

REPÚBLICA DE ANGOLA

GOVERNO DA PROVÍNCIA DE LUANDA

INSTITUTO POLITÉCNICO Nº2039 “30 DE SETEMBRO”

**COORDENAÇÃO DO CURSO DE INFORMÁTICA**

**DESENVOLVIMENTO DE UMA APLICAÇÃO DESKTOP**

**PARA O GERENCIAMENTO DE STOCK DA LOJA MANGOLÉ**

**LUANDA, 2023**

INSTITUTO POLITÉCNICO Nº2039 “30 DE SETEMBRO”

**COORDENAÇÃO DO CURSO DE INFORMÁTICA**

**DESENVOLVIMENTO DE UMA APLICAÇÃO DESKTOP**

**PARA O GERENCIAMENTO DE STOCK DA LOJA MANGOLÉ**

Elaborado por: Augusto José Francisco

Isabel Catete Milombe

Laurinda Da Costa Ferraz

Orlando Martinho Pedro Saiombo

Paulina Tchivi Augusto

Princesa Da Cruz

Teresa Chivando Adriano

Orientados por: Eng.º Gasolina

# **FOLHA DE ROSTO**

**LUANDA, 2023**

# 

# **FICHA CATALOGRÁFICA**

Autorizamos a divulgação total ou parcial deste trabalho, por qualquer meio convencional ou electrónico para fins de estudos e pesquisas desde que citada a fonte.

Augusto José Francisco

Isabel Catete Milombe

Laurinda Da Costa Ferraz

Orlando Martinho Pedro Saiombo

Paulina Tchivi Augusto

Princesa Da Cruz

Teresa Chivando Adriano

Data \_\_\_\_\_/\_\_\_\_\_/\_\_\_\_\_

|  |
| --- |
| **FRANCISCO** A.J;**MILOMBE** I.C;**FERRAZ** L.C;**SAIOMBO** O.M.P; **AUGUSTO** ,P.T ;**CRUZ** P;**ADRIANO** T.C  DESENVOLVIMENTO DE UMA APLICAÇÃO DESKTOP  PARA O GERENCIAMENTO DE STOCK DA LOJA MANGOLÉ  Orientado por: Eng.º GASOLINA.  Nº de páginas: 52  Tipo de letra: Times New Roman.  Palavras-chave: Desktop,Gerenciamento,Estoque,Controle,Produtos; |

# **DEDICATÓRIA**

Dedico este projeto à minha família pelo apoio incondicional ao longo de toda a minha jornada. Agradeço também aos meus colegas e mentores pela orientação e motivação durante o desenvolvimento deste trabalho. Este projeto é uma homenagem à equipe da loja Mangolé, cujas experiências práticas foram uma fonte de inspiração para este estudo.

# EPÍGRAFE

# **AGRADECIMENTOS**

Agradeço à Deus pelo dom de vida, por iluminar os nossos caminhos, por nos proteger sempre;

À Direção do Instituto Politécnico Nº 2039 – “30 de Setembro” e em particular à Coordenação do Curso de Informática e todos os professores, em proporcionar-nos esta formação;

Aos nossos familiares que direta e indiretamente estiveram presentes nos momentos de tristeza e alegria;

Aos nossos colegas pelo companheirismo demonstrado durante o período da nossa formação;

A todos o nosso muito obrigado.

“Na informática, um erro é apenas uma oportunidade para criar algo inesperado”

Bill Gates

# **RESUMO**

O projeto envolve o desenvolvimento de uma aplicação desktop para auxiliar no gerenciamento de estoque da loja Mangolé. A aplicação visa organizar e controlar os produtos disponíveis na loja, facilitando o acompanhamento de entradas, saídas e níveis de estoque. Com uma interface amigável, a solução permitirá que os funcionários realizem atualizações rápidas e precisas sobre os produtos, contribuindo para uma gestão mais eficiente e precisa do inventário da loja.

# **LISTAS DE FIGURAS**

# Figura 1-Ultilização de algumas funsões em C......................................................................12

# Figura 2-Ultilização de algumas funsões Node.js.....................................................................13

# Figura 3-Modelo ou processo de desenvolvimento em cascata................................................19

# Figura 4-Diagrama de caso de uso............................................................................................23

# Figura 5 – Diagrama de classes concebido para o projecto......................................................24

# Figura 6 – MER da base dados (projecto desenvolvido).........................................................25

# Figura 7– Página principal do portal (projecto desenvolvido)..................................................26

# Figura 8 – Página de detalhe dos produtos (projecto desenvolvido).......................................26

# Figura 9– Página de Fornecedores (projecto desenvolvido).....................................................27

# Figura 10 – Página da Ficha de Stock (projecto desenvolvido)...............................................28

# Figura 11 – Página dos Funcionários (projecto desenvolvido)................................................29

# Figura 12 – Página do Perfil (projecto desenvolvido).............................................................30

# **LISTAS DE TABELAS**

# Tabela 1 -Requisitos não funcionais....................................................................22

# **LISTA DE ABREVIATURAS E SIGLAS**

ALGOL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ALGOrithmic Language

BASIC\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Beginner´s All-purpose Simbolic Instruction Code

CERN\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Conselho Europeu para Investigação Nuclear

COBOL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Common Business Oriented Language

C#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_CÊ SHARP

CSS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Cascading Style Sheets

DCL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Data Control Language

DDL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Data Definition Language

DER\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Diagrama Entidade Relacionamento

DML\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Data Manipulation Language

DOM\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Document Object Model

DQL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Data Query Language

FORTRAN\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Formula Translate

FTP\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_File Transfer Protocol

GPL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_General Public License

GUI\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Graphic User Interface

HTML\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_HyperText Markup Language

HTTP\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_HyperText Transfer Protocol

HTTPS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_HyperText Tranfer Protocol

IIS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Internet Information Server

LGPL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Lesser General Public Licence

MER\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Modelo Entidade Relacionamento

MySQL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_My Structured Query Language

RUP\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Rational Unified Process

SGBD\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Sistema de Gestão de Bases de Dados SGBDR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Sistema de Gestão de Bases de Dados Relacional

SQL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Structured Query Language Sucure

URL\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Uniform Resource Locator

UML\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Unified Modeling Language

WWW\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_World Wide Web

# **ÍNDICE**

FOLHA DE ROSTO i

FICHA CATALOGRÁFICA ii

DEDICATÓRIA iii

AGRADECIMENTOS iv

EPÍGRAFE v

RESUMO vi

LISTAS DE FIGURAS vii

LISTAS DE TABELAS viii

LISTA DE ABREVIATURAS E SIGLAS xi

I. INTRODUÇÃO 1

1.1. Justificação do tema 1

1.1.1. Delimitação do tema 1

1.2. Formulação do problema 1

1.3. Formulação de hipótese 1

1.4. Objectivos 1

1.4.1. Objectivo geral 2

1.4.2. Objectivos específicos 2

II. FUNDAMENTAÇÃO TEÓRICA 3

2.1. Programação 3

2.1.1. Breve historial da programação 3

2.1.2. Linguagem de programação 4

2.1.3. Paradigmas de programação 5

2.3. Desktop 6

2.3.1. O surgimento do Desktop 7

2.3.2. Servidores Desktop 7

2.3.3. Padrões da Desktop 8

2.4. Bases de dados 9

2.4.1. Modelos de bases de dados 10

2.4.2. Modelo Relacional 11

2.5. Sistema de Gestão de Base de Dados 11

2.6. Linguagens de programação utilizada 11

2.6.1.Front-end-C# 1

2.6.2. Back-end - Node.js (JavaScript): 12

2.7. Softwares utilizado e ferramentas 13

2.7.1. XAMPP 13

2.7.2. MySQL 13

2.7.3. Visual Studio Code 14

2.7.4. GUNA UI 14

2.8.5. Visual Studio 15

2.9. Tratamento e processamento de dados 15

2.9.1. Microsoft Office Word 15

2.9.2. MySQL Workbench 15

2.9.3. StarUML 15

2.9.4. UML 17

III. METODOLOGIA 18

3.1. Tipos de pesquisa 18

3.2. Processo de desenvolvimento de software 18

3.2.1. Processo em cascata(waterfall) 18

IV. RESULTADOS E DISCUSSÃO 20

4.1. Descrição do negócio 20

4.2. Requisitos funcionas 20

4.3. Requisitos não funcionais 21

4.4. Diagrama de casos de usos 22

4.5. Diagrama de classes 23

4.6. Diagrama Entidade Relacionamento 24

4.7. Página principal 25

4.8. Página dos produtos 26

4.9. Página dos fornecedores 27

4.10. Página da Ficha de Stock 27

4.11. Página dos Funcionários 28

4.12. Página do Perfil 29

V. CONCLUSÕES 31

VI. BIBLIOGRAFIA 32

# **INTRODUÇÃO**

O gerenciamento eficiente de estoque é essencial para o sucesso de qualquer empresa varejista, pois permite um controle rigoroso dos produtos disponíveis e contribui para uma tomada de decisões mais assertiva. Neste contexto, o desenvolvimento de uma aplicação desktop para o gerenciamento de estoque da loja Mangolé é uma iniciativa que visa otimizar os processos de controle de inventário. Esta aplicação permitirá que a loja mantenha registros precisos de entrada e saída de produtos, além de monitorar os níveis de estoque em tempo real. Ao integrar uma solução tecnológica específica para a loja Mangolé, espera-se aumentar a eficiência operacional, reduzir erros e proporcionar uma melhor experiência para os clientes.

## **Justificação do tema**

A aplicação desktop para gerenciamento de estoque da loja Mangolé é necessária para otimizar o controle de produtos e reduzir erros. Ela automatiza tarefas manuais, economizando tempo e permitindo maior foco em outras atividades críticas. A aplicação fornece relatórios em tempo real para auxiliar nas decisões de compras e reposição de estoque.

### **Delimitação do tema**

O projeto trata do desenvolvimento de uma aplicação desktop para gerenciamento de estoque da loja Mangolé. A solução proposta visa aprimorar o controle de produtos, automatizar processos de inventário e gerar relatórios em tempo real, facilitando as operações da loja. Isso permitirá à loja Mangolé manter um estoque balanceado e eficiente para atender às necessidades de seus clientes.

## **Formulação do problema**

Por que criar uma aplicação desktop para o gerenciamento de estoque da loja Mangolé?

## **Formulação de hipótese**

A criação de uma aplicação desktop para o gerenciamento de estoque da loja Mangolé aumentará a eficiência e precisão no controle de produtos, minimizando erros humanos e auxiliando na tomada de decisões estratégicas para otimizar o funcionamento da loja.

## **Objectivos**

Para a elaboração do projecto foram traçados os objectivos seguintes:

### **Objectivo geral**

Desenvolver aplicação desktop para gerenciamento de estoque.

### **Objectivos específicos**

* Fazer pesquisa bibliográfica.
* Fazer o levamento dos requisitos.
* Escolher as tecnologias a serem usados para criação da aplicação web.
* Desenhar os diagramas do sistema.
* Selecionar as linguagens de programação a serem usadas.
* Criar a base de dados.
* Definir interface gráfica.
* Fazer a codificação do sistema.

# **FUNDAMENTAÇÃO TEÓRICA**

Neste capítulo foi feita abordagem sobre as linguagens de programação, linguagens utilizadas, software utilizado e ferramentas, e o método utilizado para elaboração do projeto.

## **Programação**

“É o ato de escrever instruções para que máquinas executem um determinada função ou várias funções encadeadas ”, (NOLETO, 2012).

Para Zambaldi (2020), programação é um processo de escrita, testes e manutenção de programas de computadores, sendo, este programas conjuntos de instruções determinados pelo programador que descrevem tarefas a serem realizados pela máquina que atendem diversas funcionalidades.

“O processo de programação é uma conversa controlada entre um ser humano (tecnicamente preparado) e o computador propriamente dito. O processo de comunicação se faz com o uso de uma linguagem de programação que o computador entenda”, (MANZANO, 2016, p. 21).

## **Breve historial da programação**

Segundo Zambalid (2020), a origem da programação está diretamente ligado a história dos computadores. Afirmando que a primeira programadora, foi uma mulher chamada de Ada Lovelace. Ada Lovelace escreveu um conjunto de código para possibilitar a utilização da máquina analítica de Charles Babbage (1791 - 1871), uma máquina robusta de difícil comunicação, considerada precursora dos computadores atuais.

Segundo Zambalid (2020), a origem da programação está diretamente ligado a história dos computadores. Afirmando que a primeira programadora, foi uma mulher chamada de Ada Lovelace. Ada Lovelace escreveu um conjunto de código para possibilitar a utilização da máquina analítica de Charles Babbage (1791 - 1871), uma máquina robusta de difícil comunicação, considerada precursora dos computadores atuais.

Segundo Zambalid (2020), a origem da programação está diretamente ligado a história dos computadores. Afirmando que a primeira programadora, foi uma mulher chamada de Ada Lovelace. Ada Lovelace escreveu um conjunto de código para possibilitar a utilização da máquina analítica de Charles Babbage (1791 - 1871), uma máquina robusta de difícil comunicação, considerada precursora dos computadores atuais.

## **Linguagem de programação**

De acordo com Garcia (2005), linguagem de programação é uma linguagem destinada a ser usada por uma pessoa para expressar um processo através do qual um computador pode resolver um problema. Os quatro modelos(paradigmas) de LP(Linguagem de programação) corresponde aos pontos de vista dos quatro componentes citados. A eficiência da construção e a execução de programas depende da combinação dos quatro pontos de vista.

Segundo Ribeiro (2012), a definição de uma linguagem de programação consiste na sua sintaxe e na sua semântica, descrevendo-as como:

* Sintaxe - conjunto de símbolos ou regras de escritas da linguagem de programação;
* Semântica - significados que os símbolos assumem na linguagem de programação.

Para Xavier (2005), a proximidade de uma linguagem de programação com a linguagem humana determina o nível da linguagem de programação. Levando-o a classificar as linguagens de programação em três níveis diferentes.

* Linguagem de máquina - linguagem compreendida pelo computador cujas as instruções são representadas por vários 0 e 1 (bits).
* Linguagem de baixo nível - este tipo de linguagem de programação utiliza instruções próxima à compreensão da máquina (linguagem de máquina). Exemplo: linguagem assembly.
* Linguagem de alto nível – este tipo de linguagem de programação utiliza as instruções próximas do nível da compreensão humana (linguagem humana). Exemplos: Linguagem Visual Basic, Linguagem C, Linguagem Java, Delphi, Linguagem Clipper, etc.

De acordo com Manzano (2016, p. 22) diz que:

Tanto as linguagens de programação em baixo nível como em alto nível, quando em uso em um computador eletrônico, são traduzidas internamente para uma forma numérica convencional grafada em formato binário, representada pelos símbolos numéricos 1 (um) e 0 (zero). O valor 1 indica que algum componente interno do computador se encontra ligado, e se estiver com valor 0, indica que o componente se encontra desligado. Não importa o tamanho ou tipo de computador, pois todo computador só é operado internamente com códigos binários.

Para Manzano (2016, p. 22) as linguagens de programação podem didaticamente ser divididas em quatro categorias de gerações, sendo:

* Primeira geração - representada pelas linguagens de máquina e assembly;
* Segunda geração - representada pelas linguagens FORTRAN (primeira linguagem de alto nível), ALGOL, COBOL e, de certa forma, a linguagem BASIC;
* Terceira geração - representada pelas linguagens PU1, Pascal, C, MODULA-2, C++, Java, Lua;
* Quarta geração: representada basicamente por linguagens de consulta estruturada,  
  além de outras ferramentas, destacando-se a linguagem de consulta estruturada SQL.

## **Paradigmas de programação**

De modo geral a filosofia de uma linguagem de programação segue um determinado paradigma de programação, (RIBEIRO, 2020).

Paradigmas de programação são modelos de programar computadores, (SEBESTA, 2013).

“Os paradigmas de programação passaram por cinco fases evolucionárias, sendo programação tradicional, programação estruturada, programação modular, programação com abstração de dados e programação orientada a objetos”, (ROMÁN, 2008, p. 437),

* Programação tradici­onal**:** iniciou-se no final da década de 1950. Teve como maiores representantes as linguagens de programação FORTRAN, COBOL e BASIC, nas quais se encontram as estruturas de programação de sequência, desvio condicional simples, desvio condicional composto e laço de repetição incondicional.
* Programação estruturada**:** iniciou-se em meados da década de 1960 com o surgimento da linguagem de programação Pascal. Posteriormente ocorreram mudanças nas estruturas das linguagens de programação FORTRAN, COBOL e BASICpara que suportassem os paradigmas da programação estruturada. Na década de 1970 surgiu a linguagem de programação C. Nessas linguagens de programação encontram-se, além das estruturas tradicionais, as estruturas de programação de laços de repetição condicionais e seleção, paradigma que é o assunto central desta obra.
* Programação modular**:** iniciou-se no final da década de 1970 com o surgimento da linguagem de programação MODULA-2, na qual se encontram, além das estruturas de programação existentes em outros paradigmas, o uso de encapsulamento de módulos com pacotes de dados e funções.
* Programação com abstração de dados**:** iniciou-se na década de 1980 com a linguagem de programação ADA, na qual se encontram muitas das estruturas de programação existentes em outras linguagens, além de possuir a estrutura de programação de laço de repetição condicional seletivo (também encontrada na linguagem *BASIC* estruturada) e ter introduzido o conceito de dados abstratos.
* Programação orientada à objectos**:** iniciaram-se na década de 1960 com o surgimento das linguagens SIMULA 1, depois com SIMULA 67 (1967), tendo seu auge de percepção pelo mercado a partir da década de 1980 com o surgimento da linguagem *Smalltalk* (iniciada por volta da década de 1970) nos laboratórios da empresa XEROX), depois com as linguagens C++ (1980), Eiffel e *Object Pascal* (ambas em 1985), e posteriormente com as linguagens Java (1995) e C# (2000), nas quais se encontram a definição e construção de classes, objetos, encapsulamento, herança e polimorfismo.

Segundo Ribeiro (2012), a maneira como um programa é traduzido para ser executados num computador, ela indica que uma linguagem pode ser compilada, interpretada, ou ser um misto dos dois modos.

**2.3. Desktop**

As aplicações desktop são programas de software instalados e executados em computadores pessoais ou estações de trabalho, em vez de serem acessados através de navegadores ou de se basearem em servidores remotos. Elas oferecem várias vantagens, como desempenho rápido, uma interface de usuário rica e personalizada, e podem ser executadas offline, proporcionando autonomia em relação à conectividade com a internet. Além disso, as aplicações desktop podem integrar-se diretamente com os recursos de hardware e do sistema operacional do computador, permitindo acesso a periféricos e maior controle sobre o ambiente do usuário. São amplamente utilizadas em setores que necessitam de soluções estáveis e seguras para funções específicas, como gerenciamento de inventário, processamento de dados ou edição de mídia.

### **2.3.1. O surgimento Desktop**

As aplicações desktop surgiram com a popularização dos computadores pessoais nos anos 1980. À medida que o hardware se tornou mais acessível, aumentou a demanda por software especializado para tarefas específicas, como processamento de texto, planilhas e edição de imagens. O avanço dos sistemas operacionais com interfaces gráficas, como o Apple Macintosh e o Microsoft Windows, permitiu o desenvolvimento de aplicações mais sofisticadas e amigáveis.

Isso impulsionou a indústria de software a criar uma variedade de aplicações desktop para atender às necessidades de consumidores e empresas em diferentes áreas. Apesar do crescimento de soluções baseadas em nuvem, as aplicações desktop ainda são amplamente utilizadas por sua eficiência, estabilidade e integração com o hardware local.

### **2.3.2. Servidores Desktop**

Os servidores desktop, também conhecidos como servidores locais ou servidores de escritório, são computadores configurados para funcionar como servidores em ambientes de escritório ou domésticos. Eles desempenham funções semelhantes aos servidores tradicionais, mas geralmente em uma escala menor e para um número limitado de usuários.

Algumas características dos servidores desktop incluem:

* Armazenamento e compartilhamento de arquivos: Eles podem atuar como um repositório central para armazenar e compartilhar arquivos entre os usuários de uma rede local.
* Gerenciamento de impressoras e dispositivos: Os servidores desktop podem gerenciar e compartilhar recursos, como impressoras, scanners e outros dispositivos, com os usuários da rede.
* Controle de acesso e segurança: Eles podem ser usados para gerenciar permissões de usuários, autenticação e autorizações, garantindo que apenas pessoas autorizadas tenham acesso a determinados recursos.
* Hospedagem de serviços locais: Servidores desktop podem hospedar aplicativos locais, como software de gerenciamento de inventário ou banco de dados, para uso pelos usuários conectados à rede.
* Backup e recuperação de dados: Eles podem ser configurados para realizar backups automáticos dos dados da rede, facilitando a recuperação em caso de falhas ou perda de dados.

Os servidores desktop são uma opção mais acessível para pequenas empresas ou escritórios domésticos que não precisam de servidores em grande escala. Eles proporcionam muitos dos benefícios de um servidor tradicional, como centralização e gerenciamento de recursos, em um ambiente menor e com custos reduzidos.

### **2.3.3.** **Padrões Desktop**

Os padrões desktop referem-se a convenções e práticas estabelecidas para o design e desenvolvimento de aplicações para ambientes de desktop. Esses padrões visam proporcionar uma experiência de usuário consistente e intuitiva, melhorando a usabilidade e a acessibilidade dos aplicativos. Os padrões desktop abrangem várias áreas, como:

* Interface do usuário: Padrões relacionados ao layout, disposição de elementos, ícones, botões, cores e fontes, que tornam a interação com a aplicação mais intuitiva para os usuários.
* Navegação: Inclui convenções sobre como os usuários se movimentam entre diferentes seções ou funcionalidades do aplicativo, como menus, abas e barras de ferramentas.
* Interação e controle: Define como os usuários devem interagir com o aplicativo, por meio de cliques, arrastar e soltar, atalhos de teclado, entre outros, para maximizar a eficiência e a facilidade de uso.
* Comportamento de janelas: Refere-se a como as janelas do aplicativo devem se comportar, incluindo redimensionamento, minimização, maximização e gerenciamento de múltiplas janelas.
* Padrões de usabilidade: Incluem boas práticas para garantir que o aplicativo seja acessível a todos os usuários, incluindo aqueles com deficiências, e para otimizar a eficiência das tarefas.
* Consistência: Um aspecto crucial dos padrões desktop é a consistência com outros aplicativos e com o sistema operacional, de modo que os usuários possam aplicar conhecimento prévio de outras experiências de uso.

O cumprimento desses padrões ajuda a criar aplicações desktop mais eficazes e amigáveis, melhorando a satisfação do usuário e aumentando a adoção e retenção do software. Além disso, seguir esses padrões facilita a aprendizagem e o uso do aplicativo para novos usuários.

## **Bases de dados**

Segundo Laudon (2011), bases de dados são conjuntos de arquivos relacionados entre si, com registos sobre pessoas, lugares ou informações em geral. São colecções organizados de dados que se relacionam de forma criar algum sentido (informação) e dar mais eficiência durante uma pesquisa ou estudo científico. As bases de dados são operadas pelos Sistemas de Gestão de Bases de Dados (SGBD).

### **Modelos de bases de dados**

Como qualquer tecnologia, com o passar dos anos as bases de dados foram evoluindo, o que permitiu surgir diversos modelos de bases de dados, dentre as quais destacamos os seguintes:

* Modelo em rede: é um modelo que permite que várias tabelas sejam usadas simultaneamente por meio do uso de apontadores (ou referencias). Algumas colunas contêm apontadores para outras tabelas ao invés de dados. Assim, as tabelas são ligadas por referências, o que pode ser visto como uma rede.
* Modelo hierárquico: é uma variação particular do modelo em rede, limitando as relações a uma estructura semelhante à de uma árvore (hierarquia – tronco, galhos), ao invés do modelo mais geral direcionado por garfos.
* Modelo relacional: o modelo de base de dados relacional é uma colecção de múltiplos conjuntos de dados organizados em tabelas, possuindo uma relação bem definida entre as mesmas. A linguagem SQL (*Stuctured Query Language*) é usada por oferecer uma interface facilitada para interação com a base de dados relacional (WADE; CHAMBERLIN, 2012).
* Modelo Orientado a Objectos: este modelo é um sistema de gestão de base de dados que suporta a modelagem e criação de dados como objectos. Estes podem ser referenciados de maneira similar no que acontece com objectos usados na programação orientada a objectos (KIM, 19990).

Entre os modelos de base de dados citado acima, para o desenvolvimento do projecto adaptamos por utilizar o modelo da base de dados relacional, pelo facto de fácil interação com o SQL, que foi a linguagem de definição e consulta de dados usado no desenvolvimento do projecto.

### **Modelo Não Relacional**

## **Sistema de Gestão de Base de Dados**

Segundo Teory (2014), sistema de gestão de base de dados (SGBD) é um sistema de software genérico para manipular bases de dados, seu principal objectivo é retirar da aplicação cliente a responsabilidade de gerir o acesso a persistência, a manipulação e a reorganização dos dados.

Eles são responsáveis por todo que acontece na base de dados: ele salva os dados no dispositivo de armazenamento, mantém os dados mais acessados em memória, faz criptografia deles, controla o acesso às informações em caso de falhas ele possui cópia de dados. Exemplos: MS Access, MySQL, PostgreSQL, SQLite, Oracle, SQL Server, Sybase, Informix.

Para o desenvolvimento do projecto foi o utilizado o SGBD MySQL, por ser um SGBD Relacional e por ter suporte para interface da SQL.

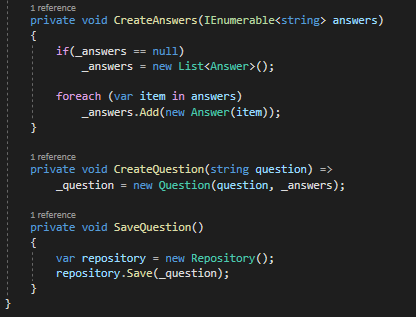
## **Linguagens de programação utilizada**

O projeto foi desenvolvido com uma combinação de linguagens de programação específicas para atender às necessidades distintas do front-end e do back-end da aplicação.

### **Front-end-C#**

C#: Usado para construir a interface do usuário e toda a lógica relacionada à camada visual da aplicação desktop. C# é uma linguagem moderna, orientada a objetos, com uma sintaxe clara e recursos robustos. Trabalhando em conjunto com frameworks como o .NET, C# permite a criação de interfaces de usuário ricas e responsivas, como as oferecidas pelo Windows Presentation Foundation (WPF) ou pela Universal Windows Platform (UWP). O uso de C# facilita a implementação de funcionalidades avançadas e a interação intuitiva com os usuários.

**Figura *1* – Utilização de algumas *funsões em C#***

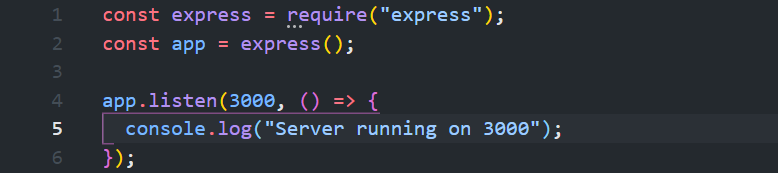


**Fonte: Alex Alves (2021)**

### **Back-end - Node.js (JavaScript):**

JavaScript (Node.js): Empregado no back-end para gerenciar a lógica de servidor, o processamento de dados e a interação com a base de dados. Node.js é uma plataforma baseada em JavaScript que possibilita o desenvolvimento de aplicações servidoras rápidas e eficientes. A programação assíncrona em Node.js permite lidar com múltiplas solicitações simultaneamente, otimizando o desempenho e escalabilidade da aplicação. A extensa biblioteca de pacotes disponíveis no npm (Node Package Manager) facilita a integração com outras ferramentas e serviços.

**Figura *2* – Utilização de algumas *funsões Node.js***



**Fonte: Thiago Costa Barbosa (2018)**

## **Softwares utilizado e ferramentas**

Para o desenvolvimento do projecto, foi utilizado diversos softwares e ferramenta de linguagens de programação, que auxiliaram para a concretização do abjectivo do projecto.

Estes softwares e ferramentas são definidos nas secções seguintes.

### **XAMPP**

XAMPP é um servidor independente de plataforma, software livre, que consiste principalmente na base de dados MySQL, o servidor web Apache e os interpretadores para linguagens de script: PHP e Perl. O nome da abreviação de X (para qualquer dos diferentes sistemas operativos), Apache, MySQL, Perl e PHP. (APACHE FRIENDS, 2023).

### **MySQL**

O MySQL é um sistema de gestão de banco de dados relacional, que suporta grande quantidade de dados com bastante eficácia, além de oferecer estabilidade, suporte por parte de seus desenvolvedores e baixo custo, (SUEHRING, 2002). O autor afirma que MySQL é um SGBDR livre e gratuito, sendo seu código fonte aberto, com isso vem rapidamente ocupando o seu espaço no mercado. Além de ser fácil usar, e com melhorias contínuas, ele é coberto pela GPL e a GNU LGPL.

### **Visual Studio Code**

Visual Studio Code é um editor de código criado pela Microsoft, que segundo a mesma, teve o intuito inicial de apresentar de forma mais simples e prático o ambiente de desenvolvimento semelhante a IDE Visual Studio, que tem intuito de ser um ambiente de desenvolvimento para aplicações C# e .NET. O foco do VSCode é o desenvolvimento de aplicações web usando JavaScript, TypeScript e Node.js. Além disso possui uma vasta biblioteca de extensões criadas pela comunidade, que permite que o mesmo seja usado para desenvolvimento de aplicações com as mais diversas linguagens. (RASK et al. 2021).

### **GUNA UI**

Guna UI é uma coleção de controles de interface de usuário (UI) aprimorados e customizáveis para aplicações desktop .NET, incluindo Windows Forms. Ele é projetado para facilitar a criação de interfaces modernas, visualmente atraentes e amigáveis ao usuário.

Os principais aspectos do Guna UI incluem:

* Variedade de Controles: Guna UI oferece uma ampla variedade de controles, como botões, caixas de texto, barras de progresso, menus e muito mais, todos com estilos atualizados e opções de personalização.
* Estilo Moderno: Uma das principais características do Guna UI é o design moderno de seus controles, com foco em usabilidade e estética contemporânea. Isso ajuda a proporcionar uma experiência de usuário mais agradável e atual.
* Fácil de Usar: Os controles do Guna UI são fáceis de adicionar a projetos existentes e configuram-se com simplicidade, graças à interface intuitiva e às opções de personalização disponíveis.
* Personalização: Além de estilos pré-definidos, o Guna UI permite que desenvolvedores personalizem os controles para atender às necessidades específicas de suas aplicações. Isso inclui cores, fontes, tamanhos e outros aspectos visuais.
* Desempenho: Guna UI é projetado para ser leve e eficiente, minimizando o impacto no desempenho da aplicação. Isso é importante para manter a fluidez da experiência do usuário.
* Documentação e Suporte: A coleção Guna UI vem com documentação e suporte ao desenvolvedor, facilitando a implementação e uso dos controles nas aplicações.

O Guna UI é uma ótima opção para desenvolvedores que desejam criar interfaces de usuário modernas e profissionais para suas aplicações desktop, especialmente com o uso de Windows Forms no ambiente .NET. Ele ajuda a elevar o design visual e a usabilidade das aplicações, proporcionando uma experiência mais satisfatória para os usuários finais

### **Visual Studio**

Visual Studio é uma IDE (ambiente de desenvolvimento integrado) da Microsoft, popular entre desenvolvedores por seus recursos abrangentes. Ele suporta diversas linguagens de programação, como C#, C++, F#, Python e JavaScript, facilitando o desenvolvimento de software em várias plataformas, incluindo web, desktop e mobile. Visual Studio oferece poderosas ferramentas de depuração, integração com serviços como Git e Azure DevOps, além de recursos de colaboração para equipes de desenvolvimento. Ele também possui uma versão mais leve, o Visual Studio Code, amplamente utilizado por desenvolvedores.

## **Tratamento e processamento de dados**

Para o tratamento e processamento de dados foi utilizado computador no qual foram foi necessário a utilização de alguns softwares, que serão descritos na secção a seguir.

### **Microsoft Office Word**

Segundo Silva (2017), Microsoft Office Word é um editor de textos compostos por uma série de ferramentas poderosas que auxiliam na elaboração de ficheiros, que nele são chamadas de documentos. O autor afirma que com este aplicativo pode se criar qualquer tipo de documento que envolva textos, revistas panfletos, joranis, etc. Além de poder enriquecer documento utilizando recursos como tabelas, gráficos, índices, entre muitos outros.

### **MySQL Workbench**

MySQL Workbench é uma ferramenta GUI para o projecto de bases de dados que integra o desenvolvimento de códigos SQL, formato padrão de script de base de dados, gestão, modelagem de dados, criação e manutenção de base de dados MySQL em um único ambiente sinérgico.

### **StarUML**

StarUML é uma ferramenta de modelagem de software que oferece suporte à criação de diagramas UML (Unified Modeling Language) e outras representações gráficas de design de software. Esta ferramenta é utilizada por engenheiros de software, analistas e designers para visualizar, documentar e planejar arquiteturas de software e sistemas.

Aqui estão alguns dos principais aspectos do StarUML:

* Suporte a UML: StarUML oferece suporte completo à UML, permitindo a criação de vários tipos de diagramas, incluindo diagramas de classes, casos de uso, sequências, atividades, componentes, entre outros. Isso auxilia no planejamento e documentação da arquitetura de software.
* Integração com outras tecnologias: A ferramenta permite exportar e importar arquivos nos formatos de modelagem mais comuns, como XMI (XML Metadata Interchange), e também se integra com outras ferramentas de desenvolvimento de software.
* Facilidade de uso: StarUML possui uma interface intuitiva, com ferramentas de arrastar e soltar para criar e editar diagramas de maneira eficiente. Isso facilita a modelagem e visualização de projetos de software.
* Personalização e extensibilidade: A ferramenta oferece recursos de personalização para adaptar os modelos às necessidades específicas de um projeto. Além disso, ela suporta extensões para adicionar funcionalidades ou diagramas personalizados.
* Documentação de projeto: StarUML facilita a documentação do projeto de software, gerando relatórios ou diagramas de design de alta qualidade. Isso ajuda a manter uma documentação clara e atualizada ao longo do ciclo de vida do projeto.
* Compatibilidade multi-plataforma: StarUML é compatível com sistemas operacionais como Windows, macOS e Linux, tornando-a acessível a uma ampla gama de desenvolvedores e organizações.

StarUML é uma ferramenta valiosa para equipes de desenvolvimento que precisam criar e manter modelos e diagramas UML para projetos de software. Ela ajuda a visualizar a arquitetura e os fluxos de dados de um sistema, auxiliando no planejamento, comunicação e documentação entre as partes interessadas de um projeto.

### **UML**

“UML (Unified Modeling Language) é uma família de notações gráficas, apoiada por um metamodelo único, que ajuda na descrição de sistemas de software, particularmente construídos utilizando o estilo orientado à objetos (OO)”, (FOWLER, 2007, p .25).

# **METODOLOGIA**

A metodologia utilizada para desenvolvimento do projecto passou por seguintes fases:

* Estabelecimento do tema.
* Formulação do problema.
* Formulação das hipóteses.
* Estabelecimento dos objectivos.
* Pesquisa bibliográfica.

## **Tipos de pesquisa**

Para o desenvolvimento do projecto realizou-se a pesquisa bibliográfia

## **Processo de desenvolvimento de software**

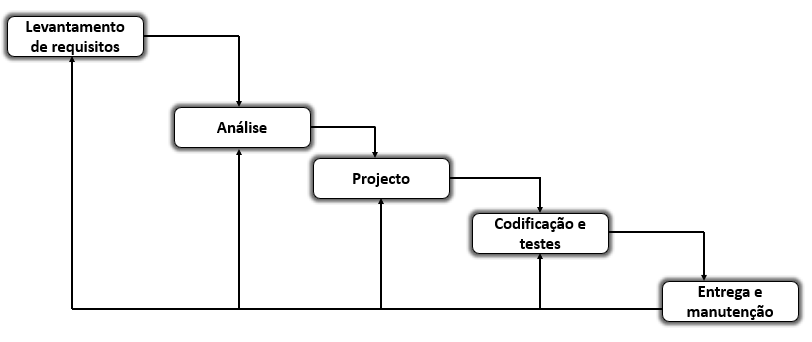
Para o desenvolvimento do projecto foi utilizado o processo em Cascata.

### **Processo em cascata(waterfall)**

O processo em cascata consiste em subdividir o projecto com base nas actividades, como: Levantamento de requisitos, análise dos requisitos, projecto, codificação e teste, (FOWLER, 2007).

Para Sommerville (2011), processo em cascata é um processo e desenvolvimento em que as actividade são estruturadas numa cascata forma linear e sequencial de forma que uma tarefa só poderá terá início quando a anterior tiver terminado, o desenvolvimento é alinhado de forma disciplinada como é mostrado na figura (Figura 10).

**Figura 3 -** **Modelo ou processo de desenvolvimento em cascata**



**Fonte: Sommerville (2011).**

O modelo cascata é composto pelas seguintes fases:

* Levantamento de requisitos: define-se os objectivos, regras e restrições, as funcionalidades e necessidades que o sistema terá, é especificada o orçamento, cronograma entre outro.
* Análise: compreende-se os requisitos e as funcionalidades do software, é especificada a descrição completa do comportamento do software.
* Projecto: descrição do contexto do software, indicar como o software devem ser implementado, é especificado a parte técnica de cada parte do software, considerando questões como: interface com usuário e armazenamento de dados.
* Codificação e testes: As unidades do software devem ser codificadas em qualquer linguagem de programação. Após a codificação as funcionalidades são testadas e corrigidas individualmente ou em conjunto. Visa garantir se o código traz o resultado esperado.
* Entrega e manutenção: liberação do sistema para utilização e são oferecidos treinamentos aos usuários envolvidos. A manutenção se encarrega de corrigir os erros que são encontrados, bem como ajustes e novas exigências conforme e evolução do sistema.

# **RESULTADOS E DISCUSSÃO**

Neste captítulo são mostrados os resultados obtidos ao desenvolver este projecto.

## **Descrição do negócio**

A loja Mangolé é uma empresa de varejo que oferece uma variedade de produtos, desde itens de consumo cotidiano até produtos específicos de nicho, atendendo a uma ampla base de clientes. O negócio se destaca por sua seleção criteriosa de produtos de qualidade, preços competitivos e atendimento ao cliente diferenciado. Como uma loja física, enfrenta desafios típicos de gerenciamento de estoque, como manter níveis adequados de mercadorias para atender à demanda dos clientes e evitar excessos ou faltas de produtos.

Para lidar com esses desafios, a loja busca otimizar seus processos internos, especialmente no gerenciamento de estoque, para manter uma operação eficiente e eficaz. A Mangolé está comprometida com a satisfação do cliente, garantindo que os produtos desejados estejam disponíveis. Ao aprimorar seus sistemas de gerenciamento de estoque, a loja pode melhorar o fluxo operacional e a lucratividade, oferecendo aos clientes uma experiência de compra melhorada.

## **Requisitos funcionas**

Os requisitos funcionais são as declarações de serviços que o sistema deve fornecer como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também estabelecer explicitamente o que o sistema não deve fazer. (SOMMERVILLE, 2007).

Para o desenvolvimento do projecto identificou-se os seguintes requisitos funcionais:

* Cadastrar
* Visualizar
* Actualizar
* Eliminar

## **Requisitos não funcionais**

Os requisitos não funcionais são restrições sobre os serviços ou as funções oferecidas pelo sistema. Eles incluem restrições de timing, restrições sobre o processo de desenvolvimento e padrões. (SOMMERVILLE, 2007). A tabela abaixo (Tabela 4) mostra os requisitos não funcionais identificado para desenvolvimento do projecto.

**Tabela 1 –** **Requisitos não funcionais do projecto**

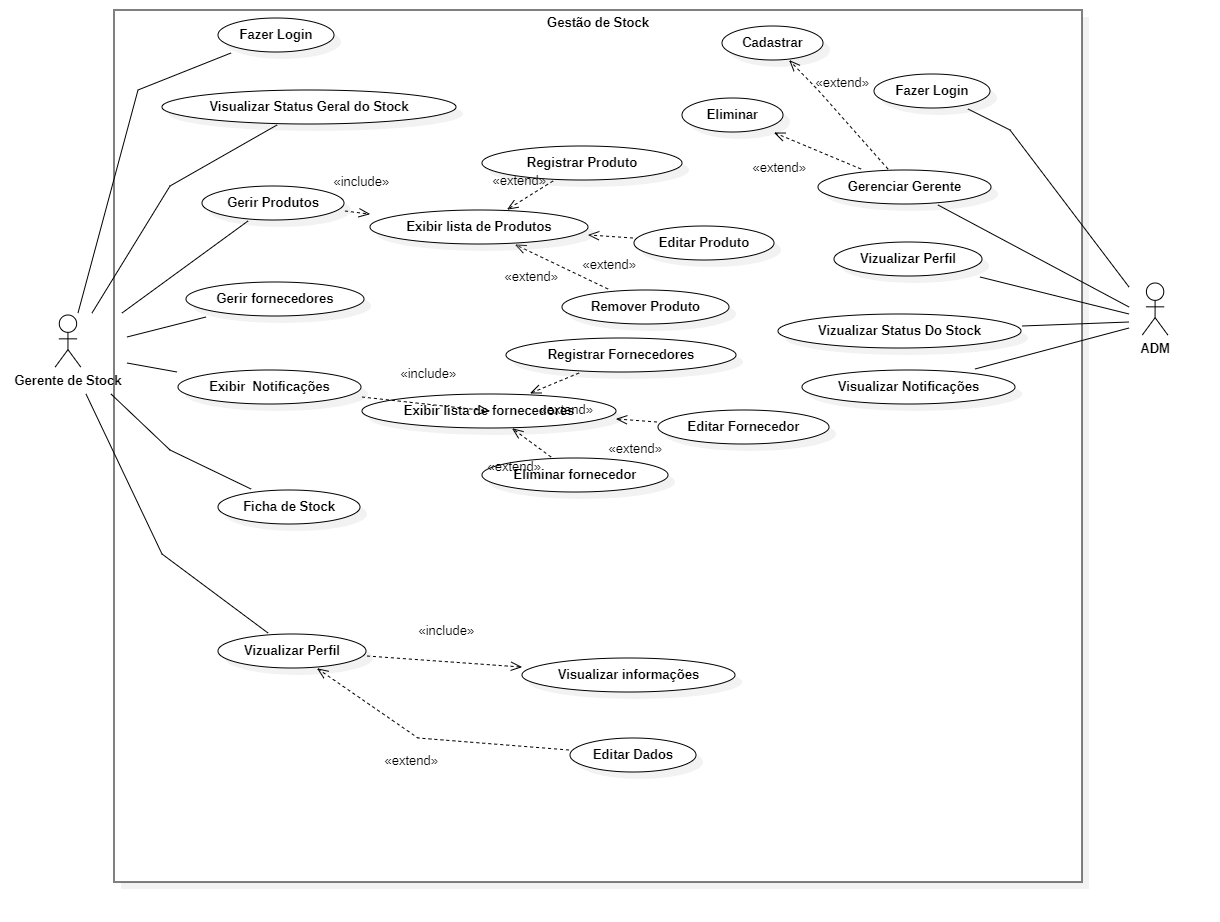
|  |  |  |
| --- | --- | --- |
| **Nº do RNF** | **Características** | **Descrição** |
| RNF1 | Aparência ou interface externa | Muito legível  Simples de usar  Interactivo |
| RNF2 | Usabilidade | Fácil de usar por pessoas sem muita experiência  Interface gráfica consistente  Rapidez na obtenção dados desejados |
| RNF3 | Rendimento | Grande disponibilidade, ao ser acessível à qualquer lugar da internet |
| RNF4 | Portabilidade | Poderá ser usado em qualquer plataforma que suporte .NET 7, JavaScript, MYSQL, Apache (GNU\Linux, Windows XP ou Superior, Mac, etc). |
| RNF5 | Software | Sistema operativos Windows XP ou superior GNU\Linux, Mac. |
| RNF6 | Desenho e implementação | Ferramenta de desenvolvimento: MySQL Workbench, XAMPP, Visual Studio Code ,Visual Studio e Insomnia  Linguagens de programação: C#, JavaScript |
| RNF7 | Segurança | Os dados confidencias são protegidos, só se pode realizar operações sobre ele depois de se fazer autentificação ao sistema, será necessária uma conta de utilizador com sua senha.  Nenhuma página restrita deverá ser acedida sem autenticação prévia, mesmo que se conheça o seu nome.  O administrador do sistema poderá controlar todas as operações feitas pelos outros utilizadores  O sistema utilizará protocolo HTTPS para encriptação do código fonte para que os usuários vejam o código fonte encriptado(codificado)  Sistema beneficiária do MD5 e que permitirá a encriptação das informações ao serem enviados para o servidor de base de dados MySQL.  Sistema utilizará o hash para encriptação das senhas, antes do armazenamento na base de dados. |

**Fonte: Autor**

## **Diagrama de casos de usos**

O diagrama de casos de usos descreve a funcionalidades e os utilizadores do sistema. Ele é utilizado para mostrar os relacionamentos entre actores quem empregam o sistema e os casos de uso utilizados por eles (LEE, TEPFENHART, 2001). A figura 5 apresenta o diagrama de casos de uso do projecto.

**Figura 4 – Diagrama de casos de uso (projecto desenvolvido)**

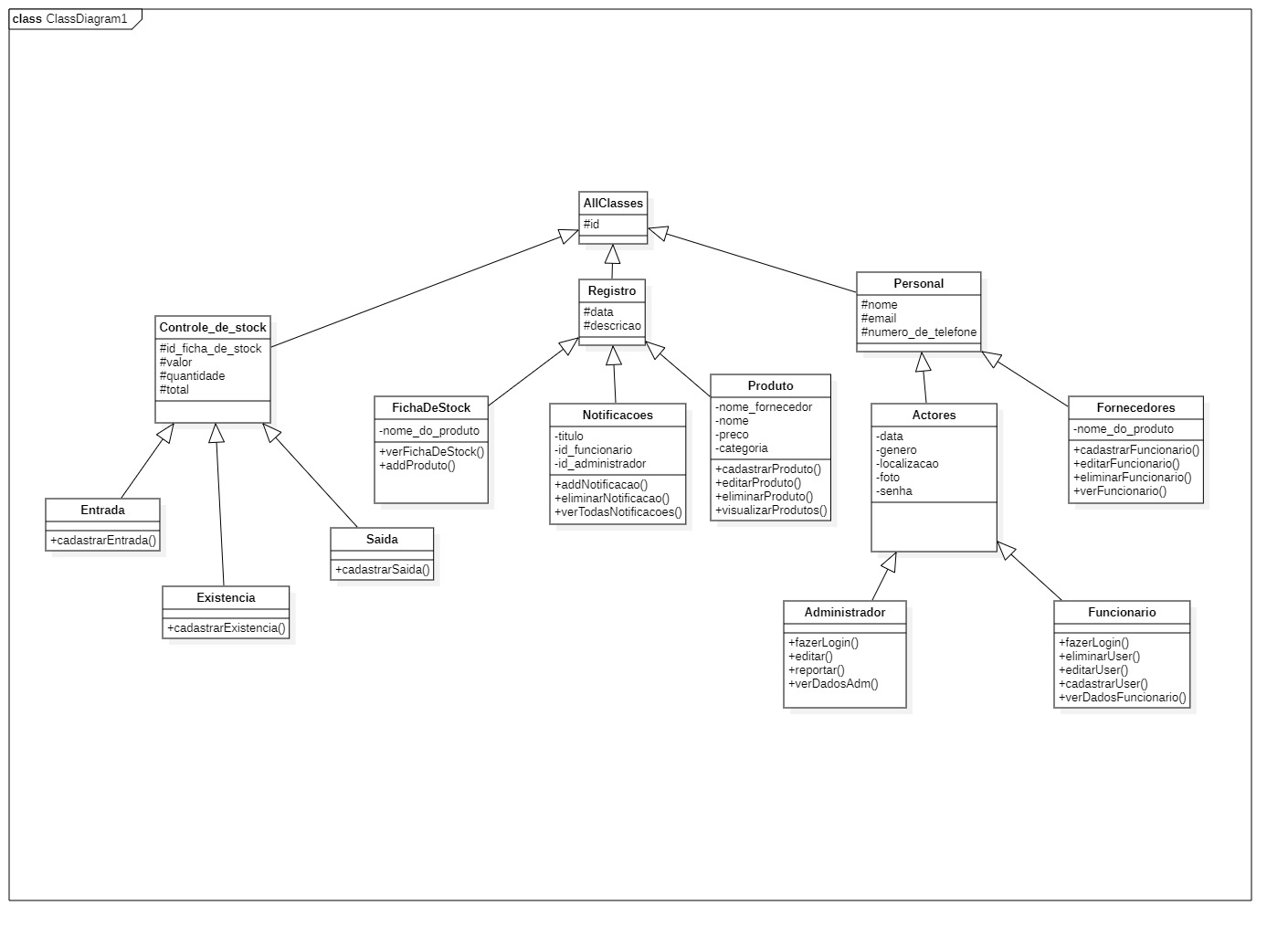


**Fonte: autor**

## **Diagrama de classes**

O diagrama de classes representa a estrutura do sistema, recorrendo ao conceito de classe e as suas relações, o modelo de classes resulta de processo de abstração onde são identificados os objectos relevantes do sistema em estudo (BEZERRA, 2006). A (Figura 6) apresenta o diagrama de classes concebido para este sistema.

**Figura 5 – Diagrama de classes concebido para o projecto**

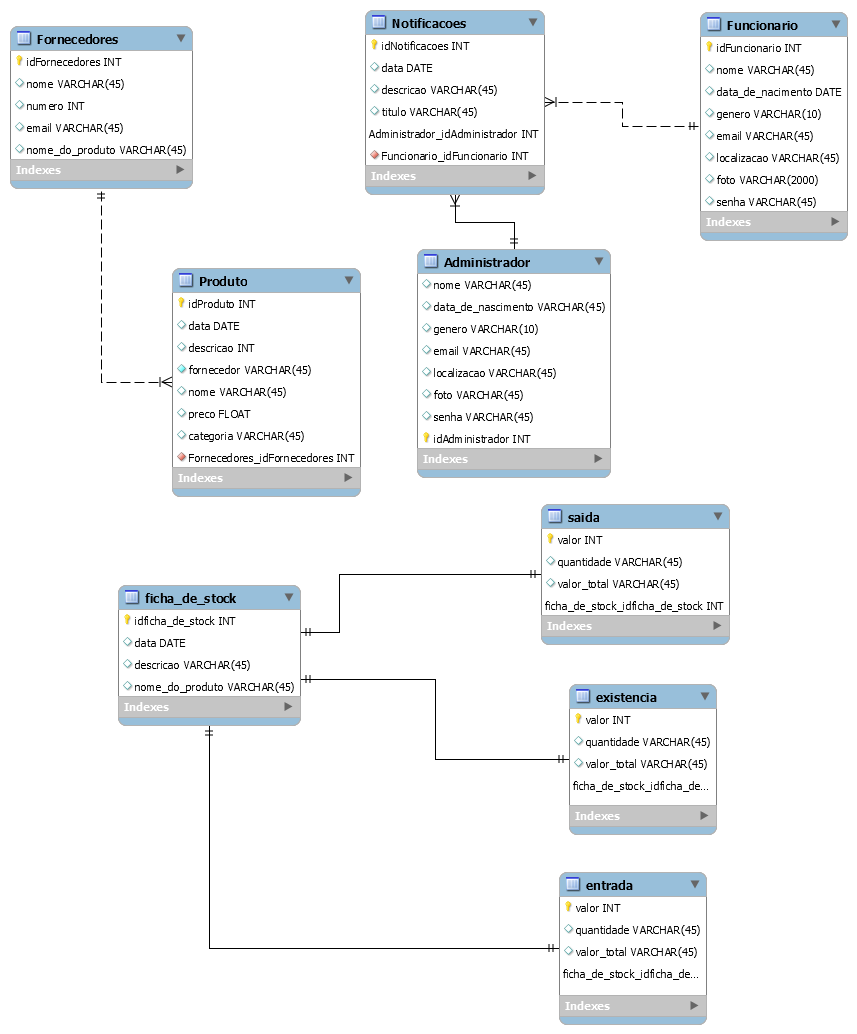


**Fonte: autor**

## **Diagrama de Entidade Relacionamento**

Modelo entidade relacionamento (Também conhecido como Modelo ER, ou simplesmente MER), é um modelo conceitual utilizado na Engenharia de Software (ES) para descrever objectos envolvidos no domínio de um sistema a ser construído, incluído seus atributos e relacionamentos. Ela permite representar de forma abstrata a estrutura que irá construir a base de dados. A (Figura 7) ilustra o modelo entidade relacionamento da base de dados do sistema desenvolvido.

**Figura 6 – MER da base dados (projecto desenvolvido)**

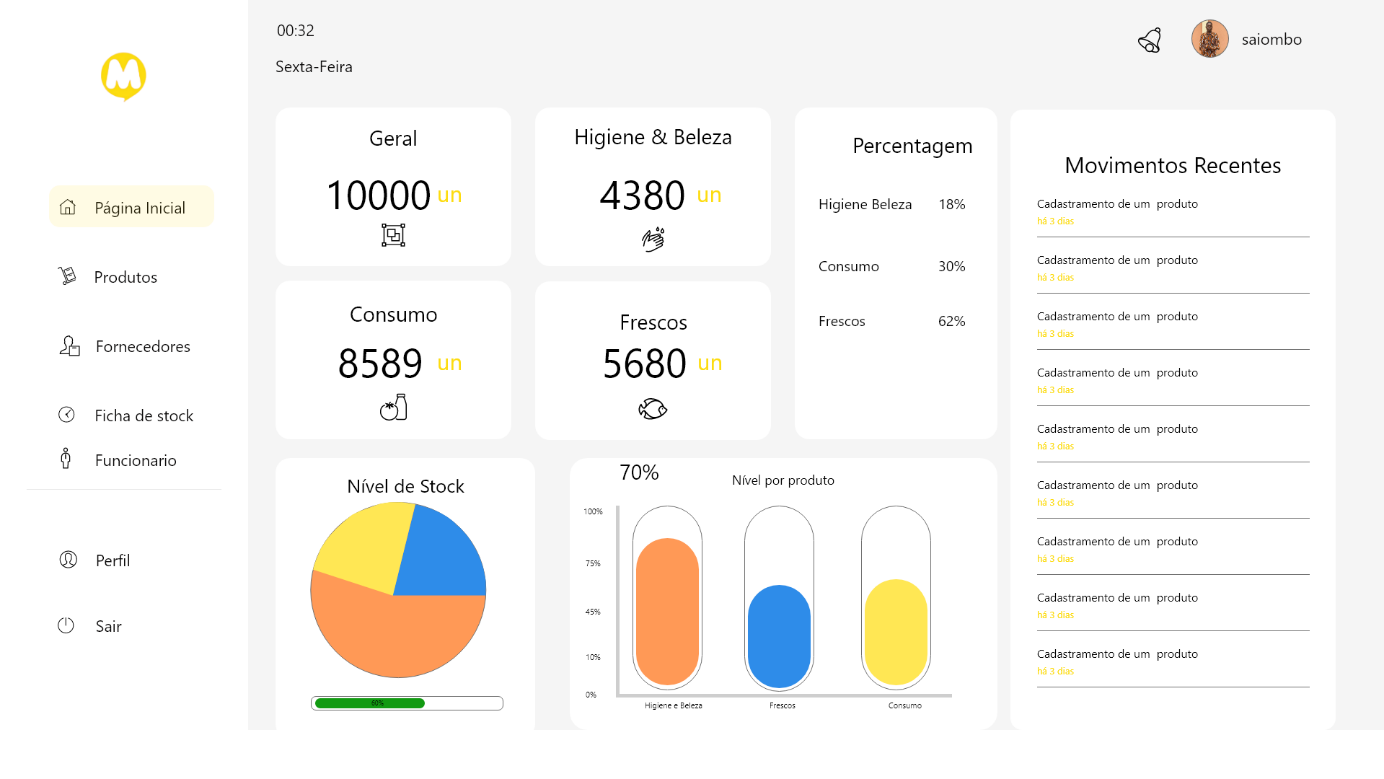


**Fonte: autor**

## **Página principal**

A página principal do sistema de Gestão de Stock, onde é ilustrado o menu de navegação, Página Inicial,Produtos, Fornecedores, Ficha de Stock,Funcionários,Perfil.A (Figura 7) mostra a página principal do site.

**Figura 7– Página principal do portal (projecto desenvolvido)**

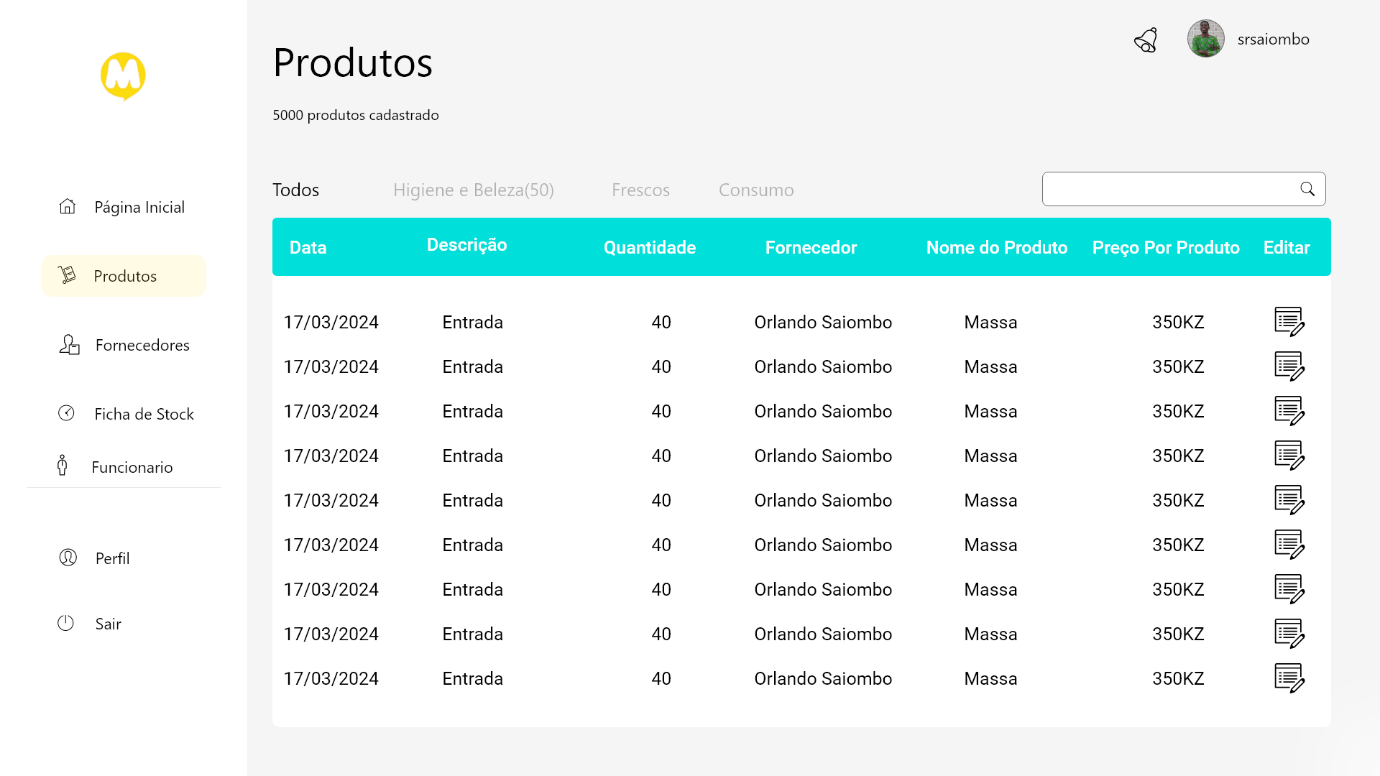


**Fonte: Autor**

## **Produtos**

Para aceder a está página da parte da visita, o internauta irá ao menu de navegação e clicando em Produtos, o levará a uma secção onde terá os produtos cadastrados. A figura seguinte (Figura 8) mostra uma página dos Produtos detalhadamente.

**Figura 8 – Página de detalhe dos produtos (projecto desenvolvido)**

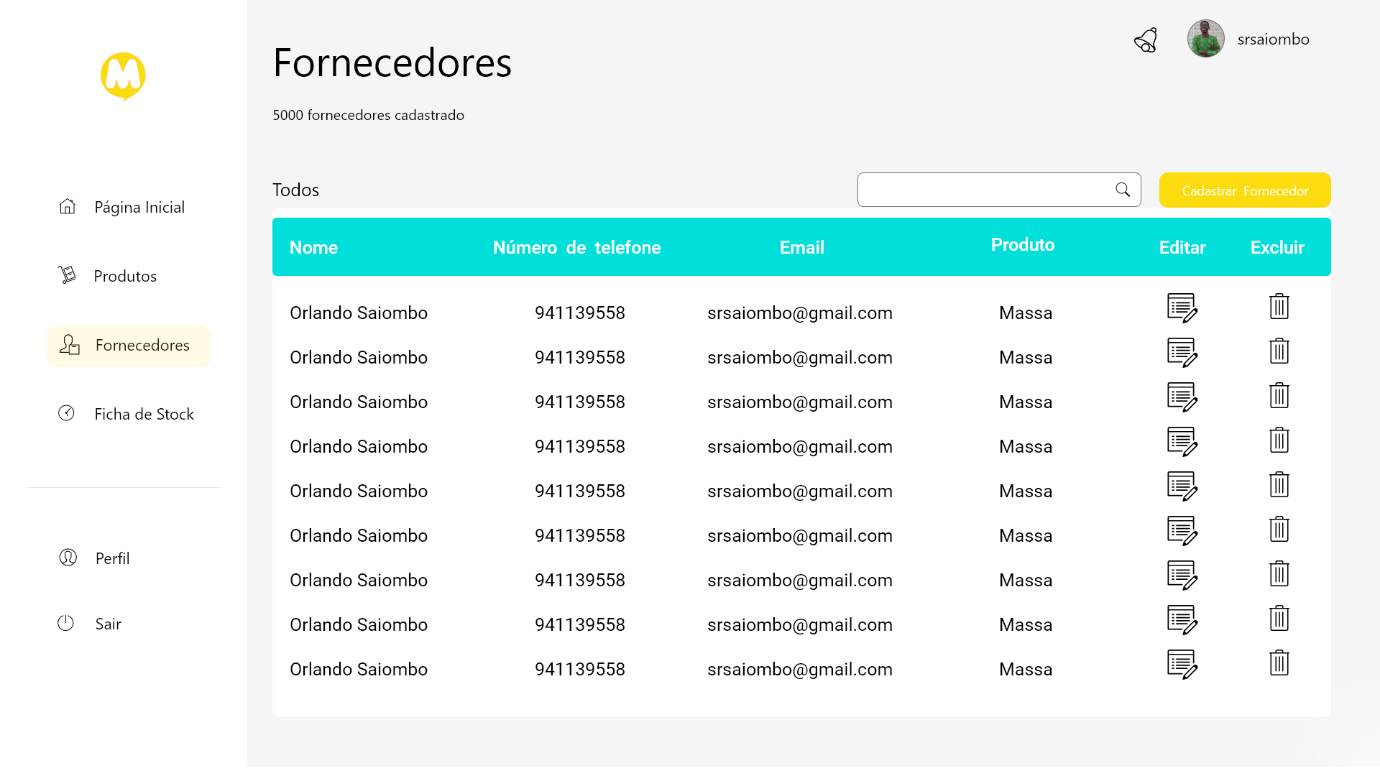


**Fonte: autor**

## **Fornecedores**

Esta página é reservada para o cadastramento do Fornecedores ,acessível apenas para o administrador como mostra a (Figura 9).

**Figura 9– Página de Fornecedores (projecto desenvolvido)**

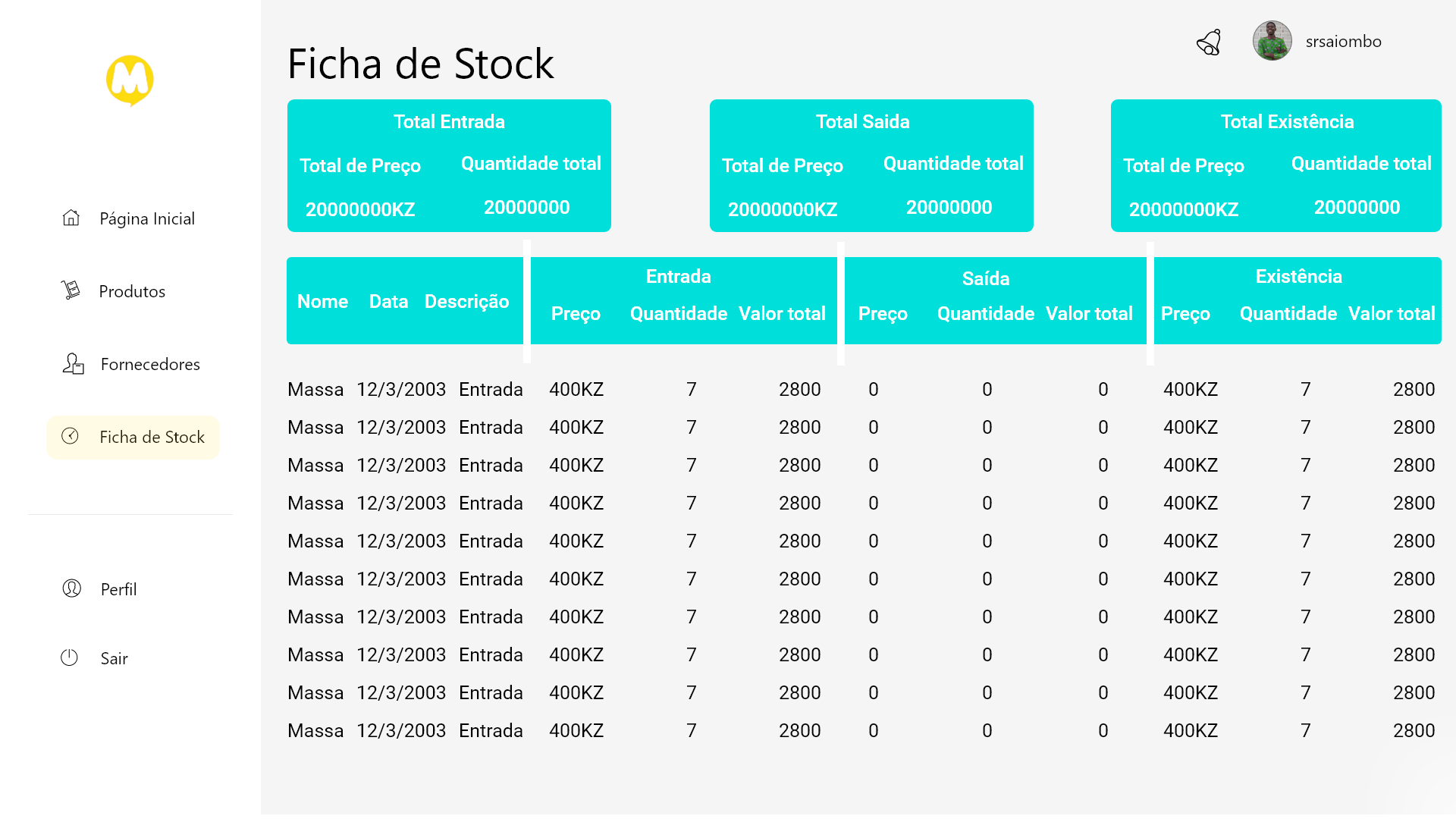


**Fonte: autor**

## **Ficha de Stock**

Esta página é reservada para ver o resumo do stock aplicando o método primeiro a entrar e primeiro a sair como mostra a (Figura 10).

**Figura 10 – Página da Ficha de Stock (projecto desenvolvido)**

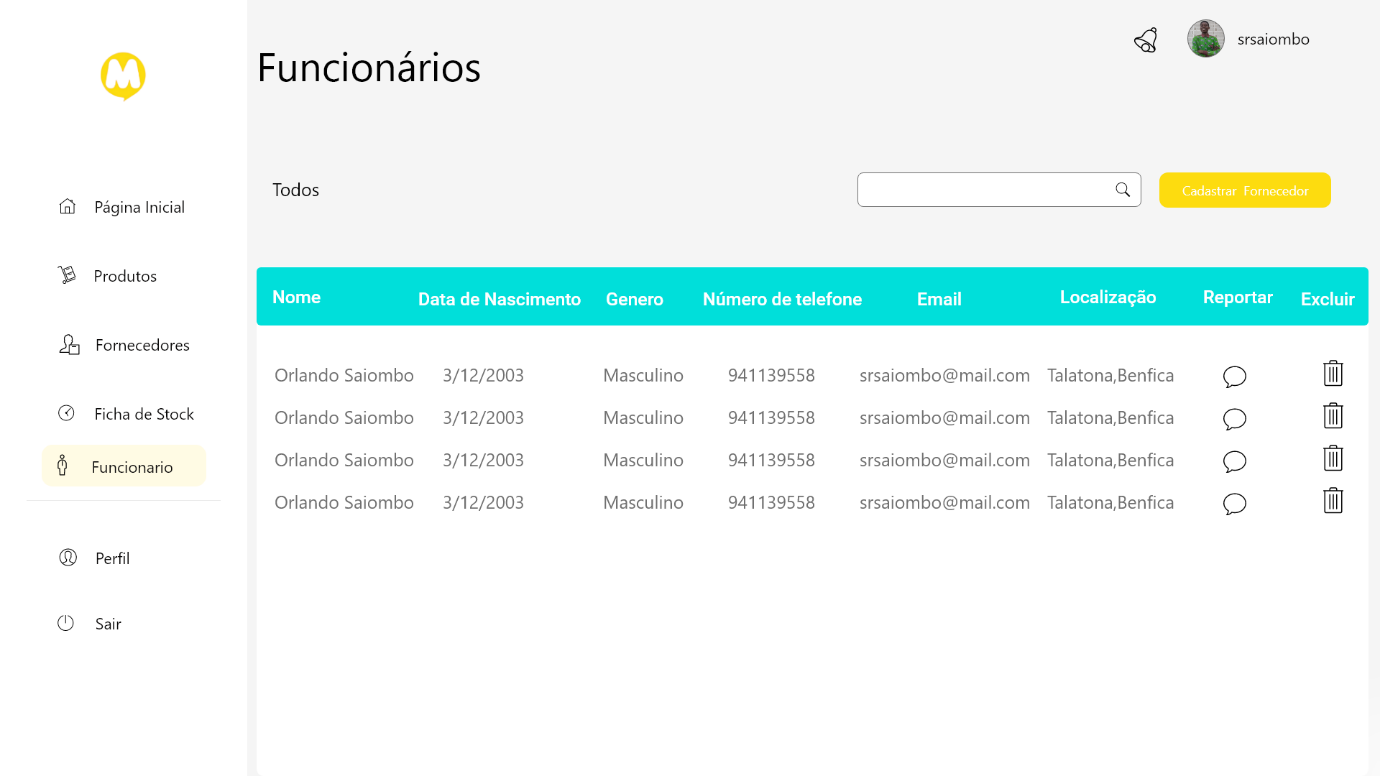


**Fonte: autor**

## **Funcionários**

Esta página é reservada para ver os funcionários do sistema reservado para o administrador como mostra a (Figura 11).

**Figura 11 – Página dos Funcionários (projecto desenvolvido)**

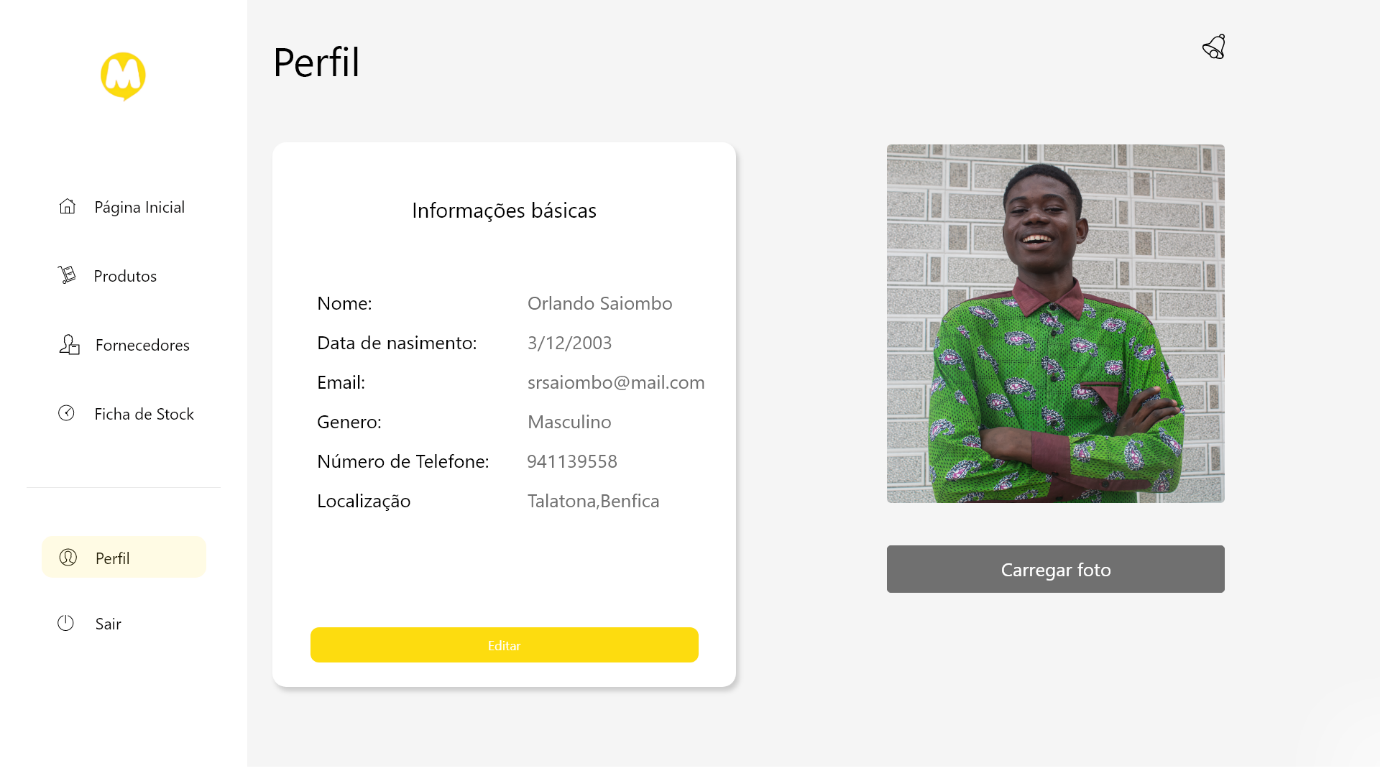


**Fonte: autor**

## **Perfil**

Esta página é reservada para ver os dados pessoais do usuário como mostra a (Figura 12).

**Figura 12 – Página do Perfil (projecto desenvolvido)**



**Fonte: autor**

# **CONCLUSÕES**

De acordo com os objectivos traçados para desenvolvimento deste projecto conseguimos concluir os seguintes:

* Conseguiu-se fazer a pesquisa bibliográfica
* Levantou-se os requisitos necessário.
* As tecnologias foram escolhidas, para o desenvolvimento da nossa aplicação web (Xampp, MySQL Workbench, StarUML, VSCode,Visual Studio).
* Selecionamos as linguagens para desenvolvimento da aplicação, como C#, JS
* Concretizou-se a criação da base de dados da aplicação.
* Definimos os Layouts da aplicação.
* Com as linguagens selecionada como C# e JS, conseguiu-se fazer a codificação da aplicação.

# **BIBLIOGRAFIA**

Ao adicionar as referências do StarUML e Visual Studio, e removendo a do Astah Community, a lista de bibliografias ficaria assim:

APACHE FRIENDS.XAMPP. Disponível em: <http://www.apachefriends.org>. Acesso em 19 janeiro 2023.

BEZERRA, E. (2006). Princípios de análise e projeto de sistemas com UML. 2ª ed. Editora Campus/Elsevier.

CASTELS, M.(2001). A galáxia da Internet: Reflexões sobre internet, os negócios e a sociedade. Edição Brasileira 2003. Rio de Janeiro: Jorge Zahar Editor Lada.

COELHO, P.(2000). Criação Fácil de Páginas Web com Office 2000 e Front Page 2000. Editora FCA.

CLARKE, I. & FLAHERTY, T. B. (2003) Web-based B2B portals. Industrial Marketing Management, 15-23.

DIAS, Cláudia.(2003). Usabilidade na web. Criando Portais mais acessíveis. Rio de Janeiro: Alta Books, 269 p.

ELMASRI, R.; NAVATHE, G. K. (2005). Sistemas de banco de dados. São Paulo: Pearson Addison, Wesley.

FOWLER, M.(2007). UML essencial: um breve guia para a linguagem-padrão de modelagem de objectos. 3ª ed. Porto Alegre: Bookman.

FREITAS, R.A; QUINTANILLA, L.W; NOGUEIRA, A.S. (2004). Portais Corporativos: Uma Ferramenta Estratégica Para Gestão Do conhecimento. Ed. Brasport Livros e Multimídia, Brasil.

GILES, D. (2010). Psychology of the media. New York. Palgrave Macmillan.

GLAZAR, Jean Eduardo. (2011). Programação para web: Curso técnico em informática. Colatina: IFES. 102 p.

KIM, W. (1990). Object-Oriented Databases: Definition and Research Directions, IEEE Trans, Knowl. Data Eng. 2ª ed, no. 3, pp. 337 – 341.

LAUDON, K.C.; LAUDON, J.P.(2011). Sistemas de Informações Gerenciais. São Paulo: Pearson Prentice Hall. 114 p.

MACHADO, V. P. (2019). Desenvolvimento para web. Teresina: EDUFPI, 149 p.

MALTZ, L. (2005). Portals: A Personal Door to the Information Enterprises. Educause Quarterly.

16. MANZANO, José Augusto N. G. (2016). Algoritmos: lógica para desenvolvimento de programação de computadores. 28ª ed, São Paulo: É rica.

MOLKA, Karten; CASALE, Ciuliano. (2015). Experiments or Simulation? A Characterization of Evaluation Methods for In-Memory Databases. Acm Sigmod Record: South Kensington, v.44, n. 0, p.1-9.

MORAIS, Carlos Tadeu Queiroz; LIMA, Valdeni de Lima de & FRANCO, Sérgio R. K. (2012). Conceitos sobre Internet e Web – Porto Alegre: Editora da UFRGS, 2012. 112 p.

REZENDE, Paulo Eduardo Lima. Documentação de Projetos Web-DER. Disponível em: <http://imasters.com.br/artigo/8568/banco-dedados/documentacao-de-projetos-web-der/>. Acesso em 23 de janeiro de 2023.

RIBEIRO, Pedro. (2012), WikiCiência, 3(06);0623.

ROMÁN, L. L. (2008). Metodologia de La Programación Orientada aObjetos. México: Alfaomega, p. 437.

SEBESTA, R.W. (2011). Conceitos de linguagens de programação. 9ª ed. Porto Alegre: Bookman.

SHELDON, T.(1994). Encyclopedia of Networking. Osborne. McGraw – Hill.

SILVA, F.S. (2017). Word 2016: básico: para pessoas com deficiência visual: Educação Profissional. InfoServer – Osasco: Fundação Bradesco.

STRAWN, George; STRAWN, Candace. (2016). Relational Databases: Codd, Stonebraker, and Ellison. It Professional. [s.l.], v 18, n. 2, p.63-65.

STRAUSS. (2002). Web Portals and Higher Educations. Tecnologies to Make IT Personal.

TEOREY, Tobey J. (2014). Projeto e modelagem de banco de dados. [et al.]; tradução Daniel Vieira. – 2. ed. – Rio de Janeiro: Elsevier.

WADE, B; CHAMBERLIN, D. (2012). IBM Relational Database Systems: The Early Years. IEEE Annals of the History of Computing. 34ª ed., n .4, pp. 38 - 48.

XAVIER, Gley, Fabiano Cardoso. (2005). Lógica de programação 9º ed. São Paulo: Editora Senac São Paulo.

StarUML. Disponível em: <https://staruml.io/>. Acesso em 23 de janeiro de 2023.

VISUAL STUDIO. Disponível em: <https://visualstudio.microsoft.com/>. Acesso em 23 de janeiro de 2023.