

## **Relatório: Defesa da Arquitetura Escolhida API REST**

### **1. Escolha da Tecnologia**

Optei por trabalhar com o backend em Node.js, junto com o framework Express.js, devido a familiaridade com a linguagem, baseada em Javascript. Além disso, foram levados em consideração pontos como performance, agilidade no desenvolvimento, visto que o npm possui diversas ferramentas que auxiliam no processo.

### **2. Persistência de Dados**

Na parte de dados, optei por trabalhar com uma simulação do banco em memória, visto que o objetivo da atividade é demonstrar os conhecimentos relacionados a APIs Restful, implementação de endpoints, compreensão do HATEOAS e tratamento de erros. Além disso, a configuração de um SGBD, como o MongoDB, levaria a uma dependência dos testes a softwares terceiros. Durante a criação da aplicação, foi cogitado o uso do H2, mas preferi trabalhar com a simulação em memória.

### **3. Design da API**

A API foi criada para seguir os princípios REST, através dos recursos e URIs, métodos HTTP correspondentes a cada ação a ser executada no banco (GET para obter dados, POST para inserir novos dados, PUT para atualizações e DELETE para remoção), e inserção dos códigos de status HTTP fornecendo o feedback sobre a ação executada.

### **4. Implementação de HATEOAS**

Ao implementar o HATEOAS, foram obtidos alguns benefícios, como a previsão de próximas ações baseados no status atual do item. Este exemplo pode ser visto no exercício no fluxo de empréstimo e devolução dos livros, onde se ele estiver disponível há uma rota para **emprestar**, e se ele estiver emprestado, há uma rota para **devolver**. Além disso, as rotas são demonstradas em tempo de execução, logo o usuário sabe quais são as possíveis ações para determinado registro.

### **5. Tratamento de Erros**

Os erros foram devidamente padronizados e possuem respostas claras e os códigos de erro estão devidamente aplicados, visando dessa forma uma identificação mais clara do problema.

### **6. Implementação da Rotina Empréstimo/Devolução**

Optei por implementar de forma funcional as rotas de empréstimo e devolução para deixar o exercício mais coeso. A intenção nas rotas foi a mesma: manter a comunicação com os componentes originais e realizar a troca dos status do HATEOAS de acordo com a mudança do status do livro (disponível/emprestado).