

## Configuração Inicial

Certifique-se de que o Git está configurado corretamente em sua máquina:

Configure seu nome e e-mail:

```
git config --global user.name "Seu Nome"  
git config --global user.email "seuemail@exemplo.com"
```

Clone o repositório:

```
git clone <URL_DO_REPOSITORIO>  
cd <NOME_DO_REPOSITORIO>
```

Verifique o status do repositório:

```
git status
```

## Fluxo de Trabalho Geral

Criar um Branch

Cada nova funcionalidade ou correção deve ser desenvolvida em um branch separado.

Nomeie o branch de forma descritiva:

Exemplo: feature/nome-da-funcionalidade ou hotfix/nome-do-problema.

Comando para criar e alternar para um branch novo:

```
git checkout -b feature/nome-da-funcionalidade
```

Fazer Alterações e Commits

Após realizar alterações no código, siga os passos abaixo:

Adicione as alterações ao stage:

```
git add .
```

Crie um commit com uma mensagem descritiva:

```
git commit -m "Implementa funcionalidade X"
```

Atualizar o Branch com o Branch Base

Antes de finalizar o desenvolvimento, atualize seu branch com as alterações do branch base (develop):

Alterne para o branch base:

```
git checkout develop
```

Obtenha as últimas atualizações do repositório remoto:

```
git pull origin develop
```

Volte para o seu branch de trabalho e mescle o branch base:

```
git checkout feature/nome-da-funcionalidade  
git merge develop
```

### Enviar o Branch para o Repositório

Quando concluir sua funcionalidade, envie o branch para o repositório remoto:

```
git push origin feature/nome-da-funcionalidade
```

### Criar um Pull Request

No GitHub, crie um Pull Request (PR) para integrar o branch ao branch base (develop).

Certifique-se de:

- Adicionar uma descrição clara do que foi implementado.

- Atribuir revisores, se necessário.

## Integração e Merges

### Testar e Aprovar o PR

Antes de mesclar o PR:

- Realize testes unitários e de integração no branch.

- Resolva quaisquer conflitos apresentados.

### Mesclar o Branch

Após aprovação, mescle o branch ao develop:

- O merge pode ser feito no GitHub pela interface gráfica.

### Promover para o Branch Principal

Quando o develop estiver estável e testado, promova as alterações para o main:

Altere para o branch main:

```
git checkout main
```

Mescle o branch develop:

```
git merge develop
```

Envie para o repositório remoto:

```
git push origin main
```

## Tratamento de Hotfixes

Se for necessário corrigir um problema crítico em produção, siga este fluxo:  
Crie um branch hotfix:

```
git checkout -b hotfix/nome-do-problema
```

Faça as alterações necessárias e commit:

```
git add .  
git commit -m "Corrige problema crítico em produção"
```

Mesclle o hotfix no main e no develop:

```
git checkout main  
git merge hotfix/nome-do-problema  
git push origin main
```

```
git checkout develop  
git merge hotfix/nome-do-problema  
git push origin develop
```

Delete o branch hotfix:

```
git branch -d hotfix/nome-do-problema  
git push origin --delete hotfix/nome-do-problema
```

## Comandos Úteis

Ação	Comando
Verificar o status do repositório	git status
Verificar histórico de commits	git log
Criar um branch novo	git checkout -b nome-do-branch
Alternar entre branches	git checkout nome-do-branch
Adicionar alterações ao stage	git add .
Criar um commit	git commit -m "Mensagem do commit"
Mesclar um branch	git merge nome-do-branch
Enviar branch para repositório	git push origin nome-do-branch
Obter alterações do repositório	git pull origin nome-do-branch