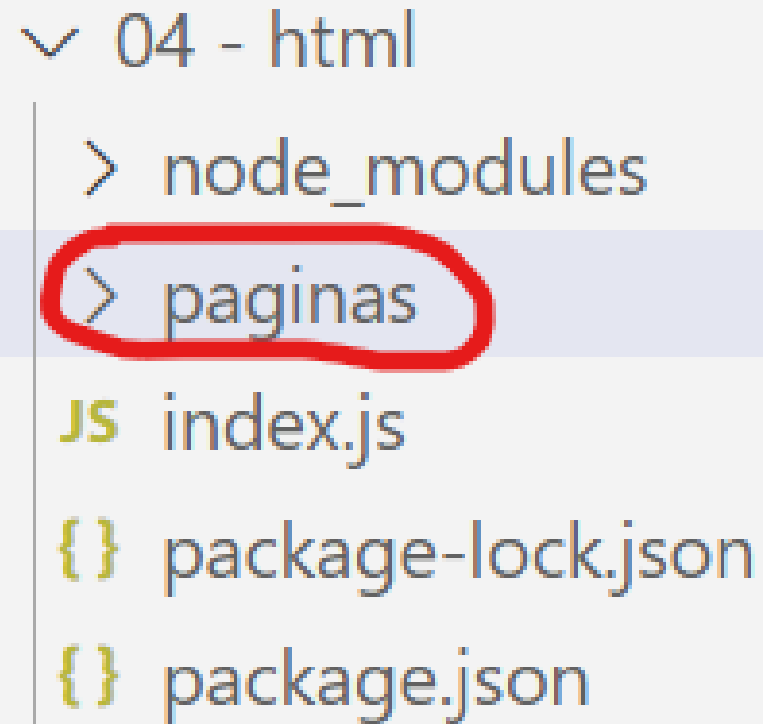




NODE.JS

IW-I – Prof. Anderson Vanin

- Crie uma nova pasta dentro do diretório raiz da sua aplicação.



```

 04 - html
├── node_modules
├── paginas
├── index.js
├── package-lock.json
└── package.json

```

A screenshot of a file explorer interface. The directory structure is as follows: a folder named '04 - html' is expanded, showing its contents. Inside '04 - html', there are four items: a folder 'node_modules', a folder 'paginas' (which is highlighted with a red circle), a file 'index.js' with a JavaScript icon, and two files 'package-lock.json' and 'package.json' with JSON icons.

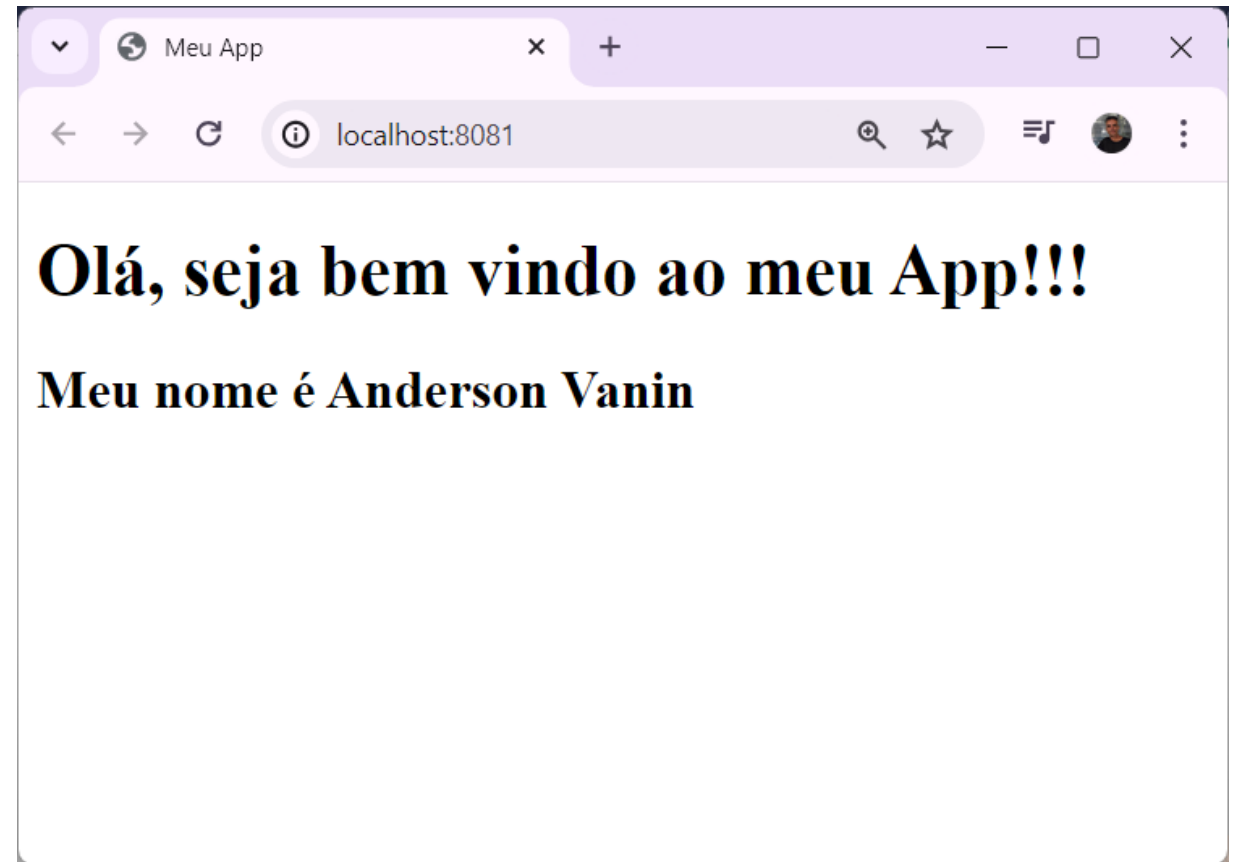
- Dentro da pasta páginas, crie um arquivo *index.html*.

04 - html > paginas > <> index.html >  html

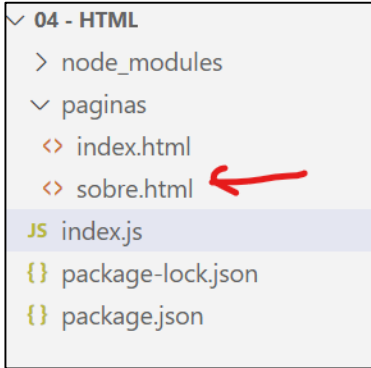
```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Meu App</title>
7  </head>
8  <body>
9      <h1>Olá, seja bem vindo ao meu App!!!</h1>
10     <h2>Meu nome é Anderson Vanin</h2>
11 </body>
12 </html>
```

- Altere a rota inicial do arquivo *index.js*. Rode a aplicação digitando na linha de comando:
- **nodemon index.js**

```
JS index.js > ...
1  const express = require('express')
2  const app = express()
3
4  app.get("/",function(req,res){
5    // __dirname retorna o diretorio raiz da sua aplicação
6    res.sendFile(__dirname + "/paginas/index.html") ←
7  })
8
9  app.get("/sobre",function(req,res){
10   res.send("Pagina Sobre")
11 })
12
13 app.get("/blog",function(req,res){
14   res.send("Pagina do Blog")
15 })
16
17 app.get("/ola/:cargo/:nome",function(req,res){
```




- Crie a página *sobre.html*.



```
paginas > <> sobre.html > html
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Página Sobre</title>
7  </head>
8  <body>
9    <h1>Essa é a página Sobre</h1>
10 </body>
11 </html>
```






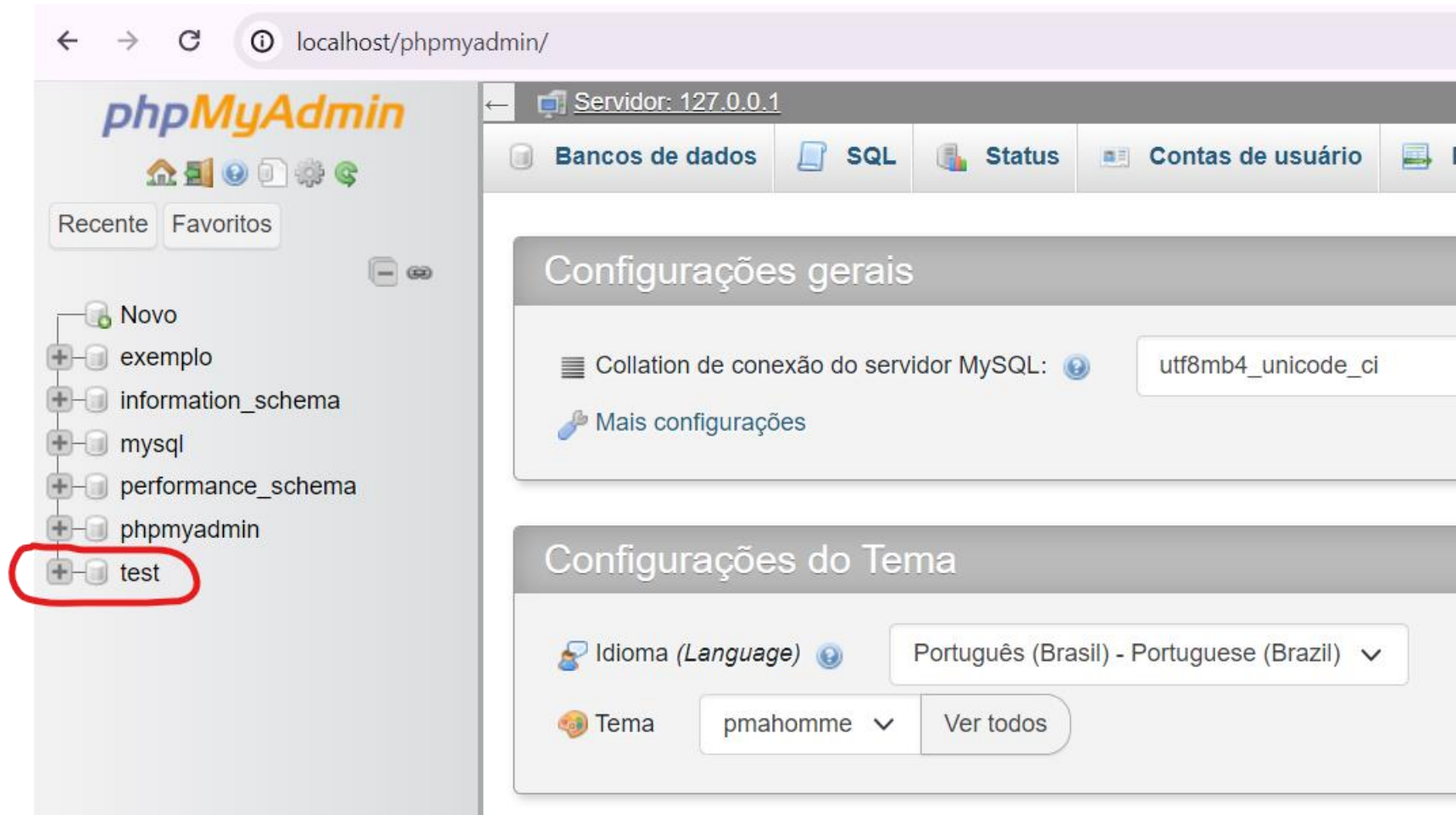
Sequelize é um ORM moderno de TypeScript e Node.js para Oracle, Postgres, MySQL, MariaDB, SQLite e SQL Server e muito mais. Apresentando suporte sólido a transações, relações, carregamento rápido e lento, replicação de leitura e muito mais.

<https://sequelize.org/>

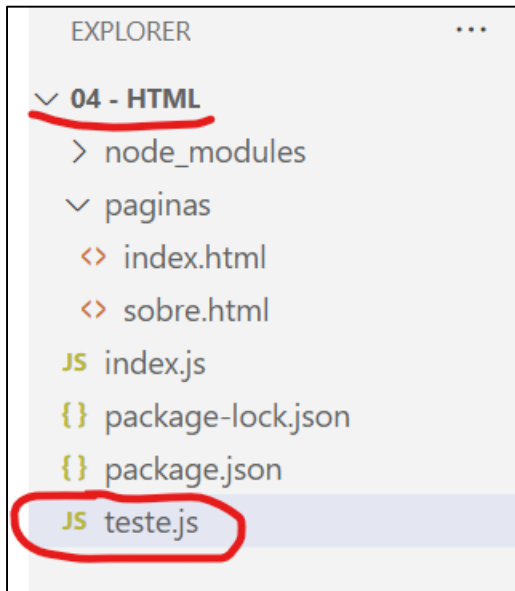
<https://www.npmjs.com/package/sequelize>

- 
- `npm install --save sequelize`
 - `npm install --save mysql2`

- Vamos fazer um pequeno teste para se conectar a uma base de dados de teste do mysql chamado test



- Crie um arquivo chamado teste.js

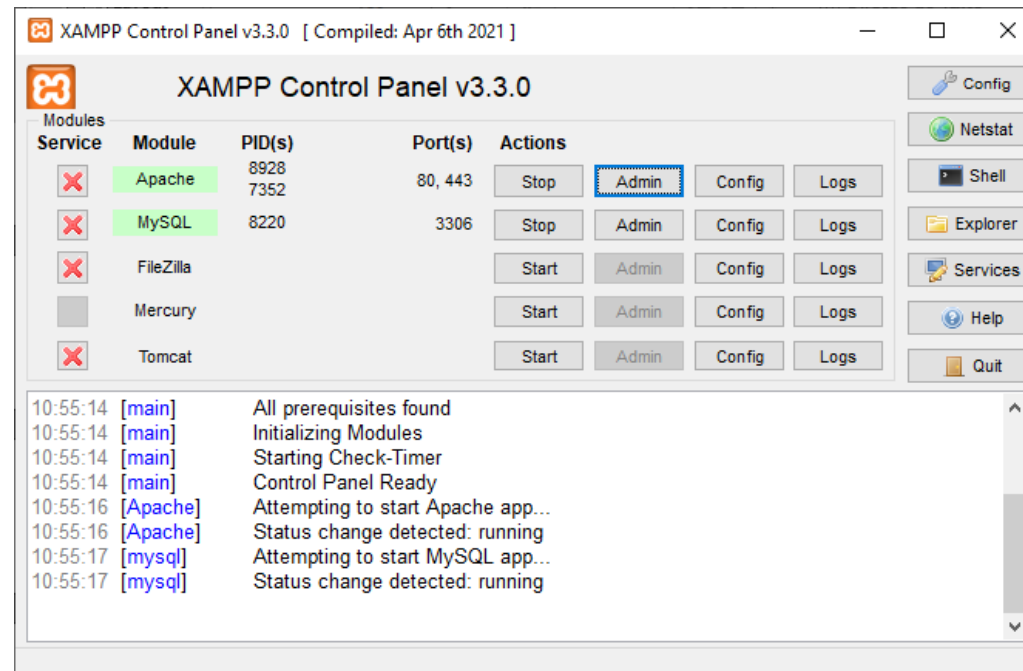


```
JS teste.js > ...
1  const Sequelize = require('sequelize')
2  const sequelize = new Sequelize('test','root','',{
3      host: "localhost",
4      dialect: 'mysql'
5  })
6
7  sequelize.authenticate().then(function(){
8      console.log("CONECTADO COM SUCESSO!")
9  }).catch(function(erro){
10     console.log("FALHA AO CONECTAR !" + erro)
11 })
```

```
C:\Users\Anderson\Desktop\2024 - ETEC\Curso_Node\04 - html>nodemon teste.js
[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node teste.js`
Executing (default): SELECT 1+1 AS result
CONECTADO COM SUCESSO!
```

MODELS NO SEQUELIZE

- Agora vamos criar um modelo que será utilizado para a criação do database e das tabelas no mysql. Vamos criar dois módulos: Postagem e Usuario.
- Ajuste o arquivo teste.js
- Deixe o XAMPP aberto e rodando o APACHE E MYSQL



MODELS NO SEQUELIZE

```
JS teste.js > ...
1  const { type } = require('express/lib/response')
2  const Sequelize = require('sequelize')
3  const sequelize = new Sequelize('test', 'root', '', {
4    host: "localhost",
5    dialect: 'mysql'
6  })
7
8  // sequelize.authenticate().then(function(){
9  //   console.log("CONECTADO COM SUCESSO!")
10 // }).catch(function(erro){
11 //   console.log("ERRO NA CONEXÃO: " + erro)
12 // })
13
14 const Postagem = sequelize.define('postagens', {
15   titulo: {
16     type: Sequelize.STRING
17   },
18   conteudo: {
19     type: Sequelize.TEXT
20   }
21 })
22
```

```
22
23 const Usuario = sequelize.define('usuarios', {
24   nome: {
25     type: Sequelize.STRING
26   },
27   sobrenome: {
28     type: Sequelize.STRING
29   },
30   idade: {
31     type: Sequelize.INTEGER
32   },
33   email: {
34     type: Sequelize.STRING
35   }
36 })
37
38
39 Postagem.sync()
40 Usuario.sync()
```

Rode o arquivo teste.js **APENAS UMA VEZ!**

MODELS NO SEQUELIZE

```
C:\Users\LAB4\Downloads\IW-II_2024-main\IW-II_2024-main\bimestre 2\curso_node\04 - Html>nodemon teste.js
```

```
[nodemon] 3.1.3
```

```
[nodemon] to restart at any time, enter `rs`
```

```
[nodemon] watching path(s): *.*
```

```
[nodemon] watching extensions: js,mjs,cjs,json
```

```
[nodemon] starting `node teste.js`
```

```
Executing (default): SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_NAME = 'postagens' AND TABLE_SCHEMA = 'test'
```

```
Executing (default): SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_NAME = 'usuarios' AND TABLE_SCHEMA = 'test'
```

```
Executing (default): SHOW INDEX FROM `usuarios`
```

```
Executing (default): SHOW INDEX FROM `postagens`
```

```
[nodemon] clean exit - waiting for changes before restart
```

MODELS NO SEQUELIZE

phpMyAdmin

Recente Favoritos

- Novo
- information_schema
- mysql
- performance_schema
- phpmyadmin
- test
 - Nova
 - postagens
 - usuarios

Servidor: 127.0.0.1 » Banco de dados: test » Tabela: postagens

Procurar Estrutura SQL Pesquisar Inserir Exportar Importar Privilégios Operações Rastrea

Estrutura da tabela Visão de relação(ões)

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1 id	int(11)			Não	Nenhum		AUTO_INCREMENT	Muda Elimina Mais
<input type="checkbox"/>	2 titulo	varchar(255)	latin1_swedish_ci		Sim	NULL			Muda Elimina Mais
<input type="checkbox"/>	3 conteudo	text	latin1_swedish_ci		Sim	NULL			Muda Elimina Mais
<input type="checkbox"/>	4 createdAt	datetime			Não	Nenhum			Muda Elimina Mais
<input type="checkbox"/>	5 updatedAt	datetime			Não	Nenhum			Muda Elimina Mais

Servidor: 127.0.0.1 » Banco de dados: test » Tabela: usuarios

Procurar Estrutura SQL Pesquisar Inserir Exportar Importar Privilégios Operações Rastre

Estrutura da tabela Visão de relação(ões)

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1 id	int(11)			Não	Nenhum		AUTO_INCREMENT	Muda Elimina Mais
<input type="checkbox"/>	2 nome	varchar(255)	latin1_swedish_ci		Sim	NULL			Muda Elimina Mais
<input type="checkbox"/>	3 sobrenome	varchar(255)	latin1_swedish_ci		Sim	NULL			Muda Elimina Mais
<input type="checkbox"/>	4 idade	int(11)			Sim	NULL			Muda Elimina Mais
<input type="checkbox"/>	5 email	varchar(255)	latin1_swedish_ci		Sim	NULL			Muda Elimina Mais
<input type="checkbox"/>	6 createdAt	datetime			Não	Nenhum			Muda Elimina Mais
<input type="checkbox"/>	7 updatedAt	datetime			Não	Nenhum			Muda Elimina Mais

MODELS NO SEQUELIZE

- Após a execução deixe comentado as últimas linhas do arquivo teste.js para não executar a criação das tabelas.

```
21  })
22
23  const Usuario = sequelize.define('usuari
24  {
25    nome:{
26      type:Sequelize.STRING
27    },
28    sobrenome:{
29      type: Sequelize.STRING
30    },
31    idade:{
32      type:Sequelize.INTEGER
33    },
34    email:{
35      type: Sequelize.STRING
36    }
37  })
38
39  //Postagem.sync()
40  //Usuario.sync()
```


MODELS NO SEQUELIZE

- Inserindo dados nas tabelas

```
13
14 const Postagem = sequelize.define('postagens',{
15     titulo:{
16         type: Sequelize.STRING
17     },
18     conteudo:{
19         type:Sequelize.TEXT
20     }
21 })
22
23 Postagem.create({
24     titulo: "UM TITULO QUALQUER",
25     conteudo: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ac neque ante. Vivamus in porttitor turpis. Fusce a dui quis mi ultricies luctus. Donec sollicitudin eros neque, non porta enim convallis sit amet. |"
26 })
27
28 const Usuario = sequelize.define('usuarios',{
29     nome:{
30         type:Sequelize.STRING
```

MODELS NO SEQUELIZE

- Inserindo dados nas tabelas

The screenshot displays the Sequelize web interface for a database named 'test' and a table named 'postagens'. The interface includes a top navigation bar with buttons for 'Procurar', 'Estrutura', 'SQL', 'Pesquisar', 'Inserir', 'Exportar', 'Importar', 'Privilegios', 'Operações', and 'Rastrear'. A green status bar indicates that 1 record is displayed. The SQL query 'SELECT * FROM `postagens`' is shown. Below the query, there are links for 'Perfil', 'Editar em linha', 'Edita', 'Explicar SQL', 'Criar código PHP', and 'Atualizar'. A control bar shows 'Mostrar tudo', 'Número de registros: 25', and a search filter 'Pesquisar esta tabela'. The table has columns: 'id', 'titulo', 'conteudo', 'createdAt', and 'updatedAt'. One record is listed with 'id' 1, 'titulo' 'UM TITULO QUALQUER', and 'conteudo' 'Lorem ipsum dolor sit amet, consectetur adipiscing...'. At the bottom, there are links for 'Editar', 'Copiar', 'Apagar', and 'Exportar' for the selected record.

Servidor: 127.0.0.1 » Banco de dados: test » Tabela: postagens

Procurar Estrutura SQL Pesquisar Inserir Exportar Importar Privilegios Operações Rastrear

✓ A mostrar registros de 0 - 0 (1 total, A consulta demorou 0,0005 segundos.)

`SELECT * FROM `postagens``

☐ Perfil [Editar em linha] [Edita] [Explicar SQL] [Criar código PHP] [Atualizar]

☐ Mostrar tudo | Número de registros: 25 ▼ Filtrar registros:

+ Opções

	id	titulo	conteudo	createdAt	updatedAt
<input type="checkbox"/> Editar Copiar Apagar	1	UM TITULO QUALQUER	Lorem ipsum dolor sit amet, consectetur adipiscing...	2024-06-05 14:36:52	2024-06-05 14:36:52

☐ Marcar todos Com os seleccionados: Editar Copiar Apagar Exportar

☐ Mostrar tudo | Número de registros: 25 ▼ Filtrar registros:

MODELS NO SEQUELIZE

- Inserindo dados nas tabelas

```
38     email:{
39         type: Sequelize.STRING
40     }
41 })
42
43 Usuario.create({
44     nome: "ANDERSON",
45     sobrenome: "VANIN",
46     idade: 49,
47     email: "anderson@email.com"
48 })
49
50 // Postagem.sync()
51 // Usuario.sync()
```

MODELS NO SEQUELIZE

- Inserindo dados nas tabelas

Servidor: 127.0.0.1 » Banco de dados: test » Tabela: usuarios

Procurar Estrutura SQL Pesquisar Inserir Exportar Importar Privilégios Operações




✓ A mostrar registros de 0 - 0 (1 total, A consulta demorou 0,0007 segundos.)






`SELECT * FROM `usuarios``

☐ Perfil [Editar em linha] [Edita] [Explicar SQL] [Criar código PHP] [Actualizar]

☐ Mostrar tudo | Número de registros: 25 ▼ Filtrar registros: Pesquisar esta tabela

+ Opções

	id	nome	sobrenome	idade	email	createdAt	updatedAt
<input type="checkbox"/>  Edita  Copiar  Apagar	1	ANDERSON	VANIN	49	anderson@email.com	2024-06-05 14:41:04	2024-06-05 14:41:04

 ☐ Marcar todos Com os seleccionados:  Edita  Copiar  Apagar  Exportar

☐ Mostrar tudo | Número de registros: 25 ▼ Filtrar registros: Pesquisar esta tabela



- Criar um banco de dados chamado: **sistemaDeCadastro**