

Repositórios Remotos



GitHub

IT BOARDING

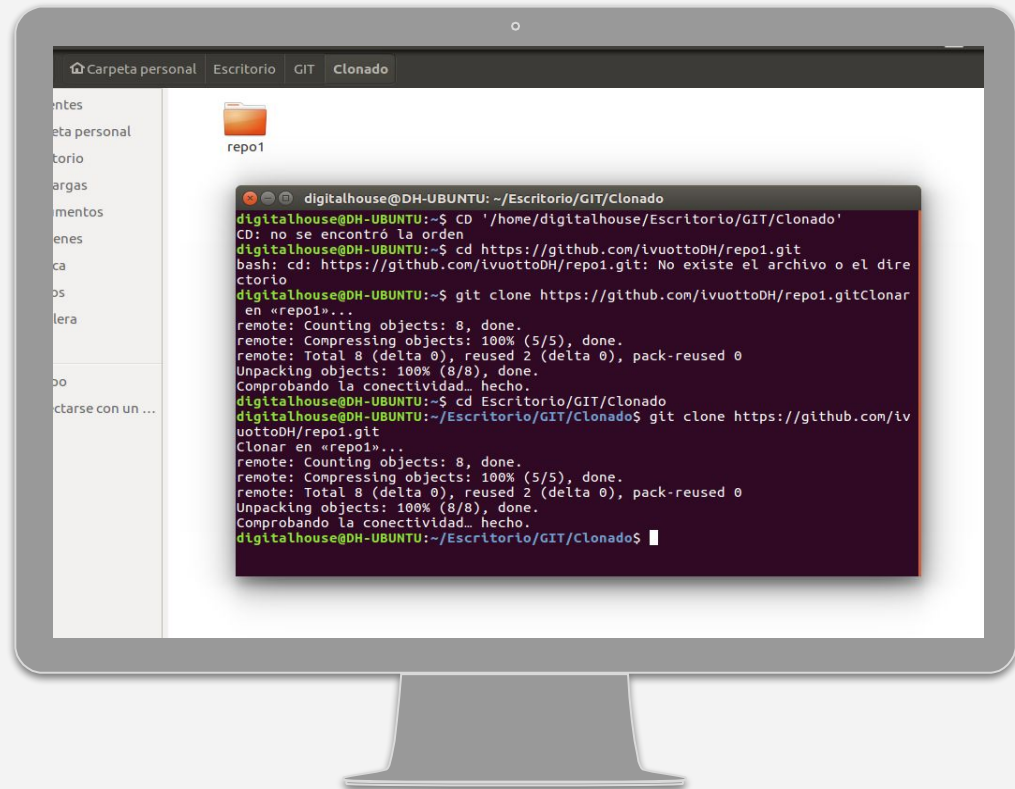
BOOTCAMP



// O que é **GitHUB**?

- É um dos provedores de repositório remoto mais amplamente usados no mundo e no ambiente de desenvolvimento.
- O GitHub se comunica com nosso repositório Git local, permitindo-nos construir diferentes versões e adicionar colaboradores para compartilhar ou colaborar com o trabalho que estamos fazendo.
- Para criar uma conta no GitHub, podemos acessar seu site oficial:
<https://github.com/>
-

Como adicionamos una identidade?





Para que tudo o que fazemos seja "assinado" por nós, precisamos informar ao repositório local quem somos para que ele possa comunicá-lo ao repositório remoto. Fazemos isso por meio dos seguintes comandos:

```
git config user.name "meu_usuario"  
git config user.email "meu_usuario@email.com"
```

```
git config user.name " "
```



Dentro das aspas colocaremos o usuário que queremos usar (geralmente corresponde ao que usaremos mais tarde no Github.com)

```
git config user.email " "
```

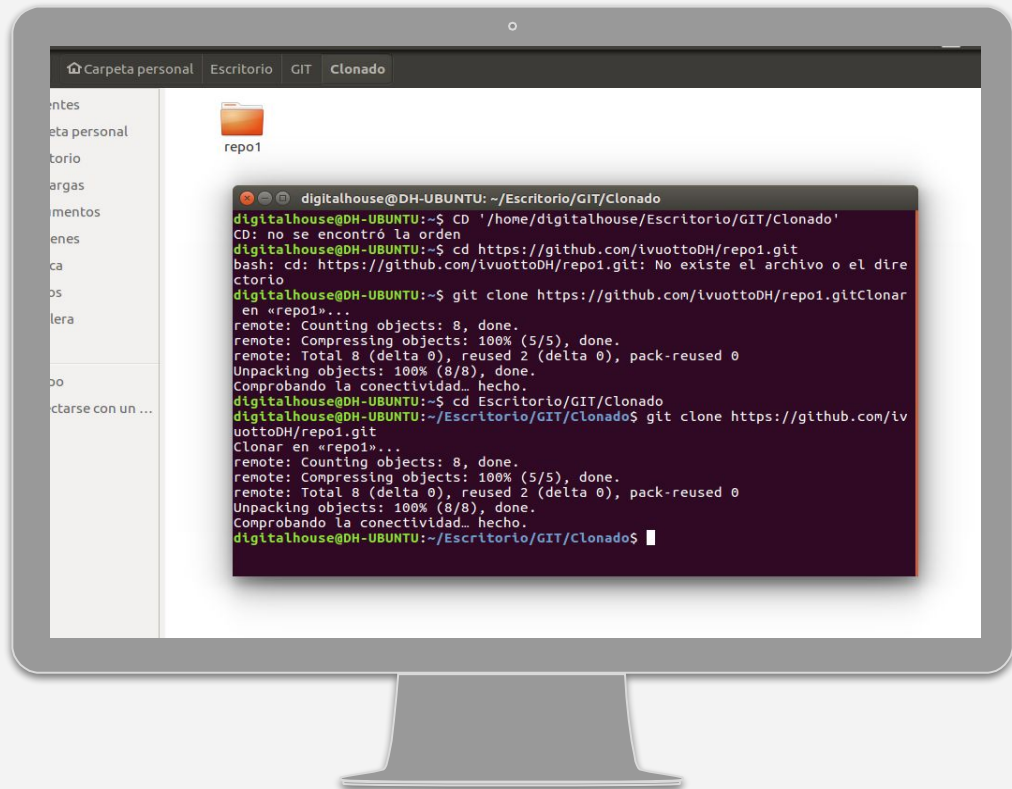
Dentro das aspas colocaremos o e-mail com o qual pretendemos nos registrar posteriormente no Github.com

IT BOARDING

BOOTCAMP



Como criamos e atribuímos um repositório REMOTO?



// Conta do GitHub

- Para conectar nosso repositório local a um repositório remoto no GitHub, primeiro devemos ter uma conta criada.
- Uma vez criada a conta, devemos proceder à criação de um repositório remoto e configurar a sua sincronização com o local e os diferentes colaboradores que irão interagir.



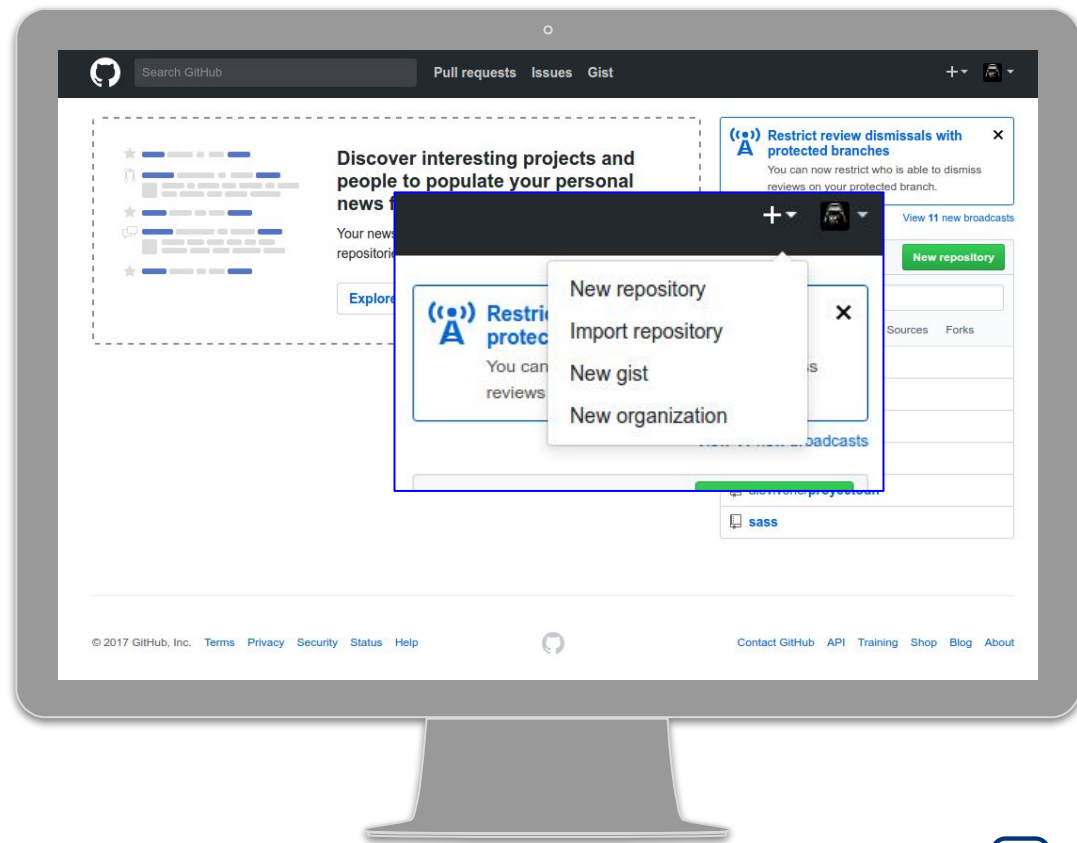
GitHub



1º Logado em nossa conta GitHub, vá para o ícone **+** e escolha a opção **New Repository**.

IT BOARDING

BOOTCAMP





2º O nome que escolhemos pode ser qualquer um (desde que não corresponda a outro repositório que criamos anteriormente)

3º Nós escolhemos a privacidade que queremos (por enquanto pública)

4º Clicamos em Criar repositório

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: **javi-dh** / Repository name: **repo-de-prueba** ✓

Great repository names are short and memorable. Need inspiration? How about **curly-octo-lamp**.

Description (optional)

☐ **Public**
Anyone can see this repository. You choose who can commit.

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

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Definir um nome

NÃO PRECISA MUDAR

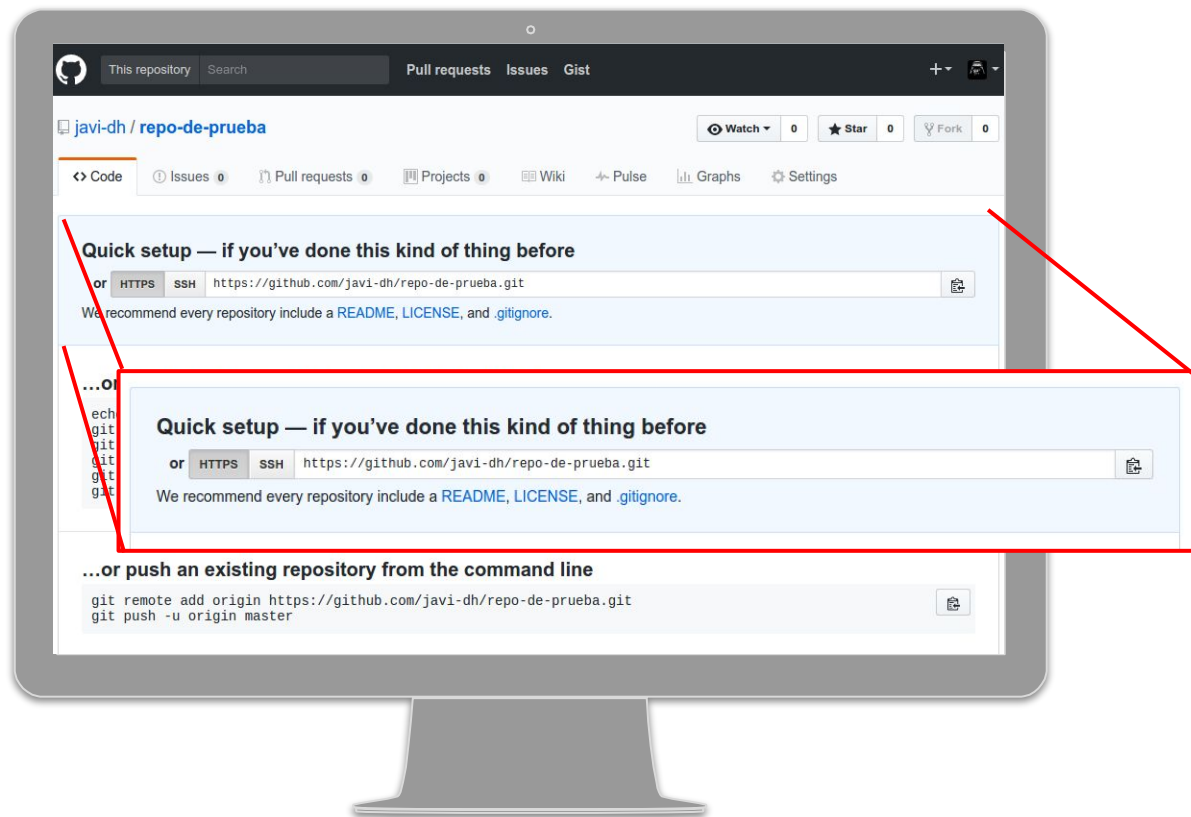
CRIAR REPOSITÓRIO



5° No final, ele nos dará esse URL que precisamos ter em mãos para poder atribuir nosso **repositório remoto**.

IT BOARDING

BOOTCAMP





// Tendo criado o Repositório Remoto e para que nosso Repositório Local saiba onde queremos carregar nossos arquivos, temos que especificá-lo da seguinte forma:

```
git remote add origin "o url que obtivemos na etapa anterior"
```

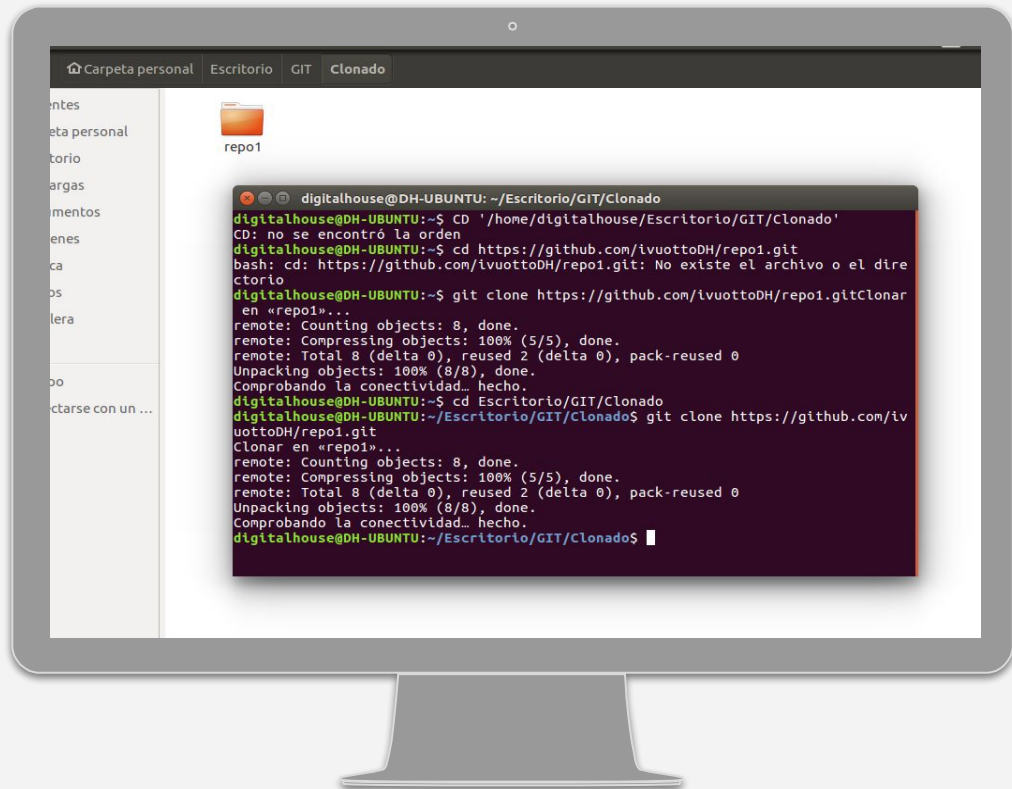
git remote add origin https://...

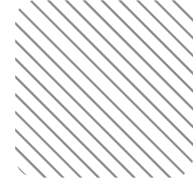
Com este comando, estamos indicando ao nosso repositório local, para onde (repositório remoto) queremos levar nossos arquivos.

O URL a ser colocado é o que obtivemos ao criar um repositório remoto no Github.com



Como adicionamos um arquivo a um repositório local?





Até agora, nossos arquivos não foram adicionados **temporariamente** ao repositório. Para adicionar arquivos ao repositório teremos que escrever o seguinte comando:

git add --all
ou
git add “arquivo.txt”

IT BOARDING

BOOTCAMP





```
git add --all
```

Adiciona ao "stage" (temporariamente) todos os arquivos que criamos em nosso projeto.

```
git add "arquivo"
```

Adiciona ao "stage" (temporariamente) apenas o arquivo referenciado.

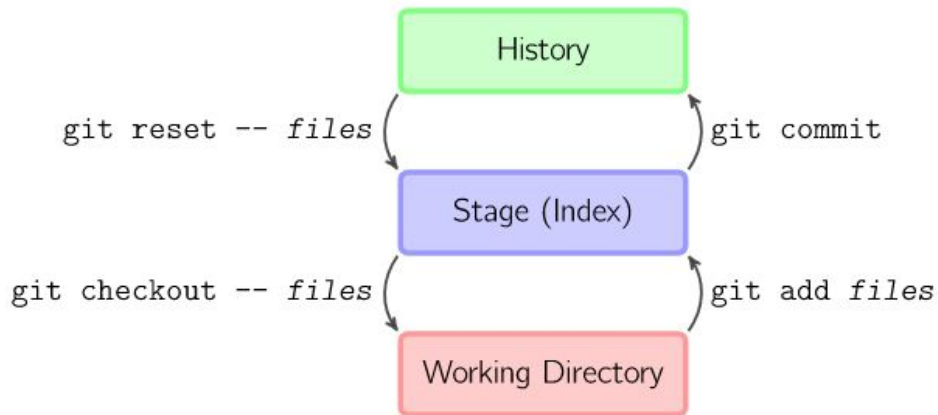
IT BOARDING

BOOTCAMP

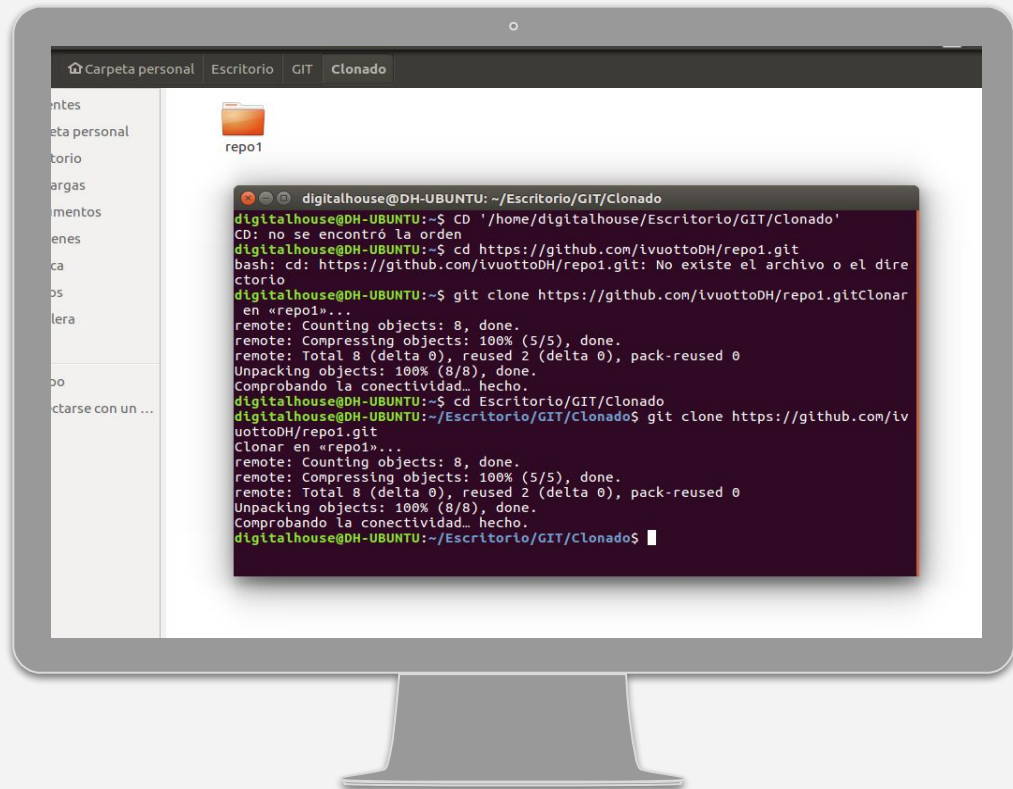


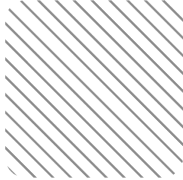
// O que é o Stage?

- Stage É um estado intermediário entre a versão local de um arquivo com o qual estamos trabalhando e a versão final que enviaremos para um repositório remoto.
- É um ponto como "Stand By", onde os arquivos estão esperando para serem selecionados (ou não) para finalmente serem versionados.



Como conhecemos o status do nosso repositório local?





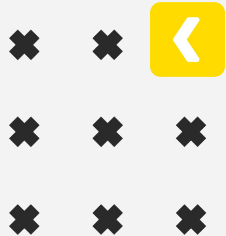
Sempre que quisermos verificar o status de nosso repositório, podemos escrever o seguinte comando:

git status

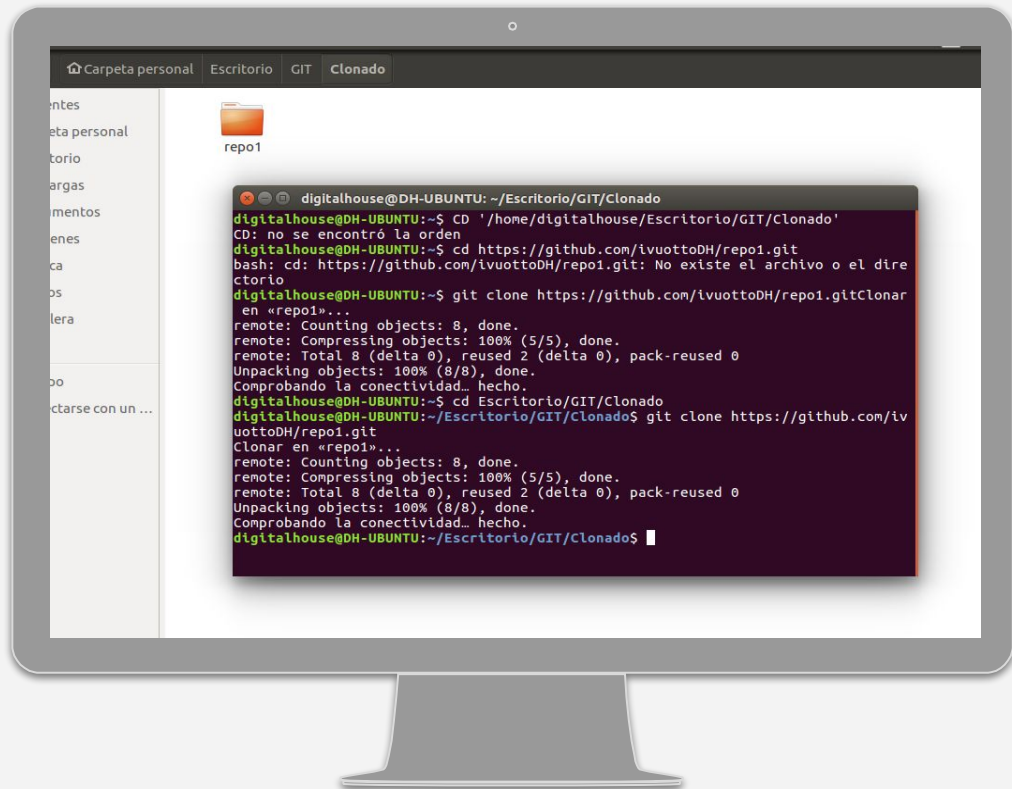


git status

Ele analisa o status do repositório, nos informa se há arquivos que não foram adicionados temporariamente ao stage, bem como se há arquivos adicionados ao stage, mas não definitivamente (**commit**).



Como adicionamos archivos de forma definitiva?





Para finalmente confirmar que os arquivos adicionados ao stage sejam definitivos, escreveremos:

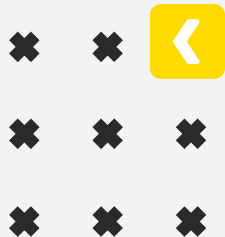
```
git commit -m "uma mensagem qualquer"
```



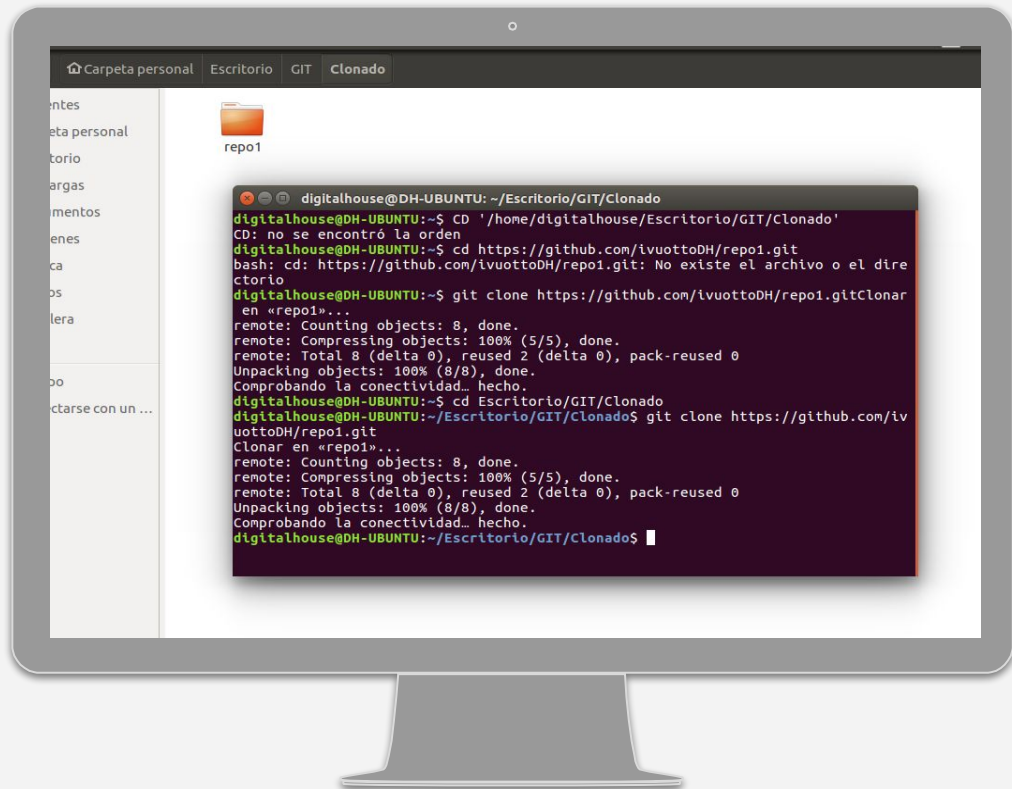
git commit -m "uma mensagem qualquer"

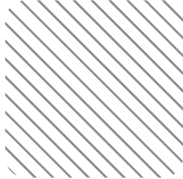
A diretiva **commit** diz ao repositório que queremos adicionar os arquivos permanentemente. O **-m** indica que a seguir adicionaremos uma mensagem que especifica o trabalho que fizemos (serve como um “documento de versionamento”).

Os **commits** servem como pequenos **backups** aos quais podemos retornar facilmente se precisarmos.



Como enviamos
nossos arquivos
do **repositório
local** para o
remoto?





Para enviar os arquivos que temos em nosso repositório local para o repositório remoto, escreveremos a seguinte instrução:

```
git push origin main
```



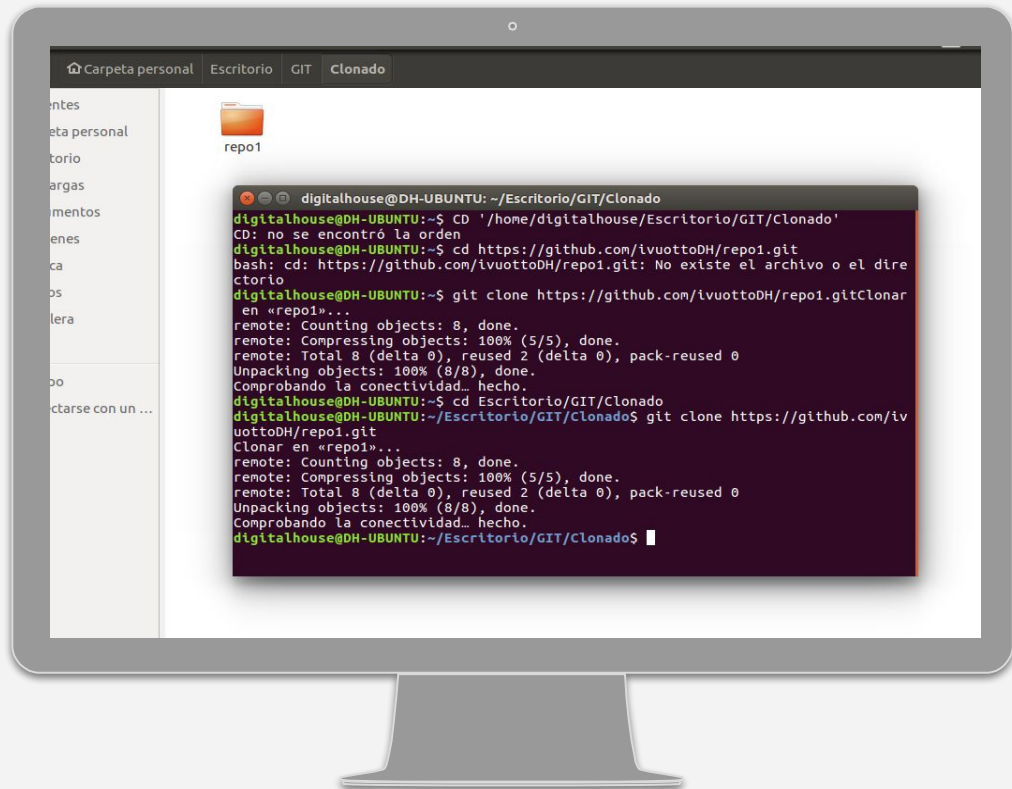
git push origin main

O push nos permite enviar os arquivos de nossa máquina (repositório local) para o repositório remoto.

Ao especificar main, estamos dizendo para qual branch do repositório queremos enviar nossos arquivos (neste caso, para o principal).



Como baixamos nossos arquivos do **repositório** **remoto** para o **repositório local**?





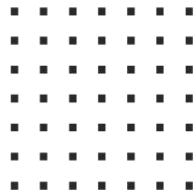
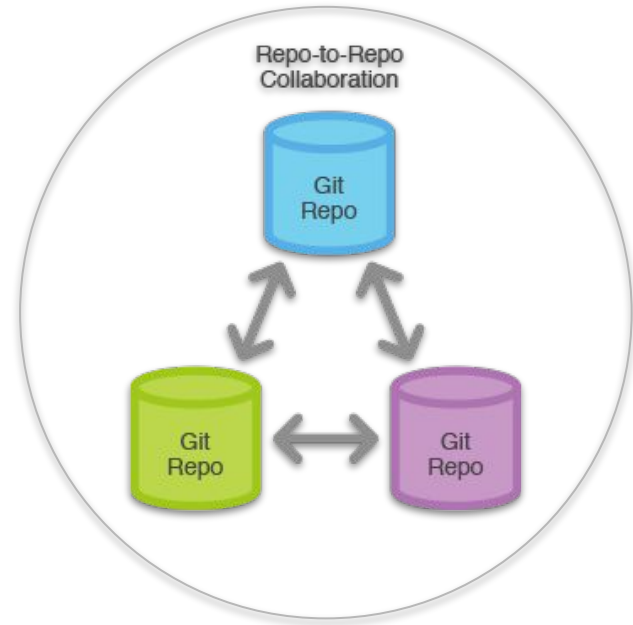
// Baixando os arquivos do repositório remoto para nosso repositório local

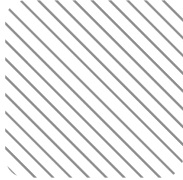
Às vezes queremos ou precisamos baixar nossos arquivos remotos para o computador local.

Para isso precisaremos clonar o repositório remoto em nossa máquina.

IT BOARDING

BOOTCAMP





Para baixar pela primeira vez um repositório remoto para nossa máquina, teremos que cloná-lo no local (pasta) que quisermos.

O comando para clonar um repositório é o seguinte:

git clone <https://github.com/user/repoName>

IT BOARDING

BOOTCAMP



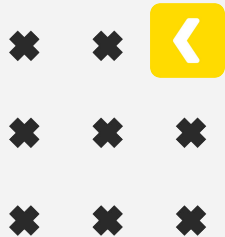


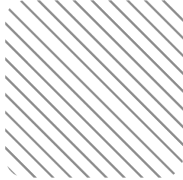
git clone https://....

git clone, permite criar uma cópia idêntica do repositório remoto em nossa máquina para que possamos trabalhar com os mesmos arquivos que temos até aquele momento.

Agora a questão é:

Como faço para atualizar os arquivos que fiz na máquina original com esses novos arquivos?





Se o que queremos é atualizar os arquivos em nosso repositório local com o que existe no repositório remoto, devemos executar o comando:

git pull origin main

IT BOARDING

BOOTCAMP

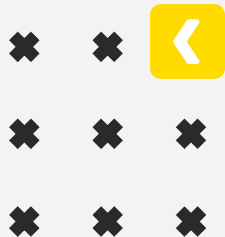




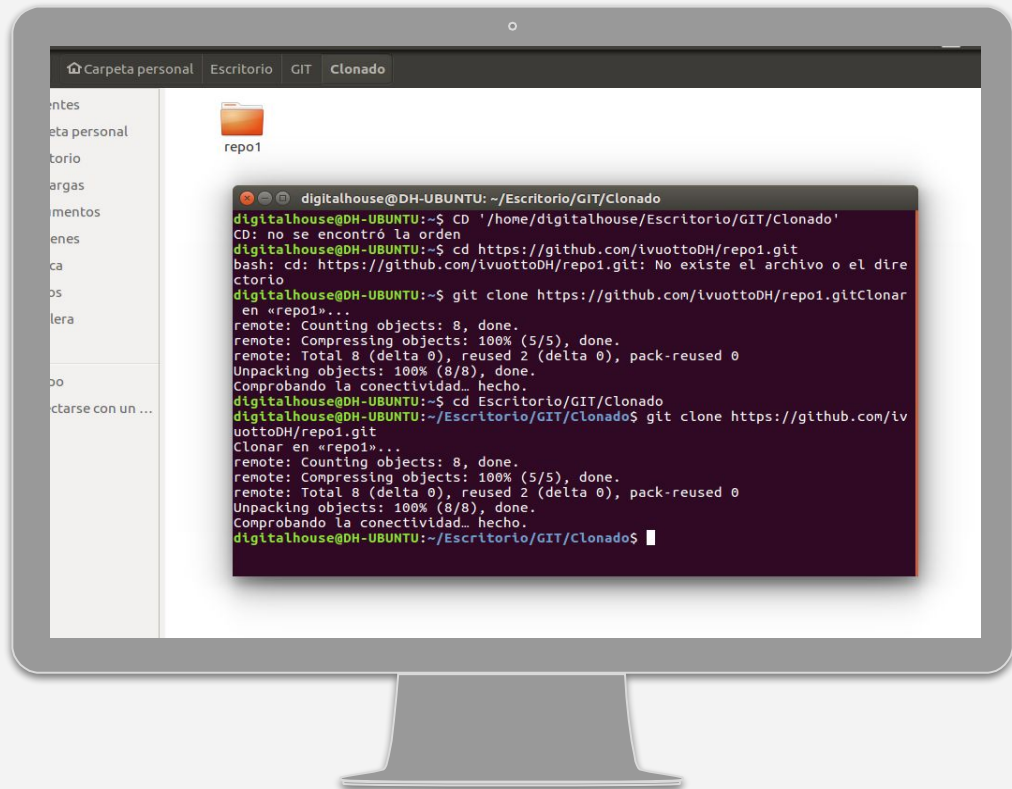
git pull origin main

git pull, baixe para seu repositório local as alterações ou novos arquivos que foram enviados para o repositório remoto de outra máquina.

Este comando é muito funcional se trabalharmos com mais colaboradores no mesmo projeto.



Como adicionamos colaboradores em nosso **repositório remoto?**



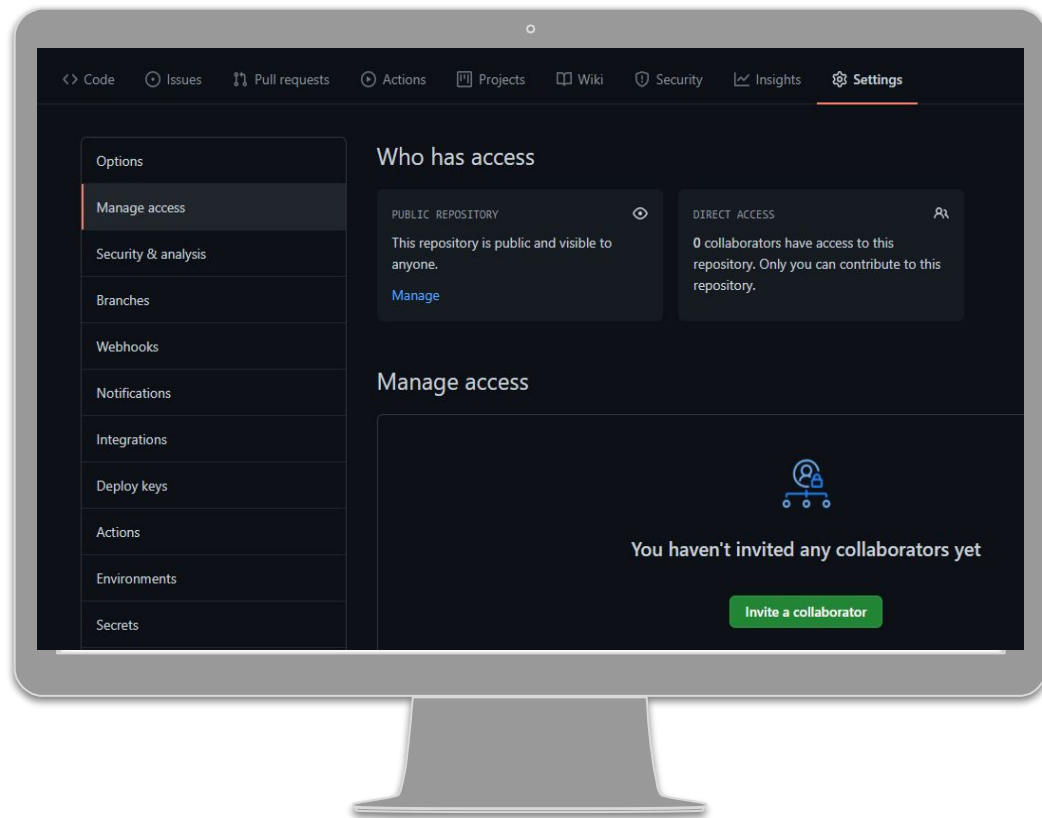


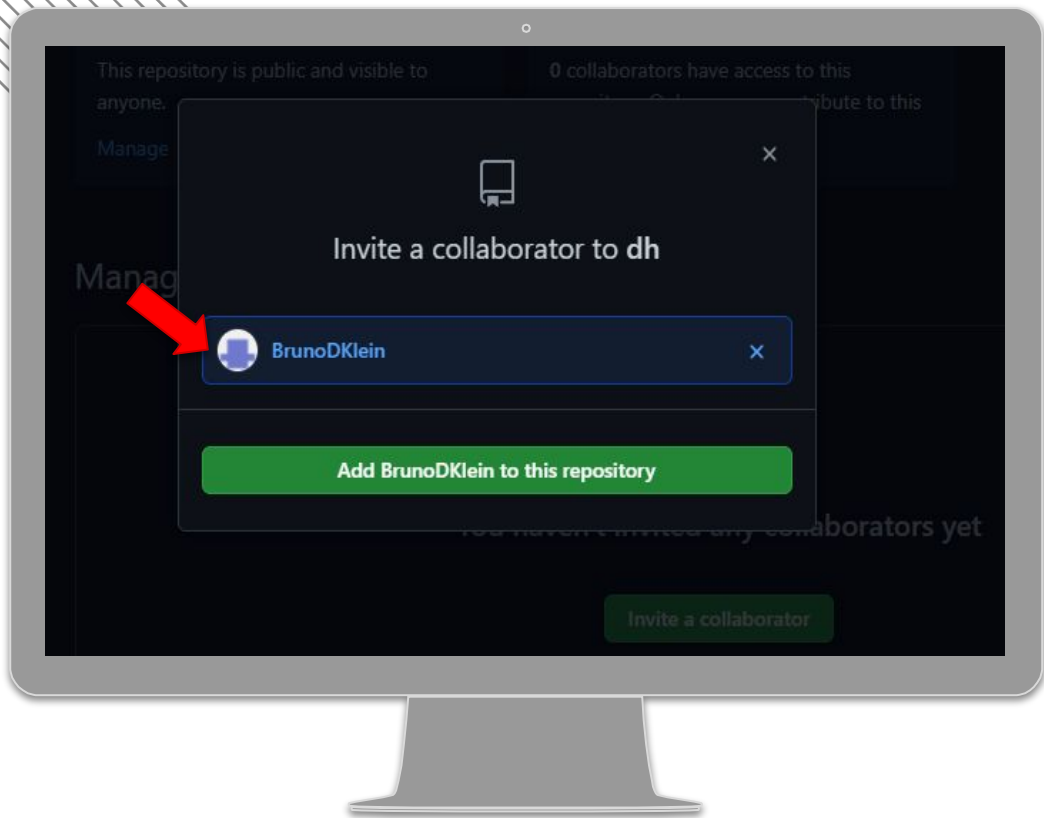
1° Vamos para o nosso repositório GitHub

2° Uma vez lá, acesse **Settings, Manage Access, Invite a Collaborators**

IT BOARDING

BOOTCAMP





IT BOARDING

BOOTCAMP

3º Escrevemos o nome de usuário ou e-mail de nosso colega para convidar e, em seguida, pressionamos o botão:

Add collaborator

4º A pessoa receberá um e-mail, onde deverá aceitar o convite.





Desta forma, adicionamos colaboradores ao nosso repositório remoto.

Agora, eles também têm o poder de enviar seu trabalho para o nosso repositório.

Por este motivo, ao sentar-se para trabalhar, antes de começar sincronize seu projeto com o repositório remoto, utilizando:

git pull origin main

IT BOARDING

BOOTCAMP





Obrigado.

IT BOARDING

BOOTCAMP

