

# Módulo 4. GitHub. Trabajo colaborativo integrado con Git



☰ Introducción

☰ 1. ¿Qué ofrece Github?

☰ Referencias

# Introducción

---

GitHub es una plataforma creada para alojar el código de aplicaciones y proyectos de desarrollo, que fue adquirida por Microsoft en junio de 2018. Está orientada a que los desarrolladores publiquen el código de sus aplicaciones y herramientas, y permite que otras personas no solo descarguen esos proyectos, sino que también accedan a la información del repositorio y colaboren en su desarrollo.

Como su nombre indica, la plataforma utiliza el sistema de control de versiones Git, diseñado por Linus Torvalds. Un sistema de control de versiones permite administrar un proyecto registrando los cambios realizados en el código a lo largo del tiempo, lo que facilita la organización de las distintas versiones y evita confusiones. Al conservar copias de cada versión, es posible recuperar estados anteriores cuando se realiza una actualización.

Git es uno de estos sistemas de control. Permite comparar archivos para identificar diferencias entre versiones, restaurar

versiones previas en caso de errores y fusionar cambios provenientes de distintas líneas de desarrollo. También posibilita el trabajo con diferentes ramas dentro de un mismo proyecto, como una rama de desarrollo para incorporar nuevas funcionalidades o una rama de producción destinada a corregir errores. Entre las características de la plataforma se encuentra su amplia adopción a nivel mundial, su funcionamiento multiplataforma y la disponibilidad de diversas interfaces de usuario.

**Resumidamente, GitHub es un portal que gestiona proyectos que utilizan Git. Además de permitir la visualización y descarga del código en sus distintas versiones, facilita la colaboración entre desarrolladores y usuarios, promoviendo el trabajo conjunto, la revisión de código y la gestión organizada de proyectos.**

CONTINUAR

# 1. ¿Qué ofrece Github?

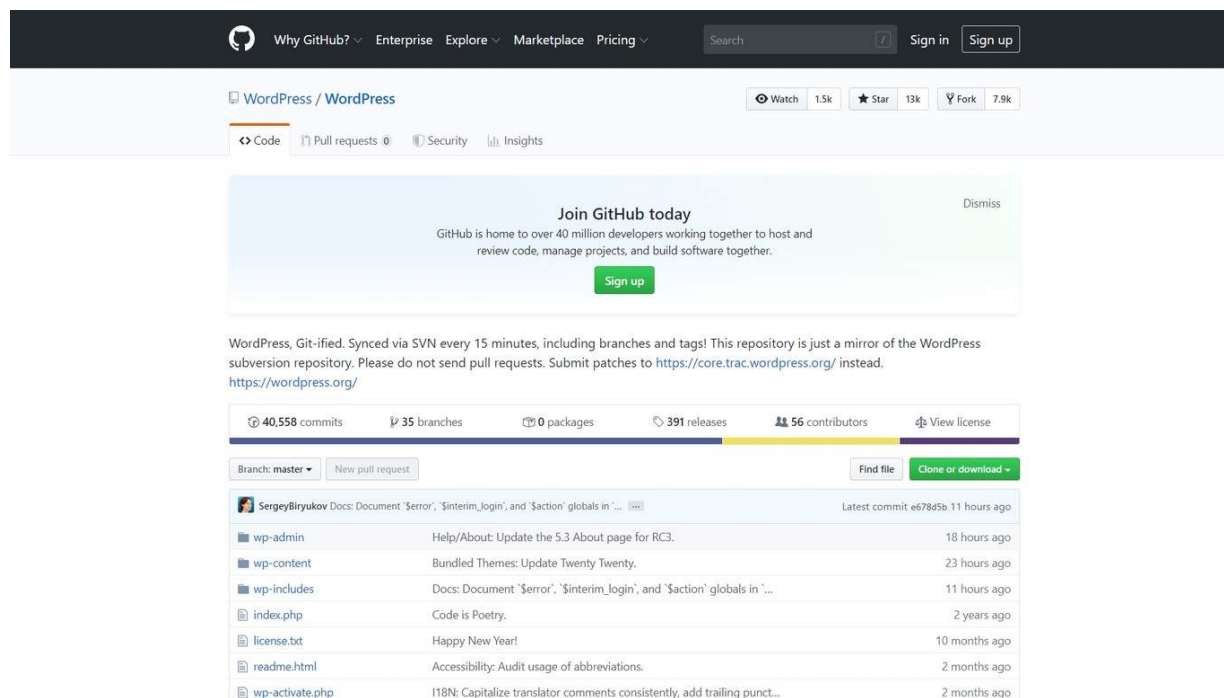
---

GitHub permite a los desarrolladores alojar proyectos mediante la creación de repositorios. Ofrece la posibilidad de crear repositorios gratuitos, tradicionalmente asociados a proyectos de código abierto. Asimismo, dispone de planes de pago que permiten mantener repositorios privados.

La plataforma facilita la interacción entre usuarios y desarrolladores: se pueden realizar comentarios sobre el código, proponer mejoras, contribuir con cambios y reportar errores para su posterior corrección.

Además, GitHub incorpora herramientas complementarias que amplían las funcionalidades propias de Git. Por ejemplo, permite crear una wiki para cada proyecto, donde se puede documentar información relevante y registrar los cambios de las distintas versiones.

## Figura 1. Interfaz de un repositorio en GitHub



The screenshot shows the GitHub interface for the WordPress repository. At the top, there is a navigation bar with links for 'Why GitHub?', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. A search bar and 'Sign in'/'Sign up' buttons are also present. Below the navigation bar, the repository name 'WordPress / WordPress' is displayed, along with statistics: 'Watch 1.5k', 'Star 13k', and 'Fork 7.9k'. The main content area features a 'Join GitHub today' banner with a 'Sign up' button. Below the banner, there is a note about the repository being a mirror of the WordPress subversion repository. A summary bar shows '40,558 commits', '35 branches', '0 packages', '391 releases', and '56 contributors'. The file list shows the current branch 'master' and a list of files with their commit messages and timestamps.

File	Commit Message	Time Ago
wp-admin	Help/About: Update the 5.3 About page for RC3.	18 hours ago
wp-content	Bundled Themes: Update Twenty Twenty.	23 hours ago
wp-includes	Docs: Document '\$error', '\$interim_login', and '\$action' globals in '...	11 hours ago
index.php	Code is Poetry.	2 years ago
license.txt	Happy New Year!	10 months ago
readme.html	Accessibility: Audit usage of abbreviations.	2 months ago
wp-activate.php	I18N: Capitalize translator comments consistently, add trailing punct...	2 months ago

Fuente: Fernández, 2019, <https://goo.su/0HCQCC>

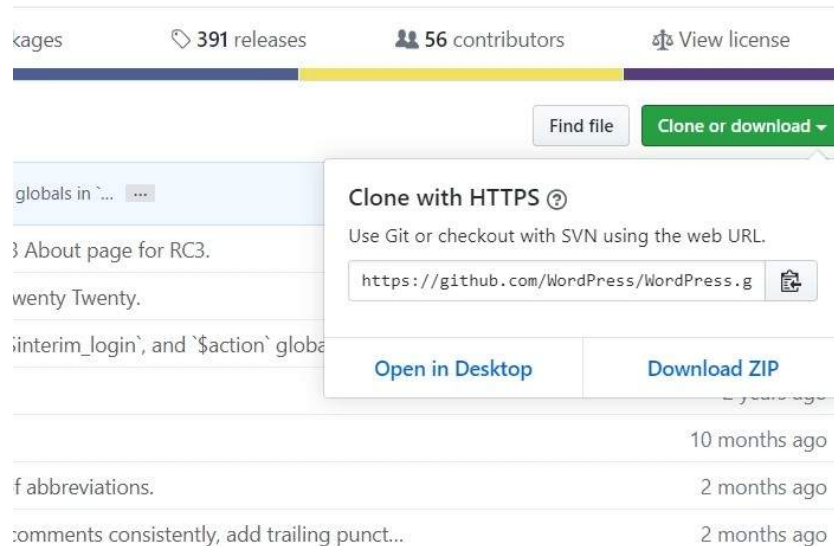
También dispone de un sistema de seguimiento de problemas, que permite a otras personas proponer mejoras, sugerencias y optimizaciones en los proyectos. Asimismo, ofrece una herramienta de revisión de código que no solo posibilita examinar el código fuente, sino también añadir anotaciones para que el autor del proyecto —o cualquier colaborador— pueda revisarlas posteriormente. A partir de estas anotaciones pueden generarse discusiones orientadas a perfeccionar el desarrollo.

**La plataforma incluye, además, gráficos que permiten visualizar la actividad del proyecto y sus bifurcaciones, así como las actualizaciones realizadas desde la primera versión y los cambios introducidos a lo largo del tiempo.**

Por último, incorpora funciones propias de una red social, como la posibilidad de seguir a otros desarrolladores y mantenerse al tanto de sus actualizaciones.

## Figura 2. Opciones para clonar o descargar un repositorio en GitHub

ding branches and tags! This repository is just a mirror of the WordPress  
omit patches to <https://core.trac.wordpress.org/> instead.



Fuente: Fernández, 2019, <https://goo.su/OHCQCC>

Cuando una aplicación dirige a su repositorio en GitHub para su descarga, la interfaz puede resultar confusa. En la parte superior derecha aparece el botón «Code», que incluye la opción «Download ZIP»; sin embargo, esta opción descarga el código fuente del proyecto, no una versión instalable del programa. Si se busca una versión lista para usar —por ejemplo, un instalador o un archivo ejecutable—, conviene revisar la sección «Releases» o la descripción del repositorio,

ya que allí suele incluirse el enlace correspondiente (Fernández, 2019).

### **¿Cómo funcionan Git y GitHub de manera conjunta?** —

Al cargar archivos en GitHub, estos se almacenan en un repositorio de Git. Cada vez que se realizan cambios —es decir, confirmaciones (*commit*)—, Git registra y gestiona automáticamente esas modificaciones.

Muchas acciones relacionadas con Git pueden realizarse directamente desde la interfaz web de GitHub, como crear un repositorio, generar ramas, cargar archivos o editarlos.

No obstante, lo habitual es trabajar de manera local en la propia computadora y, posteriormente, sincronizar los cambios y el historial de Git con el repositorio remoto alojado en GitHub. Para ello, pueden utilizarse distintas herramientas, como GitHub Desktop.

Cuando se colabora con otras personas en un mismo repositorio, el flujo de trabajo suele repetirse de forma continua:

- obtener los cambios más recientes realizados por otros colaboradores desde el repositorio remoto;
- enviar los cambios propios al repositorio remoto.

Git se encarga de integrar estos cambios de manera coherente, mientras que GitHub facilita su gestión mediante herramientas como las «solicitudes de cambios» (*pull requests*), que permiten

revisar y discutir las modificaciones antes de incorporarlas al proyecto.

## ¿Por dónde empezar? —

Si no tienes experiencia con GitHub y Git, sigue estos pasos:

- crea una cuenta en GitHub;
- aprende el «flujo de GitHub» y los principios básicos del trabajo colaborativo, como ramas, confirmaciones (*commit*), solicitudes de cambios (*pull requests*) y fusiones (*merge*);
- personaliza tu perfil para compartir tus intereses y aptitudes;
- explora la plataforma para encontrar inspiración para tus propios proyectos y conectar con otros usuarios;
- infórmate sobre cómo descargar código para utilizarlo o estudiarlo;
- conoce el procedimiento para cargar en un repositorio de GitHub un proyecto en el que estés trabajando.

## Creación de una cuenta en GitHub —

Para empezar a usar GitHub, debes crear una cuenta personal gratuita y verificar tu dirección de correo electrónico. También puedes autenticarte con Google o Apple, que son los proveedores de inicio de sesión compatibles al crear tu cuenta en GitHub. En el caso de usuarios de iOS, aunque esté habilitada la opción «Ocultar mi dirección de correo electrónico» en la cuenta de Apple, el uso de

este método de autenticación generará una nueva cuenta de GitHub.

Cada persona que utiliza GitHub inicia sesión con una cuenta de usuario. Esta cuenta constituye tu identidad en la plataforma e incluye un nombre de usuario y un perfil.

### **Registro de una nueva cuenta personal**

Para registrarte, sigue estos pasos:

- Visita <https://github.com>.
- Haz clic en «Registrarse». También puedes seleccionar «Continuar con Google» para utilizar el inicio de sesión social,
- Sigue las instrucciones que aparecen en pantalla para completar la creación de tu cuenta.

Durante el proceso, deberás verificar tu dirección de correo electrónico. Sin esta verificación, no podrás realizar algunas acciones básicas, como crear un repositorio.

#### **PASO 1. CREACIÓN DE UN REPOSITORIO**

#### **PASO 2. CREACIÓN DE UNA RAMA**

El primer paso consiste en crear un repositorio. Puede considerarse un repositorio como una carpeta que contiene elementos relacionados, como archivos, imágenes, videos u otras carpetas. Por lo general, agrupa todos los componentes que forman parte de un mismo proyecto. Con frecuencia, los repositorios incluyen un archivo

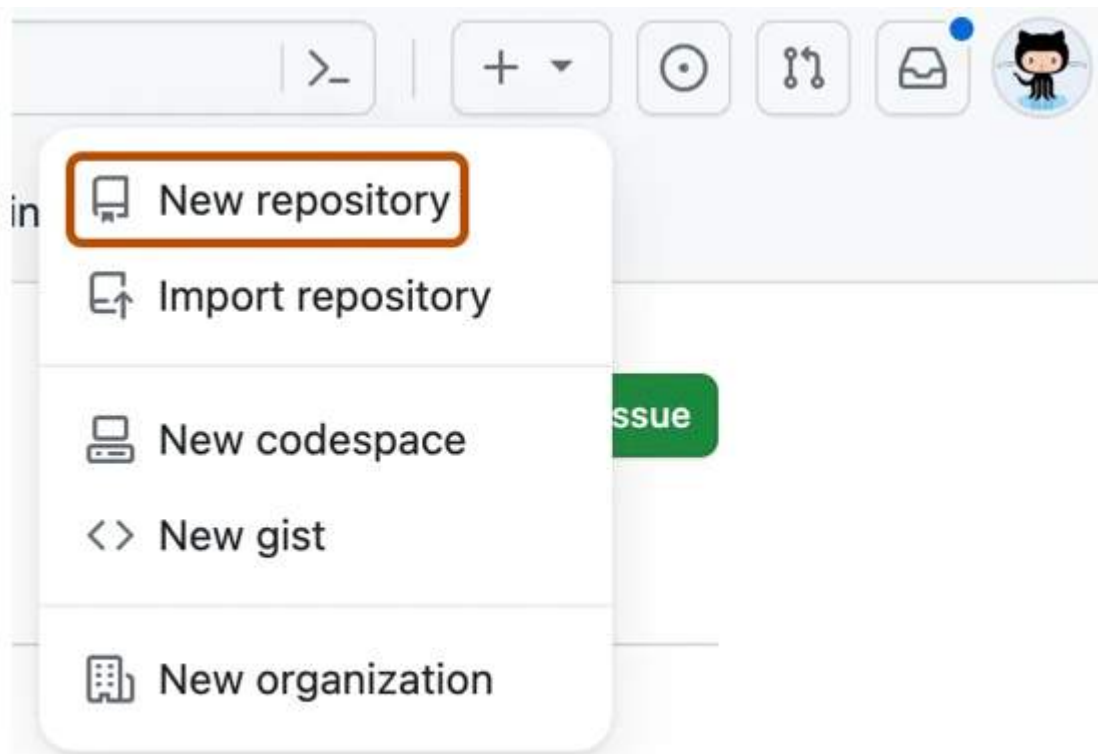
«LÉAME» (*README*), que contiene información relevante sobre el proyecto. Estos archivos suelen escribirse en *Markdown*, un lenguaje sencillo que permite dar formato al texto. GitHub ofrece la posibilidad de agregar un archivo «LÉAME» en el momento de crear el repositorio. También permite incluir otras opciones habituales, como un archivo de licencia, aunque no es obligatorio seleccionarlás en esta instancia (GitHub, s.f.).

El repositorio *hello-world* puede utilizarse para almacenar ideas, recursos o compartir y debatir contenidos con otras personas.

Para crear el repositorio, sigue estos pasos:

- En la esquina superior derecha de cualquier página, selecciona el icono «+» y haz clic en «New repository».
- En el campo «Repository name», escribe *hello-world*.
- En el campo «Description», redacta una breve descripción, por ejemplo: «Este repositorio es para practicar el flujo de GitHub».
- Elige si el repositorio será público o privado.
- Selecciona «Add a README file» (GitHub, s.f., <https://goo.su/n70iE9C>).
- Haz clic en «Create repository».

### **Figura 3. Menú para crear un nuevo repositorio en GitHub**



Fuente: GitHub, s.f., <https://goo.su/n70iE9C>

#### PASO 1. CREACIÓN DE UN REPOSITORIO

#### PASO 2. CREACIÓN DE UNA RAMA

La ramificación permite disponer de distintas versiones de un mismo repositorio en paralelo. De forma predeterminada, el repositorio cuenta con una rama denominada *main*, que se considera la rama principal. A partir de ella pueden crearse ramas adicionales dentro del repositorio.

La ramificación resulta útil cuando se desea incorporar nuevas funcionalidades sin modificar directamente el código de la rama principal. El trabajo realizado en una rama independiente no se reflejará en *main* hasta que se lleve a cabo la fusión (*merge*). De este

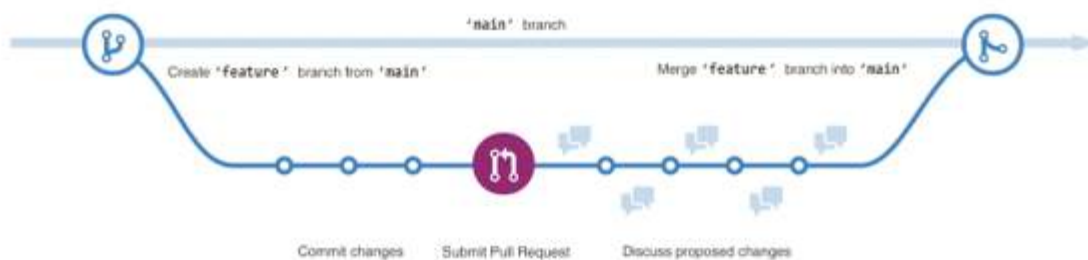
modo, es posible experimentar y efectuar cambios antes de integrarlos en la rama principal.

Al crear una rama desde *main*, se genera una copia o instantánea del estado que tenía en ese momento. Si otras personas realizan cambios en *main* mientras se trabaja en la nueva rama, posteriormente pueden incorporarse esas actualizaciones (GitHub, s.f.).

En el diagrama se observa lo siguiente:

- la rama *main*;
- una nueva rama denominada *feature*;
- el recorrido de la rama *feature* a través de las etapas «Confirmar cambios», «Enviar solicitud de incorporación de cambios» y «Analizar los cambios propuestos», antes de su fusión en *main*.

**Figura 4. Flujo de trabajo con creación de rama y posterior fusión en *main***



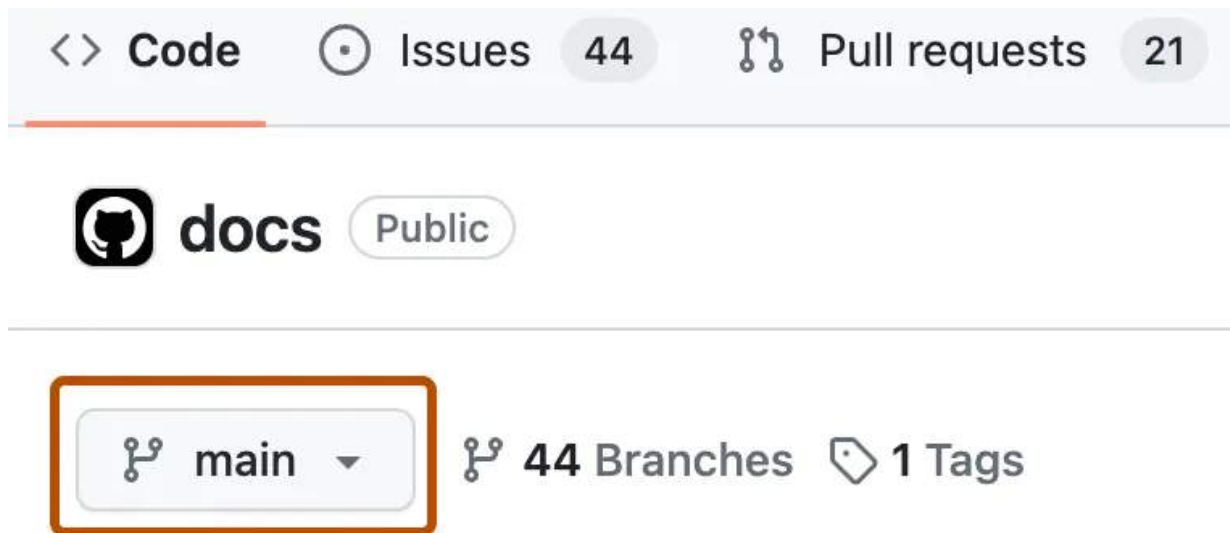
Fuente: GitHub, s.f., <https://goo.su/n70iE9C>

## ¿Cómo crear una rama?

Para crear una nueva rama en el repositorio, sigue estos pasos:

1. Haz clic en la pestaña «Code» del repositorio.
2. Encima de la lista de archivos, haz clic en el menú desplegable que muestra la rama actual, generalmente «main».

### Figura 5. Selector de rama en la vista principal del repositorio

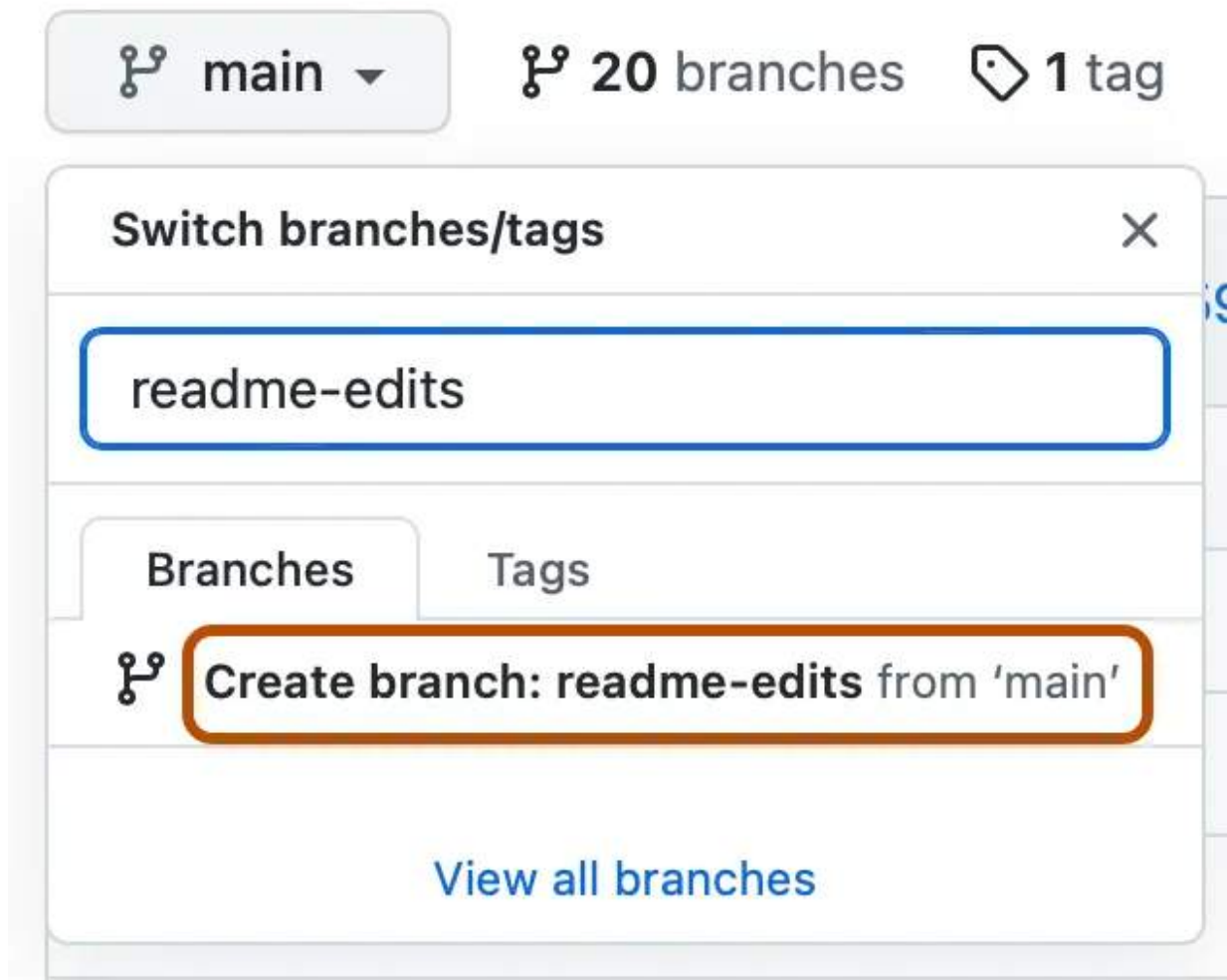


Fuente: GitHub, s.f., <https://goo.su/NyQxiz>

3. Escribe el nombre de la nueva rama, por ejemplo *readme-edits*, en el cuadro de texto.

4. Selecciona la opción «Create branch: readme-edits from 'main'».

**Figura 6. Creación de la rama *readme-edits* desde «main»**



Fuente: GitHub, s.f., <https://goo.su/NyQxiz>

Ahora el repositorio cuenta con dos ramas, *main* y *readme-edits*. En este momento, ambas son idénticas. El siguiente paso consiste en realizar cambios en la nueva rama *readme-edits*.

**PASO 3. REALIZACIÓN Y CONFIRMACIÓN DE CAMBIOS**

**PASO 4: APERTURA DE UNA SOLICITUD DE CAMBIOS**

**PASO 5. FUSIÓN DE LA SOLICITUD DE CAMBIOS**

Al crear una rama en el paso anterior, GitHub dirige automáticamente a la página de código de la nueva rama *readme-edits*, que inicialmente es una copia de *main*.

En el repositorio pueden realizarse y guardarse cambios en los archivos. En GitHub, los cambios guardados se denominan confirmaciones (*commit*). Cada confirmación incluye un mensaje asociado que describe el motivo del cambio. Estos mensajes permiten conservar el historial del proyecto y facilitan que otros colaboradores comprendan qué se modificó y por qué.

Para realizar una modificación en la rama *readme-edits*, sigue estos pasos:

- en la rama *readme-edits*, haz clic en el archivo «[README.md](#)»;
- para editarlo, selecciona el icono del lápiz;

- en el editor, escribe una breve presentación personal;
- haz clic en «Commit changes»;
- en el cuadro correspondiente, redacta un mensaje que describa la modificación realizada;
- selecciona nuevamente «Commit changes» para confirmar.

Estos cambios se aplicarán únicamente al archivo «[README.md](#)» de la rama *readme-edits*, por lo que su contenido será diferente del que se encuentra en la rama *main* (GitHub, s.f.).

**PASO 3. REALIZACIÓN Y CONFIRMACIÓN DE CAMBIOS**

**PASO 4: APERTURA DE UNA SOLICITUD DE CAMBIOS**

**PASO 5. FUSIÓN DE LA SOLICITUD DE CAMBIOS**

Una vez que se han realizado cambios en una rama derivada de *main*, es posible abrir una solicitud de cambios (*pull request*).

Las solicitudes de cambios constituyen el núcleo del trabajo colaborativo en GitHub. Al abrir una, se proponen modificaciones para que otras personas las revisen y, en su caso, las integren mediante una fusión (*merge*). Estas solicitudes muestran las diferencias (*diffs*) entre las dos ramas: las adiciones y eliminaciones aparecen destacadas en distintos colores.

Puede abrirse una solicitud de cambios inmediatamente después de realizar una confirmación, incluso si el desarrollo aún no está finalizado. Esto permite iniciar la revisión y el debate en una etapa temprana.

En este ejercicio, se abrirá una solicitud en el propio repositorio y luego se procederá a su fusión, como práctica del flujo de trabajo en GitHub.

Para ello, sigue estos pasos:

1. Haz clic en la pestaña «Pull requests» del repositorio.
2. Selecciona «New pull request».
3. En el cuadro de comparación, elige la rama *readme-edits* para compararla con *main*.
4. Revisa los cambios en la vista de comparación y verifica que sean los que deseas proponer.
5. Haz clic en «Create pull request».
6. Asigna un título a la solicitud y redacta una breve descripción de los cambios realizados. Es posible incluir emojis o arrastrar imágenes y archivos;
7. Finalmente, selecciona «Create pull request» para completar el proceso.

### **Revisión de una solicitud de incorporación de cambios**

Cuando se comienza a colaborar con otras personas, es habitual solicitar una revisión antes de fusionar los cambios. Este proceso permite que los colaboradores analicen la solicitud de cambios, realicen comentarios o propongan modificaciones antes de integrarla en la rama *main*.

**Figura 7. Vista de comparación de cambios antes de crear la solicitud**

```
Showing 1 changed file with 3 additions and 3 deletions. Split Unified
6 README.md
@@ -1,3 +1,3 @@
1 - # test-area-2
2 - edit1
3 - edit2
1 + # About me
2 +
3 + My name is Mona Lisa.
```

Fuente: GitHub, s.f., <https://goo.su/NyQxivz>

**PASO 3. REALIZACIÓN Y CONFIRMACIÓN DE CAMBIOS**

**PASO 4: APERTURA DE UNA SOLICITUD DE CAMBIOS**

**PASO 5. FUSIÓN DE LA SOLICITUD DE CAMBIOS**

En este último paso, se fusionará la rama *readme-edits* con la rama *main*. Una vez completada la fusión, los cambios realizados en *readme-edits* quedarán incorporados en la rama principal.

En algunos casos, una solicitud de cambios puede generar conflictos con el código existente en *main*. Si esto ocurre, GitHub señalará los archivos en conflicto y no permitirá completar la fusión hasta que se resuelvan. Para solucionarlos, puede realizarse una confirmación que integre los cambios necesarios o utilizar los comentarios de la solicitud para debatir las modificaciones con el equipo.

En este ejercicio no deberían presentarse conflictos, por lo que puede procederse directamente a la fusión. Sigue estos pasos:

- en la parte inferior de la solicitud de cambios, haz clic en «Merge pull request» para integrar los cambios en *main*;

- selecciona «Confirm merge». Aparecerá un mensaje indicando que la solicitud se ha fusionado correctamente y se ha cerrado;
- haz clic en «Delete branch». Una vez fusionada la solicitud y confirmada la integración en *main*, la rama *readme-edits* puede eliminarse de forma segura;
- si deseas realizar nuevas modificaciones, puedes crear otra rama y repetir el proceso;
- finalmente, vuelve a la pestaña «Code» del repositorio *hello-world* para comprobar que los cambios ya se encuentran en la rama *main* (GitHub, s.f.).

## Descarga de archivos desde GitHub

[GitHub.com](https://github.com) alberga millones de proyectos de software de código abierto que puedes copiar, adaptar y utilizar para tus propios fines. Existen distintas formas de obtener una copia de los archivos de un repositorio:

- **Descargar una instantánea en formato ZIP.** Permite obtener una copia de los archivos del repositorio en un archivo comprimido y guardarlo en tu equipo local. Esta opción no conserva el historial de cambios.
- **Clonar un repositorio con Git.** Consiste en crear una copia completa del repositorio en tu

computadora, incluyendo todo su historial. Esta alternativa resulta adecuada si deseas trabajar de manera local, realizar un seguimiento de los cambios y, posteriormente, sincronizarlos con el repositorio alojado en GitHub.

- **Bifurcar un repositorio.** Implica crear una copia del repositorio dentro de tu propia cuenta en GitHub. Puede utilizarse como base para desarrollar un proyecto propio o para proponer cambios al repositorio original. Después de bifurcarlo, también es posible clonarlo para trabajar en él desde el entorno local.

Cada uno de estos métodos responde a necesidades diferentes, que se detallarán en la sección siguiente.

## Carga de un proyecto en GitHub

Al cargar archivos en un repositorio de GitHub, puedes hacer lo siguiente:

- aplicar control de versiones a las modificaciones, lo que permite conservar y

administrar el historial del proyecto;

- disponer de una copia de seguridad del trabajo, ya que los archivos quedan almacenados en la nube;
- vincular el repositorio a tu perfil personal para que otros usuarios puedan consultar tu trabajo;
- compartir y analizar el proyecto con otras personas, ya sea de forma pública o privada.

Veamos a continuación, paso a paso, cómo cargar un proyecto.

## Paso 1

### **Crear un repositorio para el proyecto**

Para crear el repositorio, se deben seguir los pasos mencionados anteriormente.

## Paso 2

### Cargar archivos en el repositorio del proyecto

Hasta este momento, en el repositorio solo debería aparecer el archivo [«README.md»](#) creado al inicializarlo. El siguiente paso consiste en cargar los archivos propios del proyecto.

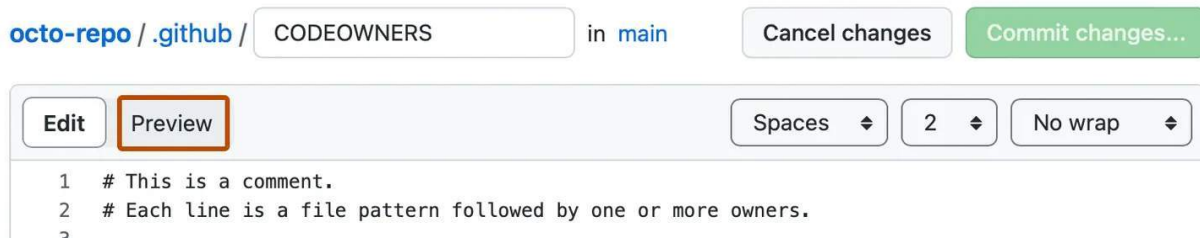
Para ello, sigue estos pasos:

- en la parte derecha de la página, selecciona el menú desplegable «Add file»;
- haz clic en «Upload files»;
- en tu equipo, abre la carpeta que contiene los archivos del proyecto y arrástralos al área del navegador, o selecciónalos manualmente;
- en la sección «Commit changes», ubicada en la parte inferior, elige la opción «Commit directly to the main branch»;
- finalmente, haz clic en «Commit changes» para completar la carga.

Con esto, los archivos quedarán almacenados en el repositorio y pasarán a formar parte del historial del proyecto.

## Paso 3

# Editar el archivo «LÉAME» del repositorio



**Figura 8. Vista previa del archivo [README.md](#) en el editor de GitHub**

El archivo «LÉAME» suele ser el primer elemento que se visualiza al acceder a un repositorio. Generalmente contiene información sobre el propósito del proyecto y su utilidad.

Como se explicó en el tutorial «Hola mundo», el archivo «[README.md](#)» se redacta en Markdown, un lenguaje sencillo que permite dar formato al texto.

En este paso, se editará el archivo «[README.md](#)» para incluir información básica sobre el proyecto.

Sigue estos pasos:

1. Haz clic en «[README.md](#)» dentro de la lista de archivos.
2. En la esquina superior derecha, selecciona el icono del lápiz para abrir el editor.
3. Observa que el archivo incluye información generada previamente, como el nombre y la descripción del repositorio.
4. Elimina el texto existente, excepto el símbolo «#», y escribe un título adecuado para tu proyecto. Ejemplo: «# Acerca de mí. Este es mi primer proyecto en GitHub».
5. Añade una breve descripción del propósito del proyecto o de sus características principales.

6. Encima del contenido nuevo, haz clic en «Preview» para visualizar cómo se mostrará el archivo.
7. Revisa el resultado y vuelve a la pestaña «Edit» si deseas realizar ajustes.
8. Continúa editando y previsualizando hasta que el contenido sea satisfactorio.
9. Haz clic en «Commit changes» en la parte superior derecha.
10. En el cuadro de diálogo, deja el mensaje predeterminado «Update [README.md](#)» y mantén seleccionada la opción «Commit directly to the main branch».
11. Haz clic nuevamente en «Commit changes».

Con estos pasos, se ha creado un repositorio, se han cargado archivos y se ha editado el archivo «LÉAME».

Si el repositorio es público, aparecerá en tu perfil y podrás compartir su dirección con otras personas. GitHub registrará cada modificación como una confirmación (commit), lo que permite gestionar el historial y la evolución del proyecto.

Al realizar cambios, recuerda que puedes crear una rama a partir de main para experimentar sin afectar la versión principal. Cuando los cambios estén listos, abre una solicitud de incorporación de cambios (pull request) para integrarlos en la rama main.

La mayoría de los usuarios prefieren trabajar en sus archivos de manera local —es decir, en su propia computadora— y, posteriormente, sincronizar los cambios realizados con el repositorio remoto alojado en GitHub. Existen diversas herramientas para llevar a cabo este proceso, como GitHub Desktop. Para comenzar, necesitarás:

instalar GitHub Desktop;

clonar el repositorio remoto para disponer de una copia en tu equipo;

sincronizar de forma periódica los cambios locales con el repositorio remoto.

## Conclusión

Al completar este tutorial, aprendiste a crear un proyecto y a realizar una solicitud de incorporación de cambios en GitHub.

En el proceso, adquiriste las siguientes habilidades:

- Crear un repositorio.
- Iniciar y administrar una nueva rama.
- Modificar un archivo y confirmar los cambios en GitHub.
- Abrir y fusionar una solicitud de incorporación de cambios.

**CONTINUAR**

# Referencias

---

**Fernández, Y.** (2019). Qué es Github y qué es lo que le ofrece a los desarrolladores. *Xataka*.  
<https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>

**GitHub**, (s.f.). *Hola mundo*. <https://docs.github.com/es/get-started/start-your-journey/hello-world>

## **Referencias bibliográficas de consulta**

**GitHub**, (s.f.). *Configuración del perfil*.  
<https://docs.github.com/es/get-started/start-your-journey/setting-up-your-profile>

**MoureDev by Brais Moure [MoureDev by Brais Moure]**, (s.f.). *Curso completo de Git y GitHub desde cero para*

*principiantes* [video]. YouTube.

<https://www.youtube.com/watch?v=3GymExBkKjE>

**NetMentor**, (s.f.). *Trabajar en equipo con Git y GitHub.*

<https://www.netmentor.es/entrada/trabajo-equipo-git-github>

CONTINUAR