Respostas da Lista de Exercícios de GitHub

Exercício 1: Configuração Inicial e Primeiro Repositório

1.1. Criar um Repositório Local

- **GitHub Desktop:**
  Criado repositório `meu-primeiro-git-repo` no caminho local especificado.

- **VS Code:**
  Repositório inicializado na pasta `meu-primeiro-git-repo`.

1.2. Criar seu Primeiro Arquivo e Fazer o Primeiro Commit

- **GitHub Desktop:**
  Criado `README.md` com conteúdo especificado.
  Commit realizado com mensagem: `Primeiro commit: Adiciona o arquivo README.md`.

- **VS Code:**
  `README.md` criado e commit realizado com mesma mensagem.

1.3. Conectar ao GitHub e Publicar o Repositório

- **GitHub Desktop:**
  Repositório publicado no GitHub (público/privado conforme selecionado).

- **VS Code:**
  Repositório publicado após autenticação no GitHub.

1.4. Verificar no GitHub

- Repositório `meu-primeiro-git-repo` visível no perfil GitHub com arquivo `README.md`.

Exercício 2: Trabalhando com Branches

2.1. Criar uma Nova Branch

- **GitHub Desktop:**
  Branch `feature/nova-secao` criada e selecionada automaticamente.

- **VS Code:**
  Branch `feature/nova-secao` criada e checkout realizado.

2.2. Criar um Novo Arquivo na Nova Branch

- **GitHub Desktop:**
  Arquivo `secao_importante.txt` criado com conteúdo especificado.
  Commit realizado com mensagem: `Adiciona nova seção de conteúdo`.

- **VS Code:**
  `secao_importante.txt` criado e commit realizado com mesma mensagem.

2.3. Publicar a Nova Branch no GitHub

- **GitHub Desktop:**
  Branch publicada com botão "Publish branch".
- **VS Code:**
  Branch publicada com "Publish Branch".

2.4. Verificar no GitHub

- Branch `feature/nova-secao` visível no repositório GitHub com arquivo `secao_importante.txt`.

Exercício 3: Mesclando Branches

3.1. Mudar para a Branch Principal (`main`)

- **GitHub Desktop/VS Code:**
  Checkout realizado para branch `main`.

3.2. Mesclar a Branch de Feature na `main`

- **GitHub Desktop:**
  Merge realizado com "Create a merge commit".
- **VS Code:**
  Merge concluído via "Merge Branch...".

3.3. Enviar as Alterações Mescladas para o GitHub

- **GitHub Desktop:**
  Alterações enviadas com "Push origin".
- **VS Code:**
  Sincronização realizada com "Sync Changes".

3.4. Excluir a Branch de Feature

- **GitHub Desktop:**
  Branch `feature/nova-secao` excluída localmente e remotamente.
- **VS Code:**
  Branch excluída localmente via ícone de lixeira; remotamente via GitHub web.

Exercício 4: Simulação e Resolução de Conflitos

4.1. Preparar para o Conflito

- Branch `bugfix/correcao-readme` criada.

- Conteúdo conflitante adicionado no `README.md` em ambas as branches (`main` e `bugfix`).

4.2. Simular e Resolver o Conflito

- **GitHub Desktop:**
  Conflito detectado no merge.
  `README.md` editado manualmente para conter:

```
# Meu Primeiro Repositório Git
Este é um repositório de exemplo para aprender Git e GitHub.
Esta linha foi adicionada diretamente na branch principal.
Esta linha foi adicionada na branch de correção de bug.
```

  Commit de merge realizado: `Resolve conflito de mesclagem no README.md`.

- **VS Code:**
  Conflito resolvido com "Accept Both Changes".
  Commit e sincronização realizados.

4.3. Excluir a Branch de Conflito

- Branch `bugfix/correcao-readme` excluída conforme passo 3.4.

Exercício 5: Verificando o Histórico e Status Final

Verificação Final:

- **GitHub Desktop:**
  - Histórico completo visível na aba "History" (incluindo commits de merge).
  - "Changes" vazio.
- **VS Code:**
  - Git Graph mostra histórico com branches mescladas.
  - Nenhuma alteração pendente no Source Control.
  - Ícone de sincronização indica estado atualizado.

Status no GitHub:

- Repositório contém:
  - `README.md` com conteúdo final (incluindo ambas as linhas adicionadas).
  - `secao_importante.txt` presente na branch `main`.
  - Branch `main` como única branch ativa (demais branches excluídas).
  - Histórico de commits reflete todas as operações realizadas.