POLÍTICA DE LA GESTIÓN DE LAS RAMAS

Fecha de Emisión: 17/10/2024
Descripción: Normativa para la creación, uso y gestión de ramas en el repositorio de GitHub para garantizar la colaboración estructurada y controlada en el proyecto.

Introducción y Propósito

Esta política tiene como objetivo proporcionar una estructura clara para la creación y gestión de ramas en Git y GitHub, asegurando una colaboración efectiva y organizada entre los integrantes del equipo. Seguiremos las convenciones estándar de flujo de trabajo, como **Git Flow** y **GitHub Flow**, para facilitar la integración continua y mantener el código en un estado estable y funcional.

Tipos de Ramas y su Propósito

Se definirán ramas específicas para cada propósito dentro del flujo de trabajo, con el fin de mantener una estructura organizada:

1. Rama Principal (`main` o `master`)

- **Descripción**: Es la rama principal del proyecto. Debe reflejar siempre el código **estable y listo para producción**.
- **Reglas**:
    - Todo **merge** debe ser aprobado mediante un **pull request** revisado y aprobado por al menos tres compañeros de equipo.
    - La rama **nunca debe contener código que rompa la compilación o funcionalidades incompletas**.

2. Rama de Desarrollo (`develop`)

- **Descripción**: Esta rama se utiliza como punto de integración para todas las funcionalidades completadas. Representa el código que está **preparado para ser testeado** y, eventualmente, lanzado a producción.
- **Reglas**:
    - Los **features**, **fixes** y otros trabajos deben ser fusionados aquí tras su revisión.
    - Todo código fusionado en esta rama debe pasar pruebas automáticas (si las hay).

3. Ramas de Funcionalidades (`feature/<nombre-descriptivo>`)

- **Descripción**: Cada nueva funcionalidad o tarea debe desarrollarse en una rama separada creada a partir de `develop`.

- **Reglas**:
  - El nombre de la rama debe seguir el patrón: `feature/<nombre-descriptivo>`. Ejemplo: `feature/autenticación-usuario`.
  - Las ramas de características **se crean desde** `develop` y **se fusionan de vuelta a** `develop`.
  - Cada rama debe estar relacionada con una tarea específica del proyecto.
  - Se debe hacer un **pull request** para fusionar la rama, con revisión obligatoria por parte de dos miembros del equipo.
  - Una vez fusionada, la rama debe ser **eliminada** del repositorio.

4. Ramas de Corrección de Errores (`fix/<nombre-descriptivo>` o `bugfix/<nombre-descriptivo>`)

- **Descripción**: Utilizadas para corregir errores específicos.
- **Reglas**:
  - El nombre de la rama debe seguir el patrón: `fix/<nombre-descriptivo>` o `bugfix/<nombre-descriptivo>`. Ejemplo: `fix/error-login`.
  - Se crean desde `develop` si es una corrección en desarrollo, o desde `main` si es una corrección crítica de producción.
  - Las correcciones de errores menores se fusionarán en `develop`, mientras que las correcciones críticas en producción deben fusionarse en `main` mediante un **hotfix** (ver más abajo).

5. Ramas de Lanzamientos (`release/<número-versión>`)

- **Descripción**: Se usan para preparar una nueva versión de producción. Permiten a los desarrolladores realizar correcciones menores y pulir detalles antes de lanzar la versión final.
- **Reglas**:
  - El nombre de la rama debe seguir el patrón: `release/<número-versión>`. Ejemplo: `release/1.0.0`.
  - Se crean desde `develop` cuando se considera que una versión está lista para su lanzamiento.
  - Se pueden aplicar cambios menores, como ajustes de versión, pero no deben incluirse nuevas funcionalidades.
  - Una vez que los cambios son aprobados, la rama `release` debe fusionarse tanto en `main` (para lanzar la versión) como en `develop` (para que el código de la versión esté disponible en futuras iteraciones).
  - La rama se elimina después de su fusión.

6. Ramas de Hotfix (`hotfix/<nombre-descriptivo>`)

- **Descripción**: Para correcciones urgentes en el código de producción.
- **Reglas**:

    - El nombre de la rama debe seguir el patrón: `hotfix/<nombre-descriptivo>`. Ejemplo: `hotfix/solucion-error-produccion`.
    - Se crean desde `main` y, una vez corregido el error, se fusionan de vuelta en `main` y en `develop` (para que la corrección se propague a las siguientes versiones).
    - Se debe realizar un **pull request** para su fusión, con revisión rápida por parte del equipo.
    - Estas ramas se eliminan tras su fusión.

Flujo de Trabajo General

El equipo seguirá un flujo de trabajo basado en **Git Flow** y **GitHub Flow**, adaptado a las necesidades del proyecto. Este es un esquema básico del flujo:

1. **Nueva funcionalidad**: Crear una rama `feature/` desde `develop`, desarrollar y realizar un **pull request**.
2. **Corrección de errores**: Crear una rama `fix/` o `hotfix/`, dependiendo de si el error es en desarrollo o en producción.
3. **Preparación para lanzamiento**: Crear una rama `release/` desde `develop`.
4. **Lanzamiento**: Fusionar la rama `release/` en `main` para generar una nueva versión.
5. **Hotfix**: Para errores críticos en producción, crear una rama `hotfix/` desde `main` y fusionar tanto en `main` como en `develop`.

Reglas de Control para Ramas y Pull Requests

1. **Creación de ramas**: Los desarrolladores deben asegurarse de que cada rama creada siga el esquema de nombres especificado y esté relacionada con una tarea o *issue* asignada.
2. **Revisiones de Pull Request**:
    - Todo **pull request** debe ser revisado por al menos otro miembro del equipo antes de ser fusionado.
    - No se permite el auto-merge sin aprobación.
3. **Protección de ramas**:
    - La rama `main` debe tener **protección** activada, lo que incluye:
        - Requerir aprobación de al menos un revisor.
        - Pasar las pruebas automáticas antes de permitir la fusión.
    - Se recomienda aplicar también protección a la rama `develop` bajo las mismas condiciones.
4. **Integración continua**: Toda rama que se fusione en `develop` o `main` debe pasar por el pipeline de integración continua (CI) para asegurar que no rompe la compilación o las pruebas.

Firma de los integrantes:

- Espinosa Naranjo, Pablo:

- Garate Fuentes, Yesica:

- Harana Mancilla, Rafael:

- Pizarro López, Eduardo:

- Portillo Sánchez, Alonso:

- Sevillano Barea, Alejandro: