



VERSIONAMENTO DE CÓDIGO COM O GITHUB

José Dario Pintor da Silva



INTRODUÇÃO

- Durante o processo de desenvolvimento de um software várias pessoas podem estar envolvidas no desenvolvimento de uma versão deste software;
- Sendo assim, como saber qual a versão mais atual do código na hora de subir para o servidor? Quem foi a pessoa que fez a mudança e porque fez? Outra coisa é, se durante um *deploy* algo dar errado, como ter acesso a versão anterior estável?
- Um sistema de **controle de versões de código** é uma ferramenta de gerenciar mudanças em arquivos, incluindo textos e imagens. Através do versionamento de código é possível saber sempre que uma alteração for realizada, quem a fez e o porquê.

GIT

- Existem vários sistemas para versionamento de código. O git é o mais popular.
- Com ele podemos:
 - Criar histórico de alterações de código;
 - Trabalhar simultaneamente com outros *stakeholders* em um mesmo projeto;
 - Manter o código organizado com a criação de históricos e funcionalidades;
 - Gerar backups automáticos;

GIT

- O Git foi inicialmente projetado e desenvolvido por Linus Torvalds para o desenvolvimento do kernel Linux, mas foi adotado por muitos outros projetos;
- O Git é um software livre, distribuído sob os termos da versão 2 da GNU General Public License. Sua manutenção é atualmente supervisionada por Junio Hamano;
- Servidores para hospedagem gratuita de código fonte para repositório Git:
 - [GitHub](#);
 - Google Code;
 - GitLab;
 - entre outros.

GITHUB

- GitHub é uma plataforma de hospedagem de código-fonte e arquivos *online* com controle de versão para o **Git**;
- O GitHub é mundialmente usado e chega a ter mais de 36 milhões de usuários ativos contribuindo em projetos comerciais ou pessoais;
- Em 4 de junho de 2018, a Microsoft anunciou a compra da plataforma por US\$ 7,5 bilhões;

GITHUB

- Gratuito para projetos públicos porém tem a versão paga para projetos privados;
- Indicado para hospedagem de códigos pessoais ou projetos open source;
- Ótima para projetos colaborativos;
- É muito utilizado por programadores para hospedar projetos que servirão como vitrine de suas habilidades no CV. Assim, os recrutadores podem ver na prática as habilidades do candidato;

INSTALANDO O GIT

INSTALANDO O GIT NO LINUX

- No terminal digitar:

```
sudo apt-get install git
```

- Para saber se foi instalando digitar:

```
git --version
```


INSTALANDO O GIT NO WINDOWS

- Ir no site do git e fazer o download do executável e seguir os passos de instalação.

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

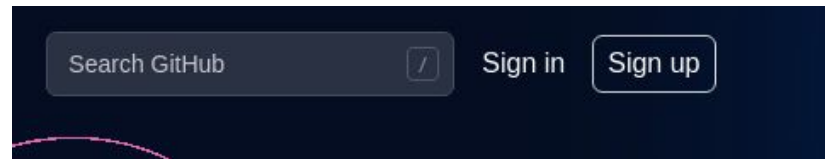
- Para saber se foi instalando digitar no terminal:

```
git --version
```

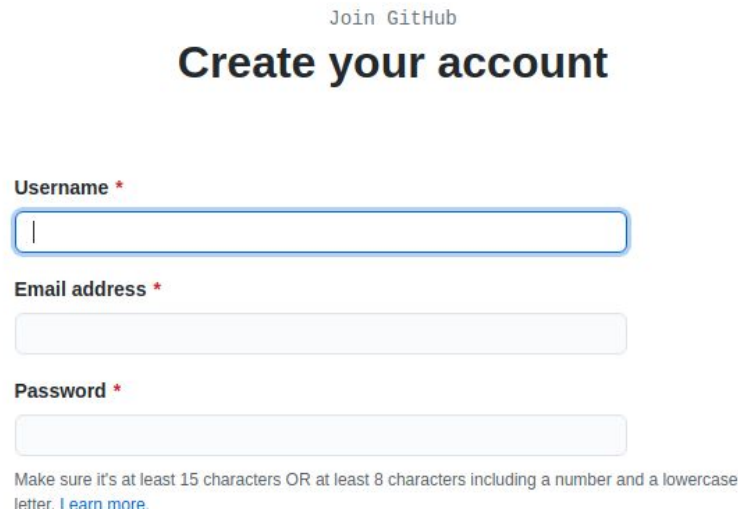
CRIANDO UMA CONTA NO GITHUB

CRIANDO UMA CONTA NO GITHUB

- Acessar o site github.com.
E clicar em **Sign up**



- Seguir os passos

A screenshot of the GitHub account creation page. At the top, it says 'Join GitHub' in small text, followed by 'Create your account' in large, bold text. Below this are three input fields: 'Username *', 'Email address *', and 'Password *'. Each field has a light blue border and a small vertical line indicating the cursor position. Below the 'Password *' field, there is a line of text: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)' A red arrow points from the text 'Seguir os passos' in the list to the 'Username' input field.

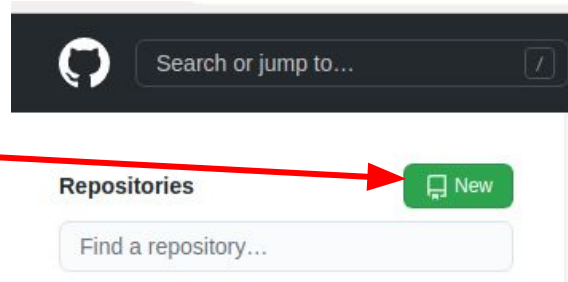


CRIANDO UM REPOSITÓRIO NO GITHUB





CRIANDO UM REPO NO GITHUB

- Após ter criado a conta e estar logado clicar no botão New repository;
- Dar nome ao repositório;





Owner * Repository name *

 dariopintor / repoTeste 

Great repository names are short and memorable. Need inspiration? How about [supreme-disco?](#)

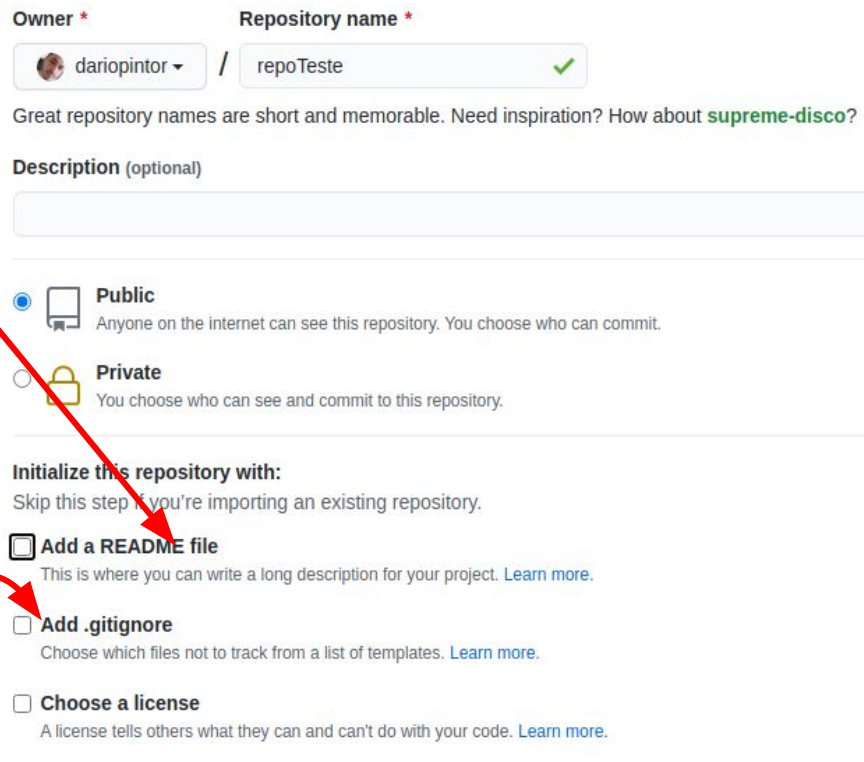
Description (optional)

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

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

CRIANDO UM REPO NO GITHUB

- Arquivo **README** serve como uma descrição mais detalhada do projeto em questão;
- O **.gitignore** é um arquivo onde você pode especificar alguns arquivos que você não quer que seja feito versionamento. Geralmente arquivos de configuração que são comuns a todos os projetos.



The screenshot shows the GitHub repository creation interface. At the top, the 'Owner' is set to 'dariopintor' and the 'Repository name' is 'repoTeste'. Below this, there is a description field and a section for repository visibility. The 'Public' option is selected. Under 'Initialize this repository with:', three options are listed: 'Add a README file', 'Add .gitignore', and 'Choose a license'. Two red arrows originate from the text on the left: one points to the 'Add a README file' checkbox, and the other points to the 'Add .gitignore' checkbox.

Owner * / Repository name *

Great repository names are short and memorable. Need inspiration? How about [supreme-disco?](#)

Description (optional)

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

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

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

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

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

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



COMANDOS DO GIT




GIT INIT

- Serve para inicializar um diretório com metadados de um projeto git;
- Após ser executado um diretório oculto .git é criado com informações do projeto;
- Primeiro comando a ser executado ao se criar um projeto Git;

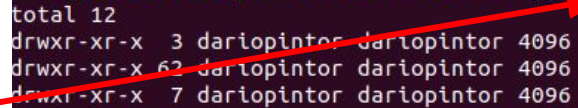

GIT INIT

- Crie um diretório (pasta) do projeto (pode ser pelo terminal ou interface);
- Pelo terminal acesso a pasta criada e digite o comando
 - `git init`
- Exiba o conteúdo pelo terminal para certificar se o diretório `.git` foi criado
 - `ls -la` (linux)
 - `dir /ah` (window)

```
darlopintor@darlopintor-Lenovo-G50-80:~$ mkdir git
```



```
darlopintor@darlopintor-Lenovo-G50-80:~$ cd git
darlopintor@darlopintor-Lenovo-G50-80:~/git$ git init
Initialized empty Git repository in /home/darlopintor/git/.git/
darlopintor@darlopintor-Lenovo-G50-80:~/git$ ls -la
total 12
drwxr-xr-x  3 darlopintor darlopintor 4096 jun 14 11:18 .
drwxr-xr-x 62 darlopintor darlopintor 4096 jun 14 11:17 ..
drwxr-xr-x  7 darlopintor darlopintor 4096 jun 14 11:18 .git
```



GIT STATUS

- Serve para informar os detalhes do repositório;
- São exibidos detalhes como novos arquivos, arquivos modificados ou excluídos;
- Também são exibidas informações sobre arquivos pendentes de commit
- no terminal digite
 - `git status`

```
dariopintor@dariopintor-Lenovo-G50-80:~/git$ git status
On branch master

No commits yet

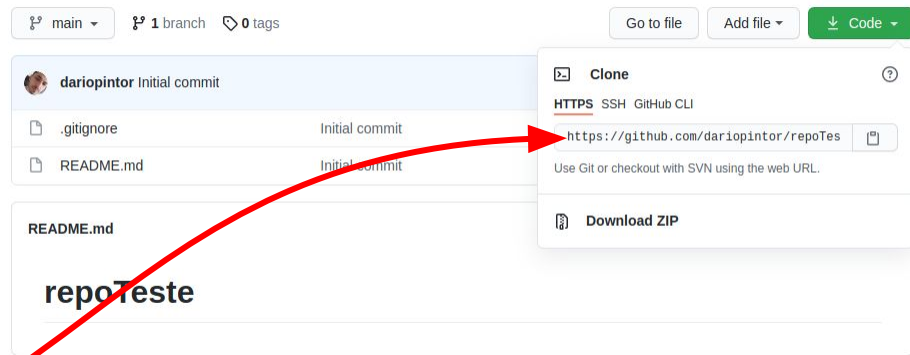
nothing to commit (create/copy files and use "git add" to track)
dariopintor@dariopintor-Lenovo-G50-80:~/git$
```

GIT REMOTE

- Serve para associar um repositório remoto com nosso projeto local;
- Utilizamos a opção 'add origin URL' para informar o repositório a qual a associação será realizada;

1. Passos

- 1.1. Acessar a página do repositório criado no github e copiar a url do mesmo,

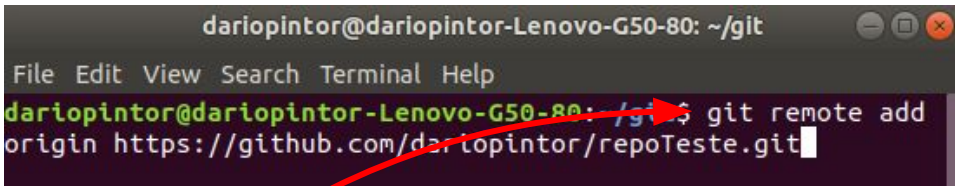


GIT REMOTE

1. Passos

- 1.1. Acessar a página do repositório no github e copiar a url do mesmo;
- 1.2. No terminal associe o repositório criado no github ao projeto inicializado na máquina local com o comando abaixo:

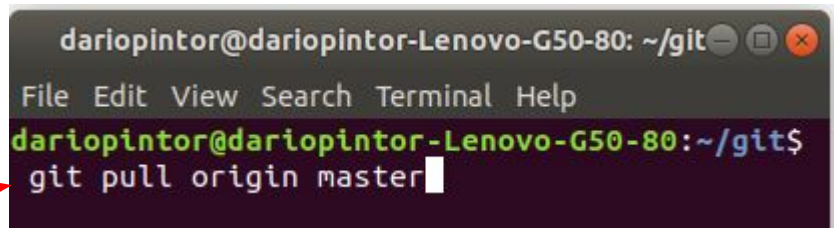
```
git remote add origin  
https://github.com/dariopinto  
r/repoTeste.git
```

A screenshot of a terminal window with a dark background. The title bar at the top reads 'dariopintor@dariopintor-Lenovo-G50-80: ~/git'. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows a prompt 'dariopintor@dariopintor-Lenovo-G50-80: ~/git\$' followed by the command 'git remote add origin https://github.com/dariopintor/repoTeste.git'. A red arrow points from the URL in the text block to the command in the terminal.

```
dariopintor@dariopintor-Lenovo-G50-80: ~/git  
File Edit View Search Terminal Help  
dariopintor@dariopintor-Lenovo-G50-80: ~/git$ git remote add  
origin https://github.com/dariopintor/repoTeste.git
```

GIT PULL

- Serve para baixar os arquivos atualizados do repositório remoto.
- Deve ser utilizado em conjunto com a origem desejada, que em nosso caso será 'origin master' (mas poderia ser uma branch secundária)
- No terminal digite
 - `git pull origin master`



```
dariopintor@dariopintor-Lenovo-G50-80: ~/git
File Edit View Search Terminal Help
dariopintor@dariopintor-Lenovo-G50-80:~/git$
git pull origin master
```

A screenshot of a terminal window with a dark background. The title bar shows the user 'dariopintor' on a machine named 'dariopintor-Lenovo-G50-80' in the directory '~/git'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The prompt is 'dariopintor@dariopintor-Lenovo-G50-80:~/git\$'. The command 'git pull origin master' has been entered and is followed by a white cursor. A red arrow points from the text 'git pull origin master' in the list above to this terminal window.

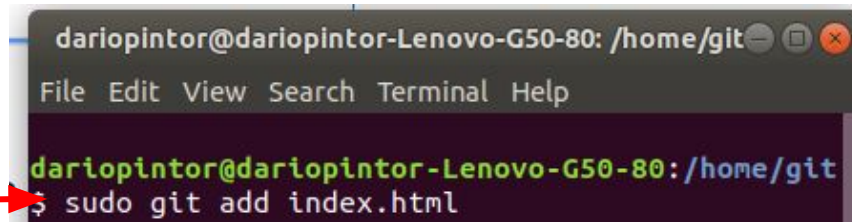
GIT ADD

- Serve para adicionar um ou mais **arquivos à uma fila do Git** para que o **commit** seja realizado posteriormente;
- Ele aceita um ou mais arquivos como parâmetro, separados por espaço. Caracteres especiais como ‘*’ podem ser utilizados para adicionar mais de um arquivo por nome ou extensão, como ‘*.html’, ‘arquivo*.txt’;
- É um comando que serve apenas para adicionar arquivos em uma lista que depois será utilizada para “upar” os arquivos pertencentes a essa lista com o comando **push**;

GIT ADD

Passos

- Crie um arquivo no seu repositório local;
- Dentro diretório que está o arquivo digite:
 - `git add index.html` para adicionar somente um arquivo ou;
 - `git add *` para adicionar à lista todos os arquivos que estão no diretório;
- Verifique o status do projeto `git status`

A screenshot of a terminal window with a dark background. The window title is 'dariopintor@dariopintor-Lenovo-G50-80: /home/git'. The menu bar shows 'File Edit View Search Terminal Help'. The prompt 'dariopintor@dariopintor-Lenovo-G50-80: /home/git' is followed by the command '\$ sudo git add index.html'. A red arrow points from the text 'para adicionar somente um arquivo ou;' in the list to the 'index.html' part of the command.

```
dariopintor@dariopintor-Lenovo-G50-80: /home/git
File Edit View Search Terminal Help
dariopintor@dariopintor-Lenovo-G50-80: /home/git
$ sudo git add index.html
```

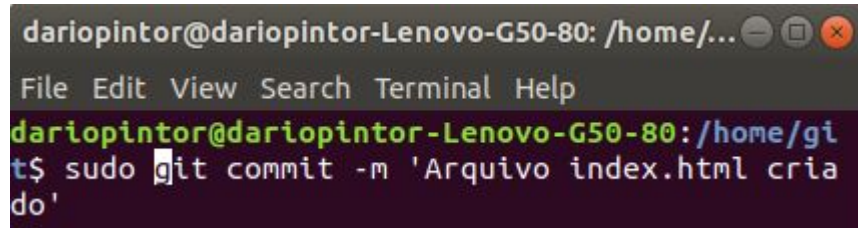
GIT COMMIT

- O comando ‘git commit’ serve para adicionar os arquivos, **adicionados na lista pelo add**, em repositório local que o Git mantém antes do código ser enviado para o servidor (git hub);
- Uma nova revisão e log são criados, porém o código continua somente sendo visível localmente;
- É o último passo antes do envio ao servidor remoto;
- Uma mensagem com a descrição da alteração é obrigatória, e deve ser informada com a utilização do parâmetro ‘-m’

GIT COMMIT

Passos

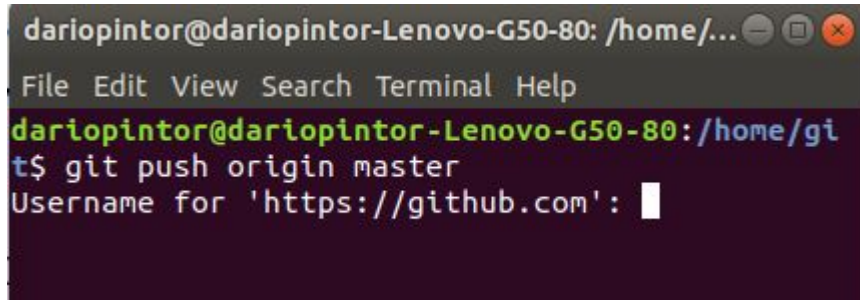
1. No terminal digite
 - `git commit -m 'Arquivo index.html criado'`
2. Verifique o status
 - `git status`
3. Verifique a revisão criada pelo commit
 - `git log`



```
dariopintor@dariopintor-Lenovo-G50-80: /home/...  
File Edit View Search Terminal Help  
dariopintor@dariopintor-Lenovo-G50-80: /home/gi  
t$ sudo git commit -m 'Arquivo index.html cria  
do'
```

GIT PUSH

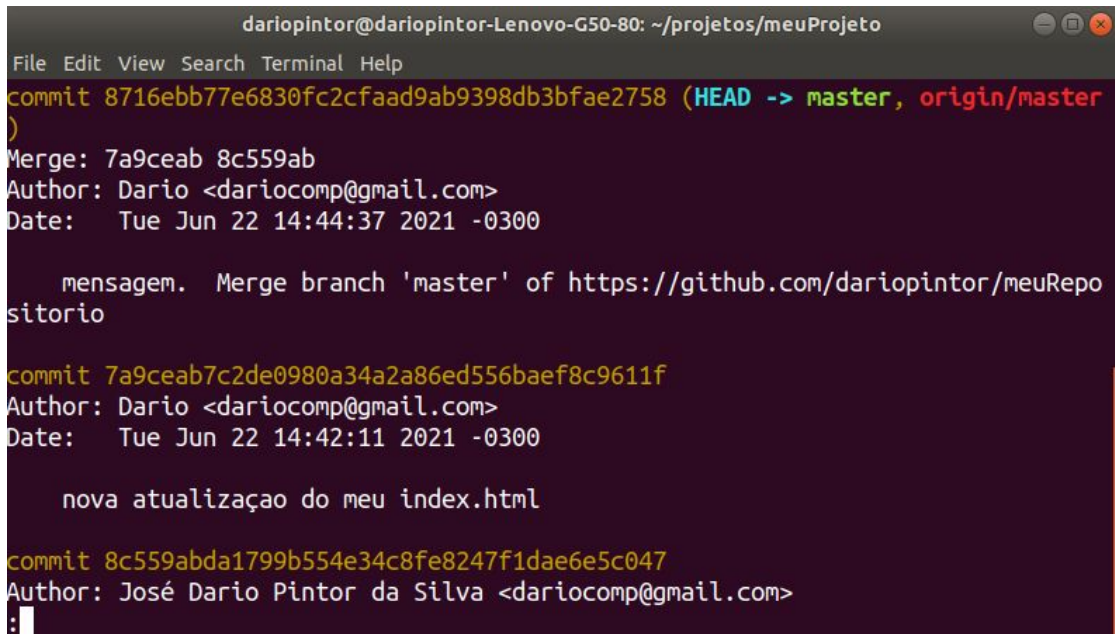
- Serve para efetivamente enviar nossas alterações para o repositório remoto;
- É necessário informar a origem com o parâmetro 'origin master', onde 'master' é a branch padrão do Git;
- No terminal digite
 - `git push origin master`
- Vai pedir o usuário e a senha do seu github;

A terminal window with a dark background and light-colored text. The window title is 'dariopintor@dariopintor-Lenovo-G50-80: /home/...'. The menu bar shows 'File Edit View Search Terminal Help'. The prompt is 'dariopintor@dariopintor-Lenovo-G50-80: /home/gi'. The command 't\$ git push origin master' has been entered. The next line shows the prompt 'Username for 'https://github.com':' followed by a white cursor block.

```
dariopintor@dariopintor-Lenovo-G50-80: /home/...  
File Edit View Search Terminal Help  
dariopintor@dariopintor-Lenovo-G50-80: /home/gi  
t$ git push origin master  
Username for 'https://github.com':
```

GIT LOG

- O comando 'git log' serve para exibir as revisões e informações dos commits realizados no repositório;
- Essa informação é útil para verificar a revisão atual, assim como pode ser utilizada para voltar o código a uma revisão anterior;



```
dariopintor@dariopintor-Lenovo-G50-80: ~/projetos/meuProjeto
File Edit View Search Terminal Help
commit 8716ebb77e6830fc2cfaad9ab9398db3bfae2758 (HEAD -> master, origin/master)
Merge: 7a9ceab 8c559ab
Author: Dario <dariocomp@gmail.com>
Date: Tue Jun 22 14:44:37 2021 -0300

    mensagem. Merge branch 'master' of https://github.com/dariopintor/meuRepositorio

commit 7a9ceab7c2de0980a34a2a86ed556baef8c9611f
Author: Dario <dariocomp@gmail.com>
Date: Tue Jun 22 14:42:11 2021 -0300

    nova atualizacao do meu index.html

commit 8c559abda1799b554e34c8fe8247f1dae6e5c047
Author: José Dario Pintor da Silva <dariocomp@gmail.com>
:
```

GIT DIFF

- O comando 'git diff' serve para comparar um arquivo em repositórios ou revisões distintas;
- Ele necessita de um parâmetro informando a revisão a ser comparada;
- Utilize 'HEAD' como parâmetro para comparar as alterações com a última revisão disponível;

A terminal window with a dark purple background and light green text. The title bar at the top reads 'dariopintor@dariopintor-Lenovo-G50-80: ~/projetos/meuProjeto'. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows the command 'git diff HEAD' being executed twice. The output shows a diff for 'index.html' between two commits, highlighting changes in the file content.

```
dariopintor@dariopintor-Lenovo-G50-80: ~/projetos/meuProjeto$ git diff HEAD
dariopintor@dariopintor-Lenovo-G50-80: ~/projetos/meuProjeto$ git diff HEAD
diff --git a/index.html b/index.html
index ba1f9c1..3512ca2 100644
--- a/index.html
+++ b/index.html
@@ -1,4 +1,4 @@
 # meuRepositorio
 video aula de versionamento de código

-nova atualização deste arquivo
+nova atualização deste arquivoddd
```



BRANCHES

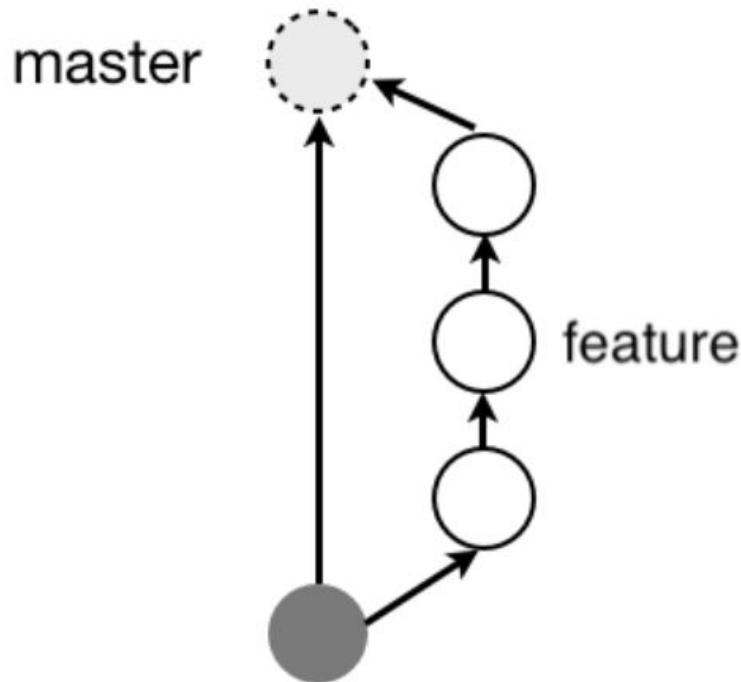


BRANCHES

- Digamos que temos um código no nosso GitHub que faz as quatro operações básicas em uma calculadora;
- A função soma já foi implementada e está funcionando corretamente (está na branch master); As outras funções foram divididas para dois programadores;
- Para evitar problemas de versões, cada programador pode criar uma “cópia” do projeto original (branch master), e então trabalhar nela adicionando as novas funções;
- Ao final é só mesclar o que foi feito (branch criada) com o que já tinha anteriormente (branch master).

BRANCHES

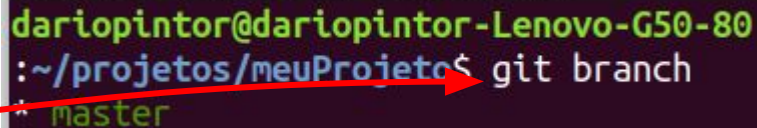
- A partir do momento que criamos uma nova branch podemos aplicar todos os comando git nela;
- Quando achamos que temos a funcionalidade funcionando corretamente podemos então aplicar as mudanças a branch principal master, usando o comando merge.



CRIANDO UM NOVA BRANCH

- Para saber em que branch estamos usamos o comando:

- `git branch:`

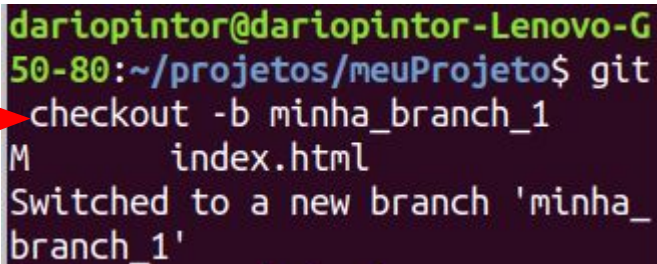


```
dariopintor@dariopintor-Lenovo-G50-80  
:~/projetos/meuProjeto$ git branch  
* master
```

A terminal window showing the command `git branch` being executed. The output is `* master`, indicating the current branch. A red arrow points from the text `git branch:` in the list to the `git branch` command in the terminal.

- Para criar uma nova branch usamos o comando

- `git checkout`
`[nome_da_branch]`




```
dariopintor@dariopintor-Lenovo-G  
50-80:~/projetos/meuProjeto$ git  
checkout -b minha_branch_1  
M      index.html  
Switched to a new branch 'minha_  
branch_1'
```

A terminal window showing the command `git checkout -b minha_branch_1` being executed. The output shows a file modification (M) for `index.html` and a message: `Switched to a new branch 'minha_branch_1'`. A red arrow points from the text `git checkout` in the list to the `git checkout` command in the terminal.

MUDANDO A BRANCH ATUAL

- Para mudar a branch em que estamos trabalhando basta usar o seguinte comando


- `git checkout [nome_branch]:`



```
dariopintor@dariopintor-Lenovo-G50-80:~/projetos/meuProjet  
o$ git checkout minha_branch_1  
M      index.html  
Switched to branch 'minha_branch_1'
```

MERGE A BRANCH MASTER

- Precisamos voltar a branch master
 - `git checkout [nome_branch]:`
- Então usamos o comando:
 - `git merge origin/[nome_branch]`



```
dariopintor@dariopintor-Lenovo-G50-80:~/projetos/neuProjet  
$ git checkout minha_branch_1  
M       index.html  
Switched to branch 'minha_branch_1'
```

REFERÊNCIAS

<https://pt.wikipedia.org/wiki/GitHub>

Material do curso “Aprenda Testes Unitários com Javascript” cedido gentilmente por Marcio Casale de Souza;