

Control de versiones con Git



Objetivo:

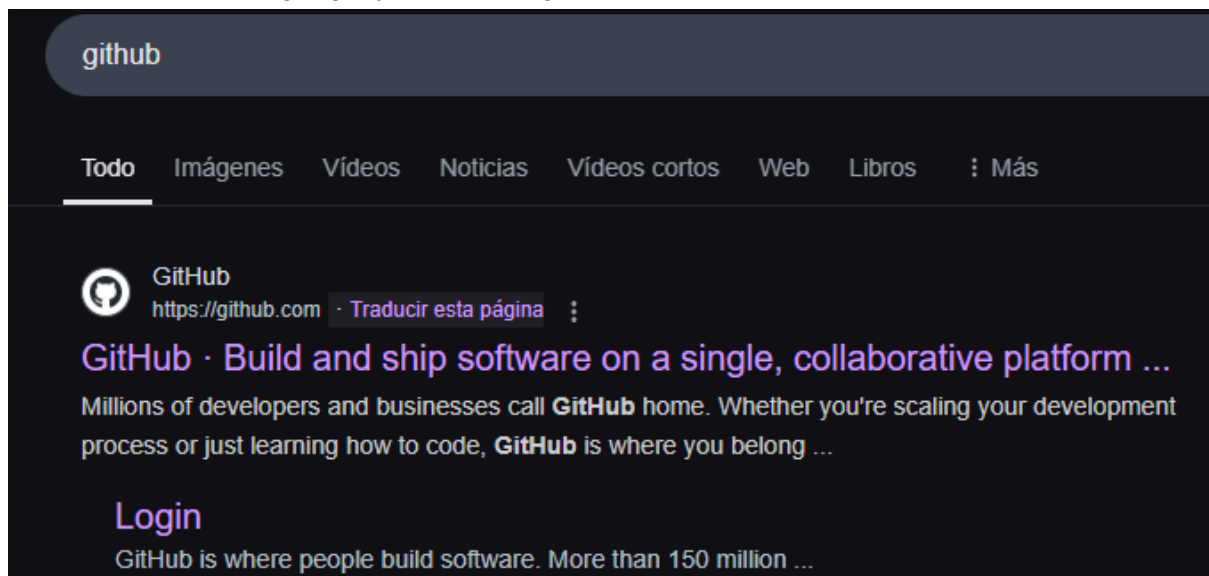
Crea un proyecto en GitHub en el que deberás subir todas las entregas del curso organizadas por tema. Añade un documento PDF en el que deberás explicar detalladamente el proceso de inicio de GitHub así como los comandos más usuales para poder llevar un control de las versiones de un proyecto.

Sube en la entrega el enlace a tu repositorio de GitHub.

1. Creación de cuenta

Para usar github necesitamos tener una cuenta registrada allí.

1. Nos vamos a google y buscamos github



2. Una vez dentro de Github hacemos click en la esquina superior derecha en donde pone "Sign up"



3. Aquí tendremos que poner nuestro email, la contraseña y el nombre que vamos a tener en Github.

Sign up to GitHub

Email*

Password*

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

Username*

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

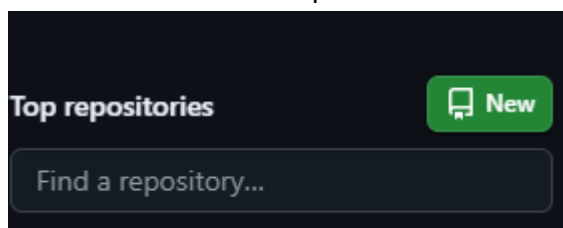
Continue >

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

2. Creación del repositorio

Una vez creada nuestra cuenta en Github podremos crear nuestros repositorios.

1. Para crear un repositorio haremos click en el recuadro verde en donde pone "NEW"




2. Una vez aquí nos encontraremos algunas opciones, debemos encargarnos de que el repositorio tenga título, una pequeña descripción, que sea público y que tenga un "README"

Create a new repository

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

Required fields are marked with an asterisk ().*


Owner * **Repository name ***


 fnr5u2fng4258gj /

✔ **AlbaFelix is available.**

Great repository names are short and memorable. Need inspiration? How about **solid-sniffle** ?

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.

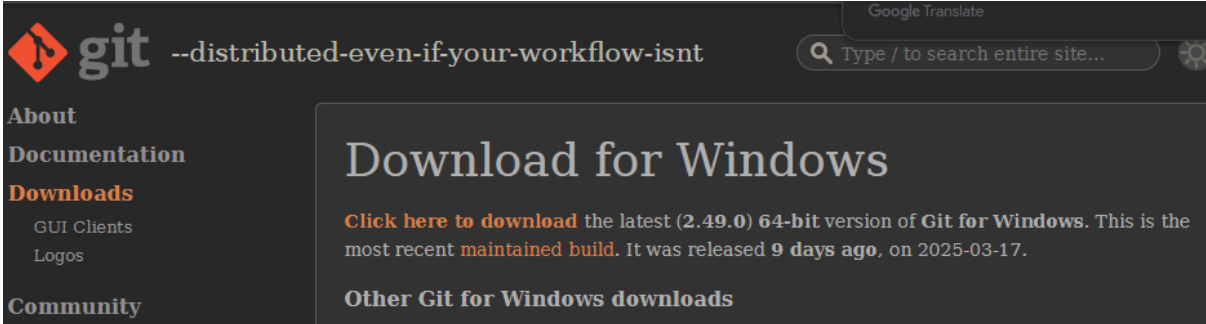
Initialize this repository with:

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

3. Añadir documentos

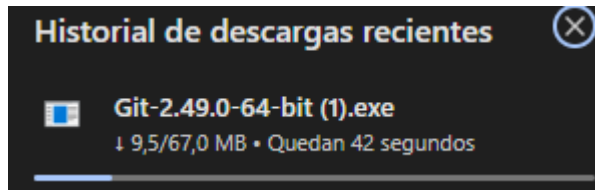
Podemos añadir documentos a Github tanto por el propio Github como por Git, en este caso usaremos Git.

1. Descargamos Git(Lo buscamos en Google y hacemos clic en descargas)



The screenshot shows the Git website with the following elements:

- Header:** Git logo, tagline "--distributed-even-if-your-workflow-isnt", Google Translate button, and a search bar.
- Left Navigation:** Links for About, Documentation, Downloads (highlighted), GUI Clients, Logos, and Community.
- Main Content:** A large section titled "Download for Windows" with the text: "Click here to download the latest (2.49.0) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 9 days ago, on 2025-03-17." Below this is a section for "Other Git for Windows downloads".



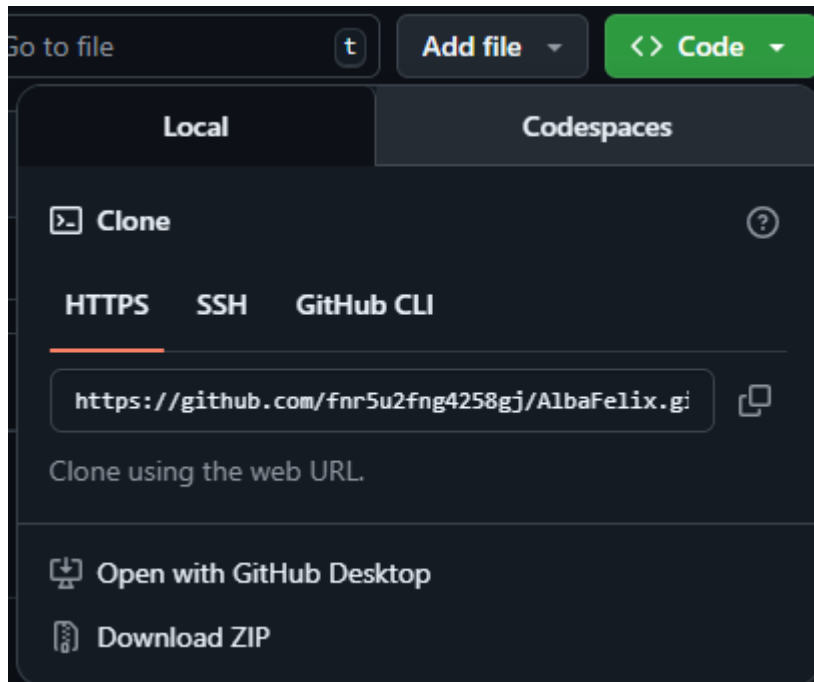
2. Una vez descargado lo iniciamos y le damos a next hasta que se descargue en nuestro dispositivo.



3. Creamos nuestro usuario en git con los siguientes comandos



4. Clonamos el repositorio en la máquina virtual mediante este comando(El enlace lo obtenemos en Github haciendo clic en el recuadro verde donde pone "code").

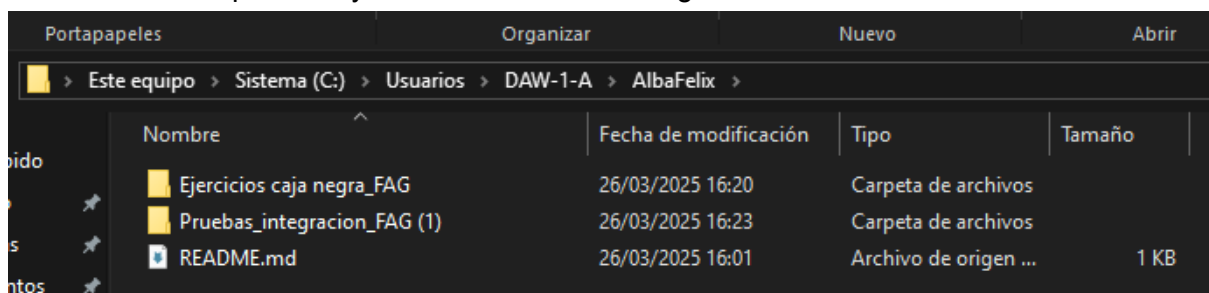


```
PS C:\Users\DAW-1-A> git clone https://github.com/fnr5u2fng4258gj/AlbaFelix.git
Cloning into 'AlbaFelix'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
PS C:\Users\DAW-1-A>
```

5. Una vez clonado el repositorio nos vamos a él, creamos una rama y cambiamos a esa rama.

```
PS C:\Users\DAW-1-A> cd .\AlbaFelix\
PS C:\Users\DAW-1-A\AlbaFelix> git branch ActividadesTema3_FAG
PS C:\Users\DAW-1-A\AlbaFelix> git checkout ActividadesTema3_FAG
Switched to branch 'ActividadesTema3_FAG'
```

6. Nos vamos a nuestro ordenador y pegamos los archivos que vamos a añadir a nuestro repositorio y los añadiremos con un git add.



7. Para finalizar, añadimos las carpetas con un git add y con git commit -m comentamos los cambios realizados.

```
PS C:\Users\DAW-1-A\AlbaFelix> git add .\EjerciciosCajaNegra_FAG\  
PS C:\Users\DAW-1-A\AlbaFelix> git commit -m "carpeta de actividades caja negra añadida"  
[main 28da0f9] carpeta de actividades caja negra añadida  
3 files changed, 81 insertions(+)  
create mode 100644 EjerciciosCajaNegra_FAG/Ejercicios caja negra_FAG/Edad.java  
create mode 100644 EjerciciosCajaNegra_FAG/Ejercicios caja negra_FAG/EquivalenciaEdadTest.java  
create mode 100644 EjerciciosCajaNegra_FAG/Ejercicios caja negra_FAG/LimiteEdadTest.java  
PS C:\Users\DAW-1-A\AlbaFelix> git add .\Pruebas_integracion_FAG\  
PS C:\Users\DAW-1-A\AlbaFelix> git commit -m "carpeta de actividades de integracion añadida"  
[main c9ef5e6] carpeta de actividades de integracion añadida  
24 files changed, 267 insertions(+)
```

8. Las carpetas deberían verse así en main.

AlbaFelix Public

main 1 Branch 0 Tags

Go to file t Add file <> Code

fnr5u2fng4258gj	carpeta de actividades de integracion añadida	c9ef5e6 · 8 minutes ago	3 Commits
EjerciciosCajaNegra_FAG/Ejercicios caja negra_...	carpeta de actividades caja negra añadida	8 minutes ago	
Pruebas_integracion_FAG/Pruebas_integracion_...	carpeta de actividades de integracion añadida	8 minutes ago	
README.md	Initial commit	35 minutes ago	