



TP 2: PREGUNTAS SOBRE GIT Y GITHUB



Actividades

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

• ¿Qué es GitHub?

GitHub es una plataforma de alojamiento de código basada en la nube que permite a los desarrolladores gestionar proyectos usando Git. Facilita la colaboración, el control de versiones y la gestión de código fuente para equipos y proyectos individuales.

• ¿Cómo crear un repositorio en GitHub?

1. Inicia sesión en GitHub.
2. Haz clic en el botón "New repository" (Nuevo repositorio) en la esquina superior derecha.
3. Escribe un nombre para el repositorio.
4. Opcionalmente, añade una descripción, eligiendo si será público o privado y seleccionamos opciones como un archivo README.
5. Haz clic en "Create repository" (Crear repositorio).

Si queremos conectarlo con Git en nuestra computadora:

```
git init
git remote add origin https://github.com/tu-
usuario/nombre-del-repo.git
git add .
git commit -m "Primer commit"
git push -u origin main
```

• ¿Cómo crear una rama en Git?

Para crear una nueva rama en Git, se usa el siguiente comando:

```
git branch nombre-de-la-rama
```

• ¿Cómo cambiar a una rama en Git?

Para cambiar a otra rama, usamos:

```
git checkout nombre-de-la-rama
```

• ¿Cómo fusionar ramas en Git?

Primero, cambiamos a la rama en la que deseamos fusionar los cambios (por ejemplo, main):

```
git checkout main
```

Luego, ejecuta:

```
git merge nombre-de-la-rama
```

Si hay conflictos, Git te pide que los resuelvas manualmente en los archivos afectados antes de confirmar los cambios con:

```
git add .
```

```
git commit -m "Resolviendo conflictos de merge"
```

• ¿Cómo crear un commit en Git?

Añadir los archivos al área de preparación:

```
git add .
```

Crear el commit con un mensaje descriptivo:

```
git commit -m "Descripción del cambio"
```

• ¿Cómo enviar un commit a GitHub?

Subimos los cambios al repositorio remoto utilizando:

```
git push origin nombre-de-rama
```

• ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión alojada en la nube (como GitHub, GitLab o Bitbucket) de tu repositorio local. Permite que varias personas colaboren y accedan al código desde distintos lugares.

• ¿Cómo agregar un repositorio remoto a Git?

1. Copia la URL del repositorio en GitHub.

2. Agregamos el repositorio remoto con:

```
git remote add origin https://github.com/tu-usuario/nombre-del-repo.git
```

3. Verificamos que el remoto se agregó correctamente con:

```
git remote -v
```

• ¿Cómo empujar cambios a un repositorio remoto?

Después de realizar commits, subimos los cambios al repositorio remoto con:

```
git push origin nombre-de-la-rama
```

Si es la primera vez que subes la rama, usa:

```
git push -u origin nombre-de-la-rama
```

• ¿Cómo tirar de cambios de un repositorio remoto?

Para obtener los cambios más recientes desde un repositorio remoto y actualizar tu repositorio local, usa el comando:

```
git pull origin nombre-de-la-rama
```

Si estás trabajando en main, por ejemplo:

```
git pull origin main
```

• ¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio que permite a los usuarios realizar cambios de forma independiente sin afectar el repositorio original. Es

común en plataformas como GitHub para contribuir a proyectos de código abierto.

Cuando haces un fork, obtienes una copia completa del proyecto en tu cuenta, donde puedes experimentar y trabajar en cambios sin intervenir en el repositorio principal.

• ¿Cómo crear un fork de un repositorio?

Para crear un fork en GitHub:

1. Vamos a la página del repositorio que deseas forkar.
2. En la esquina superior derecha de la página, haz clic en el botón "Fork".
3. GitHub creará una copia del repositorio en tu cuenta, y podrás empezar a trabajar en él.

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

1. Hacemos un fork del repositorio y realizamos los cambios que deseamos.
2. Empuja los cambios al repositorio forkeado (en tu cuenta de GitHub).
3. Vamos a la página del repositorio forkeado en GitHub.
4. Clic en el botón "Compare & Pull Request".
5. Rellenamos el título y la descripción de la solicitud de extracción y seleccionamos la rama en la que deseamos fusionar cambios (normalmente la rama main del repositorio original).
6. Por último, Clic en "Create Pull Request" para enviarla.

• ¿Cómo aceptar una solicitud de extracción?

Para aceptar una solicitud de extracción en tu repositorio:

1. Ve al repositorio donde se ha enviado la solicitud de extracción.
2. En la pestaña "Pull requests", selecciona la solicitud de extracción que deseas revisar.
3. Revisa los cambios y los comentarios de los colaboradores.

4. Si todo está correcto, haz clic en "Merge pull request" para fusionar los cambios con la rama principal.
5. Luego, confirma la fusión haciendo clic en "Confirm merge".

• ¿Qué es una etiqueta en Git?

Una etiqueta (tag) en Git es una referencia que se usa para marcar puntos específicos en la historia del proyecto, como versiones o lanzamientos importantes (por ejemplo, v1.0, v2.0). Las etiquetas no afectan el flujo de trabajo de las ramas, pero permiten un fácil acceso a momentos clave en el desarrollo, facilitando la referencia a versiones específicas del código.

• ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta, usamos el siguiente comando:

Etiqueta ligera (sin metadatos adicionales):

```
git tag nombre-de-la-etiqueta
```

Etiqueta anotada (con metadatos, como el autor y la fecha):

```
git tag -a nombre-de-la-etiqueta -m "Mensaje  
descriptivo"
```

• ¿Cómo enviar una etiqueta a GitHub?

Una vez creada la etiqueta localmente, para enviarla a GitHub utilizamos este comando:

```
git push origin nombre-de-la-etiqueta
```

Si deseamos enviar todas las etiquetas al mismo tiempo, usamos:

```
git push -tags
```

• ¿Qué es un historial de Git?

El historial de Git es el registro de todos los commits realizados en el repositorio. Cada commit contiene información sobre los cambios

realizados, como qué archivos fueron modificados, quién los modificó y cuándo. El historial te permite seguir la evolución del proyecto y revertir cambios si es necesario.

• ¿Cómo ver el historial de Git?

Para ver el historial de Git, usamos el comando:

```
git log
```

• ¿Cómo buscar en el historial de Git?

Para buscar en el historial de Git, usamos el comando `git log` con opciones de búsqueda.

1. Buscar por palabra clave en los mensajes de commit:

```
git log --grep="palabra-clave"
```

2. Buscar por autor:

```
git log --author="nombre-del-autor"
```

3. Buscar un commit específico usando una cadena de texto:

```
git log -S"texto"
```

4. Buscar por fecha: se puede buscar los commits realizados en un período específico:

```
git log --since="2025-01-01" --until="2025-03-03"
```

• ¿Cómo borrar el historial de Git?

Hay diferentes opciones en términos de borrados:

Si quieres eliminar el último commit, pero manteniendo los cambios:

```
git reset --soft HEAD~1
```

Si deseas eliminar completamente el commit y sus cambios:

```
git reset --hard HEAD~1
```

Eliminar el historial completo del repositorio:

```
rm -rf .git
```

git init

• ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un repositorio al que solo tienen acceso las personas que uno invite. A diferencia de los repositorios públicos, los repositorios privados no son accesibles a cualquier persona en Internet, lo que protege tu código y datos de miradas no autorizadas.

• ¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado en GitHub:

1. Inicia sesión en tu cuenta de GitHub.
2. Ve a la página principal de GitHub y haz clic en el botón "New" (Nuevo) para crear un nuevo repositorio.
3. En la página de creación de repositorio, rellena los campos obligatorios como nombre del repositorio.
4. Marca la opción "Private" para hacer que el repositorio sea privado.
5. Haz clic en "Create repository" para crear el repositorio privado.

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

Para invitar a colaboradores a tu repositorio privado:

1. Ve al **repositorio privado** en GitHub.
2. Haz clic en la pestaña "**Settings**" del repositorio.
3. En el menú de la izquierda, selecciona "**Manage access**" (Gestionar acceso).
4. Haz clic en el botón "**Invite a collaborator**" (Invitar a un colaborador).
5. Escribe el nombre de usuario de GitHub de la persona que deseas invitar y haz clic en "**Add**".
6. El invitado recibirá una notificación para acceder al repositorio privado.

• ¿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, clonar y contribuir al código en un repositorio público. Estos repositorios son ideales para proyectos de código abierto o cuando quieres que otros vean y colaboren en tu código.

• ¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio público en GitHub:

1. Inicia sesión en tu cuenta de GitHub.
2. Ve a la página principal de GitHub y haz clic en el botón "New" (Nuevo) para crear un nuevo repositorio.
3. En la página de creación de repositorio, rellena los campos obligatorios como nombre del repositorio.
4. Marca la opción "Public" para hacer que el repositorio sea público.
5. Haz clic en "Create repository" para crear el repositorio público.

• ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio público en GitHub, simplemente necesitas compartir la URL del repositorio.

1. Accede a tu repositorio:

Ve a la página del repositorio que deseas compartir en GitHub.

2. Copia la URL del repositorio:

En la barra de direcciones de tu navegador, encontrarás la URL del repositorio. Por ejemplo, <https://github.com/tu-usuario/nombre-del-repositorio>.

3. Comparte la URL:

Puedes compartir esta URL con cualquier persona para que vea, clone o contribuya al repositorio.

Nazareno Aranda