

# CONVENCION PARA LA CREACION DE RAMAS



Para mantener un flujo organizado, utilizaremos una convención específica para la creación de ramas según la funcionalidad y la tecnología que estemos desarrollando.

- **Convención:** <type>/<short-description>
- **Ejemplos:**
  - feature/add-leave-endpoint
  - bugfix/fix-date-validation
  - hotfix/urgent-navbar-adjustment
  - chore/setup-eslint
  - docs/update-readme

## Tipos de Cambios

Los tipos principales que utilizaremos para categorizar nuestras ramas serán:

- **feature:** Para nuevas funcionalidades.
- **bugfix:** Para correcciones de errores.
- **hotfix:** Para correcciones urgentes que deben implementarse rápidamente.
- **chore:** Para tareas de mantenimiento (configuración, ajustes menores).
- **docs:** Para documentación.
- **test:** Para agregar o modificar pruebas.

## Flujo de Trabajo (Workflow)

### 1. Crear una nueva Rama

Siempre comienza creando una nueva rama basada en la rama de desarrollo(dev):

```
git switch dev
```

```
git pull
```

```
git switch -c <type>/<short-description>
```

Ejemplo:

```
git switch -c feature/add-statistics-component
```

## 2. Realizar Cambios en el Código

Trabaja en tu rama haciendo cambios y asegurándote de mantener el código limpio y de realizar commits pequeños y específicos.

## 3. Hacer Commits con Convenciones

Sigue el formato de commits en inglés con prefijo y scope (Ver Doc. Plantilla de Prompts)

## 4. Actualizar la Rama con la Última Versión de dev

Antes de crear un Pull Request, asegúrate de que tu rama esté sincronizada con dev:

```
git pull origin dev
```

## 5. Subir la Rama al Repositorio

Sube tu rama al repositorio remoto:

```
git push
```

## 6. Crear un Pull Request (PR)

- En GitHub, crea un nuevo PR desde tu rama hacia dev.
- Añade un título descriptivo (por ejemplo, Add daily check-in functionality for employees).
- Completa la descripción del PR detallando los cambios realizados, por qué se hicieron y cualquier otro comentario relevante.
- Asigna revisores y añade etiquetas si es necesario (por ejemplo, feature).
- Revisores:
  - Frontend: @Jonnatha Figueira
  - Backend: @Lorena Acosta

## 7. Revisión de Código (Code Review)

- Los miembros del equipo revisarán el PR y dejarán comentarios o sugerencias.
- Si se solicitan cambios, realiza los ajustes necesarios en tu rama, haz commit y sube los cambios.
- Actualiza el PR hasta que sea aprobado.

## 8. Fusión del PR

Una vez que el PR haya sido revisado y aprobado, podrás fusionarlo a la rama dev.

## 9. Eliminar la Rama (Opcional)

Después de fusionar, elimina la rama para mantener limpio el repositorio:

```
git branch -d <your-branch-name>
```

```
git push origin --delete <your-branch-name>
```

## 10. Actualizar tu Copia Local

Asegúrate de actualizar tu rama dev local:

```
git switch dev
```

```
git pull
```

## Buenas Prácticas

- **Mantén los commits pequeños y específicos**, cada commit debería tener un propósito claro.
- **Revisa los PR** con atención y proporciona comentarios constructivos.
- **Ejecuta pruebas** antes de enviar tus cambios.
- **Mantén la documentación** actualizada, especialmente si introduces cambios en la API o componentes clave.
- **Resuelve los conflictos** lo antes posible para evitar problemas en la integración.
- **Haz PR con frecuencia** en lugar de acumular muchos cambios en una sola rama.