



Lista de Exercícios – Git/GitHub 2025.1

Aula 1: Introdução ao Git e ao GitHub e conceitos básicos

Objetivo: Compreender os conceitos fundamentais do Git e GitHub e realizar configurações iniciais.

Exercício 1: Configuração do Git

- Instale o Git em seu computador.
- Configure seu nome e e-mail no Git usando os comandos:

```
1 git config --global user.name "Seu Nome"
2 git config --global user.email "seu.email@exemplo.com"
```

- Verifique as configurações com:

```
1 git config --list
```

- **Entrega:** Capture a saída do comando `git config -list` e envie ao instrutor.

Exercício 2: Inicializando um Repositório

- Crie uma pasta chamada `meu-projeto`.
- Dentro dela, inicialize um repositório Git com:

```
1 git init
```

- Crie um arquivo de texto chamado `README.md` com uma breve descrição do projeto.
- Adicione o arquivo ao controle de versão com:

```
1 git add README.md
```

- Faça um commit com a mensagem: `Adiciona README inicial`.
- **Entrega:** Mostre o status do repositório com:

```
1 git status
```

e envie a saída.

Exercício 3: Primeiro Commit

- Crie um arquivo `index.html` com um código HTML simples.
- Adicione o arquivo ao controle de versão e faça um commit com a mensagem: `Adiciona página inicial`.
- Visualize o histórico de commits com:

```
1 git log
```

- **Entrega:** Envie a saída do comando `git log`.



Aula 2: Comandos no Git e repositórios remotos

Objetivo: Aprender comandos essenciais do Git e conectar repositórios locais com remotos no GitHub.

Exercício 1: Trabalhando com .gitignore

- No repositório `meu-projeto` criado na Aula 1, adicione um arquivo chamado `temp.txt`.
- Crie um arquivo `.gitignore` e adicione `temp.txt` a ele para que o Git ignore esse arquivo.
- Tente adicionar `temp.txt` com:

```
1 git add temp.txt
```

e verifique o status com:

```
1 git status
```

- **Entrega:** Mostre o conteúdo do arquivo `.gitignore` e a saída de `git status`.

Exercício 2: Conectando ao GitHub

- Crie um repositório vazio no GitHub chamado `meu-projeto`.
- Conecte seu repositório local ao remoto com:

```
1 git remote add origin <URL-do-reposit rio>
```

- Envie as alterações do repositório local para o GitHub com:

```
1 git push origin main
```

- **Entrega:** Forneça o link do repositório no GitHub.

Exercício 3: Clonando e Atualizando

- Clone o repositório `meu-projeto` em uma nova pasta usando:

```
1 git clone <URL-do-reposit rio>
```

- Faça uma alteração no arquivo `README.md` no repositório clonado, adicione e faça um commit.
- Envie as alterações com:

```
1 git push
```

- No repositório original, baixe as alterações com:

```
1 git pull
```

- **Entrega:** Mostre a saída de `git log` após o `git pull`.



Aula 3: Branches e trabalho colaborativo

Objetivo: Entender e utilizar branches para colaborar em projetos.

Exercício 1: Criando e Gerenciando Branches

- No repositório `meu-projeto`, crie uma nova branch chamada `feature-nova-pagina` com:

```
1 git branch feature-nova-pagina
```

- Mude para a nova branch com:

```
1 git checkout feature-nova-pagina
```

- Crie um arquivo `sobre.html` com um conteúdo simples e faça um commit.
- Volte para a branch principal (`main`) e verifique se o arquivo `sobre.html` não está presente.
- Entrega:** Mostre a lista de branches com:

```
1 git branch
```

e o conteúdo da pasta na branch `main`.

Exercício 2: Merge de Branches

- Na branch `feature-nova-pagina`, adicione mais uma alteração ao arquivo `sobre.html`.
- Faça um commit com a mensagem: `Atualiza página sobre`.
- Volte para a branch `main` e combine as alterações da branch `feature-nova-pagina` usando:

```
1 git merge feature-nova-pagina
```

- Entrega:** Mostre a saída de:

```
1 git log --oneline
```

após o merge.

Exercício 3: Pull Request e Conflitos

- Envie a branch `feature-nova-pagina` para o GitHub com:

```
1 git push origin feature-nova-pagina
```

- No GitHub, crie um Pull Request para mesclar `feature-nova-pagina` na branch `main`.
- Em um cenário simulado de conflito:
 - Na branch `main`, altere o arquivo `README.md` e faça um commit.
 - Na branch `feature-nova-pagina`, altere a mesma linha do `README.md` com um conteúdo diferente e faça um commit.
 - Tente mesclar as branches com `git merge` e resolva o conflito manualmente editando o arquivo.

- Entrega:** Mostre o arquivo `README.md` após a resolução do conflito e a saída de:

```
1 git log --oneline
```



Observações

- **Ferramentas necessárias:** Git instalado, conta no GitHub, editor de texto (ex.: VS Code).
- **Entrega dos exercícios:** Os alunos devem enviar as saídas dos comandos (capturas de tela ou texto) e links dos repositórios no GitHub, quando aplicável.
- **Dica:** Teste os comandos em um ambiente local antes de enviar para o GitHub para evitar erros.