

Programação Python para Ciência de Dados

Git e Github

Controle de versões

Diferentemente de arquivos comuns, é muito útil salvar programas de forma a manter um histórico de suas versões

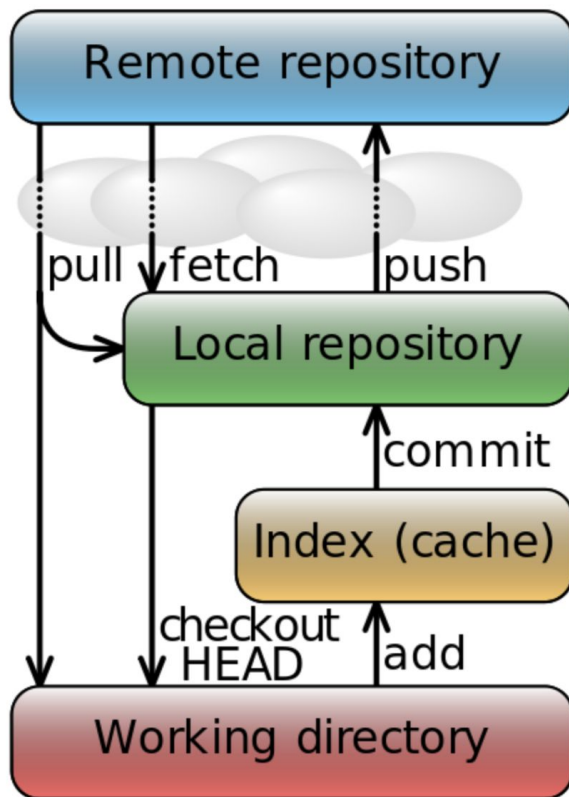
- voltar para versões anteriores quando necessário
- possibilitar o desenvolvimento simultâneo de diversas versões

Existem alguns sistemas que permitem realizar o controle de versões

Git: o que é?

- **Git** é atualmente o sistema de controle de versões mais utilizado
- Ele pode ser utilizado diretamente do terminal do computador (utilizando comandos de texto) ou através de programas com interface gráfica
- Vamos ver os principais comandos que podem ser feitos com o *git* de forma textual, mas o nome dos comandos é o mesmo nas interfaces visuais
- Iremos cobrir o necessário para o escopo do curso, acesse o link a seguir para se aprofundar no assunto :
 - https://rogerdudler.github.io/git-guide/index.pt_BR.html

Git



Git: comandos

- **init** - *git init*
 - Inicializa um repositório do *git* na pasta atual
- **add** - *git add <nome_do_arquivo>*
 - Adiciona o arquivo e suas alterações na lista de arquivos gerenciados
 - É possível adicionar todos os arquivos da pasta com o comando: *git add .*
- **commit** - *git commit -m <mensagem>*
 - Confirma as alterações nos arquivos que foram adicionados desde o último *commit*
 - É escrita uma mensagem informando o que foi alterado no programa

Git: comandos

- **Conecta repositório a servidor** - *git remote add origin <servidor>*
 - Atrela o código no computador a um servidor que irá armazenar cópias dos arquivos
 - (Já veremos como criar um servidor online)
- **push** - *git push origin master*
 - Envia o código do último *commit* para o servidor
- **pull** - *git pull*
 - Baixa a última versão presente no servidor

Git: comandos

- **clone** - *git clone <endereço_do_repositório>*
 - Cria uma cópia do código no repositório em um novo repositório já inicializado
 - O endereço pode ser local (do mesmo computador) ou de um servidor online
 - Mas aonde encontro repositórios de código?

Git: comandos

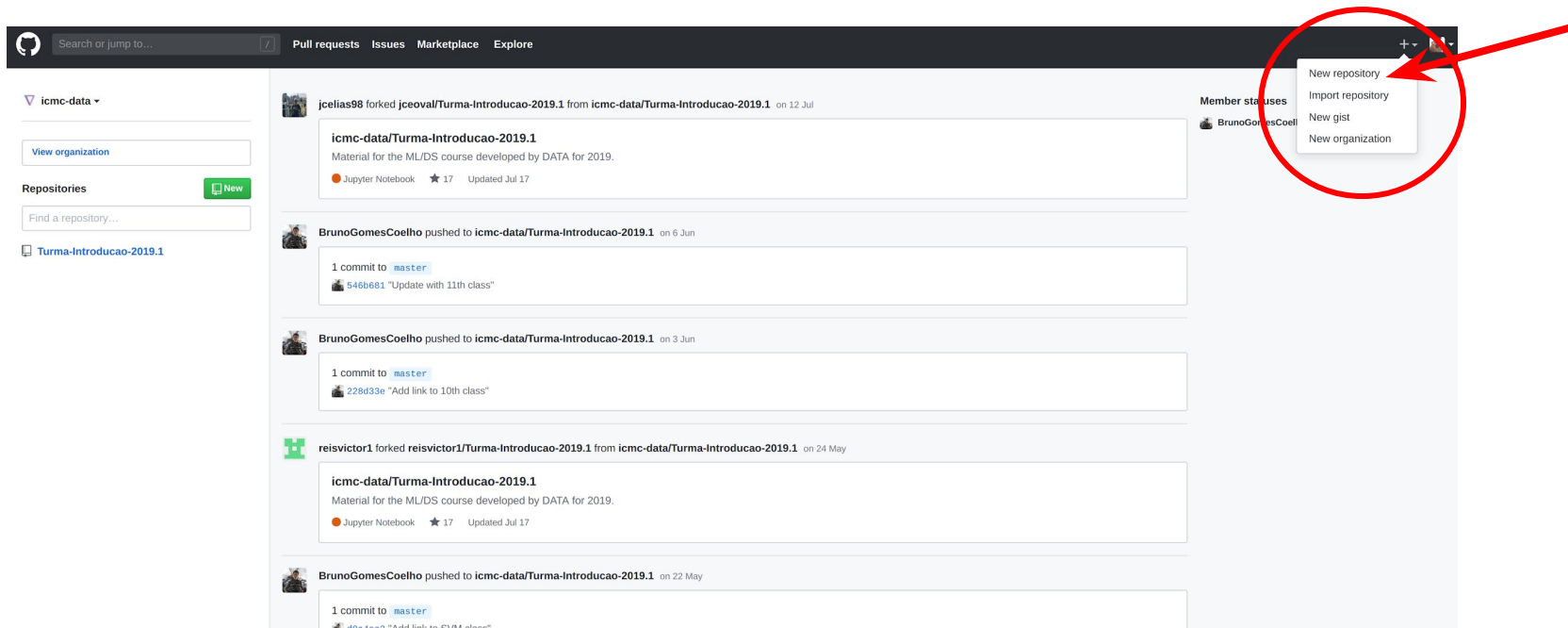
- **clone** - *git clone <endereço_do_repositório>*
 - Cria uma cópia do código no repositório em um novo repositório já inicializado
 - O endereço pode ser local (do mesmo computador) ou de um servidor online

Onde encontro repositórios de código? **Github!**

Github - <http://github.com>

- Github é uma plataforma para hospedagem de códigos, permitindo o controle de versões e trabalho colaborativo
- É possível criar repositórios abertos (que podem ser vistos por todos) ou privados
- Também é possível adicionar colaboradores que podem editar o conteúdo do repositório

Github: criando um repositório



Github: criando um repositório

Search or jump to... Pull requests Issues Marketplace Explore

Create a new repository

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

Owner: Repository name:

Great repository names are short and memorable. Need inspiration? How about [miniature-invention](#)?

Description (optional):

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

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

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: Add a license: ⓘ

Github Desktop

- Além do site, o Github possui um sistema que permite fazer o controle de versões utilizando uma interface gráfica
- O Github Desktop está disponível para Windows e macOS, podendo ser encontrado no link a seguir:
 - <https://desktop.github.com/>
- Todos os comandos de *git* ensinados possuem o mesmo nome nesta interface gráfica.