

GIT E GITHUB

Nicolly Vieira Santos Costa



INTRODUÇÃO

Controle de versão

Git

GitHub

Na prática

CONTROLE DE VERSÃO

DEFINIÇÃO

SISTEMA LOCAL

SISTEMA CENTRALIZADO

SISTEMA DISTRIBUÍDO

- Controle de versão ou versionamento é um sistema que registra as alterações em um ou mais arquivos;
- Ajuda a reverter determinados erros ou até projetos inteiros;
- Compara as mudanças ao longo do tempo;
- E caso ocorra a perda de algum arquivo, tem fácil recuperação do próprio.

CONTROLE DE VERSÃO

DEFINIÇÃO

SISTEMA LOCAL

SISTEMA CENTRALIZADO

SISTEMA DISTRIBUÍDO

- Cópia arquivos de um diretório para outro, duplicando;
- Comum e mais simples;
- Propenso a erros;
- Fácil de se confundir;
- Difícil o compartilhamento com os colegas do projeto.

CONTROLE DE VERSÃO

DEFINIÇÃO

SISTEMA LOCAL

SISTEMA
CENTRALIZADO

SISTEMA DISTRIBUÍDO

- Um único servidor;
- Contêm todos os arquivos de controle de versão;
- Vários desenvolvedores usam arquivos a partir desse lugar central;
- Por muito tempo foi o padrão.
- Se o disco for corrompido, é muito difícil a recuperação de seu conteúdo.

CONTROLE DE VERSÃO

DEFINIÇÃO

SISTEMA LOCAL

SISTEMA CENTRALIZADO

SISTEMA DISTRIBUÍDO

- Acesso total pelo repositório remoto;
- Todos tem uma cópia do projeto;
- Cada clone é um backup dos dados;
- Não necessita da internet para trabalhar;
- Clientes usam o estado mais recentes dos arquivos.

GIT

DEFINIÇÃO

DESEMPENHO

FLEXIBILIDADE

SEGURANÇA

- É um sistema de versão de controle distribuído;
- Opensource;
- Desenvolvido em 2005 por Linus Torvalds;
- Capacidade de criar pontos na história (commit);

GIT

DEFINIÇÃO

DESEMPENHO

FLEXIBILIDADE

SEGURANÇA

- Capacidade de criar ramificações do projeto (Branch);
- Análise e resolução de conflito;
- Vários desenvolvedores do mesmo projeto;
- Número impressionante de projetos de software dependem do git para controle de versão.

GIT

DEFINIÇÃO

DESEMPENHO

FLEXIBILIDADE

SEGURANÇA

- Capacidade de trabalhar em mais versões do mesmo código;
- Foca no formato do arquivo;
- Duas pessoas podem trabalhar ao mesmo tempo sem modificações;
- Utiliza o Github como servidor.

GIT

DEFINIÇÃO

DESEMPENHO

FLEXIBILIDADE

SEGURANÇA

- Suporte a vários tipos de fluxos de trabalho;
- Eficiência em projetos pequenos e grandes;
- Compatibilidade com muitos sistemas e protocolos.

GIT

DEFINIÇÃO

DESEMPENHO

FLEXIBILIDADE

SEGURANÇA

- O conteúdo dos arquivos, versões, commits, diretórios, todos esses objetos do repositório são protegidos com um algoritmo de hash de criptografia seguro chamado SHA1.

GIT



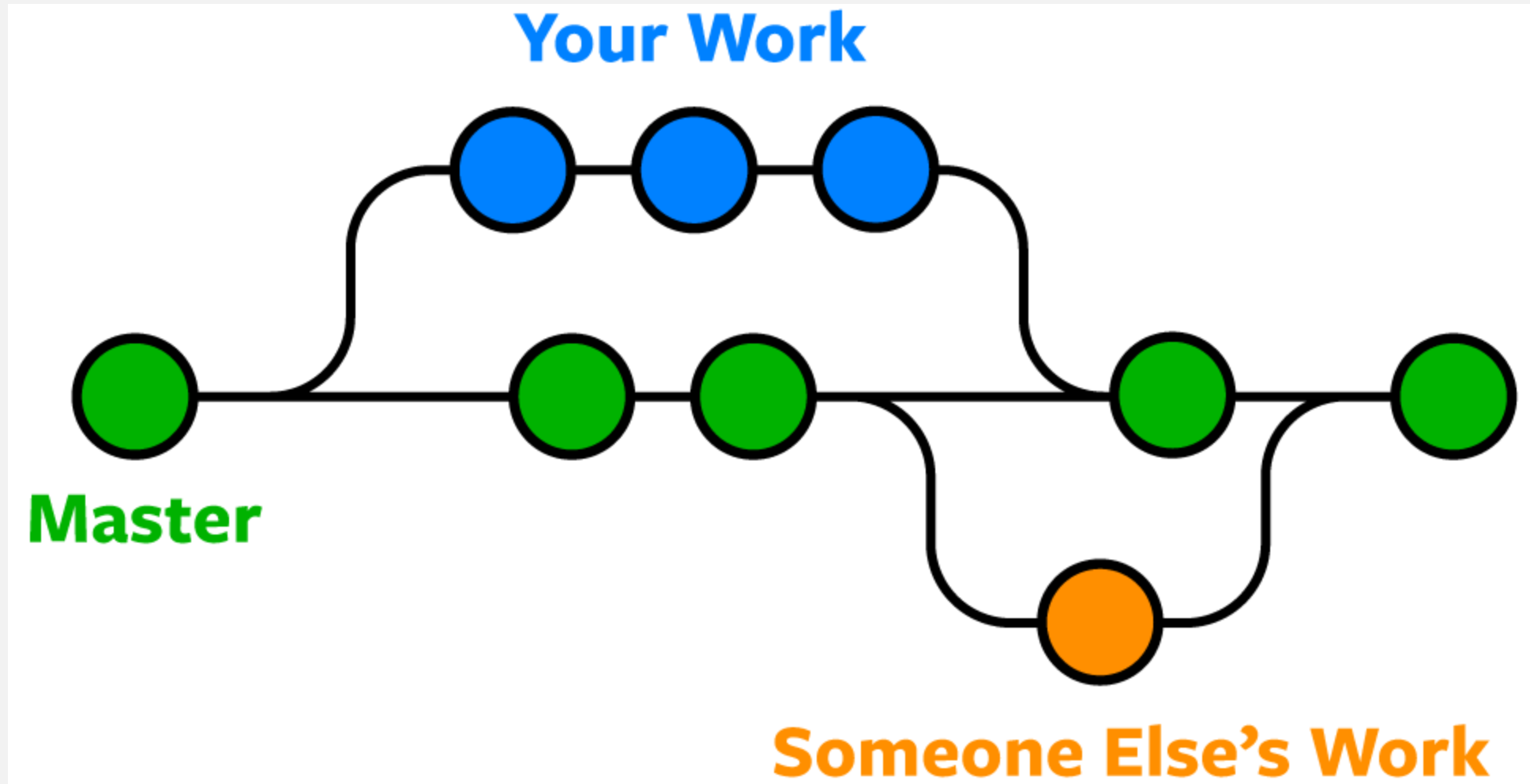
- Repositório: Local físico onde os arquivos e suas cópias ficarão armazenados.
- Commit: É a versão que foi modificada de um arquivo. O git armazena para caso você precisa recuperar uma versão anterior.
- Branch: São cópias do código original que podem ser manipuladas de forma livre pela equipe, sem afetar no código fonte.

GIT



- Merge: Após o trabalho em uma branch, é necessário realizar o merge fundindo a copia com a linha principal. Acontece apenas no repositório local.
- Push: é o envio das ramificações apos o merge para o repositório central, parar que as outras pessoas possam atualizar suas copias.
- Pull: utilizado quando outra pessoa que atua no desenvolvimento muda o ramo principal no repositório central, puxando as modificações realizadas para a sua máquina.

GIT



GITHUB



- O Github é um repositório remoto do GIT;
- Funciona como um servidor que agrega todas as modificações realizadas, unificando as diferentes versões de código e seus históricos;
- Permite compartilhamento com equipes;
- Além disso, o Github se tornou a principal rede social para o desenvolvimento de software.

NA PRÁTICA



- Baixe o GIT no git-scm.com;
- Abra seu terminal e digite:

```
git config --global user.name "Nicolly Vieira"  
git config --global user.email "Nicolly.costa@aluno.cefet-rj.br"
```

- Crie uma pasta onde ficará o seu repositório e inicie o GIT:

```
C:\Users\nyvic\Documents\GIT>git init  
Initialized empty Git repository in C:/Users/nyvic/Documents/GIT/.git/
```


NA PRÁTICA

- Crie seu projeto e faça o commit dele:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="pt-br">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Fábrica de Software</title>
8  </head>
9  <body>
10     <h1>Olá Mundo!</h1>
11
12 </body>
13 </html>
```

```
C:\Users\nyvic\Documents\GIT>git status
On branch master

No commits yet

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

NA PRÁTICA

```
C:\Users\nyvic\Documents\GIT>git add index.html

C:\Users\nyvic\Documents\GIT>git commit -m "add index"
[master (root-commit) 0c02de9] add index
1 file changed, 13 insertions(+)
create mode 100644 index.html
```

- Para acessar o histórico de commits:

```
C:\Users\nyvic\Documents\GIT>git log
commit 0c02de91ec5879c1dbbd221a6da8b0b037609f63 (HEAD -> master)
Author: Nicolly Vieira <Nicolly.costa@aluno.cefet-rj.br>
Date: Mon Apr 26 17:23:25 2021 -0300

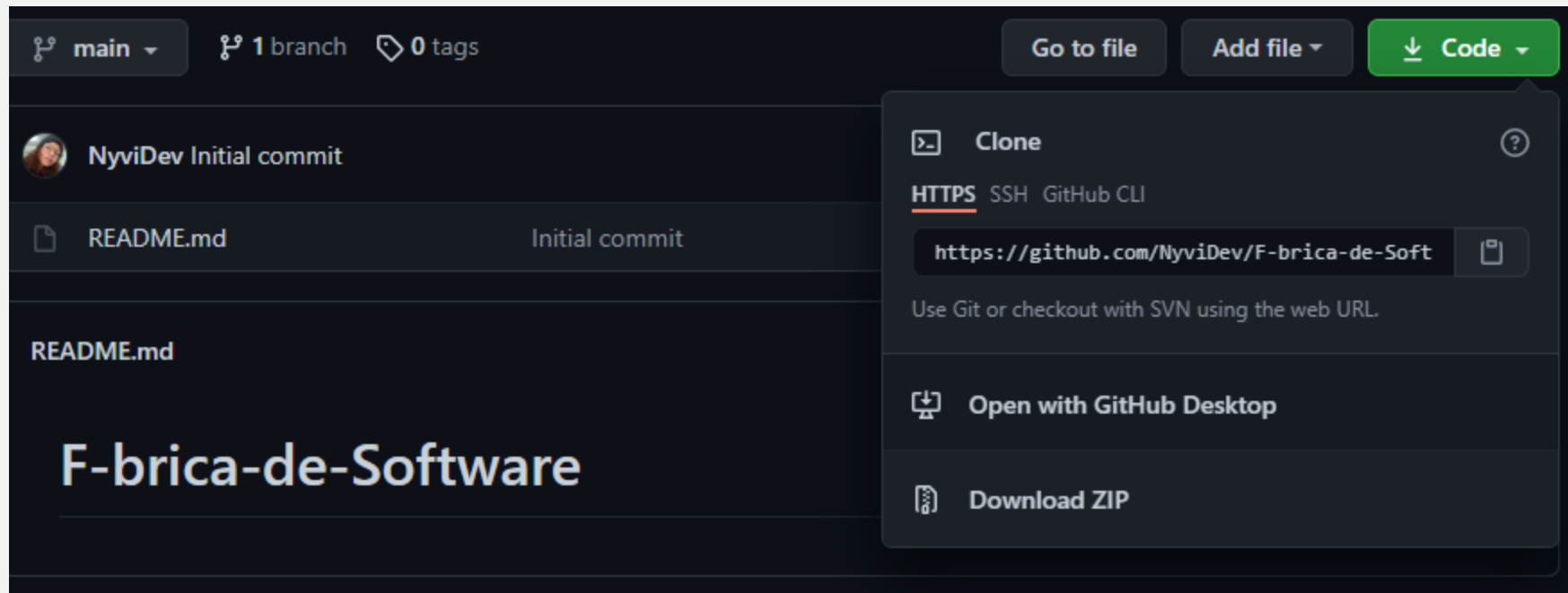
    add index
```

- Para rever os commits anteriores basta digitar “git show + nome do commit”;

NA PRÁTICA

Agora, iremos mostrar como usar o github como repositório remoto:

- Crie uma conta no github.com;
- Crie um repositório e copie o link;



NA PRÁTICA

- Voltando para o terminal:

```
C:\Users\nyvic\Documents\GIT>git remote add origin https://github.com/NyviDev/F-brica-de-Software.git
```



```
C:\Users\nyvic\Documents\GIT>git push -u origin master  
info: please complete authentication in your browser...
```

- Com o push, agora temos o nosso projeto na nuvem.



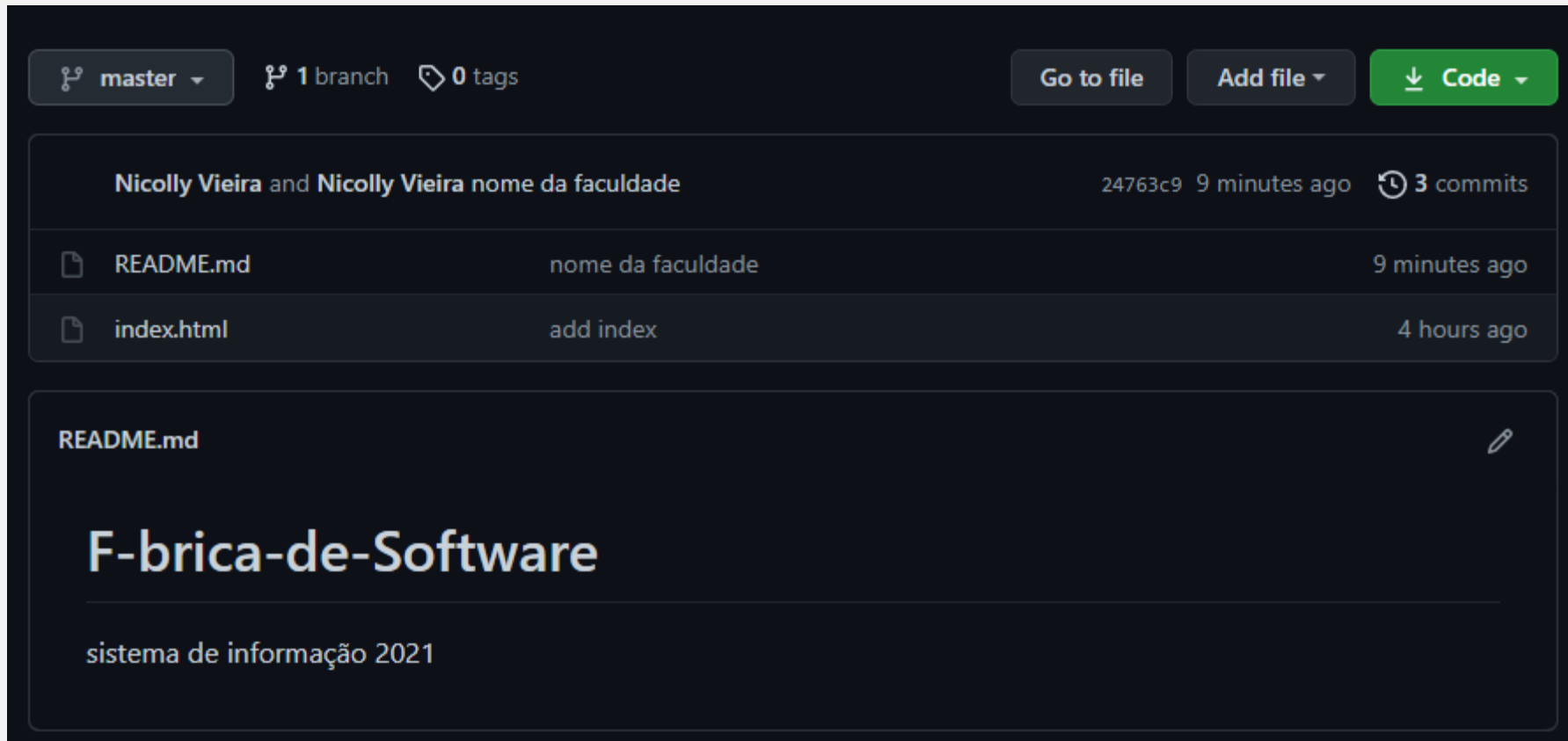
This branch is 2 commits ahead, 1 commit behind main. [Pull request](#) [Compare](#)

 **NyviDev** Create README.md 14555b7 2 minutes ago ⌚ 2 commits

 README.md	Create README.md	2 minutes ago
 index.html	add index	4 hours ago

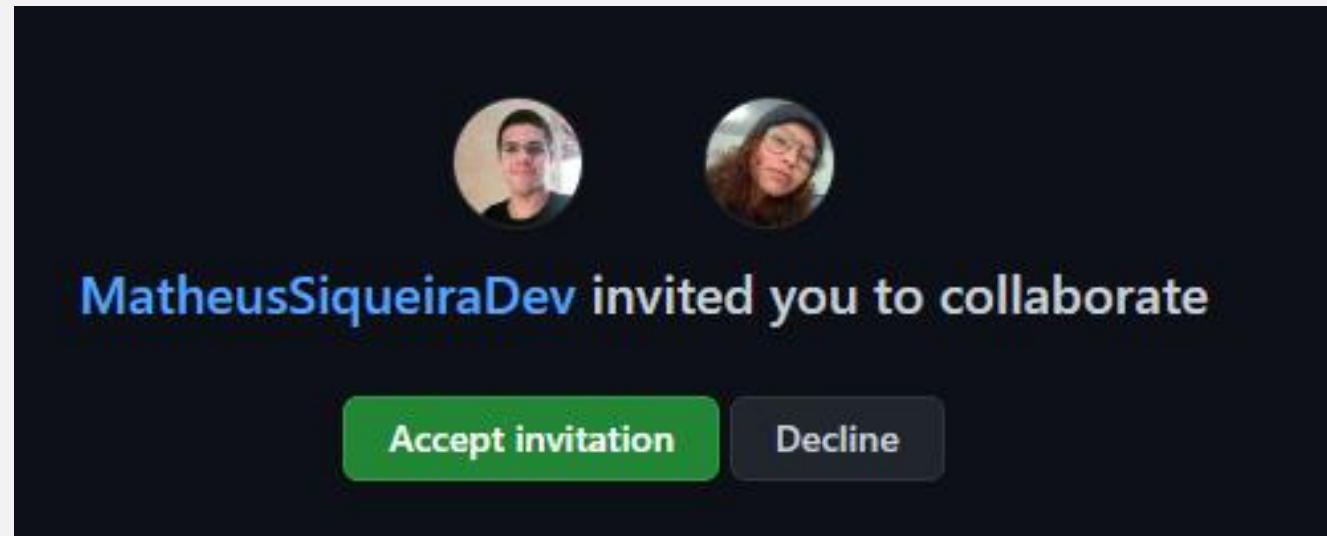
NA PRÁTICA

- Tela inicial de um repositório remoto:



NA PRÁTICA

- Agora, iremos ver como funciona se estivéssemos trabalhando em um projeto:
- Primeiro, o dono do repositório deverá te mandar um convite para ser colaborador.



NA PRÁTICA

- No terminal, digite git clone para clonar o repositório:

```
C:\Users\nyvic\Documents>git clone https://github.com/MatheusSiqueiraDev/Fabrica-de-Software.git
Cloning into 'Fabrica-de-Software'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

- Pronto, agora temos uma cópia na nossa máquina.
- Para mexer no projeto sem interferir no projeto, crie uma branch:

```
C:\Users\nyvic\Documents\Fabrica-de-Software>git branch modificacao

C:\Users\nyvic\Documents\Fabrica-de-Software>git branch
* master
  modificacao
```

NA PRÁTICA

- Com git checkout podemos mudar de branch:

```
C:\Users\nyvic\Documents\Fabrica-de-Software>git checkout modificacao
Switched to branch 'modificacao'

C:\Users\nyvic\Documents\Fabrica-de-Software>git status
On branch modificacao
nothing to commit, working tree clean
```

- Dessa forma não iremos modificar o código sem termos certeza de que queremos fazer isso.

NA PRÁTICA

- Esse é o projeto que copiamos e iremos modifica-lo.

```
index.html X
index.html > html
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Fábrica de Software</title>
8  </head>
9  <body>
10     <h1>Fábrica de Software</h1>
11     <p>CEFET</p>
12 </body>
13 </html>
```

NA PRÁTICA

- Adicionamos mais uma informação na linha 11.

```
index.html M X
index.html > html > body > p
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Fábrica de Software</title>
8  </head>
9  <body>
10     <h1>Fábrica de Software</h1>
11     <p>Sistema de Informação - CEFET/NF</p>
12 </body>
13 </html>
```

NA PRÁTICA

- Adicione e commite.

```
C:\Users\nyvic\Documents\Fabrica-de-Software>git status
On branch modificacao
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:   index.html

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

C:\Users\nyvic\Documents\Fabrica-de-Software>git add index.html

C:\Users\nyvic\Documents\Fabrica-de-Software>git commit -m "Segunda versão"
[modificacao b14fe99] Segunda versão
1 file changed, 1 insertion(+), 1 deletion(-)
```

NA PRÁTICA

- Veja que na minha branch Master, o projeto ainda é aquela primeira versão.

```
C:\Users\nyvic\Documents\Fabrica-de-Software>git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

- Quando tiver certeza de que acabou as modificações, use git merge (para isso deve estar na branch master):

```
C:\Users\nyvic\Documents\Fabrica-de-Software>git merge modificacao
Updating b5f699c..b14fe99
Fast-forward
 index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\nyvic\Documents\Fabrica-de-Software>
```

NA PRÁTICA

- Quando terminar dê git pull para exportar qualquer modificação no projeto de outro colaborador:

```
C:\Users\nyvic\Documents\Fabrica-de-Software>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 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), 318 bytes | 6.00 KiB/s, done.
From https://github.com/MatheusSiqueiraDev/Fabrica-de-Software
   b5f699c..359825b  master    -> origin/master
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

- Nesse caso, ocorreu um conflito. Significa que outra pessoa modificou ou adicionou mais linhas.

NA PRÁTICA

- O git mostra onde está o conflito e com um editor fica ainda mais claro.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Fábrica de Software</title>
8  </head>
9  <body>
10     <h1>Fábrica de Software</h1>
    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
11  <<<<<< HEAD (Current Change)
12     <p>Sistema de Informação - CEFET/NF</p>
13  =====
14  <ul>
15     <li>Menu</li>
16     <li>Menu</li>
17     <li>Menu</li>
18  </ul>
19  <p>CEFET</p>
20  >>>>>> 359825ba32c30a6c85046ec0ceb8860bc5076007 (Incoming Change)
21  </body>
22  </html>
```

NA PRÁTICA

- Accept Current Change: Ele irá escolher apenas a sua modificação.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Fábrica de Software</title>
8  </head>
9  <body>
10     <h1>Fábrica de Software</h1>
11     <p>Sistema de Informação - CEFET/NF</p>
12 </body>
13 </html>
```

NA PRÁTICA

- Accept Incoming Change: Ele irá escolher apenas a modificação do colaborador.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Fábrica de Software</title>
8  </head>
9  <body>
10     <h1>Fábrica de Software</h1>
11     <ul>
12         <li>Menu</li>
13         <li>Menu</li>
14         <li>Menu</li>
15     </ul>
16     <p>CEFET</p>
17 </body>
18 </html>
```


NA PRÁTICA

- Accept Both Changes: Ele irá escolher as duas modificações.

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Fábrica de Software</title>
8 </head>
9 <body>
10     <h1>Fábrica de Software</h1>
11     <p>Sistema de Informação - CEFET/NF</p>
12     <ul>
13         <li>Menu</li>
14         <li>Menu</li>
15         <li>Menu</li>
16     </ul>
17     <p>CEFET</p>
18 </body>
19 </html>
```

NA PRÁTICA

- Compare Changes: Mostra os dois projetos. Apenas leitura.

<pre>1 <!DOCTYPE html> 2 <html lang="pt-br"> 3 <head> 4 <meta charset="UTF-8"> 5 <meta http-equiv="X-UA-Compatible" 6 content="IE=edge"> 7 <meta name="viewport" 8 content="width=device-width, initial-scale=1. 9 0"> 10 <title>Fábrica de Software</title> 11 </head> 12 <body> 13 <h1>Fábrica de Software</h1> 14 <p>Sistema de Informação - CEFET/NF</p> 15 </body> 16 </html></pre>	<pre>1 <!DOCTYPE html> 2 <html lang="pt-br"> 3 <head> 4 <meta charset="UTF-8"> 5 <meta http-equiv="X-UA-Compatible" 6 content="IE=edge"> 7 <meta name="viewport" 8 content="width=device-width, initial-scale=1. 9 0"> 10 <title>Fábrica de Software</title> 11 </head> 12 <body> 13 <h1>Fábrica de Software</h1> 14 15 Menu 16 Menu 17 Menu 18 19 <p>CEFET</p> 20 </body> 21 </html></pre>
--	---

NA PRÁTICA

- Adicione e commite.

```
Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   index.html

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

C:\Users\nyvic\Documents\Fabrica-de-Software>git add index.html

C:\Users\nyvic\Documents\Fabrica-de-Software>git commit -m "versão unida"
[master 3d28ac6] versão unida
```

- Pronto, agora temos uma nova versão do projeto. Por isso o pull é muito importante depois que terminar de atualizar o projeto, isso faz com que não tenha duas versões diferentes no repositório remoto.

NA PRÁTICA

- Por último, dê git push para exportar para o repositório remoto.

```
C:\Users\nyvic\Documents\Fabrica-de-Software>git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 695 bytes | 63.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/MatheusSiqueiraDev/Fabrica-de-Software.git
   359825b..3d28ac6  master -> master
```

NA PRÁTICA

- Por último, iremos aprender a como criar versões do projeto.

```
C:\Users\nyvic\Documents\Fabrica-de-Software>git tag -a 1.0.0 -m "Primeira versao"

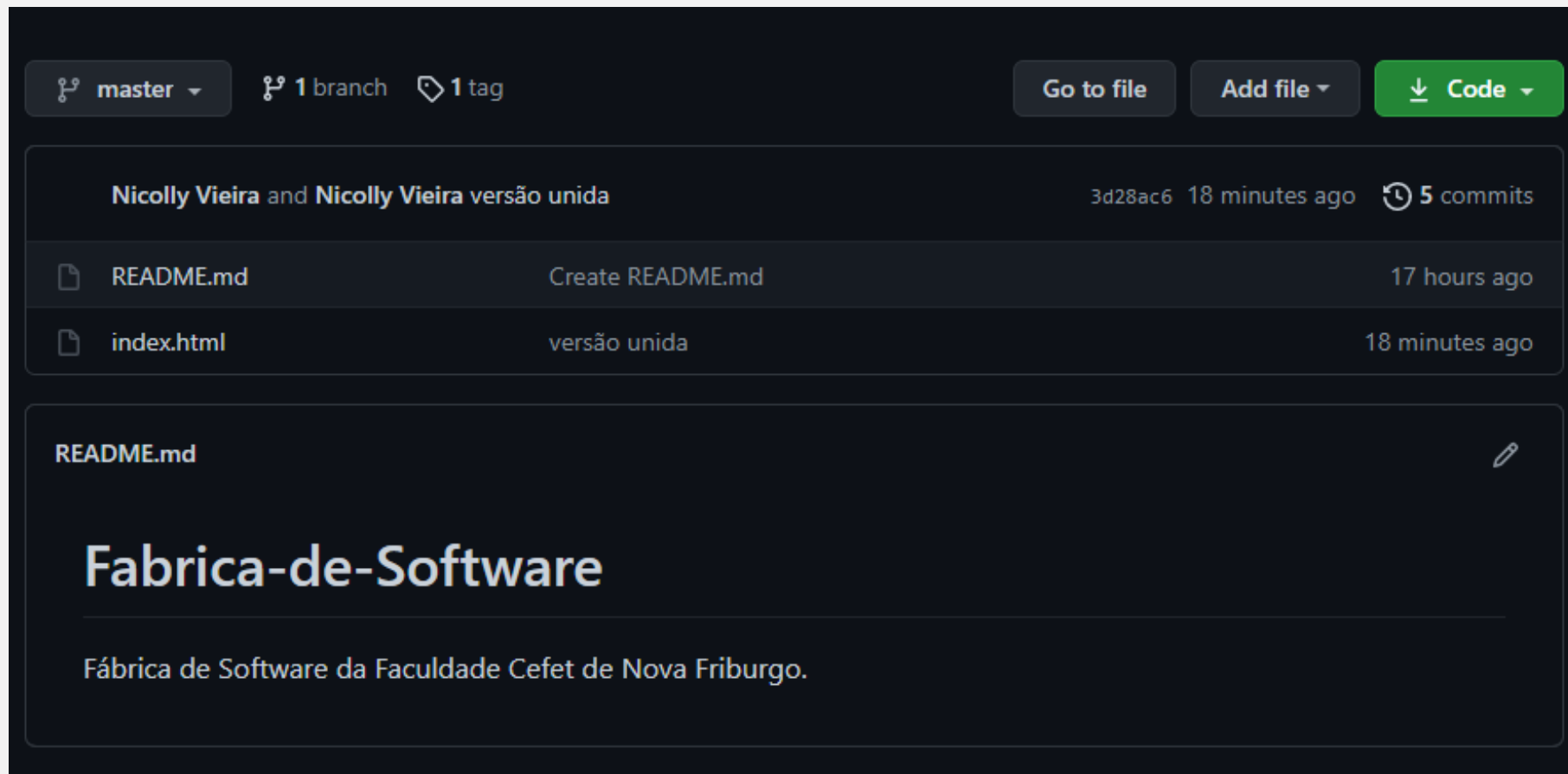
C:\Users\nyvic\Documents\Fabrica-de-Software>git tag
1.0.0
```

- Dê git push pras tags criadas também.

```
C:\Users\nyvic\Documents\Fabrica-de-Software>git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 176 bytes | 88.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MatheusSiqueiraDev/Fabrica-de-Software.git
 * [new tag]          1.0.0 -> 1.0.0
```

NA PRÁTICA

- No github está mostrando que temos uma versão do nosso projeto. Isso é importante para atualizar projetos que já estão sendo usados.



EXERCÍCIO

Formem duplas e sigam o passo a passo do exercício:

1. Crie um repositório remoto;
2. Adicione uma pessoa no projeto;
3. Crie um arquivo em python contendo uma função de soma com 2 variáveis (**n1** e **n2**) e faça retornar a soma, depois adicione no repositório;
4. Façam o clone do repositório;
5. A **pessoa A**, irá escrever a soma de **n1** e **n2**. A **pessoa B**, fará a soma de **n2** e **n1**;
6. A **pessoa A** deve ser a primeira a dar o push para que gere um conflito para a **pessoa B**;
7. Enquanto a **pessoa B** resolve o conflito, a **pessoa A** adicionará mais uma variável para função (**n3**).
8. Terminando a resolução do conflito, a **pessoa B** deverá dar o push para que gere um conflito para a **pessoa A**.
9. **Pessoa A** resolve o conflito e adicione no repositório.

OBS: O conflito deve ser resolvido com a sua versão do código.

GIT E GITHUB

Nicolly Vieira Santos Costa

