

Bootcamp IGTI: Desenvolvedor(a) Node.js**Desafio**

Módulo 04	Tópicos Especiais Desenvolvimento Back End
------------------	---

Objetivos

Exercitar os seguintes conceitos trabalhados no Módulo:

- ✓ Reforçar os conceitos teóricos sobre o TDD.
- ✓ Reforçar os conceitos teóricos sobre a Pirâmide de Testes.
- ✓ Testes de integração utilizando as bibliotecas Jest e supertest.
- ✓ Uso das ferramentas Git/GitHub.
- ✓ Containerização de aplicação utilizando o Docker.

Enunciado

Com base no **exemplo 04 (04-consulta-credito-docker)** (disponibilizado em https://github.com/lgoulartsilva/BTC_NJS_TA_PT1), deverá ser implementada uma Web API REST contendo as operações de CRUD (**C**reate, **R**etrieve, **U**ppdate e **D**eleete) de um Produto. A entidade produto é composta pelas seguintes informações, sendo todas obrigatórias: Código (Texto), Descrição (Texto) e Preço (Número Positivo).

Devem ser criadas as seguintes rotas:

- POST: realiza o cadastro de um novo produto e retorna o HTTP 201. Caso já exista um produto com o mesmo código, os dados de descrição e preço deverão ser atualizados e retornar HTTP 200. Caso o payload da requisição seja inválido, um HTTP 400 deverá ser retornado, juntamente com uma mensagem informando o(s) erro(s) observado(s).

- PUT: os dados de descrição e preço deverão ser atualizados e retornar HTTP 200. Caso o payload da requisição seja inválido, um HTTP 400 deverá ser retornado, juntamente com uma mensagem informando o(s) erro(s) observado(s). Caso não seja encontrado um produto com o mesmo código, um HTTP 405 deverá ser retornado, juntamente com uma mensagem informando o(s) erro(s) observado(s).
- GET: retorna uma lista com todos os registros de produtos encontrados em uma resposta com status HTTP 200. ATENÇÃO: não é necessário realizar a leitura para um único registro.
- DELETE: deverá ser excluído o registro do produto informado, retornando os seus dados em uma resposta com status HTTP 200. Caso não seja encontrado um produto com o mesmo código, um HTTP 405 deverá ser retornado juntamente com uma mensagem informando o(s) erro(s) observado(s).

Atividades

Os alunos deverão desempenhar as seguintes atividades:

1. Realizar a implementação da API Rest e adicioná-la à um contêiner baseado na imagem https://hub.docker.com/_/node.
2. A persistência dos dados deverá ser realizada utilizando uma base de dados postgresql. Nesse caso poderá se utilizada a instalação nativa da base, ou ainda um contêiner baseado na imagem https://hub.docker.com/_/postgres.
3. Deverão ser implementados os testes de integração com supertest + jest (não é necessária a implementação de testes unitários).
4. Deverá ser criado um repositório no **github** e feito o **push** da solução.

Com base na solução realizada, responda as questões.