

Práctica 1 - Documentación

Cómo publicar en GitHub Pages

1	Presentación	2
2	Git	2
3	Principales comandos Git	2
4	GitHub Pages	3
4.1	Crear una cuenta en GitHub	3
4.2	Crear un nuevo repositorio en GitHub	3
4.3	Ver nuestra web publicada	5

1 Presentación

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.

2 Git

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 y 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 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, encontraremos un pequeño glosario de términos que servirán para familiarizarnos con los principales comandos de Git, aunque en las redes se encuentra gran cantidad de tutoriales y manuales para utilizarlos con todo su potencial.

3 Principales comandos Git

Algunos de los comandos más importantes de Git son los siguientes:

Comando	Descripción	Ejemplos
branch	Este comando se utiliza para listar, crear o eliminar ramas.	<code>git branch</code> <code>git branch -d <branch-name></code>
init	Este comando se utiliza para crear un repositorio Git.	<code>git init</code>
add	Este comando se utiliza para añadir archivos.	<code>git add file.txt</code>
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.	<code>git commit -m "Mensaje que acompaña a commit"</code>
status	Este comando lista los archivos que se han preparado, los que no y los archivos de los que Git no hará ningún seguimiento.	<code>git status</code>
push	Este comando uno de los más básicos. Envía los cambios que se han hecho de un repositorio local a uno remoto.	<code>git push origin master</code>
remote	Este comando se utiliza para conectar con un repositorio remoto.	<code>git remote</code>

pull	Este comando se utiliza para fusionar todos los cambios realizados en el repositorio local.	<code>git pull</code>
merge	Este comando se utiliza para fusionar una rama con otra rama activa.	<code>git merge <branch-name></code>
fetch	Este comando permite buscar todos los objetos de un repositorio remoto que actualmente no están en el repositorio local.	<code>git fetch origin</code>

4 GitHub Pages

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 estilo:

`usuario.github.io`

Ahora veremos cómo en pocos pasos y de forma muy sencilla podemos publicar un repositorio en GitHub.

4.1 Crear una cuenta en GitHub

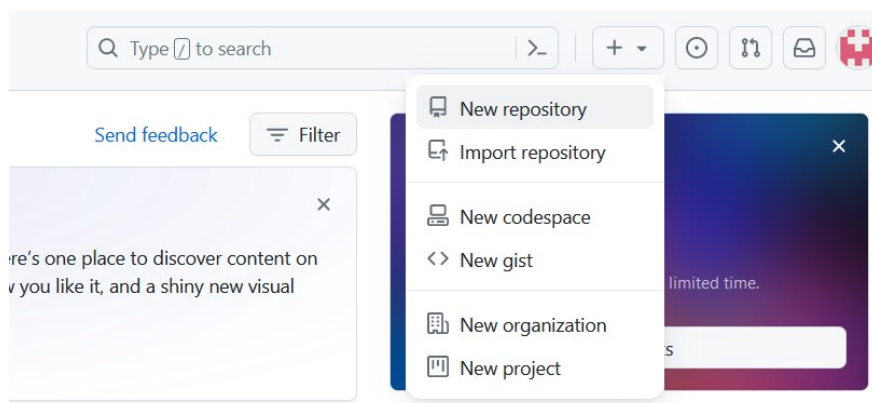
Lo primero que tenemos que hacer es registrarnos en GitHub. Es muy importante que cuando escojamos nuestro nombre de usuario, tengamos en cuenta que éste será el nombre que aparecerá en nuestro dominio. Escogeremos un nombre disponible e introduciremos nuestro correo electrónico y una contraseña.

Después, crearemos un **plan gratuito** haciendo clic en el botón «*Join a free plan*». Sólo quedará verificar que no somos una máquina escogiendo una imagen, y ya podremos seguir con la configuración de la cuenta. Tendremos que responder a algunas preguntas sobre lo que necesitamos, qué conocimientos tenemos y cuáles son nuestras preferencias en lo que a lenguajes se refiere.

Finalmente, recibiremos un correo en nuestro buzón para verificar que la dirección electrónica es correcta.

4.2 Crear un nuevo repositorio en GitHub

En primer lugar, tendremos que crear un repositorio de código en el que irá agrupado todo nuestro código para este nuevo proyecto web. En el margen superior derecho tenemos el enlace para añadir un nuevo repositorio.




A continuación, escogeremos un nombre para este repositorio, lo marcamos como público y, también, escogeremos que haya un fichero «*README.md*», qué después podremos modificar.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)



Required fields are marked with an asterisk (*).

Owner * Repository name *

 maravid /

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

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:

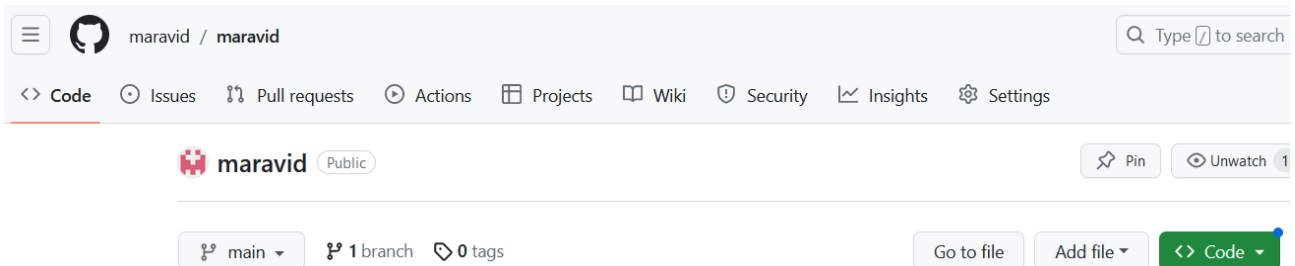
- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

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

Ya tenemos creado el proyecto al que hemos nombrado «web» – podemos ponerle el nombre que queramos, por ejemplo, en mi caso, he utilizado la primera letra de mi nombre y mi primer apellido – y en el que ahora mismo sólo hay un archivo «*README.md*». En el lateral derecho de este archivo, observamos que hay un **icono con un lápiz**. Editar este archivo será la manera más fácil de tener ya hecha la primera página de nuestro proyecto.



Una vez creado este repositorio lo podremos descargar y modificar. De hecho, ésta será la manera más cómoda de trabajar cuando ya estemos familiarizados. Pero, de momento y para esta ocasión, lo editaremos directamente en la página web de GitHub Pages.

En la casilla donde pone el nombre del archivo cambiad «*README.md*» por «*index.html*». Entonces ya podemos editar el texto que haya y escribir el código HTML correspondiente a la que será nuestra primera página.

Finalmente, nos movemos 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.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://maravid.github.io/maravid/>
Last deployed by maravid 3 months ago

[Visit site](#) [...](#)

Build and deployment

Source
Deploy from a branch ▼

Branch
Your GitHub Pages site is currently being built from the `main` branch. [Learn more about configuring the publishing source for your site.](#)

main ▼ / (root) ▼ [Save](#)

[Learn how to add a Jekyll theme](#) to your site.

4.3 Ver nuestra web publicada

Si accedemos a nuestra web, publicada en Github y todavía no la vemos, accedemos a la pestaña de «Settings» y vamos al apartado de Github Pages.

Escoged el branch «main» (ya que no hemos creado ninguna otra) y hacemos clic en el botón «Save» para guardar los cambios. A continuación, ya aparecerá publicada, justo encima de este apartado, la URL donde ha sido publicado nuestro proyecto web.

Si hacemos clic sobre esta URL el web se abrirá desde nuestro navegador. Otra opción es ir directamente a un navegador y teclear la dirección web del proyecto que acabamos de crear. A partir de aquí ya podemos ir editando esta página para añadir contenido, estilos e imágenes, así como subiendo diferentes archivos hasta tener una web completamente navegable.