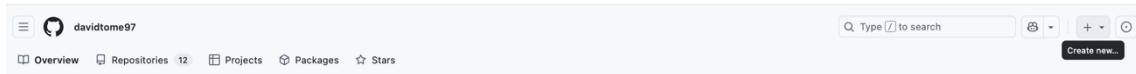
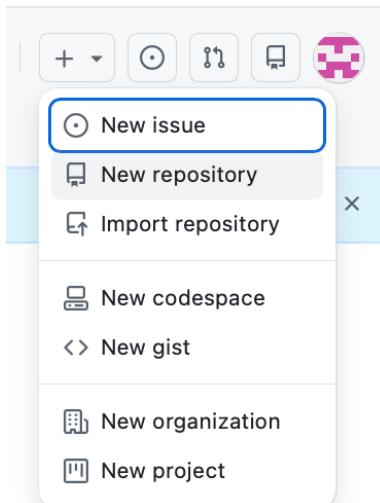


Para crear nuestro repositorio, nos logueamos en github y damos al + que aparece arriba a la derecha.



Pulsamos a “New repository”



Le ponemos un nombre a nuestro repositorio (este nombre lo vamos a necesitar en la configuración de la aplicación), recomendable no poner el mismo nombre para todo. Una vez puesto le damos a “create repository”.

**Create a new repository**  
Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).  
Required fields are marked with an asterisk (\*).

**1 General**

Owner \*  Repository name \*   
Great repository names are short and memorable. How about [curly-adventure](#)?

Description  
  
0 / 350 characters

**2 Configuration**

Choose visibility \*  Choose who can see and commit to this repository

Start with a template  Templates pre-configure your repository with files.

Add README  READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore  .gitignore tells git which files not to track. [About ignoring files](#)

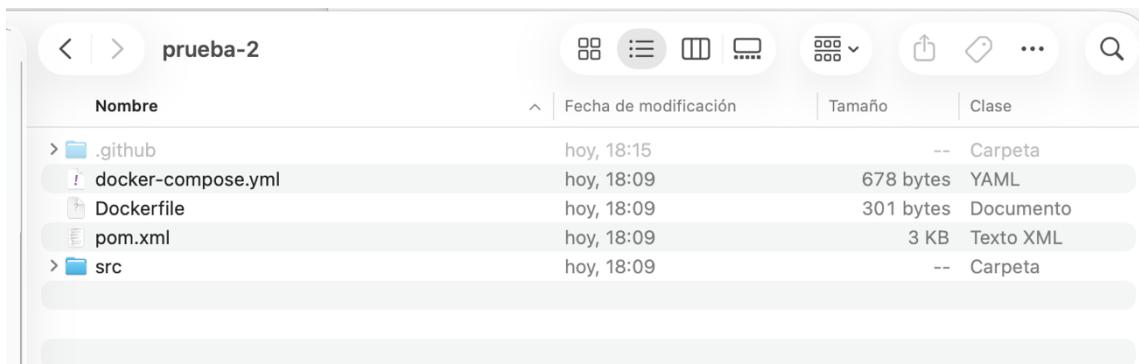
Add license  Licenses explain how others can use your code. [About licenses](#)

**Create repository**

Damos a descargar nuestro zip

Listado de aplicaciones							
Nueva aplicación							
Nombre	Lenguaje	Base de datos	Proveedor CI/CD	Fecha creación	Pasos despliegue	Estado	Acciones
demo-config-zip	JAVA	MYSQL	GITHUB	07/12/2025 19:26	0 / 6	--	<button>Editar</button> <button>Eliminar</button> <button>Descargar ZIP</button> <button>Asistente</button>
prueba	JAVA	MONGODB	GITHUB	11/12/2025 21:58	0 / 6	--	<button>Editar</button> <button>Eliminar</button> <button>Descargar ZIP</button> <button>Asistente</button>

Dentro de nuestro zip nos debería aparecer estos ficheros o similares, dependiendo de la configuración que hemos elegido.



Abrimos nuestro terminal y nos localizamos dentro de nuestro zip ya descomprimido que hemos descargado. Ahora ejecutamos los siguientes comandos:

**git init**

**git add .**

**git commit -m "chore: proyecto inicial generado"**

**git branch -M main**

**git remote add origin https://github.com/USUARIO/REPO.git**

**git push -u origin main**

no obstante, una vez que creamos nuestro repositorio en github, nos da la ayuda para hacer el primer commit en la ventana principal

The screenshot shows the initial setup steps for a new GitHub repository named 'prueba1'. It includes sections for setting up GitHub Copilot, adding collaborators, quick setup via desktop or command line, and pushing from the command line.

**Set up GitHub Copilot**  
Use GitHub's AI pair programmer to autocomplete suggestions as you code.  
[Get started with GitHub Copilot](#)

**Add collaborators to this repository**  
Search for people using their GitHub username or email address.  
[Invite collaborators](#)

**Quick setup — if you've done this kind of thing before**

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/davidtome97/prueba1.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# prueba1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/davidtome97/prueba1.git
git push -u origin main
```

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/davidtome97/prueba1.git
git branch -M main
git push -u origin main
```

Si todo sale bien debería salir algo así:

```
b.com/davidtome97/prueba1.git
[(base) david@MacBook-Pro-de-David prueba-2 % git push -u origin main
Enumerating objects: 50, done.
Counting objects: 100% (50/50), done.
Delta compression using up to 10 threads
Compressing objects: 100% (45/45), done.
Writing objects: 100% (50/50), 16.75 KiB | 8.38 MiB/s, done.
Total 50 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/davidtome97/prueba1.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
(base) david@MacBook-Pro-de-David prueba-2 % ]
```

The screenshot shows the GitHub repository page for 'prueba1' after the initial commit. It displays the commit history, file structure, and a README file.

**Code** Issues Pull requests Actions Projects Wiki Security Insights Settings

**Don't get locked out of your account. [Download your recovery codes](#) or [add a passkey](#) so you don't lose access when you get a new device.**

**prueba1 Public**

**Code** Issues Pull requests Actions Projects Wiki Security Insights Settings

**main 1 Branch 0 Tags**

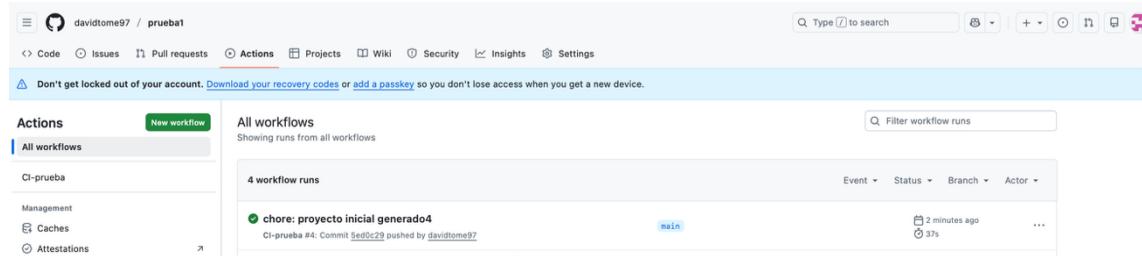
**Go to file** Add file Code

**davidtome97 chore: proyecto inicial generado** 65815ed · 10 minutes ago 1 Commit

File	Commit Message	Time
.github	chore: proyecto inicial generado	10 minutes ago
src/main	chore: proyecto inicial generado	10 minutes ago
.DS_Store	chore: proyecto inicial generado	10 minutes ago
Dockerfile	chore: proyecto inicial generado	10 minutes ago
docker-compose.yml	chore: proyecto inicial generado	10 minutes ago
pom.xml	chore: proyecto inicial generado	10 minutes ago

**README**

Despues de hacer el primer commit si nos vamos a la pestaña “actions”, nos debería aparecer nuestro workflows en verde.



The screenshot shows the GitHub Actions interface for a repository named "prueba1". The left sidebar has tabs for "Actions", "Management", "Caches", and "Attestations", with "Actions" currently selected. A "New workflow" button is also visible. The main area displays "All workflows" with a subtitle "Showing runs from all workflows". Below this, a table titled "4 workflow runs" lists four entries. The first entry is highlighted in green and labeled "chore: proyecto inicial generado4", indicating it was triggered by a commit and is successful. It includes details like "main", "2 minutes ago", and "37s".

Workflow	Status	Event	Duration
chore: proyecto inicial generado4	Success	main	2 minutes ago
CI-prueba #4: Commit 5ed0c28 pushed by davidtome97	Success	main	37s
CI-prueba #3: Commit 5ed0c28 pushed by davidtome97	Success	main	37s
CI-prueba #2: Commit 5ed0c28 pushed by davidtome97	Success	main	37s