

¿Cómo publicar en *Github Pages*?

Gran parte de los repositorios que se publican en [Github](#) son públicos - accesibles y abiertos - para todo el mundo. Esto es una excelente noticia para todos los que crean código: todos pueden aprender de todos. Esta es la gracia del servicio, en realidad. Pero antes de empezar: ¿Sabes lo que es Git?

Antes de empezar: aclaremos conceptos

Git es un software de **sistema de control de versiones** que se utiliza habitualmente para el desarrollo de software. Se trata de un sistema de código abierto y gratuito que se utiliza tanto para crear código abierto como comercial, con ventajas claras para las personas implicadas en la realización de código.

Git permite a los desarrolladores ver toda la cronología de sus cambios, decisiones y progresión de cualquier proyecto en un único sitio. Con una herramienta como esta, todos los desarrolladores implicados en un proyecto pueden trabajar de manera asíncrona y, si es necesario, también en diferentes zonas horarias.

Un programa de control de versiones básicamente nos permite saber cosas cómo: ¿Qué cambios se hicieron? ¿Quién los realizó? ¿Cuándo? O ¿para qué eran necesarios estos cambios?

Un **repositorio**, o proyecto Git, incluye toda la colección de ficheros y carpetas asociadas a un proyecto, junto con el historial de revisiones de cada fichero. Trabajar en repositorios mantiene los proyectos de desarrollo organizados y protegidos. A través de plataformas cómo **GitHub**, Git también proporciona más oportunidades para la transparencia y colaboración del proyecto.

GitHub es un **servicio de hosting** de repositorios Git, que ofrece las funcionalidades de Git, más algunas características propias. A diferencia de Git, que es estrictamente una herramienta de línea de comandos, GitHub proporciona una **interfaz gráfica** basada en web y escritorio, así como integración móvil.

Para trabajar en Git es necesario familiarizarse con los conceptos cómo **branch**, **commit**, **pull**, **push**... A continuación encontrarás un pequeño glosario de términos que te pueden ser útiles para familiarizarte con los principales comandos de Git, aunque por la red encontrarás montones de tutoriales y manuales para utilizarlos con todo su potencial. Para introducirte con el tema, te recomendamos esta guía, [Git Handbook](#), en la que encontrarás también todos los **comandos** necesarios para trabajar con Git y Github.

PRINCIPALES COMANDOS DE GIT

branch:

Este comando se utiliza para listar, crear o borrar ramas.

```
git branch
git branch -d <nombre-rama>
```

init :

Este comando se utiliza para crear un nuevo repositorio Git.

```
git init
```

add :

Este comando se utiliza para añadir archivos

```
git add nombrearchivo.txt
```

commit :

Este comando hace una instantánea de los cambios preparados en este momento en el proyecto. Las instantáneas confirmadas se pueden considerar versiones seguras del proyecto.

```
git commit -m "Mensaje que acompaña el commit"
```

status:

Este comando lista los archivos que se han preparado, los que no y los archivos de los que Git no realizará ningún seguimiento.

```
git status
```

push :

Comando de los más básicos. Envía los cambios hechos de un repositorio local a uno de remoto.

```
git push origin master
```

remote :

Este comando se utiliza para conectar a un repositorio remoto.

```
git remote
```

pull :

Este comando se utiliza para fusionar todos los cambios que se han hecho al repositorio local.

```
git pull
```

merge :

Este comando se utiliza para fusionar una rama con otra rama activa.

```
git merge <nombre-rama>
```

fetch :

Este comando permite buscar todos los objetos de un repositorio remoto que no están actualmente en el repositorio local.

```
git fetch origin
```

Para las personas que están aprendiendo a crear código, Github tiene una funcionalidad muy útil llamada **Github Pages**, que permite crear y publicar todo el código de una web pública, alojando este código gratuitamente dentro de su dominio, con una estructura de este estilo:

`tuusuario.github.io`.

En este manual vamos a ver cómo, en pocos pasos y de una manera muy simple, podemos crear y publicar un repositorio en Github.

1) Crear una cuenta en Github

Lo primero que tendrás que hacer es registrarte en Github.

Es muy importante que, cuando elijas tu nombre de usuario, tengas en mente que este será el nombre que va a aparecer en tu dominio. Asegúrate de elegir un nombre disponible e introduce tu correo electrónico y una contraseña.

Entonces tendrás que crear un plan gratuito, clicando al botón que pone “*Join a free plan*”. Solamente tendrás que verificar que no eres una máquina, eligiendo una imagen, y ya podrás seguir con la configuración de tu cuenta.

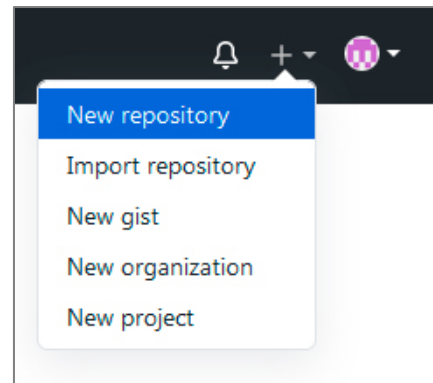
A continuación, tendrás que responder a algunas preguntas sobre lo que necesitas, qué conocimientos tienes y cuáles son tus lenguajes de preferencia.

Luego recibirás un correo en tu buzón, con el fin de verificar que tu dirección de correo electrónico es correcta.

2) Crear un nuevo repositorio en Github

Antes de empezar a introducir código, lo primero que tienes que hacer es crear un repositorio de código, que es donde irá agrupado todo tu código para este nuevo proyecto web. En el margen superior derecho, tienes el enlace para añadir un nuevo repositorio.

A continuación, eliges un nombre para este repositorio, lo marcas como público y también eliges que contenga un fichero *readme.md*, que luego podrás modificar.




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 *

 laboratori-uoc ▾


/ web ✓

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

Description (optional)

☒  Public

Anyone on the internet can see this repository. You choose who can commit.

☐  Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ **Add a README file**


This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

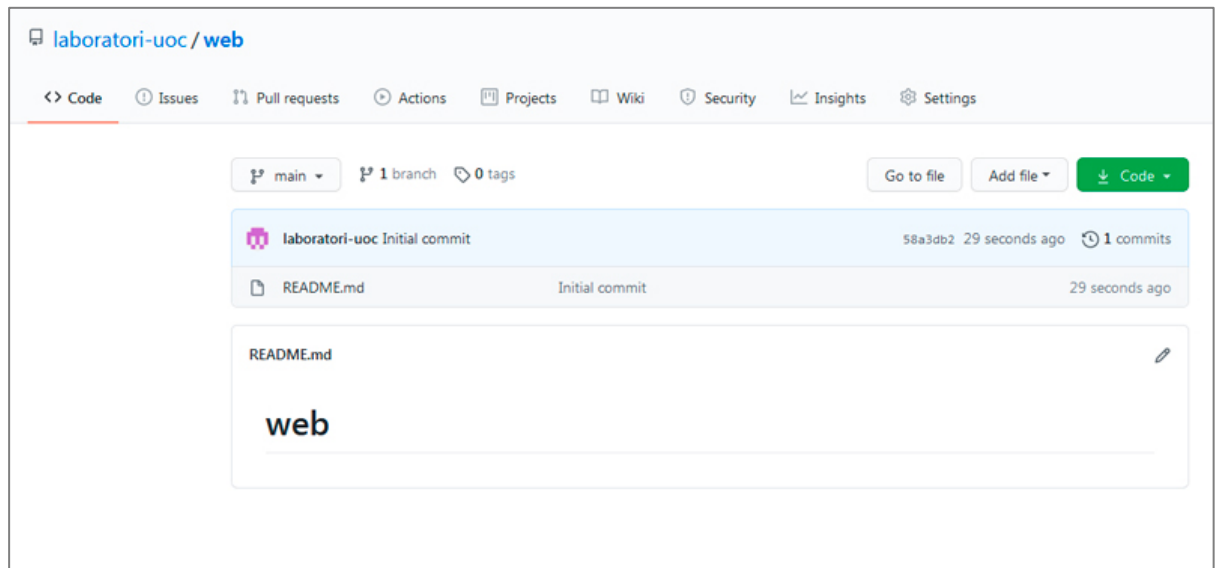
☐ **Choose a license**

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

This will set  **main** as the default branch. Change the default name in your [settings](#).

Create repository

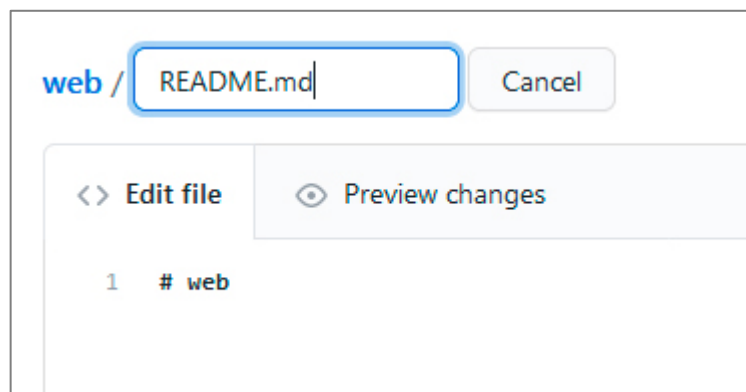
Ya tienes el proyecto que hemos llamado “web” - puedes ponerle el nombre que quieras - y en el cuál, ahora mismo, solo hay el archivo **README.md**. En el lateral derecho de este archivo, verás que hay un icono con un lápiz.



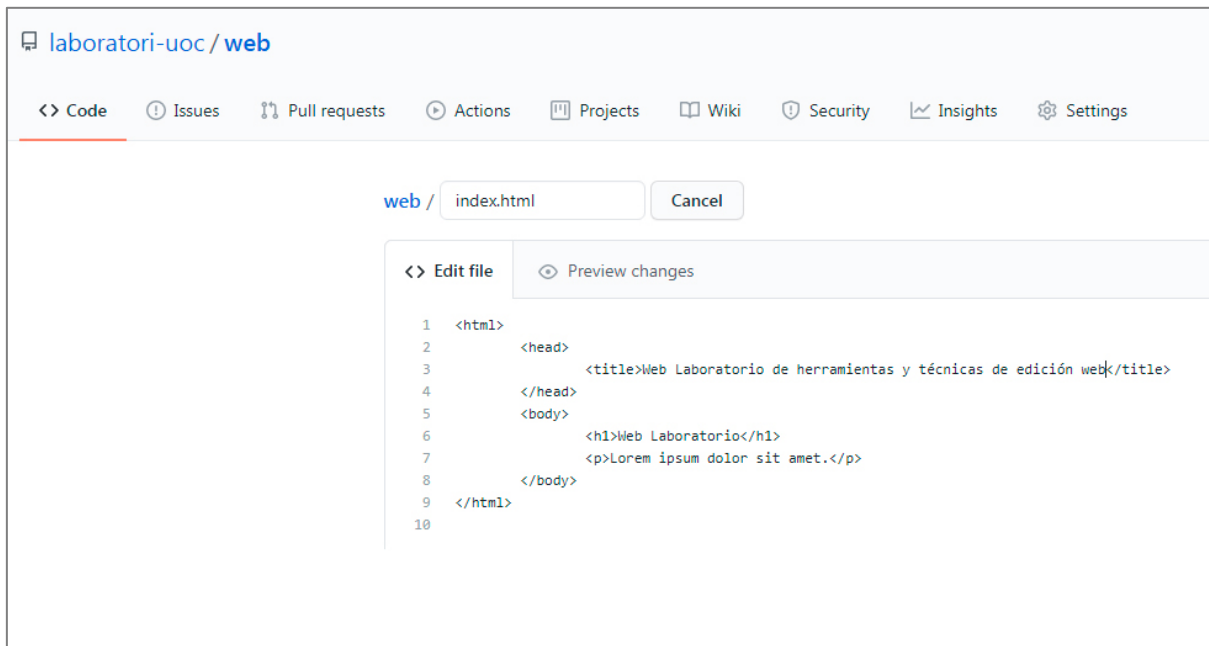
Editar este archivo será la manera más fácil de tener ya hecha la primera página de tu proyecto.

Una vez creado este repositorio, lo podrás descargar y modificar. En realidad, este será el modo más cómodo de trabajar cuando ya lo tengas más controlado. Pero, de momento y, para esta ocasión, lo editaremos directamente en la web de *Github Pages*.

En la casilla donde pone el nombre del archivo, cambia “README.md” por “index.html”.



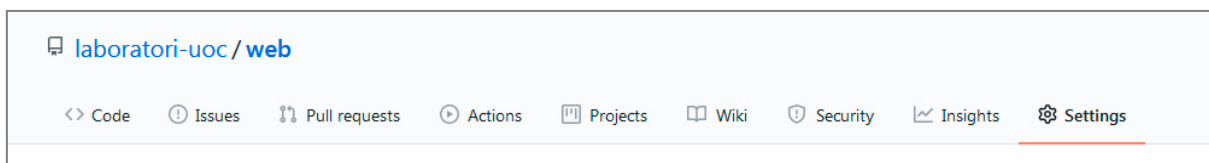
Entonces, ya puedes editar el texto que hay dentro de este archivo y puedes poner el código HTML que corresponda a la que va a ser tu primera página.



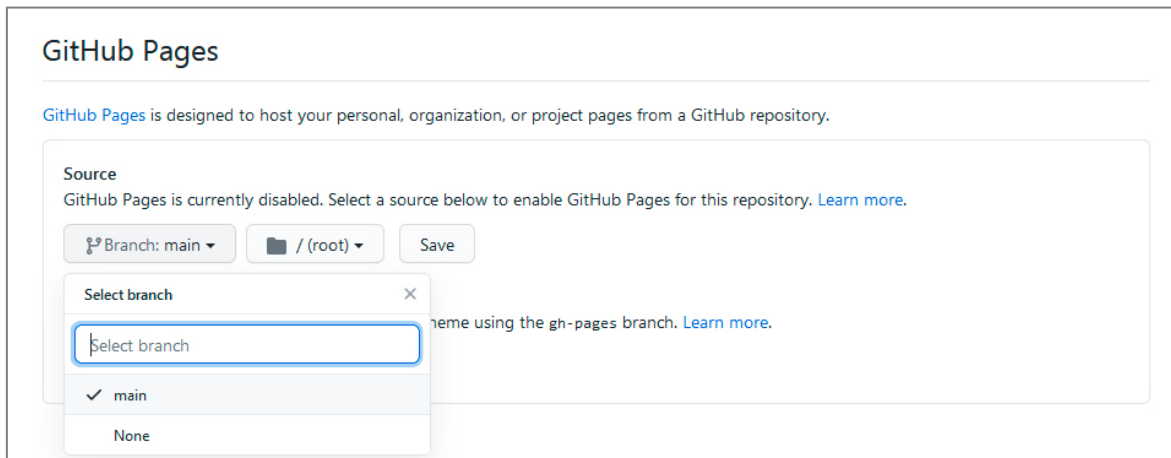
Finalmente, con el *scroll*, desplaza la página hacia abajo hasta llegar donde pone **“Commit changes”**. El cambio de `README.md` a `index.html` junto con el cambio de contenido de la página quedarán registrados como cambios realizados en el repositorio y siempre se podrán consultar en el historial de cambios.

3) Ver tu web publicada

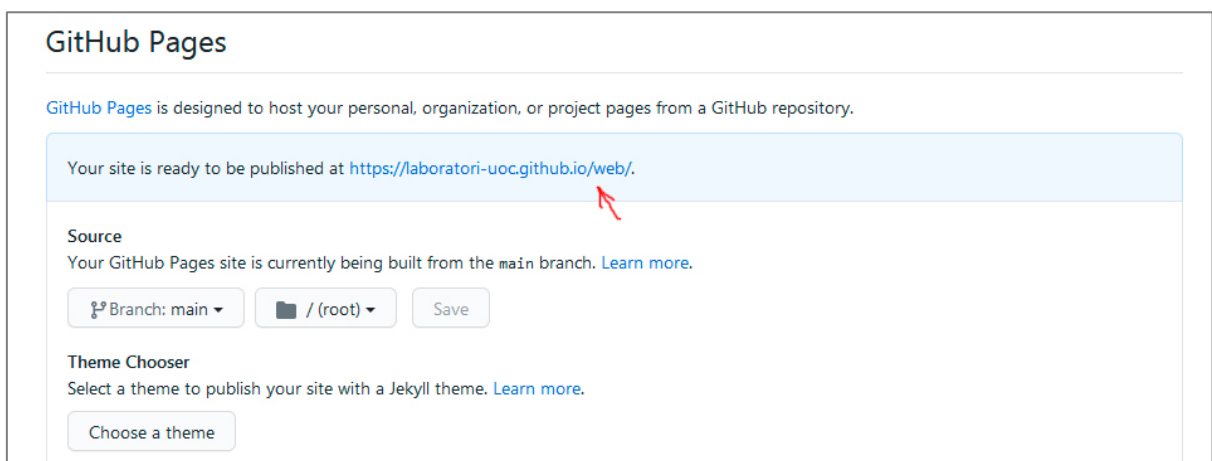
Si accedes a tu web publicada en Github y aún no la puedes ver, tendrás que acceder a la pestaña de “*settings*” e ir al apartado de **Github Pages**.



Elige el *branch* “**main**” (ya que aún no hemos creado ninguna otra) y haz clic en el botón que pone “**save**” para guardar los cambios.



A continuación ya aparecerá publicada, justo encima de este apartado, **la URL** donde se ha publicado tu proyecto web.



Si clicas en la URL, se te abrirá esta web desde tu navegador. Sino también puedes ir directamente a un navegador y teclear la dirección web del proyecto que acabas de crear.

La primera página que hemos creado, **index.html**, a la cual hemos introducido muy poco código, ya se vería de este modo:



A partir de aquí, ya puedes ir editando esta página para ponerle más contenidos y estilos CSS e imágenes relacionadas. Puedes ir subiendo archivos y trabajar con el proyecto, hasta tener una web del todo navegable.

Más información y ayuda para crear páginas con *Github Pages*

- ***Publicar tu web usando Github Pages***
<https://devcode.la/tutoriales/publicar-tu-web-usando-github-pages/>
- ***Creating a Github Pages site***
<https://docs.github.com/en/free-pro-team@latest/github/working-with-github-pages/creating-a-github-pages-site>
- ***Hosting your website on Github***
<https://www.khanacademy.org/computing/computer-programming/html-css/web-development-tools/a/hosting-your-website-on-github>
- ***Hello world***
<https://guides.github.com/activities/hello-world/>
- ***Getting started with Github Pages***
<https://guides.github.com/features/pages/>
- ***Using github pages***
https://developer.mozilla.org/es/docs/Learn/Using_Github_pages
- ***Git Handbook***
<https://guides.github.com/introduction/git-handbook/>