

# Teste técnico - Engenheiro de Software NodeJS

Para os requisitos abaixo você deve criar um projeto NodeJS, em Typescript ou Javascript, utilizando NPM como gestor de dependências. A versão do NodeJS utilizada deve ser maior ou igual a v20.

O projeto deve utilizar o framework Express para criar a API. Você poderá utilizar quaisquer outros pacotes disponíveis no repositório do NPM.

O projeto deve rodar e estar funcional em qualquer máquina que tenha NodeJS  $\geq$  v20 instalado, incluindo a camada de persistência.

Para persistência utilize qualquer ferramenta e estratégia que desejar. É necessário que o projeto seja funcional e contenha as dependências necessárias independentemente da estratégia escolhida.

Considere que essa API pode ser expandida no futuro, então pense numa arquitetura que permita escalabilidade.

## Have fun!

1. Seguindo os requisitos acima, crie uma API HTTP para realizar a gestão das solicitações de Notas Fiscais de um cliente. Sua API deve possuir as seguintes operações:

- Criar uma solicitação de Nota Fiscal. A solicitação criada deve ser persistida e permitir consulta posterior. Essa operação deve receber os seguintes parâmetros de entrada:
  - i. CNPJ do tomador do serviço - String;
  - ii. Município da prestação do serviço - String;
  - iii. Estado da prestação do serviço - String;
  - iv. Valor do serviço - Number;
  - v. Data desejada de emissão - Data ISO;
  - vi. Descrição do serviço - String

Além dos parâmetros de entrada, você deve persistir a data de criação da solicitação, data de última atualização da solicitação e o status inicial da solicitação.

Considere que uma nota fiscal pode ter os seguintes status:

- i. PENDENTE\_EMISSAO
- ii. EMITIDA
- iii. CANCELADA

O status inicial de uma solicitação de Nota Fiscal é PENDENTE\_EMISSAO.

- Listar todas as solicitações de Notas Fiscais persistidas. Todos os dados persistidos das solicitações devem ser retornados.
- Buscar uma solicitação.
- Emitir uma Nota Fiscal. Essa operação deve receber o identificador de uma solicitação criada anteriormente e consumir uma API HTTP externa que realizará a emissão.
  - i. Utilize a operação POST <https://api.drfinancas.com/testes/notas-fiscais>. Utilize a chave "87451e7c-48bc-48d1-a038-c16783dd404c" no cabeçalho "Authorization" para autenticar a chamada. Considere os mesmos campos utilizados na criação da solicitação de Nota Fiscal como payload da operação de emissão. Essa API retorna resultados aleatórios variando entre respostas de sucesso (status 200) e erros (status 400, 401 e 500).
  - ii. Caso a emissão seja feita com sucesso (200), a API retornará os campos:
    - 1. numeroNF - string
    - 2. dataEmissao - string (formato ISO)Os campos devem ser salvos na solicitação e o status deve ser atualizado para EMITIDA
  - iii. Considere que a API pode retornar erros. Realize um tratamento básico de erros para os status de retorno 400, 401 e 500.

2. Crie uma suíte de testes para as operações acima.

3. Considere o seguinte cenário: Você tem uma aplicação frontend para seus clientes. Nesta aplicação, você precisa implementar uma funcionalidade em que o usuário realizará alguma ação que levará tempo para ser concluída. A aplicação deve sempre mostrar o último status do processamento. No backend, essa ação do usuário irá disparar uma sequência de ações assíncronas independentes entre si com tempos de conclusão diferentes. Desenhe um diagrama de arquitetura para essa solução. Não é necessário codificar nenhum dos componentes.