

capital code

# GitHub

no full  
2.0



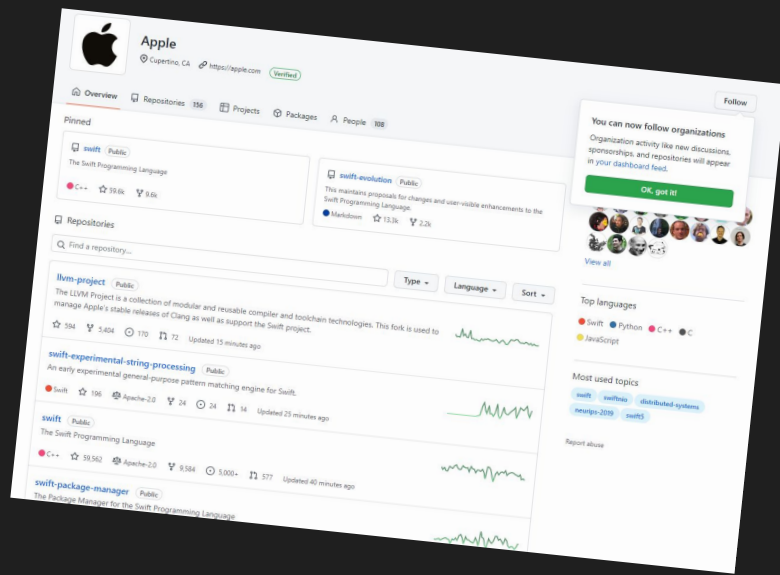
# O Git é um Sistema de controle de versão distribuído Open Source

*“open source” significa “código aberto”*



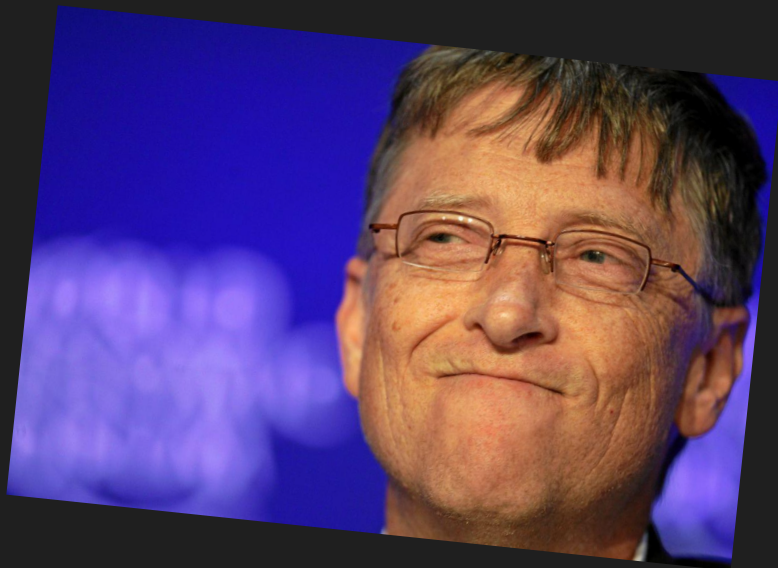
# O que é o GitHub?

Uma **plataforma social colaborativa**, onde programadores e empresas colocam seus **projetos para o desenvolvimento do código**



# Por que eu preciso do Git?

- Vários desenvolvedores **trabalhando em paralelo**;
- **Conflitos** de código;
- Acessar **versões anteriores** do mesmo código;
- Branches, muito usadas em projetos que estão em paralelo e **envolvem a mesma base de código**;

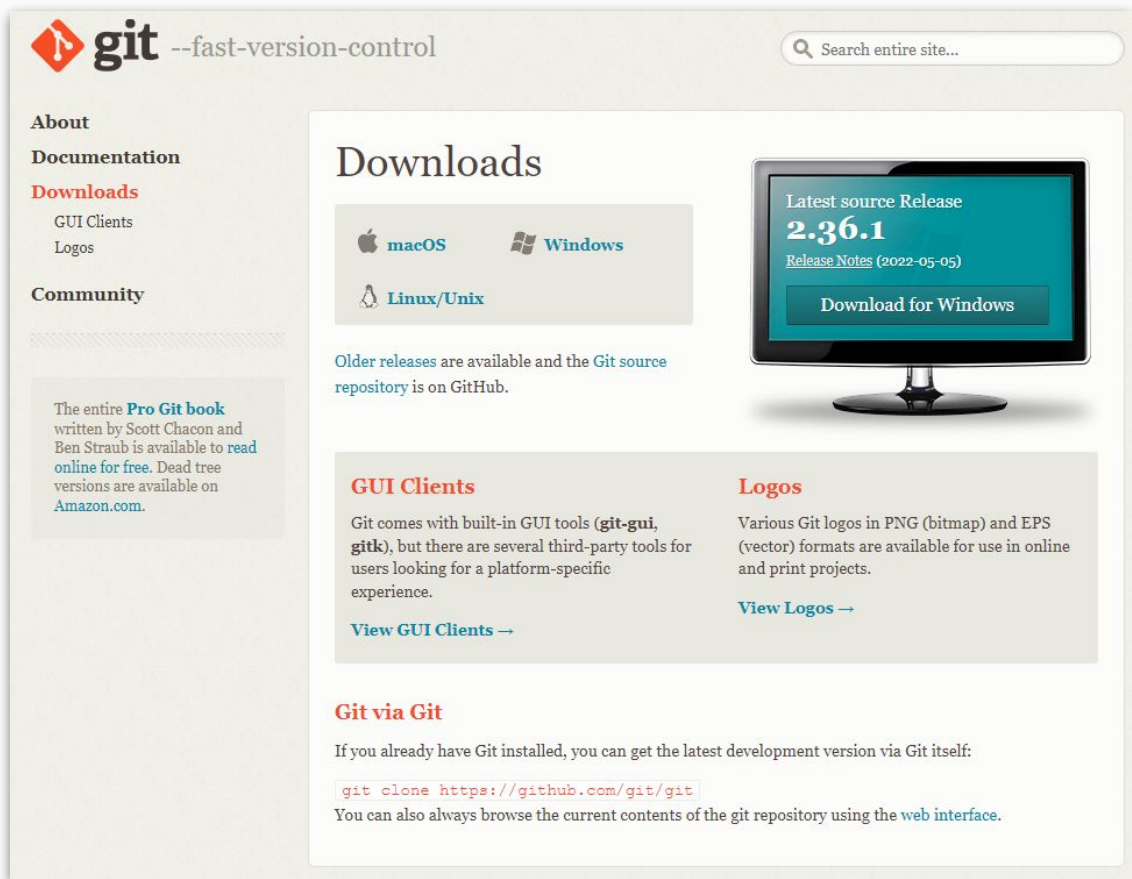


# #Bora pro código

Antes de começarmos o desenvolvimento de fato, vamos baixar a versão mais recente do git.

Acesse o link abaixo

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



The screenshot shows the Git website's Downloads page. At the top, the Git logo and tagline "--fast-version-control" are visible, along with a search bar. The left sidebar contains links for "About", "Documentation", "Downloads" (highlighted in red), "GUI Clients", "Logos", and "Community". The main content area features a "Downloads" section with icons for macOS, Windows, and Linux/Unix. To the right, a monitor displays the "Latest source Release 2.36.1" with a "Download for Windows" button. Below the download links, text states that older releases are available on GitHub. Further down, there are sections for "GUI Clients" (with a "View GUI Clients" link) and "Logos" (with a "View Logos" link). At the bottom, the "Git via Git" section provides instructions on how to clone the repository using the command `git clone https://github.com/git/git` and mentions the web interface.

**git** --fast-version-control

Search entire site...

**About**  
**Documentation**  
**Downloads**  
GUI Clients  
Logos  
**Community**

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Downloads

macOS Windows Linux/Unix

Older releases are available and the Git source repository is on GitHub.

Latest source Release  
**2.36.1**  
[Release Notes \(2022-05-05\)](#)  
Download for Windows

### GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

### Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

### Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

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

You can also always browse the current contents of the git repository using the [web interface](#).

# Verifique se o git está instalado

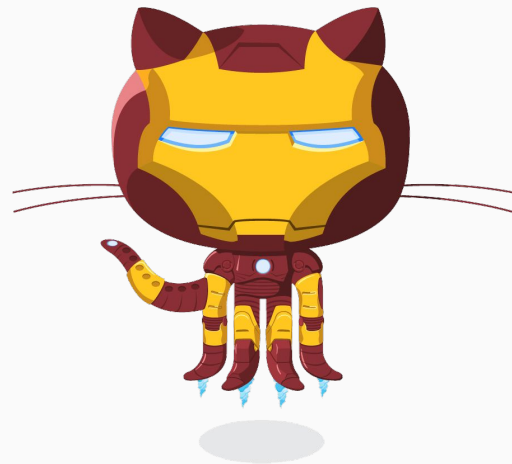
`git --version`

# Criando seu repositório local

No seu computador, crie uma pasta para o seu projeto.

Vamos chamá-la de:

**git-capital-code**



# Acesse a pasta através do GIT

```
cd git-capital-code
```

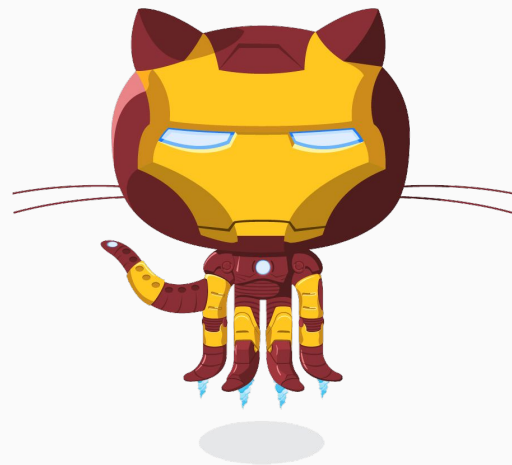
```
git init
```



# Adicionando um pouquinho de código

Crie um arquivo chamado **demo.txt** na pasta do projeto com o seguinte texto:

**“Sou Aluno Capital Code”**



# Preparando e “Commitando” o código

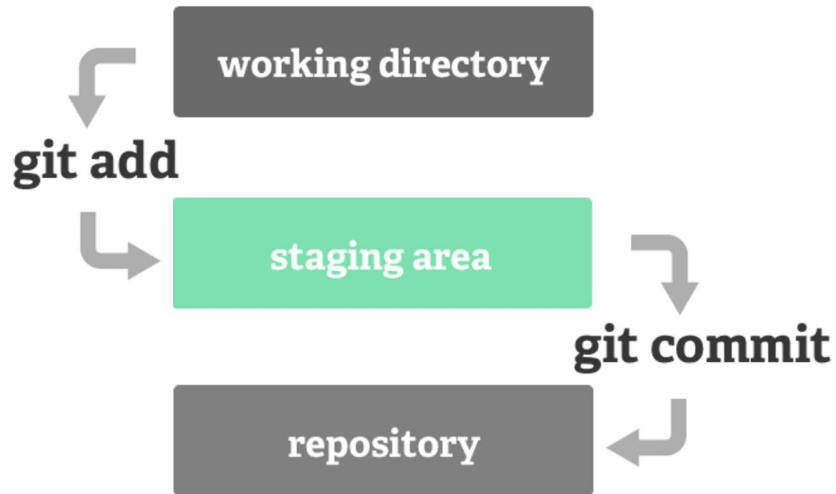
O commit é o processo no qual o código é adicionado ao repositório local.

Antes de commitar o código, **ele deve estar na área de staging.**

A área de staging **é onde são mantidas as alterações** que ainda não foram commitadas.



A área de staging é onde são mantidas as alterações que ainda não foram commitadas.



Use o seguinte comando para **preparar o arquivo para ser commitado**, ou seja mandá-lo para a área de staging

```
git add demo.txt
```

Se você deseja **adicionar todos os arquivos dentro da pasta do projeto**, use o seguinte comando

```
git add .
```

# Use o seguinte comando para commitar o arquivo:

**git commit -m "Meu primeiro Commit"**

“Meu primeiro Commit” será a mensagem do commit. Procure usar mensagens de commit que sejam relevantes e que indiquem o que as alterações do código fazem.



# Git Status

Use o comando `git status` para encontrar informações sobre quais arquivos foram modificados e quais estão na área de staging, e portanto prontos para receberem um commit.

**git status**



# Git Status

Agora modifique o arquivo `demo.txt` e adicione o seguinte texto:

“Add mais conteúdo”

**git status**

O status mostra que o **demo.txt** foi modificado e não está na área de staging.

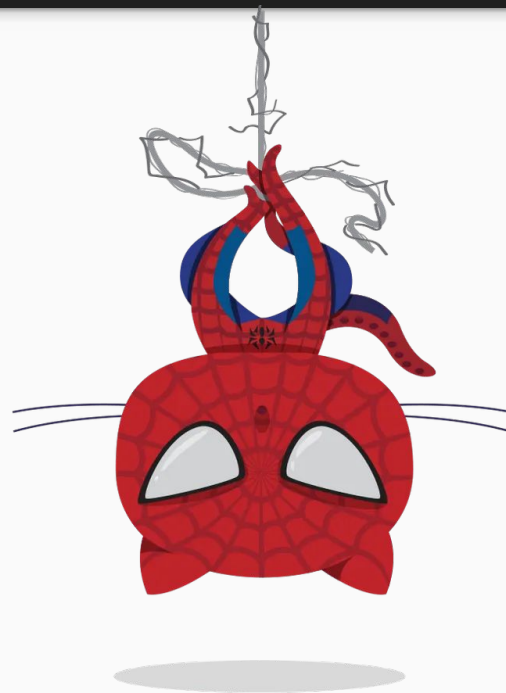




**Agora vamos adicionar o demo.txt para a área de staging e realizar o commit com os seguintes comandos**

**git add** demo.txt

**git commit -m** "arquivo modificado"

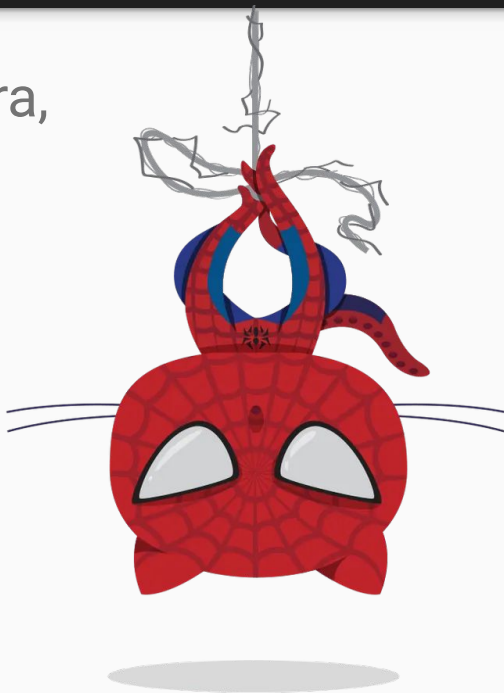


# Git Log

para exibir todos os commits realizados até agora, escreva:

**git log**

O log mostra o autor de cada commit, a data do commit, e a mensagem do commit.



capital code

# GitHub

no full  
2.0

