**Trabajo Practico Numero Dos**

**Programacion I**

**Marcos Janchuk**

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada

(Desarrollar las respuestas):

• ¿Qué es GitHub?

Github es una plantaforma donde esta permitido subir repositorios de código, o mismo, formato txt. Es perfecto para trabajar en equipo manteniendo un orden.

• ¿Cómo crear un repositorio en GitHub?

Esto se realiza desde Github con el botón crear repositorio.

• ¿Cómo crear una rama en Git?

git branch nombreDeLaRama

• ¿Cómo cambiar a una rama en Git?

git checkout -b nombreDeLaRama

• ¿Cómo fusionar ramas en Git?

git merge nombreDeLaRama

• ¿Cómo crear un commit en Git?

git commit -m “Nombre o mensaje que queramos escribir”

• ¿Cómo enviar un commit a GitHub?

git push -u origin master

• ¿Qué es un repositorio remoto?

Es básicamente una copia del proyecto que se guarda en GitHub u otra plataforma.

• ¿Cómo agregar un repositorio remoto a Git?

git remote add origin URL\_DEL\_REPO

• ¿Cómo empujar cambios a un repositorio remoto?

git push -u origin nombre

• ¿Cómo tirar de cambios de un repositorio remoto?

git pull

• ¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio de GitHub.

• ¿Cómo crear un fork de un repositorio?

En el repositorio de interés, apretamos el botón fork.

Una vez que se hacen cambios en la rama, en el github del repositorio, creamos un New pull request.

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

En Pull requests, hacemos click en los cambios propuestos y, si estás de acuerdo, hacemos un Merge pull request para integrarlos.

• ¿Cómo aceptar una solicitud de extracción?

En el repositorio original, entrá a la sección de “Pull requests”, revisá los cambios propuestos y, si estás de acuerdo, hacé clic en “Merge pull request” para integrarlos.

• ¿Qué es una etiqueta en Git?

Una etiqueta es una marca que se pone en un punto específico del historial del proyecto.

• ¿Cómo crear una etiqueta en Git?

git tag nombre-etiqueta

• ¿Cómo enviar una etiqueta a GitHub?

git push origin nombre-etiqueta

• ¿Qué es un historial de Git?

El historial es el registro completo de todos los commits realizados en el proyecto. Muestra quién hizo qué cambios y cuándo.

• ¿Cómo ver el historial de Git?

git log

• ¿Cómo buscar en el historial de Git?

git log -S "texto"

• ¿Cómo borrar el historial de Git?

git rebase -i o git filter-branch

• ¿Qué es un repositorio privado en GitHub?

Un repositorio privado es aquel al que solo pueden acceder las personas que vos autorices.

• ¿Cómo crear un repositorio privado en GitHub?

Se crea seleccionando la opción private al momento de crear el mismo.

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

En la opción “Manage access” o “Colaboradores” para invitar.

• ¿Qué es un repositorio público en GitHub?

Es un repositorio al que cualquier persona puede acceder y ver el código.

• ¿Cómo crear un repositorio público en GitHub?

Se crea seleccionando la opción public al momento de crear el mismo.

• ¿Cómo compartir un repositorio público en GitHub?

Para compartirlo se necesita copiar el link del mismo y compartirlo.

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).

2

• 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.

• 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

3

• 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.