

1. Introducción

Este informe describe el flujo de trabajo del proyecto y la gestión de las ramas principales (main y develop) para garantizar un desarrollo eficiente y controlado del código fuente.

2. Flujo de Trabajo del Proyecto

El flujo de trabajo adoptado en este proyecto sigue un modelo basado en Git Flow, que permite una gestión estructurada del código. Se utilizan las siguientes ramas principales:

- **main:** Contiene la versión estable y lista para producción del código.
- **develop:** Ramas en las que se desarrolla y prueba el código antes de fusionarse en main.

Etapas del Flujo de Trabajo:

1. Planificación y Creación de Tareas

- Se definen los requisitos y se crean las tareas en un sistema de gestión (Jira, Trello, GitHub Projects).

2. Creación de una Nueva Rama

- Se crea una rama a partir de main llamada develop donde se realizarán los nuevos cambios del proyecto.

3. Desarrollo y Pruebas Locales

- Se desarrollan las nuevas características o correcciones.
- Se realizan pruebas unitarias y de integración en el entorno local.

4. Commit y Push de Cambios

- Se realizan commits con mensajes claros y descriptivos.
- Se suben los cambios al repositorio remoto (git push origin develop).

5. Creación de un Pull Request (PR)

- Se crea un PR para fusionar la rama en develop.
- Se asignan revisores para validar el código.

6. Revisión y Aprobación del Código

- Se revisa el código, se solicitan cambios si es necesario y se aprueba el PR.

7. Pruebas en Entorno de Pruebas (Staging)

- Se implementa la versión de develop en un entorno de pruebas.
- Se validan todas las funcionalidades antes del lanzamiento.

8. Fusión en main y Despliegue a Producción

- Una vez validado, develop se fusiona en main.
- Se despliega la nueva versión a producción.

3. Gestión de Ramas main y develop

Ramas Principales:

- **main:** Solo contiene versiones estables. No se desarrollan cambios directamente en esta rama.
- **develop:** Es la rama donde se integran las funcionalidades antes de ser lanzadas a producción.

4. Herramientas Utilizadas

- **Git y GitHub:** Para la gestión de versiones.
- **CI/CD:** Para automatizar pruebas y despliegues.
- **Sistemas de Gestión de Tareas:** Para planificar y hacer seguimiento del desarrollo.

5. Conclusión

Este flujo de trabajo estructurado permite mantener un código limpio y controlado, minimizando errores en producción y asegurando una integración eficiente de nuevas funcionalidades.