

Fatec

Faculdade de Tecnologia de Presidente Prudente

Análise e Desenvolvimento de Sistemas - AMS

Técnicas Avançadas de Programação Web e Mobile

Minicurso

Sistema de Controle de Versão: GIT (PARTE 1)



Profº Me. Bruno Santos de Lima

bruno.lima105@fatec.sp.gov.br

- **Contato:** bruno.lima105@fatec.sp.gov.br
 - **Plataforma Teams**
 - ✓ Avisos, disponibilização de material de estudo, envio de Trabalhos Práticos
 - **SIGA**
 - ✓ Chamadas e Notas de Avaliações e Trabalhos Práticos

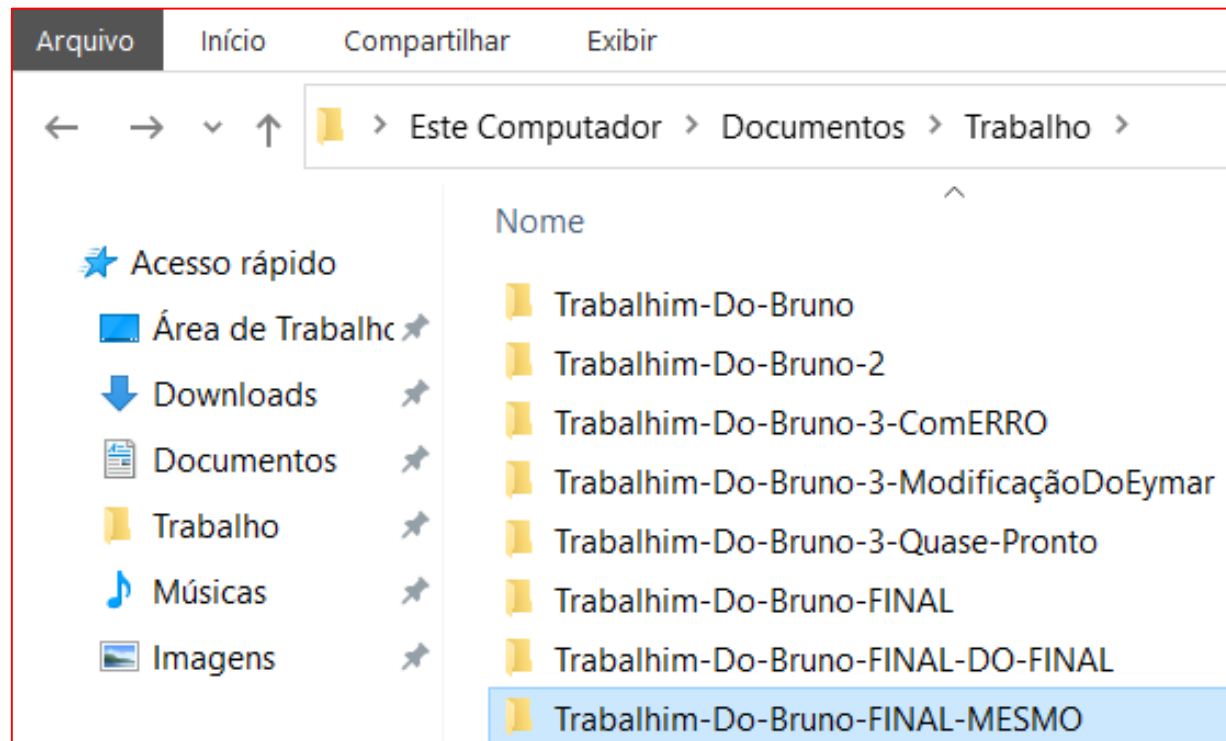
Versão de um Documento



Versão de um Documento

Ao desenvolver um trabalho, documento ou mesmo aplicação:

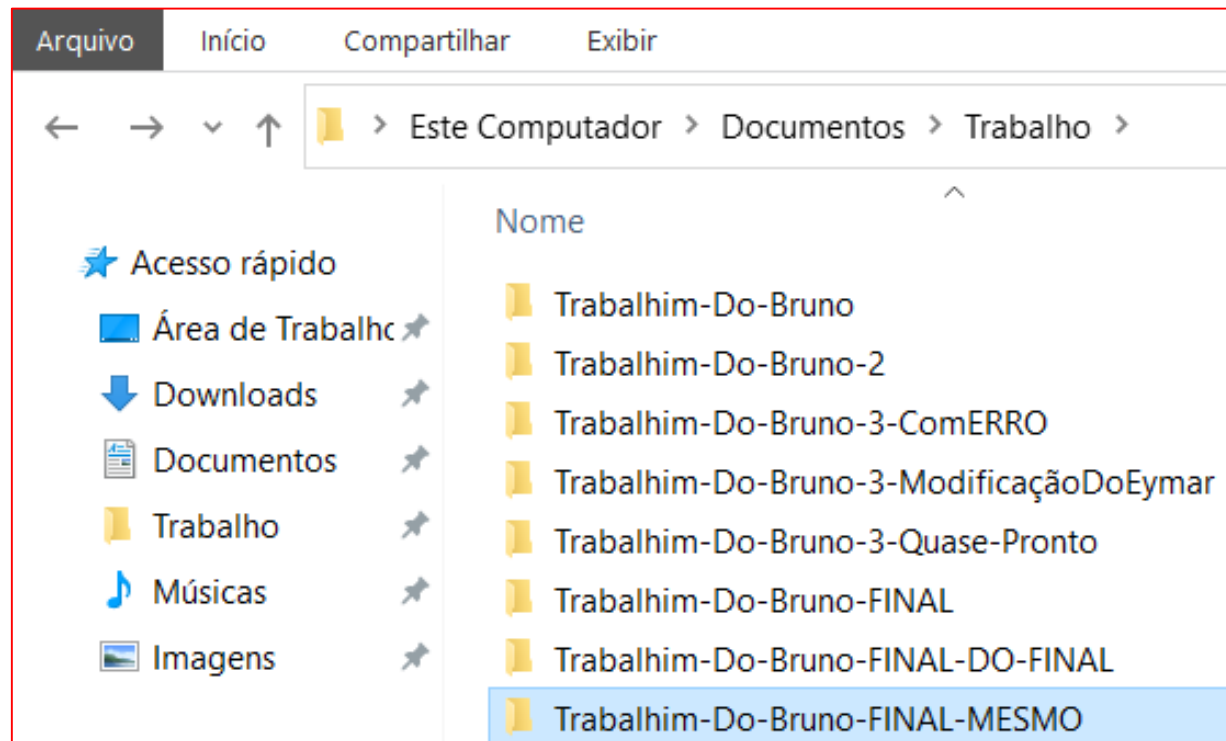
- É comum manter diversas cópias dos diferentes estágios do desenvolvimento.
- Cada cópia de um estágio finalizado é chamada de **Versão**.
- Provavelmente você já fez varias copias de um projeto para não perder as etapas anteriores do mesmo.



Versão de um Documento

Ao desenvolver um trabalho, documento ou mesmo aplicação:

- Cada cópia de um estágio finalizado é chamada de **Versão**.
- Manter **versões anteriores da liberdade de modificar** a versão atual sem a preocupação de inserir ERROS!
 - Afinal, caso um ERRO (bug) for inserido basta apagar e refazer com base na cópia anterior.



Sistema de Controle de Versão



“Um **Sistema de Controle de Versão (VCS)** tem a finalidade de **gerenciar diferentes versões** de um **documento (projeto)**, permite acompanhar o histórico de modificações, desenvolvimento em paralelo, formalizar versões ”

- Sistema de Controle de Código-fonte
 - **Source Code Management (SCM)**
 - Possibilita controlar versões do código-fonte de um projeto de software
 - ✓ Código-fonte, documentação e todo arquivo relacionado ao projeto de software.

- **Por que utilizar um VCS ou SCM?**

- **Principais vantagens:**

- ✓ **Controle de histórico:** Possibilita desfazer modificações, resgatar versões antigas e estáveis. Obter conhecimento através do histórico de desenvolvimento.
 - ✓ **Trabalho paralelo em equipe:** Diversas pessoas trabalham em paralelo sobre um mesmo conjunto de documento.
 - ✓ Exemplo: dois programadores podem modificar em paralelo um mesmo arquivo .py (python), sem a necessidade de um esperar o outro terminar sua modificação para então realizar uma nova modificação.
 - ✓ **Ramificações:** A partir de uma versão pode ser criado uma ramificação, cada equipe pode trabalhar em um ramo e depois unir para uma nova versão.

- **Por que utilizar um VCS ou SCM?**

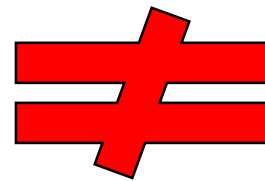
- **Principais vantagens:**

- ✓ **Marcações de versões estáveis e de teste:** Definir uma versão estável para uso e outras versões de teste com implementações que ainda estão sendo desenvolvidas.
 - ✓ **Segurança:** Possui mecanismos para evitar acesso de pessoas indevidas sem autorização ou mesmo invasão de agentes que possam danificar arquivos do projeto.
 - ✓ **Organização:** Todo o processo possibilita que o líder da equipe organize o projeto e etapas do trabalho, além de poder controlar visualmente o desenvolvimento por meio de uma interface.

Git e GitHub

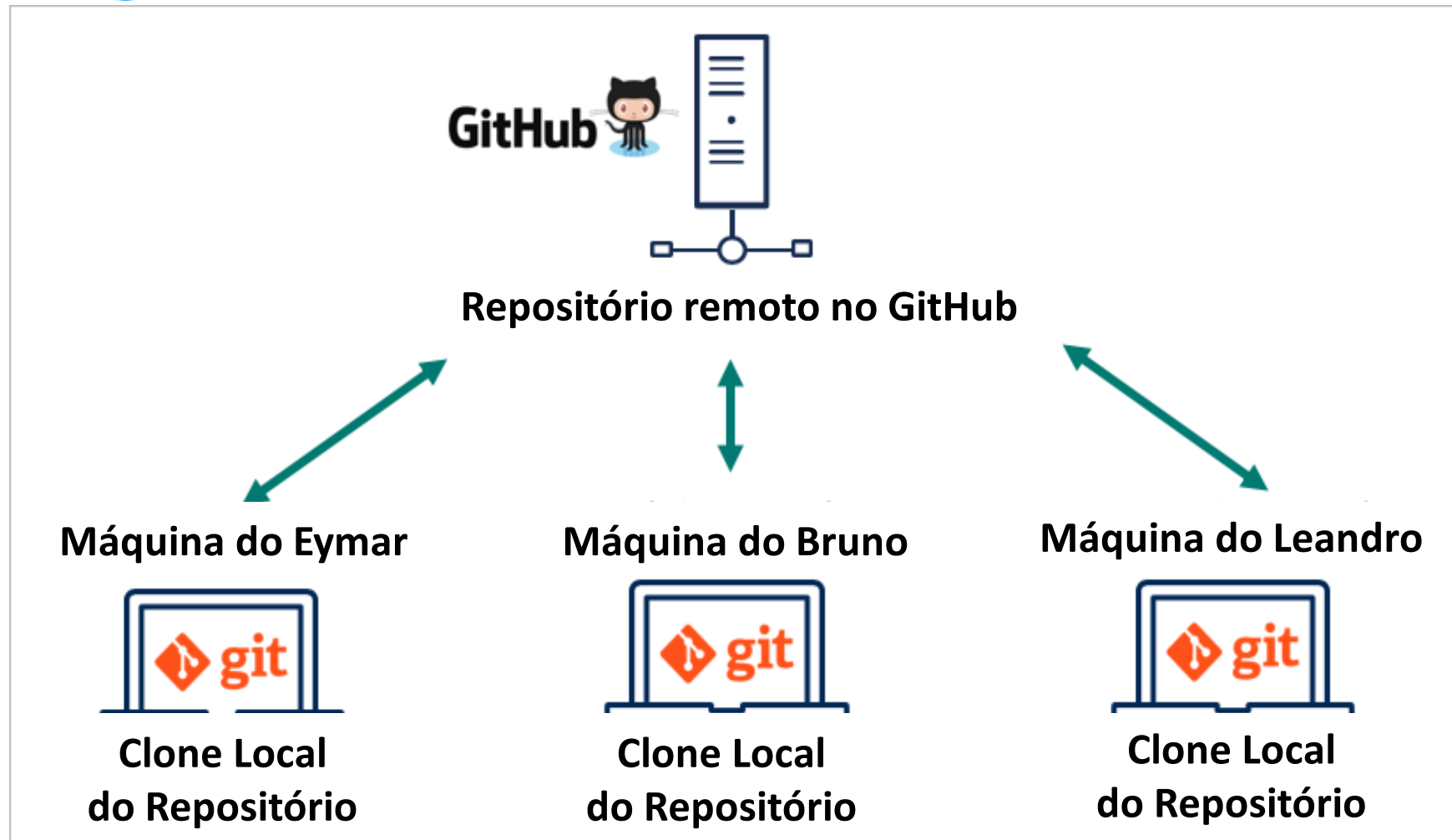


Git e GitHub são coisas diferentes!!!

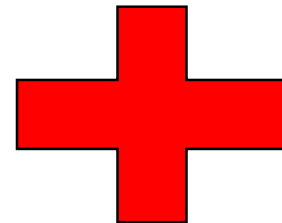


- ✓ Software de Controle de Versão (VCS)
- ✓ Rastreia alterações em arquivos locais e compartilha alterações com um repositório remoto

- ✓ Plataforma de hospedagem de projetos de software com controle de versão usando *Git*
- ✓ Rede Social para Programadores

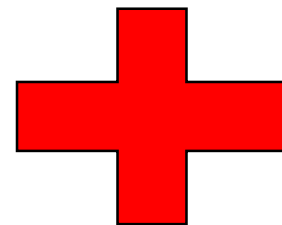


Git e GitHub: Ambos são necessários!!!



Existem outros repositórios remotos

Todos eles possuem funcionalidades semelhantes ao GitHub



- **Git:**

- **É um Sistema de Controle de Versão (VCS) multiplataforma**

- ✓ **Desenvolvedores:** Linus Torvalds e Junio Hamano

- ✓ **Página Oficial:** <https://git-scm.com>

- **Projetos de Software que utilizam o *Git*:**

- ✓ Android, Reddit, Eclipse, Distribuições Linux, PHP, JQuery, entre outros.

- **Escrito em:** C, Shell e Perl

- **Lançamento:** 7 de abril de 2005

- ✓ Versão estável 2.40.0: março de 2023



- **GitHub:**

- **É uma plataforma de hospedagem de código-fonte e arquivos de controle de versão usando *Git*.**
 - ✓ **Desenvolvedores:** Chris Wanstrath, J. Hyett, Tom Preston-Werner e Scott Chacon.
 - ✓ **Página oficial:** <https://github.com>
- Programadores do mundo inteiro podem se conectar e contribuir em projetos privados e *Open Source*.
 - ✓ **36 milhões de usuários ativos e abriga mais de 100 milhões de projetos.**
- **Proprietário:** Microsoft Corporation
 - ✓ **Adquiriu em 2018 o GitHub por 7,5 bilhões de dólares**
- **Lançamento:** fevereiro de 2008





Introducing GitHub Copilot X

Your AI pair programmer is leveling up



Let's build from here

Harnessed for productivity. Designed for collaboration.
Celebrated for built-in security. Welcome to the platform
developers love.



Email address

Sign up for GitHub

Start a free enterprise trial >



Introducing GitHub Copilot X

Your AI pair programmer is leveling up



Let's build from here

Harnessed for productivity. Designed for collaboration.
Celebrated for built-in security. Welcome to the platform
developers love.



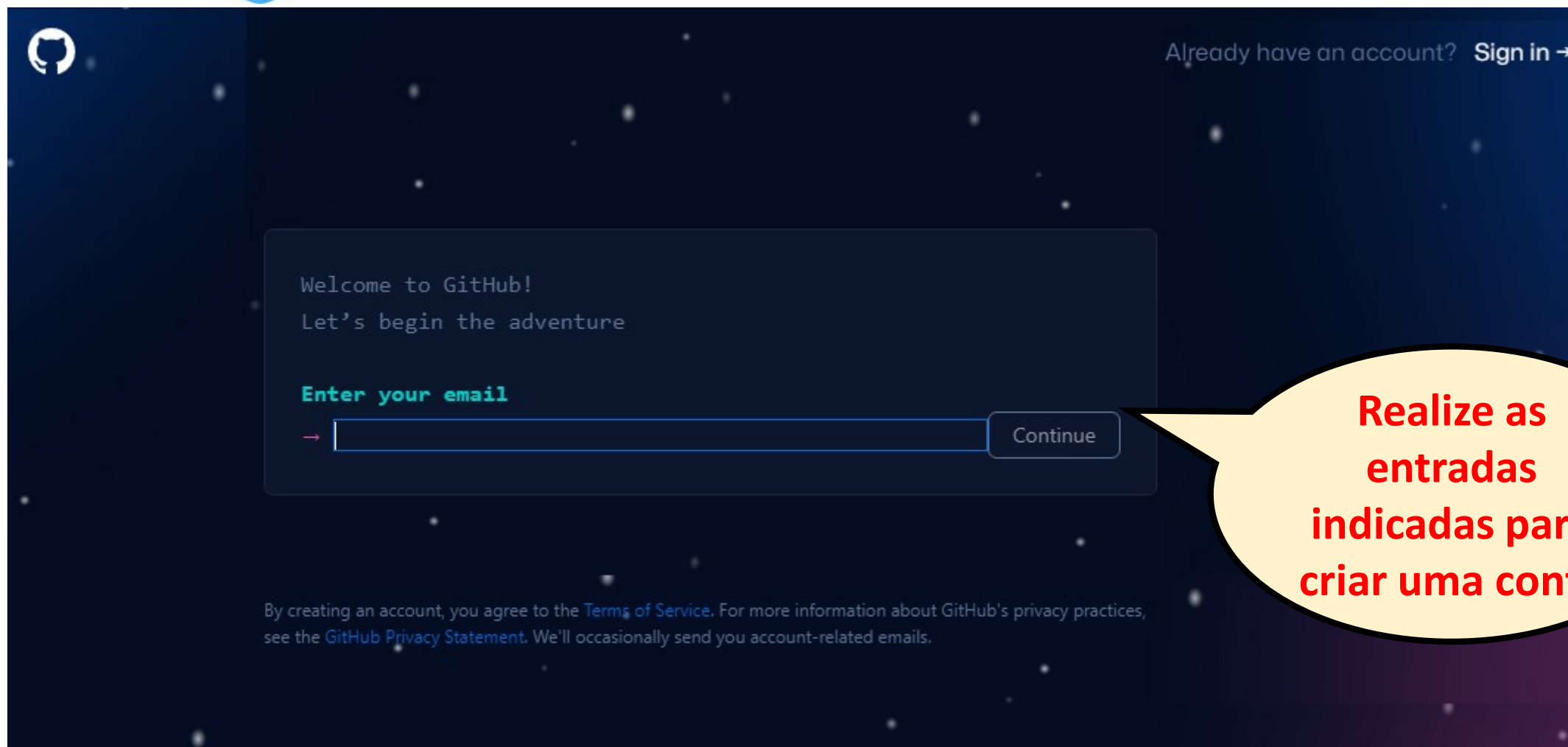
Email address

Sign up for GitHub

Start a free enterprise trial >

**Criar uma
conta!!!**

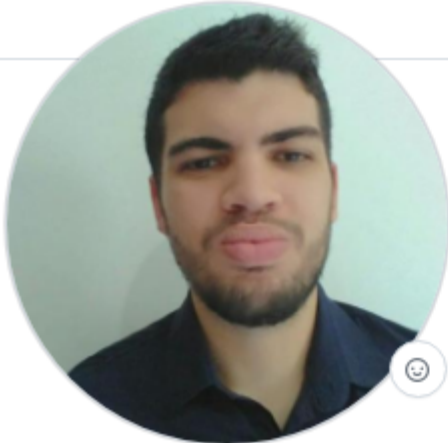
GitHub: Criar Conta



The screenshot shows the GitHub sign-up interface. At the top left is the GitHub logo. At the top right, it says "Already have an account? Sign in →". The main content area has a dark blue background with a starry pattern. A central white box contains the text "Welcome to GitHub! Let's begin the adventure". Below this, it says "Enter your email" in green. There is a text input field with a red arrow icon on the left and a "Continue" button on the right. At the bottom, there is a small text block: "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."

**Realize as
entradas
indicadas para
criar uma conta**

GitHub: Criar Conta – Perfil Overview



Bruno Santos de Lima
brunoslima · he/him

Doutorando em Ciência da Computação -
Universidade Estadual Paulista (UNESP)

Edit profile

22 followers · 32 following

Faculdade de Tecnologia de Presidente Prudente

Presidente Prudente - São Paulo - Brasil

Highlights
☆ PRO

Overview Repositories 34 Projects Packages Stars 20

Popular repositories

Technical-Debt-Catalog Public

Technical Debt Catalog - Code Debt - Repository - Scientific Initiation Project supported by the São Paulo Research Foundation - FAPESP, grant n°: 2016/15631-4

☆ 3

Compilador-LALG Public

Projeto de compiladores - FCT - Unesp - Presidente Prudente

Java ☆ 1

Minicurso-Desenvolvimento-Web-Front-end-HTML-CSS Public

Minicurso - Desenvolvimento Web Front-end: HTML e CSS

HTML ☆ 1

Dijkstra-Algorithm Public

Dijkstra Algorithm - Implementation and theory

Java ☆ 1

TDCodeVis-Technical-Debt-Visualization Public

TDCodeVis Tool - Repository - Scientific Initiation Project supported by the São Paulo Research Foundation - FAPESP, grant n°: 2016/15631-4

PHP ☆ 1

fctpp-bd-foodexpress Public

Este projeto consiste em um Sistema para gerenciamento de uma distribuidora de alimentos, com o intuito de armazenar os dados necessários para a empresa bem como fornecer os melhores caminhos para ...

PHP

Customize your pins

0 contributions in the last year

Contribution settings

	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
Mon												
Wed												
Fri												

Learn how we count contributions

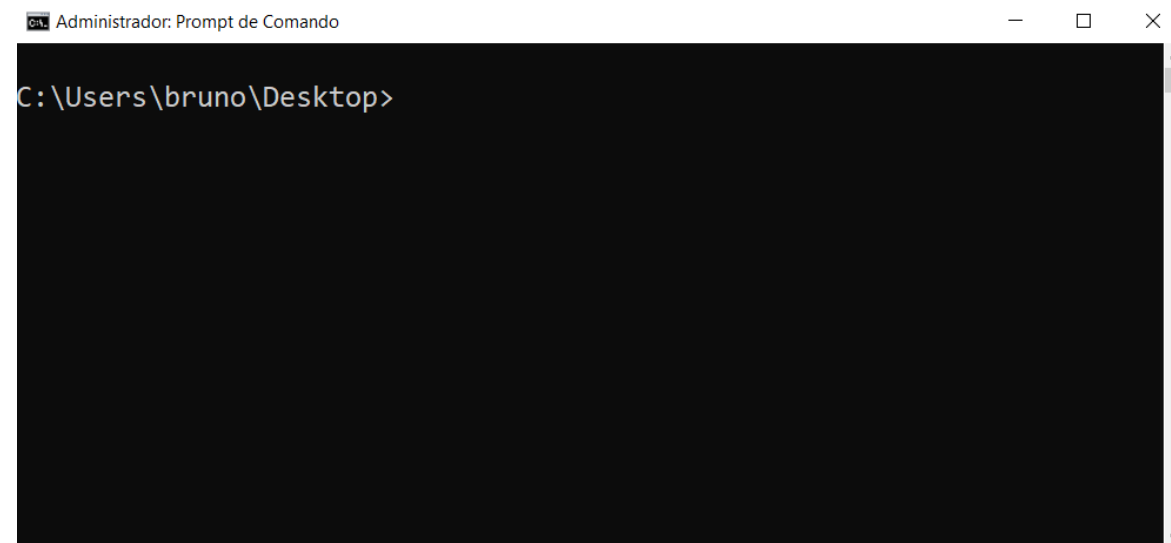
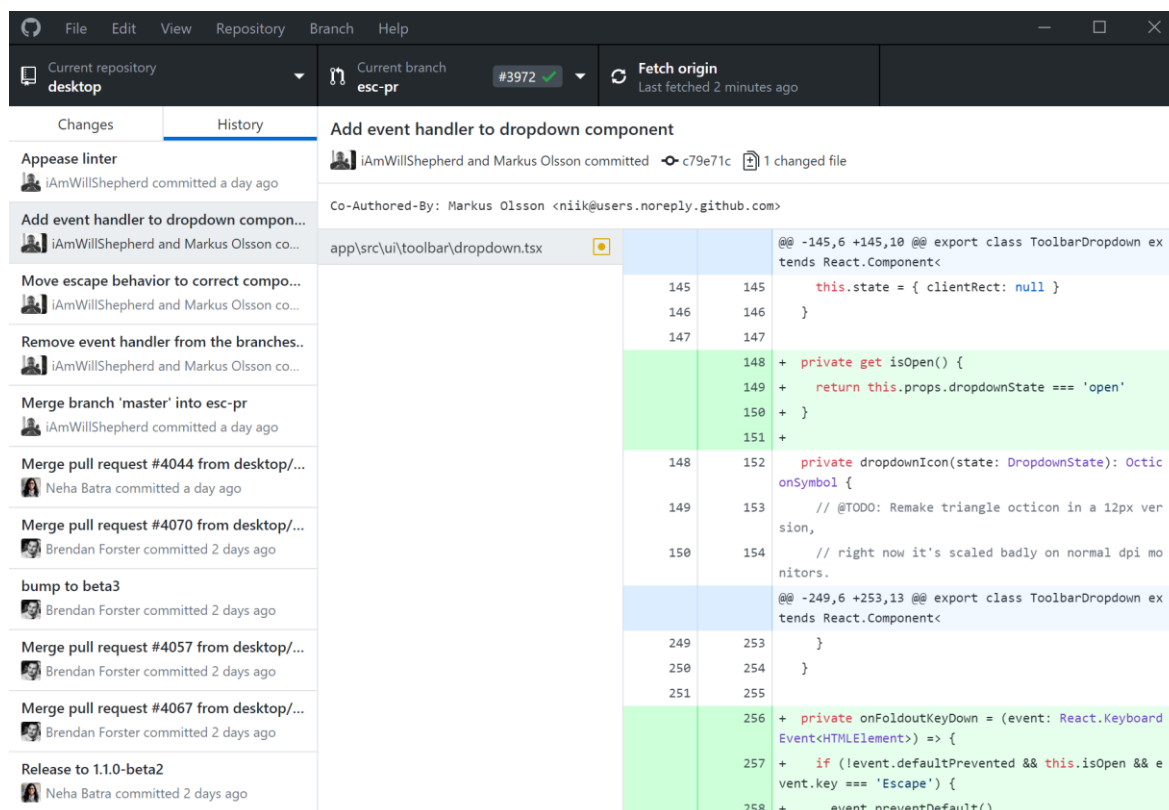
Less More

Git: Comandos de Configuração



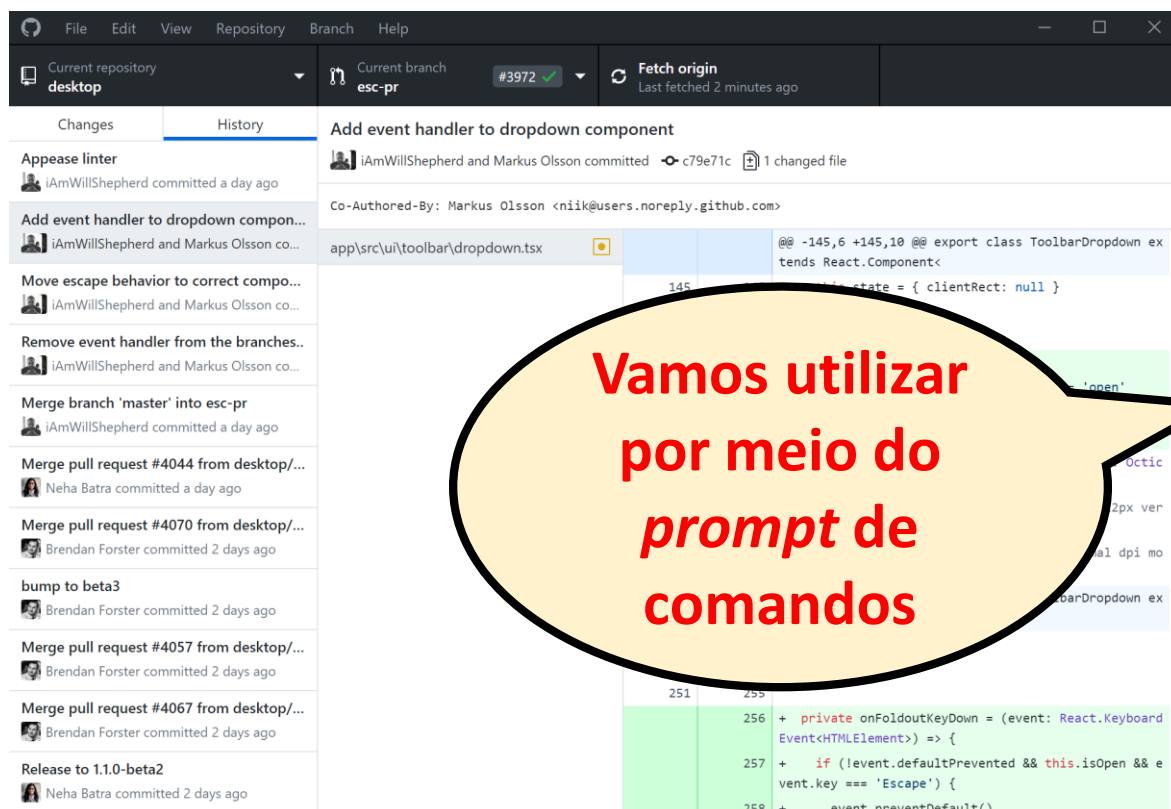
Git: Comandos de Configuração

- Existem duas formas de trabalhar com Git:
 - ✓ Usar *prompt* de comandos ou um aplicativo desktop com interface gráfica.

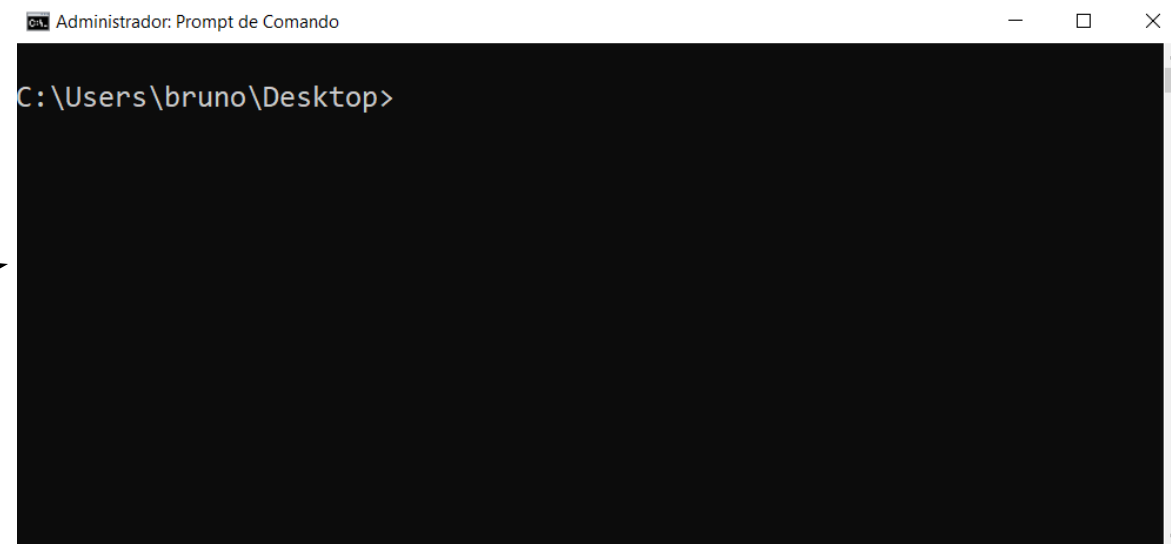


Git: Comandos de Configuração

- Existem duas formas de trabalhar com Git:
 - ✓ Usar *prompt* de comandos ou um aplicativo desktop com interface gráfica.

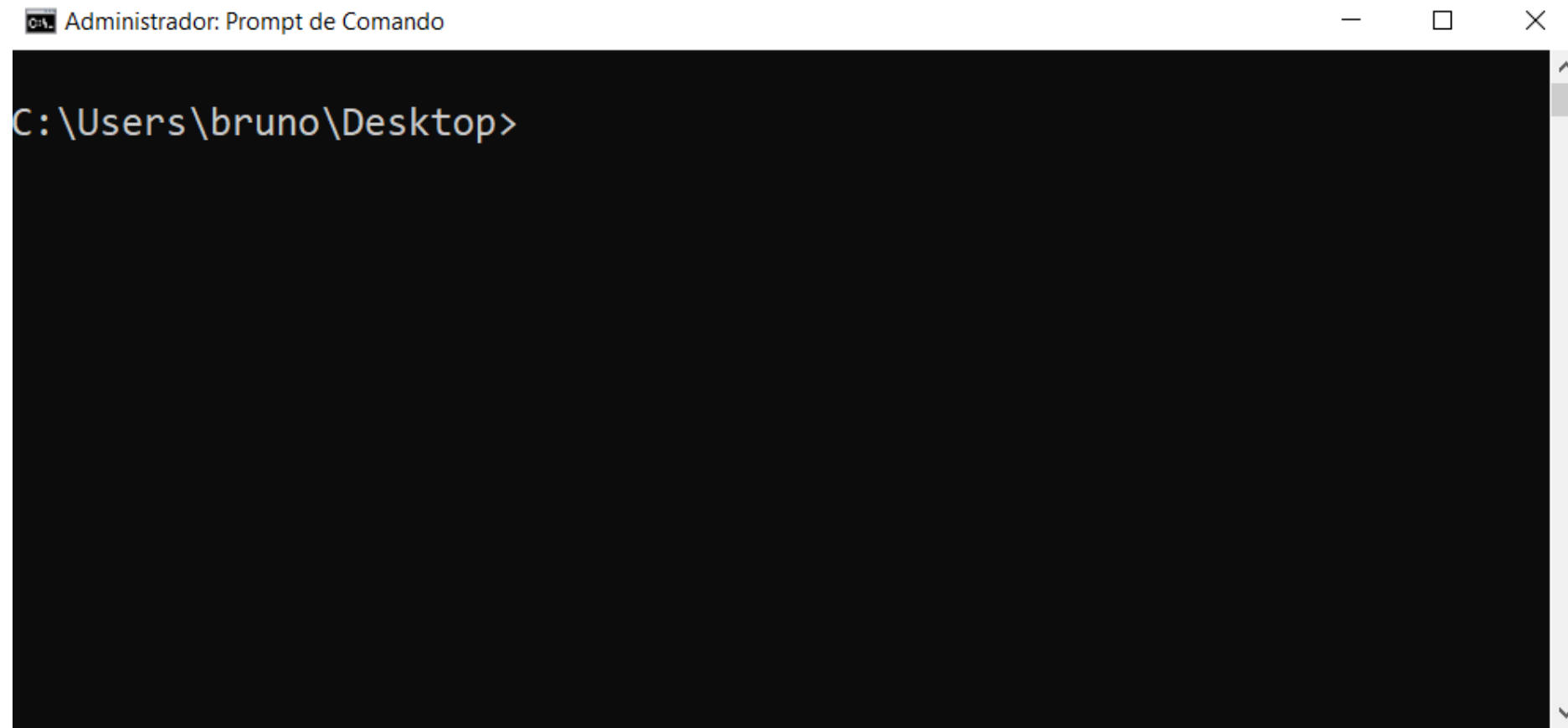


Vamos utilizar
por meio do
prompt de
comandos



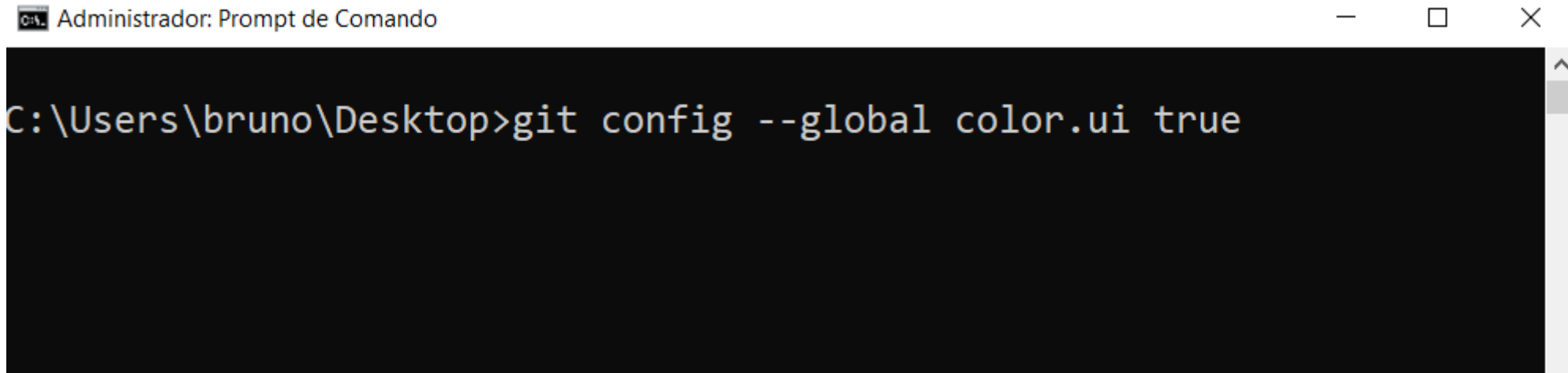
Git: Comandos de Configuração

- **Abrir o Prompt de Comando (cmd)**
 - ✓ Selecione o path do seu Desktop



Git: Comandos de Configuração

- ***git config --global color.ui true***
 - ✓ Comando de configuração global para habilitar sistema de cores ao realizar comandos




A screenshot of a Windows Command Prompt window titled "Administrador: Prompt de Comando". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt shows the current directory as "C:\Users\bruno\Desktop" and the command "git config --global color.ui true" has been entered and executed. The output is not visible, suggesting the command was successful.

```
C:\Users\bruno\Desktop>git config --global color.ui true
```

Git: Comandos de Configuração

- ***git config --global user.name "nome de usuário"***
 - ✓ Comando de configuração global para definir nome de usuário.
- ***git config --global user.email seuemail@exemplo.com***
 - ✓ Comando de configuração global para definir email.

 Administrador: Prompt de Comando

```
C:\Users\bruno\Desktop>git config --global user.name "brunoslima"  
C:\Users\bruno\Desktop>git config --global user.email brunoslima4@gmail.com
```

Git: Criar um repositório de Software



Git: Criar um repositório de Software (Local)

- **Crie uma pasta em sua área de trabalho:**
 - ✓ Repositorio-Aula
 - ✓ Acessar essa pasta pelo prompt de comando e utilizar o comando git init.
- ***git init***
 - ✓ Cria um novo repositório local

Administrador: Prompt de Comando

```
C:\Users\bruno\Desktop\Repositorio-Aula>git init
Initialized empty Git repository in C:/Users/bruno/Desktop/Repositorio-Aula/.git/
```

Git: Criar um repositório de Software (Local)

- ***git status***

✓ Verifica o status (estado) dos arquivos e diretórios do repositório, informações gerais.

Administrador: Prompt de Comando

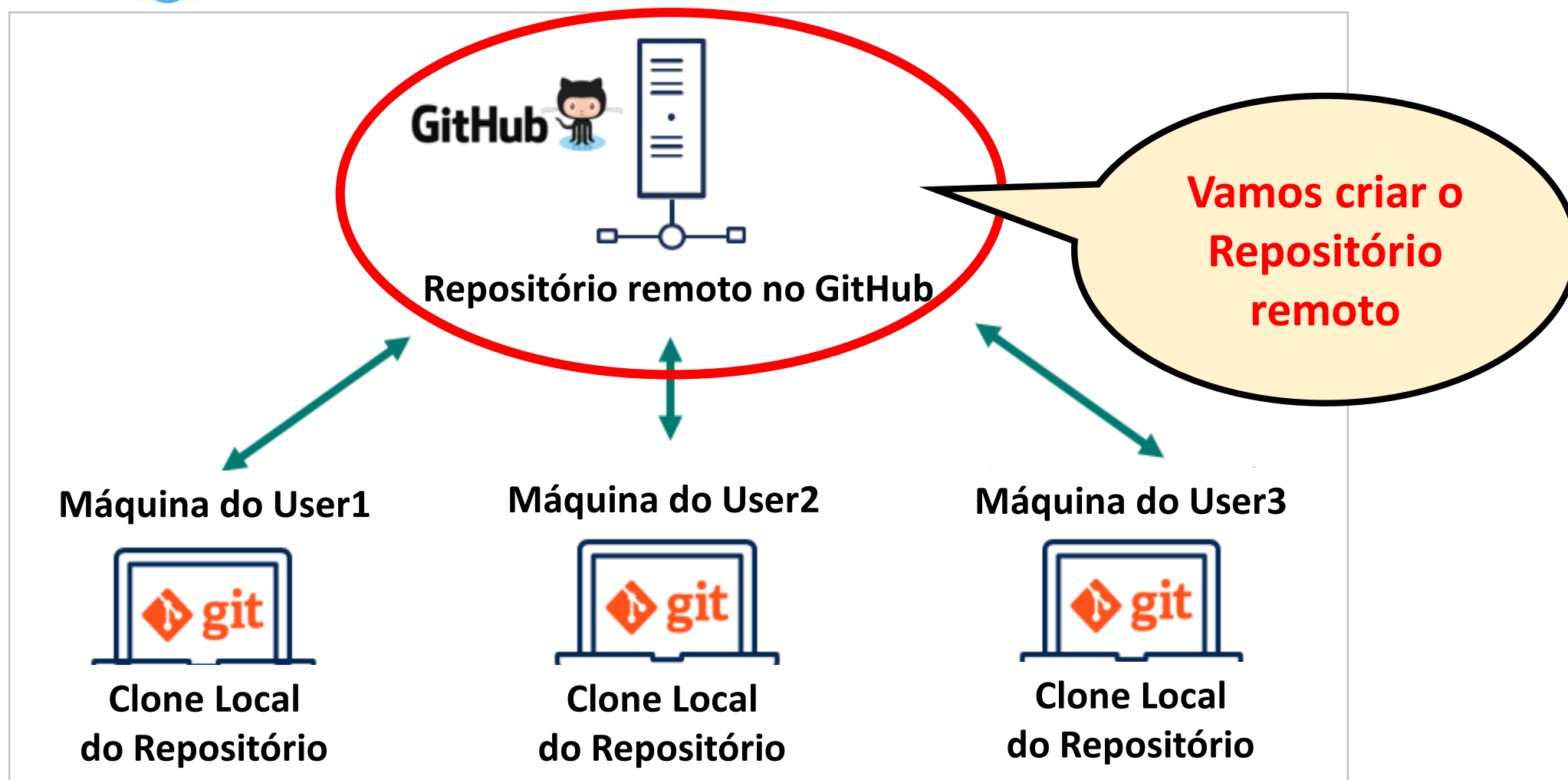
```
C:\Users\bruno\Desktop\Repositorio-Aula>git init
Initialized empty Git repository in C:/Users/bruno/Desktop/Repositorio-Aula/.git/

C:\Users\bruno\Desktop\Repositorio-Aula>git status
On branch master

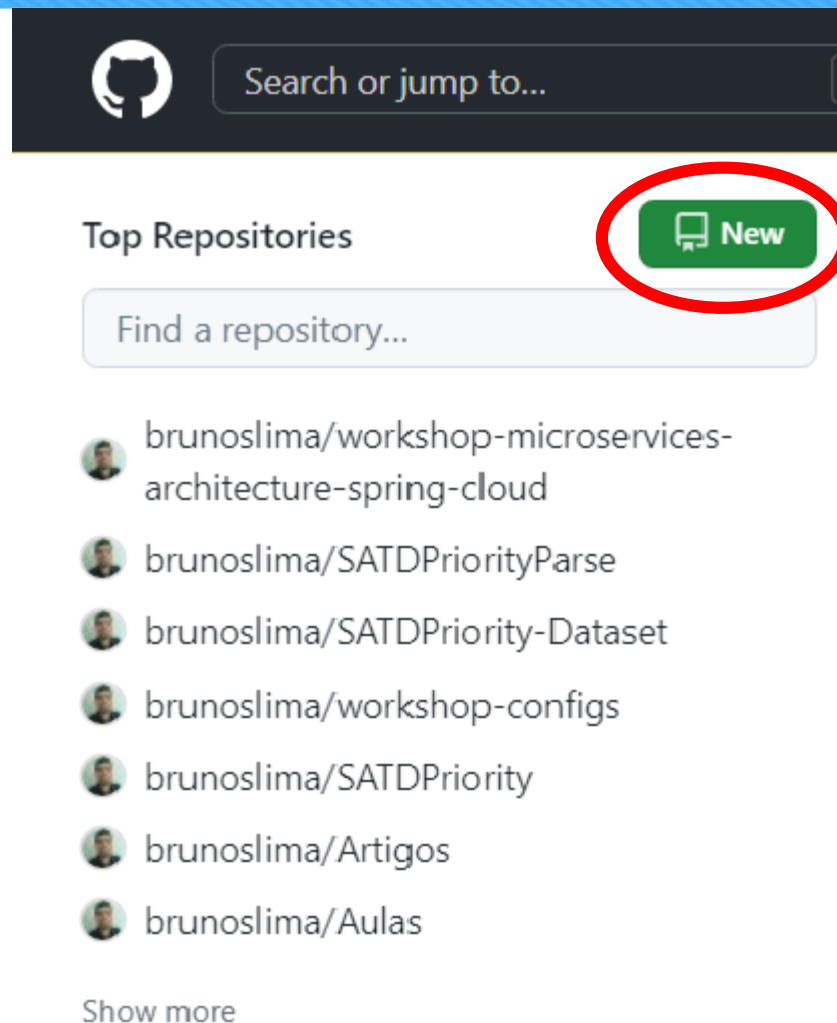
No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Git: Criar um repositório de Software (Remoto)



Git: Criar um repositório de Software (Remoto)



Git: Criar um repositório de Software (Remoto)

Create a new repository

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

Owner *



brunoslima ▼

Repository name *



repositorio-aula



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

Description (optional)

Repositório Exemplo para aula de Git



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.

Git: Criar um repositório de Software (Remoto)

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 ▼

This will set  `main` as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

Git: Criar um repositório de Software (Remoto)

The screenshot shows the GitHub interface for the repository 'brunoslima/repositorio-aula'. The repository is public and has 1 branch and 0 tags. The 'Code' button is circled in red. A dropdown menu is open, showing options to clone the repository. The 'Clone' button is also circled in red. A red arrow points to the HTTPS URL: 'https://github.com/brunoslima/repositorio-aula'. The repository name 'repositorio-aula' is displayed in large text, and the description 'Repositório Exemplo para aula de Git' is shown below it.

brunoslima / repositorio-aula Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

brunoslima Initial commit

README.md Initial commit

README.md

repositorio-aula

Repositório Exemplo para aula de Git

Local Codespaces New

Clone ?

HTTPS SSH GitHub CLI

https://github.com/brunoslima/repositorio-aula

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

Repositório Exemplo

Readme

0 stars

1 watching

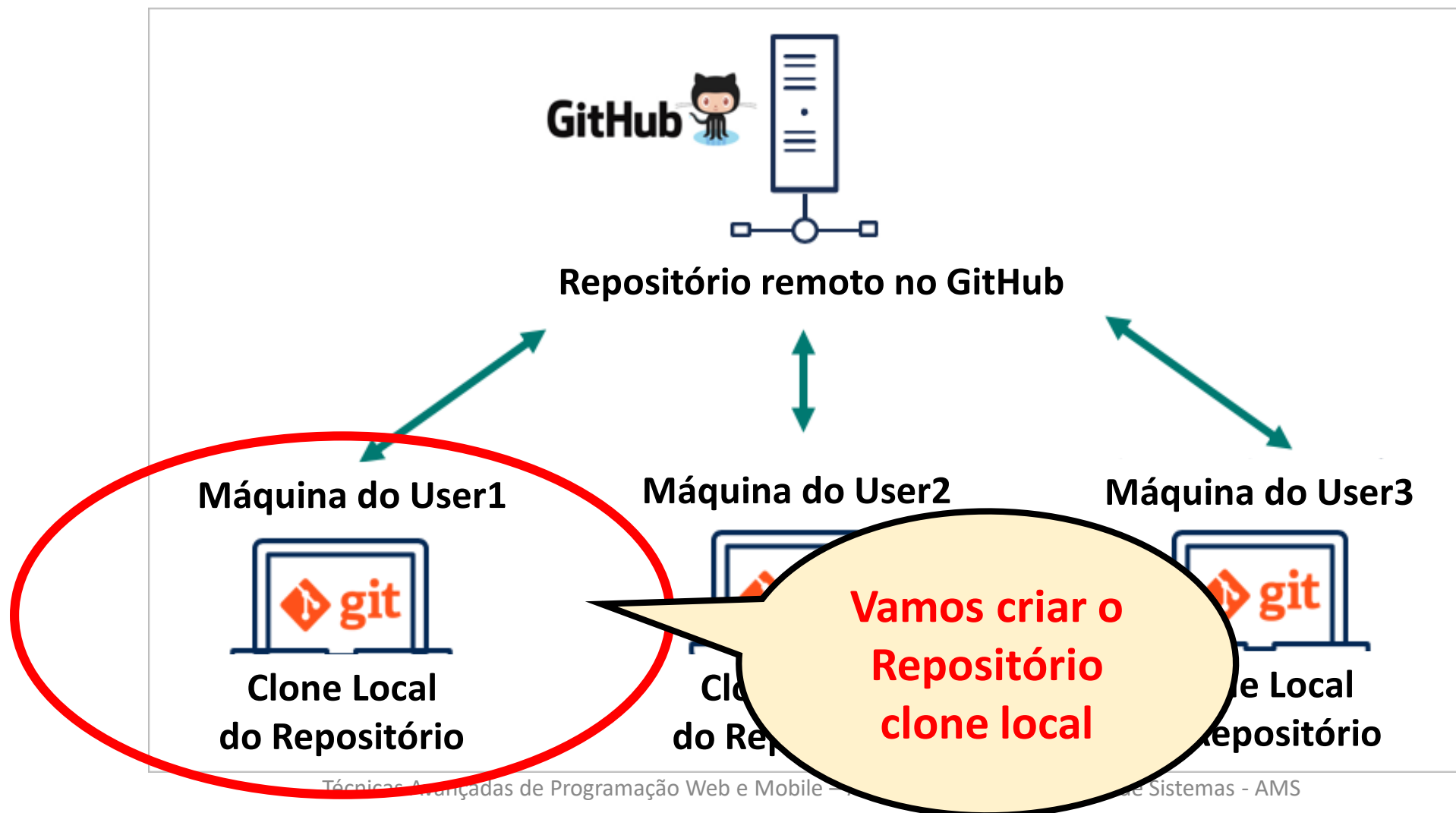
0 forks

Releases

No releases published

Create a new release

Git: Criar um repositório de Software (Remoto)




Git: Criar um repositório de Software (Remoto)

- **Crie uma pasta em sua área de trabalho:**
 - ✓ Aula-Git
 - ✓ Acessar essa pasta pelo prompt de comando e utilize o comando git clone.
- ***git clone url***
 - ✓ Cria um clone do repositório remoto e armazena esse clone como um repositório local na máquina do usuário.

Git: Criar um repositório de Software (Remoto)

- ***git clone url***

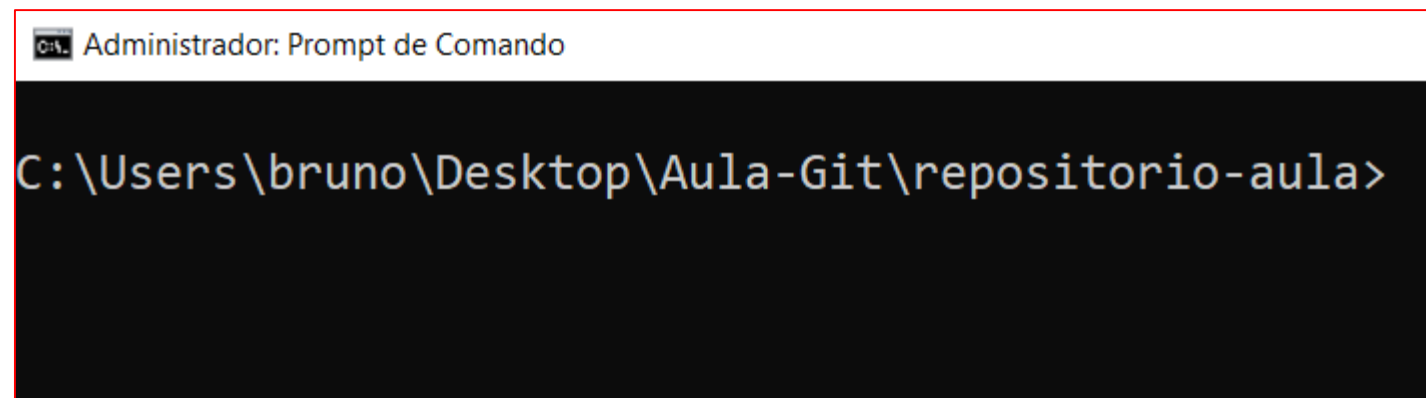
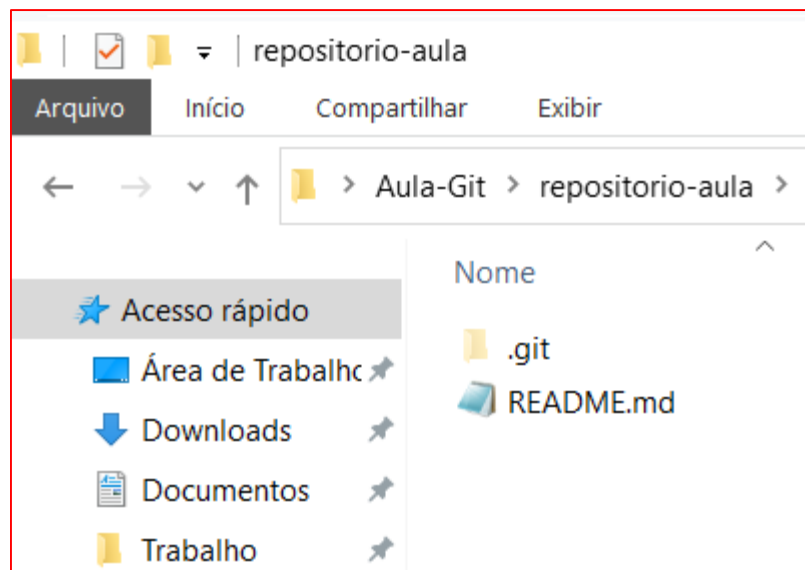
- ✓ Cria um clone do repositório remoto e armazena esse clone como um repositório local na máquina do usuário.

 Administrador: Prompt de Comando

```
C:\Users\bruno\Desktop\Aula-Git>git clone https://github.com/brunoslima/repositorio-aula.git
Cloning into 'repositorio-aula'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

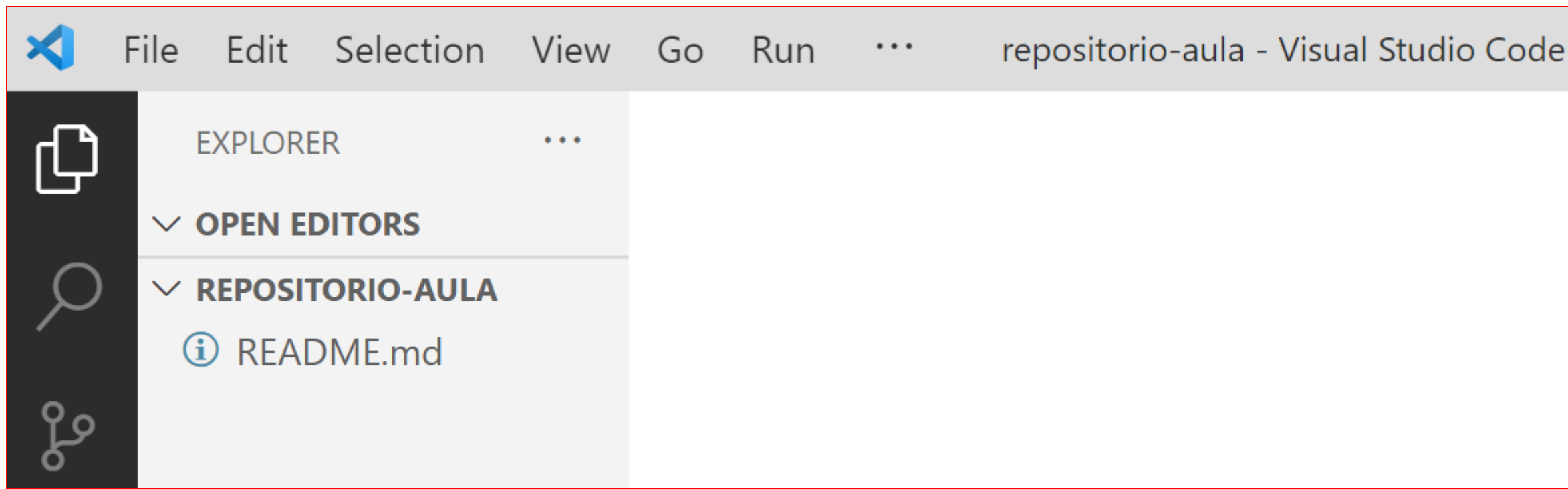
Git: Criar um repositório de Software (Remoto)

- ***git clone url***
 - ✓ O repositório foi clonado e pode ser utilizado localmente



Git: Criar um repositório de Software

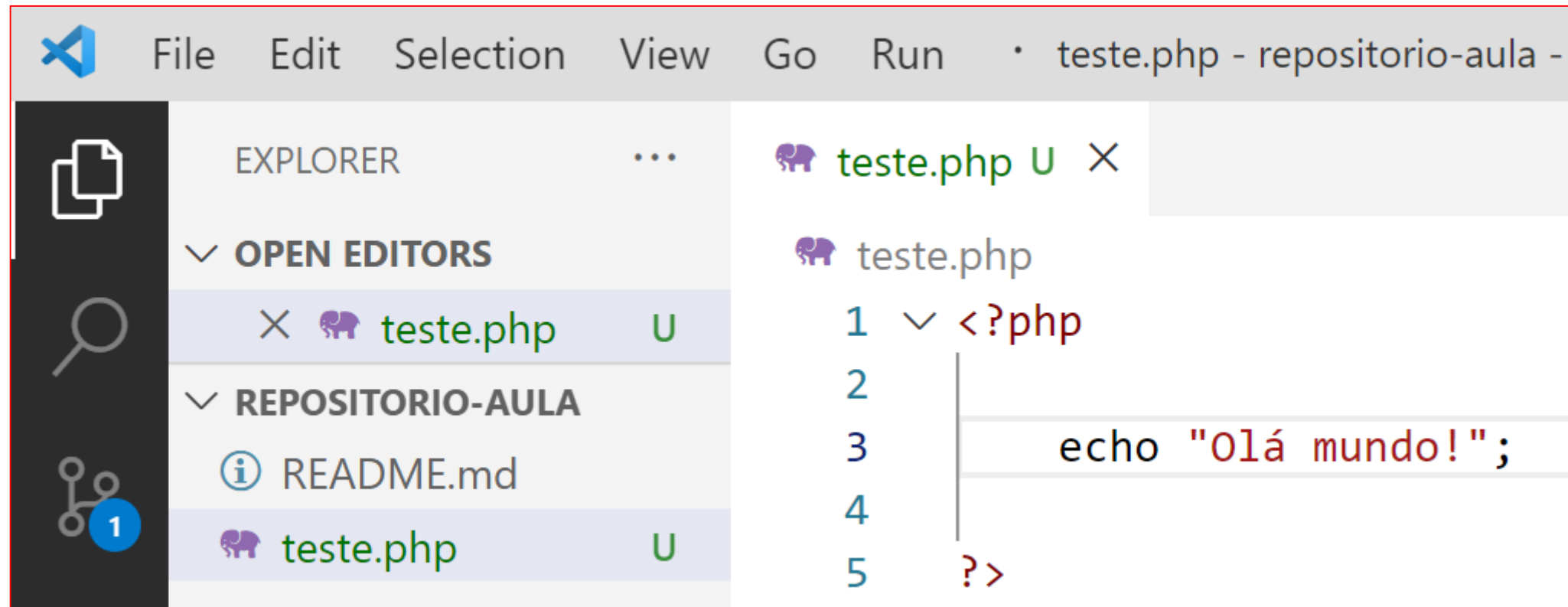
- Abrir nossa pasta do Repositório Local no VS Code



Git: Criar um repositório de Software

- **Abrir nossa pasta do Repositório Local no VS Code**

✓ Podemos criar nossos arquivos normalmente, como por exemplo teste.php



Git: Criar um repositório de Software

- Ao realizar um `git status`, podemos ver nosso arquivo `teste.php`
 - ✓ Entretanto o arquivo *teste.php* não foi adicionado ao repositório

```
Administrator: Prompt de Comando

C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

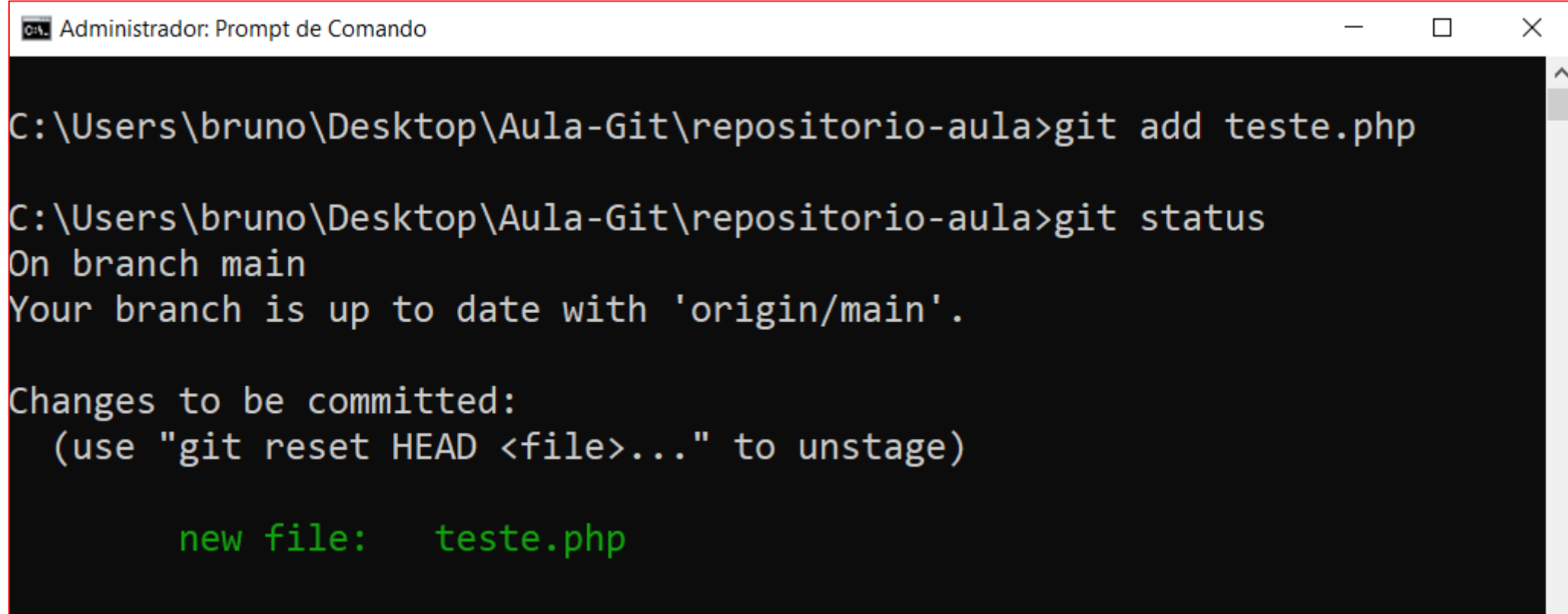
    teste.php

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

Git: Criar um repositório de Software

- ***git add nomeArquivo***

✓ Incluir alterações de um ou vários arquivos para uma futura versão estável.



```
Administrador: Prompt de Comando

C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git add teste.php

C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git status
On branch main
Your branch is up to date with 'origin/main'.

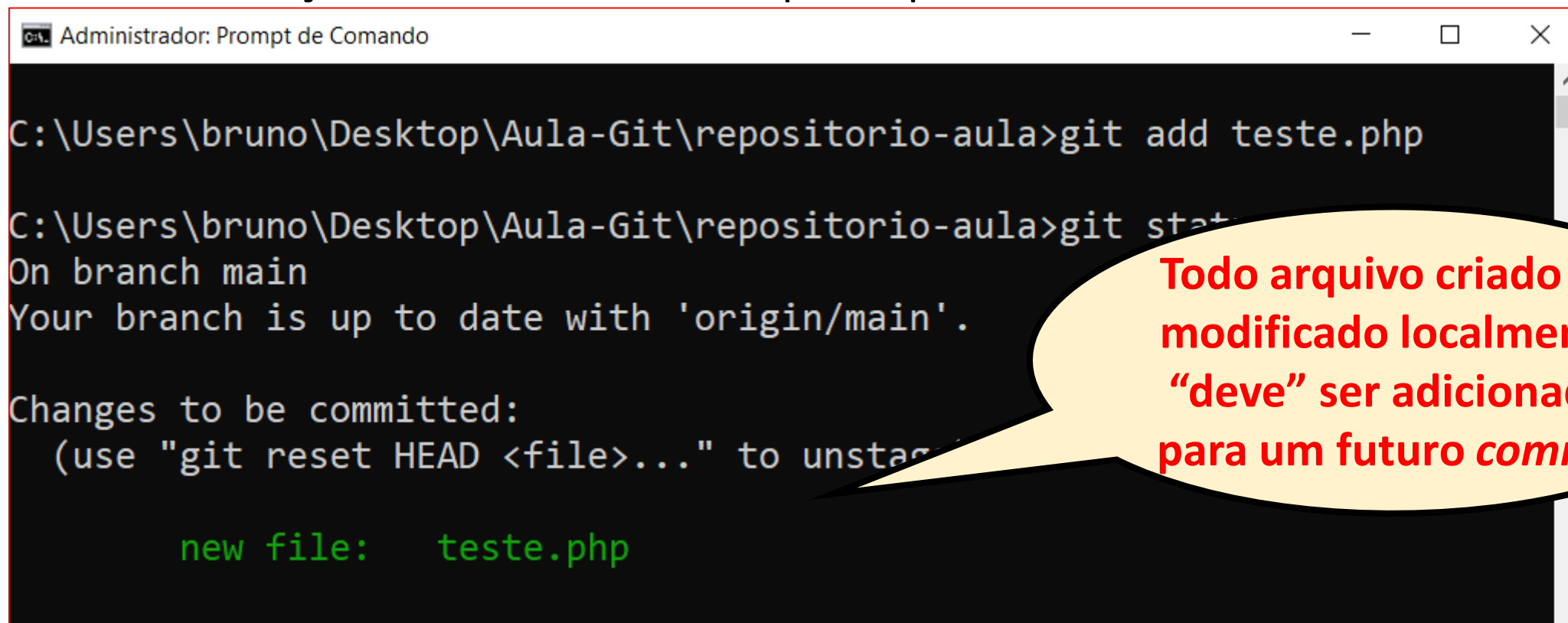
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   teste.php
```

Git: Criar um repositório de Software

- ***git add nomeArquivo***

✓ Incluir alterações de um ou vários arquivos para uma futura versão estável.



```
Administrador: Prompt de Comando

C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git add teste.php

C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

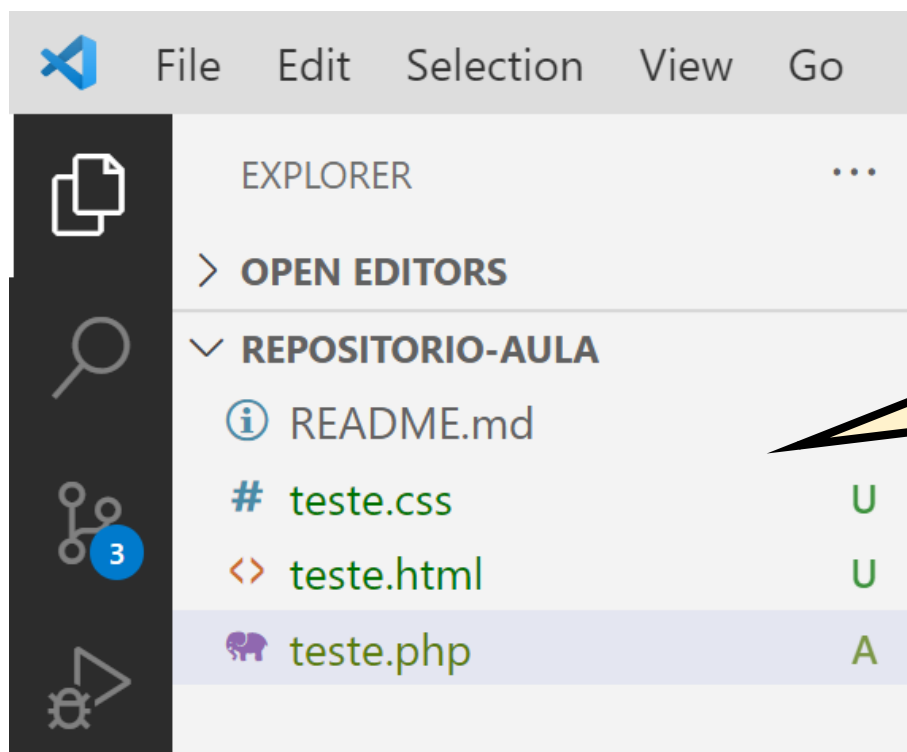
    new file:   teste.php
```

Todo arquivo criado ou modificado localmente “deve” ser adicionado para um futuro *commit*

Git: Criar um repositório de Software

- ***git add --all***

✓ Incluir alterações de **vários arquivos** para uma futura versão estável.

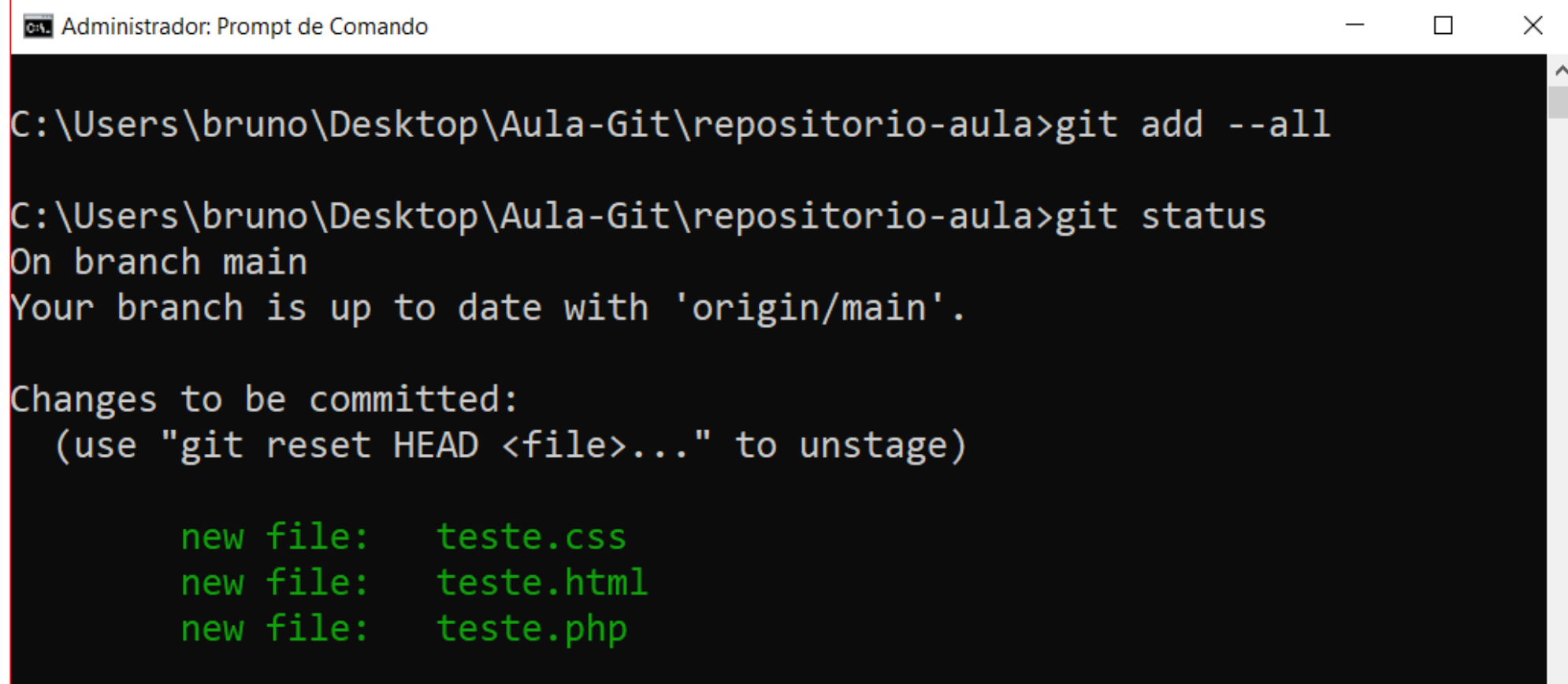


**Crie mais dois arquivos,
o *git add --all* adiciona
todos eles**

Git: Criar um repositório de Software

- ***git add --all***

✓ Incluir alterações de **vários arquivos** para uma futura versão estável.



```
Administrador: Prompt de Comando

C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git add --all

C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git status
On branch main
Your branch is up to date with 'origin/main'.

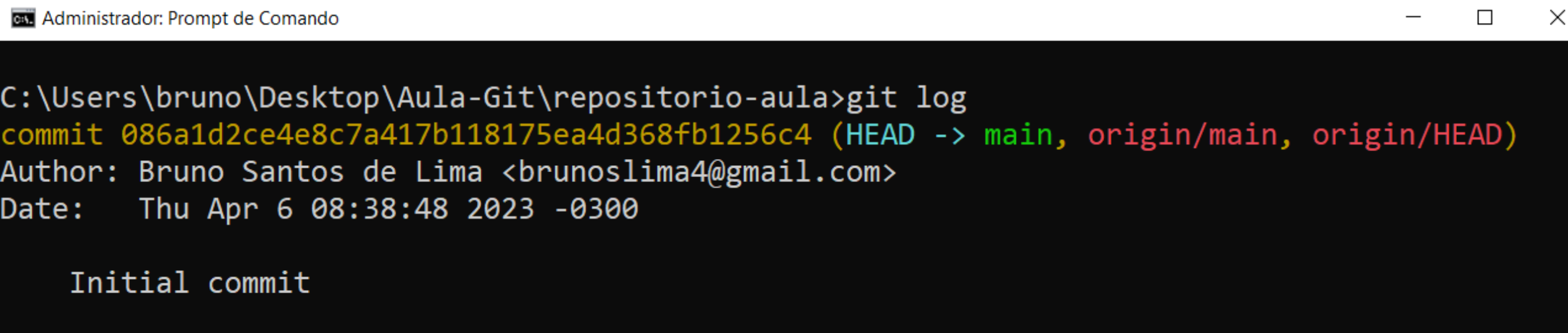
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   teste.css
        new file:   teste.html
        new file:   teste.php
```

Git: Criar um repositório de Software

- ***git log***

- ✓ Permite ver o histórico de *commits* realizados.
- ✓ Informações do autor do *commit*, *hash*, data e hora de efetivação também são visíveis.



```
C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git log
commit 086a1d2ce4e8c7a417b118175ea4d368fb1256c4 (HEAD -> main, origin/main, origin/HEAD)
Author: Bruno Santos de Lima <brunoslima4@gmail.com>
Date: Thu Apr 6 08:38:48 2023 -0300

Initial commit
```

Git: Criar um repositório de Software

- ***git commit -m “mensagem do commit”***
 - ✓ Torna permanente um conjunto de alterações
 - ✓ Neste caso torna permanente a adição dos três arquivos de teste criados no repositório.

Administrador: Prompt de Comando

```
C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git commit -m "Inserção de arquivos teste"
[main 4d5d8c0] Inserção de arquivos teste
3 files changed, 18 insertions(+)
create mode 100644 teste.css
create mode 100644 teste.html
create mode 100644 teste.php
```

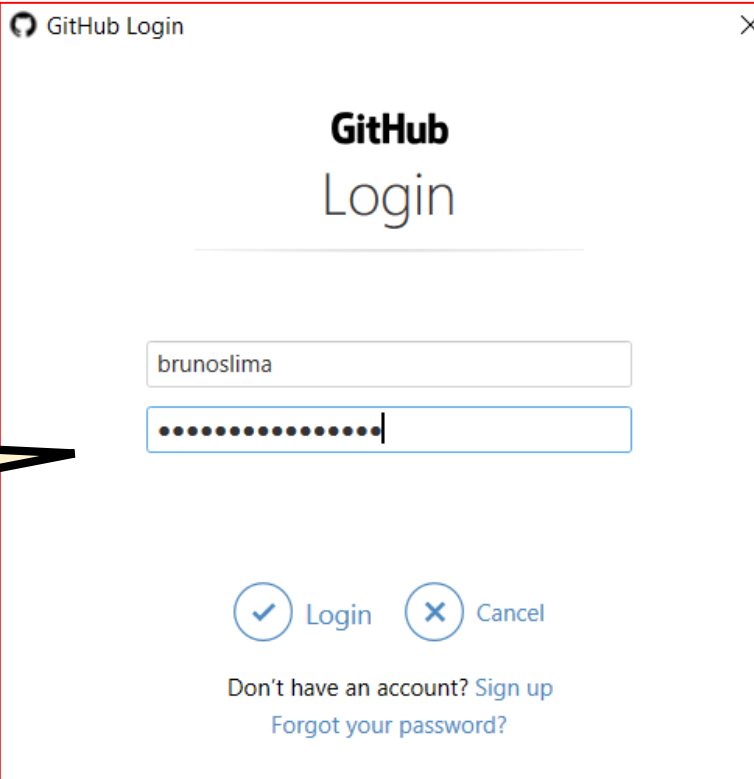
Git: Criar um repositório de Software

- ***git push***

✓ Envia os commits para o Repositório Remoto e todos podem acessar os mesmos.

```
C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git push
```

**O push faz necessário
autenticação no
Repositório Remoto**



The screenshot shows a 'GitHub Login' dialog box. It has a title bar with the GitHub logo and the text 'GitHub Login'. The main content area displays 'GitHub Login' in a large font. Below this, there are two input fields: the first contains the username 'brunoslima', and the second contains a masked password represented by dots. At the bottom of the dialog, there are two buttons: 'Login' with a checkmark icon and 'Cancel' with an 'X' icon. Below these buttons, there is a link that says 'Don't have an account? Sign up' and another link that says 'Forgot your password?'.

Git: Criar um repositório de Software

- ***git push***

- ✓ Envia os commits para o Repositório Remoto e todos podem acessar os mesmos.

```
C:\Users\bruno\Desktop\Aula-Git\repositorio-aula>git push
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': brunoslima
Password for 'https://brunoslima@github.com':
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 520 bytes | 260.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/brunoslima/repositorio-aula.git
   086a1d2..4d5d8c0  main -> main
```

- CASTRO, E. e HYLOP, B. **HTML e CSS3: Guia prático e visual**. Rio de Janeiro: Alta Books, 2013.
- ROBSON, E. e FREEMAN, E. **Use a cabeça! HTML e CSS**. Rio de Janeiro: Alta Books, 2015.
- SILVA, M.S. **JavaScript - Guia do Programador: Guia Completo das Funcionalidades de Linguagem JavaScript**. 1. ed. São Paulo: Novatec Editora, 2010.
- SOARES, W. **PHP 5: Conceitos, Programação e Integração com Banco de Dados**. São Paulo: Editora Érica, 2013.
- W3SCHOOLS. **W3 Schools**. Disponível em: <https://www.w3schools.com>

Obrigado!

