**Git e GitHub**

**GitHub é uma plataforma de gerenciamento de projetos e versionamento de código**, criada para desenvolvedores (mas não precisa ser usado apenas por desenvolvedores). Na plataforma, você pode **hospedar** as versões de seus trabalhos, códigos e/ou projetos e compartilha-los, se quiser, com o resto do globo, além é claro de poder contribuir com projetos de terceiros.

O Git (Desenvolvido em Linux) é o sistema de controle de versão que nos ajuda e acompanhar as mudanças feitas no código, e também identificar quem efetuou mudanças nele. Além disso, através do Git podemos restaurar código removido ou modificado, e utiliza-lo em linha de comando (Git bash / CMD Git) para fazer tudo isso e mais, **criar repositórios**, fazer ***fork*** e **clonar** projetos, e enviar arquivos (**commits** e **pull-request**) para um repositório.

Digamos então que o **Git** é a parte operacional, responsável pelo famoso CRUD, que cria, altera, atualiza, remove e o **Hub** é a alma, a plataforma que possibilita essas ações.

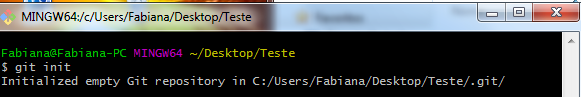
Para utilizar o Git, é necessário baixa-lo (<https://git-scm.com/downloads>). Após a instalação, ele virá com o Git em CMD /Bash que é forma de entrar com os comandos em linha de comando (espécie de terminal). Ele também possui a opção Git GUI que é nada mais que uma interface que permite que você crie, clone ou abra repositórios.

**Alguns conceitos importantes:**

**Repositório** - ou repo, é o diretório onde serão **hospedadas** as versões de seus trabalhos, códigos e/ou projetos. Para criar um repositório via linha de comando, você pode:

- Criar uma pasta local e abrir o git CMD/Bash dentro dela;

- Utilizar o comando **git init** dentro desta pasta para que a mesma seja tratada como repositório:

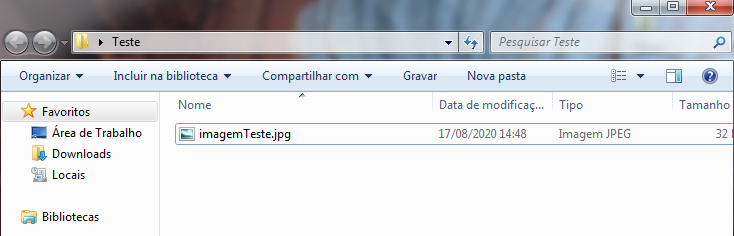


Pronto, seu repositório local foi iniciado.

**Para enviar arquivos de um repositório local para um repositório do GitHub, você pode:**

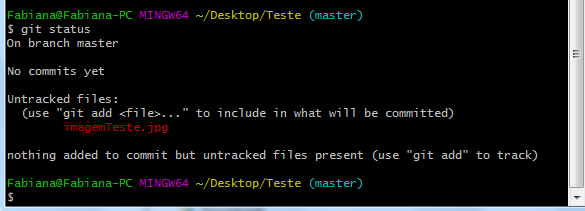
- Adicionar os arquivos que quer subir ao repositório do GitHub na pasta local que está tratando como repositório (pode ser pastas, arquivos e/ou imagens). Abaixo, demostrarei um exemplo:

Na pasta que estou tratando como repositório, adicionei uma imagem chamada ‘imagemTeste.jpg’:

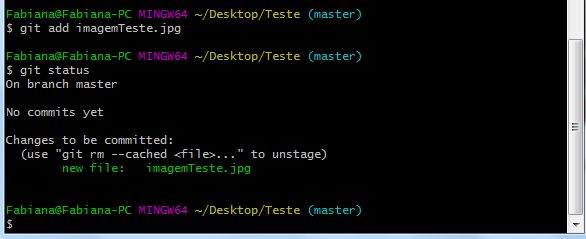


Após adicionar o(s) arquivo(s), utilize os comandos abaixo para adicionar, persistir e enviar este(s) arquivo(s) ao GitHub:

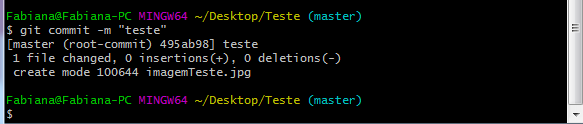
- Utilize o comando **git status** para verificar as alterações – em vermelho:

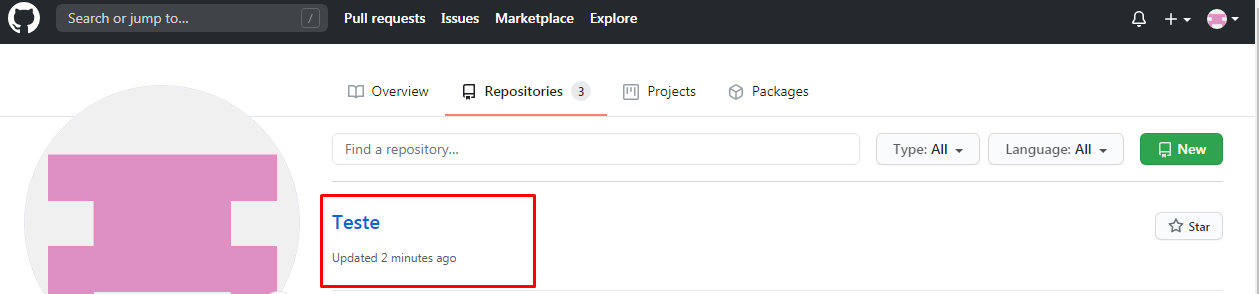


- Utilize o comando **git add .** ou **git add nomeArquivo** para adicionar as alterações localmente:

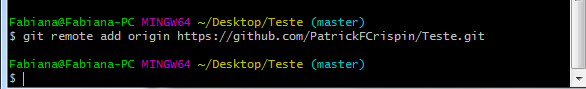


- Utilize o comando **git commit -m “mensagem”** para persistir as alterações:



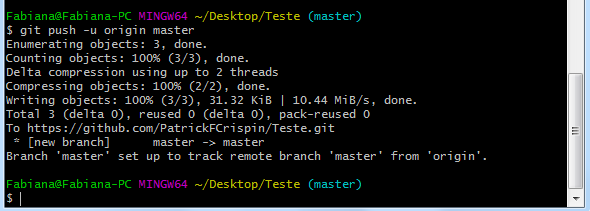
- No Github, crie um repositório com o mesmo nome do repositório local criado no computador (será utilizado para receber o(s) arquivo(s) do repositório local): 

- Volte ao git e utilize o comando **git remote add origin link.git**, onde o ‘**link** é o link para o repositório que acabou de criar no Github:

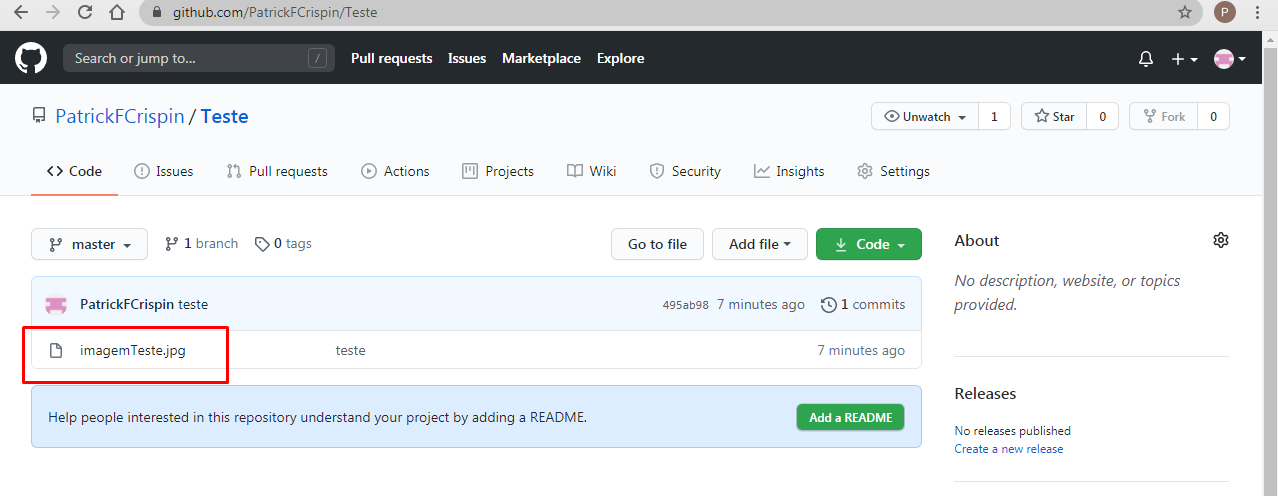


Pronto, você acabou de enviar o repositório local ao repositório do GitHub.

- Por fim, faça o upload do conteúdo do repositório local ao repositório do GitHub - utilize o comando **git push** ou **git push -u origin nome\_da\_branch**:

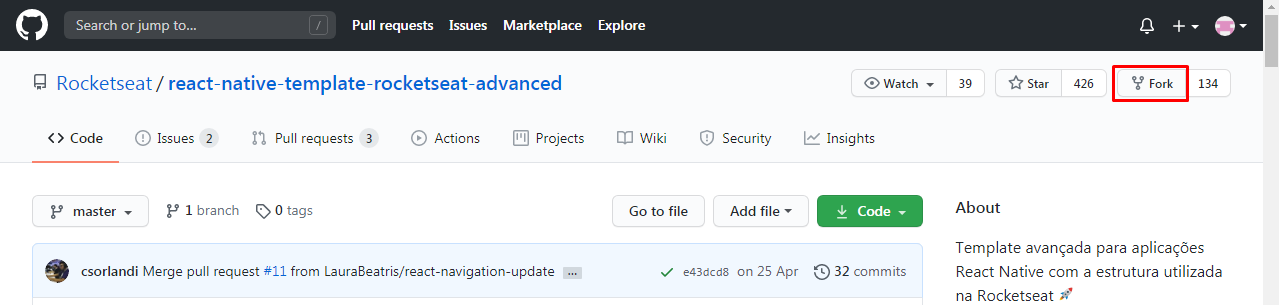


No Github, dentro do repositório criado, você terá os arquivos adicionados inicialmente em sua pasta local que foi tratada como repositório:

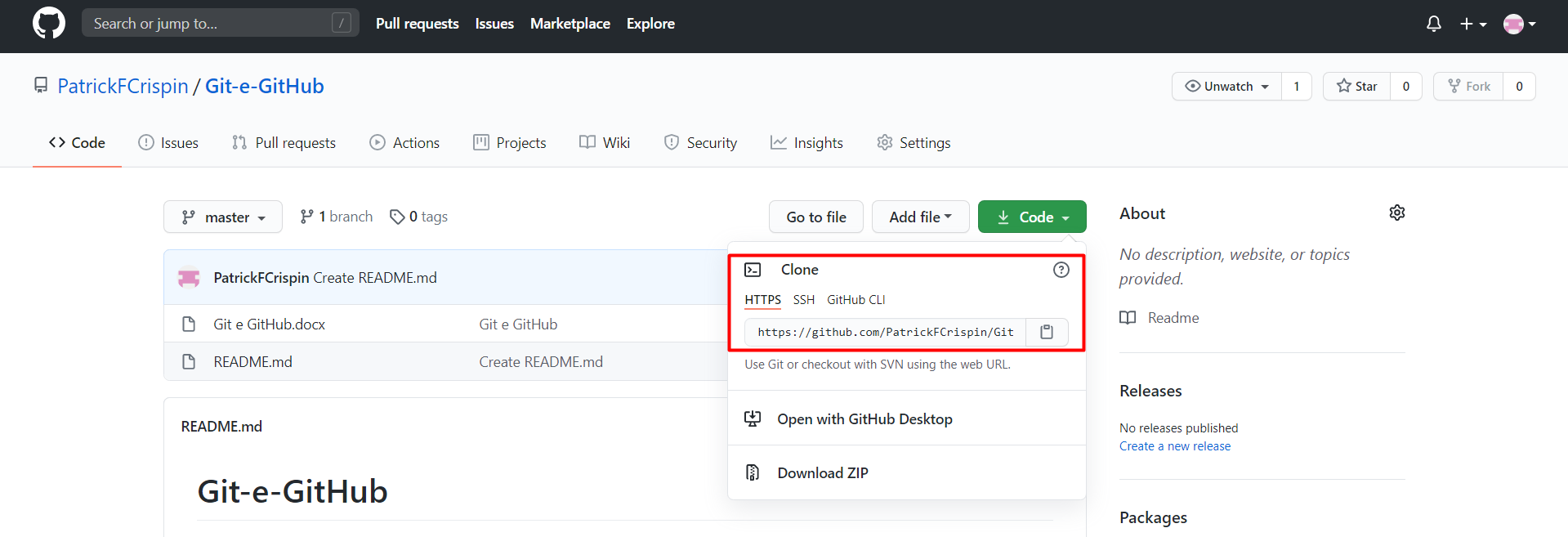


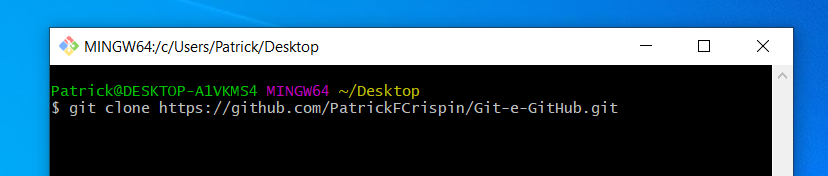
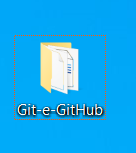
**Branch** - é uma cópia do diretório.

**Fork** - utilizado para criar uma cópia de um repositório em seus repositórios. É muito utilizado quando se deseja fazer um pull-request em algum repositório – ou seja, informar outros desenvolvedores / integrantes de um projeto que você irá implementar mudanças criadas no seu branch ao reposítório do projeto. Para fazer um Fork, basta clicar no botão ‘fork’ de algum repositório do GitHub:



**Clone** – utilizado para clonar um repositório. Junto com o git CMD/Bash, é comumente utilizado para clonar um repositório e trabalhar de forma local. Em uma pasta qualquer de seu computador, ao executar o git, você pode utilizar o comando **git clone link** onde o ‘**link’** é o código HTTPS fornecido no print abaixo:

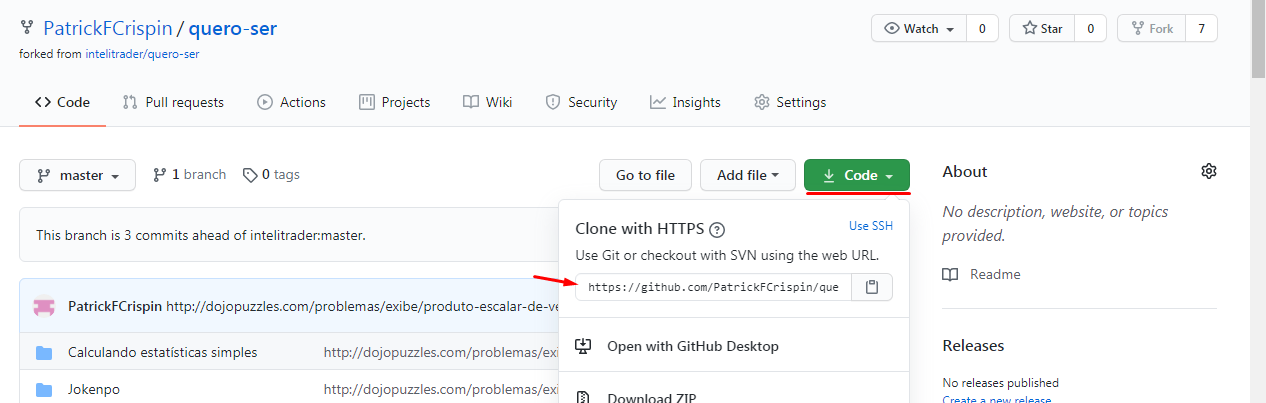


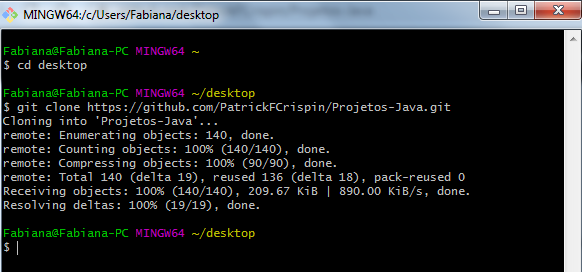
Abaixo exemplo de como clonar um repositório do GitHub: 

**Pull-request -** utilizado basicamente para informar outros desenvolvedores de um projeto que você irá implementar mudanças criadas no seu branch ao reposítório do projeto. Os colaboradores do repositório podem aceitar ou negar a Pull Request. Abaixo exemplo de como fazer um Pull-request:

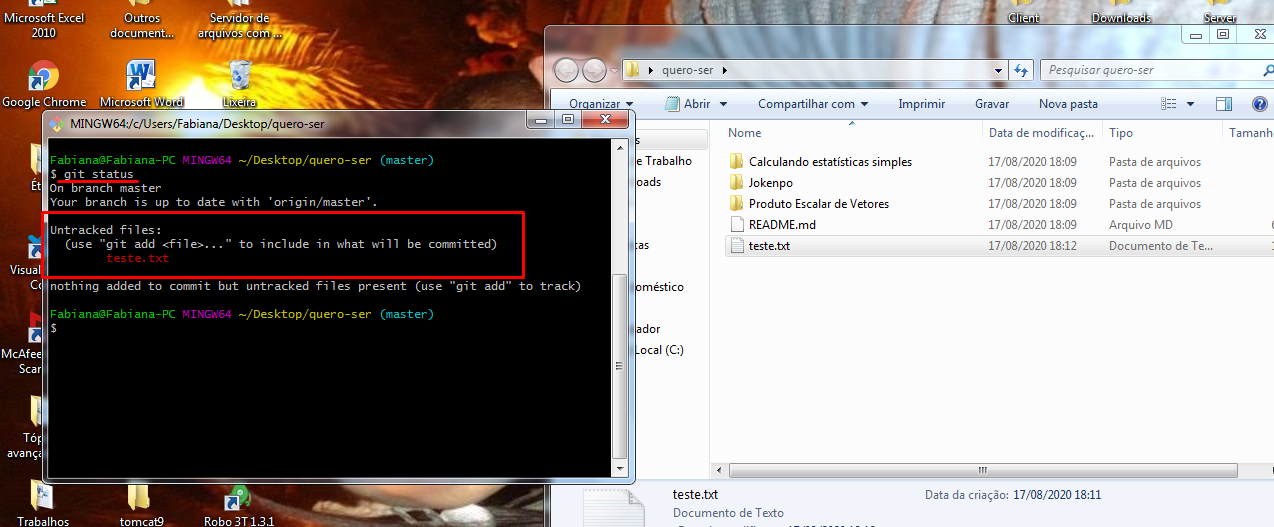
- Faça um **fork** de um repositório;

- Abra o Git CMD/Bash no seu computador (em qualquer lugar) e faça um **clone** do repositório - utilize o comando **git clone repositório**:





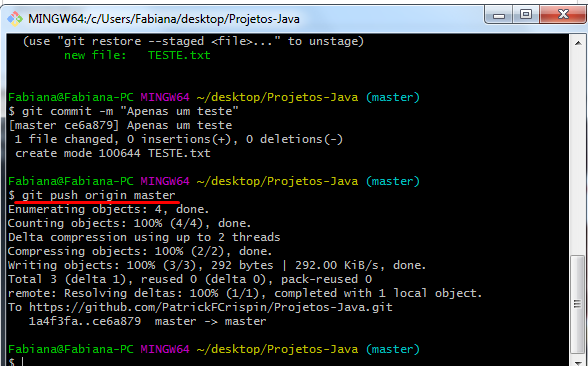
Crie, adicione e/ou remova os arquivos da pasta. Criei um ‘teste.txt’ para exemplificar.

- Utilize o comando **git status** para verificar os itens modificados/alterados:

- Utilize o comando **git add** . para adicionar os itens modificados.

- Utilize o comando **git commit -m “mensagem que quiser”** para persistir a ação de adição anterior no repositório.

-Por fim, faça o upload das modificações ao repositório - utilize o comando **git push** ou **git push origin master**:



Obs.: neste momento pode ocorrer algum erro por causa da configuração de usuário. Caso ocorra. basta adicionar **user.email** e **user.name** de forma global e estes dados serão utilizados sempre pelo git. Exemplo:

git config user.email "patrickfonseca208@gmail.com"

git config user.name "PatrickFCrispin" - adicionei conforme o nome do GitHub.

--

Pronto! Para fazer o **Pull-request** basta clicar no botão ‘Pull-request’ > ‘New pull-request’ no GitHub e então criar e enviar (send) o pull request.

--

**Comando importantes (via git CMD/Bash):**

**git init** – para iniciar um repositório.

**git clone link** -para clonar um repositório, onde ‘**link**’ é o link para um repositório do GitHub.

**git status** - para verificar as alterações.

**git add .** / **git add nome\_do\_arquivo** - para adicionar as alterações localmente. ‘git add .’ adiciona todos os arquivos. ‘git add nome\_do\_arquivo’ adiciona um arquivo específico.

**git commit -m “mensagem que quiser”** -para persistir as alterações no repositório.

**git push** / **git push origin nome\_da\_branch** - para fazer o upload das alterações na branch do GitHub.

**git fetch** – para trazer do repositório remoto todas informações sobre ele que ainda não estão no seu repositório local.

**git checkout nome\_da\_branch** – para mudar de branch.

**git checkout -b nome\_da\_branch** – para criar uma nova branch.