



Git & GitHub

Alumno y curso: Esteban Alamino 1Prog2 turno tarde

1- Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

- ¿Qué es GitHub?

Github es un sistema de almacenamiento de información en forma de repositorios que es especialmente útil para desarrolladores para así poder gestionar versiones y controlar y compartir proyectos de software

- ¿Cuáles son los beneficios de usar GitHub?

Es relativamente fácil de usar, se considera una plataforma segura, permite comunicación de manera rápida y efectiva, tiene una gran comunidad por lo que es fácil encontrar instructivos para usarlo y todo tipo de software y a su vez se puede adaptar a casi cualquier escala de cualquier proyecto.

- ¿Cómo crear un repositorio en GitHub?

Se debe tener a disposición una cuenta de github y configurarla, una vez creada dicha cuenta se pueden empezar a crear repositorios desde el perfil.

- ¿Cuáles son los tipos de ramas en Git?

Main: Es la rama principal y por defecto de la mayoría de los repositorios

Desarrollo(Develop): es una rama que esta en constante cambio, a esta rama se le integran los cambios para probarlos antes de enviarlos a la rama principal

Feature: Son ramas que se crean a partir de las ramas de desarrollo para probar funciones completamente nuevas

Bugfix: Son ramas que se crean para solucionar errores en especifico que despues se envia a la rama principal

Hotfix: Es como la rama de bugfix pero es para solucionar errores criticos de alta prioridad

- ¿Cómo crear una rama en Git?



Se crean nuevas ramas desde la consola de comando con el comando "git branch" y se le añade un nombre por lo que el comando quedaria "git branch -Nombre de la rama-"

- ¿Cómo cambiar a una rama en Git?

Usando el comando "git checkout" y el nombre de la rama a la que se desea acceder

- ¿Cómo fusionar ramas en Git?

Primero hay que ir a la rama a la que se desea añadir la rama secundaria usando el comando "git checkout" una vez en dicha rama se usar el comando "Git merge" + el nombre de la rama a fusionar

- ¿Cómo crear un commit en Git?

Primero hay que añadir un archivo nuevo al stagin area con el comando "git add" luego podemos comprobar el estado del archivo con "git status" y par crear el commit se usa "git commit -m" (-m es para añadir el mensaje acerca de los archivos que fueron añadidos)

- ¿Cómo enviar un commit a GitHub?

Lo primero es agregar un repostorio remoto a git con el comando "git remote add" seguido del nombre del repositorio remoto (En este caso vamos a usar origin) seguido de la url del repositorio de github, después usamos el comando "git push" seguido del repositorio origin y la rama cuyos cambios queremos enviar a Github por lo que el comando quedaría como "git push origin main" para la rama principal

- ¿Qué es un repositorio remoto?

Un repositorio remoto es una copia del repositorio local que se usa para almacenar información en un servidor externo donde otras personas pueden acceder a dicha información.

- ¿Cómo agregar un repositorio remoto a Git?

Se usa el comando "Git remote add" seguido del nombre del repositorio remoto y seguido de la url de dicho repositorio

- ¿Cómo empujar cambios a un repositorio remoto?

Luego de hacer un commit en git usamos el comando "git push" seguido del nombre del repositorio y de la rama en la que se hicieron los cambios que se quieren empujar (Ej: git push origin main)

- ¿Cómo tirar de cambios de un repositorio remoto?



Se usa el comando “git pull” seguido del nombre del repositorio y el nombre de la rama a la que se quieren traer los cambios (Ej: git pull origin main)

- ¿Qué es un fork de repositorio?

Un fork es una copia personal del repositorio original que se almacena en la cuenta del usuario

- ¿Cómo crear un fork de un repositorio?

Una vez que encuentres el repositorio al que le quieres hacer un fork simplemente dirígete a la esquina superior derecha de este repositorio y presiona la tecla de Fork

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

Primero se tiene que crear un fork de un repositorio, después clonamos este fork localmente con el comando “git clone” seguido de la url del fork, nos movemos al fork con `cd -nombre del fork-` y creamos una rama para los cambios, una vez hechos los cambios hacemos el commit del archivo y usamos “git push” para enviar los cambios, ahora que volvamos a la pagina del fork Github te va a informar que hay una solicitud de extracción del fork, escribimos un titulo y una descripción de los cambios y usamos el comando “create pull request” para enviar la solicitud.

- ¿Cómo aceptar una solicitud de extracción?

En la pagina de Github vamos a revisar la solicitud de extracción que queramos aceptar en la sección que dice “Pull requests” una vez verificados los cambios vamos seleccionar la opción “Approve” en la revisión para confirmar los cambios

- ¿Qué es un etiqueta en Git?

Una etiqueta en git es una referencia a un cambio específico en el historial de commits de un repositorio

- ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta se usa el comando “git tag” seguido del nombre de la etiqueta, también se puede crear una etiqueta anotada con el comando “git tag” seguido de “-a” y “-m”

- ¿Cómo enviar una etiqueta a GitHub?

Para empujar la etiqueta a Github se usa el comando “git push” seguido del nombre del repositorio y el nombre de la etiqueta.

- ¿Qué es un historial de Git?



El historial es el registro de commits que registra todos los cambios que se la hicieron a un repositorio

- ¿Cómo ver el historial de Git?

Para ver el historial de git se usa el comando “git log”

- ¿Cómo buscar en el historial de Git?

Se puede buscar con palabras específicas, con fechas o por autores

`git log --grep="palabra_clave"` te mostrara el historial de git que contenga las palabras clave solicitadas

`git log --since="fecha_inicial" --until="fecha_final"` te muestra los cambios que se hicieron en un lapso de tiempo definido

`git log --author="nombre_del_autor"` te muestra los cambios hechos por un autor en específico

Hay otra forma de buscar cambios, pero son demasiado específicos para nombrarlos todos

- ¿Cómo borrar el historial de Git?

Para borrar el historial de cambios se debería borrar el directorio `.git` del repositorio local, lo que elimina también el acceso a github. Alternativamente se puede crear un repositorio nuevo que no tenga historial de cambios

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio que solo se puede acceder por personas que tienen permiso del administrador

- ¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado hay que crear un repositorio normal pero marcar la opción de “Private” en la sección de visibilidad

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

En la sección de collaborators se puede añadir a personas de manera manual para otorgar acceso a un repositorio privado

- ¿Qué es un repositorio público en GitHub?



Un repositorio publico es un repositorio que puede ser accedido por cualquier persona que encuentre la dirección de dicho repositorio

- ¿Cómo crear un repositorio público en GitHub?

Se crea un repositorio asegurándose de marcar publico en la sección de visibilidad

- ¿Cómo compartir un repositorio público en GitHub?

Ya que un repositorio publico no requiere accesos especiales, solo se tiene que copiar la url de este repositorio y enviársela a la persona que se desea compartir.

- ¿Qué es un repositorio de código en GitHub?

Un repositorio de código es un conjunto de archivos y directorios que contienen el cosigo fuente de un proyecto, a su vez también contiene si historial de versiones, etiquetas y otros datos relacionados al control de versiones.

- ¿Cómo crear un repositorio de código en GitHub?

Se crea como cualquier otro repositorio, solo cambia el fin del mismo

- ¿Cómo compartir un repositorio de código en GitHub?

Se comparte la url si es publico o se añade un colaborador si es privado

- ¿Qué es un repositorio de documentación en GitHub?

Es un repositorio destinado a almacenar documentos, guias y manuales de proyectos

- ¿Cómo crear un repositorio de documentación en GitHub?

Se crea como cualquier otro repositorio, en la cuenta de github seleccionas la opción “New repository”

- ¿Cómo compartir un repositorio de documentación en GitHub?

Se comparte la url si es publico o se añade un colaborador si es privado

- ¿Qué es un repositorio de proyecto en GitHub?

Es un repositorio que contiene toda la información, códigos, documentación y recursos de un proyecto especifico.

- ¿Cómo usar Git para trabajar con otros?

Una vez que todos los integrantes tienen acceso a un repositorio todos pueden actualizar la información del mismo



2- Realizar la siguiente actividad:

- Crear un repositorio.
 - Dale un nombre al repositorio.
 - Elije 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 línea 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/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

- 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.
- 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.



Tecnicatura Superior en Programación
UTN-FRM
Organización Empresarial I – Git & GitHub

- Copiar los enlaces al proyecto de GitHub.

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