

Passo a Passo "Rápido"

1 - Crie uma pasta no seu computador, entre dentro dela e em seguida aperte o botão direito do mouse em um espaço vazio dentro da pasta para abrir o menu de opções, em seguida clique na opção abaixo para iniciar o terminal Git nesta pasta:

```
" Git Bash Here "
```

2 - No terminal digite o comando abaixo, para poder inicializar o Git nesta pasta e criar uma pasta onde conterà os arquivos que o Git vai usar para fazer o versionamento, esta pasta está nomeada como "git" e ela estará oculta, pois você não vai alterar os dados dela:

```
==> git init
```

3 - Para adicionar mais de um arquivo você precisa rodar o comando abaixo, assim você pode adicionar 2 arquivos ou mais de uma vez só:

```
==> git add .
```

4 - Até o momento fizemos apenas adicionar os arquivos no git, mas ainda não criamos o primeiro conjunto de dados de alterações do projeto, que no caso seria o commit, para isso precisamos fazer o commit dessa alteração, segue comando abaixo, tendo em vista que o texto entre aspas é o título descrevendo o que foi feito nessa alteração:

```
==> git commit -m "primeiro commit"
```

5 - Para enviar esse commit(conjunto de alterações do projeto) para o seu projeto na nuvem, rode o comando abaixo:

```
==> git push
```

[OPCIONAL] Caso tenha aparecido o erro "[No configured push destination.](#)", quer dizer que você ainda não criou informou para o Git qual o repositório desse projeto. Vamos aos passos:

1 - Esse comando serve para definir qual o repositório que você vai subir os commits:

```
==> git remote add origin https://github.com/RamonMartins/Testes
```

2 - Agora dê novamente o comando de push:

```
==> git push
```

3 - Caso apareça o erro "[The current branch master has no upstream branch.](#)", quer dizer que você não informou qual branch(divisão) do projeto essas alterações estão sendo subidas, para isso rode o seguinte comando:

`==> git push --set-upstream origin master`

4 - Caso tenha aparecido o erro "[error: failed to push some refs to](#)", significa que as alterações do projeto que está no Github e do que está localmente não estão iguais, para resolver isso rode o comando:

`==> git pull origin master`

5 - Caso apareça o erro "[fatal: refusing to merge unrelated histories](#)", quer dizer que o projeto no Github e o projeto local são totalmente divergentes e não tem relações entre si, para resolver rode o comando abaixo:

`==> git pull origin master --allow-unrelated-histories`

Em seguida vai abrir um editor de texto no terminal, tecla "Esc" e em seguida digite ":wq" e dê enter, para salvar o título do commit e sair do editor de texto.

6 - Agora rode novamente o comando para setar a branch master:

`==> git push --set-upstream origin master`

6 - Após isso, acesse a página do seu repositório pelo site do Github, atualize a página do repositório e você verá na tela que os arquivos que estão na sua pasta do computador agora estão dentro do seu repositório no Github na branch de nome master. Com isso você criou a primeira versão e seu primeiro commit do seu projeto.

Para alterações feitas nesse projeto:

7 - Tendo já tudo configurado e dado o primeiro commit, agora os próximos passos se tratam apenas de adicionar novos arquivos ao seu projeto e/ou editar arquivos já existentes, após ter feito uma dessas duas ações rode o comando abaixo para adicionar esse arquivo:

`==> git add .`

8 - Em seguida rode o comando para ver o status se o Git identificou que você modificou ou criou um novo arquivo:

`==> git status`

9 - Após isso rode o comando para criar o commit dessa alteração/criação, lembrando que o texto entre aspas é uma breve descrição da alteração que você fez:

`==> git commit -m "add uma nova linha no código"`

10 - Por fim, rode o comando para enviar essa alteração para o seu repositório lá no Github:

==> git push

11 - Processo concluído, os três comandos "add", "commit" e "push" precisam ser rodados sempre que você quiser enviar uma nova alteração para seu projeto, toda a configuração anterior para definir o repositório foram apenas configurações iniciais do Git que serão configuradas somente uma única vez em cada projeto.

OBS 1: Caso tenha aparecido um erro no retorno do comando anterior informando o seguinte texto "**Author identity unknown**", quer dizer que o Git não conseguiu identificar quem está fazendo o commit, sendo assim ele passou para você passar dois comandos, um para informar o email e outro para informar o nome, segue comandos abaixo:

==> git config --global user.email "gustavo.rios801208@gmail.com"

Dê enter e depois digite o outro comando:

==> git config --global user.name "Guganp-8012"

OBS 2: Para saber o status atual do repositório, digite o comando abaixo para visualizar arquivos que não foram enviados, os commits e etc:

==> git status

OBS 3: Para saber qual nome e email está configurado como autor:

==> git config --global user.name

OBS 4: Para verificar qual link de repositório está definido:

==> git remote -v

OBS 5: Para remover o link do repositório já definido:

==> git remote remove origin

OBS 6: Caso o queira setar a "main" para fazer o push, e esteja aparecendo um erro dizendo que o repositório do Github está avançou além do repositório local e você pretende mesclar ambas os repositórios:

==> git pull origin master --allow-unrelated-histories

Em seguida digite o título do commit na linha após as linhas com #. Depois de digitar, tecla "Esc" e em seguida digite ":wq" e dê enter, para salvar o título do commit e sair do editor de texto.