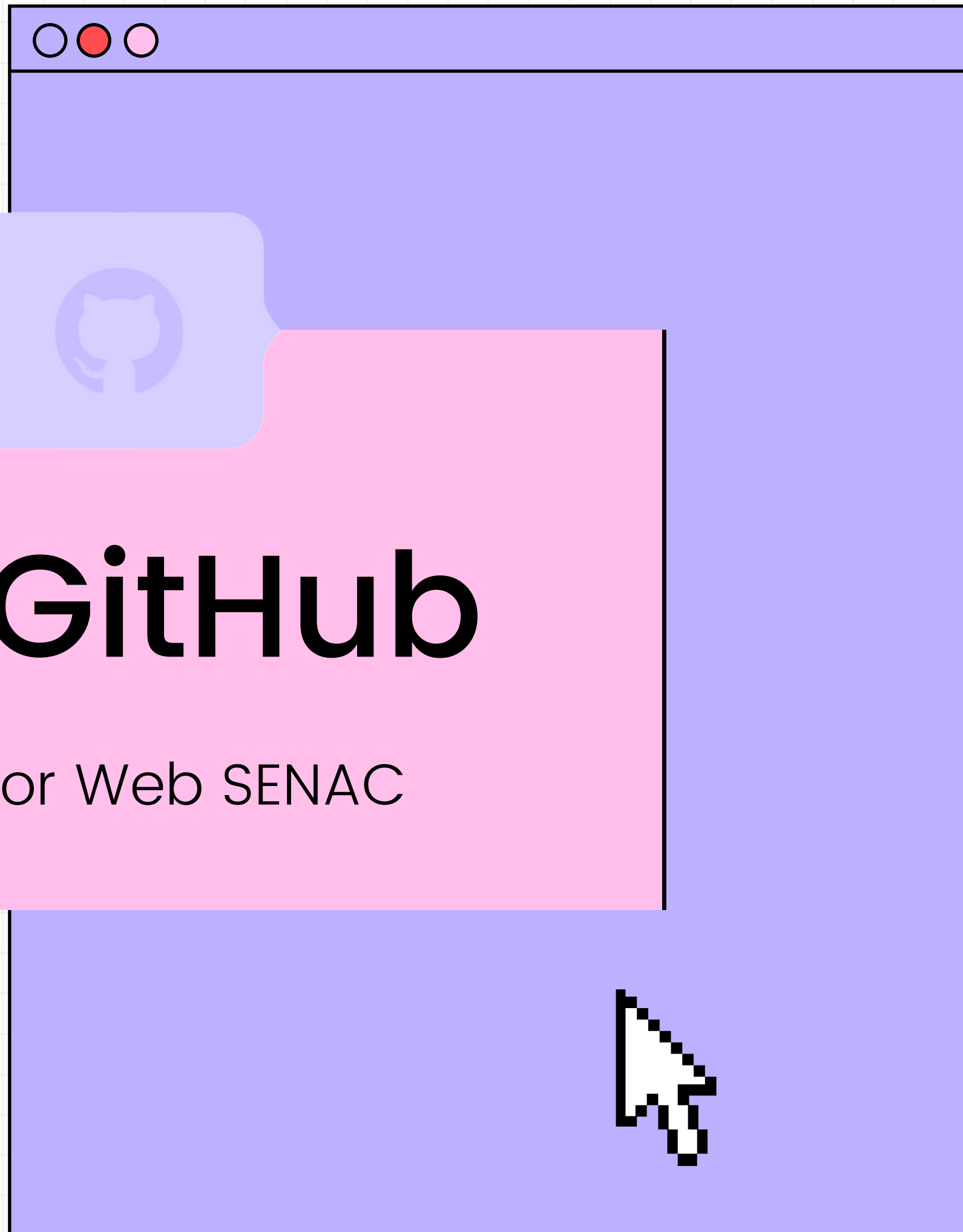


# Git e GitHub

Programador Web SENAC





# Linus Torvalds

Um engenheiro de software finlandês, nascido em 28 de dezembro de 1969

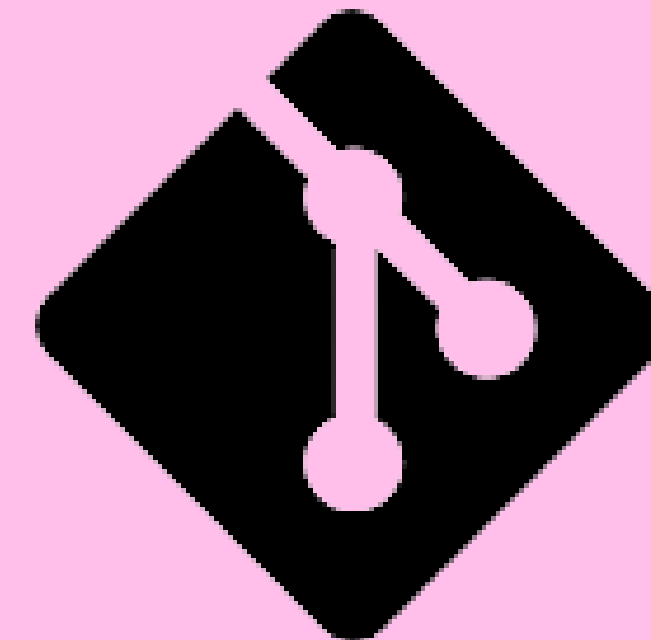
Criador do Kernel do Linux, base a qual são construídos diversos sistemas operacionais gratuitos, como as distribuições Linux Ubuntu, Debian, Fedora, Chrome OS e o Android.

Criador do git, um software usado no desenvolvimento de programas para controle de versão durante o processo de testes e compilações.

## ○ ● ○ Diferenças entre Git e Github

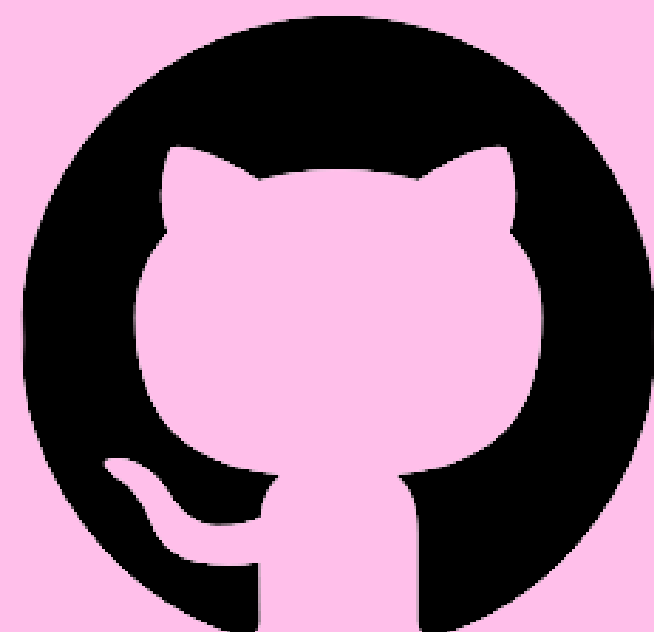
A diferença entre ambos é que enquanto o Git é um software de controle de versão, que consegue gerar registros dos projetos, para saber quais arquivos foram alterados, quando foram alterados e por quem. O GitHub é uma plataforma onde é possível subir projetos e compartilhá-los, além de criar portfólio. O Github é muito usado para compartilhar os projetos gerados pelo Git com os outros desenvolvedores.

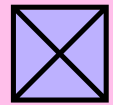




**Para que serve?**

**Qual o principal objetivo da plataforma?**



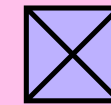


## Git

**GIT é um Sistema de Controle de Versões Distribuído — ou DVCS.**

**O DVCS registra alterações feitas em cima de um código, armazenando essas informações e permitindo que, caso seja necessário, um(a) programador(a) possa regredir a versões anteriores de uma aplicação de modo simples e rápido.**

**Ou seja, ele permite ver o código de outras pessoas colaboradoras, as diferenças que foram adicionadas ou removidas e mantém o histórico deles, ou seja, você pode voltar para qualquer versão anterior que estava funcionando.**

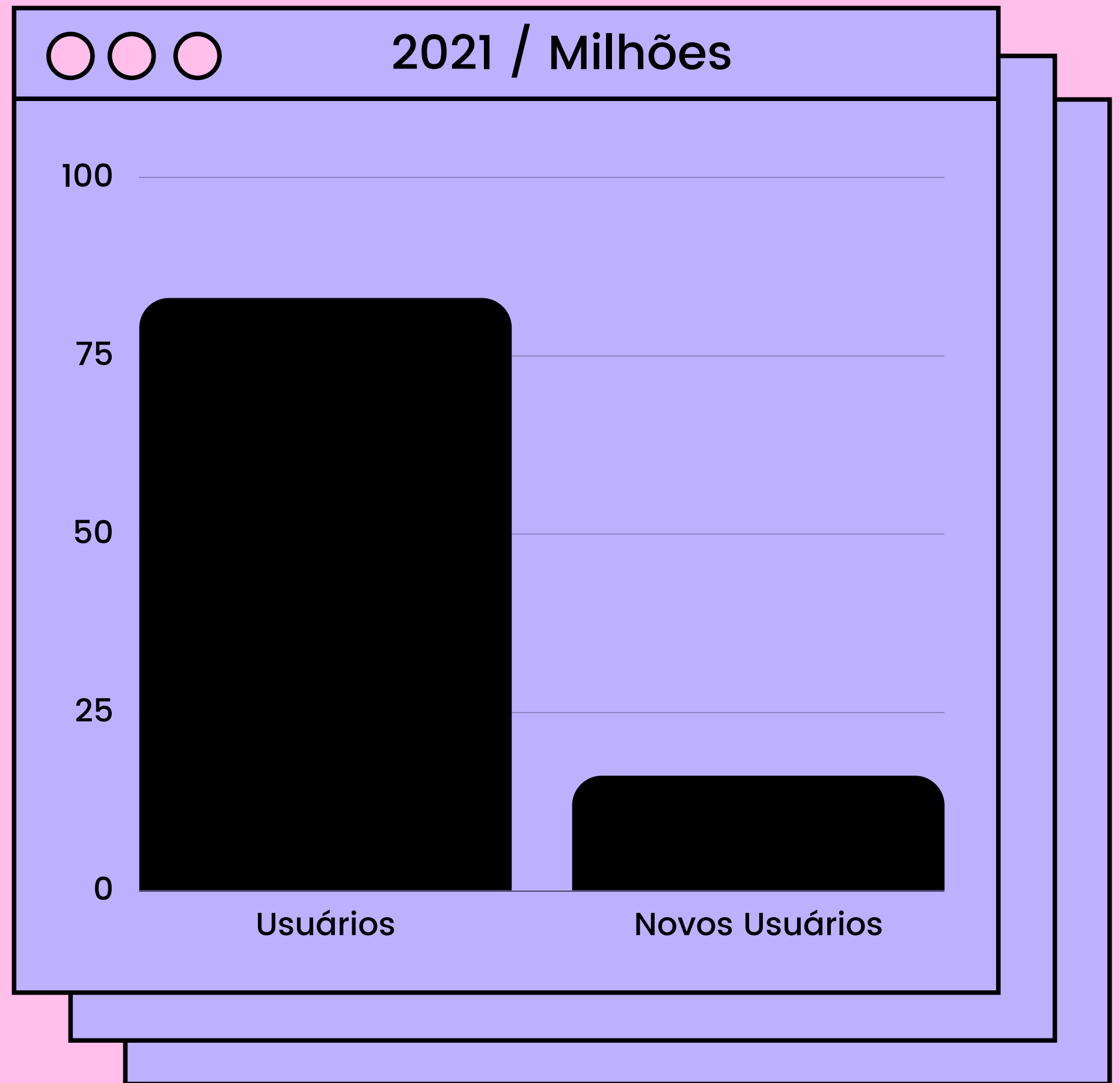


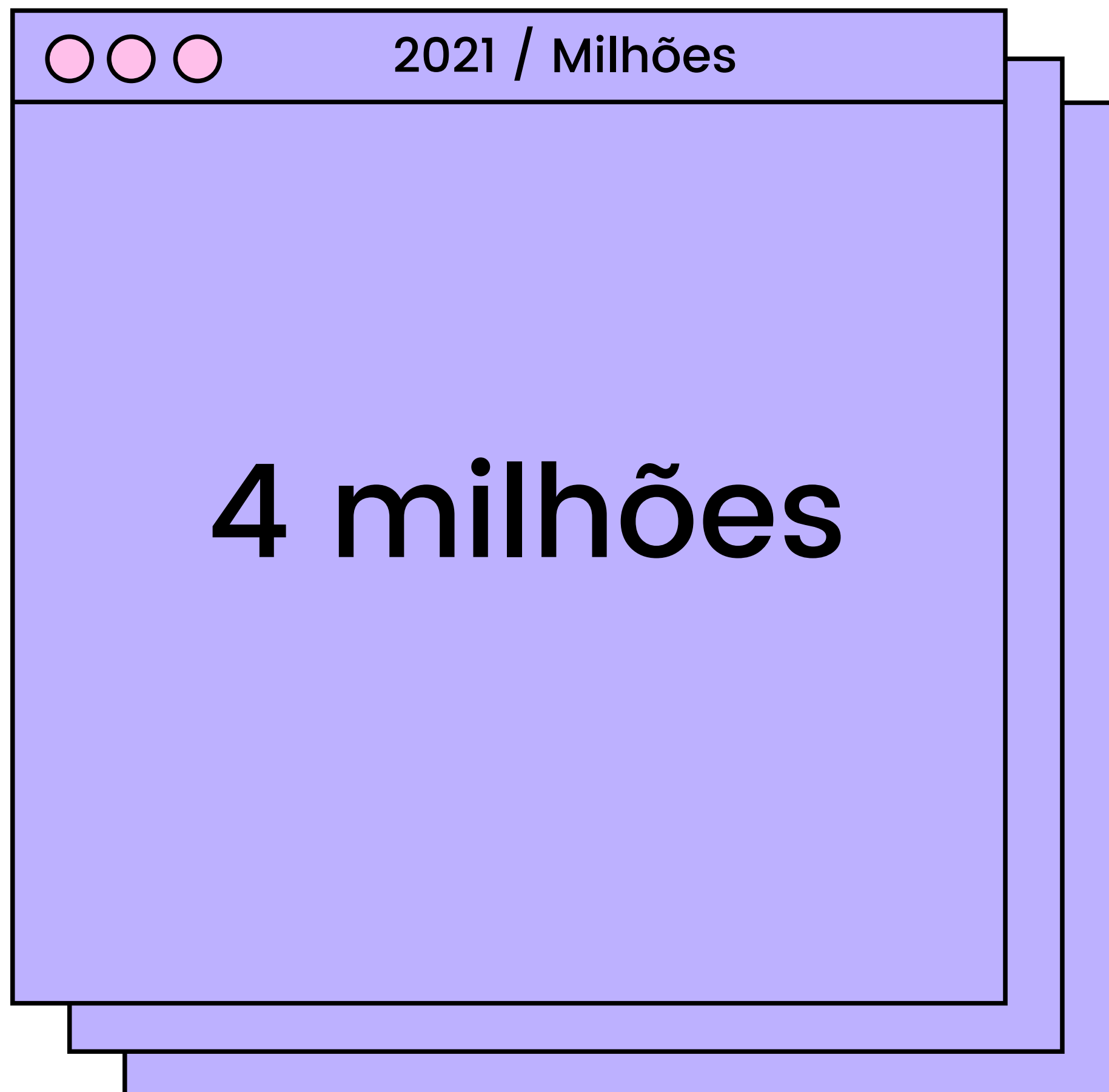
## GitHub

**GitHu é uma plataforma online onde você pode criar repositórios e hospedar neles seus projetos, colaborar com softwares open source, seguir outros(as) programadores(as) e interagir com códigos de terceiros.**

**O GitHub armazena todos estes dados em uma nuvem e você pode acessá-los de onde estiver: basta logar-se no site em qualquer navegador.**

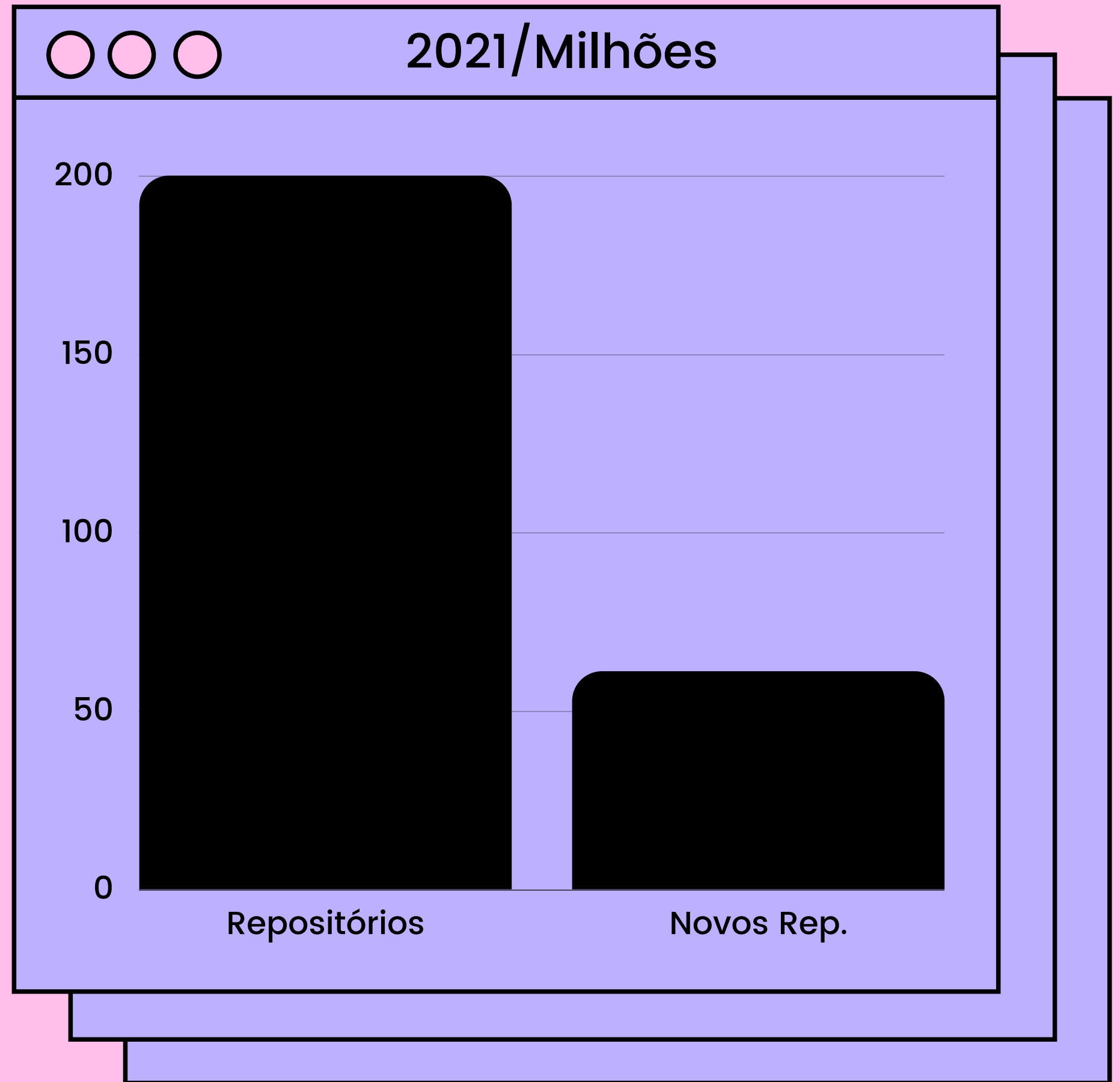
# Quantidade usuários?





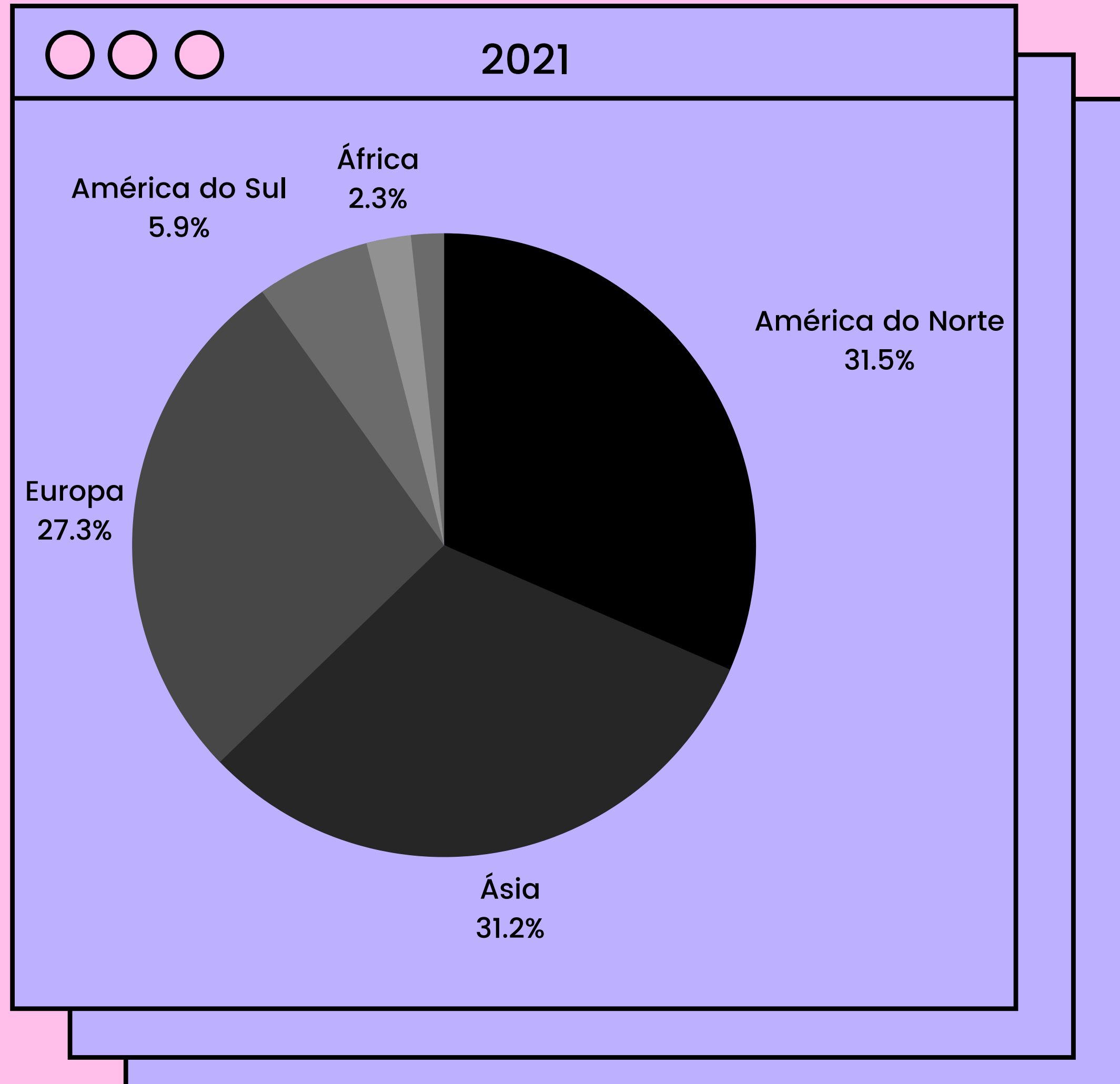
**Empresas  
cadastradas**

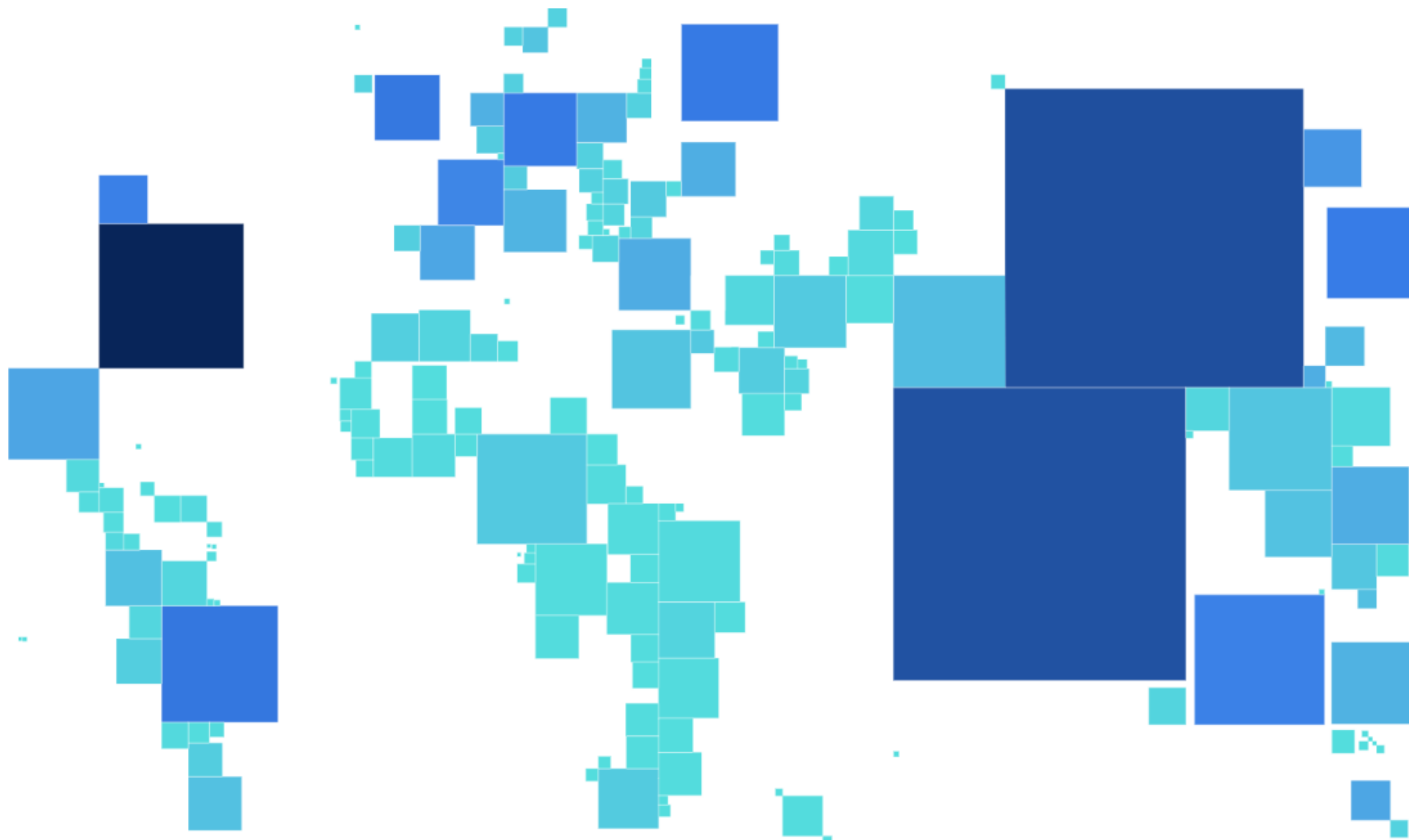
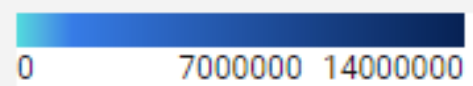
# Repositórios

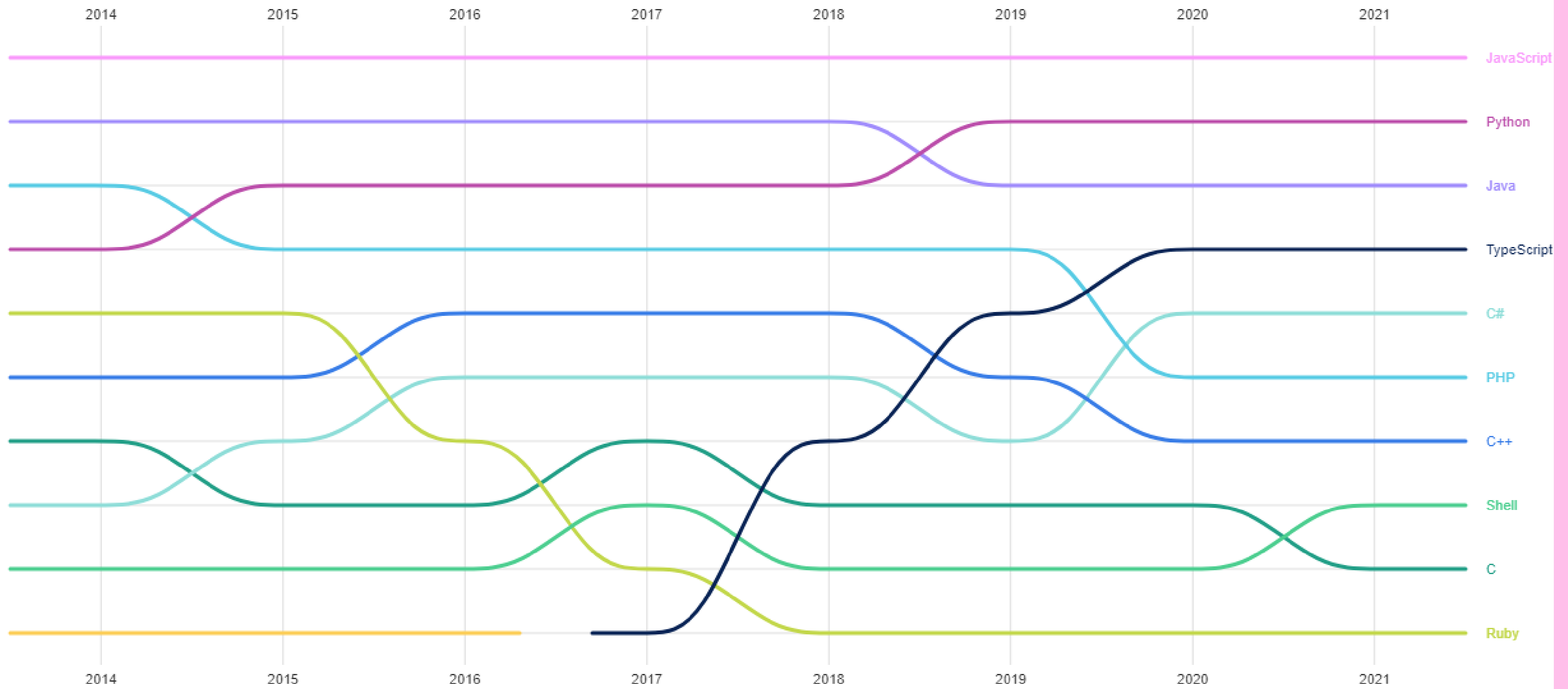




# Distribuição de usuários ativos



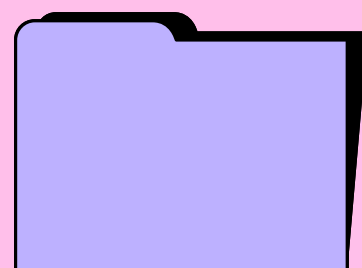
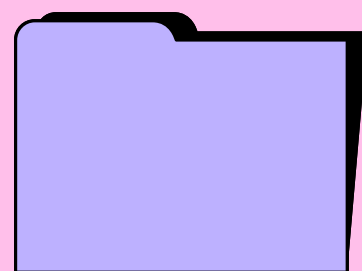
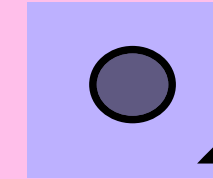
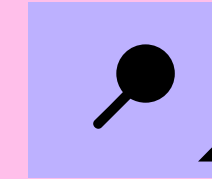
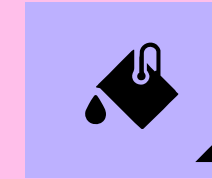
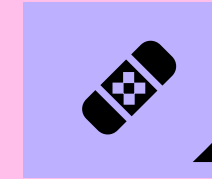






# Termos e Comandos

# Comandos Básicos do Terminal



## Criar nova pasta

`mkdir nome-da-pasta`

## Criar novo arquivo

`echo nome-do-arquivo`

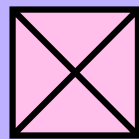
## Acessar pastas

`cd nome-da-pasta`

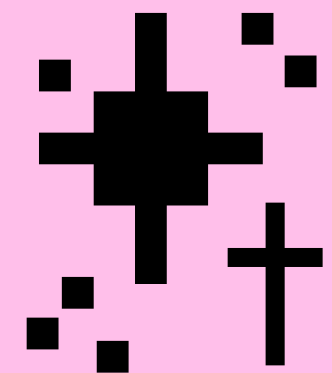
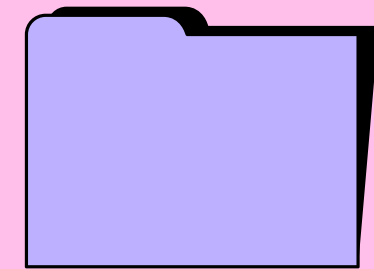
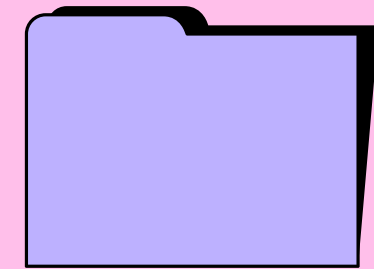
## Listar pastas

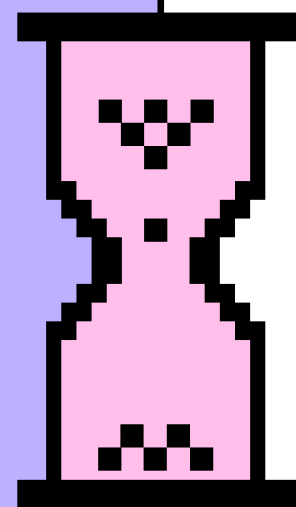
`ls`

A primeira coisa que  
você deve fazer é baixar  
o **Git Bash**, o **GitHub**  
**desktop** e uma **IDE**



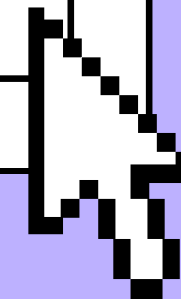
ATENÇÃO!





# Clonando o repositório

Da nuvem para sua máquina  
(utilizando o VSCode)





**Passo 1:** criar uma pasta na sua máquina

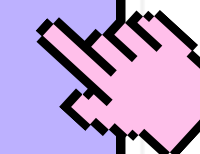
**Passo 2:** abrir a pasta na sua IDE (ex: VSCode)

**Passo 3:** ir no seu repositório, no site do GitHub, clicar no botão "code" e copiar o link HTTPS

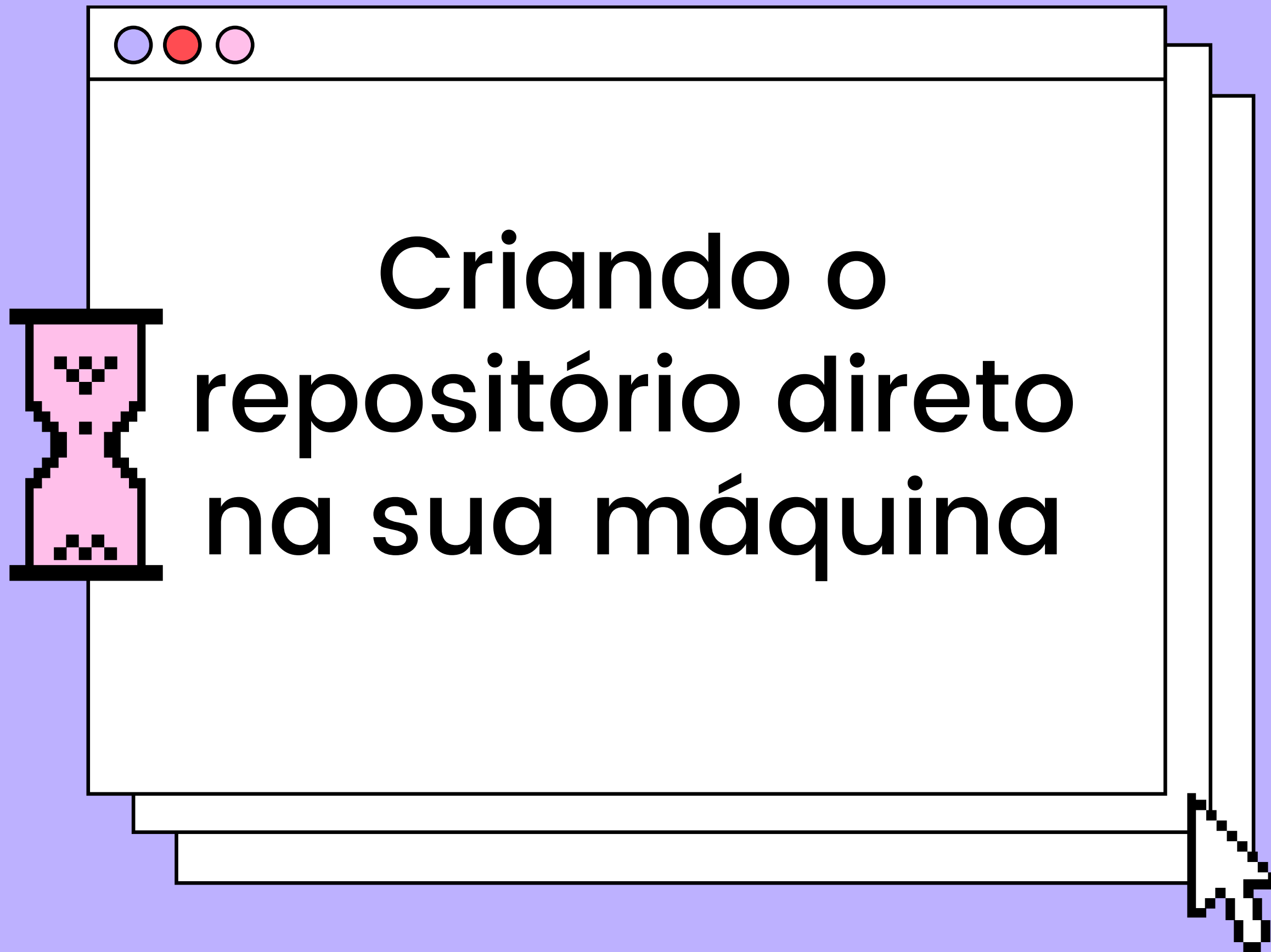
**Passo 4:** ir no seu repositório, no site do GitHub, clicar no botão "code" e copiar o link HTTPS

**Passo 5:** digitar no terminal do VSCode

```
git clone <link-do-repositorio>
```









**Passo 1:** criar uma pasta na sua máquina

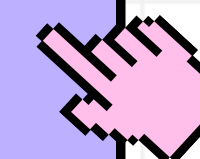
**Passo 2:** iniciar o repositório, digitando no terminal o comando:

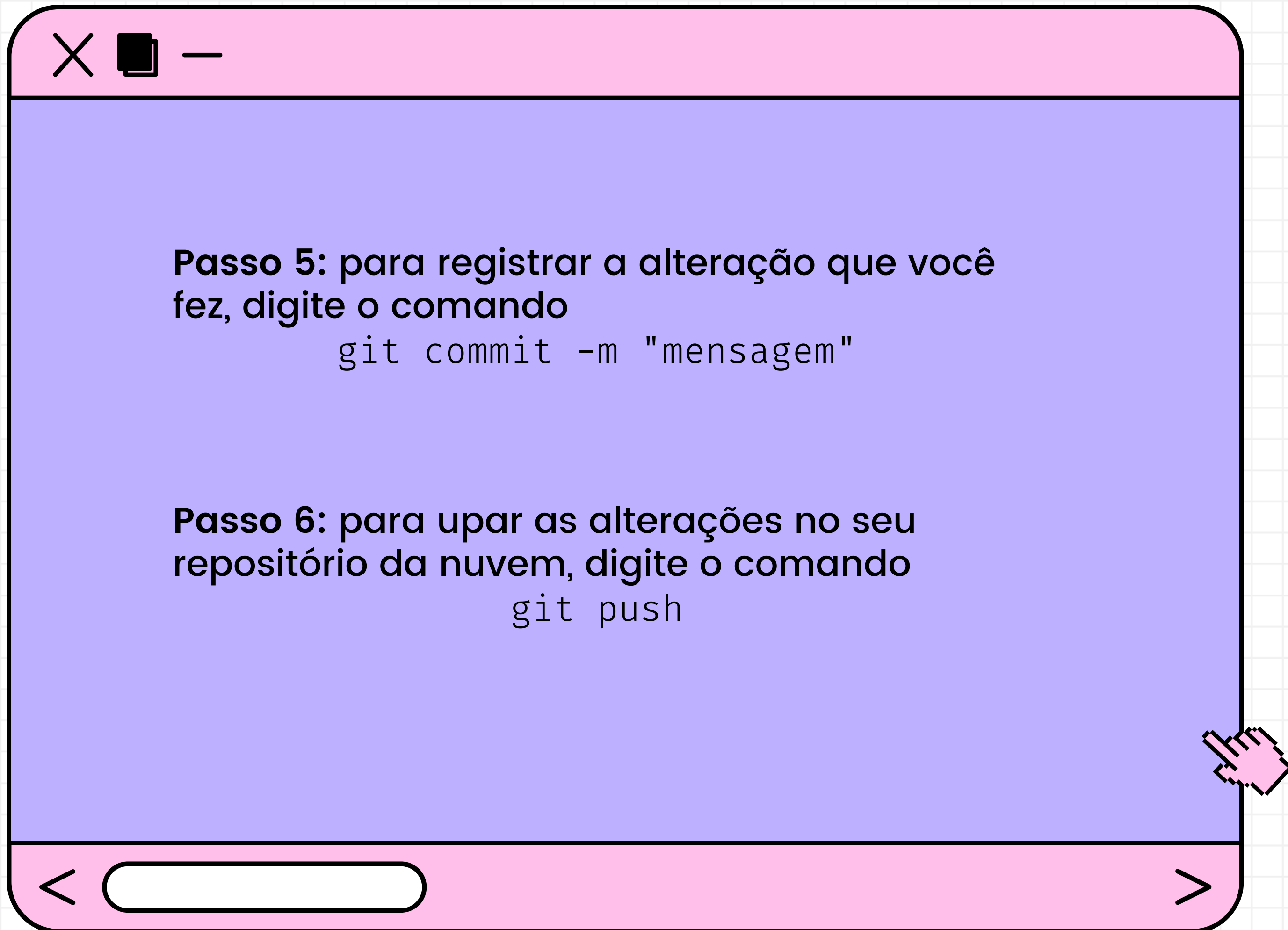
```
git init
```

**Passo 3:** criar o seu código

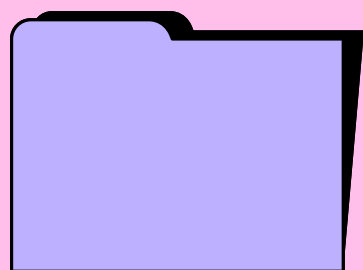
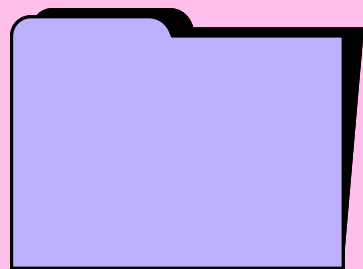
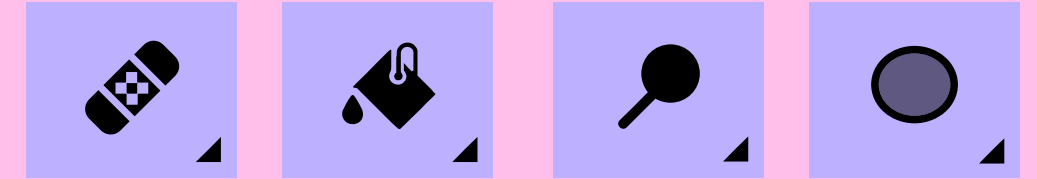
**Passo 4:** quando terminar, digitar o comando

```
git add .
```





# Branches



## Git branch

Mostra todas as branches do seu repositório

## Git branch nome-da-branch

Cria uma nova branch

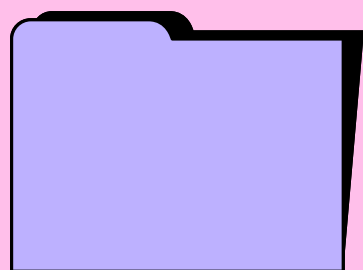
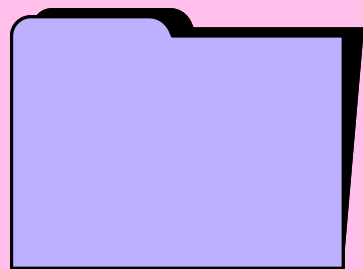
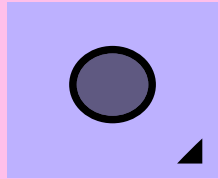
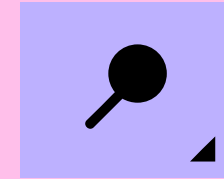
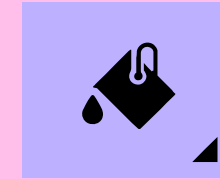
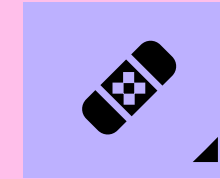
## Git checkout nome-da-branch

Acessa uma branch que já foi criada

## Git checkout -b nome-da-branch

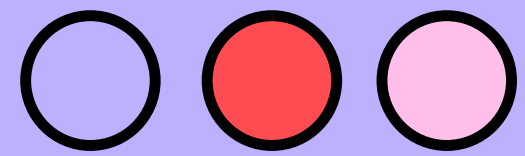
Cria e acessa uma branch

# Branches



**Git push origin  
<nome-da-branch>**

Envia suas alterações  
commitadas para uma branch  
específica no repositório  
remoto



## Pull Request (PR) e Merge

A branch principal do seu repositório chama-se main. A cada alteração que você faz no seu projeto, recomenda-se, de acordo com o git flow, que você crie uma nova branch. Após commitar seu código e dar push, você deverá criar um PR, que nada mais é que uma "solicitação" para que seu código seja revisado pelo seu líder ou seu colega de projeto (code review), por exemplo. Se o código estiver ok, suas alterações poderão ser "mergeadas" para o main.