

# ***Git para Principiantes***

O mínimo que você precisa saber



Fabio Beneditto

(v 1.1)

# Sobre



**Fabio Beditto**

Web Developer

Entusiasta de FOSS

Voluntário Tchelinux

Campusero: #CPBR8, 9,11 e 12

Mozillian – Voluntário SUMO

Praticante de #mototerapia

**[about.me/fabiobeneditto](https://about.me/fabiobeneditto)**

Github / Gitlab / Twitter / Telegram:

**[@fabiobeneditto](#)**



**DISCLAIMER**

# Motivação



- *Esse é o Git. Ele mostra nosso trabalho em projetos através de uma árvore distribuída.*
- *Legal! E como vamos usar ele?*
- *Não faço ideia. Apenas decore esses comandos e digite-os para sincronizar tudo. Se der erro, salve seu trabalho em algum lugar, apague o projeto e faça uma nova cópia.*

(tradução livre) - Fonte: <https://xkcd.com/1597/>

# O que é Controle de Versão?

# O que é Controle de Versão?

- Sistema que registra as mudanças feitas
  - Em um arquivo, ou vários
- De forma que seja fácil a recuperação
  - Versões específicas, anteriores
- Pode ser utilizado com qualquer tipo de arquivo
  - Código-fonte é mais comum

**Fonte:** <https://git-scm.com/book/pt-br/v2>

**Por que usar  
Controle de Versão?**

# Por que usar Controle de Versão?

- Manter histórico de alterações no código
  - Autores (**Quem?**)
  - Data das alterações (**Quando?**)
  - Motivo das alterações (**Por que?**)
- Facilitar a colaboração
  - pessoas podem trabalhar juntas
- Auxiliar na identificação e resolução de problemas



O que é **GIT** ?

# O que é GIT?

- Sistema de controle de versão distribuído
- Criado por Linus Torvalds em 2005
- Simples e rápido
  - Armazena *snapshots*
- É Livre



**git**



**github**  
SOCIAL CODING

# **GIT != GitHub**

- Serviço de Hospedagem de Código
- Criado em 2008
- Usa GIT
- Comprado pela Microsoft em **04/06/2018**
  - Por U\$ 7,5 bilhões
- **NÃO É Livre (ainda)**
  - As Free Speech

# Utilização básica de Git

*“Talk is Cheap: show me the code”*

# Configurando o cliente Git

```
$ git config --global user.name "John Doe"
```

```
$ git config --global user.email "john@doe.com"
```

```
$ cat .gitconfig
```

```
[user]
```

```
    name = John Doe
```

```
    email = john@doe.com
```

# Clonando um repositório

```
$ mkdir Projeto
```

```
$ cd Projeto
```

```
$ git clone \  
https://gitlab.com/fabiobeneditto/tchelinux-  
git.git Tchelinux
```

```
Cloning into 'Tchelinux'...
```

```
done.
```

# Verificando alterações

```
$ vim hello.md
```

```
$ git status
```

```
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
hello.md
```

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



# Verificando alterações

```
$ vim hello.md
```

```
$ git status -s
```

```
?? hello.md
```

# Adicionando arquivos ao index

```
$ git add hello.md
```

```
$ git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
modified:   hello.md
```

...e o que é *index* ?

# Adicionando arquivos ao index

“**Index**” é a *stage area*:

- um simples arquivo que armazena informações sobre o que irá em seu próximo *commit*. Você, durante o ciclo:
  - Modifica arquivos no seu diretório de trabalho.
  - Seleciona os arquivos, adicionando *snapshots* deles para o *index*.
  - Faz um *commit*, levando os arquivos como eles estão no *index* e os armazena permanentemente

# Fazendo seu primeiro commit

```
$ git commit -m "Versão Inicial de hello.md"
```

```
[master (root-commit) f480dd9] Versão Inicial de hello.md
```

```
1 file changed, 3 insertions(+)
```

```
create mode 100644 hello.md
```

# Fazendo os demais commits

```
$ vim hello.md
```

```
$ git add hello.md
```

```
$ git commit -m "Bloco Extra em hello.md"
```

```
[master 97e567d] Bloco Extra em hello.md
```

```
1 file changed, 1 insertion(+), 1 deletion(-)
```

# Fazendo os demais commits

```
$ vim hello.md
```

```
$ git add -p (ou --patch)
```

```
diff --git a/hello.md b/hello.md
```

```
index 9cff701..aa1314b 100644
```

```
--- a/hello.md
```

```
+++ b/hello.md
```

```
@@ -4,3 +4,5 @@ Exemplo de como fica um arquivo no Git
```

```
## Como fazer
```

```
Adicionando uma segunda linha e um ponto.
```

```
+
```

```
+...e mais uma
```

```
Stage this hunk [y,n,q,a,d,/,e,]?
```

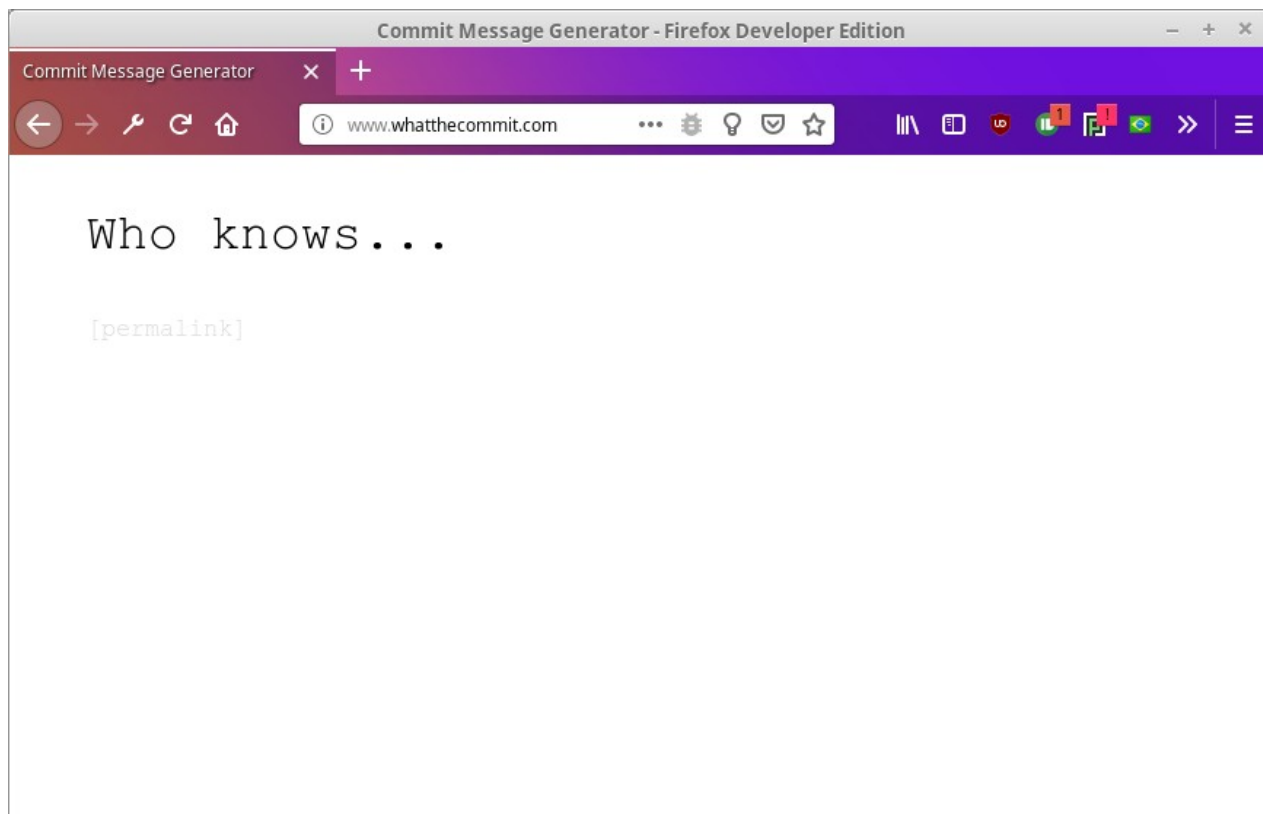
# Use mensagens explicativas!

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT  
MESSAGES GET LESS AND LESS INFORMATIVE.



# Use mensagens explicativas!



Fonte: <http://www.whatthecommit.com/>

# Use mensagens explicativas!

```
fabio@fabio-portege:[~/www/Tchelinux]: (master)$ git log
commit 3e9fc4376095e22051991cb8b1b4f736d8beedf6 (HEAD -> master)
Author: Fabio Beneditto <fabiobeneditto@gmail.com>
Date: Wed Sep 20 12:58:02 2017 -0300
```

Descricao Curta da tarefa

Slides

Descrição Detalhada do Commit, onde você pode detalhar o que foi modificado em seu código e que pode servir para futuras consultas. Nem sempre você vai ficar de dono desse código, os demais (ou o Você do Futuro) merecem saber o motivo de ter feito essas alterações. Ainda, elas não aparecem na versão resumida do git log.

```
commit cf8cc84e4ac880d6fd7ee94739f91ba4cc442c6c hello.md
Author: Fabio Beneditto <fabiobeneditto@gmail.com>
Date: Sun Sep 17 16:16:32 2017 -0300 $ git add hello.md

Bloco Extra em hello.md $ git commit
```

# Use mensagens explicativas!

```
$ vim hello.md
```

```
$ git add hello.md
```

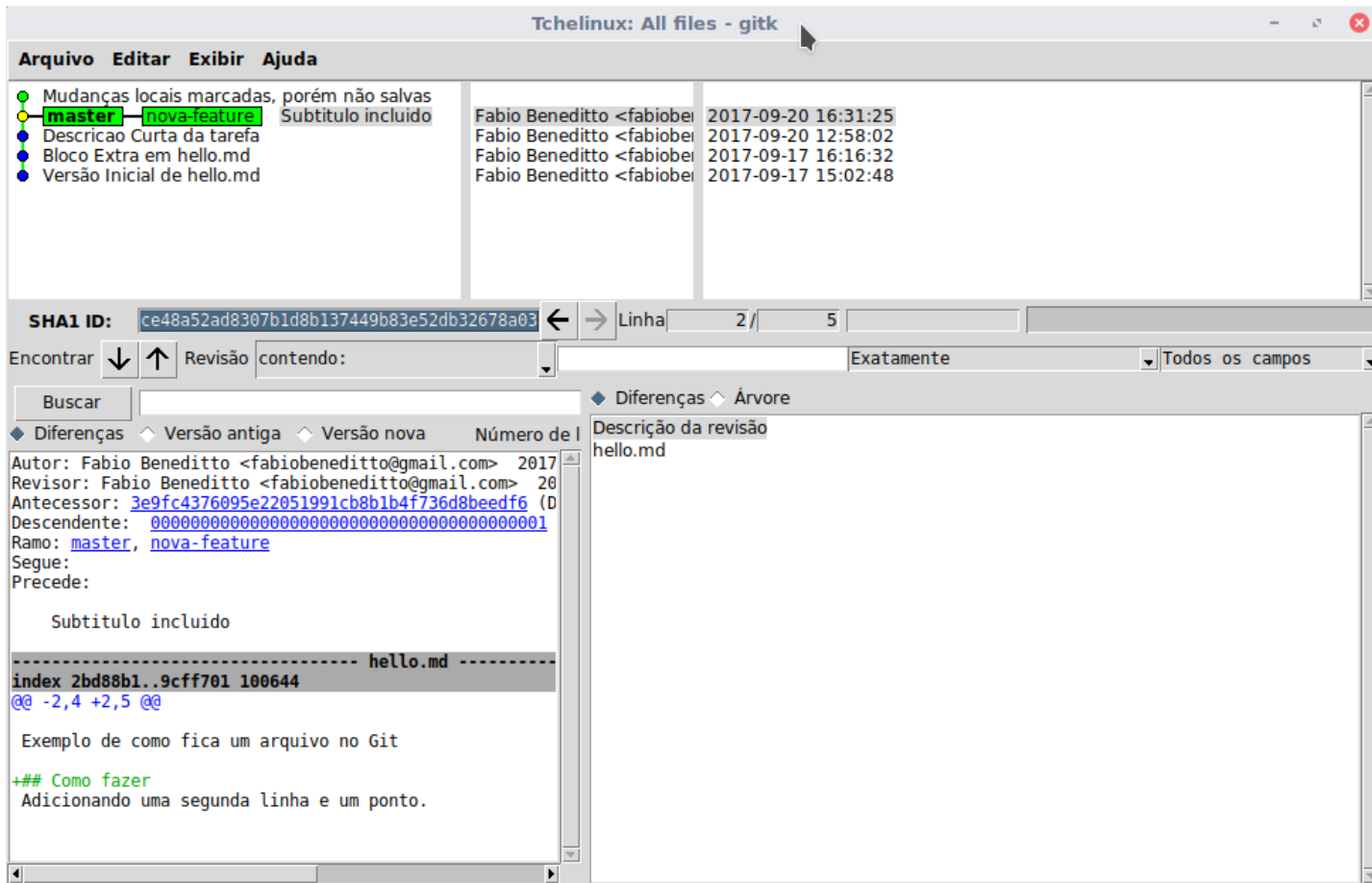
```
$ git commit
```

```
  [] Descricao Curta
```

```
Descrição Detalhada do Commit
```

```
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.  
# On branch master  
# Changes to be committed:  
#       modified:   hello.md  
#
```

# Use mensagens explicativas!



# Enviando tudo para o servidor remoto

```
$ git push -u origin -all
```

```
Counting objects: 12, done.
```

```
Delta compression using up to 4 threads.
```

```
Compressing objects: 100% (8/8), done.
```

```
Writing objects: 100% (12/12), 1.28 KiB | 653.00 KiB/s, done.
```

```
Total 12 (delta 2), reused 0 (delta 0)
```

```
To gitlab.com:fablobeneditto/tchelinix-git.git
```

```
* [new branch]      master -> master
```

```
* [new branch]      nova-feature -> nova-feature
```

```
Branch master set up to track remote branch master from origin.
```

```
Branch nova-feature set up to track remote branch nova-feature from origin.
```

# Uso menos básico de Git

*“Básica, mas nem tanto”*

# Criando um repositório

```
$ mkdir Projeto
```

```
$ cd Projeto
```

```
$ git init
```

```
Initialized empty Git repository in Projeto/.git/
```

# Reescrevendo o último commit

```
$ vim hello.md
```

```
$ git add hello.md
```

```
$ git commit --amend
```

```
[master dd259b4] Bloco Extra em hello.md
```

```
Date: Sun Sep 17 15:36:51 2017 -0300
```

```
1 file changed, 3 insertions(+), 1 deletion(-)
```



# Reescrevendo o último commit

```
$ git commit --amend
```

```
Bloco Extra em hello.md
```

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sun Sep 17 15:36:51 2017 -0300
#
# On branch master
#
# Changes to be committed:
#
#       modified:   hello.md
```

# Removendo o último commit

```
$ git reset --hard HEAD~1
```

```
HEAD is now at f480dd9 Versão Inicial de hello.md
```

# Descartando alterações antes do commit

```
$ git checkout -p
```

```
diff --git a/hello.md b/hello.md
```

```
index aba21ac..2cc5dc2 100644
```

```
--- a/hello.md
```

```
+++ b/hello.md
```

```
@@ -1,3 +1,5 @@
```

```
# Hello World
```

Exemplo de como fica um arquivo no Git

+

+Adicionando uma terceira linha

Discard this hunk from worktree [y,n,q,a,d,/,e,]?

# Listando commits realizados

```
$ git log
```

```
commit f480dd96974c5eafea736a4286db353172e960e3 (HEAD -> master)
```

```
Author: Fabio Beneditto <fabio beneditto@gmail.com>
```

```
Date:    Sun Sep 17 15:02:48 2017 -0300
```

```
Versão Inicial de hello.md
```

```
$ git log --pretty=oneline
```

```
f480dd96974c5eafea736a4286db353172e960e3 (HEAD -> master) Versão  
Inicial de hello.md
```

```
$ git log --oneline
```

```
f480dd9 (HEAD -> master) Versão Inicial de hello.md
```

# Visualizando detalhes de um commit

```
$ git show f480dd9
```

```
commit f480dd96974c5eafea736a4286db353172e960e3 (HEAD -> master)
```

```
Author: Fabio Beneditto <fabio beneditto@gmail.com>
```

```
Date:    Sun Sep 17 15:02:48 2017 -0300
```

```
Versão Inicial de hello.md
```

```
diff --git a/hello.md b/hello.md
```

```
new file mode 100644
```

```
index 0000000..aba21ac
```

```
--- /dev/null
```

```
+++ b/hello.md
```

```
@@ -0,0 +1,3 @@
```

```
+# Hello World
```

```
+
```

```
+Exemplo de como fica um arquivo no Git
```

# Exibindo diferenças entre commits

```
$ git diff f480dd9..cf8cc84
```

```
diff --git a/hello.md b/hello.md
```

```
index aba21ac..2cc5dc2 100644
```

```
--- a/hello.md
```

```
+++ b/hello.md
```

```
@@ -1,3 +1,5 @@
```

```
 # Hello World
```

Exemplo de como fica um arquivo no Git

+

+Adicionando uma terceira linha

# Criando e usando branches

```
$ git branch nova-feature
```

```
$ git branch
```

```
* master
```

```
nova-feature
```

```
$ git checkout nova-feature
```

```
Switched to branch 'nova-feature'
```

```
$ git branch
```

```
master
```

```
* nova-feature
```

# Fazendo merge de branches

```
$ git checkout master
```

```
Switched to branch 'master'
```

```
$ git merge nova-feature
```

```
Updating 3e9fc43..ce48a52
```

```
Fast-forward
```

```
hello.md | 1 +
```

```
1 file changed, 1 insertion(+)
```

```
commit f480dd96974c5eafea736a4286db353172e960e3
```

```
$ git log --oneline
```

```
ce48a52 (HEAD -> master, nova-feature) Subtitulo incluido
```

```
3e9fc43 Descricao Curta da tarefa
```

```
cf8cc84 Bloco Extra em hello.md
```

```
f480dd9 Versão Inicial de hello.md
```



# Utilização de repositórios remotos com Git

*“Já tenho um repositório. #comofas ?”*

# Atualizando um repositório local

```
$ cd Tchelinux/
```

```
$ git checkout master
```

```
$ git pull --rebase
```

# Listando *remotes* de repositório

```
$ cd Tchelinux/
```

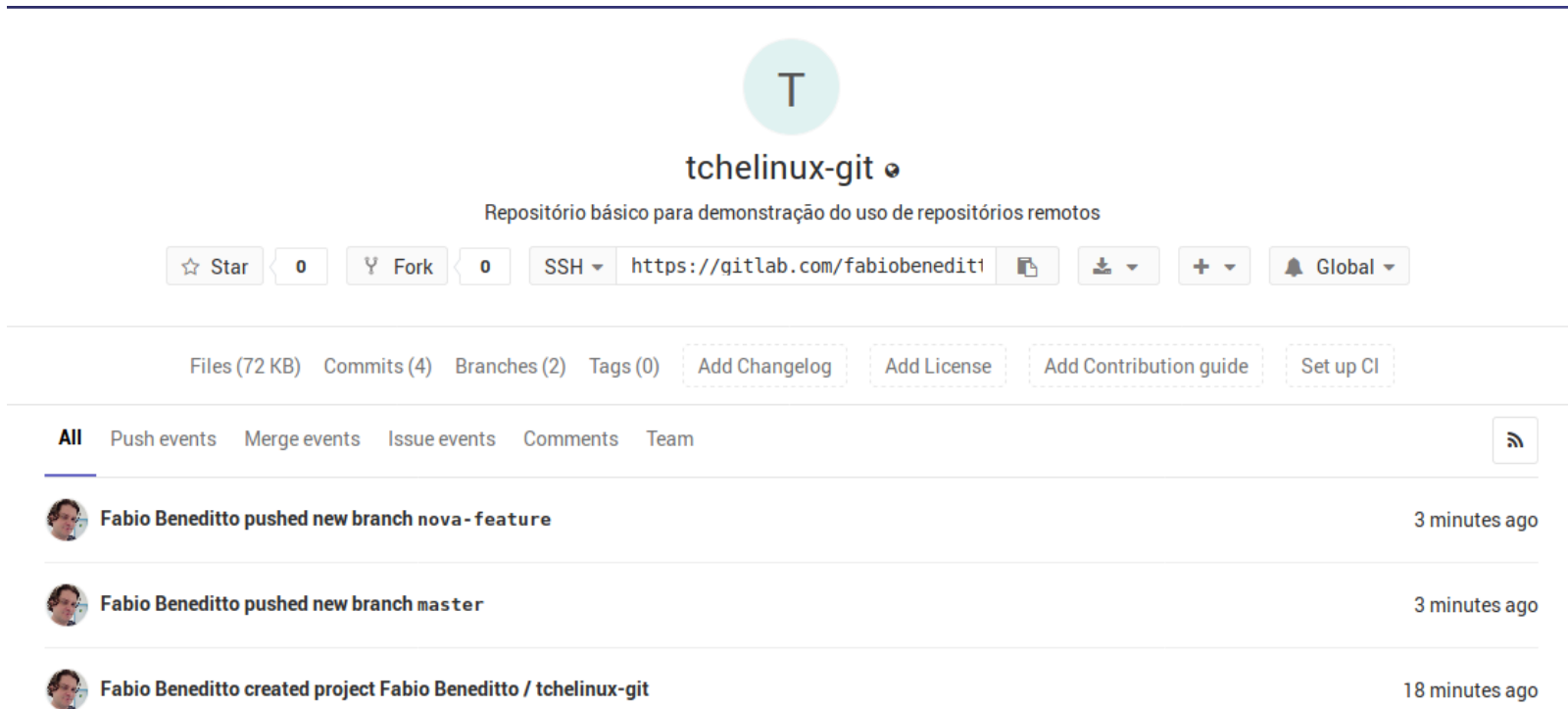
```
$ git remote
```

# Adicionando *remotes* ao repositório

```
$ cd Tchelinux
```

```
$ git remote add origin \  
git@gitlab.com:fabio beneditto/tchelinux-git.git
```

# Verificando tudo






The screenshot shows the GitLab interface for a repository named 'tchelixux-git'. At the top, there is a teal circular avatar with the letter 'T'. Below it, the repository name 'tchelixux-git' is displayed with a small lock icon. A description reads 'Repositório básico para demonstração do uso de repositórios remotos'. The interface includes buttons for 'Star' (0), 'Fork' (0), and 'SSH' with a dropdown menu showing the URL 'https://gitlab.com/fabiobenedit1'. There are also buttons for 'Add Changelog', 'Add License', 'Add Contribution guide', and 'Set up CI'. A navigation bar at the bottom shows 'All' selected, with other options like 'Push events', 'Merge events', 'Issue events', 'Comments', and 'Team'. A feed of recent activity is visible, showing three events by 'Fabio Beneditto': 'pushed new branch nova - feature' (3 minutes ago), 'pushed new branch master' (3 minutes ago), and 'created project Fabio Beneditto / tchelixux-git' (18 minutes ago).


T


tchelixux-git


Repositório básico para demonstração do uso de repositórios remotos


☆ Star 0 Fork 0 SSH <https://gitlab.com/fabiobenedit1>   +  Global

Files (72 KB) Commits (4) Branches (2) Tags (0) [Add Changelog](#) [Add License](#) [Add Contribution guide](#) [Set up CI](#)

All Push events Merge events Issue events Comments Team 

 Fabio Beneditto pushed new branch nova - feature 3 minutes ago

 Fabio Beneditto pushed new branch master 3 minutes ago

 Fabio Beneditto created project Fabio Beneditto / tchelixux-git 18 minutes ago

`https://gitlab.com/fabiobeneditto/tchelixux-git`

E como aprender mais ?

# E como aprender mais?

- **Git Book** - <https://git-scm.com/book/pt-br/v2>
- **Git Cheat Sheet** - <https://www.git-tower.com/blog/git-cheat-sheet/>
- **OwShitGit** - <http://ohshitgit.com/>
- **Git Explorer** - <https://gitexplorer.com>
- Criar contas e usar:
  - **Gitlab** - <https://gitlab.com/>
  - **Github** - <https://github.com/>
- Clonar repositórios e colaborar com projetos

# Perguntas?

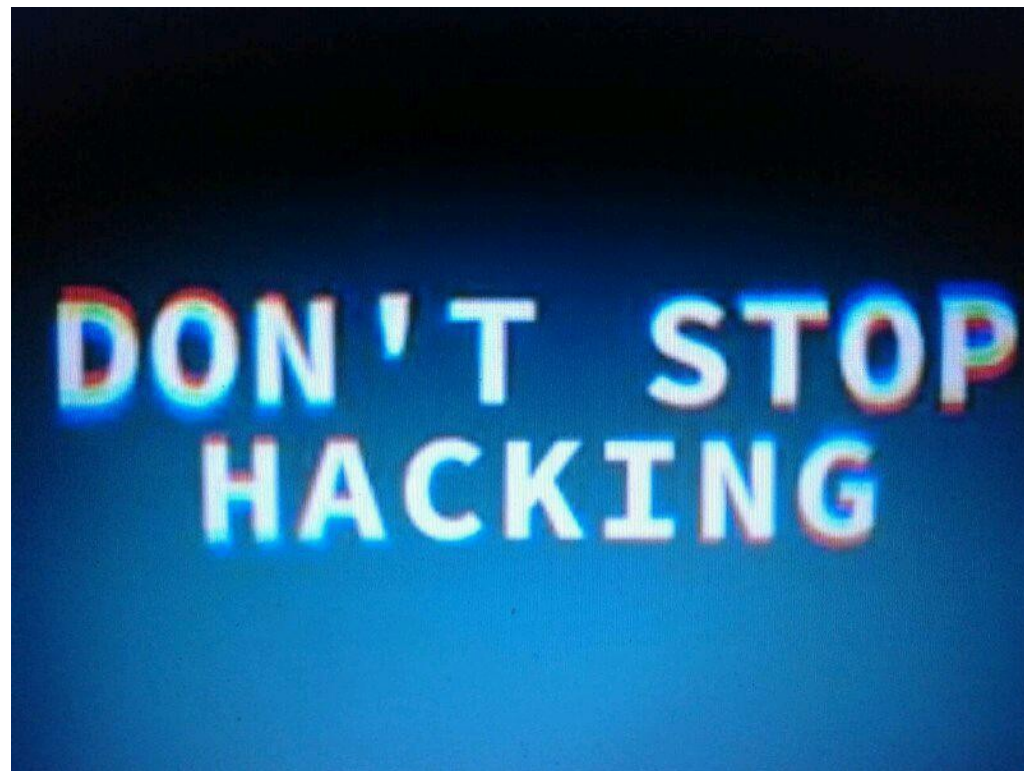




# Muito Obrigado!

## Agradecimentos:

- Leonardo Vaz
- Julio Biason
- Cláudio “Patola” Sampaio
- Tchelinux
- Campus Party Brasil



<https://slideshare.net/fabiobeneditto>