GA7-220501096-AA1-EV01

Fran Emilio Peña Pedroza
Juan Camilo Londoño Londoño
Juan David Rojas Alzate
Wilmer Ferney Estevez Piratoa

SENA
Centro Minero
Análisis y desarrollo de software
Marzo 2025

Contenido

Introducción

El control de versiones es fundamental para garantizar una gestión eficiente del código fuente y facilitar el trabajo colaborativo. Este informe presenta un análisis de las herramientas y tecnologías de versionamiento a utilizar en el desarrollo del software, enmarcadas dentro de las prácticas de integración continua. Se describirán las herramientas seleccionadas y su importancia en la optimización del proceso de construcción del software

Objetivo

El objetivo de este informe es definir las herramientas de control de versiones a emplear en el desarrollo del software, garantizando una administración eficiente del código y una mejor colaboración entre los desarrolladores.

Selección de herramientas de versionamiento

Git

Git es un sistema de control de versiones distribuido ampliamente utilizado en el desarrollo de software. Sus principales características incluyen:

- Permite el trabajo colaborativo sin conexión.
- Facilita el seguimiento de cambios en el código fuente.
- Ofrece ramas (branches) para gestionar múltiples versiones del software.

GitHub

GitHub es una plataforma basada en Git que permite almacenar repositorios de código en la nube. Características clave:

- Permite la colaboración entre equipos a través de repositorios remotos.
- Soporta pull requests y revisiones de código.
- Facilita la gestión de permisos y revisiones antes de fusionar cambios.

Proceso propuesto para el manejo de repositorios

El equipo está conformado por 4 integrantes, cada uno creara su propio repositorio local. Se contará con un **repositorio remoto en GitHub** y un **líder del equipo** que será el encargado de aprobar los pull requests y hacer el merge en la rama principal (main).

A continuación, se describe el flujo de trabajo y los comandos clave según cada rol:

1. ***Configuración inicial (Todos los integrantes)***

Cada integrante debe clonar el repositorio remoto para trabajar localmente:

git clone https://github.com/juancho-sp/GastroGo.git

Para moverse al repositorio ingresa el comando

cd GastroGo

Configurar la identidad en Git:

git config --global user.name "nombre-integrante"

git config --global user.email "correointegrante@email.com"

2. ***Creación de una nueva funcionalidad (Desarrolladores)***

Cada integrante trabajará en una nueva rama para su funcionalidad:

Crear y cambiar a una nueva rama

git checkout -b nombre-nueva-funcionalidad

Después de hacer cambios en los archivos, se guardan y se suben al repositorio:

Agregar archivos modificados al área de preparación

git add .

Confirmar los cambios

git commit -m "Descripción de los cambios realizados"

Subir los cambios al repositorio remoto en una nueva rama:

git push origin nombre-nueva-funcionalidad

3. Creación de Pull Request (Desarrolladores)

Desde GitHub, cada desarrollador debe crear un Pull Request (PR) para que el líder del equipo revise los cambios antes de fusionarlos en main.

4. Revisión y Aprobación (Líder del equipo)

El líder del equipo revisará los PRs y, si todo está correcto, procederá a fusionarlos en la rama principal:

Cambiar a la rama principal

git checkout main

Traer los últimos cambios del remoto

git pull origin main

Fusionar la rama con los cambios aprobados

git merge nombre-nueva-funcionalidad

Subir los cambios al remoto

git push origin main

Después de fusionar, se recomienda eliminar la rama para mantener ordenado el repositorio:

git branch -d feature-nueva-funcionalidad

git push origin --delete feature-nueva-funcionalidad

5. Actualización del código local (Todos los integrantes)

Para asegurarse de tener siempre la última versión de main, cada integrante debe actualizar su repositorio local regularmente:

Cambiar a la rama principal

git checkout main

Descargar los últimos cambios

git pull origin main

Bibliografía

Sena. (s.f.). integración continua. Recurso Educativo Sena

GitHub. (s.f.). *Documentación oficial de GitHub*. Disponible en: https://docs.github.com