



# Git e GitHub

Instituto Federal de Mato Grosso (IFMT)  
Pontes e Lacerda



# Conteúdo da aula

- **Introdução**
- Git e GitHub
- Comandos e conceitos
- Criando repositório individual
- Criando repositório compartilhado
- Conclusão
- Exercícios propostos
- Referências bibliográficas

---

# Conteúdo da aula

- Introdução
- **Git e GitHub**
- **Comandos e conceitos**
- **Criando repositório individual**
- **Criando repositório compartilhado**
- Conclusão
- Exercícios propostos
- Referências bibliográficas



# Conteúdo da aula

- Introdução
- Git e GitHub
- Comandos e conceitos
- Criando repositório individual
- Criando repositório compartilhado
- **Conclusão**
- **Exercícios propostos**
- **Referências bibliográficas**

# Introdução



-

---

# Conteúdo da aula

- Introdução
- **Git e GitHub**
- Comandos e conceitos
- Criando repositório individual
- Criando repositório compartilhado
  
- Conclusão
- Exercícios propostos
- Referências bibliográficas

# GIT e GitHub

---

- Git
  - Sistema de versionamento de documentos
    - Armazenar programas de computadores
    - Conter versões de um TCC
  - Utilidade do Git
    - Reintegrar versões antigas
    - Histórico de versões
    - Coordenar conflito na criação compartilhada de um arquivo
    - Criar formas seguras de alterar o código mantendo o programa rodando

# GIT e GitHub

---

- *Git*
  - Instalado na máquina do usuário
  - Uso do terminal de comandos Git Bash (Após instalado)
  - Site do Git:

<https://www.git-scm.com/>

**OBS:** Durante a instalação selecione o terminal do windows quando solicitado

# GIT e GitHub

---

- **GitHub**
  - Página web (ou programa desktop) que permite criar e compartilhar repositórios de arquivos remotos
  - Compartilhados com outras pessoas
    - Identificação para alterar os repositórios
      - Token
      - Login e senha

# GIT e GitHub

---

- **GitHub**

- É possível realizar uma ligação entre o repositório local e o repositório remoto (no GitHub)
- Rede social de programadores
- Necessária a criação de uma conta
- Site do GitHub: <https://github.com/>



# Conteúdo da aula

- Introdução
- Git e GitHub
- **Comandos e conceitos**
- Criando repositório individual
- Criando repositório compartilhado
- Conclusão
- Exercícios propostos
- Referências bibliográficas

# Comandos e conceitos

## Comandos Git

Comando	Descrição (breve)
git init	Transforma um diretório comum em um repositório
git add	Identifica todos o(s) arquivo(s) alterado(s)
git commit	Cria uma nova versão do documento
git branch -M	(Com a opção -M) Troca o nome do ramo atual
git remote add	Cria uma “variável” que armazena o endereço do repositório remoto
git push	Envia os dados do repositório local para o remoto
git clone	Copia e cola um repositório remoto na máquina local

# Comandos e conceitos

## Comandos Git

Comando	Descrição (breve)
git pull	Pega os arquivos do repositório remoto e atualiza o repositório local
git branch	Lista todos os ramos
git branch ramo1	Cria um branch de nome “ramo1”
git checkout	Entre em outro ramo
git merge	Mescla o ramo atual com outro ramo
git config user.name	Altera o nome do responsável pelo diretório local
git config user.email	Altera o email do responsável pelo diretório local

# Comandos e conceitos

Comandos Git	
Comando	Descrição (breve)
git status	Apresente a situação atual do git
git log	Apresenta o histórico dos commits

# Comandos e conceitos

## Conceitos Git e GitHub

Conceito	Descrição (breve)
Branch (ramo)	Uma sandbox para executar as alterações sem impactar o programa principal
Colaboradores	É possível adicionar pessoas para auxiliarem na elaboração dos arquivos do repositório remoto
Repositório git	Criado a partir do git init, contém as versões e informações sobre os documentos
Git Bash	Terminal de comandos usando para executar o git
Tokens	Permissões temporárias concedidas aos colaboradores
Pull Request	Solicita ao dono do repositório uma permissão para alterar o programa principal

# Comandos e conceitos

## Conceitos Git e GitHub

Conceito	Descrição (breve)
Regra de Pull Request	Regras criadas pelo dono do repositório para evitar alterações indevidas ou inapropriadas
Repositório remoto	Criado a partir do GitHub, podendo ser compartilhado



# Conteúdo da aula

- Introdução
- Git e GitHub
- Comandos e conceitos
- **Criando repositório individual**
- Criando repositório compartilhado
- Conclusão
- Exercícios propostos
- Referências bibliográficas

## Trabalho individual

---

Como usar o Git e o GitHub para codar  
**individualmente?**

# Git Init

---

- Transforma um diretório comum em um repositório
  - Execute este comando dentro do diretório que irá ser alterado
  - Cria uma pasta oculta chamada .git
    - Contém versões antigas do arquivos
    - Contém o arquivo “config” que guarda informações importantes
    - Formato do comando

```
git init
```

# Git Init

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git (master)
$ mkdir NovoRepositorio|
```

Terminal Git Bash:

Aceita comandos linux



# Git Init

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git (master)  
$ mkdir NovoRepositorio
```



Determina o Branch (Ramo)  
Atual

# Git Init

```
hugopc@Hugopc MTNGW6U ~/Repositorios_Git (master)
$ mkdir NovoRepositorio|
```



comando **mkdir** cria um novo  
diretório

# Git Init

redação

Fechar

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git (master)
$ mkdir NovoRepositorio
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git (master)
$ cd NovoRepositorio/
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ git init
```



Repositório  
corrente

Dentro do diretório execute  
o git init

# Git Init

```
hugop@Hugopc MINGW64 ~/Repositorios_Git (master)
$ cd NovoRepositorio/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ git init
Initialized empty Git repository in C:/Users/hugop/Repositorios_Git/Novo
Repositorio/.git/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ ls -la
total 8
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 .
drwxr-xr-x 1 hugop 197609 0 jun 27 14:42 ../
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 .git/
```

Mensagem apresentada  
após git init



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ |
```

# Git Init

```
hugop@Hugopc MINGW64 ~/Repositorios_Git (master)
$ cd NovoRepositorio/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ git init
Initialized empty Git repository in C:/Users/hugop/Repositorios_Git/Novo
Repositorio/.git/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ ls -la
total 8
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 .
drwxr-xr-x 1 hugop 197609 0 jun 27 14:42 ../
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 .git/
```

Comando que lista todos arquivos, inclusive os ocultos

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ |
```

# Git Init

```
hugop@Hugopc MINGW64 ~/Repositorios_Git (master)
$ cd NovoRepositorio/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ git init
Initialized empty Git repository in C:/Users/hugop/Repositorios_Git/Novo
Repositorio/.git/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ ls -la
total 8
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 .
drwxr-xr-x 1 hugop 197609 0 jun 27 14:42 /
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 .git/
```

Diretório criado pelo git

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ |
```

# Git config

---

- Duas informações são solicitadas para realizar um commit
  - Nome e email
- As configurações locais são armazenados dentro do arquivo config (localizado na pasta .git)
- Os comandos abaixo inserem o nome e o email no git para o repositório corrente apenas

```
git config user.name "Fulano"  
git config user.email "fulano@example.com"
```

# Git config

---

- Para inserir globalmente as informações de nome e senha é necessária a declaração da opção “--global”

```
git config --global user.name "Fulano"  
git config --global user.email "fulano@example.com"
```

# Git config

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ git config --global user.name "Hugo"
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ git config --global user.email "hugo@example.com"
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$
```

# Git config

```
hugop@Hugopc MINGW64 ~/Repositorios Git/NovoRepositorio (master)
$ git config --global user.name "Hugo"
```

```
hugop@Hugopc MINGW64 ~/Repositorios Git/NovoRepositorio (master)
$ git config --global user.email "hugo@example.com"
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$
```

Inserção das informações no git

OBS: Sem o global as informações  
seriam válidas apenas para o  
diretório corrente

# Git config

```
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 .git/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)  
$
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)  
$ git config --global user.name "Hugo"
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)  
$ git config --global user.email "hugo@example.com"
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)  
$ ls -la
```

```
total 8
```

```
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 ./
```

```
drwxr-xr-x 1 hugop 197609 0 jun 27 14:42 ../
```

```
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 .git/
```



# Git config

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ cd .git/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio/.git (GIT_DIR!)
```

```
$ ls -la
total 11
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 .
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 ../
-rw-r--r-- 1 hugop 197609 130 jun 27 14:49 config
-rw-r--r-- 1 hugop 197609 73 jun 27 14:49 description
-rw-r--r-- 1 hugop 197609 23 jun 27 14:49 HEAD
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 hooks/
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 info/
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 objects/
drwxr-xr-x 1 hugop 197609 0 jun 27 14:49 refs/
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio/.git (GIT_DIR!)
$
```

# Git add

---

- “Rastreia” o(s) documento(s). Em outras palavras, encontra os documentos e torna ciente da existência destes arquivos.
  - Para rastrear um documento específico (por exemplo, arquivo.txt):

```
git add arquivo.txt
```
  - Para rastrear todos os documentos do repositório

```
git add .
```

# Git add

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)  
$ vim arquivo.txt
```



Para testar o comando é criado o  
arquivo.txt

Uso do editor de texto vim

# Git add

Tela inicial do programa vim

```
arquivo.txt [unix] (20:59 31/12/1969)  
"arquivo.txt" [Novo]
```

0,0-1 Tudo

# Git add

Aperte a tecla i para começar a  
editar o arquivo

arquivo.txt [unix] (20:59 31/12/1969)

0,1 Tudo

-- INSERÇÃO --



# Git add

Primeiro commit - Hugo

~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~

Ao finalizar a edição pressione ESC seguida das teclas :wq para salvar e finalizar

w → Write (salvar)

q → Quit (Sair)

Caso queira apenas fechar o arquivo sem salvar pressione ESC seguida das teclas :q

q → Quit (Sair)

arquivo.txt[+] [unix] (20:59 31/12/1969)  
:wq

1,22 Tudo

# Git add

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ vim arquivo.txt
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ ls -l
total 1
-rw-r--r-- 1 hugop 197609 23 jun 27 16:22 arquivo.txt
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ |
```

# Git add

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio  
$ vim arquivo.txt
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio  
$ ls -l  
total 1  
-rw-r--r-- 1 hugop 197609 23 jun 27 16:22 arquivo.txt
```

Apesar do warning, a operação foi efetuada



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)  
$ git add arquivo.txt  
warning: in the working copy of 'arquivo.txt', LF will be replaced by CR  
LF the next time Git touches it
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)  
$ |
```

# Git status

---

- Comando utilizado para apresentar o estado atual do git
  - Algumas mensagens que podem aparecer:
    - Execute o commit
    - Rastreie o arquivo

**Dica**: Quando estiver com dúvida do que está acontecendo use o git status

# Git status

---

- Algumas mensagens que podem aparecer:

```
$ git status  
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   arquivo.txt
```

Falta executar  
commit

# Git status

---

- Algumas mensagens que podem aparecer:

```
$ git status  
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)  
  new file: arquivo.txt  
  new file: tes.txt
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)  
(use "git restore <file>..." to discard changes in working directory)  
  modified: tes.txt
```

Falta executar  
commit e falta  
rastrear arquivo

# Git commit

---

- Efetiva uma alteração no arquivo apresentando uma nova versão dele
- Comando conhecido como “commitar”
- Formato do comando:

```
git commit -m “mensagem sobre o commit feito”
```

- A opção -m determina que a string que aparece a seguir é uma informação sobre a alteração feita no commit

# Git commit

```
hugopc@Hugopc MINGW64 ~/Repositories_Git/NovoRepositorio (master)
$ git commit -m "Primeiro Commit"
[master (root-commit) bac79fb] Primeiro Commit
 2 files changed, 2 insertions(+)
 create mode 100644 arquivo.txt
 create mode 100644 tes.txt
```

```
hugopc@Hugopc MINGW64 ~/Repositories_Git/NovoRepositorio (master)
$ |
```

Commit realizado com sucesso

## Git commit



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/  
$ git status  
On branch master  
nothing to commit, working tree clean
```



Mensagem do git status

# Git commit

---

- Erros comuns ao commitar:
  - Arquivo não foi rastreado
  - Identificação do usuário não declarada (nome e email)

# A estrada até aqui

---

- Um repositório Git local foi criado
  - Git init - Git add - Git commit - Git config - Git status
- **Próximo passo:** Criar um diretório remoto para podermos enviar os arquivos locais para o repositório remoto
  - Criar o repositório remoto (GitHub)
  - Alterar o nome do repositório local (Git)
  - Enviar as alterações locais para o remoto (Git)

# A estrada até aqui

- Criando repositório remoto GitHub

The screenshot shows the GitHub Home page. On the left, there's a sidebar with 'Top Repositories' listing several repositories like 'hpbtakiuchi/exemplo\_20\_06\_3'. A green 'New' button is highlighted with a red box. In the top right, a red arrow points to a dropdown menu where 'New repository' is also highlighted with a red box.

**GitHub Home Page:**

- New Repository:** A green button labeled 'New' is highlighted with a red box.
- Updates to your homepage feed:** A modal window explaining the combined power of the Following and For you feeds.
- Trending repositories:** Shows 'modelscope/DiffSynth-Studio' (Python, 4.3k stars) and 'evolutionyscale/esm' (Python, 576 stars).
- Recent activity:** A placeholder message: "When you take actions across GitHub, we'll provide links to that activity here."
- Latest changes:** A list of recent updates, including:
  - 5 hours ago: SBOMs now include copyright attribution data
  - 6 hours ago: GitHub Issues & Projects – GraphQL and webhook support for project status updates...
  - 9 hours ago: Copilot Enterprise knows about pull requests, discussions, and files – June Updates
  - Yesterday: Secret scanning on demand validity checks for NuGet and Azure

**GitHub Top Right Menu:**

- + New repository
- Import repository
- New codespace
- <> New gist
- New organization
- New project

# A estrada até aqui

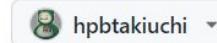
- Criando repositório remoto GitHub

## Create a new repository

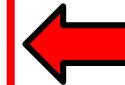
A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*



Nome do  
repositório

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

Description (optional)

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.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

# A estrada até aqui

- Criando repositório remoto GitHub

Usaremos em aula repositórios públicos.

Repositórios privados para armazenar arquivos

Create a new repository

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

Required fields are marked with an asterisk (\*).

Owner \* Repository name \*

hpbtakiuchi /

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

Description (optional)

 Public Anyone on the internet can see and commit to this repository.

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

Público ou privado

Initialize this repository with:

Add a README file  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

# A estrada até aqui

- Criando repositório remoto GitHub

Usaremos em aula repositórios públicos.

Repositórios privados para armazenar arquivos

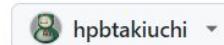
Cria automaticamente o README caso marcada a opção

## Create a new repository

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

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*



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

Description (optional)

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.

Initialize this repository with:

Add a README file

This is where you can write a long description for your repository.

Add .gitignore

.gitignore template: None

Público ou privado



Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

# Repositório remoto

- Página inicial do repositório sem or README criado

The screenshot shows the main interface of a GitHub repository named "Aula2706". At the top, there are navigation links: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the header, there are buttons for Pin, Unwatch (1), Fork (0), and Star (0). The repository name "Aula2706" is followed by "Public".

Two main sections are displayed:

- Set up GitHub Copilot:** A section with a "Copilot" icon. It encourages users to "Use GitHub's AI pair programmer to autocomplete suggestions as you code." A button labeled "Get started with GitHub Copilot" is present.
- Add collaborators to this repository:** A section with a "+" icon. It asks users to "Search for people using their GitHub username or email address." A button labeled "Invite collaborators" is present.

A large blue banner at the bottom provides "Quick setup" instructions:

- It says "if you've done this kind of thing before".
- It includes options to "Set up in Desktop" or "HTTPS" (selected) or "SSH".
- It shows the repository URL: <https://github.com/hpbtakiuchi/Aula2706.git>.
- It suggests starting by creating a new file or uploading an existing file, and recommends including a [README](#), [LICENSE](#), and [.gitignore](#).

Below the banner, another section titled "...or create a new repository on the command line" provides the following terminal commands:

```
echo "# Aula2706" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/hpbtakiuchi/Aula2706.git
git push -u origin main
```

# Repositório remoto

- Página inicial do repositório sem or README criado

The screenshot shows the GitHub repository page for 'Aula2706'. At the top, there are navigation links: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the header, there are sections for 'Set up GitHub Copilot' and 'Add collaborators to this repository'. A large blue box titled 'Quick setup — if you've done this kind of thing before' contains instructions and a command-line example. A red arrow points from the text 'Comandos iniciais a serem executados, porém estão incompletos' to the command-line example.

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Aula2706 Public

Pin Unwatch 1 Fork 0 Star 0

**Set up GitHub Copilot**  
Use GitHub's AI pair programmer to autocomplete suggestions as you code.  
Get started with GitHub Copilot

**Add collaborators to this repository**  
Search for people using their GitHub username or email address.  
Invite collaborators

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/hpbtakiuchi/Aula2706.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
echo "# Aula2706" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git branch -M main  
git remote add origin https://github.com/hpbtakiuchi/Aula2706.git  
git push -u origin main
```

Comandos iniciais a serem executados, porém estão incompletos

53

# Repositório remoto

## Comandos iniciais completos:

```
echo "# seurepositorio" >> README.md
```

```
git init
```

```
git config user.email "nome@example.com"
```

```
git config user.nome "Nome"
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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

**OBS:**  
Execute um  
por vez

# Repositório remoto

Comandos iniciais completos:

```
echo "# seurepositorio" >> README.md
```

```
git init
```

```
git config user.email "nome@example.com"
```

```
git config user.nome "Nome"
```

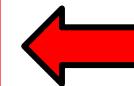
```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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



Cria um arquivo README e insere o texto "# seurepositorio"

# Repositório remoto

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$ echo "# Aula2706" >> README.md
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$ ls -la
total 14
drwxr-xr-x 1 hugop 197609 0 jun 27 23:54 ./
drwxr-xr-x 1 hugop 197609 0 jun 27 14:42 ../
drwxr-xr-x 1 hugop 197609 0 jun 27 23:47 .git/
-rw-r--r-- 1 hugop 197609 23 jun 27 16:22 arquivo.txt
-rw-r--r-- 1 hugop 197609 11 jun 27 23:54 README.md
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$ |
```

# Repositório remoto

```
# Aula2706
```



Arquivo README

README.md [unix] (23:54 27/06/2024)

"README.md" [unix] 1L, 11B

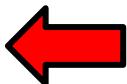
1,1 Tudo

# Repositório remoto

**Comandos iniciais completos:**

```
echo "# seurepositorio" >> README.md
```

```
git init
```



```
git config user.email "nome@example.com"
```

```
git config user.nome "Nome"
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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

# Repositório remoto

**Comandos iniciais completos:**

```
echo "# seurepositorio" >> README.md
```

```
git init
```

```
git config user.email "nome@example.com"
```



```
git config user.nome "Nome"
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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

# Repositório remoto

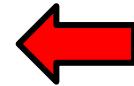
**Comandos iniciais completos:**

```
echo "# seurepositorio" >> README.md
```

```
git init
```

```
git config user.email "nome@example.com"
```

```
git config user.nome "Nome"
```



```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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

# Repositório remoto

**Comandos iniciais completos:**

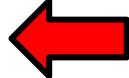
```
echo "# seurepositorio" >> README.md
```

```
git init
```

```
git config user.email "nome@example.com"
```

```
git config user.nome "Nome"
```

```
git add README.md
```



```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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

# Repositório remoto

**Comandos iniciais completos:**

```
echo "# seurepositorio" >> README.md
```

```
git init
```

```
git config user.email "nome@example.com"
```

```
git config user.nome "Nome"
```

```
git add README.md
```

```
git commit -m "first commit" ←
```

```
git branch -M main
```

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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

# Repositório remoto

**Comandos iniciais completos:**

```
echo "# seurepositorio" >> README.md
```

```
git init
```

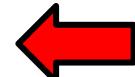
```
git config user.email "nome@example.com"
```

```
git config user.nome "Nome"
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```



**Comando novo !!**

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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

# Git branch -M

---

- Comando git branch está associado à criação e manipulação de branches (ramos)
  - Branch é uma “sandbox” para execução de alterações seguras
  - A opção **-M** renomeia o branch atual
    - Necessário trocar o nome padrão (master) do repositório local
    - Ele deve ser igual ao repositório remoto no GitHub (main)

```
git branch -M main
```

# Git branch -M

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ git branch -M main
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$
```

# Git branch -M

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ git branch -M main ←
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$
```

# Git branch -M

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (master)
$ git branch -M main ←
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$
```

The diagram illustrates the effect of the command. On the left, the terminal shows the command being run: '\$ git branch -M main'. A red box highlights this command, and a large red arrow points from it to the right. On the right, two boxes represent the state of the repository after the command is executed. The top box contains '(master)' and the bottom box contains '(main)'. A curved blue arrow originates from the '(master)' box and points to the '(main)' box, indicating that the 'master' branch has been renamed to 'main'.

Troca o nome do ramo

# Repositório remoto

**Comandos iniciais completos:**

```
echo "# seurepositorio" >> README.md
```

```
git init
```

```
git config user.email "nome@example.com"
```

```
git config user.nome "Nome"
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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

Comando novo !!



# Git remote

---

- Comando pode ser usado para endereçar o servidor remoto
  - Cria/altera o link entre o repositório local e o repositório remoto no GitHub
  - Geralmente é criada uma “variável” chamada **origin** que conterá a URL do repositório remoto

# Git remote

---

- Comando pode ser usado para endereçar o servidor remoto

- Comando para criar um link:

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

- Comando para alterar um link

```
git remote set-url origin https://github.com/seulogin/seurepositorio.git
```

# Git remote

---

- Git remote add

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$ git remote add origin https://github.com/hpbtakiuchi/Aula276.git
```

# Git remote

---

- Git remote add

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$ git remote add origin https://github.com/hpbtakiuchi/Aula276.git
```

OPS! Erro na URL



# Git remote

---

- Git remote add

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$ git remote add origin https://github.com/hpbtakiuchi/Aula276.git
```

- Git remote set-url

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$ git remote set-url origin https://github.com/hpbtakiuchi/Aula2706.git
```

# Git remote

---

- Erros que podem aparecer
  - 1) Esquecer de declarar o endereço remoto

```
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.
```

Please make sure you have the correct access rights  
and the repository exists.

# Git remote

---

- Erros que podem aparecer
  - 2) Tentar adicionar uma “variável” que já foi criada

```
$ git remote add origin https://github.com/hpbtakiuchi/Aula2706.git  
error: remote origin already exists.
```

- Para corrigir use set-url ao invés de add

# Repositório remoto

**Comandos iniciais completos:**

```
echo "# seurepositorio" >> README.md
```

```
git init
```

```
git config user.email "nome@example.com"
```

```
git config user.nome "Nome"
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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



**Comando novo !!**

# Git push

---

- Envia as modificações feita no repositório local para o remoto

- Necessário especificar o endereço remoto (GitHub)
  - Usa a “variável” origin

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

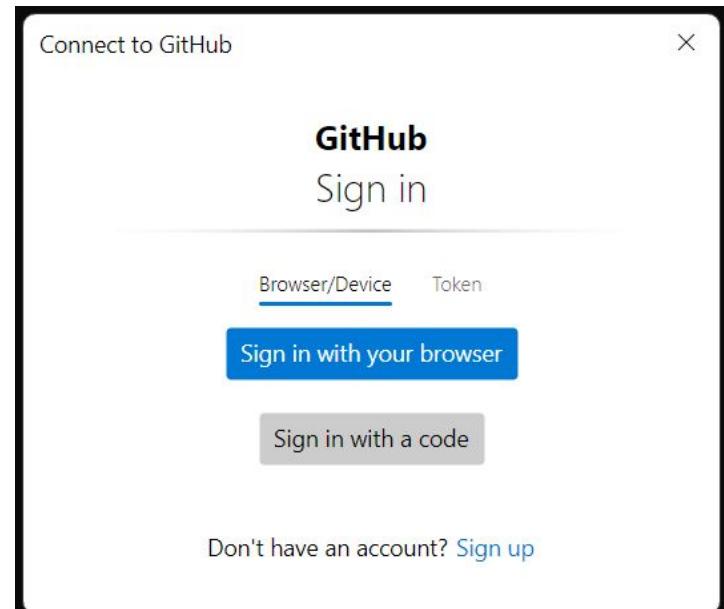
- O parâmetro -u determina que se a ligação entre os repositórios tiver sucesso esta conexão será armazenada
  - O último parâmetro determina o ramo remoto que será “linkado”

# Git push

---

- Para executar um push é necessária um tipo de identificação

- Duas formas de identificar no GitHub
  - Login e senha
  - Token



Tela que solicita a identificação

# Token

---

- Corresponde a uma “moeda” de acesso ao seu repositório remoto
  - Concede algumas permissões (temporárias sem tempo limitado)
  - É Criada no GitHub do proprietário do repositório remoto
  - Um modo alternativo à solicitação de login e senha
  - É recomendado precaução ao se logar em computadores compartilhados
    - Após o uso do token é recomendado removê-lo

# Token

- Criando um token no GitHub

The screenshot shows a GitHub repository page for 'Aula2706'. At the top right, there is a user profile icon with a red box and a red arrow pointing to it, indicating where to click to access settings. The page includes sections for 'Set up GitHub Copilot', 'Add collaborators', and 'Quick setup'. It also provides command-line instructions for creating a new repository.

hpbtakiuchi / Aula2706

Type  to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Aula2706 Public

Pin Unwatch 1 Fork 0 Star 0

Set up GitHub Copilot

Add collaborators to this repository

Quick setup — if you've done this kind of thing before

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
echo "# Aula2706" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git branch -M main
```

Acesse as opções de configurações (settings) clicando no seu perfil

# Token

- Criando um token no GitHub

The screenshot shows a GitHub repository page for 'Aula2706'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository details show it is public and has 1 watch. A sidebar on the right lists various GitHub features like Your profile, Your repositories, and Your Copilot. A red box highlights the 'Settings' link in the sidebar, with a red arrow pointing to it from below. A large red box also highlights the 'Settings' tab at the bottom of the main repository page. The page content includes sections for GitHub Copilot setup, adding collaborators, quick setup instructions, and command-line repository creation. The URL is https://github.com/hpbtkiuchi/Aula2706.git.

hpbtakiuchi / Aula2706

Type ⌘ to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Aula2706 Public

Pin Unwatch 1 Fork 0

Set status

Your profile Your repositories Your Copilot Your projects Your stars Your gists Your organizations Your enterprises Your sponsors Try Enterprise Feature preview

Settings

Get started with GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot

Add collaborators to this repository

Search for people using their GitHub username or email address.

Invite collaborators

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/hpbtkiuchi/Aula2706.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

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

```
echo "# Aula2706" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/hpbtkiuchi/Aula2706.git
```

Menu das configurações

https://github.com/settings/profile

# Token

- Criando um token no GitHub

The screenshot shows the 'Developer settings' page on GitHub. On the left, there's a sidebar with various links like 'Moderation', 'Code, planning, and automation', 'Repositories', etc. At the bottom of the sidebar, 'Developer settings' is highlighted with a red box and a black arrow pointing to it from the bottom-left. The main area has fields for 'Name' (set to 'Don't specify'), 'URL', and 'ORCID iD'. Below these are sections for 'Social accounts' and 'Time zone'. A large red box covers the right side of the page, containing the following text:

**No perfil, acesse a última opção na aba lateral esquerda chamada configurações de desenvolvedor (Developer settings)**

# Token

- Criando um token no GitHub

The screenshot shows the 'Developer settings' page on GitHub. On the left, there's a sidebar with various links like 'Moderation', 'Code, planning, and automation', 'Repositories', etc. At the bottom of the sidebar, 'Developer settings' is highlighted with a red box and a black arrow pointing to it from the bottom-left. The main area has fields for 'Name' (set to 'Don't specify'), 'URL', and 'ORCID iD'. Below these are sections for 'Social accounts' and 'Time zone'. A large red box covers the right side of the page, containing the following text:

**No perfil, acesse a última opção na aba lateral esquerda chamada configurações de desenvolvedor (Developer settings)**

# Token

- Criando um token no GitHub

The screenshot shows the GitHub Developer Settings page. At the top, there are navigation icons and a search bar. Below the search bar, there are tabs for "GitHub Apps" and "OAuth Apps", with "GitHub Apps" currently selected. A dropdown menu is open, showing options: "GitHub Apps", "OAuth Apps", and "Personal access tokens". The "Personal access tokens" option is highlighted with a red box and a large red arrow points to it from below. The main content area is titled "GitHub Apps" and contains text about building GitHub Apps. At the bottom, there is a footer with links to GitHub's Terms, Privacy, Security, Status, Docs, Contact, Manage cookies, and a "Do not share my personal information" link.

Dentro de Developer Settings  
escolha a opção Tokens de  
acesso pessoal (Personal  
access tokens)

# Token

- Criando um token no GitHub

The screenshot shows the GitHub Developer Settings page. On the left, there's a sidebar with options: GitHub Apps, OAuth Apps, Personal access tokens (selected), Fine-grained tokens (Beta), and Tokens (classic). The main area is titled "Personal access tokens (classic)" and contains a button to "Generate new token" and another to "Revoke all". Below this is a section for tokens generated, with a link to the GitHub API. A note explains that these tokens function like OAuth access tokens. At the bottom, there's a copyright notice for GitHub, Inc. and links to Terms, Privacy, Security, Status, Docs, Contact, Manage cookies, and a link to not share personal information.

**Uma lista de tokens pessoais deve aparecer.**

**É possível remover/criar tokens nesta página**

# Token

- Criando um token no GitHub

The screenshot shows the GitHub Developer Settings page. On the left, there's a sidebar with options: GitHub Apps, OAuth Apps, Personal access tokens (selected and highlighted with a red box), Fine-grained tokens (Beta), and Tokens (classic). A large red arrow points from the bottom of the slide towards the 'Tokens (classic)' button. The main content area is titled 'Personal access tokens (classic)' and contains a 'Generate new token' button and a 'Revoke all' button. Below these are sections for generated tokens and a note about their function. At the bottom, there's a footer with copyright information and navigation links.

Personal access tokens (classic)

Generate new token ▾ Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

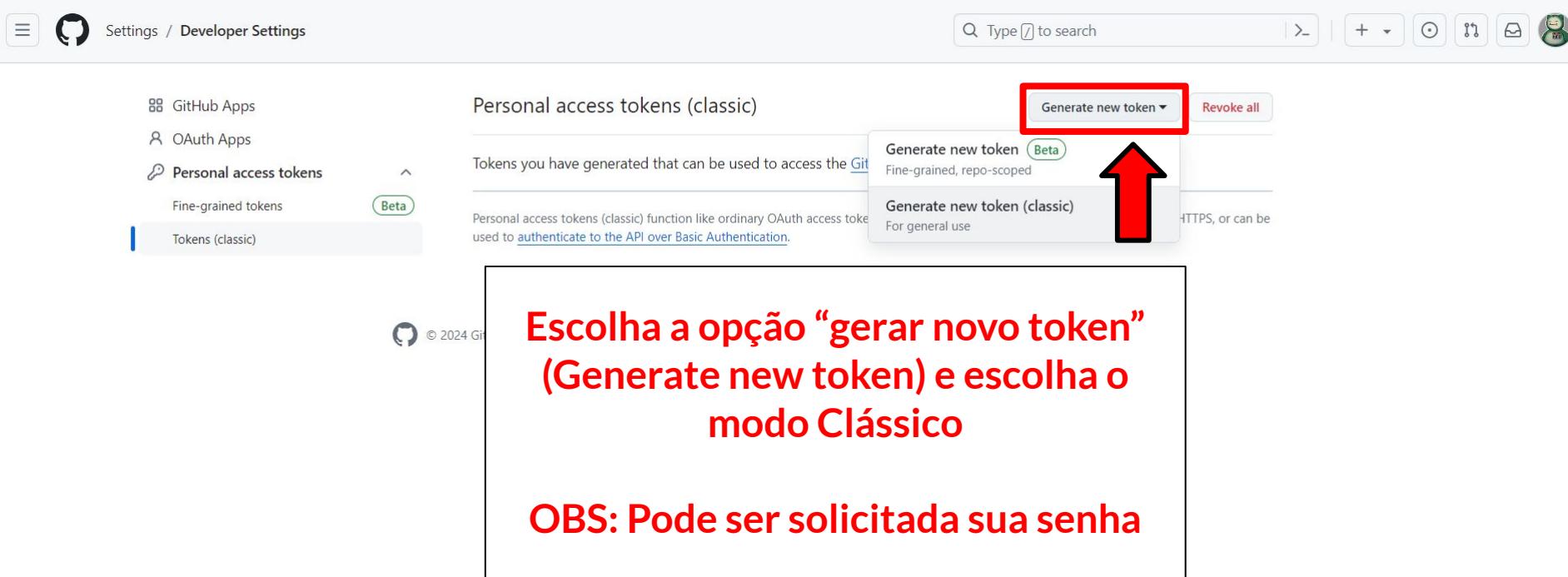
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

**Selecione a opção Tokens (Clássico)**

# Token

- Criando um token no GitHub



Personal access tokens (classic)

Generate new token Beta

Fine-grained, repo-scoped

Generate new token (classic)

For general use

HTTPS, or can be

Escolha a opção “gerar novo token” (Generate new token) e escolha o modo Clássico

OBS: Pode ser solicitada sua senha

# Token

- Criando um token no GitHub

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

What's this token for?

Na página de gerar novo token adicione uma nota identificando o token

Scopes	Description
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry

# Token

- Criando um token no GitHub

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

Token aula 27/06

What's this token for?

Expiration \*

Custom... 29/06/2024

Select scopes

Scopes define the access for personal tokens. [Read more about OA](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write/packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read/packages	Download packages from GitHub Package Registry

Escolha a data de expiração do token

# Token

- Criando um token no GitHub

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

Token aula 27/06

What's this token for?

Expiration \*

Custom... 29/06/2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

repo

Full control of private repositories

repo:status

Access commit status

repo\_deployment

Access deployment status

public\_repo

Access public repositories

repo:invite

Access repository invitations

security\_events

Read and write security events

workflow

Update GitHub Action workflows

write:packages

Upload packages to GitHub Package Registry

read:packages

Download packages from GitHub Package Registry

Marque a opção “repo” para permitir acesso ao repositório remoto

# Token

- Criando um token no GitHub

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

Token aula 27/06

What's this token for?

Expiration \*

Custom... 29/06/2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry

Ao final clique em  
gerar token

# Token

- Criando um token no GitHub

The screenshot shows the GitHub Developer Settings page under the 'Personal access tokens' section. A red arrow points to the 'token' field where a new token has been generated. The token itself is highlighted in red.

Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

GitHub Apps      OAuth Apps      **Personal access tokens**      Fine-grained tokens Beta      Tokens (classic)

Personal access tokens (classic)      Generate new token ▾      Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

token Delete

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

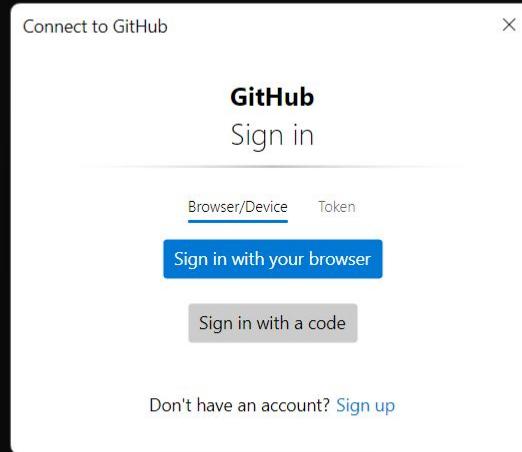
**Token será criado e o código é indicado pela seta vermelha**

**É altamente recomendado transmitir o token apenas para pessoas autorizadas a alterar o repositório**

# Voltando ao Git push ...

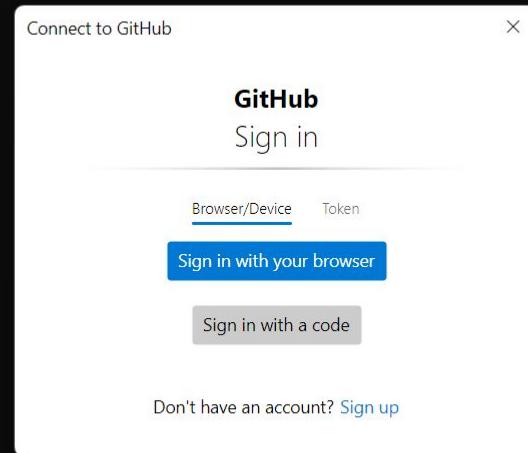
- Voltando ao push

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$ git push -u origin main
```



# Voltando ao Git push ...

```
● Voltando ao push  
bugo@Hugopc-MINGW64: ~/Repositorios_Git/NovoRepositorio (main)  
$ git push -u origin main
```



# Voltando ao Git push ...

Fechar

• Voltando ao push  
hugop@Hugopc MINGW64 ~/Repositorios\_Git/NovoRepositorio (main)  
\$ git push -u origin main



**Escolha a opção  
token e insira a  
moeda**

# Voltando ao Git push ...

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/NovoRepositorio (main)
$ git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 733 bytes | 733.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/hpbtakiuchi/Aula2706.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```



Push executado com sucesso

# Repositório remoto

**Comandos iniciais completos:**

```
echo "# seurepositorio" >> README.md
```

```
git init
```

```
git config user.email "nome@example.com"
```

```
git config user.nome "Nome"
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/seulogin/seurepositorio.git
```

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

**Comandos iniciais  
feitos** 

**Acesse o repositório  
remoto**

# Repositório remoto

## ● Repositório Aula2706

The screenshot shows a GitHub repository page for 'Aula2706'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository name 'hpbtakiuchi / Aula2706' is visible, along with a search bar and various icons for pinning, unwatching, forks, and stars.

The main content area displays the repository details:

- Branches:** main (selected), 1 Branch, 0 Tags
- Commits:**
  - Hugo commit README (e079d94 · 36 minutes ago)
  - README.md commit README (36 minutes ago)
  - arquivo.txt Primeiro Commit (8 hours ago)
- README:** A red underline is under the word 'README'.

**About:**

- No description, website, or topics provided.
- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks

**Releases:**

- No releases published
- [Create a new release](#)

**Packages:**

- No packages published
- [Publish your first package](#)

# Repositório remoto

## • Repositório Aula2706

hpbtakiuchi / Aula2706

Type  to search | [X](#) | [+](#) [-](#) [○](#) [□](#) [✉](#) [⚙️](#)

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

**Aula2706** Public

[Pin](#) [Unwatch](#) 1 [Fork](#) 0 [Star](#) 0

[main](#) [1 Branch](#) [0 Tags](#)

[Go to file](#) [Add file](#) [Code](#)

**Hugo commit README** e079d94 · 36 minutes ago [3 Commits](#)

<a href="#">README.md</a>	commit README	36 minutes ago
<a href="#">arquivo.txt</a>	Primeiro Commit	8 hours ago

[Readme](#) [Activity](#) [0 stars](#) [1 watching](#) [0 forks](#)

**About**

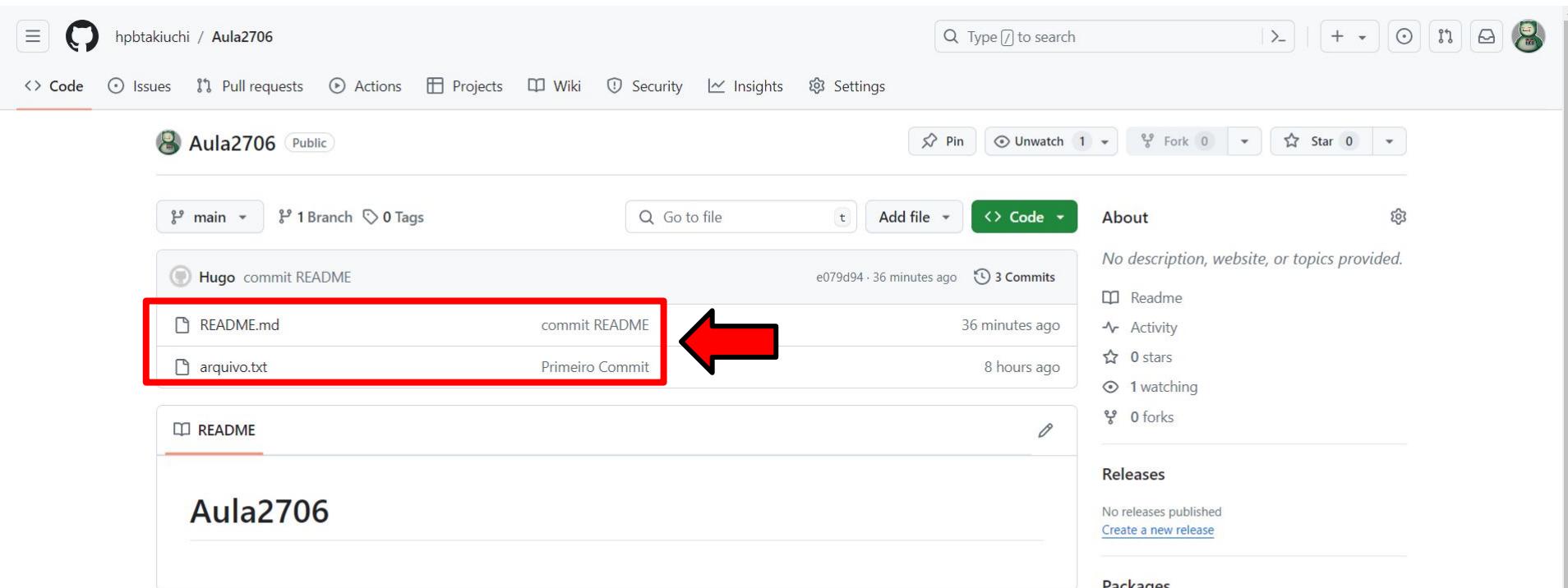
No description, website, or topics provided.

**Releases**

No releases published [Create a new release](#)

**Packages**

No packages published [Publish your first package](#)



README

Aula2706

# Trabalho individual

---

- Com os comandos apresentados até aqui é possível trabalhar individualmente
  - Etapas apresentadas:

- 1) Criar e alterar o nome de um repositório local Git
- 2) Criar, alterar e commitar um arquivo
- 3) Criar um repositório remoto GitHub
- 4) Criar um token de acesso clássico
- 5) Linkar o repositório remoto ao local
- 6) Enviar as alterações ao servidor remoto



# Conteúdo da aula

- Introdução
- Git e GitHub
- Comandos e conceitos
- Criando repositório individual
- **Criando repositório compartilhado**
- Conclusão
- Exercícios propostos
- Referências bibliográficas

# Trabalho em grupo

---

Como usar o Git e o GitHub para codar em  
**grupo?**

**OBS:** Para apresentação do conteúdo partimos do ponto de partida que já existe um trabalho remoto individual feito

# Colaboradores

---

- É possível declarar colaboradores para auxiliar no desenvolvimento dos arquivos no repositório remoto
  - Alterar no GitHub
  - O colaborador deve ter uma conta no GitHub
  - Executar trabalhos de maneira remota
- **OBS:** Talvez seja necessário um token de acesso durante todo o processo compartilhado

# Colaboradores

- Inserindo colaboradores no repositório

A screenshot of a GitHub repository page for 'Aula2706'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A large red arrow points upwards from a text box containing the Portuguese text 'Acessar as configurações do repositório' towards the 'Settings' button. The text box is positioned over the repository's commit history and file list. The repository has one branch ('main') and one tag ('0 Tags'). The commit history shows a single commit by 'Hugo' that adds a 'README' file. The repository details section on the right shows it is public, has no description, and has 0 stars, 1 watching, and 0 forks. It also indicates no releases or packages have been published.

Acessar as configurações do repositório

hpbtakiuchi / Aula2706

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Aula2706 Public

main 1 Branch 0 Tags

Hugo commit README

README.md commit README

arquivo.txt Primeiro Commit

README

Aula2706

Type to search

Pin Unwatch 1 Fork 0 Star 0

About

No description, website, or topics provided.

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

# Colaboradores

- Inserindo colaboradores no repositório

Acesse a opção Colaboradores (Collaborators)

The screenshot shows the GitHub repository settings page for 'hpbtakiuchi / Aula2706'. The 'Settings' tab is selected. On the left, the 'Access' section is expanded, showing three options: 'Collaborators' (which is highlighted with a red box and has a red arrow pointing to it), 'Moderation options', and 'Code and automation'. Below this, other sections like 'Branches', 'Tags', 'Rules', 'Actions', 'Webhooks', 'Environments', 'Codespaces', and 'Pages' are listed. The main content area is titled 'General' and contains sections for 'Repository name', 'Access', 'Default branch', 'Social preview', and 'Deploy keys'. A large red box highlights the 'Collaborators' section in the sidebar.

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Type  to search

+

Repository name

name

Repositories with the same directory structure and files. [Learn more about template repositories](#).

Based commits

Sign off on commits made through GitHub's web interface. Signing off is a way for you to verify that your code complies with the repository's terms, commonly the [Developer Certificate of Origin \(DCO\)](#). [Learn more about signing off on commits](#).

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

main

Social preview

Upload an image to customize your repository's social media preview.

<https://github.com/hpbtakiuchi/Aula2706/settings/access>

# Colaboradores

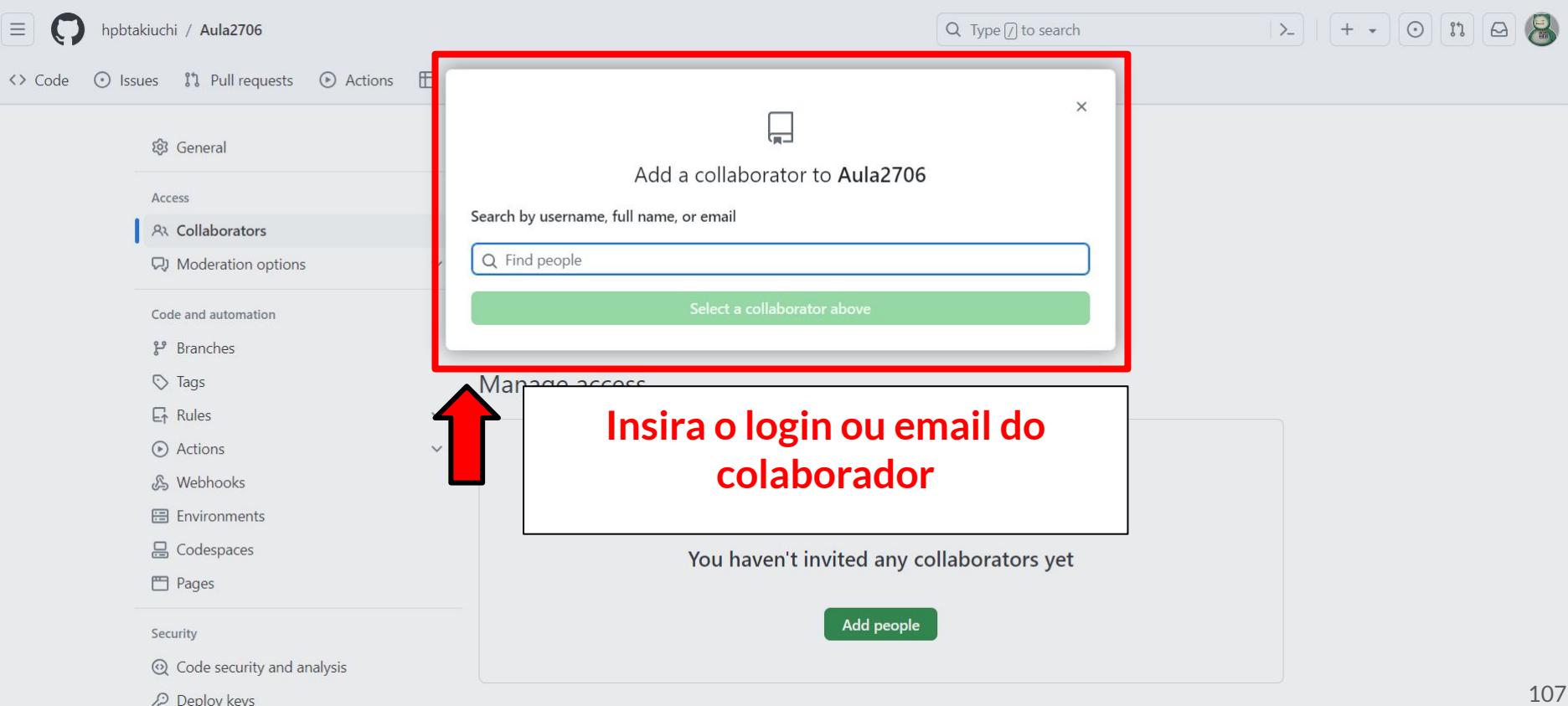
- Inserindo colaboradores no repositório

The screenshot shows the 'Who has access' section of a GitHub repository settings page. On the left sidebar, under the 'Access' heading, the 'Collaborators' option is selected, indicated by a blue vertical bar. The main area displays two sections: 'PUBLIC REPOSITORY' and 'DIRECT ACCESS'. The 'PUBLIC REPOSITORY' section states 'This repository is public and visible to anyone.' and includes a 'Manage' link. The 'DIRECT ACCESS' section states '0 collaborators have access to this repository. Only you can contribute to this repository.' A red box highlights the 'Add people' button at the bottom of the 'Manage access' section, which is also circled in red. A large red arrow points from the text 'selecionar a opção adicionar pessoas' to this button. A red box also surrounds the entire 'Manage access' section.

Nas configurações selecione a opção adicionar pessoas

# Colaboradores

- Inserindo colaboradores no repositório



A screenshot of a GitHub repository settings page. The left sidebar shows various repository management options like Code, Issues, Pull requests, Actions, General, Access, Collaborators (which is selected), Moderation options, Code and automation, Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, Pages, Security, Code security and analysis, and Deploy keys. A red box highlights the 'Find people' search bar in the 'Add a collaborator' modal. A large red arrow points upwards from a text box containing instructions to the search bar.

hpbtakiuchi / Aula2706

Type  to search

Code Issues Pull requests Actions

General

Access

**Collaborators**

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Add a collaborator to Aula2706

Search by username, full name, or email

Find people

Select a collaborator above

Insira o login ou email do colaborador

You haven't invited any collaborators yet

Add people

# Colaboradores

- Inserindo colaboradores no repositório

The screenshot shows the GitHub repository settings page for 'hpbtakiuchi / Aula2706'. The left sidebar has 'Collaborators' selected under 'Access'. The main area shows 'Who has access' with sections for 'PUBLIC REPOSITORY' and 'DIRECT ACCESS'. A red box highlights the 'Manage' link in the PUBLIC REPOSITORY section. A red arrow points from a text box above to this manage section. The text box contains the message: 'Os colaboradores podem ser gerenciados no menu abaixo'.

hpbonfimt has been added as a collaborator on the repository.

General

Access

**Collaborators**

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Type  to search

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

DIRECT ACCESS

1 has access to this repository. [0 collaborators](#). [1 invitation](#).

[Manage](#)

Manage access

Select all

Type

Find a collaborator...

hpbonfimt Awaiting hpbonfimt's response

**Os colaboradores podem ser gerenciados no menu abaixo**

108

# Colaboradores

## ● Inserindo colaboradores no repositório

Notifications

Inbox 2

All Unread Filter notifications

Clear out the clutter.

Get the most out of your new inbox by quickly and easily marking all of your previously read notifications as done.

Dismiss Get started

Conta do colaborador

Select all

Assigned

Participating 1

Mentioned

Team mentioned

Review requested

hpbtakiuchi/exemplo\_20\_0... 1

hpbtakiuchi/Aula2706 1

hpbonfimt/exemplo\_20\_06\_2

hpbonfimt/exemplo\_20\_06\_2 #2

hpbonfimt/exemplo\_20\_06\_2 #3

hpbonfimt/exemplo\_20\_06\_3 #2

hpbonfimt/exemplo\_20\_06\_3 #3

hpbonfimt/exemplo\_20\_06\_3 #4

hpbtakiuchi/exemplo\_20\_06\_3 #4

Invitation to join hpbtakiuchi/Aula2706 from hpbtakiuchi

Terceiro commit - Mara - Ramo 3

Segundo commit - José - Ramo1

Primeiro commit - José - Ramo1

Teste de oitavo commit

Primeiro commit - José

subscribed

+1 author

author

author

author

author

subscribed

2 minutes ago

last week

last week

last week

last week

last week

last week

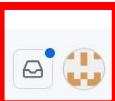
ProTip! When viewing a notification, press `shift u` to mark it as Unread.

1-6 of 6 Prev Next 109

# Colaboradores

## ● Inserindo colaboradores no repositório

Acesse as notificações



Notifications

Type  to search

Inbox 2 All Unread Filter notifications Group by: Date ▾

Saved

✓ Done

Clear out the clutter.

Get the most out of your new inbox by quickly and easily marking all of your previously read notifications as done.

Dismiss Get started

Filters

Assigned

Participating 1

Mentioned

Team mentioned

Review requested

hpbtakiuchi/exemplo\_20\_0... 1

hpbtakiuchi/Aula2706 1

hpbonfimt/exemplo\_20\_06\_2

Select all

- hpbtakiuchi/Aula2706 #4 Invitation to join hpbtakiuchi/Aula2706 from hpbtakiuchi subscribed 2 minutes ago
- hpbtakiuchi/exemplo\_20\_06\_3 #4 Terceiro commit - Mara - Ramo 3 +1 author last week
- hpbtakiuchi/exemplo\_20\_06\_3 #3 Segundo commit - José - Ramo1 author last week
- hpbtakiuchi/exemplo\_20\_06\_3 #2 Primeiro commit - José - Ramo1 author last week
- hpbonfimt/exemplo\_20\_06\_2 #2 Teste de oitavo commit author last week
- hpbonfimt/exemplo\_20\_06\_2 #1 Primeiro commit - José subscribed last week

ProTip! When viewing a notification, press **shift u** to mark it as Unread.

1-6 of 6 Prev Next 110

# Colaboradores

- Inserindo colaboradores no repositório

Notifications

Inbox 2

All Unread Filter notifications

Saved

Done

Clear out the clutter.

Get the most out of your new inbox by quickly and easily marking all of your previous notifications as done.

Select all

Assigned 1

Participating

Mentioned

Team mentioned

Review requested

hpbtakiuchi/exemplo\_20\_0... 1

hpbtakiuchi/Aula2706 1

hpbonfimt/exemplo\_20\_06\_2

Uma mensagem será enviada ao colaborador via GitHub e email

Dismiss Get started

Group by: Date

Subscriptions

Invitation to join hpbtakiuchi/Aula2706 from hpbtakiuchi subscribed 2 minutes ago

Terceiro commit - Mara - Ramo 3 +1 author last week

Segundo commit - José - Ramo1 author last week

Primeiro commit - José - Ramo1 author last week

Teste de oitavo commit author last week

Primeiro commit - José subscribed last week

ProTip! When viewing a notification, press shift u to mark it as Unread.

1-6 of 6 Prev Next

111

# Colaboradores

- Inserindo colaboradores no repositório

The screenshot shows a GitHub repository page for 'hpbtakiuchi / Aula2706'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. Below the navigation is a search bar with placeholder text 'Type to search' and various icons for filtering and searching. On the left, there's a sidebar with a 'Back to notifications' link. The main content area displays a collaboration invitation from 'hpbtakiuchi' to collaborate on the repository. The invitation includes two user icons (one green, one orange) and two buttons: 'Accept invitation' (green) and 'Decline' (gray). Below the buttons, it states: 'Owners of Aula2706 will be able to see:' followed by a list of five items. At the bottom, there's a question 'Is this user sending spam or malicious content?' with a 'Block hpbtakiuchi' link. A large red arrow points from a callout box on the left to the 'Accept invitation' button.

O colaborador decide sobre aceitar ou não o convite

→

hpbtakiuchi invited you to collaborate

[Accept invitation](#) [Decline](#)

Owners of Aula2706 will be able to see:

- Your public profile information
- Certain activity within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?

[Block hpbtakiuchi](#)

# Git clone

---

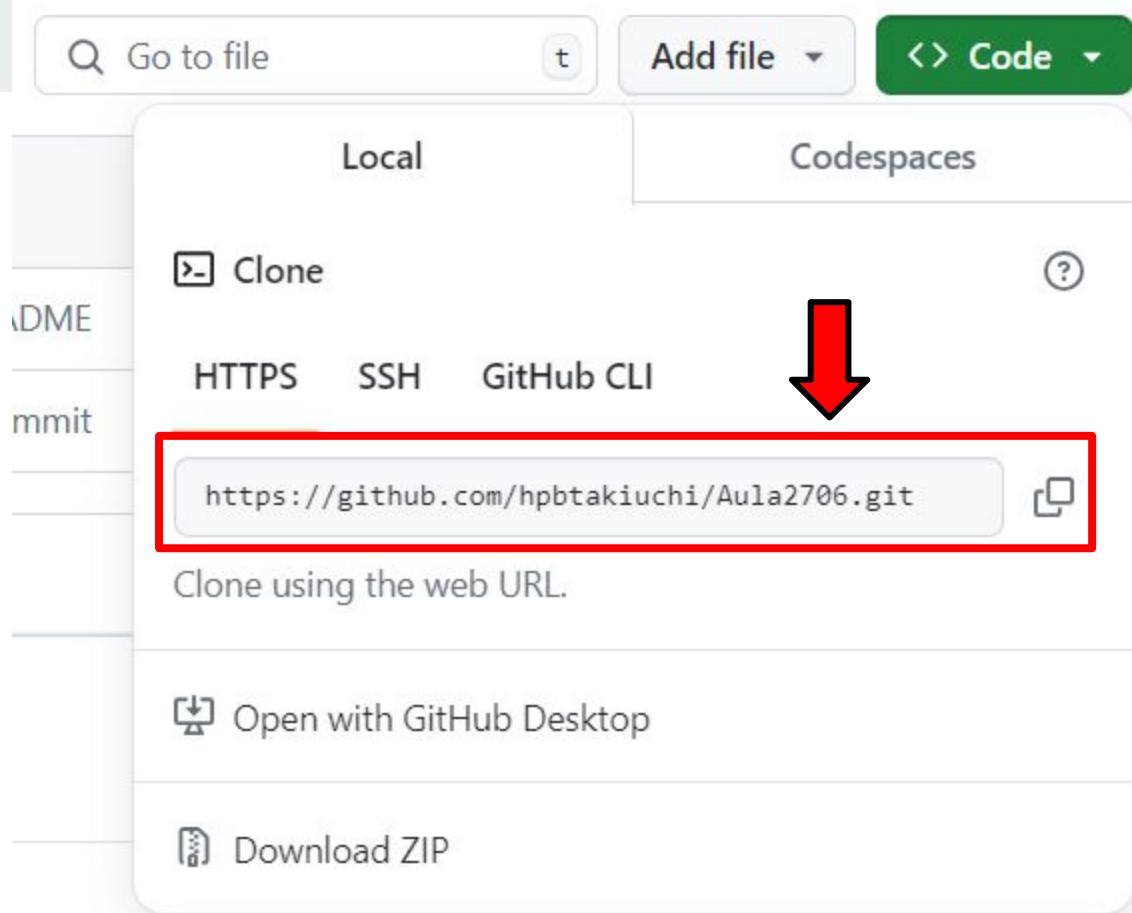
- Copia um repositório remoto do GitHub e cola na máquina local do usuário
  - Necessária a URL do repositório remoto
  - Pode ser executada em um diretório normal
  - Cria um repositório Git com o mesmo nome do repositório remoto dentro do diretório corrente

```
git clone https://github.com/seulogin/Aula2706.git
```

# Git clone

Localização da URL do repositório remoto no GitHub

**OBS 1:** Usamos na aula o protocolo HTTPS



# Git clone

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ cd ..
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git (master)
$ cd ..
```

```
hugop@Hugopc MINGW64 ~ (master)
$ cd Repositorios_Git2/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2 (master)
$ |
```



Diretório que irá receber o repositório Git

# Git clone

```
hugop@Hugopc MINGW64 ~ (master)
$ cd Repositorios_Git2/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2 (master)
$ git clone https://github.com/hpbtakiuchi/Aula2706.git
```



Clonando o repositório remoto

# Git clone

```
hugopc@Hugopc MINGW64 ~ (master)
$ cd Repositorios_Git2/
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2 (master)
$ git clone https://github.com/hpbtakiuchi/Aula2706.git
Cloning into 'Aula2706'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 0), reused 9 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), done.
```

Clonagem  
executada com  
sucesso

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2 (master)
$ cd Aula2706/
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ |
```

# Git clone

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ ls -l
total 2
-rw-r--r-- 1 hugop 197609 24 jul  2 12:24 arquivo.txt
-rw-r--r-- 1 hugop 197609 12 jul  2 12:24 README.md
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ |
```

# Colaboradores

---

- Após a execução do Git Clone o colaborador pode começar a alterar/criar arquivos
  - Necessário o uso dos comandos ADD e COMMIT para efetivar as mudanças
  - PUSH pode ser executado sem declarar o servidor remoto neste contexto

# Git clone

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ ls -l
total 2
-rw-r--r-- 1 hugop 197609 24 jul  2 12:24 arquivo.txt
-rw-r--r-- 1 hugop 197609 12 jul  2 12:24 README.md
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ vim programa1.java
```

Colaborador altera localmente

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ |
```

# Git clone

```
public class programa1 {  
    ...  
}
```

Conceúdo do programa

# Git clone

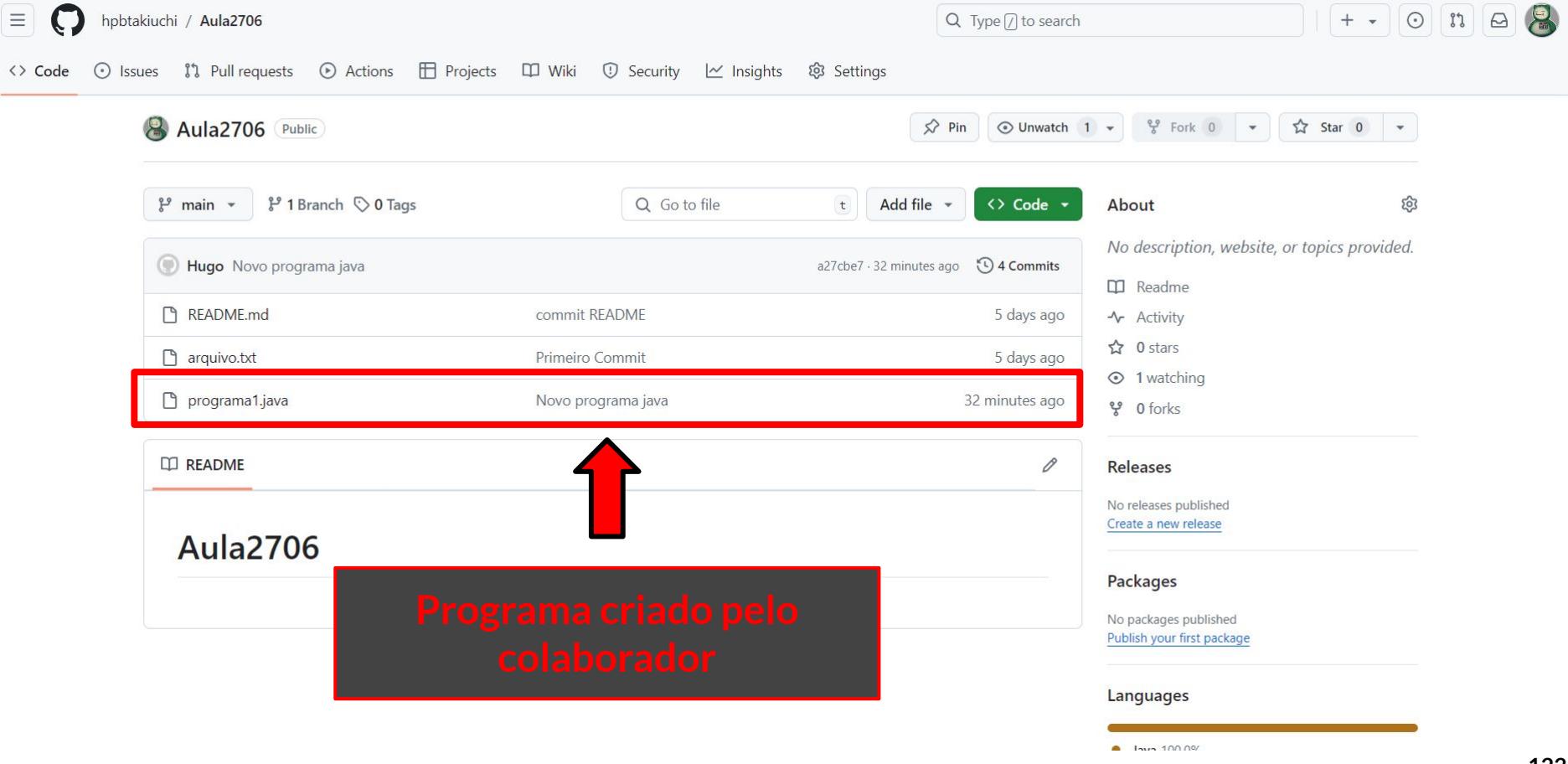
```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git add .
warning: in the working copy of 'programa1.java', LF will be replaced by CRLF
the next time Git touches it
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git commit -m "Novo programa java"
[main a27cbe7] Novo programa java
 1 file changed, 3 insertions(+)
 create mode 100644 programa1.java
```

Alteração local enviada ao  
repositório do proprietário

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 350 bytes | 350.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
```

# Git clone



## Git pull

---

- Comando usado para obter o repositório remoto atualizado
  - A declaração do servidor remoto pode ser omitida se já apresentada no comando PUSH
  -

# Git pull

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 330 bytes | 33.00 KiB/s, done.
From https://github.com/hpbtakiuchi/Aula2706
  e079d94..a27cbe7  main      -> origin/main
Updating e079d94..a27cbe7
Fast-forward
  programal.java | 3 ++
  1 file changed, 3 insertions(+)
  create mode 100644 programal.java
```

Repositório do proprietário



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ |
```

# Git pull

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 330 bytes | 33.00 KiB/s, done.
From https://github.com/hpbtakiuchi/Aula2706
  e079d94..a27cbe7  main      -> origin/main
Updating e079d94..a27cbe7
Fast-forward
  programal.java | 3 ++
  1 file changed, 3 insertions(+)
  create mode 100644 programal.java
```

Proprietário obtém o repositório atualizado



```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ |
```

# Git pull

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ ls -l
total 3
-rw-r--r-- 1 hugop 197609 23 jun 27 16:22 arquivo.txt
-rw-r--r-- 1 hugop 197609 35 jul  2 13:40 programa1.java
-rw-r--r-- 1 hugop 197609 11 jun 27 23:54 README.md
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ |
```

# Trabalho em grupo

---

- Alguns erros comuns
  - Executar um push antes de obter o repositório remoto atualizado

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git push
To https://github.com/hpbtakiuchi/Aula2706.git
 ! [rejected]      main -> main (fetch first)
error: failed to push some refs to 'https://github.com/hpbtakiuchi/Aula2706.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

# Trabalho em grupo

---

- Alguns erros comuns
  - URL incorreta pode ocasionar no seguinte erro:

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2806 (main)
$ git push -u origin main
fatal: protocol '^?https' is not supported
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2806 (main)
$ git remote set-url origin https://github.com/hpbtakiuchi/Aula2806.git
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2806 (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
```

# Trabalho em grupo

---

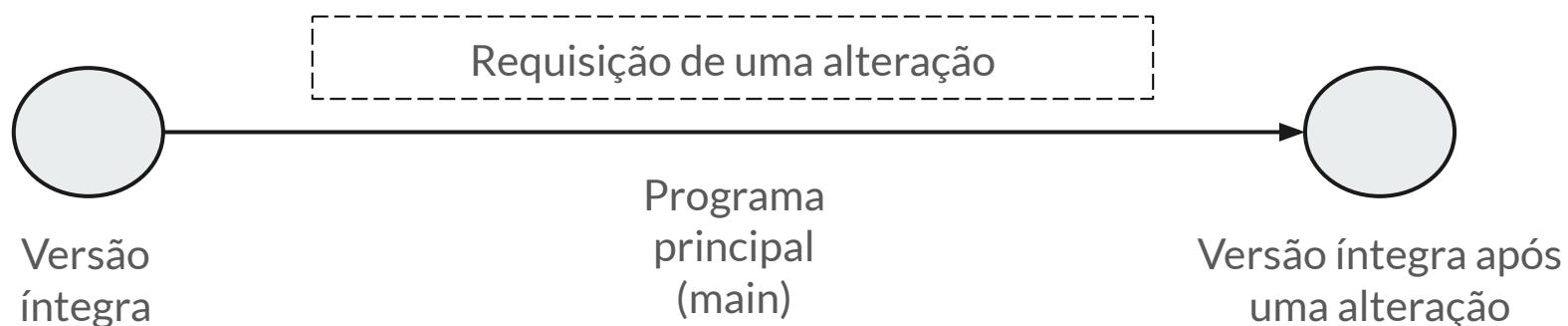
- **Branches (Ramos)**

**Definição:** São ramificações do repositório. Eles permitem que o programa continue executando enquanto alguma alteração deve ser efetuada no código. É uma forma de proteger a integridade do código. As alterações são executadas em “sandboxes”

# Trabalho em grupo

---

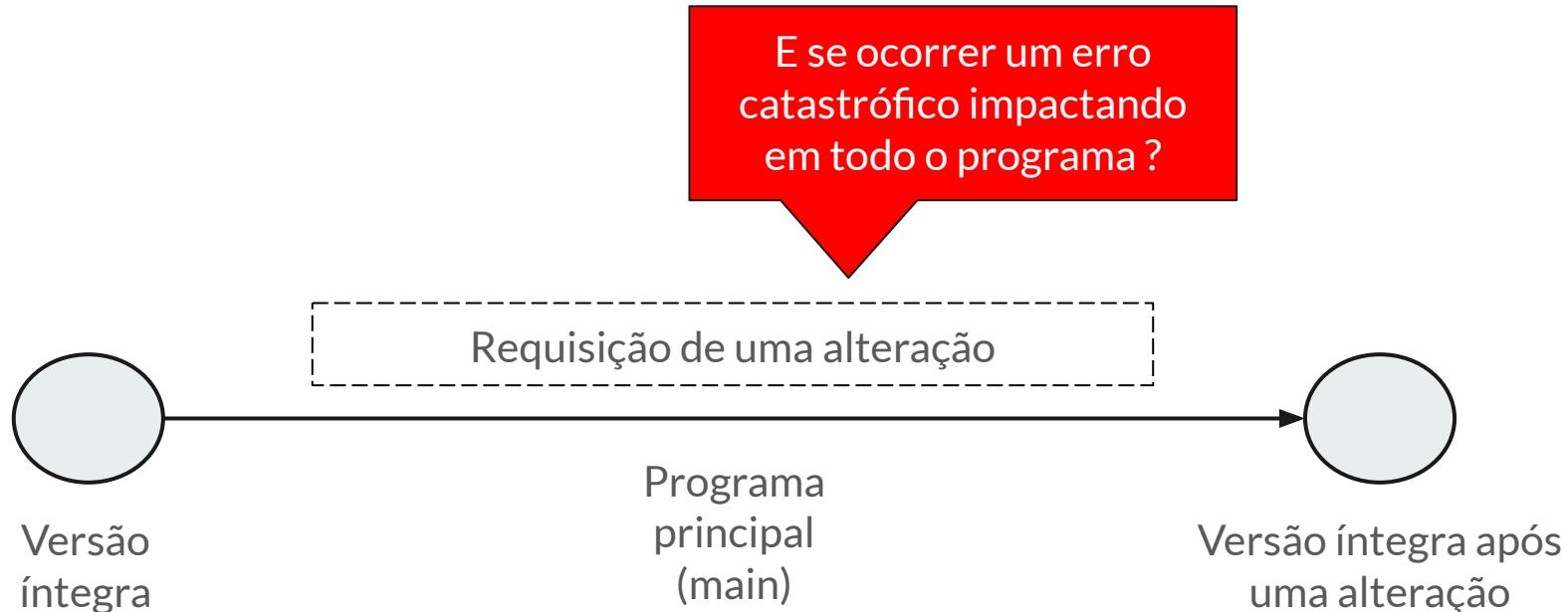
- **Branches (Ramos)**



# Trabalho em grupo

---

- **Branches (Ramos)**

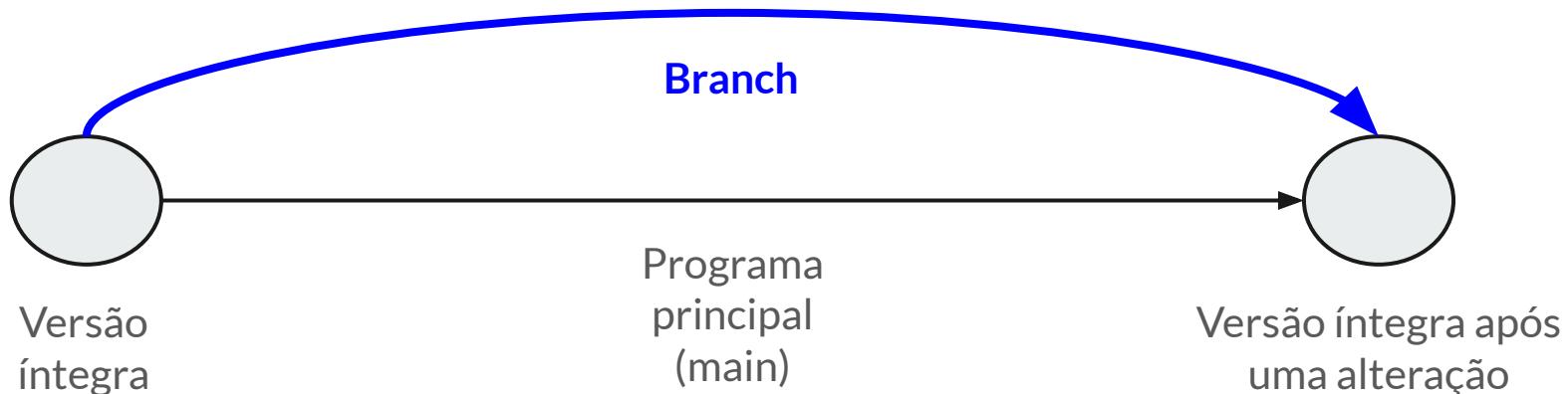


# Trabalho em grupo

---

- **Branches (Ramos)**

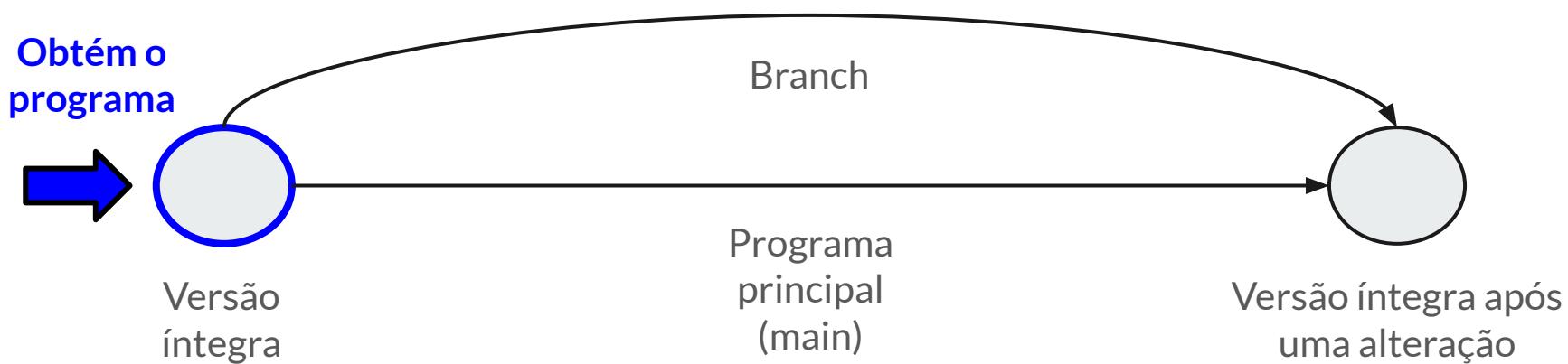
Uma solução é criar uma área segura para implementar as alterações



# Trabalho em grupo

---

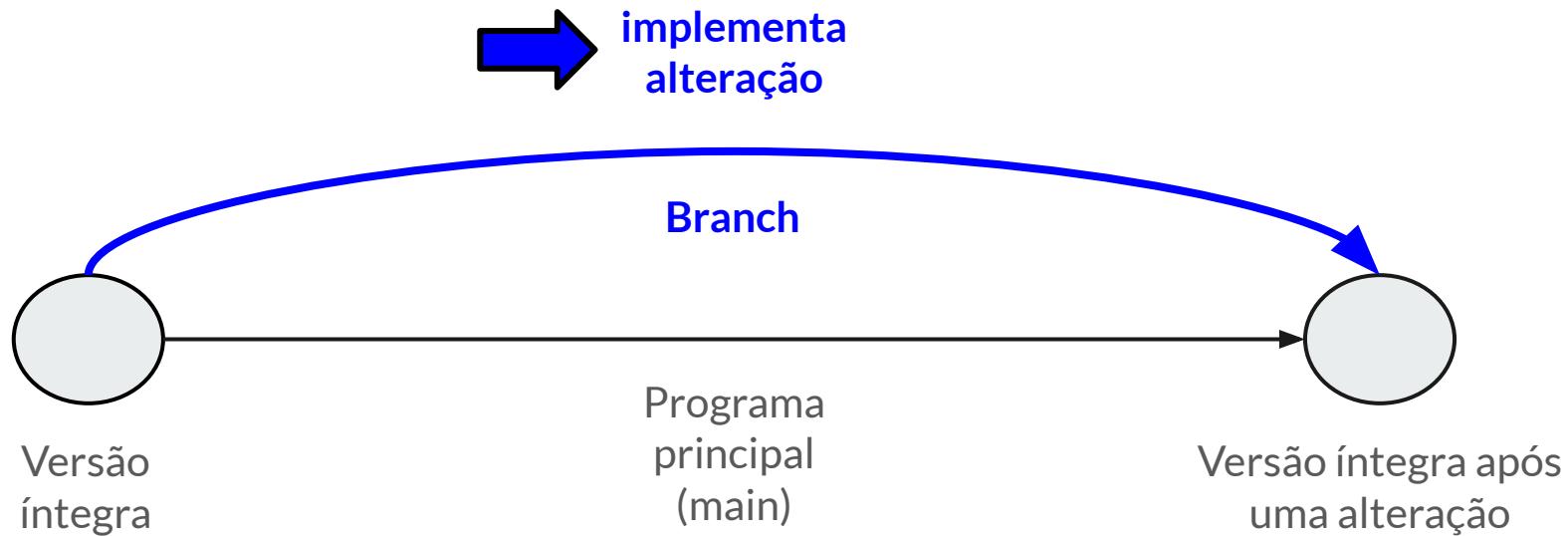
- **Branches (Ramos)**



# Trabalho em grupo

---

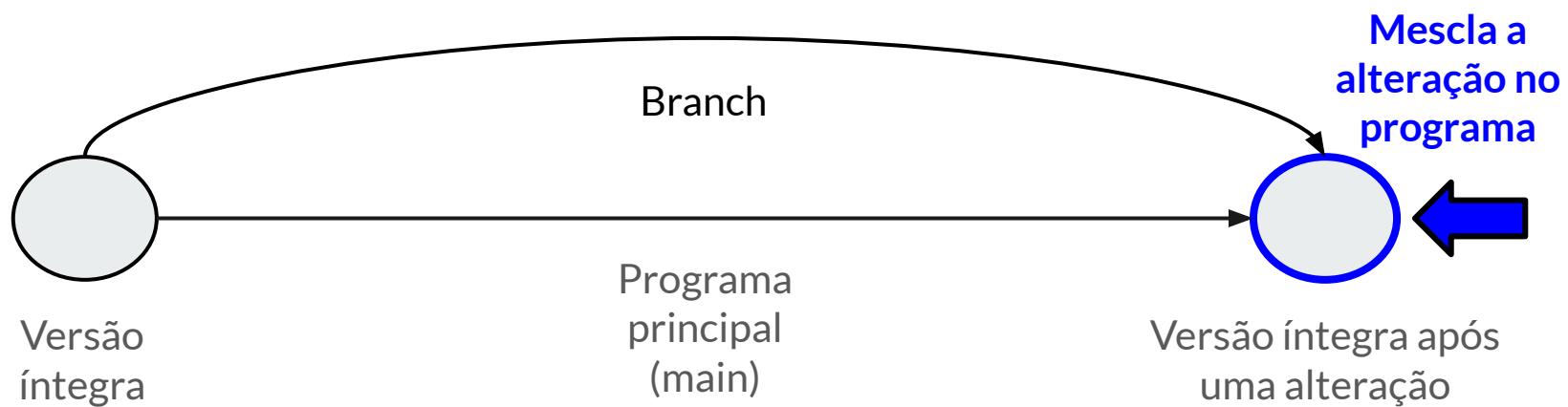
- **Branches (Ramos)**



# Trabalho em grupo

---

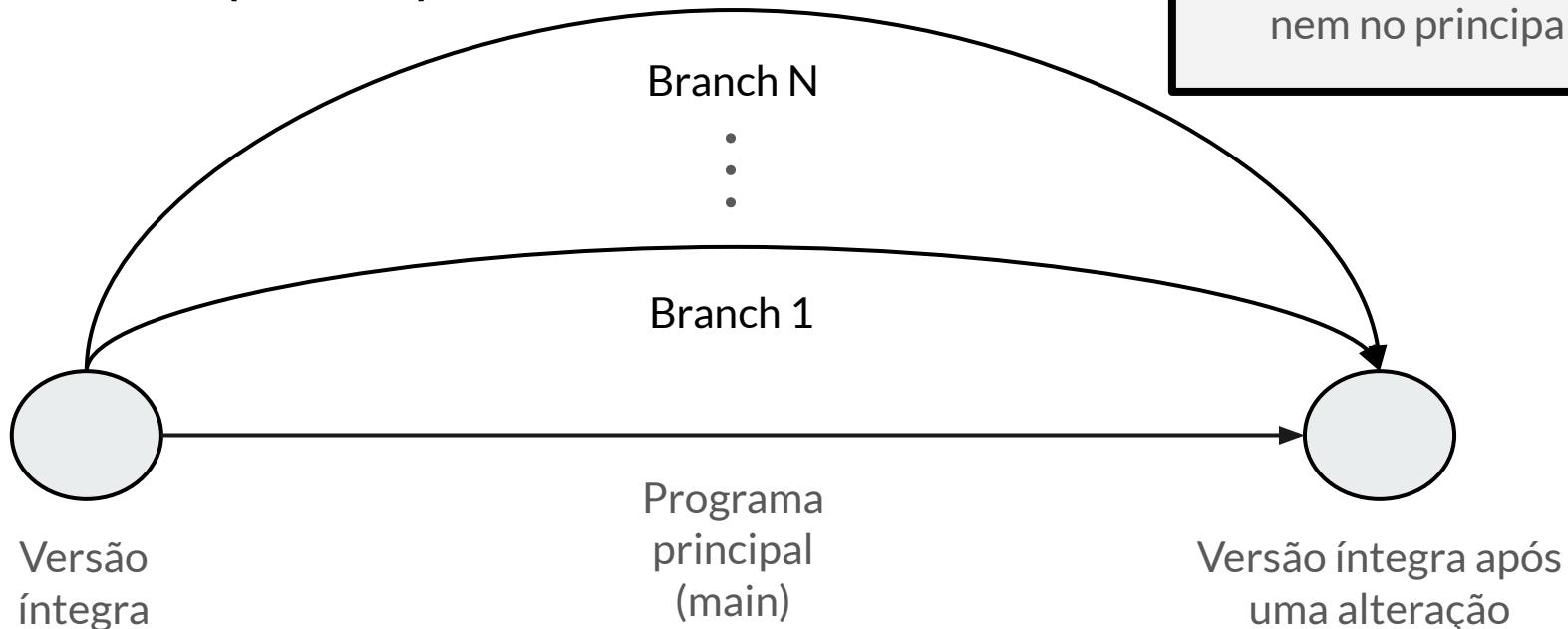
- **Branches (Ramos)**



# Trabalho em grupo

- **Branches (Ramos)**

É possível criar mais de um branch



**OBS:** As alterações, antes da mescla, em um Branch não impactam nos outros branches alternativos nem no principal

# Trabalho em grupo

---

- **Branches (Ramos)**
  - Algumas origens das alterações
    - Requisição do cliente
    - Bug ou erro
    - Necessidade de mercado
    - Otimização do código

# Trabalho em grupo

---

- **Branches (Ramos)**

- Alguns comandos para manipular Branches:



Comandos
Git Branch
Git Checkout
Git Merge

- Ramos podem ser usados tanto localmente quanto remotamente

# Git Branch

---

- Comando usado para criar um novo branch
  - Formato: `git branch nomeRamo`
  - A declaração da opção “-M” altera o nome do branch
  - A declaração da opção “-d” remove um branch
  - Sem nenhuma opção lista todos os branches
    - O que contém asteriscos é o ramo corrente
  - **OBS:** Antes de criar o branch é recomendável dar commit nos arquivos.  
(Fonte: Apostila do professor Evandro)

# Git Branch

```
hugop@Hugopc MINGW64
$ git branch
* main
```

Lista todos os branches



```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch rammmol
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
    rammmol
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch -d rammmol
Deleted branch rammmol (was a27cbe7).
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$
```

# Git Branch

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
```

```
hugop@Hugopc MINGW64 ~/Repos: $ git branch rammmol
```



Cria um ramo nomeado “rammmo1”

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
    rammmol
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch -d rammmol
Deleted branch rammmol (was a27cbe7).
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ |
```

# Git Branch

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch rammmol
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
    rammmol
```



Ramo corrente main

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch -d rammmol
Deleted branch rammmol (was a27cbe7).
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$
```

# Git Branch

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch rammmol
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
    rammmol
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch -d rammmol
Deleted branch rammmol (was a27cbe7).
```

Remoção de um  
branch

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ |
```

# Git Branch

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (rammmo1)
$ git branch -M ramo1
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$ |
```

(rammmo1)



Ramo  
corrente:  
rammmo1

# Git Branch

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (rammmo1)
$ git branch -M ramo1 ← Altera o nome do ramo corrente
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$ |
```

# Git Branch

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (rammmo1)
$ git branch -M ramo1
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$ |
```

Ramo  
corrente:  
ramo1

# Git Checkout

---

- Comando usado para acessar um ramo específico
  - Formato:

```
git checkout ramo1
```

# Git Checkout

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
  ramo1
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git checkout ramo1
Switched to branch 'ramo1'
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 24 jul  2 12:24 arquivo.txt
-rw-r--r-- 1 hugop 197609 32 jul  2 12:35 programa1.java
-rw-r--r-- 1 hugop 197609 12 jul  2 12:24 README.md
-rw-r--r-- 1 hugop 197609  6 jul  2 16:35 teste.txt
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$ |
```

# Git Checkout

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
  ramo1
```

```
hugop@Huaopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git checkout ramo1
Switched to branch 'ramo1'
```

Acesso ao ramo alternativo

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 24 jul 2 12:24 arquivo.txt
-rw-r--r-- 1 hugop 197609 32 jul 2 12:35 programa1.java
-rw-r--r-- 1 hugop 197609 12 jul 2 12:24 README.md
-rw-r--r-- 1 hugop 197609 6 jul 2 16:35 teste.txt
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$
```

# Git Checkout

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
  ramo1
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git checkout ramo1
Switched to branch 'ramo1'
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 24 jul  2 12:24 arquivo.txt
-rw-r--r-- 1 hugop 197609 32 jul  2 12:35 programa1.java
-rw-r--r-- 1 hugop 197609 12 jul  2 12:24 README.md
-rw-r--r-- 1 hugop 197609  6 jul  2 16:35 teste.txt
```

Conteúdo

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$ |
```

# Git Checkout

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git branch
* main
  ramo1
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main) ←
$ git checkout ramo1
Switched to branch 'ramo1'
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1) ←
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 24 jul  2 12:24 arquivo.txt
-rw-r--r-- 1 hugop 197609 32 jul  2 12:35 programa1.java
-rw-r--r-- 1 hugop 197609 12 jul  2 12:24 README.md
-rw-r--r-- 1 hugop 197609  6 jul  2 16:35 teste.txt
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$ |
```

# Git merge

---

- Comando que combina as alterações realizados no ramo alternativo com o ramo principal
  - Deve ser executado no ramo principal
  - formato:

```
git merge ramo1
```

# Git merge

---

- Dentro do ramo são executadas as alterações
  - Toda alteração deve ser rastreada e “commitada”
    - Git add
    - Git commit
  - Sem uso do commit as alterações no ramo alternativo podem ser propagadas para o ramo principal

# Git merge

Fechar

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
```

```
$ git branch  
* main  
  ramo1
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
```

```
$ git checkout ramo1  
Switched to branch 'ramo1'
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
```

```
$ ls -l  
total 4  
-rw-r--r-- 1 hugop 197609 23 jun 27 16:22 arquivo.txt  
-rw-r--r-- 1 hugop 197609 45 jul 2 17:28 programa1.java  
-rw-r--r-- 1 hugop 197609 32 jul 2 13:49 programa2.java  
-rw-r--r-- 1 hugop 197609 11 jun 27 23:54 README.md
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
```

# Git merge

Fechar

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ git branch
* main
  ramo1
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ git checkout ramo1
Switched to branch 'ramo1'
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 23 jun 27 16:22 arquivo.txt
-rw-r--r-- 1 hugop 197609 45 jul  2 17:28 programa1.java
-rw-r--r-- 1 hugop 197609 32 jul  2 13:49 programa2.java
-rw-r--r-- 1 hugop 197609 11 jun 27 23:54 README.md
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
```

# Git merge

Fechar

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ git branch
* main
  ramo1
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ git checkout ramo1
Switched to branch 'ramo1'
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 23 jun 27 16:22 arquivo.txt
-rw-r--r-- 1 hugop 197609 45 jul  2 17:28 programa1.java
-rw-r--r-- 1 hugop 197609 32 jul  2 13:49 programa2.java
-rw-r--r-- 1 hugop 197609 11 jun 27 23:54 README.md
```

Alteração no  
arquivo  
programa1.  
java



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
```

# Git merge

```
public class programa1 {  
    ...  
    | //teste  
}
```

Estado atual do programa

# Git merge

```
public class programa1 {  
    ...  
    //Alteração realizada no ramo1  
}
```

Alteração executada

programa1.java [dos] (18:35 02/07/2024)  
"programa1.java" [dos] 4L, 70B

3,33-38 Tudo

# Git merge

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
$ git add .
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
$ git commit -m "alteração do programa1.java"
[ramo1 47b55ce] alteração do programa1.java
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
$ |
```

# Git merge

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
$ git add .
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
$ git commit -m "alteração do programa1.java"
[ramo1 47b55ce] alteração do programa1.java
 1 file changed, 1 insertion(+), 1 deletion(-)
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
$ |
```

# Git merge

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
```

```
$ git checkout main  
Switched to branch 'main'  
Your branch is ahead of 'origin/main' by 1 commit.  
(use "git push" to publish your local commits)
```



Mensagem de erro indicando que o repositório local possui alterações que o diretório remoto não possui ainda

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)  
$ vim programa1.java
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)  
$ git merge ramo1  
Updating 35483c3..47b55ce  
Fast-forward  
programa1.java | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)  
$ |
```

# Git merge

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ vim programa1.java
```



Arquivo permanece com o texto antigo

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ git merge ramo1
Updating 35483c3..47b55ce
Fast-forward
 programa1.java | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ |
```

# Git merge

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (ramo1)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ vim programa1.java
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ git merge ramo1
Updating 35483c3..47b55ce
Fast-forward
programa1.java | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```



Mescla dos ramos

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ |
```

# Git merge

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
```

```
$ git status  
On branch main  
Your branch is ahead of 'origin/main' by 2 commits.  
(use "git push" to publish your local commits)
```

```
nothing to commit, working tree clean
```

Problema apresentado anteriormente

```
hugop@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
```

```
$ git push  
Enumerating objects: 8, done.  
Counting objects: 100% (8/8), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (5/5), done.  
Writing objects: 100% (6/6), 590 bytes | 590.00 KiB/s, done.  
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (2/2), completed with 1 local object.  
To https://github.com/hpbtakiuchi/Aula2706.git
```

# Git merge

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git/Aula2706 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

```
hugopc@Hugopc MINGW64 ~/Repositorios Git/Aula2706 (main)
```

```
$ git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 590 bytes | 590.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/hpbtakiuchi/Aula2706.git
```

Git push resolve a  
sincronização entre local e  
remoto

# Git merge

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Aula2706 Public

main 1 Branch 0 Tags Go to file Add file Code

	Hugo	alteração do programa1.java
	README.md	commit README
	arquivo.txt	Primeiro Commit
	programa1.java	alteração do programa1.java
	programa2.java	Criação do programa2

About No description, website, or topics provided.

Readme Activity 0 stars 1 watching 0 forks

Releases No releases published Create a new release

Packages No packages published Publish your first package

# Pull Request

---

- Segurança no repositório é essencial para manter sua integridade
- Alguns problemas podem ocorrer:
  - **PUSH** de algum arquivo corrompido
  - **PUSH** malicioso e intencional
  - **PUSH** de erros não intencionais
- Necessária uma coordenação e autorização das alterações no repositório remoto

# Pull Request

---

- **Pull Request** são solicitações feita pelos colaboradores ao proprietário do repositório remoto (GitHub)
  - Feitas através do GitHub
  - Proprietário decide sobre a validade das alterações
  - Necessidade de criar **Pull Request Rules**
    - Regras para alteração do repositório
    - Criada pelo proprietário

# Trabalho em grupo

---

- **Passo a passo da codificação em grupo usando GitHub de maneira segura**
  - 1) Proprietário cria as regras de Pull Request
  - 2) Colaborador clona o repositório
  - 3) Colaborador cria um ramo e realiza as alterações
  - 4) Colaborador envia as alterações ao servidor e solicita um Pull Request
  - 5) Proprietário aceita/rejeita as alterações

# 1) Proprietário cria as regras de Pull Request

The screenshot shows a GitHub repository page for 'Aula2706'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A red box highlights the 'Settings' button, and a red arrow points from it to a callout bubble containing the text 'Busque as configurações' (Search for configurations). Below the navigation bar, the repository details show 'main' branch, 1 branch, 0 tags, and 7 commits by 'Hugo'. The commits are listed with their files, times, and descriptions. To the right, there are sections for About (no description), Activity (1 watching), Releases (no releases), Packages (no packages), and Languages (Java 100%).

hpbtakiuchi / Aula2706

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Aula2706 Public

main 1 Branch 0 Tags

Go to file Add file Code

Hugo alteração do programa1.java 47b55ce · 2 hours ago 7 Commits

README.md commit README 5 days ago

arquivo.txt Primeiro Commit 5 days ago

programa1.java alteração do programa1.java 2 hours ago

programa2.java Criação do programa2 6 hours ago

README

**Aula2706**

Type to search

Pin Unwatch

Busque as configurações

About

No description, website, or topics provided.

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Languages

Java 100%

## **1) Proprietário cria as regras de Pull Request**

## **1) Proprietário cria as regras de Pull Request**

The screenshot shows the GitHub 'Branch protection rules' page for a repository named 'hpbtakiuchi / Aula2706'. The left sidebar is open, showing sections like General, Access, Collaborators, and Moderation options under 'Code and automation'. The 'Branches' section is selected. The main content area displays a message: 'You haven't protected any of your branches' with links to 'Add branch ruleset' and 'Add classic branch protection rule'. A large red box highlights the 'Add classic branch protection rule' link, which is also accompanied by a red arrow pointing to a callout box containing the text 'Seleciona "Adicionar nova regra"'. The top navigation bar includes Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.

hpbtakiuchi / Aula2706

Type  to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

You haven't protected any of your branches

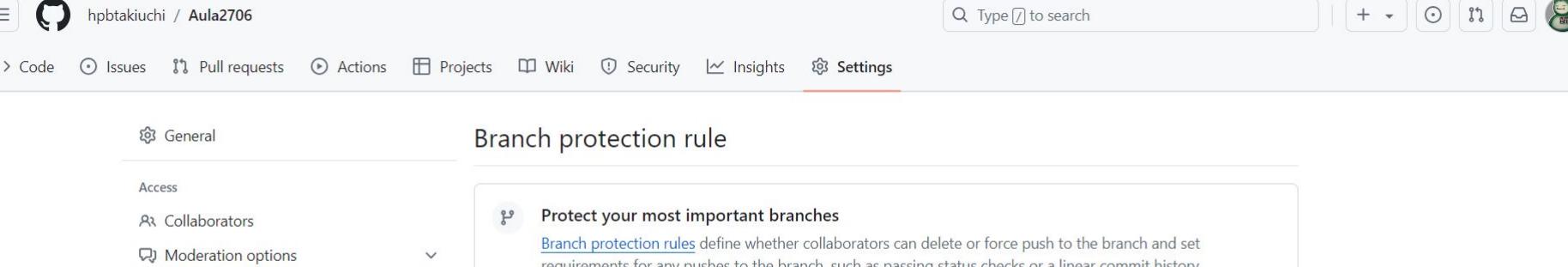
Define branch rules to disable force pushing, prevent branches from being deleted, or require pull requests. Learn more about [repository rules](#) and [protected branches](#).

Add branch ruleset Add classic branch protection rule

Seleciona "Adicionar nova regra"

## Selecione “Adicionar nova regra”

## **1) Proprietário cria as regras de Pull Request**



The screenshot shows the GitHub 'Branch protection rule' configuration page for a repository named 'hpbtakiuchi / Aula2706'. The left sidebar is open, showing sections like General, Access, Collaborators, Moderation options, Code and automation (with 'Branches' selected), Tags, Rules, Actions, Webhooks, Environments, Codespaces, Pages, Security, Code security and analysis, Deploy keys, and Secrets and variables. The main content area is titled 'Branch protection rule' and contains a section titled 'Protect your most important branches' with a note about GitHub Free plan limitations. Below this is a 'Branch name pattern \*' input field containing 'main', which is highlighted with a red box and a large red arrow pointing to it from the bottom right. The bottom part of the page shows settings for 'Protect matching branches' with two checked options: 'Require a pull request before merging' and 'Require approvals', both with explanatory text below them.

Insira o ramo “main” para indicar que a regra se aplica ao ramo principal

## **1) Proprietário cria as regras de Pull Request**

The screenshot shows the GitHub 'Branch protection rule' settings page for a repository named 'hpbtakiuchi / Aula2706'. The left sidebar is collapsed, showing sections like 'General', 'Access', 'Collaborators', 'Moderation options', 'Code and automation' (with 'Branches' selected), 'Tags', 'Rules', 'Actions', 'Webhooks', 'Environments', 'Codespaces', and 'Pages'. The main area is titled 'Branch protection rule' and contains a section titled 'Protect your most important branches' with a description of what branch protection rules do. Below this is a form for defining a 'Branch name pattern' (set to 'main') and a button 'Protect matching branches'. A large red box highlights the bottom section containing two checked checkboxes: 'Require a pull request before merging' and 'Require approvals'. Red text inside this box reads: 'Marque as opções como indicado abaixo:' followed by the instructions '“Solicitar pull request antes de mesclar”' and '“Solicitar aprovação”'.

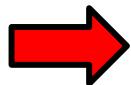
Marque as opções como indicado abaixo:

“Solicitar pull request antes de mesclar”

“Solicitar aprovação”

# 1) Proprietário cria as regras de Pull Request

Evita a criação e push de um ramo principal local sem a permissão do proprietário (Opcional)



- Require conversation resolution before merging**  
When enabled, all conversations on code must be resolved before a pull request can be merged into a branch that matches this rule.  
[Learn more about requiring conversation completion before merging.](#)
- Require signed commits**  
Commits pushed to matching branches must have verified signatures.
- Require linear history**  
Prevent merge commits from being pushed to matching branches.
- Require deployments to succeed before merging**  
Choose which environments must be successfully deployed to before branches can be merged into a branch that matches this rule.
- Lock branch**  
Branch is read-only. Users cannot push to the branch.
- Do not allow bypassing the above settings**  
The above settings will apply to administrators and custom roles with the "bypass branch protections" permission.

## Rules applied to everyone including administrators

- Allow force pushes**  
Permit force pushes for all users with push access.
- Allow deletions**  
Allow users with push access to delete matching branches.

**Create**

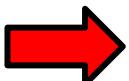
# 1) Proprietário cria as regras de Pull Request

- Require conversation resolution before merging**  
When enabled, all conversations on code must be resolved before a pull request can be merged into a branch that matches this rule.  
[Learn more about requiring conversation completion before merging.](#)
- Require signed commits**  
Commits pushed to matching branches must have verified signatures.
- Require linear history**  
Prevent merge commits from being pushed to matching branches.
- Require deployments to succeed before merging**  
Choose which environments must be successfully deployed to before branches can be merged into a branch that matches this rule.
- Lock branch**  
Branch is read-only. Users cannot push to the branch.
- Do not allow bypassing the above settings**  
The above settings will apply to administrators and custom roles with the "bypass branch protections" permission.

## Rules applied to everyone including administrators

- Allow force pushes**  
Permit force pushes for all users with push access.
- Allow deletions**  
Allow users with push access to delete matching branches.

Selezione criar regra



Create

## **1) Proprietário cria as regras de Pull Request**

A screenshot of a GitHub repository's settings page. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The Settings tab is active. A message at the top says "Branch protection rule created." On the left, a sidebar lists General, Access, Collaborators, Moderation options, Code and automation (with Branches selected), Tags, Rules, Actions, Webhooks, Environments, Codespaces, and Pages. The main content area shows a "Branch protection rules" section with a red box highlighting the "main" rule. A red arrow points upwards from a dark gray box containing the text "Regras criadas" (Rules created) towards the highlighted rule.

# Trabalho em grupo

---

- Passo a passo da codificação em grupo usando GitHub de maneira segura
  - 1) Proprietário cria as regras de Pull Request
  - **2) Colaborador clona o repositório**
  - **3) Colaborador cria um ramo e realiza as alterações**
  - 4) Colaborador envia as alterações ao servidor e solicita um Pull Request
  - 5) Proprietário aceita/rejeita as alterações

## 2) Colaborador clona o repositório

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4 (master)
$ git clone https://github.com/hpbtakiuchi/Aula2706.git
Cloning into 'Aula2706'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 21 (delta 4), reused 19 (delta 2), pack-reused 0
Receiving objects: 100% (21/21), done.
Resolving deltas: 100% (4/4), done.
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git4 (master)
$ ls -l
total 4
drwxr-xr-x 1 hugop 197609 0 jul  2 21:30 Aula2706/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4 (master)
$ cd Aula2706/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ |
```

## 2) Colaborador clona o repositório

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4 (master)
$ git clone https://github.com/hpbtakiuchi/Aula2706.git
Cloning into 'Aula2706'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 21 (delta 4), reused 19 (delta 2), pack-reused 0
Receiving objects: 100% (21/21), done.
Resolving deltas: 100% (4/4), done.
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4 (master)
$ ls -l
total 4
drwxr-xr-x 1 hugop 197609 0 jul  2 21:30 Aula2706/
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git4 (master)
$ cd Aula2706/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ |
```

## 2) Colaborador clona o repositório

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4 (master)
$ git clone https://github.com/hpbtakiuchi/Aula2706.git
Cloning into 'Aula2706'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 21 (delta 4), reused 19 (delta 2), pack-reused 0
Receiving objects: 100% (21/21), done.
Resolving deltas: 100% (4/4), done.
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4 (master)
$ ls -l
total 4
drwxr-xr-x 1 hugop 197609 0 jul  2 21:30 Aula2706/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4 (master)
$ cd Aula2706/
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ |
```

### 3) Colaborador cria um ramo e realiza as alterações

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 24 jul  2 21:30 arquivo.txt
-rw-r--r-- 1 hugop 197609 70 jul  2 21:30 programa1.java
-rw-r--r-- 1 hugop 197609 35 jul  2 21:30 programa2.java
-rw-r--r-- 1 hugop 197609 12 jul  2 21:30 README.md
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ git branch ramo1
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ git checkout ramo1
Switched to branch 'ramo1'
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ |
```

### 3) Colaborador cria um ramo e realiza as alterações

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 24 jul  2 21:30 arquivo.txt
-rw-r--r-- 1 hugop 197609 70 jul  2 21:30 programa1.java
-rw-r--r-- 1 hugop 197609 35 jul  2 21:30 programa2.java
-rw-r--r-- 1 hugop 197609 12 jul  2 21:30 README.md
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ git branch ramo1
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ git checkout ramo1
Switched to branch 'ramo1'
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ |
```

### 3) Colaborador cria um ramo e realiza as alterações

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 24 jul  2 21:30 arquivo.txt
-rw-r--r-- 1 hugop 197609 70 jul  2 21:30 programa1.java
-rw-r--r-- 1 hugop 197609 35 jul  2 21:30 programa2.java
-rw-r--r-- 1 hugop 197609 12 jul  2 21:30 README.md
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ git branch ramo1
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ git checkout ramo1
Switched to branch 'ramo1'
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ |
```

### 3) Colaborador cria um ramo e realiza as alterações

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 24 jul  2 21:30 arquivo.txt
-rw-r--r-- 1 hugop 197609 70 jul  2 21:30 programa1.java
-rw-r--r-- 1 hugop 197609 35 jul  2 21:30 programa2.java
-rw-r--r-- 1 hugop 197609 12 jul  2 21:30 README.md
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ git branch ramo1
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (main)
$ git checkout ramo1
Switched to branch 'ramo1'
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ |
```

### 3) Colaborador cria um ramo e realiza as alterações

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 24 jul  2 21:30 arquivo.txt
-rw-r--r-- 1 hugop 197609 70 jul  2 21:30 programa1.java
-rw-r--r-- 1 hugop 197609 35 jul  2 21:30 programa2.java
-rw-r--r-- 1 hugop 197609 12 jul  2 21:30 README.md
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ vim programa3.java
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ ls -l
total 5
-rw-r--r-- 1 hugop 197609 24 jul  2 21:30 arquivo.txt
-rw-r--r-- 1 hugop 197609 70 jul  2 21:30 programa1.java
-rw-r--r-- 1 hugop 197609 35 jul  2 21:30 programa2.java
-rw-r--r-- 1 hugop 197609 44 jul  2 21:36 programa3.java
-rw-r--r-- 1 hugop 197609 12 jul  2 21:30 README.md
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
```

### 3) Colaborador cria um ramo e realiza as alterações

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ ls -l
total 4
-rw-r--r-- 1 hugop 197609 24 jul  2 21:30 arquivo.txt
-rw-r--r-- 1 hugop 197609 70 jul  2 21:30 programa1.java
-rw-r--r-- 1 hugop 197609 35 jul  2 21:30 programa2.java
-rw-r--r-- 1 hugop 197609 12 jul  2 21:30 README.md
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ vim programa3.java
```

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ ls -l
total 5
-rw-r--r-- 1 hugop 197609 24 jul  2 21:30 arquivo.txt
-rw-r--r-- 1 hugop 197609 70 jul  2 21:30 programa1.java
-rw-r--r-- 1 hugop 197609 35 jul  2 21:30 programa2.java
-rw-r--r-- 1 hugop 197609 44 jul  2 21:36 programa3.java
-rw-r--r-- 1 hugop 197609 12 jul  2 21:30 README.md
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
```

# Trabalho em grupo

---

- **Passo a passo da codificação em grupo usando GitHub de maneira segura**
  - 1) Proprietário cria as regras de Pull Request
  - 2) Colaborador clona o repositório
  - 3) Colaborador cria um ramo e realiza as alterações
  - **4) Colaborador envia as alterações ao servidor e solicita um Pull Request**
  - 5) Proprietário aceita/rejeita as alterações

#### 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ git add .
warning: in the working copy of 'programa3.java', LF will be replaced by CRLF the next time Git touches it
```

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ git commit -m "Criação do programa 3"
> '
[ramo1 d5cc7ac] Criação do programa 3
 1 file changed, 3 insertions(+)
  create mode 100644 programa3.java
```



#### 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

```
hugopc@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ git add .
warning: in the working copy of 'programa3.java', LF will be replaced by CRLF the next time Git touches it
```

```
hugopc@Hugopc MTNGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ git commit -m "Criação do programa 3"
> '
[ramo1 d5cc7ac] Criação do programa 3
 1 file changed, 3 insertions(+)
  create mode 100644 programa3.java
```



#### 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$ git push -u origin ramo1
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 327 bytes | 327.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'ramo1' on GitHub by visiting:
remote:     https://github.com/hpbtakiuchi/Aula2706/pull/new/ramo1
remote:
To https://github.com/hpbtakiuchi/Aula2706.git
 * [new branch]      ramo1 -> ramo1
branch 'ramo1' set up to track 'origin/ramo1'.
```



```
hugop@Hugopc MINGW64 ~/Repositorios_Git4/Aula2706 (ramo1)
$
```

# 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

The screenshot shows a GitHub repository page for 'Aula2706'. The repository is public and has 7 commits. The commit history includes:

- commit README (5 days ago)
- Primeiro Commit (5 days ago)
- alteração do programa1.java (3 hours ago)
- Criação do programa2 (8 hours ago)

The repository has 1 watch, 0 forks, and 0 stars. It also has 1 watching and 0 forks.

**About**

No description, website, or topics provided.

**Branches**

- main (default)
- ramo1

**Tags**

**README**

**Aula2706**

**Releases**

No releases published  
[Create a new release](#)

**Packages**

No packages published  
[Publish your first package](#)

# 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

The screenshot shows a GitHub repository page for 'Aula2706'. The repository is public and has 7 commits. A modal dialog is open, titled 'Switch branches/tags', with a search bar containing 'Find or create a branch...'. It lists two branches: 'main' (selected) and 'ramo1'. Below the branches, there are links for 'View all branches' and 'README'. The main repository page displays a list of commits:

Commit Message	Date
commit README	5 days ago
Primeiro Commit	5 days ago
alteração do programa1.java	3 hours ago
Criação do programa2	8 hours ago

On the right side of the page, there are sections for 'About', 'Releases', and 'Packages'. The 'About' section notes 'No description, website, or topics provided.' The 'Releases' section says 'No releases published' and 'Create a new release'. The 'Packages' section is partially visible.

A large red box highlights the text 'GitHub do colaborador' at the bottom of the modal dialog.

#### 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

The screenshot shows a GitHub repository page for 'Aula2706'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights. The repository name 'Aula2706' is displayed, along with 'Public' status, 'Watch 1', 'Fork 0', and 'Star 0' buttons.

The main area shows a 'Switch branches/tags' dialog box with a red border. Inside the dialog, there is a search bar labeled 'Find or create a branch...', a 'Branches' tab selected, and a list of branches: 'main' (marked with a checkmark and 'default') and 'ramo1'. A large red arrow points from the right side of the dialog towards the main repository area.

The main repository area contains a dark gray box with the text: 'Criou um ramo alternativo com nome "ramo1"' in red. Below this, the commit history shows 'commit README' and 'alteração do pro...', followed by 'Criação do pro...'. At the bottom of the repository page, the text 'Aula2706' is visible.

On the right side of the page, there is an 'About' section with the following details:

- No description, website, or topics provided.
- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks
- Report repository

Below the 'About' section are sections for 'Releases' (No releases published, Create a new release) and 'Packages' (No packages published, Publish your first package).

#### 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

hpbtakiuchi / Aula2706

Code Issues Pull requests

Aula2706 Public

main 2 Branches 0 Tags

Switch branches/tags Find or create a branch...

Branches Tags

✓ main default

ramo1

View all branches

README

Aula2706

Próxima ação é criar um pull request

Type to search

Watch 1 Fork 0 Star 0

About

No description, website, or topics provided.

Readme Activity 0 stars 1 watching 0 forks Report repository

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

A GitHub repository page for 'Aula2706'. The 'Pull requests' tab is highlighted with a red box and a large red arrow pointing to it. A central callout box contains the text 'Próxima ação é criar um pull request'. The repository has 1 watch, 0 forks, and 0 stars. The 'About' section indicates no description, website, or topics are provided. The 'Releases' and 'Packages' sections show no activity. The commit history shows several commits, including 'commit README', 'Primeiro Commit', 'alteração do programa1.java', and 'Criação do programa2'.

#### 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

The screenshot shows a GitHub interface for a repository named 'hpbtakiuchi / Aula2706'. The 'Pull requests' tab is selected. A tooltip is displayed at the top center, reading 'Label issues and pull requests for new contributors' and 'Now, GitHub will help potential fi...'. A red box highlights the text 'Seleciona a opção de criar um novo pull request' (Select the option to create a new pull request). A large red arrow points from this box to a green button labeled 'New pull request'.

Label issues and pull requests for new contributors  
Now, GitHub will help potential fi...

Seleciona a opção de criar um novo pull request

New pull request

Filters ▾ Q is:pr is:open

Welcome to pull requests!

Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should [create a pull request](#).

ProTip! [no:milestone](#) will show everything without a milestone.

**4) Colaborador envia as alterações ao servidor e solicita um Pull Request**

Compare changes across branches, commits, tags, and more below. If you need to, you can also compare across forks.

base: main ▾ ← compare: main ▾

Choose different branches or forks above to discuss and review changes. [Learn about pull requests](#)

Create pull request

**4) Colaborador envia as alterações ao servidor e solicita um Pull Request**

The screenshot shows the GitHub 'Compare changes' interface. At the top, there are dropdown menus for 'base: main' and 'compare: main'. A modal window titled 'Choose a head ref' is open, showing a search bar 'Find a branch' and a list of branches. The 'main' branch is checked as the default. Below the modal, the text 'Compare and review' is visible. To the right of the modal, a large red box contains the instructions: 'Em “compare changes” selecione no componente a direita o ramo alternativo e na esquerda o ramo principal'. A red arrow points from this text towards the 'compare: main' dropdown.

Compare changes

Compare changes across branches, commits, tags, and more below. If you need to, you can also compare across forks.

base: main ▾ compare: main ▾

Choose a head ref

Find a branch

Learn about pull requests

Branches Tags

✓ main default

ramo1

Compare and review

Em “compare changes” selecione no componente a direita o ramo alternativo e na esquerda o ramo principal

# 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

The screenshot shows a GitHub repository interface with the following elements:

- Header:** hpbtakiuchi / Aula2706
- Search Bar:** Type to search
- Navigation:** Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights
- Section:** Compare changes
- Text:** Compare changes across branches, commits, tags, and more below. If you need to, you can also compare across forks.
- Form:** A modal dialog titled "Choose a head ref" with the following fields:
  - Base: main
  - Compare: main
  - Find a branch input field (highlighted with a red arrow)
  - Branches tab selected (highlighted with a red box)
  - Tags tab
  - checkbox checked next to main
  - checkbox checked next to ramo1 (highlighted with a red box)
  - default button
  - Create pull request button
- Text:** Choose different branch
- Text:** Learn about pull requests
- Text:** Ramo alternativo (highlighted with a large red box and a red arrow pointing to the ramo1 checkbox)
- Text:** Branches, tags, commit ranges, and time ranges. In the same repository and across forks.
- Table:** Example comparisons
  - ramo1 vs main (13 minutes ago)
  - main@{1day}...main (24 hours ago)

#### 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

The screenshot shows a GitHub interface for comparing changes between branches. A red box highlights the central area where the message "A página seguinte contém informações sobre o pull request a ser criado" (The next page contains information about the pull request to be created) is displayed. Below this, another red box highlights the green "Create pull request" button.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

base: main ▾ compare: ramo1 ✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison

-o 1 commit

-o Commits on Jul 2, 2024

Criação do programa 3  
Hugo committed 16 minutes ago

Showing 1 changed file with 3 additions and 0 deletions

programa3.java

```
@@ -0,0 +1,3 @@
1 + public class programa3 {
2 +     //Algum código
```

A página seguinte contém informações sobre o pull request a ser criado

Create pull request

1 contributor

d5cc7ac

Split Unified

# 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about diff comparisons here.

base: main ▾ ← compare: ramo1 ▾ ✓ Able to merge. These branches can be automatically merged.

Add a title

Criação do programa3

Add a description

Write Preview

Add your description here...

Markdown is supported

Paste, drop, or click to add files

Reviewers

No reviews—at least 1 approving review is required.

Assignees

No one [assign yourself](#)

Labels

None yet

Projects

None yet

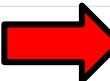
Milestone

No milestone

Development

Use [Closing keywords](#) in the description to automatically close issues

Helpful resources



**Create pull request**

# 4) Colaborador envia as alterações ao servidor e solicita um Pull Request

The screenshot shows a GitHub pull request page. At the top, there's a navigation bar with links for Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, and Insights. The main title of the pull request is "Criação do programa3 #3". Below the title, it says "hpbonfimt wants to merge 1 commit into `main` from `ramo1`". The pull request has 0 conversations, 1 commit, 0 checks, and 1 file changed. A comment from user "hpbonfimt" is shown, stating "commented now" and "No description provided." Below the comment, there's a link to "Criação do programa3". On the right side of the page, there are sections for "Assignees" (None yet), "Labels" (None yet), "Projects" (None yet), and "Milestone" (None). A red box highlights the text "Página do pull request criado" and "Próximo passo é o proprietário aceitar a alteração no ramo principal".

Página do pull request criado

Próximo passo é o proprietário aceitar a alteração no ramo principal

Criação do programa3 #3

hpbonfimt wants to merge 1 commit into `main` from `ramo1`

Conversation 0 Commits 1 Checks 0 Files changed 1

hpbonfimt commented now

No description provided.

Criação do programa3

Add more commits by pushing to the `ramo1` branch on [hpbtakiuchi/Aula2706](#).

**Review required**  
At least 1 approving review is required by reviewers with write access. [Learn more about pull request reviews.](#)

**Merging is blocked**  
Merging can be performed automatically with 1 approving review.

Assignees  
None yet

Labels  
None yet

Projects  
None yet

Milestone  
None

# Trabalho em grupo

---

- **Passo a passo da codificação em grupo usando GitHub de maneira segura**
  - 1) Proprietário cria as regras de Pull Request
  - 2) Colaborador clona o repositório
  - 3) Colaborador cria um ramo e realiza as alterações
  - 4) Colaborador envia as alterações ao servidor e solicita um Pull Request
  - **5) Proprietário aceita/rejeita as alterações**

# 5) Proprietário aceita/rejeita as alterações

The screenshot shows a GitHub repository page for 'Aula2706'. At the top, there's a yellow banner indicating a recent push from 'ramo1' 14 minutes ago. Below this, the repository details show it's public and has 2 branches and 0 tags. The commit history lists several changes made by 'Hugo':

File	Commit Message	Time Ago
programa1.java	alteração do programa1.java	47b55ce · 3 hours ago
README.md	commit README	5 days ago
arquivo.txt	Primeiro Commit	5 days ago
programa1.java	alteração do programa1.java	3 hours ago
programa2.java	Criação do programa2	8 hours ago

Below the commits, there's a link to 'README' with an edit icon. A red box highlights the text 'Página do proprietário' in the bottom right corner of the page.

Página do proprietário

# 5) Proprietário aceita/rejeita as alterações

The screenshot shows a GitHub repository page for 'Aula2706'. At the top right, there is a green button labeled 'Compare & pull request' with a red box and a black arrow pointing to it from the right. Below the button, a yellow banner indicates 'ramo1 had recent pushes 14 minutes ago'. The repository details include 'main' branch, 2 branches, 0 tags, and several commits by 'Hugo' (e.g., 'alteração do programa1.java', 'README.md', 'arquivo.txt'). A large red box covers the main content area, containing the text 'Selecione para comparar pull request'. To the right of this box, there are sections for 'Releases' (no releases published, 'Create a new release') and 'Packages' (no packages published, 'Publish your first package'). At the bottom left, the repository name 'Aula2706' is displayed.

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Aula2706 Public

Pin Unwatch 1 Fork 0 Star 0

ramo1 had recent pushes 14 minutes ago

Compare & pull request

No description, website, or topics provided.

main 2 Branches 0 Tags

Hugo alteração do programa1.java

README.md

arquivo.txt

programa1.java

programa2.java

3 days ago

5 days ago

3 hours ago

8 hours ago

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Aula2706

# 5) Proprietário aceita/rejeita as alterações

hpbtakiuchi / Aula2706

Type to search

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

Label issues and pull requests for new contributors  
Now, GitHub will help potential first-time contributors discover issues labeled with [good first issue](#)

Dismiss

Filters ▾ [is:pr is:open](#) Labels 9 Milestones 0 New pull request

1 Open 2 Closed Author ▾ Label ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾

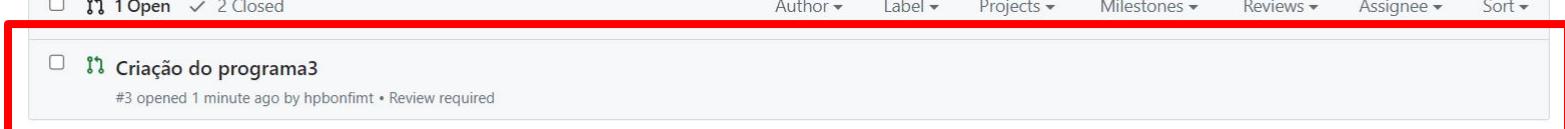
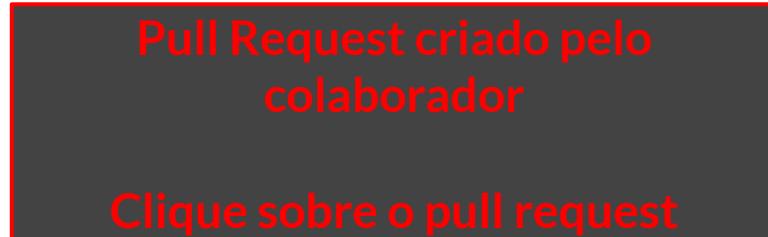
Criação do programa3  
#3 opened 1 minute ago by hpbonfimt • Review required

ProTip! Follow long discussions with [comments:>50](#).

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

Pull Request criado pelo colaborador

Clique sobre o pull request



## **5) Proprietário aceita/rejeita as alterações**

hpbtakiuchi / Aula2706

Type  to search

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

## Criação do programa3 #3

**Open** hpbonfimt wants to merge 1 commit into `main` from `ramo1`

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -0

hpbonfimt commented 3 minutes ago

No description provided.

Criação do programa3 50cfef1

Add more commits by pushing to the `ramo1` branch on [hpbtakiuchi/Aula2706](#).

**Review required**  
At least 1 approving review is required

**Merging is blocked**  
Merging can be performed automatically

Reviewers  
No reviews—at least 1 approving review is required.  
Still in progress? [Convert to draft](#)

Assignees  
No one—[assign yourself](#)

Labels  
None yet

Projects  
None yet

Milestone  
No milestone

Mesma página de pull request. Entretanto, agora na visão de proprietário

## **5) Proprietário aceita/rejeita as alterações**

hpbtakiuchi / Aula2706

Type  to search

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

## Criação do programa3 #3

**Open** hpbonfimt wants to merge 1 commit into `main` from `ramo1`

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -0

hpbonfimt commented 3 minutes ago  
No description provided.

Opção Add review visível para o proprietário

Add more commits by pushing to the `ramo1` branch on [hpbtakiuchi/Aula2706](#).

**Review required**  
At least 1 approving review is required by reviewers with write access. [Learn more about pull request reviews](#).

**Merging is blocked**  
Merging can be performed automatically with 1 approving review.

Add your review

Reviewers  
No reviews—at least 1 approving review is required.  
Still in progress? [Convert to draft](#)

Assignees  
No one—assign yourself

Labels  
None yet

Projects  
None yet

Milestone  
No milestone

# 5) Proprietário aceita/rejeita as alterações

The screenshot shows a GitHub pull request interface for a repository named 'hpbtakiuchi / Aula2706'. The pull request is titled 'Criação do programa3 #3' and is marked as 'Open'. It shows a merge from the 'ramo1' branch into the 'main' branch. The 'Files changed' tab is selected, showing a single file, 'programa3.java', with one commit. The commit message is '+ código java'. A red arrow points to the green 'Review changes' button in the top right corner of the changes view.



© 2024 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

# 5) Proprietário aceita/rejeita as alterações

The screenshot shows a GitHub pull request interface. At the top, there's a navigation bar with links for Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. The title of the pull request is "Criação do programa3 #3". Below the title, it says "hpbonfimt wants to merge 1 commit into main from ramo1". The main area shows a single commit with the file "programa3.java" and the change "1 + código java". A red arrow points from the bottom left towards a modal window titled "Finish your review". This modal contains a rich text editor toolbar, a comment input field, and three radio button options: "Comment", "Approve" (which is selected), and "Request changes". There's also a "Submit review" button at the bottom right of the modal.

Finish your review

Write Preview H B I ḥ ḡ ḣ Ḥ ḧ Ḩ Ḫ Ḫ

Leave a comment

Markdown is supported

Paste, drop, or click to add files

Comment  
Submit general feedback without explicit approval.

Approve  
Submit feedback and approve merging these changes.

Request changes  
Submit feedback that must be addressed before merging.

Submit review

# 5) Proprietário aceita/rejeita as alterações

Criação do programa3 #3  
hpbonfim wants to merge 1 commit into `main` from `ramo1`

Add more commits by pushing to the `main` branch on [hpbonfim/Aula2706](#)

**Changes approved**  
1 approving review by reviewers with write access. [Learn more about pull request reviews](#).

**1 approval**

**Continuous integration has not been set up**  
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

**This branch has no conflicts with the base branch**  
Merging can be performed automatically.

**Merge pull request** You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Add a comment

Write

Add your comment here...

Markdown is supported Paste, drop, or click to add files

None yet

Projects None yet

Milestone No milestone

Development Successfully merging this pull request may close these issues.  
None yet

Notifications Customize

You're receiving notifications because you're watching this repository.

2 participants

Lock conversation

# 5) Proprietário aceita/rejeita as alterações

Criação do programa3 #3  
hpbonfimt wants to merge 1 commit into `main` from `ramo1`

Add more commits by pushing to the `ramo1` branch on [hpbtakiuchi/Aula2706](#).

Merge pull request #3 from hpbtakiuchi/ramo1

`Criação do programa3`

This commit will be authored by 58343987+hpbtakiuchi@users.noreply.github.com

**Confirm merge** **Cancel**

Add a comment

Write

Add your comment here...

Markdown is supported Paste, drop, or click to add files

Close pull request Comment

None yet

Projects None yet

Milestone None yet

Development Successfully merging this pull request may close these issues.  
None yet

Notifications Unsubscribe Customize  
You're receiving notifications because you're watching this repository.

2 participants

Lock conversation

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

ProTip! Add comments to specific lines under [Files changed](#).

# Trabalho em grupo

---

- Alguns erros comuns
  - Deletar o branch corrente:

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (ramo1)
$ git branch -d ramo1
error: cannot delete branch 'ramo1' used by worktree at 'C:/Users/hugop/Repositorios_Git2/Aula2706'
```

- Acessar um ramo que não existe

```
hugop@Hugopc MINGW64 ~/Repositorios_Git2/Aula2706 (main)
$ git checkout ramo1
error: pathspec 'ramo1' did not match any file(s) known to git
```



# Conteúdo da aula

- Introdução
- Git e GitHub
- Comandos e conceitos
- Criando repositório individual
- Criando repositório compartilhado
- **Conclusão**
- **Exercícios propostos**
- **Referências bibliográficas**

# Conclusão



-

# Exercícios propostos

---

# Referências bibliográficas Básicas

---

- CRUZ, Fabio. Scrum e Agile em Projetos (2a. edição): guia completo. 2. ed. Rio de Janeiro:Brasport. 2018.
- PRIKLADNICKI, Rafael; WILLI, Renato, MILANI, Fabiano. Métodos Ágeis para Desenvolvimento de Software. Porto Alegre:Bookman. 2014.
- BELL , P.; BEER , B. Introdução ao GitHub: um Guia que Não é Técnico. 1. ed. São Paulo: Editora Novatec. 2014.

## Referências bibliográficas Complementares

---

- BROD, Cesar. SCRUM: Guia prático para projetos ágeis. 2<sup>a</sup>. ed. São Paulo:Novatec. 2015.
- KERZNER, H. Gerenciamento de Projetos: uma abordagem sistêmica para planejamento, programação e controle. 10<sup>a</sup>. ed. São Paulo:Blucher. 2011.
- OLIVEIRA, Bruno Souza de. Métodos Ágeis e Gestão de Serviços de TI. Rio de Janeiro:Brasport. 2018.

# Referências bibliográficas Complementares

---

- SCRUMSTUDY. Um Guia para o Conhecimento em Scrum (Guia SBOK™). ed. 2016. Phoenix. 2016. Disponível em <http://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016-Portuguese.pdf>.
- PROJECT MANAGEMENT INSTITUTE (PMI). Um guia do conhecimento em gerenciamento de projetos. 5. ed. PMI:Newton Square. 2013.

# Material de Apoio

---

- PRESSMAN, R.S. Engenharia de Software: Uma Abordagem Profissional. 7. ed. São Paulo: McGraw-Hill, 2011.
- Site contendo o manifesto ágil. Acessado em 06/03. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>
- AQUILES, Alexandre; FERREIRA, Rodrigo. Controlando versões com Git e GitHub. São Paulo: Casa do código | Alura, 2020.
- Apostilas do professor Dr. Evandro.

# Imagenes

---

- <https://www.pinterest.com/pin/rugby--66991113186539678/>