

1)A) Github es una plataforma en línea donde los programadores pueden guardar y compartir su código. Funciona como una "nube" para proyectos de software, permitiendo que varias personas trabajen juntas en el mismo código sin pisarse los cambios.

B) Para crear uno, inicias sesión en GitHub, eliges un nombre para tu repositorio y decides si será público o privado. Luego, puedes subir archivos directamente desde la web o usar Git en tu computadora para gestionarlo. Con Git, puedes clonar el repositorio, hacer cambios en los archivos y enviarlos de vuelta a GitHub para mantener tu código organizado y accesible desde cualquier lugar.

C) Para crear una, primero asegúrate de estar en el repositorio correcto. Luego, usa el comando `git branch nombre-de-la-rama` para crearla y `git checkout nombre-de-la-rama` para cambiarte a ella. También puedes combinar ambos pasos con `git checkout -b nombre-de-la-rama`. Una vez que termines de trabajar en la rama, puedes fusionarla con la principal usando `git merge`. Esto permite desarrollar nuevas características sin afectar el código estable.

D) Usando el comando `Git checkout` podrás cambiarte de rama

E) Utilizando el comando `Git merge` puedes fusionar ambas ramas.

F) Creamos un commit en git utilizando el comando `git commit -m "Mensaje descriptivo"`

G) Enviamos un commit a github utilizando el comando `Git push`

H) Un **repositorio remoto** es una versión de tu repositorio que está almacenada en un servidor en línea, como GitHub, GitLab o Bitbucket. El repositorio remoto te permite compartir tu código con otras personas y acceder a él desde cualquier lugar.

I) Para agregar un **repositorio remoto** a tu proyecto en Git, primero debes crear un repositorio en una plataforma como GitHub y copiar la URL de ese repositorio. Luego, desde tu terminal, en el directorio de tu proyecto local, usas el comando `git remote add origin <URL-del-repositorio>` para vincular tu repositorio local con el remoto. Esto permite que puedas subir tus cambios y sincronizar tu código. Finalmente, puedes verificar que el repositorio remoto esté correctamente agregado usando `git remote -v`. Ahora, tu proyecto está listo para interactuar con el repositorio remoto.

J) Utilizando el comando `git push`

K) Utilizando el comando `git pull`

L) Un **fork** de repositorio es una copia personal de un repositorio que existe en una plataforma como GitHub.

M) Para crear un **fork** de un repositorio en GitHub, tenemos que hacer esto:

Ir al repositorio original que quieres forkear en GitHub.

En la parte superior derecha de la página del repositorio, haz clic en el botón **"Fork"**.

Selecciona tu cuenta o la organización a la que quieres que se copie el repositorio.

GitHub creará una copia del repositorio en tu cuenta.

N) Para **enviar una solicitud de extracción (pull request)** a un repositorio en GitHub, primero debes haber hecho un fork del repositorio original y haber realizado cambios en tu propia copia.

Ñ) Para **aceptar una solicitud de extracción (pull request)** en GitHub, sigue estos pasos:

1. **Ve al repositorio** donde se ha enviado la solicitud de extracción (pull request).
2. En la pestaña **"Pull requests"**, selecciona la solicitud que deseas revisar y aceptar.
3. Revisa los cambios propuestos para asegurarte de que todo esté correcto.
4. Si todo está bien, en la parte inferior de la página de la solicitud de extracción, verás un botón que dice **"Merge pull request"**. Haz clic en él.
5. GitHub te pedirá que confirmes la acción. Haz clic en **"Confirm merge"** para aceptar los cambios.

O) Una **etiqueta (tag)** en Git es una referencia que apunta a un punto específico en la historia de tu proyecto, generalmente usado para marcar versiones importantes, como lanzamientos de software o hitos.

P) Utilizando el comando `Git tag "Nombre de la etiqueta"`

Q) Utilizando el comando `git push origin "nombre-de-la-etiqueta"`

R) El **historial de Git** es una lista de todos los cambios realizados en un repositorio a lo largo del tiempo. Cada vez que haces un **commit**, Git guarda un registro de ese cambio, incluyendo información sobre qué archivos se modificaron, quién hizo el cambio, cuándo se hizo y un mensaje que describe el cambio.

S) Utilizando el comando `git log`

T) Utilizando el comando `git log --oneline`

U) Utilizando el comando git reset

V) La forma más sencilla de **borrar el historial de Git** es usar un **hard reset** y luego forzar un push al repositorio remoto.

W) Un **repositorio privado en GitHub** es un repositorio cuyo acceso está restringido solo a los usuarios que el propietario del repositorio haya autorizado.

X) Para **crear un repositorio privado en GitHub**, sigue estos pasos:

1. **Inicia sesión** en tu cuenta de GitHub.
2. En la página principal de GitHub, haz clic en el botón **"New"** o **"Nuevo"** (usualmente está en la esquina superior derecha).
3. En la sección de creación de repositorio, ingresa el **nombre** del repositorio.
4. Justo debajo del campo para el nombre, verás una opción llamada **"Visibility"** (Visibilidad). Allí selecciona la opción **"Private"** (Privado).
5. Opcionalmente, puedes agregar una **descripción** para el repositorio, inicializarlo con un archivo README o un archivo .gitignore, dependiendo de tus necesidades.
6. Haz clic en el botón **"Create repository"** (Crear repositorio).

Y) Para **invitar a alguien a un repositorio privado** en GitHub, sigue estos pasos:

1. **Accede a tu repositorio privado** en GitHub.
2. En la parte superior de la página del repositorio, haz clic en la pestaña **"Settings"** (Configuración).
3. En el menú de la izquierda, busca y selecciona la opción **"Manage access"** (Gestionar acceso).
4. Haz clic en el botón **"Invite a collaborator"** (Invitar a un colaborador).
5. Se abrirá una ventana donde puedes buscar el nombre de usuario de la persona que deseas invitar. Escribe su nombre de usuario en GitHub.
6. Una vez que aparezca el nombre de la persona, selecciona su cuenta y haz clic en **"Add"** (Agregar).

La persona recibirá una invitación para acceder al repositorio privado, y podrá aceptarla para obtener acceso. Después de aceptar, podrá ver y trabajar en el repositorio según los permisos que le hayas otorgado

Z) Un **repositorio público** es un lugar en línea donde se almacenan proyectos de código fuente y otros archivos que son accesibles para cualquiera.

Inicia sesión en tu cuenta de GitHub.

En la página principal de tu cuenta de GitHub, haz clic en el icono de tu perfil en la esquina superior derecha y selecciona **Your repositories** (Tus repositorios).

Haz clic en el botón **New** (Nuevo) para crear un nuevo repositorio.

-Para compartir un repositorio público en GitHub, simplemente tienes que compartir su enlace.

Ejercicio 2

<https://github.com/fedebikart/mi-repositorio.git>

Ejercicio 3

<https://github.com/fedebikart/resolver.git>