

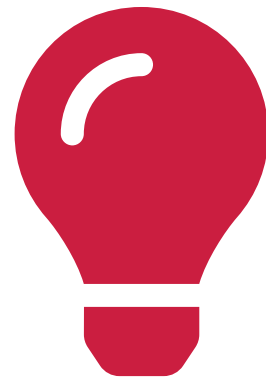




Es un software de **control de versiones** diseñado por Linus Torvalds (el creador de Linux) que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo.

Nos permite:

- Tener un historial de cómo fue cambiando el código de un proyecto a lo largo del tiempo.
- Mantener a todo el equipo de trabajo sincronizado con la última versión estable del código.
- Ser eficiente y confiable al momento de implementar cambios o mejoras en el código del proyecto.





**GIT** es un **software** pensado para ser usado por **programadores**.

Se instala en nuestras computadoras y se usa desde la **consola de comandos/ terminal**.

La instalación de **GIT** agrega sus funcionalidades a nuestras consolas/ terminales.

Al finalizar la instalación **NO** veremos un **ícono** en el escritorio como sucede con las aplicaciones comerciales tradicionales.

Podemos verificar si GIT está instalado en nuestro sistema escribiendo la sentencia ***git --version*** en la terminal.



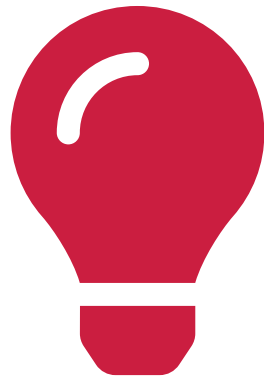


Cuando trabajamos con GIT debemos tener siempre presente el término **REPOSITORIO**.

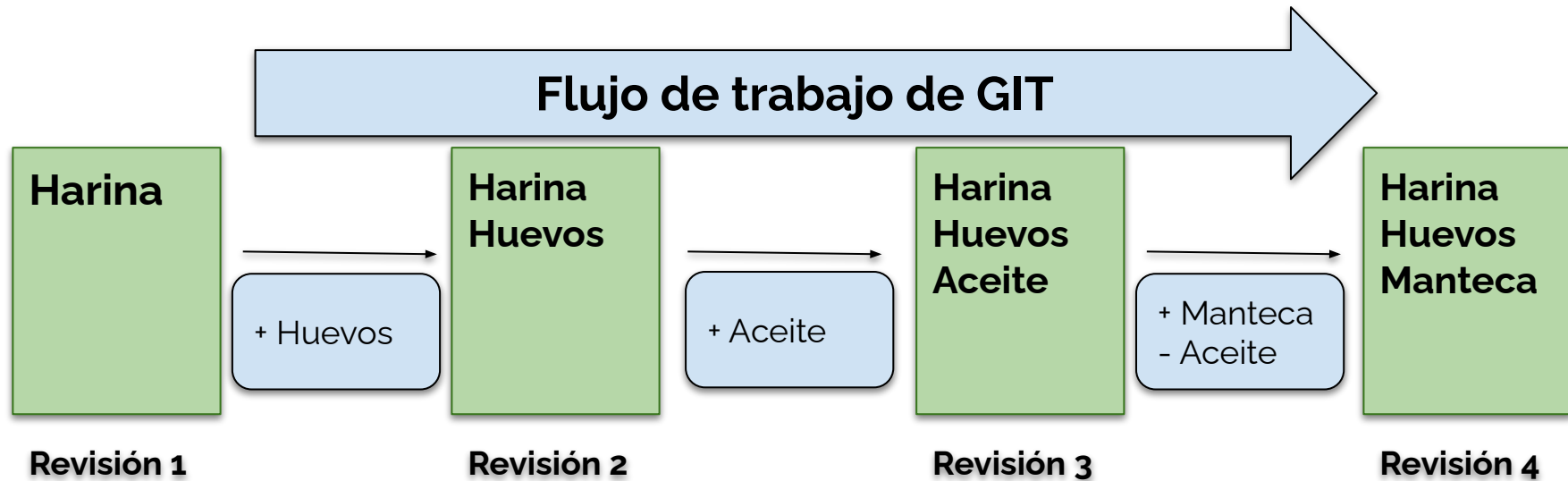
Un **repositorio** es el lugar en donde se almacenan nuestros archivos.

Usando GIT tendremos dos tipos de **repositorios**:

- **local** en nuestra máquina.
- **remoto** creado en internet.



# FLUJO DE TRABAJO EN GIT





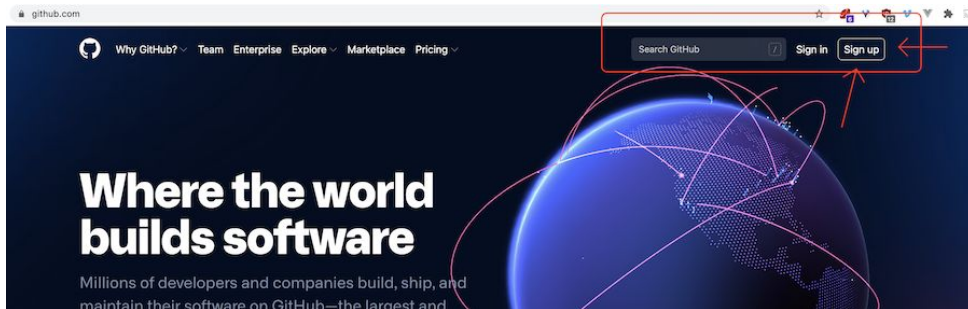
# REPOSITORIO REMOTO GITHUB

**DigitalHouse** >  
Coding School

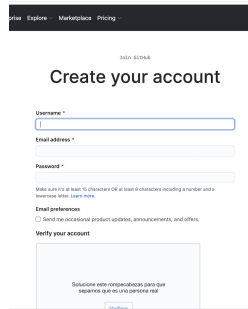
/Academy  
/Schools  
/Corporate Training  
/University

## Dos simples pasos:

1. Ingresar a la siguiente URL: <https://github.com/>



2. Completar el formulario de registro.

A screenshot of the GitHub 'Create your account' registration form. The form is titled 'Create your account' and includes fields for 'Username', 'Email address', and 'Password'. Below the password field, there is a checkbox for 'Email preferences' and a link to 'Verify your account'. The form is set against a dark background with white text and input fields.



# CREANDO EL REPOSITORIO REMOTO

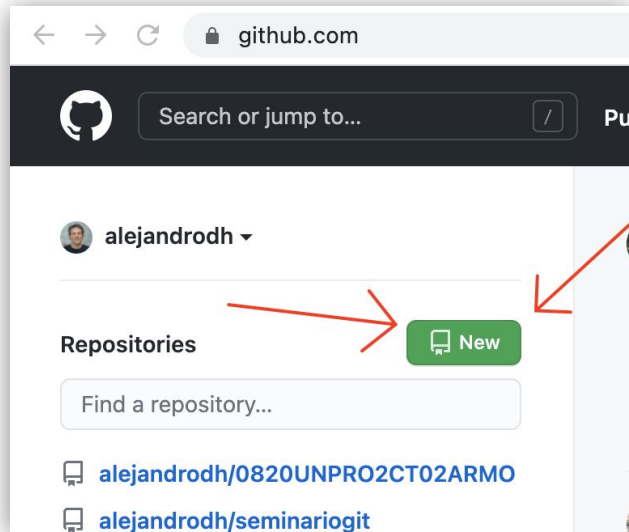
**DigitalHouse** >  
Coding School

/Academy  
/Schools  
/Corporate Training  
/University



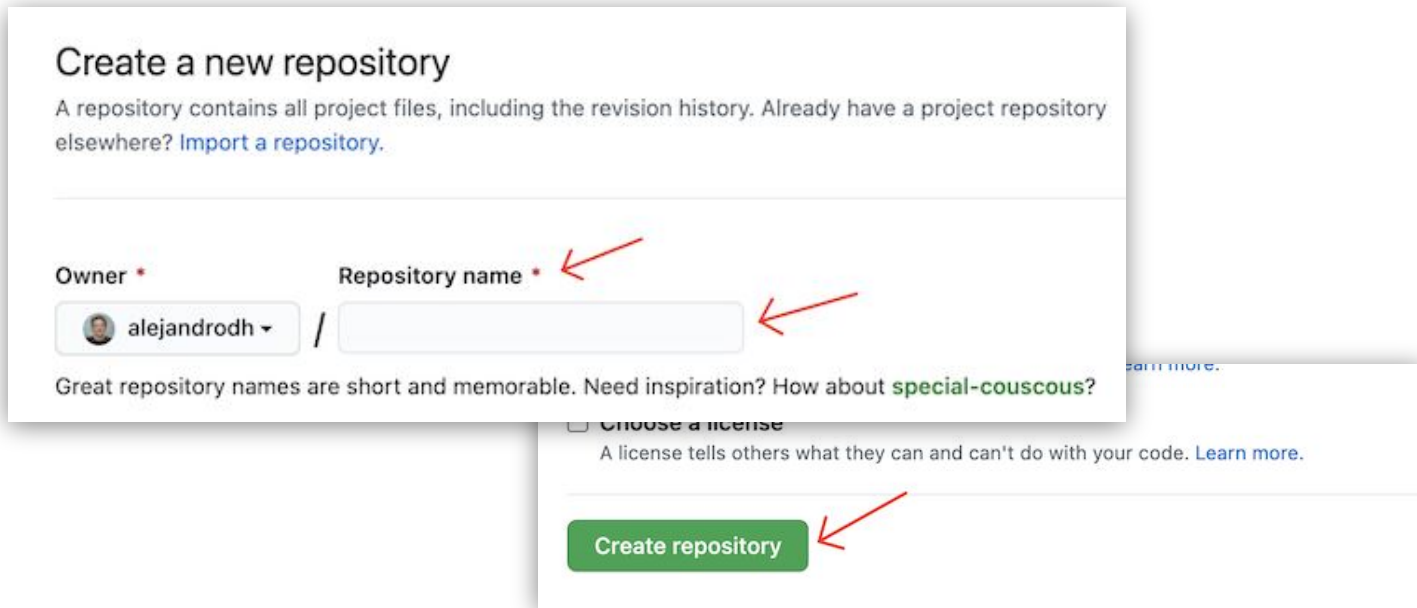
## Dos simples pasos:

1. Dentro de cuenta hacé click en el botón verde con el texto **“new”**.



## Dos simples pasos:

2. Completá el campo obligatorio con el nombre de tu repositorio y dale click al botón verde al pie del formulario.




**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 \* ←

 alejandrodh ▾ /  ←

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

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

[Create repository](#) ←



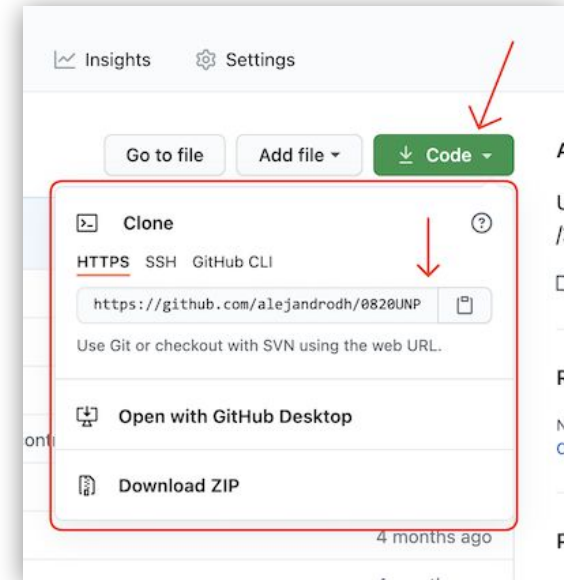
# CLONANDO EL REPOSITORIO

# CLONANDO EL REPOSITORIO

El comando ***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.

Para clonar el repositorio debemos usar el comando ***git clone*** desde la terminal seguido de la url del repositorio. Encontraremos la url en Github en el botón verde **"code"**.

Al finalizar la clonación tendremos creado nuestro **repositorio local**.



***git clone*** <https://github.com/alejandrodh/0820UNPRO2CT02ARMO.git>

# INDICANDO NUESTRA **IDENTIDAD** AL REPOSITORIO

Para que todo lo que hagamos quede "**firmado**" por nosotros necesitamos indicar nuestras credenciales al **repositorio local** ya clonado.

Necesitaremos nuevamente de la terminal y del comando ***git config*** con las siguientes variantes:

```
git config user.name "mi usuario de github"
```

```
git config user.email "miEmail@github.com"
```

Según corresponda dentro de las comillas escribiremos nuestro usuario de Github o el email con el que nos registramos, también en Github.



# AGREGANDO ARCHIVOS AL REPOSITORIO

# AGREGANDO ARCHIVOS **AL REPOSITORIO**

Para **agregar archivos** al **repositorio** debemos cumplir **con 3 pasos**. Cada uno de ellos está acompañado de un comando a ejecutar en la consola/ terminal:

1. Indicar a git que **agregue** uno o varios archivos a su lista para hacerles seguimiento.
2. Crear el **punto de control** e indicar a git cuales son los archivos que lo componen.
3. **Enviar los archivos** al repositorio remoto.

## 1 AGREGAR ARCHIVOS PARA SEGUIMIENTO

Dentro de un repositorio **git** va a identificar todos los movimientos de los archivos. Identificará el **alta**, los **cambios** internos y la **eliminación**. Es nuestra responsabilidad indicarle qué hacer cuando detecta los movimientos.

El comando **git add** indica a git que queremos **hacer seguimiento** de los archivos creados y/o modificados.

Para agregar **todos** los archivos usamos:

**git add .** (Sí con el "." al final del comando).

Para agregar archivos **individualmente**:

**git add nombreDelArchivo.extensión**



## 2 CREAR UN PUNTO DE CONTROL

El comando ***git commit*** permite crear un **punto de control** (o commit) en el historial de nuestro repositorio. Dentro del punto de control estarán **todos** los archivos agregados para **seguimiento** que incluimos en el punto anterior.

El comando ***git commit*** necesita **obligatoriamente** que incluyamos un mensaje que describa brevemente al punto de control.

***git commit -m "cambios en formulario de login"***

## 3 ENVIAR LOS ARCHIVOS AL REPOSITORIO REMOTO

El comando ***git push*** envía los archivos empaquetados dentro del commit (punto de control) al repositorio remoto.

El comando está acompañado de 2 términos más:

- ***origin***: es un alias que por detrás tiene la url del repositorio remoto. El alias se usa para no tener que recordar y escribir toda la url.
- ***master***: es el nombre de la rama\* del repositorio a donde queremos que git coloque los archivos.

***git push origin master***

(\*)El manejo de **ramas** es parte de conocimientos avanzados en el uso de GIT. Por el momento y para hacer más sencillo el uso asumamos que una rama es similar a una carpeta y allí vamos a colocar siempre nuestros archivos.



# VERIFICANDO EL ESTADO DEL REPOSITORIO

# VERIFICANDO EL ESTADO DEL REPOSITORIO

El comando ***git status*** devuelve en la consola el estado de nuestro repositorio. Allí podremos ver si hay archivos sin seguimiento o si tenemos archivos pendientes de incluir en un commit.

Podemos usar el comando ***git status*** luego de cada una de las instancias para ver como evoluciona nuestro repositorio.

***git status***



# DESCARGANDO ARCHIVOS AL REPOSITORIO LOCAL

DigitalHouse >  
Coding School

/Academy  
/Schools  
/Corporate Training  
/University

# DESCARGANDO ARCHIVOS AL REPOSITORIO LOCAL

El comando ***git pull*** baja al **repositorio local** los cambios o archivos nuevos **pusheados** (subidos) al **repositorio remoto** por otros miembros del equipo o desde otra máquina.

El comando está acompañado de 2 términos más:

- ***origin***: es un alias que por detrás tiene la url del repositorio remoto. El alias se usa para no tener que recordar y escribir toda la url.
- ***master***: es el nombre de la rama\* del repositorio desde donde queremos que git descargue los archivos.

***git pull origin master***

(\*)El manejo de **ramas** es parte de conocimientos avanzados en el uso de GIT. Por el momento y para hacer más sencillo el uso asumamos que una rama es similar a una carpeta y allí vamos a colocar siempre nuestros archivos.

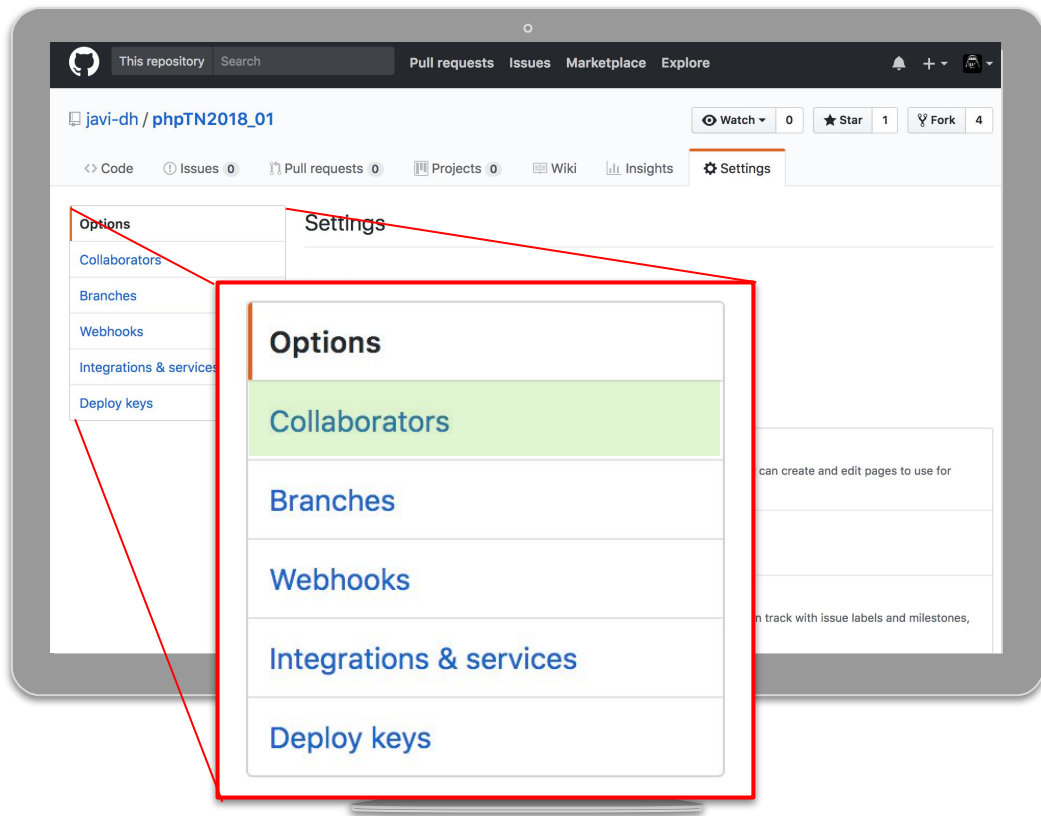


# AGREGAR COLABORADORES AL REPOSITORIO

DigitalHouse >  
Coding School

/Academy  
/Schools  
/Corporate Training  
/University

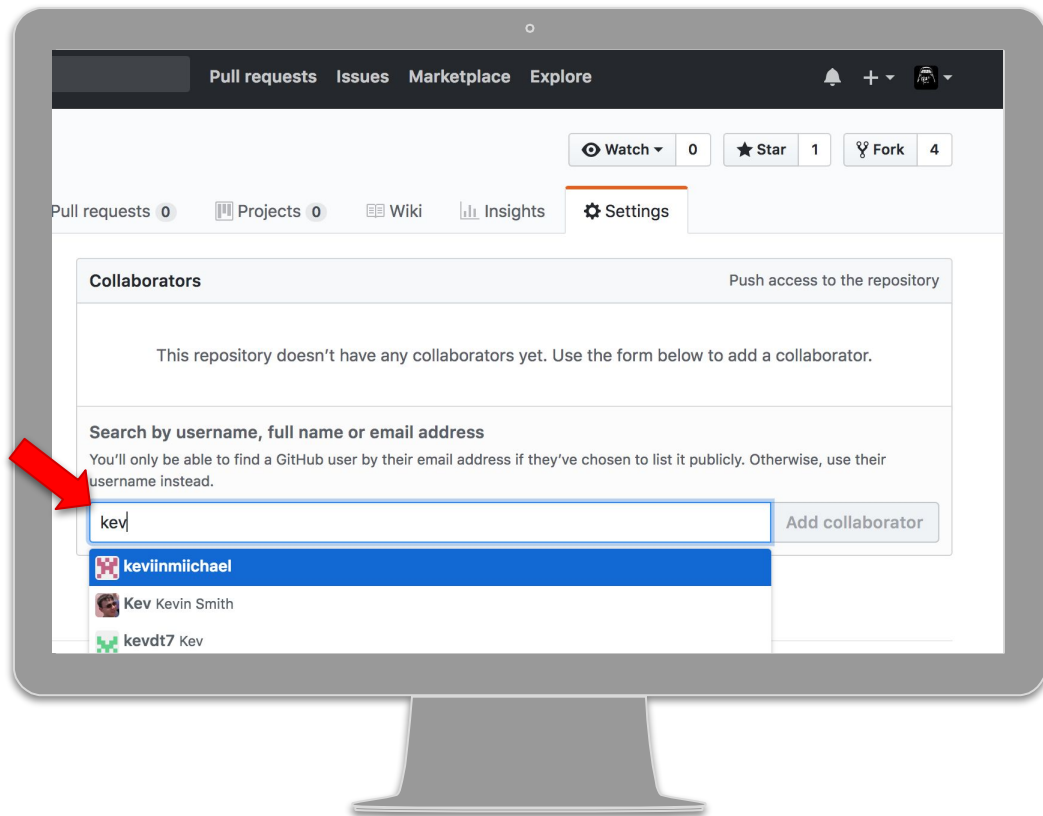
# AGREGANDO COLABORADORES AL REPOSITORIO REMOTO



Dentro de nuestro **repositorio** en **github** debemos ir a la pestaña **settings** y luego al menú **collaborators**.



# AGREGANDO COLABORADORES AL REPOSITORIO REMOTO



Aquí escribiremos el nombre de usuario y pulsaremos el botón:  
**Add collaborator**

El usuario recibirá un email con la invitación para participar.

Es **IMPORTANTE** aceptar la invitación. De lo contrario github no otorgará el permiso.

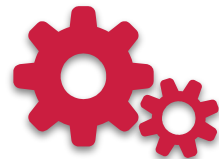


# PASO A PASO

**DigitalHouse** >  
Coding School

/Academy  
/Schools  
/Corporate Training  
/University

# PASO A PASO



- 1 Crear el repositorio **remoto** //En github.
- 2 `git clone https://...` //Clona y creo el repositorio local
- 3 `git config user.name "hanSolo"` //Agrega nuestro usuario  
`git config user.email "hansolo@starwars.com"` //Agrega nuestro email
- 4 `git add .` //Agrega todos los archivos para seguimiento.
- 5 `git commit -m "Un mensaje siempre"` //Crea un punto de control.
- 6 `git push origin master` //Sube los cambios al repositorio remoto.
- 7 `git pull origin master` //Descarga los cambios del repositorio remoto.

Los pasos de 4, 5 y 6 se repiten tantas veces como funcionalidades vayamos sumando al proyecto.

# PRACTIQUEMOS



**DigitalHouse** >  
Coding School



**¡NOS VEMOS LA PRÓXIMA!**