

Programación 1

Trabajo Práctico N° 2 - Trabajo Colaborativo

Resolución de actividades

Actividad 1:

¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo basada en la web que permite a los programadores y equipos de trabajo almacenar, gestionar y colaborar en proyectos de software. Manteniendo un control de versiones y un seguimiento de los cambios realizados en el código.

¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub, primero inicia sesión en tu cuenta y haz clic en el botón **"New Repository"**. Luego, configura el repositorio asignándole un nombre, una breve descripción del proyecto y seleccionando su visibilidad (público o privado). Opcionalmente, puedes marcar la opción **"Add a README file"** para incluir un archivo `README.md`. Finalmente, haz clic en **"Create Repository"** para completar el proceso.

¿Cómo crear una rama en Git?

Las ramas en Git permiten trabajar en nuevas funcionalidades sin afectar el código principal. Para crear una nueva rama en Git, utiliza el comando `git branch nombre-de-la-rama`.

¿Cómo cambiar a una rama en Git?

Antes de cambiar de rama, puedes ver las ramas existentes con: `git branch`. Para cambiar de rama, lo puedes hacer con `git checkout nombre-de-la-rama`. Si tienes una versión más reciente de Git, también puedes usar `git switch nombre-de-la-rama`.

Si prefieres crear y cambiar a la nueva rama en un solo paso, puedes ejecutar `git checkout -b nombre-de-la-rama` o `git switch -c nombre-de-la-rama`.

¿Cómo fusionar ramas en Git?

Fusionar ramas en Git permite combinar los cambios de una rama en otra. Esto es útil cuando has trabajado en una nueva funcionalidad y quieres integrarla en la rama principal. Antes de fusionar, tenemos que asegurarnos de estar en la rama de destino (ejemplo, main). Para fusionar una rama (ejemplo, feature-rama) en la rama actual, ejecutamos el comando `git merge feature-rama`. Si la fusión es exitosa y necesitas compartirla usa `git push origin main`.

¿Cómo crear un commit en Git? ¿Cómo enviar un commit a GitHub?

Un commit en Git guarda los cambios en el historial del repositorio, ayudan a mantener un historial de cambios claro y organizado.

Antes de crear un commit puedes verificar los archivos modificados, nuevos o eliminados con `git status`.

Para agregar todos los archivos al área de preparación usamos `git add .` . Si solo necesitas agregar un archivo específico utilizamos `git add nombre-del-archivo`.
Una vez agregados los archivos, guardamos los cambios con `git commit -m "Breve descripción"`.

¿Qué es un repositorio remoto?

Es una versión de tu repositorio almacenada en un servidor externo, como **GitHub**. Permite colaborar con otros desarrolladores y mantener copias seguras en la nube, pudiendo acceder a tu código desde cualquier lugar.

¿Cómo agregar un repositorio remoto a Git?

Desde nuestro repositorio local, abrimos la terminal y navegamos hasta la carpeta del proyecto, si todavía no está inicializado Git, usa `git init`.
Para vincular el repositorio local con el remoto `git remote add origin URL-del-repositorio`.

¿Cómo empujar cambios a un repositorio remoto?

Empujar cambios a un repositorio remoto significa enviar tus commits desde el repositorio local a GitHub.
Para subir los cambios al repositorio remoto usamos `git push origin nombre-de-la-rama`.

¿Cómo tirar de cambios de un repositorio remoto?

Utilizamos el comando `git pull origin nombre-de-la-rama` para obtener (fetch) y fusionar (merge) los cambios del repositorio remoto a tu repositorio local.
Esto es esencial para mantener tu repositorio local sincronizado con los cambios realizados por otros en el repositorio remoto.
`git pull` es equivalente a `git fetch` seguido de `git merge`

¿Qué es un fork de repositorio?

Es una copia de un repositorio que se realiza en tu cuenta de GitHub. Permite que puedas realizar cambios en el código sin afectar el repositorio original. Son comúnmente usados en proyectos de código abierto. Un fork te da la oportunidad de trabajar en tu propia versión de un proyecto, y si tus cambios son útiles, puedes proponerlos al repositorio original.

¿Cómo crear un fork de un repositorio?

Primero debes ir al repositorio original en GitHub, el que quieres forkear. Haces click en el botón "Fork" y GitHub creará automáticamente una copia del repositorio en tu cuenta. Luego puedes clonar el fork y comenzar a trabajar en él.

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

Una pull request es una forma de proponer cambios de tu fork o rama en un repositorio hacia el repositorio original.
Una vez que tus cambios estén en tu fork o en una rama, ve al repositorio original al cual deseas contribuir.
Crea el Pull Request, verás un botón "Compare & pull request" o "New Pull Request" si estás usando otra rama. Completar algunos detalles como Título y Descripción, luego seleccionar la rama de destino y enviarlo.

¿Cómo aceptar una solicitud de extracción?

Cuando se recibe un Pull Request en un repositorio, significa que un colaborador ha propuesto cambios en su rama.

Primero, revisa el código propuesto, asegúrate de que los cambios no rompan el código, ver las sugerencias y comentarios, comprobar que los cambios resuelvan o mejoren el proyecto. Si es necesario, realiza pruebas en tu máquina local.

Una vez revisado y aprobado, tienes que fusionar haciendo clic en el botón "Merge pull request" y luego confirmas haciendo clic en "Confirm merge".

¿Qué es una etiqueta en Git?

Una etiqueta (tag) es una referencia que apunta a un punto específico en la historia del repositorio. Las etiquetas se utilizan comúnmente para marcar versiones de lanzamiento, lo que permite a los desarrolladores identificar y acceder fácilmente a esos puntos clave del proyecto.

Hay dos tipos: Etiquetas ligeras (lightweight tags) y Etiquetas anotadas (annotated tags).

¿Cómo crear una etiqueta en Git?

Para crear una etiqueta ligera que apunta al commit actual usa `git tag nombre-de-la-etiqueta` y para crear una etiqueta anotada usa `git tag -a nombre-de-la-etiqueta -m "Mensaje de la etiqueta"`.

¿Cómo enviar una etiqueta a GitHub?

Las etiquetas no se envían automáticamente al hacer git push. Para enviar una etiqueta al repositorio remoto, usa el comando `git push origin nombre-de-la-etiqueta`. Si deseas enviar todas las etiquetas que has creado en tu repositorio local al remoto, usa `git push --tags`.

¿Qué es un historial de Git?

El historial de Git es un registro de todos los cambios que se han realizado en un repositorio a lo largo del tiempo. Se almacena como una serie de commits, que representan momentos específicos en los que se realizaron modificaciones en el código.

¿Cómo ver el historial de Git?

Para ver el historial de commits en tu repositorio, usa el comando `git log`. Esto te mostrará una lista de todos los commits hechos en el repositorio, desde el más reciente hasta el más antiguo.

¿Cómo buscar en el historial de Git?

Para buscar en el historial, usamos el comando `git log` con diversas opciones de búsqueda, como por ejemplo por mensaje de commit (`git log --grep="mensaje de búsqueda"`), por autor (`git log --author="Nombre del autor"`), cambios en un archivo específico (`git log <ruta-del-archivo>`), en un rango de fechas (`git log --since="2025-03-01" --until="2025-03-25"`), entre otros.

¿Cómo borrar el historial de Git?

Al eliminar el historial, se elimina la información sobre los cambios realizados. Se debe hacer con cuidado y en un repositorio privado o si estás seguro que no necesitas esa información.

Si deseas borrar todos los commits y empezar de cero, crea un nuevo commit vacío, elimina las ramas anteriores, fuerza el push al repositorio remoto. Si solo deseas eliminar uno o más commits específicos y no todo el historial, puedes usar el comando `git rebase` (elimina) o `git reset` (edita).

¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un repositorio al que solo pueden acceder las personas que tú autorices. Mantiene el código y la información del proyecto confidencial y solo los colaboradores con permisos pueden ver, clonar y modificar el código.

¿Cómo crear un repositorio privado en GitHub?

Seguimos los mismos pasos de cuando creamos un repositorio y en la opción de visibilidad lo ponemos en privado para que solo nosotros y los colaboradores autorizados puedan acceder al mismo.

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

Si deseas compartir el repositorio con otras personas, debes ir a la pestaña "Settings" en el repositorio. Haz clic en "Collaborators", luego ingresa el nombre de usuario o correo de la persona que quieres invitar, haces clic en "Add collaborator" y la persona recibirá una invitación para acceder al repositorio.

¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un repositorio que cualquier persona en Internet puede ver y clonar. Sin embargo, solo los propietarios y colaboradores autorizados pueden realizar cambios en él. Estos, son ideales para proyectos de código abierto, compartir código con la comunidad, mostrar tu trabajo a posibles empleadores o clientes y colaborar con otros desarrolladores.

¿Cómo crear un repositorio público en GitHub?

Seguimos los mismos pasos de cuando creamos un repositorio y en la opción de visibilidad lo ponemos en público (Public) para que todos puedan acceder a él.

¿Cómo compartir un repositorio público en GitHub?

Una vez creado, puedes compartirlo de varias maneras: compartir el enlace (ejemplo: <https://github.com/tuusuario/tu-repositorio>), clonar el repositorio (ejemplo: `git clone https://github.com/tuusuario/tu-repositorio.git`), crear un fork o compartir en redes o comunidades.

Actividad 2:

<https://github.com/Angelrina/mi-repositorio-p1>

Actividad 3:

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