

Laboratório 5 – MongoDB e Mongoose

1 Exercícios

Acesse o código-fonte disponibilizado no arquivo “TypeScript_Lab05.zip”. Execute o comando “npm install” dentro do diretório do projeto.

1. O projeto da solução é um modelo de implementação de acordo com o padrão de separação de responsabilidades em camadas. As entidades de negócio da aplicação (neste caso um modelo simplificado de uma Biblioteca) foram representados por interfaces presentes no diretório “entidades”. A camada de persistência da aplicação, presente no diretório “persistência”, segue o padrão de projeto Repositório (também conhecido como DAO), e encapsula através de uma classe repositório todo o modelo de persistência do mapeador Mongoose. Procure entender como o TypeScript foi utilizado para representar os objetos persistentes do Mongoose através das interfaces de negócio que representam as entidades a serem persistidas no MongoDB.

2. Execute o arquivo “index.ts” que inicializa a base de dados com alguns autores. Você poderá se basear nesse arquivo para criar o resto do projeto.

3. Abra o arquivo “autorRepositorio.ts” e complete a implementação da classe com métodos para:

- consultar o banco de dados e retornar um array com todos os autores cujo último nome seja igual ao parâmetro informado;
- consultar o banco de dados e retornar um array com todos os autores cujo primeiro nome seja igual ao parâmetro informado;
- alterar o registro de um autor no banco de dados;

4. Complete a implementação da camada de persistência com operações CRUD sobre a coleção de livros e empréstimos. Defina os seguintes métodos:

- Na classe responsável pelo repositório de livros:
 - a) criar e armazenar um novo livro no banco de dados, retornando o livro recém criado;
 - b) consultar o banco de dados e retornar um array com todos os livros;
 - c) consultar o banco de dados e retornar um array com todos os livros de um determinado autor cujo id foi passado por parâmetro.
- Na classe responsável pelo repositório de empréstimo:
 - a) criar e armazenar um novo empréstimo no banco de dados, retornando o empréstimo criado;
 - b) consultar o banco de dados e retornar um array com todos os empréstimos;
 - c) alterar os dados de um determinado empréstimo no banco de dados.

5. Crie uma biblioteca de funções de negócio (em um novo arquivo “negocio.js”) para manipular as operações sobre uma biblioteca. Defina as seguintes funções:

- consultarLivros() – para retornar um array com todos os livros da biblioteca, junto com a informação de que estão disponíveis ou não e, se estiverem indisponíveis, a data de entrega futura;
- emprestarLivro(id_livro) – para criar um novo empréstimo caso o livro esteja disponível e marcar como data futura de entrega 7 dias a partir da data de empréstimo;
- devolverLivro(id_livro) – para efetuar a devolução de um livro e calcular o valor da multa associada caso a entrega do livro tenha sido realizada em uma data posterior à data prevista de entrega (defina um valor qualquer de multa por dia de atraso).