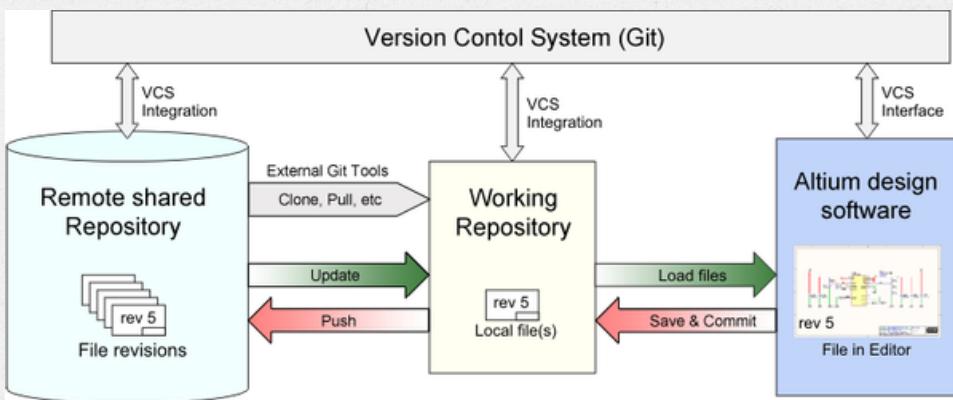


Manual de

GITHUB



Es un software de control de versiones gratis y de código abierto. Fue creado por Linus Torvalds en 2005. Es un rastreador de contenido, puede ser utilizado para almacenar contenido y se usa principalmente para almacenar código debido a otras características que proporciona.



GIT FACILITA

La contribución a proyectos de código abierto para gestionar sus proyectos. Al igual que facilita la búsqueda de trabajo ya que la mayoría de las compañías miran los perfiles de GitHub.

GIT DOCUMENTA

Su sección de ayuda y las guías tienen artículos para casi cualquier tema relacionado a Git en el que puedas pensar.

GIT INTEGRA

GitHub puede integrarse con plataformas comunes como Amazon y Google Cloud, para rastrear tus comentarios y puede resaltar la sintaxis en lenguajes de programación diferentes.

GIT RASTREA

Cuando varias personas colaboran en un proyecto, es difícil mantener el seguimiento de las revisiones. GitHub se ocupa de este problema.

1 CUENTA

Para crear una cuenta, tienes que ir al sitio web de GitHub y llenar la forma de registro.



2 INSTALACIÓN

Tendremos que instalar las herramientas de Git en nuestra computadora. Utilizaremos CLI (interfaz de línea de comandos) para comunicarnos con GitHub.

3 PARA UBUNTU

Actualiza tus paquetes:

```
sudo apt update
```

4 PARA UBUNTU

Instala Git y GitHub con apt-get:

```
sudo apt-get install git
```

5 PARA UBUNTU

Verifica que Git se instaló correctamente:

```
git --version
```

6 PARA UBUNTU

Ejecuta los siguientes comandos con tu información para establecer un nombre de usuario y un correo electrónico predeterminados para cuando vayas a salvar tu trabajo.

```
git config --global user.name "MV Thanoshan"  
git config --global user.email "example@mail.com"
```

7 PROYECTOS

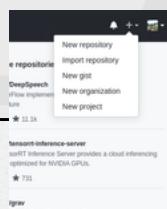
Trabajaremos con proyectos GitHub en dos maneras. **Tipo 1:** Crear el repositorio, clonarlo en tu PC y trabajar en él. (Recomendado) y **Tipo 2:** Trabajar en tu proyecto localmente y después crear el repositorio en GitHub y enviarlo a remote.

8 TIPO 1

Involucra la creación de un repositorio totalmente nuevo en GitHub, clonarlo en nuestra computadora, trabajar en nuestro proyecto y enviarlo de regreso.

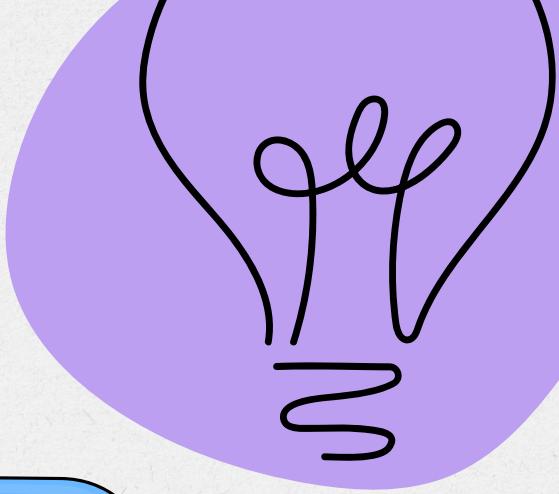
9 TIPO 1

Crea un nuevo repositorio haciendo clic en el botón de "Nuevo repositorio" en la página web de GitHub.



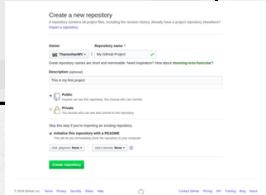
10 TIPO 1

Elije un nombre para tu primer repositorio, agrega una pequeña descripción,



11 TIPO 1

Marca la opción "Inicializar este repositorio con un README", y haz clic en el botón "Crear Repositorio".

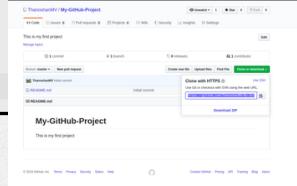


12 TIPO 1

Obtener una copia del repositorio en tu computadora. Para hacer eso, necesitas "clonar" el repositorio en tu computadora.

13 TIPO 1

En la página del repositorio, necesitas obtener la dirección "HTTPS".



14 TIPO 1

Usa el siguiente comando en tu terminal. Cuando estés listo puedes ingresar esto:

```
git clone [DIRECCION HTTPS]
```

```
Cloning into 'My-GitHub-Project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

15 TIPO 1

Ahora, tu repositorio está en tu computadora. Necesitas moverte en él con el siguiente comando:

```
cd [NAME OF REPOSITORY]
```

16 TIPO 1

Ahora, en ese folder podemos crear archivos, trabajar en ellos y guardarlos localmente.

```
$ cd My-GitHub-Project
/My-GitHub-Project$
```

17 TIPO 1

Para guardarlos en un lugar remoto como GitHub tenemos que hacer un proceso llamado "commit". Para hacer esto, regresa a tu terminal. Si la cerraste usa el comando 'cd'.

```
cd [NAME OF REPOSITORY]
```

18 TIPO 1

1. "status": La primera cosa que necesitas hacer es revisar los archivos que has modificado. Puedes escribir el siguiente comando para hacer aparecer una lista de cambios:

```
git status
```

19

TIPO 1

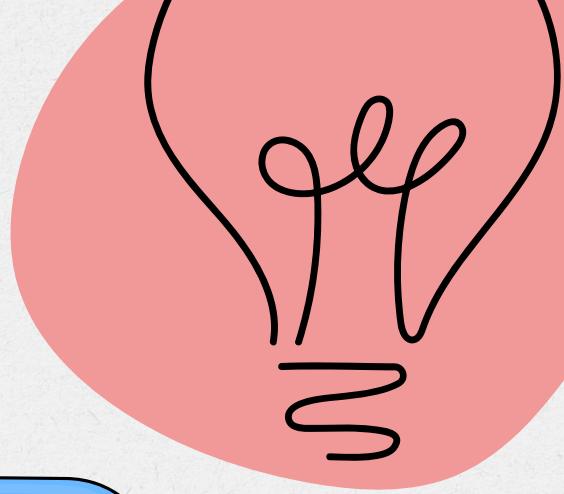
Les debe de aparecer asi:

```
thanatos18@lifecompanion:~/My-GitHub-Project$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    sample.html

nothing added to commit but untracked files present (use "git add" to track)
```



20

TIPO 1

2. "add": Con la ayuda de la lista de cambios, puedes agregar todos los archivos que quieras cargar con el siguiente comando:

```
git add [NOMBRE DE ARCHIVO] [NOMBRE DE ARCHIVO] [...]
```

22

TIPO 1

Aparecerá de la siguiente forma:

```
thanatos18@lifecompanion:~/My-GitHub-Project$ git add sample.html
thanatos18@lifecompanion:~/My-GitHub-Project$
```

21

TIPO 1

En este caso, agregaremos un archivo HTML simple:

```
git add sample.html
```

23

TIPO 1

3. "commit": Necesitamos escribir un mensaje para explicar lo que hemos hecho. Este mensaje puede ser útil después si queremos revisar el historial de cambios.

```
git commit -m "Se agregó archivo HTML de muestra que contiene sintaxis básica"
```

24

TIPO 1

Se mostrará así:

```
thanatos18@lifecompanion:~/My-GitHub-Project$ git commit -m "Added sample HTML file that contain basic syntax"
[master b234227] Added sample HTML file that contain basic syntax
 1 file changed, 12 insertions(+)
 create mode 100644 sample.html
```

25

TIPO 1

4. "push": Ahora podemos poner nuestro trabajo en GitHub. Para hacer eso, necesitamos 'enviar' nuestros archivos a Remote. Para encontrar ese nombre, escribe el siguiente comando:

```
git remote
```

26

TIPO 1

Se visualizará de la siguiente manera:

```
thanatos18@lifecompanion:~/My-GitHub-Project$ git remote
```

27

TIPO 1

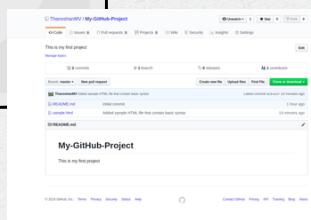
Ahora podemos 'enviar' de manera segura nuestro trabajo con el siguiente comando:

```
git push origin master
```

28

TIPO 1

Ahora, si vamos a nuestro repositorio en la página web de GitHub, podemos ver el archivo sample.html que hemos enviado a Remote.

29 **NOTA**

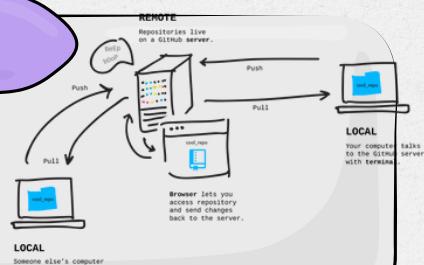
A veces cuando estás usando comandos Git en la terminal, te puede llevar al editor de texto VIM (un editor de texto basado en CLI). Para deshacerte de esto, tienes que escribir:

:q

30 **NOTA**

y presionar ENTER.

```
type :q<Enter>
```

31 **IMAGEN**32 **DATO**

Pulling es el acto de recibir de GitHub.
Pushing es el acto de enviar a GitHub.

33 **TIPO 2**

Permite hacer un nuevo repositorio de un folder existente en nuestra computadora y enviarlo a GitHub.

34 **TIPO 2**

Cuando se ejecuta cualquier comando Git, tenemos que asegurarnos que estamos en el directorio correcto en la terminal.

```
thanos18@llifecompanion:~/Code-School/WebDev/Survey Form Project$
```

35 **TIPO 2**

Cualquier directorio en nuestra computadora no es un repositorio Git pero lo podemos convertir en un repositorio Git ejecutando el siguiente comando:

```
git init
```

36 **TIPO 2**

Aparecerá de la siguiente manera:

```
thanos18@llifecompanion:~/Code-School/WebDev/Survey Form Project$ git init
Initialized empty Git repository in /home/thanos18/Code-School/WebDev/Survey Form Project/.git/
```

37 TIPO 2

Revisar los archivos que tenemos utilizando el siguiente comando:

```
git status
```

38 TIPO 2

Se mostrará una pantalla así:

```
thanosi@llifecompanion:~/Code-School/WebDev/Survey Form Projects$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    index.html
    style.css

nothing added to commit but untracked files present (use "git add" to track)
```

40 NOTA

Para "agregar" todos los archivos en nuestro repositorio podemos usar el siguiente comando:

```
git add .
```

39 TIPO 2

Así que hay dos archivos en ese directorio que necesitamos "agregar" a nuestro Repo.

```
git add [FILENAME] [FILENAME] [...]
```

42 TIPO 2

Si los archivos se encuentran en verde como en la imagen, se hizo buen trabajo.

```
thanosi@llifecompanion:~/Code-School/WebDev/Survey Form Projects$ git add .
thanosi@llifecompanion:~/Code-School/WebDev/Survey Form Projects$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   index.html
    new file:   style.css
```

44 TIPO 2

Si mi repositorio inició en GitHub y lo descargo a mi computadora, un remote ya estará unido a él (Tipo 1). Pero si estoy iniciando mi repositorio en mi computadora, no tiene un remote asociado con él, así que necesito agregarlo (Tipo 2).

41 TIPO 2

Después de que el área de preparación (el proceso de Add) está completo, podemos revisar si nuestros archivos fueron agregados satisfactoriamente o no ejecutando:

```
git status
```

43 TIPO 2

Tenemos que hacer "commit" con una descripción.

```
git commit -m "Agregando un formulario web de encuesta"
```

45 TIPO 2

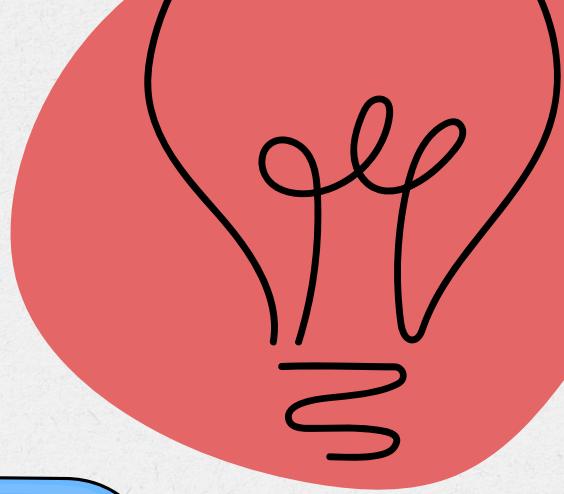
Aparecerá de la siguiente manera:

```
thanosi@llifecompanion:~/Code-School/WebDev/Survey Form Projects$ git commit -m
"Adding web Survey form"
[master (root-commit) aa8a70a] Adding web Survey form
 2 files changed, 306 insertions(+)
 create mode 100644 index.html
 create mode 100644 style.css
```

46

TIPO 2

Así que para agregar ese remote, primero tenemos que ir a GitHub. Crear un nuevo repositorio y nombrarlo como queramos almacenarlo en GitHub. Después hacer clic en el botón "Crear repositorio".



47

NOTA

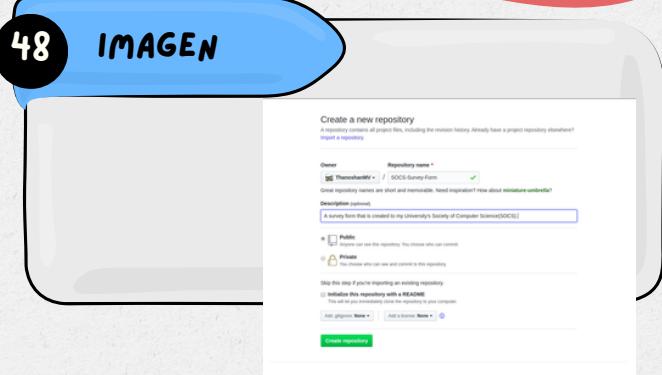
En el tipo 2, por favor no inicialices el repositorio con un archivo README cuando crees un nuevo repositorio en la página web de GitHub.

49

TIPO 2

Después de hacer clic en el botón "Crear repositorio" encontrarás la siguiente imagen como una página web.

48

IMAGEN

50

IMAGEN

51

TIPO 2

Copia la dirección HTTPS. Ahora crearemos el remote para nuestro repositorio.

```
git remote add origin [DIRECCION HTTPS]
```

52

TIPO 2

Después de ejecutar este comando, podemos revisar si hemos agregado el remote satisfactoriamente o no con el siguiente comando:

```
git remote
```

53

TIPO 2

Y si genera "origin", haz agregado el remote a tu proyecto.

54

NOTA

Solo recuerda que podemos indicar cualquier nombre para el remote cambiando el nombre "origin". Por ejemplo:

```
git remote add [NOMBRE REMOTE] [DIRECCION HTTPS]
```

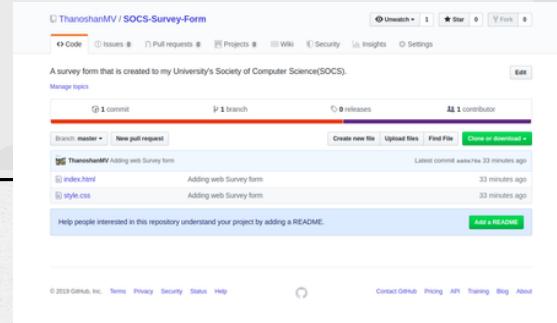
55**TIPO 2**

Ahora, podemos enviar nuestro proyecto a GitHub sin ningún problema.

```
git push origin master
```

56**TIPO 2**

Después de completar estos pasos uno por uno, si vas GitHub puedes encontrar tu repositorio con los archivos.

57**IMAGEN**

END