

Control de versiones con Git y GitHub

3.3 Colaboración con Pull Request



Autor: Lisa ERIKSEN

Fecha: 2025 / 2026

1. Cómo trabajar entre dos en un proyecto GitHub

1.1 Quién crea el repositorio

Una sola persona empieza:

1. Va a GitHub → **New repository**.
2. Le pone un nombre (ejemplo: *Git_GTA*).
3. Marca "**Public**".
4. Crea el repo.

👉 Esa persona se convierte en la **propietaria** del repositorio.

1.2 Cómo compartir el repo con otra persona

La persona propietaria hace:

1. Va al repo en GitHub.
2. Clic en **Settings**.
3. En el menú lateral → **Collaborators**.
4. "**Add people**".
5. Añade el nombre o correo GitHub del compañero.

El compañero recibirá una invitación → **Accept**.

👉 A partir de ese momento, los dos pueden trabajar y hacer push (pero no en la misma rama, porque... desastre asegurado).

1.3 Cada uno trabaja en su propia rama

Cada alumno:

1. Crea un clone del repositorio *Git_GTA* : `git clone <URL-del-repo>`
2. Crea su propia rama: `git branch ...`
3. Realiza sus cambios, puedes elegir uno de los ejemplos siguientes:
 - ! Para añadir la función, crea una nueva clase en un archivo distinto (por ejemplo, añadir una función en *GestionDinero.java*) y utilizarla desde el programa principal.

Idea 1 — Añadir el método *gastarDinero()*

Permite que el personaje pueda gastar parte de su dinero.

Ejemplo de resultado: Trevor gasta 100\$ en ropa nueva.

Idea 2 — Añadir el método *subirNivel()*

Permite que el personaje pueda subir de nivel

Ejemplo de resultado: Trevor sube al nivel 2 !

Idea 3 — Añadir el método *cambiarNombre()*

Permite que el personaje cambie de nombre

Ejemplo de resultado: Trevor ahora se llama Mike.

Idea 4 — Añadir el método *randomMission()*

Crear una misión aleatoria;

Ejemplo de resultado: Trevor recibe una misión aleatoria: "Entregar paquetes."

👉 Cuidado, si las dos personas modifican las mismas líneas de un mismo archivo, Git no puede decidir qué versión es la correcta y se crea un **conflicto**.

1.4 Hacer cambios y subirlos (push)

Cada alumno:

1. Añade sus archivos (git add ...)
2. Realiza un commit de sus cambios (git commit ...)
3. Sube su rama a GitHub (git push ...)



2. Realizar un Pull Request

2.1 ¿Qué es Pull Request?

Un **Pull Request (PR)** es una solicitud para proponer y justificar la fusión de tu rama con otra. Permite que alguien revise tus cambios antes de integrarlos en la rama principal.

GitHub muestra:

- qué cambiaste,
- en qué archivos,
- si hay conflictos,
- y un botón para fusionar cuando todo esté bien.

2.2 Cómo crear un Pull Request?

1. Ve a GitHub → a tu rama.
2. Arriba aparece un botón amarillo/verde → **Compare & pull request**.
3. Asegúrate de que el Pull Request sea desde *tu rama* hacia la rama main. (Ejemplo : mi-rama → main)
4. Escribe un mensaje:
 - qué has cambiado,
 - por qué,
 - posibles dudas.
5. Clic en **Create pull request**.

👉 El PR se queda pendiente hasta que un compañero lo revise y lo fusione.

2.3 Cómo aceptar (fusionar) un Pull Request?

Las personas con permiso de escritura en el repositorio pueden aceptar un Pull Request, lo que permite fusionar los cambios de una rama en otra.

1. Ve a **GitHub** → la pestaña **"Pull requests"** del repositorio.
2. Haz clic en el PR que quieres revisar.
3. Mira los cambios en la pestaña **Files changed**.

4. Si todo está bien:
 - Clic en **Review changes** → **Approve** (opcional pero recomendado).
 - Luego clic en **Merge pull request**.
5. Confirma con **Confirm merge**.
6. (Opcional) Clic en **Delete branch** para borrar la rama si ya no la necesitas.

👉 Cuando se fusiona el Pull Request, los cambios de la rama secundaria pasan a formar parte de la rama principal (main).

When you are gonna merge two branches after long time

