

Flujo Detallado

1. Crear la Branch de Proyecto Urgente (Hotfix):

Cuando se necesita desarrollar un proyecto, se crea una branch de hotfix desde master.

- git checkout master
- git pull origin master
- git checkout -b hotfix/nombre-del-proyecto
- git add .
- git commit -m 'comentario con estructura'
- git push

2. Desarrollar y Probar en Branch de Desarrollo (Develop):

Realiza los cambios necesarios en la branch de hotfix y realiza commits. Fusiona la branch de hotfix con develop para pruebas preliminares.

- git checkout develop
- git pull origin develop
- git merge hotfix/nombre-del-proyecto
- git push origin develop

3. Probar en Branch de QA:

Después de las pruebas preliminares, fusiona la branch de hotfix con qa para pruebas de calidad.

- git checkout qa
- git pull origin qa
- git merge hotfix/nombre-del-proyecto
- git push origin qa

4. Validar en Branch de Pre-Producción:

Una vez que las pruebas de calidad son satisfactorias, fusiona la branch de hotfix con pre-production para pruebas finales.

- git checkout pre-production
- git pull origin pre-production
- git merge hotfix/nombre-del-proyecto
- git push origin pre-production

5. Fusionar con Master:

Una vez que el proyecto ha sido validado en pre, se fusiona con master.

- git checkout master
- git pull origin master
- git merge hotfix/nombre-del-proyecto
- git push origin master

6. Sincronizar Pre, QA y Develop con Master:

- git checkout pre-production

- git pull origin pre-production
- git merge master
- git push origin pre-production

- git checkout qa
- git pull origin qa
- git merge master
- git push origin qa

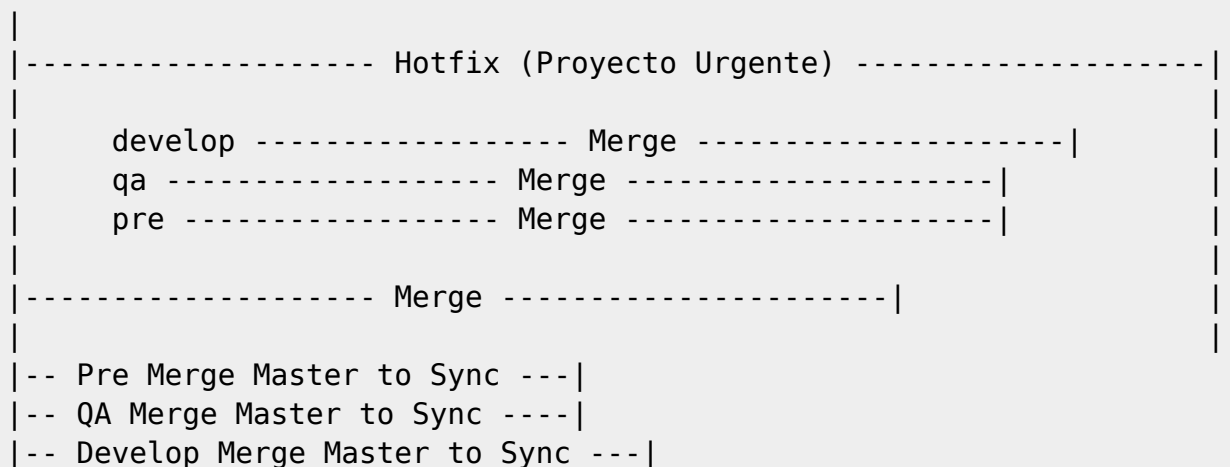
- git checkout develop
- git pull origin develop
- git merge master
- git push origin develop

7. Eliminar la Branch de Proyecto Urgente:

Después de que el proyecto ha sido desplegado a producción y validado, puedes eliminar la branch de hotfix.

- git branch -d hotfix/nombre-del-proyecto
- git push origin --delete hotfix/nombre-del-proyecto

master



Beneficios de Este Enfoque

Multinivel de Pruebas: Permite realizar pruebas en diferentes niveles (develop, qa, pre) antes de desplegar a producción.

Consistencia: Asegura que master, pre, qa y develop estén siempre sincronizadas. **Flexibilidad y Control:** Proporciona un control granular sobre el flujo de trabajo y las pruebas en diferentes etapas.

Este flujo de trabajo proporciona un enfoque estructurado y robusto para gestionar proyectos urgentes, garantizando que los cambios sean probados exhaustivamente en diferentes entornos antes de ser desplegados a producción.

Este enfoque está basado en Git Flow y GitLab Flow, también se analizó GitHub Flow. Se puede

consultar más información para comparar las tres en el siguiente artículo: Choosing the Right Git Branching Strategy: A Comparative Analysis.

From:
<http://wiki.adacsc.co/> - Wiki

Permanent link:
<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:front:flujogit&rev=1718979447>

Last update: **2024/06/21 14:17**

