

DevOps

Aula 01 – Parte 2

Objetivo da Aula



O foco será nos comandos básicos do Git e nas práticas colaborativas, fundamentais para o trabalho em equipe e o gerenciamento eficiente de código e análises.

ESTRUTURA DE UM REPOSITÓRIO



- ❑ **.git/** : Armazena os metadados do repositório.
- ❑ **.gitignore**: Lista de arquivos a serem ignorados pelo Git.
- ❑ **README.md**: Documento descritivo sobre o projeto.
- ❑ **Arquivos de código e scripts**
- ❑ **Dados**: armazenado separadamente para evitar arquivos grandes.
- ❑ **Resultados**: Gráficos, relatórios ou dashboards gerados pela análise.



CONCEITOS FUNDAMENTAIS



Commits

- ❖ um snapshot das alterações feitas no seu projeto
- ❖ tem um identificador único (hash)
- ❖ uma mensagem descreve as mudanças realizadas

Branches

Merges

The git logo, a red circle with the word "git" in white lowercase letters.

CONCEITOS FUNDAMENTAIS



Commits

Branches

- ❖ linha de desenvolvimento independente.
- ❖ permite que se trabalhe em novas funcionalidades ou correções sem afetar o código principal (geralmente o branch `main` ou `master`)

Merges



CONCEITOS FUNDAMENTAIS



Commits

Branches

Merges

- ❖ processo de combinar as alterações de um branch em outro.
- ❖ quando se finaliza o desenvolvimento de uma funcionalidade e deseja integrá-la ao branch principal.



REPOSITÓRIOS LOCAIS E REMOTOS



Repositório local:

- ❖ Criado no computador do usuário com `git init`.
- ❖ Contém o histórico do projeto

Repositório remoto:

- ❖ Hospedado em plataformas como GitHub e GitLab
- ❖ Serve para colaboração e backup.



ESTRUTURA DE DADOS DO GIT

Blobs:

- ❖ Armazena o conteúdo dos arquivos.

Trees:

- ❖ Representa diretórios e estrutura do projeto.

Commits:

- ❖ Capturam mudanças no projeto e conectam versões do código.



ÁREAS DO GIT

Working Directory

- ❖ Onde os arquivos são editados.

Staging Area

- ❖ Onde as mudanças são preparadas antes de um commit.

Repository

- ❖ Onde as versões confirmadas são armazenadas.



EXEMPLOS DE WORKFLOWS



Fluxo Individual

1. Criar um repositório (`git init`)
2. Criar e modificar arquivos (`git add`, `git commit`)
3. Versionar mudanças e voltar a versões anteriores (`git log`, `git checkout`)



EXEMPLOS DE WORKFLOWS



Colaboração entre Analistas

1. Clonar o repositório remoto (``git clone``)
2. Criar um branch para cada nova análise (``git checkout -b nova-analise``)
3. Fazer commits frequentes e descritivos (``git commit -m "Explicação clara da mudança"``)
4. Mesclar mudanças para o branch principal (``git merge``)
5. Resolver conflitos quando necessário



Convenções de Mensagens de Commit

1. Devem ser curtas e descritivas.
2. Exemplo:
 - ✓ "Corrige erro na importação de dados CSV"
 - ✗ "Correção"



BOAS PRÁTICAS



Estratégias de Branching

1. Feature Branching

- ❖ Cada nova funcionalidade recebe um branch separado.

2. Git Flow

- ❖ Usa branches fixos (`main`, `develop`, `feature`, `release`, `hotfix`).

3. Trunk-Based Development

- ❖ Desenvolvedores trabalham diretamente no `main` com pequenas mudanças.

The word "git" in white lowercase letters inside a red circle.

git



BOAS PRÁTICAS



Pull Requests

- ❖ use para revisar o código antes de fazer o merge.
- ❖ facilita a colaboração
- ❖ garante a qualidade do código

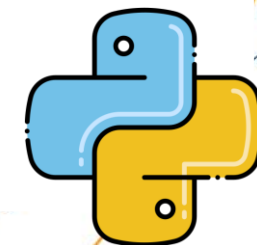


GIT E GITBUB: ATIVIDADE 1



URL da Atividade 1 Configuração e Comandos Básicos

<https://github.com/luisrodrigonet/DevOps-UC1/blob/main/README.md>

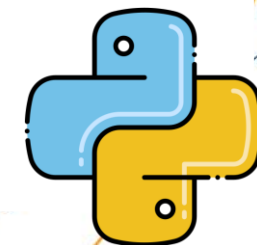


GIT E GITBUB: ATIVIDADE 2



URL da Atividade 2 Exercício Prático

<https://github.com/luisrodrigonet/DevOps-UC1/blob/main/README.md>

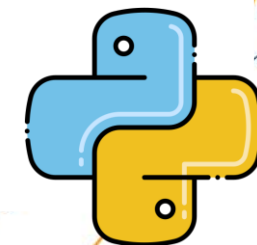


GIT E GITBUB: ATIVIDADE 3



URL da Atividade 3
Fluxo de Trabalho Colaborativo

<https://github.com/luisrodrigonet/DevOps-UC1/blob/main/README.md>

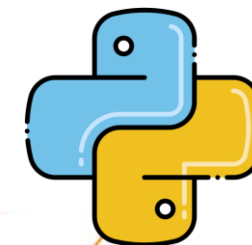


GIT E GITBUB: ATIVIDADE 4



URL da Atividade 4 Exercício Prático 2

<https://github.com/luisrodrigonet/DevOps-UC1/blob/main/README.md>



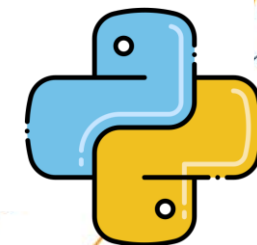
GIT E GITBUB: DESAFIO 1



URL do Desafio 1

Projeto de Análise de Dados Colaborativo

<https://github.com/luisrodrigonet/DevOps-UC1/blob/main/README.md>

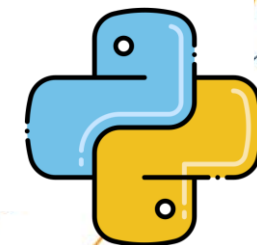


GIT E GITBUB: DESAFIO 2



URL do Desafio 2 Desafio de Branches

<https://github.com/luisrodrigonet/DevOps-UC1/blob/main/README.md>



DevOps



DevOps

Aula 01 – Parte 1

