

# Iniciación a GitHub. Práctica III

## Tarea 1.

[cómo crear una cuenta GitHub](#)

<https://guides.github.com/activities/hello-world/>

Aprenderás a:

- Crear una cuenta GitHub
- Crear y usar un repositorio
- Realizar cambios en un archivo y enviarlos a GitHub con un commit
- Iniciar y administrar una nueva rama
- Abrir y fusionar un pull request

## ¿Qué es GitHub?

GitHub es una plataforma de alojamiento de código para control de versiones y colaboración. Te permite a ti y a otros trabajar juntos en proyectos desde cualquier lugar.

Veamos los elementos esenciales de GitHub como repositorios, ramas, commits y Pull Request.

En esta práctica crearás tu propio repositorio **Hello-World** y aprenderá el flujo de trabajo Pull Request de GitHub, una forma habitual de crear y revisar código.

## Paso 0. Crear una cuenta

Para iniciar esta práctica, necesitas una [cuenta de GitHub.com](#). Créala!

Aquí puedes consultar más detalles de [cómo crear una cuenta GitHub](#)

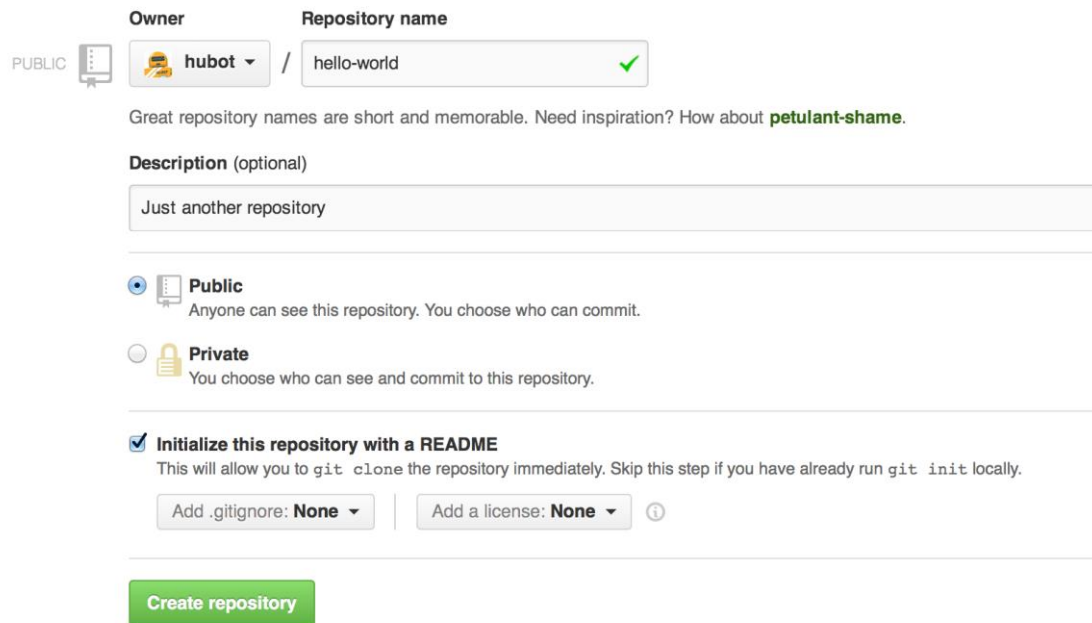
## Paso 1. Crear un repositorio

Un **repositorio** se usa generalmente para organizar un solo proyecto. Los repositorios pueden contener carpetas y archivos, imágenes, videos, hojas de cálculo y conjuntos de datos, cualquier cosa que tu proyecto necesite. Recomendamos incluir un **archivo README** o un archivo con información sobre tu proyecto. GitHub facilita agregar uno al mismo tiempo que crea su nuevo repositorio. También ofrece otras opciones comunes, como un archivo de licencia.

Tu repositorio de **hello-world** puede ser un lugar donde almacenar ideas, recursos o incluso compartir y discutir cosas con otros.

## Para crear un nuevo repositorio

1. En la esquina superior derecha, junto a tu avatar o icono de identificación, haz clic en + y luego selecciona **Nuevo repositorio**.
2. Nombra tu repositorio **hello-world**.
3. Escribe una breve descripción.
4. Selecciona **Inicializar este repositorio con un archivo README**.



The screenshot shows the GitHub 'Create new repository' interface. At the top, there's a 'PUBLIC' label and a 'Repository name' field containing 'hello-world' with a green checkmark. Below this is a 'Description (optional)' text area with the placeholder text 'Just another repository'. Further down, there are two radio button options: 'Public' (selected) and 'Private'. Below these is a checkbox labeled 'Initialize this repository with a README' which is also checked. At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None'. A green 'Create repository' button is at the very bottom.

Haz clic en **Crear repositorio**.

## Paso 2. Crear una rama

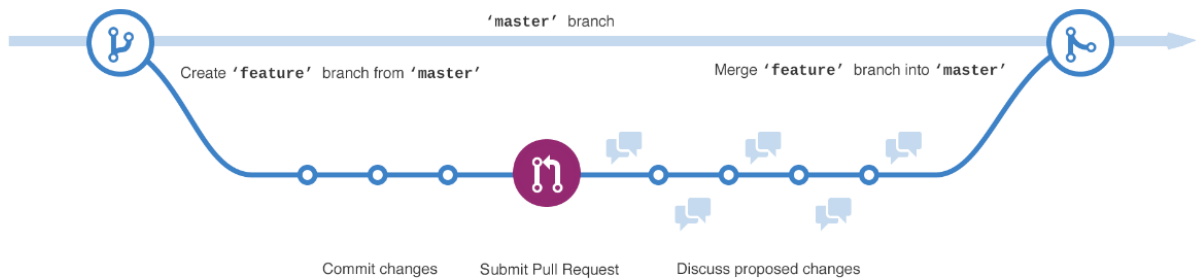
La **ramificación** es la forma de trabajar con diferentes versiones a la vez en un repositorio.

Por defecto, tu repositorio tiene una rama llamada **master** que se considera la rama principal. Usamos ramas para experimentar y hacer ediciones antes de fusionarlas a **master**.

Cuando creas una rama a partir de la rama **master**, estás haciendo una copia o una instantánea de la **master** tal como estaba en ese momento. Si alguien más realiza cambios en la rama **master** mientras trabajabas en tu rama, podrás obtener esas actualizaciones.

Este diagrama muestra:

- La rama **master**
- Una nueva rama llamada **feature** (porque estamos trabajando nuevas "características" en esta rama)
- La evolución que lleva esa **feature** antes de fusionarse en **master**



¿Alguna vez has guardado diferentes versiones de un archivo? Algo como:

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



Las ramas logran objetivos similares en los repositorios de GitHub.



Aquí en GitHub, nuestros desarrolladores, escritores y diseñadores usan ramas para mantener las correcciones de errores y variaciones del trabajo, separadas de nuestra rama **master** (producción). Cuando un cambio está listo, fusionan su rama en **master**.


## Para crear una nueva rama


1. Ve a tu nuevo repositorio **hello-world**.
2. Haz clic en el menú desplegable en la parte superior de la lista de archivos que dice **branch: master**.
3. Escribe un nombre de rama, **readme-edits**, en el nuevo cuadro de texto de rama.
4. Selecciona el cuadro azul **Crear rama** o presiona "Enter" en su 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 tienes dos ramas **master** y **readme-edits**. ¡Se ven exactamente iguales, pero no por mucho tiempo! A continuación, agregaremos nuestros cambios a la nueva rama.

## Paso 3. Realizar y confirmar cambios

Asegúrate ahora de que estás en la vista de código para tu rama de **readme-edits**, que es una copia de **master**. Hagamos algunas ediciones.

En GitHub, los cambios guardados se llaman commits. Cada commit tiene un *mensaje* asociado, que es una descripción que explica por qué se realizó un cambio en particular. Los mensajes de los commits describen el historial de sus cambios, para que otros contribuyentes puedan entender lo que ha hecho y por qué.

### Realizar y confirmar cambios

1. Haz clic en el archivo README.md.
2. Haz clic en el icono de lápiz en la esquina superior derecha de la vista de archivo para editar.
3. En el editor, escribe un poco sobre ti.
4. Escribe un mensaje de confirmación que describa sus cambios.
5. Haz clic **en el botón Commit changes**.

The screenshot shows the GitHub web interface for the repository 'hubot / hello-world'. At the top, there are navigation tabs for 'Code', 'Issues', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. Below these, the file 'hello-world / README.md' is selected. The editor shows the content of the README file with line numbers 1 through 7. The text includes a greeting, a mention of 'Hubot', and a sentence about 'tacos on the moon'. Below the editor, the 'Commit changes' dialog is open. It has a title bar with 'Commit changes' and a 'Cancel' button. The dialog contains a text input field with the placeholder 'Finish README', a larger text area with the placeholder 'And mention moon tacos', and two radio buttons. The first radio button is selected and labeled 'Commit directly to the readme-edits branch'. The second radio button is labeled 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' At the bottom of the dialog, there are two buttons: 'Commit changes' (green) and 'Cancel' (grey).

Estos cambios se realizarán solo en el archivo README en la rama de **readme-edits**, por lo que ahora esta rama contiene contenido diferente a la **master**.

## Paso 4. Abre una Pull Request

Ahora que tienes diferencias entre la segunda rama y **master**, puede hacer un Pull Request (solicitud de incorporación de cambios).

Las Pull Request son el corazón de la colaboración en GitHub. Cuando inicias una Pull Request, estás proponiendo tus cambios y solicitando que alguien revise y “suba” tu contribución y la combine en su rama. Las Pull Request muestran los *diff* o diferencias entre el contenido de ambas ramas. Los cambios, adiciones (y substracciones), se muestran en verde (y rojo).

Tan pronto como hagas un commit, puedes hacer un Pull Request e iniciar una discusión, incluso antes de que finalices el código.

También puedes abrir un Pull Request en tu propio repositorio y fusionarlas tú mismo. Es una excelente manera de aprender el flujo de GitHub antes de trabajar en proyectos más grandes. Esto es lo que vamos a hacer ahora.

### Abre una Pull Request para los cambios hechos en el archivo README

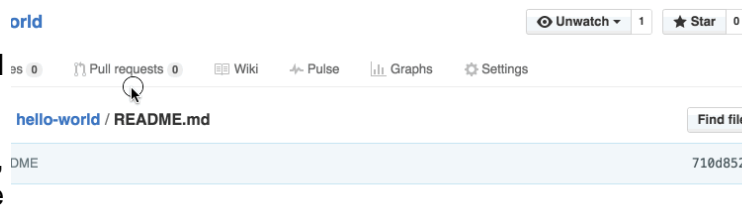
#### Paso

#### Captura de pantalla

Atención: Asegúrate de que estás situado en la rama que quieres fusionar: **readme-edits**

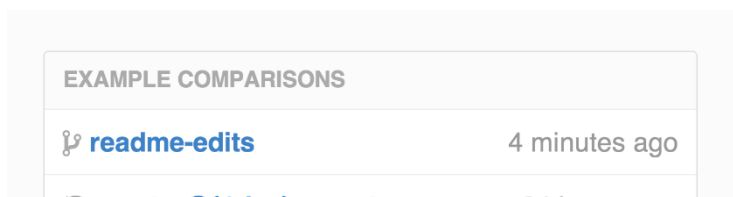
Clica en el icono de Pull Request,

Después desde esta página, haz clic en el botón verde **New pull request**.



En el cuadro Example Comparisons,

selecciona tu rama, **readme-edits**, para compararla con **master** (la rama inicial).



1 commit
1 file c

Commits on Oct 27, 2014

hubot
Finish R

Revisa tus cambios viendo los diff en la página Comparar, asegúrate de que sean los que deseas enviar.

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 ta

Cuando estés seguro de que estos son los cambios que deseas enviar, haz clic en el botón verde grande **Create pull request**.

base: master ... compare: readme-ed

Create pull request
Discuss and review the

base: master ... compare: readme-edits

Da un nombre al pull request y escribe una breve descripción de tus cambios.

Readme edits

Write Preview


Content for non-telepathic human.


Cuando haya terminado con tu mensaje, haz clic en **Create pull request**.

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


En este último paso, es hora de unir los cambios, fusionando tu rama de **readme-edits** en la rama **master**.


1. Haz clic en el botón verde de **Merge pull Request** para fusionar los cambios en **master**.
2. Haz clic en **Confirm merge**.
3. Continúa y elimina la rama, ya que sus cambios se han incorporado, con el botón **Delete branch** en el cuadro morado.



**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**

# GitHub: Fork y Pull Requests

## Tarea 2

<http://aprendegit.com/fork-de-repositorios-para-que-sirve/> y siguientes

<http://aprendegit.com/que-es-un-pull-request/>

<https://git-scm.com/book/es/v2/GitHub-Participando-en-Proyectos>

<https://www.atlassian.com/es/git/tutorials/making-a-pull-request>

### Trabajo con repositorios: forks y pull requests

1. Haz un *fork* del repositorio localizado en la siguiente url:

<https://github.com/immacabanes/practicaFork.git>

2. Clona este repositorio en tu máquina local
3. Realiza un primer commit para poner tu nombre y apellidos en el fichero README.md
4. Modifica los siguientes ficheros realizando un commit después de cada una de las tareas
  2. Añadir 3 libros
  3. Añadir 3 películas
  4. Añadir 3 discos
5. Haz capturas de pantalla de los comandos que has utilizado para clonar el repositorio en local, hacer los commits y subir los cambios a GitHub.
6. Incluye las capturas de pantalla en el repositorio dentro de la carpeta capturas. Añádelas también al repositorio de manera que queden guardadas en tu repositorio en GitHub.
7. Realiza una pull request indicando en el mensaje que has completado la tarea.