



Capacitarte

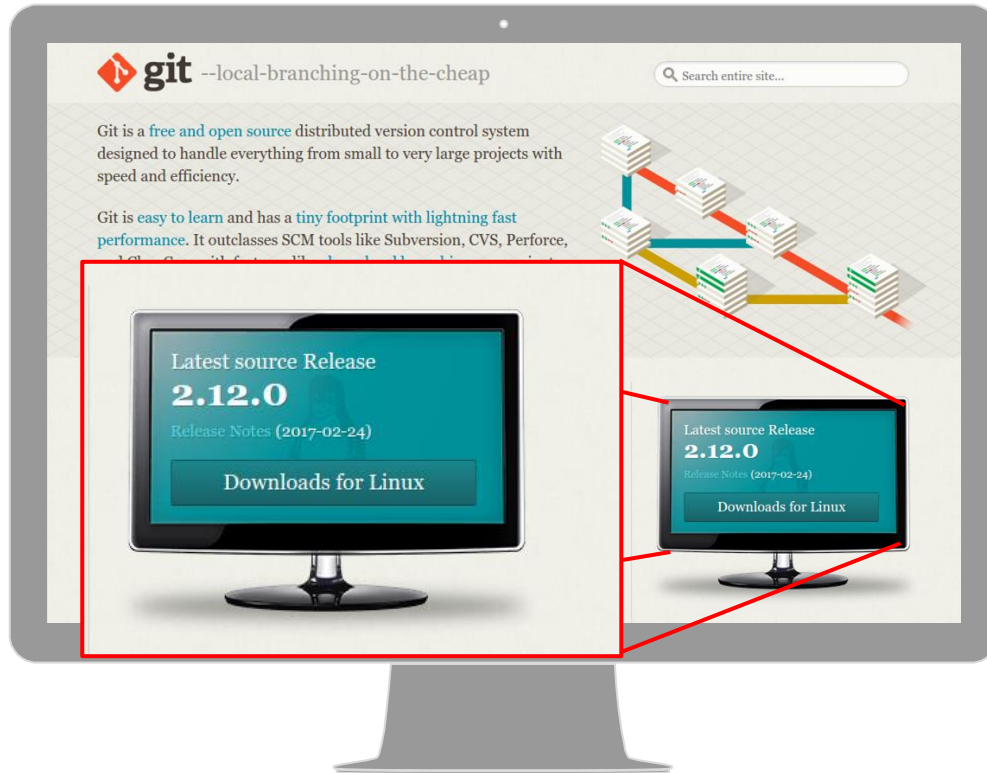
Introducción a React



git



GitHub



Instalando GIT en nuestras máquinas

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



Creando el repositorio local



*Lo primero será ubicarnos en donde
queremos crear el repositorio y
posteriormente escribir el siguiente comando:*

git init



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.



Agregando
nuestra identidad



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



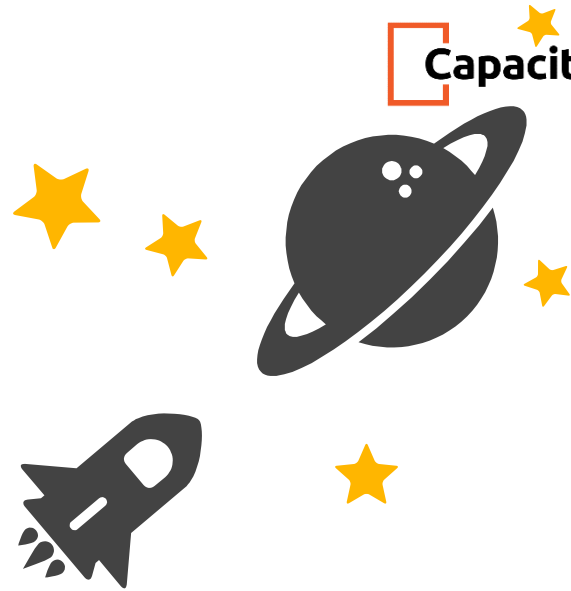
Asignando
nuestro
repositorio
remoto

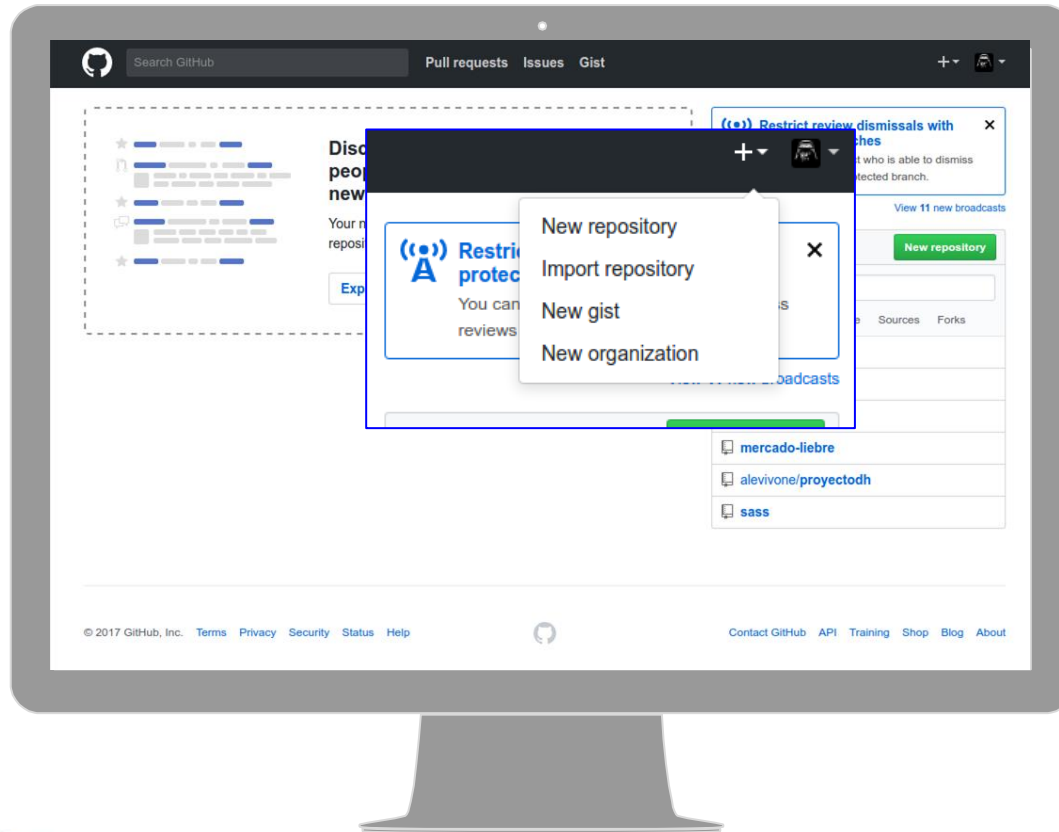
1ero

Vamos a crear
nuestro repo
en Github

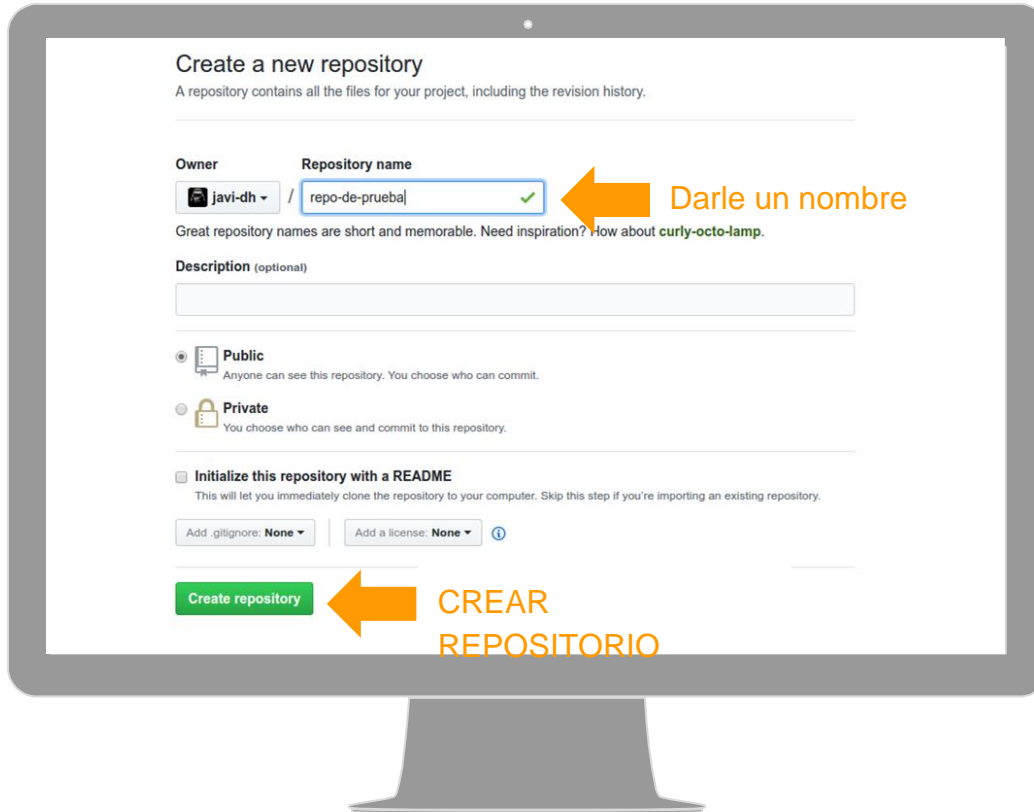
¿Cómo creamos el repositorio remoto?

Para ello vamos a utilizar nuestra cuenta de GitHub que creamos previamente.





Logueados en nuestra cuenta de GitHub vamos al ícono + y ahí elegimos la opción **New Repository**.




The screenshot shows the GitHub 'Create a new repository' form. An orange arrow points to the 'Repository name' field, which contains 'repo-de-prueba' and a green checkmark. Another orange arrow points to the 'Create repository' button at the bottom left.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner **Repository name**

 javi-dh / ✓

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

Description (optional)

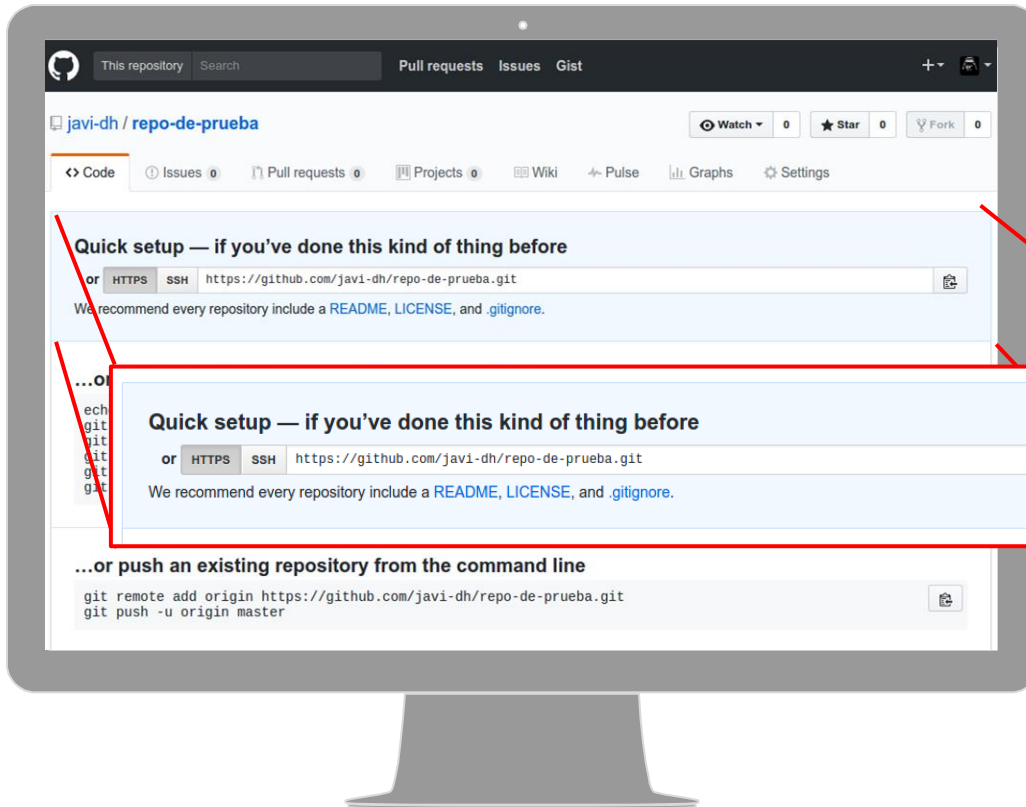
☒ **Public**
Anyone can see this repository. You choose who can commit.

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

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

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



Adicionando
nuestros archivos
al repositorio
local

“ Hasta el momento, nuestros archivos no han sido agregados **temporalmente** al repositorio (**stage**) para ello tendremos que escribir el siguiente comando:

git add .

ó

git add archivo.txt




git add --all

*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.*



Testeando el
status de nuestro
repositorio



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

git status



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**).*



Agregando
oficialmente
los archivos al
stage



*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.*



Enviando nuestros
archivos del
repositorio local al
repositorio
remoto



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

git push origin master



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.*

Gracias!



¿Consultas?