

1. Um modelo é uma classe ES6. Uma instância da classe representa um objeto desse modelo (que mapeia para uma linha da tabela no base de dados). Dessa forma, as instâncias do modelo são DAOs (Data Access Objects).
2. Definimos na aula o modelo Produto com Sequelize. O objectivo agora é usar este modelo nos diferentes middlewares definidos nos controllers admin.js e shop.js. Para isso devem então fazer uso dos métodos Sequelize para criar uma instância (que corresponde a inserir nova linha na tabela), procurar e obter dados, modificar e apagar.
3. A documentação dos métodos estão aqui <https://sequelize.org/docs/v6/core-concepts/model-instances/>. O TPC é usarem os métodos do Sequelize para models de acordo com o que é descrito na sequência.
4. O primeiro método que vão usar é o create. Este método será chamado quando recebermos os dados do form no /admin/add-product. Vão portanto modificar a função postAddProduct no admin.js e>
 - ✓ em vez de: `const product = new Product(null, title, imageUrl, description, price); product.save();`
 - ✓ devem chamar o método create com o Modelo Product. Assm:

```
Product.create({
  title: title,
  price: price,
  imageUrl: imageUrl,
  description: description,
})
```
 - ✓ Lembrem-se que os métodos de Sequelize vão retornar um Promise. Então a seguir ao create devem ter os dois ramos do programa .then() e .catch() que devem completar.

2 – No shop.js temos o middleware getIndex. Este é o código da versão anterior :

```
exports.getIndex = (req, res, next) => {
  Product.fetchAll(products => {
    res.render('shop/index', {
      prods: products,
      pageTitle: 'Shop',
      path: '/'
    });
  });
}
```

Objectivo é usar o método findAll do sequelize - `Product.findAll()` . Relembrar que findAll também retorna um Promise. Portanto devemos ter o then e catch.

3 – Devem modificar todos os restantes middlewares dos controllers shop.js e admin.js para que se possa usar métodos Sequelize. Consultem o link referenciado anteriormente para mais detalhes sobre os métodos e respectivos sintaxe.

4 – De seguida vão praticar a criação de Modelos. Definem um novo modelo para representar um utilizador da nossa aplicação. O modelo deve ter os campos id (inteiro, auto incremento e chave primária) nome e e-mail.

[dica para que o modelo posso fazer sync e criar a tabela devem importar os modelos no app.js]