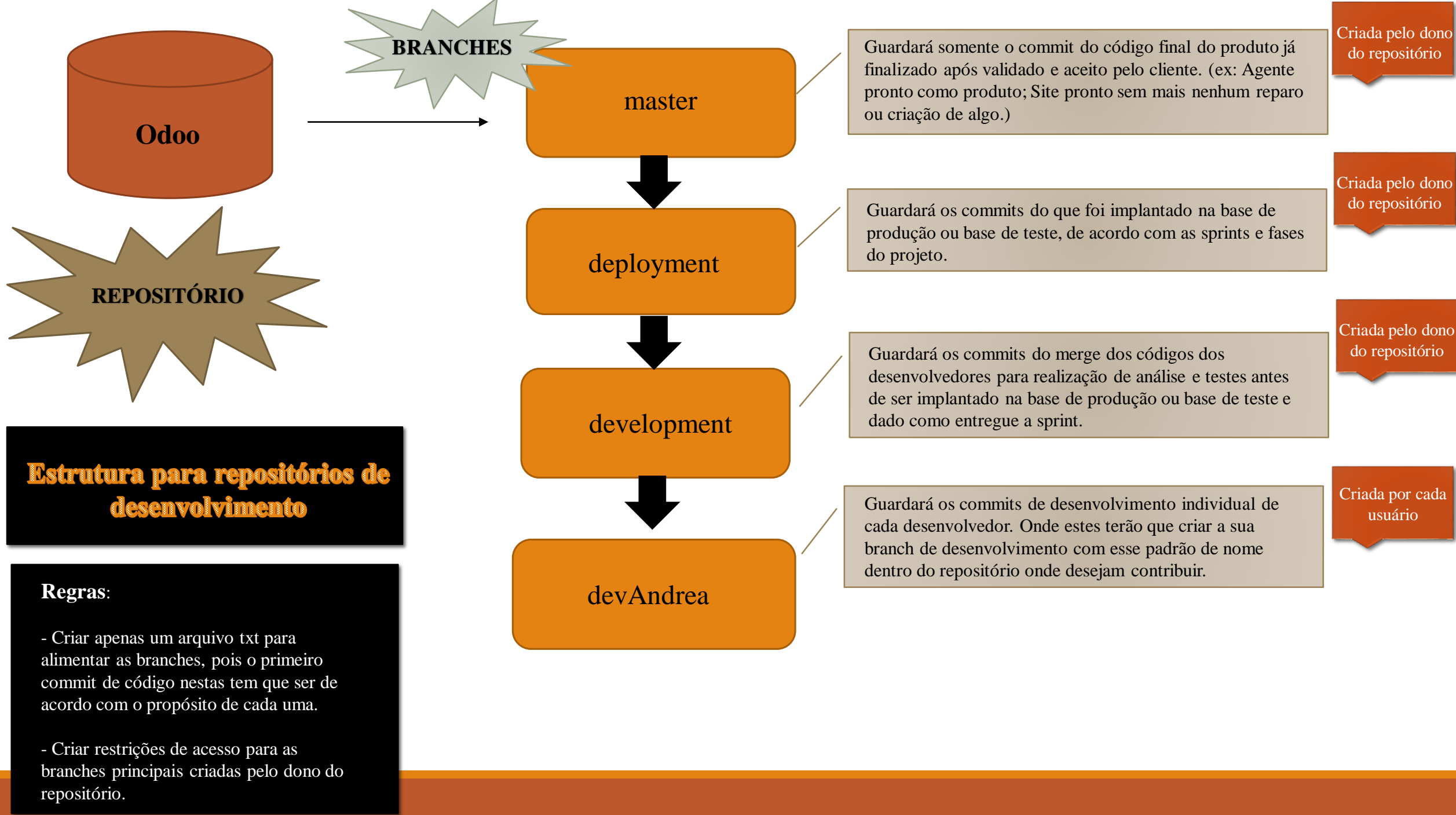
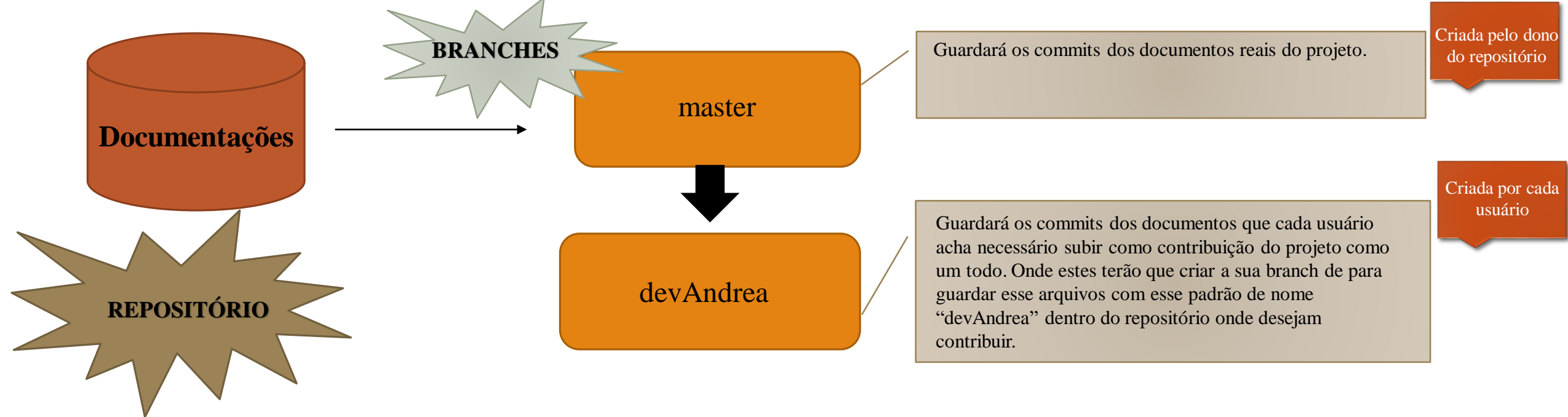


Git para Projetos

PADRÃO DE ESTRUTURA PARA EQUIPES E COMANDOS BÁSICOS







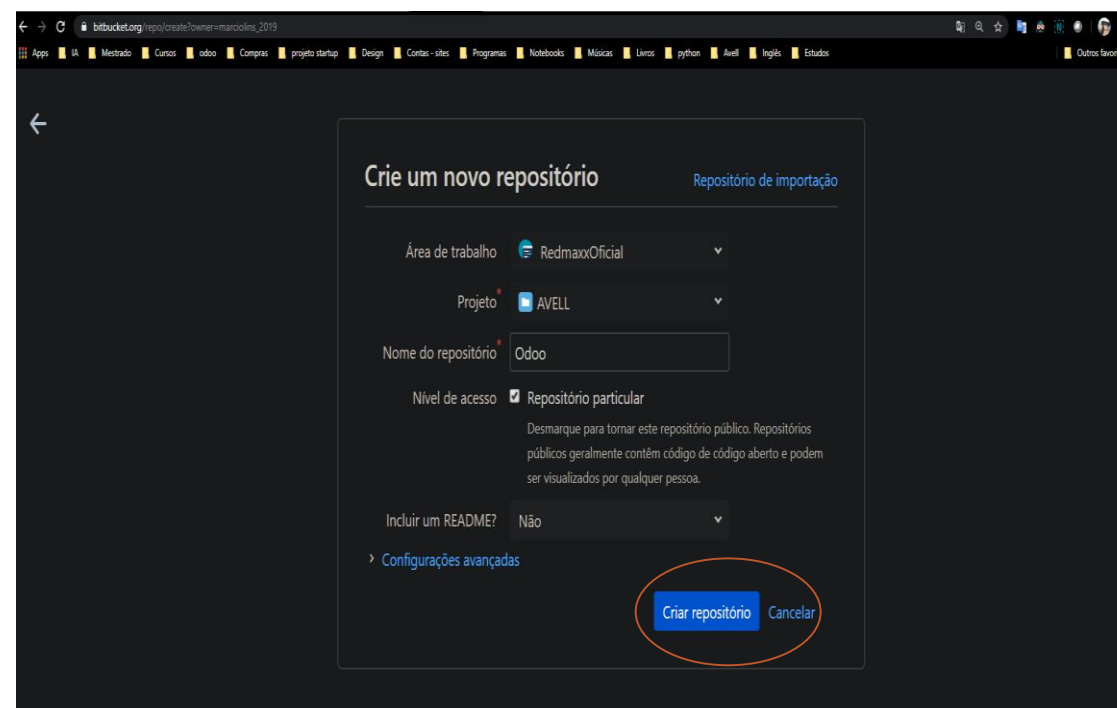
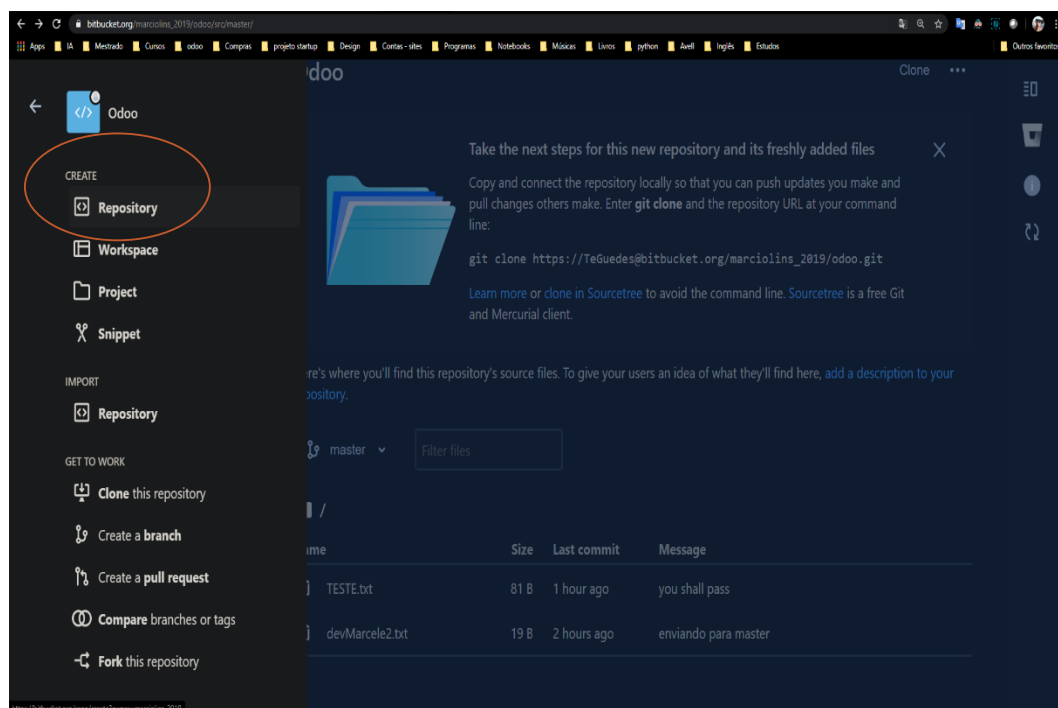
Estrutura para repositórios de documentação

Regras: *(vejam branches como pastas e commits como uma versão de um documento que está sendo guardado)*

- Na branch devNome, serão guardados os docs que cada desenvolvedor achar interessante e que contribua para o projeto como um todo.
- Na branch master serão guardados os documentos realmente necessários para o projeto. Onde o dono do repositório irá restringir o acesso a esta branch, e somente o mesmo decidirá que documentações ficarão nesta branch.

Processo para criar e subir dados locais para um repositório na plataforma de versionamento (Bitbucket, Github, Gitlab)

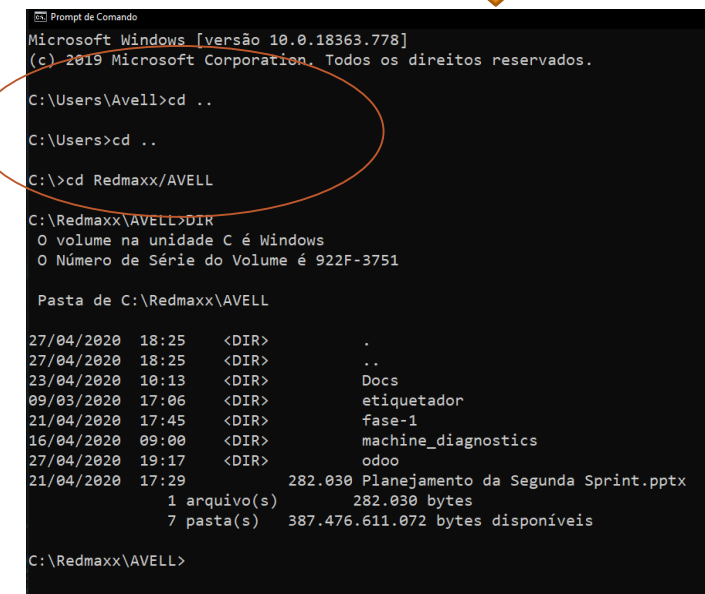
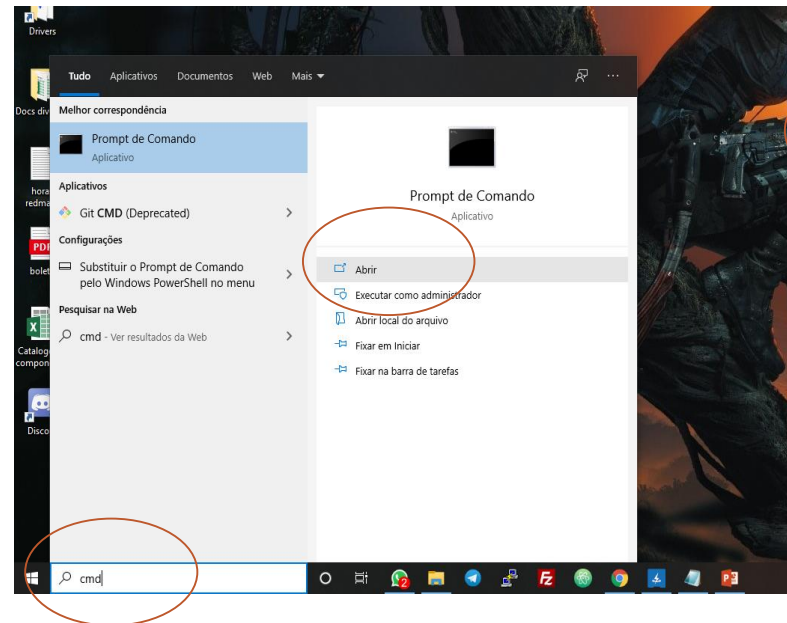
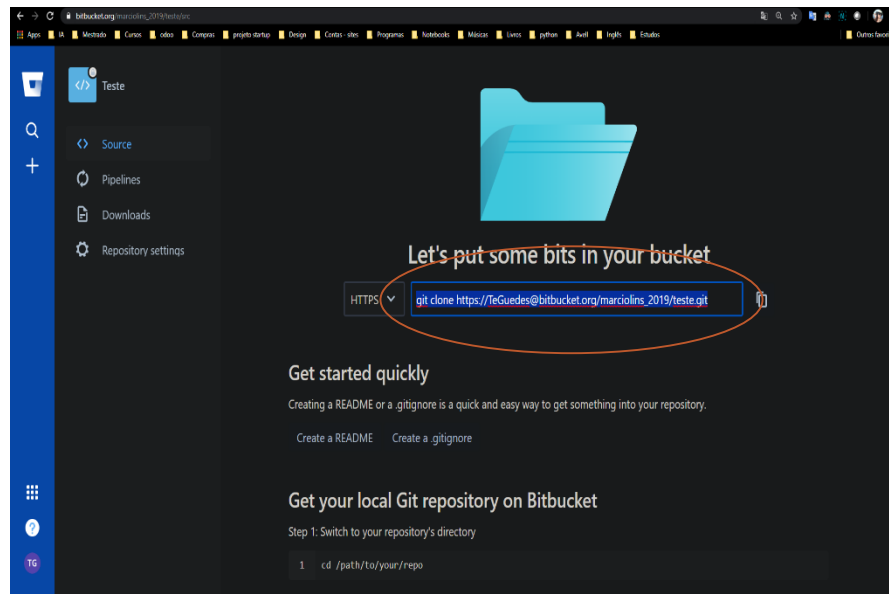
Crie um repositório dentro da plataforma especificando a área de trabalho, o projeto, nível de acesso como privado (para que nenhuma pessoa ao pesquisar seu repositório veja, plagie ou altere sem sua permissão), e com opção de não incluir README.



Processo para criar e subir dados locais para um repositório na plataforma de versionamento (Bitbucket, Github, Gitlab)

Após criar o repositório clique no mesmo, localize o clone na tela, copie o link que está sendo disponibilizado. Em seguida abra o **cmd** do Windows digitando o nome do mesmo na barra de pesquisa, e dentro dele vá para a pasta onde deseja guardar esse repositório.

cd .. (comando para sair da pasta)
cd nome da pasta
(comando para ir a uma pasta e guardar o repositório criado na plataforma)
dir (comando para ver o que tem na pasta)



Processo para criar e subir dados locais para um repositório na plataforma de versionamento (Bitbucket, Github, Gitlab)

Cole o link copiado no **cmd** com o comando **git clone** antes, e veja que ele baixará o repositório na pasta que escolhestees criando uma pasta com o nome do repositório criado na plataforma. No **cmd** após ter feito isso, escreva **cd nome do repositório**, como neste exemplo: **cd teste**. Pronto você está dentro do repositório local (na sua máquina) e agora pode subir, baixar arquivos, criar branches e aplicar comando que o git disponibiliza. 😊/ 😊/

```
Prompt de Comando
27/04/2020 18:25 <DIR> .
27/04/2020 18:25 <DIR> ..
23/04/2020 10:13 <DIR> Docs
09/03/2020 17:06 <DIR> etiquetador
21/04/2020 17:45 <DIR> fase-1
16/04/2020 09:00 <DIR> machine_diagnostics
27/04/2020 19:17 <DIR> odoo
21/04/2020 17:29 282.030 Planejamento da Segunda Sprint.pptx
                  1 arquivo(s) 282.030 bytes
                  7 pasta(s) 387.476.611.072 bytes disponíveis

C:\Redmaxx\AVELL>git clone https://TeGuedes@bitbucket.org/marciolins_2019/teste.git
Cloning into 'teste'...
warning: You appear to have cloned an empty repository.

C:\Redmaxx\AVELL>cd teste
O sistema não pode encontrar o caminho especificado.

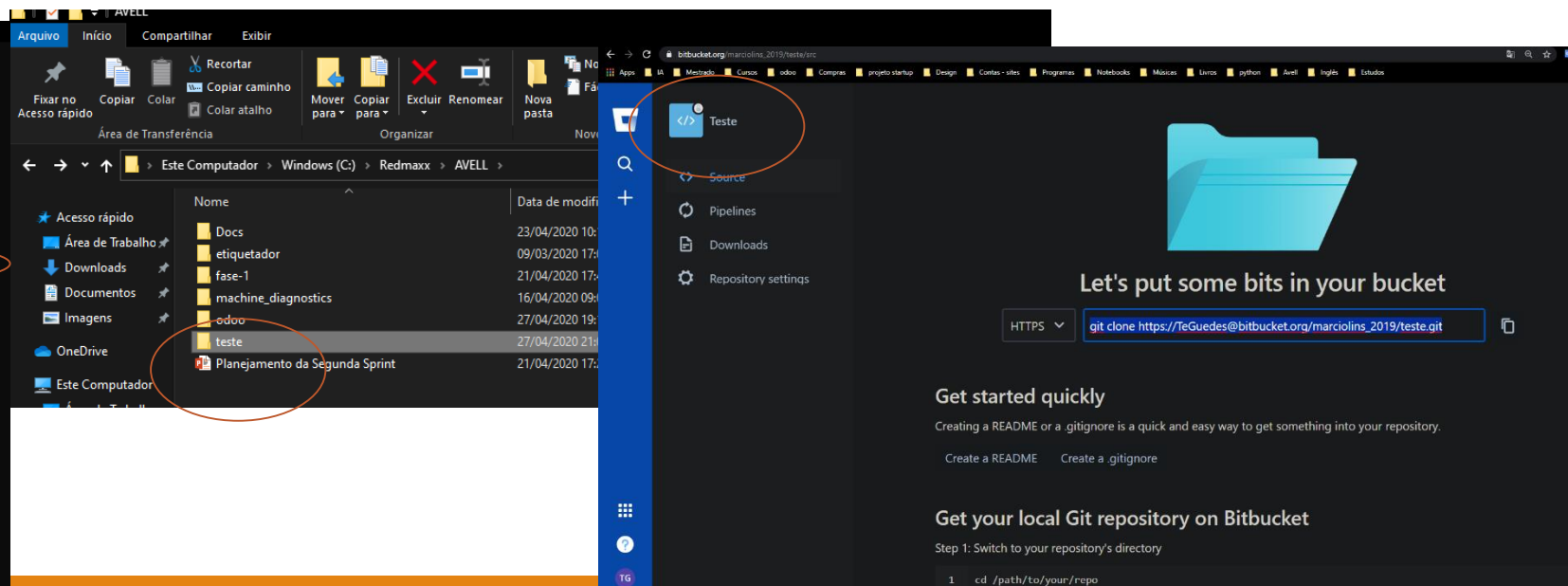
C:\Redmaxx\AVELL>cd teste

C:\Redmaxx\AVELL\teste>git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

C:\Redmaxx\AVELL\teste>
```



Processo para criar e subir dados locais para um repositório na plataforma de versionamento (Bitbucket, Github, Gitlab)

Dentro do repositório local (na minha máquina) coloquei o arquivo que quero subir para o repositório na plataforma, e este arquivo ou esta versão não está lá. Para eu enviá-lo basta eu estar dentro da pasta que é o repositório local, este ter o mesmo nome do repositório que está na plataforma para qual desejo subir, e aplicar os seguintes comandos abaixo no **cmd**. Em seguida é só conferir no repositório na plataforma que estará lá.

- **git status** (comando para verificar a branch onde estás, e se tem algum arquivo diferente ou alterado que não está na sua máquina e não no repositório na plataforma. Este é sinalizado na cor vermelha)
- **git add nome do arquivo** (ex: *git add teste* ou *git add .* – para adicionar tudo) (serve para determinar o que será enviado, um arquivo especificado ou todos que estão no repositório local e não está no da plataforma)
- **git commit -m “adicione aqui um comentário sobre essa versão do documento que está subindo”** (serve como um controle de versionamento de arquivo. Toda vez que tentares subir um arquivo, precisas commitar escrevendo um comentário que descreve o que estás subindo)
- **git push origin nome da branch** (ex: *git push origin master*) (serve para enviar o arquivo adicionado e commitado para o repositório na plataforma)

Nunca delete essa pasta! **.git**, que aparece no repositório local. Pois ela que define que este é um repositório. Permitindo que a manipule com comandos git e haja comunicação com o repositório que está na plataforma.

Fazendo um **git push** na branch que estiveres, é nesta que será guardado o arquivo que está subindo também tanto no repositório local como no da plataforma.

```
C:\Redmaxx\AVELL\teste>git status
On branch master
No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
C:\Redmaxx\AVELL\teste>git add .
C:\Redmaxx\AVELL\teste>git commit -m "arquivo de teste"
[master (root-commit) 5059de1] arquivo de teste
1 file changed, 1 insertion(+)
 create mode 100644 Hello Word!.txt
C:\Redmaxx\AVELL\teste>git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 236 bytes | 118.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://bitbucket.org/marciolins_2019/teste.git
 * [new branch] master -> master
C:\Redmaxx\AVELL\teste>
```

Obs.: Sempre nessa ordem 😊

Processo para criar e subir dados locais para um repositório na plataforma de versionamento (Bitbucket, Github, Gitlab)

Caso modifique este arquivo na sua máquina, é só subir novamente commitando esse arquivo alterado sem precisar mudar o nome, assim criando versões 😊 que é a proposta da plataforma.

Caso queira baixar algo que está lá, e não está no seu repositório local, é só utilizar o comando **git pull**, que ele baixará tudo o que está lá, atualizando seu repositório local.

Só lembre dos passos anteriores necessários.

Não esqueça de estar dentro da pasta que é o repositório local para poder aplicar os comandos git.

Para criar uma branch use o comando **git checkout -b nome_da_branch**

Toda vez que criar uma branch precisas subir um arquivo para ela. Se não subires, ele não cria, ou substitui a que vc estava com o nome da que vc criou.

Toda vez que um projeto é criado, o git já cria uma branch raiz chamada **Master**

Esta geralmente é usada para guardar as versões dos arquivos principais ou finais do projeto.

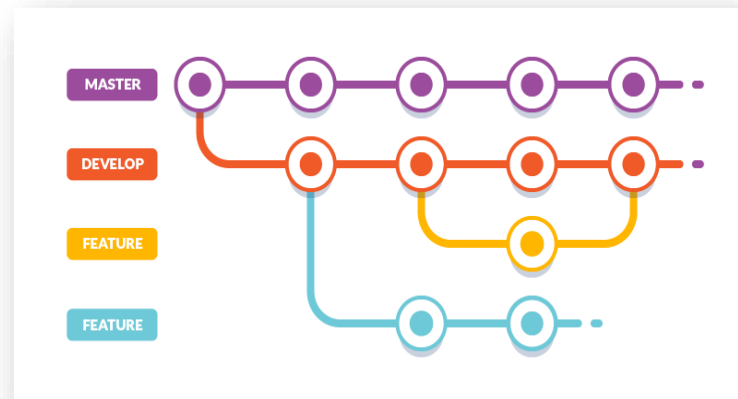
Para uma ver as branches criadas use o comando **git branch**

Para uma ir para uma branch use o comando **git checkout nome_da_branch**

Quando estiveres dentro de uma, tudo o que for feito será guardado dentro dela.

Para excluir uma branch use o comando **git branch -d nomedabbranch**

git branch -d -D (para deletar como super)



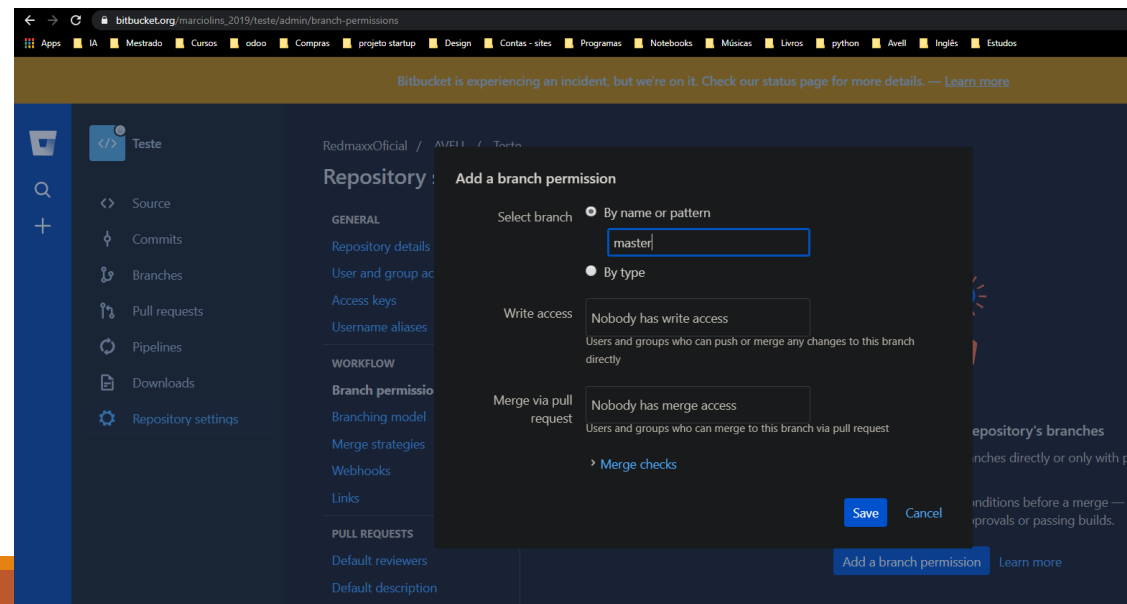
Processo para restringir uma branch dentro da plataforma bitbucket

É ideal que restrinja as branches criadas, principalmente as principais como a master, deployment e development, para que somente você altere, faça merge (união de versões) e qualquer manipulação, como um Admin.

Para isso:

1. Vá dentro do repositório na plataforma;
2. Clique em **configurações** no menu a esquerda;
3. Clique em **permissões de branch** no menu a esquerda;
4. Clique em **adicionar uma permissão de branch**;
5. Para criar permissão para uma branch específica veja a imagem e os passos abaixo:

- Selecione uma branch por nome ou padrão;
- Escreva o nome da branch a qual deseja restringir;
- Por padrão já fica definido que ninguém escreve, faz merge ou pull request, mas tens a opção de escolher um dos colaboradores.
- Em seguida é só salvar.



*Para brincar com mais comandos segue o arquivo **Workshop GIT.txt**
(<https://github.com/DevNicque/MyCods/blob/docs/Workshop%20GIT.txt>)
Neste há comandos que aprendi no curso de git com anotações ^^*

Segue links para estudo:

https://www.youtube.com/watch?v=hj_YO9tWRAY

<https://www.hostinger.com.br/tutoriais/git-gui/>

[https://digitalinnovation.one/cursos/introducao-ao-](https://digitalinnovation.one/cursos/introducao-ao-github-e-comandos-essenciais-para-trabalhar-em-equipe)

[github-e-comandos-essenciais-para-trabalhar-em-equipe](https://digitalinnovation.one/cursos/introducao-ao-github-e-comandos-essenciais-para-trabalhar-em-equipe)

https://rogerdudler.github.io/git-guide/index.pt_BR.html

Qualquer dúvida chama no zap 😊

by, Andrea Monicque

