

LINGUAGEM DE PROGRAMAÇÃO 4

NOME: _____ TURMA: _____

EXERCÍCIO 01 – PRIMEIRO COMMIT

Objetivo:

Fazer o clone do repositório, configurar autor, alterar o `README.md` e enviar a alteração.

Passos:

1. Acesse seu repositório no navegador. Se o seu repositório remoto é o `grupo01-projeto`, então acesse o endereço abaixo no navegador.

```
https://github.com/colégio-pedro-II-projetos-2025/grupo01-projeto.git
```

2. No git bash, navegue até a pasta onde você deseja armazenar o seu projeto.

```
cd "C:/caminho/do/projeto"
```

3. Faça o clone do repositório e depois acesse a pasta baixada `grupo01-projeto` onde o seu repositório local foi criado. Aqui você terá acesso a todos os arquivos que vieram do repositório remoto.

```
git clone https://github.com/colégio-pedro-II-projetos-2025/grupo01-projeto.git  
cd grupo01-projeto
```

4. Configure seu nome e email. Isso é importante para definir quem será o autor dos futuros commits.

```
git config user.name "Seu Nome Completo"  
git config user.email "seuemail@gmail.com"
```

5. Edite o arquivo `README.md` com seu grupo e nomes dos integrantes. Isso pode ser feito em qualquer editor de texto ou IDE, inclusive Visual Studio ou Visual Code.
6. Verifique as mudanças. Você notará que um indicador de "modified" sinaliza que o arquivo `README.md` foi alterado.

```
git status
```

6. Adicione o arquivo alterado à área de preparação. Lembre-se, a área de preparação vai conter os arquivos que serão “commitados”.

```
git add README.md
```

7. Crie o commit e dê um nome para ele. O nome do commit geralmente é breve e indica o que a alteração representa.

```
git commit -m "Definir grupo"
```

8. Visualize o histórico local. Esse comando é útil apenas para que você veja os commits no seu repositório local.

```
git log
```

9. Envie para o repositório remoto. Ao executar este comando, os novos commits que você tem localmente irão para o repositório remoto! Isso significa que todos os demais colaboradores terão acesso a eles.

```
git push
```

EXERCÍCIO 02 – INSERÇÃO DO PROJETO INICIAL

Objetivo:

Adicionar todos os arquivos do projeto à pasta clonada e fazer o primeiro commit real do código.

Passos:

1. Copie os arquivos do seu projeto para dentro da pasta `grupo0x-projeto`. Considere que `x` é o número do seu grupo.
2. Verifique os arquivos detectados. Você irá notar que esses arquivos estão marcados como “untracked files”, ou seja, não rastreados pelo git.

```
git status
```

3. Adicione todos os arquivos para a área de preparação. Isso fará com que o git os rastreie.

```
git add .
```

4. Execute `git status` mais uma vez e veja que esses arquivos agora estão na cor verde, preparados para serem commitados se necessário.

```
git status
```

5. Crie o commit e dê uma mensagem como “Projeto inicial”.

```
git commit -m "Projeto inicial"
```

6. Visualize o histórico. Note que o último commit é o seu. Novamente, você commitou no seu repositório local. Se você acessar o repositório remoto ainda não verá este commit.

```
git log
```

7. Envie para o repositório remoto o seu commit.

```
git push
```

EXERCÍCIO 03 – FAZENDO NOVAS ALTERAÇÕES NO PROJETO (FLUXO PRINCIPAL DE TRABALHO)

Objetivo:

Fazer uma modificação real no projeto (ex: corrigir texto, mudar cor, etc), com pull, commit e push. Praticar o fluxo principal de trabalho do git e github.

Passos:

0. Sempre que você começar a trabalhar no seu repositório remoto, traga as alterações do repositório remoto. Isso é importante pois algum colaborador pode ter enviado commits e portanto você quer trabalhar sempre na versão mais atual do projeto.

```
git pull
```

1. Altere qualquer parte do projeto (ex: um botão, texto, nome do formulário etc) usando o Visual Studio (ou qualquer ambiente de desenvolvimento).
2. Verifique o que mudou:

```
git status
```

3. Adicione os arquivos alterados à área de preparação.

```
git add .
```

4. Crie um commit com descrição apropriada:

```
git commit -m "Envia senha criptografada para o banco."
```

5. Visualize o histórico para ver que o seu commit está no repositório local.

```
git log
```

6. Envie a alteração para o repositório remoto:

```
git push
```

CONSIDERAÇÕES FINAIS

Durante o desenvolvimento de um projeto com Git, existe um conjunto de passos que se repete sempre que uma nova alteração é feita. Esse ciclo começa com o comando `git pull`, que deve ser executado antes de qualquer modificação no código. Ele serve para trazer as alterações mais recentes que estão no repositório remoto (no GitHub, por exemplo) para a cópia local do aluno. Isso evita conflitos e garante que o aluno esteja sempre trabalhando sobre a versão mais atual do projeto. Após essa sincronização, o aluno realiza suas alterações normalmente (no código, nas telas, nos arquivos de dados etc).

Feitas as alterações, o fluxo segue com os comandos `git status`, `git add`, `git commit` e `git push`. O `git status` permite visualizar quais arquivos foram modificados. O `git add` prepara esses arquivos para o `commit`. O `git commit` registra uma nova entrada no histórico local, com uma mensagem clara explicando a alteração. Por fim, o `git push` envia esses commits para o repositório remoto. Esse ciclo — `pull` → `status` → `add` → `commit` → `push` — se repete a cada alteração feita no projeto. Ele mantém o histórico limpo, o projeto sincronizado e permite que todos os membros do grupo acompanhem as contribuições de forma organizada e segura. Adotar esse padrão de trabalho é essencial para qualquer projeto em equipe ou com continuidade ao longo do tempo.

