

# ¿Cómo publicar en *GithubPages*?

Como muchos de vosotros ya sabréis, [Github](#) es un servicio que permite alojar repositorios de código además de hacer un control de versiones con una interfaz gráfica basada en web y escritorio y también con integración móvil. Gran parte de los repositorios que se publican son públicos -accesibles y abiertos- para todos. Esto es una excelente noticia para todos aquellos que crean código: todos podemos aprender de todos. De hecho, esta es la gracia del asunto. Pero antes de empezar, ¿sabéis qué es Git?

## Antes de empezar: aclaremos conceptos

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

Git permite a los desarrolladores ver toda la cronología de sus cambios, decisiones y progresión de cualquier proyecto en un solo sitio. Con una herramienta como ésta todos los desarrolladores implicados en un proyecto pueden trabajar de manera asíncrona e incluso en diferentes zonas horarias.

Un programa de control de versiones básicamente nos permite conocer en detalle cosas tales como: ¿qué cambios se hicieron? ¿Quién los hizo? ¿Cuándo?, o ¿por qué eran necesarios estos cambios?

**Un repositorio**, o proyecto Git, incluye todos los ficheros y carpetas asociados a un proyecto y también el historial de revisiones de cada fichero. Trabajar en repositorios mantiene los proyectos en desarrollo organizados y protegidos. A través de plataformas como **GitHub**, Git también proporciona mejores oportunidades para la transparencia y la colaboración en el proyecto.

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

Para trabajar en Git es necesario familiarizarse con conceptos tales como **branch**, **commit**, **pull**, **push**... A continuación, encontraréis un pequeño glosario de términos que os servirán para familiarizaros con los principales comandos de Git, aunque en las redes encontraréis gran cantidad de tutoriales y manuales para utilizarlos con todo su potencial. Para empezar a introducirnos en el tema os recomendamos esta guía, [Git Handbook](#), en la que también encontraréis todos los **comandos** necesarios para trabajar con Git i Github.

## Principales comandos Git

### **branch**

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

```
gitbranch  
gitbranch -d <branch-name>
```

### **init**

Este comando se utiliza para crear un repositorio Git.

```
gitinit
```

### **add**

Este comando se utiliza para añadir archivos.

```
gitadd nomarxiu.txt
```

### **commit**

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

```
gitcommit -m "Missatge que acompanya el commit"
```

### **status**

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

```
git status
```

### **push**

Este comando uno de los más básicos. Envía los cambios que se han hecho de un repositorio local a uno remoto.

```
gitpushorigin master
```

### **remote**

Este comando se utiliza para conectar con un repositorio remoto.

```
gitremote
```

### **pull**

Este comando se utiliza para fusionar todos los cambios realizados en el repositorio local.

```
gitpull
```

### **merge**

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

```
gitmerge<nom-branca>
```

### **fetch**

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

```
gitfetchorigin
```

Para aquellos que estáis aprendiendo a crear código, Github tiene una funcionalidad muy útil denominada **Github Pages**, la cual permite crear y publicar todo el código de una web pública alojándola gratuitamente dentro de su dominio con una estructura de este estilo:

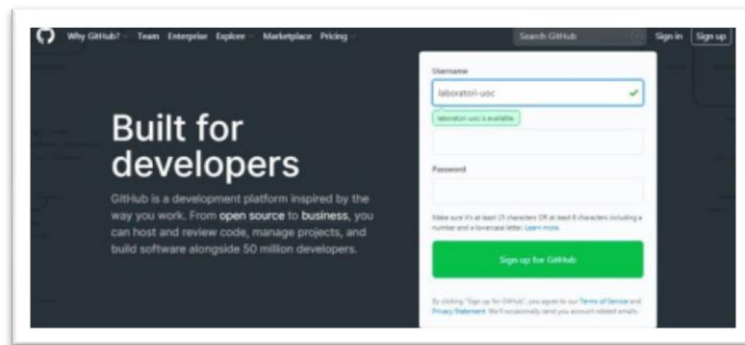
`elteusuari.github.io`.

En este manual veremos cómo en pocos pasos y de forma muy sencilla podemos publicar un repositorio en Github.

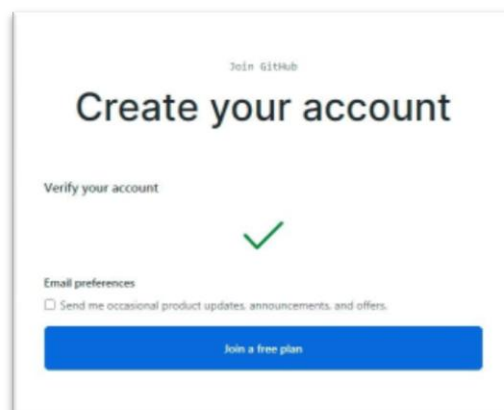
## 1) Crear una cuenta enGithub

Lo primero que tendréis que hacer es registraros en Github.

Es muy importante que cuando escojáis vuestro nombre de usuario, tengáis en cuenta que éste será el nombre que aparecerá en vuestro dominio. Aseguraos de escoger un nombre disponible e introducid vuestro correo electrónico y una contraseña.



Después tendréis que crear un plan gratuito clicando en el botón que dice "Join a free plan". Sólo quedará verificar que no sois una máquina, escogiendo una imagen, y ya podréis seguir con la configuración de vuestra cuenta.

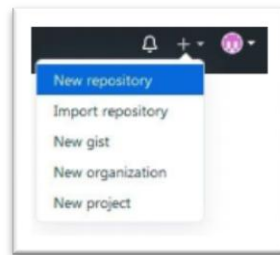


A continuación, tendréis que responder a algunas preguntas sobre lo que necesitáis, qué conocimientos tenéis y cuales son vuestras preferencias en lo que a lenguajes se refiere.

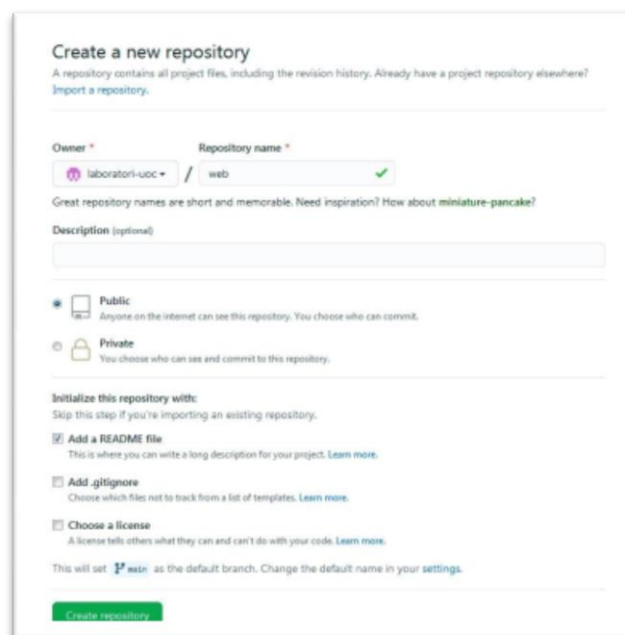
A continuación, recibiréis un correo en vuestro buzón para verificar que vuestra dirección electrónica es correcta.

## 2) Crear un nuevo repositorio enGithub

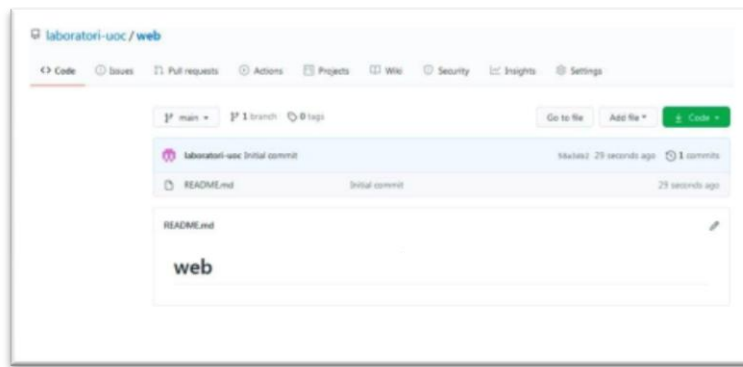
Antes de empezar a poner código, lo primero que tendréis que hacer es crear un repositorio de código en el que irá agrupado todo vuestro código para este nuevo proyecto web. En el margen superior derecho tenéis el enlace para añadir un nuevo repositorio.



A continuación, escoged un nombre para este repositorio, marcadlo como público y, también escoged que haya un fichero reame.md, qué después podréis modificar.

A screenshot of the 'Create a new repository' page on GitHub. The form includes fields for 'Owner' (laboratori-uoc) and 'Repository name' (web). Below these are instructions about repository names and a 'Description (optional)' field. There are radio buttons for 'Public' (selected) and 'Private'. Under 'Initialize this repository with:', there are three checked checkboxes: 'Add a README file', 'Add .gitignore', and 'Choose a license'. A green 'Create repository' button is at the bottom.

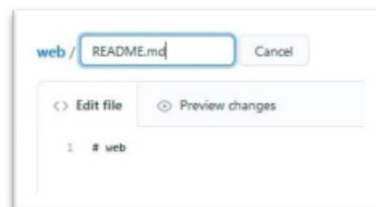
Ya tendréis creado el proyecto al que hemos nombrado “web” –podéis ponerle el nombre que queráis- y en el que ahora mismo sólo hay un archivo README.md. En el lateral derecho de este archivo observaréis que hay un ícono con un lápiz.



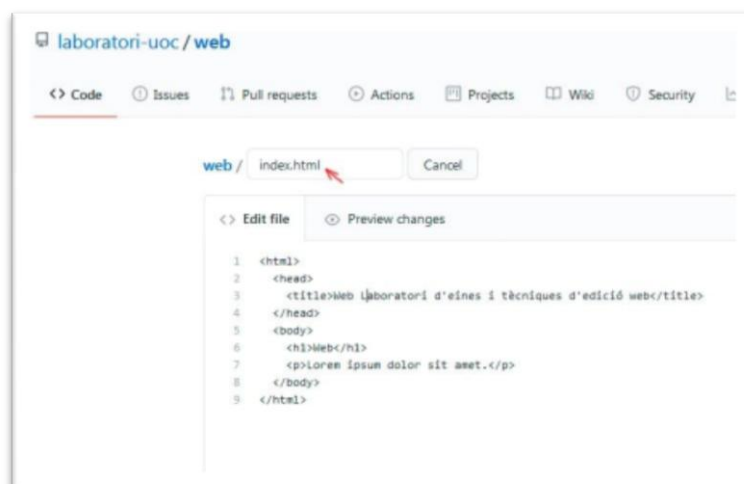
Editar este archivo será la manera más fácil de tener ya hecha la primera página de vuestro proyecto.

Una vez creado este repositorio lo podréis descargar y modificar. De hecho, ésta será la manera más cómoda de trabajar cuando ya estéis familiarizados. Pero, de momento y para esta ocasión, lo editaremos directamente en el web de *Github Pages*.

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



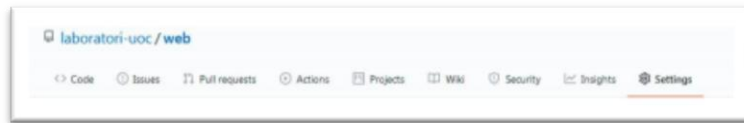
Entonces ya podréis editar el texto que haya y podréis poner el código HTML correspondiente a la que será vuestra primera página



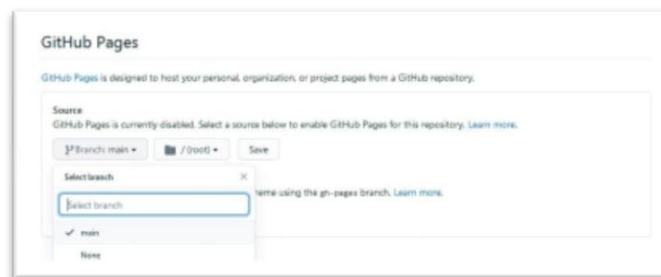
Finalmente, con el scroll, moved la página hacia abajo hasta llegar a “**Commit changes**”. El cambio de `README.md` a `index.html` y el cambio de contenido de la página quedarán registrados como cambio realizado en el repositorio y siempre podrá ser consultado en el historial de cambios.

### 3) Ver vuestra web publicada

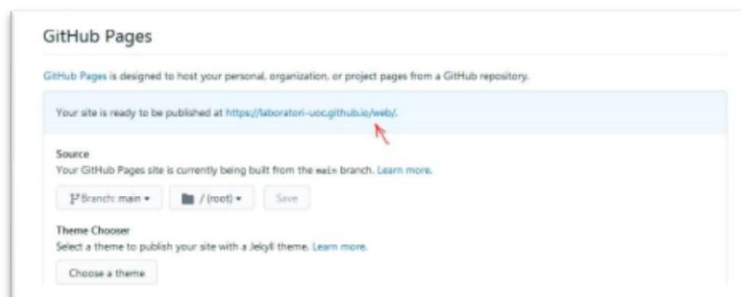
Si accedéis a vuestra web, publicada en Githuby todavía no lavéis, deberéis acceder a la pestaña de “settings” e ir al apartado de **Github Pages**.



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



A continuación, ya aparecerá publicada, justo encima de este apartado, la URL donde ha sido publicado vuestro proyecto web.



Si clicáis sobre esta URL el web se abrirá desde vuestro navegador. Otra opción es ir directamente a un navegador y teclear vuestra dirección web del proyecto que acabáis de crear.

La primera página que hemos creado, index.html, a la qué hemos puesto muy poco código, ya se vería así:



A partir de aquí ya podréis ir editando esta página para añadirle contenidos y estilos CSS e imágenes relacionadas, podréis ir subiendo archivos y trabajar en el proyecto hasta tener una web completamente navegable.

## Más información y ayuda para crear páginas con *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>
- Publicar tu web usando Github Pages  
<https://devcode.la/tutoriales/publicar-tu-web-usando-github-pages>
- Helloworld  
<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](https://developer.mozilla.org/es/docs/Learn/Using_Github_pages)
- Git Handbook  
<https://guides.github.com/introduction/git-handbook/>