

Programação em Python (Back-end)

Instrutor: Pablo Araujo
(21) 97172-1697

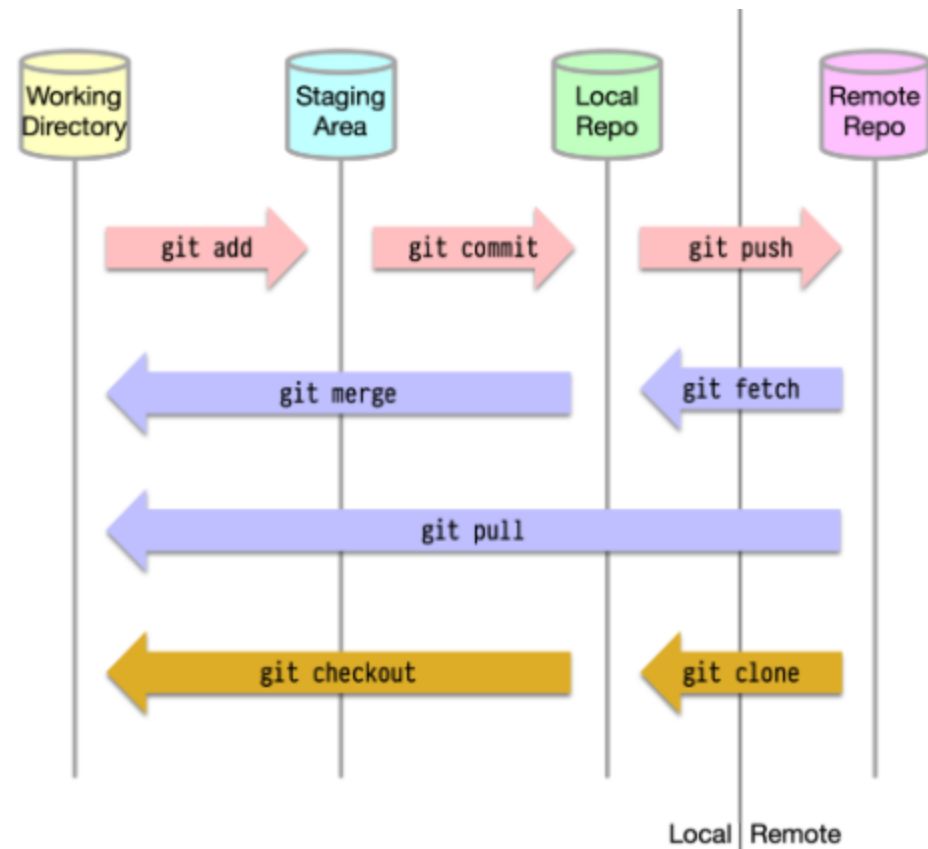


Aula 2

Aprendendo - GIT

O que é GIT?

- O Git é um sistema de controle de versão distribuído.
- Ele permite que você registre cada alteração no código, guardando um histórico completo do projeto.
- Facilita o trabalho em equipe, permitindo que vários desenvolvedores trabalhem em paralelo sem perder o controle.
- Ajuda a voltar para versões anteriores do projeto caso algo dê errado.
- É uma das ferramentas mais utilizadas no mercado de tecnologia para organizar e compartilhar código.



Aprendendo - GIT

Como instalar o GIT:

- Acesse o site oficial: <https://git-scm.com>.
- Baixe a versão correspondente ao seu sistema operacional (Windows, Mac ou Linux).
- No Windows, durante a instalação, mantenha as opções padrão (já vêm configuradas para a maioria dos casos).
- Após instalar, abra o terminal e digite:
`git --version`
Se aparecer a versão, significa que o Git foi instalado corretamente.



Aprendendo - GitHub

O que é GitHub

- O GitHub é uma plataforma online que hospeda repositórios Git na nuvem.
- Ele funciona como uma “rede social de código”, onde desenvolvedores compartilham, colaboram e versionam projetos.
- Permite armazenar projetos de forma segura, sem depender apenas da máquina local.
- Facilita o trabalho em equipe.
- É amplamente usado em projetos open source e no mercado de trabalho, sendo um portfólio público para programadores.




Aprendendo - GitHub

Fazendo cadastro no GitHub

- Acesse: <https://github.com>.
- Clique em Sign Up (Cadastrar).
- Informe:
 - E-mail válido
 - Senha
 - Nome de usuário (será o seu @ no GitHub)
- Confirme o cadastro no e-mail enviado pela plataforma.
- Após a confirmação, você terá acesso ao seu perfil GitHub e poderá criar repositórios.

Sign up for GitHub

 Continue with Google

or

Email*

Email

Password*

Password

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

Username*

Username

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

Your Country/Region*

Brazil

For compliance reasons, we're required to collect country information to send you occasional updates and announcements.

Email preferences

☐ Receive occasional product updates and announcements

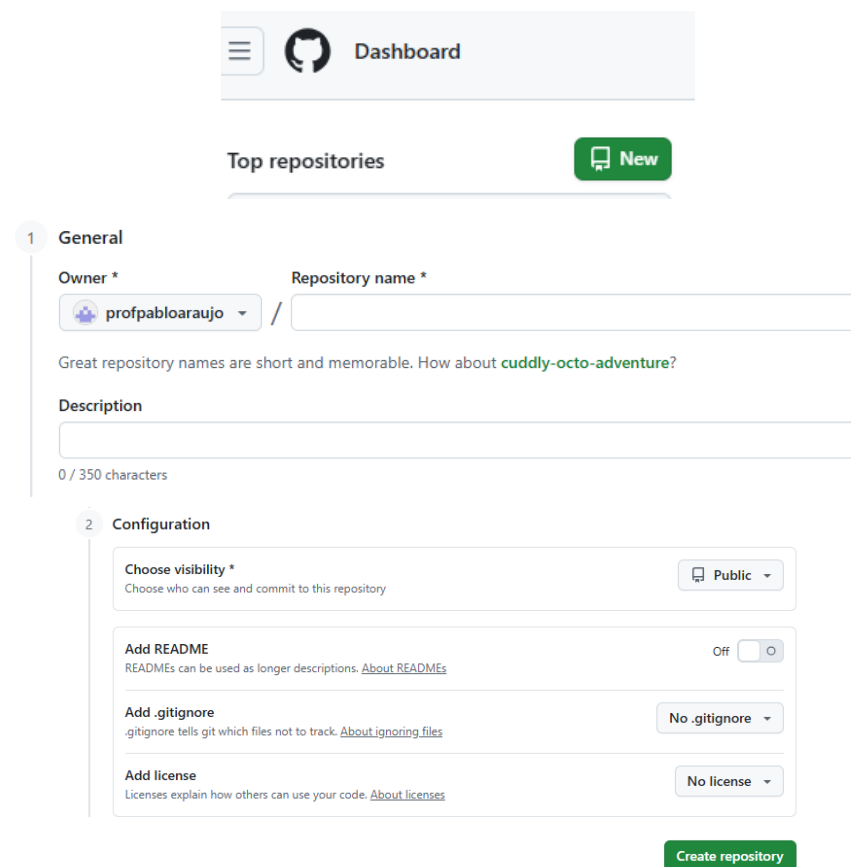
Create account >

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

Aprendendo - GitHub

Criar e clonar repositório (passo a passo)

1. Acesse github.com e faça login.
2. Clique em New (ou + » New repository).
3. Escolha nome do repositório (ex.: meu-projeto) e, opcionalmente, descrição.
4. Defina Public ou Private.
5. (Opcional) Marque Add a README / .gitignore / License se quiser já iniciar com arquivo.
6. Clique em Create repository.



The screenshot shows the GitHub 'New repository' form. At the top, there's a 'Dashboard' header with a menu icon and the GitHub logo. Below it, a 'Top repositories' section with a 'New' button. The form is divided into two tabs: 'General' (selected) and 'Configuration'. In the 'General' tab, there's a form with 'Owner *' (set to 'profpabloaraujo') and 'Repository name *' (empty). A hint suggests 'cuddly-octo-adventure?'. Below is a 'Description' field (0 / 350 characters). The 'Configuration' tab contains 'Choose visibility *' (set to 'Public'), 'Add README' (toggle 'Off'), 'Add .gitignore' (set to 'No .gitignore'), and 'Add license' (set to 'No license'). A 'Create repository' button is at the bottom right.

Dashboard

Top repositories New

1 General

Owner * Repository name *

profpabloaraujo /

Great repository names are short and memorable. How about [cuddly-octo-adventure?](#)

Description

0 / 350 characters

2 Configuration

Choose visibility * Choose who can see and commit to this repository Public

Add README READMEs can be used as longer descriptions. [About READMEs](#) Off

Add .gitignore .gitignore tells git which files not to track. [About ignoring files](#) No .gitignore

Add license Licenses explain how others can use your code. [About licenses](#) No license

Create repository

Aprendendo - GitHub

Clonar o repositório para sua máquina

- Copie a URL do repositório (HTTPS ou SSH) na página do repo > Code » escolha HTTPS ou SSH.

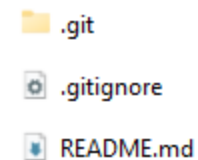
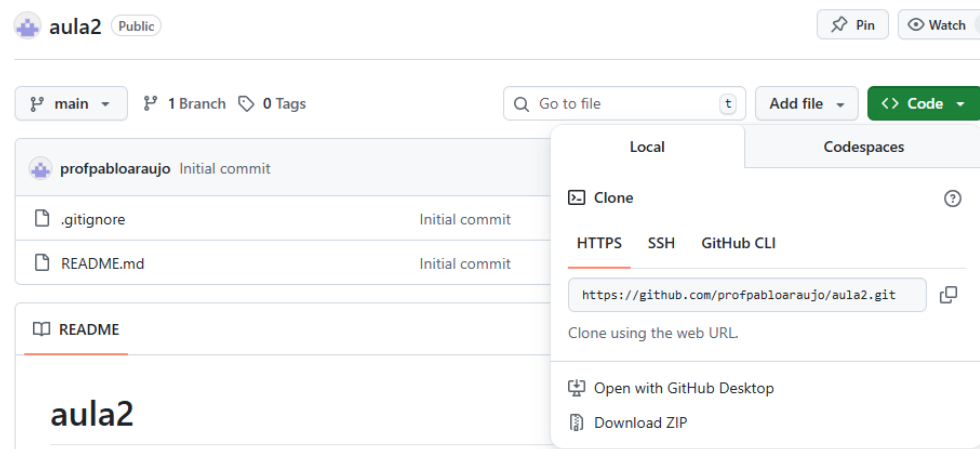
- No terminal, execute:

clonando via HTTPS

```
git clone https://github.com/seuusuario/nome-do-repo.git
```

Acesse a pasta:

```
cd nome-do-repo
```



Atividade

1 - Abrir a pasta do projeto no VS Code:

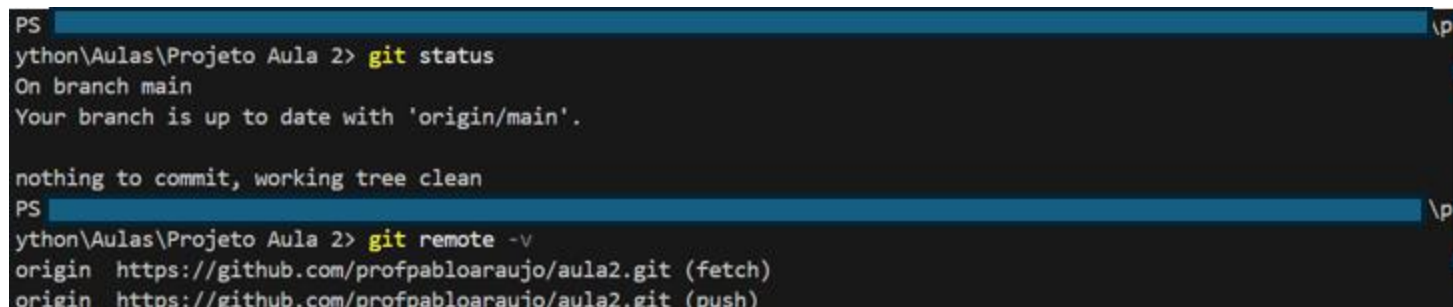
- Pelo menu: File → Open Folder... e selecione a pasta do projeto.
- Ao abrir, verifique a Explorer (barra à esquerda) com os arquivos do projeto.

2 - Abrir o terminal integrado:

- Menu: View → Terminal.
- Atalho: Ctrl+` (Windows/Linux)

3 – Comandos iniciais:

- git status
- git pull
- git remote -v



```
PS ython\Aulas\Projeto Aula 2> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS ython\Aulas\Projeto Aula 2> git remote -v
origin  https://github.com/profpabloaraujo/aula2.git (fetch)
origin  https://github.com/profpabloaraujo/aula2.git (push)
```

Atividade

Você recebeu duas listas:

`funcionariosParaCadastrar` → contém vários dicionários, onde cada um guarda as informações de uma pessoa (nome, sobrenome, idade, altura e se possui habilitação).

`cadastrosParaEnviarParaOBanco` → começa vazia e será usada para armazenar os objetos criados a partir dos dicionários.

```
funcionariosParaCadastrar = [  
    {"nome": "Pablo", "sobrenome": "Araujo", "idade": 34, "altura": 1.71, "temHabilitacao": True},  
    {"nome": "Ana", "sobrenome": "Silva", "idade": 28, "altura": 1.65, "temHabilitacao": False},  
    {"nome": "Carlos", "sobrenome": "Souza", "idade": 40, "altura": 1.80, "temHabilitacao": True}  
]
```

```
cadastrosParaEnviarParaOBanco = []
```

Seu objetivo:

1. Criar uma classe `Pessoa` que represente cada funcionário.
2. Criar uma função de cadastro que percorra a lista `funcionariosParaCadastrar`, crie objetos da classe `Pessoa` e adicione-os na lista `cadastrosParaEnviarParaOBanco`.
3. Criar uma função de salvar, que percorra a lista `cadastrosParaEnviarParaOBanco` e exiba a mensagem:
“O usuário <nome> <sobrenome> foi salvo com sucesso.”

Atividade

Passo a Passo para subir alterações no GitHub:

1. `git status` (mostra se existem arquivos novos ou modificados)
2. `git add .` (adicionar todos os arquivos modificados)
3. `git commit -m "Adicionando atividade da Aula 2 - Cadastro de Pessoas"` (Salva seu código no repositório local)
4. `git push origin main` (Enviar para o repositório remoto)

Dúvidas ?

Obrigado!

Instrutor: Pablo Araujo

