

Atividade Prática: Corrigindo Erros com Git e Entendendo Rollbacks

Atividade Prática: Corrigindo Erros com Git e Entendendo Rollbacks

Objetivo

Desenvolver autonomia e domínio sobre o controle de versões, com foco em como desfazer alterações em diferentes etapas do ciclo de versionamento com Git (antes e depois do commit).

Instruções

1. Crie um repositório local usando git init.
2. Crie 3 arquivos:
 - info1.txt: Escreva seu nome, turma e data.
 - info2.txt: Escreva um conteúdo que aprendeu em Git até agora.
 - calculadora.py: Crie a função `somar(a, b)` que retorna `a - b` (erro proposital).
3. Faça o primeiro commit com esses três arquivos.
4. Corrija o erro em `calculadora.py`:
 - Crie uma branch chamada `correcao_bug`
 - Corrija o erro (`a - b` para `a + b`)
 - Faça o commit na branch `correcao_bug`
 - Faça merge da `correcao_bug` na `main`
5. Pesquise e defina os seguintes comandos Git (coloque as definições dentro do `info2.txt`):
 - `git log`
 - `git status`
 - `git diff`
 - `git restore`
 - `git checkout`
 - `git reset`

Atividade Prática: Corrigindo Erros com Git e Entendendo Rollbacks

- git revert

6. Na prática, aplique esses comandos em seu projeto:

- Teste como desfazer antes do commit (ex: git restore, git checkout)
- Teste como desfazer depois do commit (ex: git reset, git revert)
- Registre no info2.txt como cada comando funcionou no seu caso.

7. Apresente a evidência da correção e da prática dos comandos:

- Capture os comandos usados e os efeitos no terminal
- Ou escreva um resumo no info2.txt explicando o que cada comando fez no seu projeto

Reflexão Obrigatória

Inclua no info2.txt:

- Quando é mais seguro usar git reset e quando é melhor usar git revert?
- Qual a diferença prática entre corrigir um erro antes de fazer commit e depois?