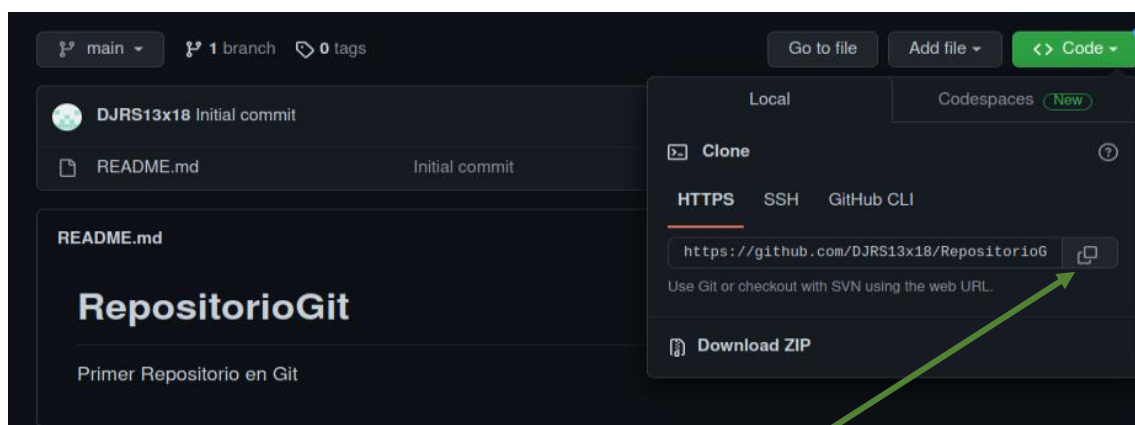


-Seremos redirigidos a nuestro repositorio. Lo siguiente que tenemos que hacer para conectar nuestro repositorio a nuestro terminal y poder subir los cambios desde allí será **clonar nuestro repositorio en el lugar donde querremos trabajar en nuestro proyecto**. En este caso en un portátil de trabajo con Linux como sistema operativo.

-Para ello necesitaremos el **link de nuestro repositorio**, que bien podemos copiarlo desde la URL de la página del mismo.

   <https://github.com/DJRS13x18/RepositorioGit>

La página de GitHub también nos da otra opción en la que tendríamos que hacer click en el botón **Code**.



Y hacer click en el botón que esta al final de la URL.

Una vez tengamos la URL de nuestro repositorio, crearemos una carpeta en la cual clonaremos el repositorio que es lo primero que necesitamos hacer para tener conexión con nuestro repositorio en GitHub y poder subir los cambios y se vean reflejados en la página de GitHub.

```
domenico@domenico-VirtualBox:~/Escritorio$ mkdir CarpetaRepositorioGit
```

Lo siguiente que haremos será entrar en la carpeta para clonar el repositorio dentro de la misma.

```
domenico@domenico-VirtualBox:~/Escritorio$ cd CarpetaRepositorioGit
```

Para clonar el repositorio dentro de la carpeta usaremos el comando git clone añadiendo la URL de nuestro repositorio.

```
$ git clone https://github.com/DJRS13x18/RepositorioGit.git
```

ejecutamos el comando y nos aparecerá lo siguiente. Indicando que nuestro repositorio ha sido clonado con éxito.

```
Clonando en 'RepositorioGit'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Recibiendo objetos: 100% (3/3), listo.
domenico@domenico-VirtualBox:~/Escritorio/CarpetaRepositorioGit$
```

El repositorio estaría ubicado dentro de la carpeta donde lo clonamos.

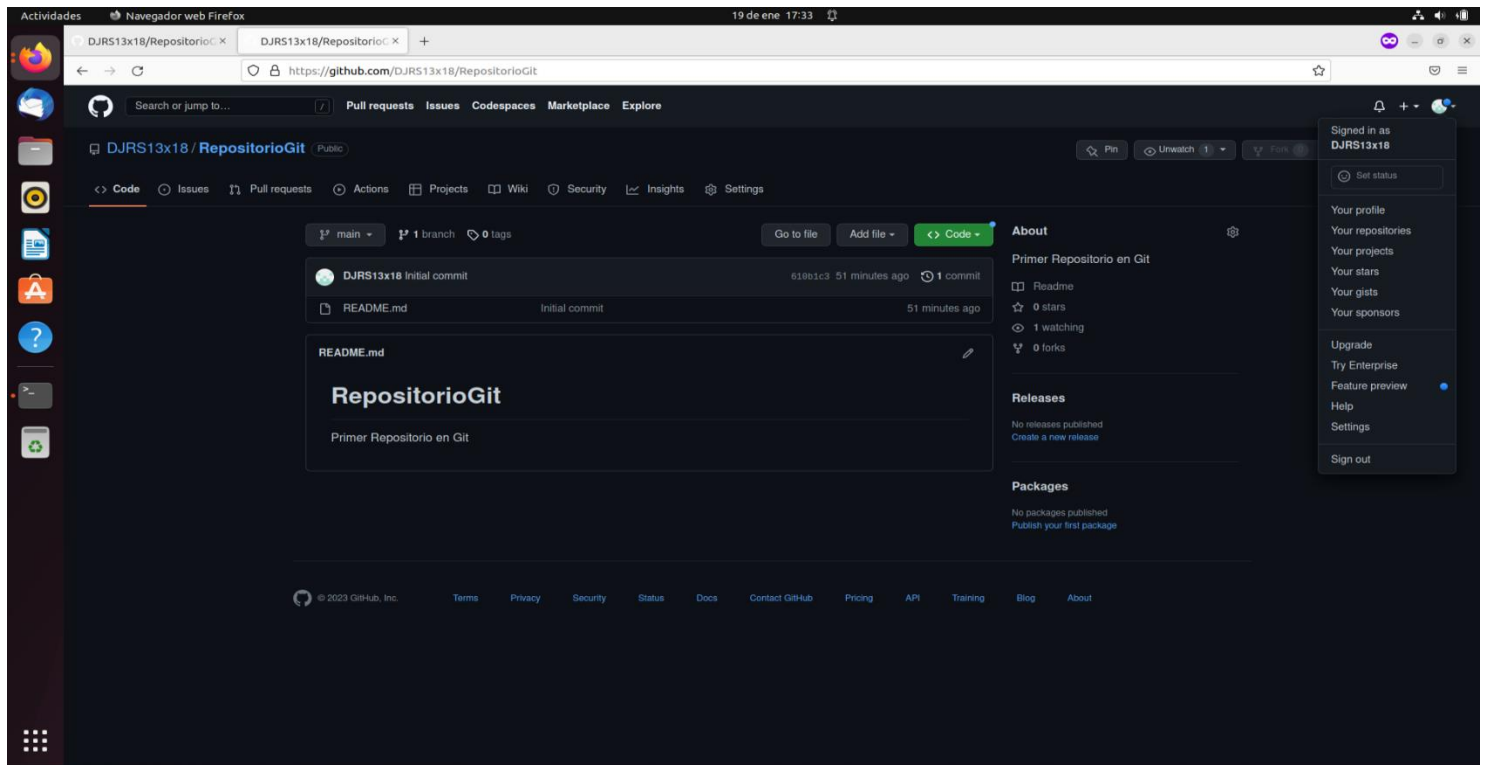
```
domenico@domenico-VirtualBox:~/Escritorio/CarpetaRepositorioGit$ ls
RepositorioGit
domenico@domenico-VirtualBox:~/Escritorio/CarpetaRepositorioGit$
```

Entramos dentro de nuestro repositorio

```
cd RepositorioGit
```

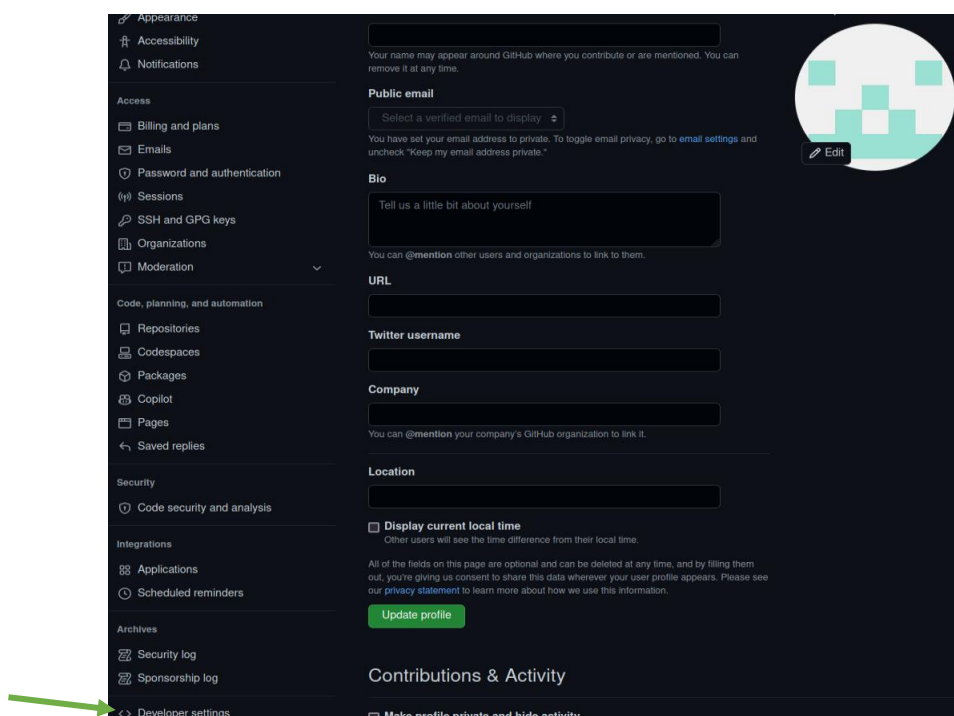
Y dentro estará todo tal cual como en el repositorio central que sería el de GitHub, la clonación es la forma más habitual en la que los desarrolladores obtienen una copia del trabajo.

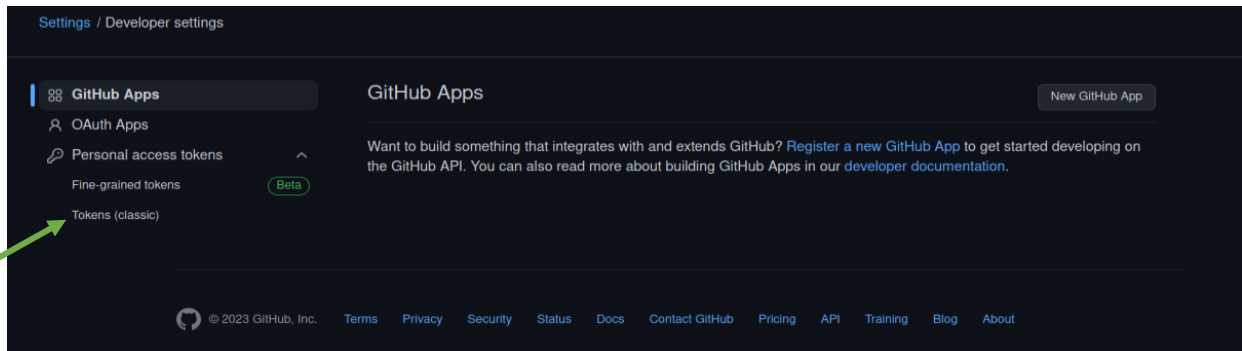
Tendremos que crear un token que será lo que nos dará el acceso remoto a nuestro repositorio. Para ello tendremos que volver a la página de GitHub.



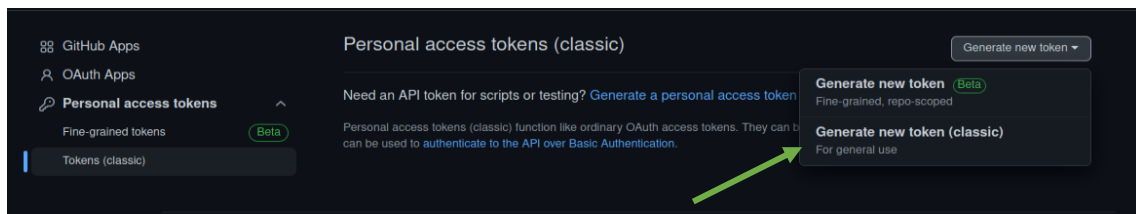
Estando en la página de GitHub hacemos click en la foto de nuestro usuario en la esquina superior derecha de la pantalla, luego hacemos click en el apartado **Settings**.

Dentro del apartado settings tendremos que ir a la última opción. **Developer Settings**.





Dentro del apartado **Developer Settings**, hacemos click en **Personal Access tokens** y luego click en **Tokens (classic)**.



Hacemos click en **Generate new token** y luego click en **Generate new token (classic)**.

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

TokenRepositorio

What's this token for?

Expiration

30 days

The token will expire on Sat, Feb 18 2023

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

- ☒ **repo** Full control of private repositories
 - ☒ repo:status Access commit status
 - ☒ repo_deployment Access deployment status
 - ☒ public_repo Access public repositories
 - ☒ repo:invite Access repository invitations
 - ☒ security_events Read and write security events
- ☐ **workflow** Update GitHub Action workflows
- ☐ **write:packages** Upload packages to GitHub Package Registry
 - ☐ read:packages Download packages from GitHub Package Registry
- ☐ **delete:packages** Delete packages from GitHub Package Registry
- ☐ **admin:org** Full control of orgs and teams, read and write org projects
 - ☐ write:org Read and write org and team membership, read and write org projects
 - ☐ read:org Read org and team membership, read org projects
 - ☐ manage_runners:org Manage org runners and runner groups
- ☐ **admin:public_key** Full control of user public keys

Tendremos que colocarle un nombre al Token, y añadir los **Scopes**.

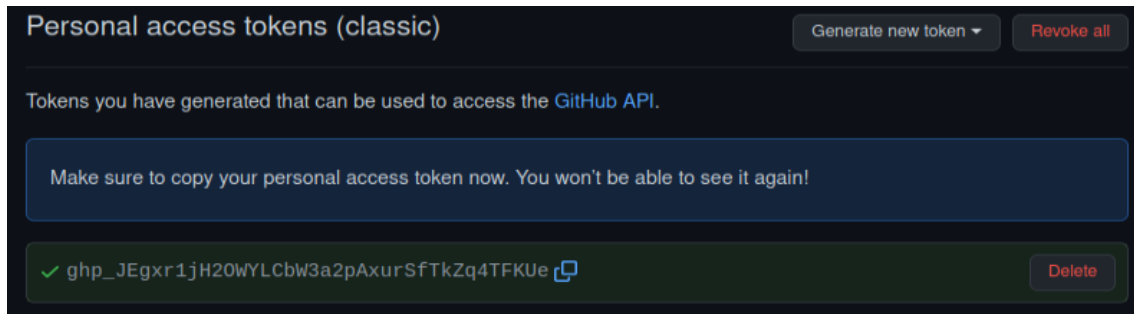
Los **Scopes** son simplemente operaciones que estarán o no permitidas desde este token. En este caso solo seleccionaremos la de **Repo**.

- ☐ **project** Full control of projects
 - ☐ read:project Read access of projects
- ☐ **admin:gpg_key** Full control of public user GPG keys
 - ☐ write:gpg_key Write public user GPG keys
 - ☐ read:gpg_key Read public user GPG keys
- ☐ **admin:ssh_signing_key** Full control of public user SSH signing keys
 - ☐ write:ssh_signing_key Write public user SSH signing keys
 - ☐ read:ssh_signing_key Read public user SSH signing keys

Generate token

Cancel

Al final de los Scopes podremos Generar el token que necesitamos para tener el acceso remoto al repositorio.



Esta sería la token ya generada que necesitamos para tener acceso remoto al repositorio, ahora lo que necesitamos es añadir el acceso remoto en el dispositivo en el cual se desea trabajar en el proyecto. En este caso que sería donde previamente clonamos el repositorio.

Para añadir el acceso remoto usaremos el comando

```
<git remote add [nombre del remoto] [URL del repositorio]>
```

Se vería de la siguiente manera

```
git remote add reporemoto https://github.com/DJRS13x18/RepositorioGit.git
```

Lo siguiente sería verificar que el acceso remoto al repositorio está vinculado correctamente.

```
domenico@domenico-VirtualBox:~/Escritorio/CarpetaRepositorioGit/RepositorioGit$ git remote -v
origin https://github.com/DJRS13x18/RepositorioGit.git (fetch)
origin https://github.com/DJRS13x18/RepositorioGit.git (push)
reporemoto https://github.com/DJRS13x18/RepositorioGit.git (fetch)
reporemoto https://github.com/DJRS13x18/RepositorioGit.git (push)
domenico@domenico-VirtualBox:~/Escritorio/CarpetaRepositorioGit/RepositorioGit$
```

Con el comando `<git remote -v>` podemos ver el acceso remoto que hemos creado con el nombre que le hemos asignado. Y de esta manera podemos **crear un repositorio y conectarlo con nuestro repositorio personal en GitHub mediante la terminal de Linux.**

- Push Y pulls a los repositorios desde el terminal con ejemplos.
- Cada miembro del equipo debe realizar cambios en el repositorio propiedad de uno de los miembros, documenta el proceso.