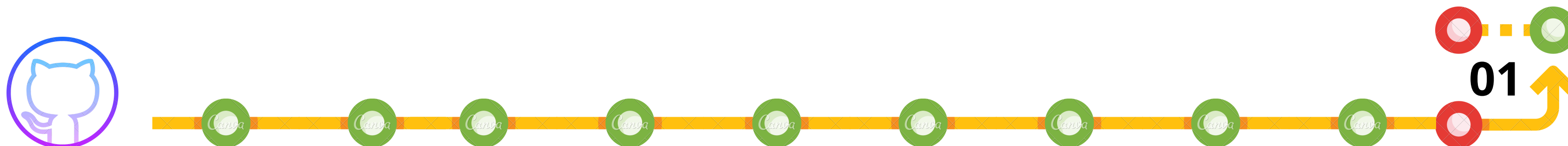


Git / Github

Criando repositório com Github e enviando arquivos com terminal **Git bash**

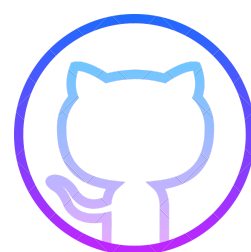


Git / Github

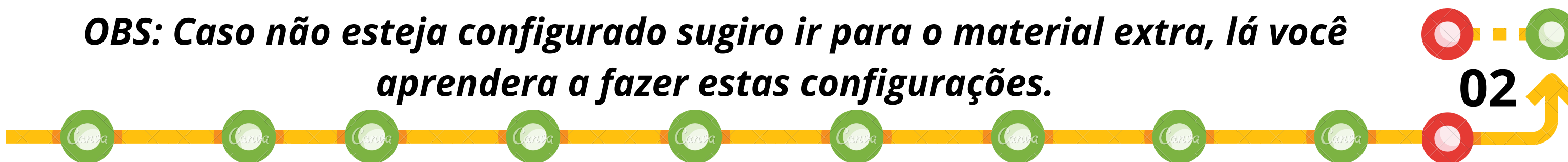
Primeiro passo vamos conferir em nosso terminal Git bash se:

- **O nome de usuário** esta configurado **(user.name)**
- **O email de usuário** esta configurado **(user.email)**

Para fazer esta verificação digite o comando: **git config --list** **no terminal**
Git bash. Este comando ira mostrar todas as configurações.



OBS: Caso não esteja configurado sugiro ir para o material extra, lá você aprendera a fazer estas configurações.



```
Tegra@DESKTOP-999JDBG MINGW64 /
$ 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=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
init.defaultbranch=master
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
user.name=HederAlves
user.email=alvesheder92@gmail.com

Tegra@DESKTOP-999JDBG MINGW64 /
$
```

comando executado

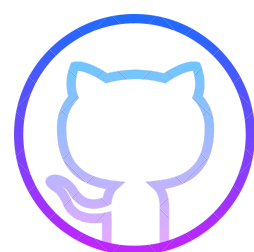
nome de usuário

email de usuário

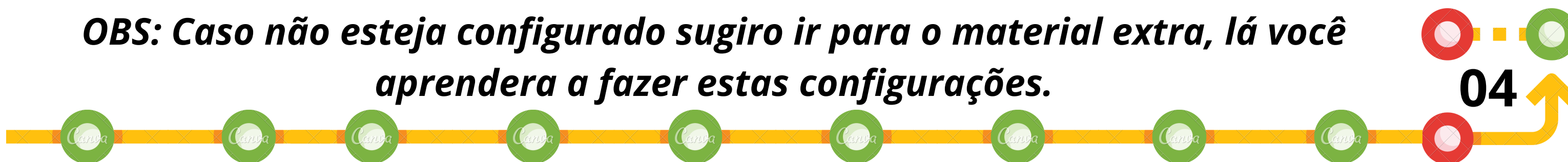
Git / Github

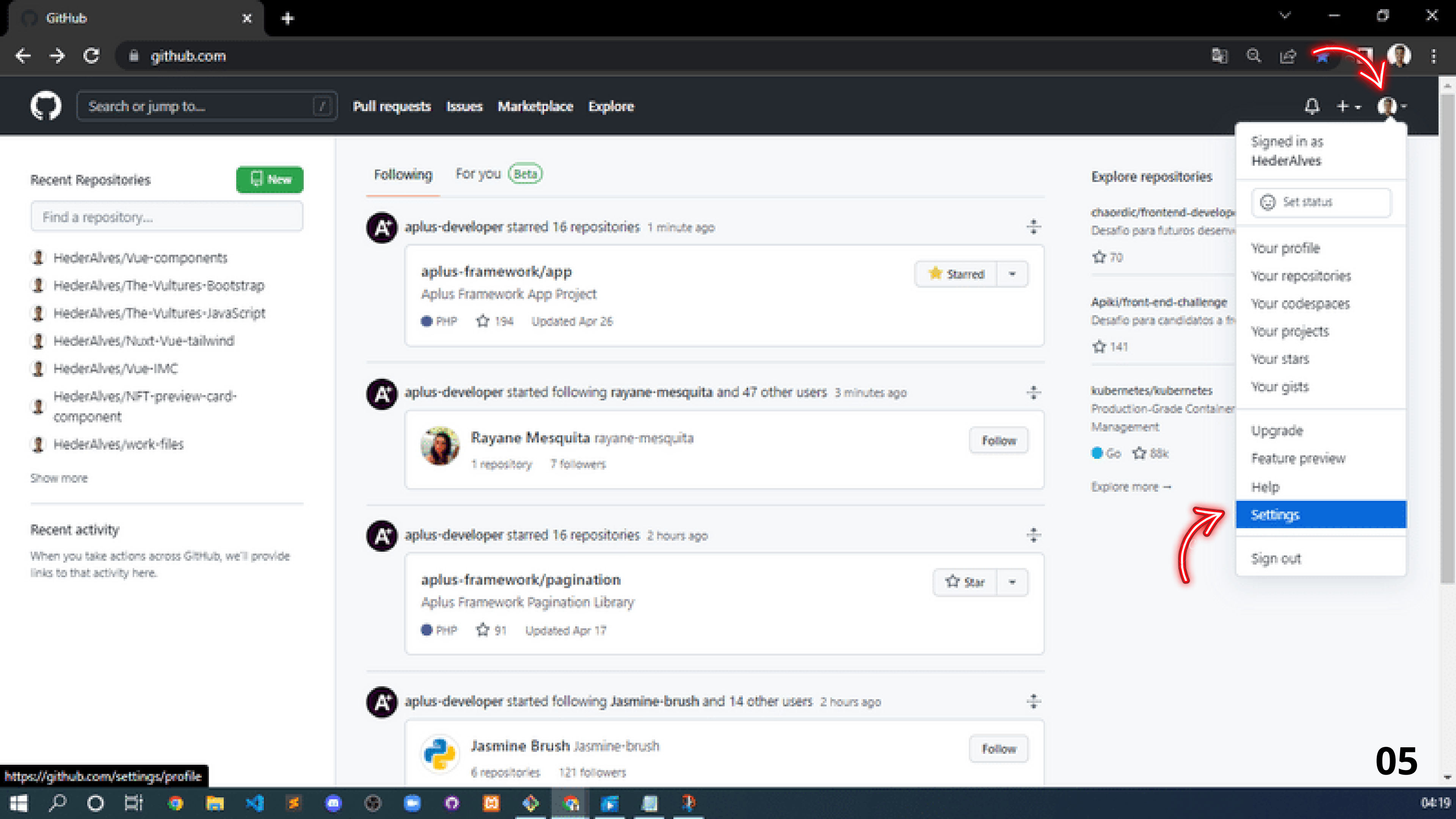
Segundo passo vamos conferir se temos uma chave SSH no site do Github.

Para fazer esta verificação **navegue até a imagem do seu perfil** e acesse a opção **Settings**. O próximo clique será na opção de **SSH and GPG keys**. Verifique se existe alguma chave **SSH configurada**.



OBS: Caso não esteja configurado sugiro ir para o material extra, lá você aprenderá a fazer estas configurações.





SSH and GPG keys

github.com/settings/keys

Search or jump to...

Pull requests

Issues

Marketplace

Explore

Heder Alves

Your personal account

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

SSH and GPG keys

Organizations

Moderation

Code, planning, and automation

Repositories

Packages

Pages

Saved replies

Go to your personal profile

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Heder

SHA256:yE17UHPctA3vFn8f0x5e3T81c/200C+y6Q42+hQ1vMhc

Added on 2 May 2022

Never used — Read/write

SSH

Delete

Check out our guide to generating SSH keys or troubleshoot common SSH problems.

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

Vigilant mode

☐ Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key.

Note that this will include your existing unsigned commits.

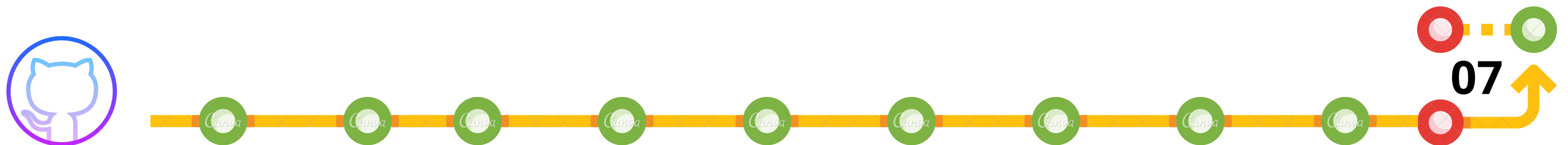
06

05:21

Git / Github

Verifique também no **terminal Git bash** usando o comando:

ls -al ~/.ssh



```
MINGW64:/

Tegra@DESKTOP-999JDBG MINGW64 / comando executado
$ ls -al ~/.ssh
total 31
drwxr-xr-x 1 Tegra 197121  0 Apr 27 10:57 ./
drwxr-xr-x 1 Tegra 197121  0 May  2 03:36 ../
-rw-r--r-- 1 Tegra 197121 464 Apr 29 03:37 id_ed25519
-rw-r--r-- 1 Tegra 197121 104 Apr 29 03:37 id_ed25519.pub
-rw-r--r-- 1 Tegra 197121 670 Apr 27 10:57 known_hosts
-rw-r--r-- 1 Tegra 197121 187 Mar 30 23:41 known_hosts.old

Tegra@DESKTOP-999JDBG MINGW64 /
$ |
```

← chave SSH criada

← chave SSH
publica

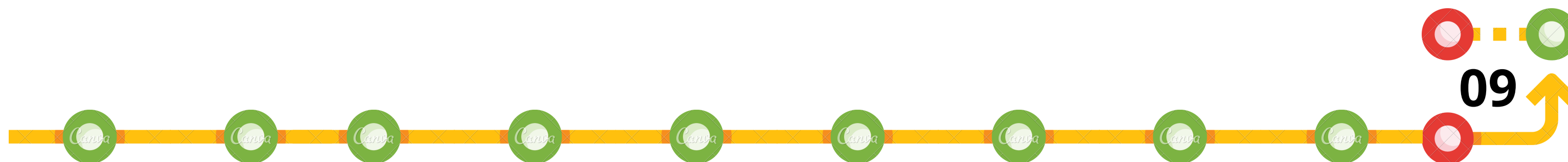
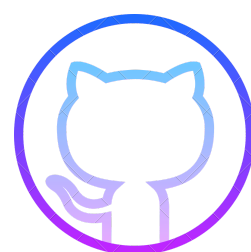


Git / Github

Vamos então iniciar nosso repositório e enviar nossos arquivos

Crie uma pasta ou abra uma pasta existente utilizando o **terminal Git bash**.

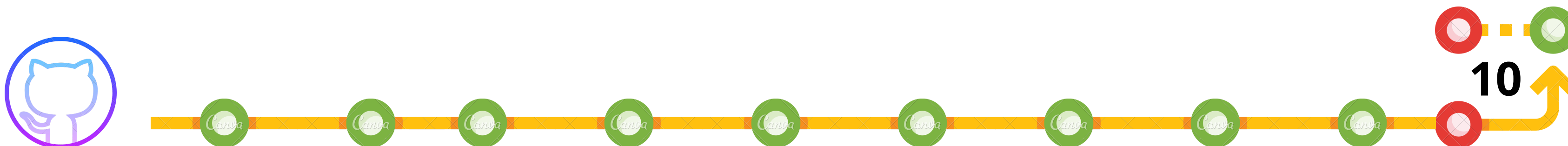
Vamos iniciar com o **comando:** **git init** Este comando faz com que o **git** **assista a pasta** e registre alterações que acontece dentro dela.



Git / Github

Nosso segundo **comando utilizado** se chama: **git status** ele será nosso amigo informante sempre que chamarmos por ele, teremos respostas do que aconteceu na pasta, e com quais arquivos.

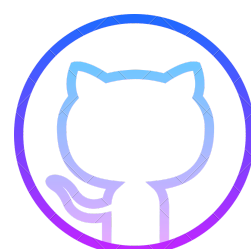
Ele ajuda muito a sabermos qual será o **próximo comando** que precisamos **executar**.



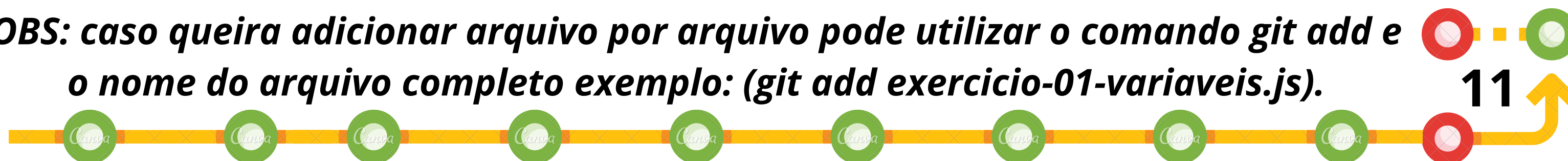
Git / Github

Vamos agora adicionar nosso ou nossos arquivos utilizando o
comando: `git add`.

Este comando irá adicionar todos os arquivos que estão dentro da pasta.



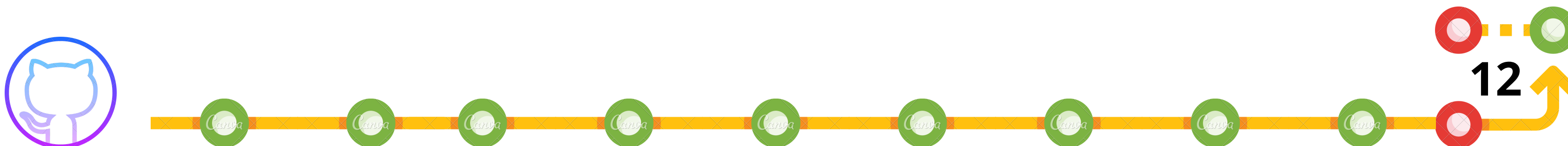
OBS: caso queira adicionar arquivo por arquivo pode utilizar o comando `git add` e o nome do arquivo completo exemplo: (`git add exercicio-01-variaveis.js`).



Git / Github

Vamos agora registrar o **commit** relacionado ao(s) arquivo(s) adicionado(s)

Para isso vamos utilizar o **comando:**
git commit -m "seu commit aqui"



```
MINGW64:/c/Users/Tegra.DESKTOP-999JDBG/Desktop/exercicios-traco
Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    exercicios-01-variaveis.js

nothing added to commit but untracked files present (use "git add" to track)

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git add .

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   exercicios-01-variaveis.js

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git commit -m "seu commit aqui"
[master (root-commit) 3efeb92] seu commit aqui
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 exercicios-01-variaveis.js

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git status
On branch master
nothing to commit, working tree clean

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$
```

←
Comando executado

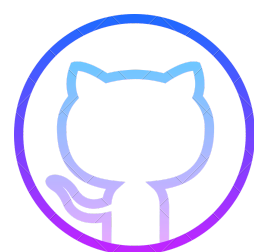
←
Arquivo ainda não
adicionado

←
Arquivo já adicionado,
aguardando commit

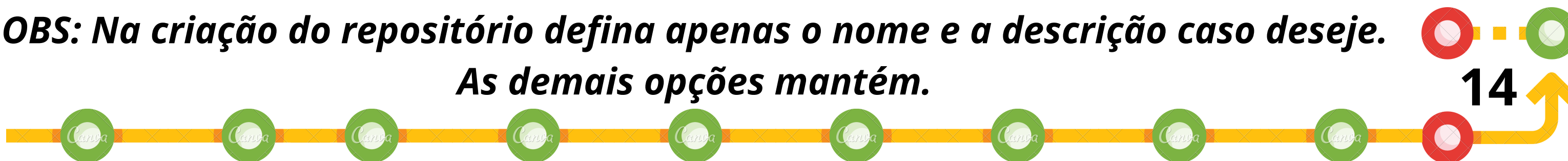
←
Todos arquivos
adicionados e com commit

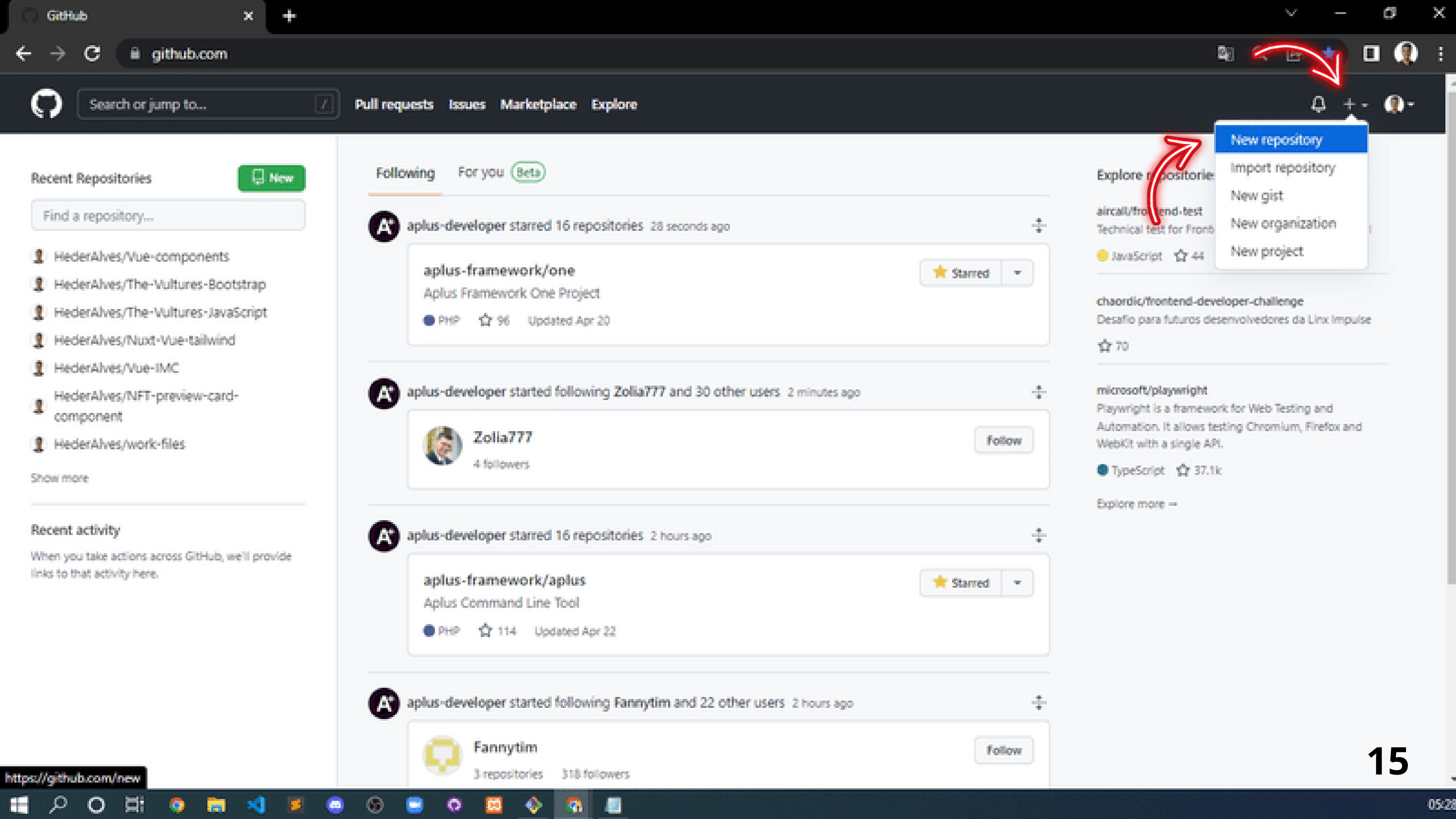
Antes de enviar nossos arquivos com o **comando** `git push` vamos criar nosso **repositório** no site do **GitHub**.

Acessamos nossa conta e navegamos até o **sinal de +** e selecionamos a opção **New repository**.

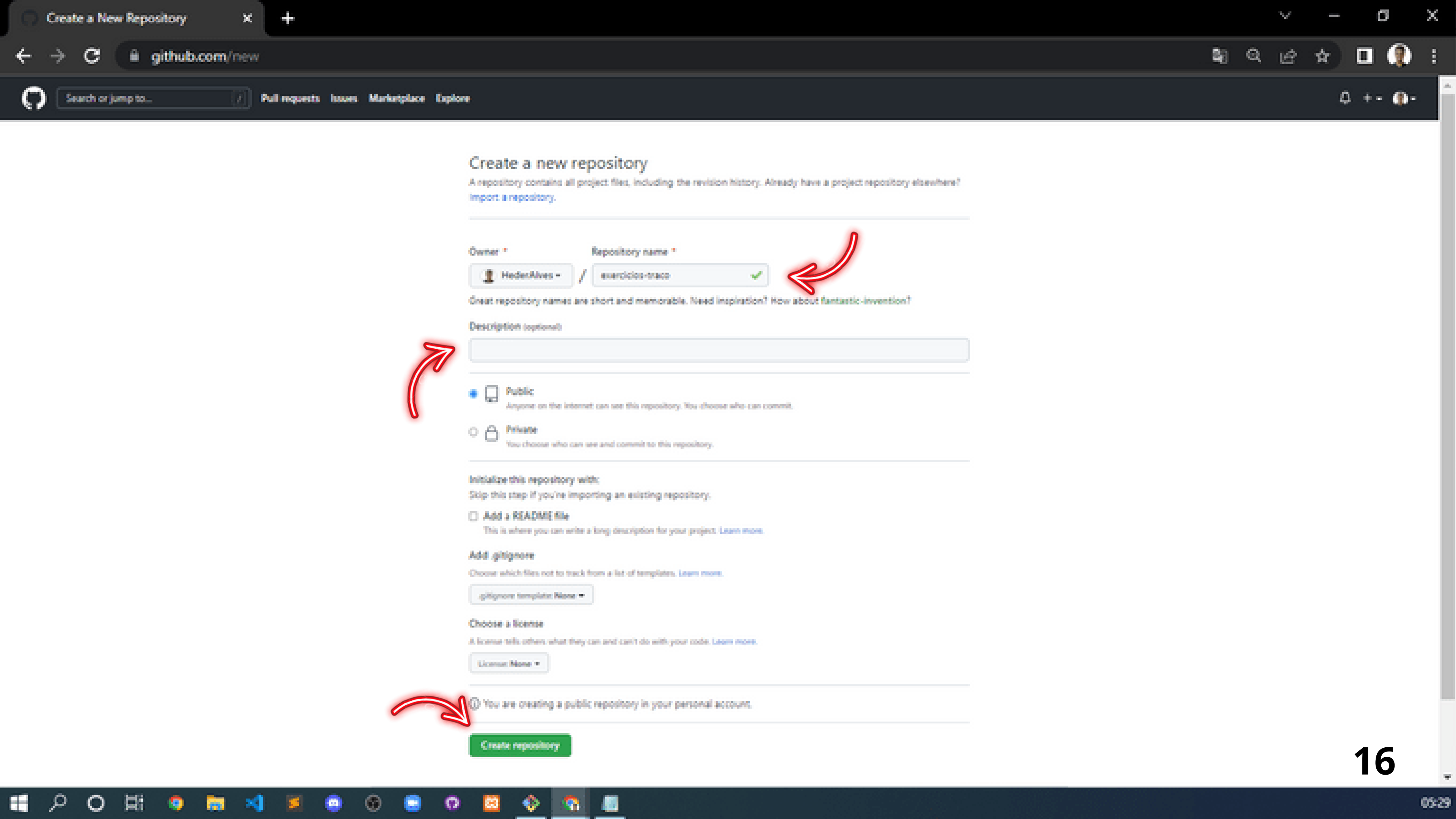


OBS: Na criação do repositório defina apenas o nome e a descrição caso deseje.
As demais opções mantém.





- New repository
- Import repository
- New gist
- New organization
- New project

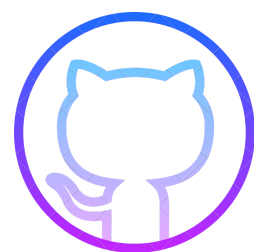


Git / Github

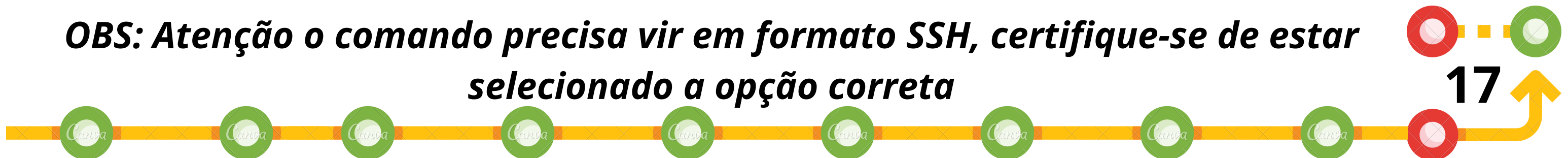
Após criado o **repositório**, você será direcionado para uma página aonde irão conter alguns **comandos**, deles **o único que vamos utilizar é o:**

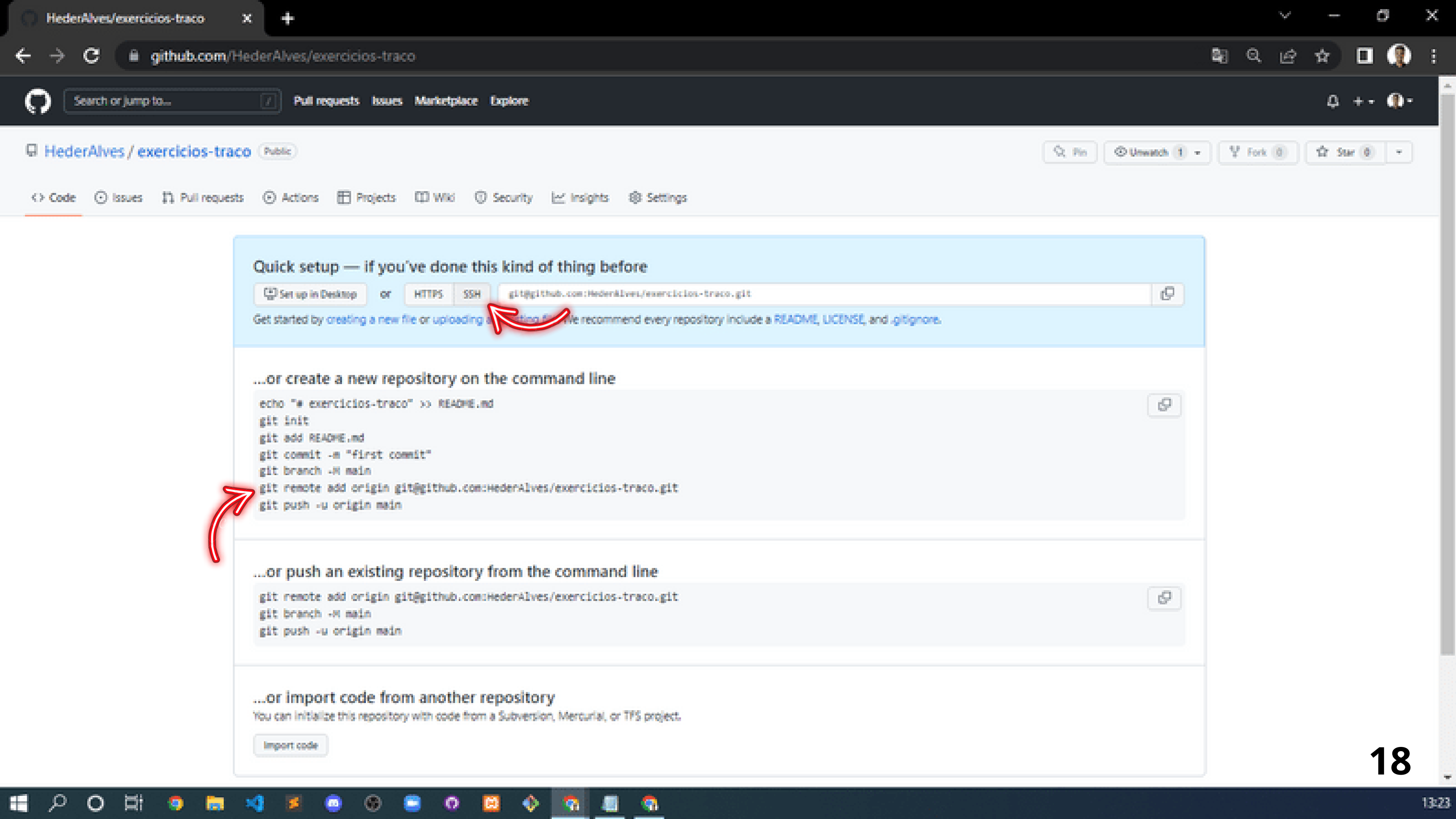
git remote add origin git@github.com:seu/link.gerado

Copie o comando com os **seus** dados e cole no terminal **Git bash**



OBS: Atenção o comando precisa vir em formato SSH, certifique-se de estar selecionado a opção correta







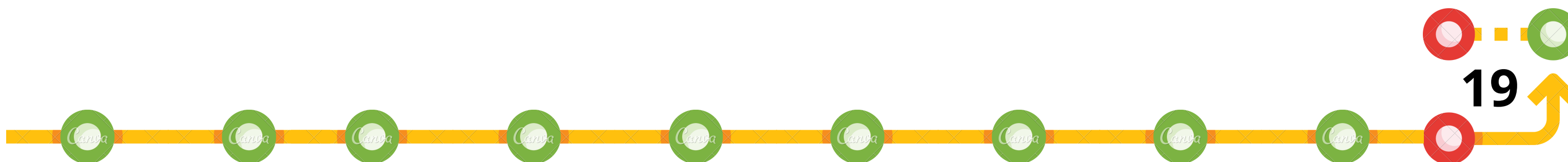
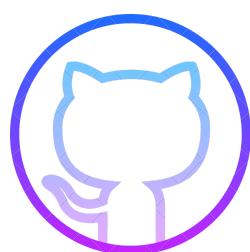
Git / Github

Agora sim vamos enviar nosso(s) arquivo(s) para o repositório no site utilizando o **comando:** `git push` no terminal **Git bash**

Ele irá retornar uma mensagem. (**fatal: the current branch master ...**) mais a baixo em um texto separado encontramos um **comando:**

`git push --set-upstream origin master`

Então **copiamos este comando e usamos** em nosso terminal **Git bash**

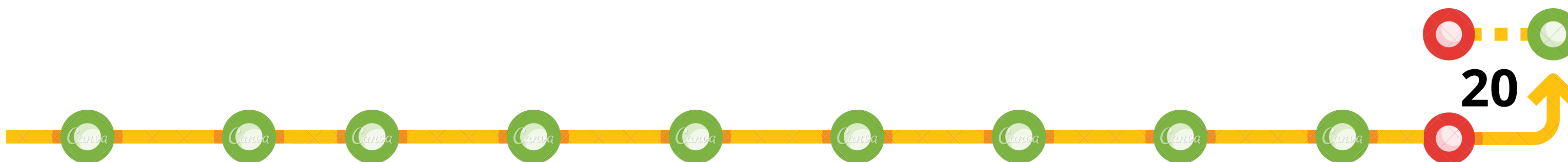
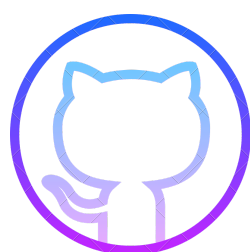




Git / Github

Ele então irá pedir uma **senha** para nós, esta **senha** é a mesma cadastrada em **nossa chave SSH**

Pronto já **enviamos nossos arquivos** e deixamos **tudo configurado** para trabalharmos com nosso **repositório**



```
MINGW64:/c/Users/Tegra.DESKTOP-999JDBG/Desktop/exercicios-traco

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git remote add origin git@github.com:HederAlves/exercicios-traco.git
Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master
Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git push --set-upstream origin master
Enter passphrase for key '/c/Users/Tegra.DESKTOP-999JDBG/.ssh/id_ed25519':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 232 bytes | 232.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:HederAlves/exercicios-traco.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$
```

←
Comando executado

←
Comando do terminal
para copiar

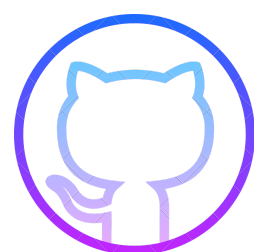
←
Senha da chave **SSH**
criada



Git / Github

Vamos **movimentar** um pouco nossa **pasta e repositório** para vermos o **mecanismo funcionando** e ficarmos ainda mais preparados

Adicione novo(s) arquivo(s) a sua pasta, e no terminal **Git bash** vamos chamar nosso amigo informante: **git status**



OBS: Notamos que novo(s) arquivo(s) estão na pasta. O terminal nos mostra o nome do arquivo em letras vermelhas informando que ainda não foi adicionado.



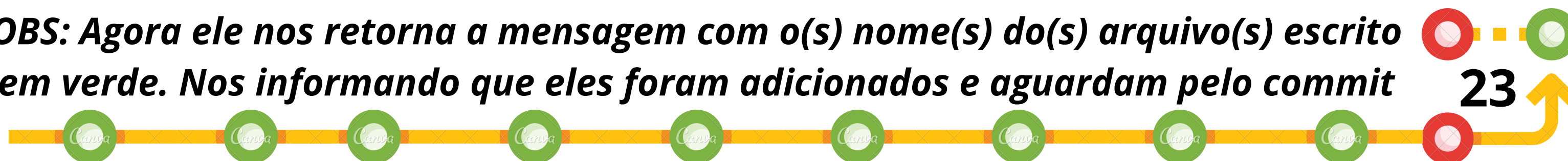
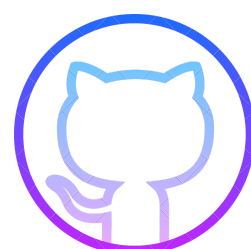
Git / Github

Para adicionar utilizamos então o **comando:** `git add`.

Como já é de costume chamamos nosso amigo informante:

`git status`

OBS: Agora ele nos retorna a mensagem com o(s) nome(s) do(s) arquivo(s) escrito em verde. Nos informando que eles foram adicionados e aguardam pelo commit



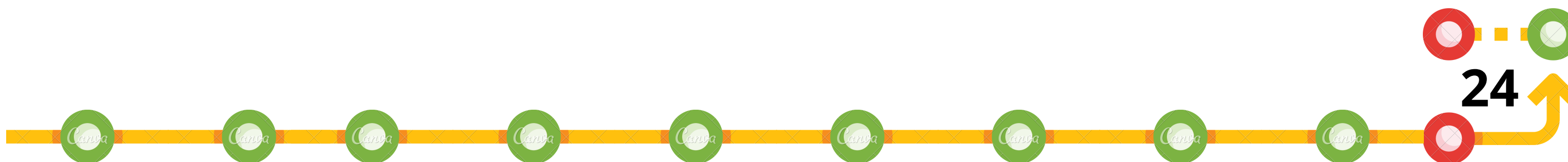
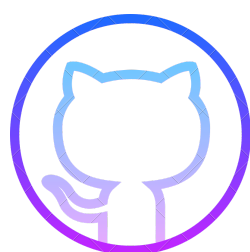


Git / Github

Próximo passo é dar o commit.

Utilizando o **comando:** `git commit -m "seu commit aqui"`

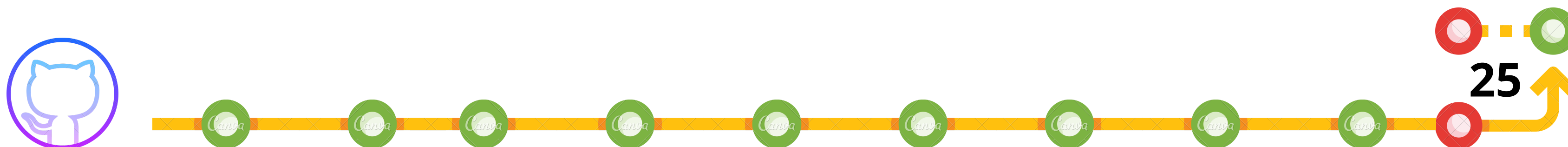
Assim deixamos então **tudo preparado** para nosso(s) arquivo(s) **serem enviados** para o nosso repositório no **site do Github**



Git / Github

E por fim **enviamos** nosso(s) arquivo(s) utilizando o **comando:**
git push

Daqui para frente não iremos receber a mensagem com o comando para copiar e colar, pois já foi resolvido anteriormente. Então irá direto para opção da senha, a qual é a mesma da **chave SSH**



```
MINGW64:/c/Users/Tegra.DESKTOP-999JDBG/Desktop/exercicios-traco
Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html
        style.css

nothing added to commit but untracked files present (use "git add" to track)

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git add .

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   index.html
        new file:   style.css

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git commit -m "seu commit aqui"
[master fc74e2b] seu commit aqui
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
 create mode 100644 style.css

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$ git push
Enter passphrase for key '/c/Users/Tegra.DESKTOP-999JDBG/.ssh/id_ed25519':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 279 bytes | 279.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:HederAlves/exercicios-traco.git
 1d9a417..fc74e2b  master -> master

Tegra@DESKTOP-999JDBG MINGW64 ~/Desktop/exercicios-traco (master)
$
```

←
Comando executado

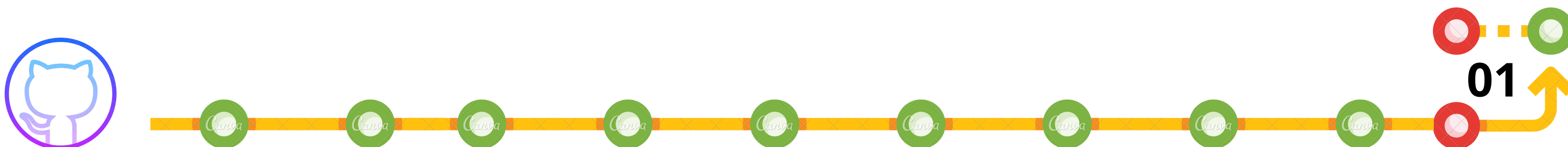
←
Arquivo ainda não
adicionado

←
Arquivo já adicionado,
aguardando commit

←
Senha da chave **SSH**
criada

A seguir novo conteúdo

Configurando **nome** de usuário e **email** de usuário



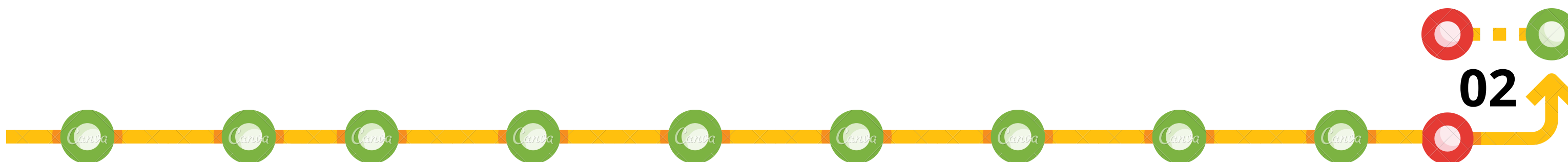
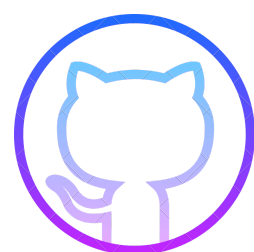


Git / Github

Primeiro passo é **conferir** como está nossas **configurações** para isso utilizamos no terminal **Git bash** o **comando**:

git config --list

Notem que **não temos nem nome de usuário e nem email de usuário configurado**. Vamos então **configurar** estas opções, uma por vez



```
MINGW64:/

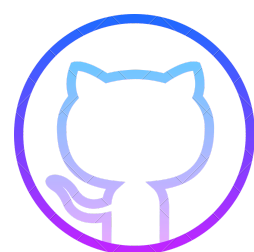
Tegra@DESKTOP-999JDBG MINGW64 /
$ git config --list ← comando executado
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
init.defaultbranch=master
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
user.name= ← nome de usuário
user.email= ← email de usuário

Tegra@DESKTOP-999JDBG MINGW64 /
$
```

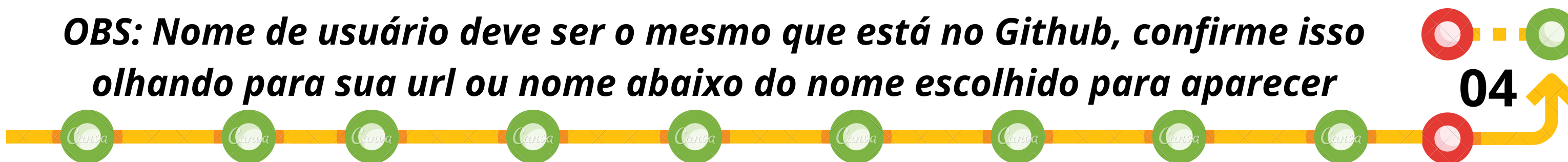
Git / Github

Primeiro vamos configurar nosso **nome de usuário** utilizando no terminal do **Git bash** o **comando:**

git config --global user.name "seu nome de usuário aqui"



OBS: Nome de usuário deve ser o mesmo que está no Github, confirme isso olhando para sua url ou nome abaixo do nome escolhido para aparecer



Heder Alves

[Edit profile](#)

© Contenda-P.R. Brasil



Customize your pins

 [The-Multiverse-Bootstrap](#) Public

Página da banda The Vultures, desenvolvida com o framework Bootstrap

Sistema-estoque-produtos Public

Sistema para controlar a entrada e saída de produtos.

Page 10 of 10

[Sistema-cobra](#)
Public

Sistema para controle de fluxo de caixa

Journal Pre-proof

React-Node-Typescript Public

Lista de tarefas desenvolvida com react

API-banco-de-dados Public

API e banco de dados, CRUD de produtos

Calcula-IMC-JavaScript Public

Programa calcula seu índice de massa corporal.

1,146 contributions in the last year

Contributions received:



Activity overview

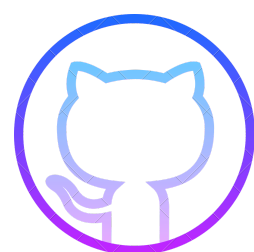
Code review

Contributed to [HederAhves/Vue-components](#),
[HederAhves/The-Vultures-Bootstrap](#),
[HederAhves/The-Vultures-JavaScri...](#)

Git / Github

Agora vamos configurar o **email de usuário** utilizando no terminal **Git bash** o comando:

git config --global user.email "seu email aqui"



OBS: Email deve ser o mesmo utilizado para criar a conta no Github



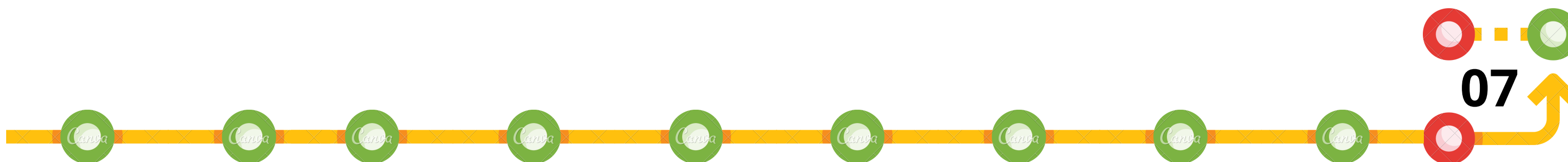
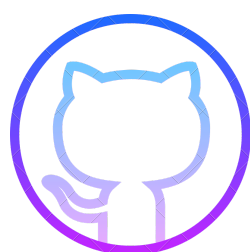


Git / Github

Feito isso já temos nossas **configurações prontas** vamos chamar novamente o **comando:**

git config --list

Para observarmos as novas configurações



```
MINGW64:/

Tegra@DESKTOP-999JDBG MINGW64 /
$ git config --global user.name "HederAlves"

Tegra@DESKTOP-999JDBG MINGW64 /
$ git config --global user.email "alvesheder92@gmail.com"

Tegra@DESKTOP-999JDBG MINGW64 /
$ 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=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
init.defaultbranch=master
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
user.name=HederAlves
user.email=alvesheder92@gmail.com

Tegra@DESKTOP-999JDBG MINGW64 /
$
```

←
Comando executado

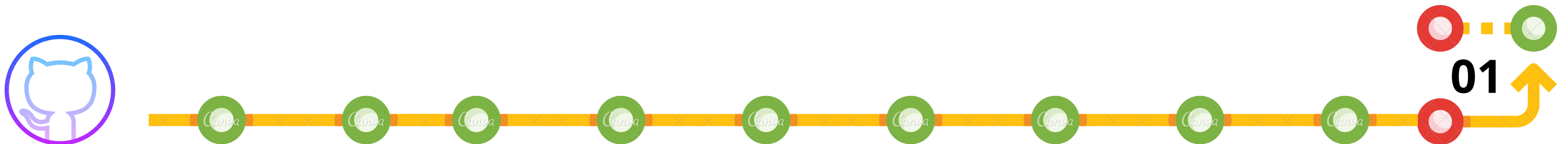
←
Nome de usuário

←
Email de usuário

A seguir novo conteúdo

Git / Github

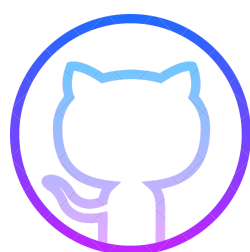
Configurando chave **SSH**








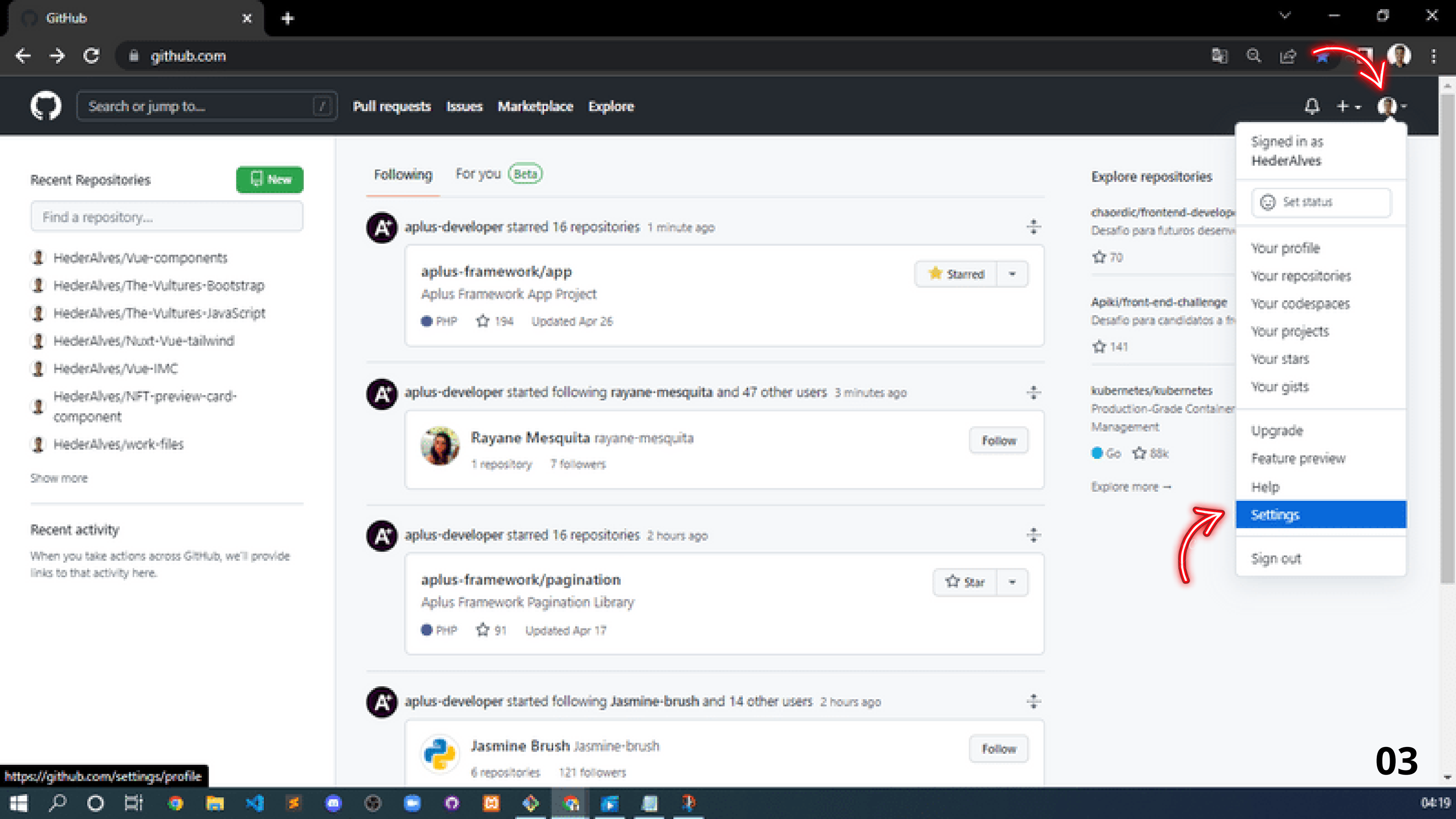
Git / Github

Primeiros vamos no site deixar tudo preparado para cadastrar a
nova chave SSH

Os **comando utilizados** estão na **documentação**, mas você irá ter **acesso** a eles
nos **próximos slides**



OBS: Para acessar a documentação, basta clicar na palavra generating **SSH** keys,  
você será redirecionado para a documentação  
 **02**



SSH and GPG keys

github.com/settings/keys

Search or jump to...

Pull requestsIssuesMarketplaceExplore

Go to your personal profile

Heder Alves

Your personal account

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

SSH and GPG keys

Organizations

Moderation

Code, planning, and automation

Repositories

Packages

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

SSH keys

There are no SSH keys associated with your account.

Check out our guide to [generating SSH keys](#) or [troubleshoot common SSH problems](#).

New SSH key

GPG keys

There are no GPG keys associated with your account.

Learn how to [generate a GPG key](#) and add it to your account.

New GPG key

Vigilant mode

☐ Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key. Note that this will include your existing unsigned commits.

[Learn about vigilant mode.](#)

04

03:18



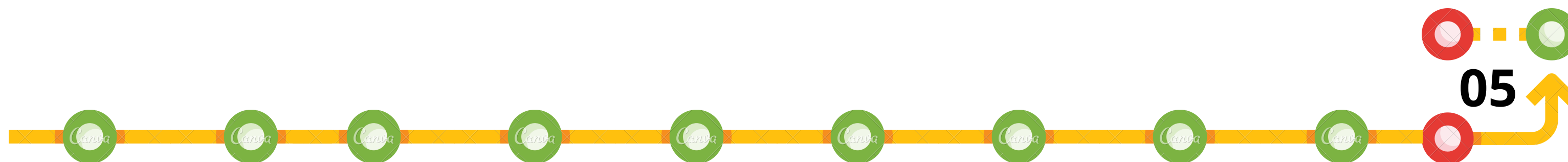
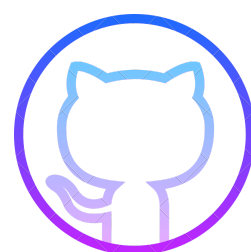
Git / Github

No terminal **Git bash** vamos utilizar **dois comandos**

Insira o primeiro **comando**:

ssh-keygen -t ed25519 -C "seu email aqui"

Apos gerar a nossa **chave** ele vai nos retornar **algumas informações**, basta dar **enter para avançar**



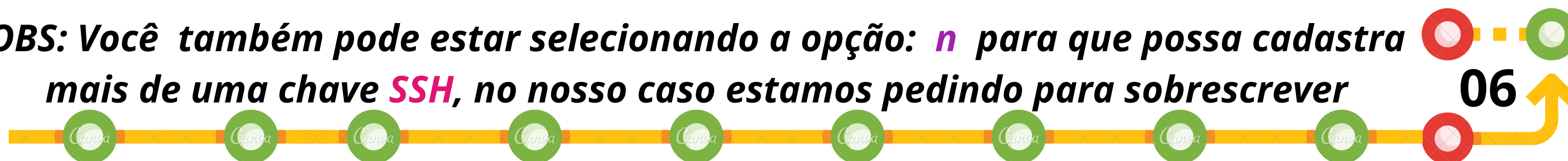
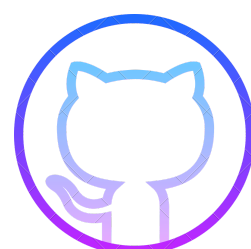
Git / Github

Novamente ele retorna uma mensagem, desta vez **ele pergunta:**

Overwrite (y/n)?

neste caso vamos **utilizar o:** **y**

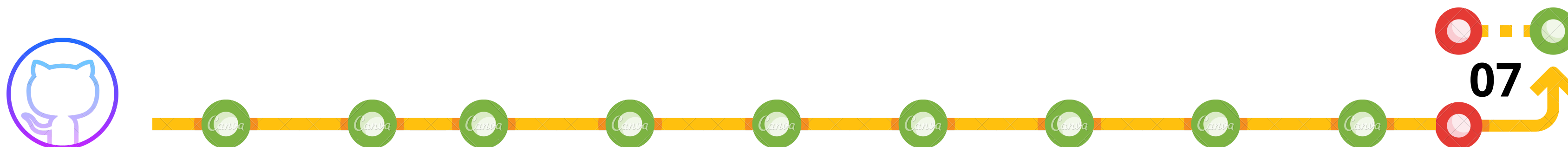
OBS: Você também pode estar selecionando a opção: **n para que possa cadastrar mais de uma chave **SSH**, no nosso caso estamos pedindo para sobrescrever**



Git / Github

Em seguida ele nos pede para **cadastrar uma senha e confirmar** a mesma.

Feito isso ele irá nos retornar **a chave** e uma espécie de figura em código

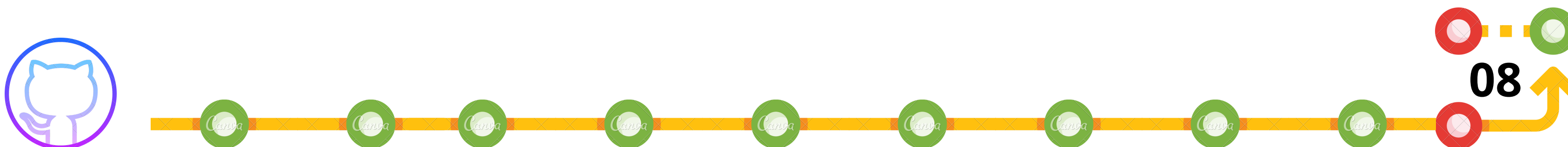


Git / Github

Nosso ultimo passo no terminal **Git bash** é adicionar a chave e para isso utilizamos o nosso segundo **comando**:

```
clip < ~/.ssh/id_ed25519.pub
```

Volte até o site sem copiar nada, uma mágica acontece após este comando, guardando a chave na memória como um **Ctrl + c**



```
MINGW64:/
Tegra@DESKTOP-999JDBG MINGW64 /
$ ssh-keygen -t ed25519 -C "alvesheder92@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Tegra.DESKTOP-999JDBG/.ssh/id_ed25519):
/c/Users/Tegra.DESKTOP-999JDBG/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Tegra.DESKTOP-999JDBG/.ssh/id_ed25519
Your public key has been saved in /c/Users/Tegra.DESKTOP-999JDBG/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:jYzHnbLEchi+zn2L9PYipHNIhqq7jciE9L6Taq3dgLo alvesheder92@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|
|      .
|    . B + .
|  .  . = S +
|o o . o*.o
|.o.+ .o.+o
|=+++oo+ooo+
|E0++o.ooo+o+.
+-----[SHA256]-----+

Tegra@DESKTOP-999JDBG MINGW64 /
$ clip < ~/.ssh/id_ed25519.pub

Tegra@DESKTOP-999JDBG MINGW64 /
$ |
```

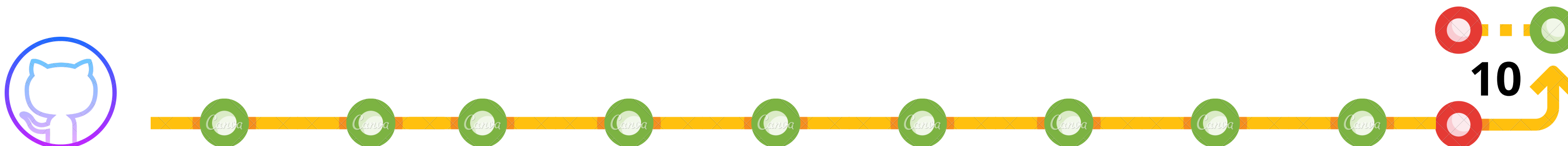
←
Comando executado

←
Digite sua senha

Git / Github

Então agora novamente **no site**, no campo abaixo do titulo da chave **cole a sua chave** e click no botão **add SSh key**

Pronto sua chave já está cadastrada e pronta para ser utilizada



Add new SSH keys

github.com/settings/ssh/new

Search or jump to...Pull requestsIssuesMarketplaceExplore

Heder Alves

Your personal account

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

SSH and GPG keys

Organizations

Moderation

Code, planning, and automation

Repositories

Packages

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

SSH keys / Add new

Title

Heder

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

Go to your personal profile

11

03:21

PRODUZIDO POR:

Academia
tegra ▶

