

- SERVE PARA PUXAR O PROJETO PARA O NOSSO PC:

git clone https://github.com/Renny3060/4-em-linha.git

SEMPRE QUE SE FIZER COMMITS E SE QUISER PUXAR PARA O PC

- COMO PUXAR AS ALTERAÇÕES SEM AFETAR O NPM RUN:

1º clique em Terminal > New Terminal

2º Colocar no terminal > git pull origin main

3º Se algum de nós adicionar pacotes novos no projeto > npm install

COMMIT

- APAGAR UM COMMIT COM UM Nº ESPECÍFICO:

1º git rebase --abort

2º git log --oneline

3º git rebase -i (meter o número do commit, pode ser visto no github online)^

```
f857a8e (HEAD -> main, origin/main, origin/HEAD) mostrar commit
e50240f (novo-main) início da aplicação
c9df2b4 first commit
PS C:\Users\admin\Desktop\4-em-linha-1> git rebase -i f857a8e^
Successfully rebased and updated refs/heads/main.
PS C:\Users\admin\Desktop\4-em-linha-1> |
```

4º trocar o pick por drop

```
pick f857a8e mostrar commit
# Rebase e50240f..f857a8e onto e50240f (1 command)
```

```
drop f857a8e mostrar commit

# Rebase e50240f..f857a8e onto e50240f (1 command)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the
#     commit's log message, unless -C is used,
#     keep only this commit's message; -C is
#     opens the editor
# x, exec <command> = run command (the rest of the line) using
# b, break = stop here (continue rebase later with 'git rebase
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# n, merge [-C <commit> | -c <commit>] <label> [ <oneline> ]
#     create a merge commit using the original merge commit
#     message (or the oneline, if no original merge commit
#     specified); use -c <commit> to reword the commit me
# u, update-ref <ref> = track a placeholder for the <ref> to
#     to this position in the new commits.
#     updated at the end of the rebase
#
# These lines can be re-ordered; they are executed from top to
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be abort
#
~
~
~
git rebase -merge/git-rebase-todo+1 [unix] (21:19 04/04/2025)
-- INSERT --
```

5º git push origin main --force

```
graph TD
    A[início da aplicação Nune...] --> B[first commit Renny3060]
    B --> C[main]
```

- REVERT DE UM COMMIT COM UM Nº ESPECÍFICO:

1º copiar o id do commit

2º abrir o terminal

3º git revert (numero do commit copiado)