Claudio A. Ferreira

2010-0316

Prof: Kelyn Tejada Belliard

Programación III

Tarea 3 – Herramientas de administración de fuentes

GIT FLOW

URL proyecto:

https://github.com/claudioaferreira/test-git-sabado-03

Desarrolla el siguiente Cuestionario

¿Qué es Git?

Es un software de control de versiones que realiza un seguimiento de los cambios en los archivos. Es especialmente útil cuando un grupo de personas están hacieno cambios en los mismos al mismo tiempo.

¿Para que funciona el comando Git init?

El comando git init se utiliza para crear un repositorio de git en un director vacío o en uno que ya existe, pero no este siendo rastreado por git.

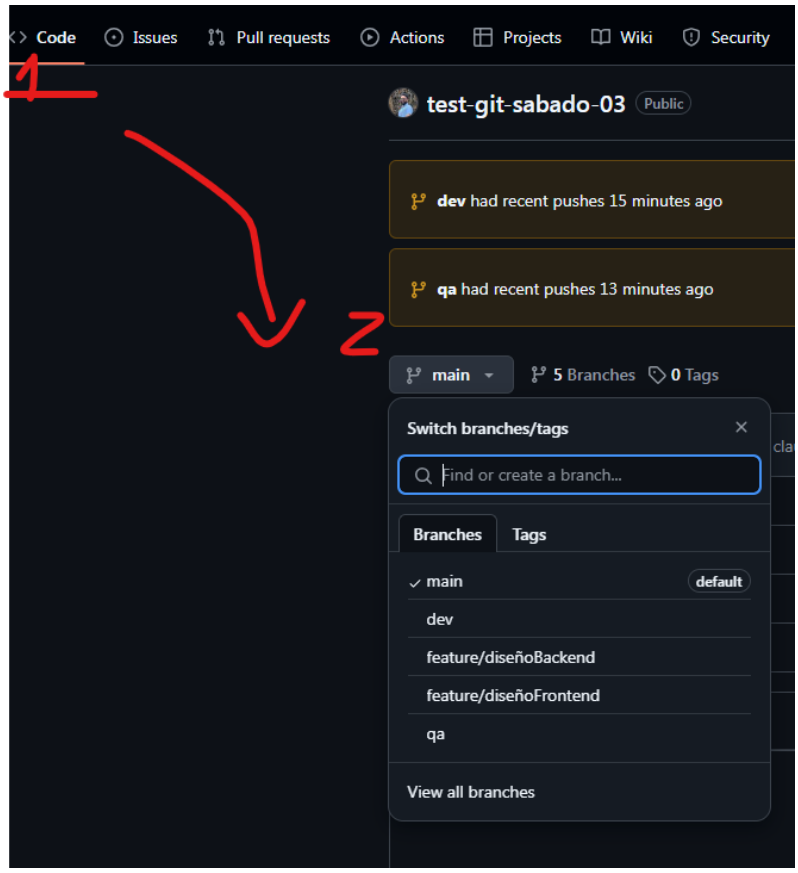
Al ejecutar el comando git init, se crea un subdirectorio (.git) en el directorio del trabajo actual y se crea una nueva rama principal. Dentro de este subdirectorios se crean los metadatos que se necesita git, como la rama default, objetos, referencias y archivos de template.

¿Qué es una rama?

Una rama o Branch es una versión del código del proyecto sobre el que se esta trabajando. Estas ramas ayudan a mantener el orden en el control de versiones y manipular el código de forma segura.

En otras palabras, una rama proviene de otra. Imaginemos un árbol, que tiene una rama gruesa, y otra mas fina, en la rama mas gruesa tenemos los commits principales y en la rama fina tenemos otros commits que pueden ser de dev entre otros.

¿Cómo saber en cual rama estoy?



También se puede utilizar el comando git Branch -a

¿Quién creo git?

Fue diseñado por Linus Torvalds quien fue el creador del kernel Linux. Git es un software de código abierto y software libre. Su desarrollo empezó en abril de 2005, cuando BitKeeper, el sistema de control de versiones propietario que se usaba para el desarrollo de Linux, revoco la licencia gratuita.

¿Cuáles son los comandos más esenciales de Git?

Los comandos más esenciales son:

- **git add**: Agrega archivos al área de preparación.
- **git commit**: Guarda una instantánea de los cambios en el directorio Git.
- **git clone**: Copia un repositorio de Git.
- **git diff**: Muestra las diferencias entre confirmaciones.

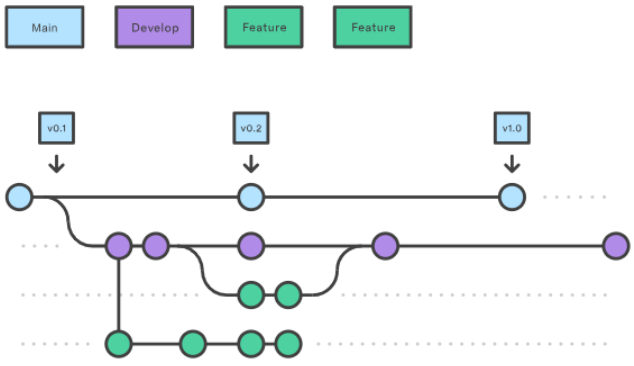
- **git fetch**: Descarga una rama de otro repositorio.
- **git init**: Crea un nuevo repositorio de Git.
- **git log**: Muestra el historial del proyecto.
- **git pull**: Fusiona los cambios del repositorio remoto con el directorio de trabajo local.
- **git push**: Envía confirmaciones locales al repositorio remoto.
- **git status**: Muestra los archivos que se han cambiado, los que están por ser preparados o confirmados.
- **git checkout**: Cambia de una rama a otra.
- **git branch**: Crea, lista o borra ramas.

¿Qué es git Flow?

El Git Flow es un flujo de trabajo basado en Git que brinda un mayor control y organización en el proceso de integración continua.

En esta se utilizan ramas de función y varias ramas principales. GitFlow se puede utilizar en proyectos que tienen un ciclo de publicación programada, así como para la practica recomendada de DevOps de entrega continua. Este flujo de trabajo no añade ningún concepto o comando nuevo, aparte de los que se necesitan para el flujo de trabajo de ramas de función.

Este se debe implementar porque aumenta la velocidad de entrega de código terminado al equipo de pruebas, disminuyen los errores humanos en la mezcla de las ramas, elimina la dependencia de funcionalidades al momento de entregar código.



¿Que es trunk based development?

Es una estrategia de git donde existe un trunk(un Branch principal, usualmente llamado master/main) en el cual todo el equipo colabora he integra directamente (hace push).

Esto nos da beneficios como:

- Nos evitamos esos grandes conflictos y esos merges extra.
- Los desarrolladores siempre van a poder trabajar con el código más reciente.
- El equipo se vuelve mas eficiente y más ágil para entregar el código.