

O objetivo é interegir com um repositório alheio de forma que outras pessoas do repositório original vejam as tuas alterações. O fluxo será:

fork → clone → branch → push → pull request

## 1 Fluxo recomendado (GitHub) — fork + PR

Esse é o jeito padrão quando você não tem permissão de escrita no repo original.

Passos rápidos (web + terminal):

No GitHub, vá ao repositório original (<https://github.com/natmourajr/CPE727-2025-03>) e clique em Fork (canto superior direito).

The screenshot shows the GitHub interface for the repository 'CPE727-2025-03'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository name 'CPE727-2025-03' is displayed in the center. Below the repository name, there are sections for 'main' (containing 14 branches and 0 tags), a commit history (with 107 commits), and file lists for README and LICENSE. On the right side, there's an 'About' section with details about the repository owner, license (MIT), and statistics (4 stars, 1 watching, 2 forks). A red box highlights the 'Fork' button in the top right corner, and a red arrow points to it from below.

-> Isso cria uma cópia do repo na sua conta ([github.com/SEU\\_USUARIO/CPE727-2025-03](https://github.com/SEU_USUARIO/CPE727-2025-03)).

The screenshot shows the 'Create a new fork' form on GitHub. It starts with a heading 'Create a new fork' and a brief explanation: 'A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.' Below this, it says 'Required fields are marked with an asterisk (\*).'. The 'Owner' dropdown is set to 'jaoovpessoa' and the 'Repository name' field is 'CPE727-2025-03'. A note indicates that 'CPE727-2025-03 is available.' Below these fields, there's a 'Description' input box containing the text 'Repositório para ser utilizado durante a disciplina do PEE (Programa de Engenharia Elétrica) da Coppe/UFRJ CI'. There's also a checked checkbox for 'Copy the main branch only' with the sub-instruction 'Contribute back to natmourajr/CPE727-2025-03 by adding your own branch.' At the bottom, a note says 'You are creating a fork in your personal account.' and a green 'Create fork' button is visible.

Clone o fork para sua máquina:

```
git clone https://github.com/SEU_USUARIO/CPE727-2025-03.git  
cd CPE727-2025-03
```

Configure o remoto upstream apontando para o repositório original, pois será útil para sincronizar depois:

```
git remote add upstream https://github.com/natmourajr/CPE727-2025-03.git  
git fetch upstream
```

Crie uma branch nova para sua alteração:

```
git checkout -b feature/sua_feature  
git add .  
git commit -m "Descrição curta do que foi alterado"
```

Envie sua branch para o fork no GitHub:

```
git push origin feature/sua_feature
```

No GitHub, entre no seu fork e verá um botão sugerindo abrir um **Pull Request** do feature/sua\_feature do seu fork para o repositório original. Escreva um título claro e descrição, pois facilita a revisão, em seguida escolha a branch alvo do repo original (normalmente main ou develop). Os mantenedores do repositório original receberão a PR, revisarão e poderão aceitar (merge) — aí suas alterações aparecerão no repo original.

## Clonou o repositório original ao invés de ter feito fork?

```
# dentro da pasta clonada  
git remote rename origin upstream          # opcional, renomeia o origin para upstream  
git remote add origin https://github.com/SEU_USUARIO/CPE727-2025-03.git  
# agora origin = seu fork, upstream = repo original  
git checkout -b feature/feature_name  
git add .  
git commit -m "minha mudança"  
git push origin feature/feature_name  
# abrir PR no GitHub do seu fork -> upstream
```

## Pedir permissão para ser colaborador

Se você precisa push direto no repositório original (sem PR), peça ao dono que te adicione como collaborator (Settings → Manage access) ou te dê permissão em uma branch. Só então você poderá:

```
git push upstream feature/feature_name
```

## Importante lembrar!

Antes de trabalhar, mantenha seu fork sincronizado com o original. Quando sua branch divergir, rebase/merge da main atualizada evita conflitos.

```
git fetch upstream
git checkout main
git merge upstream/main    # ou: git rebase upstream/main
git push origin main
```