



1º DAM/DAW Sistemas Informáticos

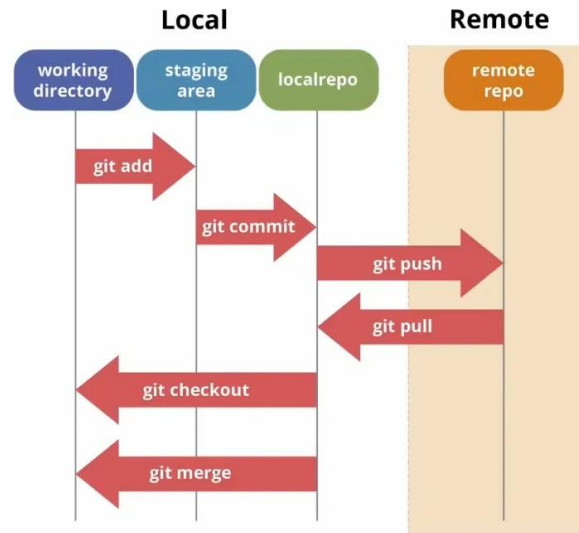
U3. Sistemas de control de versiones - Git

4 - Repositorios remotos



Git. Repositorios remotos

- Hasta este momento, hemos trabajado con repositorios en local. Es habitual, en desarrollos de cierto nivel de complejidad y participación, que el repositorio esté en un servidor remoto y los miembro@s autorizad@s del proyecto interactúen con él. Vamos a explicar ahora cuál sería el **ciclo básico de trabajo con un repositorio remoto**.



Git. Repositorios remotos. Creación

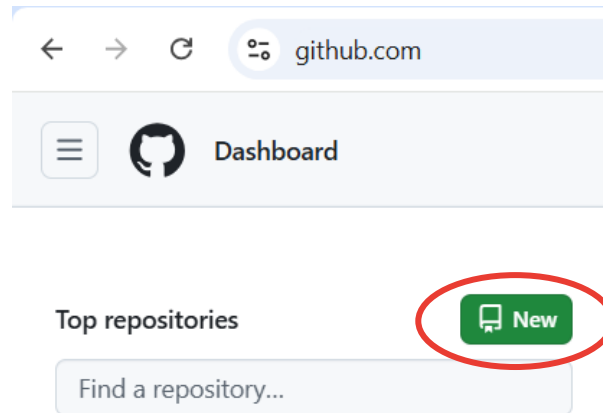
- Existen diversos **mecanismos** básicos y habituales **para iniciar los trabajos** con un repositorio remoto:

1. **Iniciar** el repositorio remoto **desde** una carpeta **local**.

Es decir, he iniciado el proyecto **desde** un **repositorio local** y quiero **sincronizarlo hasta** un **repositorio remoto**:

- a) El primer paso, de forma previa a la sincronización sería crear el repositorio remoto. Se puede hacer mediante múltiples proveedores, nosotros usaremos **GitHub**. Basta con darse de alta y en el apartado **Repositorios**, pulsar el botón **nuevo**.

Git. Repositorios remotos. Creación



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 *
✓ SIN_U3 is available.

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

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

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

Choose a license

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

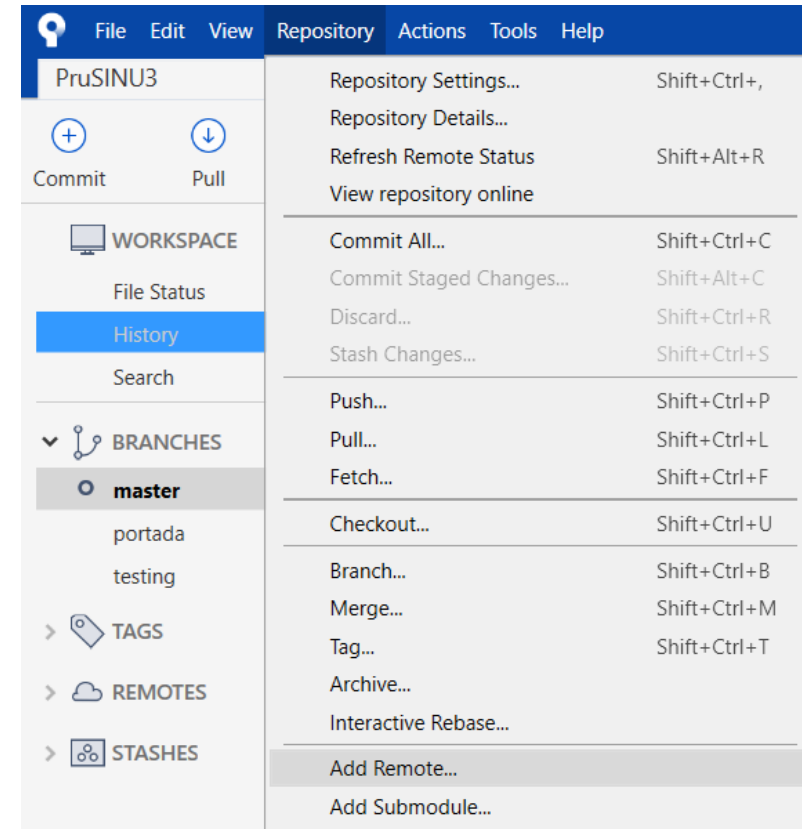
① You are creating a public repository in your personal account.

Create repository

Git. Repositorios remotos. Creación

- b) Desde SourceTree, pulsaremos el menú **“Repository”**.

Añadiremos una conexión al registro remoto mediante **“Add remote”**.



Git. Repositorios remotos. Creación

- b) Se nos abrirá un formulario donde pulsaremos el botón **“Add”**, y añadiremos la **ruta URL** del repositorio remoto, un **nombre**, por defecto se usa **“origin”**, y un nombre de usuario autorizado **“username”**.

The screenshot shows the 'Remote details' dialog box. The 'Required information' section is highlighted with a red box, containing the following fields:

- Remote name: ☒ Default remote
- URL / Path:

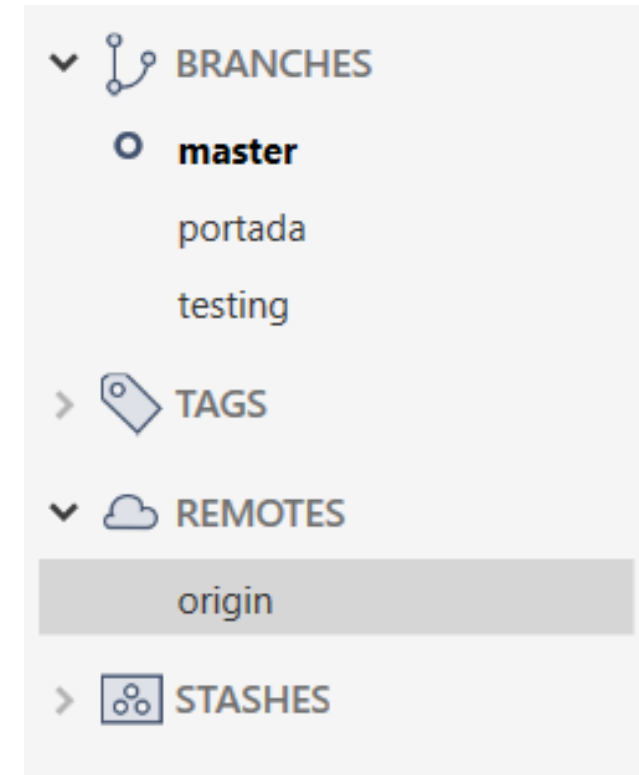
The 'Optional extended integration' section contains the following fields:

- Remote Account: Generic Host
- Legacy Account Settings:
 - Host Type:
 - Host Root URL:
 - Username:

At the bottom right are 'OK' and 'Cancel' buttons.

Git. Repositorios remotos. Creación

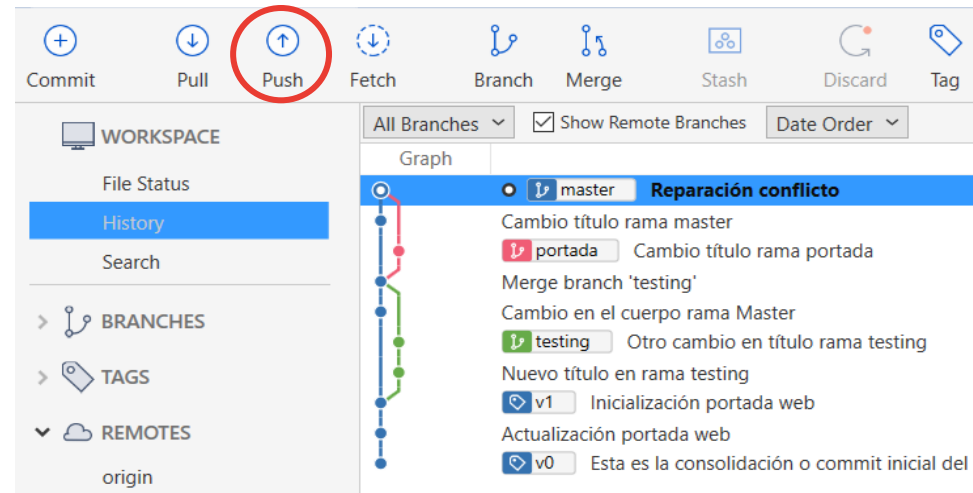
- b) Al aceptar se generará, dentro del menú de la izquierda “**REMOTES**”, la entrada del repositorio “**origin**”, que hemos generado.



Git. Repositorios remotos. Creación

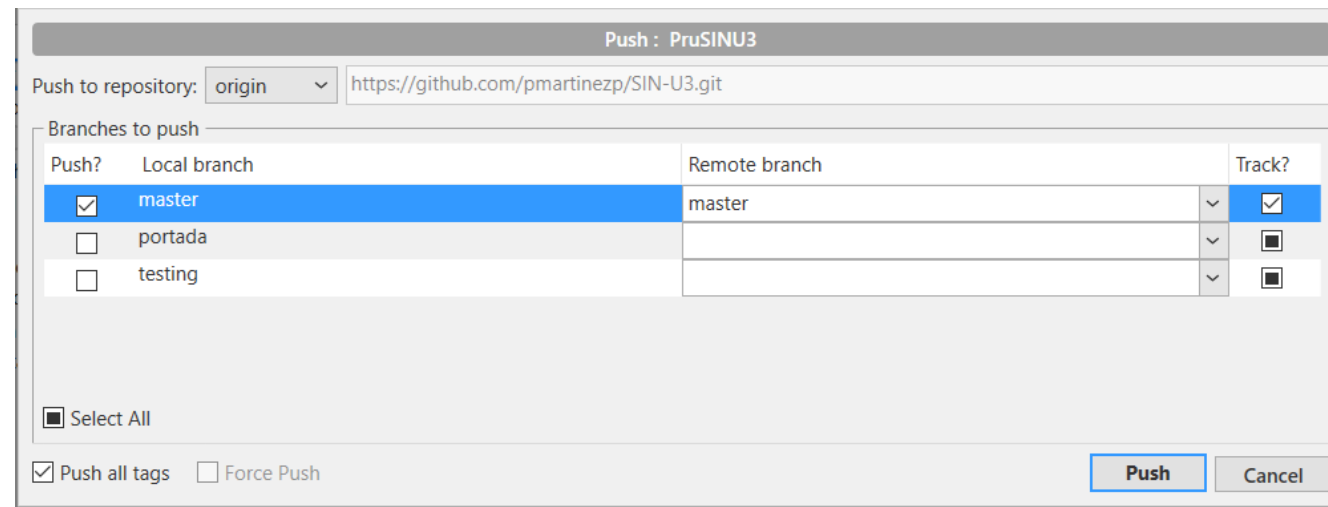
- c) Ahora tenemos que **sincronizar** el repositorio local con el repositorio remoto.

Esta acción de **sincronizar los cambios locales hacia el repositorio remoto** se conoce como: **“Push”**



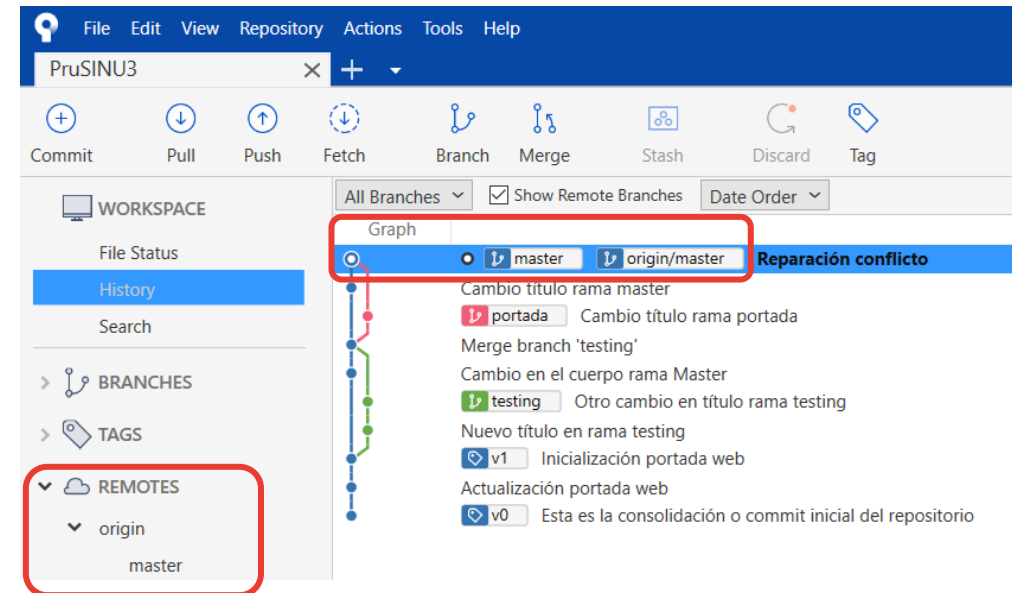
Git. Repositorios remotos. Creación

- d) Seleccionamos la/s rama/s que se sincronizará/n. En este caso, todas las ramas estaban fusionadas en master.



Git. Repositorios remotos. Creación

- d) Finalmente, dispondremos de un repositorio remoto, iniciado desde nuestro repositorio local. **Las modificaciones que vamos haciendo en el repositorio local, las iremos enviando al repositorio remoto mediante “Push”.**

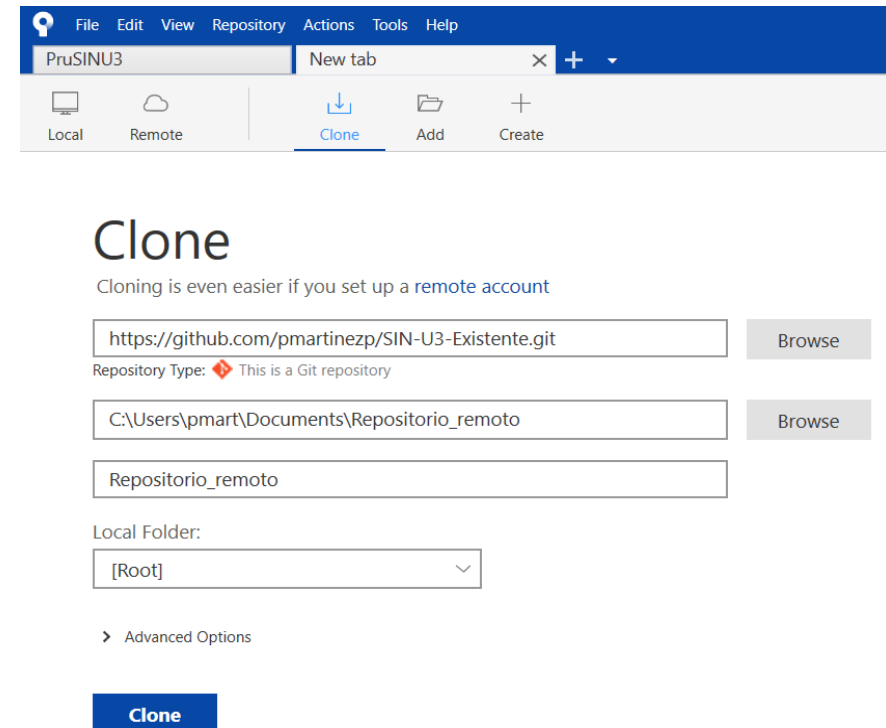


Git. Repositorios remotos. Creación

2. **Iniciar desde un repositorio remoto existente.** Por ejemplo, te incorporas a un proyecto, donde ya existe un repositorio remoto. En este caso, el mecanismo consiste en conectarte al repositorio y traértelo a local, para llevar a cabo tu parte del desarrollo:
 - a) Crear una **carpeta local**, en nuestro sistema de archivos. Será la **raíz del repositorio** (por ejemplo, la denominamos “**Repositorio_remoto**”).

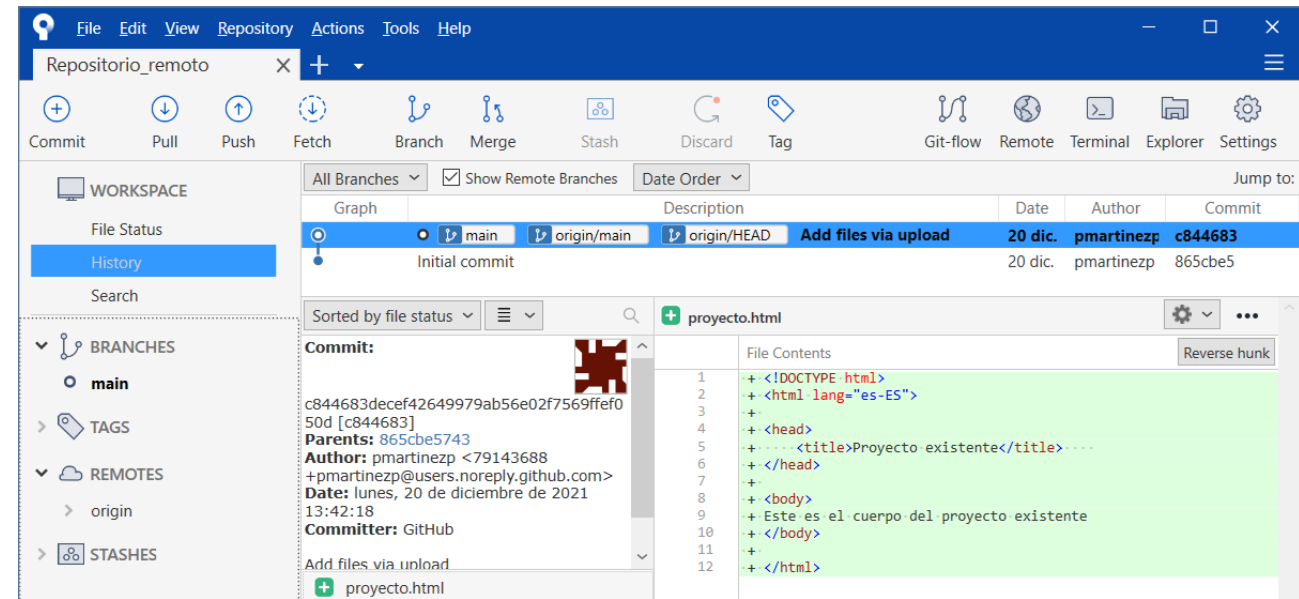
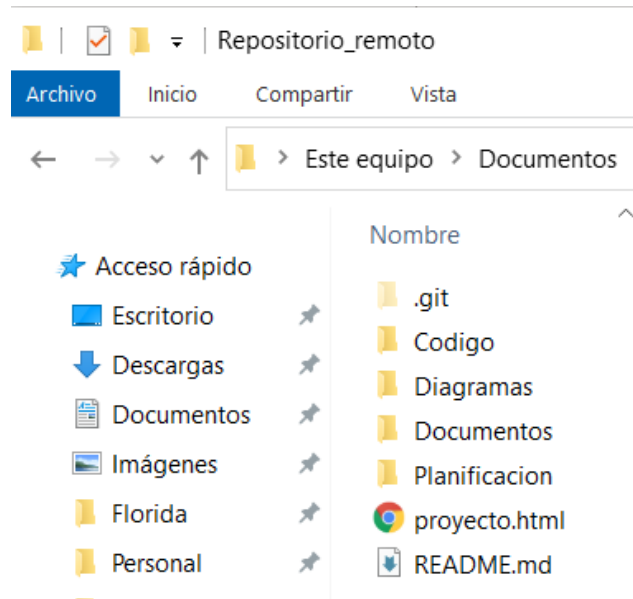
Git. Repositorios remotos. Creación

- b) Nos proporcionarán la **URL del repositorio remoto** y el acceso necesario. En SourceTree podemos **clonar un repositorio nuevo**.



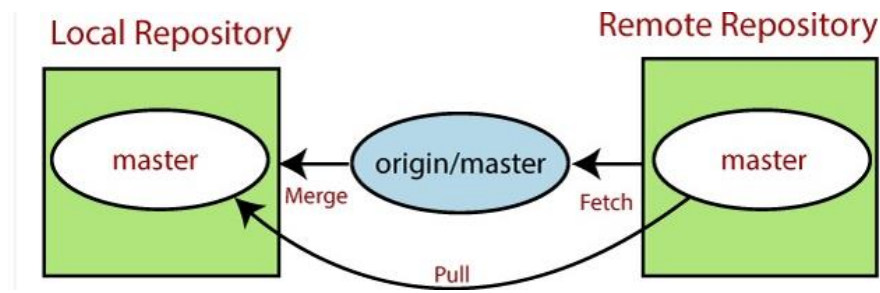
Git. Repositorios remotos. Creación

- b) Una vez clonado el repositorio remoto, disponemos de un repositorio en local, preparado para realizar modificaciones y sincronizar posteriormente.



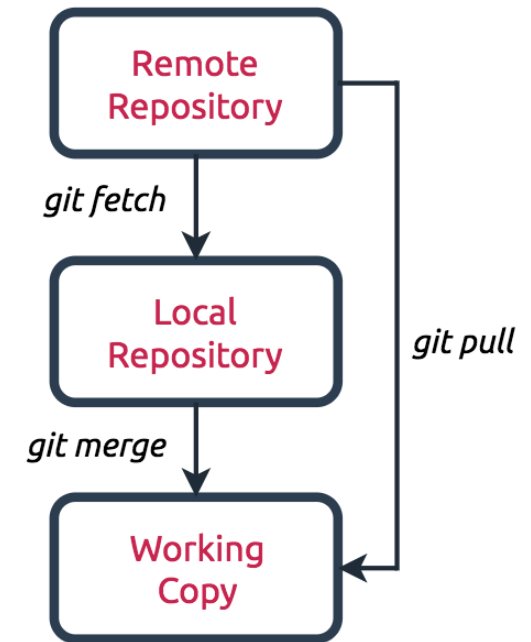
Git. Repositorios remotos. Trabajo en equipo

- El hecho de que en un mismo repositorio remoto trabaje un **equipo de desarrollo** implica, de forma habitual, que tengamos que **ir trayendo** a nuestro repositorio **local**, **cambios consolidados por otr@s compañer@s** en el repositorio remoto e **ir integrándolos** con nuestras **modificaciones locales**.
- Esta acción se conoce como “**Pull**”.
- Internamente hace dos operaciones (**Pull = Fetch + Merge**):



Git. Repositorios remotos. Trabajo en equipo

- El detalle de las 2 operaciones que componen la acción “**Pull**”, es el siguiente (**Pull = Fetch + Merge**):
 - Primero se trae los cambios desde el repositorio remoto, en forma de un nuevo commit (origin/master), al repositorio local, “**Fetch**”.
 - Después integra esos cambios con el último commit que había antes de llegar los cambios (master) al repositorio local, “**Merge**”.



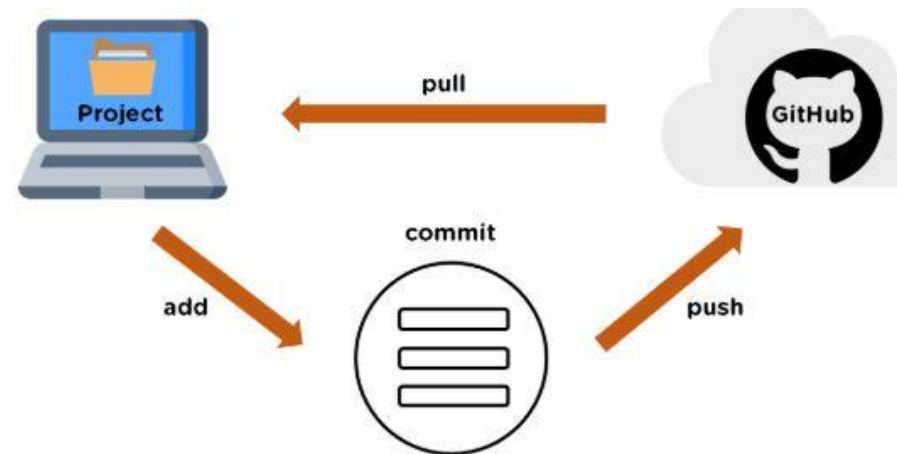
Git. Repositorios remotos. Trabajo en equipo

- Para sincronizar nuestras modificaciones locales con el repositorio remoto, usaremos la acción **“Push”**.



Git. Repositorios remotos. Trabajo en equipo

- Ahora bien, cuando hagamos “Push”, si en el repositorio remoto hay cambios que no tenemos en el repositorio local, nos dará un error y tendremos que hacer primero un “Pull”, para poner nuestro repositorio local al día y poder después hacer el “Push” desde el repositorio local contra el repositorio remoto.



Git. Repositorios remotos. Ciclo de vida

