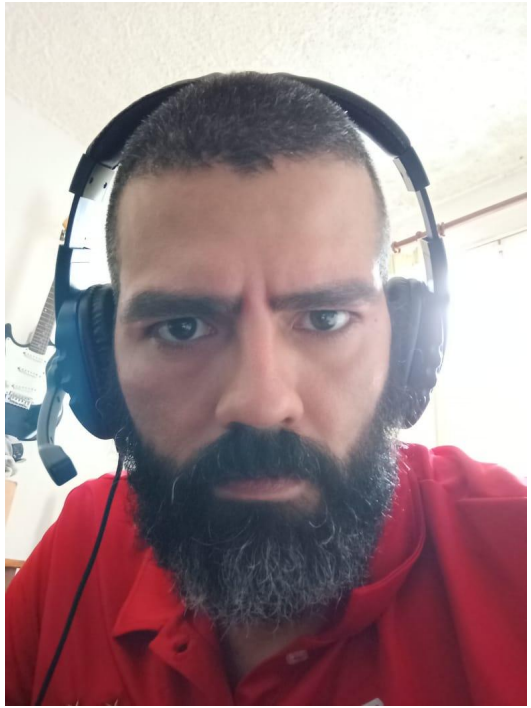
# GIT

## Introdução ao Git e Github com repositório público

# Tales Oliver

Programador FullStack Teknisa



# GIT

## O que é GIT?

> O Git é um sistema open-source de controle de versão utilizado pela grande maioria dos desenvolvedores atualmente. Com ele podemos criar todo histórico de alterações no código do nosso projeto e facilmente voltar para qualquer ponto para saber como o código estava naquela data.

VERSÃO

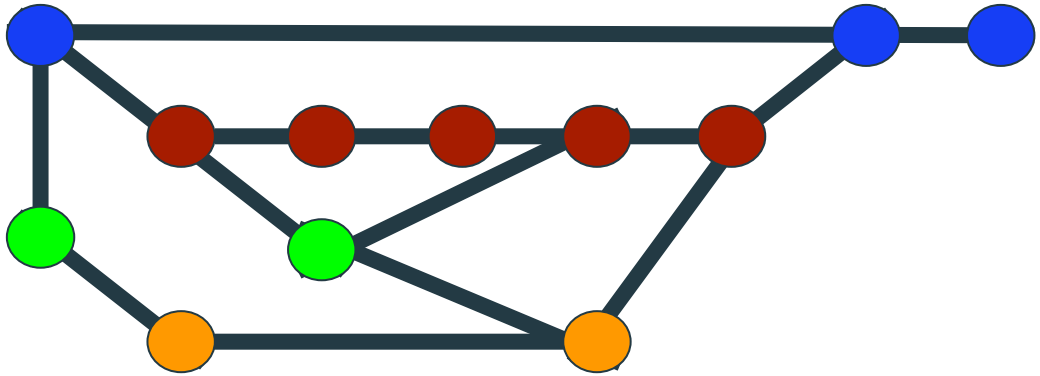
0.1 0.2 0.3 0.4 0.5 0.6 0.7 1.0

MASTER

DEV

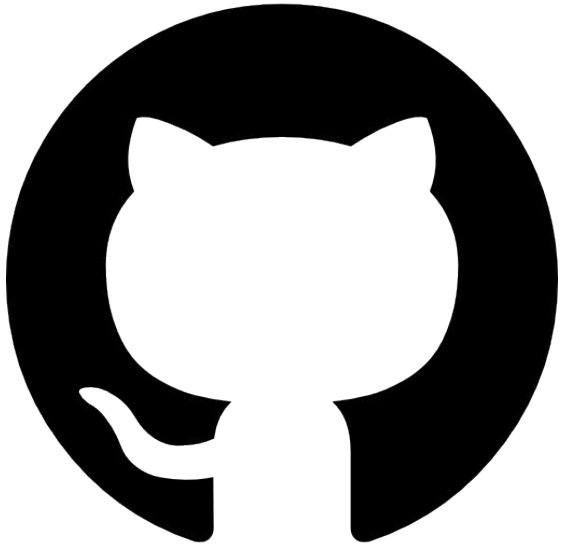
FEATURE

BUG



# GITHUB

## O que é Github?



> O **Github** é um serviço online de hospedagem de **repositórios Git** (como são chamados os projetos que utilizam Git). Com ele podemos manter todos nossos *commits* e ramos sincronizados entre os membros do time.

Under0Cover

Analista e Desenvolvedor de Sistemas - Unopar - Programador Full Stack Jr

Edit profile

**23** followers · **61** following

Teknisa
Uberaba - MG
@Oliver_Cover

## Achievements

Overview | Repositories 14 | Projects | Packages | Stars

## Popular repositories

Customize your pins

- **Google-Glass-Guanabara-** — Public — HTML
- **Cursos_Site_AulaEaD** — Public — JavaScript
- **Cursos_DevMedia** — Public — HTML
- **Portugol** — Public
- **Curso_Em_Video** — Public — Python
- **Faculdade** — Public — Java

## 450 contributions in the last year

Contribution settings


Learn how we count contributions

Less More

# GIT CONFIG

## Conferindo/Setando configurações iniciais

- ➔ git config --list
- ➔ git config --global user.name ‘seu_nome’
- ➔ git config --global user.email ‘seu_email@exemplo.com’

- ➔ git config user.name
- ➔ git config user.email

```
talis@Oliver  ~
$ 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=E:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Tales Oliver
user.email=talis.oliveira@hotmail.com.br
core.editor="C:\Users\talis\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
credential.http://code.zeedhi.com.provider=generic

talis@Oliver  ~
$
```

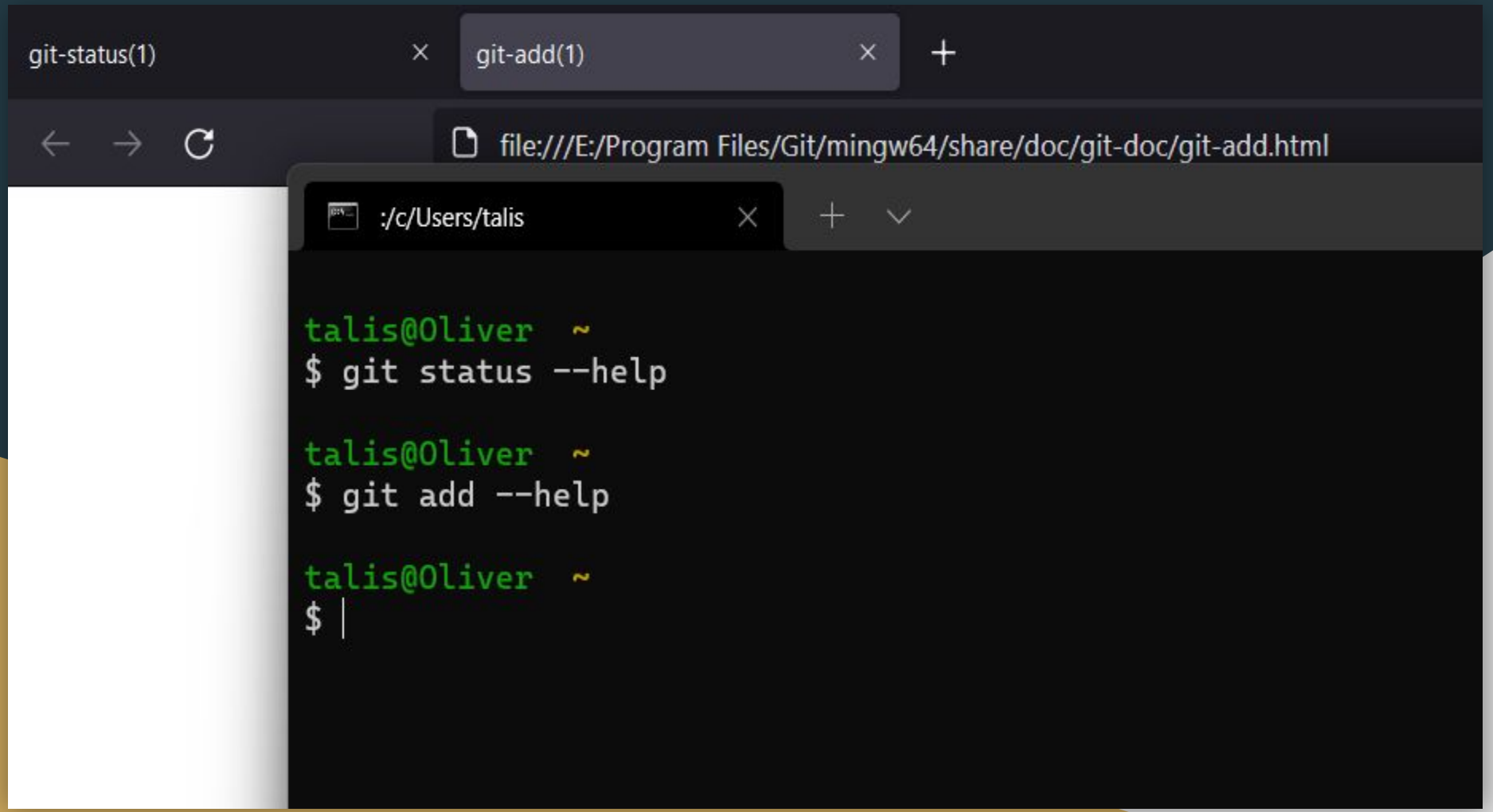
```
talis@Oliver  /d/estudos/curso_git_oliver (main)
$ git config user.name
Tales Oliver

talis@Oliver  /d/estudos/curso_git_oliver (main)
$ git config user.email
talis.oliveira@hotmail.com.br
```

# GIT HELP

Conseguindo ajuda sobre os comandos git

- ➔ git <comando> --help


git-status(1)
git-add(1)
file:///E:/Program Files/Git/mingw64/share/doc/git-doc/git-add.html
:/c/Users/talis
talis@Oliver ~
$ git status --help
talis@Oliver ~
$ git add --help
talis@Oliver ~
$

```
talis@Oliver  ~
$ git --help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone     Clone a repository into a new directory
   init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add       Add file contents to the index
   mv        Move or rename a file, a directory, or a symlink
   restore   Restore working tree files
   rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect    Use binary search to find the commit that introduced a bug
   diff      Show changes between commits, commit and working tree, etc
   grep      Print lines matching a pattern
   log       Show commit logs
   show      Show various types of objects
   status    Show the working tree status

grow, mark and tweak your common history
   branch    List, create, or delete branches
   commit    Record changes to the repository
   merge     Join two or more development histories together
   rebase    Reapply commits on top of another base tip
   reset     Reset current HEAD to the specified state
   switch    Switch branches
   tag       Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
   fetch     Download objects and refs from another repository
   pull      Fetch from and integrate with another repository or a local branch
```

# GIT FORK

Aprendendo a trabalhar com uma branch existente

➔ https://github.com/Under0Cover/curso_git_oliver

# GIT CLONE

Aprendendo a trabalhar com uma branch existente

➔ git clone https://github.com/SEU_USUARIO/curso_git_oliver

```
talis@Oliver  /d/estudos
$ git clone https://github.com/Under0Cover/curso_git_oliver.git
Cloning into 'curso_git_oliver'...
remote: Enumerating objects: 37, done.
remote: Counting objects: 100% (37/37), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 37 (delta 4), reused 37 (delta 4), pack-reused 0Receiving objects:  81% (30/37)
Receiving objects: 100% (37/37), 637.02 KiB | 8.97 MiB/s, done.
Resolving deltas: 100% (4/4), done.
```

# GIT INIT

Iniciando seu projeto

- ➔ git init

```
talis@Oliver  /d/estudos/teste
$ git init
Initialized empty Git repository in D:/estudos/teste/.git/
```

# GIT STATUS

Verificando o status do repositório

- ➔ git status

```
talis@Oliver  /d/estudos/teste (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

```
talis@Oliver  /d/estudos/teste (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        pasta1/
        pasta2/
        pasta3/

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

# GIT ADD

Adicionando arquivo a serem commitados

- ➔ git add .
- ➔ git add <file>

```
talis@Oliver  /d/estudos/teste (master)
$ git add pasta2/

talis@Oliver  /d/estudos/teste (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   pasta2/conteudo.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        pasta3/
```

```
talis@Oliver  /d/estudos/teste (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   pasta1/conteudo.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        pasta2/
        pasta3/
```

GIT ADD <file>

STAGED

LOCAL

# GIT COMMIT

## Criando o primeiro COMMIT

- ➔ git commit -m ‘message’

```
talis@Oliver  /d/estudos/teste (master)
$ git commit -m 'first commit'
[master (root-commit) 86b0b06] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 pasta1/conteudo.txt

talis@Oliver  /d/estudos/teste (master)
$
```

# GIT PUSH

Enviando sua nova versão ao Hub

- ➔ git remote add <name> <url>
- ➔ git push

- OBS: O PRIMEIRO COMANDO PODERÁ SER NECESSÁRIO PARA ADICIONAR SEU REPOSITÓRIO LOCAL COM O HUB DE ARMAZENAMENTO

```
talis@Oliver  /d/estudos/teste (master)
$ git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>      1

and then push using the remote name

    git push <name>    2
```

```
talis@Oliver  /d/estudos/curso_git_oliver (main)
$ git push
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 91.50 KiB | 22.88 MiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/Under0Cover/curso_git_oliver.git
   49f3ef5..fec6a44  main -> main

talis@Oliver  /d/estudos/curso_git_oliver (main)
$
```

# GIT PULL

Atualizando seu repositório local

- ➔ git pull

```
talis@Oliver  /d/estudos/curso_git_oliver (main)
$ git pull
Already up to date.
```

# BRANCHING

## Afinal, o que é BRANCH?

Branch, do inglês, significa ‘ramo’

É uma ramificação do código que está sendo construído

- ➔ git branch
- ➔ git branch <nova_branch>

```
talis@Oliver  /d/estudos/teste (master)
$ git branch outro_teste

talis@Oliver  /d/estudos/teste (master)
$ git branch
* master
  outro_teste

talis@Oliver  /d/estudos/teste (master)
$ git branch teste2

talis@Oliver  /d/estudos/teste (master)
$ git branch
* master
  outro_teste
  teste2

talis@Oliver  /d/estudos/teste (master)
$ git branch -D teste2
Deleted branch teste2 (was 86b0b06).

talis@Oliver  /d/estudos/teste (master)
$ git branch
* master
  outro_teste
```

# GIT LOG

Visualizando o log de gits

- ➔ git log
- ➔ git show

```
talis@Oliver  /d/estudos/curso_git_oliver/git_2022 (main)
$ git log
commit 1bf48c674ecca98fd5225390cf27fd08cc6e2ade (HEAD -> main, origin/main, origin/HEAD)
Author: Tales Oliver <talis.oliveira@hotmail.com.br>
Date:   Mon Mar 7 14:44:04 2022 -0300

    evolucao

commit ec8b68b2f8413ab529d4a765e04c63d76576f11d
Author: Tales Oliver <talis.oliveira@hotmail.com.br>
Date:   Thu Feb 24 22:21:51 2022 -0300

    atualizacoes curso

commit fec6a44784283b8db90164079ce3758a838c141c
Author: Tales Oliver <talis.oliveira@hotmail.com.br>
Date:   Mon Feb 21 22:48:11 2022 -0300

    continuacao_criacao_curso
```

código commit

autor

data e hora

comentário

```
talis@Oliver  /d/estudos/curso_git_oliver/git_2022 (main)
$ git show
commit 1bf48c674ecca98fd5225390cf27fd08cc6e2ade (HEAD -> main, origin/main, origin/HEAD)
Author: Tales Oliver <talis.oliveira@hotmail.com.br>
Date:   Mon Mar 7 14:44:04 2022 -0300

    evolucao

diff --git a/git_2022/git_contribuicoes_bruno_orlandi/notes.txt b/git_2022/git_contribuicoes_bruno_orlandi/notes.txt
deleted file mode 100644
index bf9b43d..0000000
--- a/git_2022/git_contribuicoes_bruno_orlandi/notes.txt
+++ /dev/null
@@ -1,85 +0,0 @@
-
```

# GIT CHECKOUT

Trocando de Branch ou commit

- ➔ git checkout <branch>
- ➔ git checkout -B <nova_branch>
- ➔ git checkout <commit> <file>

```
talis@Oliver  /d/estudos/curso_git_oliver/git_2022 (main)
$ git checkout ec8b68b2
Note: switching to 'ec8b68b2'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at ec8b68b atualizacoes curso

talis@Oliver  /d/estudos/curso_git_oliver/git_2022 ((ec8b68b...))
$
```

```
talis@Oliver  /d/estudos/curso_git_oliver/git_2022 ((ec8b68b...))
$ git checkout main
Previous HEAD position was ec8b68b atualizacoes curso
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

talis@Oliver  /d/estudos/curso_git_oliver/git_2022 (main)
$
```


talis@Oliver /d/estudos/teste (master)
$ git checkout -b checkteste
Switched to a new branch 'checkteste'
talis@Oliver /d/estudos/teste (checkteste)
$