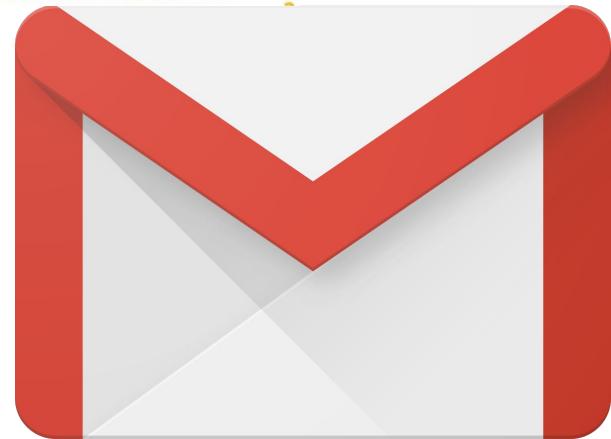
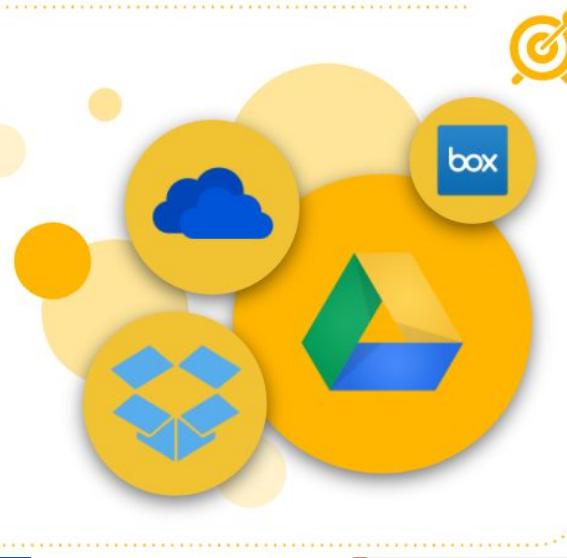


GIT & SCRUM

Completo wuu

¿Cómo compartimos archivos en la actualidad?



¿Cuál es la herramienta perfecta para nuestros de proyectos

Necesitamos
un software

que nos permita hacer un correcto
seguimiento y control de versiones.



Trabajo en ramas

RAMA PRINCIPAL --> MASTER

ORIGIN --> VERSIÓN DE
NUESTRO EQUIPO

ORIGIN MASTER --> VERSIÓN ESTA
QUE ESTÁ UTILIZANDO NUESTRO EQUIPO

“

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

LINUS TORVALDS

Ingeniero de software

Linus Benedict Torvalds es un ingeniero de software finlandés-estadounidense, conocido por iniciar y mantener el desarrollo del kernel Linux, basándose en el sistema operativo libre Minix creado por Andrew S. Tanenbaum y en algunas herramientas, varias utilidades y los compiladores desarrollados por el proyecto GNU.

De acuerdo con un artículo de Bloomberg, Linus Torvalds gana unos 1.8 millones de dólares anuales, que paga la Fundación Linux.



SOFTWARE DE CONTROL DE VERSIONES

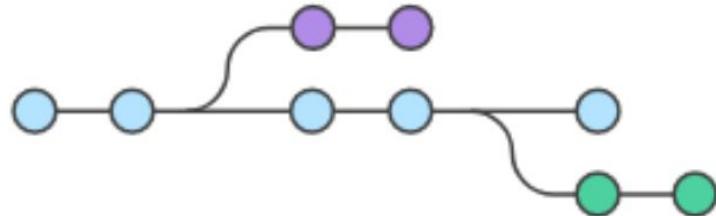


- 1. SEGUIMIENTO DE TODAS LAS VERSIONES**
- 2. SEGUIMIENTO DE CADA CAMBIO**
- 3. AUTORÍAS**

- REPOSITORIO
 1. LOCAL
Cada participante en su computadora
 2. REMOTO
En la nube
siempre tiene lo último

Ramas / Branches

Ramas / Branches





git

Repo Local



GitHub

Repo Remoto





Instalando GIT en nuestras máquinas

<https://git-scm.com/>

¿Cómo funciona GIT?

¿Qué significa ORIGIN MASTER?

TERMINAL:

- 
- 1. MAYOR PODER Y ACCESO AL SISTEMA OPERATIVO**
 - 2. SE UTILIZA MUCHO EN ENTORNOS DE DESARROLLO**
 - 3. PORQUE ALGUNOS LENGUAJES DE PROGRAMACIÓN LO "REQUIEREN"**
 - 4. PODEMOS OPTIMIZAR NUESTRO TRABAJO**

Me e

El terminal es un programa que está presente en todos los sistemas operativos y por medio del cual se pueden dar órdenes al sistema a través de líneas de comando.



¿Por qué usar la terminal?

- Para tener mayor control sobre el SO.
- Porque es muy común en los entornos de desarrollo.
- Porque algunos lenguajes de programación lo "requieren".

Si sabemos usar la terminal y nos acostumbramos a la misma, podremos optimizar mucho nuestro trabajo de programar.

-a viene de all

ls list

clear -> borrar

lh -> permisos

man ls -> todos los comandos

cd -> change director

TAB -> AUTOCOMPLETAR

cd.. -> volver

mkdir ok -> crear un directorio

touch ejemplo.txt -> nuevo archivo

rm ejemplo.txt -> remover

rm -r ok -> borrara de manera recursiva
recursiva

cp (copiar)/ mv (mover o renombrar)

Nombre del proyecto

```
sus@AR-BE-FS-00:~$ ls
Descargas  Downloads  examples.desktop  Música  out-2.ogv  Plantillas  Vídeos
Documentos Escritorio  Imágenes        out-1.ogv  out.ogv   Público

sus@AR-BE-FS-00:~$ ls -a
.          Documentos      .ICEauthority  Plantillas
..         Downloads       Imágenes       .profile
.atom      Escritorio     .local         Público
.bash_history examples.desktop .mozilla     .rnd
.bash_logout .gconf        Música        .sudo_as_admin_successful
.bashrc     .gitconfig    .mysql        .thunderbird
.cache      .gitkraken   .nano         Videos
.compirz   .gnome        .npm         .Xauthority
.config    .gnupg        out-1.ogv   .xsession-errors
.dbus      .gphoto       out-2.ogv   .xsession-errors.old
Descargas  .gtk-recordmydesktop out.ogv
.dmrcc    .gvfs          .pki

SUS@AR-BE-FS-00:~$
```

```
SUS@AR-BE-FS-00:~$ ls -lh
total 2,0M
drwxr-xr-x 4 sus sus 4,0K jul 17 15:41 Descargas
drwxr-xr-x 2 sus sus 4,0K abr 18 20:28 Documentos
drwxrwxr-x 2 sus sus 4,0K abr 19 19:29 Downloads
drwxr-xr-x 5 sus sus 4,0K jul 6 12:47 Escritorio
-rw-r--r-- 1 sus sus 8,8K abr 18 20:13 examples.desktop
drwxr-xr-x 4 sus sus 4,0K may 9 10:02 Imágenes
drwxr-xr-x 2 sus sus 4,0K abr 18 20:28 Música
-rw-rw-r-- 1 sus sus 578K may 14 10:08 out-1.ogv
-rw-rw-r-- 1 sus sus 1,2M may 14 10:10 out-2.ogv
-rw-rw-r-- 1 sus sus 162K may 14 10:07 out.ogv
drwxr-xr-x 2 sus sus 4,0K abr 18 20:28 Plantillas
drwxr-xr-x 2 sus sus 4,0K abr 18 20:28 Público
drwxr-xr-x 3 sus sus 4,0K jul 16 10:21 Vídeos

SUS@AR-BE-FS-00:~$
```

ls (en Mac y Linux muestra los archivos de la carpeta en la que estamos ubicados, en Windows también si usamos el **PowerShell**)

dir (en Windows muestra los archivos de la carpeta en la que estamos ubicados)

cd .. (nos permite retroceder a una carpeta previa)

cd nombre-carpeta (nos permite acceder a la carpeta descrita)

clear

(limpia todo lo que hayamos escrito en la consola / Mac y Linux. En Windows en el **PowerShell**)

cls

(limpia todo lo que hayamos escrito en la consola / Windows)

mkdir algo

(crea una carpeta con el nombre "**algo**")

touch archivo.txt

(crea una archivo de texto "**archivo.txt**")

rm archivo.txt

(elimina un archivo con el nombre "**archivo.txt**")

mv nombre.txt otro.txt

(cambia el nombre "**archivo.txt**" a "**otro.txt**")



**UTILIZAR GIT A TRAVÉS
DE LA TERMINAL ES
EL MODO MÁS POTENTE**

CREACIÓN DE UN REPOSITORIO

Directorio del u

Em

CASO 1:
ESTOY CREANDO
EL REPOSITORIO
DESDE CERO

CASO 2:
YO ME SUMO COMO
COLABORADOR

**1. CREAR
REPOSITORIO REMOTO**

**2. OPERAR DESDE NUESTRO
REPOSITORIO LOCAL**

CASO 2

**CLONAR
EL PROYECTO**

@AR-BE-FS-00: ~/Escritorio/HTML/clonGIT

(42%) mier 18 de jul 16:40

```
sus@AR-BE-FS-00:~/Escritorio/HTML$ ls
css          img          inicio2.php  nosotros.html
embed.html   inicio2.html  inicio.html  primero.html
sus@AR-BE-FS-00:~/Escritorio/HTML$ mkdir ejemploGIT
sus@AR-BE-FS-00:~/Escritorio/HTML$ cd ejemploGIT
sus@AR-BE-FS-00:~/Escritorio/HTML/ejemploGIT$ git init
Initialized empty Git repository in /home/sus/Escritorio/HTML/ejemploGIT/.git/
sus@AR-BE-FS-00:~/Escritorio/HTML/ejemploGIT$ git remote add origin https://github.com/dariosus/blend.git
sus@AR-BE-FS-00:~/Escritorio/HTML/ejemploGIT$ cd ..
sus@AR-BE-FS-00:~/Escritorio/HTML$ mkdir clonGIT
sus@AR-BE-FS-00:~/Escritorio/HTML$ cd clonGIT
sus@AR-BE-FS-00:~/Escritorio/HTML/clonGIT$ git clone https://github.com/dariosus/blend
.git
Clonar en «blend»...
warning: Parece que ha clonado un repositorio vacío.
Comprobando la conectividad... hecho.
sus@AR-BE-FS-00:~/Escritorio/HTML/clonGIT$
```



“

git init

Crea un repositorio local (en nuestra máquina) y nos permite comenzar a utilizar todas las funcionalidades de GIT.

Generalmente crea una carpeta oculta la cual contiene todo el repositorio y sus distintas ramificaciones.

“

Para que todo lo que hagamos quede “firmado” por nosotros, necesitamos decirle al repositorio quien somos, así:

git config user.name "Jhon_Doe"

git config user.email "jhon@email.com"

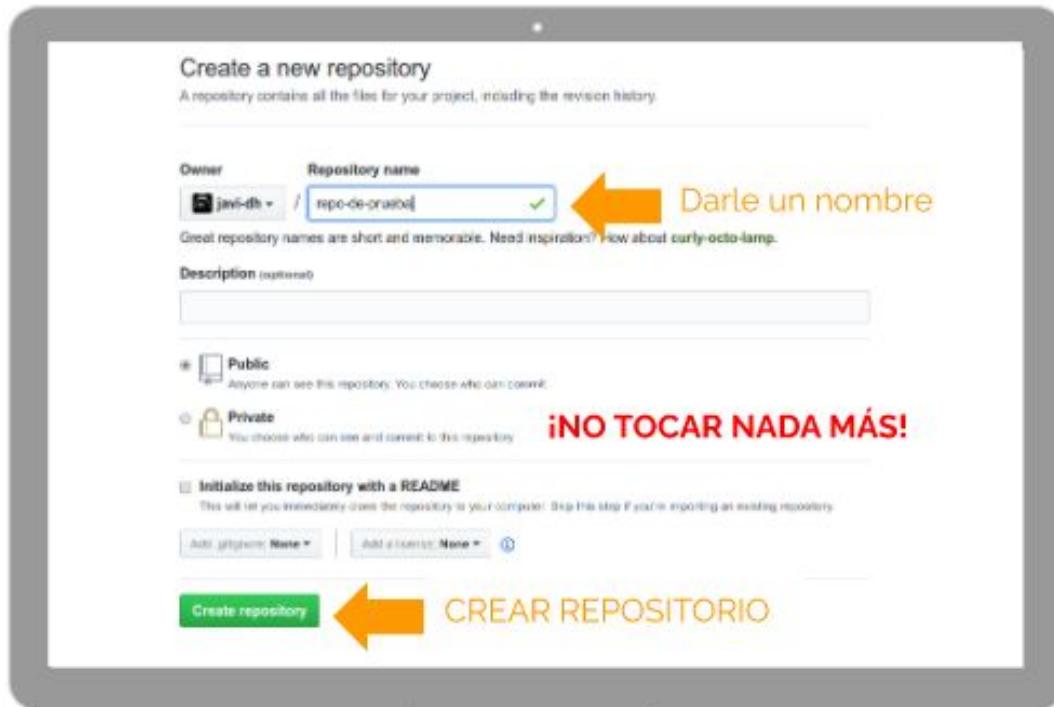
“

git config user.name " "

Dentro de las comillas pondremos nuestro usuario de Github.com

git config user.email " "

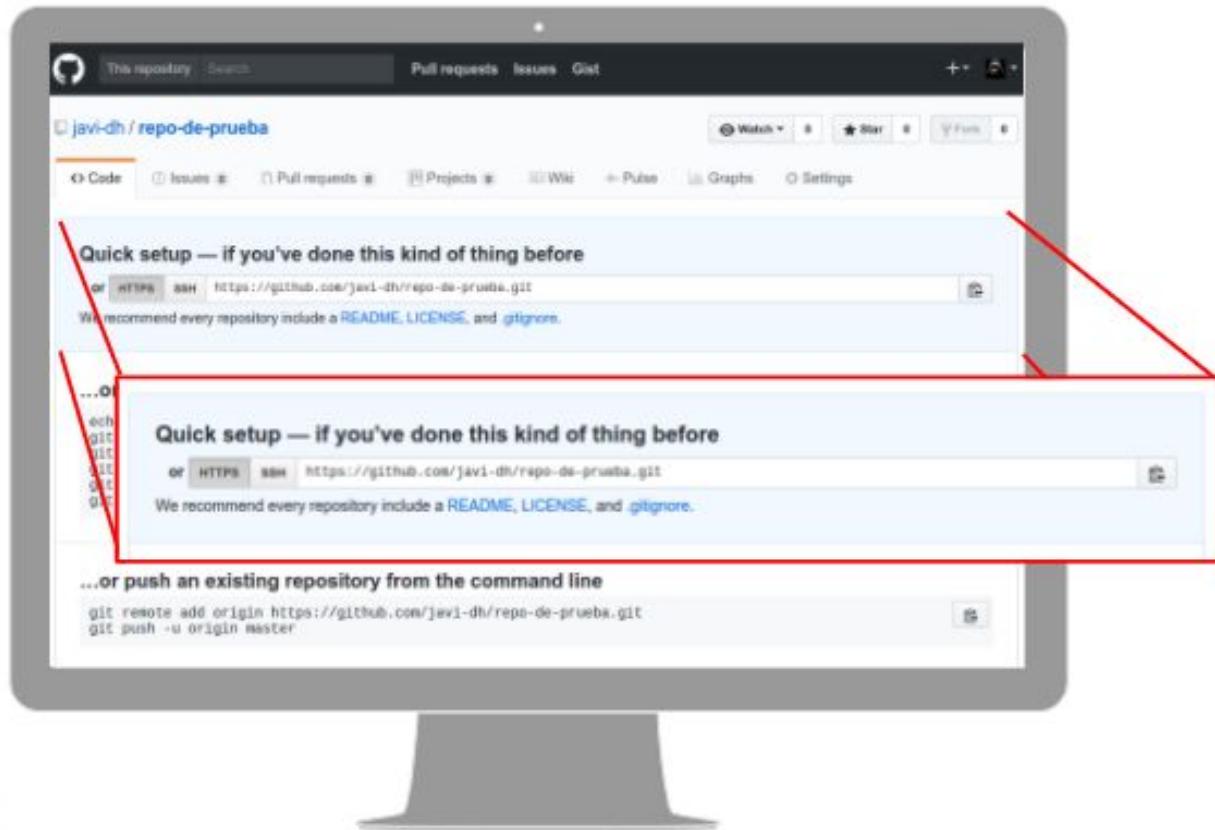
Dentro de las comillas pondremos el email con el que nos registramos en Github.com



El nombre que elijamos puede ser cualquier, uno que no hayamos usado para otro repositorio.

De resto **NO TOCAR** nada más, solo el botón

CREATE.



Luego veremos esta pantalla y ésta URL es la que necesitamos tener a mano en el paso de: **Asignando nuestro repositorio remoto.**

“Habiendo creado el **Repositorio Remoto** y para que nuestro **Repositorio Local** sepa a donde queremos subir nuestros archivos tenemos que especificarlo así:

```
git remote add origin https://github.com/user/repo.git
```

“

git remote add origin https://...

Con este comando, le estamos indicando a nuestro repositorio local, a donde queremos llevar (repositorio remoto) nuestros archivos.

La URL la obtendremos al crear un **repositorio remoto en Github.com**

“Para **descargar por 1era vez** un repositorio remoto a nuestra máquina. Tendremos que clonar el mismo en el lugar que deseemos.

El comando que necesitamos será:

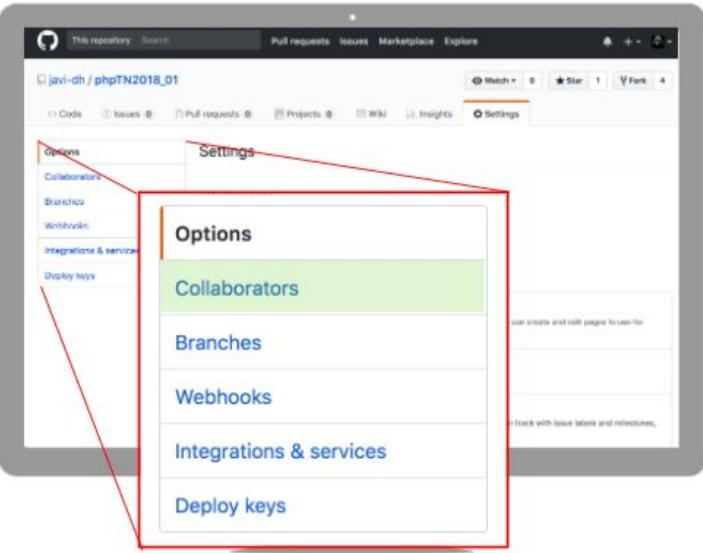
`git clone https://github.com/user/repoName`

“

`git clone https://....`

`git clone`, permite crear una copia idéntica del repositorio remoto en nuestra máquina. Para que podamos trabajar con los mismo archivos que tengamos hasta ese momento. Después de trabajarlos deberemos como siempre **pushearlos**.

Ahora, la pregunta es ¿cómo hago para actualizar los archivos que hice en la máquina original con estos nuevos archivos?



Una vez aquí iremos a la opción:
Collaborators



A screenshot of a GitHub repository settings page showing the 'Collaborators' section. The text says 'This repository doesn't have any collaborators yet. Use the form below to add a collaborator.' Below is a search bar with the placeholder 'Search by username, full name or email address'. A red arrow points to the search bar where the name 'kev' is typed. Below the search bar, a dropdown list shows results: 'kevlinmichael' (selected), 'Kev Kevin Smith', and 'kevd7 Kev'. There is a blue 'Add collaborator' button to the right of the search bar.



Aquí escribiremos el nombre de usuario de nuestro colega y después pulsaremos el botón:
Add collaborator

La persona recibirá un email, donde deberá aceptar la invitación.

COMMIT & PUSH

COMMIT -> EMPAQUETADO -> CERRAMOS UNA VERCION

git add .

git status -> indica si esta subido o no

-m -> mensaje

TODOS

```
OGIT$ git commit -m "Primer Commit"
```

git config -> si hay algun problema

git commit



git push

git add .

Agrega al **stage** (de manera temporal) todos los archivos que hayamos creado en nuestro proyecto.

git add archivo.txt

Agrega al **stage** (de manera temporal) solamente el archivo referenciado.

git status

Analiza el estado del repositorio, nos dirá si hay archivos que no se han agregado temporalmente al **stage** así como también si hay archivos agregados al stage pero no de forma (**commit**).

“

Cada vez que deseemos comprobar o verificar el estado de nuestro repositorio podremos escribir la siguiente línea de comando:

git status

“

Para finalmente confirmar que los archivos agregados al **stage** los queremos de **manera definitiva** escribiremos:

git commit -m "un mensaje cualquier"

git commit -m "un mensaje cualquier"

La directriz **commit**, le indica al repositorio que los archivos los queremos agregar de manera oficial. La **-m** indica que a continuación agregaremos un mensaje que especifique qué trabajo hicimos.

Los **commits** sirven como pequeños **backups** a los cuales podremos volver fácilmente si así lo necesitáramos.

git push origin master

El **push**, permite enviar los archivos de nuestra máquina (**repositorio local**) al **repositorio remoto**.

Al especificar **master**, estamos diciendo a qué rama del repositorio queremos enviar nuestros archivos.

Para enviar los archivos que tenemos en nuestro repositorio local al repositorio remoto, escribiremos la siguiente línea:

git push origin master

Y recuerden:
In case of fire



1. git commit



2. git push



3. leave building



pull

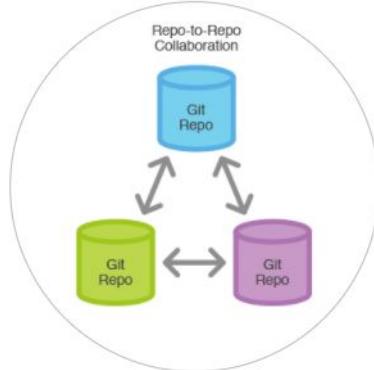
Bajando los **archivos** de **repo remoto** a **nuestro repo local**

“

git pull origin master

git pull, baja a tu repositorio local, los cambios o archivos nuevos que se hayan **pusheado** al repositorio remoto desde otra máquina,

Este comando es muy funcional si trabajamos con más colaboradores en el mismo proyecto.



Si lo que deseamos es actualizar los archivos en nuestro repositorio local con lo existente en el repositorio remoto, deberemos:

git pull origin master

git pull origin master
en caso de conflictos, git nos avisará donde debemos corregirlos

Una vez corregidos hay que hacer un commit de **merge** para dejar el estado final en el repositorio remoto



1 **git init** //crea el repositorio

2 **git config user.name "hanSolo"**
//agrega nuestra identidad - username

3 **git config user.email "hansolo@starwars.com"**
//agrega nuestra identidad - email

4 **git remote add origin https://github.com/....**
//apunta al repositorio remoto

5 **git add .**
//agrega todos los cambios al repo local

6 **git commit -m 'mensaje del commit'**
//hito histórico - comitea los cambio hechos

7 **git push origin master**
//manda los cambios al repositorio remoto

... Los pasos de 5 al 7, se repiten tantas veces como funcionalidades vayamos sumando al proyecto.



git pull

traera los repositorios de nuestros compaleros

git soluciona conflictos -> mezcla la ultima
version

¡CONFLICTO!



MISMA LÍNEA DE CÓDIGO



[Code](#)[Issues 0](#)[Pull requests 0](#)[Projects 0](#)[Wiki](#)[Insights](#)[Settings](#)

Branch: master ▾

blend / nosotros.html[Find file](#)[Copy path](#)**dariosus** Primer Commit

6831b8a 20 minutes ago

1 contributor

0 lines (0 sloc) | 0 Bytes

[Raw](#)[Blame](#)[History](#)

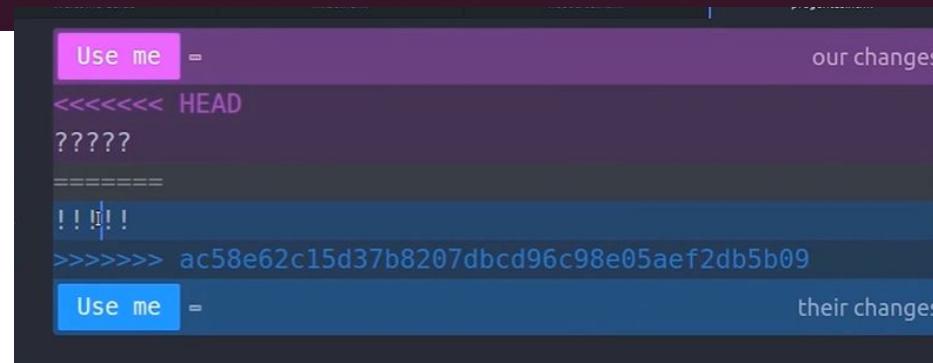
Add an optional extended description...

 ⚡ Commit directly to the `master` branch. 🛡 Create a **new branch** for this commit and start a pull request. Learn more about pull requests.[Commit changes](#)[Cancel](#)

```
sus@AR-BE-FS-00:~/Escritorio/HTML/ejemploGIT$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
De https://github.com/dariosus/blend
 * branch            master      -> FETCH_HEAD
   0b2e08f..9e096d1  master      -> origin/master
Updating 0b2e08f..9e096d1
Fast-forward
  nosotros.html | 2 ++
  1 file changed, 2 insertions(+)
sus@AR-BE-FS-00:~/Escritorio/HTML/ejemploGIT$
```

No se hace problemas

Atom preparado



Add topics

9 commits

1 branch

0



Branch: master ▾

New pull request

Cr

dariosus Merge

inicio.html

Update inicio.html

nosotros.html

Update nosotros.html

preguntas.html

???

Webs de consulta

Siempre viene bien tener a la mano documentación, para ello podemos visitar:

- <https://dev.to/davedodea/git--commands-you-need-to-git-going-3mpg>
- <https://ohshitgit.com/>
- <https://git-scm.com/>

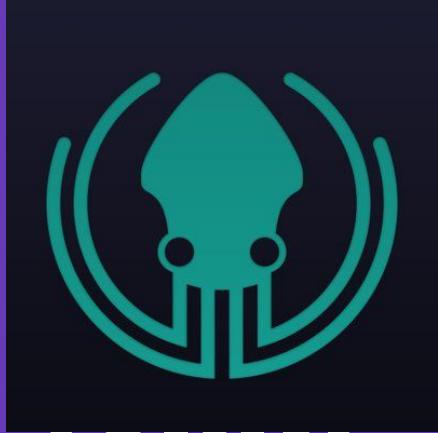


¿Cómo se actualiza el repositorio local con los cambios del repositorio remoto?

¿Cuando aparece un conflicto en GIT?

¿Cómo se resuelve un conflicto?

Gitkraken (cilk)



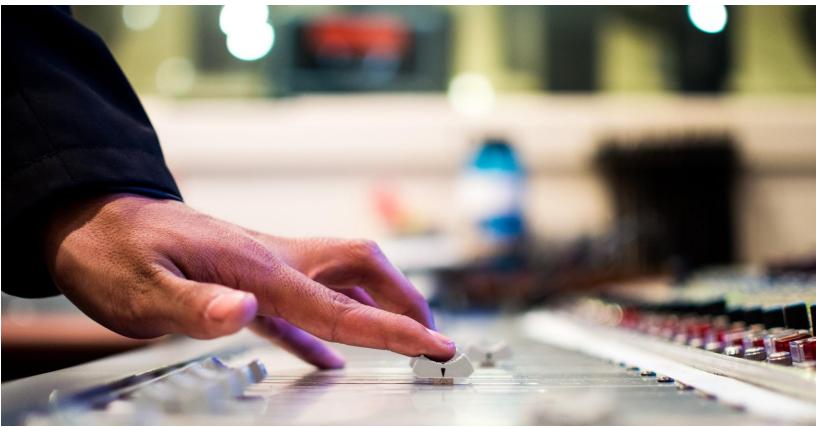
CLONANDO



INICIALIZANDO

- 1- SUBIR AL ESCENARIO
- 2- ARMAMOS EL COMMIT
- 3- HACEMOS EL PUSH

HACEMOS UN PULL



Nombre del proyecto

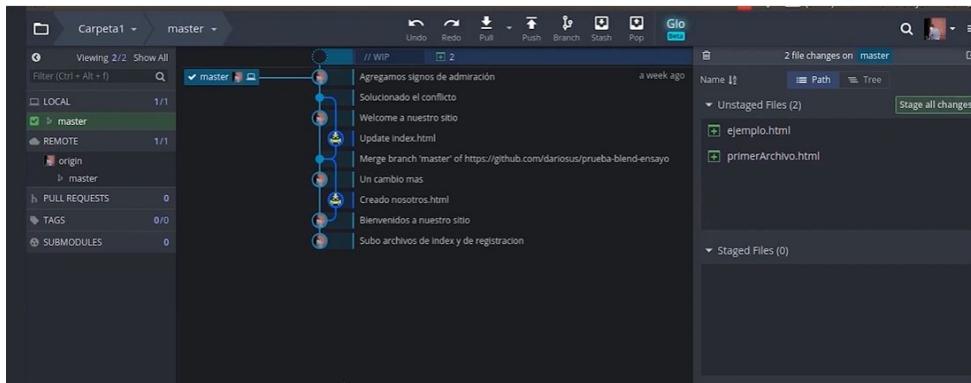
Escribe aquí tu texto Escribe aquí tu
texto Escribe aquí tu texto Escribe aquí
tu texto Escribe aquí tu texto



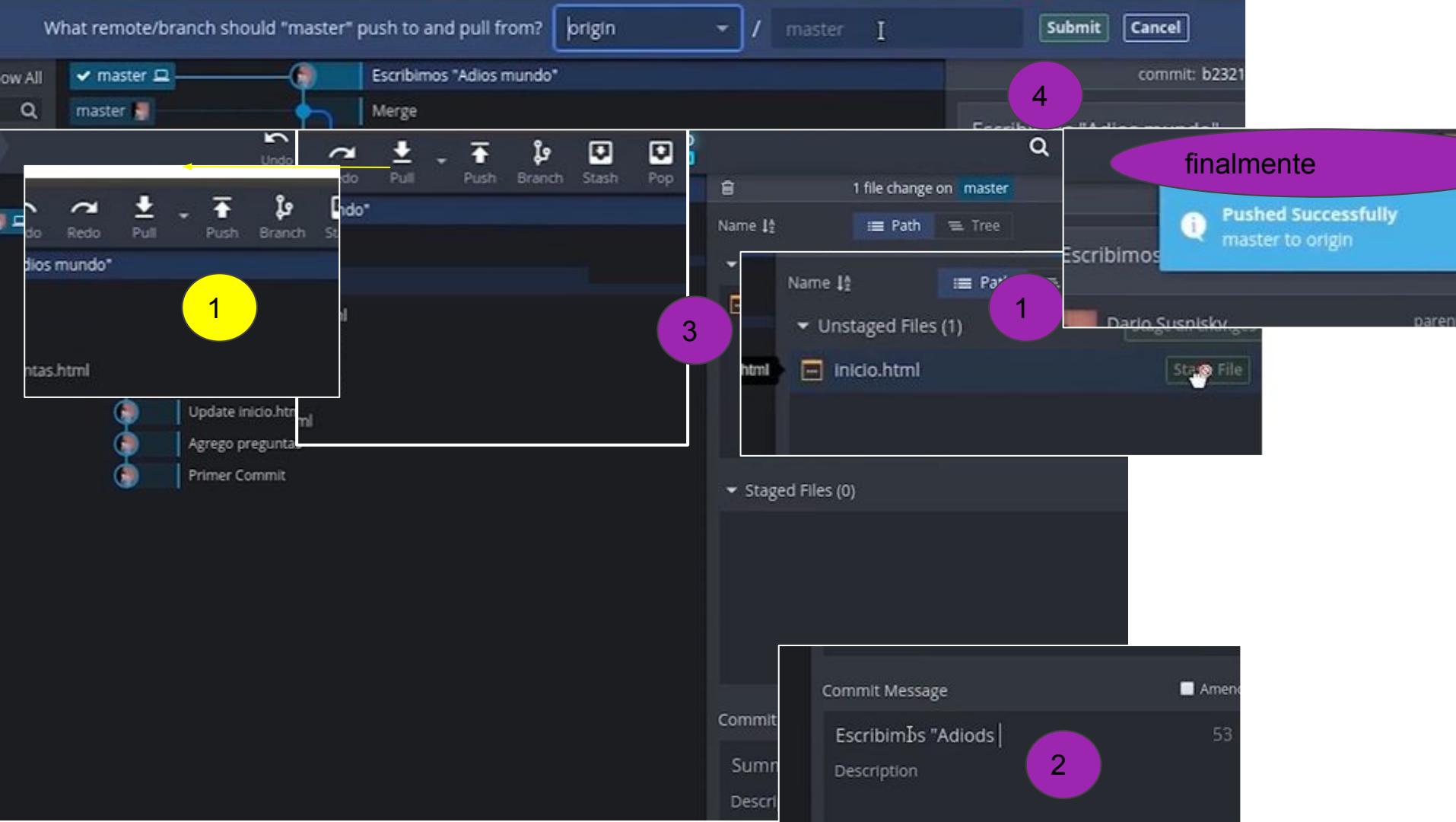
Nombre del proyecto

Escribe aquí tu texto Escribe aquí tu
texto Escribe aquí tu texto Escribe aquí
tu texto Escribe aquí tu texto

Funciona excelente en cualquier sistema operativo.



UTILIZAR TODO EL FLUJO DE GIT PERO DESDE UNA INTERFAZ GRÁFICA



Contacto

LAZO ARROYO

lazocorinafanny123@gmail.com

www.example.com

