

Trabajo Práctico Git y GitHub

- ¿Qué es GitHub?

GitHub es una plataforma que nos permite publicar y gestionar repositorios de código de forma pública o privada. Nos facilita recuperar nuestros proyectos y clonarlos (git clone) en otra computadora para continuar trabajando sin interrupciones.

Además, permite que otros usuarios *forkeen* (git fork) un repositorio, es decir, creen una copia en su cuenta para realizar modificaciones. Luego, pueden proponer cambios a través de un *pull request*, que permite al equipo del proyecto revisar y, si es aprobado, fusionar los cambios en el repositorio original.

GitHub también ofrece herramientas avanzadas como **GitHub Actions**, que automatizan tareas de integración y despliegue continuo (CI/CD), facilitando pruebas y entregas de software de manera eficiente.

- ¿Cómo crear un repositorio en GitHub?

Una vez registrado en la plataforma vamos a iniciar primero nuestro repositorio local

Pasos para crear un repositorio git:

1. **git init** : inicia un repositorio
2. **git add .** : guarda todos los cambios
3. **git commit -m:** registra los cambios en ese punto para después poder volver si es necesario
4. **git config - -global user.email** : comando para registrar email
5. **git config - -global user.name** : comando para ingresar el nombre

Desde la plataforma de Github en nuestro perfil vamos a *create a new repository* e ingresamos el nombre del repositorio y una descripción (op.)

Crear un nuevo repositorio

Un repositorio contiene todos los archivos del proyecto, incluido el historial de revisiones. ¿Ya tienes un repositorio de proyectos en otro lugar? [Importa un repositorio.](#)

Los campos obligatorios están marcados con un asterisco (*).

Dueño * / **Nombre del repositorio ***


 cristianobus0909 /


✓ mi_nuevo_repositorio está disponible.

Los buenos nombres de repositorios son cortos y fáciles de recordar. ¿Necesitas inspiración? ¿Qué te parece? **turbo-carnaval ?**

Descripción (opcional)

Mas abajo, si lo queremos público o privado, y otras opciones como agregar un README, gitignore, licencia. Seguido creamos el repositorio

☒  **Público**
Cualquier persona en internet puede ver este repositorio. Tú decides quién puede contribuir.

☐  **Privado**
Tú eliges quién puede ver y comprometerse con este repositorio.

Inicialice este repositorio con:

☐ **Agregar un archivo README**
Aquí puedes escribir una descripción detallada de tu proyecto. [Obtén más información sobre los archivos README.](#)

Agregar .gitignore


Plantilla .gitignore : **Ninguna** ▼

Seleccione los archivos que no desea rastrear de una lista de plantillas. [Obtenga más información sobre cómo ignorar archivos.](#)

Elija una licencia

Licencia : **Ninguna** ▼

Una licencia indica a otros qué pueden y no pueden hacer con tu código. [Obtén más información sobre las licencias.](#)

 Estás creando un repositorio público en tu cuenta personal.

Crear repositorio

Posteriormente nos da una serie de pasos para crear un repositorio remoto de nuestro repositorio local, esto se realiza una sola vez

6. `git remote add origin https://github.com/miusuariogithub/primer-repo-git.git`
7. `git branch -M master`: crea la rama principal
8. `git push origin master`: publica los commits de los cambios a la nueva version



- ¿Cómo crear una rama en Git?

Para crear una rama primero desde la terminal en VSC debemos estar ubicados en la raíz de nuestro repositorio local, que fue inicializado y los cambios guardados anteriormente, con el comando `git branch "nombre_de_rama"` creamos una nueva rama.

- ¿Cómo cambiar a una rama en Git?

Para cambiar de rama con el comando `git checkout "nombre_de_rama"` cambiamos a la rama indicada

- ¿Cómo fusionar ramas en Git?

Primero debemos ubicarnos en la rama que queremos fusionar los cambios, por ejemplo: *git checkout "nombre-de-la-rama"* y luego fusionar las ramas *git merge "nombre-de-la-rama"* si no hay conflictos, Git aplicara los cambios automáticamente.

- ¿Cómo crear un commit en Git?

Commit: Es un punto en la línea de tiempo del proyecto donde registramos los cambios, permitiéndonos regresar a ese estado si es necesario. Siempre va acompañado de un mensaje que ayuda a identificarlo. Además, Git asigna un código *hash* único a cada *commit* para su referencia.

1. *git status*: revisamos los cambios a guardar
2. *git add .* : guarda todos los cambios
3. *git commit -m*: registra los cambios en ese punto para después poder volver si es necesario

- ¿Cómo enviar un commit a GitHub?

Debemos realizar primero los pasos anteriores y después ingresar el comando *git push*, este nos publica el nuevo commit

- ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión del proyecto almacenado en un servidor en línea, como **GitHub, GitLab o Bitbucket**. Permite que múltiples personas trabajen en el mismo código desde diferentes computadoras y mantengan sincronizados sus cambios.

- ¿Cómo agregar un repositorio remoto a Git?

Como fue explicado anteriormente debemos ingresar a la plataforma, en este caso Github, desde nuestro perfil vamos a *create a new repository*, colocamos el nombre al repositorio y Creamos el repositorio en VSC nos ubicamos en la raíz del proyecto donde inicializamos y realizamos los cambios anteriormente y seguimos los pasos que nos indica la plataforma.

Seguimos estos pasos desde la terminal:

```
cd mi-proyecto
```

```
git init
```

```
git remote add origin https://github.com/usuario/mi-proyecto.git
```

```
git add .
```

```
git commit -m "Primer commit"
```

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

- ¿Cómo empujar cambios a un repositorio remoto?

Cuando trabajamos en un repositorio local y queremos subir los cambios al repositorio remoto en este caso Github, debemos usar el comando *git push*.

- ¿Cómo tirar de cambios de un repositorio remoto?

Cuando hay actualizaciones en un repositorio remoto y queremos obtener esos cambios en nuestro repositorio local, utilizamos el comando *git pull*. Este comando descarga los cambios y los fusiona automáticamente con la rama local.

- ¿Qué es un fork de repositorio?

Un **fork** de repositorio es una copia completa de un repositorio que se crea en tu cuenta de **GitHub**. Este proceso permite hacer cambios en el código sin afectar el repositorio original. Los *forks* son muy útiles en proyectos de código abierto, ya que facilitan las contribuciones de cualquier usuario al proyecto original.

- ¿Cómo crear un fork de un repositorio?

Primero accedemos al repositorio original. En la esquina superior derecha de la página del repositorio, encontramos un botón llamado **"Fork"**, clickeamos ahí y seleccionamos la cuenta destino que se desea hacer el fork o copia. Luego, para trabajar de forma local con el código, debemos clonar el repositorio *git clone https://github.com/tu-usuario/repositorio-forkeado.git*

Luego de realizar los cambios subimos dichos cambios al repositorio copia remoto con *git push*

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

Un **pull request (PR)** es una solicitud para que los mantenedores de un repositorio revisen y fusionen tus cambios.

1. Realizamos un **Fork y clone** al repositorio
2. Creamos una nueva rama para los nuevos cambios.
3. Realiza cambios, agrégalos un commit y súbelos.
4. Vamos a GitHub → Nuestro repositorio forkeado → "Compare & pull request".
5. Escribimos una descripción y enviamos el PR.

- ¿Cómo aceptar una solicitud de extracción?

Si eres el propietario de un repositorio y alguien envió un PR, puedes aceptarlo.

1. Vamos a la pestaña "**Pull Requests**" en el repositorio.
2. Hacemos clic en el PR que quieres aceptar.
3. Revisamos los cambios y, si todo está bien, haz clic en "**Merge pull request**".
4. Confirmamos con "**Confirm merge**".
5. (Opcional) Borramos la rama después de fusionarla.

- ¿Qué es una etiqueta en Git?

Una **etiqueta (tag)** en Git es un marcador que se usa para identificar puntos importantes en la historia del repositorio, como una versión estable de un proyecto (v1.0, v2.0, etc.).

- ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta anotada (con mensaje y metadatos)

git tag -a v1.0 -m "Versión 1.0 estable"

Para crear una etiqueta ligera (sin metadatos)

```
git tag v1.0
```

- ¿Cómo enviar una etiqueta a GitHub?

Después de crear una etiqueta, debemos subirla al repositorio remoto con el comando

```
git push origin v1.0
```

Si deseamos enviar todas las etiquetas creadas

```
git push origin --tags
```

- ¿Qué es un historial de Git?

El **historial de Git** es el registro de todos los commits realizados en el proyecto.

- ¿Cómo ver el historial de Git?

Para ver el historial de commits debemos ingresar:

```
git log
```

```
git log --oneline
```

 → Muestra los commits en una sola línea.

```
git log --graph
```

 → Visualiza el historial como un gráfico.

```
git log --author="tu-nombre"
```

 → Muestra solo los commits de un usuario específico.

- ¿Cómo buscar en el historial de Git?

Para buscar commits específicos en el historial

```
git log --grep="Texto del mensaje"
```

```
git log -- nombre-del-archivo
```

- ¿Cómo borrar el historial de Git?

Hay que estar muy seguros de hacer esto ya que es irreversible. Esto reinicializa el repositorio y borra todo el historial

```
rm -rf .git
```

```
git init
```

- ¿Qué es un repositorio privado en GitHub?

Un **repositorio privado** en GitHub es aquel que solo pueden ver y acceder las personas con permiso. Es útil para proyectos confidenciales.

- ¿Cómo crear un repositorio privado en GitHub?
 1. Vamos a [GitHub](#) y hacemos clic en "**New Repository**".
 2. Escribe el nombre del repositorio.
 3. Seleccionamos "**Private**".
 4. Creamos el repositorio.
- ¿Cómo invitar a alguien a un repositorio privado en GitHub?
 1. Vamos a la configuración del repositorio (**Settings**).
 2. En **Manage Access**, hacemos clic en "**Invite a collaborator**".
 3. Escribe el usuario de GitHub de la persona y enviamos la invitación.

- ¿Qué es un repositorio público en GitHub?

Un **repositorio público** en GitHub es un repositorio accesible por cualquier persona en internet. Es útil para proyectos de código abierto.

- ¿Cómo crear un repositorio público en GitHub?
 1. Ingresamos a [GitHub](#) y hacemos clic en "**New Repository**".
 2. Escribimos el nombre del repositorio.
 3. Seleccionamos "**Public**".
 4. Creamos el repositorio.

- ¿Cómo compartir un repositorio público en GitHub?

Solo necesitas compartir la URL del repositorio

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