

TUTORIAL – FLUXO DE TRABALHO

Este tutorial documenta as ações Git utilizadas durante o desenvolvimento do projeto e o padrão utilizado para atualização de código. Para organização das modificações cada desenvolvedor deverá atuar uma Branch conforme o padrão.

– Padrão utilizado

Atualizar main branch

`git checkout main`

`git pull origin main`

Criar e acessar nova Branch **(1)**. Se Branch já criada, apenas acessar **(2)**.

1. `git checkout -b tasks-nomedabbranch`

2. `git checkout tasks-nomedabbranch`

Commit na branch criada pelo desenvolvedor

`git add .`

`git commit -m "Commit message"`

`git push origin tasks-back-end`

Para contribuidores, realizar pull request (1) e para mantenedores, revisar e realizar merge (2).

1. Clicar no botão “Compare & Pull Request”

Selecionar Branch “main” como base e branch “tasks-nomedabbranch” como Branch de comparação

Preencher informações sobre a atualização ou modificação

Enviar Pull Request

2. Merge com a main Branch (Após revisão com equipe)

git checkout main

git pull origin main

git merge tasks-nomedabbranch

git push origin main

– Lista de comandos utilizados no projeto:

Modificações no back-end

git checkout main

git fetch origin

git pull origin main

git checkout tasks-back-end

git add .

git commit -m "Commit message"

git push origin tasks-back-end

git checkout main

git pull origin main

git merge tasks-back-end

git push origin main

Modificações no front-end

git checkout main

```
git fetch origin
git pull origin main
git checkout tasks-front-end
git add .
git commit -m "Commit message"
git push origin tasks-front-end
git checkout main
git pull origin main
git merge tasks-front-end
git push origin main
```

Atualização com docker

```
git checkout main
git pull origin main
git checkout tasks-back-end-docker
git add .
git commit -m "Commit message"
git push origin tasks-back-end-docker
git checkout main
git pull origin main
git merge tasks-back-end-docker
git push origin main
```