



**Desafio de Node 2023.1**

**Juiz de Fora, 2023**

# SUMÁRIO

<b>1.</b>	<b>Introdução.....</b>	<b>3</b>
1.1.	Propósito.....	3
1.2.	Escopo.....	3
1.3.	Definições e abreviações.....	3
1.4.	Visão geral do documento.....	3
<b>2.</b>	<b>Descrição geral.....</b>	<b>4</b>
2.1.	Perspectivas do produto.....	4
2.2.	Funções do produto.....	4
2.3.	Características do usuário.....	4
<b>3.</b>	<b>Requisitos funcionais.....</b>	<b>4</b>
<b>4.</b>	<b>Requisitos não funcionais.....</b>	<b>6</b>
<b>5.</b>	<b>Orientações.....</b>	<b>7</b>

## Especificação dos Requisitos de Software

<b>Projeto:</b> <i>Desafio de Node 2023.1</i>	
<b>Data de criação:</b> 28/01/23	<b>Versão:</b> 1.0.0

Histórico de Modificação			
Data	Versão	Descrição	Autor(es)
28/01/23	1.0.0	Início da elaboração do documento	Gabriel Gomes Bahia

# 1. Introdução:

## 1.1. Propósito:

O propósito do Documento de Especificação de Requisitos é delinear os requisitos do software a ser construído, descrevendo suas funcionalidades e características. O público alvo do documento são clientes, gerentes e desenvolvedores do projeto.

## 1.2. Escopo:

*O projeto consiste em criar uma API que uma aplicação de delivery de produtos irá utilizar. Essa API tem o objetivo de fazer o gerenciamento de produtos, pedidos e restaurantes. Os Restaurantes poderão ser cadastrados e fazer login no sistema por meio de um token JWT. Após o login os restaurantes poderão consultar, cadastrar, editar e excluir seus produtos e seus pedidos com os produtos que ele possui. Um restaurante pode ter vários pedidos e produtos, assim como um pedido pode ter vários produtos.*

## 1.3. Definições e abreviações:

- **RF:** requisito funcional;
- **RNF:** requisito não funcional.

## 1.4. Visão geral do documento:

- **Seção 2 - Descrição Geral:** apresenta uma visão geral do sistema, especificando a perspectiva do produto e detalhamento do escopo do sistema através da discretização das funções do produto. Além disso, são explicitadas as características gerais dos usuários do produto e as restrições que poderão limitar as possibilidades de desenvolvimento.
- **Seção 3 - Requisitos funcionais:** apresentação de todos os requisitos funcionais do sistema. Descreve as principais ações do produto, considerando a aceitação e processamento das entradas e o processamento e geração das saídas.
- **Seção 4 - Requisitos não funcionais:** apresentação de todos os requisitos não funcionais do sistema. Descreve todos os aspectos qualitativos do sistema, explicitando os detalhes de

facilidade de uso, manutenibilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.

## 2. Descrição geral:

### 2.1. Funções do produto:

- Gerenciamento de restaurantes(Listar, Cadastrar, Editar e Deletar)
- Gerenciamento de produtos(Listar, Cadastrar, Editar e Deletar)
- Gerenciamento de pedidos(Listar, Cadastrar, Editar e Deletar)
- Login com token JWT
- Documentação da API com o Swagger
- Modelagem do banco de dados

## 3. Requisitos funcionais:

Requisito 01 / RF001 - <b>CRUD de restaurantes (20%)</b>	
<b>1. Descrição</b>	<p>O sistema deverá ter um gerenciamento de restaurantes, sendo possível criar, visualizar, editar e deletar um restaurante. A rota de visualizar deve ser pública (qualquer pessoa pode ver os restaurantes). As demais rotas podem ser acessadas apenas por usuários que estão logados no sistema. Utilize middlewares para fazer essa verificação.</p> <p>O restaurante é também o usuário que acessa o sistema.</p>
<b>2. Dados</b>	<ul style="list-style-type: none"><li>• Id &lt;Int&gt;</li><li>• Nome &lt;String&gt;</li><li>• Email &lt;String&gt;</li><li>• Senha &lt;String&gt;</li><li>• Categoria &lt;String&gt;</li><li>• Cidade &lt;String&gt;</li><li>• Endereco &lt;String&gt;</li><li>• Telefone &lt;String&gt;</li></ul>

### Requisito 02 / RF002 - *CRUD de produtos* (20%)

<b>1. Descrição</b>	O sistema deverá ter um gerenciamento de produtos, sendo possível criar, visualizar, editar e deletar um produto. As rotas de consulta dos produtos devem ser públicas para que os clientes consigam ver os produtos de um restaurante. As demais rotas podem ser acessadas apenas por usuários que estão logados no sistema.
<b>2. Dados</b>	<ul style="list-style-type: none"><li>• id &lt;Int&gt;</li><li>• nome &lt;String&gt;</li><li>• descricao &lt;String&gt;</li><li>• quantidade &lt;String&gt;</li><li>• preco &lt;String&gt;</li><li>• categoria &lt;String&gt;</li><li>• id_restaurante &lt;Int&gt;</li></ul>

### Requisito 03 / RF003 - *CRUD de pedidos* (20%)

<b>1. Descrição</b>	O sistema deverá ter um gerenciamento de pedidos, sendo possível criar, visualizar, editar e deletar um pedido. Todas as rotas devem ser acessadas apenas por usuários que estão logados no sistema.
<b>2. Dados</b>	<ul style="list-style-type: none"><li>• id &lt;Int&gt;</li><li>• produtos &lt;Array&gt;</li><li>• valor_total &lt;float&gt;</li><li>• nome_cliente &lt;String&gt;</li><li>• cidade_cliente &lt;String&gt;</li><li>• endereco_cliente &lt;String&gt;</li><li>• telefone_cliente &lt;String&gt;</li><li>• restaurante_id &lt;int&gt;</li></ul>

### Requisito 04 / RF004 - *Login* (20%)

<b>1. Descrição</b>	O sistema deve ter Login de restaurantes no sistema, utilizando token JWT como forma de realizar o login.
<b>2. Dados</b>	<ul style="list-style-type: none"><li>• email &lt;String&gt;</li><li>• senha &lt;String&gt;</li></ul>

#### Requisito 05 / RF005 - Documentação da API (Swagger) (20%)

1. Descrição	O sistema deverá ter a documentação detalhada de todos os endpoints da API utilizando o Swagger.
2. Dados	<ul style="list-style-type: none"><li>• Todas as rotas devem ser documentadas</li><li>• As rotas POST, PUT e DELETE devem ter os dados que precisam ser enviados na documentação da API.</li></ul>

#### Requisito 06 / RF006 - Modelagem do banco de dados

1. Descrição	Deverá ser feita a modelagem do banco de dados do sistemas com todos os atributos, entidades e as relações existentes. Utilizem o Modelo Entidade Relacionamento (MER). Utilize o software que achar necessário.
2. Data de Envio	Deve ser enviado até às <b>23:59 do dia 12/02/2023</b>

## 4. Requisitos não funcionais:

#### Requisito não funcional / RNF001 - Segurança

1.Descrição	<i>O sistema deve garantir que as rotas protegidas não possam ser acessadas sem o login</i>
-------------	---

#### Requisito não funcional / RNF002 - Códigos HTTP

1.Descrição	O sistema deve enviar para o front o código HTTP certo para cada ocasião.
-------------	---

#### Requisito não funcional / RNF003 - Mensagens de Erro

1.Descrição	O sistema deve enviar para o front mensagens de erro intuitivas e detalhadas, para serem devidamente tratadas.
-------------	--

## 5. Orientações:

- O relacionamento da aplicação com o banco de dados pode ser feito utilizando TypeORM ou Prisma.
- O banco de dados utilizado fica a critério do desenvolvedor, mas priorize MySQL, PostgreSQL e MongoDB.
- O login é OBRIGATÓRIO que seja feito com JWT.
- O sistema é uma API, então não é necessário fazer um front-end para consumi-lo, utilize o Insomnia ou o Postman para fazer as requisições.
- Caso tenha dificuldades com os códigos HTTP, acesse [HTTP Cats](#) ou [HTTP Status Dogs](#)
- Fica a critério do desenvolvedor a adição de bibliotecas ao sistema.
- Será disponibilizado um template para iniciar o desenvolvimento, mas o desenvolvedor pode optar por fazer quaisquer alterações na estrutura.
- Qualquer dúvida entre em contato com o gerente responsável pelo documento de requisitos ou com o diretor de projetos
- As porcentagens de cada requisito estão no título de cada tabela acima. Os membros do departamento de projetos devem entregar 100% até a data final de entrega.
- Link para repositório base do desafio [Desafio Node](#)
- Trilha de Node: [w Trilha de NodeJS.docx](#)
- Links de apoio para a modelagem do banco de dados:
  - <https://www.youtube.com/watch?v=64ULpoaP8ZA>
  - <https://www.youtube.com/watch?v=wjDhtGhj9xg>
  - <https://www.youtube.com/watch?v=hLiBVGa1YT4> (Esse é o mais curto e mais específico)
  - Se preferir um conteúdo em Texto:
    - <https://www.devmedia.com.br/modelagem-de-dados-tutorial/20398>