

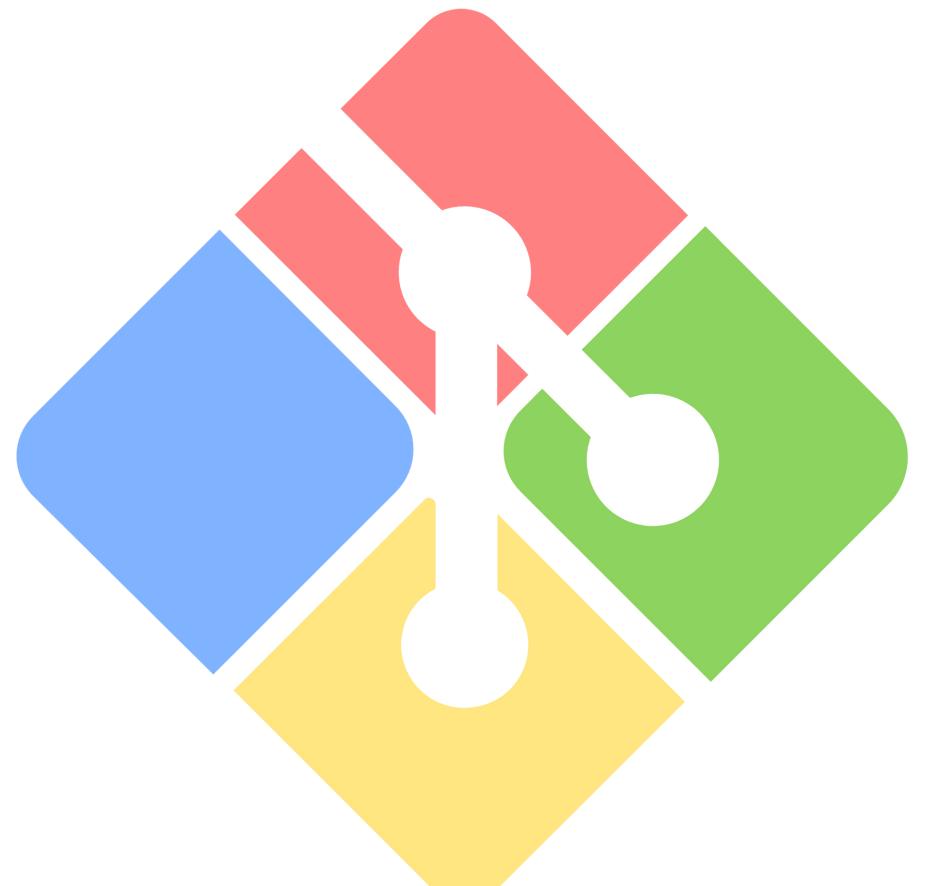
# GITHUBE GITBASH

Aula de nivelamento

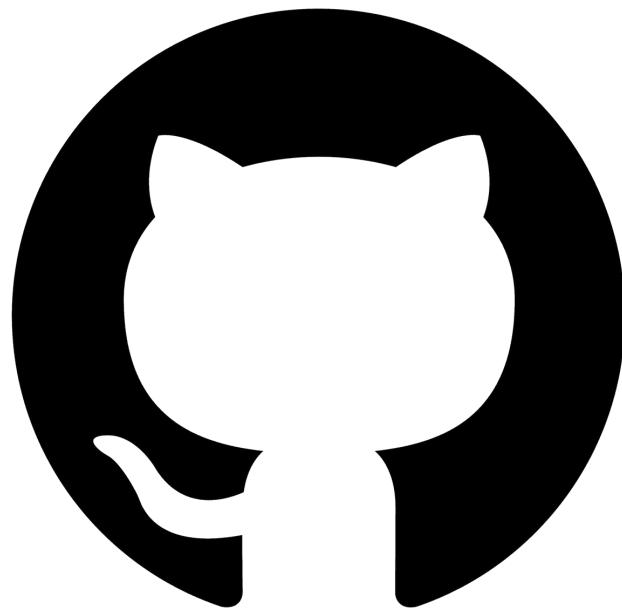
**Monitores**

Jayme Holanda

Samuel Vinícius



# O que é o Github ?



# GitHub

**O GitHub é uma plataforma de desenvolvimento colaborativo que aloja projetos na nuvem utilizando o sistema de controle de versões chamado Git.**

**A plataforma ajuda os desenvolvedores a armazenar e administrar o código e faz o registro de mudanças.**

# Programação antes do Github



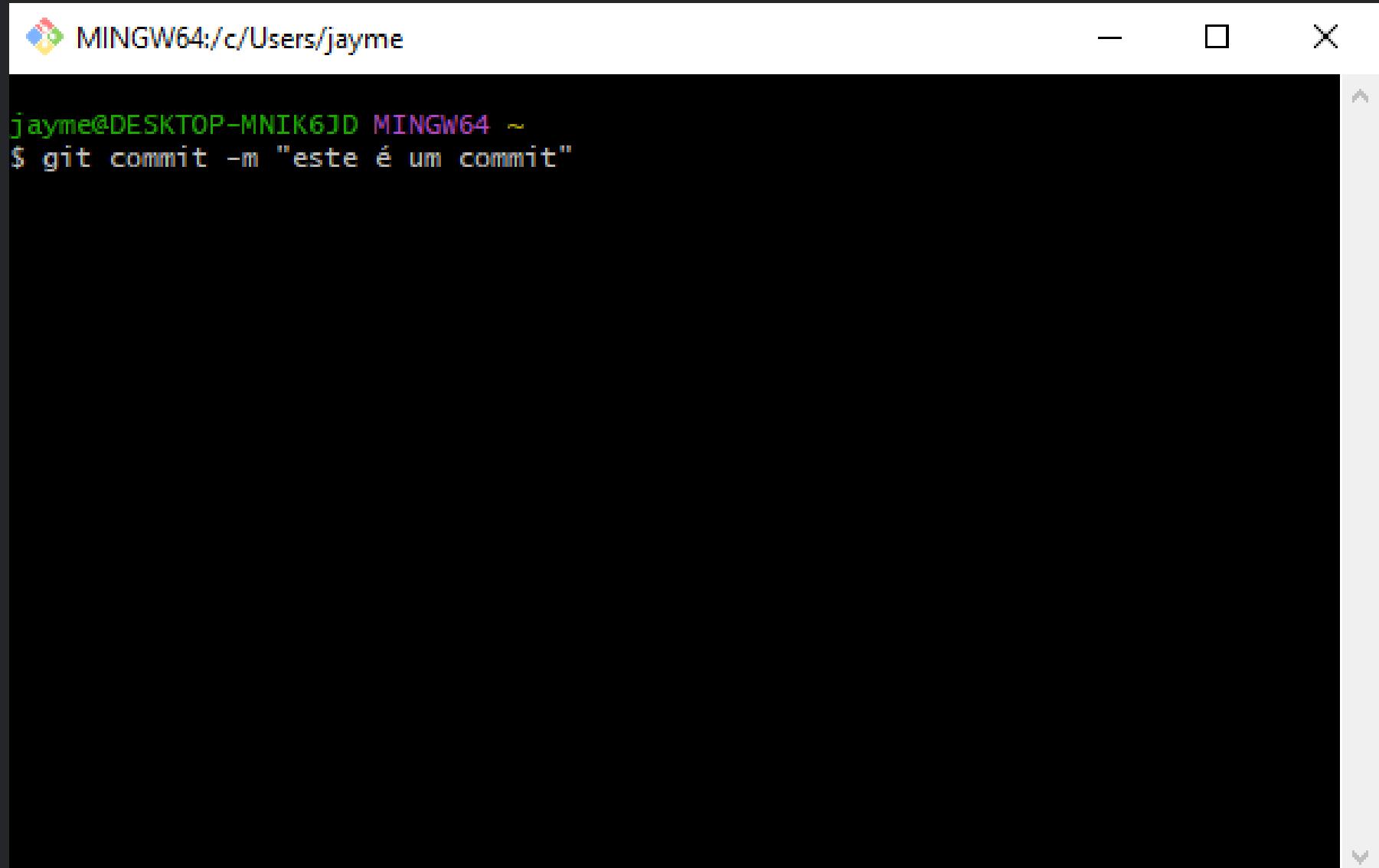
- Copias Redundantes.
- Alterações Precárias.
- Colaboração penosa.

# Git : o salvador do código



- Controle de versão preciso.
- Colaboração Eficiente.
- Fluxo de trabalho organizado.

# GIT BASH



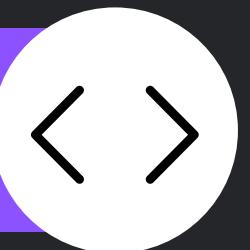
A screenshot of a Git Bash terminal window. The title bar says "GIT BASH". The window shows a command-line interface with the following text:  
MINGW64:/c/Users/jayme  
jayme@DESKTOP-MNIK6JD MINGW64 ~  
\$ git commit -m "este é um commit"

**O Git é um sistema de controle de revisão de código aberto maduro e mantido ativo usado por milhares de desenvolvedores em todo o mundo.**

**O Git Bash é uma ferramenta poderosa que facilita o gerenciamento de projetos versionados pelo Git.**

# Nomeclatura Master / Main

- **main**



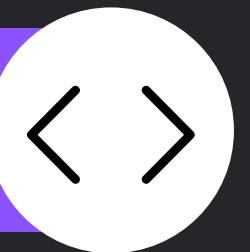
**Usar "main" em vez de "master" no Git é uma prática moderna, flexível e sem ambiguidades.**

O Git usa "main" como branch padrão desde 2020. "Master" era o padrão antigo, mas por vários motivos, "main" é a melhor escolha para projetos novos.

- **git branch -M main**

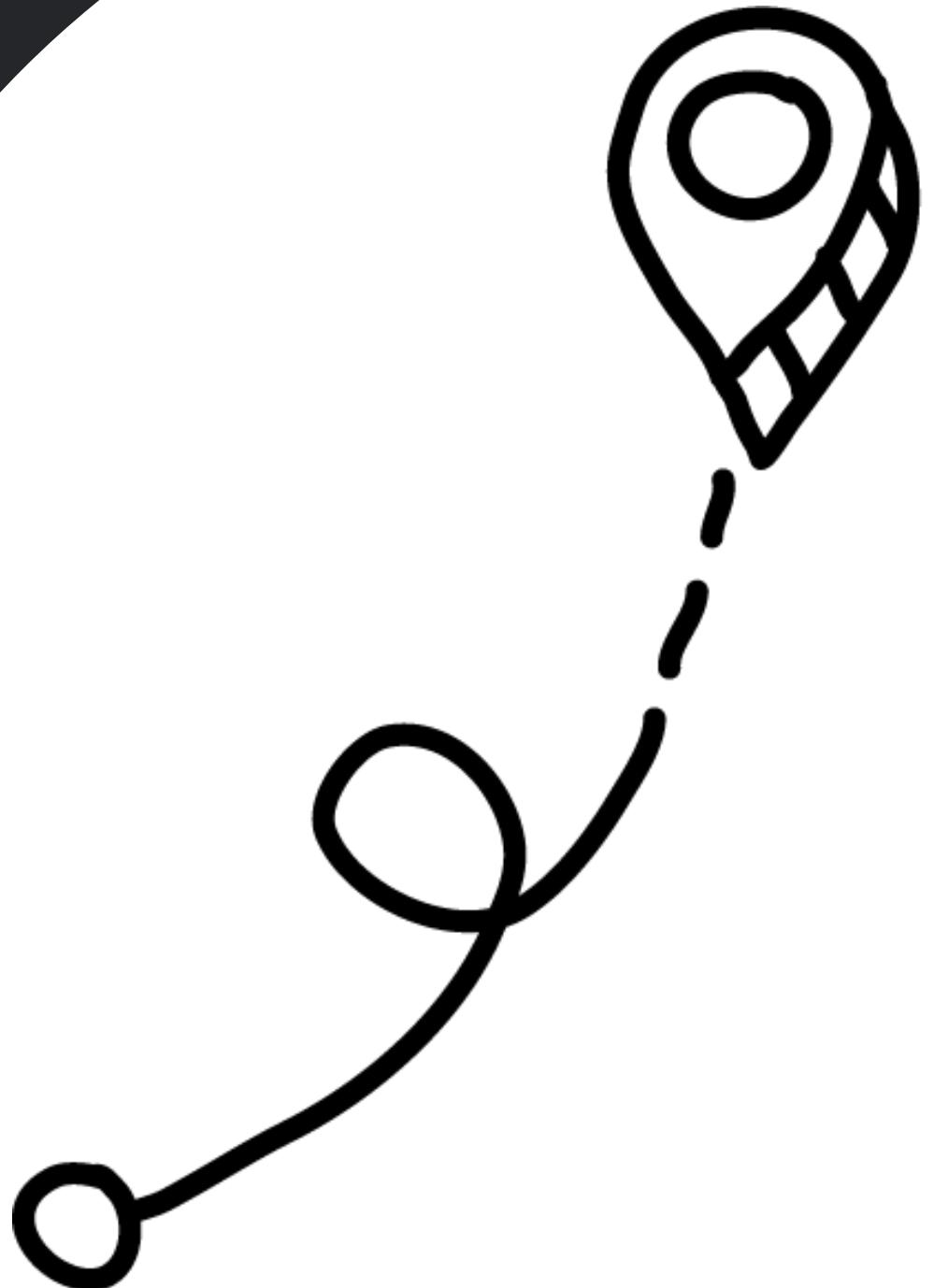
# Nomeclatura Branch ?

Branch



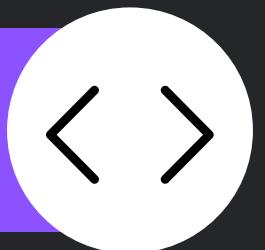
**Uma branch no Git, é como um caminho alternativo ou linha de desenvolvimento separada dentro do seu projeto. Imagine um projeto de software como uma estrada em construção.**

**A estrada principal representa a branch principal (geralmente chamada de master), que contém a versão "oficial" e estável do seu código.**



# Comandos Básicos

- `git init`

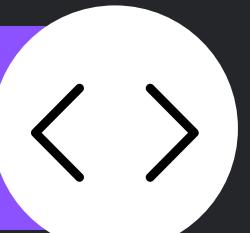


É o primeiro passo para começar a usar o Git para controlar o versionamento do seu projeto

Quando você executa `git init` no diretório do seu projeto, ele cria um subdiretório especial chamado `.git` nesse diretório. Portanto, só precisamos usar uma única vez.

# Comandos Básicos

- **git add**



No Git, o comando **git add** serve para preparar as alterações feitas em arquivos do seu projeto para serem incluídas no próximo commit.

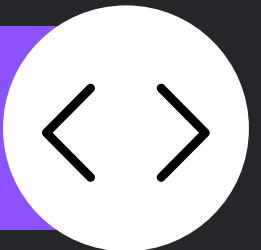
É como dizer para o Git: "Eu quero registrar essas modificações específicas como parte do próximo commit."

**git add “nome da pasta”**

**git add .**

# Comandos Básicos

git status

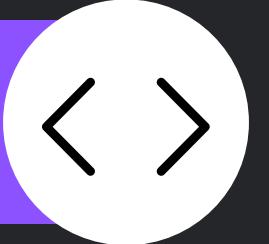


O comando **git status** mostra o estado do seu repositório Git, incluindo arquivos modificados, rastreados e não rastreados, e a diferença entre a área de trabalho e o último commit.

git status

# Comandos Básicos

- **git commit**



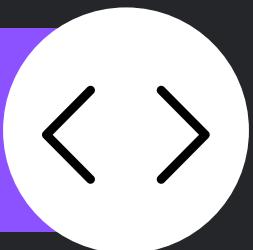
**git commit -m "Mensagem descritiva da mudança"**

O comando **git commit** registra as alterações que você fez no seu projeto, criando um ponto de referência permanente no histórico do seu código.

- -m remete a “mesage”

# Comandos Básicos

- **git push**



O comando **git push** envia as alterações que você fez no seu código para um repositório remoto, como o GitHub, para que outras pessoas possam acessá-las.

- **git push -u origin main/master** (Apenas no primeiro uso).
- **git push**
- **-u** remete a "**--set-upstream**". Uma branch upstream é uma branch remota que serve como referência para a sua branch local.



# Fazendo uma analogia

## Fila de espera:

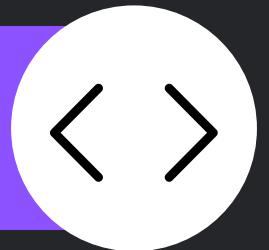
- **git add:** Alguém organiza os códigos em uma fila de espera, preparando-os para serem processados.
- **git commit:** Os códigos na fila são revisados e oficialmente registrados, como se estivessem sendo carimbados para entrar no palco.

## Palco:

- **git push:** Os códigos registrados são enviados para o palco principal, que no Git seria um repositório remoto como o GitHub, onde ficam disponíveis para todos os colaboradores.

# Comandos Básicos

- Resumindo



- **git add .:** prepara os códigos para serem processados.
- **git -m commit “texto”:** registra oficialmente os códigos.
- **git push:** envia os códigos para o repositório remoto.

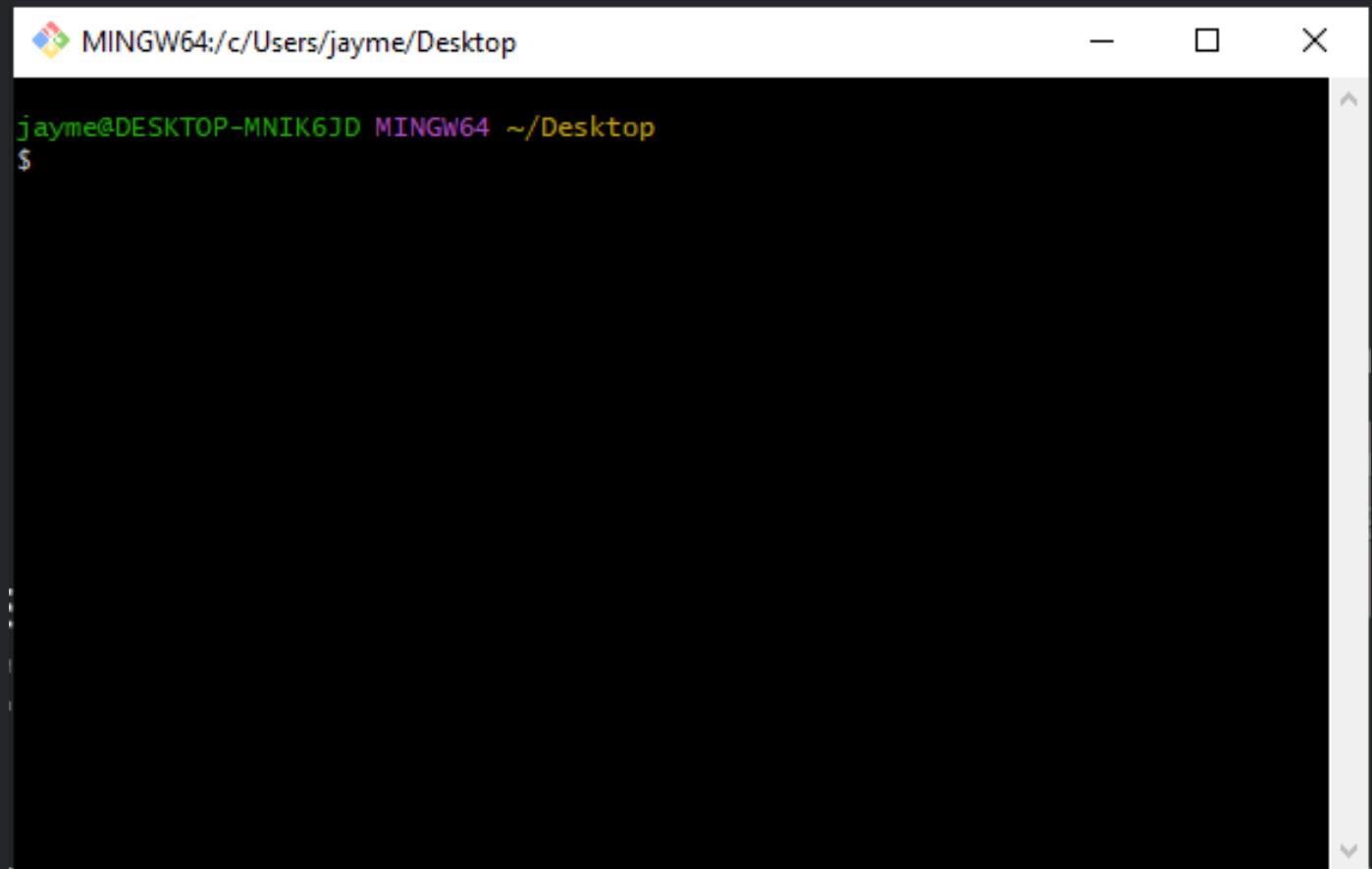
# Instalação e Configuração do GitBash



- Baixar e instalar o Git Bash:  
<https://gitforwindows.org/>



# Configurar nome e e-mail:



```
MINGW64:/c/Users/jayme/Desktop
jayme@DESKTOP-MNIK6JD MINGW64 ~/Desktop
$
```



Nome:

```
git config --global user.name "Seu Nome"
```



E-mail:

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

**Obs:** Usar o mesmo Email do  
github!

# Criando um repositório



 Set up GitHub Copilot  
Use GitHub's AI pair programmer to autocomplete suggestions as you code.  
[Get started with GitHub Copilot](#)

Quick setup — if you've done this kind of thing before  
[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/ligalatim/Teste.git>  
 Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository i

...or create a new repository on the command line

```
echo "# Este" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ligalatim/Teste.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/ligalatim/Teste.git
git branch -M main
git push -u origin main
```

...or import code from another repository  
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.  
[Import code](#)

ProTip! Use the URL for this page

- Acesse o Github e crie um novo repositorio.
- **git init**
- **git remote add origin “link” ( única vez)**

**Feche o git bash e abra-o novamente na pasta do novo arquivo.**



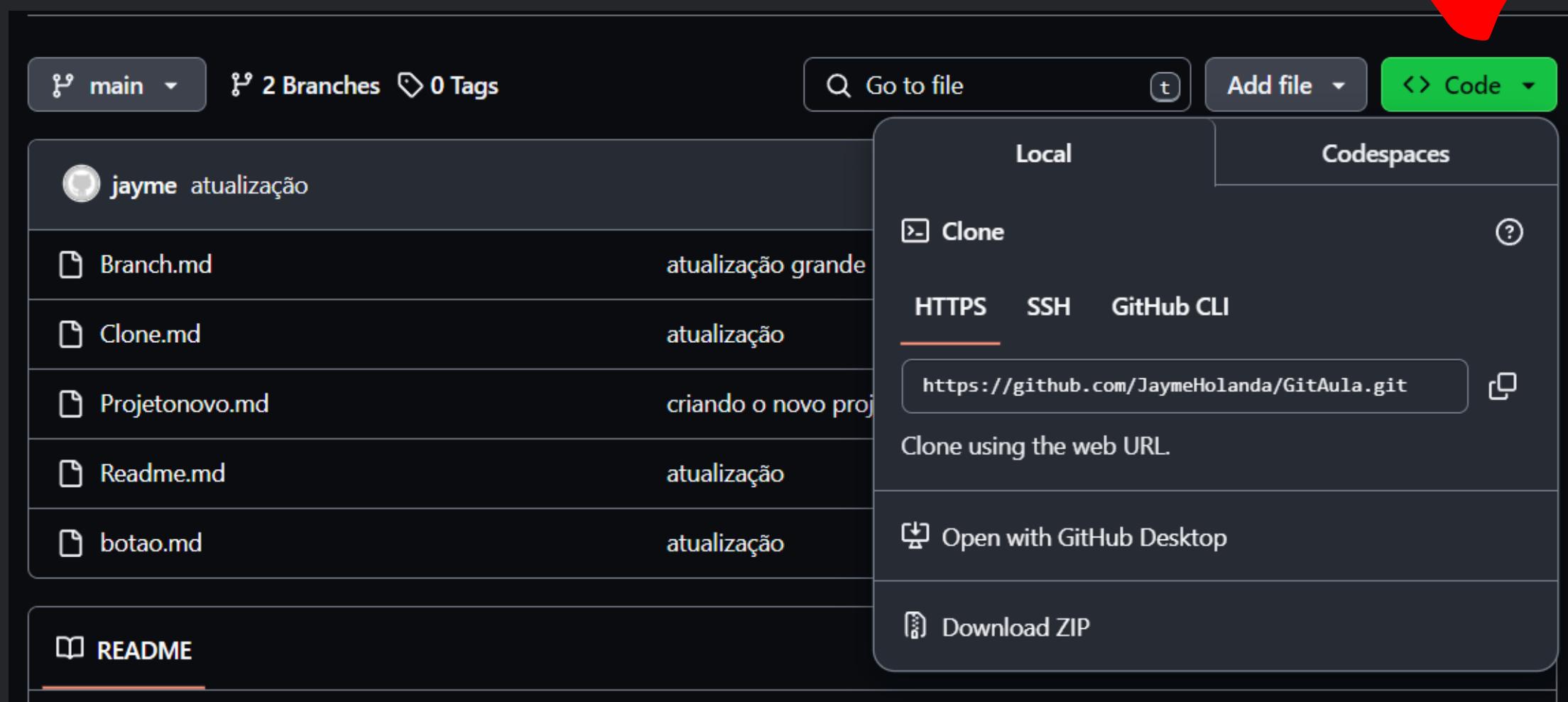
- **git add .**
- **git commit -m “texto”**
- **git push -u origin main/master**

**Dica: Faça o repositório no Desktop.**

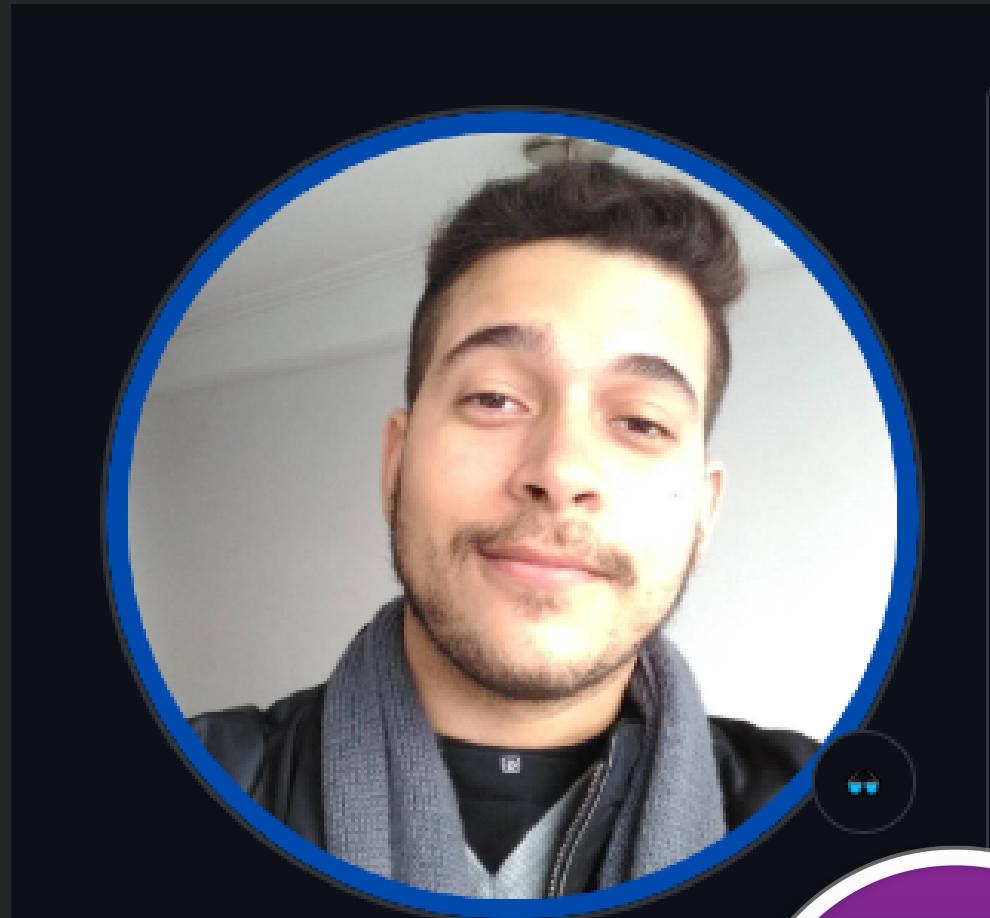
# Clonando Repositorio



# • git clone “link”



A screenshot of a GitHub repository page for a user named 'jayme'. The repository has 2 branches and 0 tags. The code tab is selected. A red arrow points to the 'Clone' button in the dropdown menu, which shows options for HTTPS, SSH, and GitHub CLI, along with the URL <https://github.com/JaymeHolanda/GitAula.git>.



**Jayne Holanda**  
JaymeHolanda



## Relembrando:

- **git init** (única vez)
- **git add .**
- **git commit -m “texto”**
- **git push -u origin main** (única vez)

```
jayme@DESKTOP-MNIK6JD MINGW64 ~/Desktop
$ git clone https://github.com/ligalatim/Teste.git
Cloning into 'Teste'...
warning: You appear to have cloned an empty repository.
```

```
jayme@DESKTOP-MNIK6JD MINGW64 ~/Desktop
$
```



# Exercício de casa

## Exercício 01

**Abra o Github da liga acadêmica latim e clone o repositório GitAula em sua máquina.**

## Exercício 02

**Faça um repositório local. Nele faça um arquivo HTML, se possível use CSS. Faça um resumo do que vimos nesta aula e suba para o seu repositório do Github.**