

Curso de Git e Github

Prof. Anderson Coimbra Choren

Vídeos: <https://www.youtube.com/watch?v=beMnH51P-T4&list=PLmMxPWmzYRGcTabffOwHBORBjtKa2wCXS>

GitHub: https://github.com/professorchoren/curso_de_git_e_github

Resumo feito por Roberto Pinheiro (<https://github.com/betopinheiro1005>)

Aula 01 - Controle de versão

SCV, o que são e qual sua importância

Modelo tradicional de versionamento

Gravando os arquivos de cada versão do projeto em um arquivo compactado ZIP diferente.

Desvantagens:

- Não sabemos o que mudou de uma versão para outra;
- Não sabemos quem e nem quando essas modificações foram feitas;
- Grande chance de haver divergências de código caso estejamos trabalhando em equipe;
- Gasto excessivo de espaço em disco;

A solução é usar um sistema de controle de versão (SCV).

Vantagens de se usar um SCV

- Controle de histórico;
- Trabalho em equipe;
- Marcação e resgate de versões estáveis;
- Ramificação do projeto;
- Segurança;
- Rastreabilidade;
- Organização;
- Confiança;

Tipos de controle de versão

Os SCV se dividem em dois grupos:

1. Centralizados
2. Distribuídos

Sistema de controle centralizado

- O servidor possui todo o controle de versão do código-fonte.
- Basicamente, os clientes sempre trabalham em rede com os arquivos no servidor.

Exemplos:

- CVS
- Microsoft Visual SourceSafe 6.0
- Apache Subversion ou SVN

Sistema de controle distribuído

- Não há a necessidade de um servidor mas pode haver;
- Os clientes possuem toda a cópia do código-fonte localmente;
- Ou seja: o cliente tem a cópia local (em seu computador) e a cópia remota (no servidor);

Exemplos:

- GIT (o mais conhecido)
- Mercurial

Aula 02 - Introdução ao GIT e sua importância para nós desenvolvedores

- O Git trabalha com o conceito de repositórios.

- Um repositório é uma pasta a qual o Git deverá monitorar. Ou seja, é a pasta onde o Git atuará gerenciando as diversas versões dos arquivos ali presentes.

Os três estados do Git

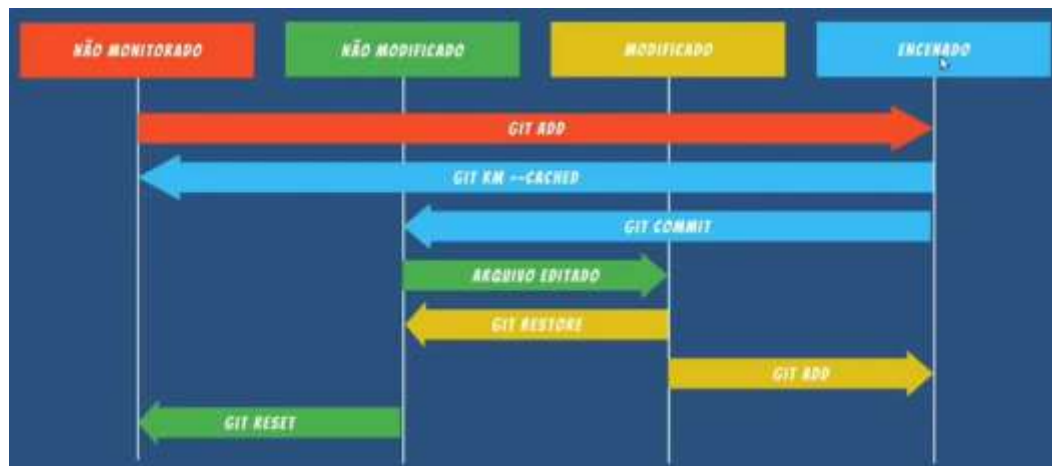
1. Diretório de trabalho;
2. Área de preparo;
3. Repositório local



Os quatro estados dos arquivos

Quando trabalhamos com Git, nossos arquivos podem ser classificados em quatro estados:

1. Untracked (não monitorado)
2. Unmodified (não modificado)
3. Modified (modificado)
4. Staged (Encenado)



O Hash SHA1

O Git utiliza o algoritmo de hash **sha1** para definir quando um arquivo foi modificado.

Dessa forma o Git é capaz de **identificar quais linhas do arquivo foram modificadas**, mantendo assim um histórico de versões para esse arquivo.

<http://www.sha1-online.com/>

Home Page | [SHA1 in JAVA](#) | [Secure password generator](#) | [Linux](#)

SHA1 and other hash functions online generator

▼

Result for sha1: 50623b414d557ef49fd90bb4576660d3179f7114

[SHA-1 MD5](#) on Wikipedia

Home Page | [SHA1 in JAVA](#) | [Secure password generator](#) | [Linux](#)

SHA1 and other hash functions online generator

▼

Result for sha1: f3fac12097840da84135d5a09eb9fb500b85cb1b

[SHA-1 MD5](#) on Wikipedia

SHA1 and other hash functions online generator

print "Olá mundo" hash

sha-1

Result for sha1: **50623b414d557ef49fd90bb4576660d3179f7114**

[SHA-1 MD5](#) on Wikipedia

Versão 1: 50623b414d557ef49fd90bb4576660d3179f7114

Versão 2: f3fac12097840da84135d5a09eb9fb500b85cb1b

Versão 3: 50623b414d557ef49fd90bb4576660d3179f7114

Aula 03 - Instalando o GIT no Windows

- Acesse o site e faça o download.

<https://git-scm.com/>



The image shows the Git website banner. At the top left is the Git logo (a red diamond with a white 'G' and a red arrow) followed by the text "git --distributed-is-the-new-centralized". To the right is a search bar with the placeholder text "Search entire site...". Below the header, there is a paragraph describing Git as a "free and open source distributed version control system" and another paragraph stating it is "easy to learn" and has a "tiny footprint with lightning fast performance". To the right of these paragraphs is a diagram showing a network of stacks of books connected by lines, representing a distributed system. At the bottom, there are four sections: "About" (advantages of Git), "Documentation" (command reference, Pro Git book, etc.), "Downloads" (GUI clients, binary releases), and "Community" (bug reporting, mailing list, etc.). On the right side of the bottom section is a monitor displaying the "Latest source Release 2.34.1" and a "Download for Windows" button.

git --distributed-is-the-new-centralized

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

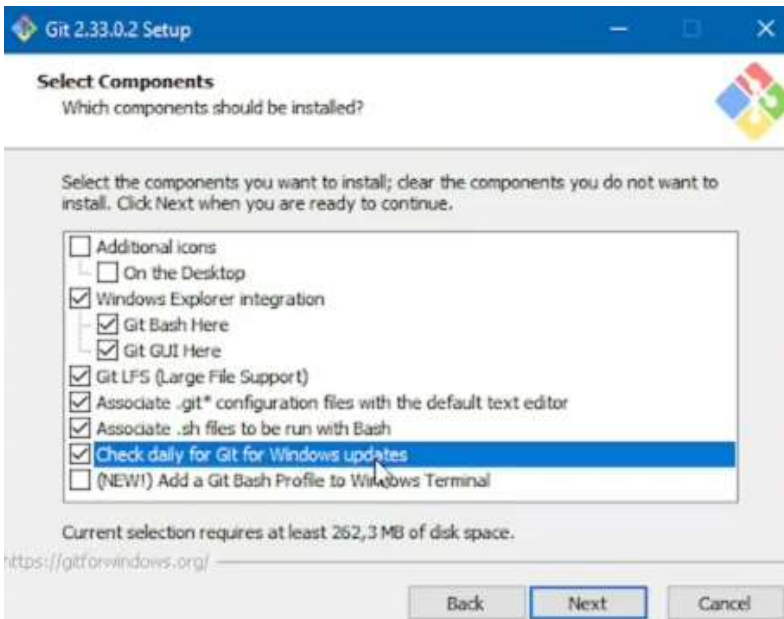
About
The advantages of Git compared to other source control systems.

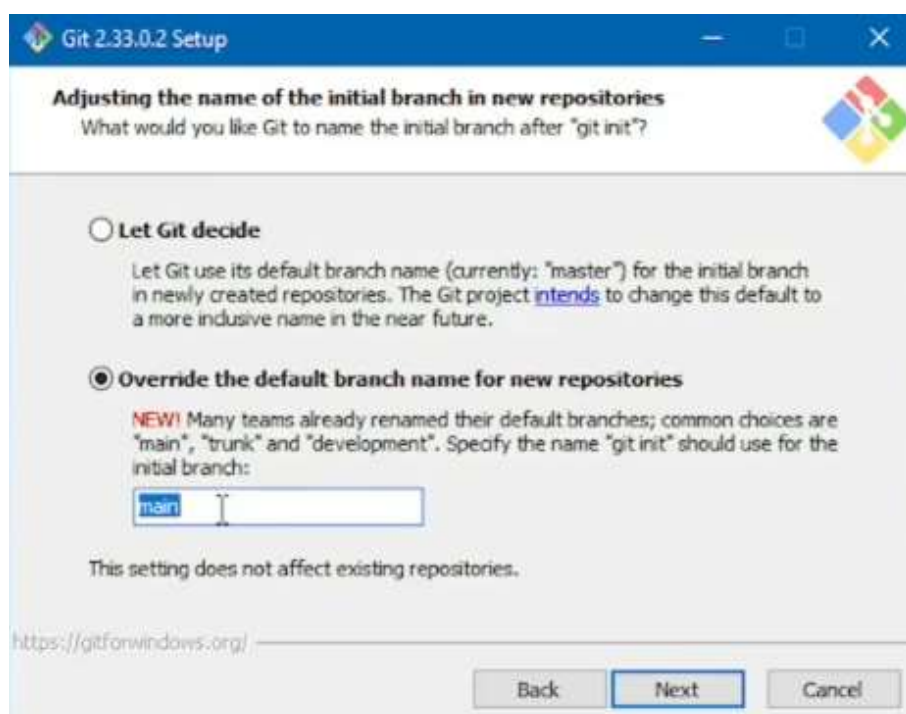
Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.34.1
[Release Notes \(2021-11-24\)](#)
[Download for Windows](#)





Configurando as variáveis de ambiente

```
git config --global user.name "Roberto Pinheiro"
```

```
git config --global email.name "betopinheiro1005@yahoo.com.br"
```

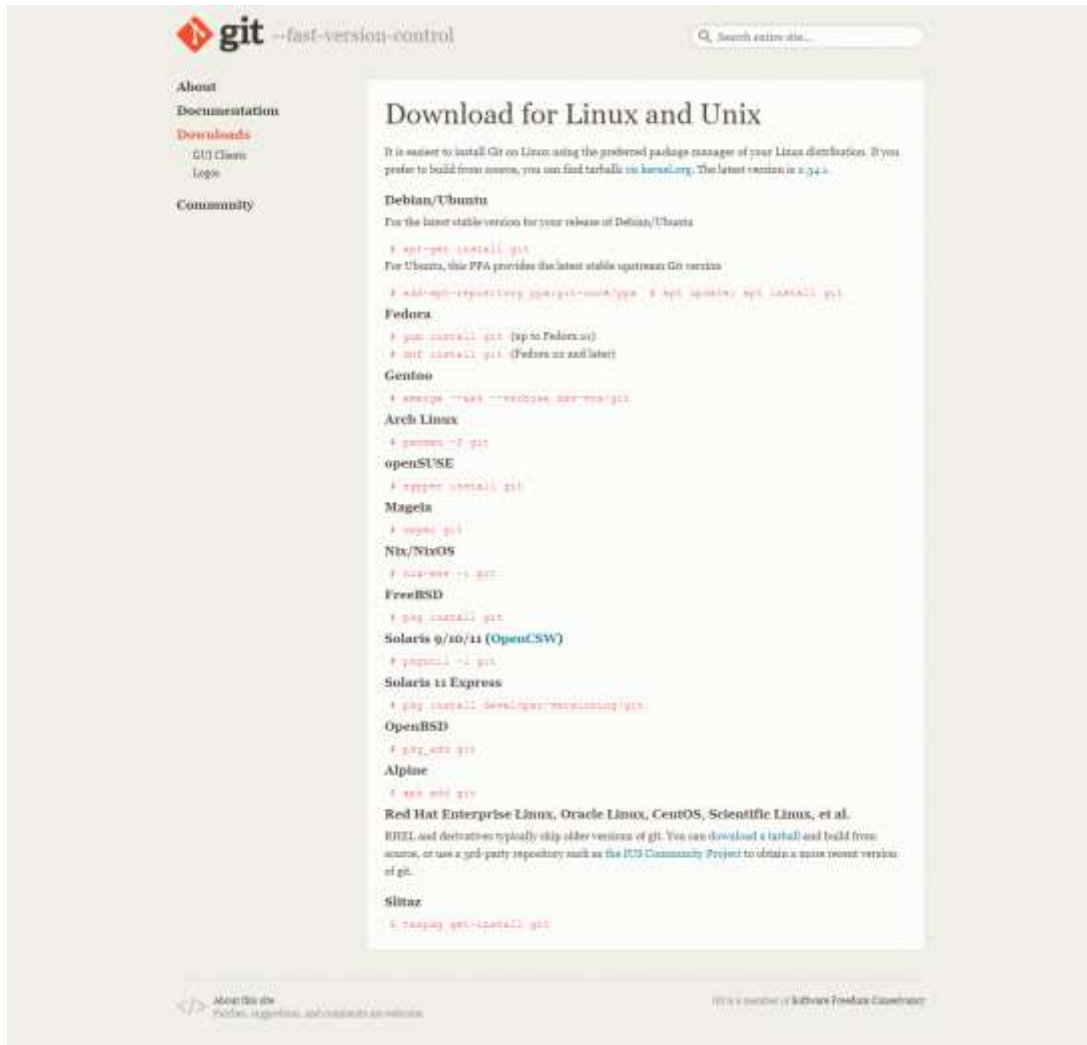
Checando se está configurado corretamente

```
git config --list
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /i/github
$ 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
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Roberto Pinheiro
user.email=betopinheiro1005@yahoo.com.br
```


Aula 04 - Instalando o GIT no Linux

<https://git-scm.com/download/linux>



Debian/Ubuntu

For Ubuntu, this PPA provides the latest stable upstream Git version

`add-apt-repository ppa:git-core/ppa`

```
root@anderson-desktop:/home/professorchoren# add-apt-repository ppa:git-core/ppa
Você está prestes a adicionar o seguinte PPA:
The most current stable version of Git for Ubuntu.

For release candidates, go to https://launchpad.net/~git-core/+archive/candidate .
Mais informações: https://launchpad.net/~git-core/+archive/ubuntu/ppa
Pressione Enter para continuar ou Ctrl+C para cancelar

Executing: /tmp/apt-key-gppghome.q8LiVY8lls/gpg.1.sh --keyserver hkps://keyserver.ubuntu.com:443 --recv-keys E1
DD270288B4E6030699E45FA1715D88E1DF1F24
gpg: chave A1715D88E1DF1F24: "Launchpad PPA for Ubuntu Git Maintainers" não mudada
gpg: Número total processado: 1
gpg:      não modificados: 1
root@anderson-desktop:/home/professorchoren#
```

apt update

```
root@anderson-desktop:/home/professorchoren# apt update
Ign:1 http://mirror.ufscar.br/mint-archive uma InRelease
Atingido:2 http://mirror.ufscar.br/mint-archive uma Release
Atingido:3 https://download.docker.com/linux/ubuntu focal InRelease
Atingido:5 https://brave-browser-apt-release.s3.brave.com stable InRelease
Atingido:6 https://typora.io/linux ./ InRelease
Atingido:7 http://ubuntu-archive.locaweb.com.br/ubuntu focal InRelease
Atingido:8 http://ubuntu-archive.locaweb.com.br/ubuntu focal-updates InRelease
Atingido:9 http://ubuntu-archive.locaweb.com.br/ubuntu focal-backports InRelease
Atingido:10 http://security.ubuntu.com/ubuntu focal-security InRelease
Atingido:11 http://archive.canonical.com/ubuntu focal InRelease
Atingido:12 http://ppa.launchpad.net/git-core/ppa/ubuntu focal InRelease
Lendo listas de pacotes... 9%
```

apt install git

```
root@anderson-desktop:/home/professorchoren# apt install git
Lendo listas de pacotes... Pronto
Construindo árvore de dependências
Lendo informação de estado... Pronto
git já é a versão mais recente (1:2.33.0-0ppa1-ubuntu20.04.1).
Os seguintes pacotes foram instalados automaticamente e já não são necessários:
  docker-ce-rootless-extras docker-scan-plugin linux-headers-5.4.0-81 linux-headers-5.4.0-81-generic
  linux-image-5.4.0-81-generic linux-modules-5.4.0-81-generic linux-modules-extra-5.4.0-81-generic pigz
  slirp4netns
Utilize 'sudo apt autoremove' para os remover.
0 pacotes atualizados, 0 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
root@anderson-desktop:/home/professorchoren# █ █
```

```
root@anderson-desktop:/home/professorchoren# git --version
git version 2.33.0
```

Configurando as variáveis de ambiente

```
professorchoren@anderson-desktop:~$ git config --global user.name "Professor Choren"
professorchoren@anderson-desktop:~$ git config --global user.email "professorchoren@gmail.com"
professorchoren@anderson-desktop:~$ git config --global core.editor "code --wait"
```

Aula 05 - Comandos básicos do GIT

- Crie a pasta `git_github_choren`

Listando os arquivos na pasta

ls

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren
$ ls
README.md  changelog.txt  logo_azul.png  logo_branco.png  logo_git.png  logo_github.png
```

Inicializando um repositório git

git init

```
professorchoren@anderson-desktop:~/Documentos/Curso de Git$ ls
changelog.txt  logo_azul.png  logo_branco.jpg  logo_github.png  logo_git.png  README.md
professorchoren@anderson-desktop:~/Documentos/Curso de Git$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Repositório vazio Git inicializado em /home/professorchoren/Documentos/Curso de Git/.git/
```

git config --global init.defaultBranch main

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren
$ git config --global init.defaultBranch main
```

git config --list

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren
$ 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
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Roberto Pinheiro
user.email=betopinheiro1005@yahoo.com.br
core.editor=code -w
init.defaultbranch=main
```

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren
$ git init
Initialized empty Git repository in C:/xampp/htdocs/git_github_choren/.git/
```

É criada uma pasta oculta chamada **.git**

git status

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md
        changelog.txt
        logo_azul.png
        logo_branco.png
        logo_git.png
        logo_github.png

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

Removendo o repositório git

rm -rf .git

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ rm -rf .git
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren
$ git status
fatal: not a git repository (or any of the parent directories): .git
```

Inicializando novamente o repositório git

git init

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren
$ git init
Initialized empty Git repository in C:/xampp/htdocs/git_github_choren/.git/
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md
        changelog.txt
        logo_azul.png
        logo_branco.png
        logo_git.png
        logo_github.png

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

- Os arquivos não estão sendo monitorados pelo Git.

Adicionando arquivos ao controle de versão

Adicionando um arquivo específico pelo nome

git add changelog.txt

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git add changelog.txt
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main

No commits yet

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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md
        logo_azul.png
        logo_branco.png
        logo_git.png
        logo_github.png
```

Adicionando arquivos com uma determinada extensão

git add *.png

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git add *.png
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   changelog.txt
        new file:   logo_azul.png
        new file:   logo_branco.png
        new file:   logo_git.png
        new file:   logo_github.png

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

Adicionando todos os arquivos

`git add --all`

ou

`git add .`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git add .
```

`git status`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
        new file:   changelog.txt
        new file:   logo_azul.png
        new file:   logo_branco.png
        new file:   logo_git.png
        new file:   logo_github.png
```

Aula 06 - Comando git commit

Configurando o editor padrão

- O editor padrão é o **vi**.
- Para configurar o **Visual Studio Code** como o editor padrão:

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

-w para aguardar o editor fechar.

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (master)
$ git config --global core.editor "code -w"
```

Convenção para commits

A estrutura dos commits é descrita por uma convenção

<https://www.conventionalcommits.org/pt-br/v1.0.0/>

A especificação do Conventional Commits é uma convenção simples para utilizar nas mensagens de commit. Ela define um conjunto de regras para criar um histórico de commit explícito, o que facilita a criação de ferramentas automatizadas baseadas na especificação. Esta convenção se encaixa com o SemVer, descrevendo os recursos, correções e modificações que quebram a compatibilidade nas mensagens de commit.

A mensagem do commit deve ser estruturada da seguinte forma:

<tipo>[escopo opcional]: <descrição>

[corpo opcional]

[rodapé(s) opcional(is)]

O commit contém os seguintes elementos estruturais, para comunicar a intenção ao utilizador da sua biblioteca:

fix: um commit do tipo fix soluciona um problema na sua base de código (isso se correlaciona com PATCH do versionamento semântico).

feat: um commit do tipo feat inclui um novo recurso na sua base de código (isso se correlaciona com MINOR do versionamento semântico).

BREAKING CHANGE: um commit que contém no rodapé opcional o texto BREAKING CHANGE:, ou contém o símbolo ! depois do tipo/escopo, introduz uma modificação que quebra a compatibilidade da API (isso se correlaciona com MAJOR do versionamento semântico). Uma BREAKING CHANGE pode fazer parte de commits de qualquer tipo.

Outros tipos adicionais são permitidos além de fix: e feat:, por exemplo @commitlint/config-conventional (baseado na Convenção do Angular) recomenda-se build:, chore:, ci:, docs:, style:, refactor:, perf:, test:, entre outros.

Outros rodapés diferentes de BREAKING CHANGE: <descrição> podem ser providos e seguem a convenção simular a git trailer format.

Observe que esses tipos adicionais não são exigidos pela especificação do Conventional Commits e não têm efeito implícito no versionamento semântico (a menos que incluam uma BREAKING CHANGE). Um escopo pode ser fornecido ao tipo do commit, para fornecer informações contextuais adicionais e está contido entre parênteses, por exemplo feat(parser): adiciona capacidade de interpretar arrays.

Versionamento semântico

<https://semver.org/lang/pt-BR/>

Versionamento Semântico 2.0.0

Dado um número de versão MAJOR.MINOR.PATCH, incremente a:

- versão Maior(MAJOR): quando fizer mudanças incompatíveis na API,
- versão Menor(MINOR): quando adicionar funcionalidades mantendo compatibilidade, e
- versão de Correção(PATCH): quando corrigir falhas mantendo compatibilidade.
- Rótulos adicionais para pré-lançamento(pre-release) e metadados de construção(build) estão disponíveis como extensão ao formato MAJOR.MINOR.PATCH.

Realizando o primeiro commit

git commit

```
COMMIT_EDITMSG .git X
.git > COMMIT_EDITMSG
1
2 # Please enter the commit message for your changes. Lines starting
3 # with '#' will be ignored, and an empty message aborts the commit.
4 #
5 # On branch main
6 #
7 # Initial commit
8 #
9 # Changes to be committed:
10 # new file:   README.md
11 # new file:   changelog.txt
12 # new file:   logo_azul.png
13 # new file:   logo_branco.png
14 # new file:   logo_git.png
15 # new file:   logo_github.png
16 #
```

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git commit
hint: Waiting for your editor to close the file... ..
```

.git\COMMIT_EDITMSG

feat: adicionar os recursos base do projeto

adição dos arquivos:

- README.md
- changelog.txt
- logo_azul.png
- logo_branco.png
- logo_git.png
- logo_github.png

- Nunca se deve utilizar . (ponto final) no commit

- Salve o arquivo e feche o editor.

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git commit
[main (root-commit) 756d36c] feat: adicionar os recursos base do projeto
6 files changed, 1 insertion(+)
create mode 100644 README.md
create mode 100644 changelog.txt
create mode 100644 logo_azul.png
create mode 100644 logo_branco.png
create mode 100644 logo_git.png
create mode 100644 logo_github.png
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
nothing to commit, working tree clean
```

Aula 07 - Comando git log

- Para saber o que já foi commitado:

git log

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log
commit 756d36c51c22cb3b65db35fa4a81dbf39e1d673c (HEAD -> main)
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date:   Wed Jan 5 23:00:24 2022 -0300

    feat: adicionar os recursos base do projeto

    adição dos arquivos:
        README.md
        changelog.txt
        logo_azul.png
        logo_branco.png
        logo_git.png
        logo_github.png
```

HEAD é a cabeça (ou ponteiro) do repositório, ou seja: para onde o Git está apontando naquele momento.

O HEAD é o apontamento do Git para um commit, ou seja: que commit atualmente eu estou trabalhando.

- Altere o conteúdo do arquivo **changelog.txt**

changelog.txt

05/01/2022 - Aprendendo a usar o comando git log.

Adicionando a modificação e realizando o commit

git commit -a

```
COMMIT_EDITMSG .git X
.git > COMMIT_EDITMSG
1
2 # Please enter the commit message for your changes. Lines starting
3 # with '#' will be ignored, and an empty message aborts the commit.
4 #
5 # On branch main
6 # Changes to be committed:
7 #   modified:   changelog.txt
8 #
```

.git\COMMIT_EDITMSG

docs: agregar informação ao log de modificações

Modificações dos arquivos:

modified: changelog.txt

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git commit -a
[main 3dc0965] docs: agregar informação ao log de modificações
1 file changed, 1 insertion(+), 1 deletion(-)
```

git log

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log
commit 3dc0965bbc67aace818125c7783f8dc3d5428a15 (HEAD -> main)
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date: Wed Jan 5 23:11:07 2022 -0300

    docs: agregar informação ao log de modificações

    Modificações dos arquivos:
        modified: changelog.txt

commit 756d36c51c22cb3b65db35fa4a81dbf39e1d673c
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date: Wed Jan 5 23:00:24 2022 -0300

    feat: adicionar os recursos base do projeto

    adição dos arquivos:
        README.md
        changelog.txt
        logo_azul.png
        logo_branco.png
        logo_git.png
        logo_github.png
```

- Observe que a cabeça do repositório (HEAD) foi movida para o novo commit.

Exibição de commits mais detalhada

git log -p

```
commit 3dc0965bbc67aace818125c7783f8dc3d5428a15 (HEAD -> main)
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date:   Wed Jan 5 23:11:07 2022 -0300

    docs: agregar informação ao log de modificações

    Modificações dos arquivos:
        modified:   changelog.txt

diff --git a/changelog.txt b/changelog.txt
index b88c44a..2742d8a 100644
--- a/changelog.txt
+++ b/changelog.txt
@@ -1,1 @@
-17/10/2021 - Aprendendo a usar o comando git log.
\ No newline at end of file
+05/01/2022 - Aprendendo a usar o comando git log.
\ No newline at end of file

commit 756d36c51c22cb3b65db35fa4a81dbf39e1d673c
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date:   Wed Jan 5 23:00:24 2022 -0300

    feat: adicionar os recursos base do projeto

    adição dos arquivos:
        README.md
        changelog.txt
        logo_azul.png
        logo_branco.png
        logo_git.png
        logo_github.png

diff --git a/README.md b/README.md
new file mode 100644
index 0000000..e69de29
diff --git a/changelog.txt b/changelog.txt
new file mode 100644
index 0000000..b88c44a
--- /dev/null
+++ b/changelog.txt
@@ -0,0 +1 @@
+17/10/2021 - Aprendendo a usar o comando git log.
\ No newline at end of file
diff --git a/logo_azul.png b/logo_azul.png
new file mode 100644
index 0000000..ba07d9e
Binary files /dev/null and b/logo_azul.png differ
:|
```

- Pressione a tecla <q> para sair.

Filtrando logs

- Para apresentar apenas o último log:

`git log -1`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log -1
commit 3dc0965bbc67aace818125c7783f8dc3d5428a15 (HEAD -> main)
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date:   Wed Jan 5 23:11:07 2022 -0300

    docs: agregar informação ao log de modificações

Modificações dos arquivos:
    modified:   changelog.txt
```

- Conta de trás para frente.

Log com informação resumida

`git log --oneline`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
3dc0965 (HEAD -> main) docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

- Apresenta os sete primeiros caracteres do hash.

Filtro resumido do último commit

`git log --oneline -1`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline -1
3dc0965 (HEAD -> main) docs: agregar informação ao log de modificações
```

Aula 08 - Comando git rm

O comando `git rm` basicamente é uma forma de remover um arquivo do controle de versão.

```
git rm --cached *.png
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git rm --cached *.png
rm 'logo_azul.png'
rm 'logo_branco.png'
rm 'logo_git.png'
rm 'logo_github.png'
```

ls

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ls
README.md  changelog.txt  logo_azul.png  logo_branco.png  logo_git.png  logo_github.png
```

- Perceba que os arquivos não foram removidos do diretório, apenas do controle de versão.

```
git status
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    logo_azul.png
        deleted:    logo_branco.png
        deleted:    logo_git.png
        deleted:    logo_github.png

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        logo_azul.png
        logo_branco.png
        logo_git.png
        logo_github.png
```

Criando um novo arquivo de texto

```
echo > remover.txt
```

ls

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ls
README.md  changelog.txt  logo_azul.png  logo_branco.png  logo_git.png  logo_github.png  remover.txt
```


git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    logo_azul.png
    deleted:    logo_branco.png
    deleted:    logo_git.png
    deleted:    logo_github.png

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logo_azul.png
    logo_branco.png
    logo_git.png
    logo_github.png
    remover.txt
```

git add remover.txt

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git add remover.txt
warning: LF will be replaced by CRLF in remover.txt.
The file will have its original line endings in your working directory
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    logo_azul.png
    deleted:    logo_branco.png
    deleted:    logo_git.png
    deleted:    logo_github.png
    new file:   remover.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logo_azul.png
    logo_branco.png
    logo_git.png
    logo_github.png
```

git commit

.git\COMMIT_EDITMSG

feat: adicionar o arquivo de teste remover.txt

arquivo adicionado:

new file: remover.txt

```
betoi@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git commit
[main d2f362e] feat: adicionar o arquivo de teste remover.txt
5 files changed, 1 insertion(+)
delete mode 100644 logo_azul.png
delete mode 100644 logo_branco.png
delete mode 100644 logo_git.png
delete mode 100644 logo_github.png
create mode 100644 remover.txt
```

git log

```
betoi@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log
commit d2f362ed35992b3017585162062d0cddc8c0d479 (HEAD -> main)
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date: Wed Jan 5 23:54:19 2022 -0300

    feat: adicionar o arquivo de teste remover.txt

    arquivo adicionado:
        new file: remover.txt

commit 3dc0965bbc67aace818125c7783f8dc3d5428a15
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date: Wed Jan 5 23:11:07 2022 -0300

    docs: agregar informação ao log de modificações

    Modificações dos arquivos:
        modified: changelog.txt

commit 756d36c51c22cb3b65db35fa4a81dbf39e1d673c
Author: Roberto Pinheiro <betopinheiro1005@yahoo.com.br>
Date: Wed Jan 5 23:00:24 2022 -0300

    feat: adicionar os recursos base do projeto

    adição dos arquivos:
        README.md
        changelog.txt
        logo_azul.png
        logo_branco.png
        logo_git.png
        logo_github.png
```

ls

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ls
README.md  changelog.txt  logo_azul.png  logo_branco.png  logo_git.png  logo_github.png  remover.txt
```

git rm remover.txt

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git rm remover.txt
rm 'remover.txt'
```

Sem o `--cached` na linha de comandos acima, o arquivo é removido do controle de versão e da pasta de arquivos.

ls

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ls
README.md  changelog.txt  logo_azul.png  logo_branco.png  logo_git.png  logo_github.png
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    remover.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        logo_azul.png
        logo_branco.png
        logo_git.png
        logo_github.png
```

Aula 09 - Comando git diff

README.txt

Curso de Git e GitHub

1. Controle de versão
2. Conhecendo o Git
3. Instalando o Git CLI no Windows
4. Instalando o Git CLI no Linux
5. Comandos básicos do Git
6. Comando git commit
7. Comando git log
8. Comando git rm
9. Comando git diff

git status

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    remover.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        logo_azul.png
        logo_branco.png
        logo_git.png
        logo_github.png
```

Mudanças em relação ao último commit

git diff

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git diff
diff --git a/README.md b/README.md
index e69de29..b3b55e9 100644
--- a/README.md
+++ b/README.md
@@ -0,0 +1,11 @@
+# Curso de Git e GitHub
+
+1. Controle de versão
+2. Conhecendo o Git
+3. Instalando o Git CLI no Windows
+4. Instalando o Git CLI no Linux
+5. Comandos básicos do Git
+6. Comando git commit
+7. Comando git log
+8. Comando git rm
+9. Comando git diff
```

Comparação entre commits

git log --oneline

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
d2f362e (HEAD -> main) feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

git diff d2f362e 756d36c

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git diff d2f362e 756d36c
diff --git a/changelog.txt b/changelog.txt
index 2742d8a..b88c44a 100644
--- a/changelog.txt
+++ b/changelog.txt
@@ -1,1 @@
-05/01/2022 - Aprendendo a usar o comando git log.
\ No newline at end of file
+17/10/2021 - Aprendendo a usar o comando git log.
\ No newline at end of file
diff --git a/logo_azul.png b/logo_azul.png
new file mode 100644
index 0000000..ba07d9e
Binary files /dev/null and b/logo_azul.png differ
diff --git a/logo_branco.png b/logo_branco.png
new file mode 100644
index 0000000..8713b37
Binary files /dev/null and b/logo_branco.png differ
diff --git a/logo_git.png b/logo_git.png
new file mode 100644
index 0000000..51f4ae5
Binary files /dev/null and b/logo_git.png differ
diff --git a/logo_github.png b/logo_github.png
new file mode 100644
index 0000000..9c9e6f1
Binary files /dev/null and b/logo_github.png differ
diff --git a/remover.txt b/remover.txt
deleted file mode 100644
index 8b13789..0000000
--- a/remover.txt
+++ /dev/null
@@ -1,0 @@
```

Diferença da versão atual para o último commit

git diff d2f362e

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git diff d2f362e
diff --git a/README.md b/README.md
index e69de29..b3b55e9 100644
--- a/README.md
+++ b/README.md
@@ -0,0 +1,11 @@
+# Curso de Git e GitHub
+
++1. Controle de versão
++2. Conhecendo o Git
++3. Instalando o Git CLI no Windows
++4. Instalando o Git CLI no Linux
++5. Comandos básicos do Git
++6. Comando git commit
++7. Comando git log
++8. Comando git rm
++9. Comando git diff
diff --git a/remover.txt b/remover.txt
deleted file mode 100644
index 8b13789..0000000
--- a/remover.txt
+++ /dev/null
@@ -1 +0,0 @@
```

Aula 10 - O arquivo .gitignore

.gitignore é um arquivo onde é declarado quais são os arquivos ou pastas que devem ser ignorados no monitoramento do versionamento do código.

Criando o arquivo .gitignore

```
touch .gitignore
```

.gitignore

```
# Ignorando um único arquivo  
logo_branco.png
```

```
git status
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   remover.txt

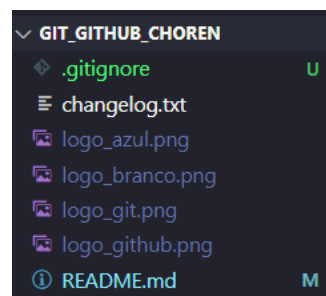
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    logo_azul.png
    logo_git.png
    logo_github.png
```

.gitignore

```
# Ignorando um único arquivo  
logo_branco.png
```

```
# Ignorando arquivos de uma extensão pré-definida  
*.png
```



git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   remover.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

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

git add .

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git add .

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore
    modified:   README.md
    deleted:    remover.txt
```

git commit

.git\COMMIT_EDITMSG

feat: adicionar o arquivo .gitignore ao sistema

arquivos modificados:

- adicionado: .gitignore
- modificado: README.md
- removido: remover.txt

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git commit
[main 2f618ea] feat: adicionar o arquivo .gitignore ao sistema
3 files changed, 21 insertions(+), 1 deletion(-)
create mode 100644 .gitignore
delete mode 100644 remover.txt
```


git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
nothing to commit, working tree clean
```

Aula 11 - Comandos git restore e git reset

Ambos comandos são utilizados para desfazer uma alteração feita no repositório.

- Crie um arquivo vazio chamado: `logs_06_01_2022.txt`:

```
touch logs_06_01_2022.txt
```

- Altere o conteúdo do arquivo `changelog.txt`

`changelog.txt`

06/01/2022 - Aprendendo a usar os comandos git restore e git reset.

`git status`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   changelog.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        logs_06_01_2022.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

`changelog.txt`

05/01/2022 - Aprendendo a usar o comando git log.

06/01/2022 - Aprendendo a usar os comandos git restore e git reset.

`git status`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   changelog.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        logs_06_01_2022.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

git add .

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git add .

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   changelog.txt
        new file:   logs_06_01_2022.txt
```

Desfazendo alterações ainda não commitadas

- Não era para ter adicionado o arquivo logs_06_01_2022.txt.

- Para desfazer as alterações que estão no staging area preparadas para serem commitadas, entre com:

git restore --staged logs_06_01_2022.txt

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   changelog.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        logs_06_01_2022.txt
```

Desfazendo alterações que já foram commitadas

Realizando o commit

git commit -m "Reverter esse commit"

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git commit -m "Reverter esse commit"
[main 8da0637] Reverter esse commit
1 file changed, 2 insertions(+), 1 deletion(-)
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt

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

git log --oneline

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
8da0637 (HEAD -> main) Reverter esse commit
2f618ea feat: adicionar o arquivo .gitignore ao sistema
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

Modos de reversão



--soft

Desfaz o commit mas não descarta nada da área de preparo ou do diretório de trabalho.

--mixed

É o valor padrão caso não se coloque nenhum desses três parâmetros.

Vai desfazer o commit e manter os arquivos no staging area (não desfaz o git add, desfaz apenas o git commit). Este comando não afeta o diretório de trabalho.

--hard

Desfaz tudo. Qualquer trabalho que não tenha sido commitado é desfeito. Volta para como era no commit anterior ao qual eu estou desfazendo.

Exemplos:

git add .

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git add .

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   logs_06_01_2022.txt
```

git commit -m "Adicionar o log"

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git commit -m "Adicionar o log"
[main 6434035] Adicionar o log
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 logs_06_01_2022.txt
```

git log --oneline

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
6434035 (HEAD -> main) Adicionar o log
8da0637 Reverter esse commit
2f618ea feat: adicionar o arquivo .gitignore ao sistema
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
nothing to commit, working tree clean
```

Usando --soft

A partir do ponteiro voltar 1 commit:

```
git reset --soft HEAD~1
```

```
git status
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git reset --soft HEAD~1

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   logs_06_01_2022.txt
```

- Commit desfeito, porém o arquivo está monitorado (na staging area), pronto para ser commitado.

```
git log --oneline
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
8da0637 (HEAD -> main) Reverter esse commit
2f618ea feat: adicionar o arquivo .gitignore ao sistema
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

Refazendo o commit:

```
git commit -m "Adicionar o log"
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git commit -m "Adicionar o log"
[main 3fc8ae5] Adicionar o log
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 logs_06_01_2022.txt
```

```
git log --oneline
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
3fc8ae5 (HEAD -> main) Adicionar o log
8da0637 Reverter esse commit
2f618ea feat: adicionar o arquivo .gitignore ao sistema
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

Usando --mixed

git reset --mixed HEAD~1

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git reset --mixed HEAD~1
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt

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

- Agora o arquivo está não monitorado (não se encontra mais na staging area).

git log --oneline

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
8da0637 (HEAD -> main) Reverter esse commit
2f618ea feat: adicionar o arquivo .gitignore ao sistema
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

Usando --hard

git reset --hard HEAD~1

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git reset --hard HEAD~1
HEAD is now at 2f618ea feat: adicionar o arquivo .gitignore ao sistema
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt
```

git log --oneline

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
2f618ea (HEAD -> main) feat: adicionar o arquivo .gitignore ao sistema
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

Revertendo o commit a partir de um ponto específico

git reset d2f362e

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git reset d2f362e
Unstaged changes after reset:
M   README.md
D   remover.txt
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md
        deleted:    remover.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        logs_06_01_2022.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

git add README.md .gitignore

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git add README.md .gitignore

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore
        modified:   README.md

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    remover.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        logs_06_01_2022.txt
```


git rm remover.txt

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git rm remover.txt
rm 'remover.txt'

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore
    modified:   README.md
    deleted:    remover.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt
```

git commit -m "Reversão do estado antes dos comandos git restore e git reset"

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git commit -m "Reversão do estado antes dos comandos git restore e git reset"
[main 4d60e60] Reversão do estado antes dos comandos git restore e git reset
3 files changed, 21 insertions(+), 1 deletion(-)
create mode 100644 .gitignore
delete mode 100644 remover.txt
```

git status

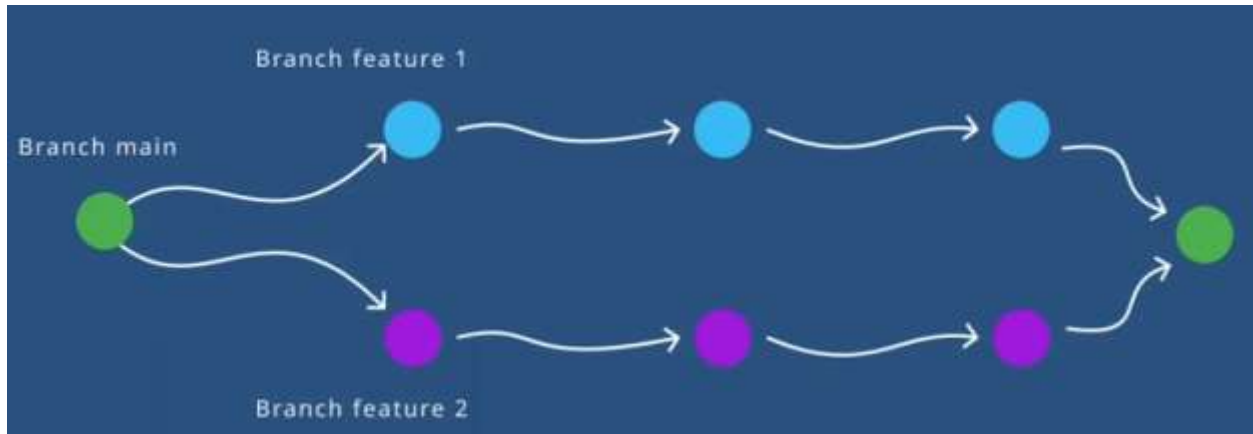
```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt

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

Aula 12 - Comando git checkout

O comando `git checkout` permite alternar entre as branches.

Uma **branch** é um ramo dentro do repositório.



- O que acontece nos ramos secundários não afeta o ramo principal.

`git branch`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git branch
* main
```

Criando uma nova branch

`git checkout -b feature_1`

-b → cria a branch, caso ela não exista.

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git checkout -b feature_1
Switched to a new branch 'feature_1'
```

`git branch`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ git branch
* feature_1
  main
```

- A branch foi criada e HEAD aponta para ela.

`touch feature_1.txt`

`git status`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ touch feature_1.txt

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ git status
On branch feature_1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    feature_1.txt
    logs_06_01_2022.txt

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

`git add feature_1.txt`

`git status`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ git add feature_1.txt

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ git status
On branch feature_1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   feature_1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt
```

`git stash save "Trabalhando na feature 1"`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ git stash save "Trabalhando na feature 1"
Saved working directory and index state On feature_1: Trabalhando na feature 1
```

- Dessa forma, é salvo o estado com um nome.

Voltando para a branch main

git checkout main

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ git checkout main
Switched to branch 'main'
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt

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

- Perceba que aquela pendência não veio junto.

Criando uma nova branch

git checkout -b feature_2

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git checkout -b feature_2
Switched to a new branch 'feature_2'
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_2)
$ git status
On branch feature_2
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    feature_2.txt
    logs_06_01_2022.txt

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

git add feature_2.txt

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_2)
$ git add feature_2.txt

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_2)
$ git status
On branch feature_2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   feature_2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt
```

git stash save "Trabalhando na feature 2"

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_2)
$ git stash save "Trabalhando na feature 2"
Saved working directory and index state On feature_2: Trabalhando na feature 2
```

Voltando para a branch main

git checkout main

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt

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

Exibindo quais são as pendências

git stash list

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git stash list
stash@{0}: On feature_2: Trabalhando na feature 2
stash@{1}: On feature_1: Trabalhando na feature 1
```

Desempilhando a feature_1

git checkout feature_1

git stash pop stash@{1}

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ git stash pop stash@{1}
On branch feature_1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   feature_1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt

Dropped stash@{1} (fa132df93f2bd3f8a076ea8ec7cbb25c31ae243b)
```

git commit

.git\COMMIT_EDITMSG

feat: agregará uma nova funcionalidade ao sistema

No ramo feature_1

Mudanças submetidas:

feature_1.txt

- Salve e feche o arquivo.

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ git commit
[feature_1 63ac638] feat: agregará uma nova funcionalidade ao sistema
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 feature_1.txt
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ git status
On branch feature_1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt

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

Alternando para a branch feature_2

git checkout feature_2

git stash pop stash@{0}

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_2)
$ git stash pop stash@{0}
On branch feature_2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   feature_2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt

Dropped stash@{0} (b0d3ef5e08d1a2eccc2dc3d5619fa3a699477bd8)
```

git branch

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_2)
$ git branch
  feature_1
* feature_2
  main
```

git commit

.git\COMMIT_EDITMSG

feat: agregará uma nova funcionalidade ao sistema

No ramo feature_2

Mudanças submetidas:

feature_2.txt

- Salve e feche o arquivo.

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_2)
$ git commit
[feature_2 ddc99d1] feat: agregará uma nova funcionalidade ao sistema
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 feature_2.txt
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_2)
$ git status
On branch feature_2
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    logs_06_01_2022.txt

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

git log --oneline

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_2)
$ git log --oneline
ddc99d1 (HEAD -> feature_2) feat: agregará uma nova funcionalidade ao sistema
4d60e60 (main) Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```


git checkout main

git log --oneline

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
4d60e60 (HEAD -> main) Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

Aula 13 - Comando git merge

`git merge` serve para unir o conteúdo de uma branch com alguma outra branch.

Unindo as branches `feature_1` e `feature_2` ao ramo principal (`main`)

`git branch`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git branch
  feature_1
  feature_2
* main
```

`git log --oneline`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
4d60e60 (HEAD -> main) Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

`ls`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ls
README.md  changelog.txt  logo_azul.png  logo_branco.png  logo_git.png  logo_github.png  logs_06_01_2022.txt
```

- Os arquivos criados nas branches `feature_1` e `feature_2` não existem na branch `main`.

`git checkout feature_1`

`ls`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git checkout feature_1
Switched to branch 'feature_1'

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ ls
README.md  changelog.txt  feature_1.txt  logo_azul.png  logo_branco.png  logo_git.png  logo_github.png  logs_06_01_2022.txt
```

Voltando para a branch main

git checkout main

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (feature_1)
$ git checkout main
Switched to branch 'main'

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$
```

Unindo a branch feature_1 ao ramo principal (branch main)

git merge feature_1

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git merge feature_1
Updating 4d60e60..63ac638
Fast-forward
 feature_1.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature_1.txt
```

ls

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ls
README.md  changelog.txt  feature_1.txt  logo_azul.png  logo_branco.png  logo_git.png  logo_github.png  logs_06_01_2022.txt
```

Agora temos o arquivo **feature_1.txt** na branch **main**.

git log --oneline

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
63ac638 (HEAD -> main, feature_1) feat: agregará uma nova funcionalidade ao sistema
4d60e60 Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

Unindo a branch feature_2 ao ramo principal (branch main)

git merge feature_2

ls

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git merge feature_2
Merge made by the 'recursive' strategy.
 feature_2.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature_2.txt

betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ls
README.md  changelog.txt  Feature_1.txt  Feature_2.txt  Logo_azul.png  Logo_branco.png  Logo_git.png  Logo_github.png  Logs_06_01_2022.txt
```

Agora temos o arquivo **feature_2.txt** na branch **main**.

git log --oneline

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
963c4bb (HEAD -> main) Merge branch 'feature_2'
ddc99d1 (feature_2) feat: agregará uma nova funcionalidade ao sistema
63ac638 (feature_1) feat: agregará uma nova funcionalidade ao sistema
4d60e60 Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

git branch

```
betol@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git branch
  feature_1
  feature_2
* main
```

Deletando as branches feature_1 e feature_2

`git branch -d feature_1 feature_2`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git branch -d feature_1 feature_2
Deleted branch feature_1 (was 63ac638).
Deleted branch feature_2 (was ddc99d1).
```

`git branch`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git branch
* main
```

`git log --oneline`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
963c4bb (HEAD -> main) Merge branch 'feature_2'
ddc99d1 feat: agregará uma nova funcionalidade ao sistema
63ac638 feat: agregará uma nova funcionalidade ao sistema
4d60e60 Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

- **Importante:** Para evitar problemas, recomenda-se que após fazer o merge do branch main com o branch criado, o branch criado seja sempre deletado.

Aula 14 - Comando git tag

Tags são forma de definir marcadores no nosso repositório onde a versão do projeto pode ser utilizável.

Criando uma tag

```
git tag -a v1.0.0-beta -m "Versão 1.0.0 beta 1"
```

Exibindo as tags criadas

```
git tag
```

```
betoi@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git tag
v1.0.0-beta
```

Importante: a tag é criada a partir do commit que está na cabeça (HEAD) do repositório.

```
git log --oneline
```

```
betoi@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
963c4bb (HEAD -> main, tag: v1.0.0-beta) Merge branch 'feature_2'
ddc99d1 feat: agregará uma nova funcionalidade ao sistema
63ac638 feat: agregará uma nova funcionalidade ao sistema
4d60e60 Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

Adicionando uma tag para uma determinada funcionalidade

Funcionalidade 1 → hash resumido 63ac638

Funcionalidade 2 → hash resumido ddc99d1

```
git tag -a v0.1.0 -m "Funcionalidade 1" 63ac638
```

```
git log --oneline
```

```
betoi@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
963c4bb (HEAD -> main, tag: v1.0.0-beta) Merge branch 'feature_2'
ddc99d1 feat: agregará uma nova funcionalidade ao sistema
63ac638 (tag: v0.1.0) feat: agregará uma nova funcionalidade ao sistema
4d60e60 Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

```
git tag -a v0.2.0 -m "Funcionalidade 2" ddc99d1
git log --oneline
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git tag -a v0.2.0 -m "Funcionalidade 2" ddc99d1

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
963c4bb (HEAD -> main, tag: v1.0.0-beta) Merge branch 'feature_2'
ddc99d1 (tag: v0.2.0) feat: agregará uma nova funcionalidade ao sistema
63ac638 (tag: v0.1.0) feat: agregará uma nova funcionalidade ao sistema
4d60e60 Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

Exibindo as tags criadas

```
git tag
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git tag
v0.1.0
v0.2.0
v1.0.0-beta
```

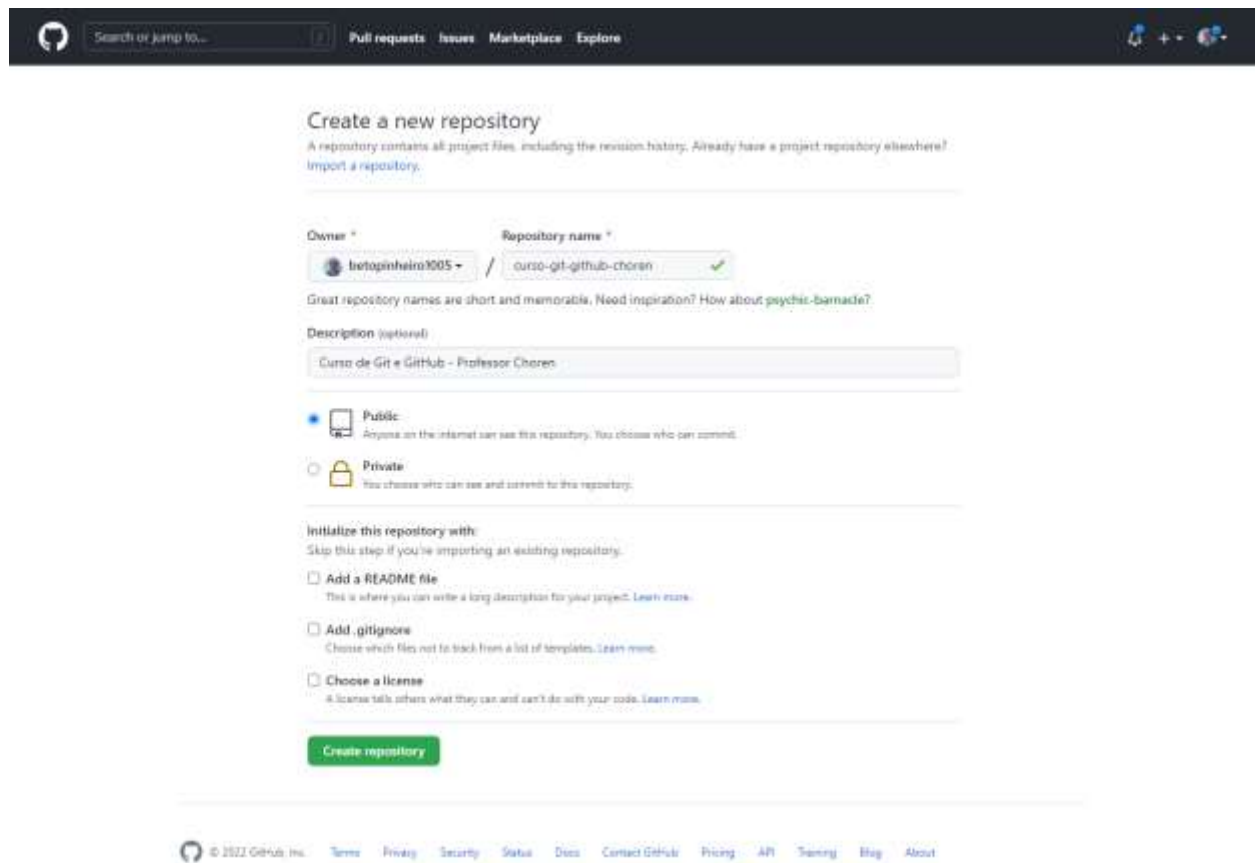
Aula 15 - Conhecendo o GitHub

Ferramentas de versionamento de código

1. GitHub
2. GitLab
3. Bitbucket

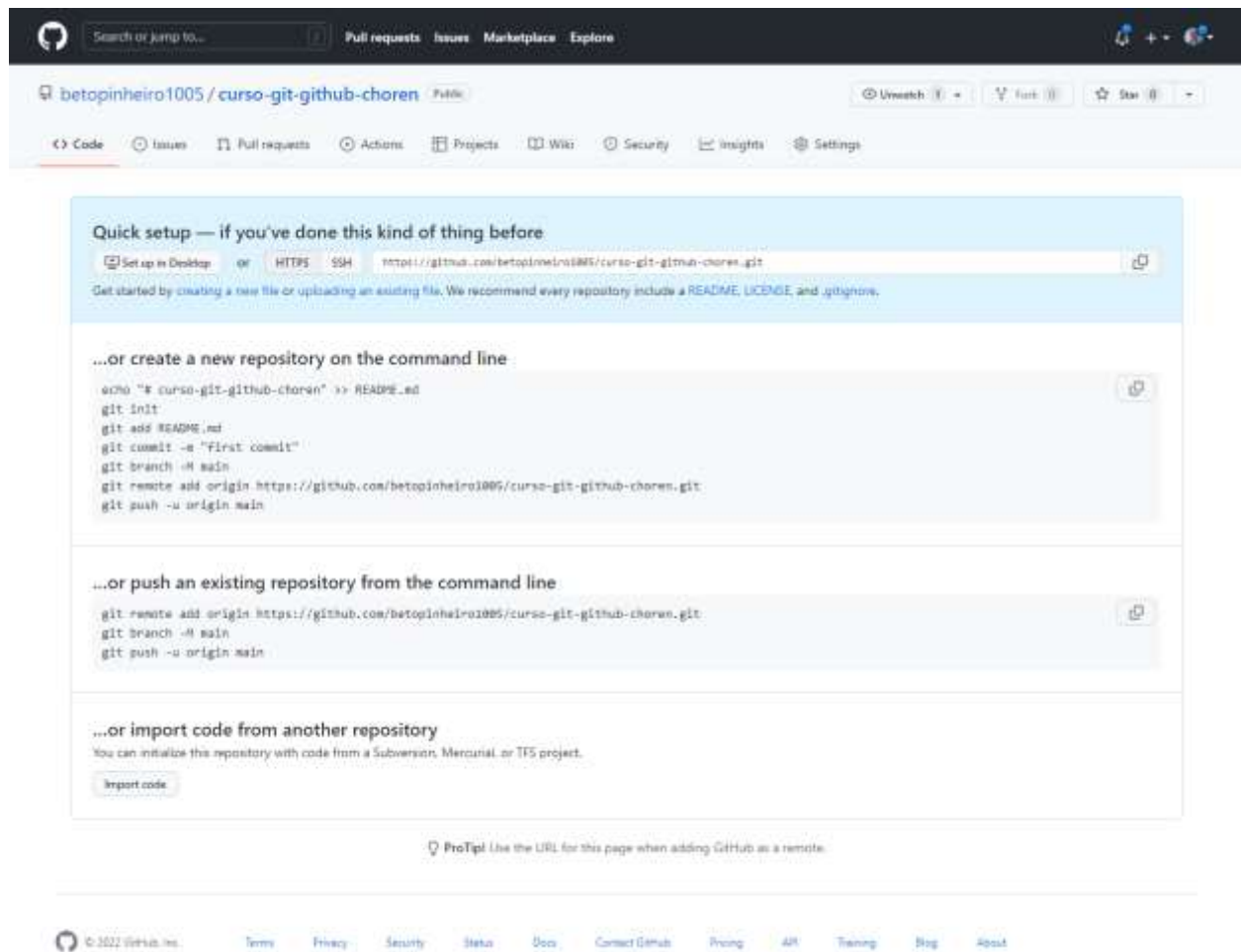
- Crie uma conta no GitHub ([Sign up](#)) e logue nela ([Sign in](#)).

Criando um novo repositório



The screenshot shows the GitHub 'Create a new repository' page. At the top, there's a dark navigation bar with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this, the main heading is 'Create a new repository'. A subtext says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. The form has two input fields: 'Owner' with a dropdown menu showing 'betopinheiro1005' and 'Repository name' with a text input showing 'curso-git-github-choran'. Below these, a note says 'Great repository names are short and memorable. Need inspiration? How about [psychic-barnacle?](#)'. The 'Description (optional)' field contains 'Curso de Git e GitHub - Professor Choren'. There are two radio buttons for visibility: 'Public' (selected) and 'Private'. Under 'Initialize this repository with:', there are three checkboxes: 'Add a README file', 'Add .gitignore', and 'Choose a license'. At the bottom, there is a green 'Create repository' button. The footer contains the GitHub logo, copyright information, and various links like 'Terms', 'Privacy', 'Security', 'Status', 'Docs', 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.

- Clique no botão "[Create repository](#)"



- Inicialmente, remova o arquivo **logs_06_01_2022.txt**:

```
rm logs_06_01_2022.txt
```

```
git status
```

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ rm logs_06_01_2022.txt

beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git status
On branch main
nothing to commit, working tree clean
```

- Para enviar para o GitHub não pode haver nenhuma pendência, portanto antes sempre confira com o comando **git status**

Adicionando o endereço remoto

git remote add origin https://github.com/betopinheiro1005/curso-git-github-choren.git

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git remote add origin https://github.com/betopinheiro1005/curso-git-github-choren.git
```

git remote

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git remote
origin
```

Gerando uma chave SSH

ssh-keygen -o

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ssh-keygen -o
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/beto1/.ssh/id_rsa): .....
```

- Pressione a tecla <Enter>

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ssh-keygen -o
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/beto1/.ssh/id_rsa):
/c/Users/beto1/.ssh/id_rsa already exists.
Overwrite (y/n)? .....
```

- Digite **y** e pressione a tecla <Enter>

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ssh-keygen -o
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/beto1/.ssh/id_rsa):
/c/Users/beto1/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase): .....
```

Essa é a senha que será usada para enviar os arquivos.

```





beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ ssh-keygen -o
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/beto1/.ssh/id_rsa):
/c/Users/beto1/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/beto1/.ssh/id_rsa
Your public key has been saved in /c/Users/beto1/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:wYVuGEIkSq8K2NZdFwD2fu3un8rC1leStGeuAAyd1/E beto1@DESKTOP-HFTIOCI
The key's randomart image is:
+---[RSA 3072]-----+
| ..oo +=.+o... |
| .....o*..... |
| . . . +=o. E |
| .... ..ooo.. + |
| o.o . ..S ...+ + |
| o. . . . = . |
| . . . . . o |
| . . . . . +o. .. |
| . . . . . o=+o |
+-----[SHA256]-----+

```

Your identification has been saved in [/c/Users/beto1/.ssh/id_rsa](#)
 Your public key has been saved in [/c/Users/beto1/.ssh/id_rsa.pub](#)

A chave pública e privada foram criadas na pasta "[/c/Users/beto1/.ssh/](#)"

Vá até essa pasta e abra o arquivo: [id_rsa.pub](#)

<div> <div></div> <div>Este Computador > W10-161 (C:) > Usuários > beto1 > .ssh</div> </div>				
Nome	Data de modificação	Tipo	Tamanho	
 id_rsa	06/01/2022 14:22	Arquivo	3 KB	
 id_rsa.pub	06/01/2022 14:22	Arquivo PUB	1 KB	
 known_hosts	20/09/2021 03:52	Arquivo	1 KB	
 known_hosts.old	15/08/2021 11:48	Arquivo OLD	1 KB	

- Copie todo o conteúdo dessa chave.
- No GitHub, clique na sua conta em "[Settings](#)"
- Clique na opção "[SSH and GPC keys](#)".
- Delete a chave existente.

SSH keys

[New SSH key](#)

There are no SSH keys associated with your account.

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

- Clique no botão "**New SSH Key**"

SSH keys / Add new

Title

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](#)

- Cole a chave copiada anteriormente e dê um título para essa chave.
- Em seguida clique no botão **Add SSH key**

SSH keys

[New SSH key](#)

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



SSH

Roberto Pinheiro - Chave do meu notebook

SHA256:wYVvGEIkSq8K2NZdFwD2fu3un8rC11eStGEuAAyd1/E

Added on 6 Jan 2022

Never used — Read/write

[Delete](#)

git branch

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git branch
* main
```

Atenção: O ramo principal deve se chamar **main**, não pode se chamar **master**.

- Caso na sua máquina apareça **master**, entre com o seguinte comando:

git branch -M main

Enviando os arquivos para o GitHub

- Na primeira vez que enviar arquivos, entre com o comando:

`git push -u origin main`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git push -u origin main
Enumerating objects: 24, done.
Counting objects: 100% (24/24), done.
Delta compression using up to 2 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (24/24), 65.50 KiB | 554.00 KiB/s, done.
Total 24 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/betopinheiro1005/curso-git-github-choren.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

The screenshot shows the GitHub web interface for the repository 'betopinheiro1005/curso-git-github-choren'. The repository is in the 'main' branch and has 1 branch and 0 tags. The file list includes .gitignore, README.md, changelog.txt, feature_1.txt, and feature_2.txt. The README.md file is selected and displays the title 'Curso de Git e GitHub' and a list of 9 topics. The right sidebar shows the repository's status, including 'Ready', 0 stars, 1 watching, and 0 forks. The footer of the page shows the GitHub logo and copyright information.

betopinheiro1005 Merge branch 'feature_2'

Go to file Add file - Code -

betopinheiro1005 Merge branch 'feature_2' 3 hours ago 7 commits

File	Description	Time
.gitignore	Reversão do estado antes dos comandos git restore e git reset	3 hours ago
README.md	Reversão do estado antes dos comandos git restore e git reset	3 hours ago
changelog.txt	docs: Agregar informação ao log de modificações	10 hours ago
feature_1.txt	feat: agregar uma nova funcionalidade ao sistema	4 hours ago
feature_2.txt	feat: agregar uma nova funcionalidade ao sistema	3 hours ago

README.md

Curso de Git e GitHub

1. Controle de versão
2. Conhecendo o Git
3. Instalando o Git CLI no Windows
4. Instalando o Git CLI no Linux
5. Comandos básicos do Git
6. Comando git commit
7. Comando git log
8. Comando git rm
9. Comando git diff

Curso de Git e GitHub - Professor Choren

Ready

0 stars

1 watching

0 forks

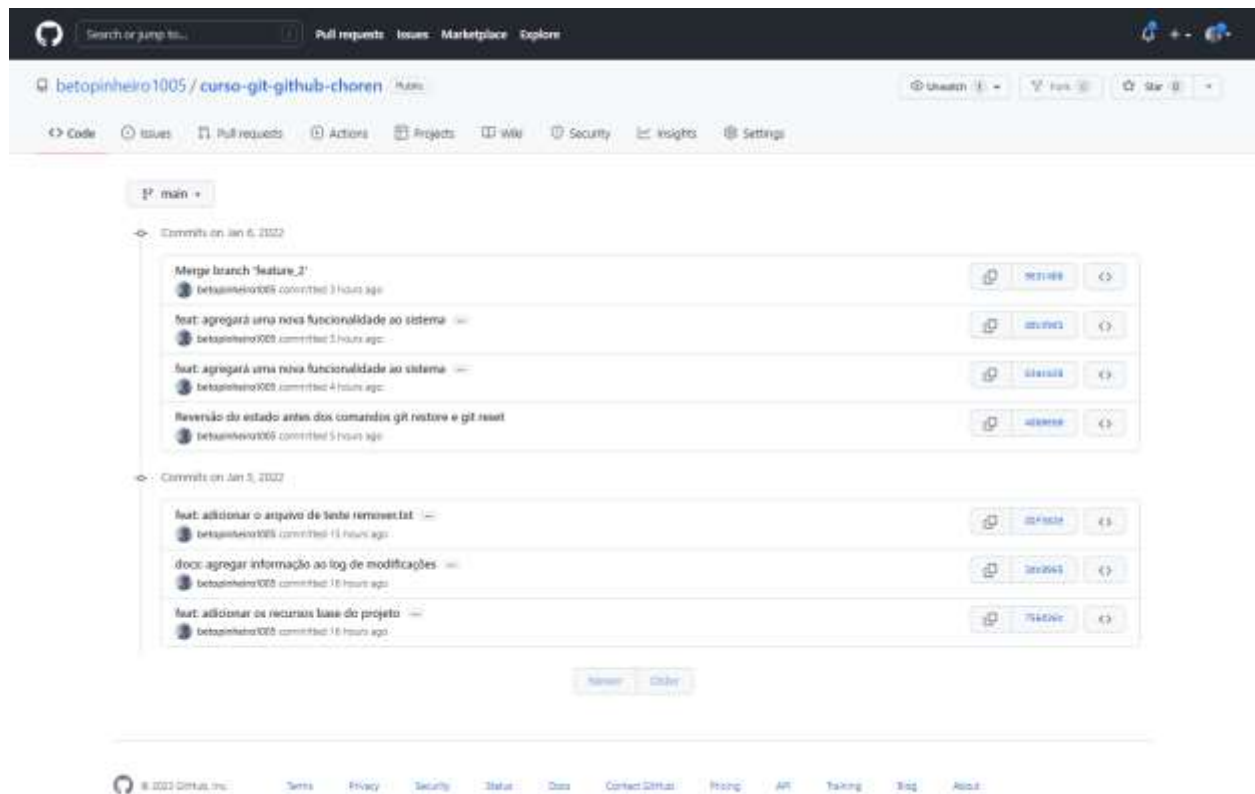
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About



- Na próxima aula iremos atualizar o arquivo **README.md**

Aula 16 - Comandos git clone e git push

- Apague a pasta que contém o repositório local (pasta `C:\xampp\htdocs\git_github_choren`).

Clonando o repositório remoto

- No GitHub, acesse o repositório remoto.

- Clique no botão "**Code**" e copie o endereço SSH:

`https://github.com/betopinheiro1005/curso-git-github-choren.git`

- Na pasta que ficará o repositório local (`I:\github`), entre com o comando:

`git clone https://github.com/betopinheiro1005/curso-git-github-choren.git`

```
betoi@DESKTOP-HFTIOCI MINGW64 /i/github
$ git clone https://github.com/betopinheiro1005/curso-git-github-choren.git
Cloning into 'curso-git-github-choren'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 27 (delta 7), reused 23 (delta 6), pack-reused 0
Receiving objects: 100% (27/27), 66.20 KiB | 1.44 MiB/s, done.
Resolving deltas: 100% (7/7), done.
```

- Será criada a pasta `curso-git-github-choren`.

`cd curso-git-github-choren`

`ls`

```
betoi@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ ls
README.md  changelog.txt  colega.txt  feature_1.txt  feature_2.txt
```


Alterando o arquivo README.md

README.md

Curso de Git e GitHub

Professor Choren

Lista de aulas - [Vídeos do curso](https://www.youtube.com/watch?v=beMnH51P-T4&list=PLmMxPWmzYRGcTabffOwHBORBJtKa2wCXS)

Aula 01 - Controle de versão

Aula 02 - Conhecendo o Git

Aula 03 - Instalando o Git CLI no Windows

Aula 04 - Instalando o Git CLI no Linux

Aula 05 - Comandos básicos do Git

Aula 06 - Comando git commit

Aula 07 - Comando git log

Aula 08 - Comando git rm

Aula 09 - Comando git diff

Aula 10 - Comando git restore e git reset

Aula 11 - Arquivo .gitignore

Aula 12 - Comando git checkout

Aula 13 - Comando git merge

Aula 14 - Comando git tag

Aula 15 - Conhecendo o GitHub e utilizando o comando git push

Aula 16 - Comando git clone e git push

Aula 17 - Comando git pull

Aula 18 - Publicando tags no GitHub

Aula 19 - Gestão de conflitos no merge

Aula 20 - Contribuição com repositório no GitHub

Enviando as alterações para o GitHub

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

git commit -a

.git\COMMIT_EDITMSG

doc: atualização do arquivo README.md, adição das aulas 10 a 20

Mudanças submetidas:

README.md

- Salve e feche o arquivo

```
beto1@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git commit -a
[main 50d3231] doc: atualização do arquivo README.md, adição das aulas 10 a 20
1 file changed, 26 insertions(+), 11 deletions(-)
rewrite README.md (89%)
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

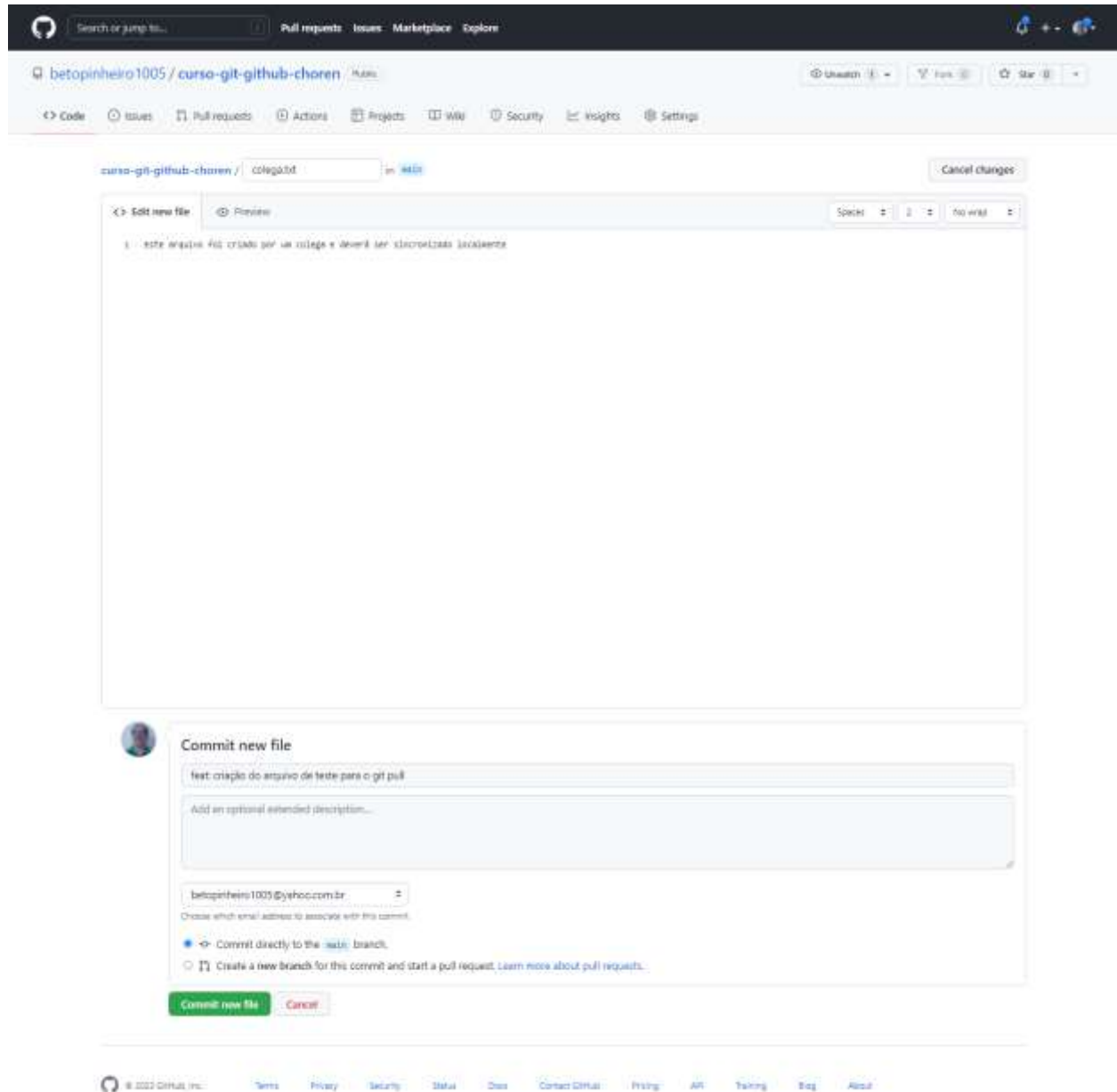
nothing to commit, working tree clean
```

git push

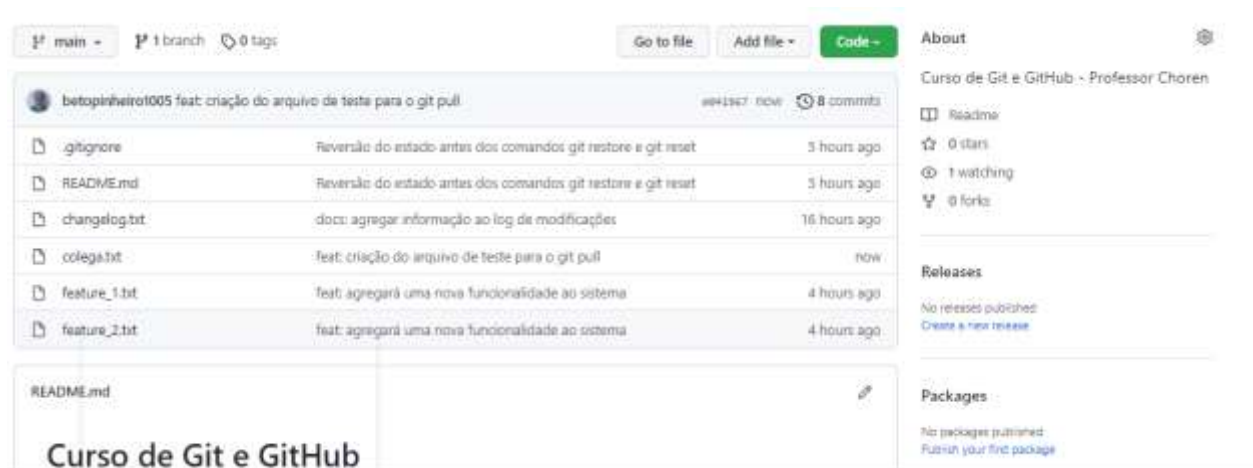
```
betol@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 475 bytes | 237.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/betopinheiro1005/curso-git-github-choren.git
  50d3231..3fd75eb  main -> main
```

Aula 17 - Comando git pull

- No GitHub clique no botão "Add File → Create new file"



- Faça as anotações e em seguida clique no botão "Commit new file"



Baixando a alteração para o repositório local

`git pull`

ou:

`git pull origin main`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 751 bytes | 3.00 KiB/s, done.
From https://github.com/betopinheiro1005/curso-git-github-choren
   963c4bb..e041967  main       -> origin/main
Updating 963c4bb..e041967
Fast-forward
   colega.txt | 1 +
   1 file changed, 1 insertion(+)
   create mode 100644 colega.txt
```

`git log --oneline`

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ git log --oneline
e041967 (HEAD -> main, origin/main) feat: criação do arquivo de teste para o git pull
963c4bb (tag: v1.0.0-beta) Merge branch 'feature_2'
ddc99d1 (tag: v0.2.0) feat: agregará uma nova funcionalidade ao sistema
63ac638 (tag: v0.1.0) feat: agregará uma nova funcionalidade ao sistema
4d60e60 Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

cat colega.txt

```
beto1@DESKTOP-HFTIOCI MINGW64 /c/xampp/htdocs/git_github_choren (main)
$ cat colega.txt
Este arquivo foi criado por um colega e deverá ser sincronizado localmente
```

Aula 18 - Publicando tags no GitHub

Criando uma tag no repositório local

```
git tag -a v1.0.0-beta -m "Versão 1.0.0.beta"
```

```
git tag
```

```
betoi@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git tag -a v1.0.0-beta -m "Versão 1.0.0.beta"

betoi@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git tag
v1.0.0-beta
```

```
git status
```

```
betoi@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Enviando a tag para o repositório remoto

```
git push --tags
```

```
betoi@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 182 bytes | 60.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/betopinheiro1005/curso-git-github-choren.git
* [new tag]          v1.0.0-beta -> v1.0.0-beta
```

The screenshot shows the GitHub web interface for the repository 'betopinheiro1005/curso-git-github-choren'. At the top, there are tabs for 'main', '1 branch', and '1 tag'. Below this is a table of files in the repository:

File	Description	Time
.gitignore	Reversão do estado antes dos comandos git restore e git reset	10 hours ago
README.md	doc: atualização do arquivo README.md, edição das aulas 10 e 20	2 hours ago
changelog.txt	docs: agregar informação ao log de modificações	21 hours ago
colega.txt	feat: criação do arquivo de teste para o git pull	5 hours ago
feature_1.txt	feat: agregará uma nova funcionalidade ao sistema	6 hours ago
feature_2.txt	feat: agregará uma nova funcionalidade ao sistema	6 hours ago

On the right side, there is a sidebar with repository statistics: 'About', 'Curso de Git e GitHub - Professor Choren', '0 stars', '1 watching', and '0 forks'. Below this, there is a 'Releases' section showing '1 tag' and a link to 'Create a new release'.

Releases Tags

Tags

v1.0.0-beta

8 minutes ago 1475kb zip tar

Releases Tags

Create release from tag Delete

v1.0.0-beta
1475kb

Compare

v1.0.0-beta

betopinheiro1005 tagged this 9 minutes ago

Version 1.0.0-beta

Assets 2

Source code (zip)

Source code (tar.gz)

Aula 19 - Gestão de conflitos no merge

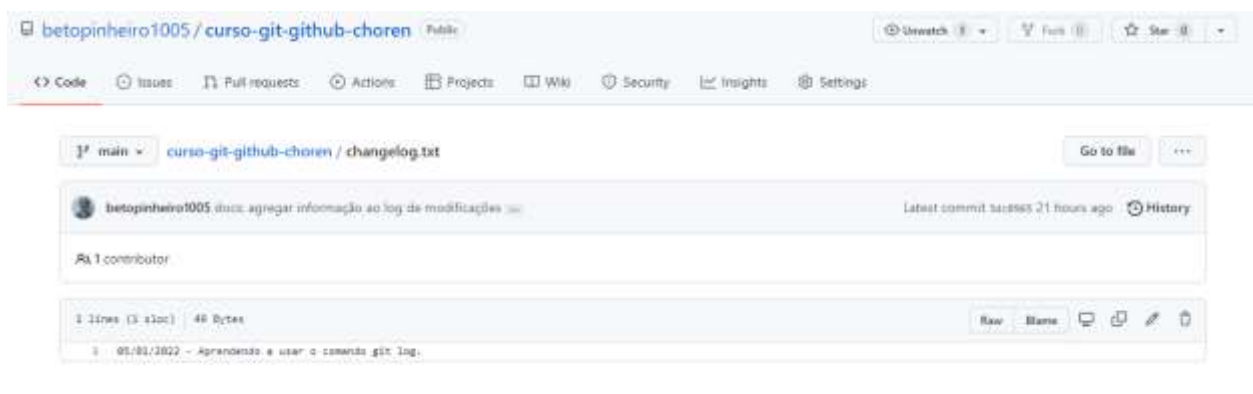
É muito comum, que em projetos grandes ou até mesmo pequenos, haja conflitos entre versões de códigos, principalmente quando trabalhamos em equipes.

Um conflito ocorre quando dois programadores estão trabalhando encima do mesmo arquivo e eles modificam as mesmas linhas, cada um em seu ambiente de trabalho. Será necessário informar qual é a versão que se quer manter.

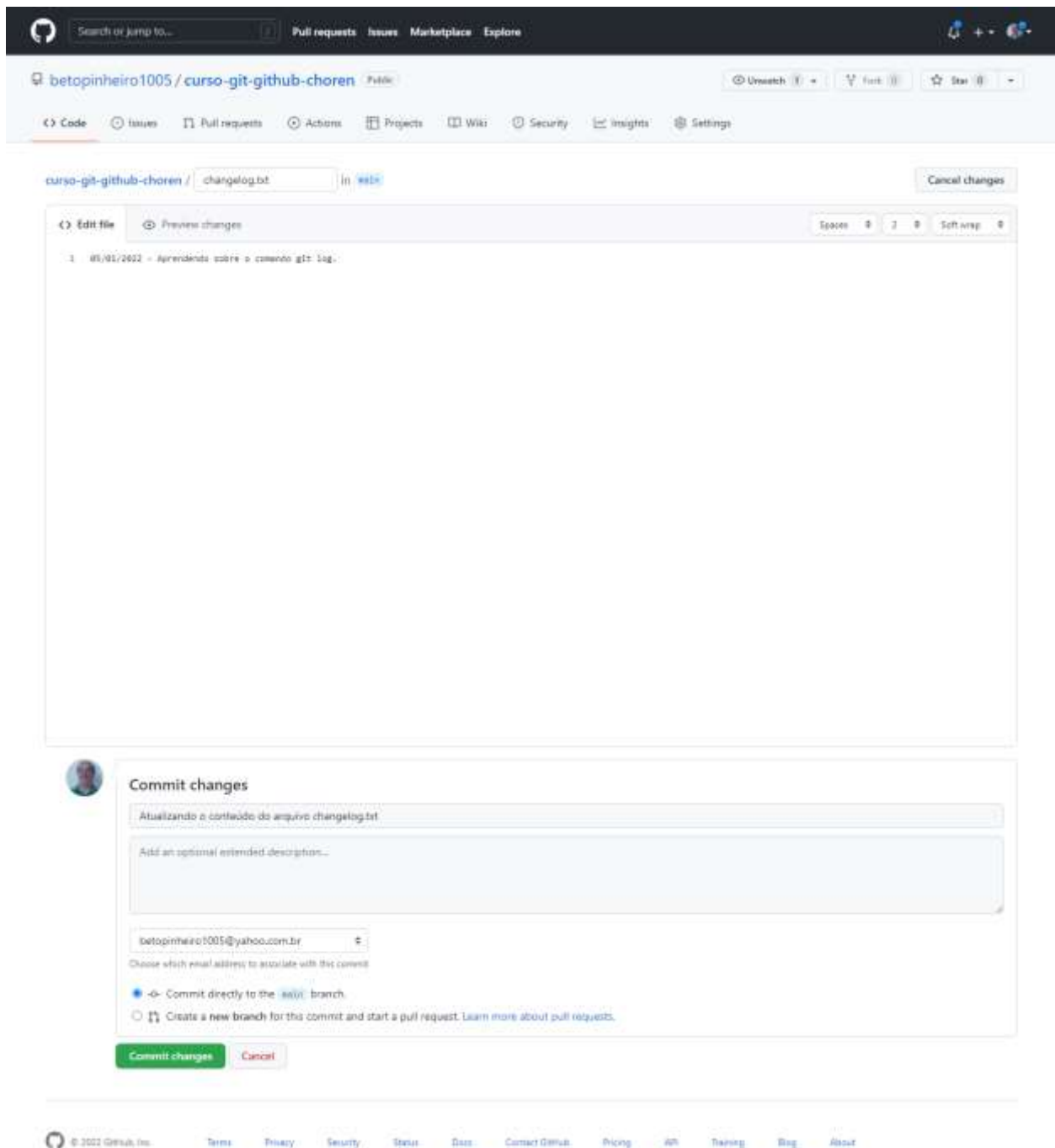
Não pode haver duas versões do mesmo código. Só podemos ter uma única fonte de dados, uma única fonte de código.

Exemplo de conflito

- No repositório remoto (servidor):

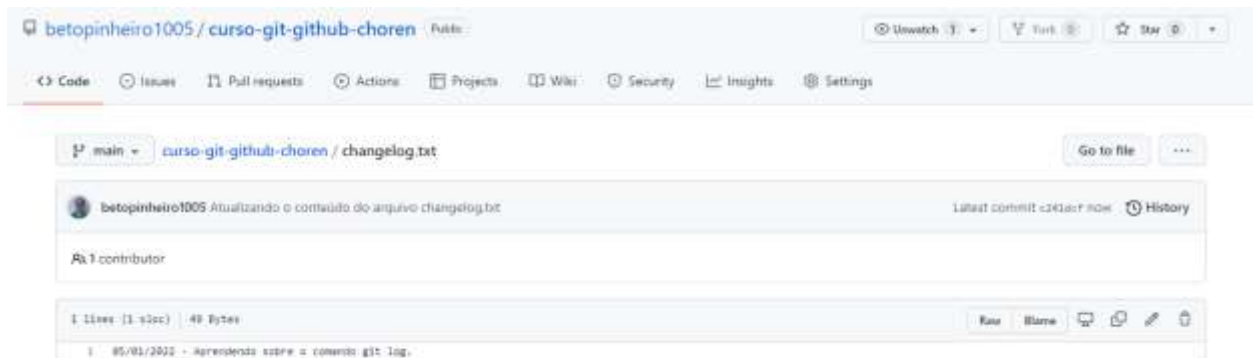


Fazendo uma edição do arquivo changelog.txt diretamente pelo GitHub



The screenshot displays the GitHub web interface for editing a file. At the top, the navigation bar includes a search field and links for Pull requests, Issues, Marketplace, and Explore. The repository path is `betopinheiro1005 / curso-git-github-choreen`, with options to Unwatch, Fork, Star, and a dropdown menu. Below the navigation bar, a breadcrumb trail shows `curso-git-github-choreen / changelog.txt` with a link to view the file's history. The main editing area has two tabs: `Edit file` (active) and `Preview changes`. The `Edit file` tab shows the content of `changelog.txt` with a single line: `1 05/01/2022 - Aprendendo sobre o comando git log.` To the right of the editor are settings for `Spaces` (0), `Trailing spaces` (0), and `Soft wrap` (0). A `Cancel changes` button is located in the top right corner of the editor area. Below the editor, the `Commit changes` dialog is open. It features a profile picture of the user, a text input field with the placeholder `Atualizando o conteúdo do arquivo changelog.txt`, and a larger text area for an optional extended description. The commit email address is set to `betopinheiro1005@yahoo.com.br`. Below this, there is a prompt to choose which email address to associate with the commit. Two options are available: `Commit directly to the main branch.` (selected with a radio button) and `Create a new branch for this commit and start a pull request. Learn more about pull requests.` (unselected with a radio button). At the bottom of the dialog are two buttons: `Commit changes` (green) and `Cancel` (red). The footer of the page shows the GitHub logo, copyright information (© 2022 GitHub, Inc.), and links to Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

- Clique no botão "Commit changes"



- No repositório local:

`cat changelog.txt`

```
beto1@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ cat changelog.txt
05/01/2022 - A
```

- Edite o arquivo `changelog.txt`:

`changelog.txt`

05/01/2022 - A

06/01/2022 - Aprendendo sobre a gestão de conflitos

`git commit -a`

`.git\COMMIT_EDITMSG`

doc: atualização do changelog.txt

Adição da linha de log sobre gestão de conflitos

Mudança submetida:

`changelog.txt`

- Salve e feche o arquivo

```
beto1@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git commit -a
[main 5a83817] doc: atualização do changelog.txt
1 file changed, 3 insertions(+), 2 deletions(-)
```

git status

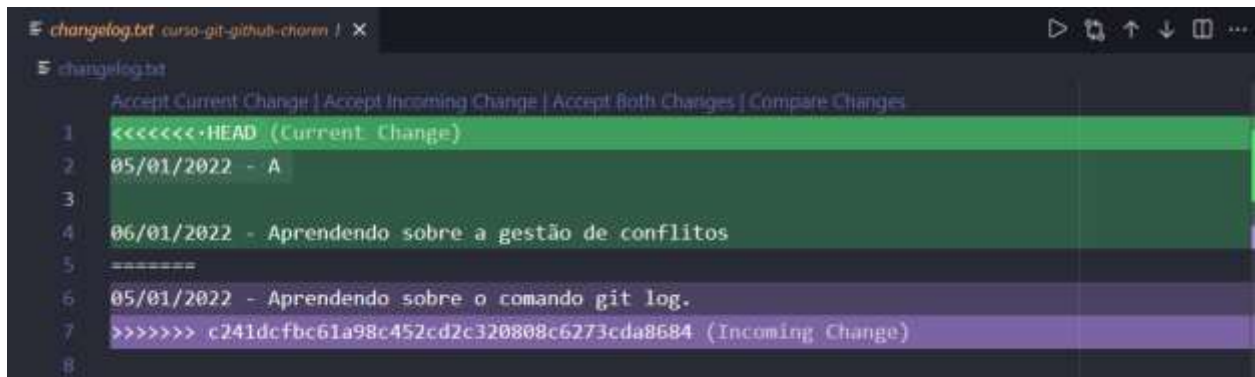
```
betoi@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

git pull

```
betoi@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 709 bytes | 13.00 KiB/s, done.
From https://github.com/betopinheiro1005/curso-git-github-choren
   3fd75eb..c241dcf  main       -> origin/main
Auto-merging changelog.txt
CONFLICT (content): Merge conflict in changelog.txt
Automatic merge failed; fix conflicts and then commit the result.
```

No arquivo **changelog.txt**:



O conflito é exibido no arquivo.

São exibidas as seguintes opções:

- Accept Current Change (Aceite a mudança atual)
- Accept Incoming Change (Aceite a mudança recebida)
- Accept Both Changes (Aceite ambas as mudanças)
- Compare Changes (Compare mudanças)

- Se for clicada na opção Accept Current Change, teremos como resultado:

05/01/2022 - A

06/01/2022 - Aprendendo sobre a gestão de conflitos

- Se for clicada na opção Accept Incoming Change, teremos como resultado:

05/01/2022 - Aprendendo sobre o comando git log.

- Se for clicada na opção Accept Both Changes, teremos como resultado:

05/01/2022 - A

06/01/2022 - Aprendendo sobre a gestão de conflitos

05/01/2022 - Aprendendo sobre o comando git log.

Clique na última opção:

changelog.txt

05/01/2022 - A

06/01/2022 - Aprendendo sobre a gestão de conflitos

05/01/2022 - Aprendendo sobre o comando git log.

- Apague as duas primeiras linhas:

changelog.txt

06/01/2022 - Aprendendo sobre a gestão de conflitos

05/01/2022 - Aprendendo sobre o comando git log.

- Salve o arquivo

git commit -a

.git\COMMIT_EDITMSG

Merge branch 'main' of https://github.com/betopinheiro1005/curso-git-github-choren

```
# Conflicts:
#  changelog.txt
#
# It looks like you may be committing a merge.
# If this is not correct, please run
#  git update-ref -d MERGE_HEAD
# and try again.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Your branch and 'origin/main' have diverged,
# and have 2 and 1 different commits each, respectively.
# (use "git pull" to merge the remote branch into yours)
#
# All conflicts fixed but you are still merging.
#
# Changes to be committed:
#   modified:   changelog.txt
#
```

- Altere para:

.git\COMMIT_EDITMSG

fix: Merge branch 'main' of https://github.com/betopinheiro1005/curso-git-github-choren

```
Conflitos:
  changelog.txt
Mudança submetida:
  changelog.txt
```

- Salve e feche o arquivo

```
betol@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main|MERGING)
$ git commit -a
[main f5d7d1d] fix: Merge branch 'main' of https://github.com/betopinheiro1005/curso-git-github-choren
```

git log --oneline

```
betol@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git log --oneline
f5d7d1d (HEAD -> main) fix: Merge branch 'main' of https://github.com/betopinheiro1005/curso-git-github-choren
c241dcf (origin/main, origin/HEAD) Atualizando o conteúdo do arquivo changelog.txt
5a83817 doc: atualização do changelog.txt
fcb6fd0 doc: atualização do changelog.txt
3fd75eb (tag: v1.0.0-beta) doc: atualização do arquivo README.md, adição das aulas 10 a 20
50d3231 doc: atualização do arquivo README.md, adição das aulas 10 a 20
e041967 feat: criação do arquivo de teste para o git pull
963c4bb Merge branch 'feature_2'
ddc99d1 feat: agregará uma nova funcionalidade ao sistema
63ac638 feat: agregará uma nova funcionalidade ao sistema
4d60e60 Reversão do estado antes dos comandos git restore e git reset
d2f362e feat: adicionar o arquivo de teste remover.txt
3dc0965 docs: agregar informação ao log de modificações
756d36c feat: adicionar os recursos base do projeto
```

git status

```
betol@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

git push

```
betol@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 2 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 1015 bytes | 169.00 KiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 1 local object.
To https://github.com/betopinheiro1005/curso-git-github-choren.git
   c241dcf..f5d7d1d  main -> main
```

git status

```
betol@DESKTOP-HFTIOCI MINGW64 /i/github/curso-git-github-choren (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

- No GitHub:

betopinheiro1005 / curso-git-github-choren Public

Unwatch Fork Star

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 1 tag Go to file Add file Code

betopinheiro1005 fix: Merge branch 'main' of https://github.com/betopinheiro1005/curso-git-github-choren · f547c2a · 15 minutes ago · 14 commits

.gitignore	Reversão do estado antes dos comandos git restore e git reset	12 hours ago
README.md	doc: atualização do arquivo README.md, adição das aulas 10 e 20	4 hours ago
changelog.txt	fix: Merge branch 'main' of https://github.com/betopinheiro1005/curso-git-github-choren	15 minutes ago
coliga.txt	feat: criação do arquivo de teste para o git pull	6 hours ago
feature_1.txt	feat: agregará uma nova funcionalidade ao sistema	10 hours ago
feature_2.txt	feat: agregará uma nova funcionalidade ao sistema	10 hours ago

About

Curso de Git e GitHub - Professor Choren

Readme

0 stars

1 watching

0 forks

Releases

1 tags

Create a new release

betopinheiro1005 / curso-git-github-choren Public

Unwatch Fork Star

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main curso-git-github-choren / changelog.txt Go to file

betopinheiro1005 fix: Merge branch 'main' of https://github.com/betopinheiro1005/curso-git-github-choren · f547c2a · 15 minutes ago · History

1 contributor

2 lines (2 sloc) 102 bytes Raw Blame

```
1 06/01/2022 - Aprendendo sobre a gestão de conflitos
2 05/01/2021 - Aprendendo sobre o comando git log.
```


main

Commits on Jan 5, 2022

file Merge branch 'main' of https://github.com/betopinheiro1005/curso...	#5c7e5a
betopinheiro1005 committed 18 minutes ago	
Atualizando o conteúdo do arquivo changelog.txt	Verified 1281a1c4
betopinheiro1005 committed 1 hour ago	
doc: atualização do changelog.txt	5a638317
betopinheiro1005 committed 1 hour ago	
doc: atualização do changelog.txt	9c06f088
betopinheiro1005 committed 1 hour ago	
doc: atualização do arquivo README.md, adição das aulas 10 a 20	3f075eb0
betopinheiro1005 committed 4 hours ago	
doc: atualização do arquivo README.md, adição das aulas 10 a 20	59d01121
betopinheiro1005 committed 4 hours ago	
feat: criação do arquivo de teste para o git pull	Verified e9419e07
betopinheiro1005 committed 6 hours ago	
Merge branch 'feature_2'	885c489c
betopinheiro1005 committed 9 hours ago	
feat: agregará uma nova funcionalidade ao sistema	89c79d12
betopinheiro1005 committed 10 hours ago	
feat: agregará uma nova funcionalidade ao sistema	65ac8558
betopinheiro1005 committed 10 hours ago	
Reversão do estado antes dos comandos git restore e git reset	4000e008
betopinheiro1005 committed 12 hours ago	

Commits on Jan 5, 2022

feat: adicionar o arquivo de teste remover.txt	a147362e
betopinheiro1005 committed 22 hours ago	
doc: agregar informação ao log de modificações	1ac89605
betopinheiro1005 committed 22 hours ago	
feat: adicionar os recursos base do projeto	754ac36c
betopinheiro1005 committed 23 hours ago	

Previous Older

Aula 20 - Contribuição com repositório no GitHub

Comando pull request

pull request é uma forma de enviar solicitações de alterações para o repositório cujo dono não é você, ou que você não faz parte da equipe de desenvolvimento.

Na realidade significa solicitar um request a partir de um **fork** que você faz.

fork é uma cópia exata do projeto para sua conta no GitHub que possibilita que você faça melhorias.

Após isso, é solicita-se ao dono desse repositório que puxe essas informações do seu repositório para o repositório dele, aceitando suas modificações e fazendo esse merging dentro do projeto.

Fazendo o fork do repositório

<https://github.com/professorchoren>

The screenshot shows the GitHub profile page for 'Professor Choren' (username: professorchoren). The profile includes a blue circular avatar with a white silhouette of a person wearing a graduation cap and the text 'PROFESSOR CHOREN'. Below the avatar, the name 'Professor Choren' and username 'professorchoren' are displayed, along with a 'Follow' button. Social links for 'andersonchoren.com.br', '@professorchoren', and a 'Block or Report' option are listed. The 'Popular repositories' section shows two repositories: 'curso_de_git_e_github' and 'curso_de_gulp', both described as 'Material produzido durante o curso de Git e GitHub do Canal Professor Choren no Youtube'. The '18 contributions in the last year' section features a calendar grid with green squares indicating contributions, primarily in November and December. The 'Contribution activity' section shows a timeline for January 2022, stating 'professorchoren has no activity yet for this period.' with a 'Show more activity' link. The footer contains the GitHub logo, copyright notice '© 2022 GitHub, Inc.', and links for Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

main 1 branch 1 tag

Go to file Add file Code

professorchoren Update README.md	4188f74 · on 3 Dec 2021	14 commits
gitignore	Reverte o estado antes dos comandos git restore e git reset	3 months ago
README.md	Update README.md	last month
changelog.txt	fix: Merge branch 'main' of github.com:professorchoren/curso_de_git_e_github	2 months ago
criaiga.txt	feat: criação do arquivo de teste para o git pull	2 months ago
feature_1.txt	feat: agregará um nova funcionalidade ao sistema	2 months ago
feature_2.txt	feat: agregará um nova funcionalidade ao sistema	2 months ago

README.md

Curso de Git e GitHub

1. Controle de versão
2. Conhecendo o Git
3. Instalando o Git CLI no Windows
4. Instalando o Git CLI no Linux
5. Comandos básicos do Git
6. Comando git commit
7. Comando git log
8. Comando git mv
9. Comando git diff
10. Comandos git restore e reset
11. O arquivo .gitignore
12. Comando git checkout
13. Comando git merge
14. Comando git tag
15. Conhecendo o GitHub e utilizando o comando git push
16. Comando Git push - Publicando alterações no GitHub
17. Comando Git pull
18. Publicando tags no GitHub
19. Gestão de conflitos no merge
20. Contribuição com repositório no GitHub

About

Material produzido durante o curso de Git e GitHub do Canal Professor Choren no Youtube

Readme

0 stars

1 watching

0 forks

Releases

1 tags

Packages

No packages published

- Clique no botão "Fork"

The screenshot shows the GitHub interface for a repository named 'curso_de_git_e_github' by user 'betopinheiro1005'. The repository is forked from 'professorchoren/curso_de_git_e_github'. The main branch is 'main', and there is 1 branch and 1 tag. The repository contains several files: .gitignore, README.md, changelog.txt, colga.txt, feature_1.txt, and feature_2.txt. The README.md file is open, showing a list of 20 topics related to Git and GitHub. The right sidebar shows the repository's statistics: 0 stars, 0 watching, and 1 fork. The footer of the page displays the GitHub logo and copyright information for 2022.

Search or jump to... Pull requests Issues Marketplace Explore

betopinheiro1005 / curso_de_git_e_github Public

Watch Fork Star

Code Pull requests Actions Projects Wikis Security Insights Settings

main 1 branch 1 tag

Go to file Add file Code

This branch is up to date with professorchoren/main. Contribute Fetch upstream

professorchoren Update README.md at8874: Jan 3 Dec 2021 14 commits

.gitignore	Reversao do estado antes dos comandos git restore e git reset	3 months ago
README.md	Update README.md	last month
changelog.txt	fix: Merge branch 'main' of github.com:professorchoren/curso_de_git_e_github	2 months ago
colga.txt	feat: criação do arquivo de teste para o git pull	2 months ago
feature_1.txt	feat: agregará um nova funcionalidade ao sistema	2 months ago
feature_2.txt	feat: agregará um nova funcionalidade ao sistema	1 month ago

README.md

Curso de Git e GitHub

1. Controle de versão
2. Conhecendo o Git
3. Instalando o Git CLI no Windows
4. Instalando o Git CLI no Linux
5. Comandos básicos do Git
6. Comando git commit
7. Comando git log
8. Comando git rm
9. Comando git diff
10. Comandos git restore e reset
11. O arquivo .gitignore
12. Comando git checkout
13. Comando git merge
14. Comando git tag
15. Conhecendo o GitHub e utilizando o comando git push
16. Comando Git push - Publicando alterações no GitHub
17. Comando Git pull
18. Publicando tags no GitHub
19. Gestão de conflitos no merge
20. Contribuição com repositório no GitHub

About

Material produzido durante o curso de Git e GitHub do Canal Professor Choren no Youtube

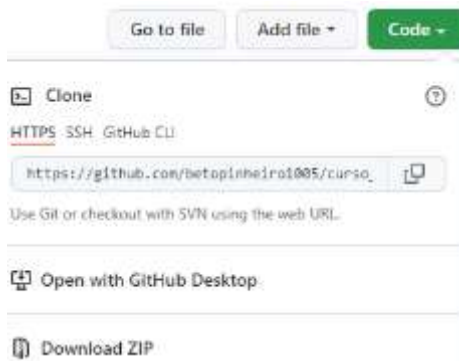
Readme 0 stars 0 watching 1 fork

Releases 1 tags Create a new release

Packages No packages published Publish your first package

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

- Clique no botão "Code" e copie o endereço SSH:



https://github.com/betopinheiro1005/curso_de_git_e_github.git

Clonando o repositório remoto para sua máquina

git clone https://github.com/betopinheiro1005/curso_de_git_e_github.git

```
betol@DESKTOP-HFTIOCI MINGW64 /i/github
$ git clone https://github.com/betopinheiro1005/curso_de_git_e_github.git
Cloning into 'curso_de_git_e_github'...
remote: Enumerating objects: 45, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 45 (delta 17), reused 32 (delta 11), pack-reused 0
Receiving objects: 100% (45/45), 146.27 KiB | 1.68 MiB/s, done.
Resolving deltas: 100% (17/17), done.
```

cd curso_de_git_e_github

```
betol@DESKTOP-HFTIOCI MINGW64 /i/github
$ cd curso_de_git_e_github

betol@DESKTOP-HFTIOCI MINGW64 /i/github/curso_de_git_e_github (main)
$ .....
```

ls

```
betol@DESKTOP-HFTIOCI MINGW64 /i/github/curso_de_git_e_github (main)
$ ls
README.md  changelog.txt  colega.txt  feature_1.txt  feature_2.txt
```

- Faça alterações no projeto original ([curso_de_git_e_github](#)), que está em sua máquina, para acrescentar alguma melhoria, recurso ou funcionalidade que seja útil ao projeto.

OBS.: Como contribuição adicionei esta apostila (em formato PDF) chamada "[Curso_Git_Github-Prof_Choren.pdf](#)" que elaborei e que contém o resumo do curso, com o código parcial de cada uma das aulas, além de anotações importantes visando facilitar os estudos deste curso.

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /i/github/curso_de_git_e_github (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)
      Curso_Git_Github-Prof_Choren.pdf

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

git add .

```
beto1@DESKTOP-HFTIOCI MINGW64 /i/github/curso_de_git_e_github (main)
$ git add .
```

git status

```
beto1@DESKTOP-HFTIOCI MINGW64 /i/github/curso_de_git_e_github (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:   Curso_Git_Github-Prof_Choren.pdf
```

git commit

.git\COMMIT_EDITMSG

doc: adicionado apostila com resumo do curso

Mudança submetida:

Curso_Git_Github-Prof_Choren.pdf

- Salve e feche o arquivo

```
betol@DESKTOP-HFTIOCI MINGW64 /i/github/curso_de_git_e_github (main)
$ git commit
[main 2db7fa5] doc: adicionado apostila com resumo do curso
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Curso_Git_Github-Prof_Choren.pdf
```

git push

```
betol@DESKTOP-HFTIOCI MINGW64 /i/github/curso_de_git_e_github (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.62 MiB | 733.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/betopinheiro1005/curso_de_git_e_github.git
   d3b9774..2db7fa5  main -> main
```

- No GitHub:

The screenshot shows a GitHub repository page for 'betopinheiro1005 / curso_de_git_e_github'. The repository is a fork of 'professorchoren/curso_de_git_e_github'. The main branch is 'main', which is 1 commit ahead of 'professorchoren/main'. The repository contains 15 commits and 1 tag. The commit history shows a series of updates to the README, including a new section for 'Curso de Git e GitHub' with 20 topics. The README content is as follows:

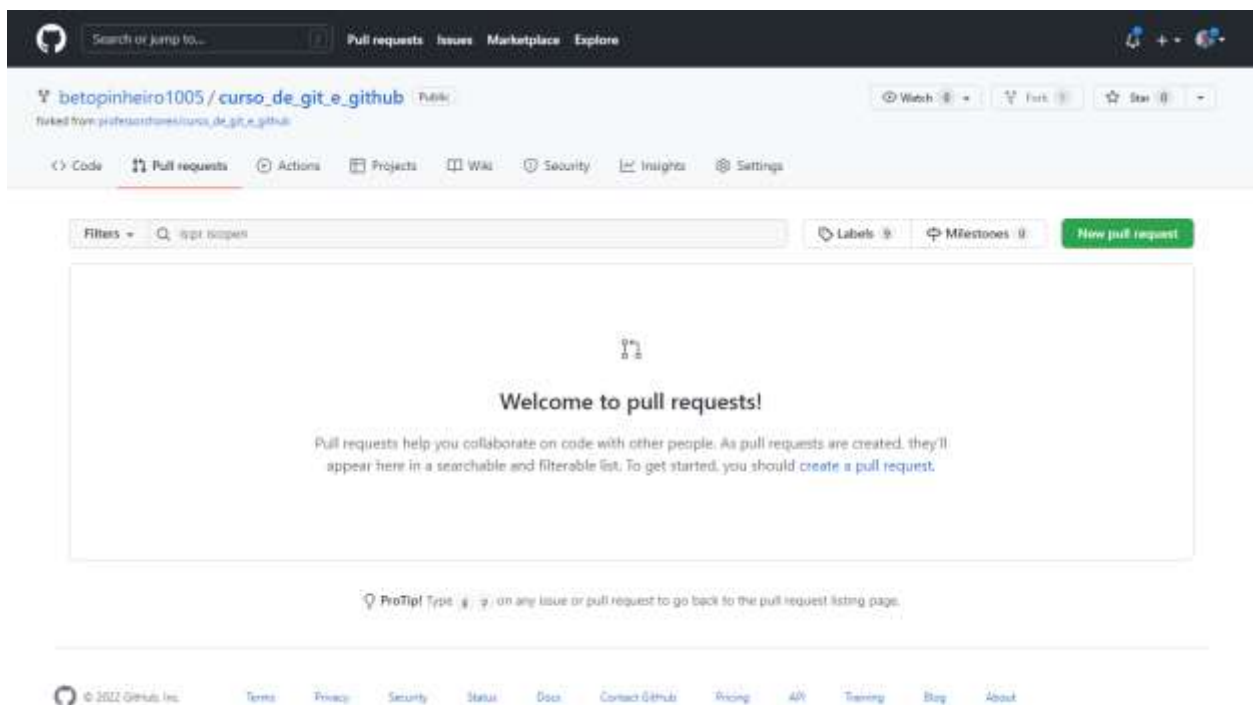
Curso de Git e GitHub

1. Controle de versão
2. Conhecendo o Git
3. Instalando o Git CLI no Windows
4. Instalando o Git CLI no Linux
5. Comandos básicos do Git
6. Comando git commit
7. Comando git log
8. Comando git rm
9. Comando git diff
10. Comandos git restore e reset
11. O arquivo .gitignore
12. Comando git checkout
13. Comando git merge
14. Comando git tag
15. Conhecendo o GitHub e utilizando o comando git push
16. Comando Git push - Publicando alterações no GitHub
17. Comando Git pull
18. Publicando tags no GitHub
19. Gestão de conflitos no merge
20. Contribuição com repositório no GitHub

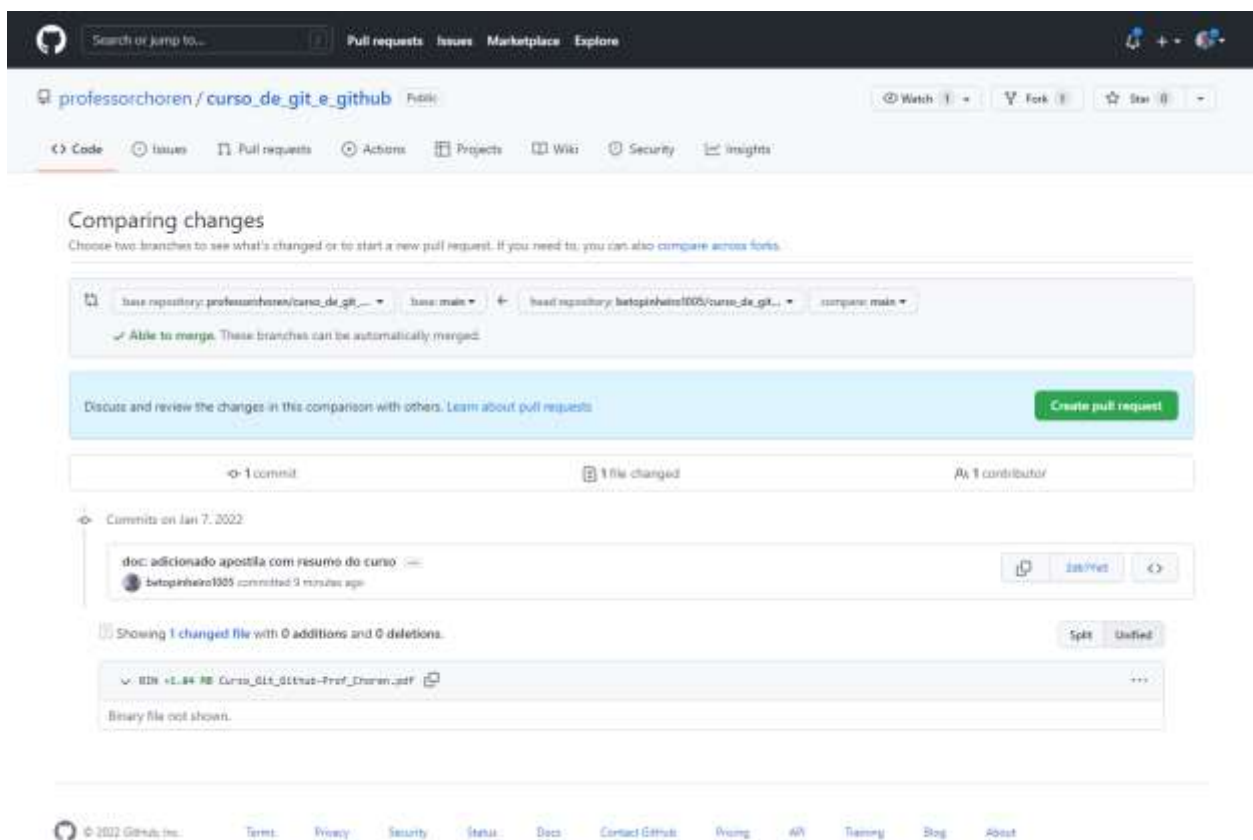
The right sidebar shows the repository's 'About' section, which states: 'Material produzido durante o curso de Git e GitHub do Canal Professor Choren no Youtube'. It also shows 0 stars, 0 watchers, and 1 fork. The 'Releases' section shows 1 tag. The 'Packages' section shows 0 packages published.

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

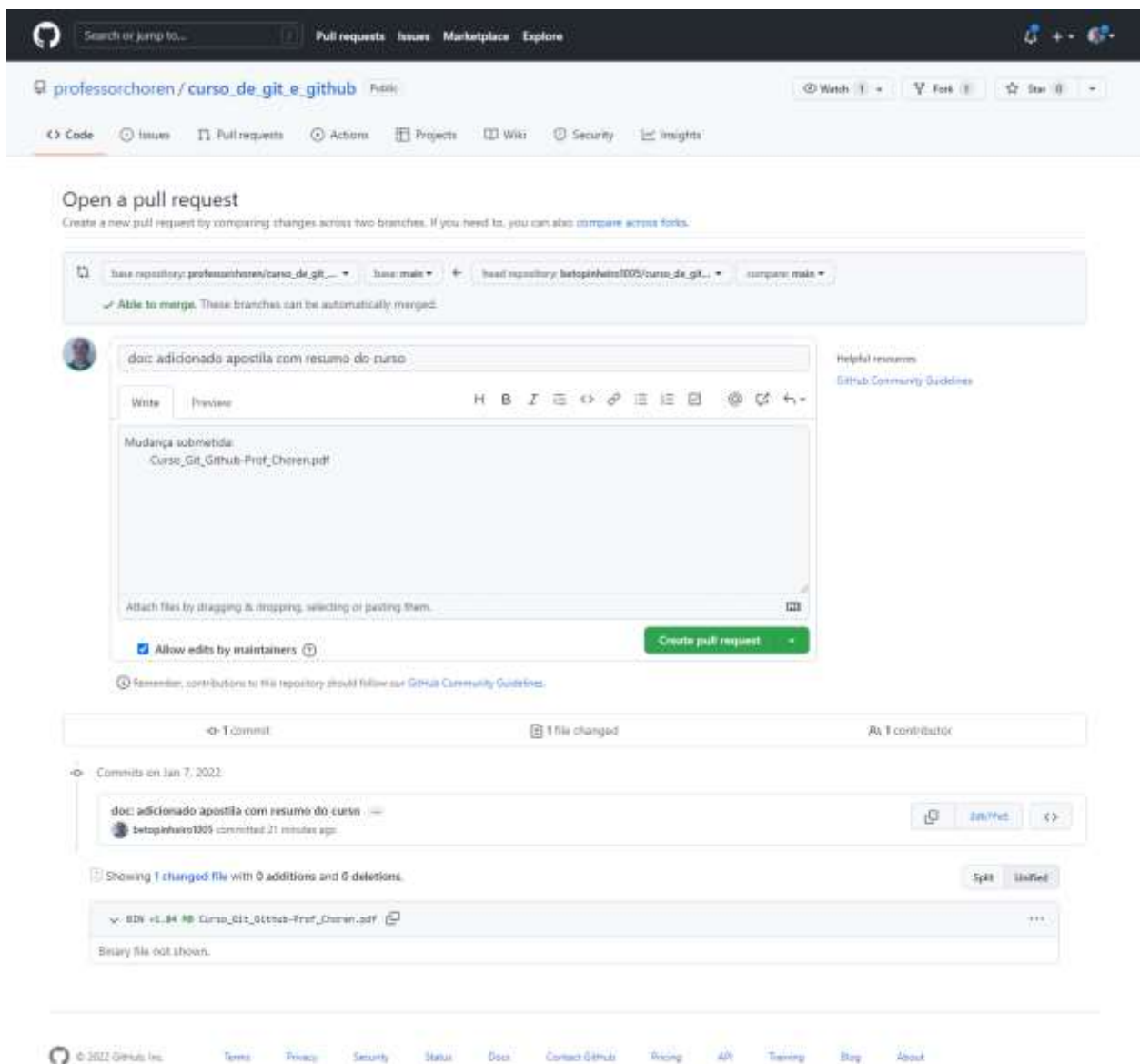
- clique no botão "Pull requests"



- Clique no botão "New pull request"



- Clique no botão "Create pull request"



- Clique no botão "Create pull request"

- Na tela que se abrir, há um campo (Leave a comment) que permite que você deixe um comentário e/ou crie uma mensagem para o dono do canal.

- Neste campo, digite uma solicitação ao autor do projeto para que ele analise sua contribuição e caso esteja de acordo aprove-a fazendo a aceitação para incorporá-la ao repositório dele.

- Em seguida clique no botão "Comment". Faça quantos comentários desejar.

- Aguarde a resposta.