# Política de Pull Requests del Equipo

## Integrantes del Equipo y Roles

|  |  |
| --- | --- |
| Nombre Completo | Rol Asignado |
| MEZA VILORIO AMAIRANY | Control de documentación |
| ROMERO AGUILAR LUIS SALVADOR | Revisión de código (QA) |
| TORRES BERNABÉ MOISÉS | Programador Backend |
| XOCUA MÁRQUEZ CÉSAR HÉCTOR | Programador Frontend |
| ZAMORA VEGA LUIS ÁNGEL | Coordinador general / DevOps |

## Política de Pull Requests

Para asegurar un flujo de trabajo eficiente y mantener la calidad del código, todo el equipo debe seguir las siguientes normas al hacer Pull Requests (PR):  
  
**1. Crear una nueva rama desde 'main' antes de comenzar una tarea.  
2. Usar una nomenclatura clara para las ramas:** - feature/nombre-tarea  
 - fix/nombre-arreglo  
 - docs/nombre-documento  
**3. Hacer commits con mensajes en formato claro y descriptivo:** - feat: agrega nueva funcionalidad  
 - fix: corrige error en validación  
 - docs: actualiza documentación  
**4. Subir los cambios con:**  
 git add .  
 git commit -m "mensaje claro"  
 git push origin nombre-rama  
**5. Crear un Pull Request en GitHub con:** - Título claro usando el acrónimo (feat, fix, docs, etc.)  
 - Descripción de los cambios realizados  
 - Archivos modificados  
 - Captura de pantalla (si aplica)  
**6. Asignar al menos un revisor (QA o Coordinador)  
7. Esperar revisión y aprobación antes de hacer merge a 'main'  
8. Si hay conflictos, resolverlos antes de continuar**

## Acrónimos para Commits y PR

- feat: Nueva funcionalidad  
- fix: Corrección de errores  
- docs: Cambios en la documentación  
- refactor: Reestructuración del código sin cambiar su funcionalidad  
- test: Agregado o modificación de pruebas  
- chore: Cambios menores o tareas administrativas