

Projeto Integrado III

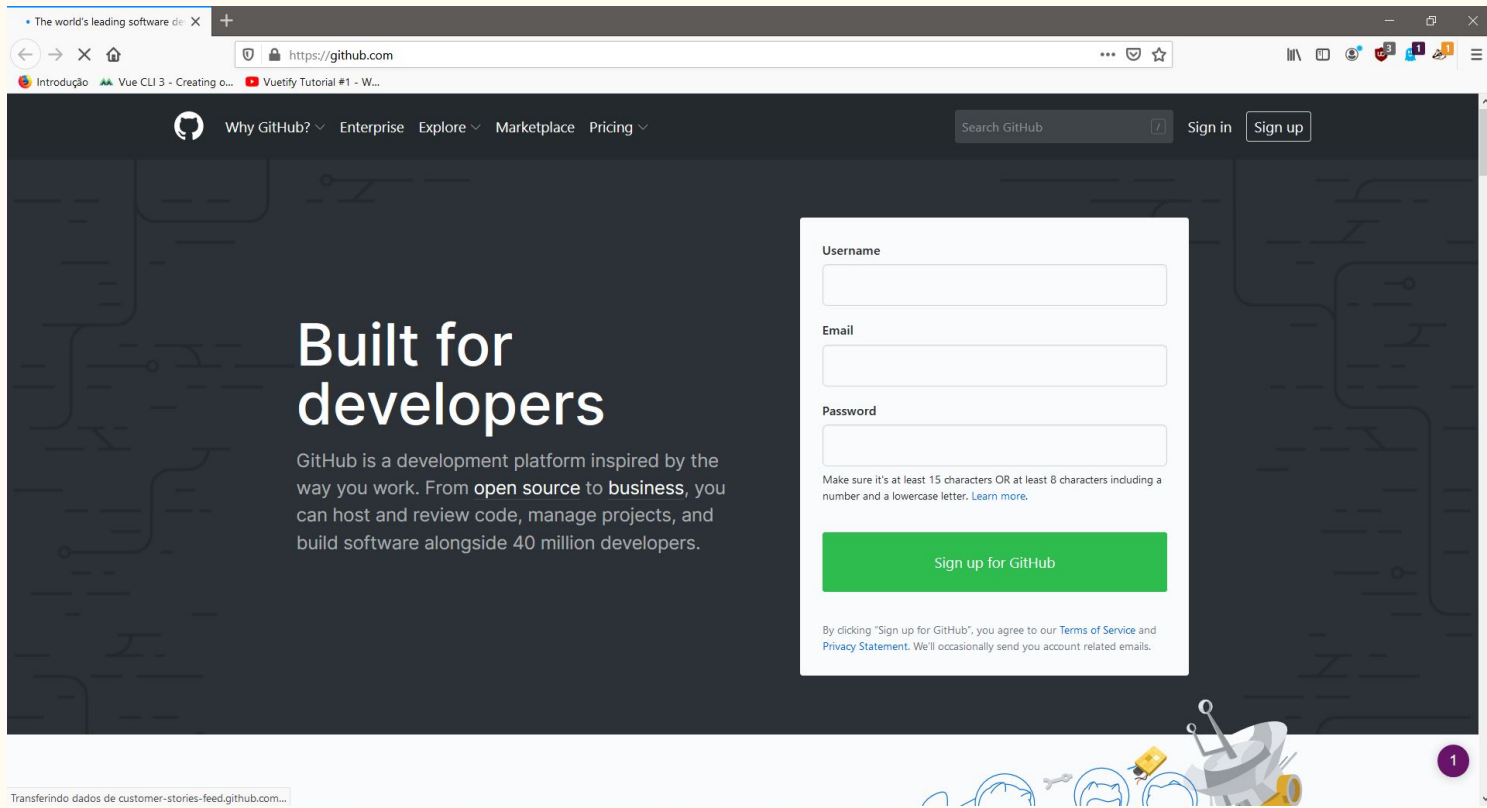
DD - UFC - Quixadá



Prof.: Aníbal Cavalcante

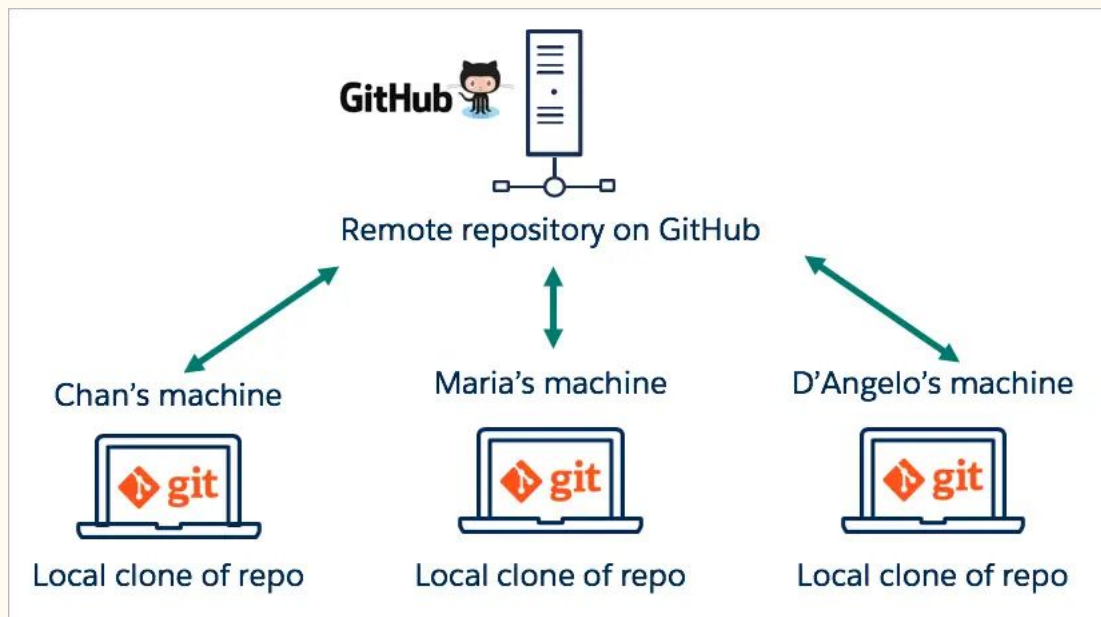
Acessando o GitHub e criando uma conta.

<https://github.com/>



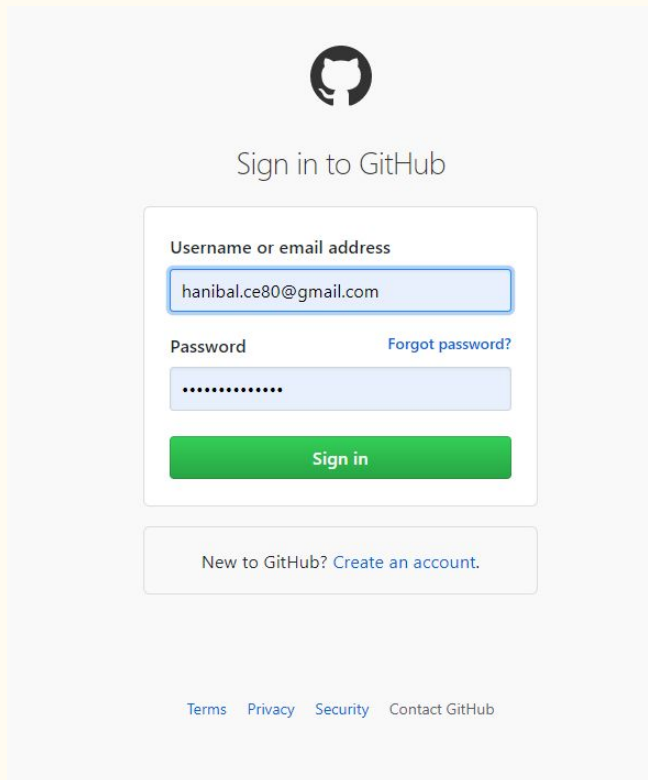
O Github

GitHub é uma plataforma de hospedagem de código-fonte com controle de versão usando o Git. Nele podemos criar projetos e versioná-los com o Git. Cada projeto terá um repositório remoto.



Vamos criar o repositório do nosso projeto fictício...

1 - Faça o login na sua conta: <https://github.com/login>



The image shows the GitHub login page. At the top is the GitHub logo (Octocat). Below it is the text "Sign in to GitHub". There are two input fields: "Username or email address" with the value "hanibal.ce80@gmail.com" and "Password" with masked characters ".....". To the right of the password field is a link "Forgot password?". Below the input fields is a green "Sign in" button. At the bottom of the form is a link "New to GitHub? Create an account.". At the very bottom of the page are links for "Terms", "Privacy", "Security", and "Contact GitHub".

Sign in to GitHub

Username or email address

hanibal.ce80@gmail.com

Password [Forgot password?](#)

.....

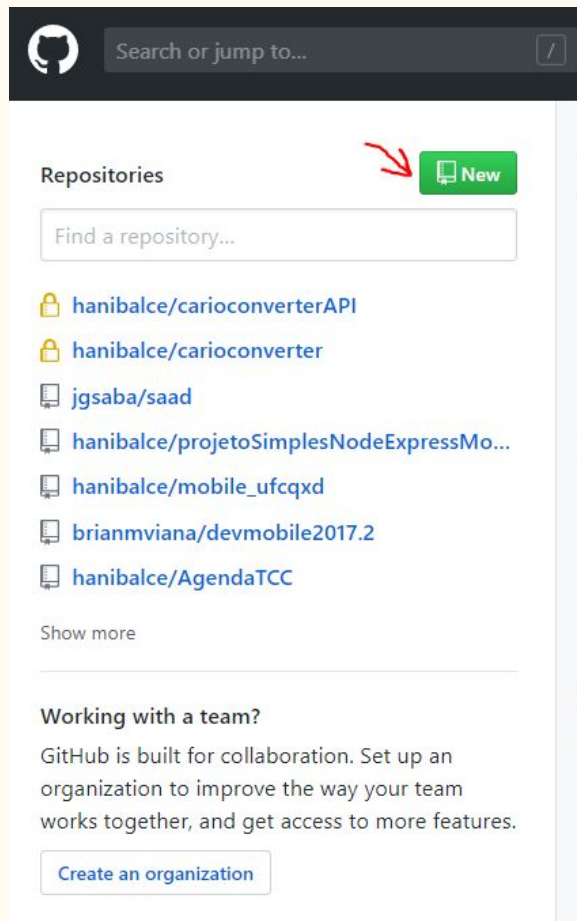
Sign in

New to GitHub? [Create an account.](#)

[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)

Vamos criar o repositório do nosso projeto fictício...

2 - Selecione a opção New.



Vamos criar o repositório do nosso projeto fictício...

3 - Preencha os dados do seu projeto.


1. Dê um nome ao repositório
2. Inicialize o README
3. Add .gitignore Node
4. Add GNU License v3.0
5. Clique em **Create Repository**

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

Repository name *


 hanibalce ▾

 /


projetoIntegrado3 ✓

Great repository names are short and memorable. Need inspiration? How about [special-octo-meme?](#)

Description (optional)

☒  Public

Anyone can see this repository. You choose who can commit.

☐  Private

You choose who can see and commit to this repository.

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

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: Node ▾

Add a license: GNU General Public License v3.0 ▾ ⓘ

Create repository

Vamos criar o repositório do nosso projeto fictício...

4 - O resultado deve ser esse.

The screenshot shows a GitHub repository page for 'hanibalce / projetoIntegrado3'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area starts with a description placeholder 'No description, website, or topics provided.' and an 'Edit' button. Below this is a 'Manage topics' link. A summary bar shows '1 commit', '1 branch', '0 packages', '0 releases', and '1 contributor'. Below the summary bar are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. The commit history table shows three initial commits for '.gitignore', 'LICENSE', and 'README.md', all made 1 minute ago. The 'README.md' file is expanded, showing the text 'projetoIntegrado3'.

hanibalce / projetoIntegrado3

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

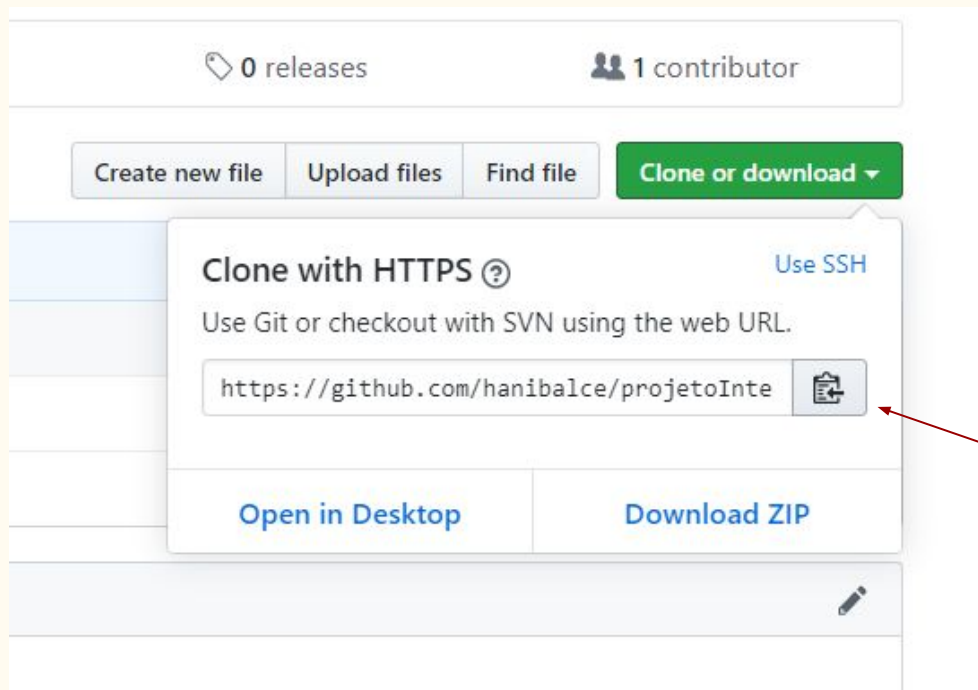
hanibalce Initial commit	Latest commit aff62e8 1 minute ago	
.gitignore	Initial commit	1 minute ago
LICENSE	Initial commit	1 minute ago
README.md	Initial commit	1 minute ago

README.md

projetoIntegrado3

Vamos clonar nosso repositório remoto para nosso repositório local

5 - Copie o link do repositório remoto.



Vamos clonar nosso repositório remoto para nosso repositório local

6 - Crie na sua máquina local uma pasta chamada repositórios e entre dentro dela.

```
PS C:\Users\Anibal\Desktop> mkdir repositorios
```

```
Diretório: C:\Users\Anibal\Desktop
```

Mode	LastWriteTime	Length	Name
----	-----	-----	----
d-----	03/03/2020 14:07		repositorios

```
PS C:\Users\Anibal\Desktop> cd .\repositorios\
```

```
PS C:\Users\Anibal\Desktop\repositorios>
```

Vamos clonar nosso repositório remoto para nosso repositório local

7 - Vamos clonar nosso repositório através do comando `git clone`.

```
PS C:\Users\Anibal\Desktop\repositorios> git clone https://github.com/hanibalce/projetoIntegrado3.git
Cloning into 'projetoIntegrado3'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), done.
PS C:\Users\Anibal\Desktop\repositorios>
```

Vamos clonar nosso repositório remoto para nosso repositório local

8 - Verifique se foi criado uma pasta com o nome do repositório. Use os comandos `ls` ou `dir` dependendo do S.O.

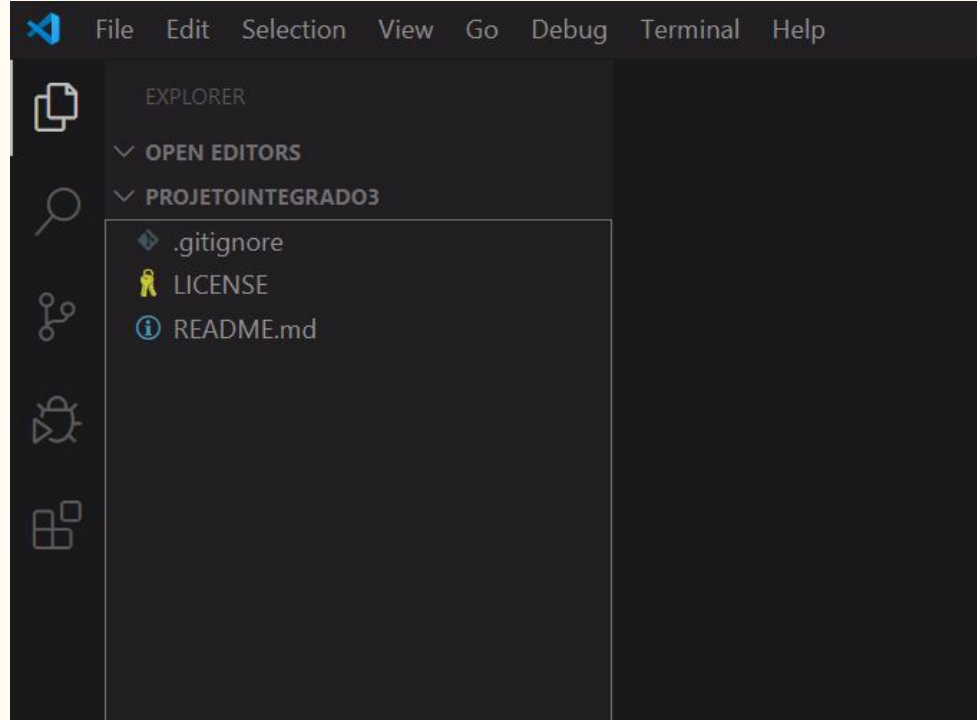
```
PS C:\Users\Anibal\Desktop\repositorios> ls
```

```
Diretório: C:\Users\Anibal\Desktop\repositorios
```

Mode	LastWriteTime	Length	Name
----	-----	-----	----
d-----	03/03/2020 14:10		projetoIntegrado3

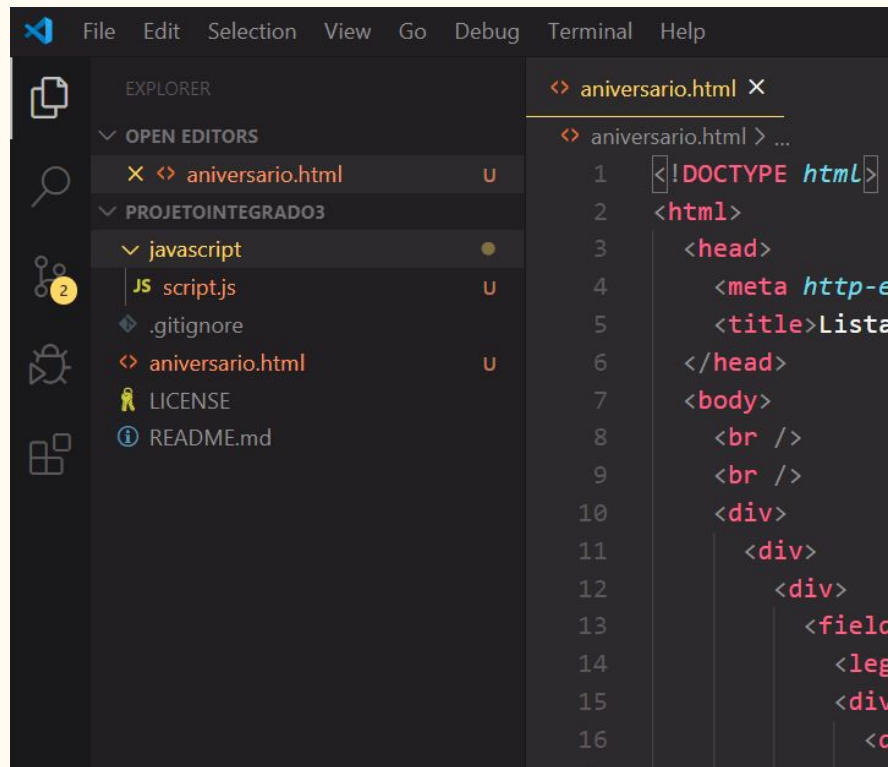
Vamos verificar os arquivos criados no repositório local.

9 - Abra o Visual Studio Code selecione a pasta do seu repositório.



Vamos adicionar novos arquivos ao nosso projeto.

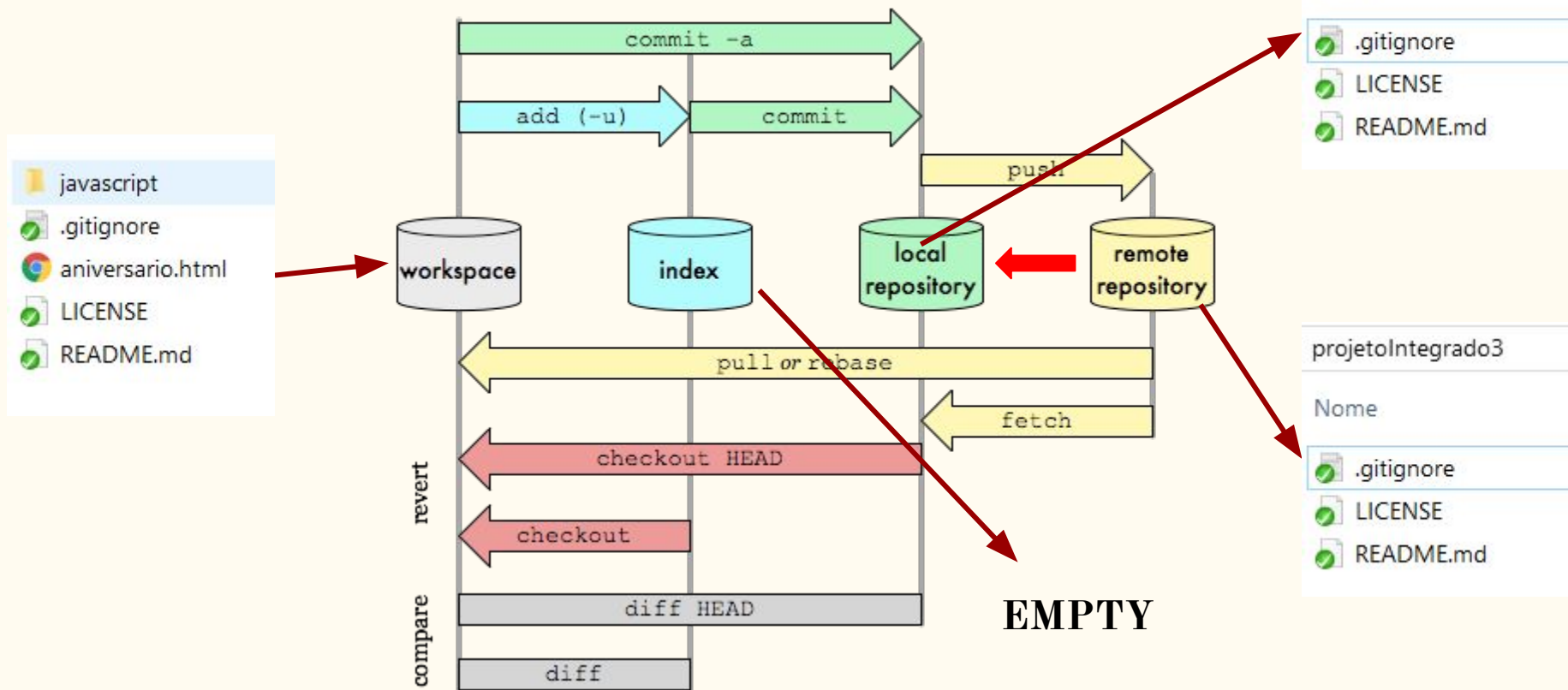
10 - Copie para dentro do repositório o arquivo **aniversario.html** e a pasta **javascript** com o arquivo **script.js**.



Relembrando o Fluxo de Trabalho do Git (Git Workflow)



Fluxo de Trabalho do Git (Git Workflow)



Vamos adicionar os arquivos a nossa Staging Area

11 - Vamos agora adicionar os novos arquivos a nossa área de testes com o comando `git add`.

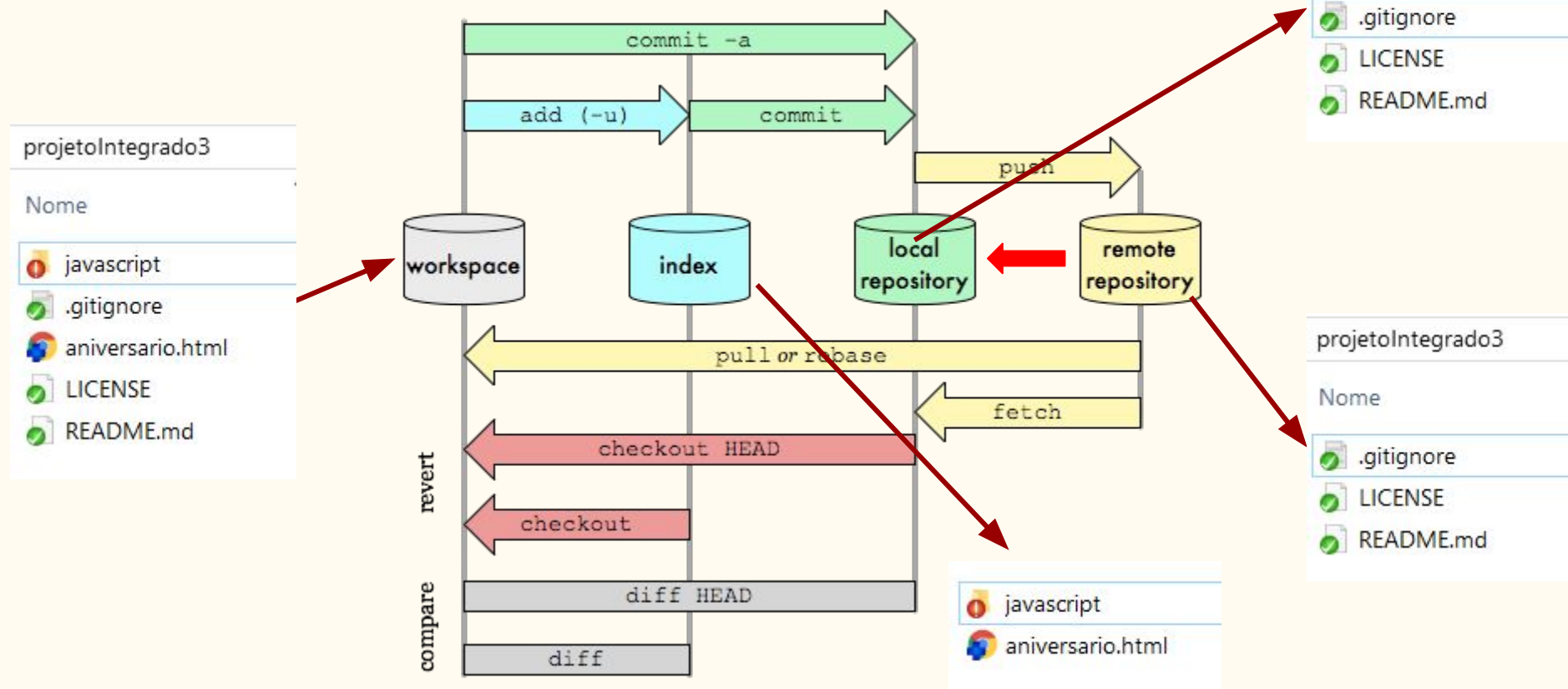
```
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> ls

Diretório: C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3

Mode                LastWriteTime         Length Name
----                -
d-----          03/03/2020    14:20             javascript
-a----          03/03/2020    14:10           1714 .gitignore
-a----          03/03/2020    14:26           1474 aniversario.html
-a----          03/03/2020    14:10          35823 LICENSE
-a----          03/03/2020    14:10            19 README.md

PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> git add .
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3>
```


Fluxo de Trabalho do Git (Git Workflow)



Vamos verificar se os arquivos realmente estão na área de testes.

12 - Vamos usar o comando `git status` para exibir os arquivos que estão na área de testes (Stagin Area).

```
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
```

```
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   aniversario.html
    new file:   javascript/script.js
```

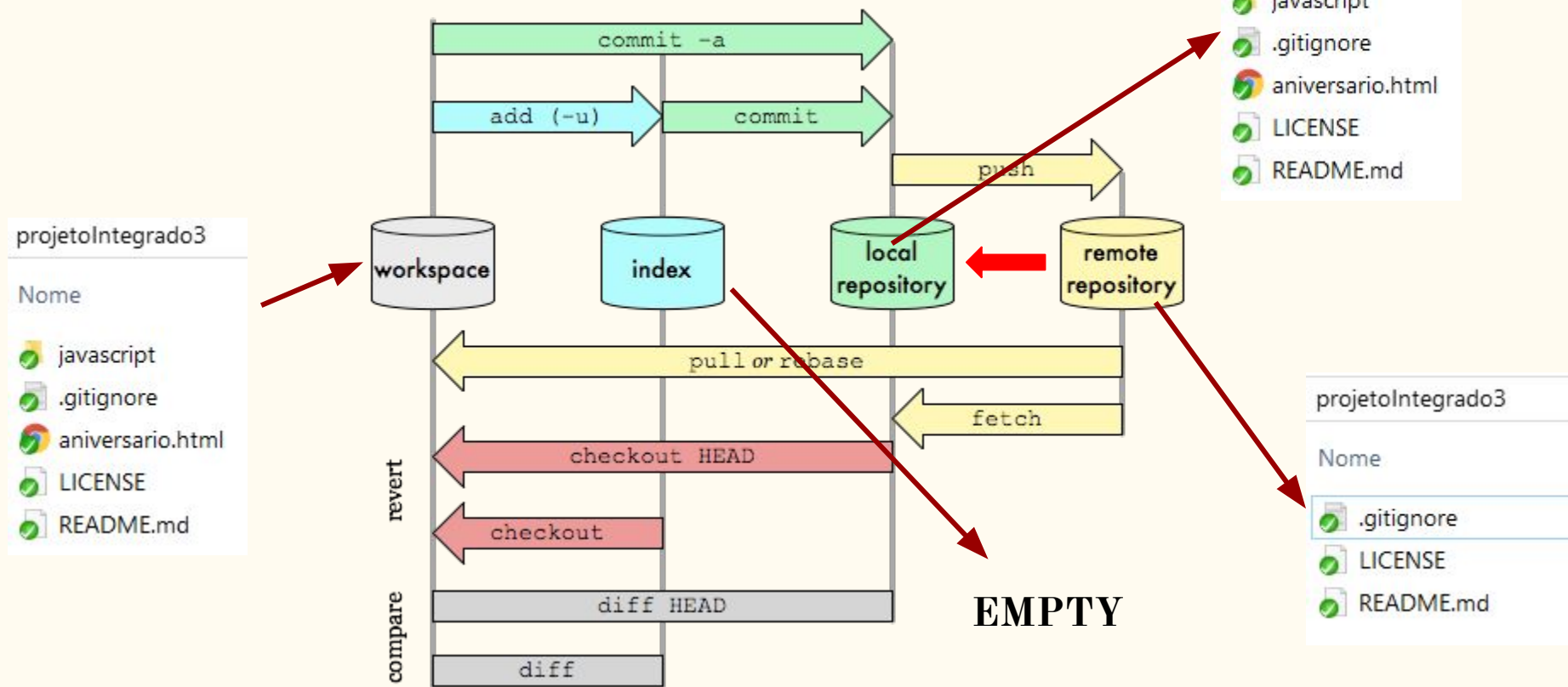
```
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3>
```

Vamos submeter os arquivos para nosso repositório local

13 - Agora vamos submeter os novos arquivos para o nosso repositório local através do comando `git commit -m "Commit Inicial"`.

```
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> git commit -m "Commit Inicial"
[master aae22bd] Commit Inicial
 2 files changed, 55 insertions(+)
 create mode 100644 aniversario.html
 create mode 100644 javascript/script.js
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3>
```

Fluxo de Trabalho do Git (Git Workflow)




Vamos enviar as alterações feitas para repositório remoto.


14 - Agora vamos deixar o repositório local igual ao repositório remoto, para isso vamos enviar todos os arquivos através do comando `git push`. Antes use o comando `git diff` para ver o que aparece.


```
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 1.12 KiB | 1.12 MiB/s, done.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/hanibalce/projetoIntegrado3.git
   aff62e8..eb11b69  master -> master
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3>
```


Vamos abrir o Github para verificar se o arquivos foram enviados.


15 - Abra seu projeto no Github.


 4 commits

 1 branch

 0 packages

 0 releases

 1 contributor

 GPL-3.0

Branch: master ▼


New pull request






Create new file

Upload files

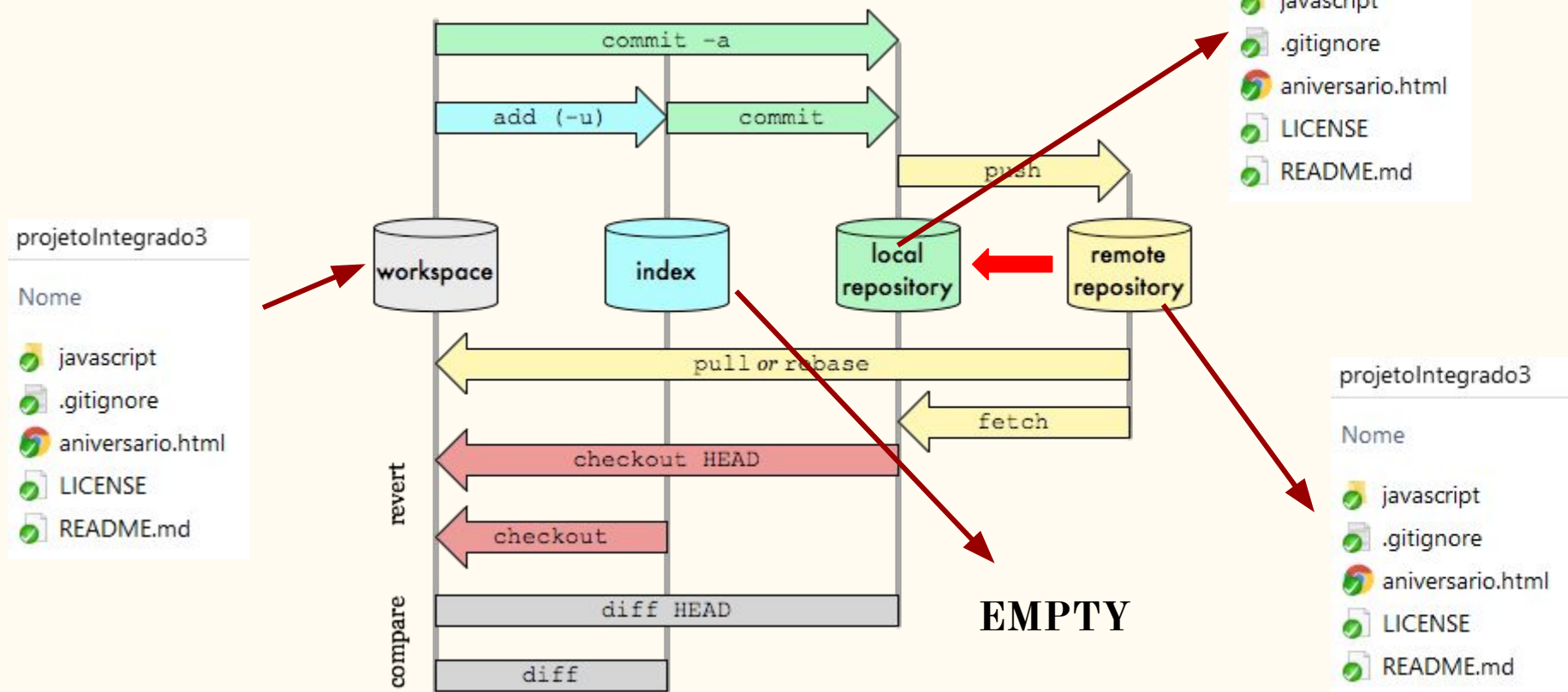
Find file

Clone or download ▼

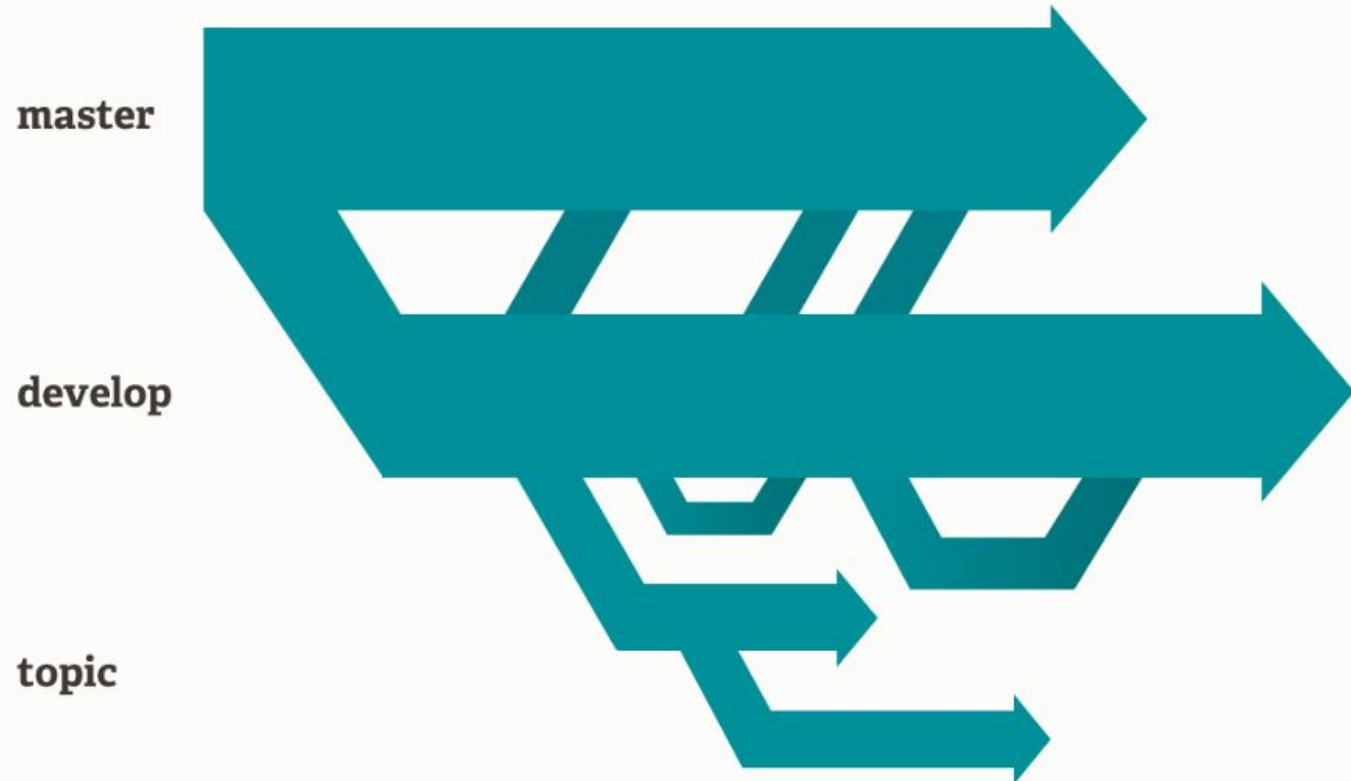
 **hanibalce** Commit Inicial Latest commit eb11b69 6 minutes ago

 javascript	Commit Inicial	6 minutes ago
 .gitignore	Initial commit	1 hour ago
 LICENSE	Initial commit	1 hour ago
 README.md	Initial commit	1 hour ago
 aniversario.html	Commit Inicial	6 minutes ago

Fluxo de Trabalho do Git (Git Workflow)



Trabalhando com Ramos (Branches)



Trabalhando com Ramos (Branches)

16 - Vamos criar uma nova branch de desenvolvimento chamada development no nosso repositório local. Execute o comando `git pull` para ter certeza que o repositório local está atualizado e em seguida o comando `git checkout -b development`

```
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> git pull
Already up to date.
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> git checkout -b development
Switched to a new branch 'development'
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3>
```

Trabalhando com Ramos (Branches)

17 - Vamos enviar a branch development para o repositório remoto.
Execute o comando `git push origin development`

```
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> git push origin development
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'development' on GitHub by visiting:
remote:   https://github.com/hanibalce/projetoIntegrado3/pull/new/development
remote:
To https://github.com/hanibalce/projetoIntegrado3.git
* [new branch]      development -> development
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3>
```

Trabalhando com Ramos (Branches)

18 - O Git permite vermos todas as branches criadas através do comando `git branch -a`.

```
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> git branch -a
* development
master
```

Trabalhando com Ramos (Branches)

19 - Se quisermos voltar para branch master, podemos usar o comando `git switch master`

```
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> git switch master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
PS C:\Users\Anibal\Desktop\repositorios\projetoIntegrado3> git branch -a
  development
* master
```

Façam os passos abaixo e me digam o aconteceu.

20 - Uma parte da equipe vai retornar para a branch development e realizar algumas alterações nos arquivos html e javascript. Em seguida vamos commitar e tentar realizar o comando **git push**.

21 - Outra parte da equipe vai continuar na branch master, também vai realizar alterações nos arquivos html e javascript, commitar e realizar o **push**.

22 - Agora a equipe que trabalho na branch development vai tentar realizar um merge das alterações na branch master.

O comando a ser utilizado é: **git merge origin/master**

Resolvam os conflitos....