

TRABAJO PRÁCTICO 2 PROGRAMACIÓN 1

Franco Ghilardi

1. Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :
 - a. ¿Qué es GitHub?
GitHub es una plataforma de desarrollo colaborativo basada en la nube que permite a los desarrolladores alojar, administrar y compartir proyectos de software utilizando Git. Proporciona herramientas para el control de versiones, colaboración en equipo, gestión de problemas, integración continua y despliegue de código.
 - b. ¿Cómo crear un repositorio en GitHub?
Para crear un repositorio en GitHub, sigue estos pasos:
 1. Inicia sesión en **GitHub**.
 2. Haz clic en el ícono de tu perfil en la esquina superior derecha y selecciona "Your repositories".
 3. Presiona el botón "New".
 4. Ingresa un nombre para el repositorio y una descripción opcional.
 5. Selecciona si el repositorio será público o privado.
 6. Opcionalmente, puedes agregar un archivo README.md, una licencia y un archivo .gitignore.
 7. Haz clic en "Create repository".
 - c. ¿Cómo crear una rama en Git?
Para crear una nueva rama, usa: git branch nombre-de-la-rama
 - d. ¿Cómo cambiar a una rama en Git?
Para cambiar a una rama existente, usa: git checkout nombre-de-la-rama
 - e. ¿Cómo fusionar ramas en Git?
Para fusionar una rama en la actual, usa:
Fusionar rama:git merge nombre-de-la-rama
 - f. ¿Cómo crear un commit en Git?
Un commit es una instantánea del estado del código en un momento dado. Para crear un commit:
git add .
git commit -m "Mensaje descriptivo del cambio"
 - g. ¿Cómo enviar un commit a GitHub?
Después de realizar un commit, envíalo al repositorio remoto con:
git push origin nombre-de-la-rama
 - h. ¿Qué es un repositorio remoto?
Un repositorio remoto es una versión almacenada en un servidor (como GitHub, GitLab o Bitbucket) que permite la colaboración entre varios desarrolladores
 - i. ¿Cómo agregar un repositorio remoto a Git?
Para vincular un repositorio local con un remoto, usa: git remote add origin URL-del-repositorio
 - j. ¿Cómo empujar cambios a un repositorio remoto?
Para enviar los cambios al repositorio remoto, usa: git push origin nombre-de-la-rama
 - k. ¿Cómo tirar de cambios de un repositorio remoto?
Para descargar y fusionar cambios desde el repositorio remoto, usa: git pull origin nombre-de-la-rama

TRABAJO PRÁCTICO 2 PROGRAMACIÓN 1

Franco Ghilardi

- l. ¿Qué es un fork de repositorio?
**Un fork es una copia de un repositorio en la cuenta de otro usuario.
Permite modificar el código sin afectar al proyecto original.**
- m. ¿Cómo crear un fork de un repositorio?
**Visita el repositorio en GitHub.
Haz clic en el botón "Fork" en la parte superior derecha.
Selecciona tu cuenta para alojar el fork.**
- n. ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
**Desde tu proyecto o fork en GitHub, ve a la pestaña "Pull requests".
Haz clic en "New pull request".
Selecciona la rama con los cambios y compárala con la rama que deseas del repositorio original.
Agrega un título y descripción.
Haz clic en "Create pull request".**
- o. ¿Cómo aceptar una solicitud de extracción?
**Ve a la pestaña "Pull requests".
Selecciona la solicitud.
Revisa los cambios y haz clic en "Merge pull request".
Confirma con "Confirm merge".**
- p. ¿Qué es una etiqueta en Git?
Una etiqueta (tag) es una referencia a un punto específico en la historia del repositorio, generalmente utilizada para marcar versiones de software.
- q. ¿Cómo crear una etiqueta en Git?
Para crear una etiqueta: `git tag -a v1.0 -m "Versión 1.0"`
- r. ¿Cómo enviar una etiqueta a GitHub?
Después de crear la etiqueta, súbela con: `git push origin v1.0`
- s. ¿Qué es un historial de Git?
El historial de Git muestra los commits realizados en el repositorio, junto con información como autor, fecha y mensaje de cada cambio.
- t. ¿Cómo ver el historial de Git?
Para ver el historial de commits: `git log`
- u. ¿Cómo buscar en el historial de Git?
**Para buscar un commit específico: `git log --grep="palabra clave"`
Para buscar por autor: `git log --author="nombre"`**
- v. ¿Cómo borrar el historial de Git?
**Git no permite borrar el historial directamente, pero puedes sobrescribir con: `git rebase -i HEAD~n` # Donde n es el número de commits a modificar
O reiniciar completamente el historial (esto es irreversible): `rm -rf .git git init`**
- w. ¿Qué es un repositorio privado en GitHub?
Un repositorio privado es un repositorio que solo puede ser accedido por los colaboradores autorizados.
- x. ¿Cómo crear un repositorio privado en GitHub?
**Inicia sesión en GitHub y haz clic en "New repository".
Especifica el nombre y marca la opción "Private".
Haz clic en "Create repository".**

TRABAJO PRÁCTICO 2 PROGRAMACIÓN 1

Franco Ghilardi

- y. ¿Cómo invitar a alguien a un repositorio privado en GitHub?
Ve al repositorio.
Haz clic en "Settings" > "Collaborators".
Ingresa el nombre de usuario o correo electrónico y presiona "Add collaborator".
 - z. ¿Qué es un repositorio público en GitHub?
Un repositorio público es accesible para cualquier usuario de GitHub y puede ser clonado sin restricciones.
 - aa. ¿Cómo crear un repositorio público en GitHub?
Inicia sesión en GitHub y haz clic en "New repository".
Especifica el nombre y marca la opción "Public".
Haz clic en "Create repository".
 - bb. ¿Cómo compartir un repositorio público en GitHub?
Puedes compartir la URL del repositorio con otros usuarios.
2. Realizar la siguiente actividad:
- a. Crear un repositorio.
 - i. Dale un nombre al repositorio.
 - ii. Elige que el repositorio sea público.
 - iii. Inicializa el repositorio con un archivo.
 - b. Agregando un Archivo
 - i. Crea un archivo simple, por ejemplo, "mi-archivo.txt".
 - ii. Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.
 - iii. Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).
 - c. Creando Branchs
 - i. Crear una Branch
 - ii. Realizar cambios o agregar un archivo
 - iii. Subir la Branch

URL REPOSITORIO DE LA ACTIVIDAD

<https://github.com/FrancoGhilardi/E2-TP2-Programacion1/tree/main>

3. Realizar la siguiente actividad:
- a. Paso 1: Crear un repositorio en GitHub
 - i. Ve a GitHub e inicia sesión en tu cuenta.
 - ii. Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio.
 - iii. Asigna un nombre al repositorio, por ejemplo, conflict-exercise.
 - iv. Opcionalmente, añade una descripción.
 - v. Marca la opción "Initialize this repository with a README".
 - vi. Haz clic en "Create repository".
 - b. Paso 2: Clonar el repositorio a tu máquina local
 - i. Copia la URL del repositorio (usualmente algo como <https://github.com/tuusuario/conflict-exercise.git>).
 - ii. Abre la terminal o línea de comandos en tu máquina.

TRABAJO PRÁCTICO 2 PROGRAMACIÓN 1

Franco Ghilardi

- iii. Clona el repositorio usando el comando:
`git clone https://github.com/tuusuario/conflict-exercise.git`
- iv. Entra en el directorio del repositorio:
`cd conflict-exercise`
- c. Paso 3: Crear una nueva rama y editar un archivo
 - i. Crea una nueva rama llamada feature-branch:
`git checkout -b feature-branch`
 - ii. Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo:
Este es un cambio en la feature branch.
 - iii. Guarda los cambios y haz un commit:
`git add README.md`
`git commit -m "Added a line in feature-branch"`
- d. Paso 4: Volver a la rama principal y editar el mismo archivo
 - i. Cambia de vuelta a la rama principal (main):
`git checkout main`
 - ii. Edita el archivo README.md de nuevo, añadiendo una línea diferente:
Este es un cambio en la main branch.
 - iii. Guarda los cambios y haz un commit:
`git add README.md`
`git commit -m "Added a line in main branch"`
- e. Paso 5: Hacer un merge y generar un conflicto
 - i. Intenta hacer un merge de la feature-branch en la rama main:
`git merge feature-branch`
 - ii. Se generará un conflicto porque ambos cambios afectan la misma línea del archivo README.md.
- f. Paso 6: Resolver el conflicto
 - i. Abre el archivo README.md en tu editor de texto. Verás algo similar a esto:
`<<<<<<< HEAD`
Este es un cambio en la main branch.
`=====`
Este es un cambio en la feature branch.
`>>>>>>> feature-branch`
 - ii. Decide cómo resolver el conflicto. Puedes mantener ambos cambios, elegir uno de ellos, o fusionar los contenidos de alguna manera.
 - iii. Edita el archivo para resolver el conflicto y guarda los cambios (Se debe borrar lo marcado en verde en el archivo donde estés solucionando el conflicto. Y se debe borrar la parte del texto que no se quiera dejar).
 - iv. Añade el archivo resuelto y completa el merge:
`git add README.md`
`git commit -m "Resolved merge conflict"`
- g. Paso 7: Subir los cambios a GitHub
 - i. Sube los cambios de la rama main al repositorio remoto en GitHub:
`git push origin main`

TRABAJO PRÁCTICO 2 PROGRAMACIÓN 1

Franco Ghilardi

- ii. También sube la feature-branch si deseas:
`git push origin feature-branch`
- h. Paso 8: Verificar en GitHub
 - i. Ve a tu repositorio en GitHub y revisa el archivo README.md para confirmar que los cambios se han subido correctamente.
 - ii. Puedes revisar el historial de commits para ver el conflicto y su resolución.

**URL REPOSITORIO DE LA ACTIVIDAD (Pedir acceso si desean
revisar)**

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