

Notas de GitHub

Fuentes URL's:

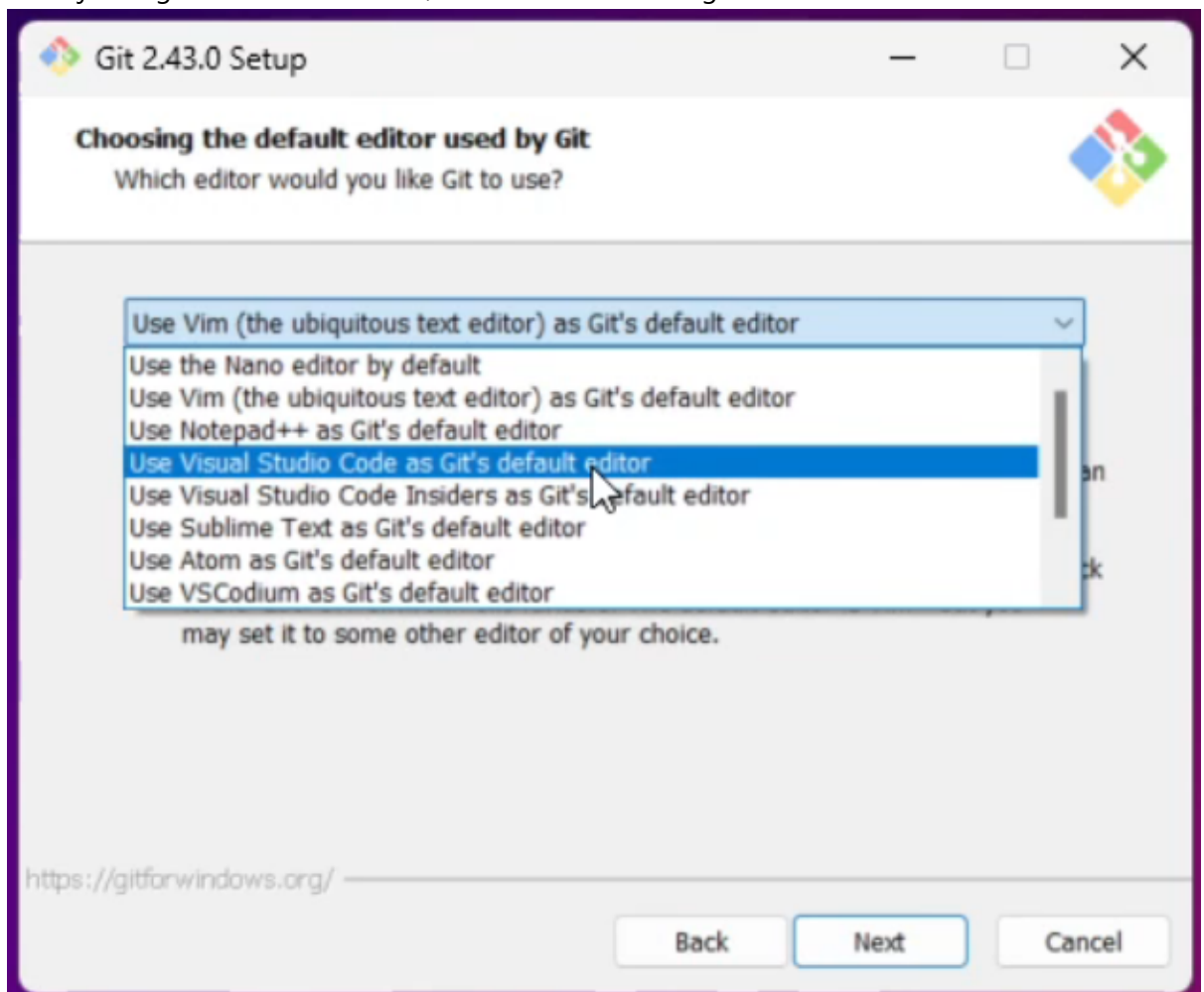
1. [certificacion-github-foundations-temario-completo-y-examen](#)

Primeros pasos

00 - Instalaciones

Deberemos instalar 2 cosas principalmente:

1. Git, de la página oficial www.git-scm.com en nuestro navegador.
1. **NOTA IMPORTANTE** : NO seleccionar VIM como editor de Git.
2. Es mejor elegir Visual Studio Code, aún cuando no lo tengamos instalado todavía.

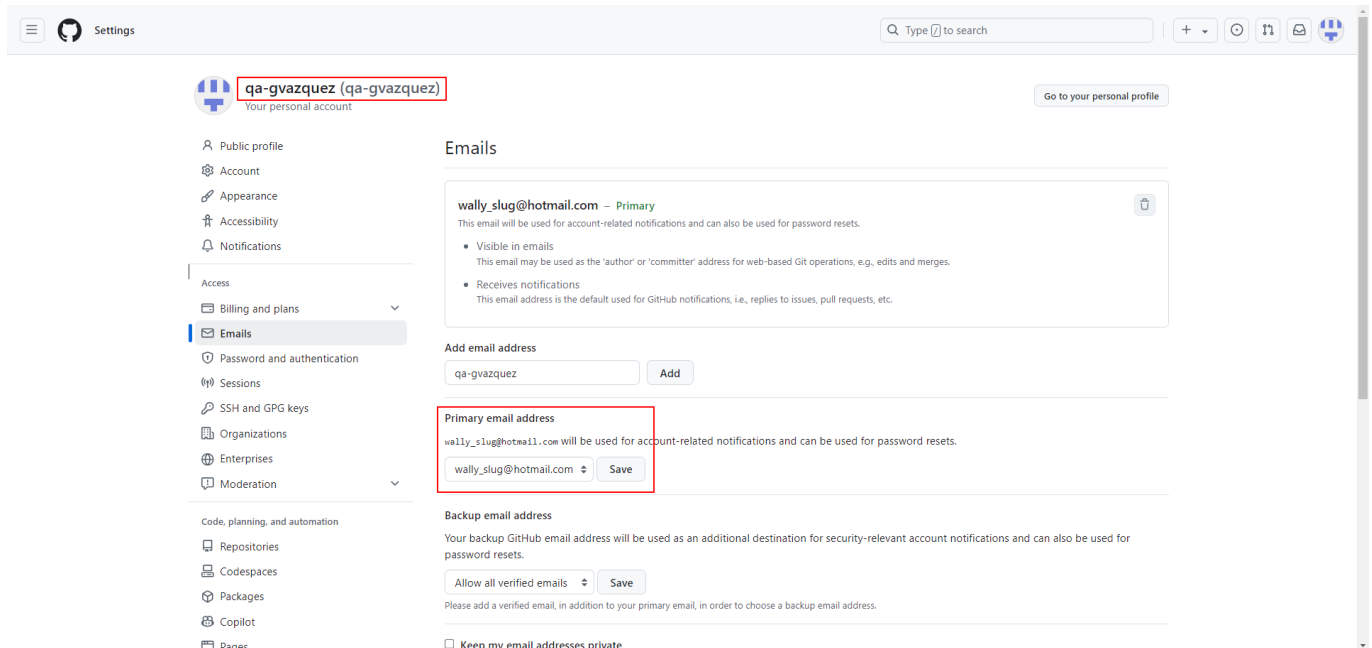


3. Hay otro paso donde nos pide cambiar el nombre de la rama principal de Master a **main**, o lo que querremos. Elegir esa casilla.

2. Y el IDE [Visual Studio Code](#)

01 - Creación de una cuenta en GitHub, en línea.

1. Entremos a la página de GitHub y únicamente con un Email podemos crear una cuenta.
2. Entremos en el enlace <https://github.com/settings/emails> y busquemos dos datos importantes:
 1. Nombre de usuario
 2. Email

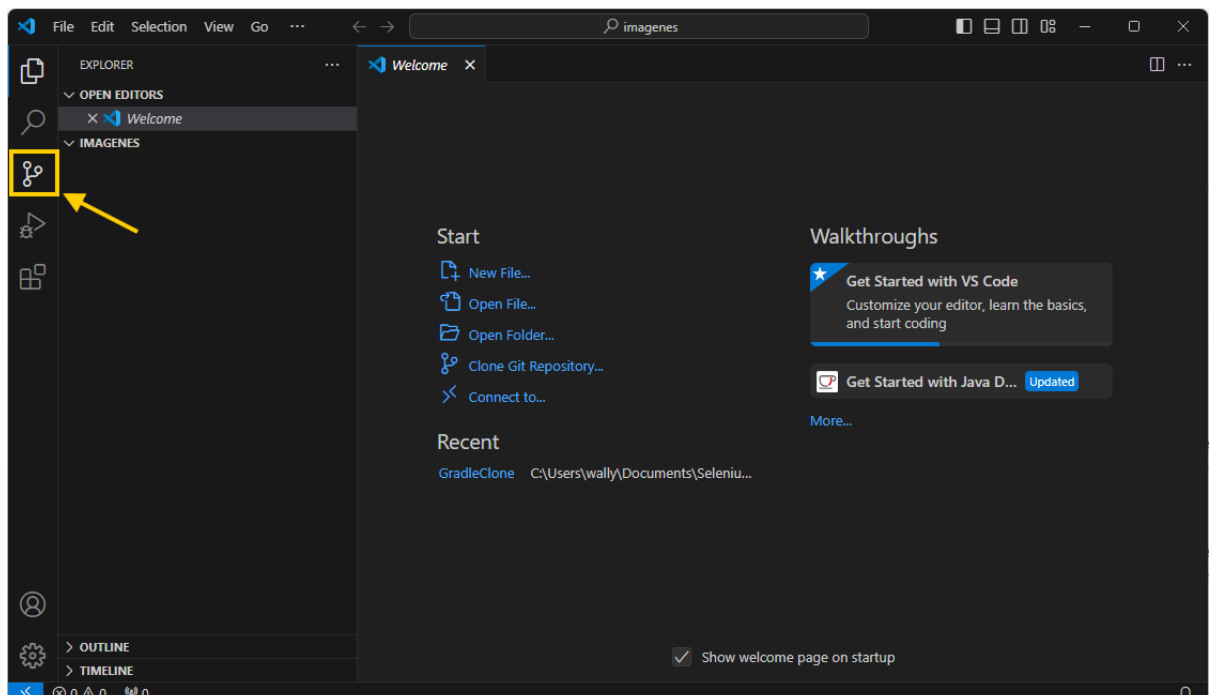


02 - Abrir el Directorio, Fichero o Carpeta de los archivos a respaldar.

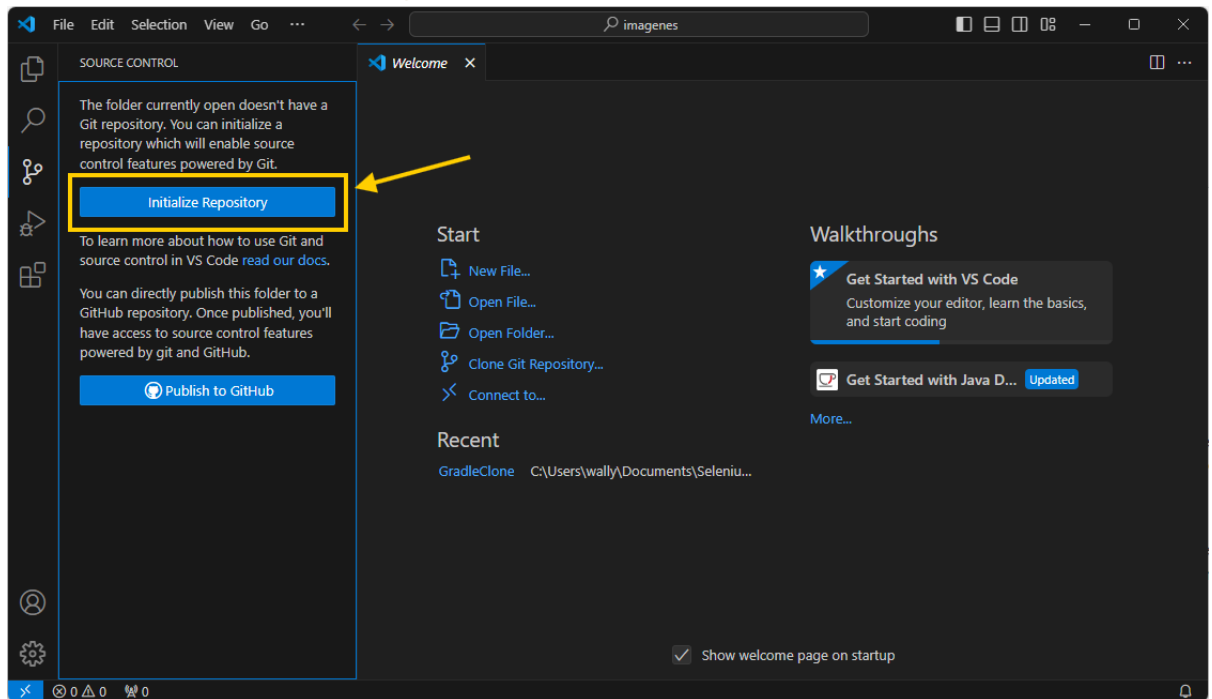
Ya sea que creemos una carpeta nueva, o sencillamente nos dirigimos a una que ya contenga archivos de interés.

1. Clic derecho > **Open Git Bash Here**
2. Teclear: **git init** o
3. Clic derecho > Mostrar mas opciones > **Abrir con Code** (Visual Studio Code)

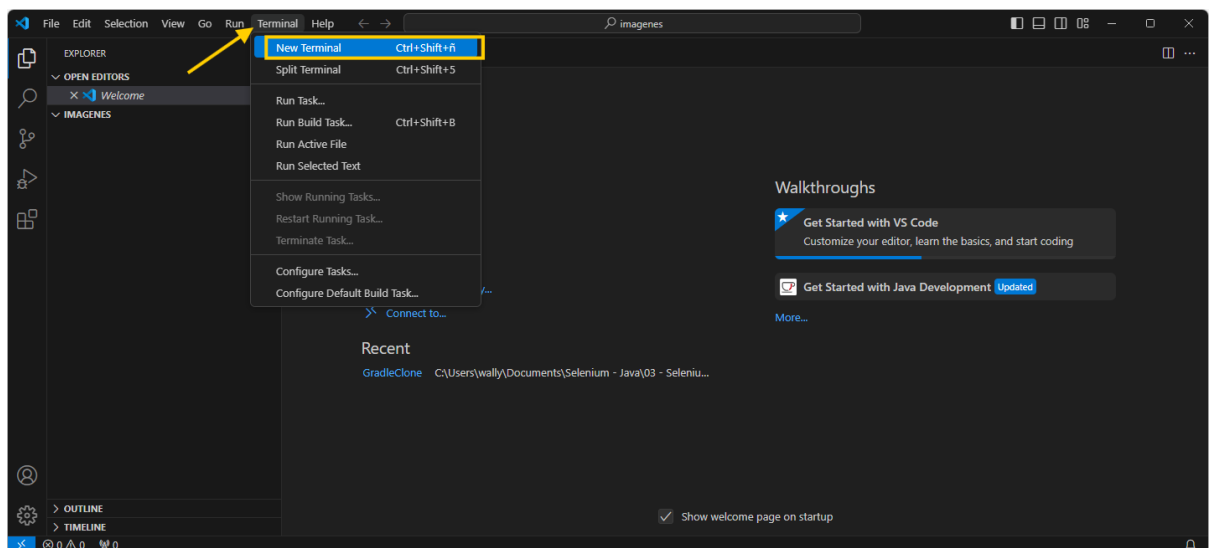
1. Dar clic en el ícono de **Source Control**



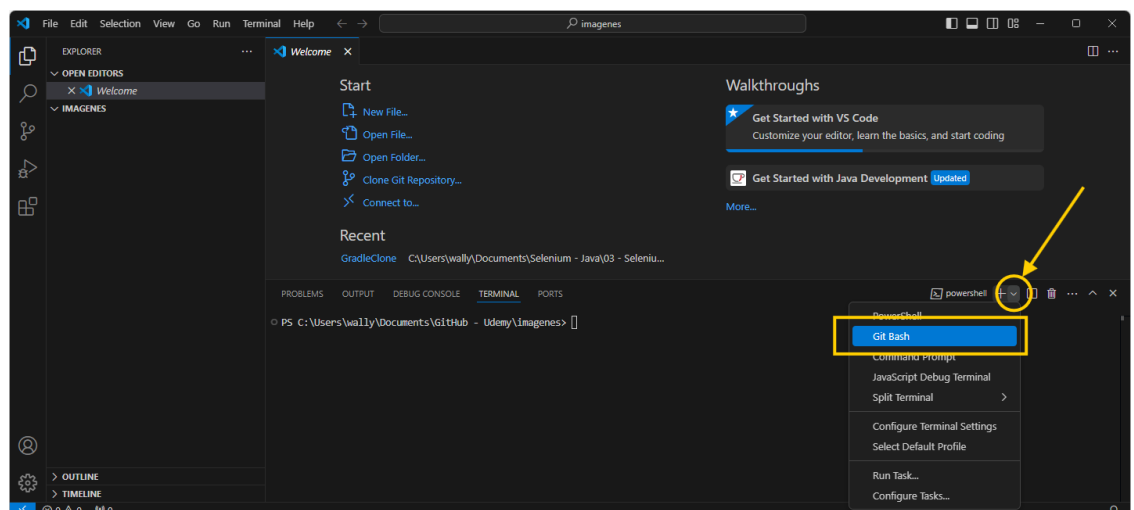
2. Clic en **Initialize Repository**



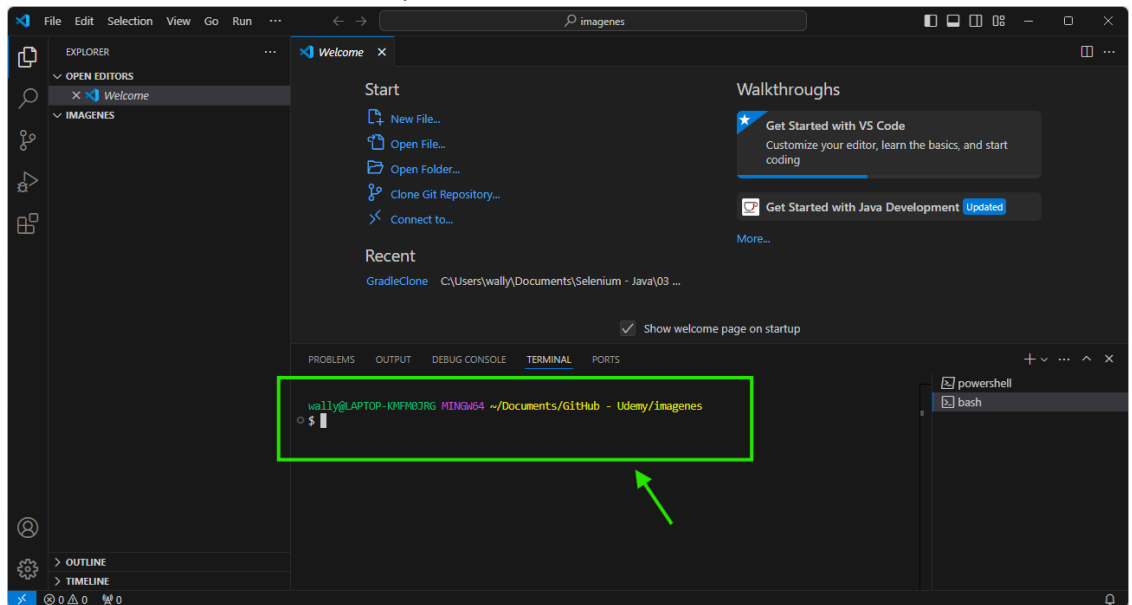
3. Abrir la Terminal en VSC.



1. Debe ser la BASH



2. La terminal BASH se caracteriza por ser colorida:



03 - Configuración de Git, local, desde la terminal BASH.

Debemos sincronizar la información de usuario del Repositorio en línea **GitHub**, con nuestro sistema de control de versiones local **Git**.

La información de **Nombre** e **Email** se debe configurar los datos del usuario reflejando los datos del Repositorio de GitHub.

Debemos teclear los datos manualmente:

```
git config --global user.name "qa-gvazquez"
git config --global user.email "wally_slug@hotmail.com"
```

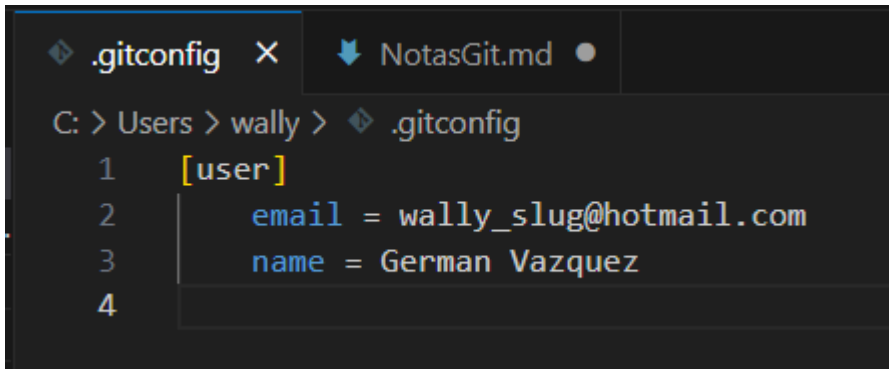
Para verificar que se aplicaron los cambios, en la Terminal **Bash** debemos teclear:

```
git config --global -e
```



Se abrirá una ventana nueva en el editor por defecto (Debería ser Visual Studio Code, si hicimos bien la instalación de Git) con el archivo **.gitconfig** donde se enlistan los ajustes de **Git** local.

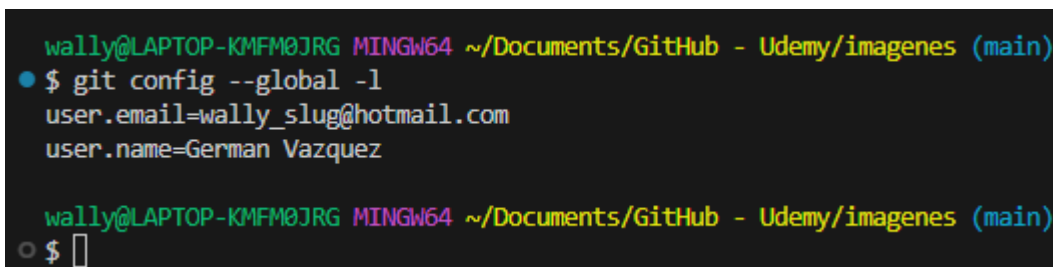
Sencillamente cerramos la ventana.



```
.gitconfig x NotasGit.md
C: > Users > wally > .gitconfig
1 [user]
2     email = wally_slug@hotmail.com
3     name = German Vazquez
4
```

También podemos hacer lo mismo con una sola línea de comando

```
git config --global -l
```

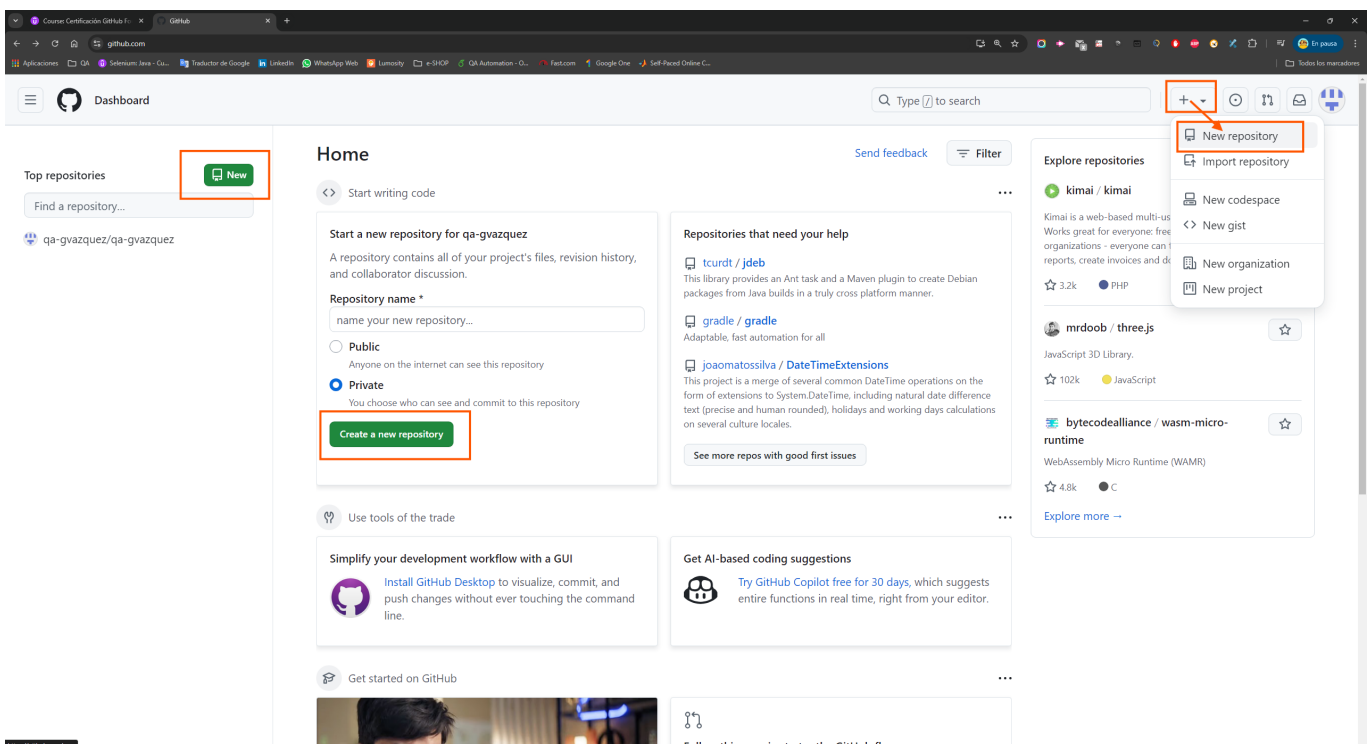


```
wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
$ git config --global -l
user.email=wally_slug@hotmail.com
user.name=German Vazquez

wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
$
```

04 - Creando un nuevo Repositorio en GitHub, en línea.

Una vez en nuestra cuenta de GitHub, tendremos muchas opciones para crear un Nuevo Repositorio, o un espacio en la Web para compartir nuestro proyecto.



Únicamente, debemos colocar un nombre al Repo, sin espacios.

También, elegir si debe ser **Público** o **Privado**

y dar clic en **Create repository**

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 * qa-gvazquez / Repository name * GitHub-101
GitHub-101 is available.

Great repository names are short and memorable. Need inspiration? How about [redesigned-rotary-phone](#) ?

Description (optional)
How to Git? Detailed guide to create your first Version Control System project on local 'Git' and upload to 'Git'

☒ 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.](#)

☐ Choose a license
License: None
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

☐ You are creating a public repository in your personal account.

Create repository

© 2024 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

05 - Subiendo el proyecto a un Repositorio

1. Si no lo hemos hecho aún, debemos abrir la carpeta de los archivos a respaldar, e inicializar el proyecto Git (local) con el comando:

```
git init
```

Esto ya se hizo en un paso anterior, pero podemos comprobar que nuestra carpeta está siendo monitoreada, con el comando:

```
ls -al
```

que enlista todo el contenido de nuestra carpeta, deberíamos poder ver una carpeta oculta llamada **.git** que contiene los archivos del sistema de control de versiones.

```
wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes
● $ git config --global -e

wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes
● $ ls -al
total 8
drwxr-xr-x 1 wally 197618 0 Sep 20 12:45 ./
drwxr-xr-x 1 wally 197618 0 Sep 20 13:07 ../
drwxr-xr-x 1 wally 197618 0 Sep 20 12:45 .git/

wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
○ $
```

2. Debemos comenzar a monitorear los cambios (versiones) realizados a nuestro proyecto.

```
git status
```

```
wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
● $ git status
On branch main

No commits yet

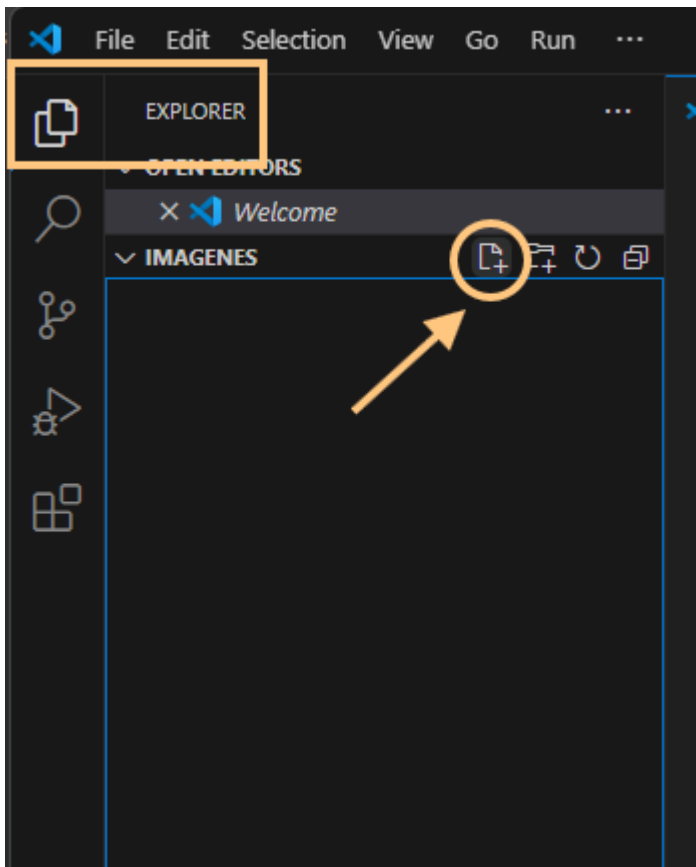
nothing to commit (create/copy files and use "git add" to track)

wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
○ $
```

En este caso, como la carpeta está vacía, no hay nada que monitorear.

Creamos 2 archivos de prueba, para tener un cambio después de haber inicializado el sistema de control de versiones.

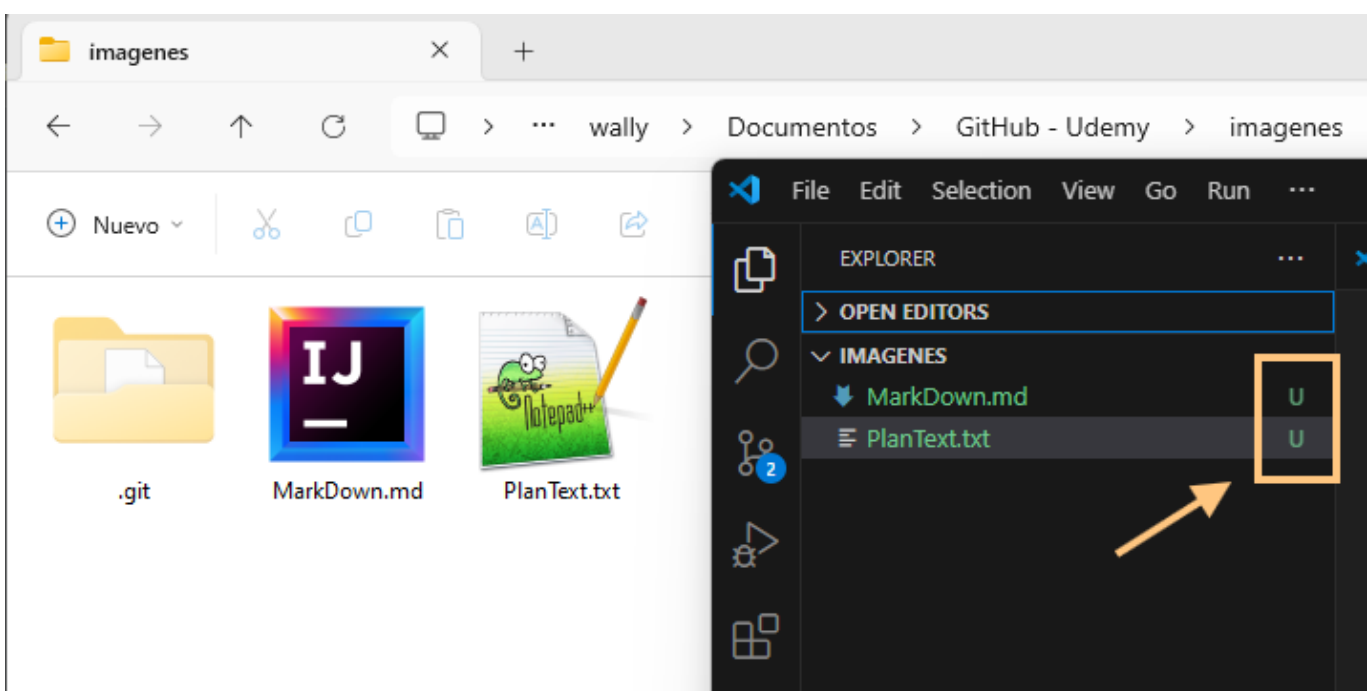
En la barra de herramientas vertical de Visual Studio Code (VSC o Code) seleccionamos el **Explor**er y luego el ícono de 'New File'



Se crean 2 archivos de texto, como ejemplo, pero deberíamos tener los archivos a trabajar. El objetivo es tener 'algo' que monitorear.

NOTA: Git en VSC y GitHub se destacan en el control de cambios de archivos de TEXTO, pero podemos gestionar los cambios de cualquier tipo de archivos.

Como ya tenemos una 'diferencia' desde que inicializamos el proyecto, VSC nos apoya visualmente con marcadores para indicar que tenemos archivos 'nuevos' en el **WorkSpace** sin 'rasterar'.



Ahora, debemos especificar que SÍ queremos que se guarden estos cambios.

Hay 2 maneras:

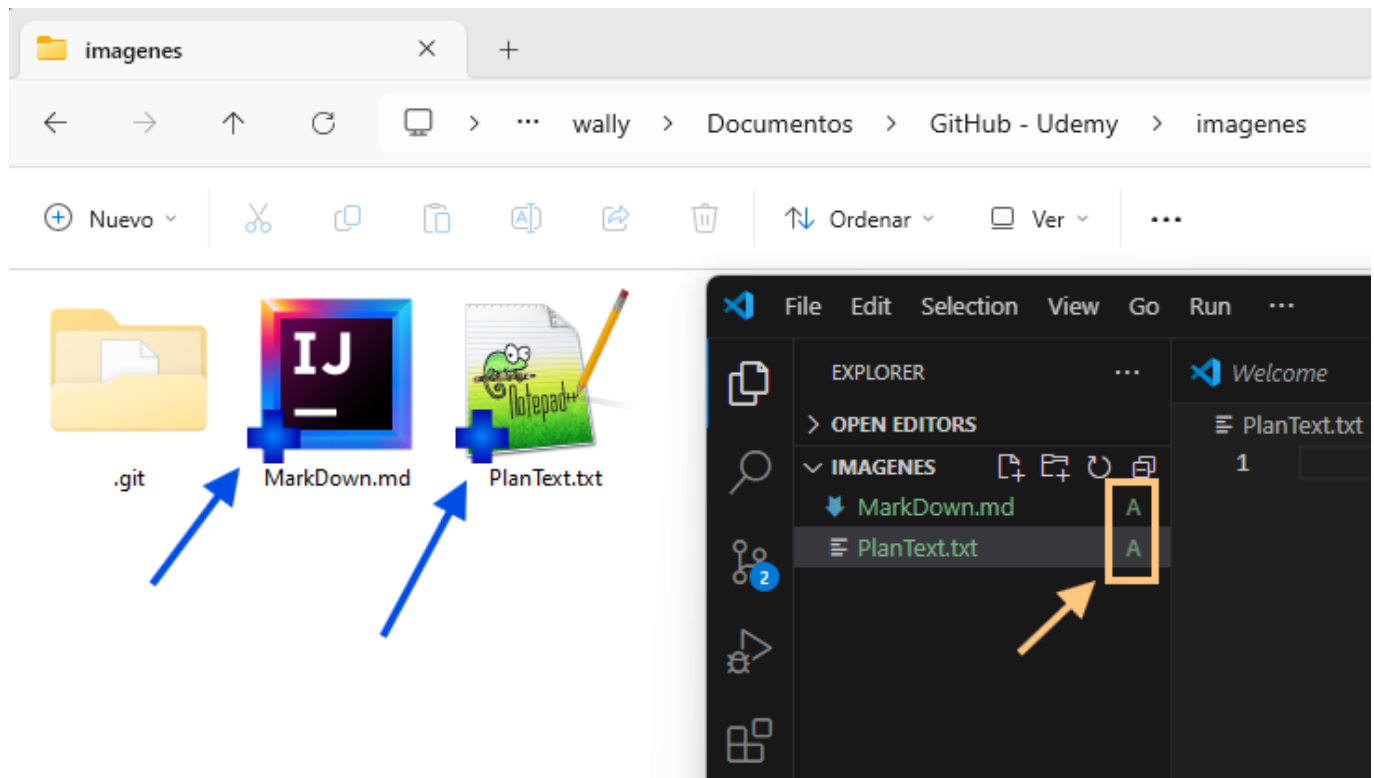
1. Individualmente cada archivo **Cherry Pick** con el comando:

```
git add Mar + TAB
```

Esto, nos ayuda a autocompletar el nombre del archivo individual, pero debemos repetirlo con el otro 2. Todos dentro de la carpeta del proyecto:

```
git add .
```

Ahora, el indicador visual cambió a **Index Added** e incluso en la carpeta misma, el ícono ha cambiado para identificar esos archivos como 'monitoreados'.



Volvemos a monitorear los cambios (versiones) realizados a nuestro proyecto.

```
git status
```

```
wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
• $ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Markdown.md
        new file:   PlanText.txt

wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
• $
```

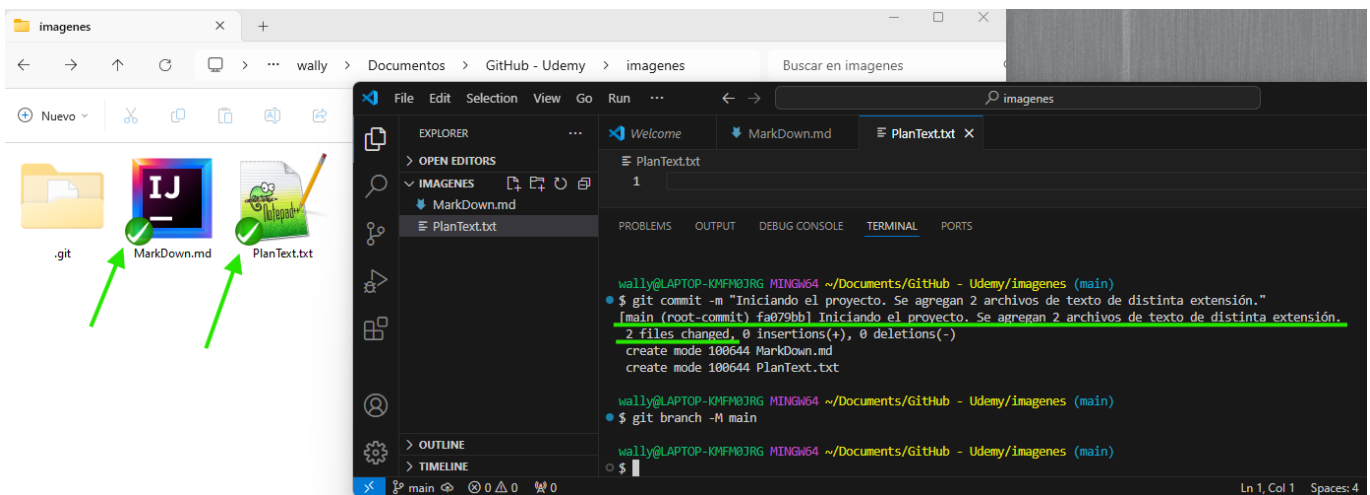
Esto significa que nuestros cambios están en el **Staging Area**, es decir, que ya están listos para ser agregados a la siguiente 'version' o ser guardados en el sistema de control de cambios.

Esta versión guardada, o estado, se denomina **Commit** y es el principal elemento del sistema de control de versiones de Git.

Como tal, el **Commit** es un punto de restauración que se identifica con un ID único, denominado **HASH**, y requiere de una breve descripción de los cambios realizados desde la última vez que se guardaron.

En este caso, es la primera vez que los guardamos, y escribimos el comando:

```
git commit -m "first commit"
```



A continuación, homologamos el nombre de la rama principal de nuestro Git local, con el que utiliza GitHub.

Generalmente, Git lo nombra **MASTERY** GitHub le denomina **MAIN**.

Hacemos ese cambio de nombre en nuestro repositorio local con el comando:

```
git branch -M main
```

Aunque, si configuramos correctamente Git en la instalación, esto ya debería llamarse así.


```
wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
● $ git branch -M main

wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
○ $
```

Así se vería el cambio de nombre de la rama principal

```
kevin@Kirito MINGW64 ~/OneDrive/Imágenes/Escritorio/Github (master)
$ git branch -M main

kevin@Kirito MINGW64 ~/OneDrive/Imágenes/Escritorio/Github (main)
$ git remote add origin https://github.com/Udemyg/Curso-GitHub.git
```



Finalmente, definimos en la configuración de nuestro repositorio local GIT, la dirección URL de nuestro repositorio en línea GITHUB, para que todos los cambios puedan ser compartidos con el mundo. El comando es:

```
git remote add origin https://github.com/qa-gvazquez/GitHub-101.git
```

```
wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
● $ git branch -M main

wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
● $ git remote add origin https://github.com/qa-gvazquez/GitHub-101.git

wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
○ $
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF

```
git remote add origin https://github.com/qa-gvazquez/GitHub-101.git
git push -u origin main
```

Y, para cerrar con Broche de Oro, finalmente subimos los cambios:

```
git push -u origin main
```

Y, falla por que teníamos registradas las credenciales de otro usuario de GitHub.. La solución está en otra sección de este documento.

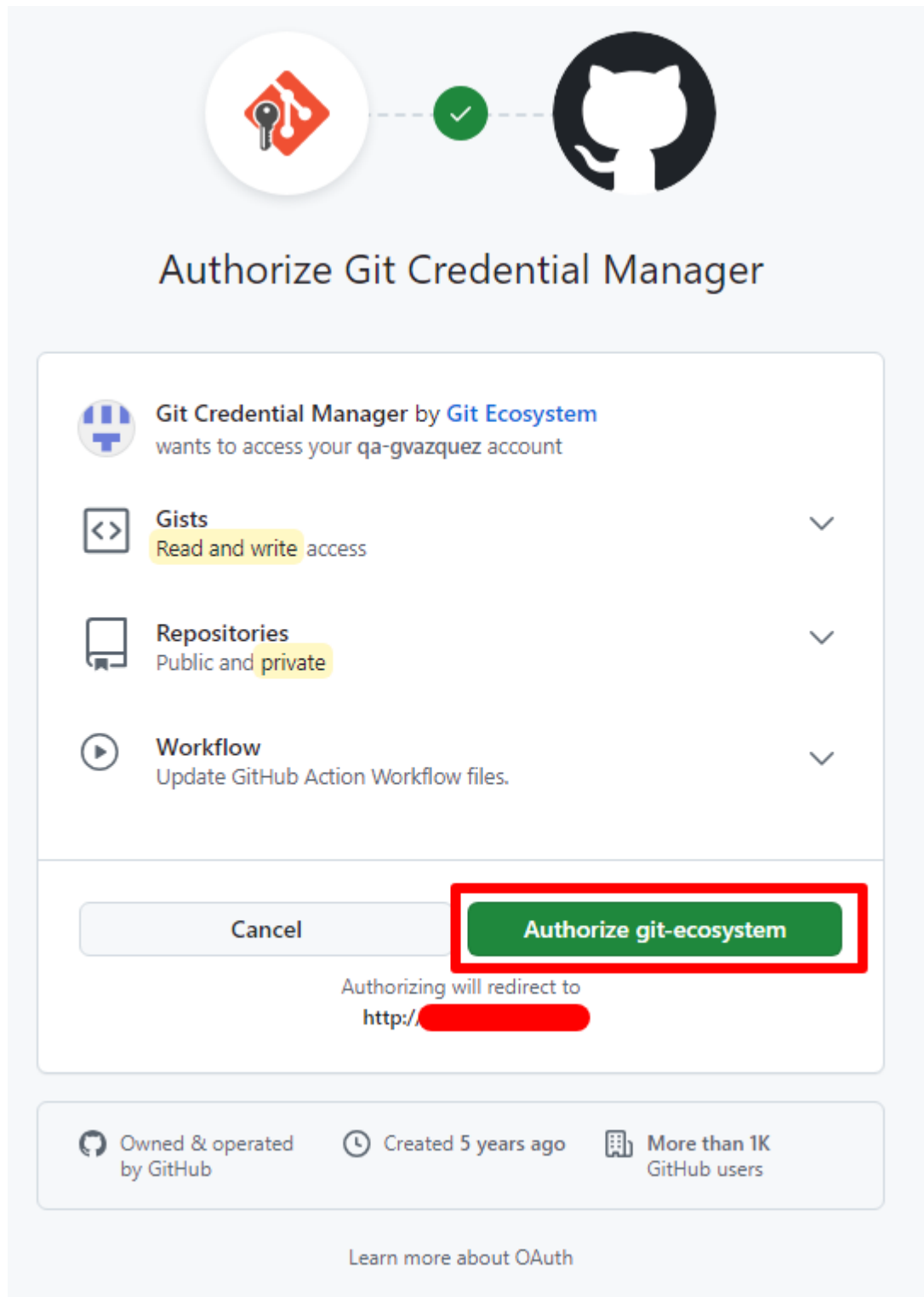
Una vez corregido el detalle, volvemos a ejecutar el comando

```
git push -u origin main
```

Y la extensión de GIT preinstalada en Visual Studio Code, nos abre la ventana de conexión a GitHub:



Elegimos el navegador en el que ya tenemos abierta la sesión de GitHub, y elegimos la siguiente opción:



Finalmente, debemos teclear nuestra contraseña de la cuenta de GitHub y dar clic en **Confirm**.



Confirm access



Signed in as @qa-gvazquez

Password

[Forgot password?](#)

Confirm

Tip: You are entering [sudo mode](#). After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity.

En la terminal de Bash en VSC aparecerá el siguiente mensaje:

```
wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 287 bytes | 287.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/qa-gvazquez/GitHub-101.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemy/imagenes (main)
$
```

Debemos cerrar la ventana del Navegador, y refrescar la ventana del Repositorio Remoto, donde aparecen los pasos a seguir para este proceso:

qa-gvazquez / GitHub-101

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

GitHub-101 Public

Pin Unwatch 1 Fork 0 Star 0

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/qa-gvazquez/GitHub-101.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 "# GitHub-101" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/qa-gvazquez/GitHub-101.git
git push -u origin main
```

Pasos para un proyecto Git que nunca se ha inicializado.

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

```
git remote add origin https://github.com/qa-gvazquez/GitHub-101.git
git branch -M main
git push -u origin main
```

Pasos para un proyecto local, con varios Commit ya realizados.

ProTip! Use the URL for this page when adding GitHub as a remote.

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

Finalmente, podemos ver nuestros cambios en la Web, a disposición de todo el mundo para compartir nuestros conocimientos y respaldar nuestros avances.

06 - Solucionando errores de GitHub.

El error, es que ya teníamos configurado anteriormente un Usuario de GitHub, que tenía otra URL para subir cambios a otro Repositorio.

```
wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemylimagenes (main)
$ git push -u origin main
remote: Permission to qa-gvazquez/GitHub-101.git denied to mx-gvazquez.
fatal: unable to access 'https://github.com/qa-gvazquez/GitHub-101.git/': The requested URL returned error: 403

wally@LAPTOP-KMFM0JRG MINGW64 ~/Documents/GitHub - Udemylimagenes (main)
$
```

The [OP Fre1234](#) adds [in the comments](#):

I finally found the solution.

Go to: **Control Panel -> User Accounts -> Manage your credentials -> Windows Credentials**

Under **Generic Credentials** there are some credentials related to Github, Click on them and click "**Remove**".

That is because the default installation for [Git for Windows](#) set a [Git-Credential-Manager-for-Windows](#).

See `git config --global credential.helper` output (it should be `manager`)

Share Improve this answer Follow

edited Dec 10, 2017 at 18:58

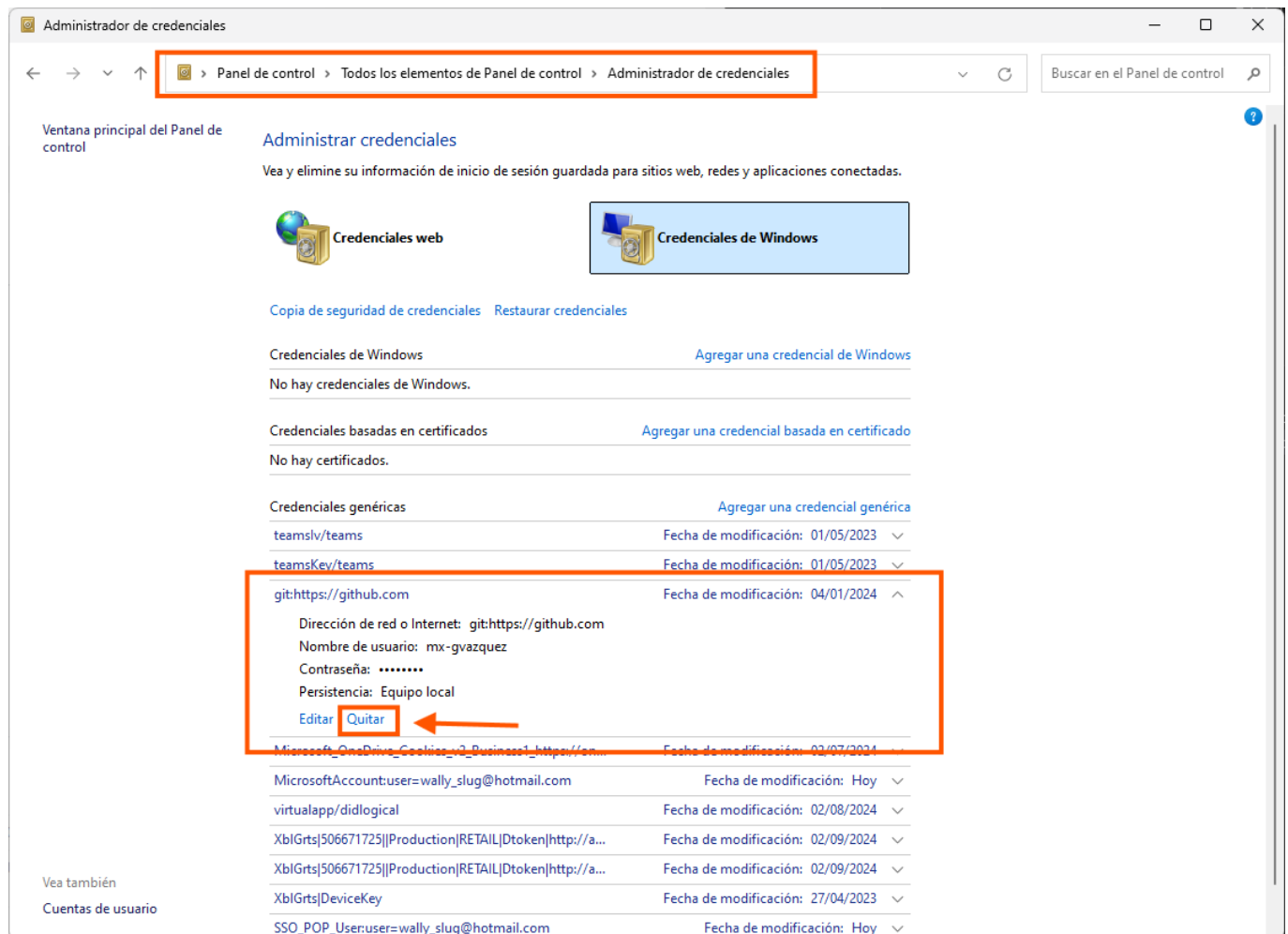
answered Sep 13, 2016 at 5:05



VonC

1.3m ● 555 ● 4.7k ● 5.5k

Source



Con este 'reset' al Administrador de Credenciales, podemos continuar.