

## Práctico 2: Git y GitHub

Lerda Fernando

- 1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas):

- ¿Qué es GitHub?:

GitHub es una plataforma online, un espacio de desarrollo colaborativo, que utiliza el sistema de control de versiones Git para alojar y gestionar proyectos de código

- ¿Cómo crear un repositorio en GitHub?

Hay que iniciar sesión en GitHub y luego seleccionar “New” para crear un nuevo repositorio

- ¿Cómo crear una rama en Git?

Mediante los siguientes comandos en la consola: `git branch` y el nombre de la rama que se desea crear.

- ¿Cómo cambiar a una rama en Git?

Con los comandos “`git checkout` y el nombre de la rama deseada”

- ¿Cómo fusionar ramas en Git?

Nos posicionamos en la rama que deseamos guardar los cambios y luego escribimos “`gi merge` y el nombre de la rama a unir”

- ¿Cómo crear un commit en Git?

Mediante los comandos `git commit -m` “mensaje descriptivo de los cambios”

- ¿Cómo enviar un commit a GitHub?

Con los comandos `git add .` y luego `git commit -a`

- ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión de tu proyecto almacenada en un servidor en línea. Como GitHub

- ¿Cómo agregar un repositorio remoto a Git?

Con los comandos `git add .`

- ¿Cómo empujar cambios a un repositorio remoto?

Hay que verificar el estado de los cambios con “`git status`”

Luego añadir los cambios con “git add .”

Por ultimo se confirman los cambios con un commit “git commit -m mas comentario”

- ¿Cómo tirar de cambios de un repositorio remoto?

Se debe hacer un git pull:

1. Verifica el estado de tu repositorio local “git status”
2. Descarga los últimos cambios del remoto “git pull origin mas nombre de rama”
3. Resuelve conflictos (si los hay) “git add archivo-con-conflicto”
4. Completa el merge: “git commit -m "Resuelto conflicto de merge"”

- ¿Qué es un fork de repositorio?

Un fork es una copia personal de un repositorio de otro usuario en GitHub.

Te permite:

Modificar libremente el código sin afectar el proyecto original.

Contribuir al proyecto original mediante Pull Requests (solicitudes de cambios).

Experimentar con nuevas ideas en tu propia versión.

- ¿Cómo crear un fork de un repositorio?

Ir al repositorio original (ej: [github.com/...](https://github.com/...)).

Hacer clic en "Fork" (esquina superior derecha).

Elige cuenta y listo.

- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

1. Hacer un Fork del repositorio original
2. Clona tu Fork localmente:  

```
git clone https://github.com/tu-usuario/repo.git
```

```
cd repo
```
3. Configura el repositorio original como "upstream"  

```
git remote add upstream https://github.com/autor/repo.git
```
4. Crea una rama nueva para tus cambios  

```
git checkout -b mi-feature
```
5. Haz tus modificaciones y envíalas a tu Fork  

```
git add .
```

```
git commit -m "Descripción clara de los cambios"
```

```
git push origin mi-feature
```
6. Ir a GitHub y abre el Pull Request  
Entra a tu Fork en GitHub.

Verás un botón "Compare & Pull Request" (aparece después de git push).

Completa el formulario:

Título claro: Ej: "Arregla el bug en el login".

Descripción detallada: Explica qué cambiaste y por qué.

Selecciona la rama destino: Usualmente main o master del repo original.

Haz clic en "Create Pull Request".

- ¿Cómo aceptar una solicitud de extracción?

Desde la Interfaz Web de GitHub

Ve al repositorio donde se encuentra la solicitud de extracción.

Haz clic en la pestaña "Pull requests" en la parte superior.

Selecciona la pull request que deseas revisar.

Revisa los cambios en la pestaña "Files changed" para asegurarte de que todo está correcto.

Aprueba o solicita cambios en la pestaña "Review changes":

Si estás satisfecho, haz clic en "Approve" y luego en "Submit review".

Si necesitas cambios, selecciona "Request changes".

Haz clic en "Merge pull request" y luego en "Confirm merge" para aceptar la solicitud.

(Opcional) Elimina la rama haciendo clic en "Delete branch" si ya no es necesaria.

- ¿Qué es una etiqueta en Git?

En Git, una etiqueta (tag) es una referencia estática a un commit específico, generalmente usado para marcar versiones importantes (ej: v1.0.0, v2.3.1).

A diferencia de las ramas (que cambian con nuevos commits), las etiquetas son inmutables y funcionan como "puntos de control" en el historial.

- ¿Cómo crear una etiqueta en Git?

1. Crear una etiqueta ligera "git tag nombre\_etiqueta"
2. Crear una etiqueta anotada "git tag -a nombre\_etiqueta -m "Mensaje de la etiqueta""

- ¿Cómo enviar una etiqueta a GitHub?

Con "git push origin v1.0"

- ¿Qué es un historial de Git?

El historial de Git es el registro de todos los cambios realizados en un repositorio a lo largo del tiempo. Contiene información sobre cada commit, incluyendo quién lo hizo, cuándo se hizo y qué cambios se aplicaron.

- ¿Cómo ver el historial de Git?

`git log` “muestra lista de commits con detalle basico”

- ¿Cómo buscar en el historial de Git?

Git ofrece varias formas de buscar en el historial de commits para encontrar cambios específicos en un proyecto

1. Buscar por palabra clave en los mensajes de commit

`git log --grep="texto_a_buscar"`

2. 2. Buscar commits hechos por un autor específico

`git log --author="nombre_autor"`

- ¿Cómo borrar el historial de Git?

1. Crear una nueva rama sin historial

`git checkout --orphan nueva_rama`

`git add .`

`git commit -m "Nuevo comienzo sin historial"`

`git branch -D main` # Borra la rama anterior

`git branch -m main` # Renombra la nueva rama a 'main'

`git push -f origin main` # Sobrescribe el repositorio remoto

2. Borrar todo el historial pero mantener los archivos

`rm -rf .git` # Borra todo el historial de Git

`git init` # Inicia un nuevo repositorio

`git add .`

`git commit -m "Inicio de nuevo historial"`

`git remote add origin URL_DEL_REPO` # Vuelve a enlazar el repositorio remoto

`git push -f origin main` # Sobrescribe el repositorio remoto

3. Resetear el historial a un commit específico

`git reset --hard ID_DEL_COMMIT`

`git push --force origin main`

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un repositorio cuyo contenido solo puede ser visto y accedido por ti y las personas a las que les des permisos.

- ¿Cómo crear un repositorio privado en GitHub?

1 Inicia sesión en GitHub y ve a GitHub.

- 2 Haz clic en "New" (Nuevo repositorio) en la esquina superior derecha.
- 3 Elige un nombre para el repositorio.
- 4 Selecciona "Private" (Privado) en la sección de visibilidad.
- 5 Opcional: Agrega un README, .gitignore o una licencia.
- 6 Haz clic en "Create repository" (Crear repositorio).

- ¿Cómo invitar a alguien a un repositorio privado en GitHub?

- 1 Ve a tu repositorio en GitHub.
- 2 Entra a "Settings" (Configuración).
- 3 En el menú lateral, selecciona "Collaborators" (Colaboradores).
- 4 Haz clic en "Add people" (Agregar personas).
- 5 Escribe el nombre de usuario o correo de la persona.
- 6 Selecciona el nivel de acceso (lectura, escritura, administrador).
- 7 Envía la invitación.

- ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un repositorio cuyo contenido es accesible para cualquier persona en internet. Cualquier usuario puede ver el código, clonarlo y utilizarlo, pero solo quienes tengan permisos pueden modificarlo.

- ¿Cómo crear un repositorio público en GitHub?

- 1 Inicia sesión en GitHub y ve a GitHub.
- 2 Haz clic en "New" (Nuevo repositorio) en la esquina superior derecha.
- 3 Elige un nombre para el repositorio.
- 4 Selecciona "Public" (Público) en la sección de visibilidad.
- 5 Opcional: Agrega un README, .gitignore o una licencia.
- 6 Haz clic en "Create repository" (Crear repositorio).

- ¿Cómo compartir un repositorio público en GitHub?

Método 1: Compartir el enlace del repositorio

- 1 Abre tu repositorio en GitHub.
- 2 Copia la URL desde la barra de direcciones del navegador.

Ejemplo: <https://github.com/tu-usuario/tu-repositorio>

3 Pega el enlace en donde quieras compartirlo (redes sociales, mensajes, correos, etc.).

Método 2: Clonar el repositorio con Git

Si otra persona quiere trabajar con tu código, puede clonarlo usando Git:

`git clone https://github.com/tu-usuario/tu-repositorio.git`