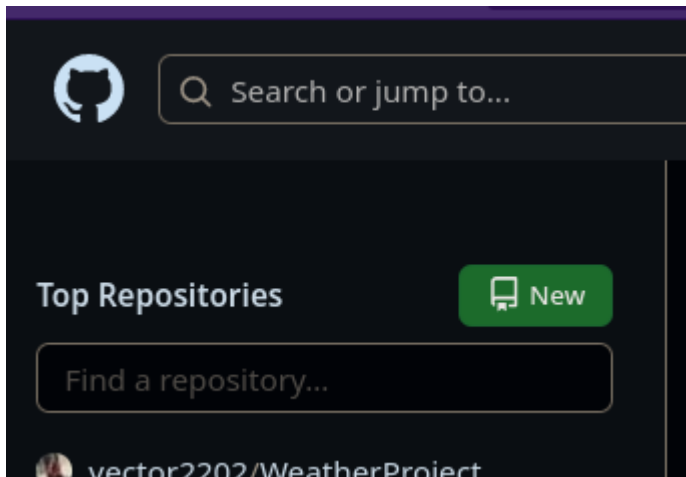


## GITHUB

Github es basicamente un controlador de versiones, sirve para trabajar en equipo que es basicamente todo programar, la mayoría de los proyectos son en equipos y github es un software que te permite trabajar sin por ejemplo estar haciendo zip a tu parte y enviarla a todos y que ellos descarguen y que si te falto algo debas volver a enviarlo, basicamente es como un chat grupal, donde tu subes tus cosas y es visible para todos los que esten en ese chat, todos pueden usar y ver lo que haces asi como puedes usar y ver lo de todos, la ventaja de github tambien es que si borras la carpeta del proyecto o un codigo, lo digo porque me paso, almacena las versiones, cada cambio que tu hiciste se guarda, se hace como una captura y se guarda, si haces otro se guarda tambien, por lo que si en algun momento te equivocaste y quieres volver a la version que si funcionaba, se hace un reset, la ventaja de github igual es que cada quien tiene su "mesa de trabajo", no importa si tu haces un desman en tu mesa, no afectas la de las demas. De igual manera se puede ver quien hizo que y las empresas es algo que te piden, el github y cuando subes muchas cosas a lo largo del tiempo se nota, por lo que yo te recomendaria que cada proyecto de programacion de ahora en adelante lo subas a github, solo para que quede registrado. Bueno ahora vamos a ver como crearlo desde la terminal, no es tan dificil usar la terminal en github, yo la uso y solo he sufrido cuando he hecho reset o cuando hay dos versiones actuales pero no hay tanto problema o bueno espero que esos casos no se den mucho.

Paso 1:

Entras a github y despues de iniciar sesion con tu correo institucional, le das aquí



Despues te aparecera esto:


Actividades Firefox 20 de jun 23:47

https://github.com/new

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk (\*).


Owner \*  vector2202 /

Repository name \*

Great repository names are short and memorable. Need inspiration? How about [expert-guide](#) ?

Description (optional)

☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

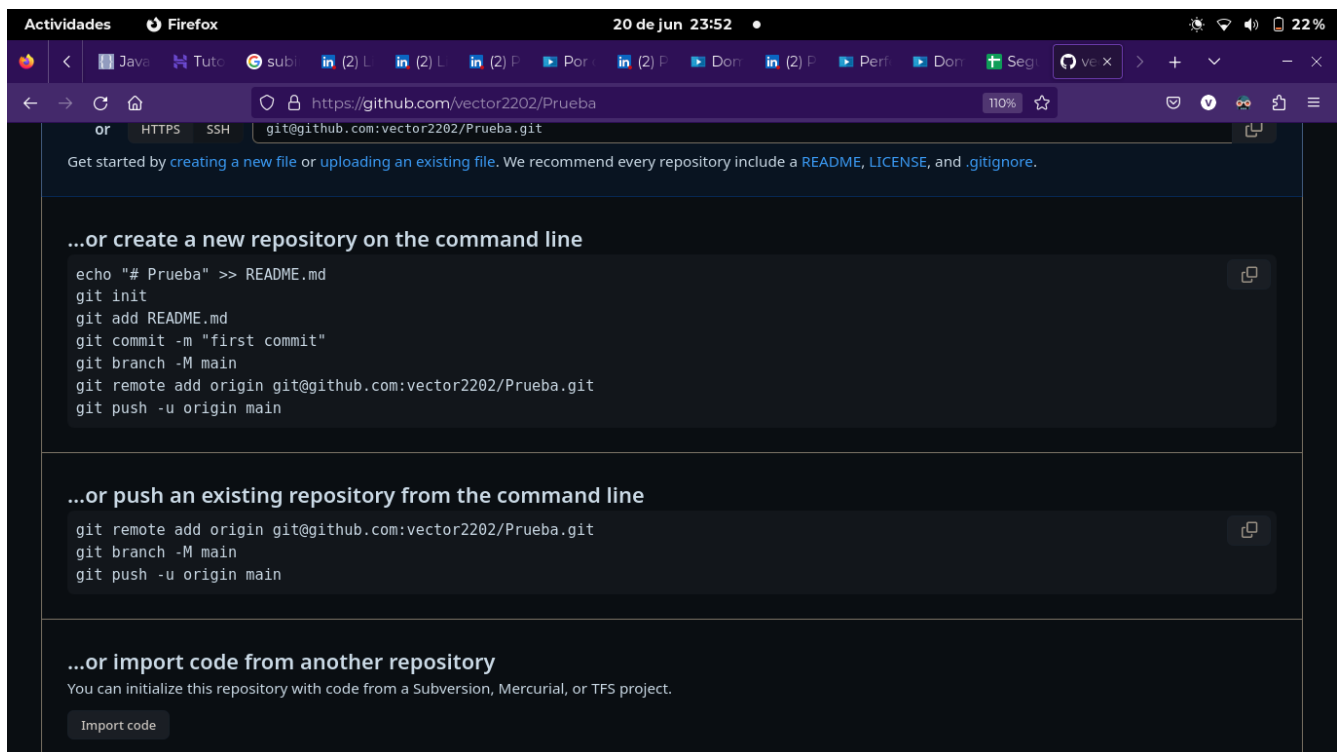
☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

☐ **Add .gitignore**

Le das un nombre a tu repositorio, en este caso le llamaremos Programacion, generalmente tiene que ser el nombre de tu proyecto, como ahorita solo lo iremos usando para subir codigos problemas asi se llamara pero en el caso por ejemplo de la tarea final yo le pondria algo asi como TicketSales o algo asi, una descripcion que le vas a poner una breve descripcion de tu proyecto, en este caso ponle solo codigos de programacion o algo asi, puedes no poner nada tambien

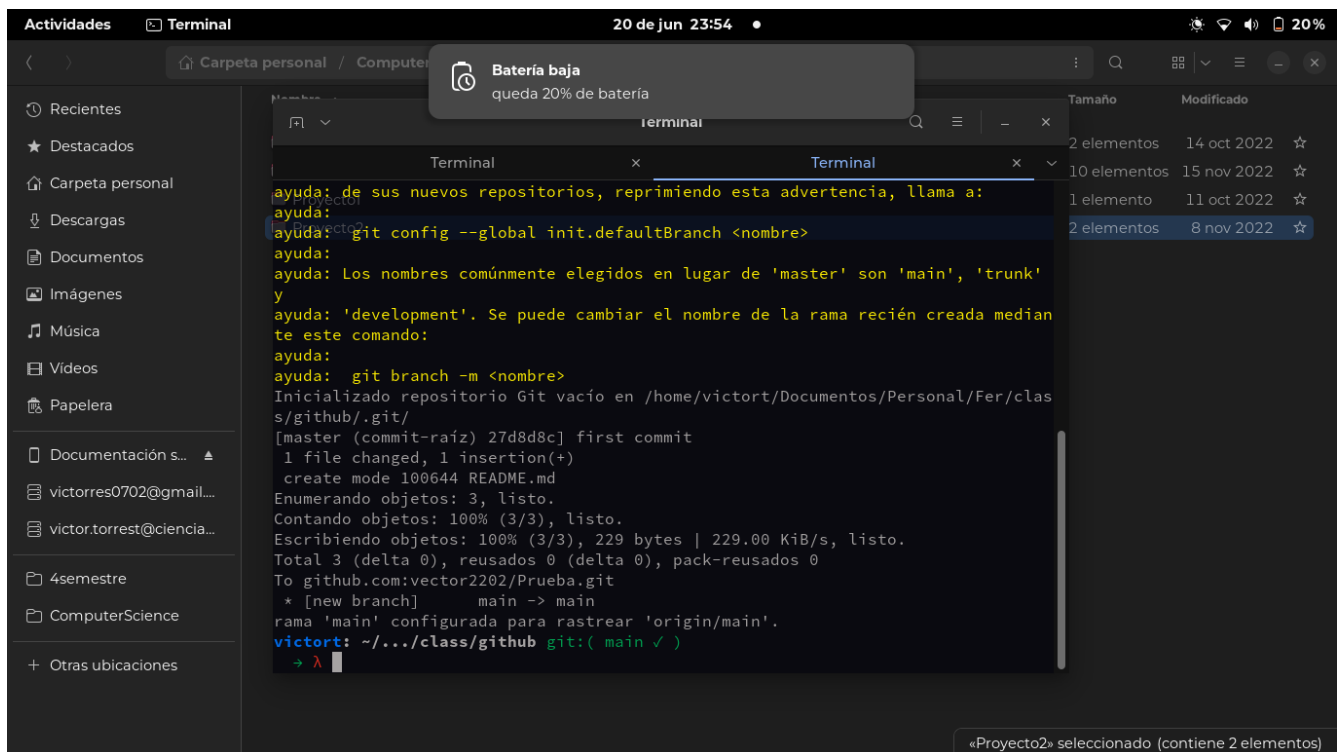
En publico casi siempre se deja, yo solo lo pongo privado cuando es una proyecto de la escuela por la paranoia que no se lo pirateen. Lo demas luego te digo para que sirve. Le das en create repository.

Luego te da dos opciones



La primera es crear un repositorio es decir un folder en tu compu y linkearlo con el actual o si ya tienes código hecho entonces hay que linkearlo con el repositorio actual, por lo general se usa el primero ya que siempre que se inicia un repositorio en github es porque estás iniciando el proyecto, es decir, no tienes ningún código ya hecho, por lo que corres las líneas que te aparecen ahí. Pero antes creas un folder con de preferencia el mismo nombre que tu repositorio, es decir Programacion, una vez dentro de ese folder en la terminal corres las líneas

Después de correr esas líneas te aparece esto



en la terminal, esto quiere decir que tu folder ya esta linkeado con el repositorio

Linea por linea lo que copiaste y ejecutaste es:

```
echo "# Nombre" >> README.md
```

Esto lo que hace es crear un archivo llamado README.md, md es la extension que llevan los readme en github, esto porque ya a la hora de abrir el repositorio se ve con un formato especial, # Es como para indicar que es el titulo.

`git init`

Le indica a la terminal que inicialice git

```
git add README.md
```

Añade README.md a los archivos que se van a subir al repositorio, add lo que hace es agregarlos a como velo como una caja, que vamos a enviar, add los mete en esa caja, por lo general yo uso: `git add -A` que quiere decir todos los archivos modificados agregalos a la caja,

```
git commit -m "first commit"
```

`git commit` empaqueta la caja y le pone una nota, el `-m` indica que vas a escribir la nota y entre comillas va el mensaje, en este caso le pone first commit porque es la primera "entrega", por lo general entre comillas va los cambios que estas subiendo, por ejemplo `archivo.cpp` funcionando bien, o la clase Punto ya funciona, o le agregamos el menu, corregimos el error tal, se agrego tal funcionalidad, mensajes breves pero concisos.

```
git branch -M main
```

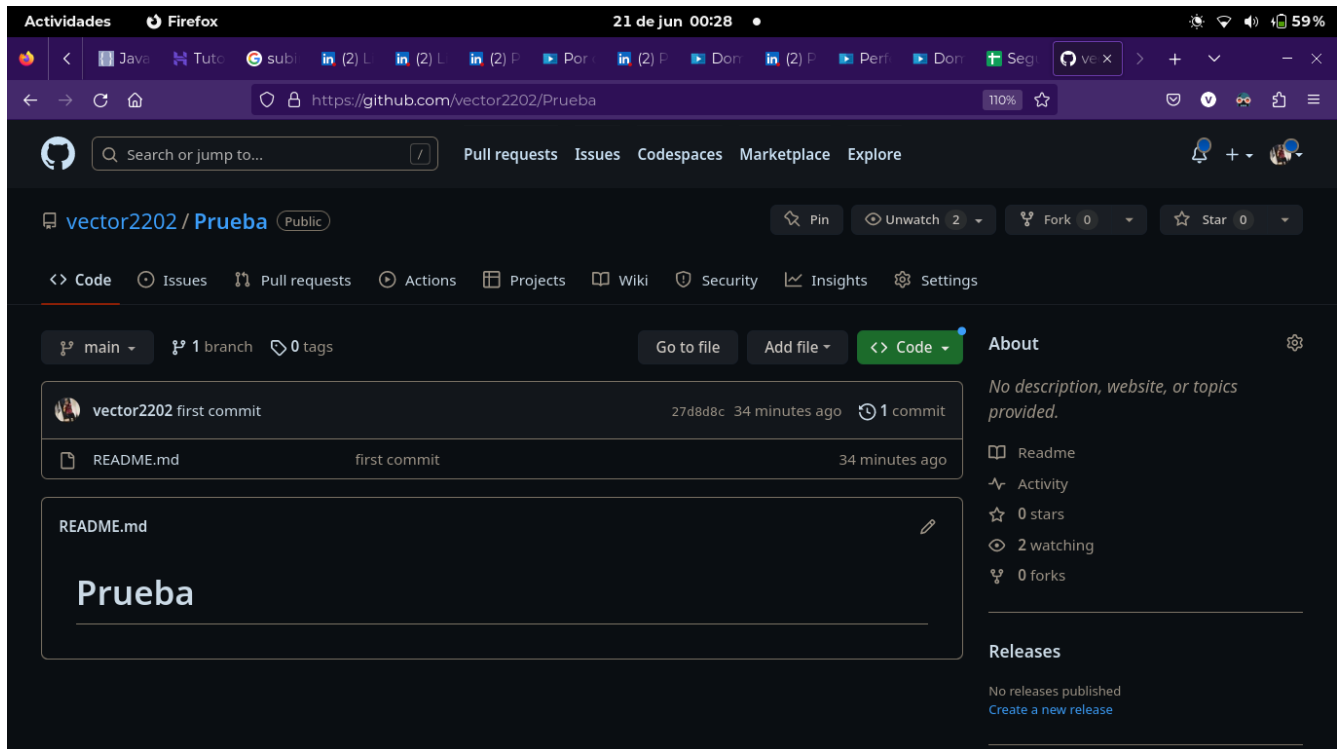
Este solo renombra la rama principal, porque por default se llama master pero lo consideraron racista y ahora es main.

```
git remote add origin git@github.com:vector2202/Prueba.git
```

Linkea el repo con tu folder

git push -u origin main

Mete la "caja" en el repositorio, todos los cambios que hiciste y etiquetaste los mete en linea por asi decirlo.



Asi se deberia ver despues de refrescar, y listo asi se crea un repositorio desde terminal.

Ahora lo que puedes hacer es crear archivos en ese folder donde estas, git add -A, git commit -m "mensaje" y git push origin y con eso deberian de actualizarse los archivos, si modificas archivos igual, si eliminas igual, puedes intentar hacerlo y veras como si cambia, para ver las diferencias entre lo que tienes online y lo que tienes en tu compu:  
git diff