

Git e Github

Adaptelas

Thiago Figueiredo - Assessor de
Desenvolvimento



20

24

Introdução

- Thiago de Figueiredo Souza
- Ciência da Computação - UFES
- Assessor de Desenvolvimento - Adapti

Git e GitHub

- O que são
- Como usar





Git

Git é um sistema de controle de versão distribuído, ou seja, ele permite que várias pessoas trabalhem no mesmo projeto e ao mesmo tempo, sem interferir no trabalho uns dos outros.



Sistema de controle de versão



trabalho

Sistema de controle de versão



trabalho



trabalho2

Sistema de controle de versão



trabalho



trabalho2



trabalho_final

Sistema de controle de versão



trabalho



trabalho2



trabalho_final



trabalho_final_final

Sistema de controle de versão



trabalho



trabalho2



trabalho_final



trabalho_final_final



trabalho_agora_vai

Sistema de controle de versão



trabalho



trabalho2



trabalho_final



trabalho_final_final



trabalho_agora_vai

???

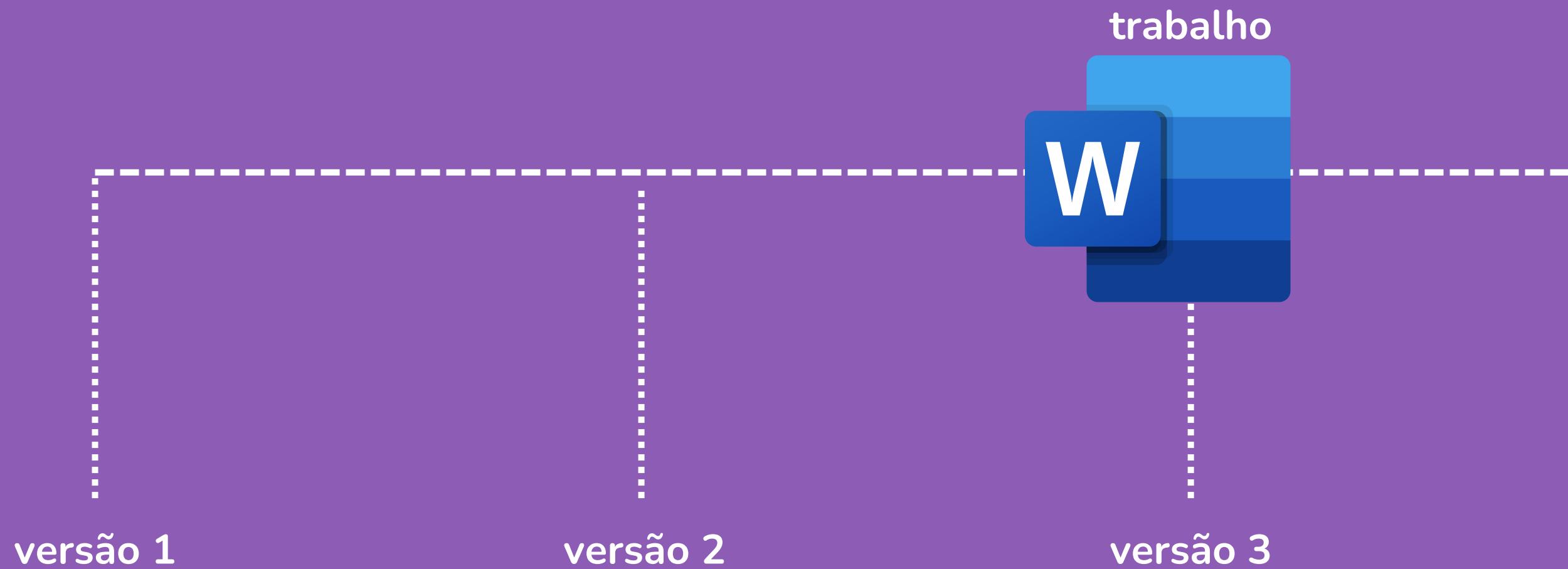
Sistema de controle de versão



Sistema de controle de versão



Sistema de controle de versão



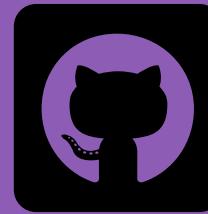


Local



Remote



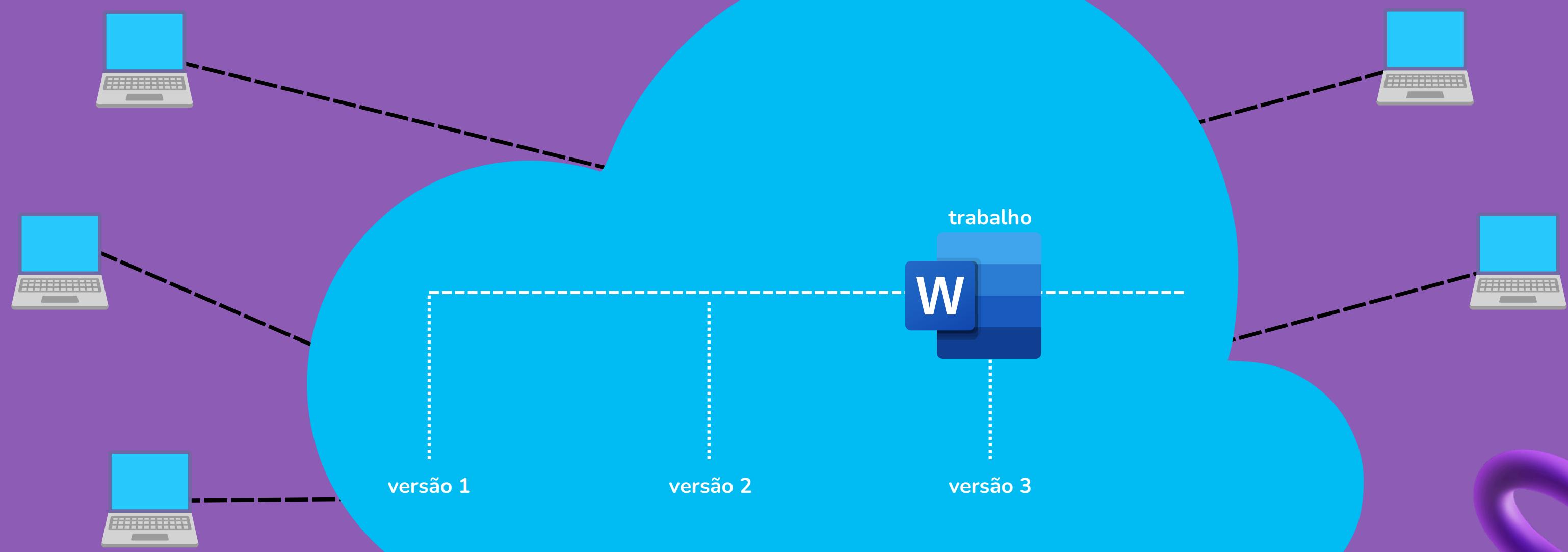


GitHub

**GitHub é uma plataforma de
hospedagem de arquivos com
controle de versão usando o Git**



Acesso remoto



Resumo

Git é a ferramenta que usamos para controlar versões do nosso arquivo, e GitHub é o serviço que usamos para hospedar e colaborar em projetos utilizando o Git



Commit

Commit é o processo de tornar permanente um conjunto de alterações, ou seja, de efetivar as alterações.



Commit



Commit

segunda_guerra_mundial



Commit

segunda_guerra_mundial



commit



Commit

segunda_guerra_mundial



commit



Commit

- adicionar arquivos

Commit

- adicionar arquivos

```
git add arquivo.docx
```

Commit

- adicionar arquivos

```
git add arquivo.docx
```

```
git add .
```

Commit

- adicionar arquivos

```
git add arquivo.docx
```

```
git add .
```

- realizar commit

Commit

- adicionar arquivos

```
git add arquivo.docx
```

```
git add .
```

- realizar commit

```
git commit -m "mensagem"
```

Commit

- adicionar arquivos

```
git add arquivo.docx
```

```
git add .
```

- realizar commit

```
git commit -m "mensagem"
```

- enviar arquivos

Commit

- adicionar arquivos

```
git add arquivo.docx
```

```
git add .
```

- realizar commit

```
git commit -m "mensagem"
```

- enviar arquivos

```
git push
```

Commit

segunda_guerra_mundial



Commit

segunda_guerra_mundial



```
git add .
```

Commit

segunda_guerra_mundial



```
git add .
```

```
git commit -m "adicionando introducao"
```

Commit

segunda_guerra_mundial



git add .

```
git commit -m "adicionando introducao"
```

git push

Branches



Branches

Maria



Introdução

Branches

Maria



Introdução

Antônia



Desenvolvimento

Branches

Maria



Introdução

Antônia



Desenvolvimento

Sebastiana



Conclusão

Branches



Maria

Introdução



Antônia

Desenvolvimento



Sebastiana

Conclusão

segunda_guerra_mundial



Branches

Maria



Introdução

Antônia



Desenvolvimento

Sebastiana



Conclusão

segunda_guerra_mundial



Branches

Maria



Introdução

Antônia



Desenvolvimento

Sebastiana

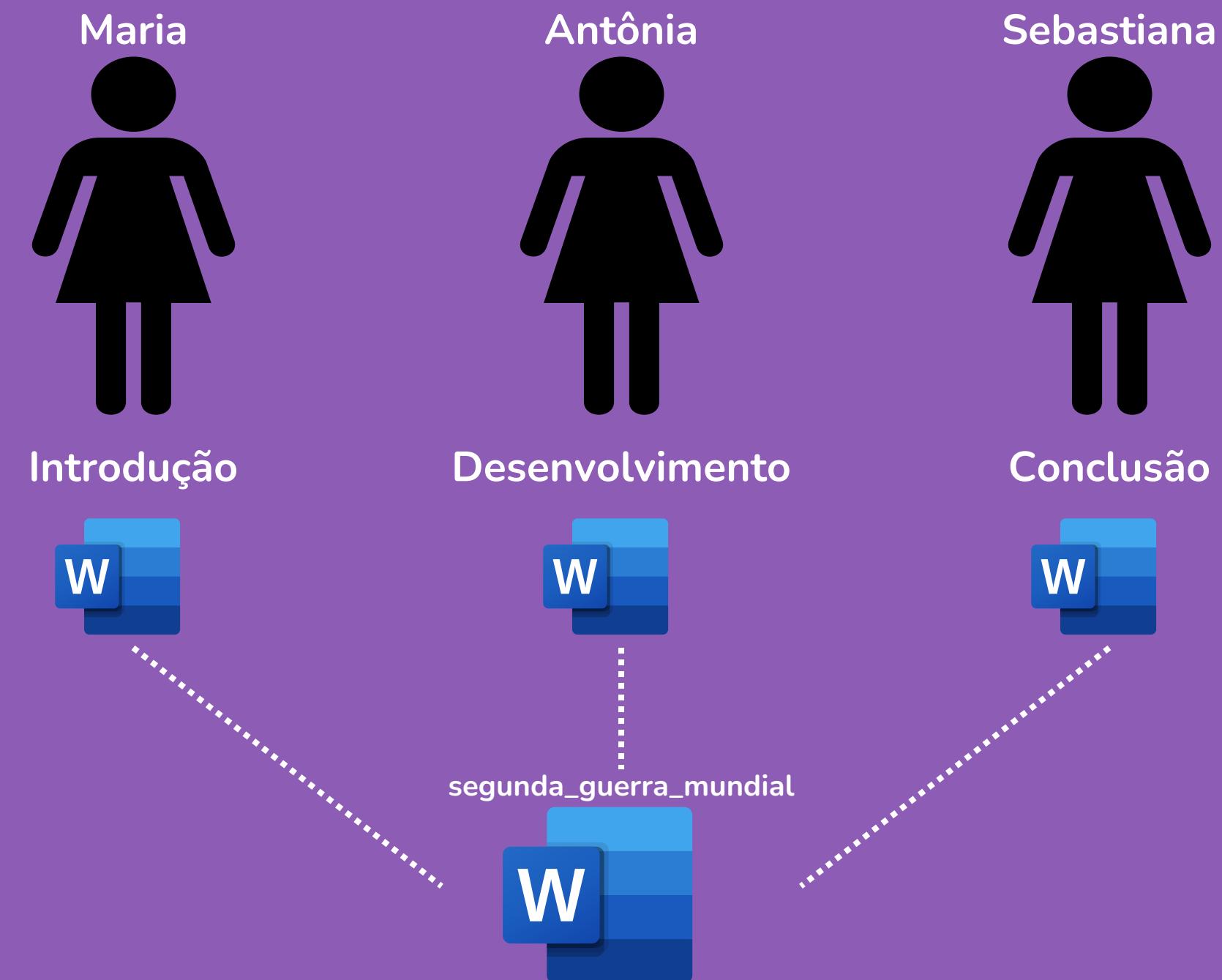


Conclusão

segunda_guerra_mundial



Branches

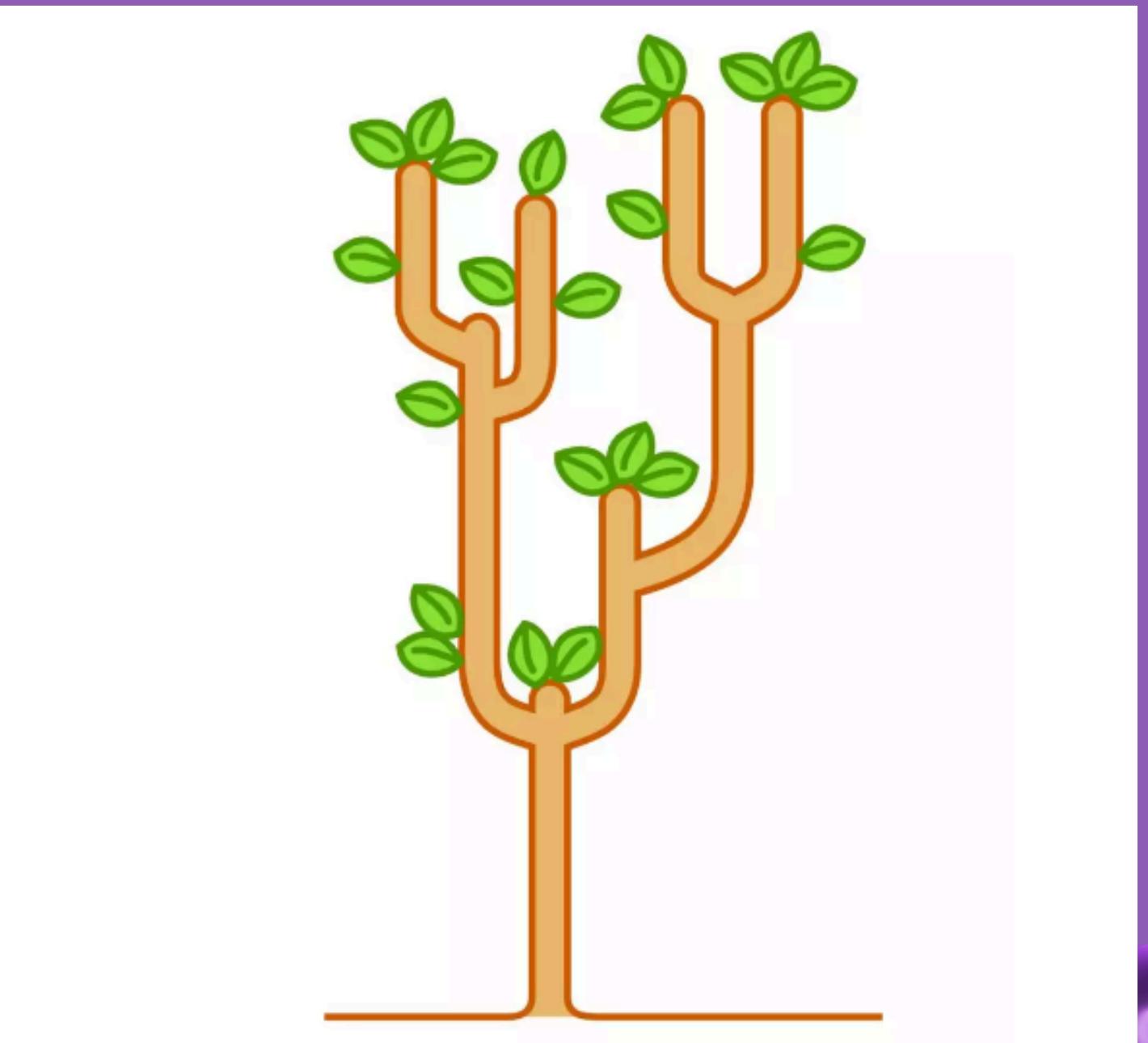
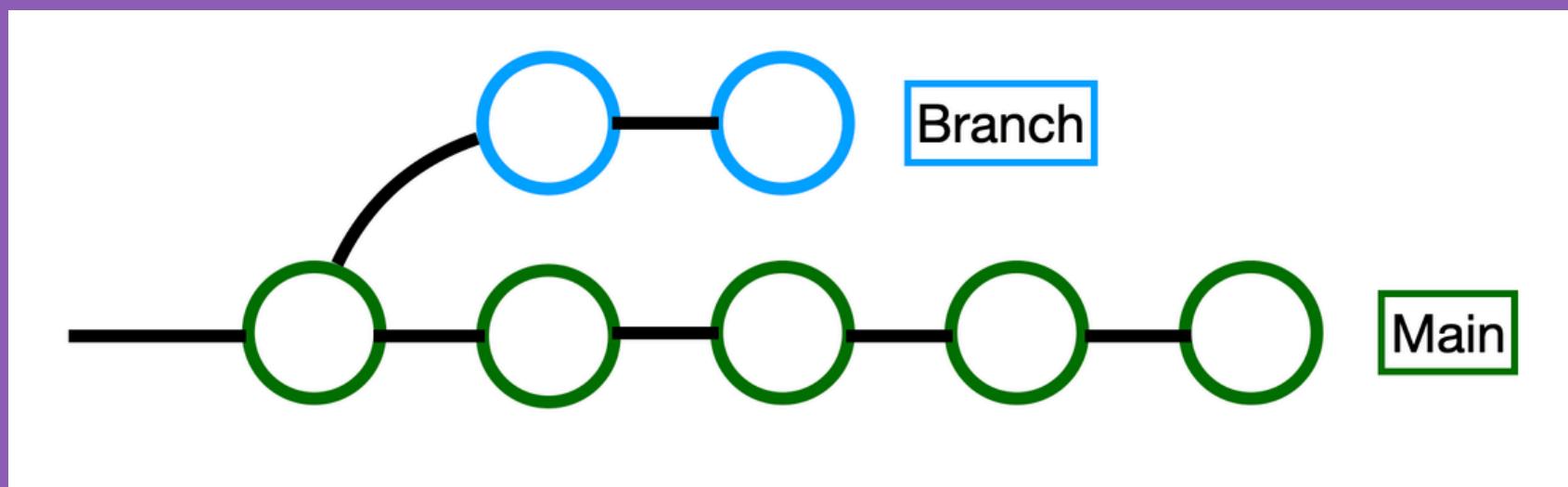


Branches

Branch é uma linha de desenvolvimento que permite trabalhar em várias versões de um repositório ao mesmo tempo. Cada branch pode ser considerada uma cópia independente do código



Branches



Branches



Introdução

Maria

Commit



Desenvolvimento

Antônia

Branches



Maria

Introdução

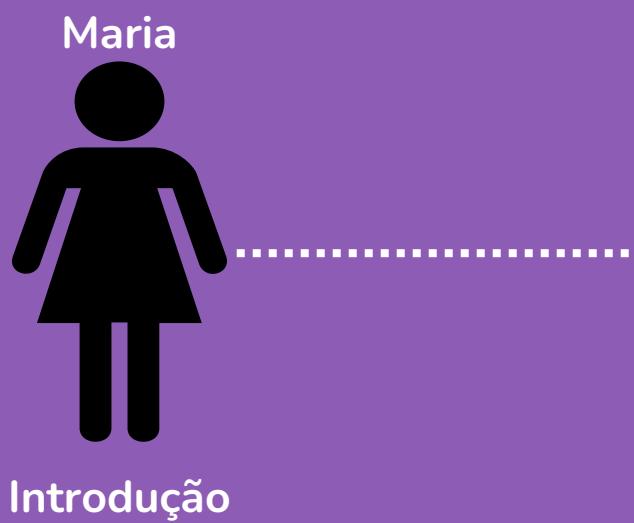


Antônia

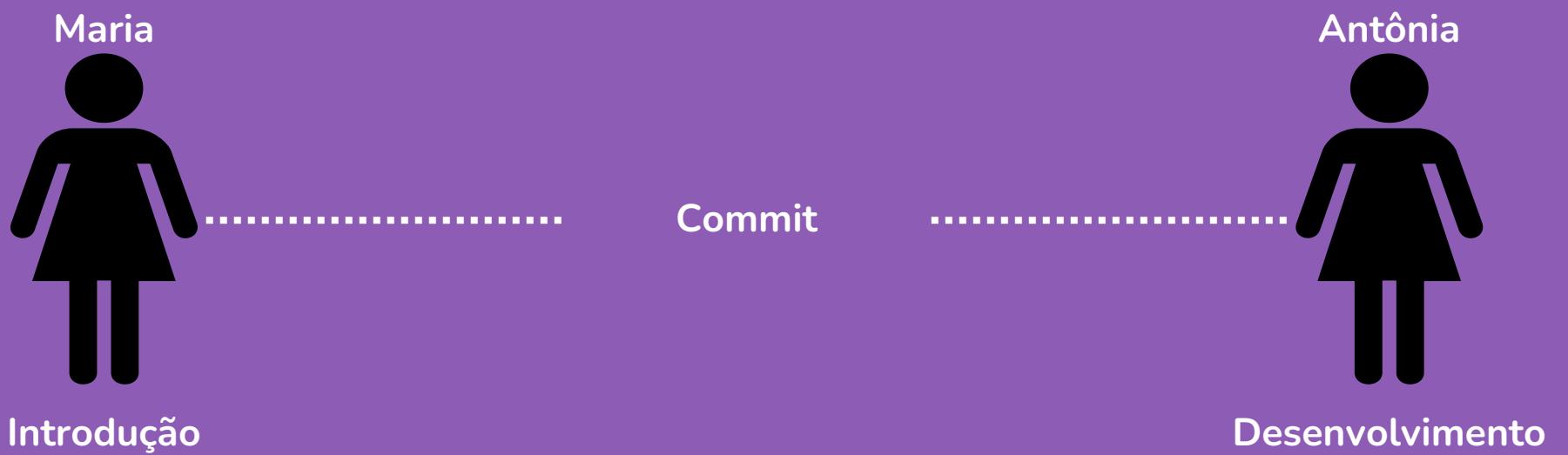
Desenvolvimento

Commit

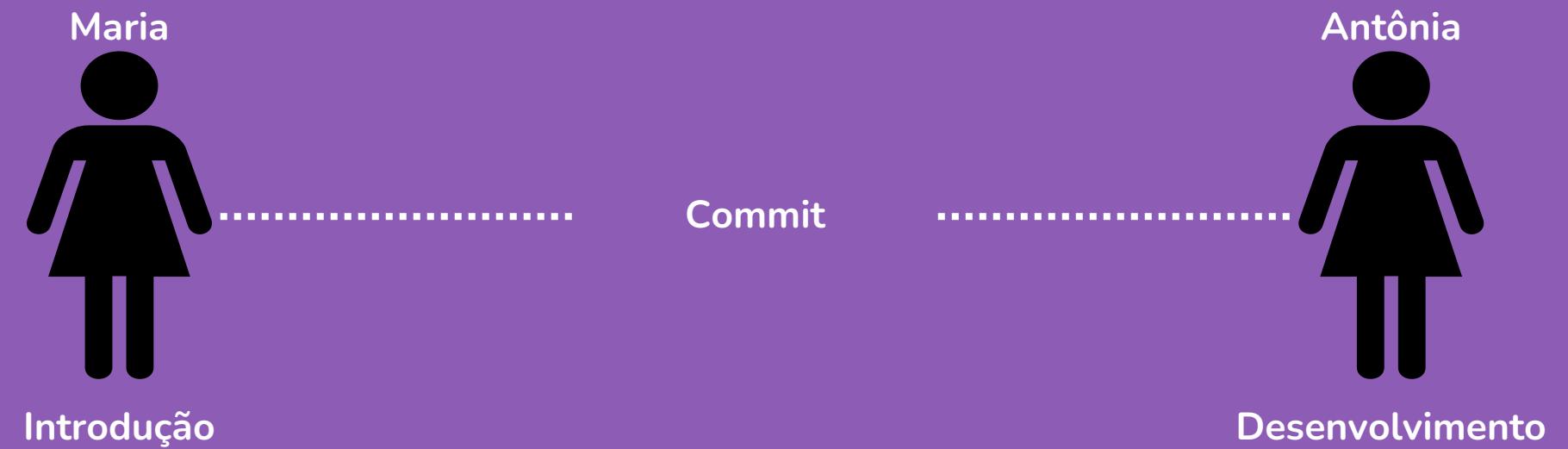
Branches



Branches



Branches



Branches



Introdução

Maria

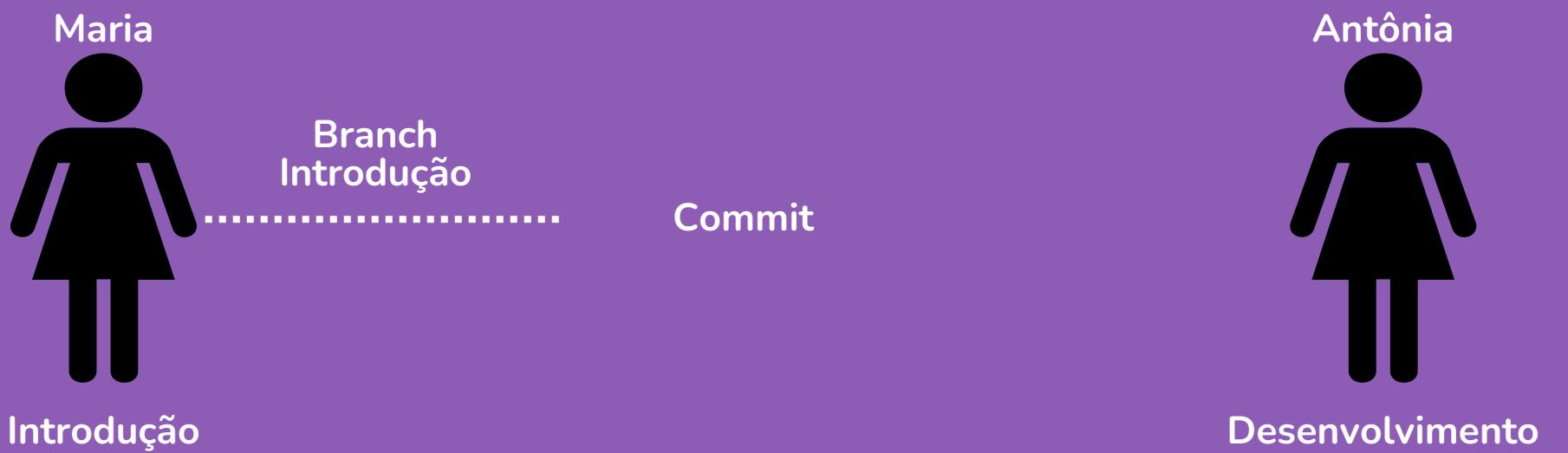
Commit

Antônia

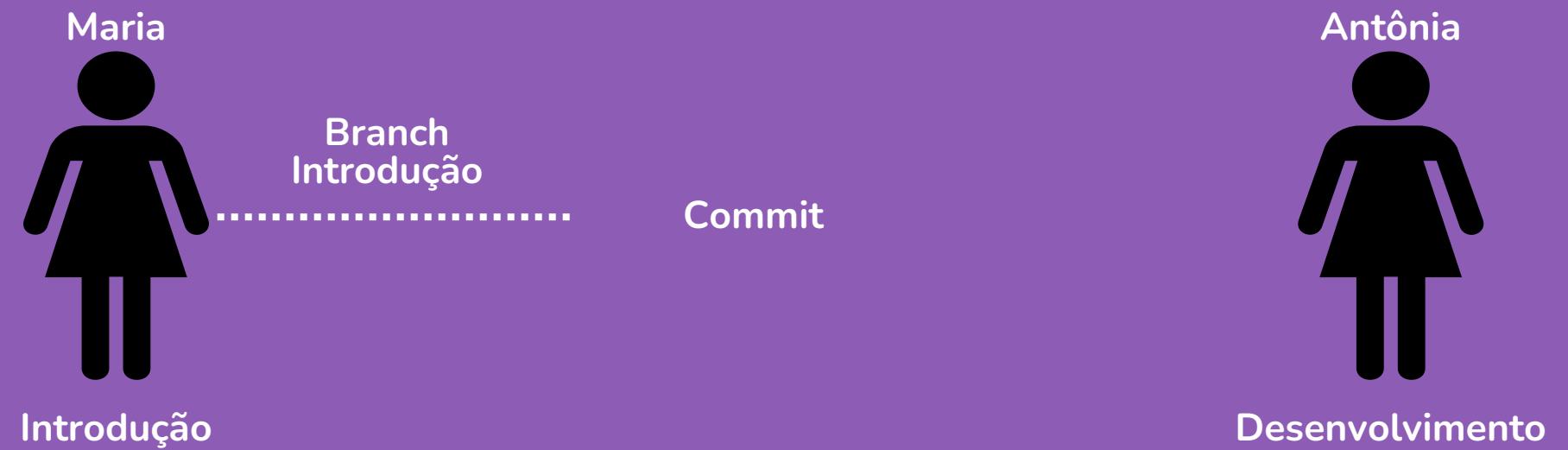


Desenvolvimento

Branches



Branches



Branches



Branches



Branches

- comandos

```
git branch
```

verifica branch atual

```
git branch nome-branch
```

cria uma nova branch

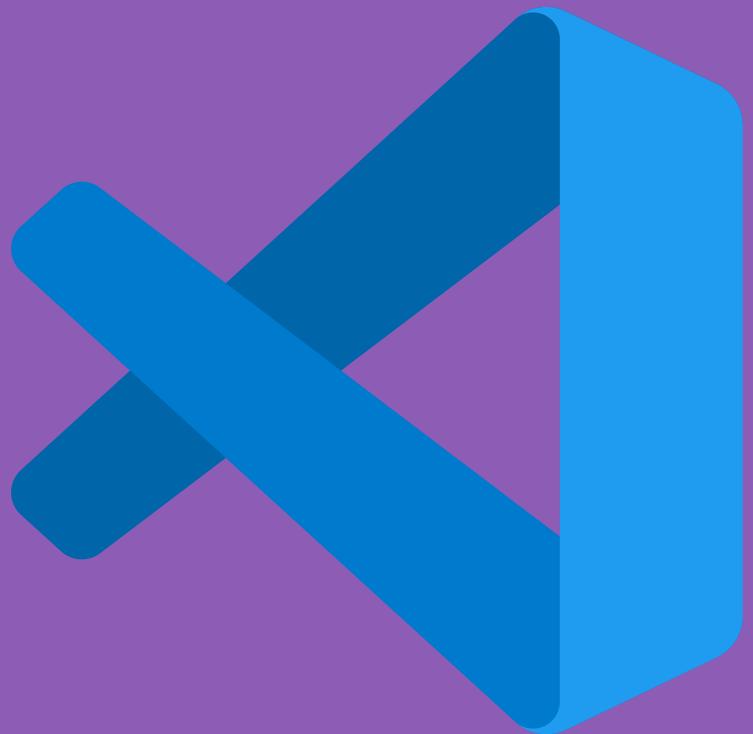
```
git checkout nome-branch
```

muda de branch

```
git checkout -b nome-branch
```

cria uma nova branch e já muda pra ela

Editor de Código



clique na imagem para instalar

Git



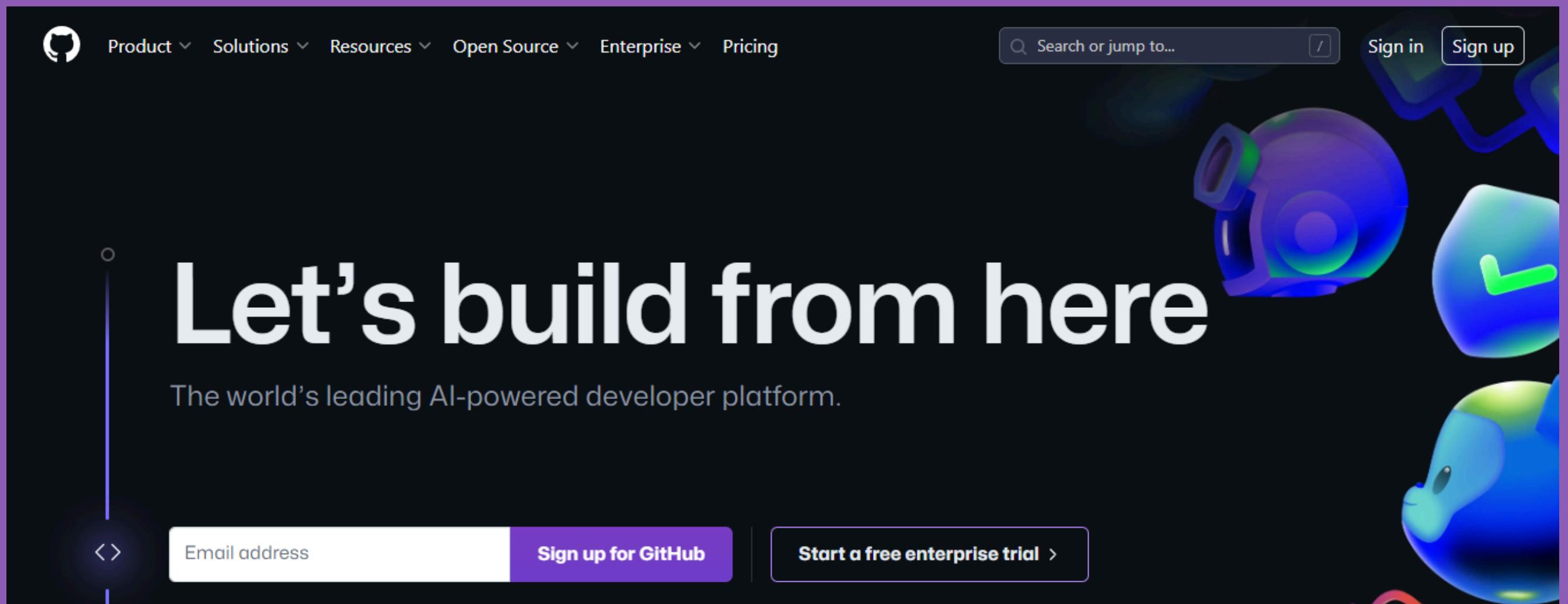
clique na imagem para instalar

GitHub

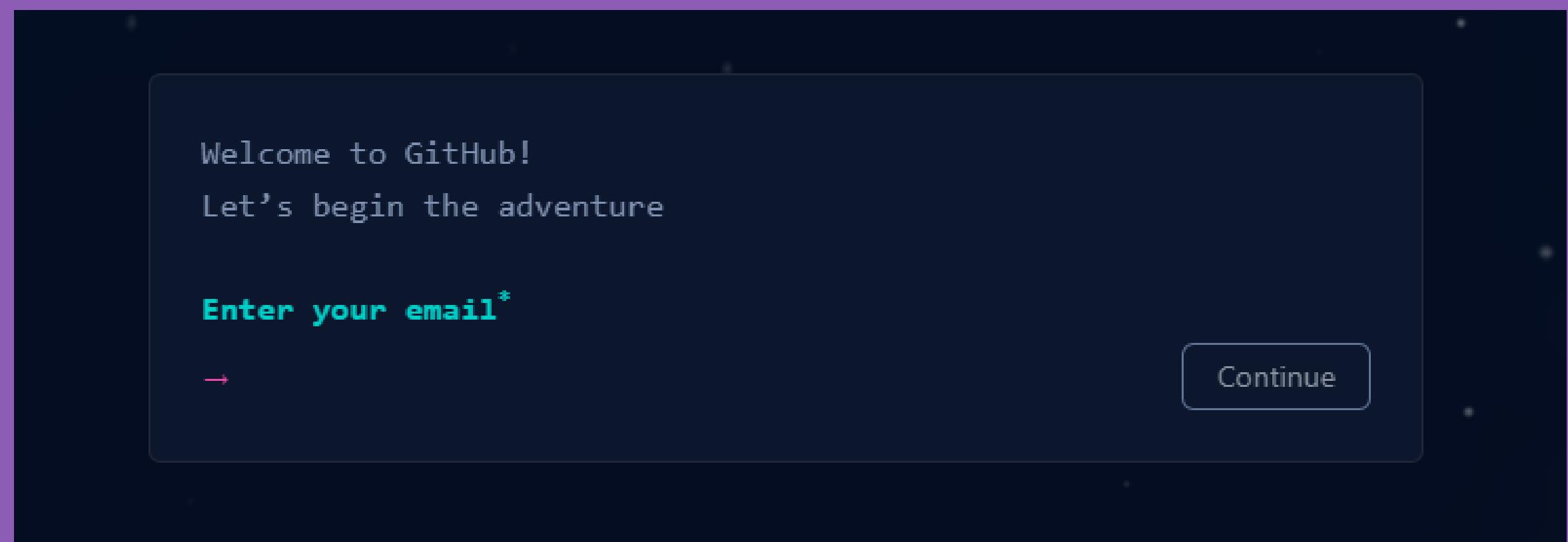


clique na imagem para ser direcionado

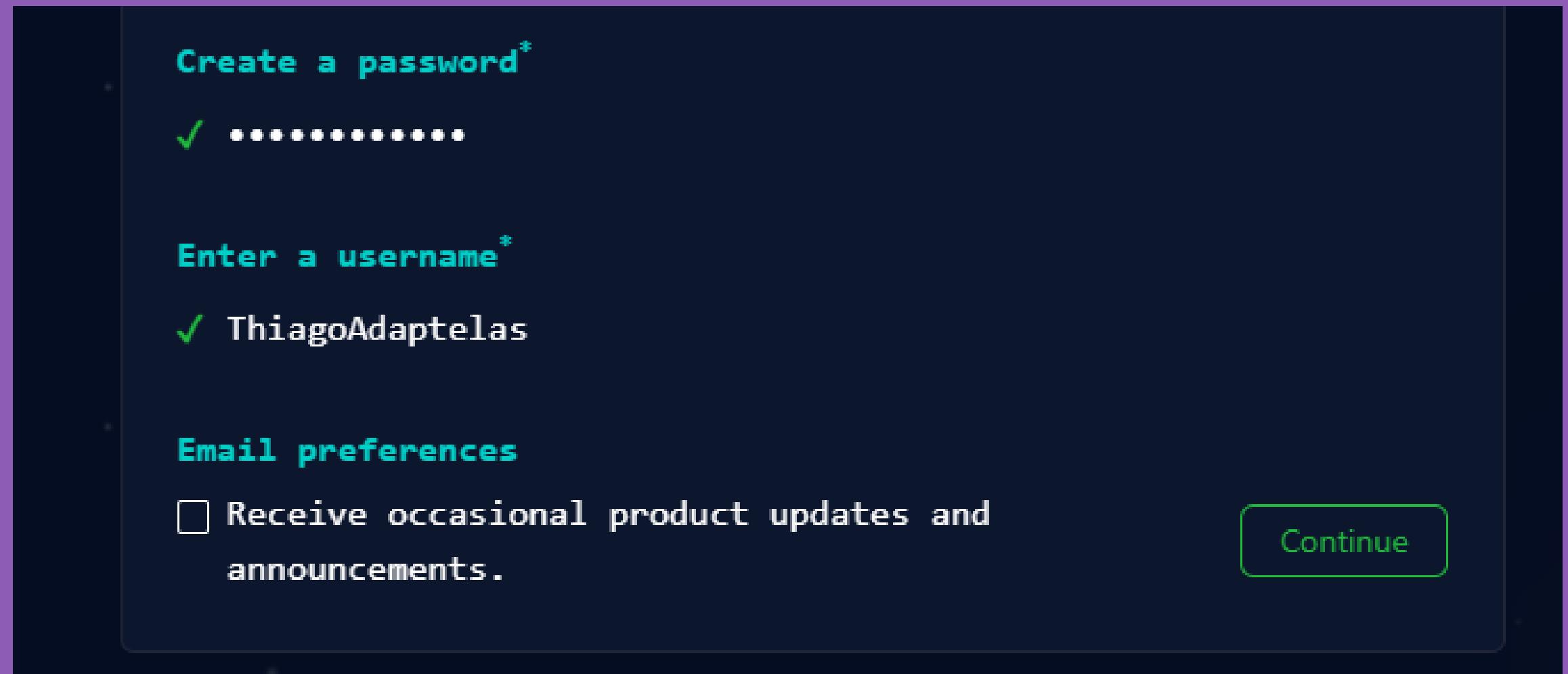
GitHub



GitHub



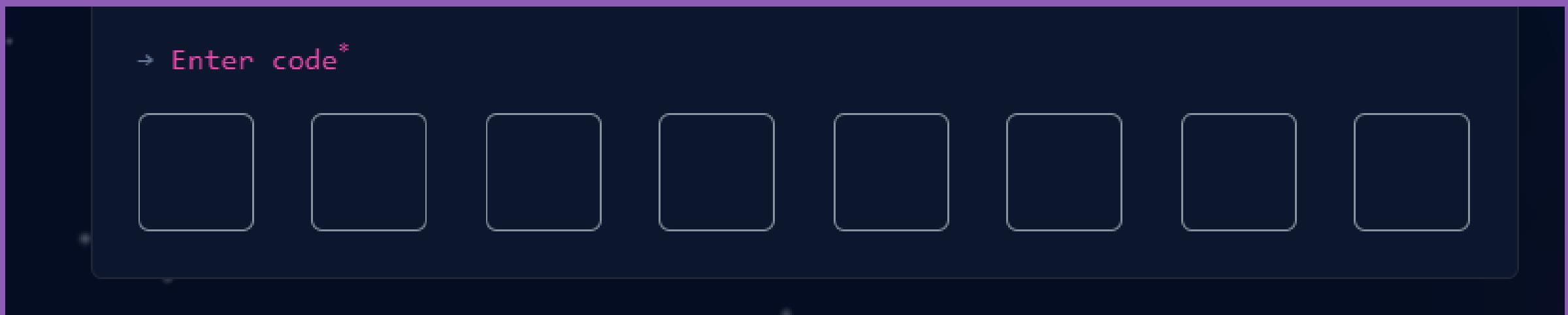
GitHub



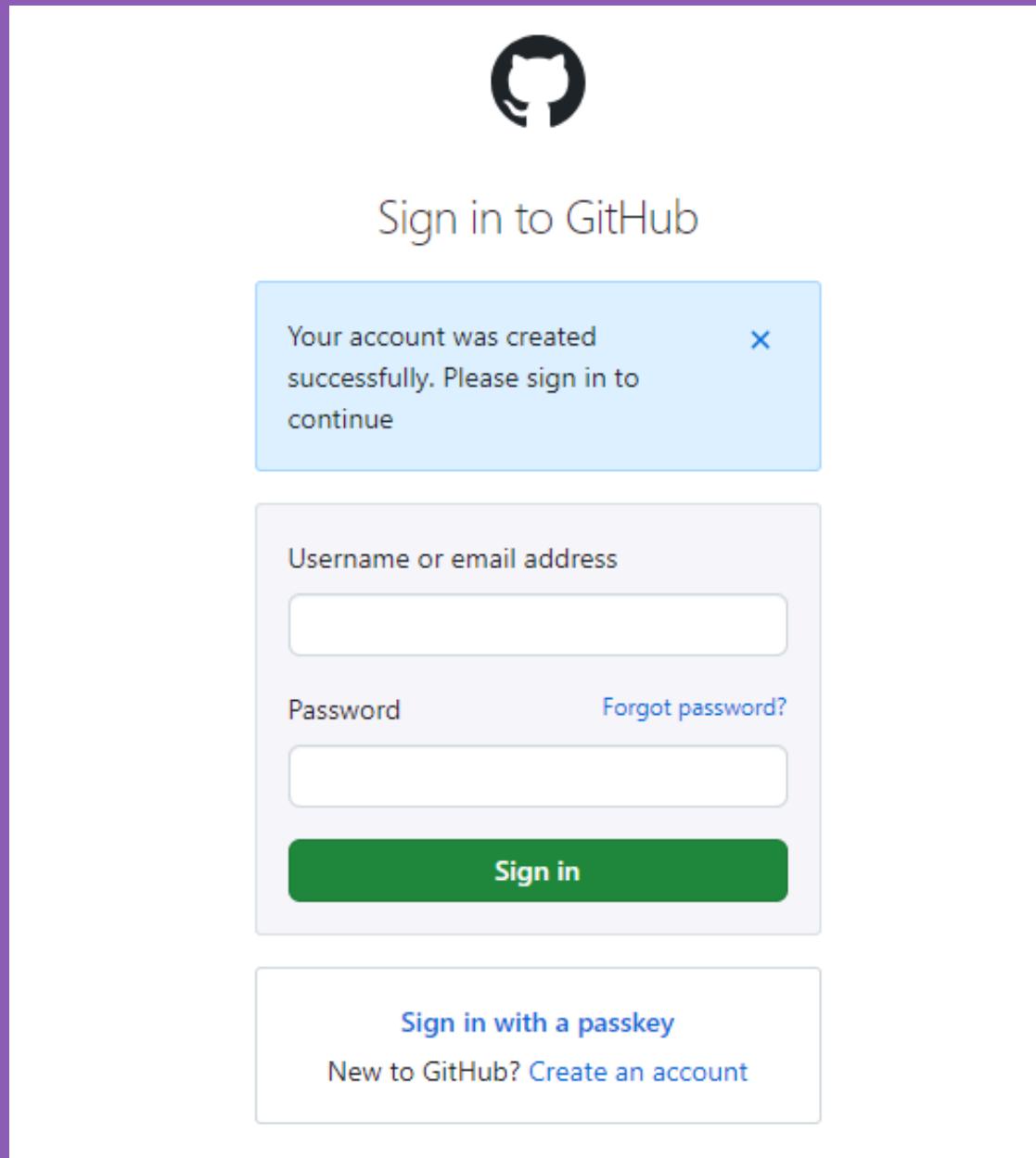
The image shows a screenshot of the GitHub account creation process. It features a dark-themed interface with white text and light gray input fields. The steps are as follows:

- Create a password***: A password has been entered, indicated by a green checkmark and a series of eight dots.
- Enter a username***: A username has been entered, indicated by a green checkmark and the text "ThiagoAdaptelas".
- Email preferences**: A checkbox labeled "Receive occasional product updates and announcements." is present, but it is not checked.
- Continue**: A green button labeled "Continue" is located at the bottom right of the form.

GitHub



GitHub



GitHub

The image shows the GitHub welcome screen. On the left, there's a dark sidebar with two icons: a magnifying glass over code and an '@' symbol. The main area has a dark background with white text. It says "Welcome to GitHub" in large blue letters, followed by "We are glad you're here." in smaller white text.

Sidebar (Personalization Settings):

- A progress bar at the top.
- A text block: "This will help us guide you to the tools that are best suited for your projects."
- A question: "How many team members will be working with you?" with six radio button options:
 - Just me
 - 2-5
 - 5-10
 - 10-20
 - 20-50
 - 50+
- A question: "Are you a student or teacher?" with three radio button options:
 - N/A
 - Student
 - Teacher
- A large blue "Continue" button at the bottom.

[Skip personalization](#) [Ativar o Windows](#)

GitHub

The screenshot shows the GitHub Home page with the following elements:

- Dashboard:** Shows a "Create your first project" section with a "Create repository" button and an "Import repository" link. It also displays a "Recent activity" section with a note about providing links to actions across GitHub.
- Updates to your homepage feed:** A card explaining the combination of Following and For you feeds, improved filtering, and a new visual design. It includes a "Learn more" link.
- Start writing code:** A section with a "Start a new repository" form for ThiagoAdaptelas. It includes fields for "Repository name *", "Public" (radio button), and "Private" (radio button). The "Private" description states: "You choose who can see and commit to this repository".
- Introduce yourself with a profile README:** A section encouraging users to create a profile README. It includes a "Create" button and a sample README.md content.
- Explore repositories:** A sidebar listing popular repositories:
 - elastic / apm-server (1.2k stars, Go)
 - microsoft / pyright (12.5k stars, Python)
 - MaterialDesignInXAML / MaterialDesignInXamlToolkit (14.9k stars, C#)
- Ativar o Windows:** A watermark at the bottom right corner.

GitHub

Owner * Repository name *

 ThiagoAdaptelas / adaptelas

 adaptelas is available.

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

Create repository

GitHub

The screenshot shows a GitHub repository page for the user 'ThiagoAdaptelas' with the repository name 'adaptelas'. The page has a light gray header with navigation links: Code (highlighted), Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A search bar and various icons are also present in the header.

The main content area displays two boxes: one for 'Set up GitHub Copilot' and another for 'Add collaborators to this repository'. Below these, a 'Quick setup' section provides instructions for cloning the repository via Desktop, HTTPS, or SSH, along with a link to the repository's URL (<https://github.com/ThiagoAdaptelas/adaptelas.git>). It also suggests creating a new file or uploading an existing one, and recommends including a README, LICENSE, and .gitignore.

At the bottom, there is a command-line setup section with the following code:

```
echo "# adaptelas" >> README.md  
git init
```

A watermark for 'Ativar o Windows' (Activate Windows) is visible in the bottom right corner.

Configurando

```
git config --global user.name "nomeGitHub"
```

```
git config --global user.email "seu@email.com"
```

Conectando

```
git init
```

```
git add .
```

```
git commit -m "Primeiro Commit"
```

```
git branch -M main
```

```
git remote add origin "https://github.com/ThiagoAdaptelas/adaptelas.git"
```

```
git push -u origin main
```

Na prática

Dúvidas?

Obrigado!

