

DevSuperior - Desafio Git Merge

Documentação Completa

Guia de Versionamento com Git e GitHub

Gerado em: 20 de November de 2025 às 18:02

■ Conteúdo

- Descrição do Projeto
- Estrutura do Repositório
- Como Começar
- Fluxo de Trabalho Git
- Gerenciamento de Branches

Conteúdo Completo do README

DevSuperior - Desafio Git Merge

■ Descrição do Projeto

Este repositório foi criado como parte do curso de **Fundação de Programação da DevSuperior com foco em versionamento de código utilizando Git e GitHub**. O objetivo é demonstrar o uso de comandos Git essenciais e a resolução de conflitos de merge.

■ Objetivos

- Aprender os fundamentos do Git e controle de versão
- Dominar comandos essenciais do Git
- Praticar criação e gerenciamento de branches
- Resolver conflitos de merge em repositórios
- Implementar boas práticas de versionamento

■ Estrutura do Projeto

DevSuperiorDesafioGitMerge/

■■■ READMECOMPLETO.md # Este arquivo

■■■ docs/

■ ■■■■ FUNDAMENTOSGIT.md # Conceitos fundamentais do Git

■ ■■■■ COMANDOSGIT.md # Referência de comandos Git

■■■ blog.html # Página blog

■■■ catalog.html # Página catálogo

■■■ index.html # Página inicial

■■■ sobre.html # Página sobre

■■■ image/ # Imagens do projeto

■ Como Começar

Pré-requisitos

- Git instalado na máquina

- Conta no GitHub
- Visual Studio Code ou outro editor de código

Clonar o Repositório

```
git clone https://github.com/gabrielsalesdavid/DevSuperiorDesafioGitMerge.git
```

```
cd DevSuperiorDesafioGitMerge
```

Estruturar o Ambiente

Inicializar Git (se necessário)

```
git init
```

Criar branch principal

```
git checkout -b main
```

Adicionar repositório remoto

```
git remote add origin git@github.com:gabrielsalesdavid/DevSuperiorDesafioGitMerge.git
```

■ Documentação

Para aprender mais sobre Git e seus comandos, consulte a documentação disponível:

- **[Fundamentos do Git](docs/FUNDAMENTOS_GIT.md)** - Conceitos essenciais e estrutura do Git
- **[Comandos Git](docs/COMANDOS_GIT.md)** - Referência completa de comandos com exemplos

■ Fluxo de Trabalho

1. Status: Verificar alterações

```
```bash
```

```
git status
```

```
```
```

2. Adicionar: Preparar arquivos para commit

```
```bash
```

```
git add .
```

```
```
```

3. Commit: Registrar alterações

```
```bash
```

```
git commit -m "Descrição das alterações"
```

```
```
```

4. Push: Enviar para o repositório remoto

```
```bash
```

```
git push origin [branch-name]
```

```
```
```

5. Pull: Obter atualizações do repositório remoto

```
```bash
```

```
git pull origin main
```

```
```
```

■ Gerenciamento de Branches

Listar branches locais

```
git branch
```

Listar todas as branches

```
git branch -a
```

Criar nova branch

```
git checkout -b nome-da-branch
```

Mudar de branch

```
git checkout nome-da-branch
```

Deletar branch

`git branch -d nome-da-branch`

■ Merge e Resolução de Conflitos

Fazer merge de uma branch

`git merge nome-da-branch`

Resolver conflitos manualmente nos arquivos

Depois, adicionar e fazer commit

`git add .`

`git commit -m "Resolve conflito de merge"`

■ Tópicos Cobertos

- ■ Inicialização de repositórios
- ■ Configuração remota
- ■ Status e alterações
- ■ Staging e commits
- ■ Branches e checkout
- ■ Merge e conflitos
- ■ Push e Pull
- ■ Histórico de commits

■ Contribuindo

1. Crie uma branch para sua feature (``git checkout -b feature/nova-funcionalidade``)
2. Commit suas alterações (``git commit -am 'Adiciona nova funcionalidade'``)

3. Push para a branch (`git push origin feature/nova-funcionalidade`)

4. Abra um Pull Request no GitHub

■ Recursos Adicionais

- [Documentação Oficial do Git](https://git-scm.com/doc)
- [GitHub Guides](https://guides.github.com)
- [DevSuperior](https://devsuperior.com.br)

■ Autor

Gabriel Sales David

- GitHub: [@gabrielsalesdavid](https://github.com/gabrielsalesdavid)

■ Licença

Este projeto é parte do curso DevSuperior e está disponível para fins educacionais.

Última atualização: 20 de novembro de 2025

—

DevSuperior - Fundação de Programação | Git e GitHub | 2025