

Exercício Prático de Git

Vamos criar um simples projeto de site utilizando HTML, CSS e JavaScript, e aplicar as boas práticas de Git listadas anteriormente.

Etapas do Exercício:

1.Inicialização do Repositório:

- Crie um repositório no GitHub ou GitLab chamado **meu-site**.
- Clone o repositório localmente:

```
git clone https://github.com/seu-usuario/meu-site.git
```

- Navegue até a pasta do repositório:

```
cd meu-site
```

- Inicialize o repositório localmente:

```
git init
```

2.Estrutura de Branches:

- Crie um branch **develop**:

```
git checkout -b develop
```

- Crie um novo branch ``feature/header`` a partir de ``develop``:

```
git checkout -b feature/header
```

- No branch ``feature/header``, crie o arquivo ``index.html`` com o seguinte conteúdo:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Meu Site</title>
</head>
<body>
  <header>
    <h1>Bem-vindo ao Meu Site</h1>
  </header>
</body>
</html>
```

- Adicione o arquivo ao Git:

```
git add index.html
```

- Faça um commit com uma mensagem clara:

```
git commit -m "Adiciona cabeçalho ao index.html"
```

3.Uso de Commits:

- Crie um arquivo `styles.css` e adicione o seguinte conteúdo:

```
header {
  background-color: #333;
  color: white;
  padding: 20px;
  text-align: center;
}
```

- Adicione o arquivo ao Git:

```
git add styles.css
```

- Faça outro commit:

```
git commit -m "Adiciona estilo ao cabeçalho"
```

4.Criação de Pull Request (PR):

- Faça o push do branch **feature/header** para o repositório remoto:

```
git push origin feature/header
```

- Crie um pull request no GitHub para mesclar o branch **feature/header** no branch **develop**.
- Descreva o que foi feito no PR, mencionando as mudanças feitas.

5.Simulação de Conflito:

- Volte para o branch **develop**:

```
git checkout develop
```

- Crie um novo branch **feature/footer** a partir de **develop**:

```
git checkout -b feature/footer
```

- No branch **feature/footer**, adicione um rodapé ao arquivo **index.html**:

```
<footer>
```

```
    <p>&copy; 2024 Meu Site. Todos os direitos  
reservados.</p>
```

```
</footer>
```

- Adicione e comite as mudanças:

```
git add index.html
```

```
git commit -m "Adiciona rodapé ao index.html"
```

- Faça o merge do branch `develop` no branch `feature/footer`:

```
git checkout feature/footer
```

```
git merge develop
```

- Se houver conflitos, o Git exibirá os arquivos com conflitos. Edite esses arquivos para resolver os conflitos e finalize o merge:

```
git add index.html
```

```
git commit -m "Resolve conflitos no merge do develop"
```

6.Tags e Versionamento:

- Após mesclar todas as mudanças, crie uma tag para a primeira versão do site:

```
git tag -a v1.0.0 -m "Versão 1.0.0 do site"
```

- Faça o push da tag para o repositório remoto:

```
git push origin v1.0.0
```

- Crie um arquivo `README.md` com a descrição do projeto. Exemplo de conteúdo:

```
# Meu Site
```

Este é um projeto simples de um site com HTML, CSS e JavaScript.

```
## Funcionalidades:
```

```
-Cabeçalho com estilo
```

```
-Rodapé
```

7.Finalização:

- Crie um arquivo **.gitignore** para evitar versionar arquivos desnecessários (por exemplo, se você estiver usando Node.js, adicione **node_modules/**):

```
echo "node_modules/" >> .gitignore
```

- Adicione e faça o commit:

```
git add .gitignore
```

```
git commit -m "Adiciona .gitignore para ignorar  
node_modules"
```

- Faça o push do branch `develop` para o repositório remoto:

```
git push origin develop
```