

# Ferramentas Controle Versão

Quando trabalhamos em equipes de desenvolvimento para gerenciamento dos projetos utilizamos o controle de versão que serve para controlar as alterações realizadas e solucionar problemas de conflito em códigos do projeto. São gerados históricos por quem efetuou determinada alteração. Algumas características do controle de versão:

- Comparar um código com uma versão anterior
- Recuperar um código de versão anterior
- Histórico de atualizações
- Facilita o trabalho em equipe

Um controle de versão possui duas partes: o repositório e a área de trabalho. O repositório armazena todo o histórico de evolução do projeto, onde são registradas todas as modificações para determinada versão. Os desenvolvedores trabalham com cópias do repositório no computador.

# Git

Foi desenvolvido por Linus Torvalds em 2005 e tornou-se a ferramenta mais utilizada para controle de versão. No Git todo desenvolvedor tem uma cópia local do projeto. O repositório fica armazenado em um servidor Git como o Github. Quando alguém da equipe vai trabalhar no projeto, uma cópia local do repositório é copiada para sua máquina.

# Github

O Github é um serviço gratuito que oferece diversas funcionalidades adicionais aplicadas ao git, diversos projetos open source estão hospedados no github, onde os desenvolvedores podem contribuir com correções ou códigos adicionais.

# Instalando git

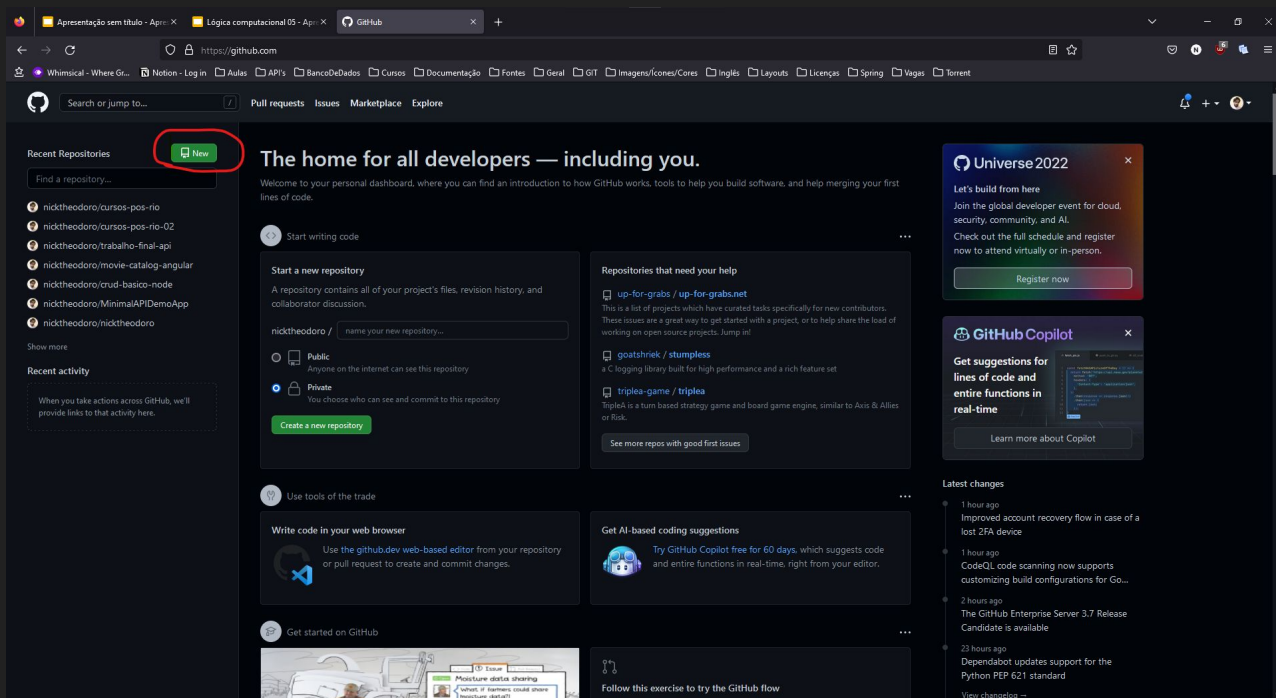
Link para instalação no windows: [Git - Downloading Package](#)

# Criação da Conta no GitHub

1. Acesse: <https://github.com/>
2. Crie sua conta
3. Cadastre seu login de acesso

# Criação do repositório

Para criação de um novo repositório clique no botão conforme imagem abaixo.



## Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner \*  nicktheodoro / Repository name \* projeto-testes 

Great repository names are short and memorable. Need inspiration? How about [ideal-octo-dollop](#)?

Description (optional)

Projeto de testes com o git

☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None

This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

**Public** - deixa o projeto disponível para qualquer usuário realizar alterações.

**README file** - permite uma descrição inicial com detalhes sobre o projeto.

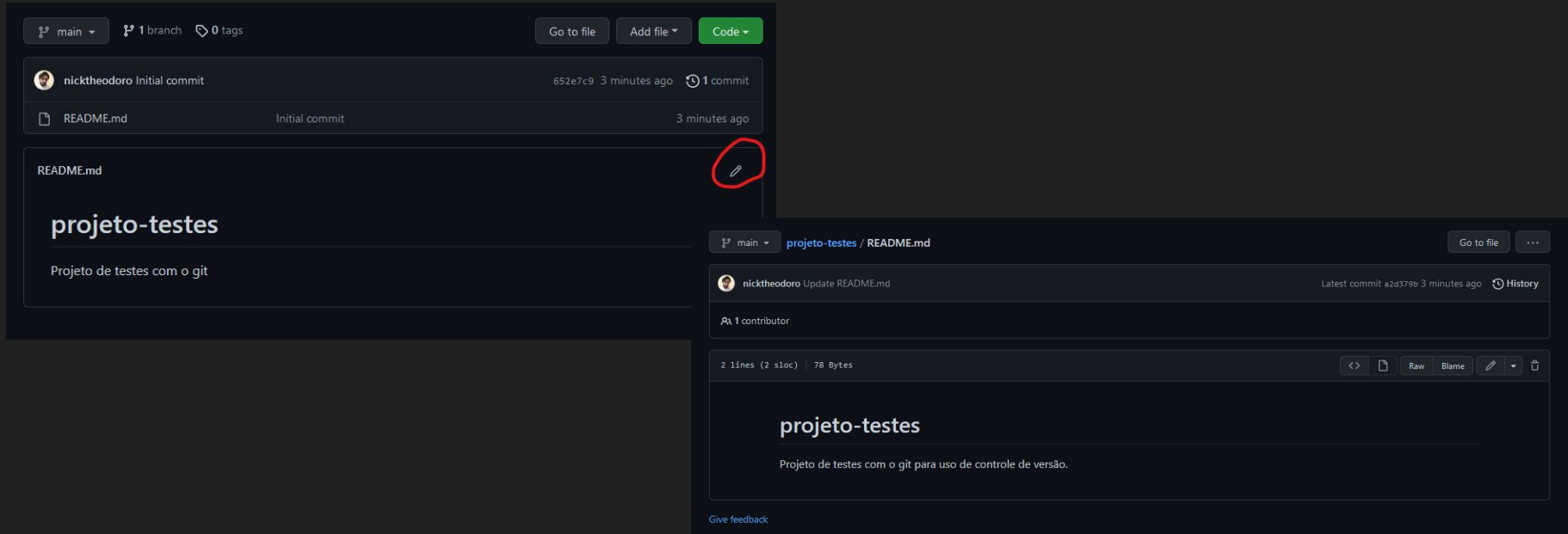
**.gitignore** - Arquivos que não são importantes são ignorados no envio do projeto.

**License** - Podemos escolher um tipo de licença.



# Alterando o README file

Vamos clicar no botão editar e adicionar o texto e gravar as alterações no botão Commit.



The image shows a GitHub repository interface for 'projeto-testes'. The left sidebar displays the repository overview, including the 'main' branch, 1 branch, and 0 tags. The 'README.md' file is highlighted, showing its initial commit by 'nicktheodoro' 3 minutes ago. The main content area shows the README file content, which includes the title 'projeto-testes' and the description 'Projeto de testes com o git'. A red circle highlights the 'Edit' button (pencil icon) next to the file name. The right sidebar shows the file's commit history, including the latest commit 'Update README.md' by 'nicktheodoro' 3 minutes ago. The file content is displayed in a code editor view, showing the title 'projeto-testes' and the description 'Projeto de testes com o git para uso de controle de versão.'.

main 1 branch 0 tags Go to file Add file Code

nicktheodoro Initial commit 652e7c9 3 minutes ago 1 commit

README.md Initial commit 3 minutes ago

README.md

**projeto-testes**

Projeto de testes com o git

main projeto-testes / README.md Go to file ...

nicktheodoro Update README.md Latest commit a2d379b 3 minutes ago History

1 contributor

2 lines (2 sloc) 78 Bytes <> Raw Blame Edit

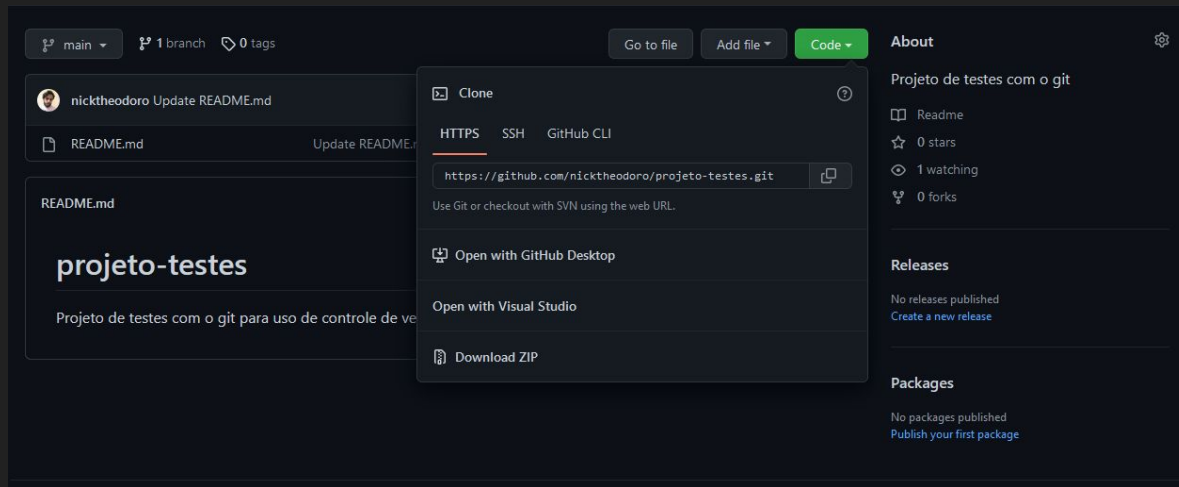
**projeto-testes**

Projeto de testes com o git para uso de controle de versão.

Give feedback

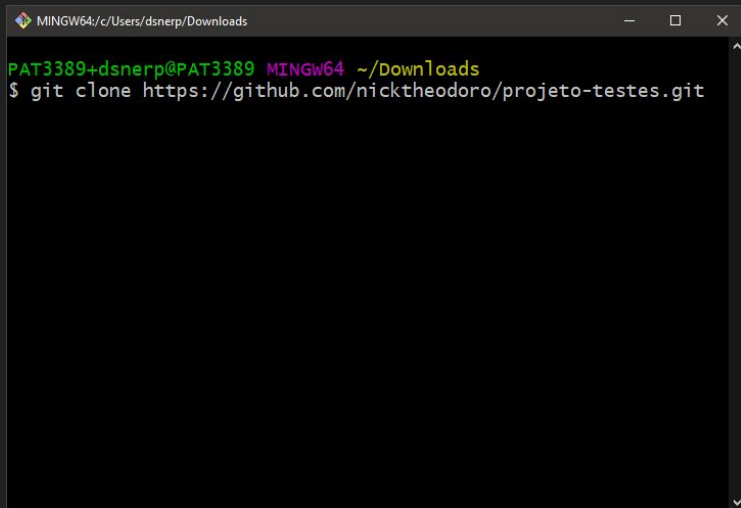
# Clonar Repositório

Na página inicial do repositório temos o botão Code onde é possível fazer o download do arquivo zip ou copiar a url do repositório. Clique no botão para copiar a url.

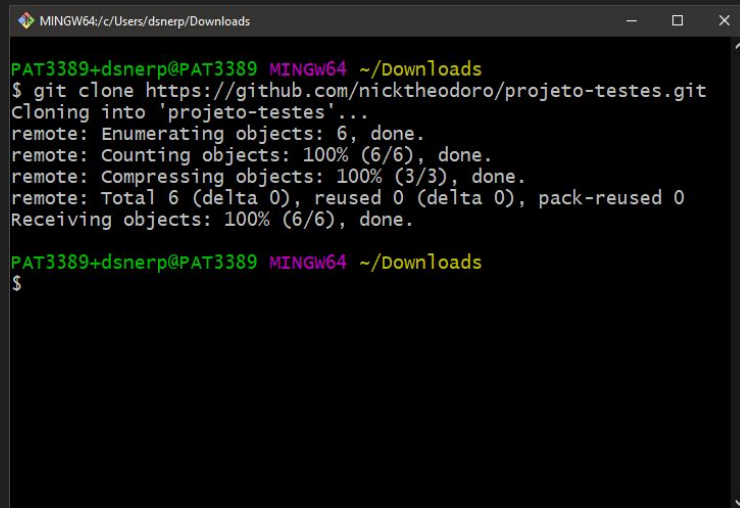


# Clonando Repositório

Abrir o gitbash, digitar `git clone` e colar a url do nosso projeto. Nossa máquina terá uma cópia do repositório remoto. Todos os commits que forem feitos não afetaram o repositório remoto, mas temos comandos também que enviam as alterações para o repositório remoto quando for preciso.



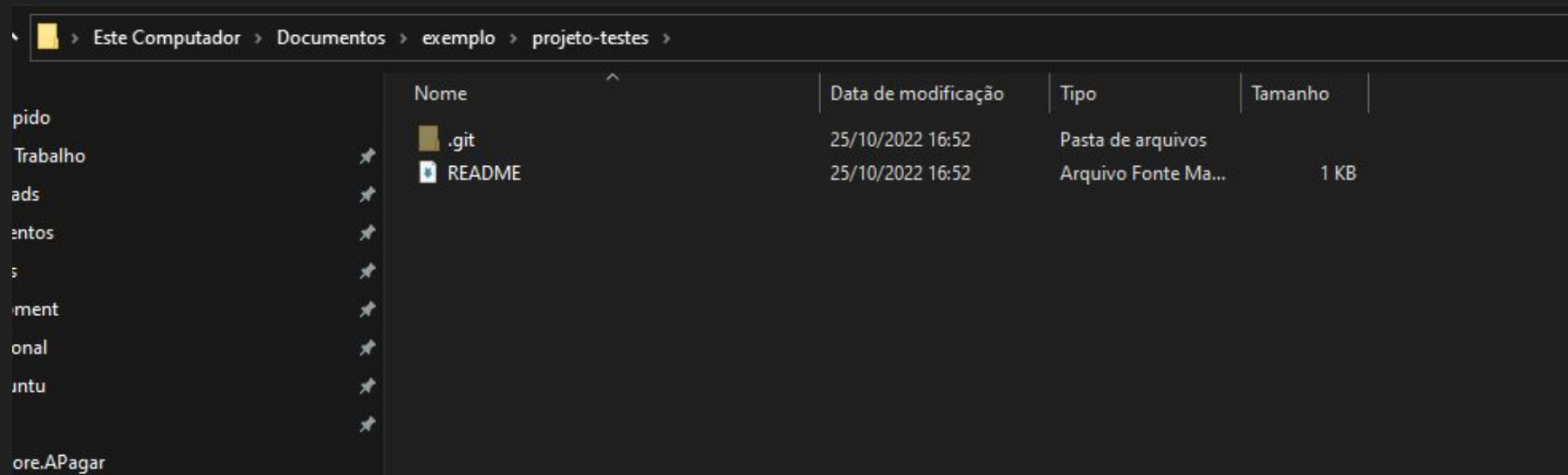
```
MINGW64/c/Users/dsnerp/Downloads
PAT3389+dsnerp@PAT3389 MINGW64 ~/Downloads
$ git clone https://github.com/nicktheodoro/projeto-testes.git
```



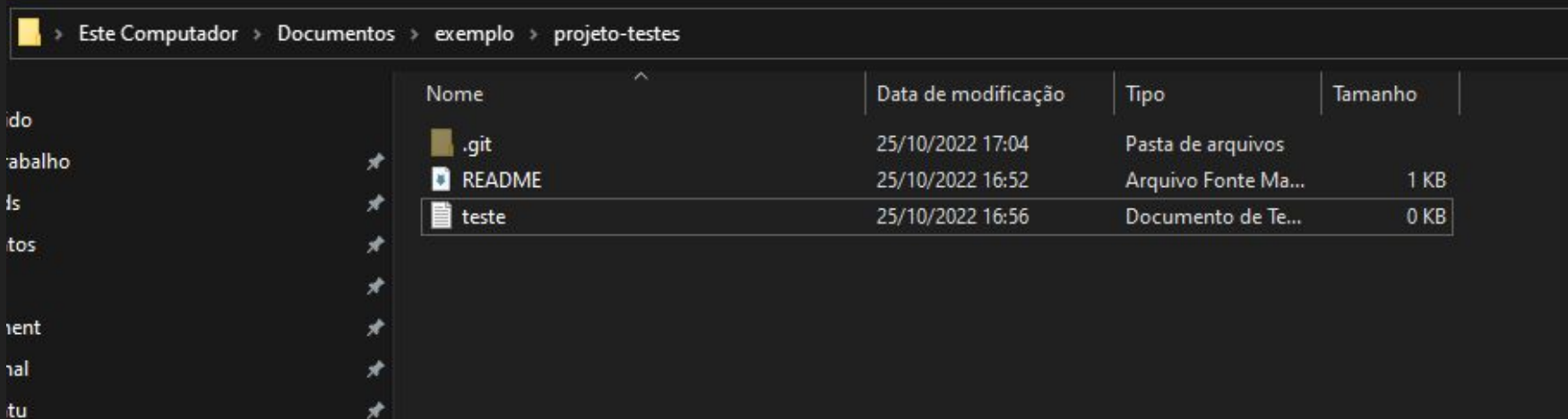
```
MINGW64/c/Users/dsnerp/Downloads
PAT3389+dsnerp@PAT3389 MINGW64 ~/Downloads
$ git clone https://github.com/nicktheodoro/projeto-testes.git
Cloning into 'projeto-testes'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

PAT3389+dsnerp@PAT3389 MINGW64 ~/Downloads
$
```

# A pasta projeto-testes foi criada

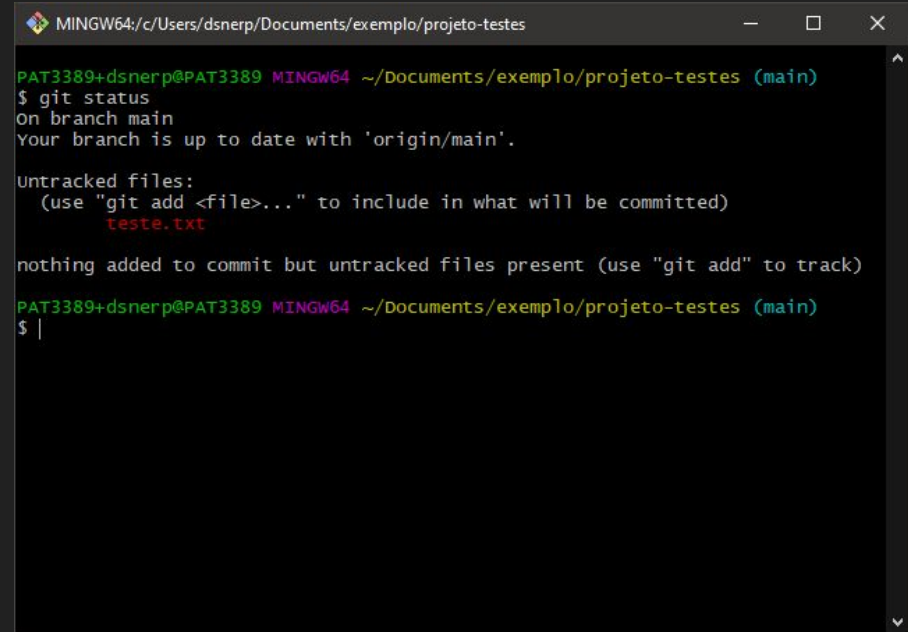


# Vamos criar um arquivo texto no nosso repositório local.



# Git Status

Exibe o status do repositório, se existe algum arquivo novo adicionado, alterado ou removido, caso esteja rastreado com uma branch no servidor o comando verifica o estado da versão. No exemplo o comando retornou que um arquivo não está sendo rastreado e para colocar o arquivo para fazer parte do nosso repositório utilizamos o comando `git add`.

A terminal window with a dark background and light-colored text. The title bar shows 'MINGW64: c:/Users/dsnerp/Documents/exemplo/projeto-testes'. The prompt is 'PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)'. The command '\$ git status' has been entered. The output shows the current branch is 'main' and it is up to date with 'origin/main'. It lists 'teste.txt' as an untracked file. A message at the bottom says 'nothing added to commit but untracked files present (use "git add" to track)'.

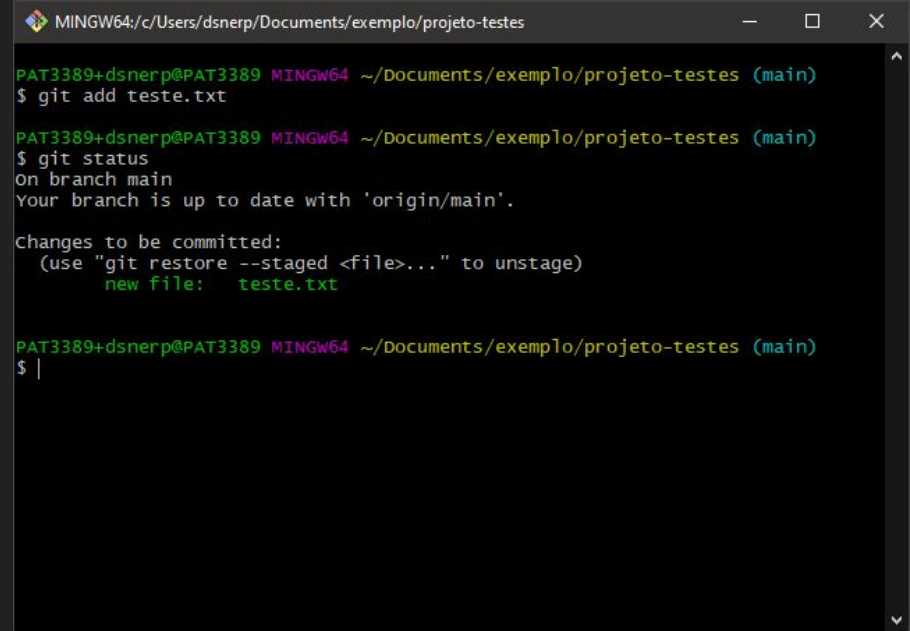
```
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        teste.txt

nothing added to commit but untracked files present (use "git add" to track)
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ |
```

# git add <nome do arquivo>

Adiciona um arquivo alterado e deixa ele preparado para o commit indicando ao git que este arquivo será rastreado. No exemplo ao executar o comando git add e após o git status temos o arquivo registrado no git e pronto para o commit.

A terminal window titled 'MINGW64: c:/Users/dsnerp/Documents/exemplo/projeto-testes' with standard window controls. The terminal shows a user at 'PAT3389+dsnerp@PAT3389' in a 'MINGW64' environment at the directory '~/Documents/exemplo/projeto-testes' on the '(main)' branch. The user enters '\$ git add teste.txt'. The prompt returns, and the user enters '\$ git status'. The output shows 'On branch main', 'Your branch is up to date with \'origin/main\'', and a section 'changes to be committed:' which includes '(use "git restore --staged <file>..." to unstage)' and 'new file: teste.txt'. The prompt returns again, and the user enters '\$ |'.

```
MINGW64: c:/Users/dsnerp/Documents/exemplo/projeto-testes

PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git add teste.txt

PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   teste.txt

PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ |
```

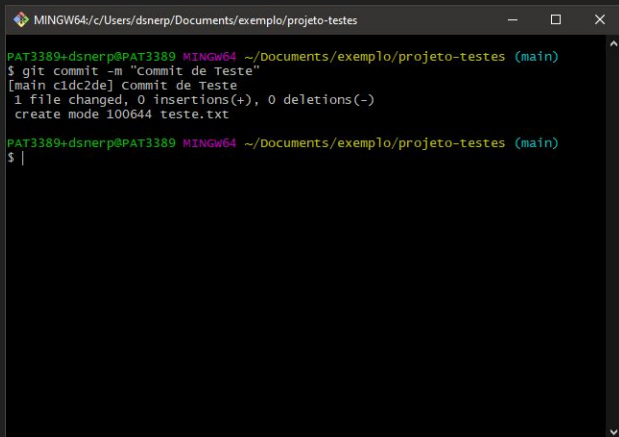
## git add --all

Quando temos vários arquivos podemos usar o git add com a opção --all para registros de vários arquivos.



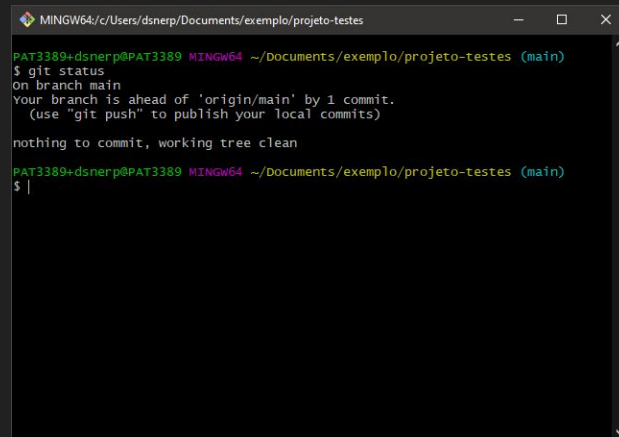
# git commit

Salva as alterações no repositório local e adiciona uma mensagem obrigatória na realização do commit. Quando o commit é realizado as alterações são efetuadas somente no nosso repositório local e não reflete no github.



```
MINGW64~/c:/Users/dsnerp/Documents/exemplo/projeto-testes
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git commit -m "Commit de Teste"
[main c1dc2de] Commit de Teste
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 teste.txt

PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ |
```



```
MINGW64~/c:/Users/dsnerp/Documents/exemplo/projeto-testes
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ |
```

# git log

Exibe o histórico de alterações. Utilizando a opção `--stat` são exibidos quais arquivos foram alterados. As linhas em vermelho abaixo são os commits no repositório remoto e o azul no repositório local.

```
MINGW64: c:/Users/dsnerp/Documents/exemplo/projeto-testes
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git log
commit c1dc2de2fee7007282eb49678d3dc32d18a9ca52 (HEAD -> main)
Author: nicolas.dsn.erp <nicolas.dsn.erp@alterdata.com.br>
Date: Tue Oct 25 17:09:06 2022 -0300

    Commit de Teste

commit a2d379b9c363d1f09d987be92af4e6b10bfade3e (origin/main, origin/HEAD)
Author: Nicolas Theodoro <nicolastheodoro@gmail.com>
Date: Tue Oct 25 16:43:26 2022 -0300

    Update README.md

commit 652e7c9b0f87d59c007ad7483298f75477000922
Author: Nicolas Theodoro <nicolastheodoro@gmail.com>
Date: Tue Oct 25 16:40:24 2022 -0300

    Initial commit

PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$
```

```
MINGW64: c:/Users/dsnerp/Documents/exemplo/projeto-testes
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git log --stat
commit c1dc2de2fee7007282eb49678d3dc32d18a9ca52 (HEAD -> main)
Author: nicolas.dsn.erp <nicolas.dsn.erp@alterdata.com.br>
Date: Tue Oct 25 17:09:06 2022 -0300

    Commit de Teste

    teste.txt | 0
    1 file changed, 0 insertions(+), 0 deletions(-)

commit a2d379b9c363d1f09d987be92af4e6b10bfade3e (origin/main, origin/HEAD)
Author: Nicolas Theodoro <nicolastheodoro@gmail.com>
Date: Tue Oct 25 16:43:26 2022 -0300

    Update README.md

    README.md | 2 +-
    1 file changed, 1 insertion(+), 1 deletion(-)

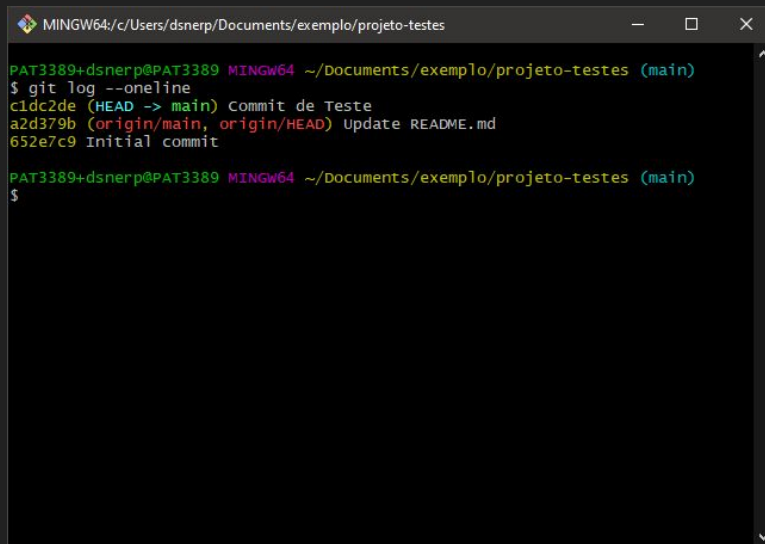
commit 652e7c9b0f87d59c007ad7483298f75477000922
Author: Nicolas Theodoro <nicolastheodoro@gmail.com>
Date: Tue Oct 25 16:40:24 2022 -0300

    Initial commit

    README.md | 2 ++
    1 file changed, 2 insertions(+)
```

# git log --oneline

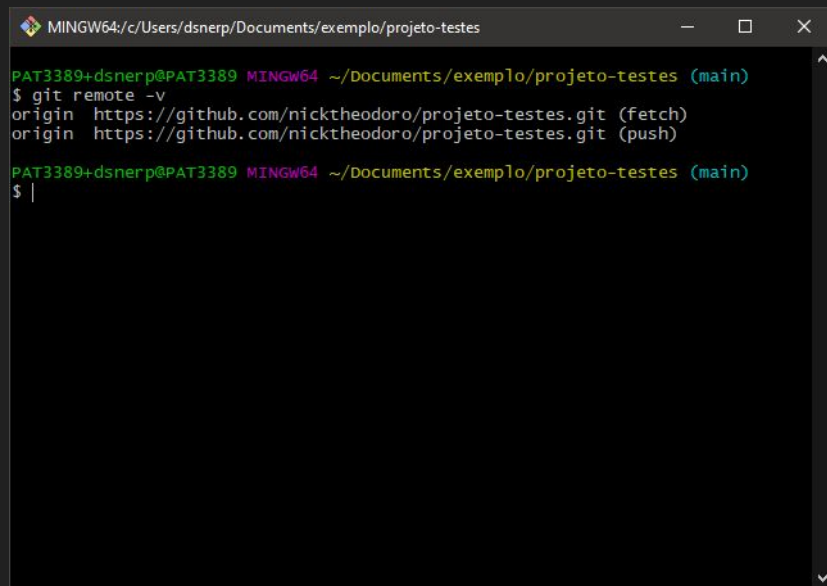
Exibe os commits em apenas uma linha

A screenshot of a Windows terminal window titled 'MINGW64: c:/Users/dsnerp/Documents/exemplo/projeto-testes'. The terminal shows the command 'git log --oneline' being executed. The output displays three commits in a compact, one-line format. The first commit is 'c1dc2de (HEAD -> main) Commit de Teste', the second is 'a2d379b (origin/main, origin/HEAD) Update README.md', and the third is '652e7c9 Initial commit'. The terminal prompt '\$' is visible at the bottom.

```
MINGW64: c:/Users/dsnerp/Documents/exemplo/projeto-testes
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git log --oneline
c1dc2de (HEAD -> main) Commit de Teste
a2d379b (origin/main, origin/HEAD) Update README.md
652e7c9 Initial commit
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$
```

# git remote

Retorna informações sobre o nosso repositório remoto.

A screenshot of a Windows terminal window titled "MINGW64/c/Users/dsnerp/Documents/exemplo/projeto-testes". The terminal shows the command "git remote -v" being executed, which outputs the remote repository information for the "origin" remote, including the fetch and push URLs. The prompt "\$" is visible at the end of the output lines.

```
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git remote -v
origin  https://github.com/nicktheodoro/projeto-testes.git (fetch)
origin  https://github.com/nicktheodoro/projeto-testes.git (push)

PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ |
```

# git push

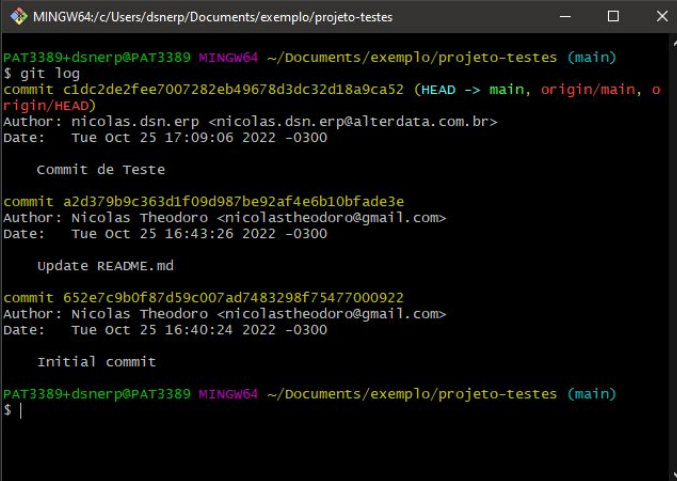
É utilizado para enviar todos os commits locais para o github.

```
MINGW64:/c/Users/dsnerp/Documents/exemplo/projeto-testes
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git push origin main
```

```
MINGW64:/c/Users/dsnerp/Documents/exemplo/projeto-testes
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 282 bytes | 282.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/nicktheodoro/projeto-testes.git
   a2d379b..c1dc2de  main -> main
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$
```

**Inserir usuário e senha do github**

Com os comando foi feito envio da branch local (main) para o servidor pela primeira vez, caso a branch que está sendo enviada não exista no servidor, ela será criada. A partir desse momento, a branch local está configurada para ser rastreada com essa origem no servidor. Obs: Para versionar outras alterações usamos somente o comando git push.

A screenshot of a terminal window titled "MINGW64/c/Users/dsnerp/Documents/exemplo/projeto-testes". The terminal shows the output of the "git log" command. The output lists three commits in reverse chronological order. The first commit is the HEAD, pointing to main, origin/main, and origin/HEAD. The second commit is an update to README.md. The third commit is the initial commit. The terminal prompt is "\$ |".

```
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git log
commit c1dc2de2fee7007282eb49678d3dc32d18a9ca52 (HEAD -> main, origin/main, o
rigin/HEAD)
Author: nicolas.dsn.erp <nicolas.dsn.erp@alterdata.com.br>
Date: Tue Oct 25 17:09:06 2022 -0300

    Commit de Teste

commit a2d379b9c363d1f09d987be92af4e6b10bfade3e
Author: Nicolas Theodoro <nicolastheodoro@gmail.com>
Date: Tue Oct 25 16:43:26 2022 -0300

    Update README.md

commit 652e7c9b0f87d59c007ad7483298f75477000922
Author: Nicolas Theodoro <nicolastheodoro@gmail.com>
Date: Tue Oct 25 16:40:24 2022 -0300

    Initial commit

PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ |
```

Utilizando o comando git log podemos ver a alteração nos dois repositórios local e remoto que possuem as alterações do commit.

```
MINGW64:~/Documents/exemplo/projeto-testes
PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ git log
commit c1dc2de2fee7007282eb49678d3dc32d18a9ca52 (HEAD -> main, origin/main, origin/HEAD)
Author: nicolas.dsn.erp <nicolas.dsn.erp@alterdata.com.br>
Date: Tue Oct 25 17:09:06 2022 -0300

    Commit de Teste

commit a2d379b9c363d1f09d987be92af4e6b10bfade3e
Author: Nicolas Theodoro <nicolastheodoro@gmail.com>
Date: Tue Oct 25 16:43:26 2022 -0300

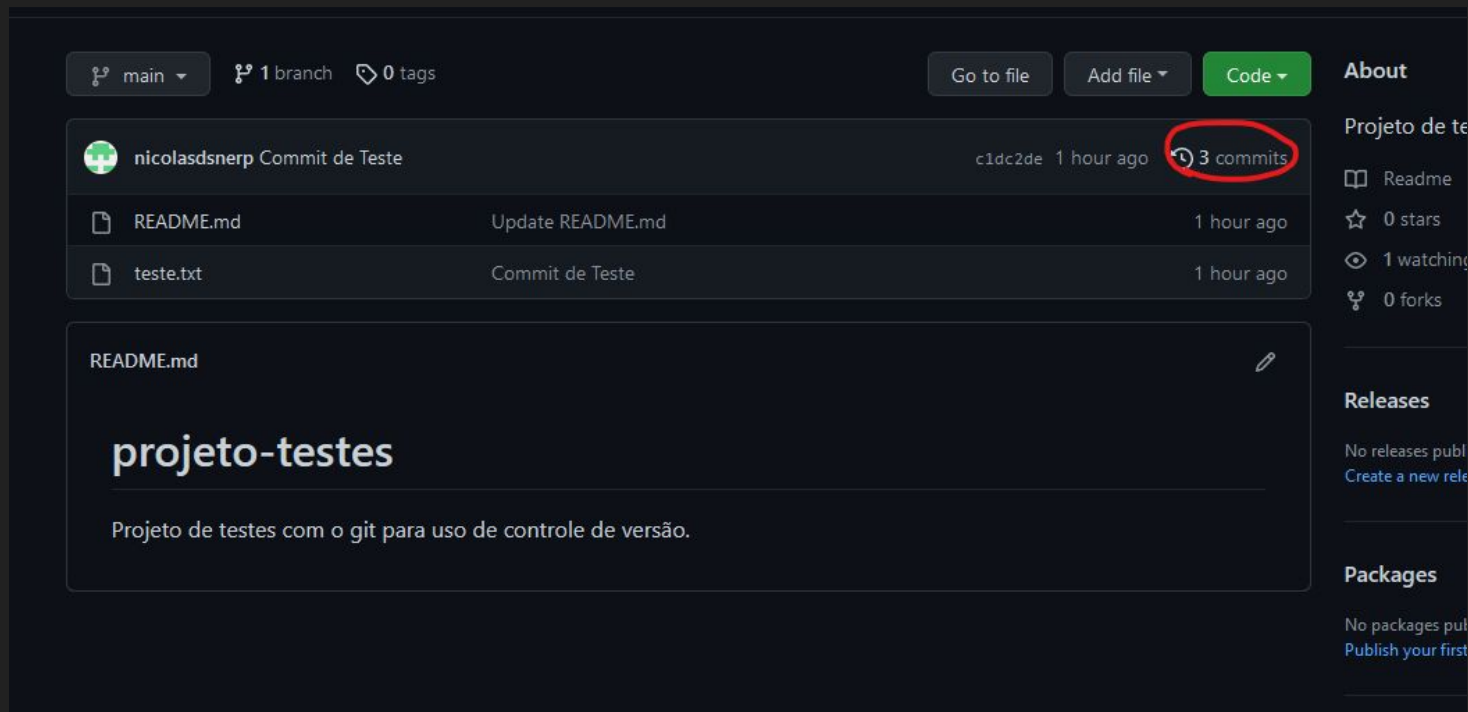
    Update README.md

commit 652e7c9b0f87d59c007ad7483298f75477000922
Author: Nicolas Theodoro <nicolastheodoro@gmail.com>
Date: Tue Oct 25 16:40:24 2022 -0300

    Initial commit

PAT3389+dsnerp@PAT3389 MINGW64 ~/Documents/exemplo/projeto-testes (main)
$ |
```

# Visualização do commit no github



The screenshot displays the GitHub interface for a repository. At the top, navigation links include 'main', '1 branch', and '0 tags'. Action buttons for 'Go to file', 'Add file', and 'Code' are visible. The commit history shows a commit by 'nicolasdsnerp' titled 'Commit de Teste' (hash c1dc2de, 1 hour ago) with a red circle highlighting '3 commits'. Below this, a table lists file changes: 'README.md' (Update README.md, 1 hour ago) and 'teste.txt' (Commit de Teste, 1 hour ago). The 'README.md' content is shown below, featuring the title 'projeto-testes' and the description 'Projeto de testes com o git para uso de controle de versão.' The right sidebar contains sections for 'About', 'Releases', and 'Packages'.

main 1 branch 0 tags

Go to file Add file Code

nicolasdsnerp Commit de Teste c1dc2de 1 hour ago 3 commits

File	Commit Message	Time
README.md	Update README.md	1 hour ago
teste.txt	Commit de Teste	1 hour ago

README.md

## projeto-testes

Projeto de testes com o git para uso de controle de versão.

**About**

Projeto de testes

Readme

0 stars

1 watching

0 forks

**Releases**

No releases published

[Create a new release](#)

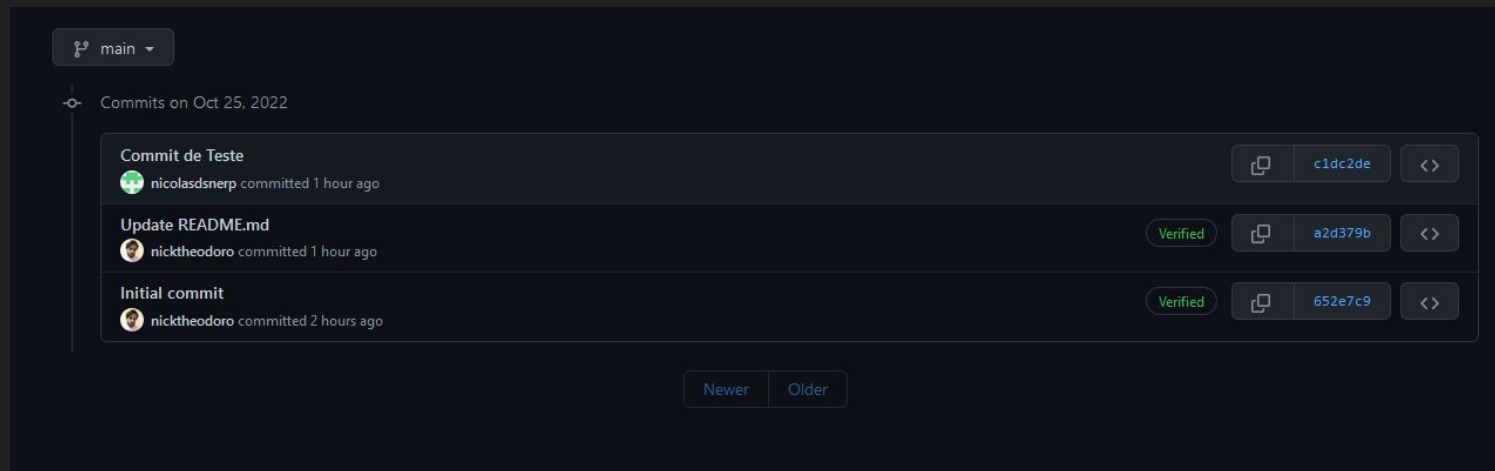
**Packages**

No packages published

[Publish your first package](#)



# É como se fosse o gitlog no github



# Clicando em cada commit conseguimos ver as informações dos arquivos no commit

