

Git & GitHub

Introdução

Prática de Laboratório
Wesley Dias Maciel / Thiérs Hofman

2023
Exercício 01

Objetivo:

Instalar o Git e criar conta no GitHub.

Aluno: vanderson henrique da silva correia filho

Ra: 4231922369

Git.

Sistema de controle de versão (Version Control System, VCS). Criado pelo Linus Torvalds.



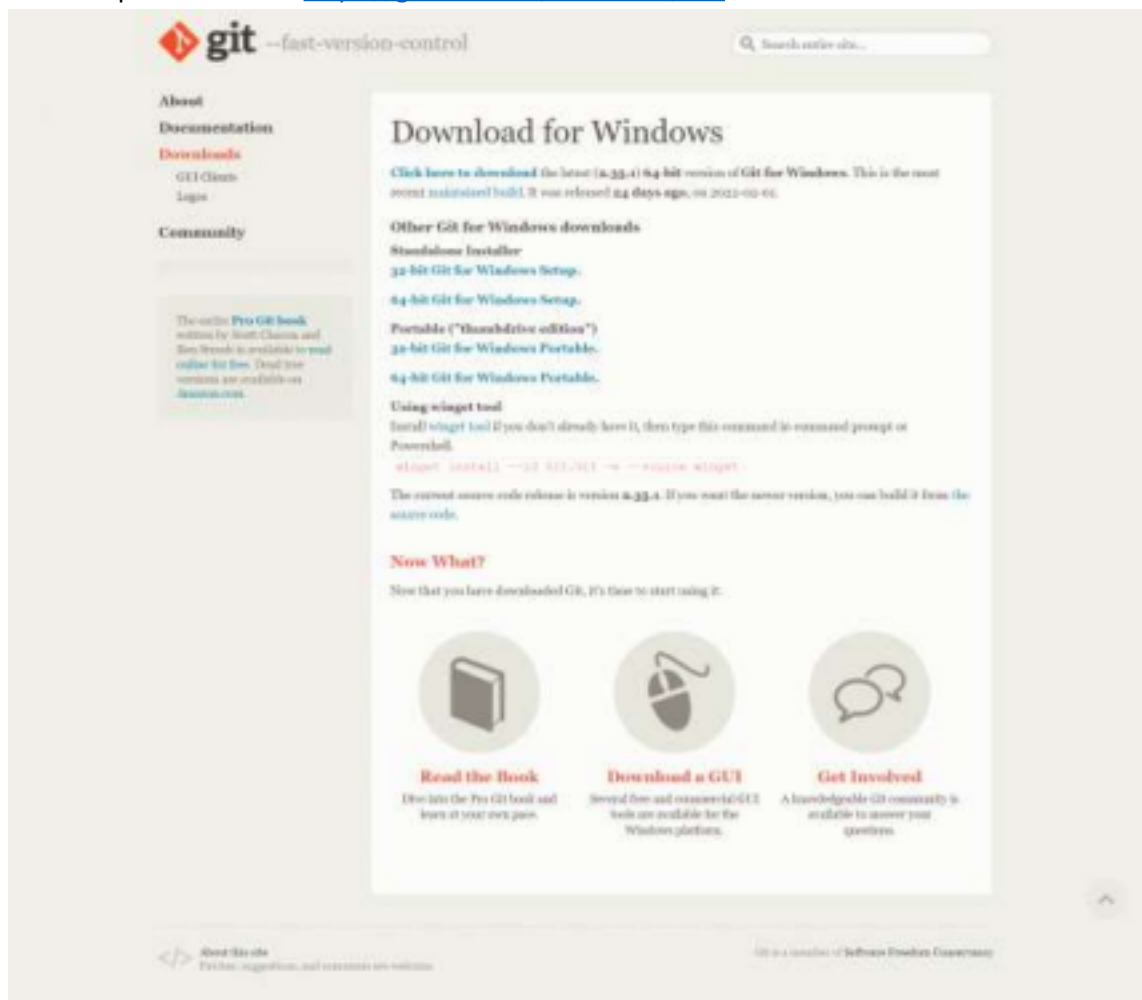
“O controle de versão é um sistema que registra as mudanças feitas em um arquivo ou em um conjunto de arquivos ao longo do tempo de forma que você possa recuperar versões específicas. Mesmo que os exemplos desse livro mostrem arquivos de código fonte sob controle de versão, você pode usá-lo com praticamente qualquer tipo de arquivo em um computador”.

(Scott Chacon e Ben Strub, 2 Edição, 2014)

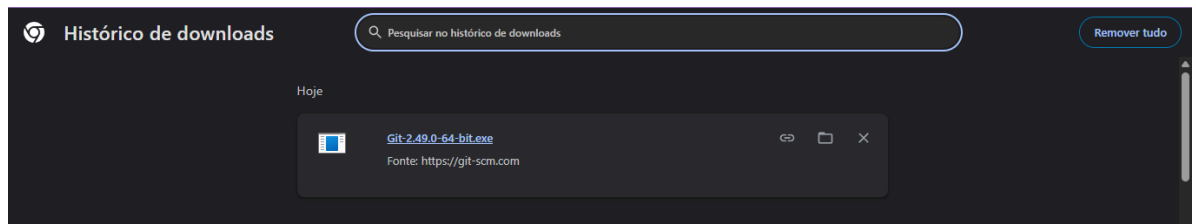
1) Página: <https://git-scm.com/>.



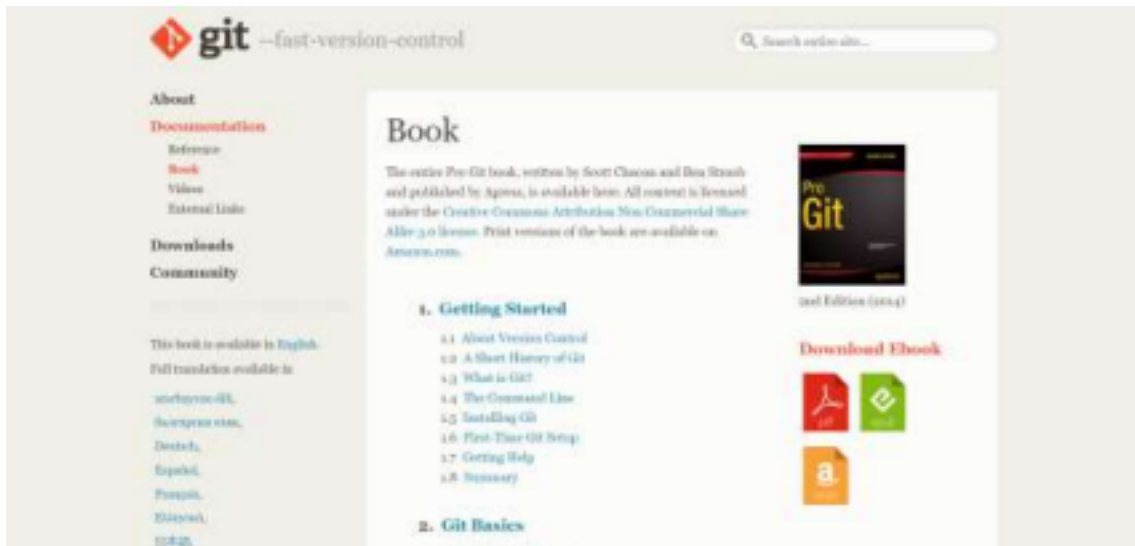
2) Download para Windows: <https://git-scm.com/download/win>.

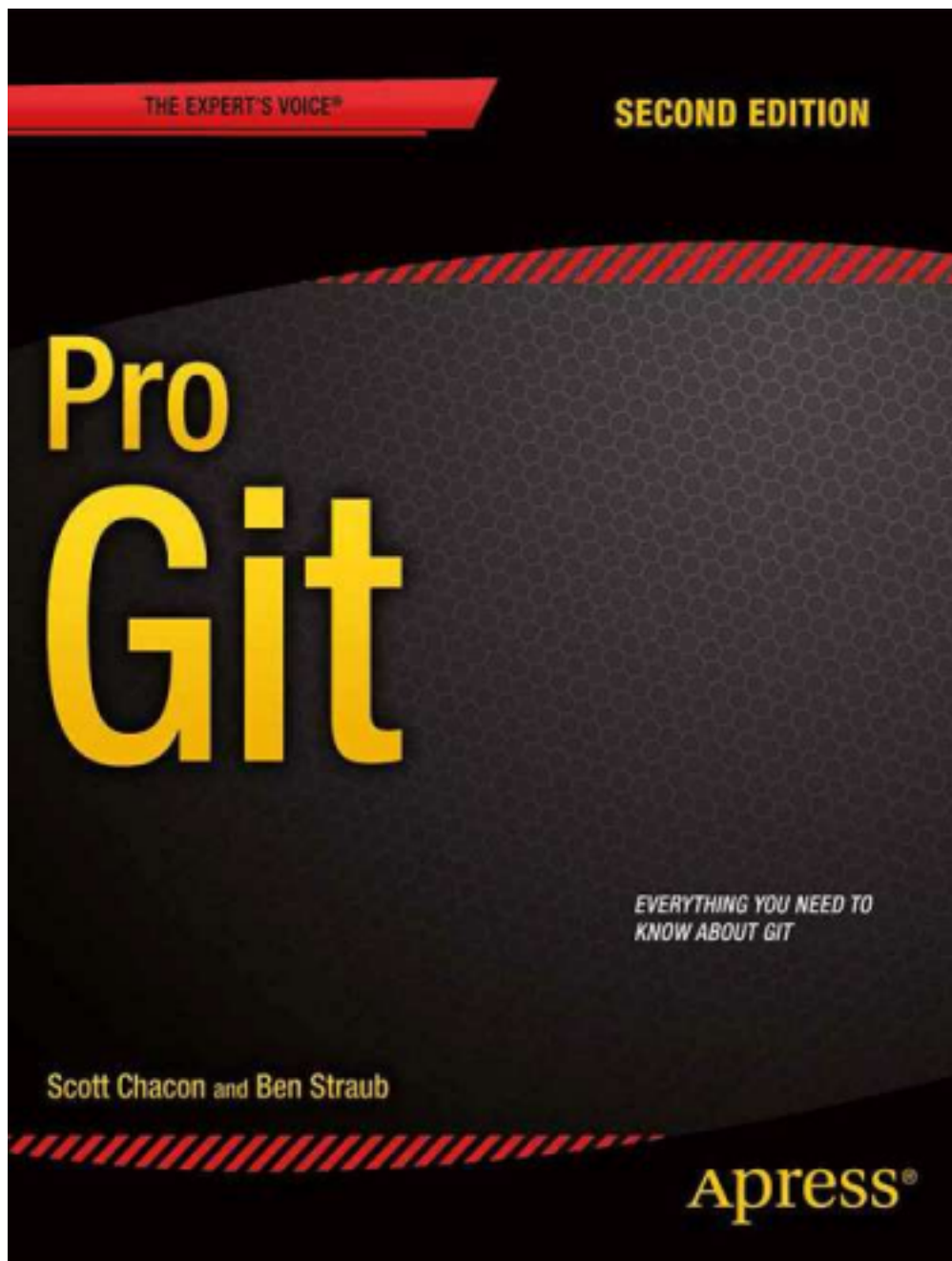


3) Instalar o Git localmente em sua máquina, seguindo os passos do instalador

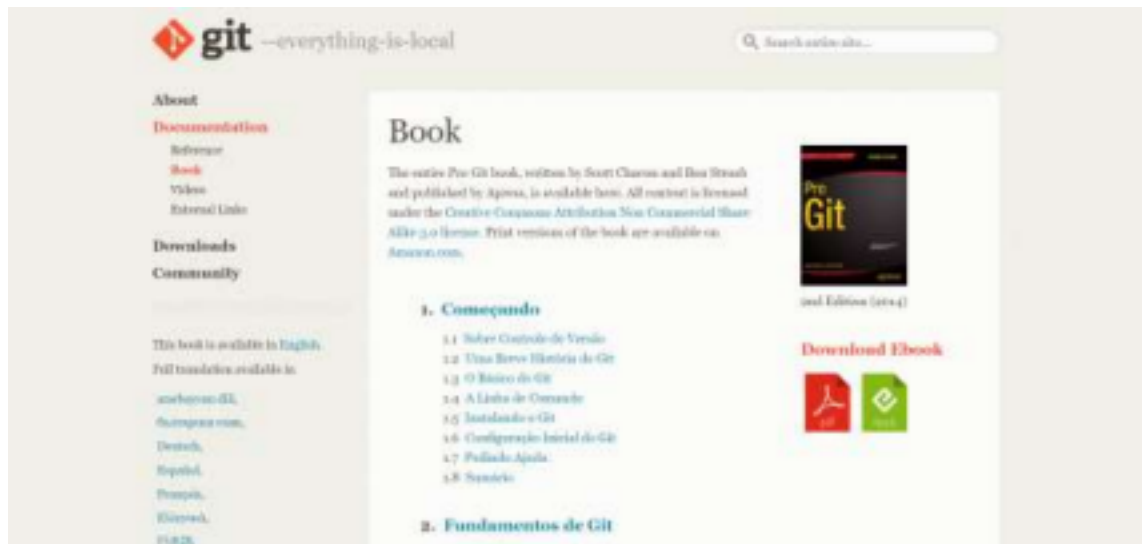


4) Livro: <https://git-scm.com/book/en/v2>



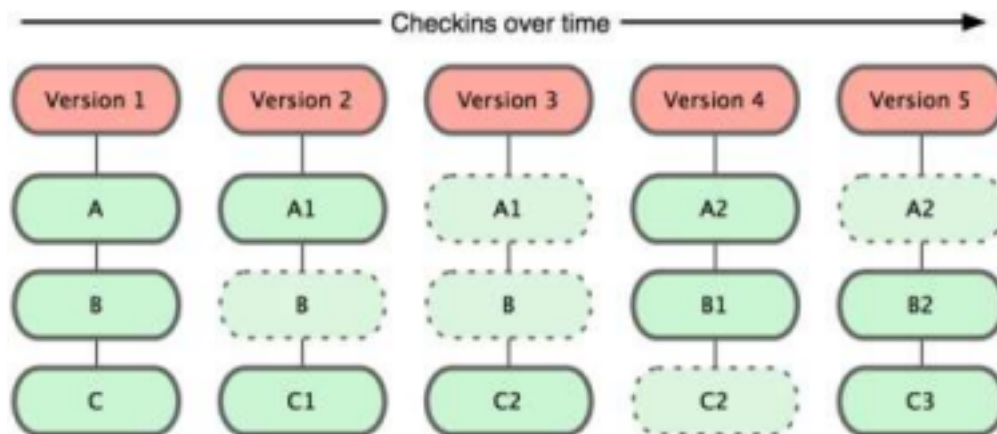


5) Link para o livro em português: <https://git-scm.com/book/pt-br/v1/Primeiros-passos>



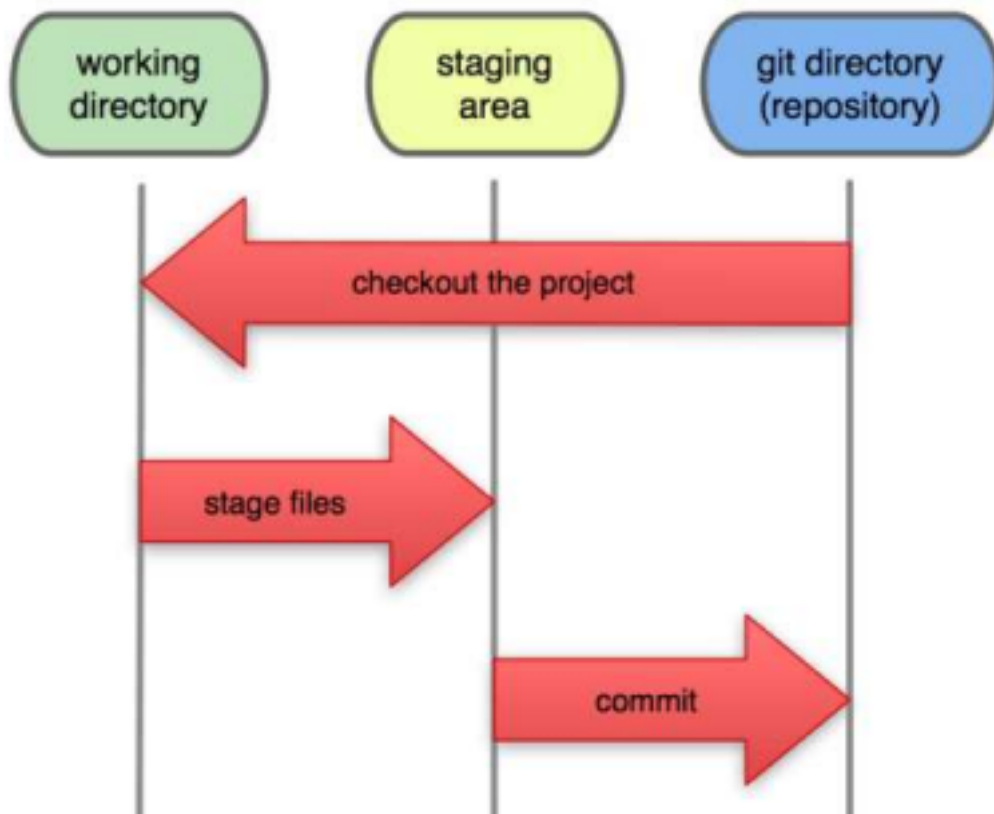
6) Curso: <https://www.udemy.com/course/git-e-github-para-iniciantes/>

7) Git armazena dados como “snapshots” do projeto ao longo do tempo.



8) Os três estados: diretório de trabalho, área de preparação (“staging area”) e o diretório do Git.

Local Operations



Diretório do Git:

- 1) Local onde o Git armazena os metadados e o banco de objetos de seu projeto.
- 2) Essa é a parte mais importante do Git.
- 3) É a parte copiada quando você clona um repositório de outro computador.

Diretório de trabalho:

- 1) Uma versão do projeto que está sendo trabalhada.
- 2) Esses arquivos são obtidos a partir da base de dados comprimida no diretório do Git e colocados em disco para que você possa utilizar ou modificar.

“Staging área”:

- 1) É um arquivo, geralmente armazenado no diretório do Git.
- 2) Armazena informação sobre o que entrará no próximo “commit”.
- 3) Algumas vezes, é chamado de índice, “index”, “stage” ou “staging area”.

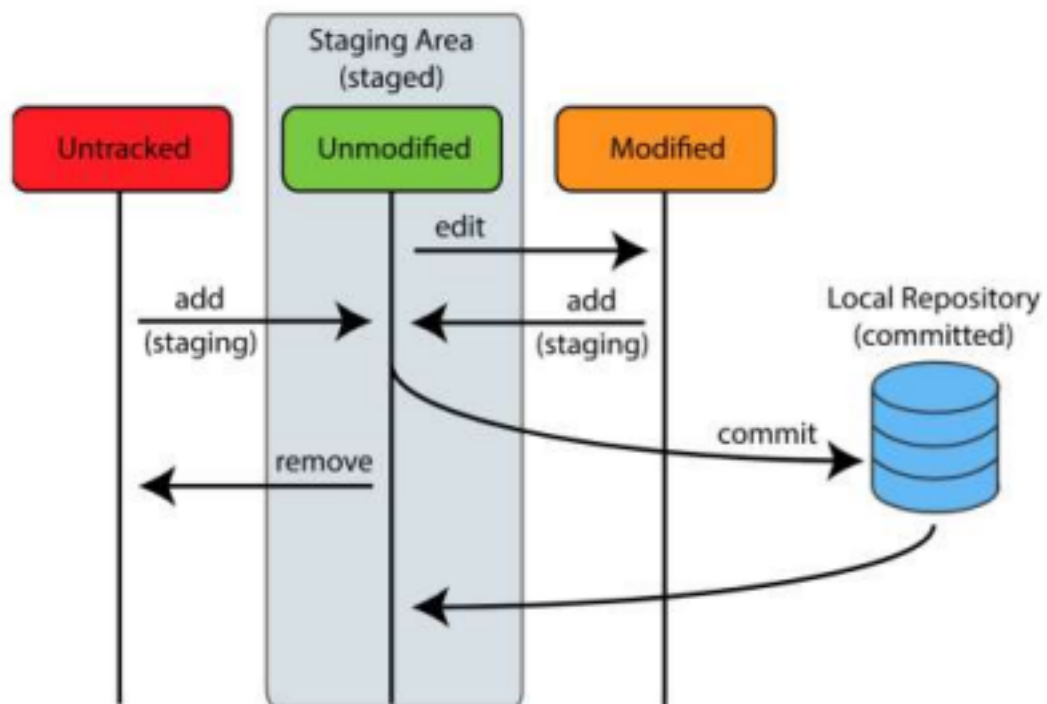
Fluxo básico do Git:

- 1) Você modifica arquivos no seu diretório de trabalho.
- 2) Você seleciona os arquivos, adicionando “snapshots” deles na “staging area”.
- 3) Você faz um “commit”, que leva os arquivos como eles estão na “staging area” e os armazena permanentemente no seu diretório do Git.

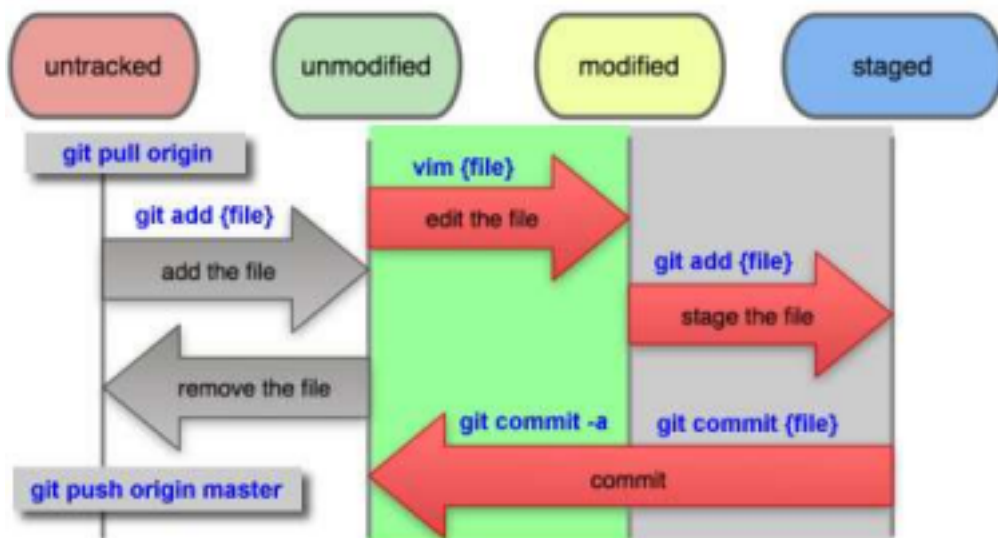
OBS:

- 1) Se uma versão de um arquivo está no diretório do Git, ela é considerada “committed”.
- 2) Se uma versão de um arquivo foi adicionada à “staging area”, ela é considerada “staged”.
- 3) Se uma versão de um arquivo foi alterada, mas não foi adicionada à “staging area”, ela é considerada modificada.

9) Ciclo de vida dos arquivos.



File Status Lifecycle



- 1) **untracked**: arquivo adicionado ao projeto, mas ainda não rastreado pelo Git. O arquivo ainda não é conhecido pelo Git, não faz parte do diretório do Git. O arquivo não é monitorado pelo Git.
- 2) **unmodified**: arquivo adicionado ao Git e ainda não modificado. O arquivo faz parte do diretório e não foi alterado.
- 3) **modified**: arquivo do diretório e que foi modificado.
- 4) **staged**: arquivos que farão parte de uma nova versão, quando o usuário executar o próximo comando “commit”. Esses arquivos são considerados “unmodified”.
- 5) **committed**: arquivos que fazem parte de uma versão.

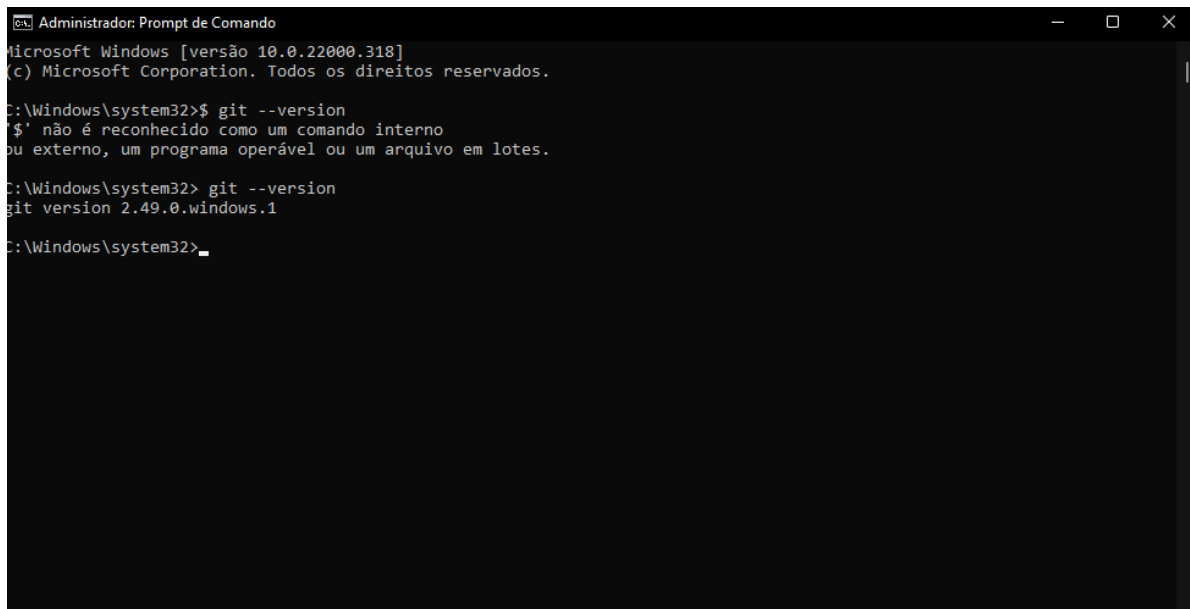
Exercício 02

Git: configurações iniciais.

- 1) Num terminal de sua máquina local, verificar a instalação local, reportando a versão do Git instalada.

\$ git --version

git version 2.35.1.windows.2



```
Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.22000.318]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Windows\system32>$ git --version
'$' não é reconhecido como um comando interno
ou externo, um programa operável ou um arquivo em lotes.

C:\Windows\system32> git --version
git version 2.49.0.windows.1

C:\Windows\system32>_
```

git config

Configurações para todo o sistema, todos os usuários: informar o parâmetro “--system”.

Configurações para o usuário corrente: informar o parâmetro “**--global**”. Configurações locais do projeto: não informar parâmetro, deixar em branco.

2) Definir nome, e-mail e editor de texto padrão do usuário.

```
$ git config --global user.name "Wesley Dias Maciel"
```

```
$ git config --global user.email "wesleydiasmaciel@gmail.com"
```

```
$ git config --global core.editor notepad
```

OBS: VS Code:

```
git config --global core.editor "code --wait"
```

ou

```
git config --global core.editor "code"
```

```
C:\Windows\system32>git config --global user.name "Vanderson henrique da silva correia filho"
C:\Windows\system32>git config --global user.email "vandersonyi@gmail.com"
C:\Windows\system32>git config --global core.editor notepad
```

3) Exibir as configurações fornecidas.

```
$ git config user.name
```

Wesley Dias Maciel

```
$ git config user.email
```

wesleydiasmaciel@gmail.com

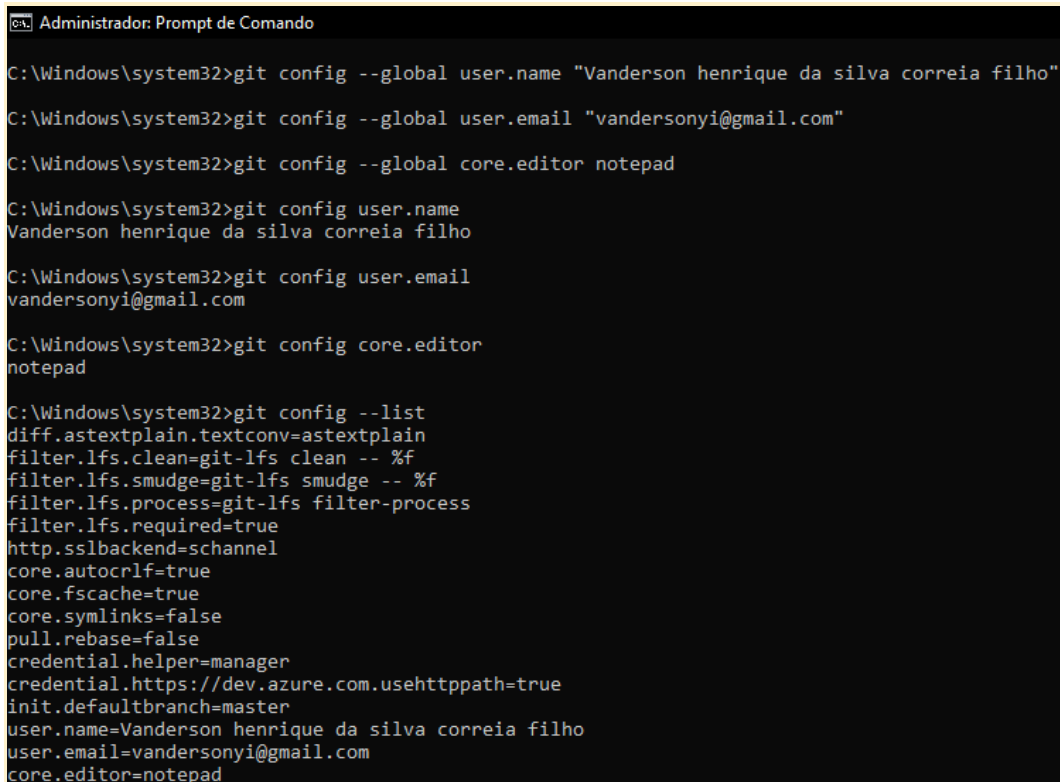
```
$ git config core.editor
```

notepad

```
$ git config --list
```

```
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
```

```
rebase.autosquash=true
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
http.sslbackend=openssl
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.required=true
filter.lfs.process=git-lfs filter-process
credential.helper=manager
user.name=Wesley Dias Maciel
user.email=wesleydiasmaciel@gmail.com
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
core.editor=notepad
```



```
Administrador: Prompt de Comando

C:\Windows\system32>git config --global user.name "Vanderson henrique da silva correia filho"
C:\Windows\system32>git config --global user.email "vandersonyi@gmail.com"
C:\Windows\system32>git config --global core.editor notepad
C:\Windows\system32>git config user.name
Vanderson henrique da silva correia filho
C:\Windows\system32>git config user.email
vandersonyi@gmail.com
C:\Windows\system32>git config core.editor
notepad
C:\Windows\system32>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Vanderson henrique da silva correia filho
user.email=vandersonyi@gmail.com
core.editor=notepad
```

Exercício 03

GitHub.

Repositório remoto, público ou privado. Plataforma de hospedagem (de código-fonte) que usa o Git como sistema de controle de versão. Local na Web para armazenamento de projetos. Também usado como rede social.

1) Página: <https://github.com/>.



Why GitHub? · Team · Enterprise · Explore · Marketplace · Pricing ·

Search GitHub

Sign in

Sign up

Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

Email address

Sign up for GitHub

73+ million

developers

4+ million

organizations

200+ million

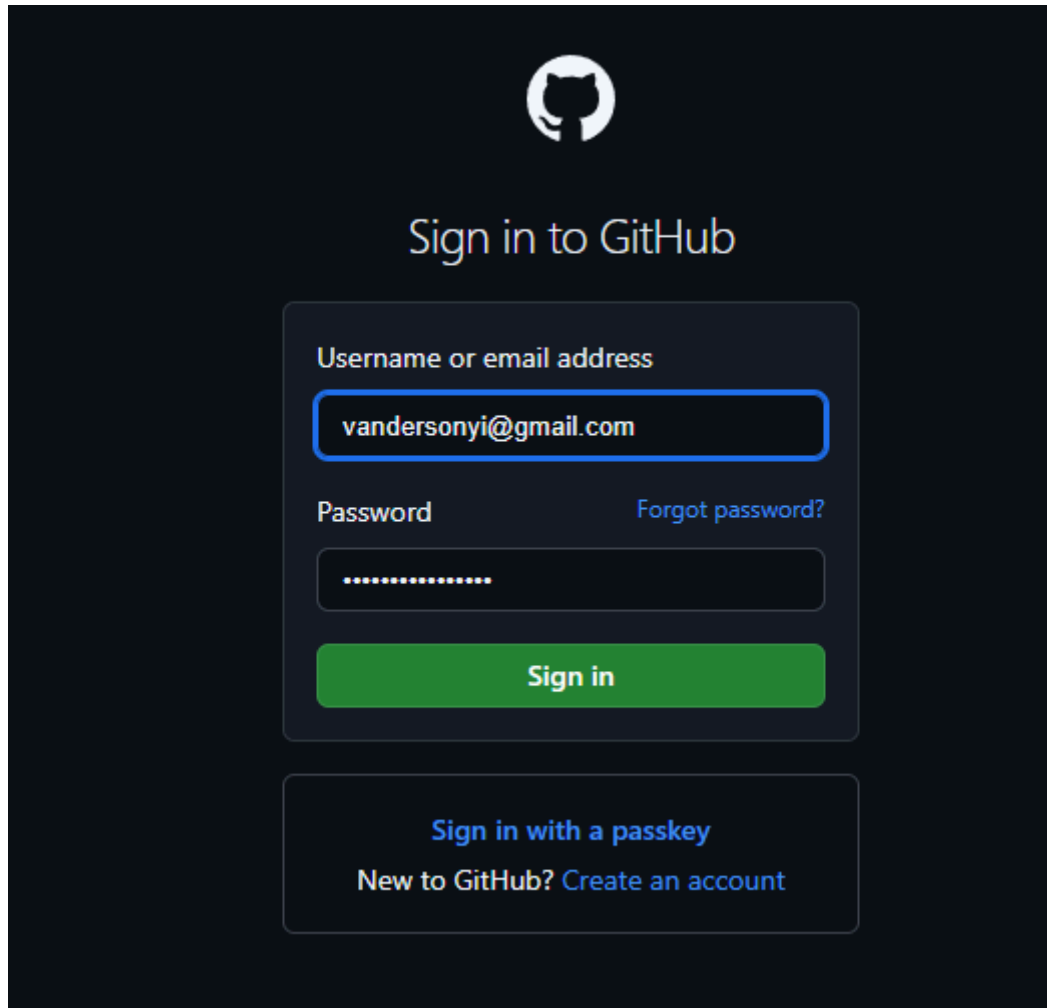
repositories

84%

market share



2) Criar uma conta e entrar: <https://github.com/login>.

The image shows the GitHub login page. At the top center is the GitHub logo (Octocat). Below it, the text "Sign in to GitHub" is displayed. The main form is a dark gray box with rounded corners. It contains two input fields: "Username or email address" with the value "vandersonyi@gmail.com" and "Password" with masked characters. To the right of the password field is a link "Forgot password?". Below the inputs is a green "Sign in" button. At the bottom of the form is a section with the text "Sign in with a passkey" and "New to GitHub? Create an account".

Sign in to GitHub

Username or email address

vandersonyi@gmail.com

Password

Forgot password?

Sign in

Sign in with a passkey

New to GitHub? [Create an account](#)

3) Livro:



Resumo

- Comandos Básicos

- **git init** //Inicia um repositório e começa a observar as alterações do projeto. **git add .** //Realiza adições na staging area.
- git commit -m 'mensagem'** //Cria uma versão.
- git remote add origin <url do repositório remoto>** //Associar o repositório local ao repositório remoto.
- git push origin main** //Realizar envio do repositório local para o repositório remoto.

- Comandos Principais

- **git --version** //Reportar a versão do sistema.
- git config** //Configurar o sistema.
- git init** //Inicia um repositório e começa a observar as alterações do projeto.
- git status** //Reportar o estado do repositório.
- git add .** //Realiza adições na staging area.
- git commit -m 'mensagem'** //Cria uma versão.
- git log** //Histórico dos commits realizados.
- git shortlog** //Histórico resumido dos commits realizados.
- git show <id do commit>** //Apresenta informação sobre um commit.
- git diff** //Exibe diferenças entre alterações.
- git remote add origin <url do repositório remoto>** //Associar o repositório local ao repositório

remoto.

git remote //Listar a associação criada.

git remote -v //Listar detalhes sobre a associação criada.

git push origin main //Realizar envio do repositório local para o repositório remoto.

git push -u origin main //Realizar envio do repositório local para o repositório remoto. **-u**: para não ter que informar “origin” e “main” nas próximas vezes que o repositório local tiver que ser enviado para o repositório remoto.

origin: repositório remoto.

main: para enviar do “branch” “main” local, para o “branch” “main” remoto. Se o “branch” “main” remoto não existir, ele é criado. Equivale a:

git push -u origin main:main

Que significa:

git push -u origin <branch local>:<branch remoto>

git push //Enviar do repositório local para o repositório remoto.

git fetch origin //Buscar o repositório remoto para o repositório local, sem realizar o “merge” dos “branches”.

git pull //Realiza o “merge” dos “branches”.

git pull origin //Buscar o repositório remoto para o repositório local, realizando o “merge” dos “branches”.