UTN - PROGRAMACIÓN I

Práctico 2: Git y GitHub

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

¿Qué es GitHub?

GitHub es una plataforma web colaborativa gratuita basada en Git, que permite a los usuarios publicar, compartir y colaborar en proyectos de programación. Los usuarios pueden gestionar repositorios, corregir errores, mejorar proyectos ajenos y aceptar cambios en su propio código. Su interfaz puede ser compleja y está solo en inglés, pero es accesible y facilita el trabajo en equipo en proyectos de código.

¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub se deben seguir los siguientes pasos:

- **Iniciar sesión en GitHub:** crear una cuenta en GitHub y acceder a la misma.
- **Crear nuevo repositorio:** en la cuenta principal de GitHub, en el margen superior izquierdo, se encontrará un botón color verde llamado “New” (nuevo) > clic en el botón.
- **Definir nombre y descripción:** una vez dentro del nuevo repositorio debemos definir un nombre único y una descripción opcional que explique de que trata el proyecto.
- **Configurar visibilidad:** elegir si el repositorio va a ser Público o Privado.
  Al colocarlo en modo “Público” cualquier persona podrá visualizarlo.
  En modo “Privado” solo podrán verlo los colaboradores que invites al repositorio.
- **Crear el repositorio:** una vez completado los campos requeridos, seleccionar botón verde ubicado en el margen inferior derecho llamado “Create repository”(crear repositorio).
  ¡Listo! Repositorio creado.

¿Cómo crear una rama en Git?

Una rama en Git es una copia del proyecto en la que podemos realizar cambios sin afectar la rama principal. Es útil para desarrollar nuevas funcionalidades o probar ideas.
Para crear una nueva rama en Git, usamos el comando:
git branch nombre_rama

¿Cómo cambiar a una rama en Git?

Para cambiar a una rama en Git, utilizamos el comando:
git checkout nombre_rama

Esto nos permite movernos a la rama indicada y trabajar en ella sin afectar las demás ramas.
A partir de ese momento, todos los cambios se realizarán en esa rama.
En versiones recientes de Git, también podemos usar el comando:
git switch nombre_rama

¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, utilizamos el comando git merge seguido del nombre de la rama que queremos fusionar con la rama actual en la que nos encontramos.
El comando básico es:
git merge nombre_rama

Este comando toma los cambios realizados en la rama especificada (por ejemplo, nombre_rama) y los combina con la rama activa (la que estamos trabajando actualmente). Esto es útil cuando queremos integrar nuevas características o cambios desarrollados en una rama separada a la rama principal (como main o master).

Es importante asegurarse de estar en la rama correcta antes de hacer el merge, para evitar fusionar ramas equivocadas. Para ello, usamos:
git checkout nombre_rama_actual
para cambiar a la rama en la que deseamos realizar la fusión.

¿Cómo crear un commit en Git?

Un commit en Git es un registro de cambios realizados en el proyecto.
Para crear uno, primero agregamos los archivos modificados con:
git add .
Luego, realizamos el commit con:
git commit -m "mensaje con cambios aplicados"

El mensaje describe los cambios. El commit se guarda en el repositorio local y se sube al remoto con git push cuando sea necesario.

¿Cómo enviar un commit a GitHub?

Para enviar un commit a GitHub, primero debemos asegurarnos de que nuestro repositorio local esté conectado a un repositorio remoto en GitHub.
Luego, después de realizar los cambios y haber creado el commit, podemos usar el siguiente comando:
git push origin nombre_rama

Esto envía los cambios de la rama local (nombre_rama) al repositorio remoto en GitHub.
Si es la primera vez que empujan esa rama, GitHub la creará automáticamente en el repositorio remoto.

¿Qué es un repositorio remoto?

Un repositorio remoto es una copia de nuestro repositorio que se encuentra almacenada en un servidor en línea, como GitHub, GitLab o Bitbucket. Este repositorio facilita el trabajo en equipo, ya que permite a varios usuarios acceder, modificar y sincronizar los cambios de un proyecto de manera centralizada.

¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git, se debe asociar nuestro repositorio local con un repositorio remoto en una plataforma como GitHub.
Utilizar el siguiente comando para vincularlos:
git remote add origin [URL_del_repositorio]

Este comando configura la URL del repositorio remoto, permitiendo que puedas enviar (push) y obtener (pull) cambios entre tu repositorio local y el remoto.

¿Cómo empujar cambios a un repositorio remoto?

Una vez que hayamos realizado cambios en nuestro repositorio local y hayamos hecho un commit, podemos enviarlos al repositorio remoto utilizando el siguiente comando:
git push origin nombre_rama

Este comando empuja (sube) los cambios de la rama especificada (nombre_rama) al repositorio remoto en GitHub u otra plataforma.

¿Cómo tirar de cambios de un repositorio remoto?

Si otras personas han realizado cambios en el repositorio remoto y deseamos traer esos cambios a nuestro repositorio local, podemos usar:
git pull origin nombre_rama

Este comando descarga y fusiona los cambios de la rama especificada del repositorio remoto con nuestra rama local.

¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio creada en otra cuenta.
Permite hacer cambios sin afectar el repositorio original.A diferencia del clonado, que descarga el repositorio localmente, el fork se genera directamente en la cuenta del usuario.

¿Cómo crear un fork de un repositorio?

Para crear un fork se deben seguir los siguientes pasos:

a. Iniciar sesión en GitHub.
b. Acceder al repositorio: ingresar al enlace de un repositorio determinado.
c. Crear un Fork: buscar el botón “Fork” y presionarlo para crear una copia del repositorio en nuestra cuenta personal.
d. Cambiar la descripción del Fork.
e. Confirmar y crear Fork: buscar el botón “Crear Fork” y presionarlo.

¡Listo! Repositorio creado.

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

Para enviar una solicitud de extracción (pull request) a un repositorio, debemos seguir los siguientes pasos:

a. Subir nuestros cambios a nuestro repositorio remoto.
   git push origin nombre_rama
b. Acceder al repositorio en GitHub: dirigirse a la página del repositorio donde se desee enviar la solicitud de extracción.
c. Inicia la solicitud de extracción: dar clic en el botón "Pull Requests" y luego en "New Pull Request".
d. Seleccionar la rama: elegir la rama desde la que se desea hacer la solicitud de extracción (nuestra rama) y la rama a la que deseas que se fusionen los cambios (generalmente la rama main o master).
e. Agregar título y descripción: agregar un título y una descripción detallada de los cambios que se están proponiendo.
f. Crear Pull Request: dar clic en "Create Pull Request" para enviar la solicitud.

¿Cómo aceptar una solicitud de extracción?

Para aceptar una solicitud de extracción (pull request) en GitHub, debemos revisar el código propuesto por otro colaborador.

Si todo está en orden, debemos seguir los siguientes pasos:

- Ir al repositorio en GitHub y buscar la sección de "Pull Requests".
- Dar clic sobre la solicitud de extracción que deseamos revisar.
- Una vez revisado, dar clic en el botón "Merge pull request".
- Luego, confirma la acción dando clic en "Confirm merge".

Esto fusionará los cambios propuestos con la rama principal o la rama especificada.

¿Qué es un etiqueta en Git?

Una etiqueta (tag) en Git es una referencia especial que señala un punto específico en el historial de commits, generalmente usado para marcar versiones importantes, como lanzamientos o hitos del proyecto. A diferencia de las ramas, las etiquetas no cambian ni se actualizan con nuevos commits; simplemente sirven como marcadores permanentes para versiones significativas.

¿Cómo crear una etiqueta en Git?

Para crear una etiqueta en Git debemos utilizar el siguiente comando.

git tag nombre_etiqueta

Esto crea una etiqueta en el commit actual, la cual puede ser usada para identificar una versión o punto específico del proyecto.

¿Cómo enviar una etiqueta a GitHub?

Una vez que creamos una etiqueta en nuestro repositorio local, para enviarla (push) al repositorio remoto de GitHub, debemos usar el siguiente comando:

git push origin nombre_etiqueta

Este comando empuja la etiqueta específica al repositorio remoto, lo que permite que otros colaboradores vean y utilicen esa etiqueta en su repositorio local.

¿Qué es un historial de Git?

El historial de Git es un registro completo de todos los commits realizados en un repositorio. Cada commit guarda una instantánea de los cambios realizados en el código, junto con información sobre quién hizo el cambio, cuándo y por qué. El historial permite a los desarrolladores revisar el progreso del proyecto y retroceder a versiones anteriores si es necesario.

¿Cómo ver el historial de Git?

Para ver el historial de Git debemos utilizar el siguiente comando.
git log

Este comando muestra una lista de todos los commits realizados en el repositorio.
Se presentan detalles como el identificador único (hash) del commit, el autor, la fecha y el mensaje del commit. Además, podemos ver los cambios realizados y retroceder en el tiempo si es necesario. Podemos usar opciones adicionales para personalizar la vista del historial, como `--oneline` para un formato más compacto.

¿Cómo buscar en el historial de Git?

Para buscar en el historial de Git debemos utilizar el siguiente comando.
git log --grep="palabra_clave"
Esto nos permitirá buscar dentro de los mensajes de commit cualquier palabra clave específica. Es útil para encontrar commits relacionados con un tema o cambio particular.
Podemos también combinarlo con otros filtros, como `--author` para buscar commits de un autor específico.

¿Cómo borrar el historial de Git?

Para eliminar el historial de Git debemos utilizar el siguiente comando.

- git reset --hard HEAD~n

Este comando elimina los últimos `n` commits, donde `n` es el número de commits que deseamos eliminar. Tener en cuenta que esto modifica el historial de forma irreversible, por lo que es importante usarlo con precaución. Esto puede ser útil para deshacer cambios erróneos o restablecer el proyecto a un estado anterior.

¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un tipo de repositorio cuyo contenido solo puede ser visto, modificado o gestionado por usuarios específicos que hayan sido invitados o autorizados por el propietario del repositorio. A diferencia de los repositorios públicos, que son accesibles a cualquier persona en internet, los repositorios privados permiten un control total sobre quién puede acceder a la información, lo que es ideal para proyectos sensibles o colaboraciones en entornos más controlados. Los propietarios y administradores pueden añadir colaboradores mediante invitaciones y asignar permisos de acceso según sea necesario, asegurando que solo personas autorizadas puedan interactuar con el repositorio.

¿Cómo crear un repositorio privado en GitHub?
Para crear un repositorio privado en GitHub, iniciar sesión en nuestra cuenta y dar clic en el botón verde "New" para crear un nuevo repositorio. En el formulario de creación, ingresar el nombre y una descripción opcional para nuestro repositorio. Bajo la sección de visibilidad, seleccionar la opción "Private" o "Privado". Esto hará que solo los usuarios que invitemos o autoricemos puedan acceder a él. Finalmente, dar clic en "Create repository" para completar la creación de nuestro repositorio privado.

¿Cómo invitar a alguien a un repositorio privado en GitHub?
Para invitar a alguien a un repositorio debemos ir a la configuración del repositorio, seleccionar "Manage access" y agregar a los colaboradores.

¿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 proyecto, siempre y cuando el propietario del repositorio lo permita. Los repositorios públicos son ideales para proyectos de código abierto, ya que fomentan la colaboración global, permiten a otros usuarios revisar el código, sugerir mejoras o incluso proponer cambios mediante solicitudes de extracción (pull requests). Además, los repositorios públicos son indexados por motores de búsqueda, lo que facilita que otros encuentren el proyecto.

¿Cómo crear un repositorio público en GitHub?
Para crear un repositorio público en GitHub, primero debemos iniciar sesión en nuestra cuenta de GitHub. Luego, en la página principal de nuestra cuenta, dar clic en el botón verde "New" para crear un nuevo repositorio. En el formulario de creación, debemos asegurarnos de ingresar un nombre único para nuestro repositorio y una descripción opcional. Después, seleccionar la opción "Public" o "Público" bajo la sección de visibilidad. Esto hará que el repositorio sea accesible para cualquier persona en Internet. Finalmente, dar clic en "Create repository" para finalizar la creación del repositorio público.

¿Cómo compartir un repositorio público en GitHub?
Para compartir un repositorio público en GitHub, simplemente copiar la URL del repositorio desde la barra de direcciones del navegador, y luego compartirla con otras personas. Cualquier usuario con esa URL podrá acceder al repositorio y ver nuestro contenido, ya que es público.

2. **Crear un repositorio en GitHub, agregar un archivo y crear branchs.**
   Link del repositorio: https://github.com/AriMaldo19/Programacion1-UTN

3. **Crear un repositorio en GitHub**, c**lonar el repositorio a tu máquina local**, **crear una nueva rama y editar un archivo, volver a la rama principal y editar el mismo archivo, Hacer un merge y generar un conflicto, resolver el conflicto, subir los cambios a GitHub y verificar en GitHub.**
   Link del repositorio: https://github.com/AriMaldo19/conflict-exercise