

AVALIAÇÃO - UNIDADE II

TAD0203 - GESTÃO DE QUALIDADE DE SOFTWARE - Docente: Joel Santos

Objetivo: Trabalhar com testes de integração com o banco de dados. O foco principal será nas operações CRUD (Create, Read, Update, Delete) para gerenciar categorias, produtos e estoque.

Estrutura do Projeto

A aplicação Node.js será composta pelos seguintes módulos:

1. **Categoria:** Gerenciamento de categorias de produtos.
2. **Produto:** Gerenciamento de produtos.
3. **Estoque:** Gerenciamento do estoque dos produtos.

Passos para Iniciar o Projeto

1. **Download do Código Base:**
 - Baixe o arquivo `avaliacao.zip` contendo o código base do projeto.
 - Descompacte o arquivo em seu ambiente de desenvolvimento.
2. **Instalação das Dependências:**
 - Navegue até o diretório descompactado do projeto.
 - Execute o comando `npm install` para instalar todas as dependências necessárias.
3. **Estrutura dos Arquivos:**
 - **app.js:** Arquivo principal da aplicação.
 - **models/:** Contém os modelos Sequelize (`Categoria`, `Produto`, `Estoque`).
 - **controllers/:** Contém os controladores para gerenciar as operações de CRUD.
 - **routes/:** Contém os arquivos de rotas para as diferentes entidades (`categoriaRoutes.js`, `produtoRoutes.js`, `estoqueRoutes.js`).
 - **test/:** Contém os arquivos de teste de integração.

Testes que Devem Ser Feitos

Categoria

1. **Criar Categoria**
 - Verificar se uma nova categoria pode ser criada com sucesso.
 - Validar que o retorno tenha um status 201 e os dados da categoria criados estejam corretos.
2. **Listar Categorias**
 - Verificar se todas as categorias podem ser listadas.
 - Validar que o retorno tenha um status 200 e a lista contenha as categorias previamente criadas.
3. **Atualizar Categoria**
 - Verificar se uma categoria específica pode ser atualizada com sucesso.
 - Validar que o retorno tenha um status 200 e os dados da categoria atualizados estejam corretos.
4. **Remover Categoria**
 - Verificar se uma categoria específica pode ser removida com sucesso.

- Validar que o retorno tenha um status 204 e a categoria não exista mais na listagem.

Produto

1. Criar Produto

- Verificar se um novo produto pode ser criado com sucesso.
- Validar que o retorno tenha um status 201 e os dados do produto criados estejam corretos.

2. Listar Produtos

- Verificar se todos os produtos podem ser listados.
- Validar que o retorno tenha um status 200 e a lista contenha os produtos previamente criados.

3. Atualizar Produto

- Verificar se um produto específico pode ser atualizado com sucesso.
- Validar que o retorno tenha um status 200 e os dados do produto atualizados estejam corretos.

4. Remover Produto

- Verificar se um produto específico pode ser removido com sucesso.
- Validar que o retorno tenha um status 204 e o produto não exista mais na listagem.

Estoque

1. Criar Registro de Estoque

- Verificar se um novo registro de estoque pode ser criado com sucesso.
- Validar que o retorno tenha um status 201 e os dados do estoque criado estejam corretos.

2. Listar Estoque

- Verificar se todos os registros de estoque podem ser listados.
- Validar que o retorno tenha um status 200 e a lista contenha os registros de estoque previamente criados.

3. Atualizar Registro de Estoque

- Verificar se um registro de estoque específico pode ser atualizado com sucesso.
- Validar que o retorno tenha um status 200 e os dados do estoque atualizados estejam corretos.

4. Remover Registro de Estoque

- Verificar se um registro de estoque específico pode ser removido com sucesso.
- Validar que o retorno tenha um status 204 e o registro de estoque não exista mais na listagem.

Submeter o link do github com as respostas.