



Trabajo Practico N°2

Git y GitHub

Programación I

Alumna: Lis Araceli Lezcano Thomas

Carrera: Tecnicatura Universitaria en Programación a Distancia

Mes y Año: Marzo 2025

Ejercicios Resueltos

1. Contestar las siguientes preguntas utilizando las guías y documentación proporcionada:

- ¿Qué es GitHub?

GitHub es una plataforma de alojamiento y colaboración para proyectos que utilizan Git. Es como una "red social para desarrolladores" donde puedes compartir código, clonar proyectos o contribuir a otros.

Permite:

- Almacenar repositorios públicos o privados.
- Gestionar versiones de código (commits, ramas, pull requests).
- Colaborar con otros usuarios (issues, forks, revisiones).
- Personalizar perfiles para mostrar contribuciones y proyectos.

- ¿Cómo crear un repositorio en GitHub?

Pasos:

1. Iniciar sesión en GitHub
2. Hacer clic en "+ > New repository" en la esquina superior derecha.
3. Completar:
 - a. Nombre del repositorio
 - b. Descripción (opcional).
 - c. Elegir visibilidad (público o privado).
4. Hacer clic en "Create repository".

- ¿Cómo crear una rama en Git?

Las ramas son líneas de tiempo independientes para desarrollar features sin afectar el código principal.

Para crear una rama, hay que ejecutar el comando:

git branch nombre-de-la-rama

- ¿Cómo cambiar a una rama en Git?

Para cambiar de rama hay que ejecutar el comando

git checkout nombre-de-la-rama

- ¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, hay que posicionarse en la rama principal, donde queremos que queden guardados los cambios, ejecutando **git checkout nombre-del-archivo**

Una vez posicionado, fusionamos la rama ejecutando **git merge nombre-del-archivo**

Si hay conflictos, Git te pedirá resolverlos manualmente.

- ¿Cómo crear un commit en Git?

Un commit es una instantánea del proyecto hasta el momento, cuando tenemos una nueva dependencia generamos un commit.

Para crear un commit en Git, realizar los comandos:

1. **git add .** (para guardar los cambios)
2. **git commit -m "mensaje "** (crear el commit con un mensaje descriptivo)

- ¿Cómo enviar un commit a GitHub?

Estos son los pasos para enviar un commit a GitHub:

- 1- Verificar el estado de los cambios con **git status**
- 2- Agregar los cambios con **git add**.
- 3- Crear un commit con un mensaje descriptivo **git commit -m "mensaje"**
- 4- Enviar el commit a GitHub **git push origin nombre-de-la-rama**

- ¿Qué es un repositorio remoto?

Es una versión de tu repositorio alojada en un servidor, como GitHub, y permite colaborar con otros, sincronizar cambios (push y pull) y hacer un backup del código.

- ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto, hay que obtener la URL del repositorio remoto. Desde GitHub hay que ir al repositorio, hacer clic en "Code" y copiar la URL. Desde la terminal, hay que ejecutar **git remote add origin URL-del-repositorio**

- ¿Cómo empujar cambios a un repositorio remoto?

Pasos para empujar (push) cambios a un repositorio remoto

- 1- Verificar el estado de cambios con **git status**
- 2- **git add .**
- 3- **git commit -m "Descripción de los cambios"**
- 4- Enviar los cambios al repositorio remoto **git push origin nombre-rama**

- ¿Cómo tirar de cambios de un repositorio remoto?

Para tirar (pull) cambios de un repositorio remoto estos son los pasos:

- 1- Asegurarse de estar en la rama correcta con **git checkout nombre-rama**
- 2- Obtener los últimos cambios del remoto **git pull origin nombre-rama**

- ¿Qué es un fork de repositorio?

Un fork es una copia exacta de un repositorio en tu cuenta de GitHub, que puedes modificar sin afectar al original. Se utiliza comúnmente para contribuir a proyectos open source y experimentar con cambios propios.

- ¿Cómo crear un fork de un repositorio?

Haciendo clic en "**Fork**" en la página del repositorio

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

Para enviar una pull request a tu repositorio hay que:

- 1- Hacer cambios en tu fork y subirlos ejecutando **git push origin tu-rama**
- 2- En GitHub, ir al repositorio original y hacer clic en "New Pull Request".
- 3- Comparar las ramas (tu fork vs. original) y describir los cambios.

- ¿Cómo aceptar una solicitud de extracción?

Pasos para aceptar un Pull Request en GitHub:

1. Entrar al PR:
 - Ir al repositorio en GitHub.
 - Hacer clic en "**Pull requests**".
 - Elegir el PR que quieres aceptar.
2. Revisar los cambios:
 - Mirar los archivos modificados en "**Files changed**".
 - Si GitHub dice "**This branch has no conflicts**", puedes fusionarlo.
3. Fusión:
 - Hacer clic en "**Merge pull request**".
 - Elegir un método (normalmente "**Merge commit**").
 - Confirmar con "**Confirm merge**".
4. Opcional:
 - Si ya no necesitas la rama, hacer clic en "**Delete branch**".

- ¿Qué es una etiqueta en Git?

Una etiqueta (tag) en Git es como un punto de guardado especial que marca una versión importante de tu proyecto. Es como ponerle un nombre a un commit para encontrarlo fácilmente. Las etiquetas sirven para marcar versiones estables, saber que

código tenía el proyecto en un momento clave y generar releases en GitHub (paquetes descargables).

- ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta ejecutas **git tag v1.0.0** (etiqueta ligera) o **git tag -a v1.0.0 -m "Version 1.0 estable"**

- ¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta a GitHub Subir las etiquetas a GitHub ejecutando **git push --tags**

- ¿Qué es un historial de Git?

El historial de Git es como el libro de registro de tu proyecto. Guarda todos los cambios (commits) que hiciste, mostrando quién hizo cada cambio, cuándo se hizo, qué archivos se modificaron y los mensajes que explicaron por qué,

- ¿Cómo ver el historial de Git?

El **git reflog** es como un diario secreto de Git que registra todo lo que haces (incluso cambios "borrados"). Muestra: commits, cambios de ramas, resets, merges, etc. Este comando solo guarda cambios por 90 días, luego Git lo limpia.

- ¿Cómo buscar en el historial de Git?

- **--grep** para buscar en mensajes.
- **-S** para buscar cambios en código.
- **--since/--until** para filtrar por fechas.

- ¿Cómo borrar el historial de Git?

Para tener un nuevo historial sin perder archivos actuales:

- Borra la carpeta .git (esto elimina todo el historial) con **rm -rf .git**
- Inicializa un nuevo repositorio con **git init**
- Vuelve a agregar los archivos con :
- **git add .**
- **git commit -m "Primer commit"** (historial nuevo)

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un proyecto de código accesible únicamente para usuarios autorizados. Destaca por tener un acceso restringido, solo es visible y puede ser editado por colaboradores explícitamente invitados.

- ¿Cómo crear un repositorio privado en GitHub?

Pasos para crear un repositorio privado en GitHub:

1. Ingresar a GitHub e iniciar sesión.
2. Para crear nuevo repositorio tocar el botón "+" y "New repository".
3. Configurar:
 - Nombre y descripción
 - Seleccionar "Private" en visibilidad
4. Confirmar: Clic en "Create repository".

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

1. Ir al repositorio, donde dice "Settings" (Configuración).
2. En el menú izquierdo, seleccionar "Collaborators".
3. Hacer clic en "Add people" e ingresar el usuario GitHub o email del colaborador.
4. Elegir permisos (Read, Write o Admin) y confirmar con "Add [usuario]".

El invitado recibirá un email y debe aceptar la invitación para acceder.

- ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un espacio de almacenamiento de código accesible para cualquier usuario de la plataforma, donde el contenido queda disponible de manera abierta y gratuita, permitiendo su visualización, modificación y distribución por parte de la comunidad.

- ¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio público en GitHub:

1. Iniciar sesión en la cuenta personal de GitHub.
2. Seleccionar la opción "New repository" desde el menú principal.
3. Completar los campos requeridos:
 - Asignar un nombre al repositorio
 - Agregar descripción (opcional)
 - Seleccionar la visibilidad "Public"
4. Confirmar la creación mediante el botón "Create repository".

Una vez creado, el repositorio estará disponible públicamente y podrá ser clonado o bifurcado por cualquier usuario de la plataforma.

- ¿Cómo compartir un repositorio público en GitHub?

Pasos para compartir un repositorio público en GitHub:

- 1- Acceder a tu repositorio en GitHub.
- 2- Copiar la URL del repositorio (está en la barra de direcciones o en el botón "Code").
- 3- Compartir el enlace directamente o usar las opciones de GitHub (como "Invite a collaborator" si querés dar permisos).

Cualquiera con el link puede verlo, pero para editar, necesitarán permisos.

2. Crear un repositorio, agregar un archivo y crear una Branch

<https://github.com/lislezcanothomas/trabajo-practico-2>

3. Crear un repositorio en GitHub, Clonar el repositorio a maquina local, crear una rama y editar un archivo, hacer un merge y generar un conflicto, resolver el conflicto y subir los cambios a GitHub

<https://github.com/lislezcanothomas/conflict-exercise>