Práctico 2: Git y GitHub
Ezequiel Pereyra

¿Qué es GitHub?
es una comunidad donde nosotros podemos crear y compartir nuestros repositorios de forma publica

¿Cómo crear un repositorio en GitHub?
ingresamos a https://github.com/ (luego de haber creado una cuenta)
arriba a la derecha hay un simbolo “ + “ , presionamos y seleccionamos “New repository”
le asignamos un nombre como por ejemplo, “miPrimerRepositorio” , seleccionamos si sera publico o privado y luego clickeamos en “Create repository”

¿Cómo crear una rama en Git?
abrimos la consola de git haciendo click derecho en la carpeta donde estamos trabajando en nuestro proyecto y seleccionamos “Open git bash here”, opcionalmente puede ejecutarse desde la terminal de visual studio code ejecutando el comando git branch rama(como nombre de ejemplo)

¿Cómo cambiar a una rama en Git?
con el comando git checkout rama

¿Cómo fusionar ramas en Git?
con el comando git merge rama fusionar con rama, en rama actual

¿Cómo crear un commit en Git?
git commit -m "mensaje" Confirmar los cambios

¿Cómo enviar un commit a GitHub?
git push origin master

¿Qué es un repositorio remoto?
un repositorio remoto es una copia del repositorio local, que se almacena en un servidor, (GitHub) estos repositorios permiten que tanto yo como otras personas accedan al codigo desde cualquier lugar , esto permite colaborar en algun proyecto y se mantengan sincronizadas las versiones del codigo

¿Cómo agregar un repositorio remoto a Git?
con el comando git remote add origin url se añade un nuevo repositorio remoto

¿Cómo empujar cambios a un repositorio remoto?
con el comando git push -u origin master

¿Cómo tirar de cambios de un repositorio remoto?
con el comando git pull origin main se aplican los cambios de un repositorio remoto al repositorio local

¿Qué es un fork de repositorio?
un fork es una copia de un repositorio creada en una cuenta diferente permitiendo desarrollar cambios sin afectar el original, a diferencia del clonado, que descarga el repositorio localmente, el fork se realiza generando una copia en la cuenta del usuario

¿Cómo crear un fork de un repositorio?
iniciamos sesion con nuestra cuenta, luego ingresamos al link de programacion 1
y vamos al boton Fork en la parte superior derecha de la pagina del repositorio

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

1. **Asegurarse de que el repositorio esté sincronizado:**

   -Realiza un commit y sube tus cambios a la rama en tu fork usando el comando: git push origin nombreRama

2. **Ir a www.github.com y abrir el repositorio fork:**

   -Accede a tu cuenta y selecciona el repositorio donde realizaste los cambios.

3. **Hacer click en “Compare & pull request”:**

   -GitHub mostrará esta opción si detecta cambios en tu fork. Al hacer click, se abrirá un formulario para crear la solicitud de extracción.

4. **Rellena los detalles de la solicitud:**

   -Asegúrate de que la rama de tu fork se compare con la rama original del repositorio al que deseas enviar la solicitud.

   Escribe los cambios realizados en el mensaje, explicando qué hiciste y por qué son importantes.

5. **Haz click en “Create pull request”:**

   -Esto enviará tu solicitud al propietario del repositorio original, quien podrá revisarla, discutir cambios y aceptarla si todo está en orden.

6. **Revisa tu solicitud:**

   Puedes seguir el estado de tu pull request en la pestaña “Pull request” del repositorio original.

¿Cómo aceptar una solicitud de extracción?

1. **Accede al repositorio de GitHub:**

   -Ve al repositorio donde se realizó la solicitud de extracción.

2. **Dirigirse a la pestaña “Pull requests”:**

   -En la parte superior del repositorio, haz click en la pestaña “Pull requests” para ver todas las solicitudes enviadas.

3. **Selecciona la solicitud de extracción:**

   -Haz click en el título de la pull request que deseas revisar.

4. **Revisa los cambios propuestos:**

   -Examina los archivos modificados y el código que se agrega o cambia.

   Si tienes dudas o sugerencias de ajustes, puedes agregar comentarios antes de aceptarla.

5. **Acepta la solicitud de extracción:**

   -Si todo está en orden, haz click en el botón “Merge pull request”. Esto fusionará los cambios en la rama de destino.

6. **Confirma el merge:**

   -Aparecerá un mensaje pidiéndote confirmar. Haz click en “Confirm merge”.

7. **(Opcional) Elimina la rama:**

   -Después de realizar el merge, GitHub te da la opción de eliminar la rama que se utilizó para la solicitud de extracción. Puedes hacerlo si ya no la necesitas.

¿Qué es un etiqueta en Git?

Una etiqueta (o tag) en Git es una referencia que apunta a un commit específico en el historial. Las etiquetas se utilizan comúnmente para marcar puntos importantes en el desarrollo de un proyecto, como versiones de software (por ejemplo, v1.0, v2.1.3, etc.). Hay dos tipos principales de etiquetas en Git:

Etiquetas ligeras (lightweight):

Simplemente actúan como un apuntador a un commit específico.

Son más básicas y no almacenan metadatos adicionales (como nombre del autor o fecha)

etiquetas anotadas(annotated)incluyen informacion adicional, como un mensaje, el nombre del autor y la fecha.

son mas recomendadas para marcar versiones oficiales porque preservan mas informacion sobre el contexto

¿Cómo crear una etiqueta en Git?
git tag nombreDeEtiqueta

¿Cómo enviar una etiqueta a GitHub?
git push origin nombreDeLaEtiqueta

¿Qué es un historial de Git?
Un historial en Git es el registro completo de los cambios realizados en tu proyecto. Cada vez que haces un commit, Git guarda información sobre esos cambios, como el autor, la fecha, y el mensaje del commit. Esto te permite rastrear quién hizo qué y cuándo, facilitando la colaboración y el manejo de versiones.

¿Cómo ver el historial de Git?
con el comando git log
Este comando muestra información detallada sobre cada commit, incluyendo:
El identificador único del commit (hash).
Fecha y hora del commit.
Autor del commit.
Mensaje descriptivo del commit.

¿Cómo buscar en el historial de Git?
Para buscar en el historial de Git, puedes utilizar varios comandos dependiendo de lo que necesites encontrar:

Buscar texto en los mensajes de los commits: Utiliza el siguiente comando para buscar palabras clave en los mensajes de los commits:
git log --grep="palabra clave"

Buscar cambios específicos en el contenido: Si necesitas buscar una línea de código o texto en los archivos modificados de los commits, usa:

git log -S "texto a buscar"

¿Cómo borrar el historial de Git?
Borrar el historial completo de Git: esto elimina todos los commits de tu repositorio. hay que tener cuidado porque esto es irreversible
rm -rf .git elimina completamente la carpeta .gut de un proyecto, lo que equivale a borrar todo el historial de git asociado con ese repositorio, esto incluye todos los commits, ramas y configuraciones de control de versiones

¿Qué es un repositorio privado en GitHub?
Un repositorio privado en GitHub es un espacio de trabajo donde se puede almacenar tu código, archivos y proyectos de manera segura y accesible únicamente para uno mismo y las personas que invites. A diferencia de los repositorios públicos, los privados no son visibles para el público ni pueden ser encontrados por usuarios que no tengan permiso. Esto

es ideal cuando trabajas en proyectos confidenciales, en desarrollo o que aún no se desea compartir públicamente.

¿Cómo crear un repositorio privado en GitHub?
se inicia sesion en GitHub , hacemos click en el boton + y seleccionamos create new repository, ingresamos el nombre y luego seleccionamos la opcion “Private”

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

1. iniciamos sesion en GitHub , en tu lista de repositorios, selecciona el repositorio privado al que desees invitar a alguien
2. una vez dentro del repositorio has click en la pestaña “Settings” ubicada en la parte superior de la página
3. selecciona “Collaborators and teams”.
4. en la seccion de colaboradores, haz clic en el botón “Add people”
   ingresa el nombre del usuario o el correo electronico de la persona que deseas invitar
5. asigna permisos(opcional):
   se puede seleccionar el nivel de acceso que tendra el colaborador(por ejemplo lectura, escritura o administrador)
6. envia la invitacion:
   hacer click en el boton “Send Invitation”. la persona recibira una notificacion en su cuenta de gitHub para aceptar la invitacion

¿Qué es un repositorio público en GitHub?
un repositorio publico en GitHub es un espacio donde puedes almacenar tu codigo, proyectos y archivos que estan disponibles para cualquier persona en el mundo. a diferencia de un repositorio privado, los repositorios publicos son visibles para todos los usuarios de GitHub e incluso para quienes no tienen una cuenta, lo que fomenta la colaboracion abierta y el aprendizaje compartido

¿Cómo crear un repositorio público en GitHub?
iniciamos sesion en gitHub , hacemos click en “New” o “Create repository”en la parte superior derecha de la pagina principal, encontraras un boton verde que te permitira iniciar la creacion de un nuevo repositorio

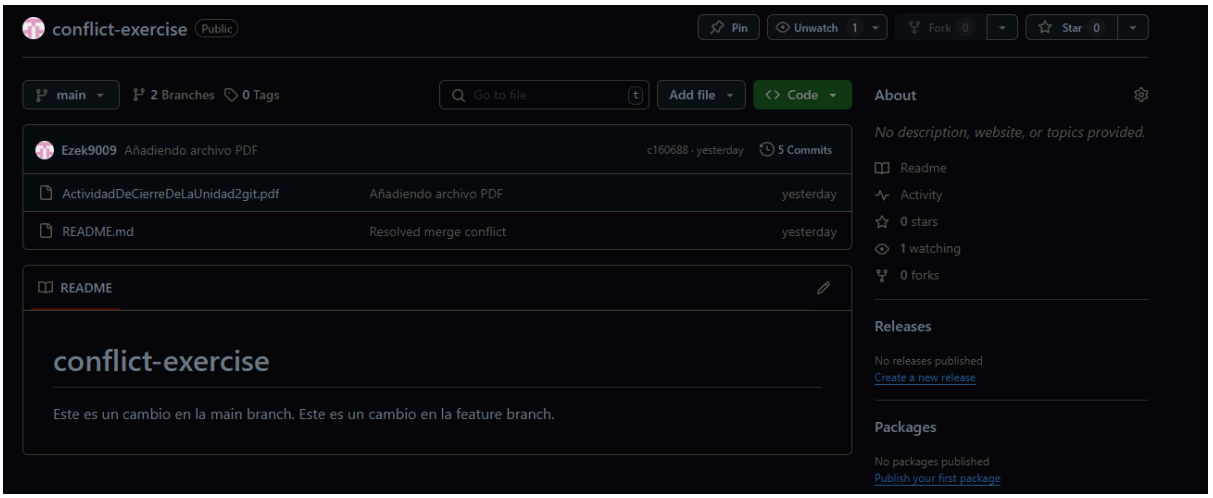
1. configura el repositorio:
   escribe el nombre para tu repositorio en el campo “Repository name”.
2. opcionalmente, agrega una descripcion para explicar de que trata tu proyecto.
3. selecciona la opcion “public” para que el repositorio sea visible para todos
4. inicializa el repositorio:
   marca la casilla de “Add a README file” para añadir un archivo inicial con informacion basica.
   opcionalmente, puedes agregar un archivo .gitignore para ignorar ciertos archivos en el seguimiento de GIt, o elegir una licencia si deseas que otros puedan usar tu proyecto
5. hacer click en “Create repository”:
   esto completara el proceso y tu repositorio publico estara listo para ser utilizado
6. comparte el repositorio(opcional):

copia la URL del repositorio publico para compartirlo con otros. ahora cualquier usuario de GitHub podra verlo, clonarlo o incluso colaborar si se lo permite

¿Cómo compartir un repositorio público en GitHub?
ir al repositorio en GitHub que deseas compartir.
en la pagina principal del repositorio, haz click en el boton verde “Code” ubicado en la parte superior
copia la URL que aparece (puede ser https, ssh o github CLI, dependiendo de tu configuracion) y envia esta URL a las personas con quienes deseas compartir el repositorio

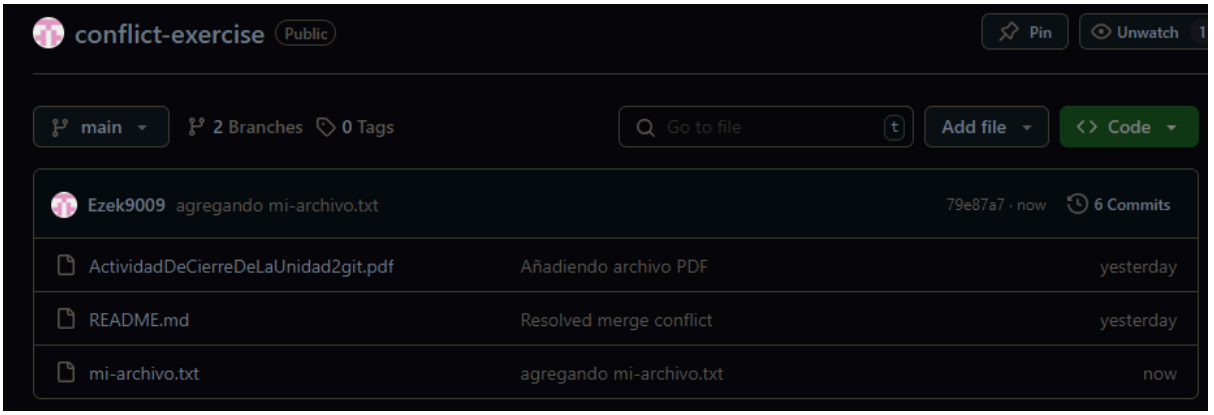
2)Realizar la siguiente actividad:

-Crear un repositorio.
-Dale un nombre al repositorio.
-Elige el repositorio sea público.
-Inicializa el repositorio con un archivo.



Agregando un archivo:

-Crea un archivo simple, por ejemplo, "mi-archivo.txt".
-Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la linea de comandos.
-sube los cambios al repositorio en GitHub con git push origin main(o el nombre de la rama correspondiente)

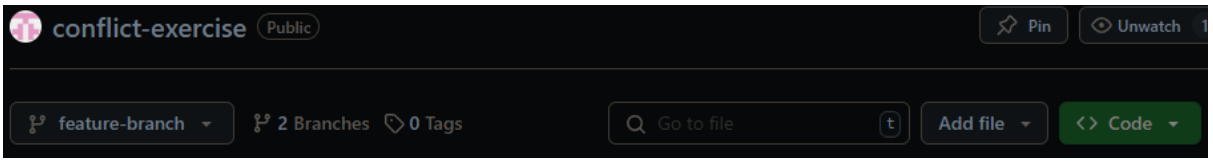


Creando branchs:

-Crear una Branch

-Realizar cambios o agregar un archivo

-Subir la Branch



3) Realizar la siguiente actividad:

Paso 1: Crear un repositorio en GitHub

-Ve a GitHub e inicia sesión en tu cuenta.

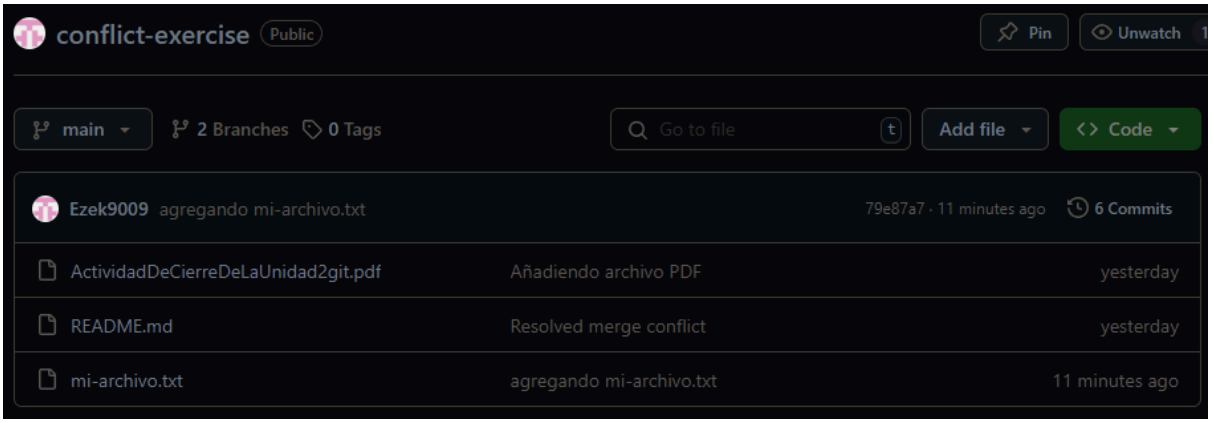
-Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio.

-Asigna un nombre al repositorio, por ejemplo, conflict-exercise.

-Opcionalmente, añade una descripción.

-Marca la opción "Initialize this repository with a README".

-Haz clic en "Create repository".



Paso 2: Clonar el repositorio a tu máquina local

-Copia la URL del repositorio (usualmente algo como https://github.com/Ezek9009/conflict-exercise.git).
-Abre la terminal o línea de comandos en tu máquina.
-Clona el repositorio usando el comando:
-git clone https://github.com/Ezek9009/conflict-exercise.git
-Entra en el directorio del repositorio:
cd conflict-exercise

Paso 3: Crear una nueva rama y editar un archivo

-Crea una nueva rama llamada feature-branch
-git checkout -b feature-branch
-Guarda los cambios y haz un commit
git add README.md
git commit -m "Added a line in feature-branch"

Paso 4: Volver a la rama principal y editar el mismo archivo

-Cambia de vuelta a la rama principal (main):
-git checkout main
-Edita el archivo README.md de nuevo, añadiendo una línea diferente
-Este es un cambio en la main branch.
-Guarda los cambios y haz un commit:
git add README.md
git commit -m "Added a line in main branch"

Paso 5: Hacer un merge y generar un conflicto

-Intenta hacer un merge de la feature-branch en la rama main
-git merge feature-branch
-Se generará un conflicto porque ambos cambios afectan la misma línea del archivo README.md.

Paso 6: Resolver el conflicto

-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
```

-Decide cómo resolver el conflicto. Puedes mantener ambos cambios, elegir uno de ellos, o fusionar los contenidos de alguna manera.

-Edita el archivo para resolver el conflicto y guarda los cambios(Se debe borrar lo marcado en verde en el archivo donde estes solucionando el conflicto. Y se debe borrar la parte del texto que no se quiera dejar).

-Añade el archivo resuelto y completa el merge:
git add README.md

git commit -m "Resolved merge conflict"

Paso 7: Subir los cambios a GitHub

Sube los cambios de la rama main al repositorio remoto en GitHub:
git push origin main

También sube la feature-branch si deseas:

git push origin feature-branch

Paso 8: Verificar en GitHub

Ve a tu repositorio en GitHub y revisa el archivo README.md para confirmar que los cambios se han subido correctamente.

Puedes revisar el historial de commits para ver el conflicto y su resolución.