DESENVOLVIMENTO DE SISTEMAS



UC13

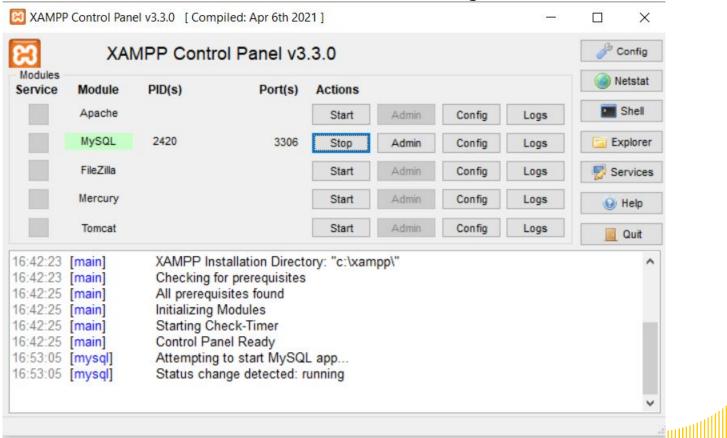
Prof. Viviane de Lima

viviane.lfrancelino@sp.senac.br

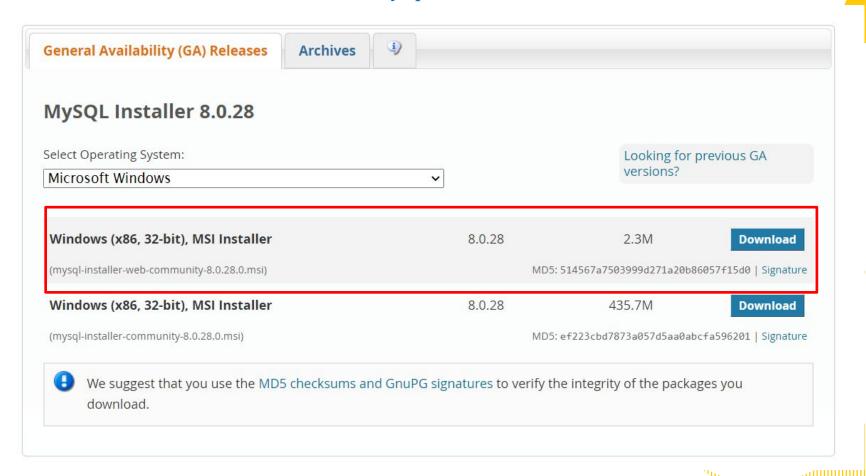
AULA 09

INSTALANDO O MYSQL NO WINDOWS

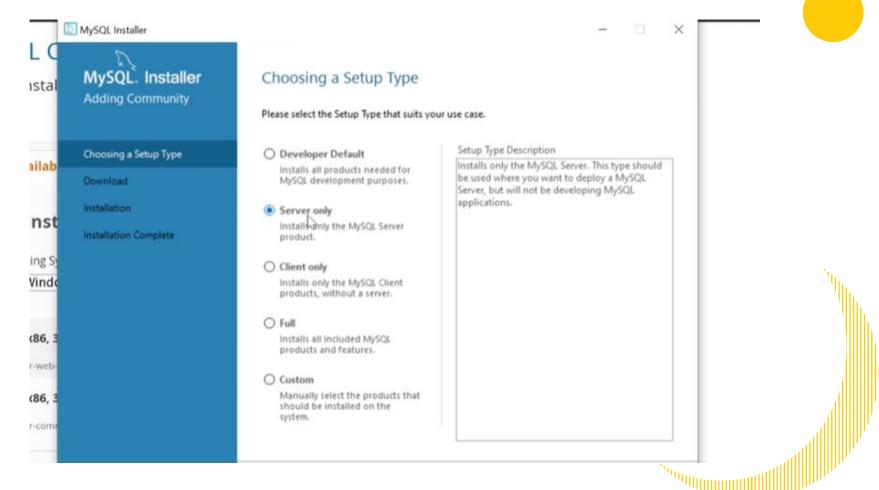
SE VOCÊ JÁ TEM O XAMPP / WAMP / LAMP INSTALADO NO SEU COMPUTADOR NÃO É NECESSÁRIO SEGUIR O PASSO A PASSO, APENAS ABRA O SEU XAMP E ATIVE O BANCO DE DADOS MYSQL



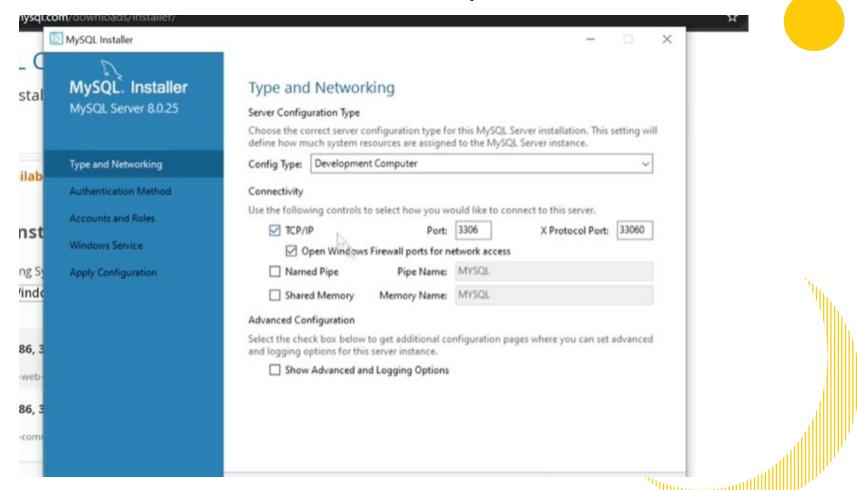
ACESSE O SITE dev.mysql.com/downloads/installer/



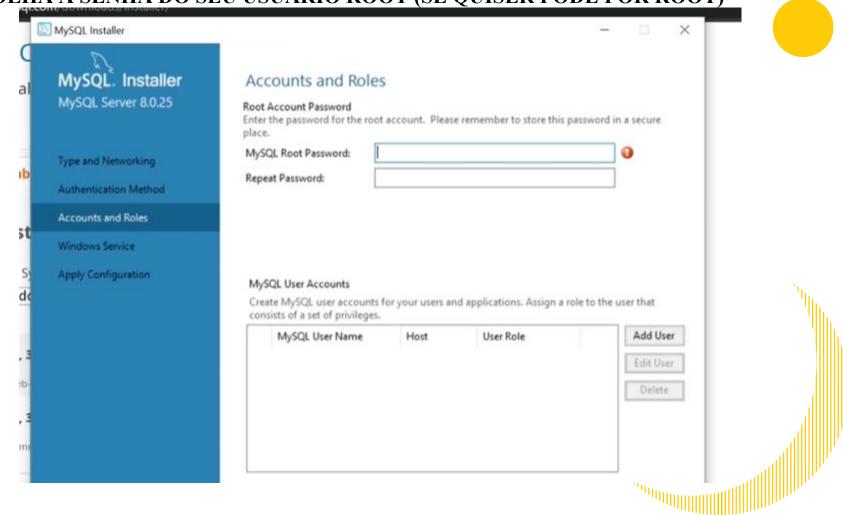
DEIXE NA OPÇÃO SERVER ONLY E CLIQUE EM AVANÇAR



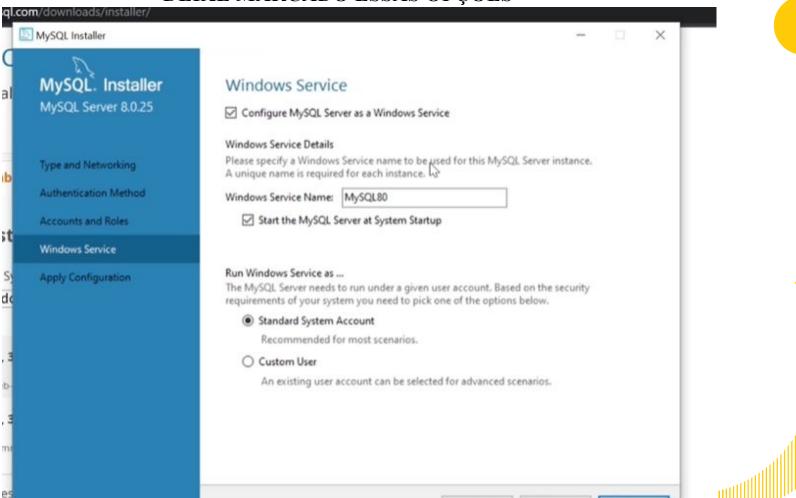
DEIXE MARCADA ESSAS OPÇÕES



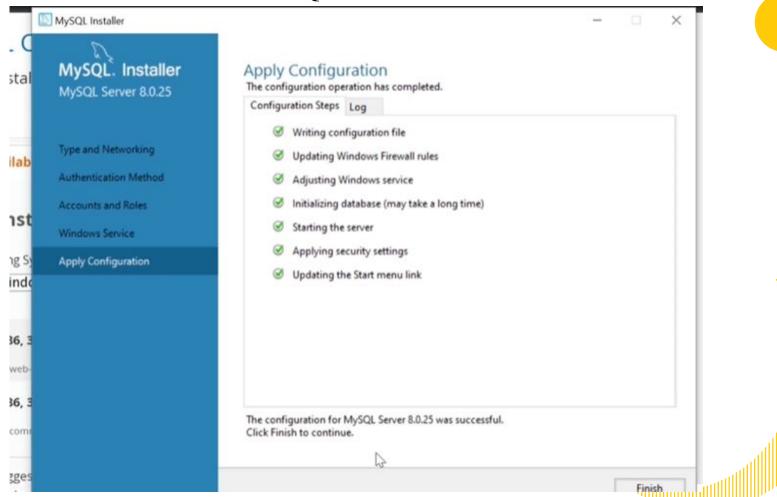
ESCOLHA A SENHA DO SEU USUÁRIO ROOT (SE QUISER PODE POR ROOT)



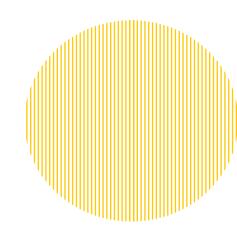
DEIXE MARCADO ESSAS OPÇÕES



POR FIM CLIQUE EM FINISH



INSTALANDO UM CLIENTE SQL

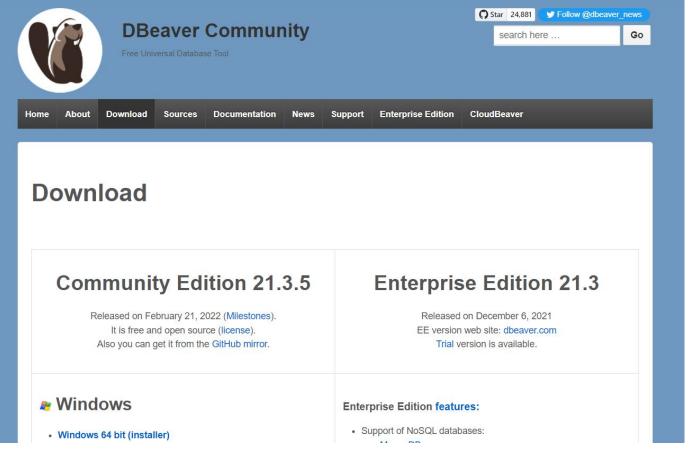


O QUE É UM CLIENTE SQL?



Não é o tipo de cliente que você está pensando! O cliente **SQL serve apenas para facilitar o seu acesso/manuseio do banco de dados**, ao invés de fazer tudo via linha de comando. Com ele você da uns cliques, digita os nomes e seu banco de dados é criado!

EXISTEM VÁRIOS CLIENTES SQL, VAMOS TRABALHAR COM O



POR QUE O DBEAVER? COM O DBEAVER VOCÊ CONSEGUE USAR COM BANCOS RELACIONAIS E NÃO RELACIONAIS

BAIXE A VERSÃO COMMUNITY EDITION Download



Community Edition 21.3.5

Released on February 21, 2022 (Milestones).

It is free and open source (license).

Also you can get it from the GitHub mirror.

Enterprise Edition 21.3

Released on December 6, 2021 EE version web site: dbeaver.com Trial version is available.

Windows

- · Windows 64 bit (installer)
- Windows 64 bit (zip)
- · Install from Microsoft Store
- Chocolatey (choco install dbeaver)

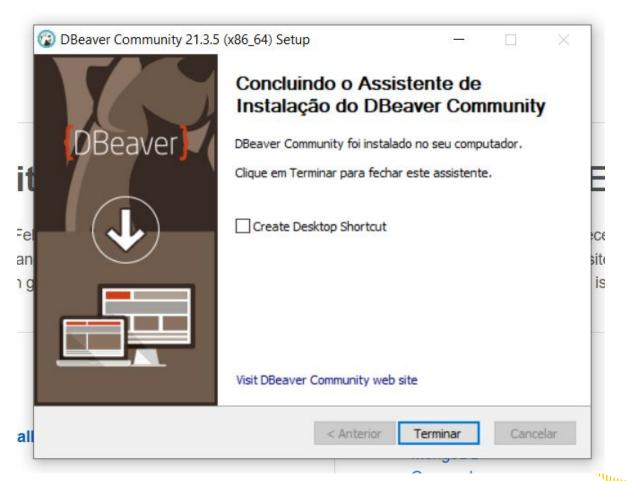
Mac OS X

- Mac OS X Intel (dmg)
- . Mac OS X M1 (dmg) beta
- Brew Cask (hrew install -- cask dheaver-community)

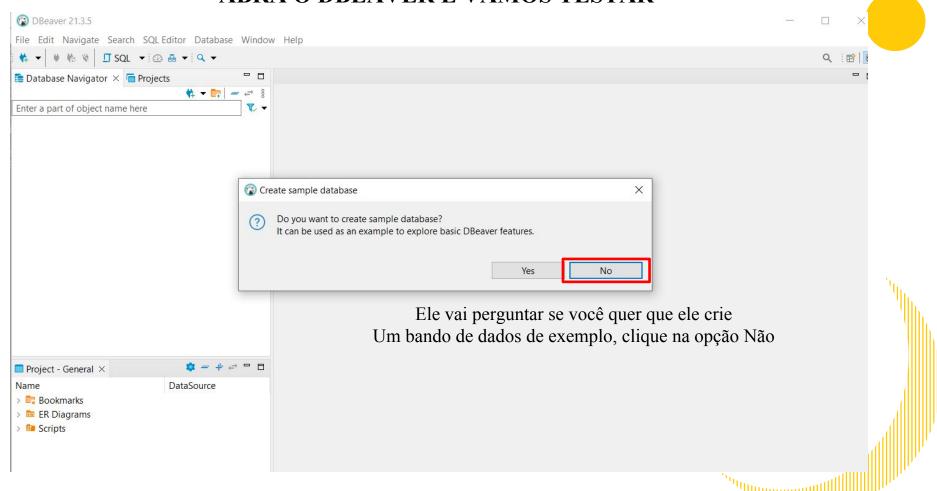
Enterprise Edition features:

- · Support of NoSQL databases:
 - MongoDB
 - Cassandra
 - InfluxDB
 - Redis
 - Amazon DynamoDB
 - Amazon DocumentDBAmazon Keyspaces
 - Google Bigtable
 - Couchbase
 - CouchDB
 - Yugabyte

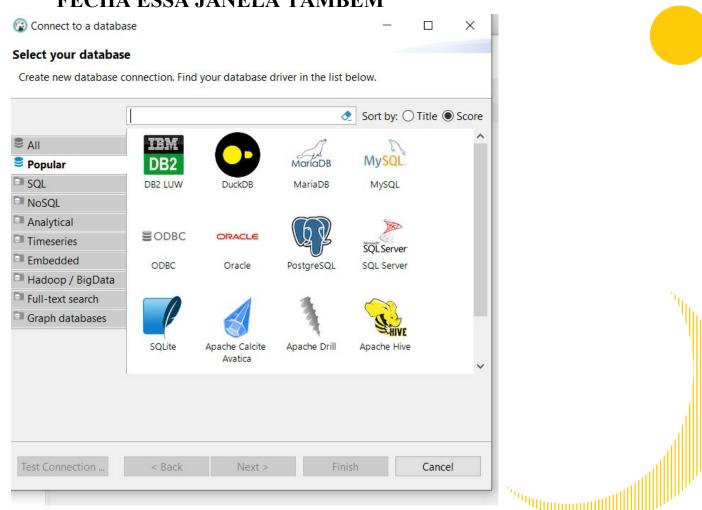
FAÇA A INSTALAÇÃO NORMAL



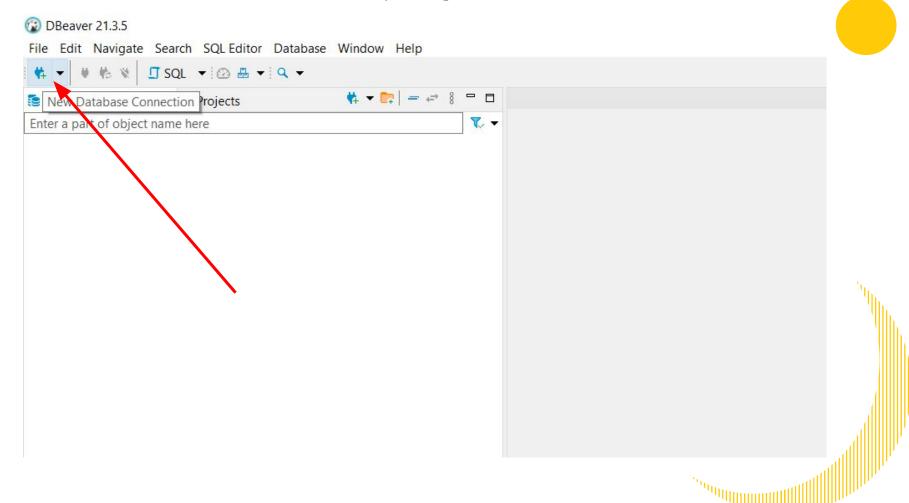
ABRA O DBEAVER E VAMOS TESTAR



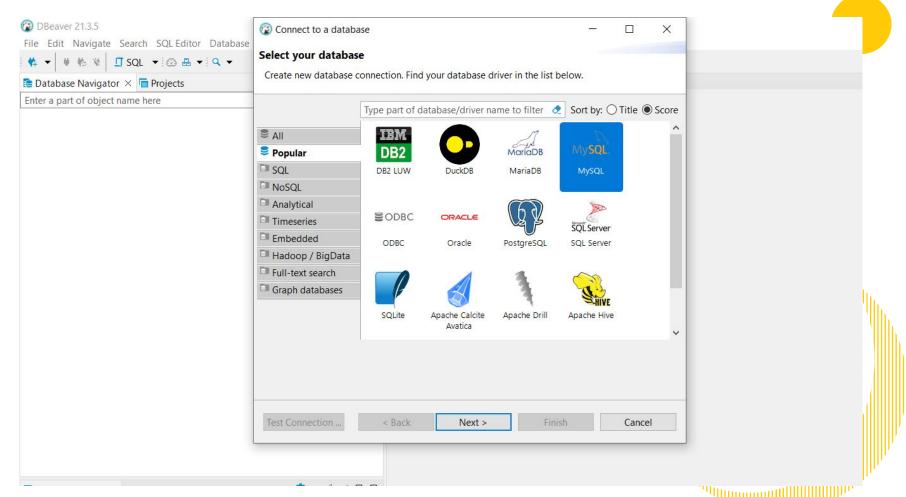
FECHA ESSA JANELA TAMBÉM



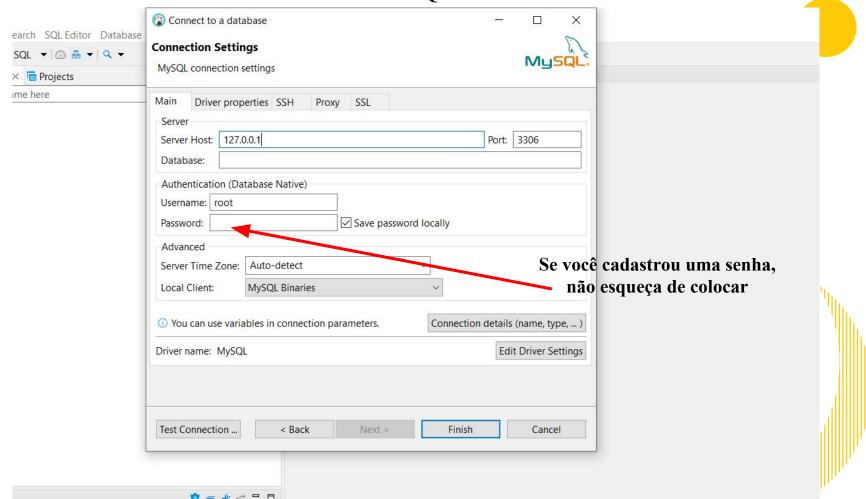
COMO DBEAVER ABERTO, CLIQUE EM NOVA CONEXÃO



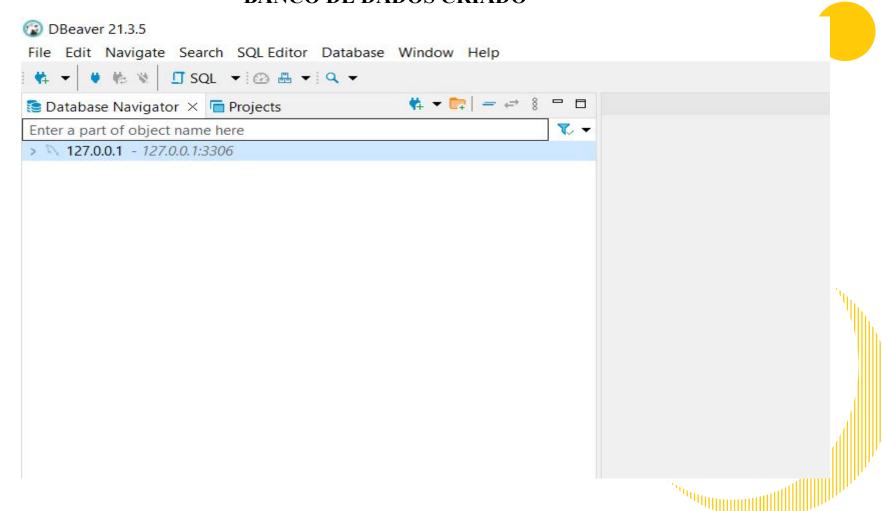
SELECIONE A OPÇÃO MYSQL



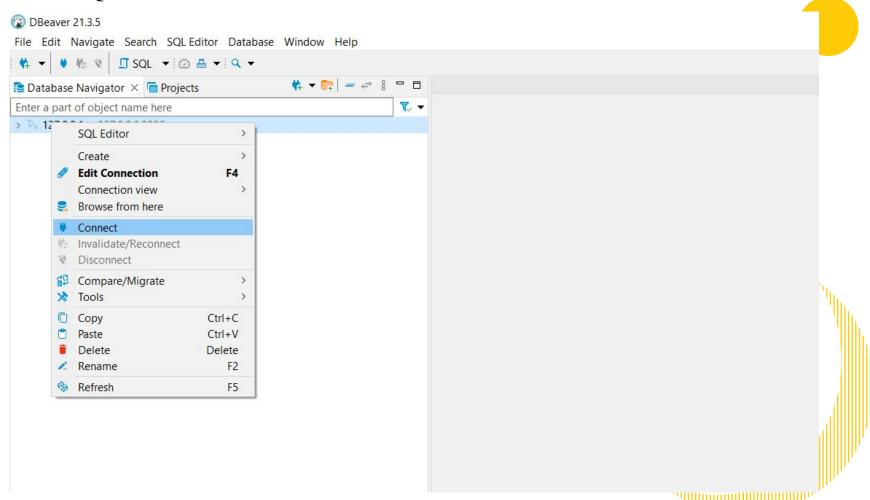
DEIXE DESSE MODO E CLIQUE EM FINISH



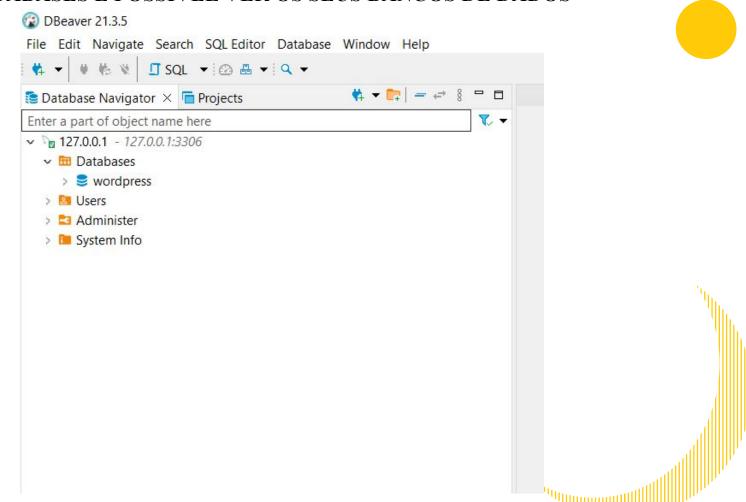
BANCO DE DADOS CRIADO



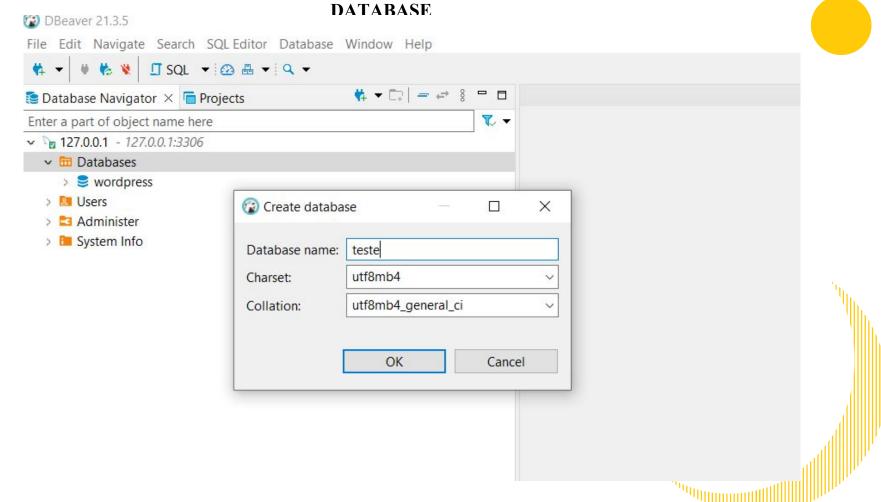
CLIQUE COM O BOTÃO DIREITO E CONECTE COM O BANCO



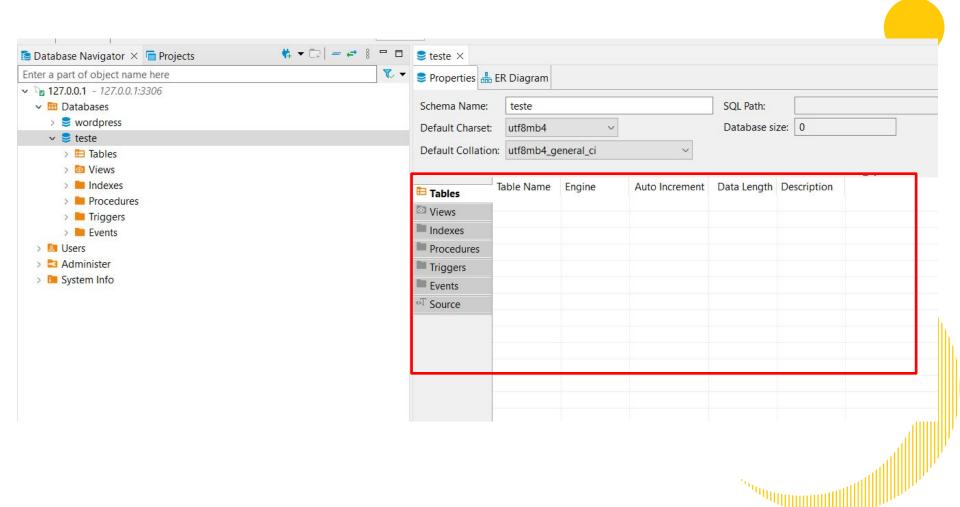
EM DATABASES É POSSÍVEL VER OS SEUS BANCOS DE DADOS



CASO QUEIRA CRIAR UM NOVO BANCO, CLIQUE COM O BOTÃO DIREITO E CREATE

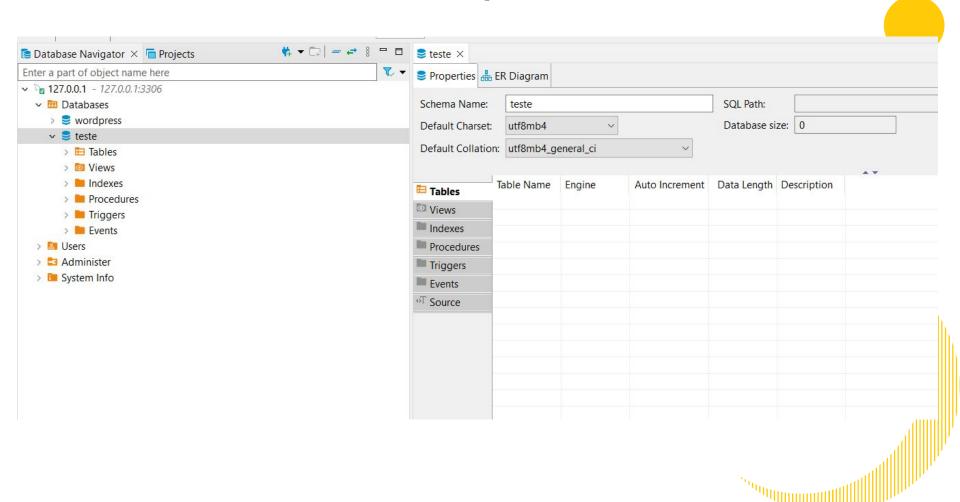


CLIQUE DUAS VEZES NO BANCO CRIADO E VEJA AS ESPECIFICAÇÕES DELE

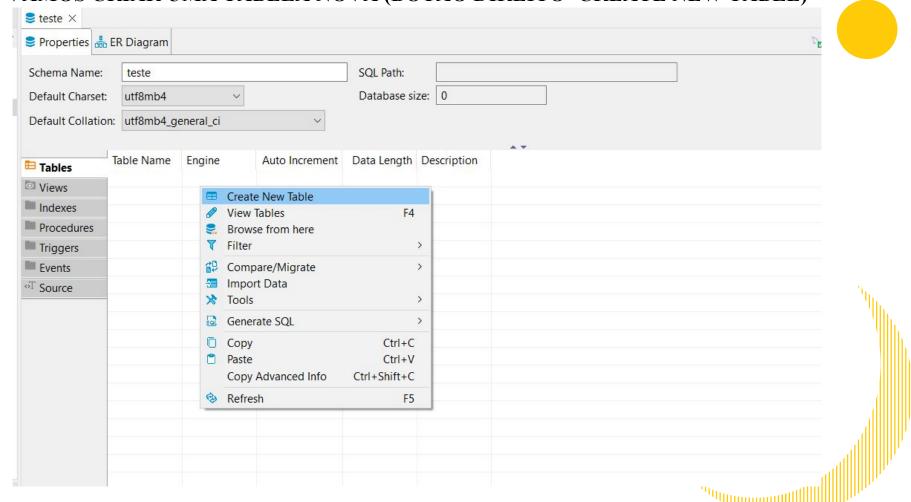


CRIANDO UM BD EXEMPLO NO MYSQL

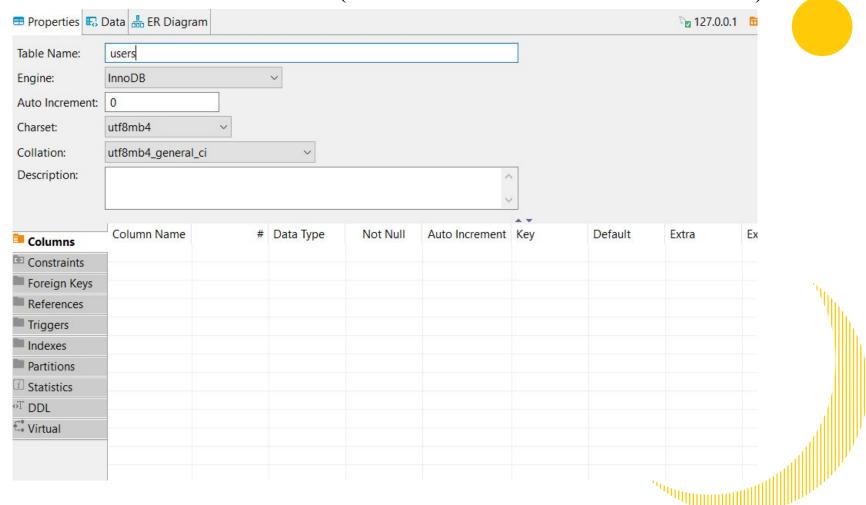
VAMOS USAR O BANCO TESTE QUE ACABAMOS DE CRIAR



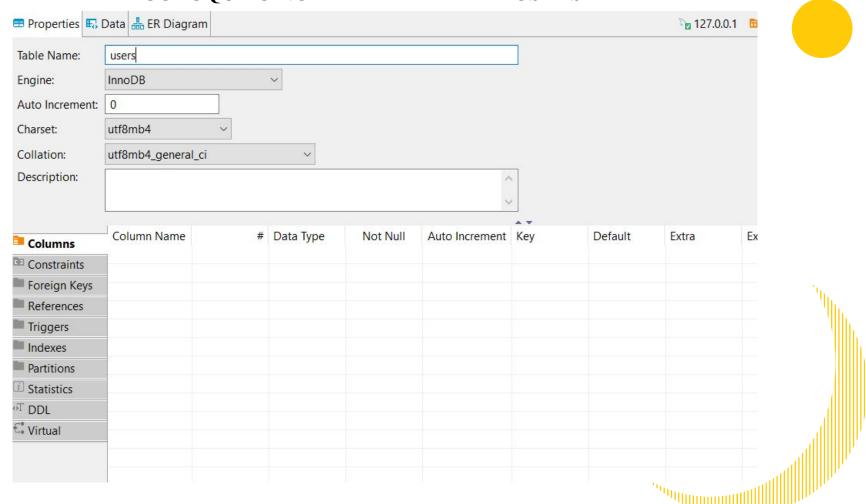
VAMOS CRIAR UMA TABELA NOVA (BOTÃO DIREITO>CREATE NEW TABLE) steste X



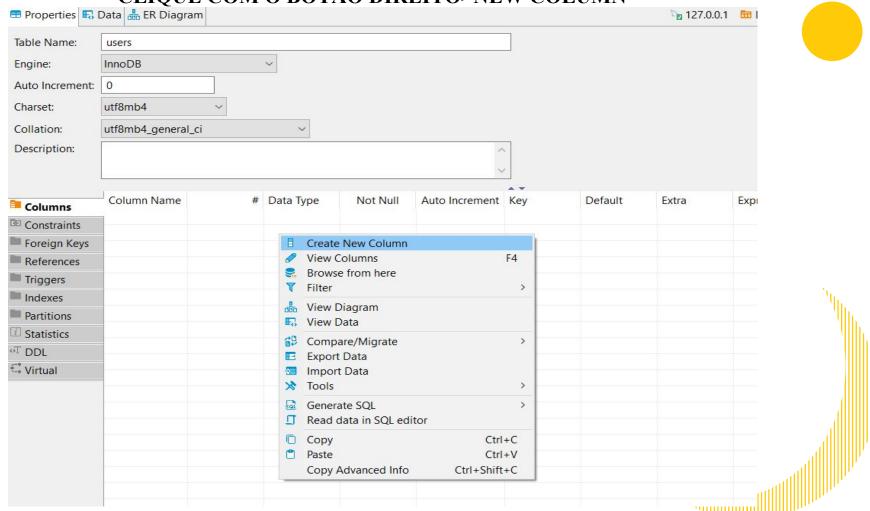
VAMOS CRIAR UMA TABELA NOVA (BOTÃO DIREITO>CREATE NEW TABLE)



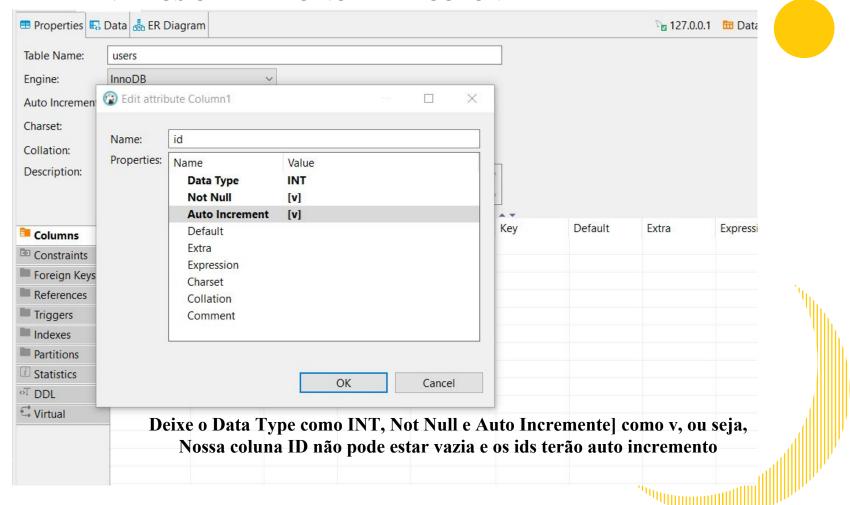
COLOQUE O NOME DA TABELA DE USERS



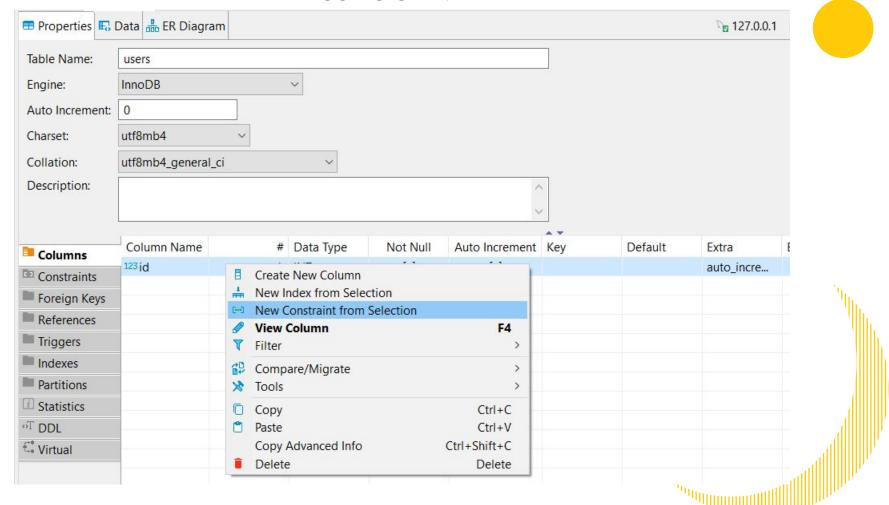
CLIQUE COM O BOTÃO DIREITO>NEW COLUMN



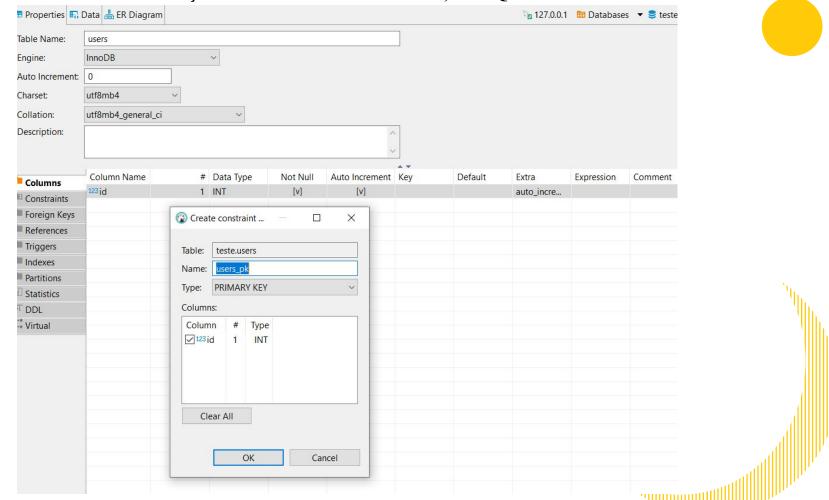
VAMOS CHAMAR O NOME DA COLUNA DE ID



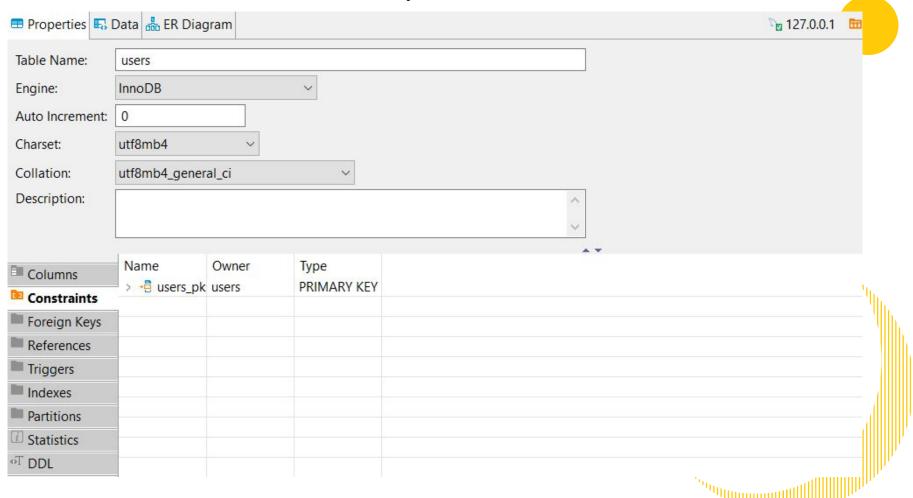
DEIXAR A ID COMO CHAVE PRIMÁRIA



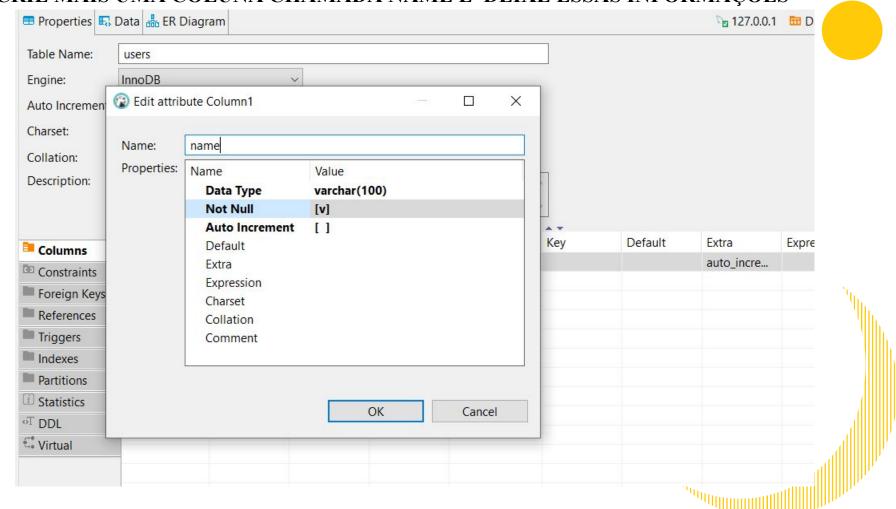
AS INFORMAÇÕES JÁ ESTÃO CERTAS, CLIQUE EM OK



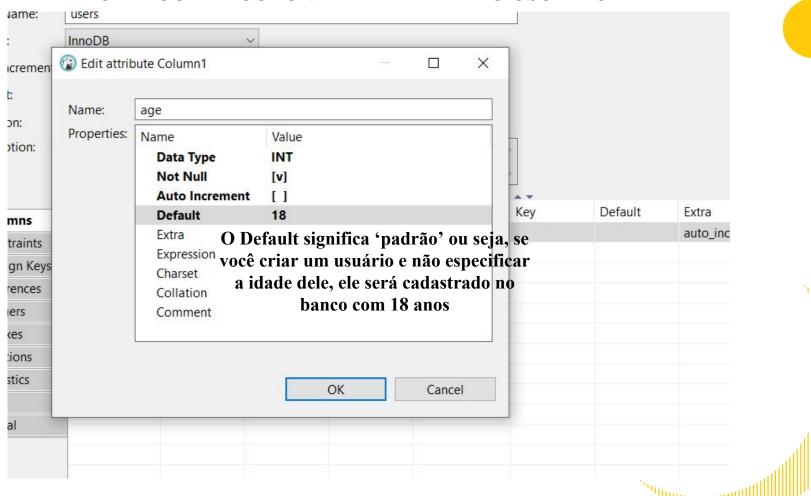
NA ABA CONSTRAINTS A INFORMAÇÃO CHAVE PRIMÁRIA JÁ APARECE



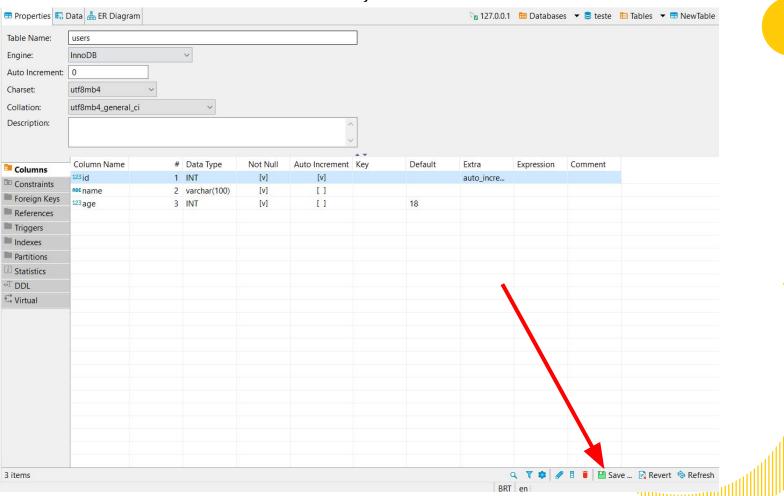
CRIE MAIS UMA COLUNA CHAMADA NAME E DEIXE ESSAS INFORMAÇÕES



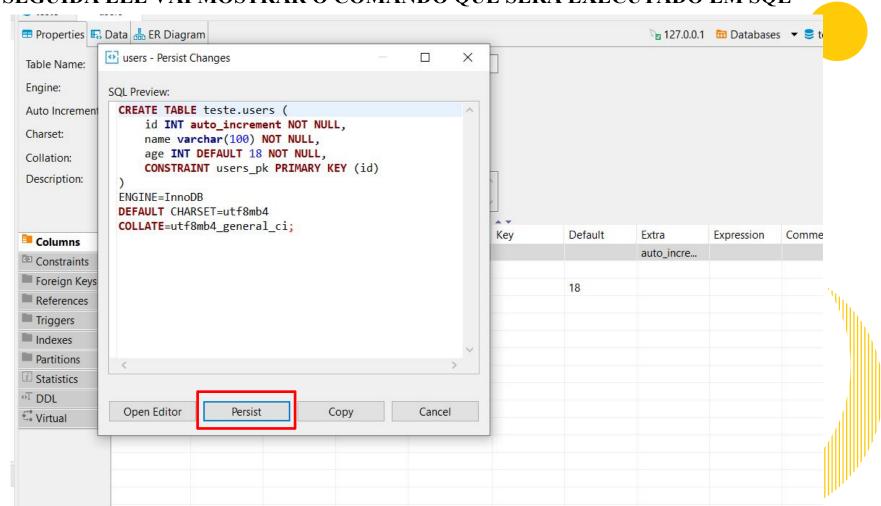
CRIE OUTRA COLUNA PARA IDADE DO USUÁRIO



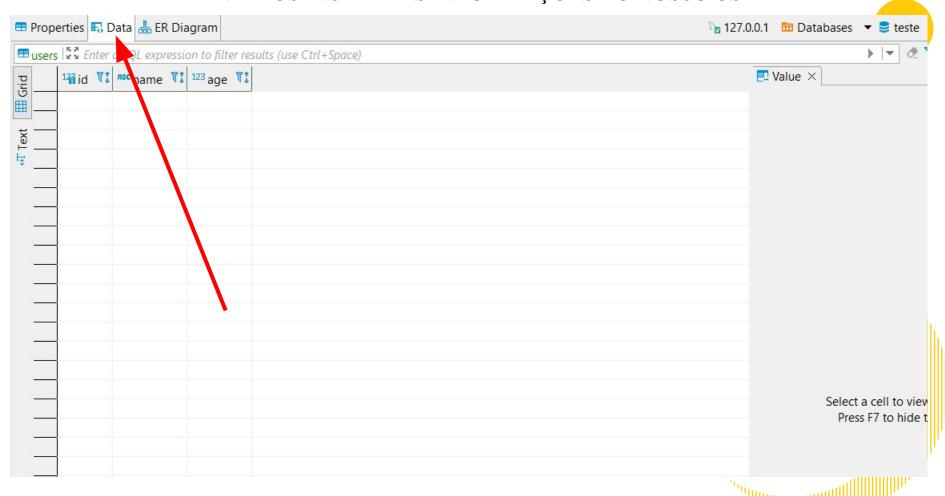
VAMOS SALVAR AS INFORMAÇÕES DA NOSSA TABELA



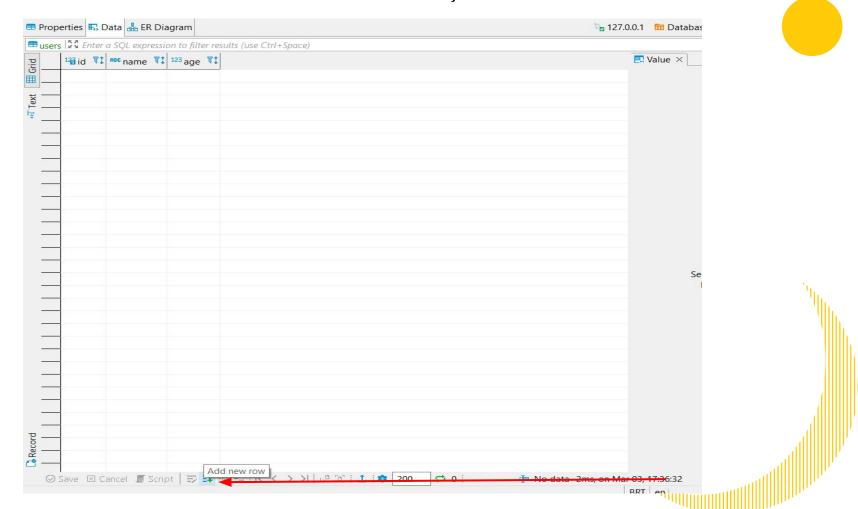
EM SEGUIDA ELE VAI MOSTRAR O COMANDO QUE SERÁ EXECUTADO EM SQL



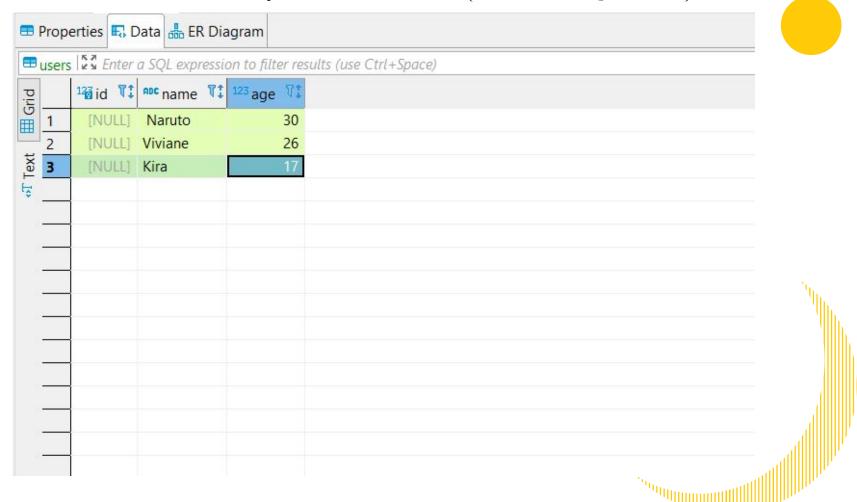
EM DATA VAMOS INSERIR AS INFORMAÇÕES DO NOSSO USER



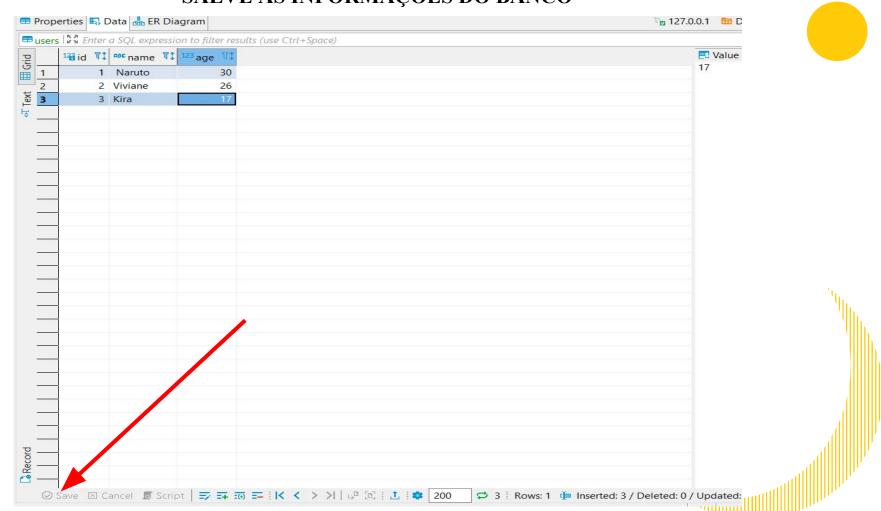
EM DATA VAMOS INSERIR AS INFORMAÇÕES DO NOSSO USER



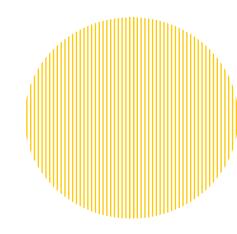
ADICIONE AS INFORMAÇÕES DO USUÁRIO (NÃO COLOQUE O ID)



SALVE AS INFORMAÇÕES DO BANCO

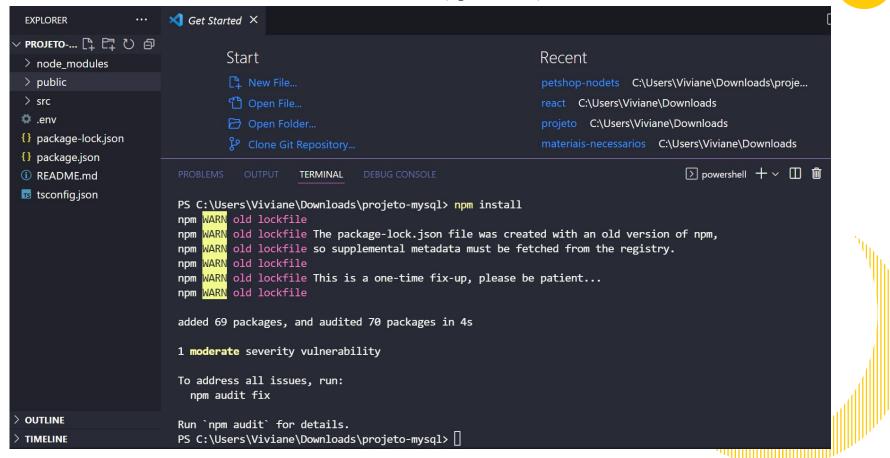


PROJETO BASE PARA SEGUIR AS AULAS

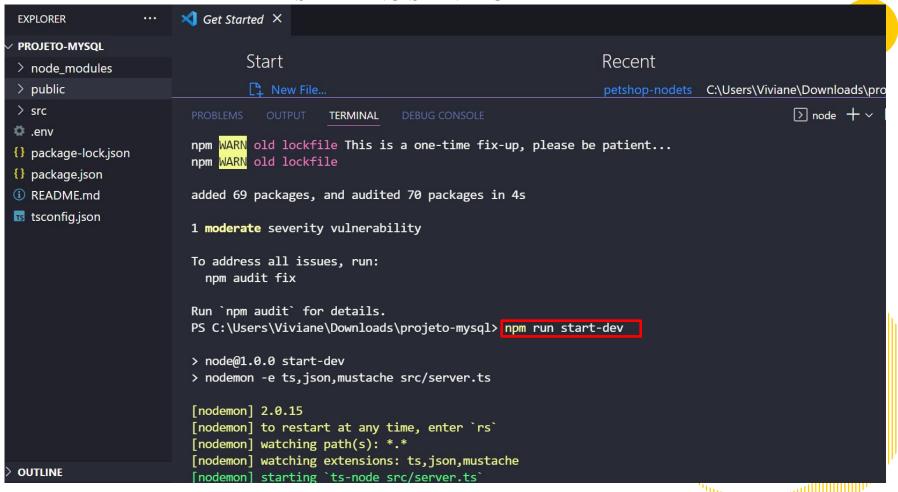


PARA SEGUIRMOS JUNTOS...

A professora vai disponibilizar um arquivo pronto para seguirmos juntos nas próximas aulas, baixe o arquivo e instale o node modules (**npm install**)



START NO SERVIDOR



VEJA SE ESTÁ FUNCIONANDO





Novo Título

C

Opa Viviane de Lima, tudo bem? Como vai a vida?

localhost:2000

Produtos

- Produto X R\$ 10
- Produto Y R\$ 15
- Produto W R\$ 20
- Produto G R\$ 5

Produtos a partir de R\$ 12

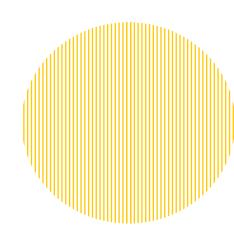
- Produto Y R\$ 15
- Produto W R\$ 20

Não há frases motivacionais hoje.

Todos os direitos reservados



ORM



O QUE É UM ORM?

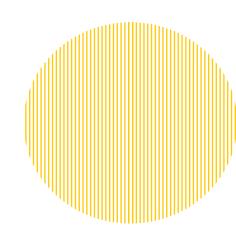
Existem **várias formas** de você conectar seu código ao banco de dados, uma dessas formas é usando o **ORM (Object Relational Mapper).** Existem algumas bibliotecas em que você instala no seu sistema e você cria uma "cópia" do banco de dados no seu código.



Por que eu vou repetir o código se eu já fiz ele no MySql?

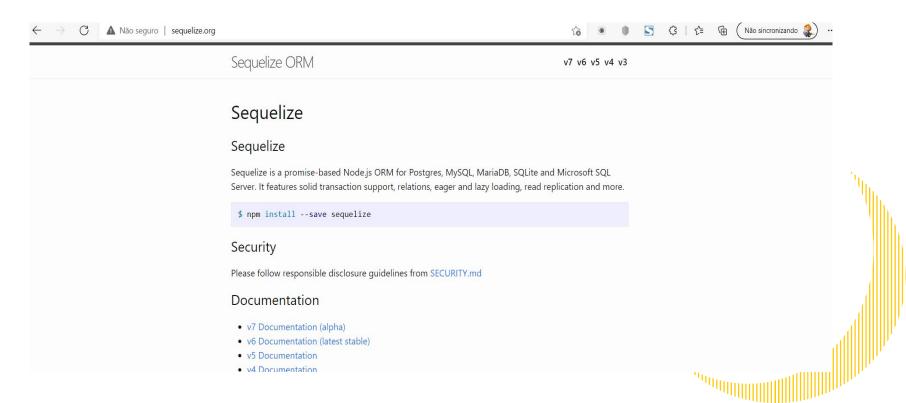
Você vai repetir a informação no seu código, para que o seu código entenda como é que funciona o banco de dados. Dessa forma seu código já vai saber qual é o banco, qual é a tabela etc. Então você ensina o seu código como é o seu banco de dados, e uma vez que ele aprendeu ele consegue executar tranquilamente, isso torna o uso do banco de dados mais fácil.

INTRODUÇÃO AO SEQUELIZE



O QUE É O SEQUELIZE?

O Sequelize é uma das bibliotecas mais utilizadas no mercado, vários sistemas utilizam o sequelize. Ele é uma biblioteca promise-based, ou seja você manda uma coisa e fica esperando o resultado dessa coisa (Exemplo: mando uma solicitação para o banco de dados, ele retorna para mim e em seguida eu faço o que eu quiser)



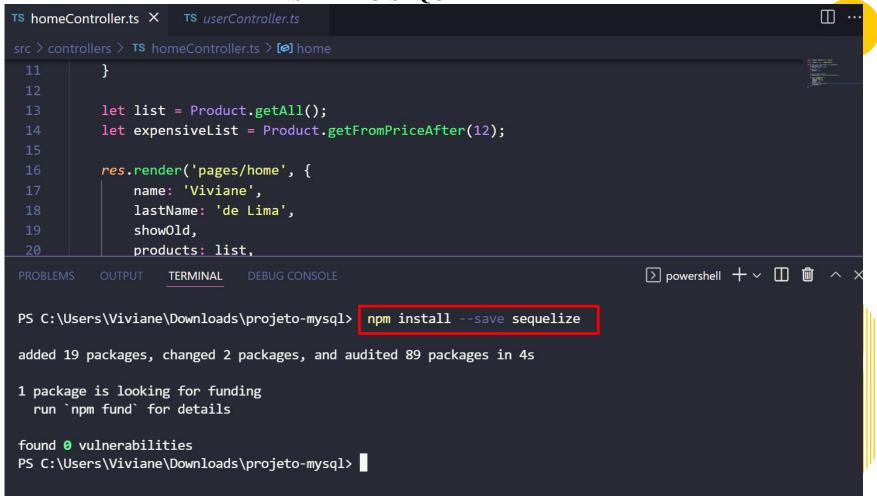
MYSQL = ORM

id	name	price	stock
1	Nike shoes	123.45	35
2	Reebok sneakers	110.34	45

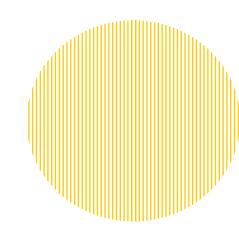
```
id: 1,
name: 'Nike shoes',
price: 123.45,
stock: 35
id: 3,
name: 'Reebok sneakers',
price: 110.34,
stock: 45
```

Através do ORM conseguimos fazer um mapeamento para evitar o uso do SQL no código

INSTALE O SEQUELIZE



SEQUELIZE E TYPESCRIPT

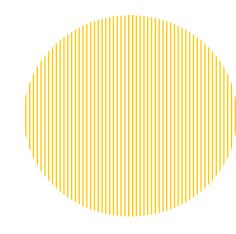


PARA CONSEGUIRMOS USAR O SEQUELIZE COM TYPESCRIPT PRECISAMOS INSTALAR OS TYPES DO SEQUELIZE

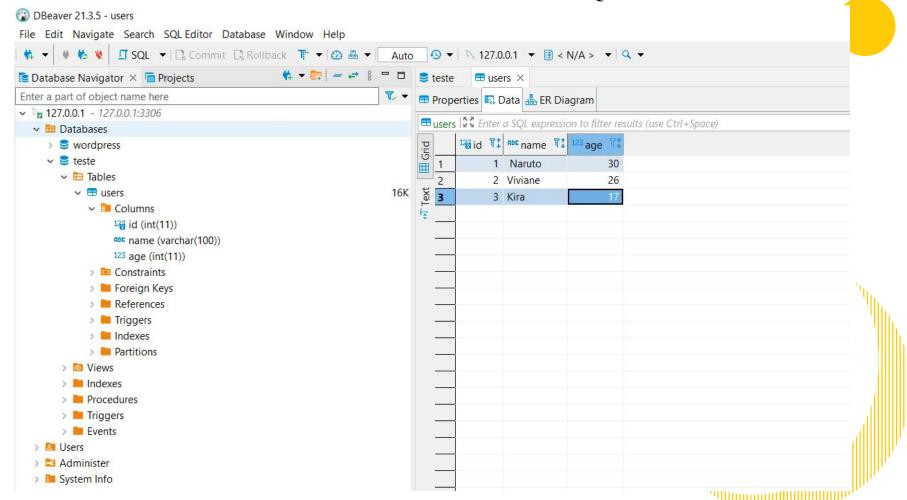




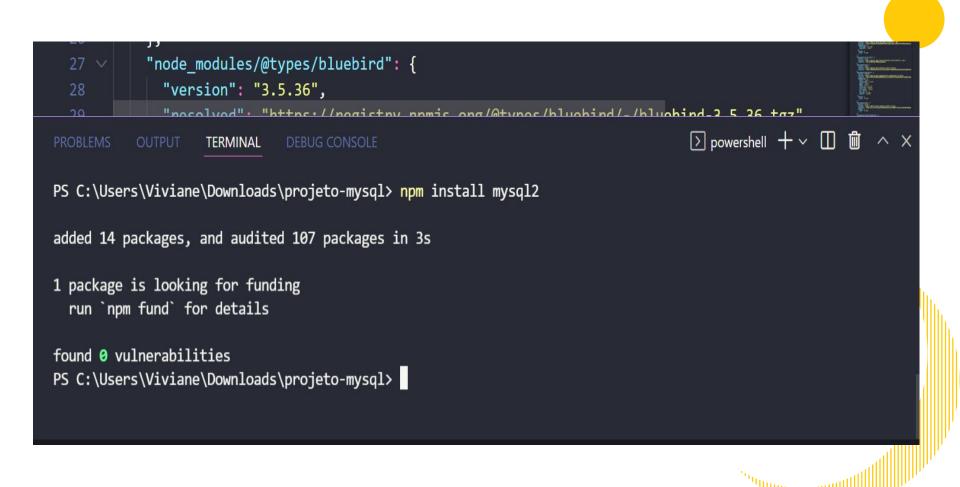
CONECTANDO AO BANCO MYSQL



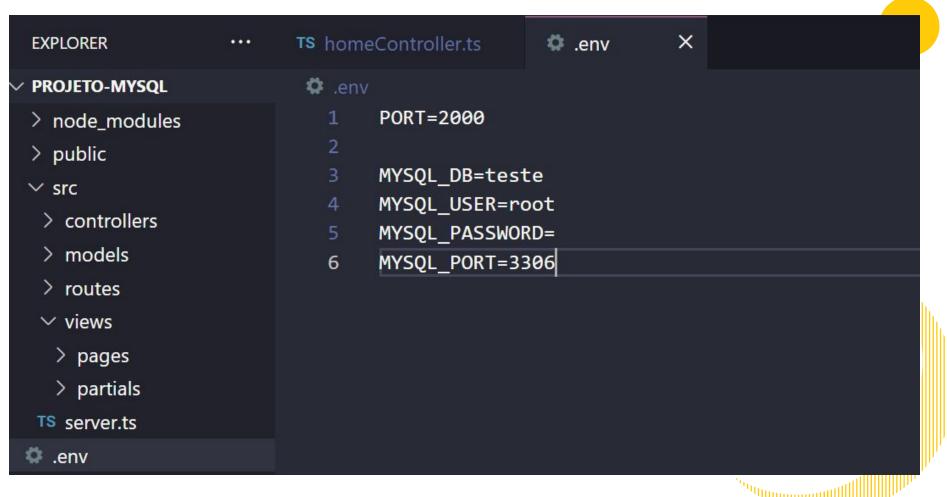
VAMOS CONECTAR NO BANCO DE DADOS TESTE QUE CRIAMOS



INSTALE A BIBLIOTECA DE GERENCIAMENTO DO MYSQL NO SEU PROJETO

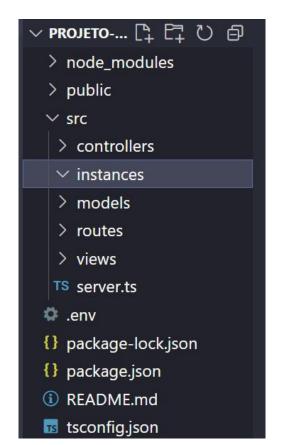


VAMOS USAR O .ENV PARA CONFIGURAR NOSSO BANCO DE DADOS

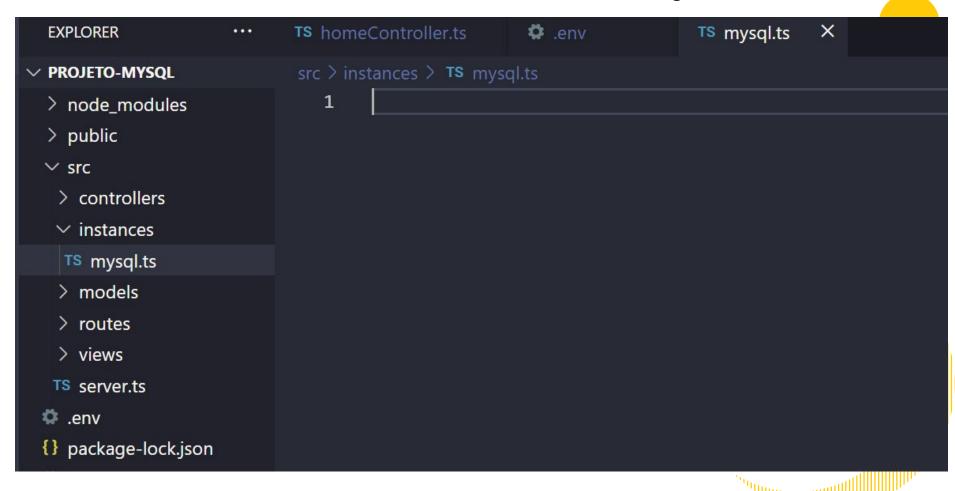


EM SEGUIDA CRIE UMA PASTA DENTRO DE SRC CHAMADA INSTANCES ESSA PASTA VAI SERVIR PARA COLOCARMOS NOSSO BANCO DE DADOS, PODE SER QUE VOCÊ TENHA MAIS DE UM BANCO PARA O SEU SISTEMA, POR ISSO É IMPORTANTE A CRIAÇÃO DESSA PASTA





DENTRO DA PASTA INSTANCES CRIE O MYSQL.TS



INICIANDO A CONFIGURAÇÃO

EXPLORER	•••	🌣 .env	TS mysql.ts X	
✓ PROJETO-MYSQL		src > ins	tances > TS mysql.ts > [�] sequelize	
> node_modules			//importando o sequelize	
> public			<pre>import { Sequelize } from 'sequelize'</pre>	
∨ src				
> controllers			//importando o dotenv para acessar as variáveis de ambiente	
✓ instances		5 6	import dotenv from 'dotenv'	
TS mysql.ts			<pre>dotenv.config()</pre>	
> models				
> routes		9	//importando uma constante que será a nossa conexão	
> views		10	export const sequelize = new Sequelize(
TS server.ts		11		
.env		12 13	/*aqui vamos receber os parâmetros para nosso banco de dados precisamos falar para o typescript que vamos receber uma string*/	\t <u>i.</u>
{} package-lock.json		14	process.env.MYSQL DB as string,	կի,
{} package.json		15	process.env.MYSQL_USER as string,	()
① README.md		16	process.env.MYSQL_PASSWORD as string,	
stsconfig.json			{	
		18	//aqui é o banco que estou usando que é mysql	,
		19	dialect:'mysql',	.(1111
		20 21	/*aqui é a porta (o typescript espera um NUMBER, entao devemos usar o parseInt) */	اااااان
		21	port: parseInt(process.env.MYSQL_PORT as string)	ااااااااا
> OUTLINE		23	}	
> TIMELINE		24	bl	
THVILLINL				l

VAMOS TESTAR A CONEXÃO EM HOMECONTROLLER USANDO O

```
env env
                                TS homeController.ts X
src > controllers > TS homeController.ts > [] home
       import { Request, Response } from 'express';
       import { Product } from '../models/Product';
       import { sequelize } from '../instances/mysql';
       export const home = async (req: Request, res: Response)=>{
           //usar o AUTENTICATE para ver se deu tudo certo
           try{
                await sequelize.authenticate()
                console.log("Conexão estabelecida com sucesso!")
            }catch(error){
                console.log("Deu problema: ",error)
           let age: number = 90;
           let showOld: boolean = false;
           if(age > 50) {
                showOld = true:
           let list = Product.getAll();
            let expensiveList = Product.getFromPriceAfter(12);
```

RODE NPM START-DEV E ABRA O NAVEGADOR





Novo Título

Opa Viviane de Lima, tudo bem? Como vai a vida?

Produtos

- Produto X R\$ 10
- · Produto Y R\$ 15
- Produto W R\$ 20
- · Produto G R\$ 5

Produtos a partir de R\$ 12

- Produto Y R\$ 15
- Produto W R\$ 20

Não há frases motivacionais hoje.

Todos os direitos reservados



VEJA NO TERMINAL SE A MENSAGEM DE CONEXÃO É EXIBIDA

```
.env
                           TS homeController.ts X
src > controllers > TS homeController.ts > [@] home
       import { Request, Response } from 'express';
       import { Product } from '../models/Product';
       import { sequelize } from '../instances/mysql';
       export const home = async (req: Request, res: Response)=>{
           //usar o AUTENTICATE para ver se deu tudo certo
           try{
PROBLEMS
                    TERMINAL
 [nodemon] 2.0.15
 [nodemon] to restart at any time, enter `rs`
 [nodemon] watching path(s): *.*
 [nodemon] watching extensions: ts,json,mustache
 [nodemon] starting `ts-node src/server.ts`
Executing (default): SELECT 1+1 AS result
Conexão estabelecida com sucesso!
```

<u>antillillillin</u>