

Passo a passo:

1. Clonar o repositório

1. Para abrir o Git Bash:

- Clique com o botão direito em uma área vazia da área de trabalho ou dentro de uma pasta e selecione **Git Bash Here**.

2. Comando para clonar o repositório

```
git clone https://github.com/JosianeCaroliny/PI_FATECOSASCO2024-2.git
```

Isso vai copiar o repositório do GitHub para o computador.

2. Atualizar o repositório antes de fazer alterações (comando pull)

SEMPRE atualizar o projeto antes de qualquer modificação, para evitar conflitos na hora do merge.

1. No Git Bash, ainda dentro da pasta do repositório, executar:

```
git pull origin main
```

Comando para atualizar o projeto de acordo com as mudanças já comitadas por outras pessoas e disponíveis no repositório.

2. Verifique o status do repositório para ver o que foi atualizado:

```
git status
```

Esse comando mostrará se há arquivos modificados ou conflitos que precisam ser resolvidos.

Erro mais comum: Se não fizer o pull antes de começar, pode haver o erro de **"conflito de merge"** ao tentar subir as novas atualizações.

3. Fazer alterações no projeto

Agora que o repositório está atualizado, abra o projeto em alguma IDE para fazer as alterações.

4. Adicionar as alterações (comando add)

Após terminar as modificações, é necessário adicionar os arquivos alterados para que o Git saiba quais mudanças serão enviadas.

1. **No terminal (ou no Git Bash, dentro da pasta do projeto), executar:**

```
git add .
```

Esse comando adiciona **todos os arquivos modificados**. Se você preferir adicionar arquivos específicos, pode usar o comando:

```
git add nome-do-arquivo.ext
```

5. Criar um commit com as alterações (comando commit)

Crie um commit descrevendo o que foi alterado.

1. **Executar o comando adicionando uma mensagem de descrição do que foi feito entre os parênteses:**

```
git commit -m "Descrição das alterações feitas"
```

Exemplo:

```
git commit -m "Alterações na tela x do produtor"
```

6. Atualizar o repositório local novamente antes de enviar as mudanças

Antes de enviar suas alterações para o GitHub, garanta que o repositório local ainda esteja atualizado com as últimas modificações feitas por outras pessoas. Isso evita o erro de **"conflito de merge"**.

1. **Execute o comando de pull novamente:**

```
git pull origin main
```

2. **Verifique o status do repositório para garantir que está tudo em ordem:**

```
git status
```

Se o Git encontrar alterações feitas por outras pessoas que conflitam com as suas, resolva os conflitos manualmente. Se o repositório já estiver atualizado, pode prosseguir.

7. Enviar as alterações para o GitHub (comando push)

1. Execute o comando:

git push origin main

Esse comando envia as alterações para o repositório no GitHub.

8. Verificar o status do repositório (comando status)

Se quiser ver se há arquivos pendentes de commit ou push, ou se seu repositório está em dia, utilizando novamente o comando:

git status