



**Mondragon
Unibertsitatea**

Faculty of
Engineering

Introducción a Git/GitHub

David Romero

dromero@mondragon.edu

Índice

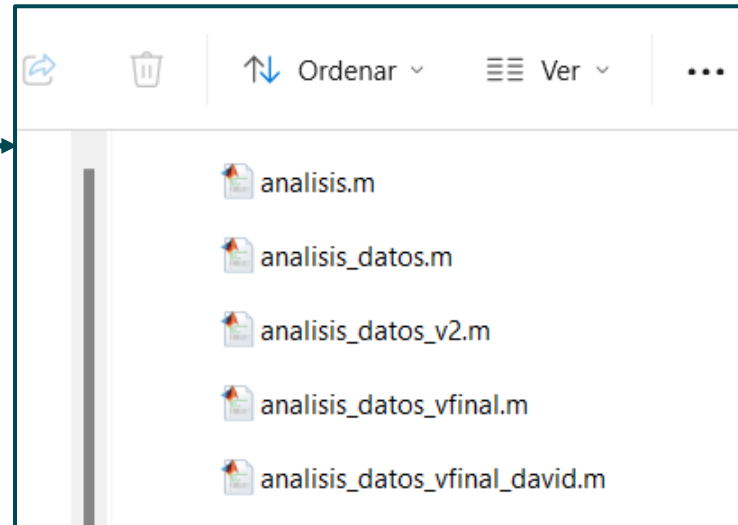
- Introducción: Git y GitHub
- Tutorial 1: uso individual
- Tutorial 2: entornos colaborativos
- PBL
- Comentarios finales

The background features two overlapping organic shapes. A large teal shape occupies the upper and right portions of the frame, while a smaller, darker teal shape is positioned on the left, partially overlapping the teal one. The word 'Introducción' is centered within the teal area.

Introducción

Flujo de trabajo habitual...

Escenario común



Problemas

- Duplicidades.
- No hay un histórico real del proyecto.
- Difícil entender el código.

Git

Qué es: Un software de **control de versiones**

Para qué se usa

- Generar un histórico del proyecto (versiones).
- Poder retroceder a un punto del proyecto “Fallback”.
- Generar ramas (desarrollo en paralelo).

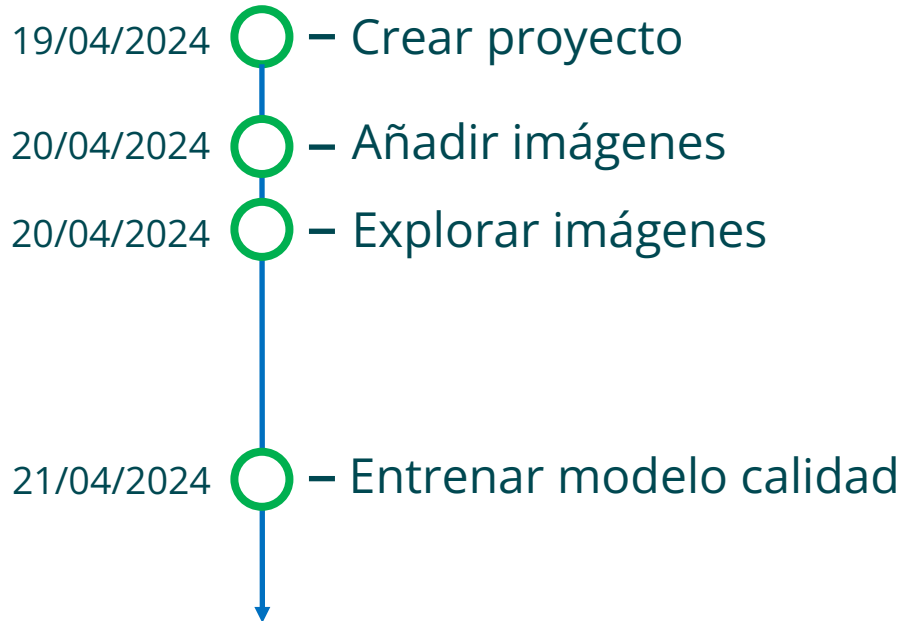


Cómo funciona

- Git constantemente monitoriza cambios en los ficheros.
- Nosotros decidimos cuándo guardar cambios en la historia del proyecto.
- Basado en comandos.
- La historia se guarda en una carpeta oculta llamada *.git*

Cómo funciona Git

Historia del proyecto



- **Commit:** punto de guardado en la historia del proyecto.
 - Identifica autor, fecha y ficheros modificados.
 - Tiene un mensaje asociado.
 - Código hash único.

Otras circunstancias

¿Cómo generar una copia de seguridad online?

¿Cómo sincronizar código entre ordenadores?

¿Cómo colaborar con otras personas en un proyecto?

¿Cómo compartir mi proyecto con otros?

Git por sí solo no da respuesta a esto

GitHub

Qué es: página web para alojar proyectos Git.


Para qué se usa

- Guardar copia de seguridad online.
- Sincronizar código entre equipos.
- Compartir tus proyectos (portfolio).
- Descargar código de proyectos públicos.



Cómo funciona

- GitHub mantiene una copia online de nuestro proyecto.
- Nosotros decidimos cuándo sincronizar (no es automático como con Dropbox/Drive).
- Incluye funcionalidades para colaboración.

The background features two overlapping abstract shapes. A large teal shape with a rounded right edge occupies the upper and middle portions of the frame. Overlapping its left side is a smaller, darker teal shape, also with rounded edges. The text is positioned within the teal area.

Tutorial 1

Git/GitHub para uso individual

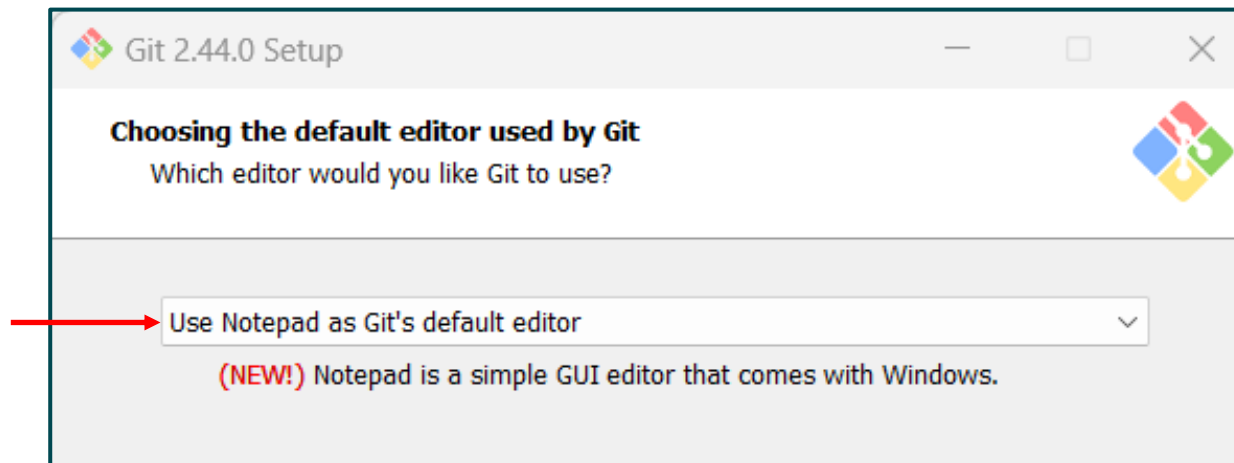
1. Descargar Git

- Ir a <https://git-scm.com/download/win>
- Descargar el instalador (Git-2.44.0-64-bit.exe)



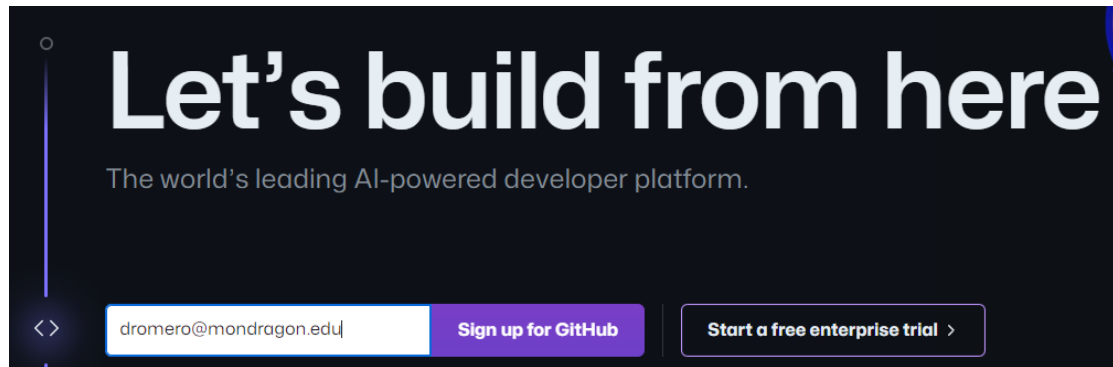
2. Instalar Git

- Ejecutar instalador
- Aceptar las opciones por defecto salvo el editor de texto que usará Git (cambiar a Notepad):



3. Crear cuenta en GitHub

- Ir a: <https://github.com>
- Introducir email y click en *Sign up for GitHub*

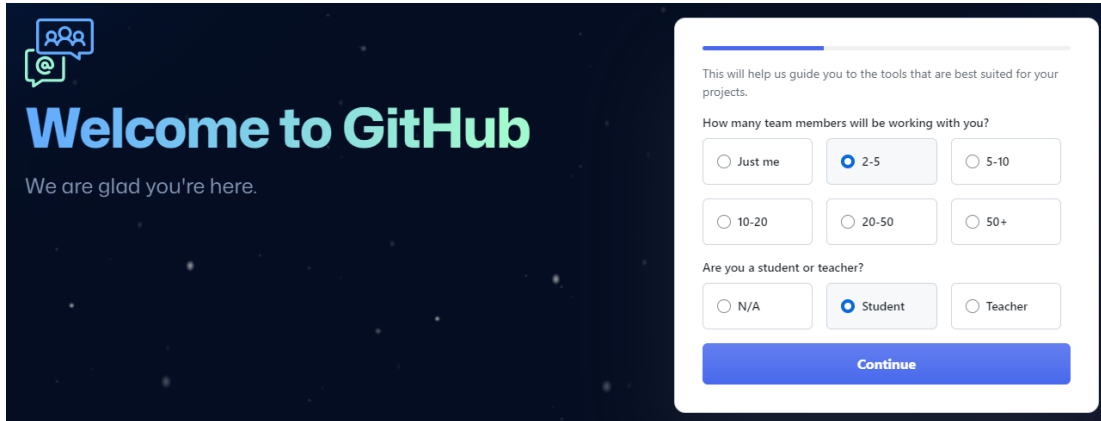


- Crear contraseña y nombre de usuario

A screenshot of the GitHub sign-up form. It has a dark blue background with white text. The form is divided into sections: "Enter your email*", "Create a password*", "Enter a username*", and "Email preferences". The email field contains "dromero@mondragon.edu" with a green checkmark. The password field contains a series of dots with a green checkmark. The username field contains "davidmgep" with a green checkmark. At the bottom, there is a checkbox for "Receive occasional product updates and announcements." and a green "Continue" button.

3. Crear cuenta en GitHub

- Seleccionar la opción estudiante y 2-5 miembros



The screenshot shows the GitHub sign-up interface. On the left, a dark blue banner with the text "Welcome to GitHub" and "We are glad you're here." is displayed. On the right, a white form with a blue border contains the following elements:

- A progress bar at the top.
- A heading: "This will help us guide you to the tools that are best suited for your projects."
- A question: "How many team members will be working with you?"
- Radio button options for team size: "Just me", "2-5" (selected), "5-10", "10-20", "20-50", and "50+".
- A question: "Are you a student or teacher?"
- Radio button options for role: "N/A", "Student" (selected), and "Teacher".
- A blue "Continue" button at the bottom.

- Seleccionar la versión gratuita

4. Crear repositorio en GitHub

Click en *Create repository*

- Escoger nombre
- Privado (por ahora)
- Añadir README

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repository

Import 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 (*).

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner *

drombas ▾

Repository name *

tutorial

✔ tutorial is available.

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

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

.gitignore template: None ▾

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

Choose a license

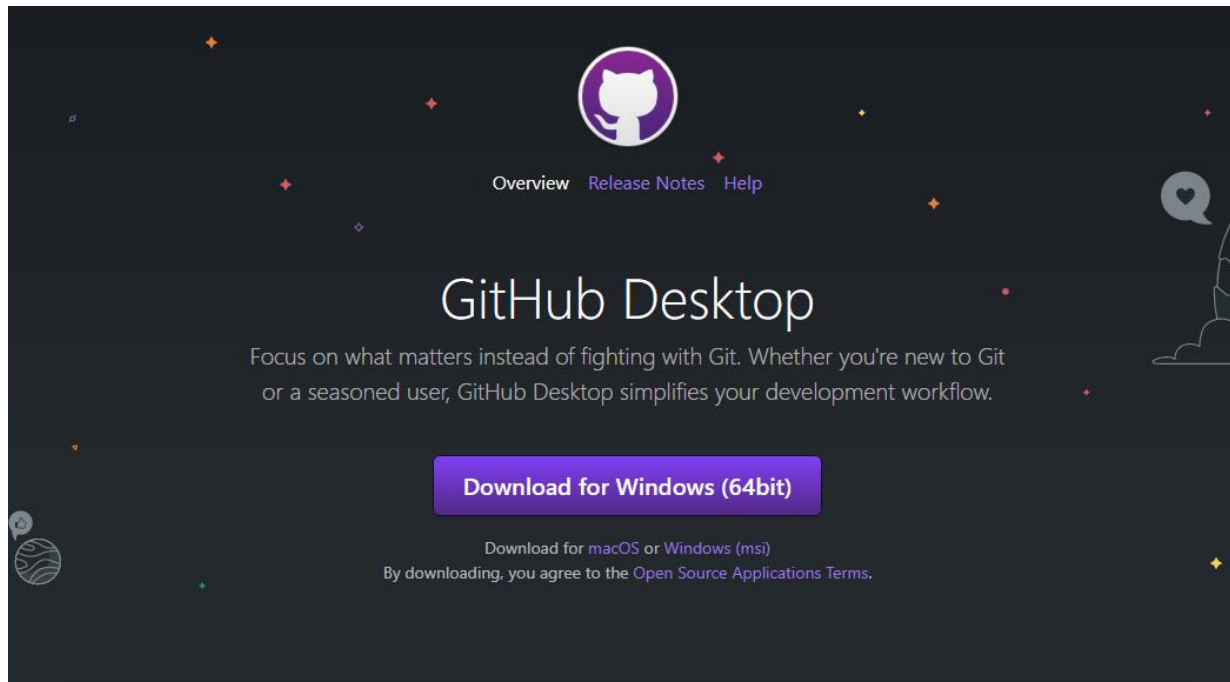
License: None ▾

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

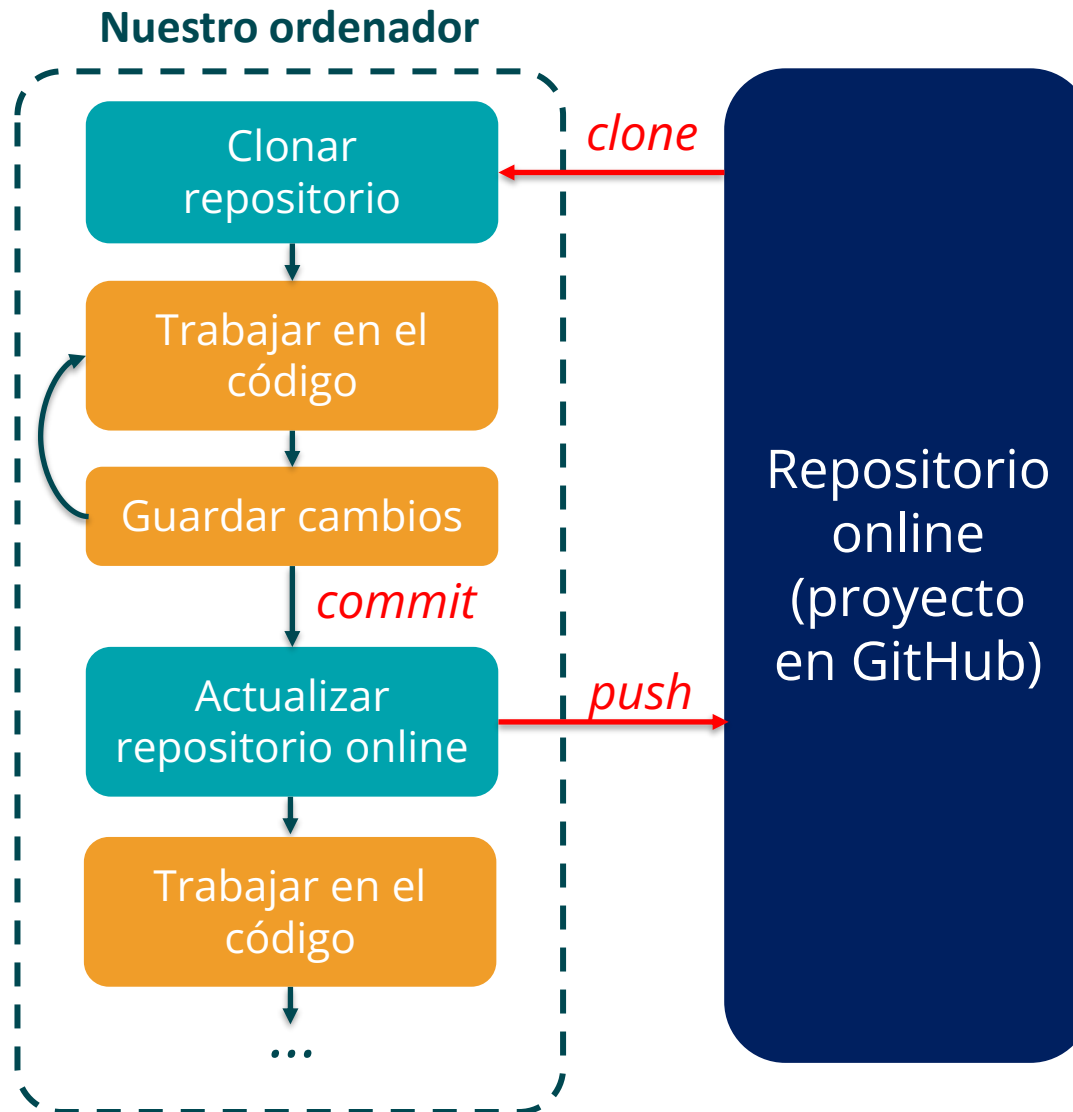
This will set `main` as the default branch. Change the default name in your [settings](#).

5. Instalar GitHub Desktop

- GitHub Desktop es el programa que usaremos para interactuar con Git
- Ir a <https://desktop.github.com>
- Descargar el instalador (*GitHubDesktopSetup-x64.exe*) e instalar el programa

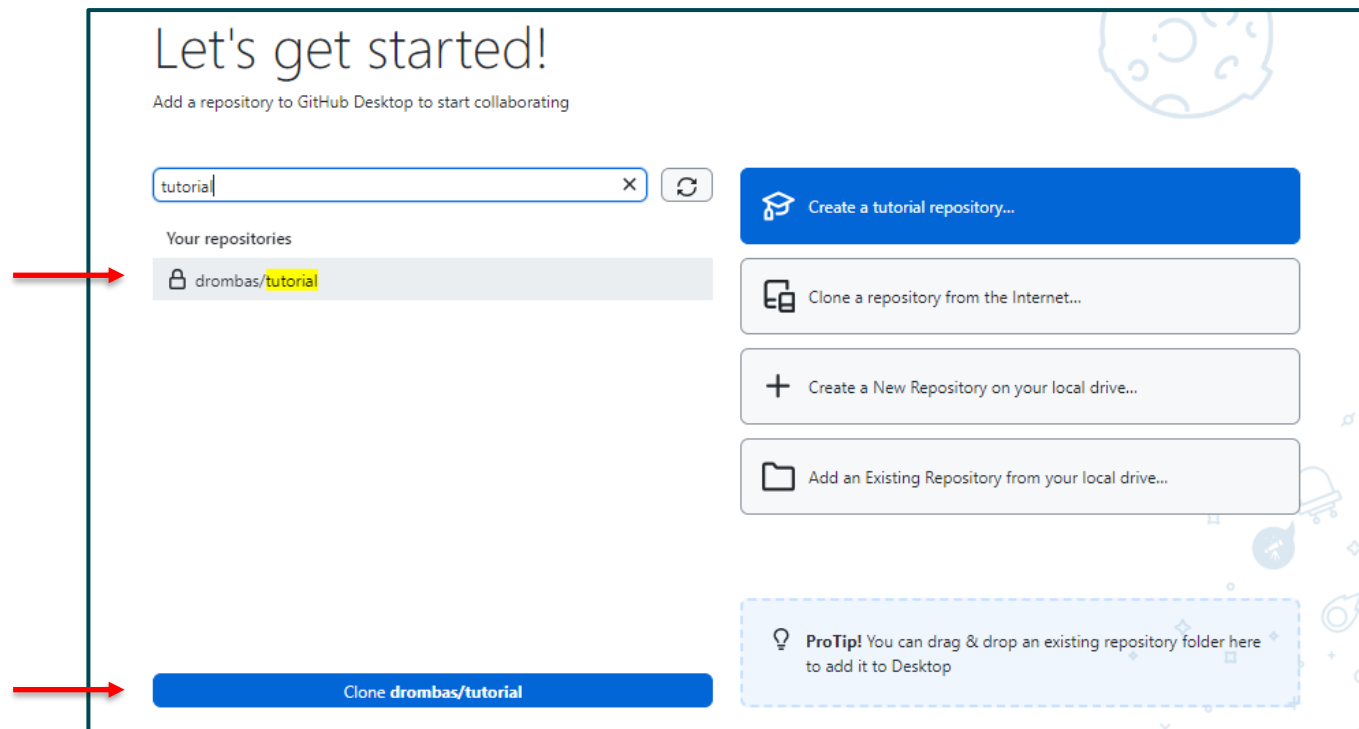


Flujo de trabajo: 1 persona 1 PC



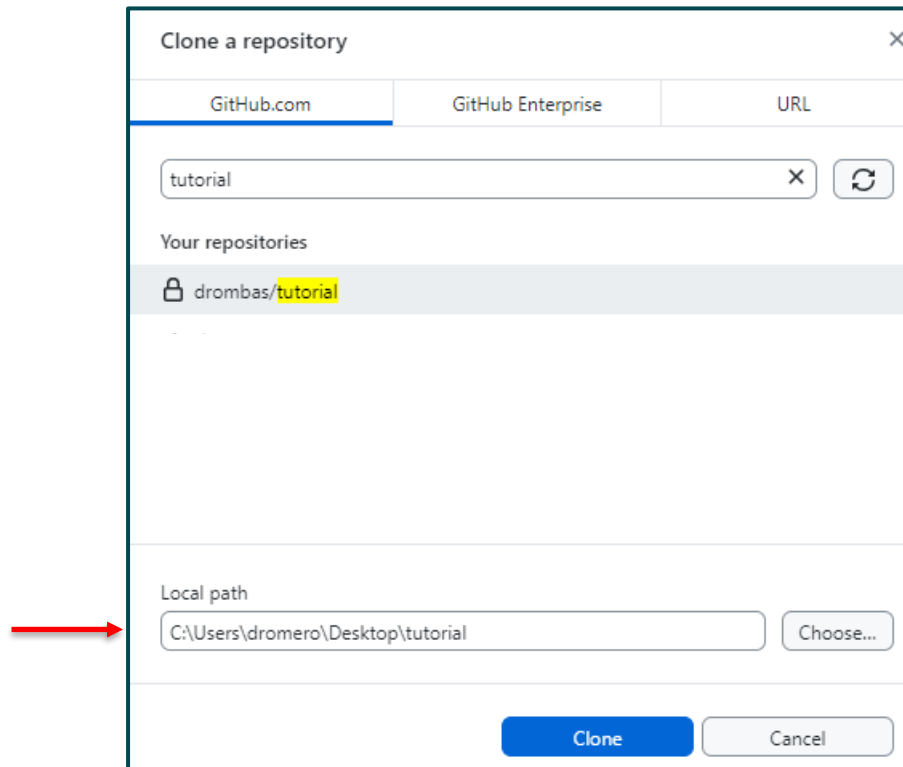
6. Clonar repositorio (crear copia local)

- Abrir GitHub Desktop
- Logearse (en File-Options)
- Clonar repositorio: buscar por nombre



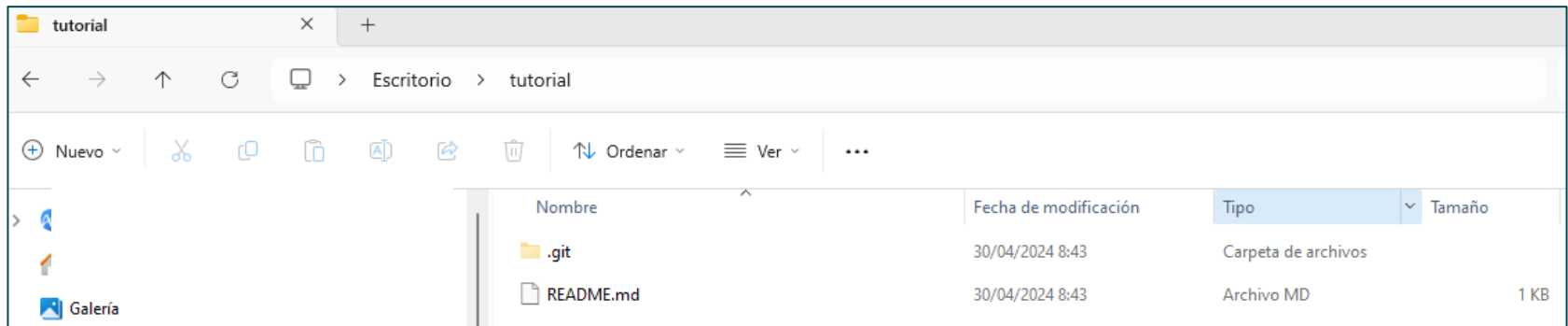
6. Clonar repositorio

- Escoger dónde crear la copia local del proyecto
- Ha de estar fuera de Dropbox/Drive/One-Drive



6. Clonar repositorio

- El proyecto es simplemente una carpeta en el ordenador



- Con el proyecto listo podemos comenzar a trabajar
- Usaremos GitHub Desktop para guardar cambios y sincronizar con GitHub.

7. Guardar cambios

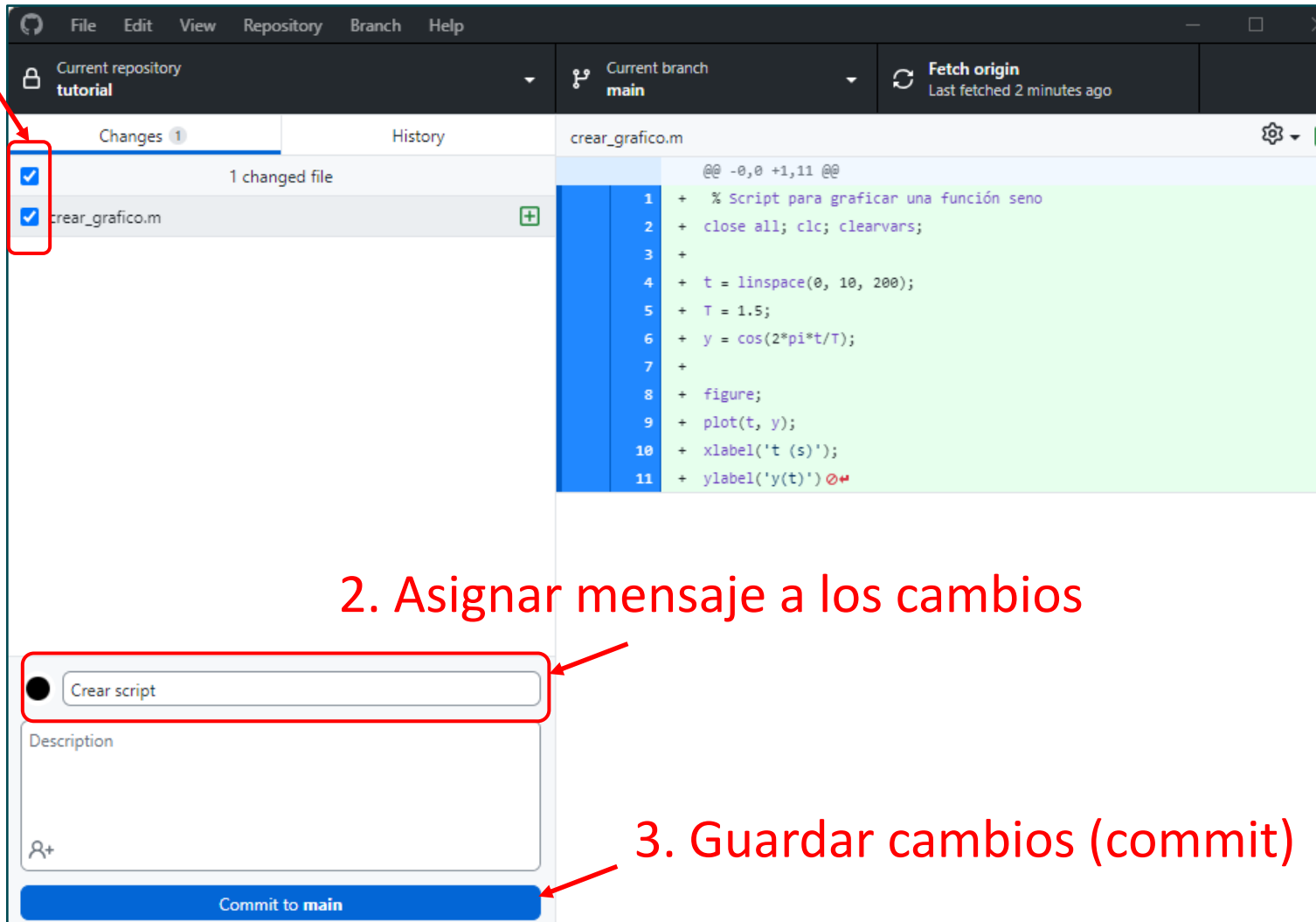
Ejemplo: añadimos un script de MATLAB

Acciones a realizar GitHub Desktop

- Seleccionar cambios
- Guardar cambios con mensaje (commit)
- Actualizar repositorio (push)

7. Guardar cambios (commit)

1. Seleccionar cambios a guardar (ticks en ficheros)

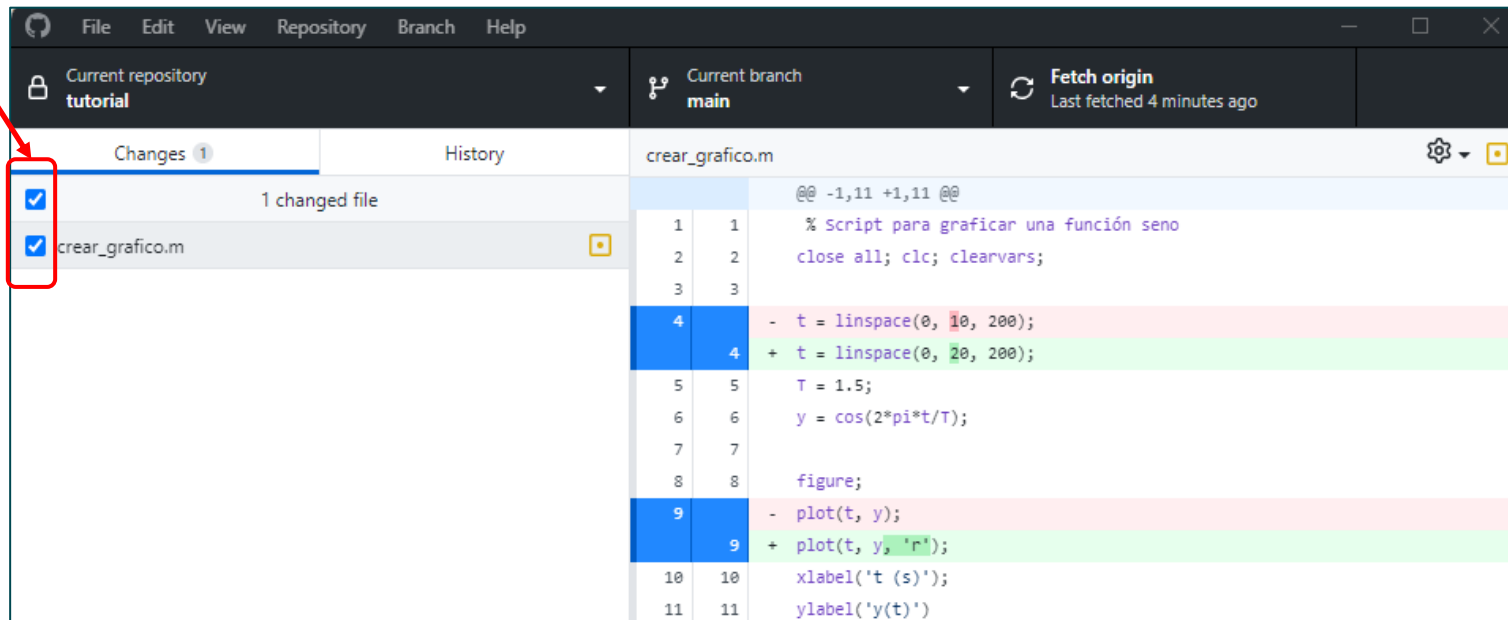


2. Asignar mensaje a los cambios

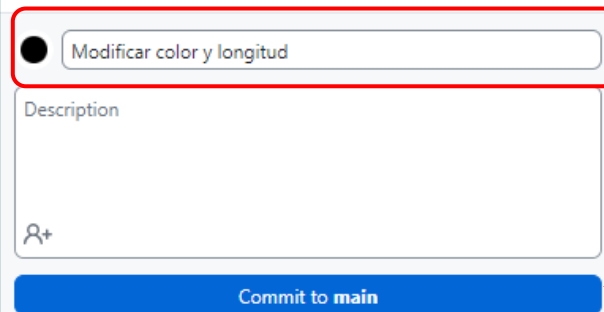
3. Guardar cambios (commit)

7. Guardar cambios (commit)

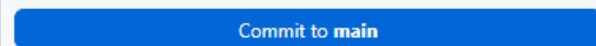
1. Seleccionar cambios a guardar (ticks en ficheros)



2. Asignar mensaje a los cambios

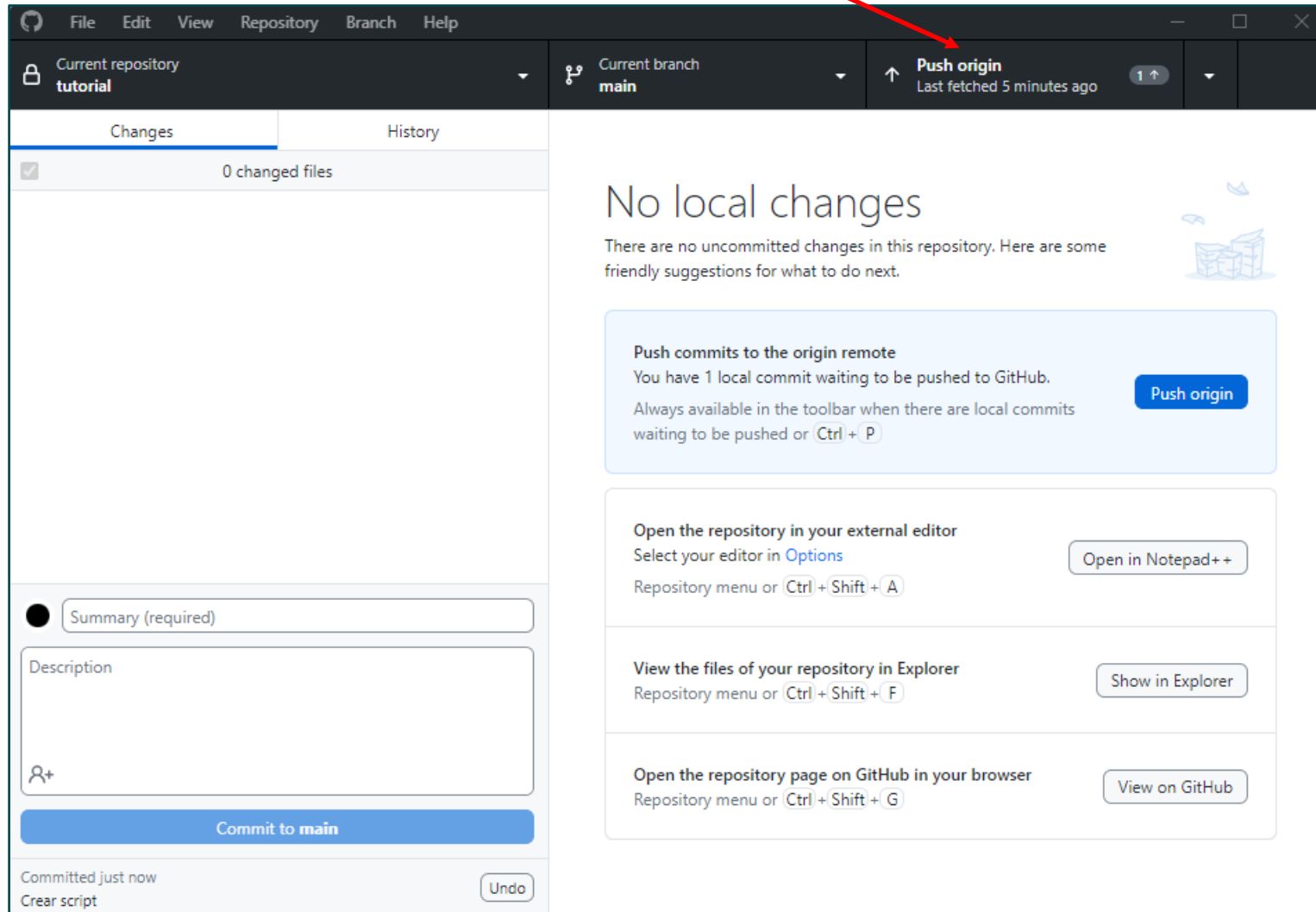


3. Guardar cambios (commit)

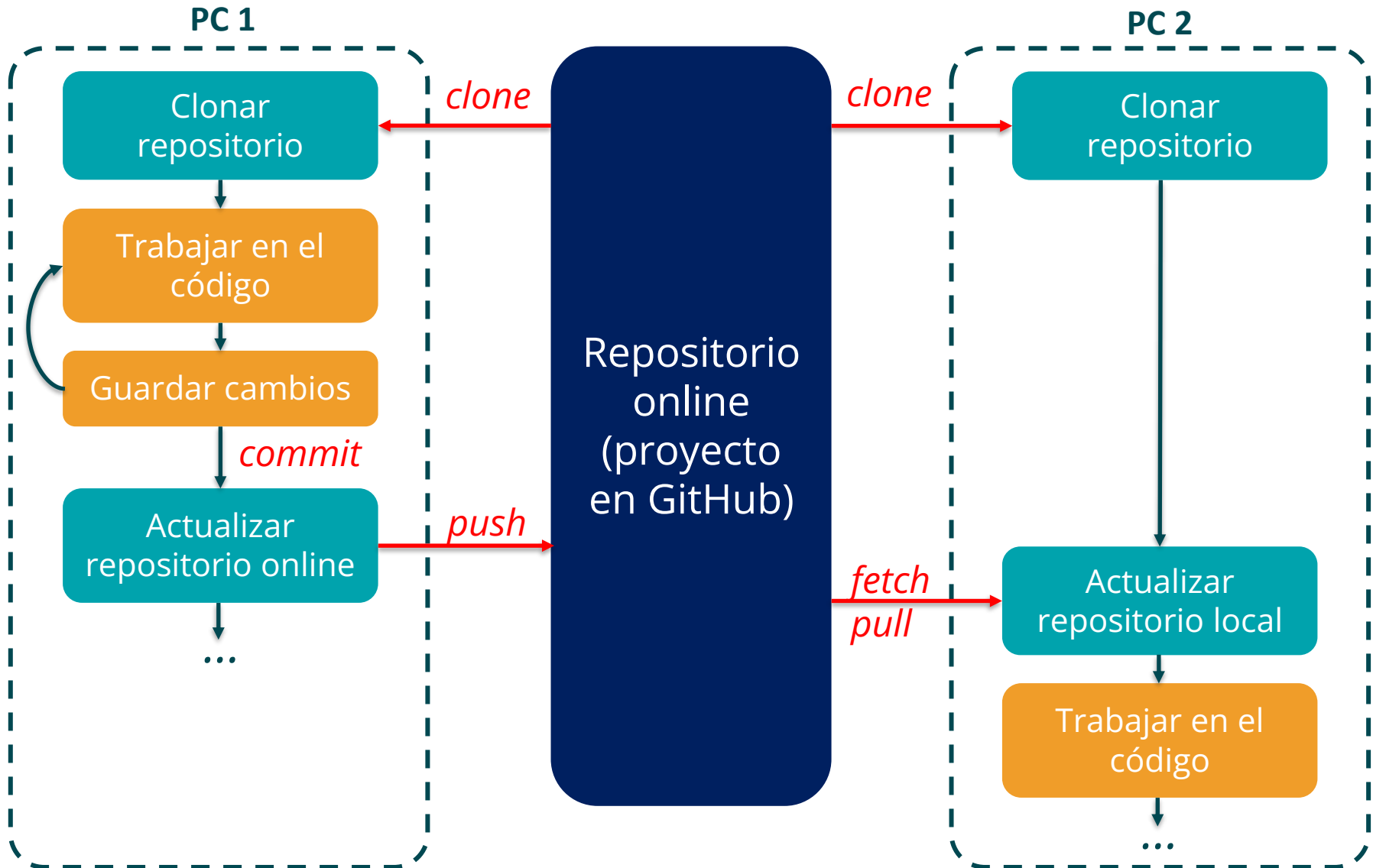


8. Actualizar repositorio online (push)

Click en la opción Push origin



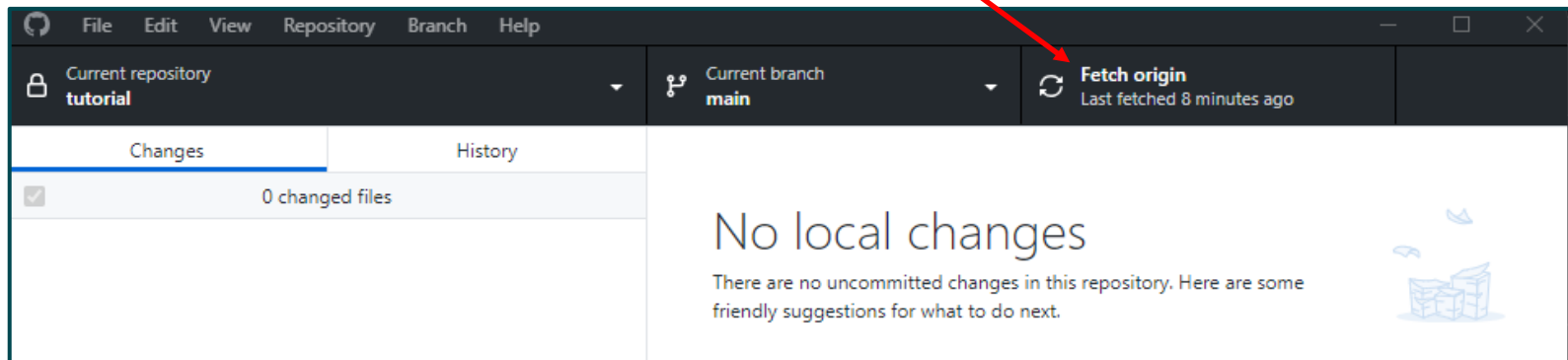
Flujo de trabajo: 1 persona >1 PC



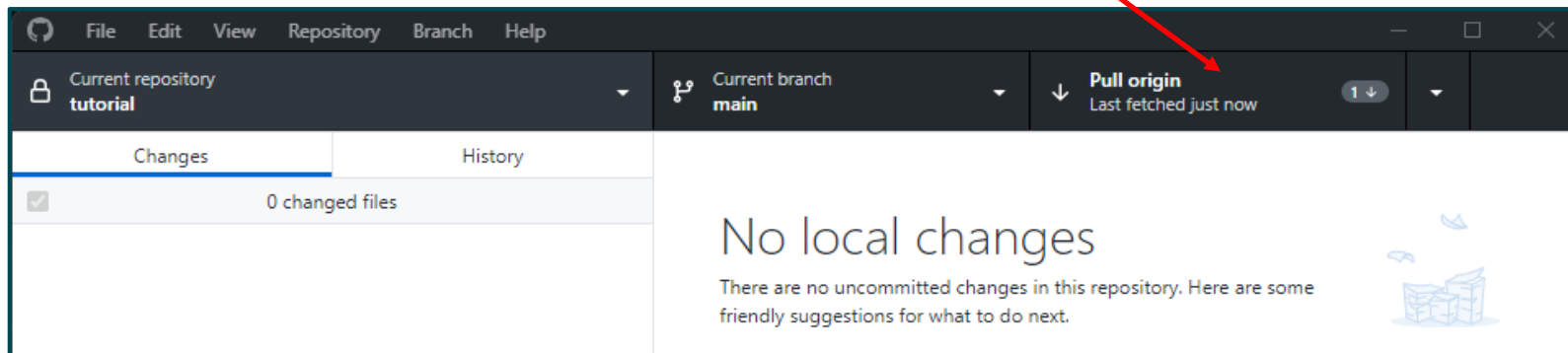
9. Actualizar proyecto local (fetch/pull)


2 pasos

1. Fetch (descargar cambios)



2. Pull (incorporar cambios)



The background features two overlapping abstract shapes. A large teal shape with a rounded right edge occupies the upper and middle portions of the frame. Overlapping its left side is a smaller, darker teal shape, also with rounded edges. The text is positioned within the teal area.

Tutorial 2:

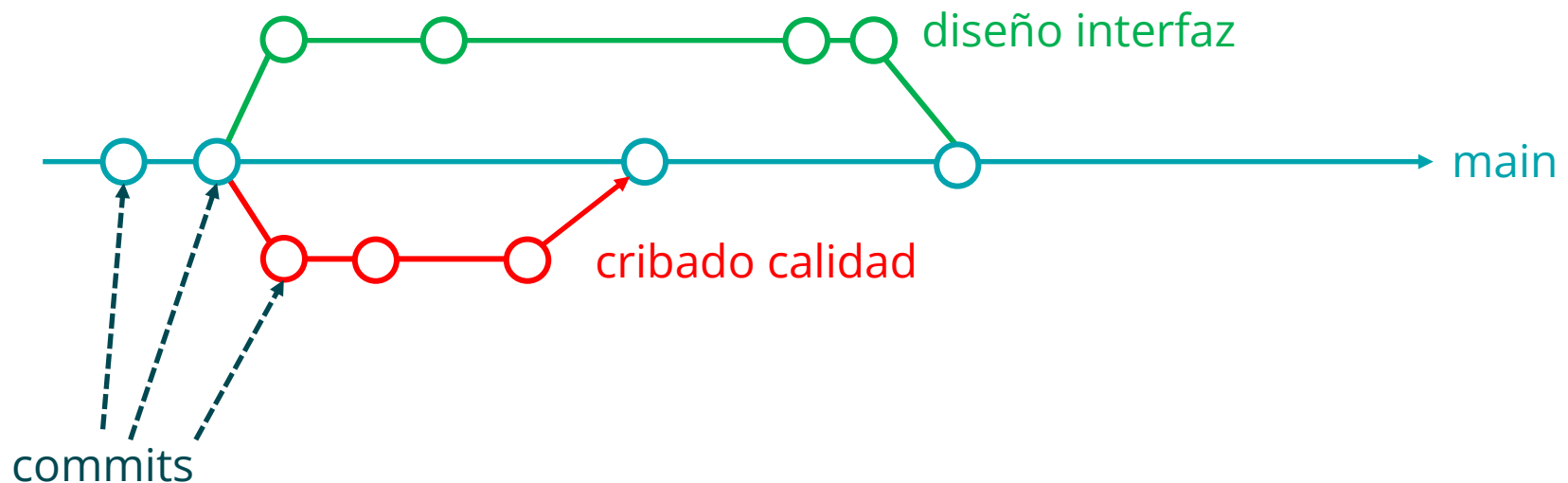
Git/GitHub en
proyectos colaborativos

Problemática

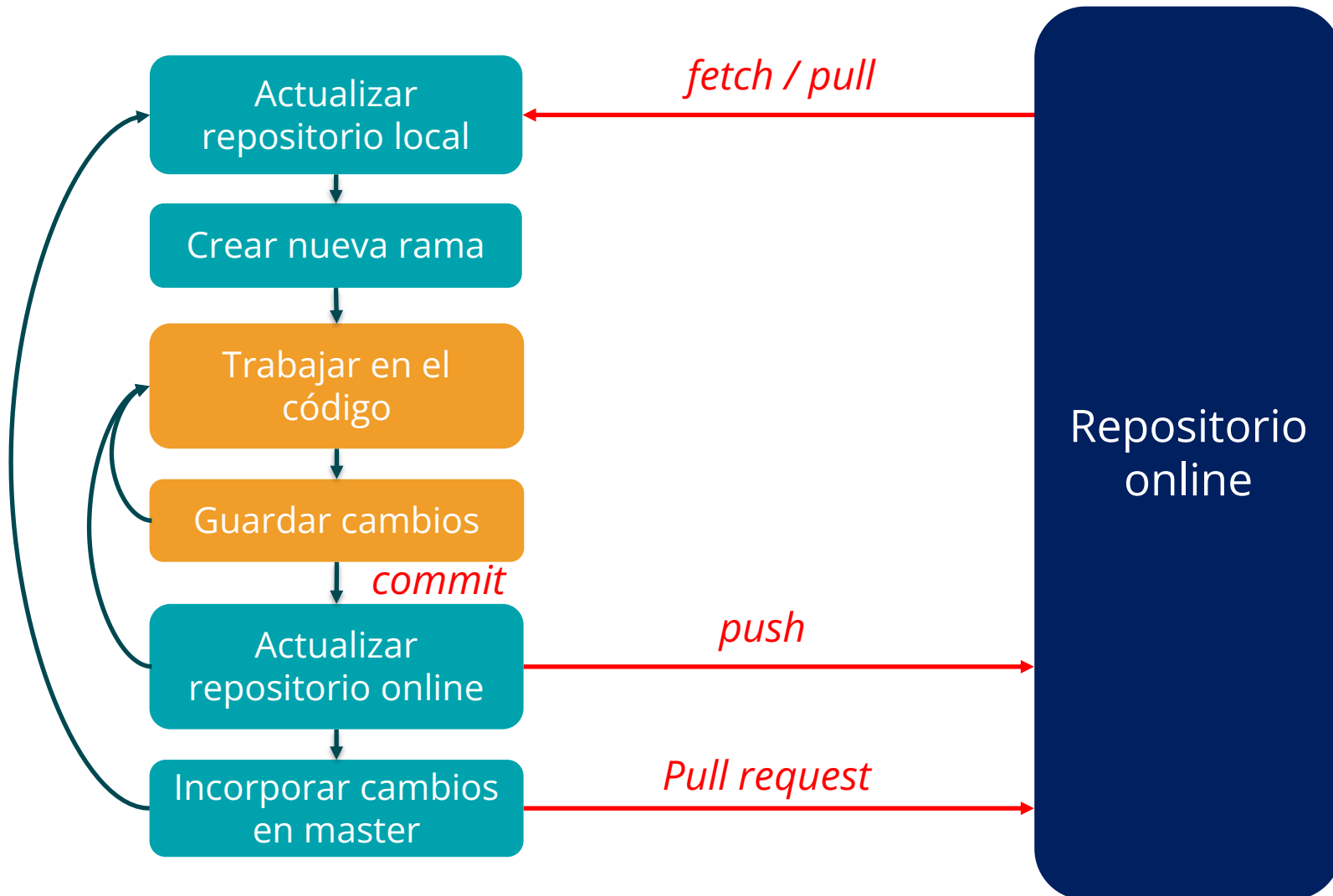
Problema: Múltiples usuarios → posibles conflictos.

Solución

- Dividir el proyecto en ramas (trabajar en paralelo).
- Nunca trabajar directamente en la rama principal (main).
- Trabajar en rama de forma individual e integrar después en main.



Flujo de trabajo: varias personas



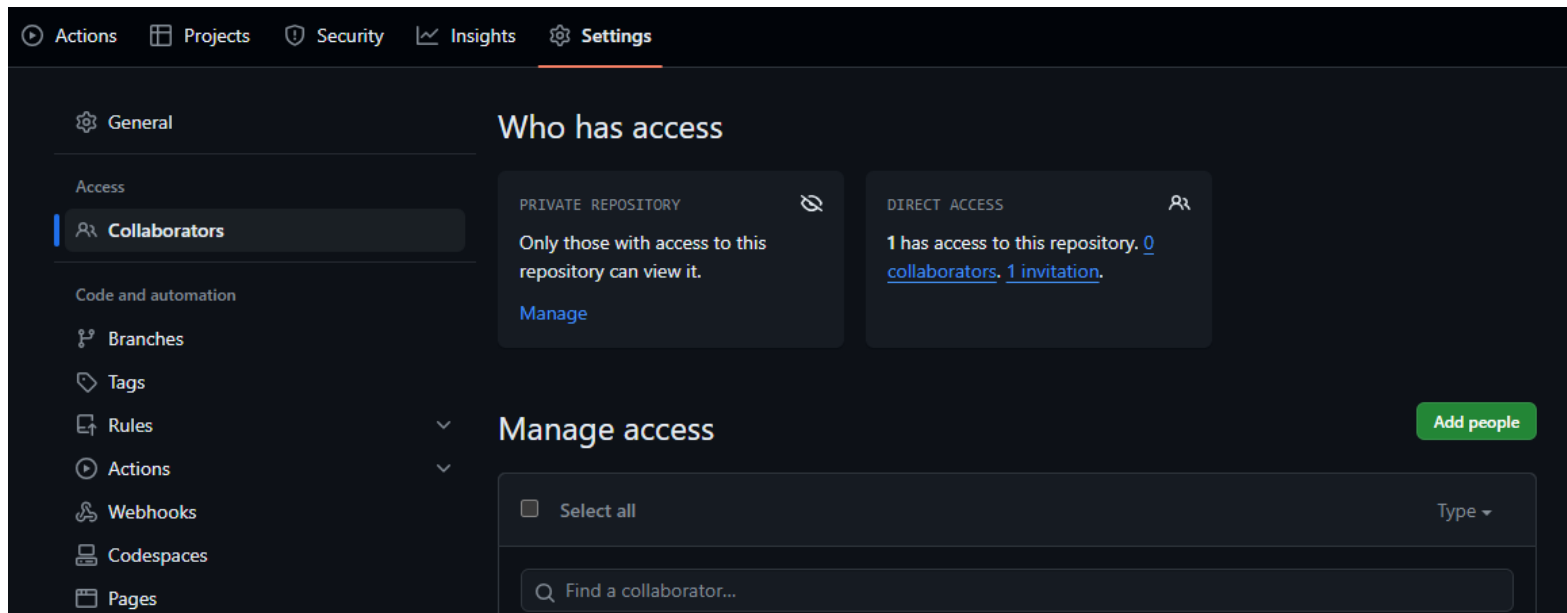
Flujo de trabajo: varias personas

Pasos a seguir

- Actualizar proyecto local (fetch + pull)
- Crear nueva rama (p.ej., drb-cribado-calidad)
- Trabajar en la rama (commit/push)
- Integrar rama en main (**Pull-Request**)
- Eliminar rama (local y repositorio)
- Actualizar rama main en local

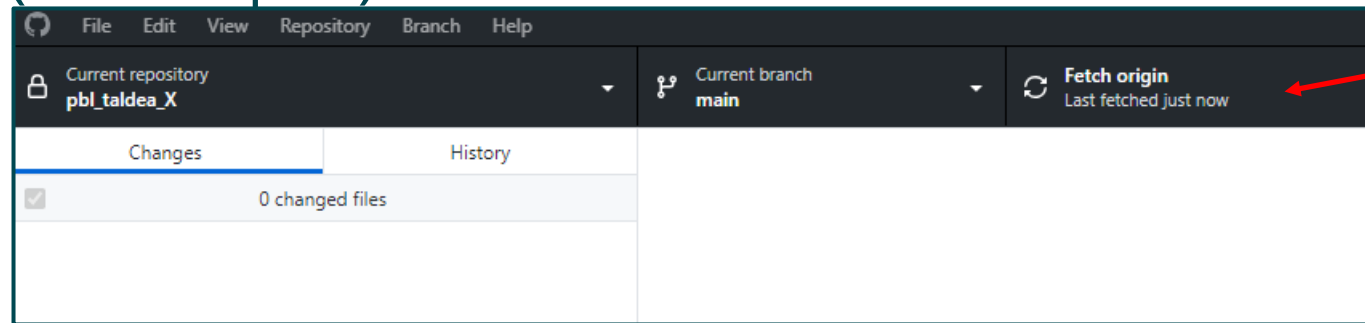
1. Puesta en marcha del proyecto

- Crear repositorio **pbl_grupo_X** (un miembro del equipo).
- Añadir como colaboradores al resto del equipo (buscar por nombre de usuario).
- Cada miembro clona el repositorio de forma local.

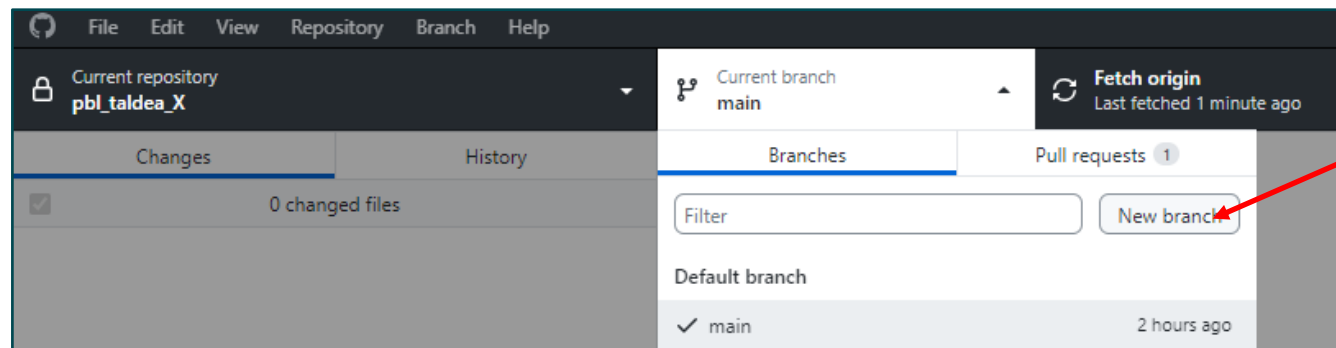


2. Crear nueva rama

- Siempre actualizar previamente la rama main local (fetch + pull)

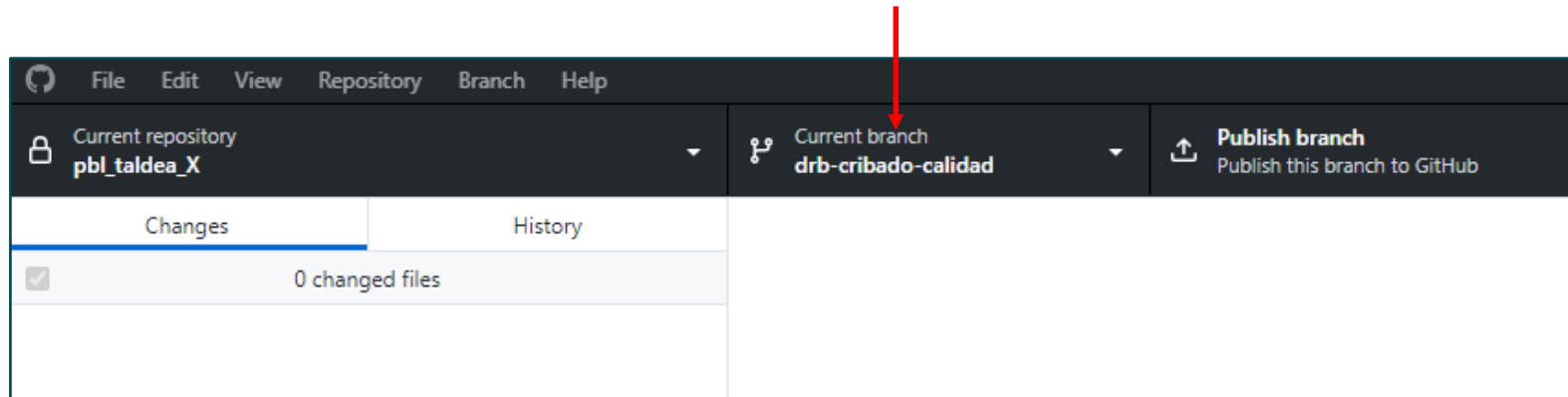


- Crear nueva rama (p.ej., drb-cribado-calidad)



3. Trabajo individual en la nueva rama

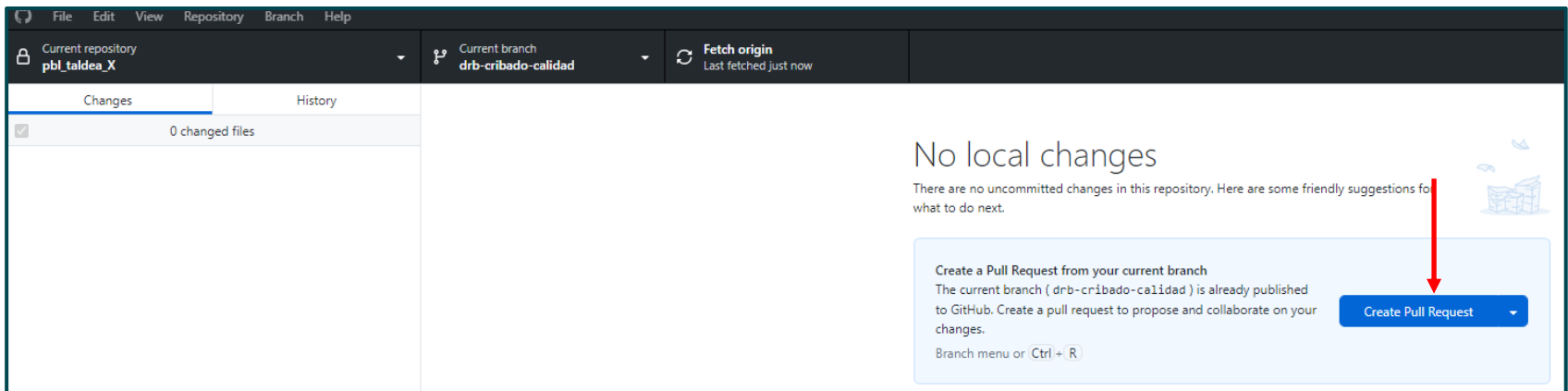
Rama en la que estamos trabajando



- Trabajamos en nuestra rama siguiendo los pasos habituales
 - Guardar cambios (commit).
 - Actualizar rama online (Publish Branch / Push).

4. Integrar rama en repositorio

- Cuando hemos finalizado la tarea debemos integrar los cambios en la rama main del repositorio en GitHub
- Para ello realizamos una solicitud de integración llamada **Pull-Request (PR)**.



4. Integrar rama en repositorio

- En GitHub describimos el contenido de la PR
- Revisamos los cambios y validamos que no hay conflictos

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). [Learn more about diff comparisons here](#).

base: main ← compare: drb-cribado-calidad ✓ **Able to merge.** These branches can be automatically merged.

Add a title

Añadir creación del modelo de calidad

Add a description

Write Preview H B I ≡ <> @ ↺ ↻

Esta PR añade el código usado para entrenar un modelo para el cribado de calidad.

Markdown is supported Paste, drop, or click to add files

Reviewers No reviews

Assignees No one—[assign yourself](#)

Labels None yet

Projects None yet

Milestone No milestone

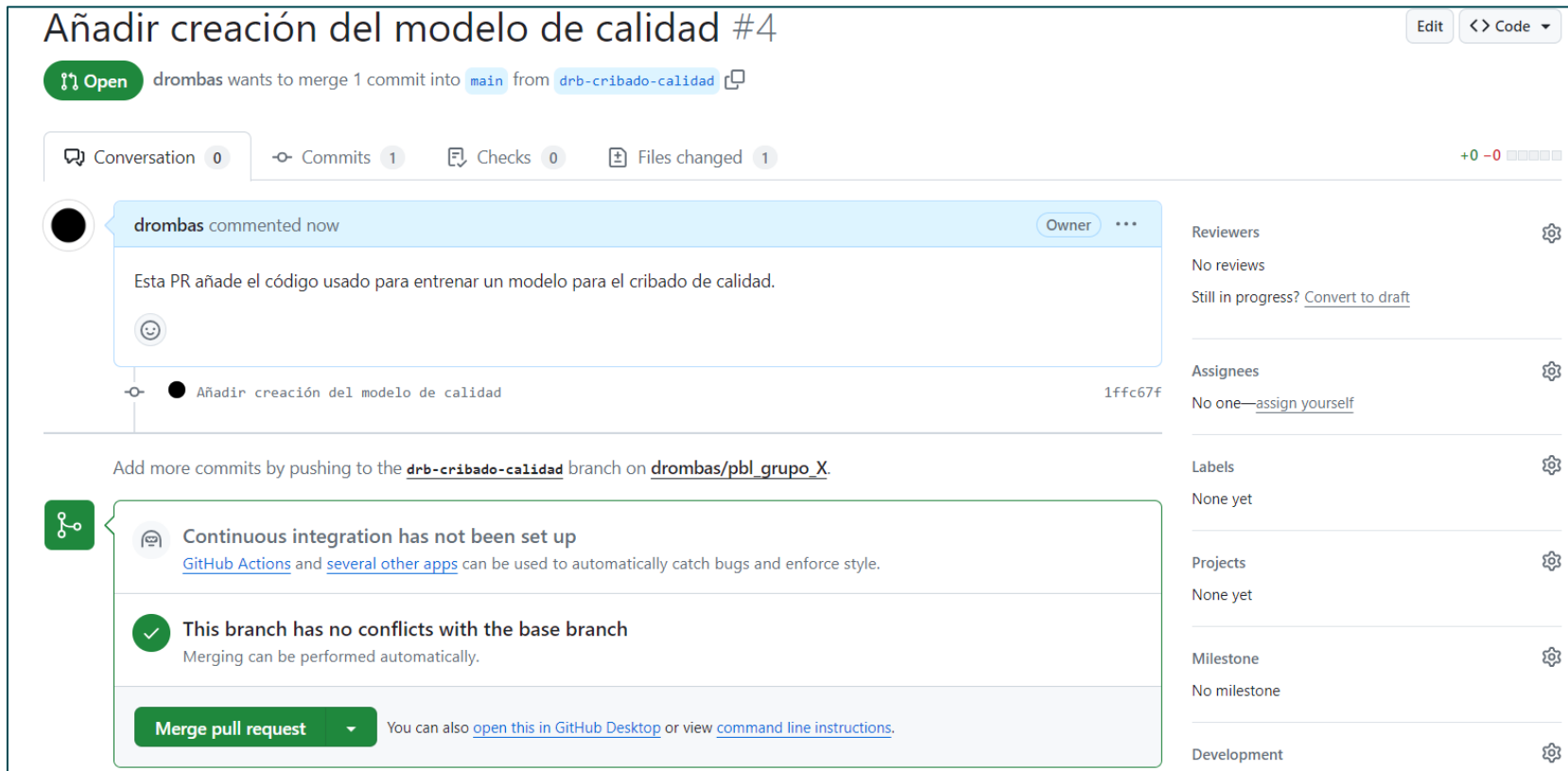
Development Use [Closing keywords](#) in the description to automatically close issues

Helpful resources

Create pull request

4. Integrar rama en repositorio

- Otros pueden comentar y revisar los cambios
- Aceptamos la integración (Merge pull request)



The screenshot shows a GitHub Pull Request (PR) titled "Añadir creación del modelo de calidad #4". The PR is open, showing a conversation with a comment from "drombas" stating: "Esta PR añade el código usado para entrenar un modelo para el cribado de calidad." Below the comment is a commit titled "Añadir creación del modelo de calidad" with hash "1ffc67f". The PR status bar indicates "Continuous integration has not been set up" and "This branch has no conflicts with the base branch". A green "Merge pull request" button is visible at the bottom. The right sidebar shows settings for Reviewers, Assignees, Labels, Projects, Milestone, and Development.

Añadir creación del modelo de calidad #4

Open drombas wants to merge 1 commit into `main` from `drb-cribado-calidad`

Conversation 0 Commits 1 Checks 0 Files changed 1 +0 -0

drombas commented now

Esta PR añade el código usado para entrenar un modelo para el cribado de calidad.

Owner

• Añadir creación del modelo de calidad 1ffc67f

Add more commits by pushing to the `drb-cribado-calidad` branch on [drombas/pbl_grupo_X](#).

Continuous integration has not been set up
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Reviewers
No reviews
Still in progress? [Convert to draft](#)

Assignees
No one—[assign yourself](#)

Labels
None yet

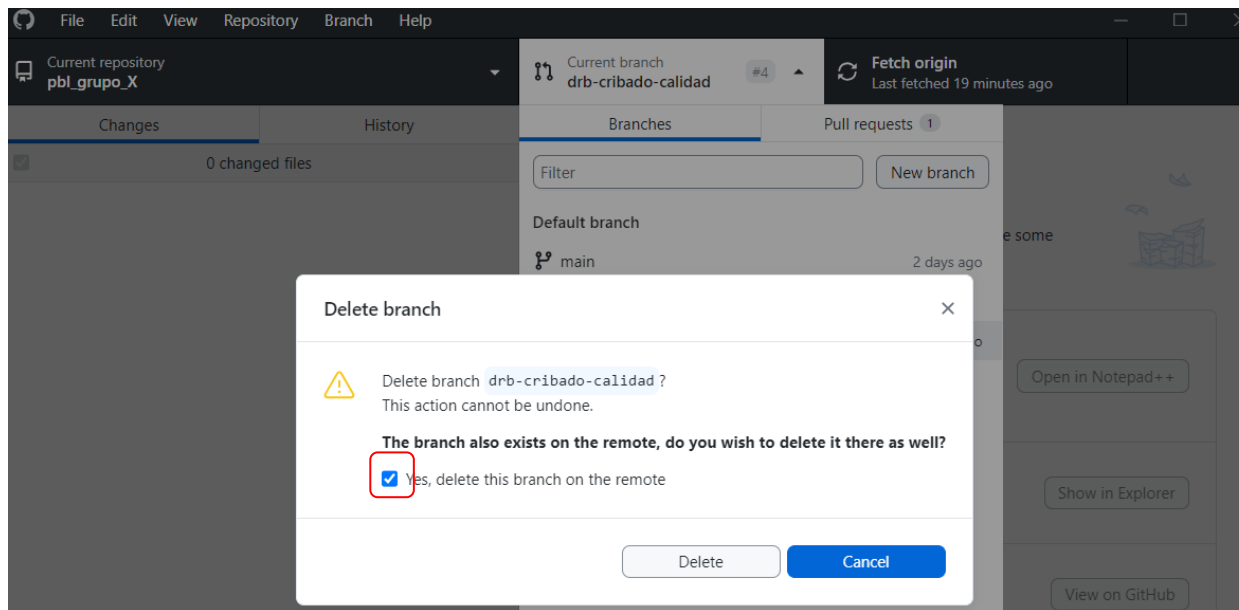
Projects
None yet

Milestone
No milestone

Development

5. Eliminar rama ya integrada

- Para evitar conflictos eliminamos la rama que ha sido integrada desde GitHub Desktop
 - Click derecho sobre la rama → remove



- Además, actualizamos la copia local de la rama main (Fetch / Pull)

The background features two overlapping organic shapes. A large, light teal shape occupies the upper and right portions of the frame. A smaller, dark teal shape overlaps its left side, extending towards the bottom left corner.

PBL

Qué se pide

Hito GitHub (24/05)

- Crear un repositorio GitHub por equipo.
- Validar que se está usando favorablemente.
- Demostrar que todos los miembros han contribuido.

Entrega final (parte software)

- Proyecto en GitHub con código.
- Despliegue web en dominio GitHub.

Estructura del proyecto

The screenshot displays the GitHub interface for the repository 'pbl_grupo_X'. At the top, it shows the repository name, a 'Public' badge, and interaction buttons like 'Pin', 'Unwatch', 'Fork', and 'Star'. Below this, the 'main' branch is selected, showing '2 Branches' and '0 Tags'. A search bar and 'Add file' button are present. The main content area lists the repository's structure:

File/Folder	Description	Time
data	add images	7 hours ago
docs	Añadir carpetas procesado	2 hours ago
interfaz	Crear interfaz	6 hours ago
procesado_datos	Añadir carpetas procesado	2 hours ago
procesado_imagen	Añadir carpetas procesado	2 hours ago
README.md	Update README.md	2 hours ago

Below the file list, the 'README' file is expanded, showing the text 'PBL grupo X'. On the right side, the 'About' section includes a link to 'drombas.github.io/pbl_grupo_X' (highlighted with a red box and labeled 'Link a página web'), 'Readme', 'Activity', '0 stars', '1 watching', and '0 forks'. The 'Contributors' section lists 'drombas David Romero-Bascones' and 'davidmgep'.

Estructura del proyecto

/data: datos empleados

- Imágenes, modelos, features...

/docs: código página-web / xml

/interfaz

- Código de la interfaz desarrollada
- Ejecutable standalone (.exe)

/procesado_datos

- Código empleado para procesar los datos

/procesado_imagen

- Código empleado para procesar las imágenes

Despliegue página web

GitHub ofrece la posibilidad de alojar una página-web

Pasos a seguir

- Configurar el repositorio GitHub como público
 - Settings → Cambiar visibilidad
- Añadir código página web a la carpeta */docs/*
- Configurar web
 - Settings → Pages
 - Deploy from branch (main) + */docs*

The background features two large, overlapping organic shapes. The larger shape is a vibrant teal color, and the smaller shape overlapping its left side is a darker teal. The word 'Conclusiones' is centered within the teal area.

Conclusiones

Resumen

Git / GitHub

- Una herramienta imprescindible a nivel profesional.
- Posibilita la sincronización, backup y colaboración.
- Buena forma de construir un portfolio de proyectos.

Aprendizaje

- Algo difícil al comienzo.
- El aprendizaje requiere de práctica diaria.
- GitHub Desktop es un buen comienzo para iniciarse, pero un uso avanzado de Git/GitHub requiere de comandos.
- Recursos
 - [GitHub Student Developer Pack](#)
 - [Pro Git Book](#)

Otros aspectos

Conflictos

- Cuando dos ramas han modificado el mismo fichero.
- Es necesario resolverlos cuidadosamente y decidir qué cambios integrar.

Trabajo con ficheros grandes (datos, imágenes)

- GitHub no está diseñado para alojar bases de datos.
- Pueden excluirse algunas carpetas con el fichero .gitignore

Proyectos open-source

- Mediante GitHub es posible colaborar con proyectos de código abierto de forma altruista.
- El flujo de trabajo es algo distinto (fork / pull-request).

Git en base a comandos

Git Bash



```

MINGW64:/c/Users/dromero/Desktop/pbl_ejemplo

dromero@MLANIESEKOp18 MINGW64 ~/Desktop/pbl_ejemplo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   procesado_imagen/crear_modelo_calidad.m

no changes added to commit (use "git add" and/or "git commit -a")

dromero@MLANIESEKOp18 MINGW64 ~/Desktop/pbl_ejemplo (main)
$ git add -A

dromero@MLANIESEKOp18 MINGW64 ~/Desktop/pbl_ejemplo (main)
$ git commit -m "modificar modelo calidad"
[main 625c214] modificar modelo calidad
 1 file changed, 4 insertions(+)

dromero@MLANIESEKOp18 MINGW64 ~/Desktop/pbl_ejemplo (main)
$ git push

```

Comando	Para qué sirve
clone <url del repositorio>	Crear copia local del proyecto desde un repositorio online
add <ficheros/carpetas> add -A	Seleccionar ficheros antes de commit Seleccionar todos los ficheros antes de commit
commit -m "mensaje"	Asignar mensaje
push	Actualizar repositorio online con cambios en local
pull	Actualizar proyecto local con cambios online
status	Visualizar ficheros con cambios
log	Mostrar histórico del proyecto



**Mondragon
Unibertsitatea**

Faculty of
Engineering

David Romero-Bascones

Mondragon Unibertsitatea

dromero@mondragon.edu