

Lista de Exercícios de GitHub

Exercício 1: Configuração Inicial e Primeiro Repositório

Nesta parte, você criará um novo repositório local e o conectará ao GitHub.

1.1. Criar um Repositório Local

* **Usando GitHub Desktop:**

1. Abra o GitHub Desktop.
2. Clique em **"File" > "New Repository..."** (Arquivo > Novo Repositório...).
3. Em **"Name"** (Nome), digite `meu-primeiro-git-repo`.
4. Em **"Local path"** (Caminho local), escolha uma pasta em seu computador onde deseja salvar o projeto (ex: sua pasta de Documentos ou Projetos).
5. Clique em **"Create Repository"** (Criar Repositório).

* **Usando Visual Studio Code:**

6. Abra o VS Code.
7. Clique em **"File" > "Open Folder..."** (Arquivo > Abrir Pasta...).
8. Crie uma nova pasta em seu computador (ex: `meu-primeiro-git-repo`) e selecione-a. Clique em **"Select Folder"** (Selecionar Pasta).
9. Com a pasta aberta, clique no ícone de **"Source Control"** (Controle de Código Fonte) na barra lateral esquerda (parece um ícone de três círculos conectados).
10. Clique no botão **"Initialize Repository"** (Inicializar Repositório).

1.2. Criar seu Primeiro Arquivo e Fazer o Primeiro Commit

* **Usando GitHub Desktop:**

1. No GitHub Desktop, clique em **"Show in Explorer"** (Mostrar no Explorador) ou **"Show in Finder"** (Mostrar no Finder) para abrir a pasta do seu repositório.
2. Dentro desta pasta, crie um novo arquivo de texto chamado `README.md`.
3. Abra o `README.md` com um editor de texto simples (como Bloco de Notas no Windows ou TextEdit no Mac) e adicione o seguinte conteúdo:

```
# Meu Primeiro Repositório Git
Este é um repositório de exemplo para aprender Git e GitHub.
```

4. Salve o arquivo e feche o editor.
5. Volte para o GitHub Desktop. Você verá o `README.md` listado na seção **"Changes"** (Alterações).

6. Na parte inferior da tela, em "Summary (required)" (Resumo - obrigatório), digite `Primeiro commit: Adiciona o arquivo README.md`.
7. Clique no botão **"Commit to main"** (Fazer commit para main).

* **Usando Visual Studio Code:**

8. No VS Code, na barra lateral esquerda, clique no ícone de **"Explorer"** (Explorador - parece uma folha de papel).
9. Clique no ícone **"New File"** (Novo Arquivo) e digite `README.md`. Pressione Enter.
10. No editor que se abre, adicione o seguinte conteúdo:

```
# Meu Primeiro Repositório Git
Este é um repositório de exemplo para aprender Git e GitHub.
```

11. Salve o arquivo (**"File" > "Save"** ou `Ctrl+S/Cmd+S`).
12. Clique no ícone de **"Source Control"** (Controle de Código Fonte) na barra lateral esquerda.
13. Você verá o `README.md` listado em "Changes" (Alterações). Clique no ícone **+** ao lado do `README.md` para "Stage Changes" (Preparar Alterações).
14. Na caixa de mensagem de commit, digite `Primeiro commit: Adiciona o arquivo README.md`.
15. Clique no ícone de "check" (marca de seleção) na parte superior da barra lateral de Source Control para **"Commit"** (Fazer Commit).

1.3. Conectar ao GitHub e Publicar o Repositório

* **Usando GitHub Desktop:**

1. No GitHub Desktop, clique no botão **"Publish repository"** (Publicar repositório) no topo da tela.
2. Certifique-se de que "Name" (Nome) seja `meu-primeiro-git-repo`.
3. Marque ou desmarque "Keep this code private" (Manter este código privado) conforme sua preferência.
4. Clique em **"Publish Repository"** (Publicar Repositório).
5. O GitHub Desktop fará o upload do seu repositório para o GitHub.

* **Usando Visual Studio Code:**

6. No VS Code, clique no ícone de **"Source Control"** (Controle de Código Fonte).
7. Clique no botão **"Publish Branch"** (Publicar Branch) na parte inferior da barra lateral.
8. O VS Code pode perguntar se você deseja publicar para o GitHub. Confirme.
9. Se for a primeira vez, você pode precisar fazer login na sua conta GitHub através do VS Code.

10. O VS Code fará o upload do seu repositório para o GitHub.

1.4. Verificar no GitHub

1. Abra seu navegador e vá para github.com.
 2. Faça login e navegue até a página do seu perfil ou para a lista de seus repositórios.
 3. Você deverá ver o repositório `meu-primeiro-git-repo` listado, contendo o arquivo `README.md`.
-

Exercício 2: Trabalhando com Branches

Nesta parte, você criará uma nova branch para desenvolver uma funcionalidade separadamente.

2.1. Criar uma Nova Branch

* **Usando GitHub Desktop:**

1. No GitHub Desktop, clique no botão "**Current Branch**" (Branch Atual) no topo da tela (provavelmente dirá `main`).
2. Clique em "**New Branch...**" (Nova Branch...).
3. Em "Name" (Nome), digite `feature/nova-secao`.
4. Clique em "**Create Branch**" (Criar Branch).
5. O GitHub Desktop mudará automaticamente para esta nova branch.

* **Usando Visual Studio Code:**

6. No VS Code, clique no nome da branch atual na barra de status inferior (provavelmente dirá `main`).
7. No menu que aparece no topo, selecione "**Create new branch...**" (Criar nova branch...).
8. Digite `feature/nova-secao` e pressione Enter.
9. Selecione "`feature/nova-secao`" no próximo prompt para mudar para a nova branch.

2.2. Criar um Novo Arquivo na Nova Branch

* **Usando GitHub Desktop:**

1. Clique em "**Show in Explorer**" (Mostrar no Explorador) ou "**Show in Finder**" (Mostrar no Finder) para abrir a pasta do seu repositório.
2. Dentro desta pasta, crie um novo arquivo de texto chamado `secao_importante.txt`.
3. Abra o `secao_importante.txt` e adicione o seguinte conteúdo:

```
Esta é uma nova seção de conteúdo.  
Ela está sendo desenvolvida em uma branch separada.
```

4. Salve o arquivo e feche o editor.

5. Volte para o GitHub Desktop. Você verá o `secao_importante.txt` listado em "Changes" (Alterações).
6. Na parte inferior, em "Summary (required)", digite `Adiciona nova seção de conteúdo`.
7. Clique em "**Commit to feature/nova-secao**" (Fazer commit para feature/nova-secao).

* **Usando Visual Studio Code:**

8. No VS Code, no "Explorer" (Explorador), clique no ícone "**New File**" (Novo Arquivo).
9. Digite `secao_importante.txt` e pressione Enter.
10. No editor, adicione o seguinte conteúdo:

```
Esta é uma nova seção de conteúdo.  
Ela está sendo desenvolvida em uma branch separada.
```

11. Salve o arquivo.
12. No "Source Control" (Controle de Código Fonte), clique no ícone **+** ao lado do `secao_importante.txt` para "Stage Changes".
13. Na caixa de mensagem de commit, digite `Adiciona nova seção de conteúdo`.
14. Clique no ícone de "check" para "**Commit**".

2.3. Publicar a Nova Branch no GitHub

* **Usando GitHub Desktop:**

1. No GitHub Desktop, clique no botão "**Publish branch**" (Publicar branch) no topo da tela.

* **Usando Visual Studio Code:**

2. No VS Code, clique no ícone de "**Source Control**" (Controle de Código Fonte).
3. Clique no botão "**Publish Branch**" (Publicar Branch) na parte inferior da barra lateral.

2.4. Verificar no GitHub

1. Atualize a página do seu repositório no GitHub.
2. Você deverá ver uma notificação sobre a nova branch `feature/nova-secao`.
3. Mude para esta branch no GitHub (usando o seletor de branch) para ver o arquivo `secao_importante.txt`.

Exercício 3: Mesclando Branches

Agora que a nova funcionalidade está pronta, você irá mesclá-la de volta à branch principal (`main`).

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

* **Usando GitHub Desktop:**

1. No GitHub Desktop, clique no botão **"Current Branch"** (Branch Atual) no topo da tela (dirá `feature/nova-secao`).
2. Selecione `main` na lista de branches.

* **Usando Visual Studio Code:**

3. No VS Code, clique no nome da branch atual na barra de status inferior (dirá `feature/nova-secao`).
4. No menu que aparece no topo, selecione `main`.

3.2. Mesclar a Branch de Feature na `main`

* **Usando GitHub Desktop:**

1. Com a branch `main` selecionada, clique em **"Branch"** no menu superior.
2. Selecione **"Merge into current branch..."** (Mesclar na branch atual...).
3. Na janela que aparece, selecione `feature/nova-secao` na lista suspensa.
4. Clique em **"Create a merge commit"** (Criar um commit de mesclagem).

* **Usando Visual Studio Code:**

5. No VS Code, clique no ícone de **"Source Control"** (Controle de Código Fonte).
6. Clique nos três pontos (...) na barra de título da seção "Source Control".
7. Vá em **"Branch" > "Merge Branch..."** (Branch > Mesclar Branch...).
8. Selecione `feature/nova-secao` na lista de branches para mesclar.

3.3. Enviar as Alterações Mescladas para o GitHub

* **Usando GitHub Desktop:**

1. Após a mesclagem, o GitHub Desktop mostrará um botão **"Push origin"** (Enviar para origin) no topo. Clique nele.

* **Usando Visual Studio Code:**

2. No VS Code, clique no ícone de **"Source Control"** (Controle de Código Fonte).
3. Clique no botão **"Sync Changes"** (Sincronizar Alterações) na parte inferior da barra lateral (ou o ícone de nuvem com setas). Confirme o "Push & Pull".

3.4. (Opcional) Excluir a Branch de Feature

* **Usando GitHub Desktop:**

1. No GitHub Desktop, clique no botão **"Current Branch"** (Branch Atual) e certifique-se de que você está na branch `main`.
2. Clique novamente no botão **"Current Branch"** e localize `feature/nova-secao` na lista.

3. Clique com o botão direito do mouse sobre `feature/nova-secao` e selecione **"Delete..."** (Excluir...).
4. Marque a opção "Also delete on GitHub" (Excluir também no GitHub) e clique em **"Delete"** (Excluir).

* **Usando Visual Studio Code:**

5. No VS Code, clique no nome da branch atual na barra de status inferior e certifique-se de que você está na branch `main`.
6. Clique novamente no nome da branch na barra de status inferior.
7. Localize `feature/nova-secao` na lista.
8. Clique no ícone de lixeira ao lado de `feature/nova-secao` para excluí-la localmente.
9. Para excluir remotamente, você pode precisar ir para o GitHub no navegador, ir em "Branches" e excluir a branch de lá.

Exercício 4: Simulação e Resolução de Conflitos de Mesclagem

Conflitos ocorrem quando o Git não consegue mesclar automaticamente as alterações. Vamos simular um e resolvê-lo visualmente.

4.1. Preparar para o Conflito

* **Usando GitHub Desktop:**

1. Crie uma nova branch: Clique em **"Current Branch" > "New Branch..."**. Nomeie-a `bugfix/correcao-readme`. Clique em **"Create Branch"**.
2. Abra o `README.md` (via "Show in Explorer"). Adicione a seguinte linha no final:

```
Esta linha foi adicionada na branch de correção de bug.
```
3. Salve e feche.
4. No GitHub Desktop, faça o commit: "Summary": `Adiciona linha de correção de bug no README`. Clique em **"Commit to bugfix/correcao-readme"**.
5. **Não publique esta branch ainda.**
6. Mude para a branch `main`: Clique em **"Current Branch"** e selecione `main`.
7. Abra o `README.md` novamente. Adicione a seguinte linha no final (no *mesmo local* da alteração anterior):

```
Esta linha foi adicionada diretamente na branch principal.
```
8. Salve e feche.
9. No GitHub Desktop, faça o commit: "Summary": `Adiciona linha na main que causará conflito`. Clique em **"Commit to main"**.

* **Usando Visual Studio Code:**

10. Crie uma nova branch: Clique no nome da branch na barra de status inferior > **"Create new branch..."**. Nomeie-a `bugfix/correcao-readme`. Pressione Enter e selecione-a.
11. Abra o `README.md` no editor. Adicione a seguinte linha no final:

```
Esta linha foi adicionada na branch de correção de bug.
```
12. Salve o arquivo.
13. No "Source Control", prepare e faça o commit: "Summary": `Adiciona linha de correção de bug no README`. Clique no ícone de "check".
14. **Não publique esta branch ainda.**
15. Mude para a branch `main`: Clique no nome da branch na barra de status inferior e selecione `main`.
16. Abra o `README.md` novamente. Adicione a seguinte linha no final (no *mesmo local* da alteração anterior):

```
Esta linha foi adicionada diretamente na branch principal.
```
17. Salve o arquivo.
18. No "Source Control", prepare e faça o commit: "Summary": `Adiciona linha na main que causará conflito`. Clique no ícone de "check".

4.2. Simular e Resolver o Conflito

* Usando GitHub Desktop:

1. Certifique-se de que você está na branch `main`.
2. Clique em **"Branch"** no menu superior > **"Merge into current branch..."**.
3. Selecione `bugfix/correcao-readme` na lista suspensa.
4. Clique em **"Create a merge commit"**.
5. O GitHub Desktop detectará o conflito e mostrará uma mensagem. Clique em **"View Conflicts"** (Ver Conflitos) ou **"Open in Visual Studio Code"** (Abrir no Visual Studio Code) se você tiver o VS Code instalado e configurado como editor padrão.
6. Se você abrir o arquivo em um editor de texto simples, verá marcadores de conflito:

```
<<<<<< HEAD
Esta linha foi adicionada diretamente na branch principal.
=====
Esta linha foi adicionada na branch de correção de bug.
>>>>>> bugfix/correcao-readme
```

7. **Para resolver:** Edite o arquivo `README.md` para que ele contenha a versão final desejada, removendo os marcadores de conflito. Por exemplo, para manter ambas as linhas:

```
# 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.
```

8. Salve o arquivo e feche o editor.
9. Volte para o GitHub Desktop. Ele detectará que o conflito foi resolvido.
10. Em "Summary", digite **Resolve conflito de mesclagem no README.md**.
11. Clique em "**Commit merge**" (Fazer commit da mesclagem).
12. Clique em "**Push origin**" para enviar as alterações para o GitHub.

* **Usando Visual Studio Code:**

13. Certifique-se de que você está na branch **main**.
14. No "Source Control", clique nos três pontos (...) > "**Branch**" > "**Merge Branch...**".
15. Selecione **bugfix/correcao-readme**.
16. O VS Code detectará o conflito e abrirá o **README.md** com uma interface de resolução de conflitos. Você verá opções como "Accept Current Change" (Aceitar Alteração Atual), "Accept Incoming Change" (Aceitar Alteração Recebida), "Accept Both Changes" (Aceitar Ambas as Alterações).
17. **Para resolver:** Clique em "**Accept Both Changes**" (Aceitar Ambas as Alterações) ou edite manualmente o código para a versão desejada.
18. Salve o arquivo (**Ctrl+S/Cmd+S**).
19. No "Source Control", o **README.md** aparecerá em "Merged Changes" (Alterações Mescladas).
20. Na caixa de mensagem de commit, digite **Resolve conflito de mesclagem no README.md**.
21. Clique no ícone de "check" para "**Commit**".
22. Clique no botão "**Sync Changes**" (Sincronizar Alterações) ou no ícone de nuvem com setas para enviar as alterações para o GitHub.

4.3. (Opcional) Excluir a Branch de Conflito

Após a mesclagem e resolução, você pode excluir a branch **bugfix/correcao-readme** usando os mesmos passos da Parte 3.4.

Exercício 5: Verificando o Histórico e Status Final

* **Usando GitHub Desktop:**

1. Clique na aba "**History**" (Histórico) para ver todos os seus commits, incluindo os de mesclagem.
2. A aba "**Changes**" (Alterações) deve estar vazia, indicando que não há alterações pendentes.

* **Usando Visual Studio Code:**

3. No "Source Control", clique nos três pontos (...) > **"View & Sort"** > **"Show Git Graph"** (se você tiver a extensão Git Graph instalada, altamente recomendado). Isso mostrará um histórico visual de seus commits e branches.
 4. A seção "Changes" (Alterações) no "Source Control" deve estar vazia.
 5. Na barra de status inferior, o ícone de nuvem com setas deve indicar que não há alterações para sincronizar.
-