

Deploy de alterações do código via git

Deploy de alterações do código via git

- Fluxo de Desenvolvimento com GitLab

Fluxo de Desenvolvimento com GitLab

1. Criar uma Issue

Antes de iniciar qualquer desenvolvimento, deve ser criada uma Issue no GitLab descrevendo a atividade a ser realizada.

A Issue deve conter:

- Título claro e objetivo;
- Descrição detalhada da demanda;
- Critérios de aceitação, quando aplicável;
- Labels apropriadas (feature, bug, hotfix, etc.).

Após a criação, o GitLab atribuirá um identificador único (ID) à Issue, por exemplo:

#123

Esse ID será utilizado na nomenclatura da branch e nas mensagens de commit.

2. Criar uma Branch

A branch deve ser criada utilizando o ID da Issue.

Para novas funcionalidades

Padrão:

feature/<id-da-issue>-<descricao>

Exemplo:

```
feature/123-cadastro-de-usuarios
```

Para correções urgentes (Hotfix)

Padrão:

```
hotfix/<id-da-issue>-<descricao>
```

Exemplo:

```
hotfix/124-correcao-validacao-cpf
```

3. Atualizar as Referências Locais

Antes de trocar para a nova branch, atualize as referências do repositório:

```
git pull
```

4. Trocar para a Branch de Trabalho

Após a criação da branch:

```
git switch nome-da-branch
```

Exemplo:

```
git switch feature/123-cadastro-de-usuarios
```

5. Recuperar Alterações Temporariamente Armazenadas

Caso tenha utilizado stash anteriormente:

```
git stash pop
```

6. Verificar Arquivos Alterados

Antes de realizar o commit:

```
git status
```

7. Corrigir Padrão de Código

Caso existam inconsistências no padrão de escrita do código, execute:

```
make pint-fixer
```

Esse comando aplica automaticamente as regras de formatação definidas pelo projeto.

8. Adicionar Arquivos ao Commit

Após validar as alterações:

```
git add .
```

9. Verificar Novamente o Status

Confirme os arquivos que serão enviados:

```
git status
```

10. Realizar o Commit

A mensagem do commit deve obrigatoriamente referenciar a Issue correspondente.

Padrão:

```
git commit -m "#<id> descrição da alteração"
```

Exemplos:

```
git commit -m "#123 adiciona cadastro de usuários"
```

```
git commit -m "#124 corrige validação de CPF"
```

11. Enviar Alterações para o Repositório Remoto

```
git push
```

Caso seja o primeiro push da branch:

```
git push -u origin nome-da-branch
```

Exemplo:

```
git push -u origin feature/123-cadastro-de-usuarios
```

12. Criar Merge Request

Após concluir o desenvolvimento:

1. Acesse o GitLab;
2. Crie um Merge Request da branch criada para a branch principal do projeto;
3. Vincule a Issue correspondente;
4. Solicite revisão de código;
5. Aguarde aprovação e merge.

Resumo do Fluxo

```
git pull
git switch feature/123-cadastro-de-usuarios
git stash pop
git status
make pint-fixer
git add .
git status
git commit -m "#123 adiciona cadastro de usuários"
git push
```