Apellido y nombre: Jimenez Ulises

¿Qué es GitHub?

GitHub es un servicio de alojamiento de repositorios de control de versiones, en el que los desarrolladores pueden almacenar y gestionar sus proyectos de software. Es una herramienta gratuita y de código abierto que permite a los usuarios trabajar juntos en proyectos, compartir código y colaborar de forma eficiente.

Ventajas de GitHub

Hay muchas ventajas de utilizar GitHub para control de versiones y colaboración, entre ellas:

* Facilidad de uso: GitHub es una herramienta muy fácil de usar, incluso para los desarrolladores que no están familiarizados con el control de versiones.
* Escalabilidad: GitHub puede escalar para satisfacer las necesidades de proyectos de cualquier tamaño, desde proyectos personales hasta proyectos de grandes empresas.
* Seguridad: GitHub ofrece una variedad de características de seguridad para proteger los proyectos de los usuarios.
* Comunidad: GitHub cuenta con una gran comunidad de desarrolladores que están dispuestos a ayudar y colaborar.

¿Cómo crear un repositorio?

Para crear un repositorio tendremos que ir a GitHub, una vez allí iremos hacia nuestro perfil que aparece en la esquina superior derecha y haremos click allí, entre esas opciones buscaremos la que dice “Yourrepositories”, allí veremos que hay una opción en verde que dice “New” al apretar allí empezaremos a crear nuestro repositorio por el nombre del repositorio. Al terminar la configuración del repositorio añadiremos el README.txt y lo crearemos. Y listo, ya tendrías tu repositorio.

¿Cuáles son los tipos de ramas en Git?

Los tipos de ramas en Git son:

* Local: La rama “Local” es la copia original del archivo o carpeta que quieras pasar al repositorio
* Remoto: La rama “Remoto” es la copia de la original para tener un tipo de “Copia de seguridad” del archivo

¿Cómo se crea una rama en Git?

Con el comando: “gitbranch [nombre de la rama]”

¿Cómo cambiar una rama en Git?

Con el comando: “gitcheckout” podremos cambiar de rama

¿Cómo fusionar las ramas?

Con el comando: “gitbranch –d” podremos fusionar las ramas

¿Cómo crear un commit en Git?

Con el comando: “gitcommit”

¿Cómo mandar un commit a GitHub?

Con el comando: “gitpush”

¿Qué es un repositorio remoto?

Un repositorio remoto es una “copia” del repositorio original que esta subido a la nube.

¿Cómo agregar un repositorio remoto a git?

Con el comando “gitremoteaddorigin<url de github>”

¿Cómo empujar cambios al repositorio remoto?

Con el comando “gitpush –u origin master”

¿Cómo tirar de cambios de un repositorio remoto?

Con el comando “gitpullorigin<nombre de la rama>”

¿Qué es un fork de repositorio?

Un fork de un repositorio es una copia independiente de un repositorio de código fuente en un sistema de control de versiones, como Git.

¿Cómo crear un fork de un repositorio?

Con el comando “git clone” se crea un fork de un repositorio.

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

Ve a tu repositorio en GitHub y verás un botón llamado "Pull request", has clic en él

¿Cómo aceptar una solicitud de extracción?

En la lista de “Pull Requests”, haz clicl en la solicitud de extracción que quieras aceptar.

¿Qué es un etiqueta en Git?

Las etiquetas son referencias que apuntan a puntos concretos en el historial de Git.

¿Cómo crear una etiqueta en Git?

Haz click en incidencias, a continuación en Labels, a la derecha del campo de búsqueda, haga click en Nueva etiqueta.

¿Cómo enviar una etiqueta a GitHub?

Con el comando “git push origin [etiqueta]”

¿Qué es un historial de Git?

El historial de Git es el registro de todos los cambios en un repositorio a lo largo del tiempo. Se muestra como una secuencia de commits interconectados que muestran quién hizo cambios, cuándo y qué se cambió. El historial sigue una línea de desarrollo y puede ramificarse en nuevas funcionalidades. Las fusiones y etiquetas también son parte del historial. Es esencial para rastrear la evolución del proyecto y colaborar en equipo.

¿Cómo ver el historial de Git?

Con el comando “git log” podras ver el historial

¿Cómo buscar en el historial de Git?

Hay varias maneras de buscar por el historial con el comando “git log”

1. Búsqueda por mensaje de commit\* Usar `git log --grep="palabra\_clave"` para buscar commits con ciertas palabras en el mensaje.

1. 2. Búsqueda por autor: Usar `git log --author="nombre\_del\_autor"` para encontrar commits de un autor específico.
2. Búsqueda por rango de fechas: Usar `git log --since="fecha\_inicio" --until="fecha\_fin"` para buscar commits dentro de un rango de fechas.
3. Búsqueda por archivos modificados: Usar `git log -- archivo\_modificado.txt` para buscar commits que afecten un archivo en particular.

¿Cómo borrar el historial de Git?

Yendo a la carpeta raíz del historial y borrando la carpeta .git

¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un espacio donde puedes almacenar proyectos de código y archivos de manera segura y privada. Solo las personas que invites y autentiquen con sus cuentas de GitHub pueden acceder y colaborar en el repositorio. Es ideal para proyectos confidenciales, trabajos en equipo restringidos y para proteger la propiedad intelectual. Está disponible en planes de pago de GitHub y ofrece opciones de control de acceso y colaboración.

¿Cómo se crea un repositorio privado en GitHub?

1. Inicia sesión en tu cuenta de GitHub.

2. Ve a "Your repositories".

3. Haz clic en "New".

4. Completa los detalles del nuevo repositorio y marca "Private".

5. Opcionalmente, selecciona opciones de inicialización.

6. Crea el repositorio.

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

En el nombre del repositorio haz click en configuración, a continuación en la sección “Acceso” y allí haz click en colaboradores y para terminar en Agregar personas.

¿Qué es un repositorio público en GitHub?

Los repositorios públicos son repositorios abiertos a toda persona que desee ver los códigos y para compartir software de código abierto.

¿Cómo crear un repositorio publico en GitHub?

Para crear un repositorio público en GitHub:

1. Inicia sesión en tu cuenta de GitHub.

2. Ve a "Your repositories".

3. Haz clic en "New".

4. Completa los detalles del nuevo repositorio.

5. Asegúrate de que "Private" no esté marcado (debe estar desmarcado para que sea público).

6. Opcionalmente, selecciona opciones de inicialización.

7. Crea el repositorio.

¿Cómo compartir un repositorio público en GitHub?

Compartiendo el link del repositorio ya estaría compartido a la otra persona a la cual esta destinada.

¿Qué es un repositorio de código en GitHub?

Es el lugar en el que se almacena y se puede realizar**la distribución del código de una aplicación o un programa**. Este debe ser un servidor seguro que utiliza sistemas de control de versiones.

¿Cómo compartir un repositorio de código en GitHub?

Para compartir un repositorio de código en GitHub:

1. Crea un repositorio: Inicia sesión en GitHub, ve a "Your repositories" y crea un nuevo repositorio.

2. Configura la visibilidad: Al crear el repositorio, elige "Public" (público) o "Private" (privado) según tus necesidades.

3. Invita colaboradores: En la pestaña "Settings" (Configuración) de tu repositorio, ve a "Manage access" (Gestionar acceso) y agrega colaboradores por sus nombres de usuario o direcciones de correo electrónico.

4. Comparte el enlace: Proporciona el enlace del repositorio a tus colaboradores para que puedan clonar, hacer cambios y colaborar.

5. Utiliza ramas y solicitudes de extracción: Para colaborar de manera efectiva, crea ramas para características o correcciones, y usa solicitudes de extracción para revisar y fusionar cambios.

6. Establece reglas: Si es un proyecto colaborativo, establece reglas y pautas de colaboración para mantener un flujo de trabajo ordenado.

¿Qué es un repositorio de documentación en GitHub?

Un repositorio de documentación es un repositorio que contiene todos los archivos de tu proyecto y el historial de revisiones de cada uno de ellos.

¿Cómo crear un repositorio de documentación en GitHub?

Para crear un repositorio de documentación en GitHub:

1. Inicia sesión: Accede a tu cuenta de GitHub.

2. Crea un repositorio: Ve a "Your repositories" y haz clic en "New" (Nuevo).

3. Configura el repositorio: Ingresa un nombre y descripción para tu repositorio. Puedes marcarlo como público o privado según tus necesidades.

4. Incluye archivos de documentación: Agrega archivos de documentación, como Markdown o archivos de texto, que contengan la información necesaria.

¿Cómo compartir un repositorio de documentación en GitHub?

Copiando el link y enviándolo a quienes queremos compartirlo.

¿Qué es un repositorio de proyectos en GitHub?

Un repositorio de proyecto en GitHub es un espacio en línea donde se almacenan y gestionan archivos, código fuente y recursos relacionados con un proyecto específico. Puede contener código, documentación, imágenes y otros archivos. GitHub ofrece herramientas para colaborar, rastrear cambios y administrar el desarrollo y la evolución de un proyecto de manera efectiva. Puede ser público (visible para todos) o privado (accesible solo para colaboradores autorizados).

¿Cómo usar git para trabajar con otros?

**Paso 1**

Cada programador miembro del equipo debe tener una copia del proyecto de la rama **Dev**, porque con esa copia es con la que van a trabajar. El proceso de sacar una copia del repositorio se llama clonar y se realiza con el comando git clone

Con ese código clonado, cada uno de los miembros del equipo debe trabajar en una funcionalidad

## ****Paso 2****

Para que cada uno trabaje individualmente en su repositorio dentro de un entorno seguro, cada programador debe sacar una rama particular (branch), de esta manera su trabajo no se mezcla con el de los demás. El comando para hacerlo es git branch

✅ Para cada funcionalidad en la que estés trabajando debes crear una rama, así no mezclas tu trabajo. Es importante trabajar dentro de un entorno seguro: si algo falla, dejará de funcionar solamente en esa rama sin comprometer al resto.

## ****Paso 3****

Cuando termines todas tus tareas, debes unir tu rama con la rama de desarrollo principal (La rama **Dev**), para que tu código esté disponible para el resto de los programadores. Esto se hace con el comando git merge.

Ojito aquí: Cuando envías tu trabajo a la rama principal, deja de estar en ese entorno seguro. Debes tener cuidado porque si tu código tiene problemas, **será un caos.**

Por eso hay muchos memes en internet acerca del comando git merge. Si tu código no es correcto o no sigue los estándares, el merge tendrá conflictos.

Pero calma, a pesar de que es tedioso, estos conflictos se pueden solucionar revisando el código línea por línea, o aún mejor, a través de test automatizados que devuelven el código si encuentran conflictos

## ****Paso 4****

Si tu código pasa las pruebas y se integra correctamente a la rama **Dev**, significa que pasará a la rama principal a través de un Merge Request en GitLab (MR), o también conocido como Pull Request en GitHub (PR). Si tu código llegó a Main, ya está listo para pasar a producción ¡Bravo!

GitHub: https://github.com/Fl1XEvac/Laboratorio\_1