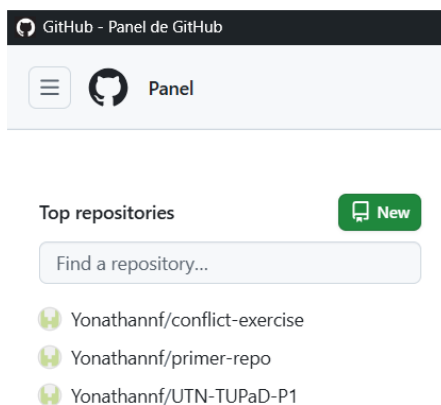


Actividad N1

- ¿Qué es GitHub?

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir, trabajar junto con otros usuarios para escribir código siendo perfecto para proyectos grupales ya que tienen varias funciones que ayuda al usuario en su proceso como dividir el proyectos en varias ramas con varias modificaciones y también poder unir sus ideas.

- ¿Cómo crear un repositorio en GitHub?



1.iniciar sesión en tu cuenta de GitHub

2.hacer click en New para generar tu nuevo repositorio

3.configurar el nuevo repositorio con un nombre, descripción, si será publico o privado

Una vez que esta todo como nos parezca, damos clic en Créate Repositorio. Nos saldrán varias líneas de comandos como:

...or create a new repository on the comand line: Nos indica cada uno de los pasos que debemos de hacer para poder mandar desde nuestro repositorio local al repositorio que hemos creado en github y enlazarlos

...or push an existing repository from the command line: Este caso hace referencia a que ya tienes un repositorio local y solamente deseas mandarlo a la plataforma. Para ello deberás de tipear los siguientes comandos en el bash:

```
$ git remote add origin https://github.com/repositorio_nuevo.git
```

```
$ git push -u origin master
```

luego refrescar la página de github y podrás ver como se ha subido nuestro repositorio.

- ¿Cómo crear una rama en Git?

Las ramas en Git permiten trabajar en nuevas características o correcciones de errores sin afectar la versión principal del código. Esto es útil para el desarrollo colaborativo y la gestión de versiones en un proyecto.

Para crearlas debemos ejecutar el siguiente comando:

```
git branch nueva_rama
```

aplicando el comando 'git branch' y agregamos el nombre de la nueva rama

- ¿Cómo cambiar a una rama en Git?

Cambiar de rama en Git te permite trabajar en diferentes versiones o funcionalidades del proyecto sin alterar otras partes del código. Es una de las funciones más poderosas de Git para el trabajo en equipo y el desarrollo organizado.

Con el siguiente comando "git Branch" podremos ver las distintas ramas que tengamos en nuestro repositorio

Con el comando "git checkout nombre-de-la-rama-a-cambiar" podemos movernos a la rama que deseamos

- ¿Cómo fusionar ramas en Git?

Fusionar ramas en Git es el proceso de combinar los cambios de una rama secundaria (por ejemplo, una rama de desarrollo o corrección) con otra rama principal (como main o master). Este proceso se conoce como **merge** y es fundamental para integrar trabajo colaborativo o nuevas funcionalidades sin perder historial de cambios.

Los pasos para fusionar ramas en Git son:

1.cambiar a la rama en la que deseamos fusionar con el comando "git checkout main (por dar un ejemplo ya que podría ser cualquier nombre)"

2.luego ejecutamos el siguiente comando para fusionar la rama siguiente (por ejemplo una rama llamada nueva_rama) en main el cual deberíamos escribir "git merge nueva_rama"

3.Resolver conflicto (si es que hay) por ejemplo si en ambas cambios afectaron las mismas líneas de código, Git genera un conflicto en el que deberíamos editar el

archivo involucrados para resolverlo manualmente, luego guardar y confirmar los cambios con los siguientes comandos

Git add archivo_resulto (corregimos de manera manual el archivo)

Git commit (guardamos el archivo)

- ¿Cómo crear un commit en Git?

Un **commit** en Git representa un “punto de guardado” de los cambios en tu proyecto como cuando estamos jugando y queremos guardar nuestra partida para que podamos volver a ese momento

Los pasos para crear un commit son:

1.abrir nuestra terminal y entrar al proyecto que estamos realizando

Cd nombre del proyecto

2.antes de realizar una commit debemos realizar el siguiente comando

Git add nombre_archivo.txt

3.despues de añadir los archivos creamos el commit

Git commit -m “algún mensaje de nuestros cambios realizados”

- ¿Cómo enviar un commit a GitHub?

Una vez creado nuestra commit podemos enviarlo al repositorio remoto con los siguientes pasos:

1.debemos asegurarnos en saber en que rama estamos antes de enviarlo

Git Branch (la rama actual estará marcada con el *)

2.asegurarnos de haber hecho el guardado y el commit de nuestro proyecto

Git add .

Git commit -m “nombre de nuestro cambios realizados”

3.ejecutamos el siguiente comando con el cual podremos subirlo a nuestro repositorio remoto

Git push origin nombre_de_la_nueva_rama

¿Qué es un repositorio remoto?

Es una copia de nuestro proyecto que estará alojada en la nube de Git Hub en donde podemos podremos compartirlo ya sea para un trabajo grupal o solo para que lo vean las demás personas, además podremos agregar permiso de lectura o modificación y lectura dependiendo de lo que deseamos hacer con nuestro repositorio remoto

Además es excelente para los trabajos grupales en donde cada participante podrá subirlo a la red con distintas ramas del proyecto, ver quien realizo los cambios del proyecto y poder organizarse de una forma muy eficiente respecto al proyecto al realizar

- ¿Cómo agregar un repositorio remoto a Git?

Para poder agregar nuestro repositorio remoto a Git debemos realizar los siguientes pasos:

1.abrir nuestra terminal y entrar al proyecto

Cd ruta_del_proyecto

2.agregar el repositorio remoto con Git

git remote add origin https://github.com/usuario/nuevo_proyecto.git

3.verificamos que se haya agregado correctamente

Git remote -v

Deberíamos ver una lista de todos los remotos configurados con sus urls

- ¿Cómo empujar cambios a un repositorio remoto?

Para empujar los cambios a nuestro repositorio remoto primero deberíamos chequear que nuestros archivos no tenga ningún conflicto entre si y luego de haber hecho este paso ejecutamos el siguiente comando

git push <nombre del repositorio> <nombre de la rama>

- ¿Cómo tirar de cambios de un repositorio remoto?

Hacer un pull en Git significa descargar los últimos cambios desde el repositorio remoto hacia tu repositorio local. Es una forma de mantener tu proyecto actualizado con los cambios hechos por otros colaboradores o por ti mismo desde otra máquina.

Los pasos para tirar de cambios desde un repositorio remoto son:

1.asegurarse de estar en la rama correcta

Git checkout main (o cualquier otro nombre de nuestra rama creada)

2.tirar de los cambios desde el repositorio remoto

Git pull origin main

- ¿Qué es un fork de repositorio?

Un **fork** es una copia de un repositorio en tu cuenta de GitHub que te permite experimentar y proponer cambios sin afectar el original.

- ¿Cómo crear un fork de un repositorio?

Hacer un fork es muy útil cuando quieres trabajar sobre un proyecto que no te pertenece, especialmente si planeas hacer mejoras o colaboraciones.

Los pasos que debemos realizar son:

1.iniciar sesión en Github

2.busca el repositorio que deseamos forkar

3.hacer click en el botón fork en donde Git nos preguntara si queremos forkarlo en nuestra cuenta personal o alguna organización

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

Una Pull Request (PR) es una manera de proponer cambios a un repositorio. Es muy útil cuando haces un fork de un proyecto o cuando estás trabajando en una rama distinta dentro del mismo repositorio

Para hacer el pull request nos dirigiremos a la solapa de Pull requests allí hacemos click en new pull request, en donde saldra una ventana a modo de resumen en donde se reflejarán los cambios que hemos hecho nosotros en comparación al repositorio original. hacemos click en Create pull request donde veremos el asunto (colocamos algún mensaje global) y deslizando para abajo tenemos suficiente lugar para poder explayarnos en mencionar el porque ese cambio que hemos realizado nosotros, sería considerado como algo que a el repositorio original le vendrían bien agregarlo.

- ¿Cómo aceptar una solicitud de extracción?

Aceptar una solicitud de extracción (pull request) en GitHub es un proceso sencillo si tienes permisos para hacerlo (por ejemplo, si eres colaborador del repositorio o el propietario).

Para aceptar una solicitud deberíamos realizar los siguientes pasos:

1. En la solicitud de extracción, haz clic en Archivos modificados
2. Revisa los cambios en un archivo por vez
3. Deja comentarios en los cambios específicos
4. Marca cada archivo como visto

Paso 2 Aprueba la solicitud

1. Haz clic en Revisar cambios
2. Escribe un comentario que resuma tu opinión sobre los cambios propuestos
3. Haz clic en Aprobar
4. Haz clic en Enviar revisión

- ¿Qué es un etiqueta en Git?

Una etiqueta en Git es una referencia fija a un commit específico. Se usa principalmente para marcar versiones importantes del código, como por ejemplo un lanzamiento de software.

- ¿Cómo crear una etiqueta en Git?

Crear una etiqueta (tag) en Git es muy fácil, y puedes hacerlo de dos formas principales: ligera o anotada, dependiendo de si quieres solo marcar el commit o también agregarle información como un mensaje.

Etiqueta ligera: Este tipo de etiqueta solo apunta a un commit, sin metadatos (autor, fecha, mensaje, etc.)

```
git tag v1.0.0
```

etiqueta anotada: Contiene información adicional (autor, fecha, mensaje)

```
git tag -a v1.0.0 -m "Versión 1.0.0 - primera release estable"
```

-a es para "annotated"

-m permite añadir un mensaje

- ¿Qué es un historial de Git?
El historial de Git es un registro de los cambios realizados en un repositorio, almacenado como un gráfico de confirmaciones.
- ¿Cómo ver el historial de Git?

Para ver el historial de Git, puedes usar el comando `git log`. También puedes usar la vista de actividad en GitHub.

Usar el comando `git log`

`git log` muestra las confirmaciones de un repositorio en orden cronológico inverso.

Para ver el historial de confirmaciones de la rama actual, puedes usar `git log --oneline`.

Para ver el historial de confirmaciones en un gráfico ASCII, puedes usar `git log --oneline --graph`.

Para filtrar los resultados, puedes agregar opciones como `--since="6 days"` para mostrar solo los commits de los últimos 6 días.

Usar la vista de actividad en GitHub muestra un historial detallado de cambios en un repositorio, muestra todas las inserciones, combinaciones, inserciones forzadas y cambios de rama.

Asocia estos cambios con confirmaciones y usuarios autenticados.

Ver el historial de un archivo en GitHub

Ve a la página principal del repositorio.

Haz clic para abrir el archivo cuyo historial de líneas quieres ver.

Sobre el contenido del archivo, haz clic en "Atribuir".

- ¿Cómo buscar en el historial de Git?

Para buscar en el historial de Git, puedes usar los comandos `git log` y `git grep`. También puedes usar la vista de actividad en GitHub.

Comando `git log`

Muestra el historial de commits

Puedes usar opciones para personalizar la salida, como `--stat`, `--name-status`, `-graph`, `--author`, y `--grep`

Para ver el historial de una función o línea de código, usa `git log -L`

Para ver el historial de cambios de un archivo, usa `git log -- [filename]`

Comando `git grep`

Busca un patrón en todo el historial de commits

Usa `git grep <pattern> $(git rev-list --all)`

Vista de actividad en GitHub

Muestra un historial detallado de cambios en el repositorio

- ¿Cómo borrar el historial de Git?

Para borrar el historial de Git localmente, puedes borrar la carpeta `.git`. También puedes reescribir la historia del repositorio para eliminar un archivo o datos confidenciales.

También podemos usar los siguientes comandos para realizarlo

`git reset ->` Quita del stage todos los archivos y carpetas del proyecto.

`git reset nombreArchivo ->` Quita del stage el archivo indicado.

`git reset nombreCarpeta/ ->` Quita del stage todos los archivos de esa carpeta.

`git reset nombreCarpeta/nombreArchivo ->` Quita ese archivo del stage (que a la vez está dentro de una carpeta).

`git reset nombreCarpeta/*.extensión ->` Quita todos los archivos que cumplan con la condición indicada previamente dentro de esa carpeta del stage.

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado es un repositorio al que solo nosotros y las personas que nosotros autorizamos pueden acceder y no es visible públicamente.

- ¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado en GitHub debemos realizar los siguientes pasos:

1. Ir a la esquina superior derecha de cualquier página en GitHub
2. Seleccionar Nuevo repositorio
3. Escribir un nombre para el repositorio
4. Añadir una descripción opcional
5. Elegir la visibilidad del repositorio
6. Hacer clic en Crear repositorio

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

Para invitar a alguien a un repositorio privado en GitHub, debemos seguir estos pasos:

1. Ir a la página principal del repositorio
2. Hacer clic en Configuración
3. En la barra lateral, hacer clic en Colaboradores y equipos
4. Debajo de Administrar acceso, hacer clic en Agregar personas
5. Escribir el nombre de usuario, el nombre completo o la dirección de correo electrónico de la persona a la que se quiere invitar
6. Elegir un rol de repositorio
7. Hacer clic en Agregar NOMBRE al REPOSITORIO

- ¿Qué es un repositorio público en GitHub?

Los repositorios públicos de GitHub se suelen utilizar para compartir software de código abierto. Para que tu repositorio sea verdaderamente de código abierto, tendrás que generarle una licencia. De este modo, las demás personas podrán usar, modificar y distribuir el software con libertad.

- ¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio público en GitHub, debemos realizar los siguientes pasos:

1. Ir a [GitHub](#)
2. En la esquina superior derecha, seleccionar Nuevo repositorio
3. Escribir un nombre para el repositorio
4. Añadir una descripción opcional
5. Elegir la visibilidad pública
6. Seleccionar Inicializar este repositorio con un README
7. Hacer clic en Crear repositorio

- ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio público en GitHub, puedes invitar a colaboradores a tu repositorio. Podemos hacerlo directamente en GitHub o por correo electrónico.

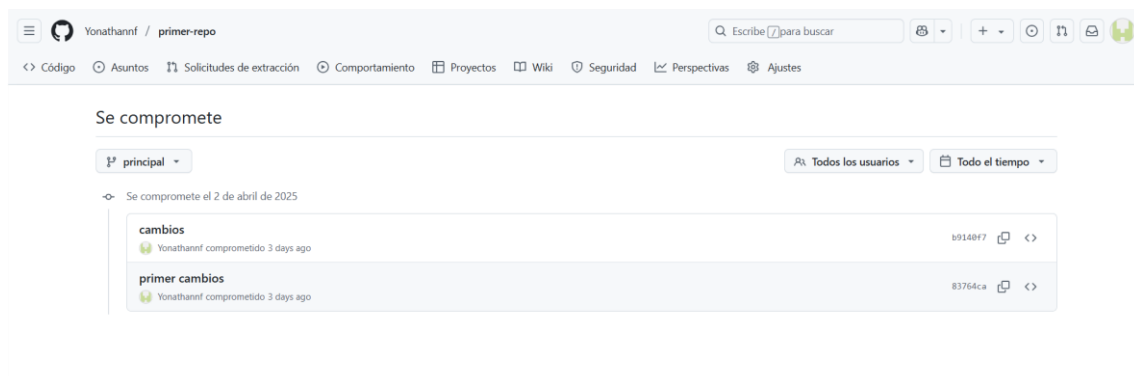
Para invitar a colaboradores a un repositorio:

1. Ve a la página principal del repositorio
2. Haz clic en Configuración
3. En la barra lateral, haz clic en Colaboradores y equipos
4. Haz clic en Agregar personas o Agregar equipos
5. Escribe el nombre de la persona a la que quieres invitar
6. Haz clic en un nombre de la lista de coincidencias
7. Haz clic en Añadir NOMBRE al REPOSITORIO

La persona invitada recibirá un correo electrónico. Una vez que acepte la invitación, tendrá acceso de colaborador a tu repositorio.

GitHub limita la cantidad de personas a las que se puede invitar a un repositorio en un período de 24 horas.

2) segunda actividad:



Se creo el repositorio publico primer-repo cargando un archivo de formato txt

<https://github.com/Yonathannf/primer-repo.git>

3) tercera actividad

Yonathannf / ejercicio de conflicto

Escribe para buscar

Código

Asuntos

Solicitudes de extracción

Comportamiento

Proyectos

Wiki

Seguridad

Perspectivas

Ajustes

ejercicio de conflicto

Público

Añadir

Dejar de ver 1

Tenedor 0

Estrella 0

principal

2 sucursales

0 etiquetas

Ir al archivo

Agregar archivo

Código

Yonathannf

Conflicto de fusión resuelto

7ae3a73 · Hace 3 días

4 confirmaciones

README.md

Conflicto de fusión resuelto

Hace 3 días

LÉAME

ejercicio de conflicto

actividad 3

Nuevo cambio realizado

Este es un cambio en la rama característica.

Acerca de

actividad 3

Léame

Actividad

0 estrellas

1 mirando

0 tenedores

Lanzamientos

No hay comunicados publicados

[Crear un nuevo lanzamiento](#)

Paquetes

No hay paquetes publicados

[Publica tu primer paquete](#)

Se compromete

principal

Todos los usuarios

Todo el tiempo

Se compromete el 2 de abril de 2025

Conflicto de fusión resuelto

Yonathannf comprometido 3 days ago

7ae3a73

<>

Se agregó una línea en la rama principal

Yonathannf comprometido 3 days ago

d1e694c

<>

Se agregó una línea en la rama de características

Yonathannf comprometido 3 days ago

3d32614

<>

Compromiso inicial

Yonathannf de autoría 3 days ago

Verified e8222b2

<>

<https://github.com/Yonathannf/conflict-exercise.git>

