31-ORM\_com\_NodeJS\_API\_com\_Sequelize\_e\_MySQL

* **Programas necessários:**

**MySQL: instalação full: https:**

**Tutorial:** [www.alura.com.br/artigos/mysql-do-download-e-instalacao-ate-sua-primeira-tabela](http://www.alura.com.br/artigos/mysql-do-download-e-instalacao-ate-sua-primeira-tabela)

**Sera usado no curso o MySQL Command Line Client**

**Download:** <https://dev.mysql.com/downloads/workbench/>

**NodeJS: versão 10 ou superior:** <https://nodejs.org/en/download>

**Postman:** ambiente onde faremos os testes da nossa API: https://www.postman.com/downloads/

* **Instalação de dependências:**

npm init -y = Iniciar o projeto:

npm install express

npm install body-parser = converte arquivos externos(POST) p/ json.

Npm install –save-dev nodemon = atualiza o servidor do **Postman** automaticamente

Npm install mysql2

npm install sequelize@5.21.7 [sequelize-cli@5.5.1](mailto:sequelize-cli@5.5.1) = sequelize-cli é para usar linha de comando na comunicação com o banco de dados

  "dependencies": {

    "body-parser": "^1.20.2",

    "express": "^4.18.2",

    "mysql2": "^3.3.1",

    "path": "^0.12.7",

    "sequelize": "^6.31.1",

    "sequelize-cli": "^6.6.0"

  }

* Comandos no decorrer do projeto

npx sequelize-cli init = inicia o sequelize-cli e cria algumas pastas(config,models,migratórios,seeders)

Instalação do banco de dados usando o mysql no cmd

Obs: depois de instalar o mysql, precisa configurar o local de instalação do mysql da pasta bin que fica em:

Propriedades do sistema/variável de ambientes/variável do sistema/Path. Dentro do Path precisa adicionar o caminho da bin do mysql. No meu caso foi: C:\Program Files\MySQL\MySQL Server 8.0\bin

Depois basta da o comando de inicio: mysql -u root -p e a senha que coloquei root. Exemplo está abaixo:

Microsoft Windows [versão 10.0.22000.1936]

(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Charles Antunes>mysql

ERROR 1045 (28000): Access denied for user 'ODBC'@'localhost' (using password: NO)

C:\Users\Charles Antunes>mysql --version

mysql Ver 8.0.33 for Win64 on x86\_64 (MySQL Community Server - GPL)

C:\Users\Charles Antunes>mysql -u root -p

Enter password: \*\*\*\*

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 19

Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

* Criando uma tabela com suas propriedades direto do sequelize(vai para seeders)

npx sequelize-cli model:create --name Pessoas --attributes nome:string,ativo:boolean,email:string,role:string

* Inicializando todos os sistemas ao ligar o PC.

npm run start = inicializa com o nodemon para atualizar o sistema sempre que houver alterações

mysql -u root -p = = iniciando a comunicação com o banco de dados, pelo prompt

* Migração

São alterações que realizamos nas tabelas do banco de dados(depois de fazer todas as conf, migrate manda p bd)

npx sequelize-cli db:migrate

* comandos SQL

|  |  |
| --- | --- |
| show databases; | consulta todos os bancos existentes |
| select user from mysql.user; | mostra todos os usuários do banco |
| use escola\_ingles; | entra no banco de dados selecionado |
| show tables; | exibe todas as tabelas do banco de dados selecionado |
| describe Pessoas; | exibe as informações da tabela, como as propriedades |
| select \* from Pessoas; | Exibe todas os dados adicionados na tabela Pessoas |

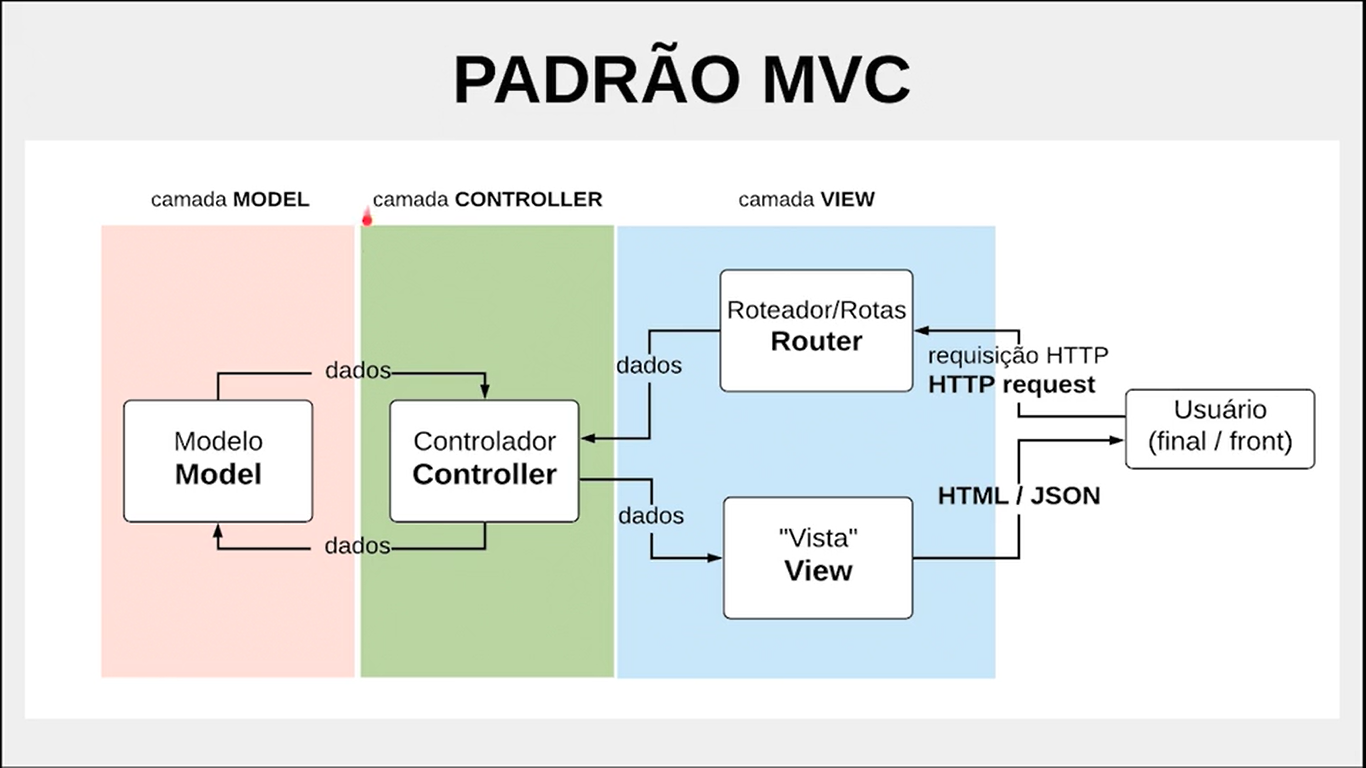
* Comandos usados mais usado do sequelize-cli, uns substituem os comandos do SQL

|  |  |
| --- | --- |
| npx sequelize-cli seed:generate --name demo-pessoas | Cria na pasta seeders com dado de exemplo e saber como os dados serão inseridos na tabela. |
| npx sequelize-cli db:seed:all | Na pasta seeders, serão adicionados alguns dados para tabela pessoas e executar o comando ao lado. |
| npx sequelize-cli db:migrate:undo | Desfazer somente a última migração feita, na ordem em que os arquivos são lidos, pode repetir. |
| npx sequelize-cli db:seed:undo | Desfazer o último seed feito. |
| npx sequelize-cli db:seed:undo --seed nomeDoArquivo | Desfazer seeds de uma tabela específica. |
| npx sequelize-cli db:seed:undo:all | Para desfazer todos os seeds feitos. |

* **Comandos CLI:**

| **Comandos** | **Descrição** |
| --- | --- |
| sequelize db:migrate | executa todas as migrações pendentes para atualizar o banco de dados. |
| sequelize db:migrate:schema:timestamps:add | Atualiza uma tabela de migração para ter marcação de data/hora |
| sequelize db:migrate:status | Exibe o status de todas as migrações |
| sequelize db:migrate:undo | reverte a migração mais recente do banco de dados. |
| sequelize db:migrate:undo:all | Reverte todas as migrações |
| sequelize db:seed | Inicializa um seeder específico |
| sequelize db:seed:undo | Deleta arquivo do banco de dados |
| sequelize db:seed:all | Inicializa todos os seeders |
| sequelize db:seed:undo:all | Deleta todos os dados do banco de dados |
| sequelize db:create | Cria um banco com uma configuração específica |
| sequelize db:drop | exclui o banco de dados especificado na configuração. |
| sequelize init | Inicializa um projeto |
| sequelize init:config | Inicializa as configurações |
| sequelize init:migrations | Inicializa as migrações |
| sequelize init:models | Inicializa as models |
| sequelize init:seeders | Inicializa os seeders |
| sequelize migration:generate [alias: migration:create] | Gera um novo arquivo de migração |
| sequelize model:generate [alias: model:create] | Gera uma model e sua migração |
| sequelize seed:generate | Gera um novo arquivo de seed |

====================================================================================================



Model as tabelas inseridas no banco de dados

Controller é onde ficam as regras de negócio, onde estamos codificando para comunicar o banco com o usuário.

View é os dados recebido do bando e entregue ao front-end para tratar, via html ou json.

Populando as tabelas com dados já feitos para o curso, primeiro populando os seeders

npx sequelize-cli seed:generate --name demo-niveis

npx sequelize-cli seed:generate --name demo-turmas

npx sequelize-cli seed:generate --name demo-matriculas

* agora só copiar os dados com informações criados para o curso e colar nos seeders, depois rodar o comando para mandar para o banco de dados.

npx sequelize-cli db:seed:all

* Agora precisa criar os controladores(crud) do sistema que esta na pasta controllers.
* Depois dos controllers, criar as rotas que está em routes.
* Por último, precisamos avisar o routes/index.js que novas rotas foram adicionadas à aplicação.
* Comandos CRUD.

**Consultar** = consultar pessoa com id 7, poderia também incluir a matricula, http://localhost:3000/pessoas/7/matricula8

GET http://localhost:3000/pessoas/7

Resposta:

{

    "id": 7,

    "nome": "Charles Antunes",

    "ativo": true,

    "email": "charleswcatunes@gmail.com",

    "role": "estudante",

    "createdAt": "2023-05-25T06:09:52.000Z",

    "updatedAt": "2023-05-25T06:09:52.000Z"

}

**Adicionar** = adicionado a pessoa de id\_2 na com status de “confirmado” na turma 4

Postman:

POST http://localhost:3000/pessoas/2/matricula/

{

    "status": "confirmado",

    "turma\_id": 4

}

Resposta:

{

    "id": 9,

    "status": "confirmado",

    "turma\_id": 4,

    "estudante\_id": 2,

    "updatedAt": "2023-05-27T06:45:45.125Z",

    "createdAt": "2023-05-27T06:45:45.125Z"

}

**Atualizar** = Atualizar a pessoa de id\_2 e matricula 9, alterar para status “cancelado”

Postman:

PUT http://localhost:3000/pessoas/2/matricula/9

{

    "status": "cancelado"

}

Resposta:

{

    "id": 9,

    "status": "confirmado",

    "turma\_id": 4,

    "estudante\_id": 2,

    "updatedAt": "2023-05-27T06:45:45.125Z",

    "createdAt": "2023-05-27T06:45:45.125Z"

}

DELETE http://localhost:3000/pessoas/2/matricula/9

Resposta:

{

    "mensagem": "id 9 deletado"

}