

# ESPECIFICAÇÃO E MODELAGEM DOS REQUISITOS

## Requisitos Funcionais (RF)

- **RF01:** Cadastro e login de usuários.
- **RF02:** Pesquisa e visualização de suplementos.
- **RF03:** Exibição detalhada dos produtos (informações e tabela nutricional).
- **RF04:** Adição ao carrinho e finalização de compra.
- **RF05:** Área do cliente com histórico de pedidos.
- **RF06:** Avaliações e comentários nos produtos.
- **RF07:** Painel administrativo para gestão da plataforma.

## Requisitos Não Funcionais (RNF)

- **RNF01:** Interface simples, responsiva e fácil de usar.
- **RNF02:** Segurança dos dados e transações (uso de SSL).
- **RNF03:** Carregamento rápido e bom desempenho.
- **RNF04:** Disponível 24h/dia, compatível com vários dispositivos e navegadores.
- **RNF05:** Estrutura escalável para futuras melhorias.

## JUSTIFICATIVA DA ESCOLHA DO TIPO DE ARQUITETURA

Para o desenvolvimento da plataforma de e-commerce, adotamos uma **arquitetura em três camadas** composta por:

- **Frontend:** HTML, CSS e JavaScript
- **Backend:** Node.js com Express.js
- **Banco de Dados:** MySQL

Essa arquitetura foi escolhida por oferecer **flexibilidade, organização e desempenho**, permitindo a separação entre interface, lógica de negócio e armazenamento de dados. Isso facilita a manutenção, os testes e futuras expansões da plataforma.

### Justificativa referente aos RF e RNF:

- **Frontend com JavaScript** permite interatividade e melhor usabilidade (RNF01), essencial para funções como filtros de produtos, atualizações em tempo real no carrinho (RF03, RF05) e feedbacks visuais para o usuário.
- **Node.js no backend** garante agilidade no processamento de requisições, ideal para sistemas com muitas interações simultâneas (RF04, RNF03).

- **Banco de dados MySQL** assegura consistência e integridade nos dados de usuários, pedidos e produtos (RF01, RF05, RF07).
- **Boa segurança (RNF02)**, com autenticação e tratamento de dados sensíveis no backend, além de boas práticas como uso de HTTPS e hash de senhas.
- **Alta compatibilidade e escalabilidade (RNF04 e RNF05)**, permitindo que o sistema cresça com o aumento da demanda.

Essa arquitetura moderna e robusta atende perfeitamente às necessidades do projeto, unindo performance, segurança e uma experiência fluida para o usuário.

## MODELAGEM COMPLETA DA ARQUITETURA COM MÓDULOS/CONTAINERS

A arquitetura será composta por três camadas principais: **Frontend**, **Backend** e **Banco de Dados**, com cada camada dividida em módulos e containers específicos para garantir uma boa organização e separação de responsabilidades.

### 7.1 Camada de Frontend

- **Módulos:**
  - **Interface de Usuário (UI):** Contém os componentes visuais da plataforma, como formulários de cadastro/login, telas de listagem de produtos, carrinho de compras, e detalhes do produto.
  - **Interatividade e Dinâmica (JavaScript):** Responsável pela manipulação dinâmica da interface, como atualizações em tempo real do carrinho de compras, filtros de pesquisa de produtos, e interações com o backend via APIs.
  - **Validação e Feedback:** Realiza validações no lado do cliente (ex.: verificação de campos obrigatórios em formulários) e fornece feedback visual ao usuário.
  - **Responsividade:** Garante que o layout seja adaptável a diferentes dispositivos, utilizando CSS e frameworks como **Bootstrap** ou **Flexbox**.
- **Containers/Componentes:**
  - **Componentes HTML/CSS:** Arquivos estáticos contendo as estruturas e estilos da interface do usuário.
  - **Componentes JavaScript:** Responsáveis pela interação com o backend, animações e lógica do frontend (AJAX para requisições assíncronas).

## 7.2 Camada de Backend

- **Módulos:**
  - **Controle de Usuários:** Gestão de autenticação, cadastro, login e gerenciamento de perfis de usuário (RF01, RF02, RF05).
  - **Gestão de Produtos:** Módulo responsável por gerenciar as informações dos produtos, incluindo a inserção, edição e exclusão no banco de dados (RF03).
  - **Carrinho e Pedidos:** Gerenciamento do carrinho de compras, cálculo de valores, e finalização de pedidos (RF04, RF06).
  - **Sistema de Avaliações:** Gestão das avaliações e comentários dos produtos feitos pelos usuários (RF08).
  - **Administração:** Painel para administradores gerenciarem usuários, produtos, pedidos e visualizações de métricas (RF09).
  - **Segurança e Autenticação:** Middleware de segurança para autenticação de usuários (ex: JWT, cookies de sessão), garantindo que somente usuários autorizados acessem áreas restritas.
  - **APIs (RESTful):** Interface entre o frontend e o backend. O frontend se comunica com o backend por meio de chamadas API para buscar informações sobre produtos, criar pedidos, etc.
- **Containers/Componentes:**
  - **Node.js com Express.js:** Servidor backend que gerencia as requisições, validações e interações com o banco de dados.
  - **Middleware de Segurança:** Camada responsável pela segurança, como autenticação (JWT) e verificação de permissões.
  - **Controladores de Requisição:** Funções que gerenciam as requisições recebidas do frontend e as respostas correspondentes (ex: resposta de sucesso ou erro ao realizar login, ou ao buscar produtos).

## 7.3 Camada de Banco de Dados

- **Módulos:**
  - **Gestão de Usuários:** Tabelas de usuários, com informações como nome, e-mail, senha criptografada e dados de histórico de pedidos.
  - **Gestão de Produtos:** Tabelas que armazenam os detalhes dos produtos, como nome, descrição, preço, tabela nutricional, e quantidade em estoque.
  - **Pedidos:** Tabelas para registrar os pedidos feitos pelos usuários, incluindo dados de pagamento, endereço de entrega e status do pedido.

- **Avaliações:** Tabelas para armazenar as avaliações dos produtos feitas pelos usuários.
- **Administração:** Dados relacionados aos administradores da plataforma, com permissões de acesso e controle.
- **Containers/Componentes:**
  - **MySQL:** Banco de dados relacional utilizado para armazenar as informações de usuários, produtos, pedidos, avaliações e administração. O uso de **relacionamentos** entre tabelas (ex.: usuários e pedidos) ajuda a garantir a integridade dos dados.

## 7.4 Fluxo de Comunicação Entre os Módulos

1. **Frontend** (HTML, CSS, JavaScript) interage com o **Backend** através de chamadas **AJAX** para as APIs RESTful.
2. O **Backend** (Node.js com Express.js) processa as requisições e interage com o **Banco de Dados MySQL** para buscar ou modificar dados (ex.: consulta de produtos, criação de pedidos).
3. O **Banco de Dados MySQL** armazena e recupera informações essenciais, como detalhes de produtos, pedidos, usuários e avaliações.
4. O **Backend** retorna os dados processados para o **Frontend**, que os exibe ao usuário de forma interativa e dinâmica.