

Control de versiones con Git

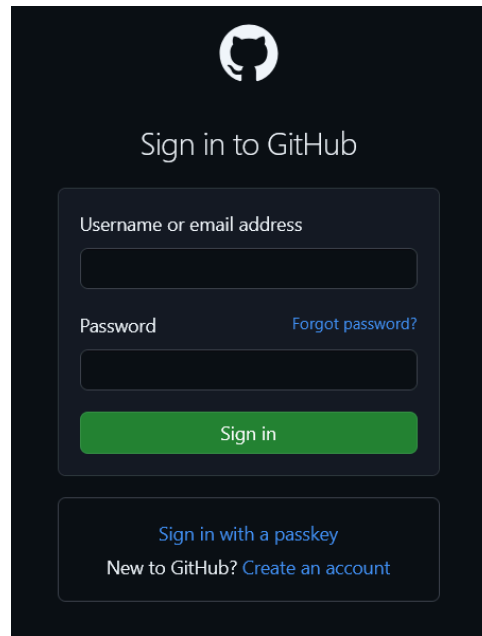


Índice

Registrarse en GitHub	1
Como crear un repositorio	1
Instalación de Git	3
Administración del repositorio con Git	3

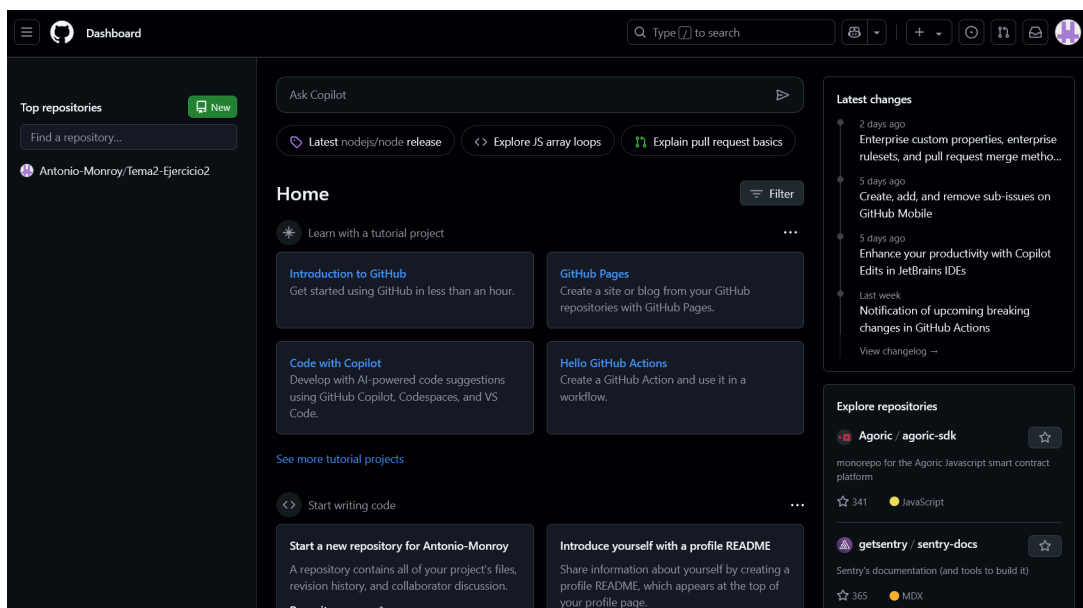
Registrarse en GitHub

Lo primero que tendremos que hacer para crear un repositorio en GitHub es tener una cuenta creada, en el caso de que ya la tengamos, tendremos únicamente que iniciar sesión.

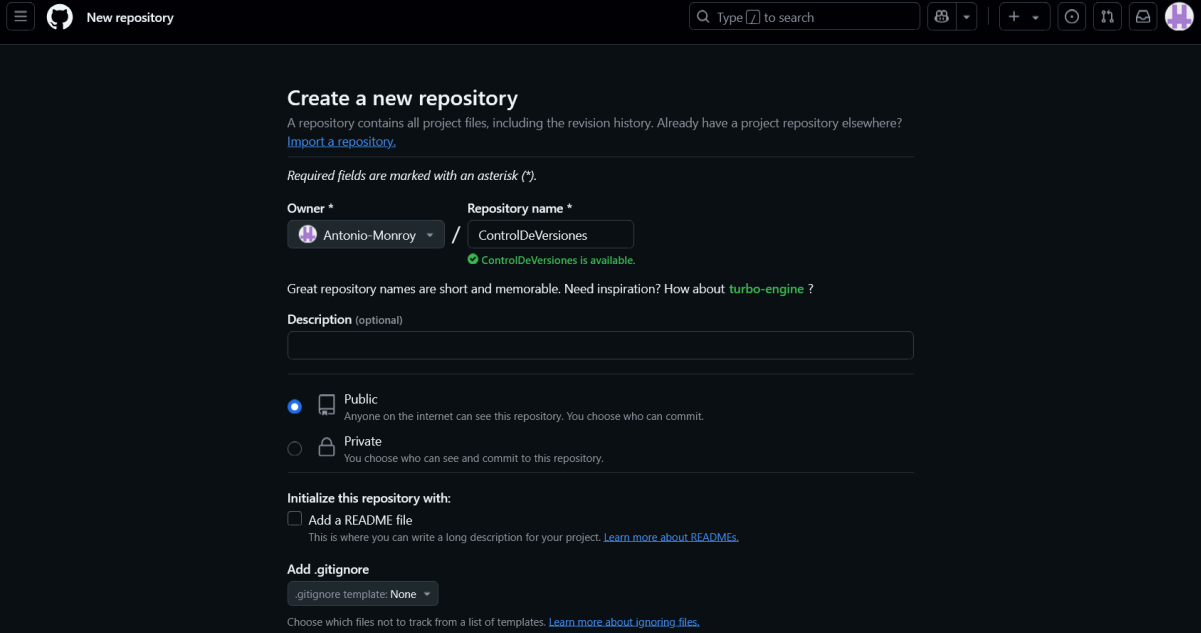


Como crear un repositorio

Una vez tengamos nuestra cuenta ya, estaremos en el menú principal de GitHub, lo que tendremos que hacer es darle al botón de "New" que encontramos arriba a la izquierda, este botón es para poder crear un repositorio nuevo.

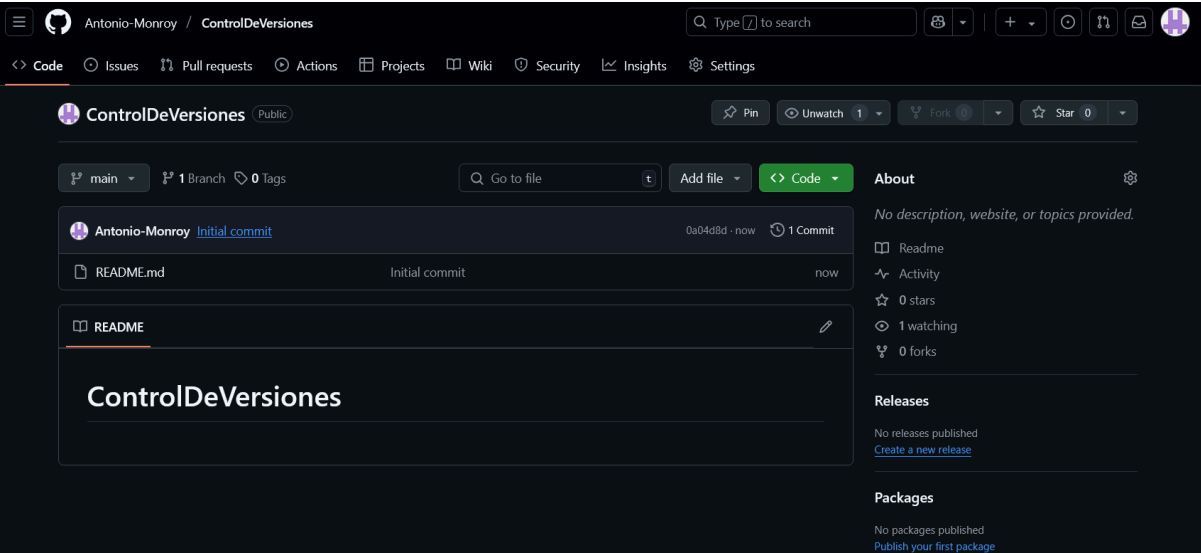


Una vez le hayamos dado al botón de “New”, se nos abrirá un menú donde tendremos que rellenar los datos que nos piden para la creación del repositorio, así como su nombre o su estado (público o privado). Una vez terminada la configuración le daremos al botón que dice “Create Repository”.



The screenshot shows the 'Create a new repository' page on GitHub. At the top, there's a search bar and navigation icons. The main heading is 'Create a new repository', followed by a subtext: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. Below this, a note states 'Required fields are marked with an asterisk (*)'. The form has two main sections: 'Owner' and 'Repository name'. The 'Owner' is set to 'Antonio-Monroy' with a dropdown arrow. The 'Repository name' is 'ControlDeVersiones', with a green checkmark indicating it's available. A hint says 'Great repository names are short and memorable. Need inspiration? How about [turbo-engine](#) ?'. There's a 'Description (optional)' text area. Below that, there are radio buttons for 'Public' (selected) and 'Private'. The 'Public' option has a subtext: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option has a subtext: 'You choose who can see and commit to this repository.' Under 'Initialize this repository with:', there's a checkbox for 'Add a README file' with a subtext: 'This is where you can write a long description for your project. [Learn more about READMEs.](#)'. Below that, there's a section for 'Add .gitignore' with a dropdown menu set to 'None' and a subtext: 'Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)'.

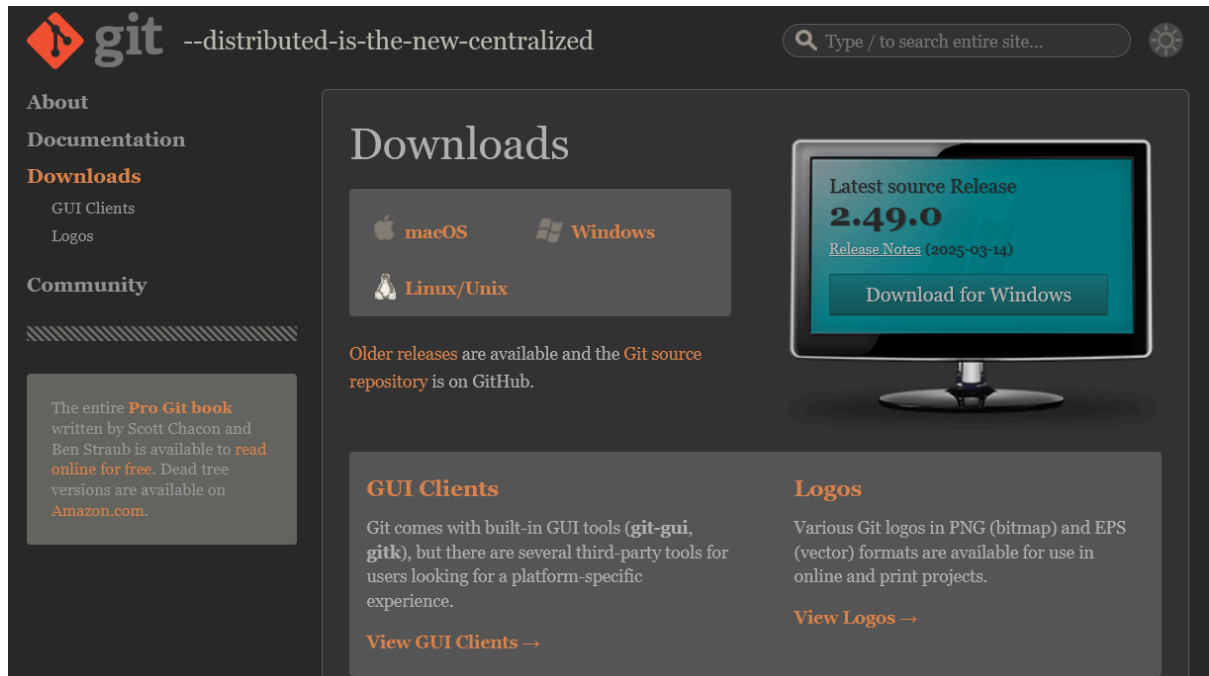
Una vez creado el repositorio, nos llevará al menú de este repositorio para poder trabajar con él, se puede trabajar con él desde GitHub pero en nuestro caso, lo vamos a administrar desde el cmd.



The screenshot shows the GitHub repository page for 'ControlDeVersiones' by 'Antonio-Monroy'. The repository is public. At the top, there's a navigation bar with icons for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there's a header section with the repository name 'ControlDeVersiones' and its status 'Public'. There are buttons for 'Pin', 'Unwatch' (1), 'Fork' (0), and 'Star' (0). Below the header, there's a section for the 'main' branch, showing '1 Branch' and '0 tags'. There's a search bar 'Go to file' and buttons for 'Add file' and 'Code'. The main content area shows the 'Initial commit' by 'Antonio-Monroy' with the commit hash '0a04d8d' and the time 'now'. Below the commit, there's a 'README.md' file. The right sidebar contains an 'About' section with the text 'No description, website, or topics provided.' and links for 'Readme', 'Activity', 'Stars' (0), 'Watching' (1), and 'Forks' (0). There's also a 'Releases' section with the text 'No releases published' and a link 'Create a new release'. At the bottom, there's a 'Packages' section with the text 'No packages published' and a link 'Publish your first package'.

Instalación de Git

Para poder administrar nuestro repositorio desde el cmd, tendremos que instalar “Git”. Una vez descarguemos el instalador, le tendremos que dar a siguiente todo el rato hasta que empiece la instalación.



Administración del repositorio con Git

Primero tendremos que configurar nuestro usuario, dándole el nombre y correo electrónico que tenemos en nuestra cuenta de GitHub. Esto lo haremos con el comando “Git config”.

```
PS C:\Users\anton> git config --global user.name "Antonio-Monroy"
PS C:\Users\anton> git config --global user.email "a.lopez.pina@iescristobaldemonroy.es"
```

Una vez configurado el usuario, tendremos que seleccionar un directorio donde queremos guardar nuestro repositorio. Una vez que estemos dentro del directorio, tendremos que hacer una copia de nuestro repositorio con el comando “Git clone” y el enlace de nuestro repositorio.

```
PS C:\Users\anton> cd C:\Users\anton\Documents\ControlDeVersiones\
PS C:\Users\anton\Documents\ControlDeVersiones> git clone https://github.com/Antonio-Monroy/ControlDeVersiones.git
Cloning into 'ControlDeVersiones'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

Para poder trabajar con el repositorio, tendremos que utilizar el comando “Git init”, este comando se utiliza para que Git reconozca que está trabajando con un repositorio.

```
PS C:\Users\anton\Documents\ControlDeVersiones> Git init
Initialized empty Git repository in C:/Users/anton/Documents/ControlDeVersiones/.git/
```

Una vez hayamos creado la copia del repositorio, tendremos que añadir los archivos que queramos meter en nuestro repositorio, esto lo haremos con el comando “Git add” y el archivos que queramos subir.

```
PS C:\Users\anton\Documents\ControlDeVersiones> Git add Tema3
```

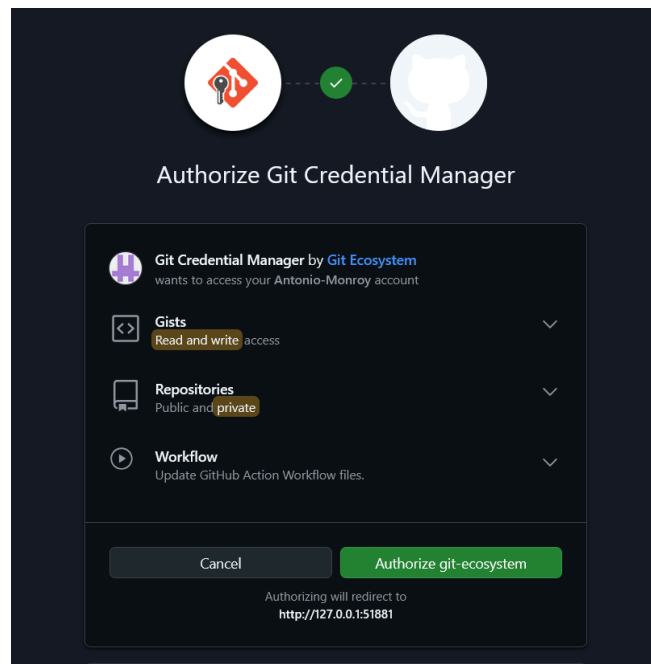
Ahora tendremos que usar el comando “Git commit -m” para poder añadir una nueva versión a nuestro repositorio, podemos añadir un mensaje para tener algo de información a la hora de que tengamos que trabajar con el repositorio.

```
PS C:\Users\anton\Documents\ControlDeVersiones> git commit -m "Hemos añadido las carpetas con las actividades"
[master (root-commit) 684b13d] Hemos añadido las carpetas con las actividades
8 files changed, 191 insertions(+)
create mode 100644 Tema3/EjercicioCajaNegra/EjercicioCajaNegra/Edad.java
create mode 100644 Tema3/EjercicioCajaNegra/EjercicioCajaNegra/Equivalencia.java
create mode 100644 Tema3/EjercicioCajaNegra/EjercicioCajaNegra/Limite.java
create mode 100644 Tema3/Preguntas tema 3_ALP.pdf
create mode 100644 "Tema3/Pruebas de Integraci\303\263n/PruebasEj1.java"
create mode 100644 "Tema3/Pruebas de Integraci\303\263n/PruebasEj2.java"
create mode 100644 "Tema3/Pruebas de Integraci\303\263n/PruebasEj3.java"
create mode 100644 "Tema3/Pruebas de Integraci\303\263n/PruebasEj4.java"
```

Para poder subir el repositorio a GitHub tendremos que usar el comando “Git push” como es la primera vez que lo usamos en este repositorio, tenemos que añadir también el nombre del repositorio y la palabra “master”.

```
PS C:\Users\anton\Documents\ControlDeVersiones> git push ControlDeVersiones master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 201.43 KiB | 3.41 MiB/s, done.
Total 14 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
To ControlDeVersiones
 * [new branch]      master -> master
```

Como es la primera vez que subimos un repositorio a GitHub, nos van a pedir una confirmación por tema de seguridad.



Una vez subida la versión, si entramos a nuestro repositorio desde GitHub, podremos ver que tenemos la carpeta que hemos añadido anteriormente.

