

# UC13 - Realizar os Processos de Codificação, Manutenção e Documentação de Aplicativos Computacionais para Desktop

---

## Exercícios - Aula 01

### Objetivo

Este documento relata sobre a aula que tive na Aula 01, com a utilização das operações Git, incluindo a criação de commits com mensagens descritivas, reversão de alterações e manipulação de branches e merges.

---

## Exercício 1 - Criando um Histórico de Commits com Mensagens Descritivas

*Objetivo:* Aprender a estabelecer um histórico de commits organizado utilizando mensagens de commit.

*Passos:*

1. Criei uma nova pasta chamada git-historico usando o comando mkdir. Que tinha no passo a passo da documentação teste.
  2. Iniciei essa pasta como um repositório Git: `bash git init` depois verifiquei com o `git status`
  3. Criei três arquivos distintos:
    - notas.txt: Incluí informações relevantes.
    - resumo.md: Adicionei um resumo ou descrição.
    - tarefa.txt: Descrevi uma tarefa ou objetivo, como pedido na documentação.
  4. Realizei o commit de cada arquivo separadamente, utilizando mensagens de commit detalhadas que explicam o conteúdo e a intenção de cada alteração:  
`bash git add notas.txt git commit -m "Adicionando notas sobre o projeto no arquivo notas.txt"`  
`git add resumo.md git commit -m "Adicionando resumo do projeto no arquivo resumo.md"`  
`git add tarefa.txt git commit -m "Descrevendo a tarefa principal no arquivo tarefa.txt"`
  5. Exibi o histórico de commits para verificar as mensagens: `bash git log`
- 

## Exercício 2: Revertendo Alterações com git reset e git restore

*Objetivo:* Pratiquei a reversão de alterações não commitadas e commitadas no repositório Git.

*Passos:*

1. Criei uma nova pasta como suporte.
2. Iniciei essa pasta como um repositório Git: `bash git init`

3. Criei um arquivo `experimento.txt` e adicionei o conteúdo inicial.
  4. Adicionei e comitei o arquivo: `bash git add experimento.txt git commit -m "Adiciona conteúdo inicial no experimento.txt"`
  5. Modifiquei o arquivo `experimento.txt` (sem realizar o commit) e utilizei `git restore` para desfazer as alterações não commitadas: `bash git restore experimento.txt`
  6. Fiz mais uma modificação no arquivo e, em seguida, comitei: `bash echo "Nova alteração" >> experimento.txt git add experimento.txt git commit -m "Adicionando nova alteração no experimento.txt"`
  7. Usei `git reset --hard` para reverter o último commit: `bash git reset --hard HEAD~1`
- 

## Exercício 3: Criando Branches e Realizando Merge

*Objetivo:* Pratiquei a criação e o merge de branches, lidando com possíveis conflitos que poderia acontecer durante o processo.

*Passos:*

1. Criei uma nova pasta chamada `branch-teste`.
2. Inicializei a pasta como um repositório Git e criei um arquivo `principal.txt` com texto: `bash git init echo "Conteúdo inicial do principal.txt" > principal.txt git add principal.txt git commit -m "Adicionar conteúdo inicial no principal.txt na branch main"`
3. Criei uma nova branch chamada `melhorias`: `bash git checkout -b melhorias`
4. Adicionei um novo arquivo `novidades.md` e realizei o commit na branch `melhorias`: `bash echo "Informações sobre novidades" > novidades.md git add novidades.md git commit -m "Adicionando novidades.md com informações sobre atualizações"`
5. Retornei para a branch `main` e modifiquei o arquivo `principal.txt`, realizando o commit da alteração: `bash git checkout main echo "Novas informações no principal.txt" >> principal.txt git add principal.txt git commit -m "Atualiza principal.txt com novas informações na branch main"`
6. Tentei realizar o merge da branch `melhorias` na `main` e resolver quaisquer conflitos que surgissem: `bash git merge melhorias`
  - Fiz um commit para finalizar o merge
  - Eu tive dificuldade no encontro das pastas para conectar