Especificação dos Requisitos de Software

Para Sistema de Gerenciamento de Produtos Vendidos - Nunes Sports

Versão 1.0

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 19/08/2024 | 1.0 | Documento inicial | Ronan B. F. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice Analítico

1. Introdução 4

1.1 Finalidade 4

1.2 Escopo 4

1.3 Definições, Acrônimos e Abreviações 4

1.4 Referências 4

2. Descrição Geral 5

2.1 Perspectiva do Produto 5

2.2 Funções do Produto 5

2.3 Usuários e Suas Características 5

3. Diagrama de Casos de Uso 6

3.1 Descrição dos Atores 6

4. Requisitos Específicos 7

4.1 Funcionalidade 7

4.1.1 Exibir produtos em tabela 7

4.1.2 Criar produtos 7

4.1.3 Editar produtos 7

4.1.4 Deletar produtos 7

4.2 Usabilidade 8

4.2.1 Facilidade de Uso 8

4.3 Confiabilidade 8

4.3.1 Disponibilidade 8

4.3.2 Exatidão 8

4.4 Desempenho 8

4.4.1 Tempo de resposta 8

4.5 Suportabilidade 8

4.5.1 Facilidade de manutenção 8

4.6 Restrições de Design 9

4.6.1 Linguagens de programação 9

4.6.2 Gerenciamento de banco de dados 9

4.6.3 Virtualização de ambiente 9

4.7 Padrões de Arquitetura 9

4.7.1 Virtualização de ambiente 9

4.8 Interfaces 9

4.8.1 Interfaces do Usuário 9

4.8.2 Interfaces de Software 9

4.9 Padrões Aplicáveis 10

Especificação dos Requisitos de Software

# Introdução

## Finalidade

A finalidade desta Especificação de Requisitos de Software (SRS) é, através da descrição da solicitação do stakeholder, elicitar os requisitos prévios para reflexão e planejamento do desenvolvimento do software atendendo as demandas esperadas.

Este documento servirá como base inicial para o desenvolvimento, validação e manutenção do sistema, definindo requisitos funcionais, não funcionais, restrições de design e outros fatores críticos que forem observados durante a confecção deste documento.

## Escopo

Este documento aplica-se ao Sistema de Gerenciamento de Produtos Vendidos do cliente Nunes Sports, cujo objetivo principal é automatizar e otimizar o processo de gerenciamento, dando a possibilidade ao usuário de visualizar (em tabela), criar, editar e deletar produtos que estão previamente salvos na base de dados através de uma interface web.

## Definições, Acrônimos e Abreviações

* Sistema: Refere-se ao Sistema de Gerenciamento de Produtos Vendidos
* Usuário: Pessoa autorizada pelo cliente que interage com o sistema para gerenciar os produtos

## Referências

* Especificações do cliente: Instruções de **Everymind | a compass.uol company**.
* Documentações técnicas das tecnologias utilizadas

# Descrição Geral

## Perspectiva do Produto

O sistema será uma aplicação web integrada a um banco de dados, onde os produtos cadastrados serão armazenados e gerenciados. Em suma, estruturando a interação em “CRUD (Create-Read-Update-Delete)”.

## Funções do Produto

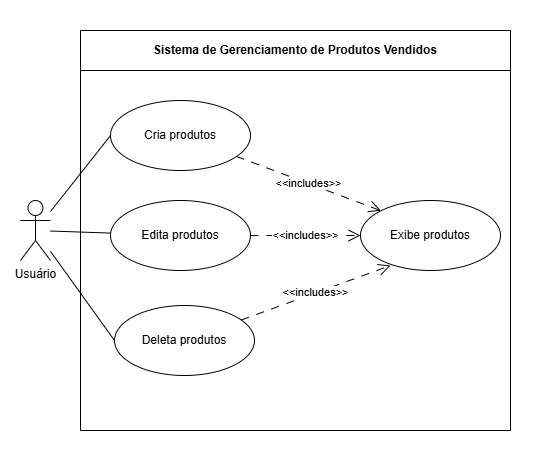
O sistema terá sua implementação inicial pautada na interação em “CRUD”, como mencionado no item anterior, possibilitando ao seu usuário:

* Visualizar produtos cadastrados
* Criar produtos
* Editar produtos existentes
* Deletar produtos

## Usuários e Suas Características

O sistema será utilizado por pessoa autorizada pelo cliente, gerenciando os produtos no banco de dados através da interface fornecida pelo sistema.

# Diagrama de Casos de Uso



## Descrição dos Atores

**Usuário**: Pessoa autorizada pelo cliente responsável por gerenciar os produtos vendidos

# Requisitos Específicos

## Funcionalidade

### Exibir produtos em tabela

**Descrição:** O sistema deve permitir que o usuário visualize uma lista de produtos cadastrados

**Detalhamento:**

* A lista deve apresentar informações básicas dos produtos, como nome, código, descrição e seu preço
* O usuário poderá realizar buscar por seu nome e código
* O usuário poderá filtrar os dados pelo preço (do maior para o menor ou inverso) ou por nome / código (ordem alfabética, ordem de inserção)

### Criar produtos

**Descrição:** O sistema deve permitir que o usuário cadastre produtos no banco de dados

**Detalhamento:**

* O usuário deve preencher os campos com o código do produto, nome, descrição e seu preço
* Todos os campos são obrigatórios, exceto o de descrição, e devem ser validados antes da inserção ser confirmada
* O sistema deve garantir que não haja produtos com o mesmo código
* O sistema deve permitir que haja produtos com o mesmo nome, pela existência de produtos de mesmo nome com características diferentes

### Editar produtos

**Descrição:** O sistema deve permitir que o usuário edite informações de um produto existente

**Detalhamento:**

* O usuário pode modificar os campos de produto, nome, descrição e preço, porém, o código se manterá fixo ao produto a ser editado
* Todos os campos são obrigatórios, exceto o de descrição, e devem ser validados antes da inserção ser confirmada

### Deletar produtos

**Descrição:** O sistema deve permitir que o usuário remova produtos do catálogo

**Detalhamento:**

* O sistema deve solicitar confirmação antes da realização da exclusão

## Usabilidade

### Facilidade de Uso

**Descrição:** O sistema deve ser intuitivo e fácil de usar, permitindo que novos usuários aprendam a utilizá-lo em menos de 30 minutos

**Detalhamento:**

* A interface deve ser consistente e limpa, direcionando o usuário através de botões de intenção clara e layout simples e organizado
* Mensagens de erro devem ser claras e fornecer sugestões para correção

## Confiabilidade

### Disponibilidade

**Descrição:** O sistema deve estar disponível para uso em 99,99% do tempo

**Detalhamento:**

* Manutenções planejadas devem ser comunicadas com antecedência

### Exatidão

**Descrição:** O sistema deve garantir a precisão dos dados exibidos e armazenados

**Detalhamento:**

* A exibição dos dados deve refletir exatamente o que está armazenado no banco de dados
* Realizar parametrização de entrada de dados é desejável, para reduzir a variabilidade de armazenamento

## Desempenho

### Tempo de resposta

**Descrição:** O sistema deve apresentar um tempo de resposta médio de até 5 segundos para as operações de exibição, criação, edição e exclusão de produtos

**Detalhamento:**

* Operações de busca e filtro devem ser concluídas em, no máximo 3 segundos
* O sistema deve ser capaz de gerenciar 10.000 produtos sem degradação significativa do desempenho

## Suportabilidade

### Facilidade de manutenção

**Descrição:** O sistema deve ser facilmente mantido e atualizado

**Detalhamento:**

* O sistema deve ser projetado seguindo os conceitos de SOLID e CLEAN CODE, aliados à arquitetura MVC

## Restrições de Design

### Linguagens de programação

**Descrição:** O sistema deve utilizar de tecnologias das ferramentas de mercado atuais

**Detalhamento:**

* Para a linguagem de programação backend, deve-se utilizar C# (ASP.NET)
* Para a linguagem de programação fronted, deve-se utilizar Angular
* Para gerenciamento do banco de dados, deve-se utilizar PostgreSQL, mas, para distribuição interna e teste, deve ser utilizado SQLite.

### Gerenciamento de banco de dados

**Descrição:** O sistema deve utilizar de tecnologias das ferramentas de mercado atuais

**Detalhamento:**

* Para gerenciamento do banco de dados, deve-se utilizar PostgreSQL ou SQLite, quando distribuição interna

## Padrões de Arquitetura

### Virtualização de ambiente

**Descrição:** Para aproveitamento das práticas de SOLID, deverá ser utilizada o padrão arquitetural, no que for cabível, do MVC (Model-View-Controller)

**Detalhamento:**

* O modelo deve ser responsável pela lógica de negócios e interação com o banco de dados
* A visão deve ser responsável pela apresentação dos dados, atuada pelo frontend
* O controlador deve mediar as interações entre modelos e visão

## Interfaces

### Interfaces do Usuário

**Descrição:** Baseada em interface web, acessível via navegadores modernos

**Detalhamento:**

* A interface deve ser responsiva, ajustando-se a diferentes tamanhos de tela
* Elementos interativos, como botões, devem ser claramente identificáveis

### Interfaces de Software

**Descrição:** O sistema deve ser capaz de comunicar-se através de API

**Detalhamento:**

* A comunicação entre backend e frontend deverá ser realizada através de criação e consumo de APIs
* CODE.