

## UD1

# Tutorial de Git con GitHub

### ¿Qué vamos a aprender?:

- Crear y usar un repositorio en GitHub
- Comenzar y manejar una nueva “Branch” (rama) sobre un repositorio.
- Hacer cambios a un archivo y registrar (push) los cambios como un “commit”.
- Abrir y fusionar (Merge) un pull request.

### ¿Qué es GitHub?

GitHub es una plataforma de almacenamiento de código mediante control de versiones. Este te permite a ti y otros trabajadores trabajar juntos en un proyecto desde cualquier parte.

Este tutorial te enseñará las 4 actividades esenciales: Crear un repositorios y branches, hacer commits , pull request y merges (fusionar).

### No vamos a codificar ni una sola línea!

Para completar este tutorial, necesitarás una cuenta en GitHub.com y acceso a Internet. Como todo se va a realizar en GitHub no necesitarás instalar Git en tu ordenador.




## Paso 1. Crear un Repositorio

Un **repositorio** es usado usualmente para organizar un **único** proyecto. Los repositorios pueden contener directorios y archivos, imágenes, videos, etc, **cualquier cosa que un proyecto pueda necesitar**. Siempre se recomienda incluir un archivo **README**, el cual es un archivo con información sobre el proyecto. GitHub hace muy fácil añadir este archivo al proyecto a la misma vez que este se crea. También se ofrecerá añadir un tipo de **licencia** a nuestro proyecto.

### Para crear un nuevo repositorio

1. En la esquina superior derecha, cerca del avatar, haz click y selecciona **New repository**.
2. Da un nombre a tu repositorio.
3. Escribe una descripción.
4. Selecciona **“Inicializar este repositorio con README”**.


**Owner** **Repository name**


PUBLIC   **hubot** /  

Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.

**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 allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

|  

---

**Create repository**

Click **Create repository**. 

## Paso 2. Crear una Branch / Rama

Crear **ramas** es la manera en la que se debe trabajar para mantener diferentes versiones de un mismo repositorio.

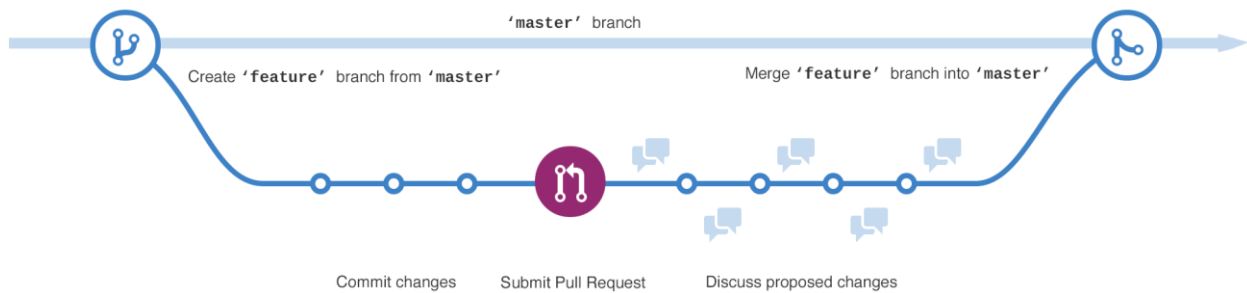
Por defecto, un repositorio tiene una rama llamada **master** la cual es considerada como la rama raíz. Usar ramas te permitirá hacer y crear nuevas cosas, probarlas y decidir finalmente si debes o no fusionarla con la rama **master**.

Cuando creas una rama de la rama **master**, lo que se hace es hacer una copia, una foto, de la rama principal.

Si antes de que tú fusiones una rama con la rama principal, hay otro trabajador que ha hecho cambios en la **master**, antes de hacer la fusión con la master de tu rama, deberías mezclar en tu rama los cambios realizados por el otro trabajador para comprobar si hay conflictos.

Este diagrama muestra:

- La rama principal/master.
- Una nueva rama llamada “feature”
- Los cambios/modificaciones que la rama “feature” tiene antes de que se haya hecho un “merge” con la rama master.



Un símil que podemos hacer para entender las ramas es si alguna vez has hecho copias de seguridad de una principal cambiándole el nombre y haciendo copias de seguridad del mismo.

- story.txt
- story-joe-edit.txt
- story-joe-edit-reviewed.txt

### Para crear una nueva branch/rama.

1. Ir al repositorio.
2. Haz click Click en el menú desplegable donde dice **branch: master**.
3. Escribe un nombre de rama nueva.
4. Selecciona el texto azul **Create branch** box o pulsa “Enter” en tu teclado.

## Just another repository — Edit

1 commit

1 branch

branch: master ▾ **hello-world** /

Initial commit

**hubot** authored just now

**README.md** Initial

**README.md**

Ahora tendremos 2 ramas: **master** y la nueva rama. Las dos deben de contener exactamente el mismo contenido.

## Paso 3. Realiza cambios y haz commit de tus cambios.

En GitHub, salvar cambios es llamado “commits” . Cada commit debe tener asociado un mensaje, el cual debe ser una descripción que explique los cambios que se han realizado. Es muy útil para que otros trabajadores puedan ver todo el historial de cambios que has realizado y gracias a la explicación, los entiendan.

### Haz cambios y realiza un “commit”:

1. Haz click en el archivo README.md .
2. Haz click sobre el icono del lápiz en la esquina derecha para abrir el editor en línea.
3. Haz algún cambio en el archivo.
4. Escribe un mensaje de commit que describa tus cambios.
5. Haz click en el botón de **Commit changes**.

The screenshot shows the GitHub interface for the repository 'hubot / hello-world'. At the top, there are buttons for 'Unwatch', 'Star', and 'Fork'. Below this is a navigation bar with links for 'Code', 'Issues', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main content area shows the 'hello-world / README.md' file being edited. The editor has a line number margin on the left and a preview pane on the right. The code in the editor is:

```
1 # hello-world
2
3 Hi Humans!
4
5 Hubot here, I like Node.js and Coffeescript (that's what I'm made of!).
6 I've had tacos on the moon and find them far superior to Earth tacos.
7
```

Below the editor is the 'Commit changes' dialog. It has a title bar with a Hubot icon and the text 'Commit changes'. Inside the dialog, there is a text input field with the value 'Finish README' and a larger text area with the value 'And mention moon tacos'. At the bottom of the dialog, there are two radio buttons: the first is selected and labeled 'Commit directly to the readme-edits branch', and the second is labeled 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' At the very bottom of the dialog are two buttons: 'Commit changes' (green) and 'Cancel' (red).

Este cambio estará guardado solo en el archivo README de la rama en la que estás situado, así que esta rama tiene diferente contenido a la rama master.

## Paso 4. Abre un Pull Request

Ahora que tenemos cambios en una rama diferentes a los de la master, puedes pensar en hacer un **pull request**.

**Pull Request** son el corazón de la colaboración en GitHub. Cuando abres un “pull request”, estás proponiendo tus cambios y preguntando a alguien que revise estos cambios y los fusiones dentro de otra rama. “Pull Request” muestra las diferencias entre las 2 ramas.

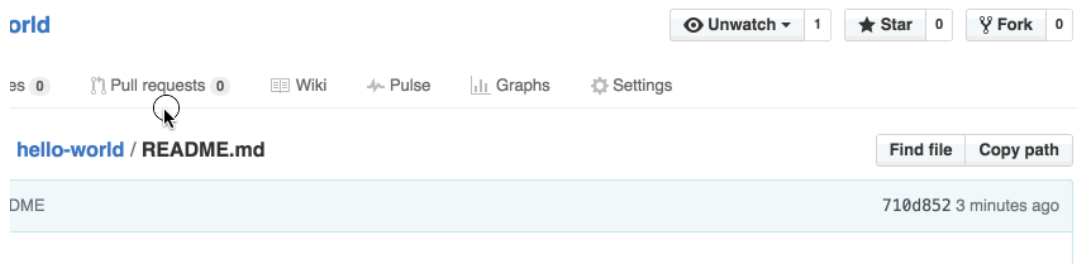
Los cambios que se **añaden** se muestran en **verde**, y los cambios que **eliminan** parte del contenido previo aparecen en **rojo**.

Tan pronto como hagas un **commit**, estás en disposición de abrir un **pull request** y empezar una discusión, incluso antes de que el código esté terminado por completo.

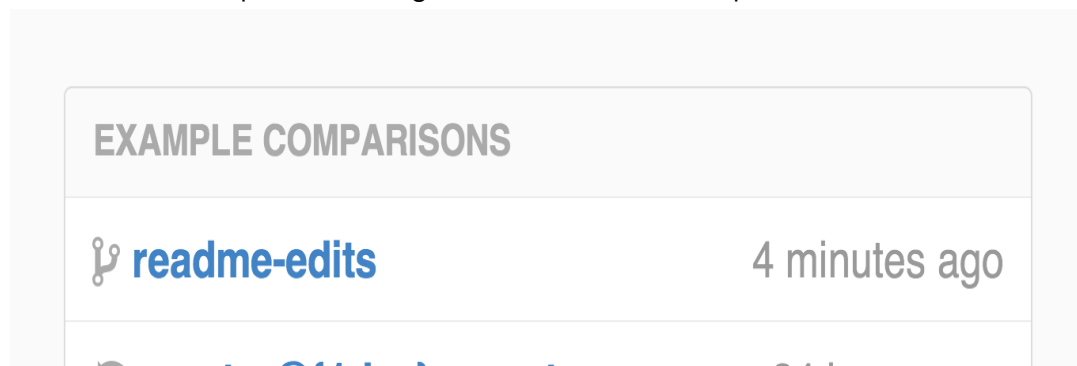
Es más, incluso para aprender a usar el “flujo de trabajo” con git, podrías ensayar a abrir pull request y hacer fusiones (merge) con tu propia rama.

**Abre un Pull Request para los cambios en el archivo README:**

**1. Click en la pestaña de Pull Request**, entonces clicka en el botón verde **New pull request**.





**2. En el cuadro Example Comparisons** verás la comparación entre tu rama, y la rama master a la que hemos sugerido realizar el Pull Request.






**3. Debes de mirar los cambios en el cuadro de comparación y estar seguro que son**


estos los cambios que quieres fusionar.


 **1** commit

 **1** file changed


 Commits on Oct 27, 2014


  **hubot** Finish READ

 Showing **1** changed file with **1** addition and **1** deletion

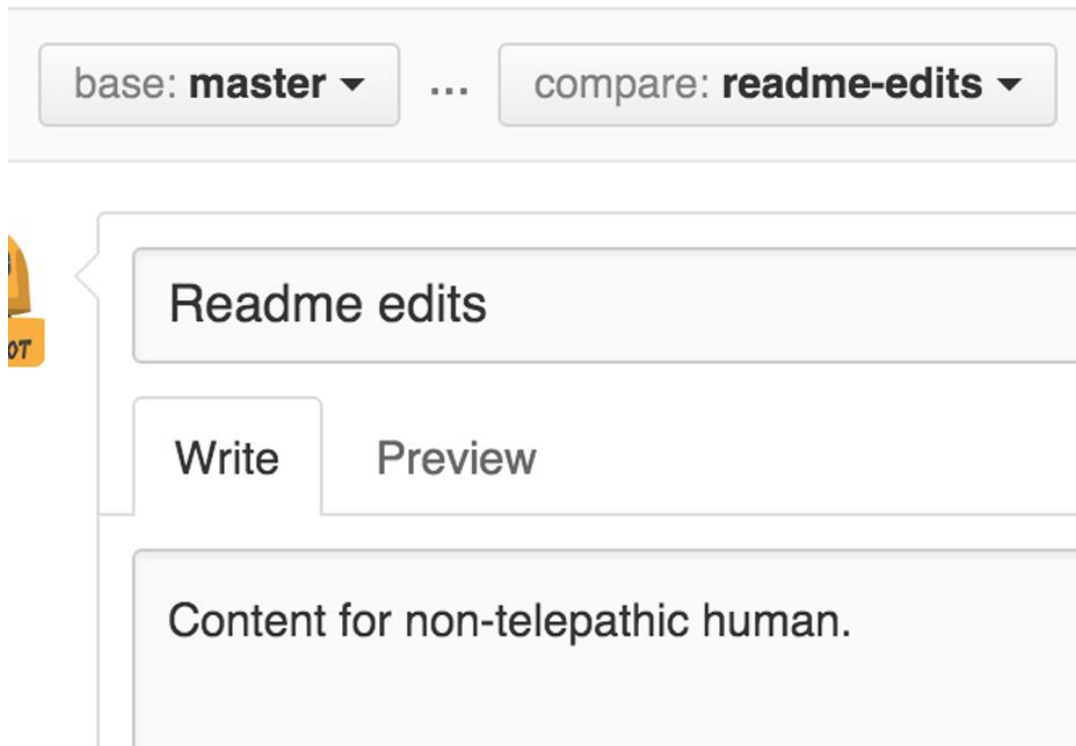
2  README.md		
...	...	@@ -1,4 +1,4 @@
1	1	hello-world
2	2	=====
3	3	
4		-Just another repository
	4	+Hubot here, I like Node.js and them far superior to Earth taco

4. Cuando estés satisfecho con los cambios haz click sobre el botón verde. **Create Pull Request Button.**

 base: **master** ... compare: **readme-ed**

 **Create pull request** Discuss and review the changes

Debes de dar un título al **pull request** y crealo definitivamente.



**Consejo:** Puedes usar [emoji](#) and [drag and drop images and gifs](#) en los comentarios del Pull Requests.

## Paso 5. Fusiona (Merge) tu Pull Request

En este último paso, es hora de fusionar tus cambios de forma que la tu rama y el master contengan el mismo contenido.

1. Click el botón verde **Merge pull request** para fusionar los cambios dentro de la rama master.
2. Click **Confirm merge**.
3. Y ahora puedes incluso borrar esa rama, ya que todos sus cambios están en la master, y no tiene ningún sentido mantener esta branche. Puedes hacerlo en el botón **Delete branch** en la caja color púrpura.



**This branch has no conflicts with the base branch**

Merging can be performed automatically.



**Merge pull request**

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



**Pull request successfully merged and closed**

You're all set—the `readme-edits` branch can be safely deleted.



**Delete branch**

**Fin!**

