

Actividad Evaluable

| Módulo | Servicios de red |
|---|-----------------------|
| Nombre y Apellidos: | Iván Sánchez Bonacasa |
| 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**
- **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.**

Antes de proceder con la práctica, quisiera explicar para así entender qué es y en qué consiste el famoso **GitHub**. Github es ni más ni menos que un portal creado para alojar el código de las aplicaciones de cualquier desarrollador. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones o herramientas, y que como usuario no sólo puedas descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.

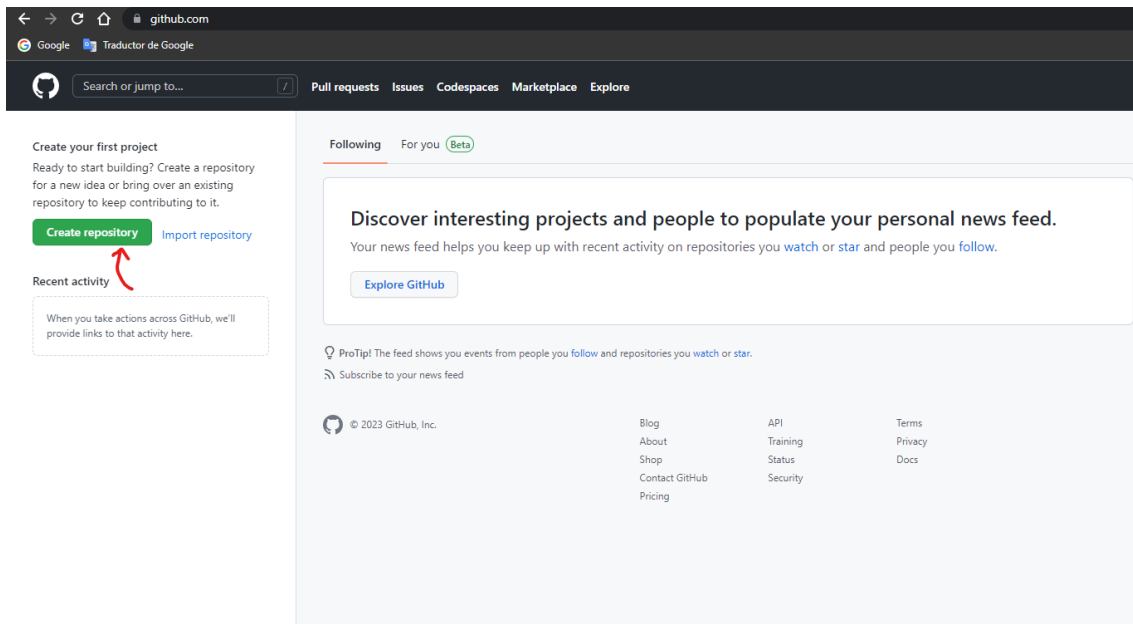
Como su nombre indica, la web utiliza el sistema de control de versiones con el que los **desarrolladores pueden administrar su proyecto**, ordenando el código de cada una de las nuevas versiones que sacan de sus aplicaciones para evitar confusiones. Así, al tener copias de cada una de las versiones de su aplicación, no se perderán los estados anteriores cuando se vaya a actualizar.

En resumidas cuentas, Git es uno de estos sistemas de control que permite comparar el código de un archivo para **ver las diferencias entre las versiones**, restaurar versiones antiguas si algo sale mal, y fusionar los cambios de distintas versiones. También permite trabajar con distintas ramas de un proyecto, como la de desarrollo para meter nuevas funciones al programa o la de producción para depurar los bugs.



Ahora, a modo de guía o tutorial desarrollaré el proceso proceso por el cual puedo crear un repositorio en la plataforma mencionada donde podré subir cualquier contenido/repositorio personal a través de una terminal.

En primer lugar, necesitamos crearnos una cuenta de Github para poder crear el repositorio y subir contenido al mismo. Para ello, nos dirigimos a la página principal (<https://www.github.com>) y hacemos clic en **Sign Up**. Una vez nos hayamos creado la cuenta, iniciamos sesión y seleccionamos la opción de **Create Repository**:




El siguiente paso será configurar el nuevo repositorio que vamos a crear, empezando por elegir su nombre. Un dato muy importante a tener en cuenta (según lo que he leído), es fundamental que el nombre del repositorio sea igual al nombre de la carpeta que vamos a subir al mismo.

En mi caso, el repositorio lo llamaré **“pruebas”** para más adelante nombrar así a la carpeta que crearemos. Aparte, se puede elegir si establecer el repositorio como público o privado, elegir una licencia, entre otras cosas.

Create a new repository

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


Owner * Repository name *


 ivansuitoo / pruebas ✓

Great repository names are short and memorable. Need inspiration? How about [curly-octo-chainsaw?](#)

Description (optional)

Esto es una práctica de Iván Sánchez para presentar en redes de 2º de ASIR

☒  **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.

Create a new repository

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

Owner *



ivansuitoo

Repository name *

pruebas



Great repository names are short and memorable. Need inspiration? How about [curly-octo-chainsaw?](#)

Description (optional)

Esto es una práctica de Iván Sánchez para presentar en redes de 2º de ASIR



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:

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

You are creating a public repository in your personal account.

Create repository



Vemos que al crearlo nos proporciona un enlace, ese link será el que nos permita vincular el contenido a subir con el repositorio. Además, el propio GitHub nos informa de los comandos básicos a seguir para proceder con esta práctica.

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

Set up in Desktop

or

HTTPS

SSH

<https://github.com/ivansuitoo/pruebas.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 "# pruebas" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ivansuitoo/pruebas.git
git push -u origin main
```



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

```
git remote add origin https://github.com/ivansuitoo/pruebas.git
git branch -M main
git push -u origin main
```



Ahora toca moverse al entorno virtual, en mi caso un Ubuntu 22.04 donde crearé el directorio que deseo subir a la plataforma. Nos dirigimos al escritorio o a la ruta donde quieras tener almacenada la carpeta, y la creamos con el comando **mkdir nombre**, recordando que debe ser igual que el nombre del repositorio recientemente creado en GitHub.

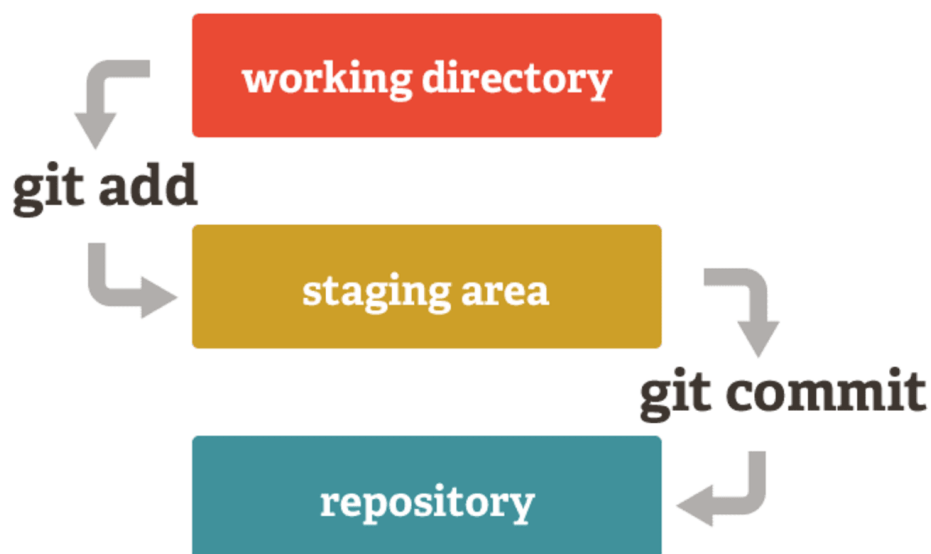
```
ivan@ivan-VirtualBox:~/Escritorio$ mkdir pruebas
ivan@ivan-VirtualBox:~/Escritorio$ ls
pruebas
ivan@ivan-VirtualBox:~/Escritorio$
```

Acto seguido creo en su interior un documento de texto en el cual escribiré cualquier cosa para que la carpeta tenga algo de contenido. Creado el fichero de extensión .txt, procedo a escribir “**git init**” para convertir la carpeta existente en un repositorio Git.

```
ivan@ivan-VirtualBox:~/Escritorio$ mkdir pruebas
ivan@ivan-VirtualBox:~/Escritorio$ ls
pruebas
ivan@ivan-VirtualBox:~/Escritorio$ cd pruebas/
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ touch a.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ nano a.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ cat a.txt
esto es el primer contenido que creo antes de hacer un commit
ivan@ivan-VirtualBox:~/Escritorio/pruebas$
```

```
ivan@ivan-VirtualBox:~/Escritorio$ mkdir pruebas
ivan@ivan-VirtualBox:~/Escritorio$ ls
pruebas
ivan@ivan-VirtualBox:~/Escritorio$ cd pruebas/
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ touch a.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ nano a.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ cat a.txt
esto es el primer contenido que creo antes de hacer un commit
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git init
Iniciado repositorio Git vacío en /home/ivan/Escritorio/pruebas/.git/
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ ls -a
.  ..  a.txt  .git
ivan@ivan-VirtualBox:~/Escritorio/pruebas$
```

Ahora que tenemos la carpeta como un repositorio local, si hacemos un **git status -s** para que nos informe del estado de los archivos de nuestro proyecto y que liste los archivos en verde o en rojo. Los archivos en verde se han añadido al “**staging area**” pero no se han confirmado todavía. Los archivos marcados en rojo son aquellos que no han sido añadidos al “**staging area**”. Al realizar el comando mencionado, veremos que debemos añadir el archivo creado para que esté preparado, para ello hacemos un **git add .** (con el “.” especificamos que queremos añadir todos los archivos de la carpeta en la que estamos).



```

ivan@ivan-VirtualBox:~/Escritorio$ mkdir pruebas
ivan@ivan-VirtualBox:~/Escritorio$ ls
pruebas
ivan@ivan-VirtualBox:~/Escritorio$ cd pruebas/
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ touch a.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ nano a.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ cat a.txt
esto es el primer contenido que creo antes de hacer un commit
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git init
Iniciado repositorio Git vacío en /home/ivan/Escritorio/pruebas/.git/
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ ls -la
.  ..  a.txt  .git
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git status -s
?? a.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git add .
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git status -s
A  a.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$
  
```

Una vez añadidos los cambios al **staging area** queremos confirmarlos en nuestro repositorio mediante un **git commit -m "Comentario"**. Al ser la primera vez que lo hacemos en ese repositorio, debemos configurar un correo y nombre de usuario para completar el commit correctamente:

```
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git commit -m "Este es mi primer commit desde linea de comandos"

*** Por favor cuéntame quién eres.

Ejecuta

  git config --global user.email "you@example.com"
  git config --global user.name "Tu Nombre"

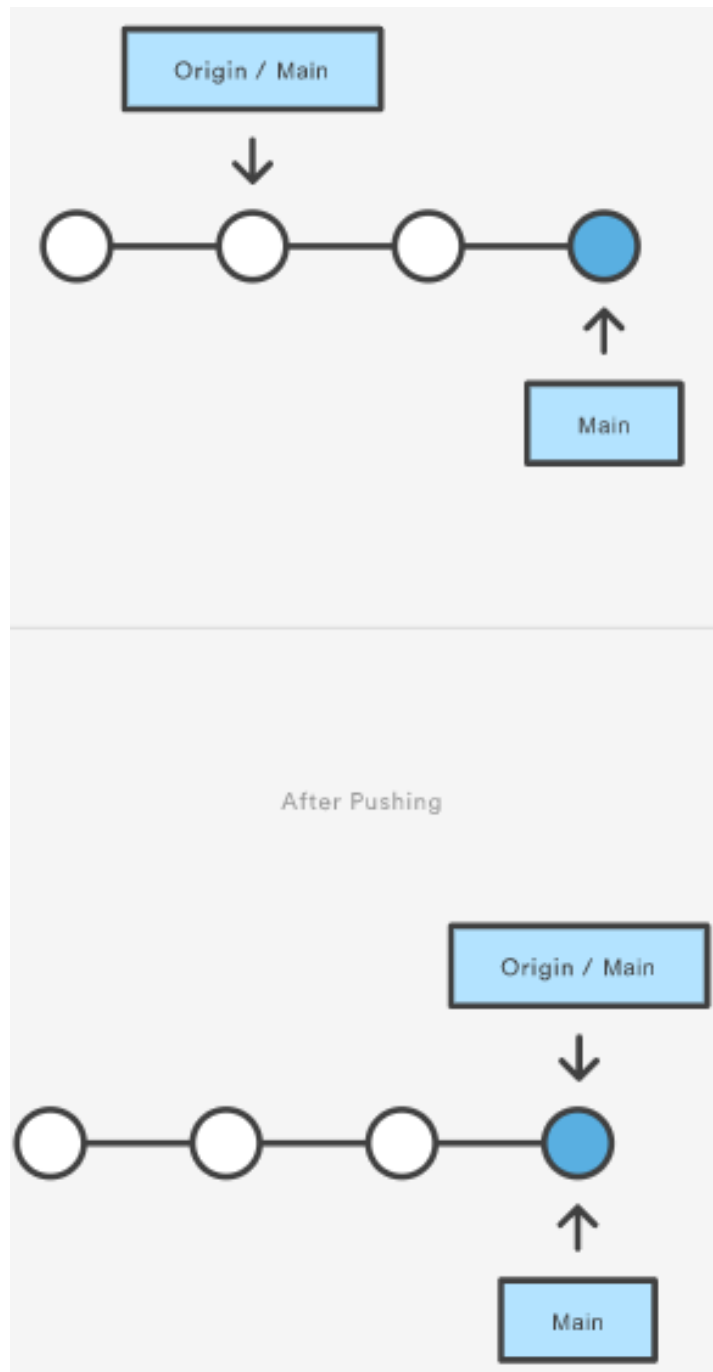
para configurar la identidad por defecto de tu cuenta.
omite --global para configurar tu identidad solo en este repositorio.

fatal: no es posible auto-detectar la dirección de correo (se obtuvo 'ivan@ivan-VirtualBox.(none)')
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git config --global user.email "ivansanchezbona@gmail.com"
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git config --global user.name "ivansuitchoo"
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git commit -m "Este es mi primer commit desde linea de comandos"
[master (commit-raiz) 5b61428] Este es mi primer commit desde linea de comandos
 1 file changed, 1 insertion(+)
 create mode 100644 a.txt
```

El siguiente paso será a partir del comando **git remote add origin (link de mi github)**. El comando git remote te permite crear, ver y eliminar conexiones con otros repositorios, en este caso lo usaremos para ver y añadir una conexión remota con el repositorio que creamos al principio de la práctica.

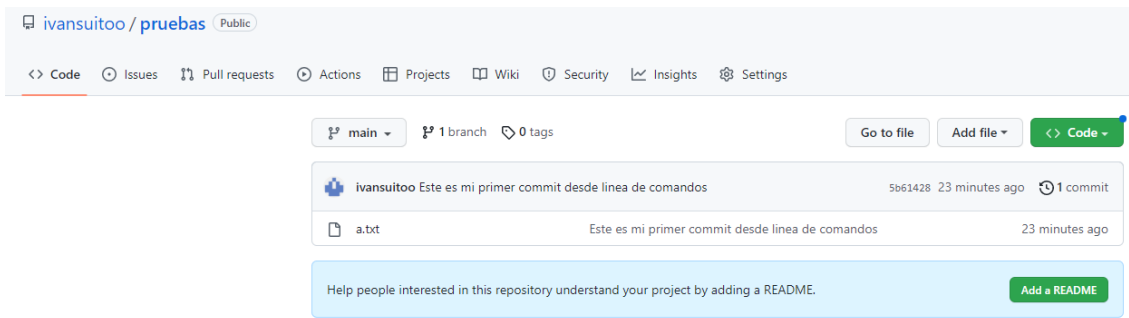
```
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git remote add origin https://github.com/ivansuitchoo/pruebas.git
ivan@ivan-VirtualBox:~/Escritorio/pruebas$
```

Por último, queremos subir lo que hemos creado o añadido a nuestro repositorio, por lo que utilizaremos la sentencia de comando que nos indica en la propia página de GitHub **git push -u origin main**. El comando git push se usa para cargar contenido del repositorio local a un repositorio remoto, es decir, el envío es la forma de transferir confirmaciones desde tu repositorio local a un repositorio remoto.



```
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git push -u origin main
Username for 'https://github.com': ivansuitoo
Password for 'https://ivansuitoo@github.com':
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Compresión delta usando hasta 2 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 286 bytes | 286.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
To https://github.com/ivansuitoo/pruebas.git
 * [new branch]      main -> main
Rama 'main' configurada para hacer seguimiento a la rama remota 'main' de 'origin'.
```

Actividad Evaluable



Respecto al siguiente ejercicio, si soy miembro del equipo del proyecto y quisiera bajarme el repositorio para cambiar algo desde mi equipo, debo utilizar el comando **git pull** (El comando git pull se emplea para extraer y descargar contenido desde un repositorio remoto y actualizar al instante el repositorio local para reflejar ese contenido).

Para el ejemplo me he creado otra carpeta en el escritorio a la que he llamado pruebas2, dentro de ella con un **git clone (link de mi gitHub)** me he bajado el repositorio y con **git pull** me puedo bajar cualquier actualización del mismo que se haya subido. Es decir, si yo estoy trabajando en el repositorio actualizando su contenido, dichas modificaciones las subo y otra persona de mi equipo hace un git pull se bajaría el contenido con las actualizaciones que mencionaba. Por último, con otra modificación que yo cree como miembro del equipo la subo de la misma manera que había desarrollado anteriormente, añadiendo los cambios al *staging area*, confirmando los en el repositorio local y subiéndolos con **git push**.

```
ivan@ivan-VirtualBox:~/Escritorio/pruebas2$ git clone https://github.com/ivansuitoo/pruebas.git
Clonando en 'pruebas'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (3/3), 266 bytes | 266.00 KiB/s, listo.
ivan@ivan-VirtualBox:~/Escritorio/pruebas2$ ls
pruebas
ivan@ivan-VirtualBox:~/Escritorio/pruebas2$ git remote
ivan@ivan-VirtualBox:~/Escritorio/pruebas2$ cd pruebas/
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git remote
origin
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git pull
Ya está actualizado.
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git branch
* main
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ sudo git pull origin main
Desde https://github.com/ivansuitoo/pruebas
* branch      main      -> FETCH_HEAD
Ya está actualizado.
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$
```

```
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ sudo git pull origin main
Desde https://github.com/ivansuitoo/pruebas
* branch          main      -> FETCH_HEAD
Ya está actualizado.
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ ls
a.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ touch b.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ nano b.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git add b.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git commit -m "Este es un co
mmmit hecho por otro miembro del equipo"
[main 9b7e074] Este es un commit hecho por otro miembro del equipo
 1 file changed, 1 insertion(+)
 create mode 100644 b.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git remote add origin https:
//github.com/ivansuitoo/pruebas.git
fatal: remoto origin ya existe.
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git push -u origin main
Username for 'https://github.com': ivansuitoo
Password for 'https://ivansuitoo@github.com':
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 2 hilos
```

```
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git add b.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git commit -m "Este es un co
mmmit hecho por otro miembro del equipo"
[main 9b7e074] Este es un commit hecho por otro miembro del equipo
 1 file changed, 1 insertion(+)
 create mode 100644 b.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git remote add origin https:
//github.com/ivansuitoo/pruebas.git
fatal: remoto origin ya existe.
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ git push -u origin main
Username for 'https://github.com': ivansuitoo
Password for 'https://ivansuitoo@github.com':
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 2 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 348 bytes | 348.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
To https://github.com/ivansuitoo/pruebas.git
 5b61428..9b7e074  main -> main
Rama 'main' configurada para hacer seguimiento a la rama remota 'main' de 'origi
n'.
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$
```

ivansuitoo / pruebas (Public)

[Code](#)
[Issues](#)
[Pull requests](#)
[Actions](#)
[Projects](#)
[Wiki](#)
[Security](#)
[Insights](#)
[Settings](#)

main 1 branch 0 tags

Go to file
Add file
Code

ivansuitoo

Este es un commit hecho por otro miembro del equipo

9b7e074 2 minutes ago 2 commits

| | | |
|-------|---|----------------|
| a.txt | Este es mi primer commit desde línea de comandos | 51 minutes ago |
| b.txt | Este es un commit hecho por otro miembro del equipo | 2 minutes ago |

Help people interested in this repository understand your project by adding a README.

Add a README

Ahora, para demostrar lo mencionado en el anterior párrafo, si yo me voy a la anterior carpeta donde solo tenía creado el primer archivo y ejecuto un **git pull** se me bajarán las modificaciones hechas actualizando mi repositorio local. De esta manera, podríamos trabajar en equipo para realizar un proyecto.

```
ivan@ivan-VirtualBox:~/Escritorio/pruebas2/pruebas$ cd ..
ivan@ivan-VirtualBox:~/Escritorio/pruebas2$ cd ..
ivan@ivan-VirtualBox:~/Escritorio$ cd pruebas
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ ls
a.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (3/3), 328 bytes | 328.00 KiB/s, listo.
Desde https://github.com/ivansuitoo/pruebas
    5b61428..9b7e074  main      -> origin/main
Actualizando 5b61428..9b7e074
Fast-forward
 b.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 b.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$ ls
a.txt b.txt
ivan@ivan-VirtualBox:~/Escritorio/pruebas$
```