

CONTROL DE VERSIONES CON GITFLOW

Resumen: La actividad consistió en implementar un flujo de trabajo con Git y GitFlow para gestionar el control de versiones de un proyecto, configurando un repositorio local enlazado a dos remotos en GitHub y GitLab, donde se realizaron los pasos de inicialización, creación del primer commit, definición de las ramas principales (Main y Development), desarrollo de una funcionalidad en una rama llamada Feature, integración de los cambios, publicación de todas las ramas y etiquetas en ambos repositorios, y finalmente la preparación de un Release (versión 1.0.0= siguiendo buenas prácticas de versionado y colaboración.

Introducción: La presente actividad tiene como propósito aplicar el control de versiones utilizando Git bajo la metodología GitFlow, con el fin de organizar y gestionar de manera adecuada el ciclo de vida de un proyecto de software. A través de este flujo de trabajo se busca mantener un orden en las ramas principales y secundarias, facilitar la integración de nuevas funcionalidades, preparar versiones estables para producción y garantizar la trazabilidad de los cambios. Asimismo, se configuró la sincronización del repositorio local con plataformas remotas como GitHub y GitLab, permitiendo mantener copias actualizadas y disponibles en entornos colaborativos.

Metodología: Paso a paso de lo realizado

1. Clonación de los repositorios remotos:

```
juans@PinzitasPC MINGW64 ~  
$ git clone https://github.com/Juansps1612/Control-de-versiones.git
```

```
juans@PinzitasPC MINGW64 ~  
$ git clone git@gitlab.com:ingenieriaweb2/control-de-versiones-lab.git
```

2. Inicialización de GitFlow:

```
juans@PinzitasPC MINGW64 ~  
$ git flow init
```

2.1 Se dio nombre a la rama de producción como Main

2.2 Se dio nombre a la rama de desarrollo como Development

3. Se crearon todas las funcionalidades (Feature):

```
juans@PinzitasPC MINGW64 ~  
$ git flow feature start sumar
```

Nota: Se usó el mismo comando para las demás funciones (Restar, Multiplicar, Dividir y Menú interactivo)

3.1 Una vez se acaba de desarrollar la función se hacía el respectivo commit especificando que hacía esa función y se fusionaba (o se cerraba) la rama relacionada a esa función:

```
juans@PinzitasPC MINGW64 ~  
$ git add calculadora.py
```

```
juans@PinzitasPC MINGW64 ~  
$ git commit -m "Añadir función sumar"
```

```
juans@PinzitasPC MINGW64 ~  
$ git flow feature finish sumar
```

Nota: Este proceso fue igual en todas las funciones.

4. Regresamos a la rama Main y procedemos a fusionar la rama Development para poder crear nuestro primer Release:

```
juans@PinzitasPC MINGW64 ~  
$ git checkout main
```

```
juans@PinzitasPC MINGW64 ~  
$ git merge development
```

5. Una vez hemos fusionado la rama Development con la rama Main, procedemos a realizar nuestro primer Release con la etiqueta v1.0.0:

```
juans@PinzitasPC MINGW64 ~  
$ git push origin main
```

```
juans@PinzitasPC MINGW64 ~  
$ git push origin development
```

```
juans@PinzitasPC MINGW64 ~  
$ git tag -a v1.0.0 -m "Versión 1.0.0 - Calculadora con menú"
```

```
juans@PinzitasPC MINGW64 ~  
$ git push origin v1.0.0
```

6. Una vez hemos realizado todos esos pasos en GitHub, procedemos a traer todos esos documentos a GitLab:

```
juans@PinzitasPC MINGW64 ~  
$ cd control-de-versiones
```

```
juans@PinzitasPC MINGW64 ~  
$ git remote add gitlab git@gitlab.com:ingenieriaweb2/control-de-versiones-lab.git
```

```
juans@PinzitasPC MINGW64 ~  
$ git push gitlab --all
```

```
juans@PinzitasPC MINGW64 ~  
$ git push gitlab --tags
```

```
juans@PinzitasPC MINGW64 ~  
$ git pull gitlab main --rebase
```

```
juans@PinzitasPC MINGW64 ~  
$ git push gitlab main
```

Resultados: Los resultados obtenidos evidencian la correcta implementación de GitFlow como estrategia de control de versiones, logrando una estructura organizada de ramas para el desarrollo, integración y liberación de versiones. Se configuró exitosamente la conexión del repositorio local con GitHub y GitLab, asegurando la publicación sincronizada de ramas y etiquetas en ambos entornos. Además, se generó un reléase estable (v1.0.0) que consolidó los cambios desarrollados en las ramas Feature, permitiendo contar con un punto de referencia confiable para futuras iteraciones. En general, la actividad permitió establecer un flujo de trabajo claro, seguro y escalable para el proyecto.

Conclusión: La actividad permitió comprender y aplicar el uso de GitFlow como una metodología eficaz para gestionar el control de versiones, logrando una mejor organización del proyecto y garantizando un flujo de trabajo ordenado al sincronizar los cambios en GitHub y GitLab.