Rasmoo – GitHub

Criando um repositório

1. Para trazer o repositório para a máquina:
2. Clicar em **Code**.
3. Copiar o link **HTTPS**.
4. Criar uma pasta no computador, chamarei de **repositório**.
5. Clonar o projeto com o comando **git clone**, fica assim: **git clone https://github.com/alexandremcp/teste.git**
6. Após isso será criada dentro da pasta **repositório** uma outra pasta onde ficará o projeto, essa pasta receberá o mesmo nome do projeto que está no GitHub.

Enviando arquivos

1. Entre na pasta do projeto que chamei de **teste**.
2. Crie um arquivo chamado **identidade.txt**
3. Após criar um arquivo, verifique o status do projeto com o comando **git status**.
4. Os arquivos que não forem salvos no GitHub irão aparecer numa listagem na cor vermelha.
5. Adicione na fila para ser salvo com:
   1. **git add .** (adiciona todos os arquivos na fila)
   2. **git add identidade.txt** (adiciona apenas este arquivo na fila)
6. Após isso os arquivos que foram incluídos na fila aparecerão numa listagem na cor verde, basta dar **git status** para ver.
7. Para enviar basta dar o comando: **Git Commit -m “Arquivo identidade criado”**
8. **Observação**: Se não estiver logado no GitHub irá aparece uma mensagem de erro informando que preciso me logar e para isso tem os comandos:
   1. Git config --global user.email [alexandremcp@gmail.com](mailto:alexandremcp@gmail.com)
   2. Git config – global user.name “alexandremcp”
9. Para gravar os arquivos enviados dar o comando **git push**.

Exibindo commits realizados

**Git reflog**

f082983 (HEAD -> main, origin/main, origin/HEAD) HEAD@{0}: commit: Alterado o nome da identidade e adicionado o CPF

eb3feab HEAD@{1}: commit: Craido arquivo identidade

461f9f9 HEAD@{2}: clone: from https://github.com/alexandremcp/teste.git

Voltando resultados anteriores

Para voltar para a versão original use o comando **git reset --hard** e o identificador exibido pelo **git reflog** ficando assim: **git reset --hard eb3feab** e isso voltara ao commit identificado como “Craido arquivo identidade”

Voltando para a versão com a alteração: **git reset --hard f082983**

Branch

Uma branch é uma ramificação do projeto que pode receber alterações para fins de teste como uma nova versão.

**git branch desenvolvimento** (cria uma ramificação chamada desenvolvimento)

Trocando de branch

**Git checkout desenvolvimento** (vai para a branch desenvolvimento)

**Git branch** (mostra a branch atual)

Estando na brach **desenvolvimento**, todas as alterações realizadas serão gravadas no arquivo porem apenas serão vistas nessa branch.

Alexandre@ALEXANDRE MINGW64 ~/Desktop/repositorio/teste (desenvolvimento)

**git add identidade.txt**

Alexandre@ALEXANDRE MINGW64 ~/Desktop/repositorio/teste (desenvolvimento)

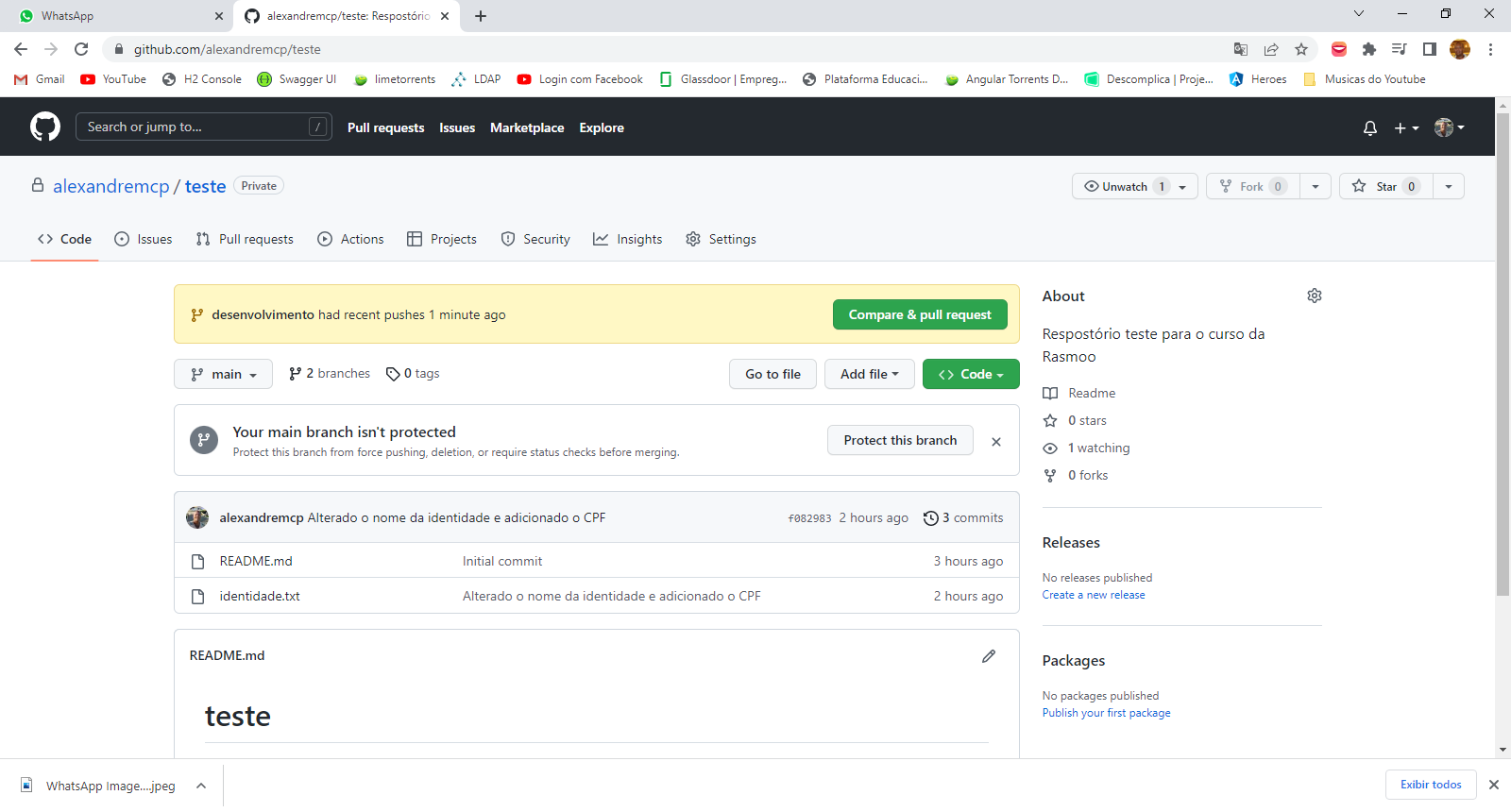
**git commit -m "Adicionado novos campos na identidade"**

[desenvolvimento e02394e] Adicionado novos campos na identidade

1 file changed, 3 insertions(+), 1 deletion(-)

Antes de dar um **git push** a branch **desenvolvimento** precisa ser criada conforme o comando **git push --set-upstream origin desenvolvimento**







Mesclar duas branchs

**Git checkout main** (volta para a branch principal que receberá as alterações)

Alexandre@ALEXANDRE MINGW64 ~/Desktop/repositorio/teste (main)

**git branch** (certificando que estou na brach correta)

desenvolvimento

\* main

Para trazer as informações novas da branch desenvolvimento dê o seguinte comando:

**Git merge desenvolvimento** (Mescla as informações vindas de desenvolvimento na main)

Updating f082983..e02394e

Fast-forward

identidade.txt | 4 +++-

1 file changed, 3 insertions(+), 1 deletion(-)

**Git push** (atualiza a branch main, não precisa de commit)

Puxando informações feitas por outro usuário

**Git pull**

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), 761 bytes | 84.00 KiB/s, done.

From https://github.com/alexandremcp/teste

e02394e..9ca78ca main -> origin/main

Updating e02394e..9ca78ca

Fast-forward

identidade.txt | 4 +++-

**1 file changed, 3 insertions(+), 1 deletion(-)**

Atualizando a brach desenvolvimento

**git checkout desenvolvimento** (vai para a branch desenvolvimento)

**git merge main** (mescla o conteúdo da main em desenvolvimento)

Updating e02394e..9ca78ca

Fast-forward

identidade.txt | 4 +++-

**1 file changed, 3 insertions(+), 1 deletion(-)**

**Git push** (atualiza a branch desenvolvimento, não precisa de commit)

Ignorando arquivos para envio

**git status** (mostra que existe uma pasta chamada imagens não enviada ainda)

On branch main

Your branch is up to date with 'origin/main'.

Untracked files:

(use "git add <file>..." to include in what will be committed)

imagens/

Crie um arquivo chamado **.gitignore** e dentro dele informe o nome do que deseja ignorar, ou seja, não levar para o github como: **/imagens**

**git status** (Agora só exibi que existe um arquivo chamado .gitignore e não mais a pasta)

On branch main

Your branch is up to date with 'origin/main'.

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

**git add .gitignore** (adiciona o gitignore ao projeto)

**git commit -m "Criado o arquivo gitignore"**

[main 6472d26] Criado o arquivo gitignore

1 file changed, 1 insertion(+)

create mode 100644 .gitignore

**git push** (grava o arquivo gitignore no github)