

Tutorial de Fluxo de Trabalho Git

Este tutorial descreve o fluxo de trabalho Git recomendado para este projeto, garantindo colaboração eficiente e controle de versão.

Visão Geral do Fluxo

Utilizamos um fluxo baseado em branches, geralmente inspirado no Gitflow, mas simplificado para este projeto. As branches principais são:

- **main** : Contém o código de produção estável. Ninguém deve fazer commit diretamente aqui.
- **develop** (Opcional, mas recomendado): Branch de integração principal para funcionalidades. Se não existir, use **main** como base para features.
- **feature/<nome-descriptivo>** : Branches criadas a partir de **develop** (ou **main**) para desenvolver novas funcionalidades.
- **fix/<nome-descriptivo>** : Branches criadas a partir de **develop** (ou **main**) para corrigir bugs.

O fluxo geral é:

1. Criar uma branch **feature** ou **fix** a partir da branch base (**develop** ou **main**).
2. Desenvolver e fazer commits na sua branch.
3. Abrir um Pull Request (PR) para mesclar sua branch de volta na branch base.
4. Após revisão e aprovação, o PR é mesclado.

Veja o diagrama visual: [Fluxo de Trabalho Git](#)

Comandos Essenciais

Assumindo que você já clonou o repositório (`git clone URL_DO_REPOSITORIO`) e está no diretório do projeto.

1. Atualizar sua Cópia Local

Antes de iniciar qualquer trabalho, certifique-se de que sua branch base local (**main** ou **develop**) está atualizada com o repositório remoto.

```
# Mude para a branch base (ex: main)
git checkout main

# Baixe as últimas alterações do repositório remoto
git pull origin main
```

(Substitua `main` por `develop` se essa for sua branch base)

2. Criar uma Nova Branch

Crie uma branch específica para sua tarefa (nova funcionalidade ou correção).

```
# Crie e mude para a nova branch a partir da base atual (ex:
main)
git checkout -b feature/adiciona-login-google

# Ou para uma correção:
git checkout -b fix/corrige-layout-responsivo
```

Use nomes descritivos e prefixos como `feature/` ou `fix/`.

3. Desenvolver e Fazer Commits

Faça suas alterações no código. Adicione e faça commit das suas mudanças frequentemente com mensagens claras.

```
# Verifique o status dos arquivos modificados
git status

# Adicione os arquivos que deseja incluir no commit
git add caminho/para/arquivo_modificado.ts
git add outro/arquivo.html
# Ou adicione todos os arquivos modificados (use com cuidado)
# git add .

# Faça o commit com uma mensagem descritiva
git commit -m "Implementa autenticação inicial com Google"
```

Boas Práticas para Mensagens de Commit: * Use o imperativo (ex: "Adiciona", "Corrige", "Refatora"). * Mantenha a linha de assunto curta (máx 50 caracteres). * Se necessário, adicione um corpo mais detalhado após uma linha em branco.

4. Enviar sua Branch para o Repositório Remoto

Quando estiver pronto para compartilhar seu trabalho ou abrir um Pull Request, envie sua branch local para o repositório remoto (geralmente `origin`, que é o seu fork ou o repositório principal se tiver permissão).

```
# Envia a branch 'feature/adiciona-login-google' para o 'origin'
git push origin feature/adiciona-login-google
```

Na primeira vez que enviar uma nova branch, você pode precisar usar `git push --set-upstream origin nome-da-sua-branch`.

5. Abrir um Pull Request (PR)

Vá para a interface web do seu serviço Git (GitHub, GitLab, etc.).

1. Navegue até o repositório original (não o seu fork, se aplicável).
2. Geralmente, haverá um aviso sugerindo criar um PR a partir da sua branch recém-enviada. Clique nele.
3. Se não houver aviso, vá para a seção "Pull Requests" e clique em "New Pull Request".
4. Selecione a branch base (ex: `main` ou `develop`) e a branch que você deseja mesclar (ex: `feature/adiciona-login-google`).
5. Dê um título claro e uma descrição detalhada ao PR, explicando o que foi feito e por quê. Referencie Issues relacionadas (ex: `Closes #42`).
6. Crie o Pull Request.

6. Revisão e Merge

Sua PR será revisada por outros membros da equipe ou mantenedores. Eles podem solicitar alterações.

- Se forem necessárias alterações, faça novos commits na sua branch local e envie-os (`git push origin nome-da-sua-branch`). O PR será atualizado automaticamente.
- Após a aprovação, um mantenedor mesclará (merge) sua PR na branch base.

7. Limpeza (Opcional)

Após o merge da sua PR, você pode deletar sua branch local e remota, se desejar.

```
# Mude de volta para a branch base
git checkout main

# Delete a branch local
git branch -d feature/adiciona-login-google

# Delete a branch remota (no origin)
git push origin --delete feature/adiciona-login-google
```

Este tutorial cobre os comandos básicos. O Git é uma ferramenta poderosa com muitos outros recursos. Consulte a [documentação oficial do Git](#) para aprender mais.