

# TRABALHO TESTES DE SOFTWARE

## Enunciado do Projeto: API de Cadastro de Clientes (CRUD)

### Visão Geral

O objetivo deste projeto é desenvolver uma **API RESTful** completa para gerenciar o cadastro de clientes. O projeto deve ser construído utilizando **Node.js** e o framework **Express.js**, com ênfase na implementação do padrão CRUD (Create, Read, Update, Delete). Uma parte fundamental do projeto é a **cobertura de testes unitários** para garantir a qualidade e a confiabilidade do código.

### Requisitos da API

A API deve ser capaz de realizar as seguintes operações em um recurso de cliente:

- **POST /clientes**: Cria um novo cliente. O corpo da requisição deve incluir, no mínimo, **nome**, **email** e **telefone**.
- **GET /clientes**: Retorna a lista de todos os clientes cadastrados.
- **GET /clientes/:id**: Retorna os detalhes de um cliente específico, utilizando o **id** como parâmetro de busca.
- **PUT /clientes/:id**: Atualiza as informações de um cliente existente. O corpo da requisição deve conter os dados a serem atualizados.
- **DELETE /clientes/:id**: Remove um cliente do cadastro.

### Tecnologias e Ferramentas Obrigatórias

- **Node.js**: Ambiente de execução do servidor.
- **MySQL**: Sistema Gerenciador de Banco de dados
- **Express.js**: Framework para a criação da API.
- **Jest**: Framework de testes para Node.js.
- **Insomnia ou Postman**: Para testar a API localmente.

### Critérios de Avaliação

1. **Funcionalidade (40%):** A API deve implementar corretamente todas as operações CRUD (**POST**, **GET**, **PUT**, **DELETE**).
2. **Testes Unitários (60%):**
  - Deve haver testes para, no mínimo, **75%** das funcionalidades da API.
  - Utilize o **Jest** para escrever os testes.
  - Os testes devem cobrir os casos de sucesso (ex: cliente criado com sucesso) e os casos de falha (ex: **id** não encontrado, dados inválidos na requisição).

### Passos para o Desenvolvimento

1. **Configuração Inicial:** Crie a estrutura de pastas, inicialize o projeto com **npm init** e instale as dependências (**express**, **jest**).
2. **Estrutura de Dados:** Defina como os dados dos clientes serão armazenados. Para este projeto, pode-se usar um array de objetos em memória para simplificar, sem a necessidade de um banco de dados.
3. **Desenvolvimento da API:** Implemente as rotas e a lógica para cada uma das operações CRUD.
4. **Desenvolvimento dos Testes:** Escreva os testes unitários em paralelo com a implementação da API, usando o Jest.
5. **Documentação:** Escreva um arquivo **README.md** explicando como rodar o projeto e os testes, e listando as rotas disponíveis.

### Entrega do Projeto

O projeto final deve ser enviado em um repositório no **GitHub**. O link do repositório deve ser enviado dentro do prazo estipulado. O repositório deve conter todo o código-fonte, incluindo o arquivo **package.json** para que a equipe de avaliação possa reproduzir o ambiente de desenvolvimento.