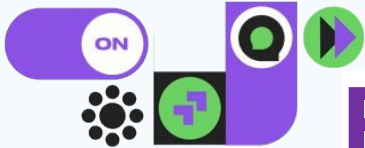


GIT



Aula de hoje:

- **Conceitos básicos - o que é programação?**
- **O que é linha de comando, Git e GitHub**
- **Configuração**
- **Comandos básicos no Git Bash**
- **Versionamento local**
- **Versionamento remoto**
- **Clonando projetos**
- **Contribuição no projeto**
- **Atualização do projeto**



Front-end e back end

A parte visual de uma aplicação, onde é possível interagir com o sistema através de uma interface gráfica.

Na web, as principais tecnologias para o desenvolvimento front-end são **JavaScript, HTML e CSS**. Mas também existem diversas bibliotecas e frameworks

O que tem por trás de uma aplicação, é a parte responsável pela implementação das regras de negócio.

Muitas tecnologias são comuns no back-end como Java, Python, Ruby, C# e até mesmo JavaScript (Node.js)



Hardware e Software

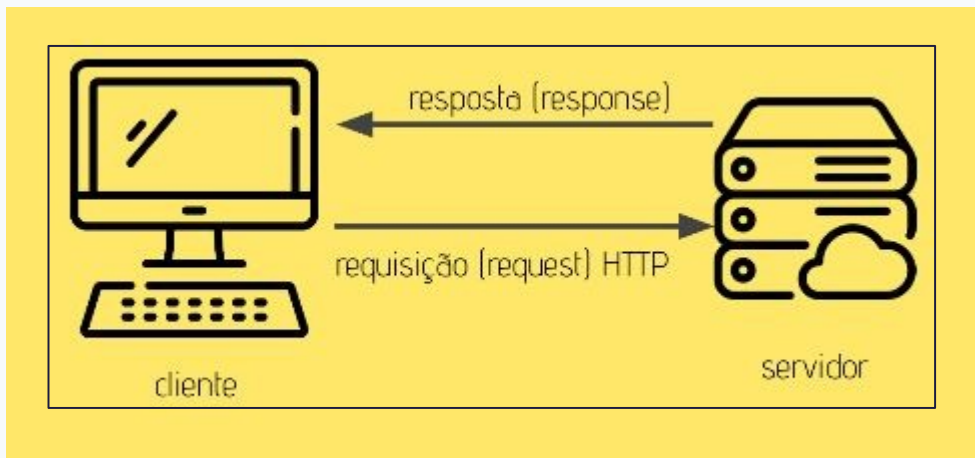
Componentes físicos de um computador, funcionam através de uma linguagem composta por bits, os famosos zeros e uns.

Conjuntos de instruções que são interpretados para linguagem de máquina.

As instruções normalmente são escritas em uma linguagem mais fácil de ser lida por nós, e então são **compiladas ou interpretadas** para linguagem de máquina.



Arquitetura Cliente - Servidor

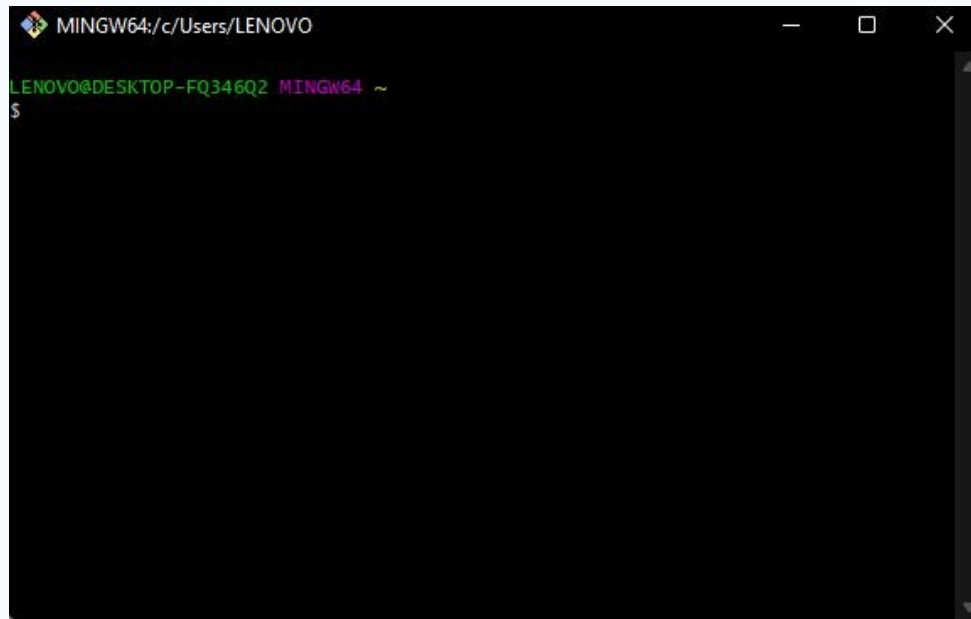


Cliente é a nossa aplicação front-end executando na máquina da nossa usuária

Servidor é a nossa aplicação back-end rodando em algum computador remoto, normalmente uma hospedagem profissional



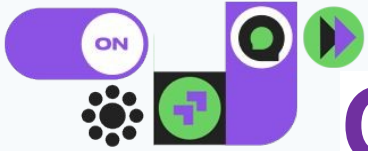
Tipos de ferramentas



Há tipos diferentes dessa interface, o que varia de acordo com o sistema operacional utilizado.

Há dois tipos de Shell mais comuns: Windows Shell (no Windows) e o Bash (OS X e Linux).

Vamos usar o git bash!



O que é Git?





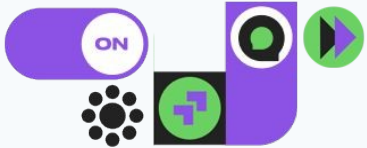
O que é Git?

Git é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software para registrar o histórico de edições dos arquivos.

Foi desenvolvido por Linus Torvalds (criador do Linux).

Com o Git podemos desenvolver projetos colaborativos, com diversas pessoas trabalhando simultaneamente no mesmo código sem riscos de perdermos o que fizemos.

O Git guarda um histórico de tudo que foi alterado nos arquivos ao longo do tempo, além de mostrar quem foi o autor da mudança.



Desenvolvimento de software por uma equipe:

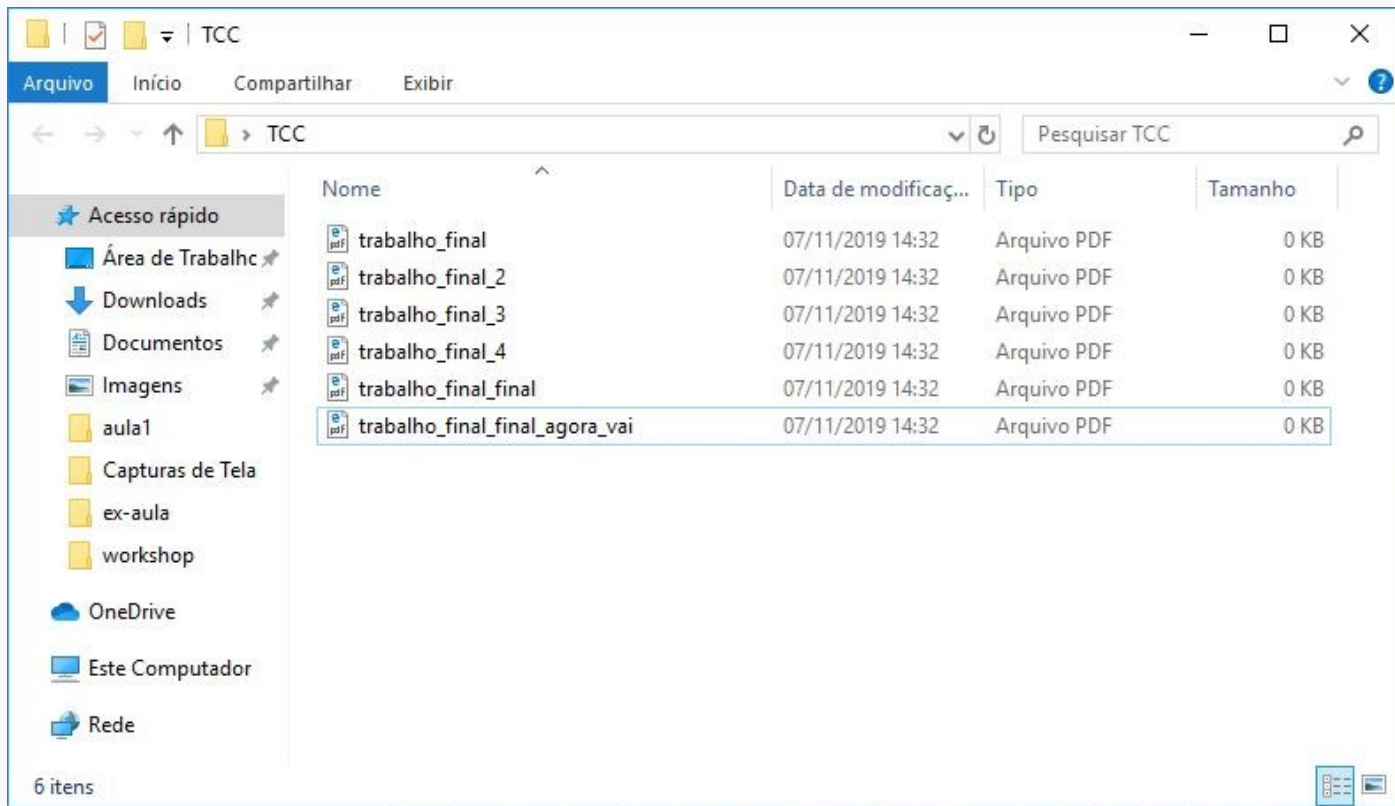


Criadora do software



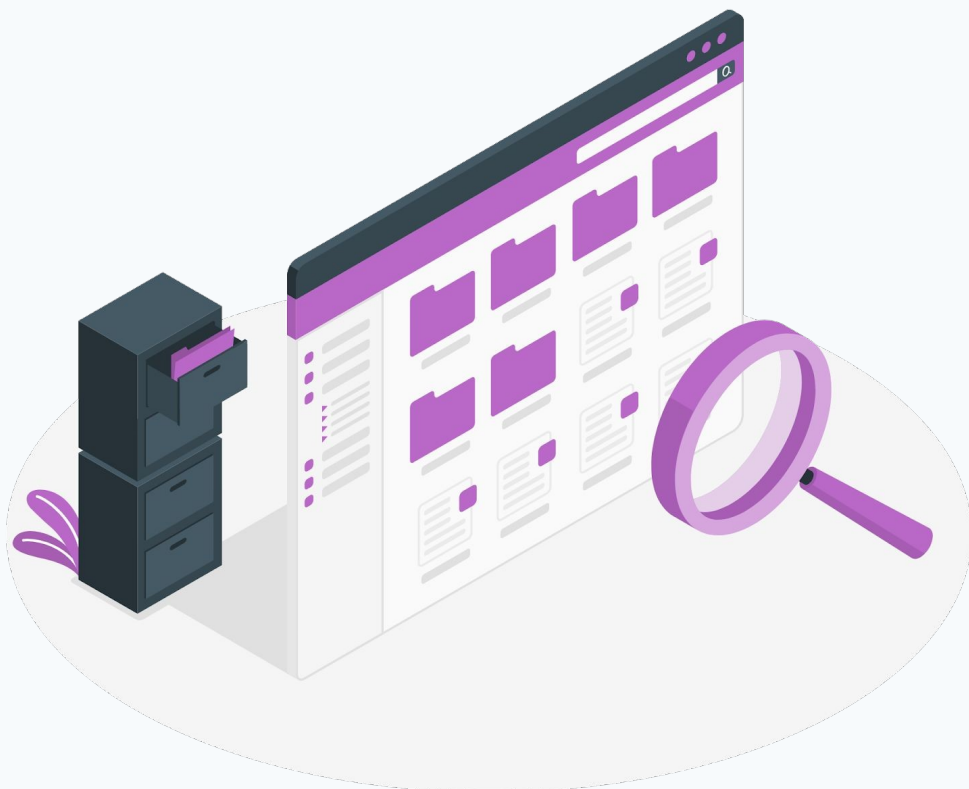


Quem nunca?

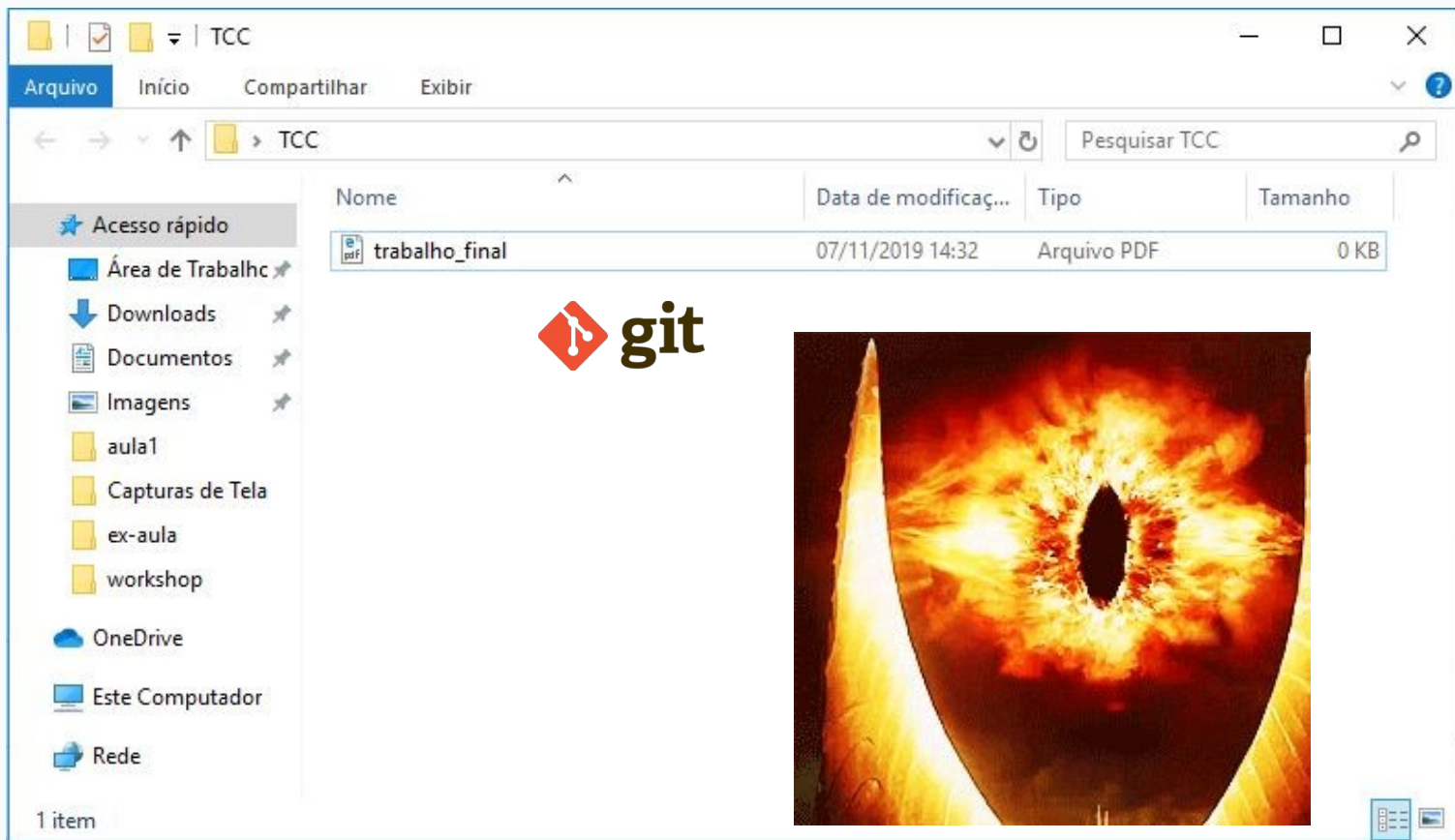




Necessidade de controle de versão

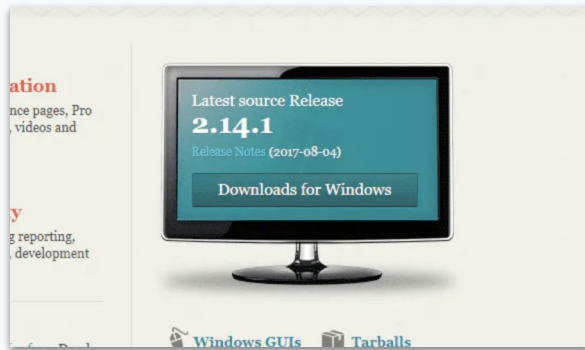


Como ficaria com o Git:



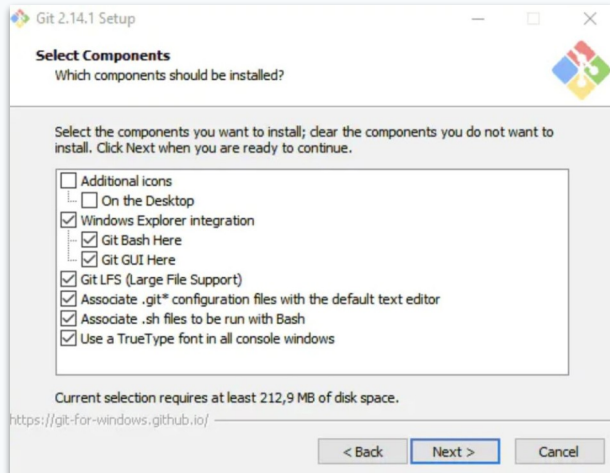


Instalando git na máquina:

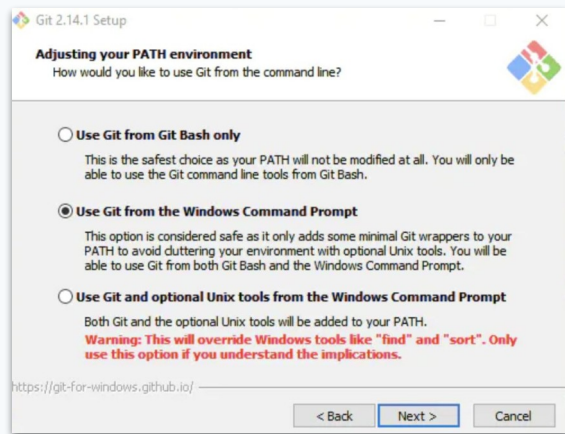
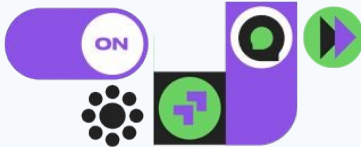


No site oficial do Git
(<https://git-scm.com/downloads>)
clique em “Downloads for Windows”.

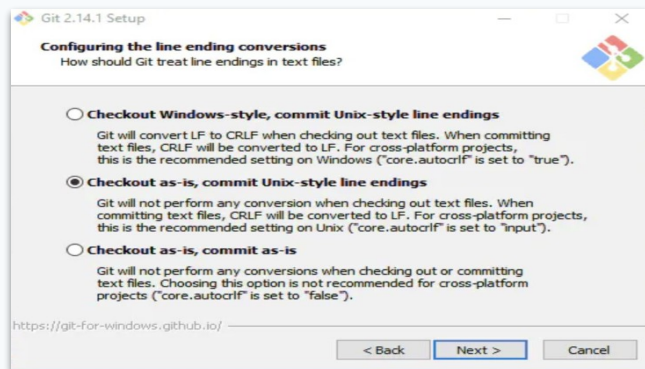
Execute o arquivo baixado e vá dando “Next” até a tela “Select Components”. Nesta tela escolho as opções como na imagem.



Essas opções me permitem abrir no prompt de comandos do Git (Git Bash) em qualquer pasta, basta clicar com o botão direito e “Git Bash Here”. A última opção também é interessante, porque instala uma fonte melhorzinha para o prompt de comandos.



Esta opção permite usar o comando `git` tanto no Git Bash quanto no terminal padrão do Windows



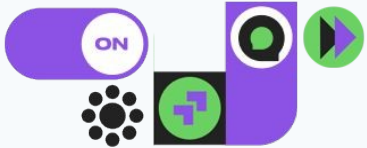
Essa opção converte o seu arquivo para padrão Unix quando você comita. Feito isso, “Next”, “Finish” e o Git está instalado

Como disponibilizar meu projeto para o mundo



Da minha máquina para o mundo



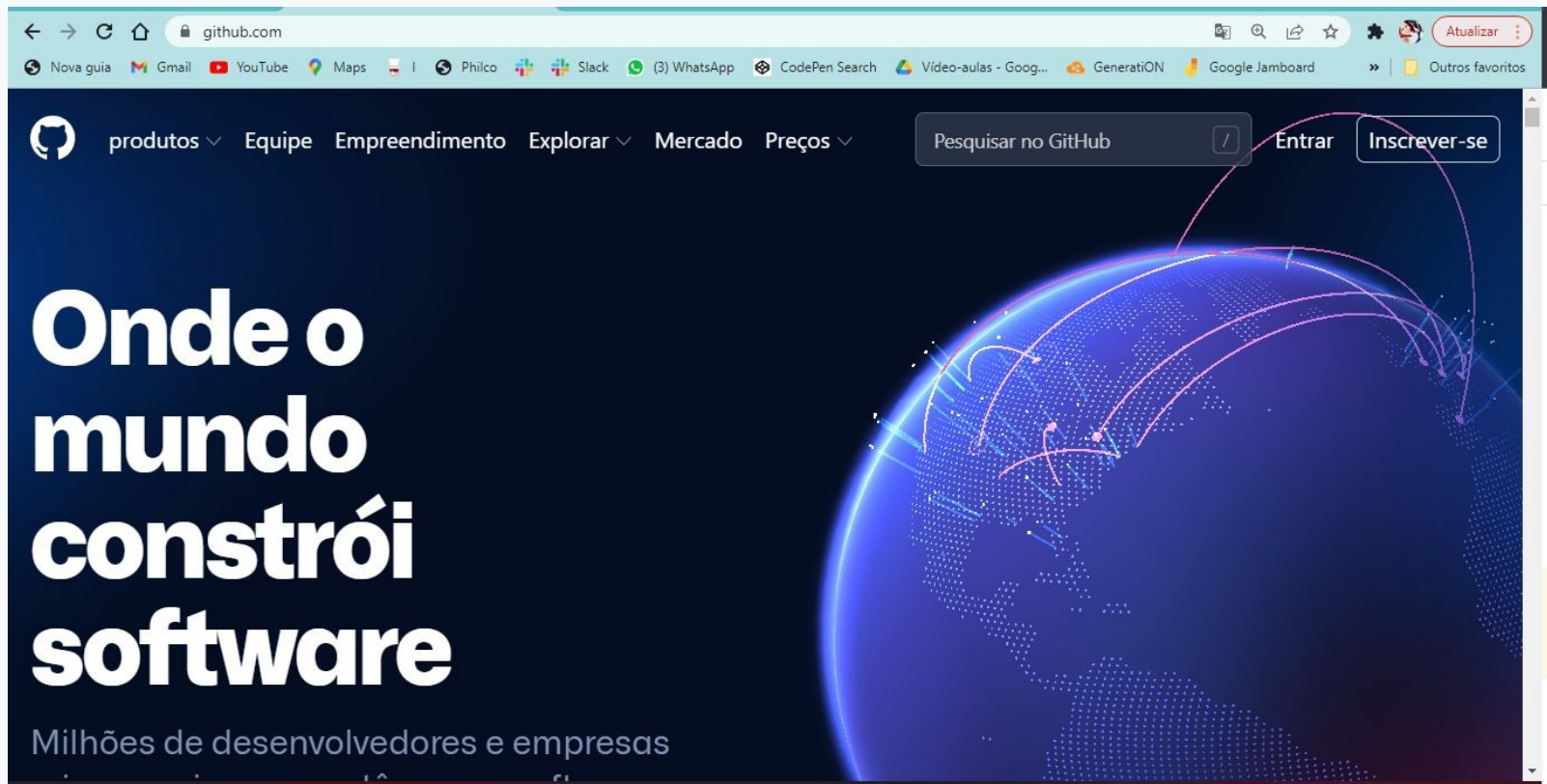


Hospedagem de repositórios





Escolhendo o site de hospedagem do nosso código

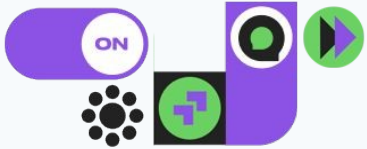




O que é GitHub?

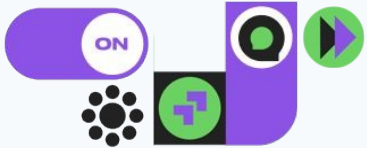
GitHub é uma plataforma de hospedagem de código-fonte com controle de versão usando o Git.

GitHub é amplamente utilizado por programadores para divulgação de seus trabalhos ou para que outros programadores contribuam com o projeto.

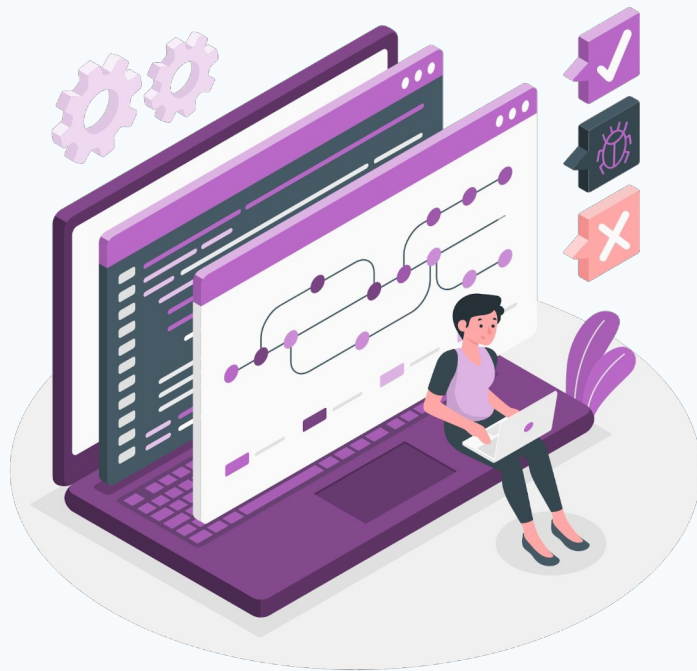


Bora nessa fazer nossa conta no GitHub?





Fluxo de trabalho

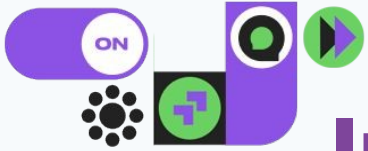




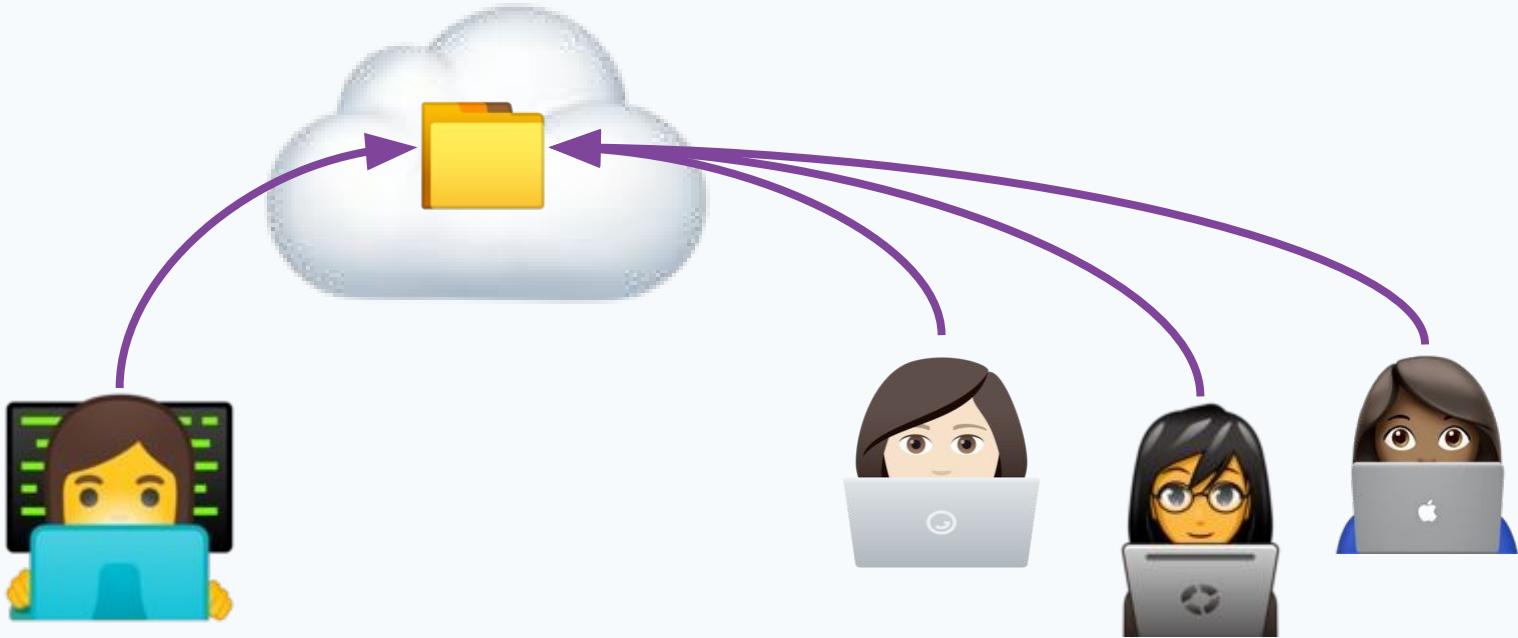
Como começar?



- **Este é o GIT. Ele rastreia o trabalho colaborativo em projetos através de um belo distribuído modelo de árvore da teoria dos grafos.**
- **Legal. Como usamos isso?**
- **Não tenho ideia. Apenas memorize esses comandos em shell e digite-os para sincronizar. Se você receber erros, salve seu trabalho em outro lugar, exclua o projeto e baixe uma nova cópia.**

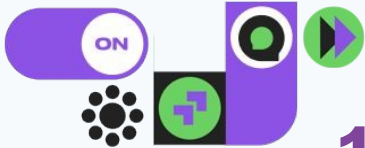


Imaginando esse cenário:



Criadora do software

Time de desenvolvimento



1. Rastreamento local do meu projeto

Criando repositório Local

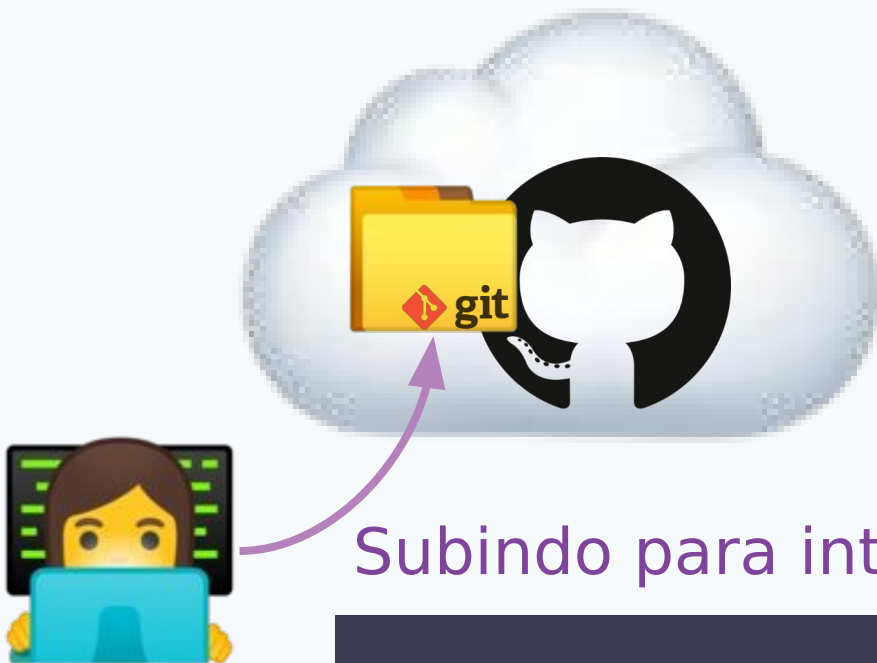


Rastreando Local

```
git init  
git add --all  
git commit -m "Meu maravilhoso projeto"
```




2. Disponibilizar o projeto na internet



Subindo para internet

```
git remote add origin https://github.com/bootwipro/meu-projeto.git  
git push
```

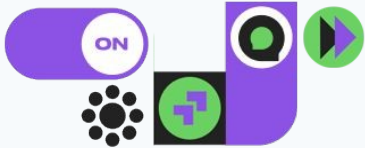


3. O time baixando o projeto



Clonando o projeto

```
git clone  
https://github.com/bootwipro/meu-projeto.git
```



4. Criando uma “cópia” do projeto

Não mexo no projeto
na branch master!

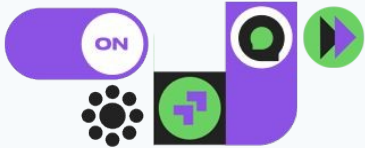


Altero o projeto que
está na branch nova

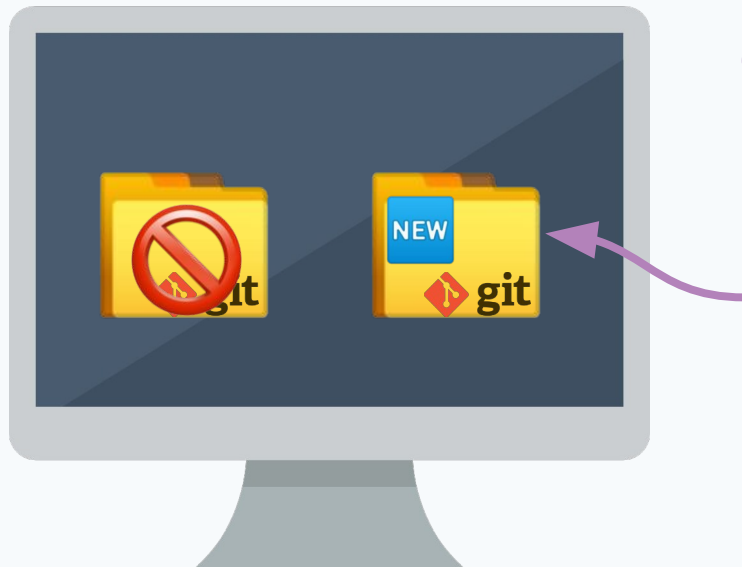


Criando branch

```
git checkout -b novaBranch
```



5. Salvando alterações na branch



Código novo

```
1 import java.util.Scanner;
2
3 public class WiproLoja {
4
5
6     public static void main(String[] args) {
7
8         Scanner leia = new Scanner(System.in);
9
10        //Variáveis:
11        char resposta;
12        int qtdeConsumir;
13        int contador = 1;
14        int produtoSelecionado;
15        boolean comprando = true;
16        boolean loop = true;
17        double tributos;
18        double desconto = 0;
```

Salvando...

```
git add --all
git commit -m "Adição de produtos"
```





6. Subindo alterações na nuvem



Subindo o projeto

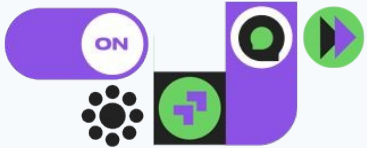
```
git push origin novaBranch
```





7. Adicionando as alterações no projeto



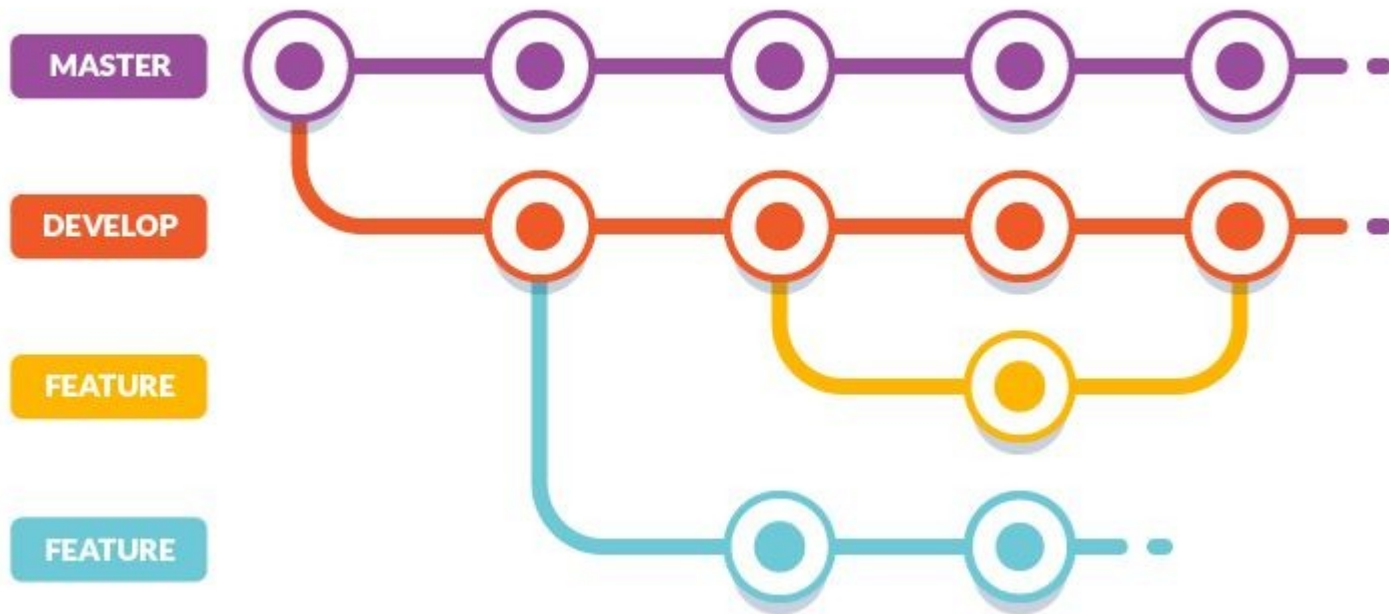


Outra explicação



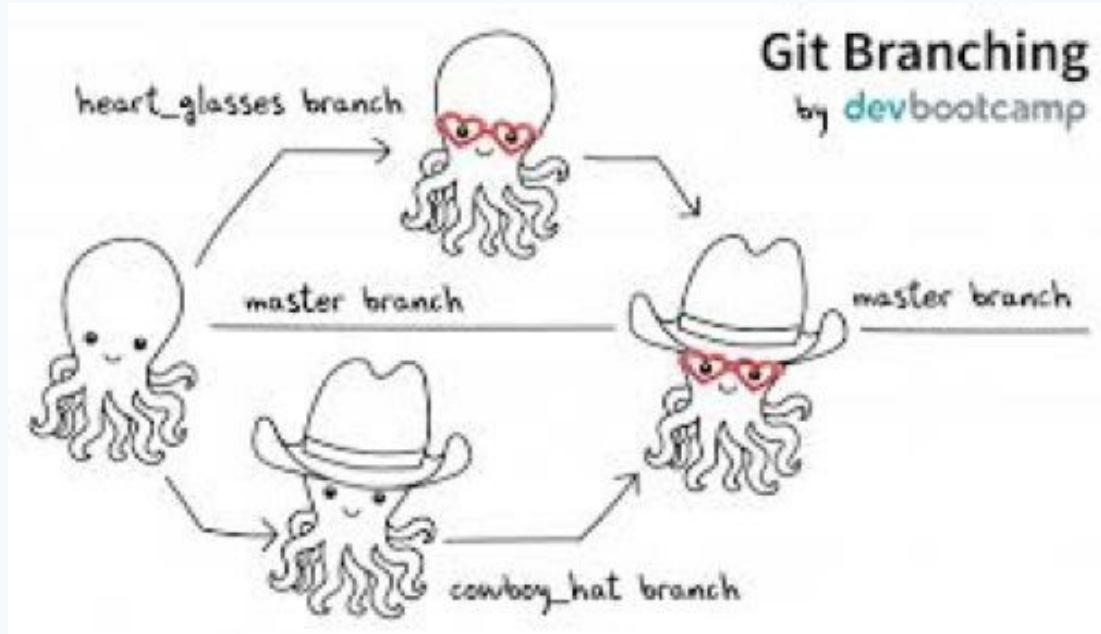


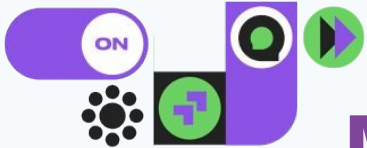
Git Flow





Git Flow





Mexendo no terminal

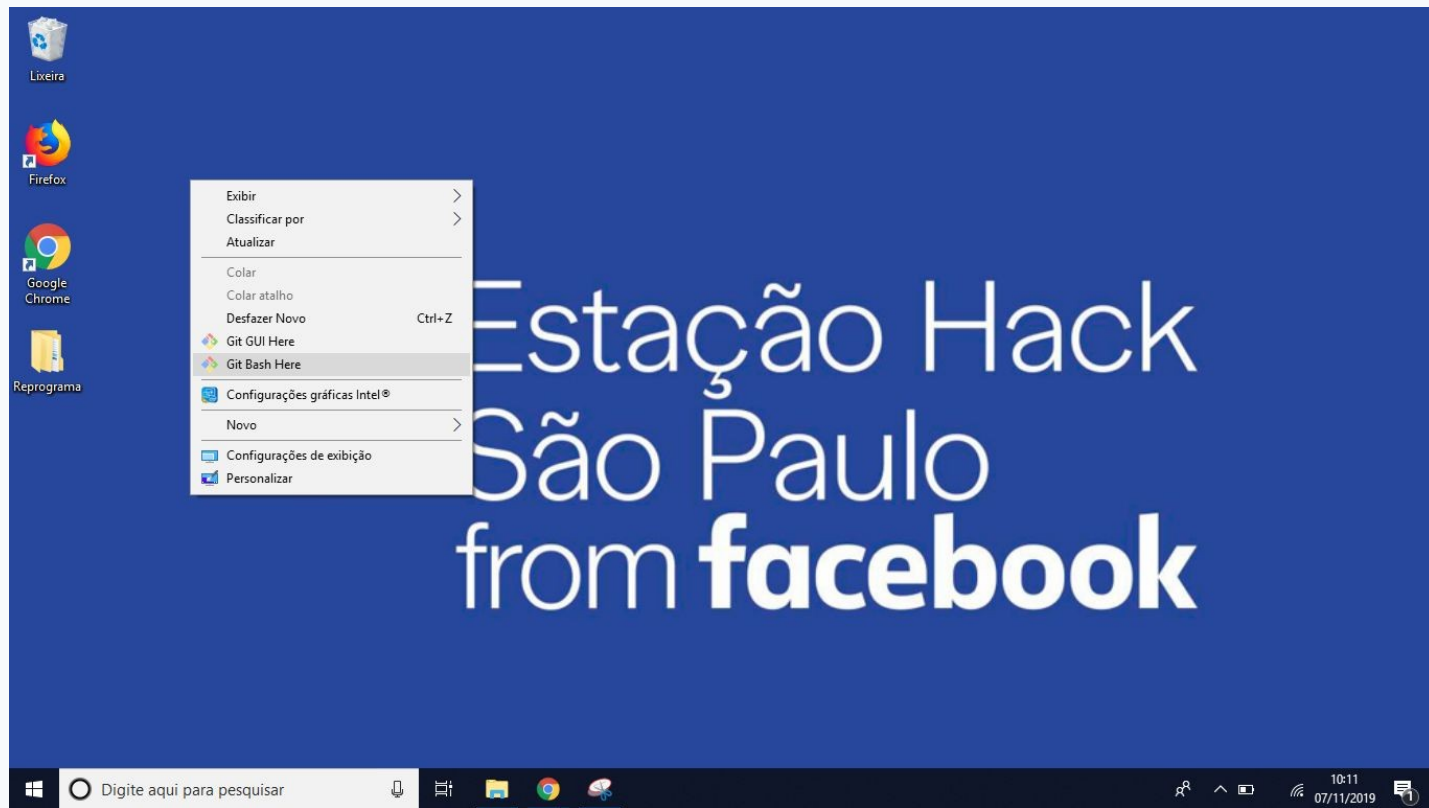


Vamos começar !



Abrindo o Git Bash

Git Bash here





Onde estou `pwd`

```
MINGW64:/c/Users/LENOVO/OneDrive/Área de Trabalho

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho
$ pwd
/c/Users/LENOVO/OneDrive/Área de Trabalho

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho
$ |
```



Listar o que tem aqui

```
MINGW64:/c/Users/LENOVO/OneDrive/Área de Trabalho

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Area de Trabalho
$ pwd
/c/Users/LENOVO/OneDrive/Area de Trabalho

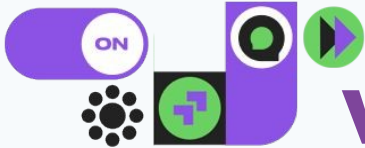
LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Area de Trabalho
$ ls
22ba2d63-5ca1-420b-9046-50af3f237553.pdf
291181c4-abd6-4b6b-9321-82730b01fc04.pdf
4079b85e-adc3-463c-a7c4-d7d79371bc2c.pdf
47ab71a2-8c8f-4ed9-8f35-3f066540e50d.pdf
9838012a-e8f7-4609-b8c4-4601267e7f5f.pdf
Banco PAN - Desafio final - BlueBank.docx.pdf'
Bancopan/
Challenge_Daikin_TDSOR.pdf
Contrato Salão.pdf'
Desafio1/
Eclipse IDE for Enterprise Java and Web Developers - 2021-09.lnk'*
GAMA PAN ACADEMY MARKETPLACE.gif'
Itaú.lnk*
```



Criar uma pasta mkdir WiproBoot

```
MINGW64:/c/Users/LENOVO/OneDrive/Área de Trabalho
Nova pasta'/
PSCS6 - Atalho.lnk'*
PanAcademy/
Pessoa 2 - Chrome.lnk'*
PrimeiroDesafioT1/
SuperPanStore/
WhatsApp.lnk*
Zoom.lnk*
a986b601-2696-41eb-9e16-eba631451549.pdf
pyback-main/
c9378c10-68ce-4893-be04-6e54eaeddcae.pdf
da05c971-740f-4950-b5fd-d77375b87ba0 (1).pdf'
da05c971-740f-4950-b5fd-d77375b87ba0.pdf
desktop.ini
Fc5f5c69-136d-44e5-9dec-5e278fbc6110.pdf
start_mongodb.bat
teste/
teste1/

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho
mkdir WiproBoot
```

Verificar a pasta no Desktop



PSCS6 -
Atalho



Verificar pelo terminal 1s

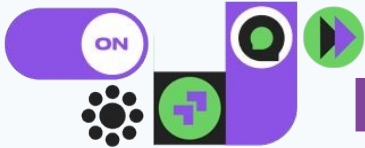
```
MINGW64:/c/Users/LENOVO/OneDrive/Área de Trabalho
Microsoft Teams.lnk'*
MongoDBCompass.lnk*
Nova pasta/
PSCS6 - Atalho.lnk'*
PanAcademy/
Pessoa 2 - Chrome.lnk'*
PrimeiroDesafioT1/
SuperPanStore/
WhatsApp.lnk*
WiproBoot/
Zoom.lnk*
a986b601-2696-41eb-9e16-eba631451549.pdf
byback-main/
c9378c10-68ce-4893-be04-6e54eaeddcac.pdf
da05c971-740f-4950-b5fd-d77375b87ba0 (1).pdf'
da05c971-740f-4950-b5fd-d77375b87ba0.pdf
desktop.ini
fc5f5c69-136d-44e5-9dec-5e278fbc6110.pdf
start_mongodb.bat
teste/
teste1/

ENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho
```



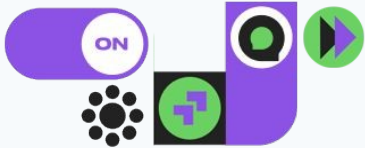

Entrar na pasta `cd WiproBoot`

```
MINGW64:/c/Users/LENOVO/OneDrive/Área de Trabalho
LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Area de Trabalho
cd WiproBoot|
```



Entrar na pasta cd WiproBoot

```
MINGW64:/c/Users/LENOVO/OneDrive/Área de Trabalho
LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho
cd WiproBoot|
```



Criar arquivo index.html com o texto *oi*

```
echo "oi" > index.html
```

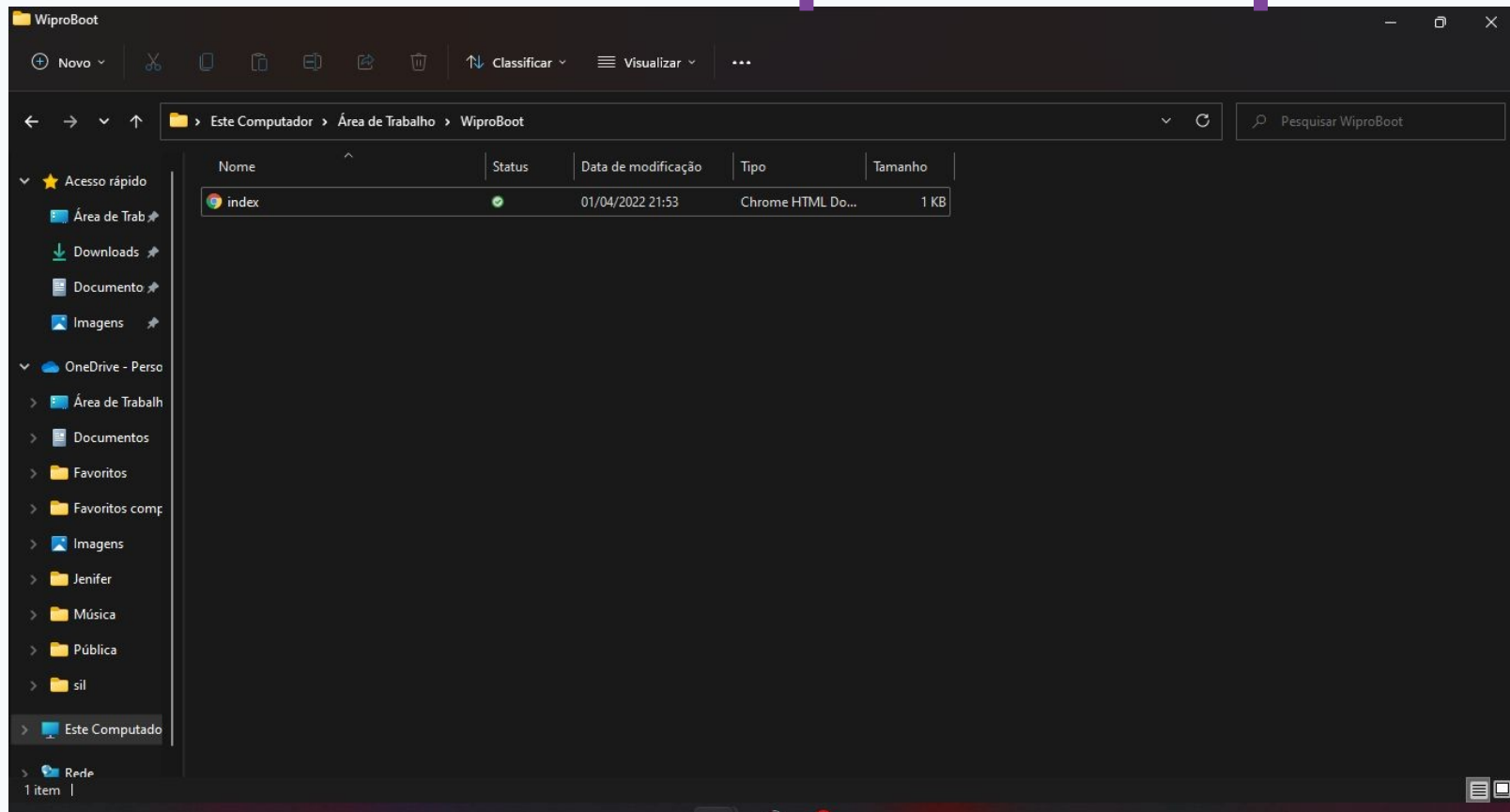
```
MINGW64:/c/Users/LENOVO/OneDrive/Área de Trabalho/WiproBoot

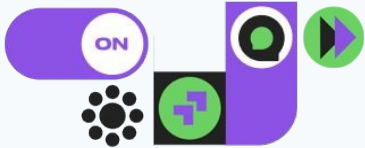
ENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho
cd WiproBoot

ENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
echo "oi" > index.html
```



Visualizando pelo Desktop





Visualizando pelo terminal

```
MINGW64:/c/Users/LENOVO/OneDrive/Área de Trabalho/WiproBoot

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho
$ cd WiproBoot

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ echo "oi" > index.html

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ ls
index.html

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ |
```



Deletando o arquivo `rm index.html`

```
MINGW64:/c/Users/LENOVO/OneDrive/Area de Trabalho/WiproBoot

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Area de Trabalho
$ cd WiproBoot

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Area de Trabalho/WiproBoot
$ echo "oi" > index.html

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Area de Trabalho/WiproBoot
$ ls
index.html

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Area de Trabalho/WiproBoot
$ rm index.html

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Area de Trabalho/WiproBoot
$
```



Verificando pelo terminal

```
MINGW64:/c:/Users/LENOVO/OneDrive/Área de Trabalho/WiproBoot

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho
$ cd WiproBoot

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ echo "oi" > index.html

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ ls
index.html

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ rm index.html

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ ls

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ |
```



Voltando uma pasta acin

cd ..

```
MINGW64:/c/Users/LENOVO/OneDrive/Área de Trabalho

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho
$ cd WiproBoot

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ echo "oi" > index.html

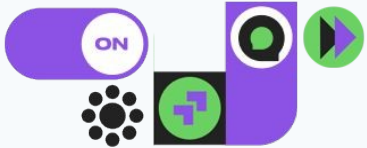
LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ ls
index.html

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ rm index.html

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ ls

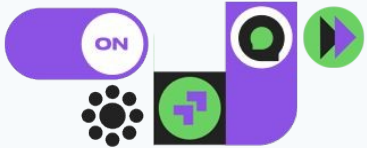
LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho/WiproBoot
$ cd ..

LENOVO@DESKTOP-FQ346Q2 MINGW64 ~/OneDrive/Área de Trabalho
$ |
```

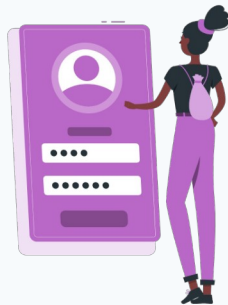



Parabéns! Usamos o Git Bash!

Terminal █



Configuração



Autoria no Git



Git vai precisar saber seu nome e e-mail

- Comando para dizer qual seu nome:

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

- Comando para dizer qual seu email:

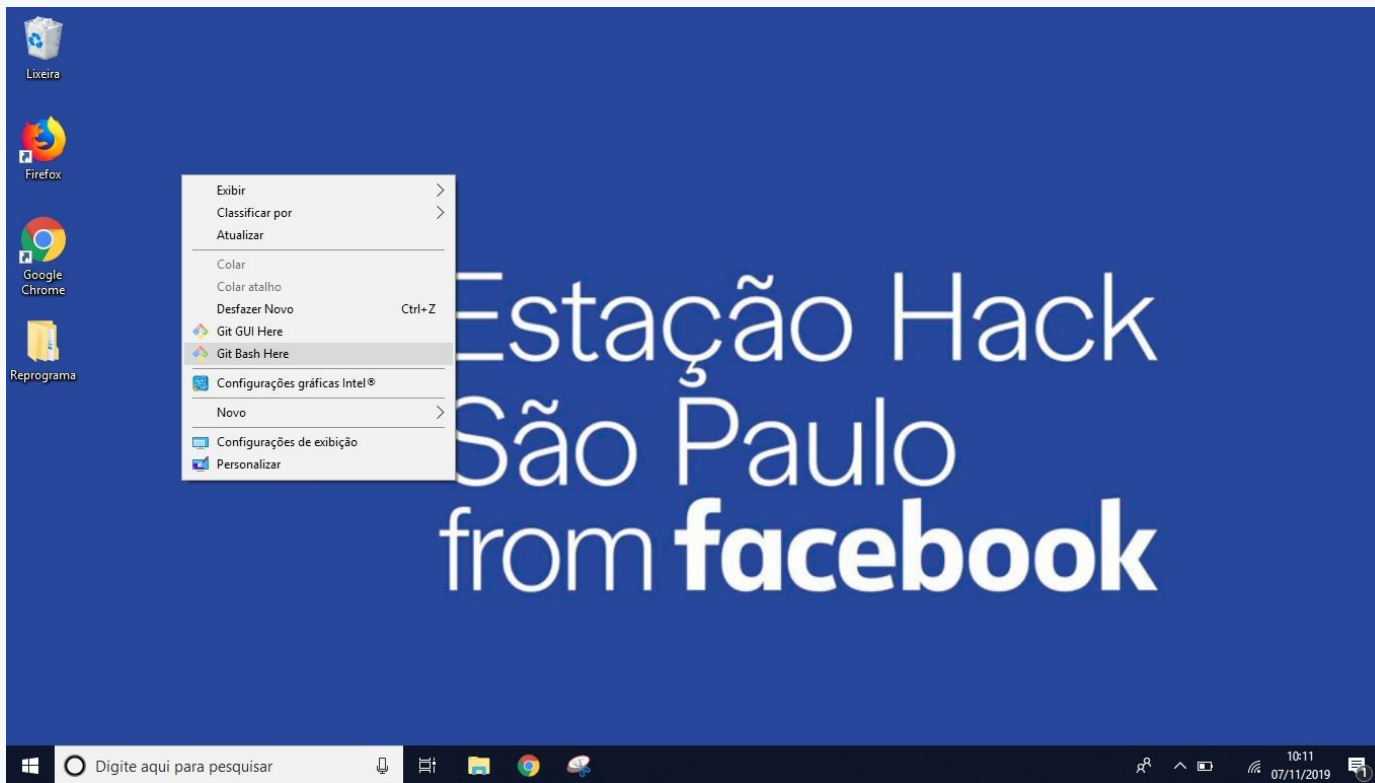
```
git config --global user.email "seu@email.com"
```

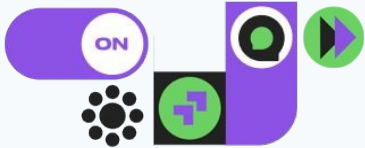
- Comando de verificação se a configuração foi realizada com sucesso:

```
git config --list
```



Botão direito do mouse no Desktop e Git Bash here



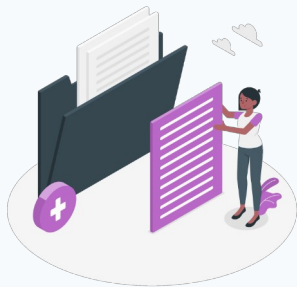


Avisar o Git quem sou eu:

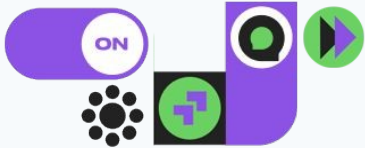
```
git config --global user.name "Jenifer Plácido"  
git config --global user.email  
"jenifer@gmail.com"  
git config --list
```



Iniciando o controle de versão



Versionamento local



Queremos rastrear com o Git





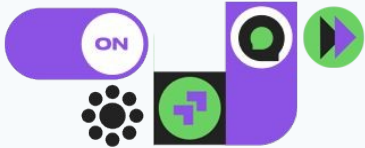
Git - configuração inicial

1. Crie conta GitHub: <https://github.com/login>
2. Instale o Git na sua máquina
3. Configure suas informações (nome e email Github)
 - a. `git config --global user.name "Seu Nome"`
 - b. `git config --global user.email "Seu Email"`
4. Verifique suas informações
 - a. `git config --list` OU
 - b. `git config user.name` OU
 - c. `git config user.email`



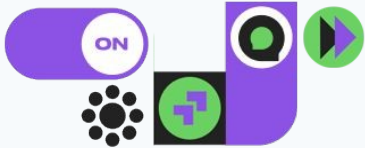
Terminal - comandos básicos

ls (macOS/Linux) dir (Windows)	lista todos os arquivos presentes no diretório atual (macOS/Linux)
mkdir nome-da-pasta	cria uma nova pasta
cd nome-da-pasta	navega para a pasta especificada (exemplo: cd documentos)
cd ..	sobe um nível de pasta
touch nome-do-arquivo dir > nome-do-arquivo	cria um novo arquivo
clear	limpa todas as informações do terminal
<nome do programa> --version	Exibe versão instalada

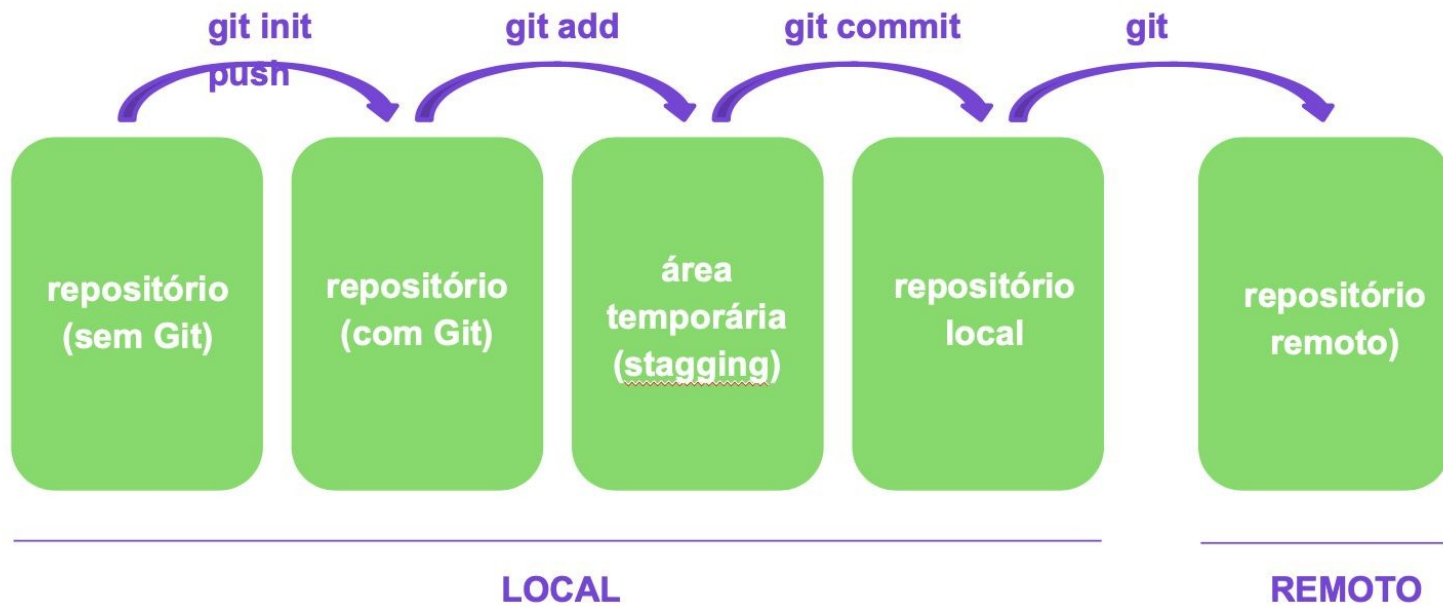


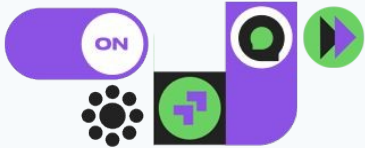
Git - comandos básicos

git init	inicializa o git no repositório local
git add nome-do-arquivo ou .	adiciona um arquivo modificado ao staging (área temporária)
git status	mostra os status dos arquivos modificados
git commit -m "mensagem"	cria um commit
git pull	puxa as atualizações mais recente (remoto -> local)
git push	envia as atualizações mais recentes (local -> remoto)
git remote add origin <caminho>	adiciona o seu repositório local ao remoto
git checkout -- <nome-arquivo>	descarta as alterações locais do arquivo informado



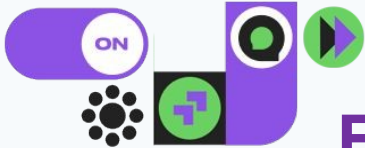
Git - comandos básicos





Dicas importantes

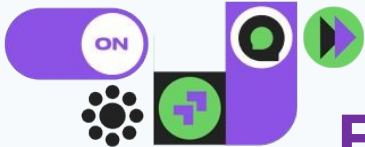
1. As mensagens do commit devem ser claras e sucintas, descrevendo bem a alteração
2. Um push só funciona se houver um commit feito!
3. Não se desespere, Git pode parecer complicado no começo mas depois você faz de olhos fechados (ou quase isso haha)



Exercícios

1. Crie um repositório localmente e inicialize o git
2. Adicione um arquivo markdown chamado README com seu nome e prato favorito e faça um commit
3. Adicione uma curiosidade sobre você e faça outro commit
4. Publique o repositório no seu GitHub

[Guia básico sobre markdown](#)



Exercício Extra

1. Faça um fork do repositório.
2. Clone o repositório para a sua máquina.
3. Crie uma nova branch com seu nome (exemplo: jenifer).
4. Faça commits com a resolução dos exercícios.
5. Atualize seu repositório remoto.

3. DESAFIO EXTRA:

Abra um Pull Request para o repositório original

Obrigada

