Tecnicatura Superior en Programación

UTN-FRM

Organización Empresarial I - Git & GitHub

**Git & GitHub**

**1- Contestar las siguientes preguntas utilizando las guías y documentación proporcionada**

**(Desarrollar las respuestas) :**

• ¿Qué es GitHub?

Git Hub es un control de versiones que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas.

• ¿Cuáles son los beneficios de usar GitHub?

La principal diferencia entre Git y cualquier otro VCS (incluyendo Subversion y sus amigos) es la forma en la que manejan sus datos. Git maneja sus datos como un conjunto de copias instantáneas de un sistema de archivos miniatura. Cada vez que confirmas un cambio, o guardas el estado de tu proyecto en Git, él básicamente toma una foto del aspecto de todos tus archivos en ese momento y guarda una referencia a esa copia instantánea.

• ¿Cómo crear un repositorio en GitHub?

Se puede crear un prepositorio de dos maneras:

1. Tomar un proyecto o directorio existente e importarlo en Git.

Para enlazarlo deberás poner el comando “ $ git push -u origin master “.

1. Clonar un repositorio existente en Git desde otro servidor.

• ¿Cuáles son los tipos de ramas en Git?

La rama por defecto de Git es la rama master. A partir de ahí se pueden crear mas ramas las cuales se le puede asignar a cualquiera de ellas como HEAD.

• ¿Cómo crear una rama en Git?

Usando el comando “ $ git branch (nombre de rama) “

• ¿Cómo cambiar a una rama en Git?

Para saltar de una rama a otra, tienes que utilizar el comando “ $ git checkout (nombre de la rama) “

• ¿Cómo fusionar ramas en Git?

Hay que activar el checkout de la rama donde deseas fusionar y lanzar el comando “ $ git merge (rama) ”

• ¿Cómo crear un commit en Git?

Debes hacer commit cuando termines una tarea o feature en tu proyecto. Lo siguiente es confirmarlo usando el comando “ git commit –m “

• ¿Cómo enviar un commit a GitHub?

• ¿Qué es un repositorio remoto?

Los repositorios remotos son versiones de tu proyecto que están hospedadas en Internet o en cualquier otra red.

• ¿Cómo agregar un repositorio remoto a Git?

• ¿Cómo empujar cambios a un repositorio remoto?

• ¿Cómo tirar de cambios de un repositorio remoto?

• ¿Qué es un fork de repositorio?

Un fork, vendría siendo la copia del repositorio de otro usuario en GitHub. Esto nos permite crear un repositorio local con su mismo sistema.

• ¿Cómo crear un fork de un repositorio?

Para ver los remotos que tienes configurados, debes ejecutar el comando git remote. Mostrará los nombres de cada uno de los remotos que tienes especificados.

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

Te deberías dirigir al repositorio del usuario al que le ha sido clonado y hacer click en la opción “new pull request” para que aparezca tu cambio realizado como opción de aprobación.

• ¿Cómo aceptar una solicitud de extracción?

Para aceptar el cambio por request debes recurrir al comando “ merge “ dentro de la plataforma. Entonces se añadira esa “rama” al repositorio principal.

• ¿Qué es un etiqueta en Git?

Una etiqueta vendría siendo el versionado que le puedes aplicar en tus distintas etapas de “commit”. Sirve para marcar los distintos pasos de cambio en el código en modos generales y categorizarlos en escalas.

• ¿Cómo crear una etiqueta en Git?

Para etiquetar un commit utilizaremos el comando “ $ git tag.(nombre de etiqueta) (numero de commit) “

• ¿Cómo enviar una etiqueta a GitHub?

• ¿Qué es un historial de Git?

El historial te permite llevar un control visual y cronológico del proceso en el repositorio. Gracias al historial puedes viajar entre commits, recuperar datos de ellos o resetear lapsos.

• ¿Cómo ver el historial de Git?

• ¿Cómo buscar en el historial de Git?

Tecnicatura Superior en Programación

UTN-FRM

Organización Empresarial I - Git & GitHub

• ¿Cómo borrar el historial de Git?

• ¿Qué es un repositorio privado en GitHub?

• ¿Cómo crear un repositorio privado en GitHub?

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

• ¿Qué es un repositorio público en GitHub?

• ¿Cómo crear un repositorio público en GitHub?

• ¿Cómo compartir un repositorio público en GitHub?

• ¿Qué es un repositorio de código en GitHub?

• ¿Cómo crear un repositorio de código en GitHub?

• ¿Cómo compartir un repositorio de código en GitHub?

• ¿Qué es un repositorio de documentación en GitHub?

• ¿Cómo crear un repositorio de documentación en GitHub?

• ¿Cómo compartir un repositorio de documentación en GitHub?

• ¿Qué es un repositorio de proyecto en GitHub?

• ¿Cómo usar Git para trabajar con otros?

**2- Realizar la siguiente actividad:**

• Crear un repositorio.

o Dale un nombre al repositorio.

o Elije el repositorio sea público.

o Inicializa el repositorio con un archivo.

• Agregando un Archivo

o Crea un archivo simple, por ejemplo, "mi-archivo.txt".

o Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la

línea de comandos.

o Sube los cambios al repositorio en GitHub con git push origin main (o el

nombre de la rama correspondiente).

Tecnicatura Superior en Programación

UTN-FRM

Organización Empresarial I - Git & GitHub

• Creando Branchs

o Crear una Branch

o Realizar cambios o agregar un archivo

o 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/tuusuario/conflict-exercise.git).

• Abre la terminal o línea de comandos en tu máquina.

• Clona el repositorio usando el comando:

git clone https://github.com/tuusuario/conflict-exercise.git

• Entra en el directorio del repositorio:

cd conflict-exercise

Paso 3: Crear una nueva rama y editar un archivo

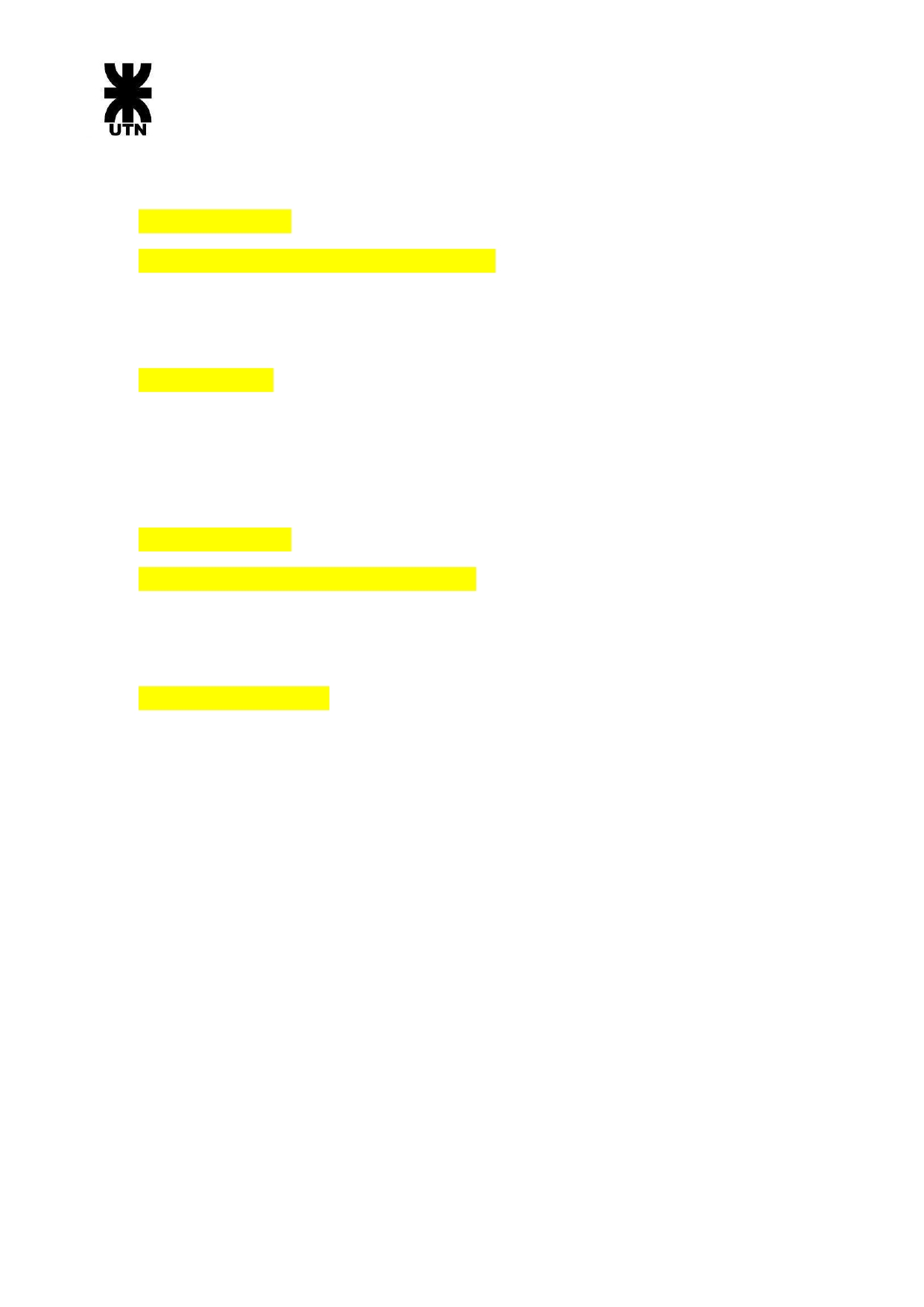
• Crea una nueva rama llamada feature-branch:

git checkout -b feature-branch

• 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.

Tecnicatura Superior en Programación

UTN-FRM

Organización Empresarial I - Git & GitHub

• 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:

Copiar código

<<<<<<< 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.

Tecnicatura Superior en Programación

UTN-FRM

Organización Empresarial I - Git & GitHub

• 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.

**Entrega**:

• Responder las preguntas en el documento de Word.

• Colocar Apellido y Nombre.

• Copiar los enlaces al proyecto de GitHub.

**Importante**: La entrega de este trabajo se realizará de forma individual.