

Aula #8: Repositório Local vs Repositório Remoto

Repositório Local	1
Repositórios Remotos	2
Git vs Github	2
Como os repositórios Remoto e Local interagem entre si?	3
Exercícios	13
Próxima Aula	13
Fontes e Links Complementares	14

Repositório Local

Conforme vimos nas aulas anteriores, o Git cria um repositório onde ele guarda as informações e o histórico de alterações feitas no código ao longo do tempo. Mas o que é um repositório?

Um repositório nada mais é que diretório, ou pasta, onde o Git armazena todas as informações necessárias para executar o versionamento do código ou projeto.

No caso do Git, o repositório é o diretório `.git` que vimos nas aulas anteriores. Ou seja, a pasta `.git` é o repositório Git, e por ela estar na nossa máquina, de forma local, ele é conhecido como repositório local. Ou seja, todas as alterações que fizermos em nosso projeto ao longo do tempo, dentro do nosso repositório local, ficarão somente em nosso repositório local, até que façamos essa transferência para um repositório remoto.

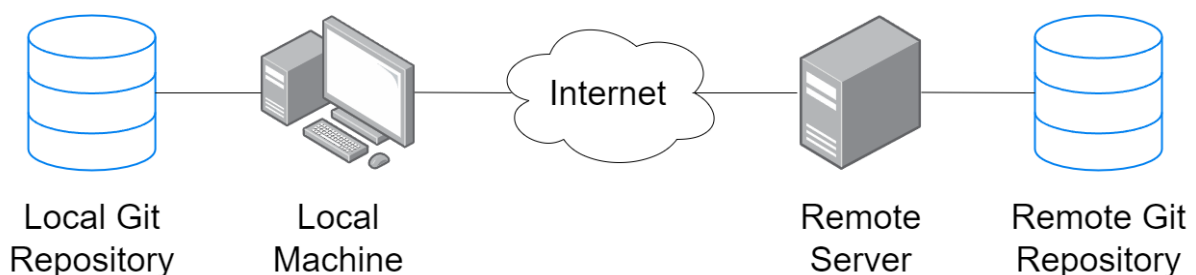
Cada projeto, que possui a sua própria pasta, e é versionado pelo Git, possui o seu próprio repositório local. Ou seja, na nossa máquina, podemos ter um ou mais repositórios, cada um contendo um projeto diferente.

Dessa forma, se desejamos manter o nosso repositório local funcionando, não devemos alterar nada do que existe dentro da pasta `.git`, uma vez que ela é o nosso repositório. E, caso ela seja deletada ou não copiada, todo o repositório local é perdido.

Repositórios Remotos

Como vimos anteriormente, um repositório local é um repositório que está na nossa máquina local. Já um repositório remoto, é um repositório que está na nuvem, sendo hospedado por algum serviço de hospedagem de repositórios Git, como o Github, Gitlab, Sourceforge ou Bitbucket, ou qualquer outro repositório que não esteja dentro da sua máquina local e que você tenha acesso.

Por tanto, um repositório remoto nada mais é que um repositório que não está em nossa máquina, e que temos uma cópia desse repositório dentro da nossa máquina local, possibilitando que nós possamos sincronizar o trabalho feito em nosso repositório local com o repositório remoto, e vice-versa.



Git vs Github

Conforme vimos anteriormente, o Git é uma ferramenta de versionamento de código, que cria repositórios para poder fazer esse controle do que foi realizado ao

longo do tempo. Mas qual a diferença entre Git e Github? Conforme vimos na seção anterior, o Github é um serviço que utiliza o Git para nos dar acesso a repositórios que ficam hospedados na nuvem, sendo que esses repositórios remotos podem ou não ser de nossa propriedade.

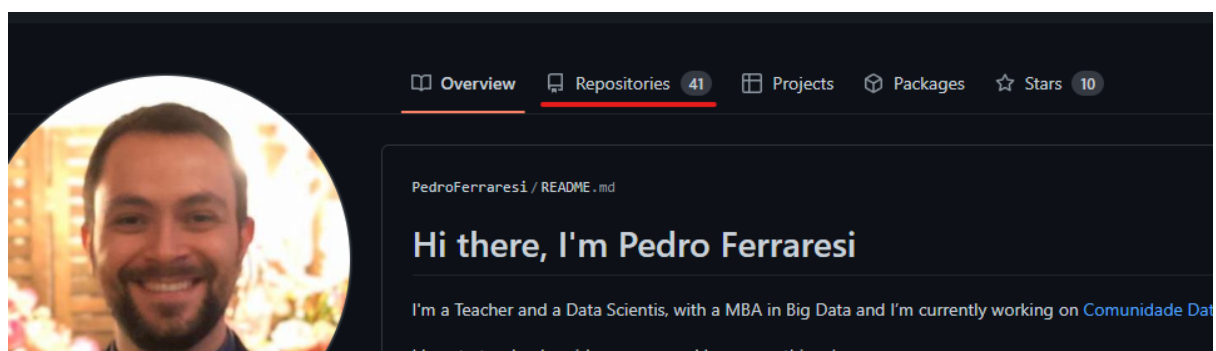
Por exemplo, se temos um projeto que está hospedado no Github, significa que temos uma instância de repositório Git localmente, na nossa máquina, e uma instância de repositório Git remoto, localizado nos servidores do Github.

Por tanto, o Github não é o Git, mas sim um serviço que se utiliza do Git. Assim como os serviços Gitlab e Bitbucket, por exemplo. Qualquer serviço que nos forneça acesso a servidores remotos, pode ser considerado um “irmão” ou “primo” do Github!

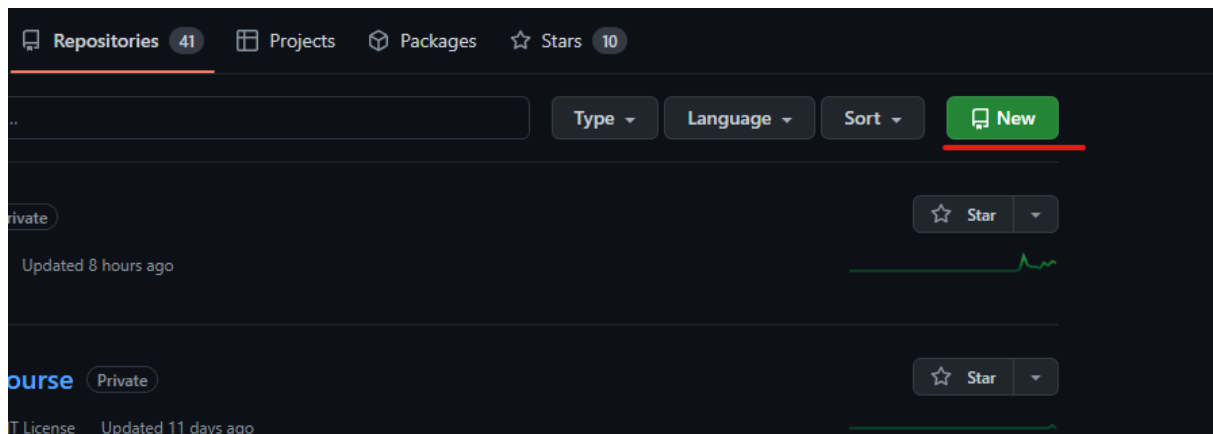
Como os repositórios Remoto e Local interagem entre si?

Para exemplificar novamente a diferença entre um repositório remoto e local, vamos criar o nosso primeiro repositório remoto dentro do github.

Primeiro, acesse a sua aba de repositórios dentro da sua página do Github:



Dentro da página de repositórios, crie um novo repositório clicando no botão verde chamada New:




Feito isso, coloque o nome do repositório de `curso_cds_git`. Faça também as seguintes configurações:

- Deixá-lo público
- Não adicionar o README.md
- Não configurar o arquivo .gitignore
- Não adicionar Licença de uso
- Caso queira adicionar alguma descrição, fique à vontade

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner *

 PedroFerraresi ▾

 /


Repository name *

curso_cds_git


✓

Great repository names are short and memorable. Need inspiration? How about **vigilant-system**?

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

 You are creating a public repository in your personal account.

Uma vez feito isso, clique em Criar repositório:

Conteúdo licenciado para André Brito Araujo - 433.934.378-10

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

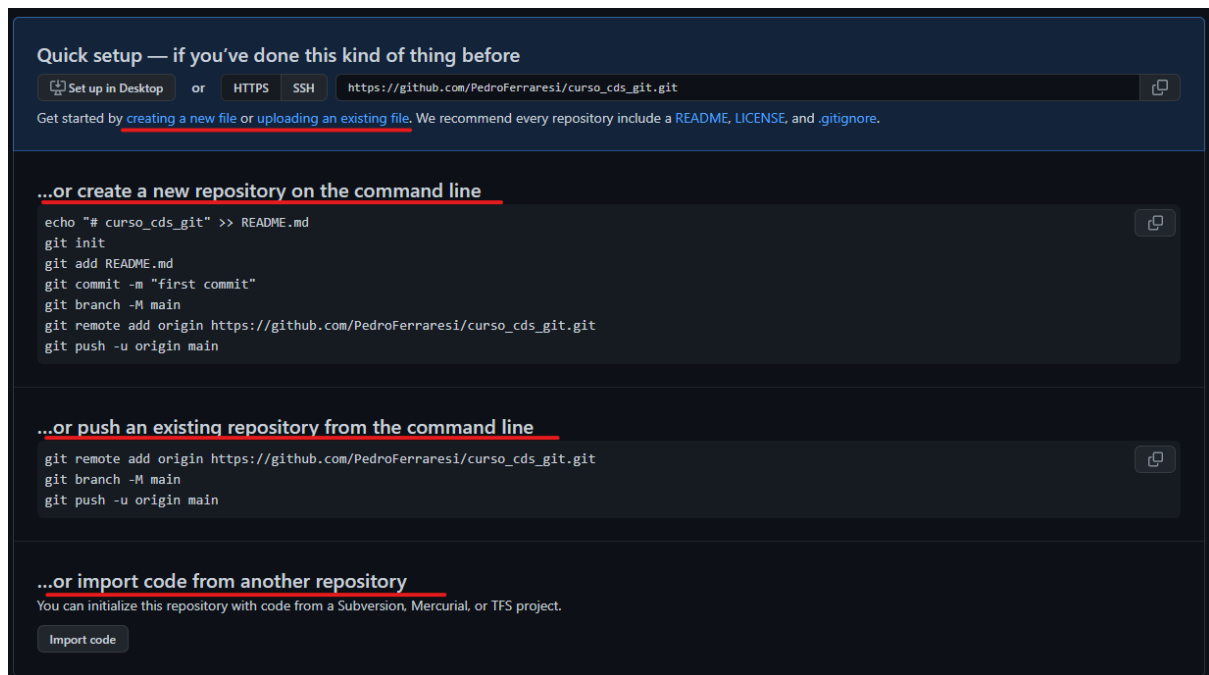
License: None ▾

You are creating a public repository in your personal account.

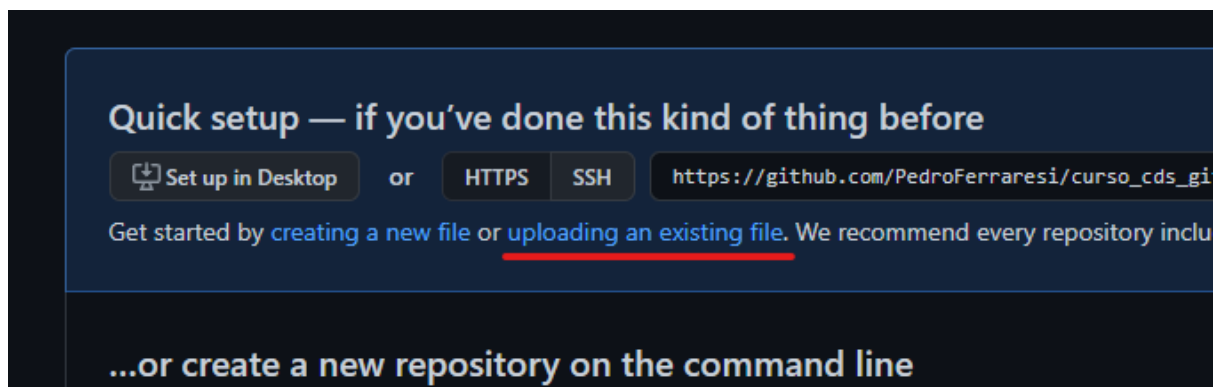
Create repository

Com o repositório criado, nos é oferecido algumas opções:

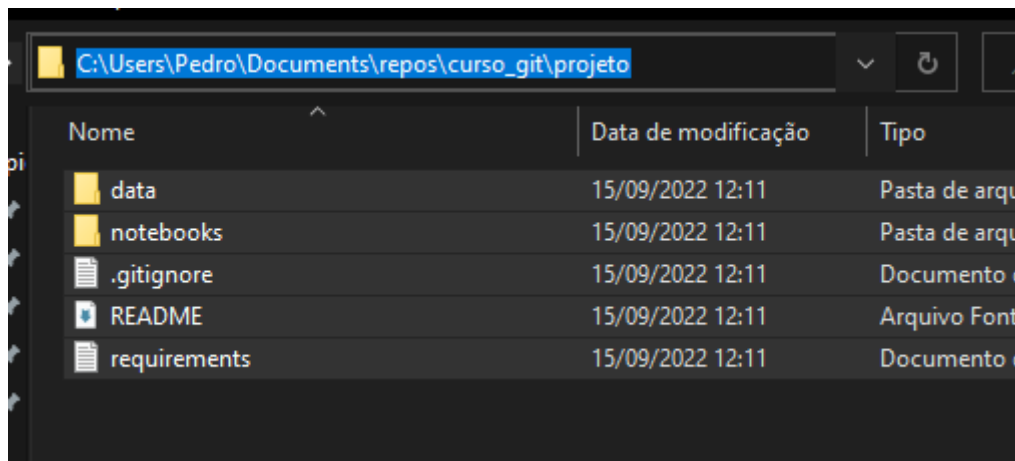
- Fazer a criação de um novo arquivo ou upload de um projeto
- Criar um novo repositório a partir da linha de comando (terminal Git)
- Fazer o Upload de um repositório existente através da linha de comando (terminal Git)
- Importar os códigos de um outro repositório



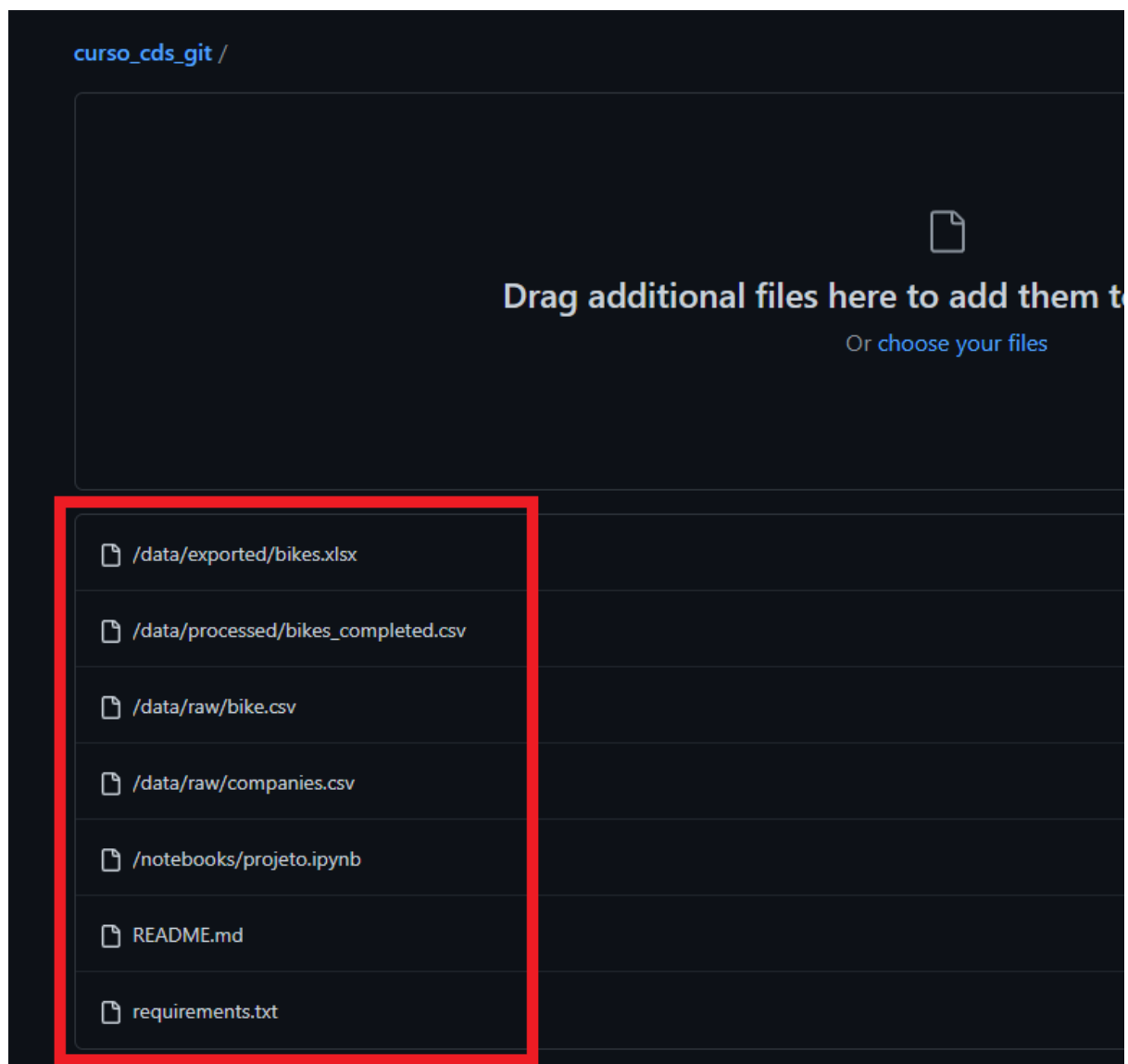
Como já temos um projeto feito, vamos fazer o upload dos nossos arquivos existentes. Para isso, selecione a opção upload an existing file:



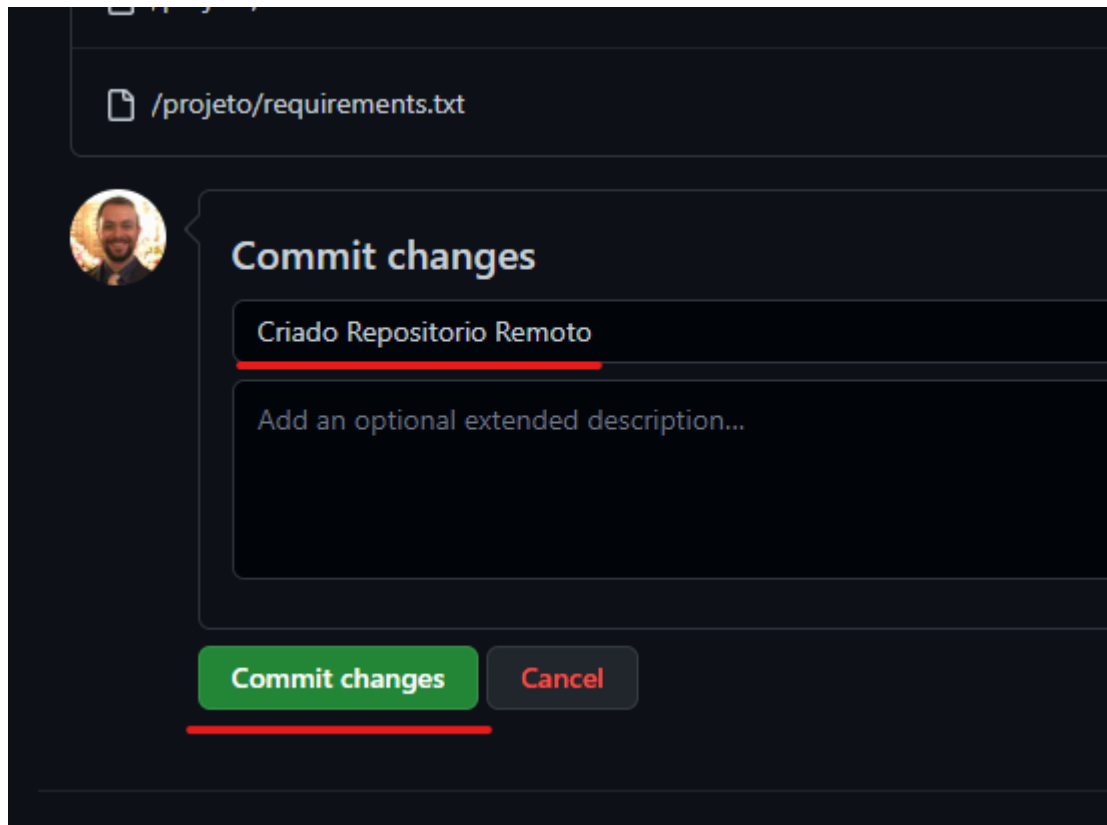
Será aberta uma nova janela onde poderemos adicionar o nosso projeto. Selecione todos os arquivos que estão dentro da pasta do projeto que está dentro da pasta de curso_git:



E arraste-os para dentro da área do github. Com isso, todos os arquivos e pastas serão adicionados:



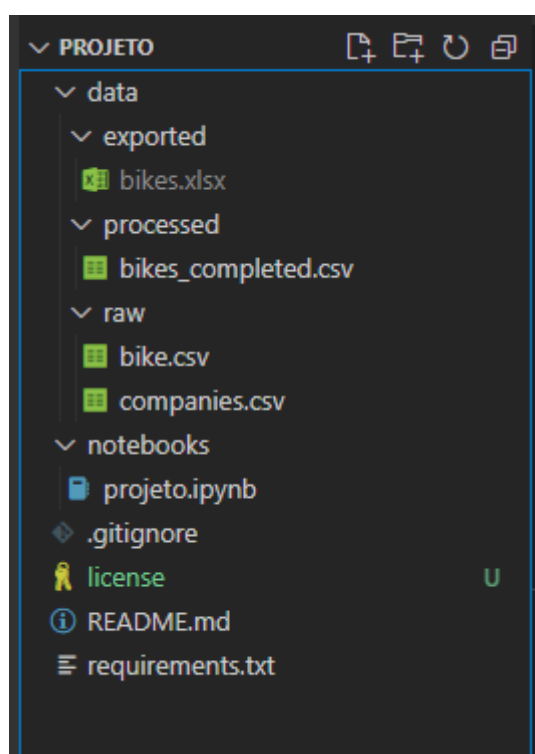
Com os arquivos adicionados, temos que commitar em nosso repositório remoto, no Github, essas alterações para podermos gravá-las. Vamos adicionar a mensagem Criado Repositorio Remoto e clicar Commit changes:



Com isso feito, será criado o repositório remoto do nosso projeto.



Com o repositório remoto criado, vamos fazer commits dentro do repositório local. Para isso, crie o arquivo chamado LICENSE, sem nenhuma extensão:



E coloque o seguinte texto dentro dele:

MIT License

Copyright (c) [year] [fullname]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Substituindo os campos [year] e [fullname] pelo ano e o seu nome.

```
license U x
license
1 MIT License
2
3 Copyright (c) 2022 Pedro Ferraresi
4
5 Permission is hereby granted, free of charge, to any person obtaining a
  copy of this software and associated documentation files (the
  "Software"), to deal in the Software without restriction, including
  without limitation the rights to use, copy, modify, merge, publish,
  distribute, sublicense, and/or sell copies of the Software, and to permit
  persons to whom the Software is furnished to do so, subject to the
  following conditions:
6
7 The above copyright notice and this permission notice shall be included
  in all copies or substantial portions of the Software.
8
9 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
  OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
  MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN
  NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
  DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
  OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
  USE OR OTHER DEALINGS IN THE SOFTWARE.
10 |
```

Feito isso, vamos commitar esse novo arquivo em nosso repositório local. Para isso, vamos utilizar novamente os comandos visto nas aulas anteriores: `git add <nome_arquivo>` e `git commit -m "Mensagem"`. O arquivo que desejamos informar ao Git que está pronto para ser commitado, é o arquivo `license`, que acabamos de criar. E a mensagem desse commit será Adicionado licença de uso:

```
Pedro@Ryzen MINGW64 ~/Documents/repos/curso_git_prep/projeto (main)
$ git add license
warning: in the working copy of 'license', CRLF will be replaced by LF t
it

Pedro@Ryzen MINGW64 ~/Documents/repos/curso_git_prep/projeto (main)
$ git commit -m "Adicionado licença de uso"
[main ca6e4b9] Adicionado licença de uso
1 file changed, 9 insertions(+)
create mode 100644 license
```

Mensagens de commit não devem possuir caracteres especiais, como acentos ou cedilhas. O padrão é a descrição do que foi feito no commit. Além disso, deve-se sempre utilizar o inglês como idioma padrão. No caso do nosso curso, será feito todo em português para fins didáticos.

Se recarregarmos a página do repositório remoto, que está dentro do Github, podemos observar que o arquivo que criamos ainda não está lá. Isso porque não fizemos nem o upload dele, e nem indicamos ao Git, na nossa máquina, que o repositório que está no Github é um repositório remoto. Sem essa indicação, ou configuração, não será possível fazer o upload do arquivo que acabamos de criar.

Ou seja, precisamos configurar ainda em nosso repositório local qual o nosso repositório remoto. Sem isso, não é possível fazermos a sincronia entre os repositórios. Porém, isso será feito em uma próxima aula!

A ideia desta aula é somente demonstrar as diferenças entre repositório remoto, as diferenças entre Git e um sistema armazenamento de repositórios remotos, como o Github, e que para que o repositório local seja integrado ao remoto, temos que configurar isso dentro do nosso repositório local.

Exercícios

Link para o formulário de exercícios de fixação de conteúdo: [Exercícios](#)

Próxima Aula

Na próxima aula, iremos verificar como funciona o ciclo de vida de um arquivo que está sendo versionado.

Fontes e Links Complementares

[Livro Pro Git - Capítulo 1.1 - Começando com Controle de Versões](#)

[Livro Pro Git - Capítulo 1.2 - Uma breve história do Git](#)

[Livro Pro Git - Capítulo 1.3 - O que é o Git?](#)