

Automação Cypress - Aula 1

criado: 18/12/2023

autor: Dionisio Soares

Objetivos

Nesta aula, foram abordados os conceitos e aplicações do uso Git e Github, e promovemos o primeiro arquivo relacionado ao projeto de automação.

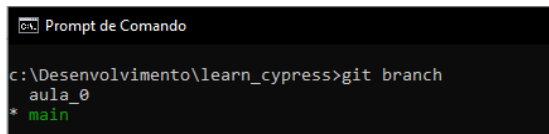
- Consulta do Ramo Atual (Branch);
- Criação e Alteração de Ramos (Branches);
- Criação do Arquivo package.json;
- Consulta e Adição de Alterações no Projeto;
- Captura Instantânea de Modificações (Commit);
- Principais problemas;
- Promoção de Alterações para o Repositório Remoto;
- Exercícios de Fixação e Extra;

Github

Consultando seu Ramo. (Branch)

Antes de iniciarmos, acesse via prompt de comando a pasta de seu projeto. Vamos consultar em que ramo do seu projeto está. Para isso vamos usar o comando abaixo:

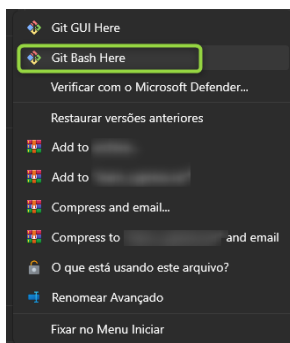
comando: `git branch`

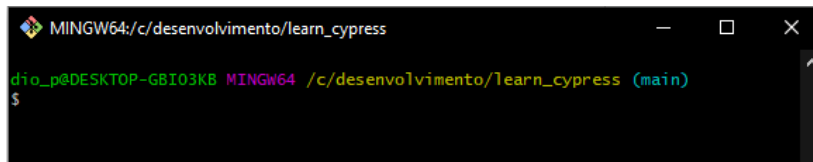


```
cmd Prompt de Comando
c:\Desenvolvimento\learn_cypress>git branch
aula_0
* main
```

Neste caso, estamos no ramo (**branch**) main! Está indicado com uma sinalização colorida e com “*”.

Outros prompts podem passar essa informação de forma mais direta, como o Bash. Você pode acessá-lo já na pasta que deseja, clicando com o botão direito na pasta de seu projeto, em “**bash here**”, conforme apresentado abaixo.



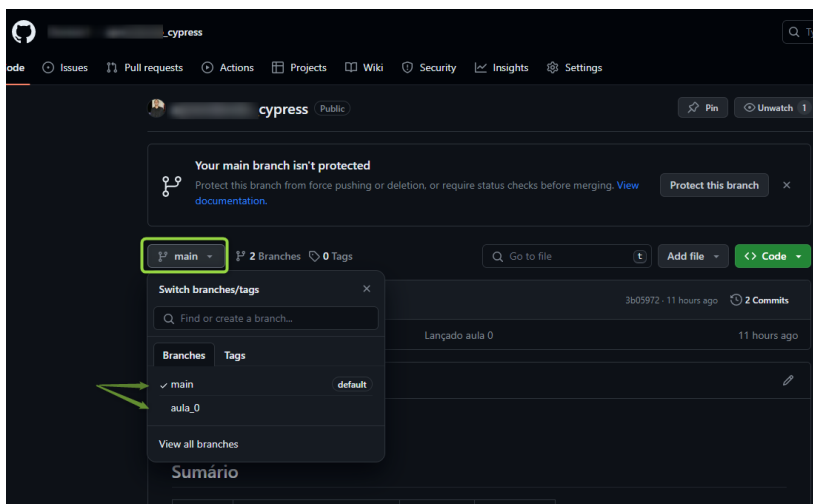


```

MINGW64/c/desenvolvimento/learn_cypress
dio_p@DESKTOP-GBI03KB MINGW64 /c/desenvolvimento/learn_cypress (main)
$

```

Observe que o último nome neste prompt, remete ao ramo que estais no projeto. Vamos dar uma olhada no repositório remoto do Github para vermos quantos ramos existem.

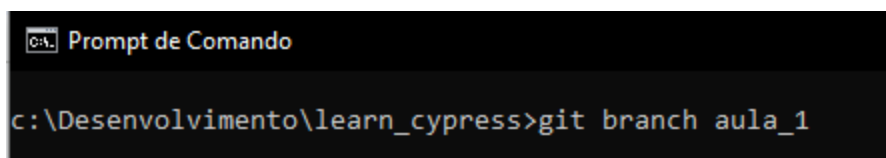


Observe que neste exemplo temos duas branches onde a principal é a **main**. No decorrer das aulas vamos abordar mais conteúdo sobre *branches* locais e remotas.

Criando um ramo (branch)

Nosso primeiro passo será criar um novo ramo local (*na sua máquina*) para trabalharmos de forma separada. No prompt de comando desejado, vamos usar o comando:

comando: `git branch "nome do ramo"`



```

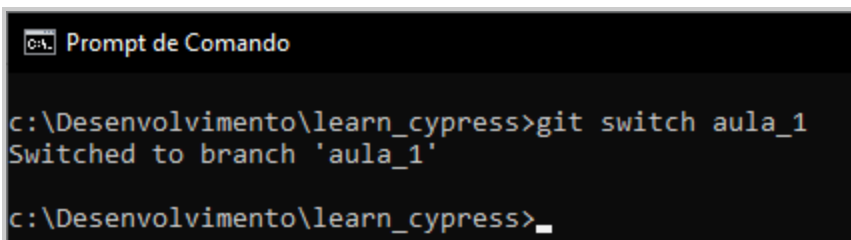
Prompt de Comando
c:\Desenvolvimento\learn_cypress>git branch aula_1

```

Alterando de ramo (branch)

Criamos! Agora precisamos acessá-la. Vamos usar o comando:

comando: `git switch "nome do ramo"`



```

Prompt de Comando
c:\Desenvolvimento\learn_cypress>git switch aula_1
Switched to branch 'aula_1'

c:\Desenvolvimento\learn_cypress>_

```

Criando arquivo package.json

Nosso segundo arquivo a ser criado é um arquivo importante que vai dar características e informações importantes para o projeto.

Para criarmos de forma prática, utilizaremos o comando a seguir:

comando: `npm init -y`

No arquivo criado, terá informações como:

"name", "version", "description", "main", "scripts", "keywords", "author", "license".

Caso tenha mais interesse, no blog do Gabriel tem mais informações sobre o arquivo. Consulte para mais informações. <https://gabrieluizramos.com.br/entendendo-o-package-json>

Consultando alterações no projeto

Para consultar o que houve de alterações no projeto, é simples, basta utilizarmos o comando a seguir:

comando: `git status`

```
c:\Desenvolvimento\learn_cypress (aula_1) (learn_cypress@1.0.0)
λ git status
On branch aula_1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    package.json

nothing added to commit but untracked files present (use "git add" to track)
```

O resultado desse comando apresenta a lista de arquivos novos ou modificados no projeto. No nosso exemplo apresenta apenas o arquivo "package.json".

Nesta imagem utilizei outro prompt de comando, que apresenta de forma mais amigável informações da branch em linha de comando. (<https://cmder.app>)

Promovendo as alterações ao projeto

Para que posteriormente essas alterações sejam contempladas no nosso repositório remoto, precisamos informar quais alterações irão ser contempladas no nosso ambiente local. No nosso exemplo, vamos indicar que desejamos incluir o arquivo "package.json", utilizando o comando abaixo:

comando: `git add "nome do arquivo"`

Em caso de ter mais de um arquivo, e deseje contemplá-los todos eles, não será preciso utilizar várias vezes o comando informado acima, utilize o comando abaixo:

comando: `git add .`

para mais comandos consulte: <https://gist.github.com/leocomelli/2545add34e4fec21ec16>

```
c:\Desenvolvimento\learn_cypress (aula_1) (learn_cypress@1.0.0)
λ git add .
```

Após ter adicionado, observe que ao consultar novamente as alterações no projeto, o mesmo apresenta como adicionado.

```
c:\Desenvolvimento\learn_cypress (aula_1) (learn_cypress@1.0.0)
λ git status
On branch aula_1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   package.json
```

Captura instantânea de modificações (Commit)

Quando decidimos as alterações que serão propostas no nosso projeto, precisamos unir todas essas informações que podem remeter a um ou vários arquivos. No nosso exemplo, temos apenas um arquivo. Essa captura é uma proposta instantânea das mudanças preparadas do projeto no momento. Os instantâneos podem ser considerados versões "seguras" de um projeto, o Git nunca os altera, a menos que você peça a ele.

Para consolidarmos as nossas alterações, utilizaremos o comando:

comando: `git commit -m "mensagem"`

```
c:\Desenvolvimento\learn_cypress (aula_1) (learn_cypress@1.0.0)
λ git commit -m "Promovendo arquivo package.json"
[aula_1 ef8c873] Promovendo arquivo package.json
 1 file changed, 12 insertions(+)
 create mode 100644 package.json
```

Promovendo as alterações para o repositório remoto

Tudo está quase pronto para promovermos nossas alterações. Faltava utilizarmos o comando para esse fim. Usaremos o comando a seguir:

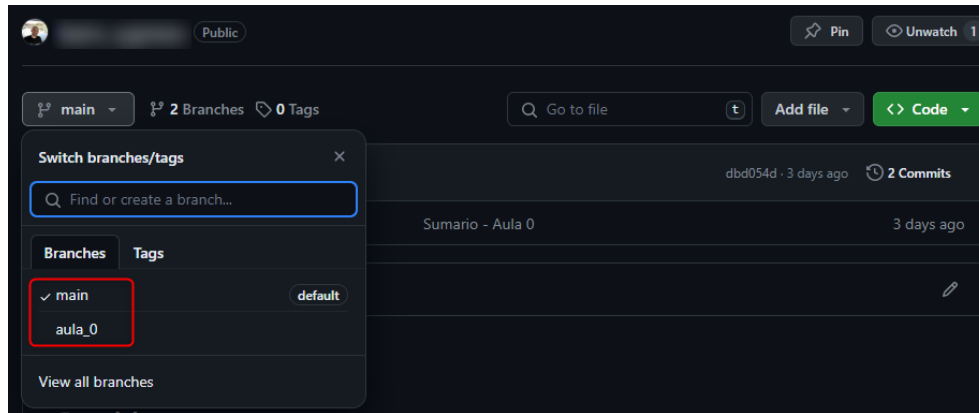
comando: `git push`

```
c:\Desenvolvimento\learn_cypress (aula_1) (learn_cypress@1.0.0)
λ git push
fatal: The current branch aula_1 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin aula_1
```

Opa! Aconteceu um problema! O Git não permitiu promovermos nossas alterações.

Isso ocorreu porque não temos a branch “aula_1” no repositório remoto. Vamos conferir, conforme imagem abaixo.



Para promovermos essa alteração e criarmos nossa branch, o próprio Git sugere o comando, conforme apresentado na imagem do prompt.

comando: `git push --set-upstream origin "branch"`

```
c:\Desenvolvimento\learn_cypress (aula_1) (learn_cypress@1.0.0)
λ git push --set-upstream origin aula_1
remote: Permission to /learn_cypress.git denied to .
fatal: unable to access 'https://github.com/learn_cypress.git/': The requested URL
returned error: 403
```

Ocorreu outro erro! O Git informou que não temos permissão!(403)

Precisamos configurar o nome e o e-mail nas configurações. Para isso no prompt de comando vamos adicionar o nome de nosso usuário, utilizando o comando a seguir.

comando: `git config --global user.name "seu nome"`

```
c:\Desenvolvimento\learn_cypress (aula_1) (learn_cypress@1.0.0)
λ git config --global user.name "Dionisio Soares"
```

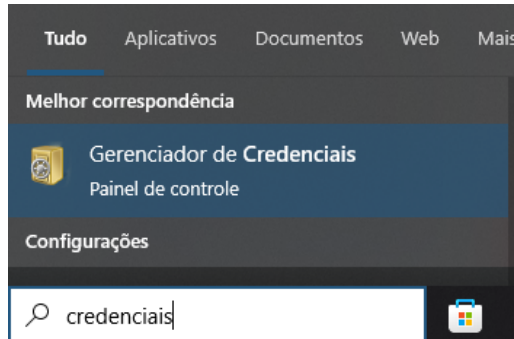
Agora vamos configurar o e-mail com o comando semelhante a seguir.

comando: `git config --global user.email seu e-mail`

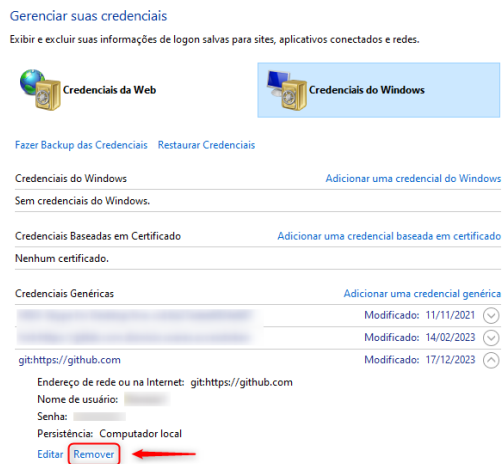
```
c:\Desenvolvimento\learn_cypress (aula_1) (learn_cypress@1.0.0)
λ git config --global user.email dionisio.soares@acct.global
```

Se você já tiver outro repositório Git, de uma outra conta, se faz necessário você remover essas credenciais do windows. Veremos a seguir.

Em pesquisar digite “credenciais”

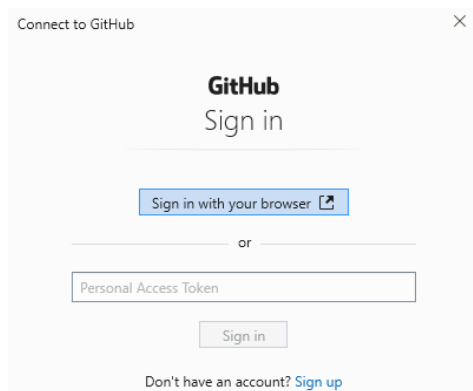


Encontre a credencial do Github e remova, conforme apresentado abaixo.

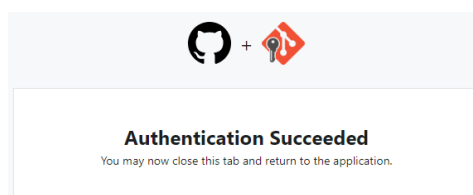


Agora vamos utilizar o comando push novamente!

O Github solicitará a confirmação de login, veja abaixo.



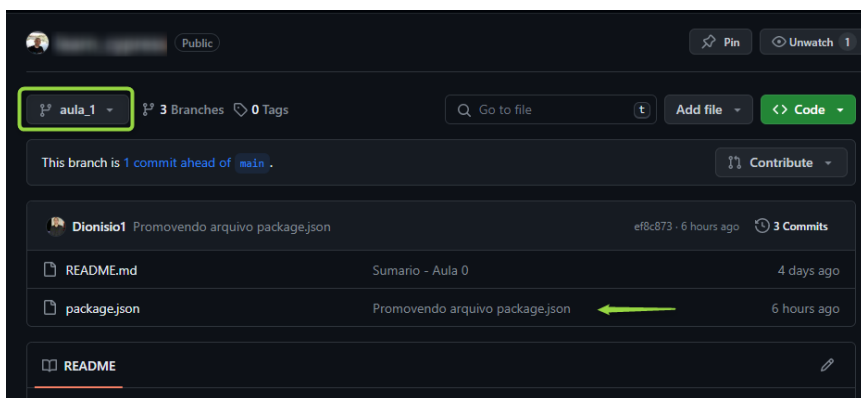
Clique em “Sign with your browser” e será concedido o devido acesso, conforme imagem abaixo.



Automaticamente a execução de seu comando push será executada.

```
c:\Desenvolvimento\learn_cypress (aula_1) (learn_cypress@1.0.0)
λ git push --set-upstream origin aula_1
info: please complete authentication in your browser...
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 547 bytes | 547.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'aula_1' on GitHub by visiting:
remote:   https://github.com/[redacted]/learn_cypress/pull/new/aula_1
remote:
To https://github.com/[redacted]/learn_cypress.git
 * [new branch]      aula_1 -> aula_1
Branch 'aula_1' set up to track remote branch 'aula_1' from 'origin'.
```

Pronto! Sua nova branch está remotamente!



Exercícios de fixação

Editar o README

Na última branch criada referente a essa aula (AULA 01), edite o arquivo README, para compor informações do que vimos nessa aula.

---- TEXTO ----

AULA 1

- Consulta do Ramo Atual (Branch);
- Criação e Alteração de Ramos (Branches);
- Criação do Arquivo package.json;
- Consulta e Adição de Alterações no Projeto;
- Captura Instantânea de Modificações (Commit);
- Principais problemas;
- Promoção de Alterações para o Repositório Remoto;
- Exercícios de Fixação e Extra:

---- TEXTO ----

Exercício Extra ★

Editar o README da aula anterior informando que conseguiu instalar os arquivos necessários.

---- TEXTO ----

AULA 0

- Instalação de ferramentas como Visual Studio Code Git e Node.js;
- Criação de um repositório no GitHub;
- Processo de baixar o repositório para a máquina local usando comandos Git;
- Comandos prompt;
- Exercícios práticos, como criar repositórios no GitHub e baixá-los;
- Exercícios extras, como a exclusão de repositórios no GitHub.

---- TEXTO ----