

FIC 2 – Módulo III

# Aula 4 – Entendendo o ORM Sequelize.

Persistência de dados com  
Sequelize.  
Antonio Izo Júnior

# Sumário

- O que é Sequelize e ORM.
- Instalando o módulo sequelize no Repl.it.
- Testando o módulo sequelize.

# Sequelize

- O Sequelize é um **ORM** para Node.js.
- ORM é uma técnica de **mapeamento objeto relacional** que permite fazer uma relação dos objetos com os dados que eles representam.
- Quando estamos construindo aplicações com bancos de dados, independente da linguagem, existem muitas atividades repetitivas mesmo entre sistemas completamente diferentes.



# Sequelize

- Uma delas é a escrita dos comandos e consultas SQL para realizar operações nas tabelas do seu banco e a outra é o mapeamento das entidades e relacionamentos em objetos ou módulos da sua aplicação.
- Mapear tabelas para código é um padrão muito comum, independente de linguagem ou framework, pois permite programar mais próximo da regra de negócio da empresa.

# Sequelize

- Uma vez que o mapeamento é feito, atividades triviais, porém trabalhosas, como ficar escrevendo os mesmos comandos SQLs de sempre se tornam apenas simples chamadas de **funções ou métodos**.
- Por exemplo, se você tem uma tabela de alunos, você pode ter uma classe ou módulo alunos na sua aplicação e, em vez de escrever um INSERT para salvar um novo aluno na base, você usaria uma função ou método salvar/adicionar



# Sequelize

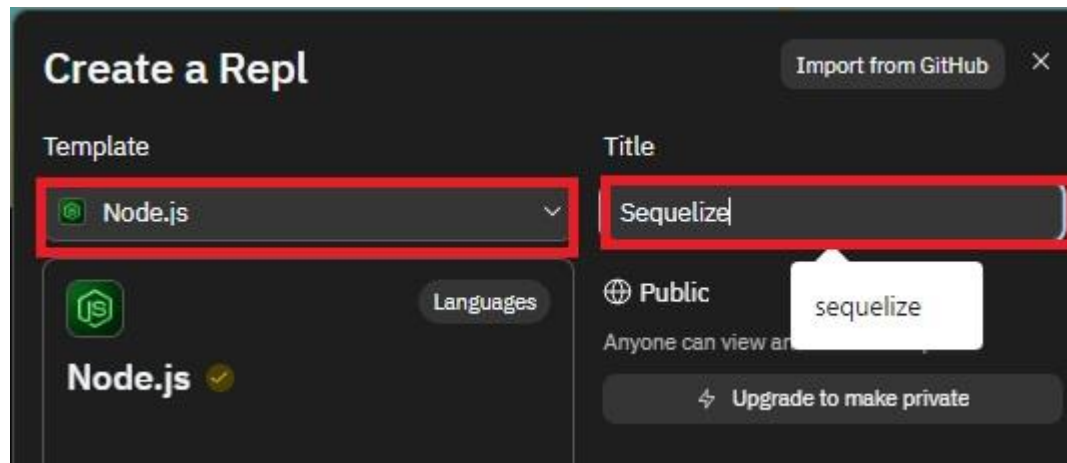
- Assim, você estará manipulando o banco SQL sem precisar escrever SQL de fato, usando apenas a sua linguagem de programação favorita.
- Um ORM lhe permite fazer este mapeamento de forma automática.

# Vamos praticar?



# Instalação do módulo sequelize no Repl.it.

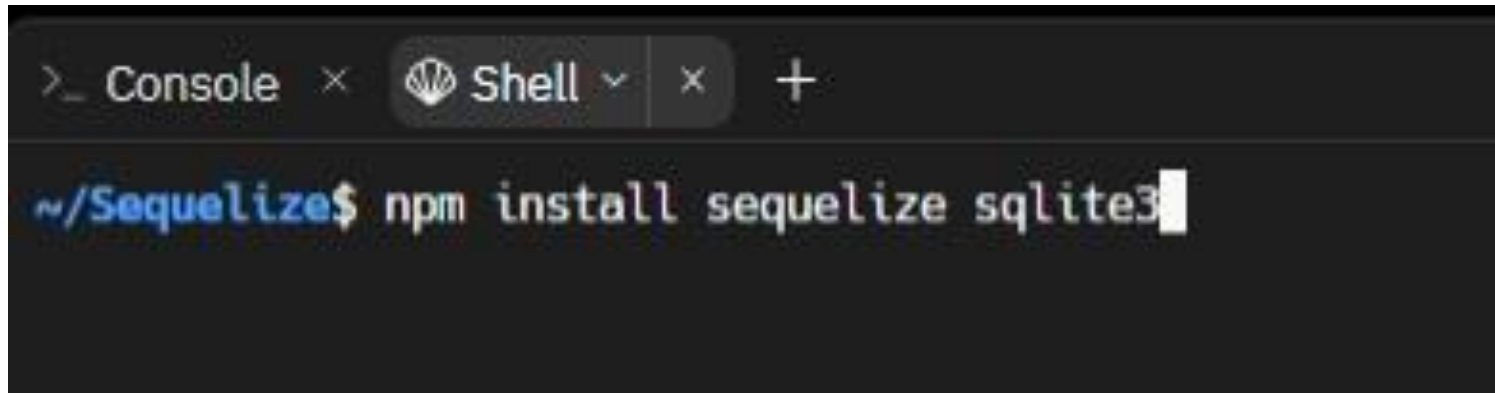
- **Passo1:** Abra um novo arquivo no repl.it com o tipo Node.js e com o nome de sequelize.





# Instalação do módulo sequelize no Repl.it.

- **Passo2:** Na aba "Shell", instale os módulos sequelize e sqlite3 conforme a figura abaixo.



The image shows a terminal window with a dark background. At the top, there are two tabs: 'Console' and 'Shell'. The 'Shell' tab is active, indicated by a shell icon and a dropdown arrow. Below the tabs, the terminal prompt is '~ /Sequelize\$'. The command 'npm install sequelize sqlite3' is entered and highlighted in blue. A white cursor is at the end of the command.

Npm install sequelize sqlite3

# Instalação do módulo sequelize no Repl.it.

- **Passo3:** Verifique seu arquivo package.Json. Se a instalação foi bem sucedida, os dois pacotes instalados aparecerão.

```
1 {  
2   "name": "nodejs",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "index.js",  
6   "scripts": {  
7     "test": "echo \"Error: no test specified\" && exit 1"  
8   },  
9   "keywords": [],  
10  "author": "",  
11  "license": "ISC",  
12  "dependencies": {  
13    "@types/node": "^18.0.6",  
14    "node-fetch": "^3.2.6",  
15    "sequelize": "^6.31.0",  
16    "sqlite3": "^5.1.6"  
17  }  
18 }
```

# Instalação do módulo sequelize no Repl.it.

- Nosso projeto já está preparado para utilizar o sequelize.
- Agora iremos testar a criação do banco empresa pelo pacote.

# Criação do banco empresa pelo sequelize

- **Passo1:** No arquivo index.js, vamos declarar as bibliotecas que utilizaremos no projeto.

```
const { Sequelize, Model, DataTypes } =  
require("sequelize");
```

```
1 // Importando as bibliotecas que iremos utilizar  
2 const { Sequelize, Model, DataTypes } = require("sequelize");
```

# Criação do banco empresa pelo sequelize

- **Passo2:** Agora iremos abrir a conexão com o banco de dados. Caso o banco não exista, ele será criado.

```
//abrindo uma conexão
const sequelize = new Sequelize({
  dialect: "sqlite",
  storage: "empresa.sqlite"
});
```

- Criamos uma variável chamada **sequelize**, após definimos o tipo de banco de dados no **dialect**, no caso **sqlite** e, por fim, colocamos como será o armazenamento no **storage**.

# Criação do banco empresa pelo sequelize

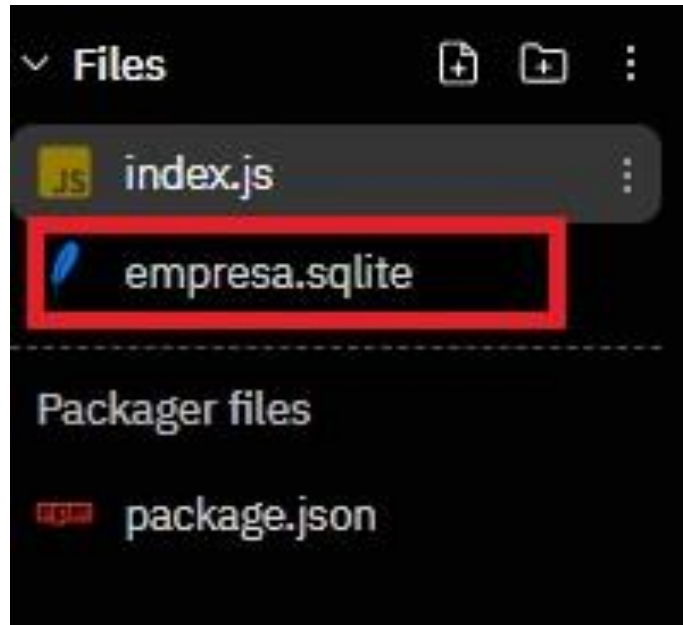
- **Passo3:** Para finalizarmos e conseguirmos criar o banco ao executar, devemos incluir o código abaixo.

```
(async () => {  
  // Sincronizando automaticamente  
  await sequelize.sync({ force: true });  
})();
```

- O código acima fará uma sincronização do nosso modelo que, por enquanto, possui só a criação do banco de dados.

# Criação do banco empresa pelo sequelize

- **Passo4:** Repare que nosso banco de dados, chamado de empresa.sqlite, foi criado.





# Criação do banco empresa pelo sequelize

- Criamos nosso primeiro banco de dados via ORM sequelize.
- Na próxima semana, faremos a manipulação das tabelas no nosso banco de dados.

*Até lá...*

# Referências Bibliográficas

- ZHAO, Alice. **SQL Guia Prático: Um guia para o uso de SQL**. Editora Novatec, 2023.
- NEWMAN, Chris. **SQLite**. 1ª. Editora Sams, 2004.



**INSTITUTO  
FEDERAL**  
Espírito Santo