



**INSTITUTO POLITÉCNICO DOM DAMIÃO FRANKLIN – Nº 8028**  
**COORDENAÇÃO DO CURSO DE INFORMÁTICA**  
**CURSO TÉCNICO DE INFORMÁTICA**

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA  
OPTIMIZAR A AQUISIÇÃO DE MEDICAMENTOS.**

**Caso de Estudo (Fármacias e Depósitos)**

**Elaborado por:**

Armando Pascoal Aquiles António.

Emanuel de Jesus Miguel António.

Edmar Miguel Vieira.

João Mendes Cabonde.

**Orientado por:** prof. Luís Domingos Marques.

Relatório da Prova de Aptidão Profissional  
(PAP) apresentado ao Instituto Dom  
Damião Franklin (IPDDF) para obtenção  
do grau de Técnico Médio de Informática.

**LUANDA, 2024**



**INSTITUTO POLITÉCNICO DOM DAMIÃO FRANKLIN N.º - “8028”**

Armando Pascoal Aquiles António.

Emanuel de Jesus Miguel António.

Edmar Miguel Vieira.

João Mendes Cabonde.

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA  
OPTIMIZAR A AQUISIÇÃO DE MEDICAMENTOS.**

**Caso de Estudo (Fármacias e Depósitos)**

**LUANDA, 2024**

## DEDICATÓRIA

*Dedicamos este trabalho aos nossos pais e familiares, que sempre estiveram ao nosso lado, que sempre cuidaram de nós, lutaram e se sacrificaram para que hoje estivéssemos aqui, no dia mais importante das nossas vidas.*

## **AGRADECIMENTO**

Agradecemos primeiramente à Deus todo poderoso, por ter nos dado a chance de ter chegado até aqui e de ter nos dado a chance de terminar este trabalho.

Agradecemos a instituição por todo o suporte dado e pelos anos de acolhimento. Agradecemos também aos nossos pais que nos incentivaram e apoiaram em todos os sentidos durante todos os anos sem parar.

Aos nossos professores e colegas que nos apoiaram e nos ajudaram a seguir com os nossos sonhos.

O nosso muito obrigado à todas às pessoas que contribuíram direta ou indiretamente para que este dia chegasse.

## RESUMO

A nossa aplicação web que tem como objectivo principal otimizar o processo da aquisição de medicamentos nos depósitos de medicamentos em Angola. Em Angola, o processo de aquisição de medicamentos por parte das farmácias enfrenta desafios significativos. Frequentemente, esse processo exige um esforço físico considerável por parte dos gestores devido a fatores como a variação repentina de preços, a disponibilidade limitada ou inexistente de certos medicamentos, e a busca incansável por depósitos que possam atender às necessidades específicas das farmácias. O método atual de aquisição é, muitas vezes, demorado e ineficiente, uma vez que os responsáveis são obrigados a se deslocar de depósito em depósito até encontrar o medicamento desejado. Para resolver essa questão, foi desenvolvida uma aplicação web para facilitar o acesso ao sistema de forma rápida e eficiente. Construímos uma aplicação totalmente responsiva e intuitiva, utilizando um conjunto de tecnologias focadas no desenvolvimento de aplicações web eficientes e seguras. A aplicação foi desenvolvida usando o framework Reactjs. O back-end foi implementado com Node.js, utilizando JavaScript com tipagem estática graças ao TypeScript. O banco de dados escolhido é um sistema relacional baseado em SQL, e o sistema também faz uso de APIs REST e uma API de geolocalização para fornecer as funcionalidades necessárias.

**Palavras-chaves:** Aplicação web; ReactJs; JavaScript; API.

## **ABSTRACT**

Our web application that has as main objective to optimize the process of purchase of medicines in medicine warehouses in Angola. In Angola, the process of purchasing medicines by pharmacies faces significant challenges. This process often requires considerable physical effort on the part of managers due to factors such as sudden price variation, limited or non-existent availability of certain medicines, and the relentless search for deposits that can meet the specific needs of pharmacies. The current method of acquisition is often time-consuming and inefficient, since those responsible are required to move from deposit to deposit until they find the desired drug. To solve this issue, a web application was developed to facilitate access to the system quickly and efficiently. We built a fully responsive and intuitive application, using a set of technologies focused on the development of efficient and secure web applications. The application was developed using the Reactjs framework. The back-end was implemented with Node.js, using JavaScript with static typing thanks to TypeScript. The chosen database is a SQL-based relational system, and the system also makes use of REST APIs and a geolocation API to provide the necessary functionalities.

**Keywords:** Web application; ReactJs; JavaScript; API.

## **LISTA DE ABREVIACÕES E SIGLAS**

**API** – Application Programming Interface (Interface de programação de aplicação);

**CSS** – Cascading Style Sheets (Folhas de estilos em cascata);

**DER** – Diagrama Entidade Relacionamento;

**HTML** – Hypertext Markup Language (Linguagem de marcação de hipertexto);

**JS** – JavaScript;

**MER** – Modelo Entidade Relacionamento;

**ORM** – Object Relational Mapper (Mapeador relacional de objetos);

**REST** – Representational State Transfer (Transferência de Estado Representacional);

**SQL** – Structured Query Language (Linguagem de Consulta Estruturada);

**SGBD** – Sistema Gerenciador de Base de Dados;

**UX** – User Experience (Experiência de usuário);

**UI** – User Interface (Interface de usuário);

**VS Code** – Visual Studio Code;

## LISTA DE FIGURAS

Figura 1. NodeJs (fonte: Google) .....	15
Figura 2. Arquitectura de uma API Rest (fonte: Google).....	18
Figura 3. ReactJs (fonte: Google).....	19
Figura 4. Express JS (fonte: Medium).....	20
Figura 5. Diagrama de casos de uso (fonte: Autor).....	24
Figura 6. Diagrama de actividade (fonte: Autor). ....	25
Figura 7. Diagrama de sequência (fonte: Autor). ....	25
Figura 8. Diagrama de Entidade Relacional (fonte: Autor).....	26
Figura 9. Modelo Entidade Relacionamento (Fonte: Autor).....	27
Figura 10. Landing Page da aplicação (fonte: Autor) .....	29
Figura 11. Telas para criação de contas (fonte: Autor) .....	29
Figura 12. Tela de login (fonte: Autor) .....	30
Figura 13. Tela de cadastro de usuários (fonte: Autor) .....	30
Figura 14. Tela dos depositos encontrados (fonte: Autor) .....	31
Figura 15. Tela da aquisição de medicamentos. (fonte: Autor).....	31
Figura 16. Código fonte da inicialização do Servidor da aplicação com ExpressJS (fonte: Autor).....	32



## ÍNDICE

DEDICATÓRIA .....	I
AGRADECIMENTO .....	II
RESUMO.....	III
ABSTRACT .....	IV
LISTA DE ABREVIACÕES E SIGLAS.....	V
LISTA DE FIGURAS.....	VI
INTRODUÇÃO .....	11
Contextualização do problema .....	11
Formulação do Problema e das Questões de Investigação .....	12
Objectivo Geral.....	13
Objectivos Específicos .....	13
Justificativa .....	13
CAPÍTULO I: FUNDAMENTAÇÃO TEÓRICA.....	14
1.1 Aplicação web .....	14
1.2. Linguagem de Programação.....	14
1.2.1. Lado do Cliente.....	14
1.2.1.1. Javascript .....	14
1.2.1.2 TypeScript .....	15
1.2.2. Lado do Servidor .....	15
1.2.2.1 Node.js .....	15
1.2.2.2 Linguagem SQL.....	16

<b>1.3. Base de Dados .....</b>	<b>16</b>
<b>1.3.1. SGBD (Sistema Gerenciador de Base de Dados) .....</b>	<b>16</b>
<b>1.3.1.1. MySql.....</b>	<b>16</b>
<b>1.3.2. MER (Modelo Entidade Relacionamento) .....</b>	<b>17</b>
<b>1.3.3. DER (Diagrama Entidade Relacionamento).....</b>	<b>17</b>
<b>1.4 API (application programming interface).....</b>	<b>17</b>
<b>1.4.1 API Rest.....</b>	<b>17</b>
<b>1.4.1.1 API de geolocalização .....</b>	<b>18</b>
<b>1.4.2 ORM .....</b>	<b>18</b>
<b>1.5.5 Architectura Client-Server.....</b>	<b>18</b>
<b>1.5 Ferramentas e Tecnologias de desenvolvimento.....</b>	<b>19</b>
<b>1.5.1 ReactJs.....</b>	<b>19</b>
<b>1.5.2 Express Js .....</b>	<b>19</b>
<b>1.5.3 Visual Studio Code .....</b>	<b>20</b>
<b>1.5.4 Insomnia .....</b>	<b>20</b>
<b>1.5.5 Xampp .....</b>	<b>20</b>
<b>1.5.6 Figma .....</b>	<b>20</b>
<b>CAPITULO II – METODOLOGIA DE ESTUDO .....</b>	<b>21</b>
<b>2.1 Metodologia de Investigação Científica .....</b>	<b>21</b>
<b>2.1.1 Método Quantitativo .....</b>	<b>21</b>
<b>2.1.2 Método Quantitativo .....</b>	<b>21</b>
<b>2.2. Metodologia de Desenvolvimento.....</b>	<b>22</b>

2.2.1. Modelo Cascata.....	22
2.3 Requisitos funcionais.....	22
2.4 Requisitos não funcionais.....	22
2.5. Ferramentas Utilizadas .....	23
2.7. Diagrama de Actividades .....	25
2.8. Diagrama de Sequência.....	25
2.9. Modelo Entidade Relacionamento .....	26
2.10 Modelo Entidade Relacionamento .....	27
<b>CAPÍTULO III – APRESENTAÇÃO, ANÁLISE E DISCUSSÃO DOS RESULTADOS OBTIDOS.....</b>	<b>28</b>
3.1 Instalação e Configuração do Ambiente de Desenvolvimento .....	28
3.1.1 Vs Code.....	28
3.1.2 Apache .....	28
3.1.3 Node JS .....	28
3.2 Uma breve apresentação da aplicação.....	29
3.3 Aquisição de medicamentos .....	31
3.4 Servidor para a API Rest.....	32
3.5. Análise e Discussão dos Resultados Obtidos .....	32
3.5.1 Eficiência do Software da Aplicação.....	33
3.5.2 Eficiência da Aplicação .....	33
3.6 CONCLUSÕES.....	34
3.7 RECOMENDAÇÕES.....	34

<b>REFERÊNCIAS .....</b>	<b>35</b>
--------------------------	-----------

## **INTRODUÇÃO**

Nos dias de hoje, a necessidade de automatizar, otimizar e inovar processos tornou-se cada vez mais evidente. Aquilo que, no passado, estava a metros ou quilômetros de distância, agora está a um clique. Isso reduziu significativamente o esforço físico, o tempo de resposta e os recursos necessários para realizar tarefas. Com este fenômeno, surge a urgência de desenvolver soluções cada vez mais inovadoras que acompanhem a globalização e facilitem a vida das pessoas. À medida que a tecnologia avança, torna-se cada vez mais claro que processos manuais e desgastantes estão perdendo espaço para soluções digitais que permitem maior agilidade e precisão. No setor de saúde, em particular, a eficiência no gerenciamento de estoques, a rapidez na obtenção de informações e a otimização dos processos de compra são cruciais para garantir o abastecimento contínuo e o atendimento de qualidade à população. É nesse contexto que a uma aplicação web para otimizar a aquisição de medicamentos em depósitos se insere. Esta solução vem responder a uma necessidade premente de tornar o processo mais rápido, prático e eficiente, garantindo que farmácias tenham acesso aos medicamentos de forma ágil, minimizando os problemas de indisponibilidade e variação de preços, e proporcionando uma melhor gestão dos recursos disponíveis.

### **Contextualização do problema**

Em Angola, especialmente na província de Luanda, o processo de aquisição de medicamentos pelas farmácias tem se tornado um desafio crescente. As farmácias enfrentam inúmeras dificuldades, incluindo a variação repentina dos preços, a indisponibilidade ou redução da quantidade dos medicamentos desejados e outros fatores que complicam o processo de compra. Diante dessas circunstâncias, os responsáveis pelas farmácias são frequentemente obrigados a buscar de depósito em depósito até encontrarem o fármaco necessário, o que demanda um esforço físico significativo e resulta em uma perda de tempo considerável. Esses obstáculos impactam diretamente a eficiência e a agilidade na gestão de estoque, dificultando o atendimento adequado à demanda das farmácias e, consequentemente, dos pacientes.

## **Formulação do Problema e das Questões de Investigação**

De que maneira uma aplicação web pode otimizar o processo de aquisição de medicamentos por parte das farmácias?

A aplicação baseando-se na sua geolocalização actual fornecerá dados atualizados sobre a disponibilidade de medicamentos em diferentes depósitos de medicamentos. Isso permitirá que as farmácias consultem rapidamente quais fármacos estão disponíveis e seus preços, eliminando a necessidade de buscas físicas sem sucesso em múltiplos locais.

## **Objectivo Geral**

- Desenvolver uma aplicação web para otimizar o processo de aquisição de medicamentos nos depósitos de medicamentos.

## **Objectivos Específicos**

1. Implementar um sistema eficiente de aquisição baseando-se na geolocalização actual do usuário;
2. Implementar uma interface de usuário intuitiva e responsiva para garantir o fácil uso da aplicação;
3. Utilizar medidas de segurança robusta;

## **Justificativa**

A nossa aplicação surge da necessidade de melhorar o processo de aquisição de medicamentos nos depósitos por parte das farmácias em Angola. A partir dos dados coletados, chegamos à conclusão de que, se existir uma plataforma web, fatores como exaustão física, perda de tempo e ineficiência na aquisição de medicamentos pode ser significativamente reduzida.

## **CAPÍTULO I: FUNDAMENTAÇÃO TEÓRICA**

Neste ponto, será apresentado, explicitamente, todos os processos teóricos, sendo eles um conjunto de conhecimentos recolhidos, previamente, antes da concepção laboral do projecto, isto é, toda informação necessária, para a construção da aplicação.

### **1.1 Aplicação web**

Uma aplicação web é um software que pode ser acessado e executado através de um navegador. Ou seja, diferente dos programas tradicionais que precisam ser instalados para serem utilizados, a aplicação web roda diretamente em navegadores, como Firefox, Google Chrome e Safari. Dessa forma, mesmo possuindo todos os requisitos de uma aplicação, com funcionalidades completas, a aplicação web caracteriza-se pelo fato de ser projetada para funcionar de forma nativa na internet (King Host, 2024).

### **1.2. Linguagem de Programação**

Linguagem de Programação é um conjunto estruturado de instruções usadas na construção de softwares possibilitando uma comunicação com o computador. As linguagens de programação podem ser classificadas em níveis de linguagens, sendo que os níveis mais baixos são mais próximas da linguagem interpretada pelo processador e mais distante das linguagens naturais. (Willrich, 2020).

#### **1.2.1. Lado do Cliente**

##### **1.2.1.1. Javascript**

JavaScript (frequentemente abreviado como JS) é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da WorldWide Web. JavaScript permite páginas da Web interactivas e, portanto, é uma parte essencial dos aplicativos da Web. A grande maioria dos sites usa, e todos os principais navegadores têm um mecanismo JavaScript dedicado para executá-lo. (Jorge Borges, 2022). É actualmente a principal linguagem para programação client-side em navegadores Web. É também bastante utilizada do lado do servidor através de ambientes como o node.js. (Jorge Borges, 2022).



### 1.2.1.2 TypeScript

TypeScript (abreviado como TS ) é uma linguagem de programação de alto nível, livre e de código aberto, desenvolvida pela Microsoft que adiciona tipagem estática com anotações de tipo opcionais ao JavaScript . Ele é projetado para o desenvolvimento de grandes aplicativos e transpila para JavaScript. (Wikipédia, s.d).

### 1.2.2. Lado do Servidor

O servidor é a parte remota, que será o software no computador responsável pelo tratamento das informações enviadas.

#### 1.2.2.1 Node.js

O NodeJs é uma plataforma que permite os desenvolvedores usem JavaScript para escrever ferramentas de linha de comando e para scripts do lado do servidor. A capacidade de executar código JavaScript no servidor é frequentemente usada para gerar conteúdo dinâmico de página da web antes que a página seja enviada ao navegador da web do usuário. Consequentemente, o Node.js representa um paradigma de "JavaScript em todos os lugares", unificando o desenvolvimento de aplicativos da web em torno de uma única linguagem de programação. (Wikipédia, s.d).



Figura 1. NodeJs (fonte: Wikipédia.com)

### **1.2.2.2 Linguagem SQL**

Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional). Muitas das características originais do SQL foram inspiradas na álgebra relacional. (André, O Que é MySQL – Guia para Iniciantes, 2022).

## **1.3. Base de Dados**

Bancos de dados (português brasileiro) ou bases de dados (português europeu) são conjuntos de arquivos relacionados entre si com registos sobre pessoas, lugares ou coisas. São colecções organizadas de dados que se relacionam de forma a criar algum sentido (informação) e dar mais eficiência durante uma pesquisa ou estudo científico. São de vital importância para empresas e, há mais de duas décadas, se tornaram a principal peça dos sistemas de informação e segurança. Normalmente existem por vários anos sem alterações em sua estrutura sistemática. (Lame, 2021).

### **1.3.1. SGBD (Sistema Gerenciador de Base de Dados)**

Um Sistema de Gerenciamento de Banco de Dados (SGBD) do inglês Data Base Management System (DBMS) é o sistema de software responsável pelo gerenciamento de um ou mais bancos de dados. Seu principal objectivo é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, a persistência, a manipulação e a organização dos dados. O SGBD disponibiliza uma interface para que seus clientes possam incluir, alterar ou consultar dados previamente armazenados. (Wattson, 2021).

#### **1.3.1.1. MySql**

MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. É actualmente um dos sistemas de gerenciamento de bancos de dados mais populares da Oracle Corporation, com mais de 10 milhões de instalações pelo mundo. (André, O Que é MySQL – Guia para Iniciantes, 2021)

### **1.3.2. MER (Modelo Entidade Relacionamento)**

De forma simples, podemos dizer que o Modelo de Entidade e Relacionamento (MER) são diagramas utilizados para projectar Bancos de Dados Relacionais, utilizando como base a relação de objectos reais, e sendo representado por meio de entidades e relacionamentos. (Cortes, 2021)

### **1.3.3. DER (Diagrama Entidade Relacionamento)**

Ao falar de MER, sempre nos deparamos com o Diagrama de Entidade Relacionamento (DER). Na verdade, o DER nada mais é que a representação gráfica do que foi escrito no MER. (Cortes, 2021)

## **1.4 API (application programming interface)**

Uma API, ou interface de programação de aplicativos, é um conjunto de regras ou protocolos que permite que os aplicativos de software se comuniquem entre si para trocar dados, recursos e funcionalidades. APIs simplificam e aceleram o desenvolvimento de aplicativos e software, permitindo que os desenvolvedores integrem dados, serviços e recursos de outros aplicativos, em vez de desenvolvê-los do zero. (Goodwin, 2024)

### **1.4.1 API Rest**

Uma API REST (também chamada de API RESTful ou web API RESTful) é uma interface de programação de aplicativos (API) que fornece uma maneira flexível de integrar aplicações e conectar componentes em arquiteturas de microsserviços, baseado no estilo arquitetônico REST (transferência de estado representacional) (IBM, 2024)

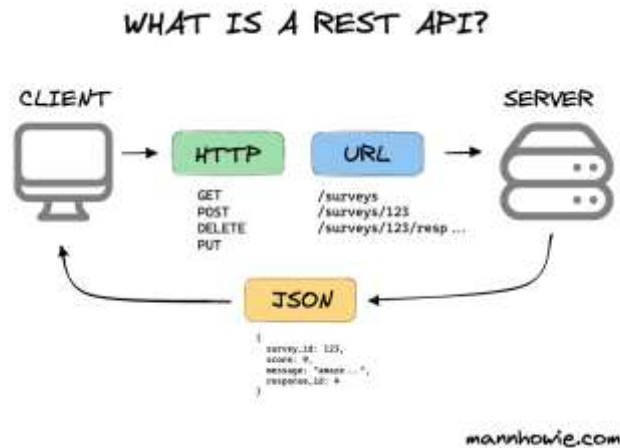


Figura 2. Arquitetura de uma API Rest (fonte: mannhowie.com)

#### 1.4.1.1 API de geolocalização

Uma API de geolocalização é um serviço que aceita uma solicitação HTTPS com uma torre de celular e pontos de acesso Wi-Fi que um cliente móvel pode detectar. Ela retorna coordenadas de latitude/longitude e um raio indicando a precisão do para cada entrada válida. (Dev, 2023)

#### 1.4.2 ORM

Um ORM , ou Object Relational Mapper, é um pedaço de software projetado para traduzir entre as representações de dados usadas por bancos de dados. Da perspectiva de um desenvolvedor, um ORM permite que você trabalhe com dados apoiados por banco de dados usando as mesmas estruturas e mecanismos orientados a objetos que você usaria para qualquer tipo de dado interno. ( Ellingwood, 2024).

#### 1.5.5 Arquitetura Client-Server

A tecnologia cliente/servidor é uma arquitetura na qual o processamento da informação é dividido em módulos ou processos distintos. Um processo é responsável pela manutenção da informação (servidores) e outros responsáveis pela obtenção dos dados (os clientes). (Grupo 4,2012) Os processos cliente enviam pedidos para o processo servidor, e este por sua vez processa e envia os resultados dos pedidos. (Grupo 4, 2012)

## 1.5 Ferramentas e Tecnologias de desenvolvimento

### 1.5.1 ReactJs

O React é uma biblioteca JavaScript front-end gratuita e de código aberto para construção de interfaces de usuário baseadas em componentes do Facebook Inc. É mantido pela Meta (anteriormente Facebook) e uma comunidade de desenvolvedores individuais e empresas. O React pode ser usado para desenvolver aplicativos de página única, móveis ou renderizados por servidor com estruturas como Next.js. (Wikipedia, s.d).

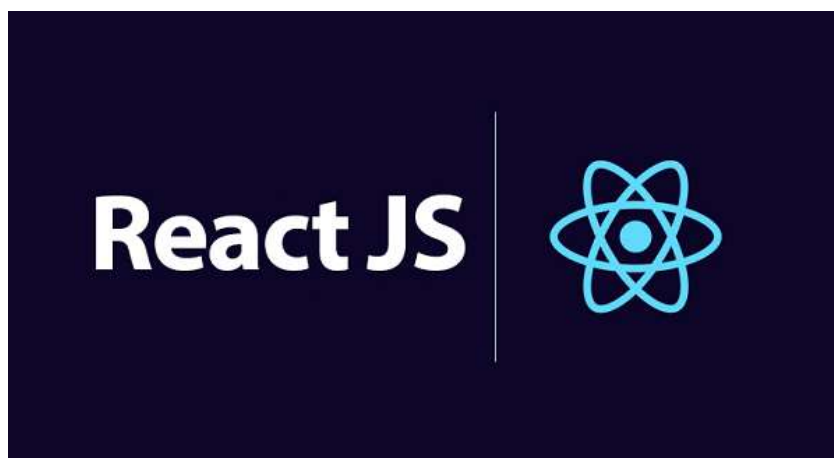


Figura 3. ReactJs (fonte: leetcode.com).

### 1.5.2 Express Js

O Express.js, ou simplesmente Express, é uma estrutura de aplicativo web back-end para construir APIs RESTful com Node.js, lançada como software livre e de código aberto sob a Licença MIT. Ele foi projetado para construir aplicativos web e APIs. (Wikipédia, s.d). Express é o componente de back-end de pilhas de desenvolvimento populares como a pilha MEAN, MERN ou MEVN, junto com o software de banco de dados MongoDB e uma estrutura ou biblioteca de front-end JavaScript.



Figura 4. Express JS (fonte: Medium.com).

### **1.5.3 Visual Studio Code**

O Visual Studio Code, é um editor de código desenvolvido e fornecido pela equipa da Microsoft com uma distribuição de licença gratuita. Foi usado, aqui, como o principal e único programa para a edição do código todo usados no aplicativo e no servidor.

### **1.5.4 Insomnia**

Insomnia é um aplicativo de desktop de código aberto que elimina a dor de interagir e projetar, depurar e testar APIs. Insomnia combina uma interface fácil de usar com funcionalidades avançadas como auxiliares de autenticação, geração de código e variáveis de ambiente.

### **1.5.5 Xampp**

XAMPP é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte às linguagens PHP e Perl. É um software livre, que consiste principalmente na base de dados MySQL. (Didática Tech, 2021).

### **1.5.6 Figma**

Figma é um aplicativo web colaborativo para design de interface , com recursos offline adicionais habilitados por aplicativos de desktop para macOS e Windows . O conjunto de recursos do Figma se concentra no design de interface e experiência do usuário , com ênfase na colaboração em tempo real,utilizando uma variedade de editores de gráficos vetoriais e ferramentas de prototipagem . O aplicativo móvel Figma para Android e iOS permite visualizar e interagir com protótipos Figma em tempo real em dispositivos móveis e tablets. (Wikipédia, s.d).

## **CAPITULO II – METODOLOGIA DE ESTUDO**

### **2.1 Metodologia de Investigação Científica**

Para a elaboração do projecto, foram usados métodos, que consistem em pesquisas de informações inerentes à taxa de eficiência na aquisição de medicamentos por parte das farmácias nos depósitos da província de Luanda. Metodologia de investigação científica é a ciência que estuda como se conduz cientificamente a investigação e a vida de solução sistemáticas dos problemas de investigação. (Norgueira e Isabel, 2023)

#### **2.1.1 Método Quantitativo**

O Método Qualitativo foi utilizado para explorar e compreender percepções subjectivas dos participantes acerca do uso de tecnologias emergentes no desenvolvimento de soluções de software. Este método envolveu a realização de entrevistas semiestruturadas e grupos focais, permitindo a coleta de dados ricos e detalhados sobre a experiência do usuário. A pesquisa qualitativa possibilita uma análise mais profunda dos comportamentos e das opiniões, fornecendo uma compreensão mais ampla dos contextos em que os fenômenos ocorrem. (Minayo, 2010).

#### **2.1.2 Método Quantitativo**

Por outro lado, o Método Quantitativo foi implementado através da coleta e análise de dados numéricos relacionados ao uso de ferramentas tecnológicas específicas no mercado. Foram aplicados questionários estruturados a uma amostra significativa de usuários para obter insights sobre a aceitação de novas tecnologias no desenvolvimento de software. Como apontado por Lakatos e Marconi (2017), a abordagem quantitativa permite uma análise objetiva dos dados e a obtenção de resultados generalizáveis, especialmente quando utilizada combinação com análise estatística.

## **2.2. Metodologia de Desenvolvimento**

As metodologias de desenvolvimento de software consistem, basicamente, no conjunto de abordagens que podem ser utilizadas para a criação de sistemas de processamento de dados. O sucesso de qualquer projeto voltado à elaboração de software depende diretamente da escolha da metodologia mais adequada, sendo assim, foi escolhido o modelo cascata no processo de desenvolvimento.

### **2.2.1. Modelo Cascata**

A abordagem em Cascata visa a adoção de controles e processos lineares muito rigorosos. Nela, uma nova etapa só pode ser começada quando as anteriores forem concluídas, para que as ações sejam orientadas gradualmente até as próximas responsabilidades previstas. Funcional, gradual e analítico, esse método também é considerado simples, já que impõe uma sequência restrita e bem dimensionada de tarefas.

## **2.3 Requisitos funcionais**

1. Disponibilizar a criação de conta para farmácias e depósitos ;
2. Possibilitar o login das entidades(fármacia e depósito);
3. Possibilitar que os depósitos cadastrem os fármacos disponíveis de acordo com a sua categoria para aquisição, quantidade disponível, e preço unitário.
4. Possibilitar a solicitação e aquisição de fármacos nos depósitos de medicamentos mais próximos consoante a geolocalização da farmácia.
5. Possibilitar a visualização de depósitos mais próximos da geolocalização da farmácia.

## **2.4 Requisitos não funcionais**

1. Otimizar consultas SQL e minimizar o tempo de resposta da API utilizando caching para reduzir o tempo de resposta.
2. O sistema deve ser seguro e utilizar um gateway de pagamentos.
3. O sistema deve ser resiliente a falhas de conexão e manter uma interface amigável.
4. Interface flexível, com padrões de UI e UX



## 2.5. Ferramentas Utilizadas

Aqui, é descrito, as ferramentas usadas na construção do software, desde a customização da interface, utilização de frameworks, bem como ferramentas de modelagem, sendo elas classificadas abaixo:

- IDE's (Ambiente de Desenvolvimento Integrado)
  - VS Code (Editor de código padrão)
- Software de construção de UI (Interface de Usuário)
  - Figma
- Framework's
  - Express JS
  - React js
- XAMPP (Pacote de servidores)
  - Apache Server
  - Mysql Server
- Softwares de Modelação
  - StarUML
  - Mysql Workbench
- Linguagens
  - JavaScript
  - TypeScript
  - SQL
  - HTML
  - CSS
- ORM
  - Prisma

## 2.6. Diagrama de Caso de Uso

Aqui, é apresentado um diagrama que demonstra quais recursos estão acessíveis para determinados actores, sendo que, existem dois tipos de actores distintos, um com funções administrativas e outro com funções mais restritas.

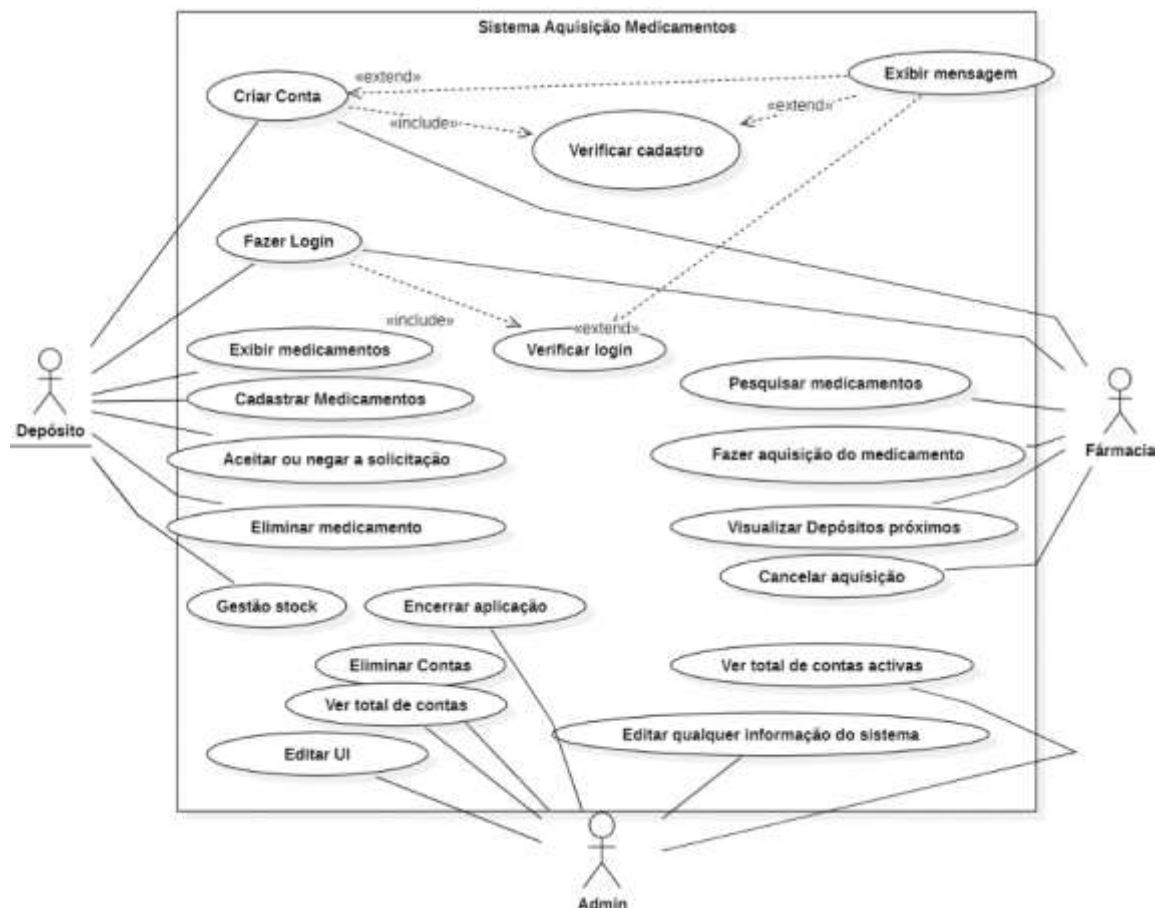


Figura 5. Diagrama de casos de uso (fonte: Autor).

## 2.7. Diagrama de Actividades

Aqui, é aqui apresentado de forma explícita, como é que a aplicação funcionará, mostrando o fluxo de funcionamento das actividades por parte de um usuário comum

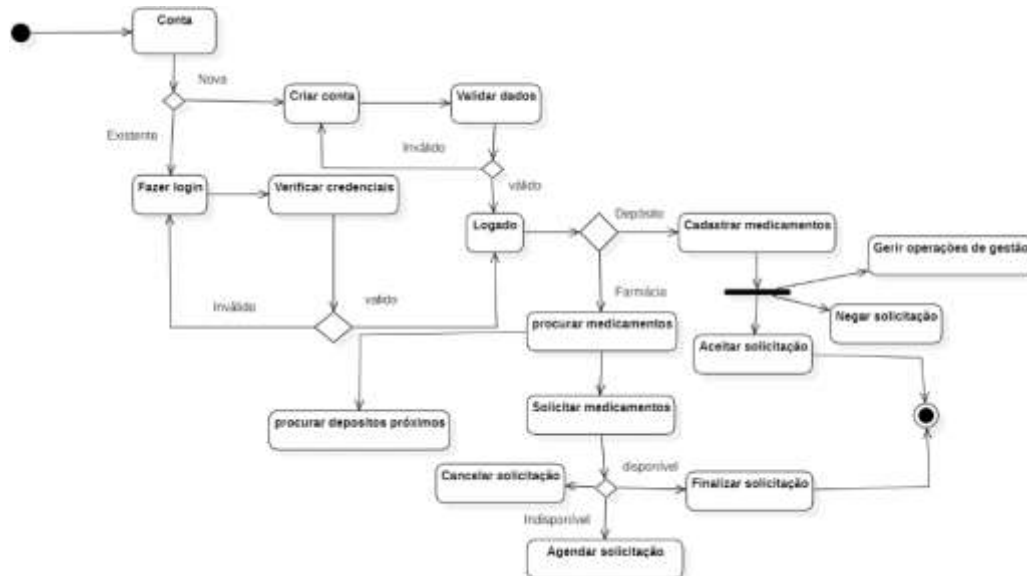


Figura 6.Diagrama de actividade (fonte: Autor).

## 2.8. Diagrama de Sequência

Durante a elaboração do projecto foram recolhidas informações complementares que ajudaram na estruturação do projecto. No caso, foi elaborado um diagrama que demonstra intuitivamente algumas sequências de funcionamento interno da aplicação..

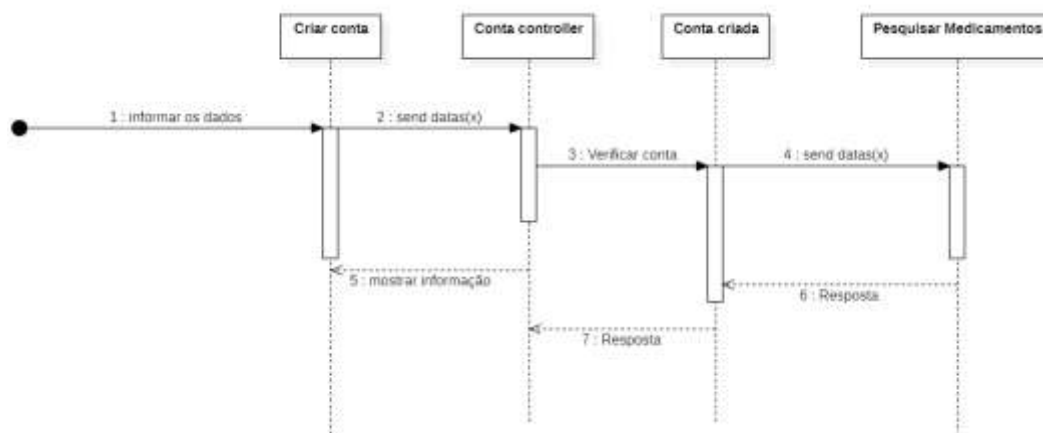


Figura 7.Diagrama de sequência (fonte: Autor).

## 2.9. Modelo Entidade Relacionamento

Aqui é apresentado um modelo lógico do banco de dados, sendo uma representação mais formal do DER.

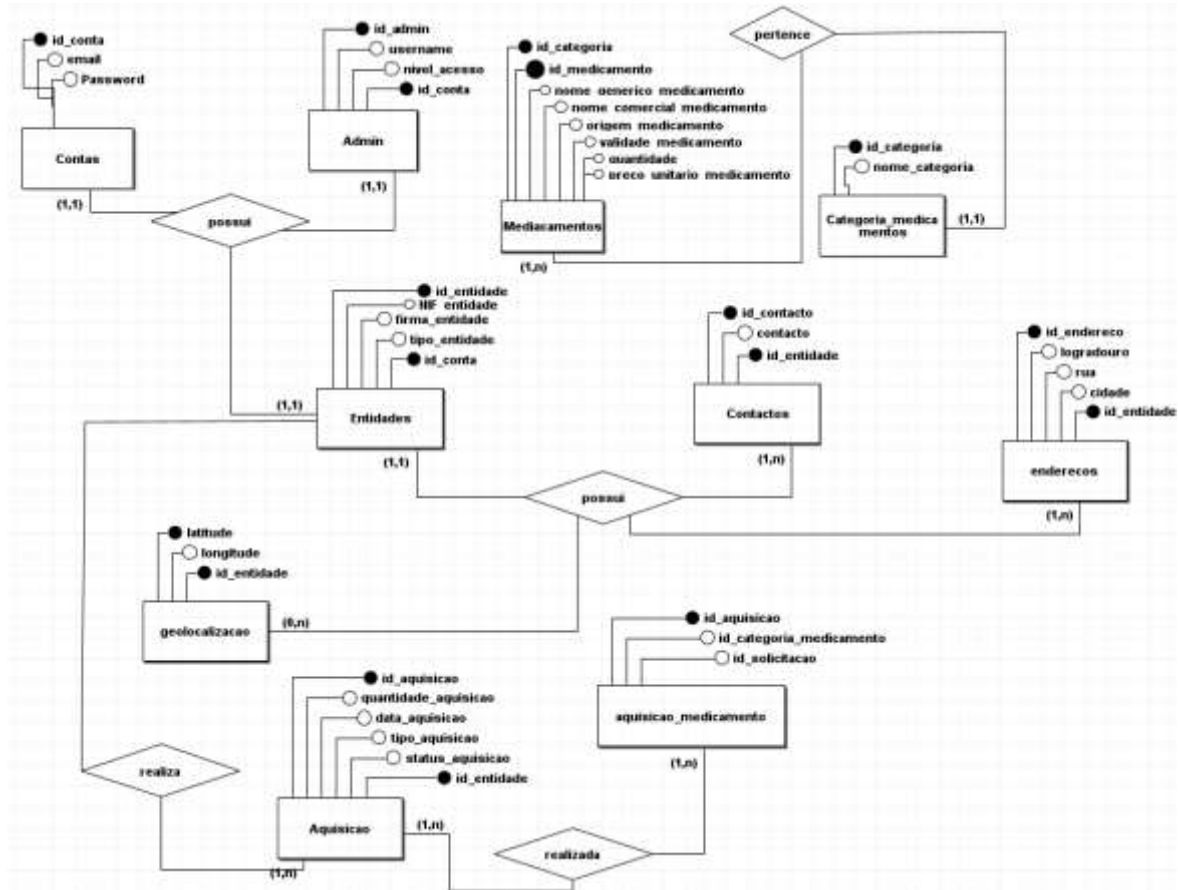


Figura 8. Diagrama de Entidade Relacional (fonte: Autor)

## 2.10 Modelo Entidade Relacionamento

Aqui é apresentado um modelo lógico do banco de dados, sendo uma representação mais formal do DER.

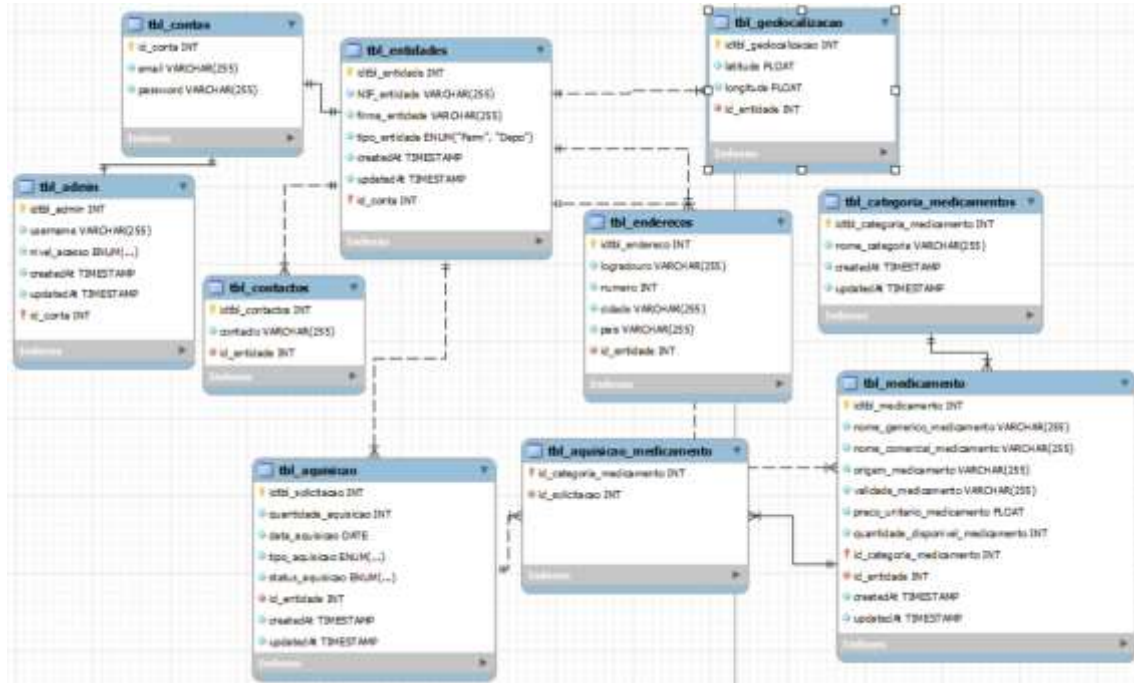


Figura 9. Modelo Entidade Relacionamento (Fonte: Autor).

## **CAPÍTULO III – APRESENTAÇÃO, ANÁLISE E DISCUSSÃO DOS RESULTADOS OBTIDOS**

Para dar uma melhor percepção do leitor, neste ponto será apresentado às ferramentas usadas para a concepção da aplicação, bem como as etapas correctas para instalação e configuração do ambiente de desenvolvimento.

### **3.1 Instalação e Configuração do Ambiente de Desenvolvimento**

Os passos, aqui listados, foram aplicados no ambiente Windows, mas serão funcionais tanto para o Linux quanto para o Mac Os, para esta aplicação precisaremos de um editor de código, nós escolhemos o VsCode, NodeJS, um pacote de servidores, recomendamos o Apache, precisaremos também de um serviço de Geolocalização.

#### **3.1.1 Vs Code**

Para este projecto, utilizamos como ferramenta de desenvolvimento o VS Code pela simplicidade e produtividade na hora de escrever os códigos da aplicação.

#### **3.1.2 Apache**

Para este projecto, utilizamos como pacote de servidores o Apache pela simplicidade.

#### **3.1.3 Node JS**

Nós precisaremos do Node Js para construir os códigos JavaScript da nossa aplicação no lado do servidor, e rodar o framework escolhido para o front-end (React JS), recomendamos usar a versão 20 do Node Js, recomendamos também o uso de gerenciador de pacotes para JavaScript, no nosso projecto decidimos usar o Yarn, pela sua eficiência e velocidade. Para todos os efeitos, devem ser usados os seguintes comandos:

- yarn install – para instalar todas às dependências da aplicação.
- yarn run dev – para iniciar os servidores da aplicação.

### 3.2 Uma breve apresentação da aplicação

Aqui, é possível visualizar a tela inicial da aplicação onde há uma breve introdução sobre os serviços oferecidos, bem como algumas telas como de aquisição, cadastro e login, em uma interface bastante simples e flexível, com recursos de UX disponíveis, para melhorar a interação usuário-máquina.



Figura 10. Landing Page da aplicação (fonte: Autor)



Figura 11. Telas para criação de contas (fonte: Autor)

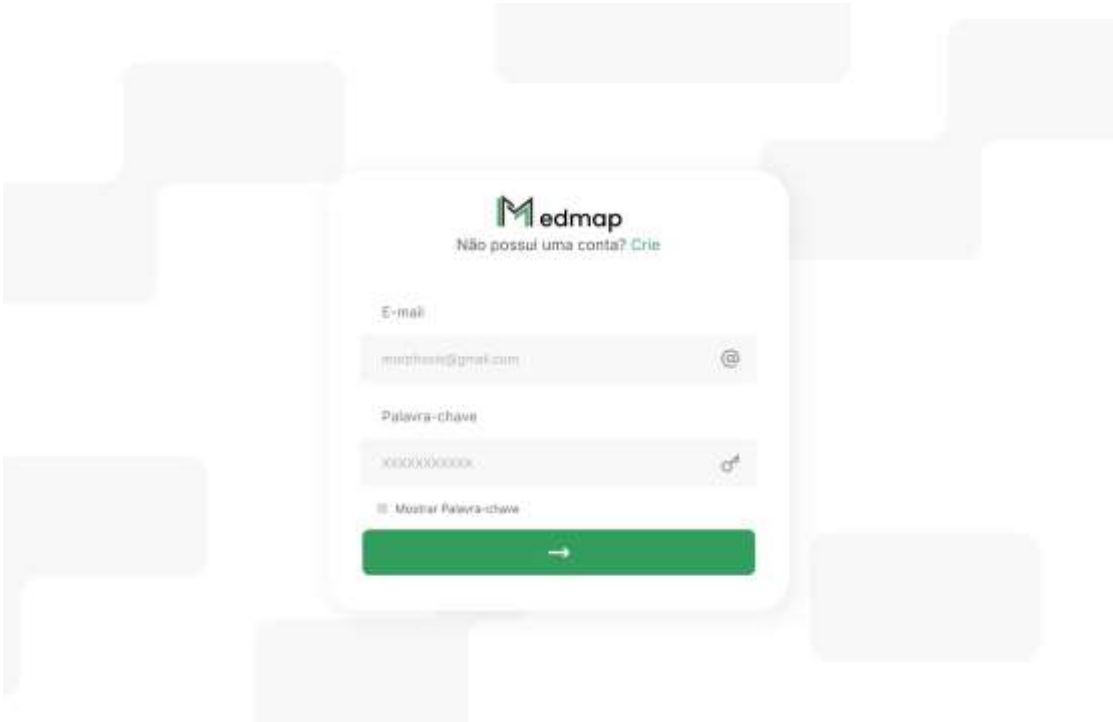


Figura 12. Tela de login (fonte: Autor)

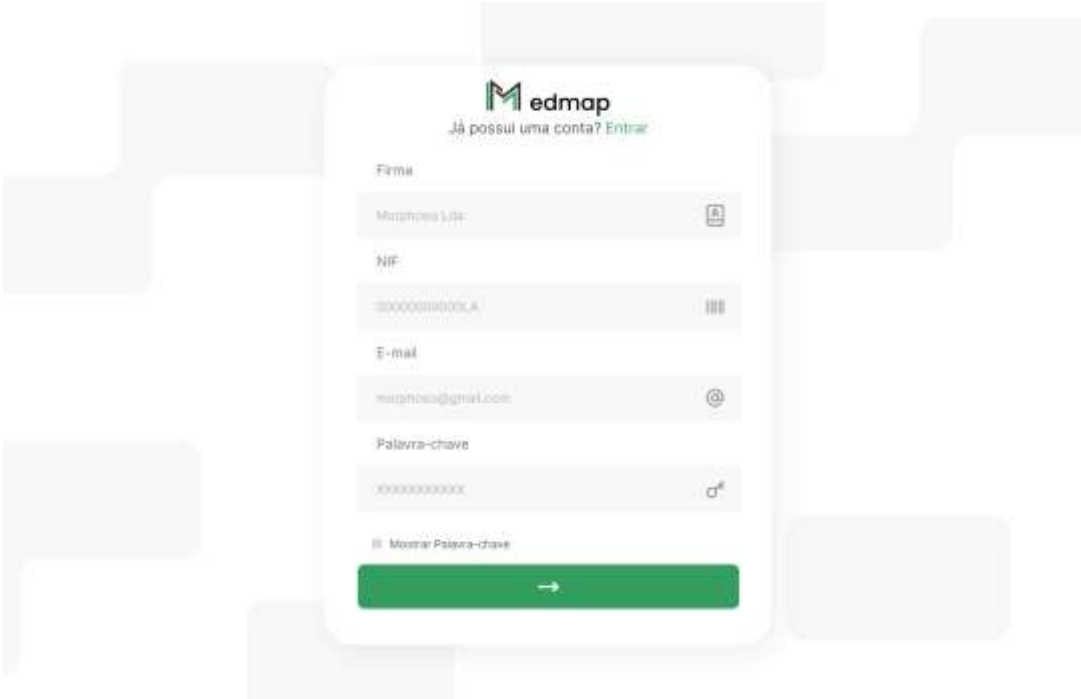


Figura 13. Tela de cadastro de usuários (fonte: Autor)



### 3.3 Aquisição de medicamentos

Para otimizar a procura de medicamentos, a nossa aplicação oferece a funcionalidade de procurar os depósitos que possuem o determinado medicamento de acordo com a geolocalização do depósito mais próximo da localização da farmácia.

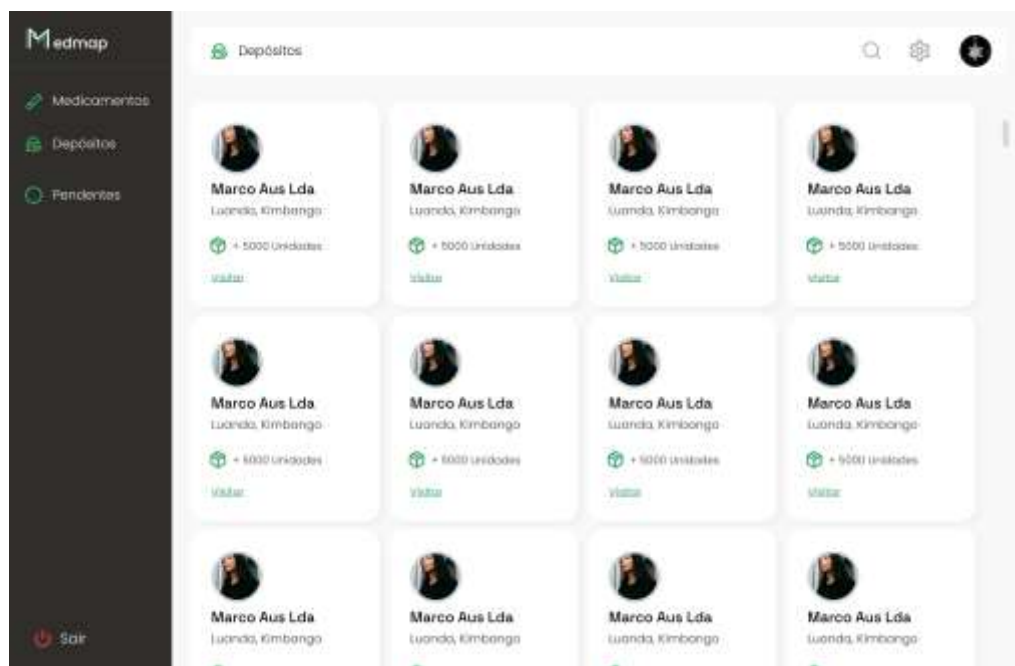


Figura 14. Tela dos depósitos encontrados (fonte: Autor)



Figura 15. Tela da aquisição de medicamentos. (fonte: Autor)

### 3.4 Servidor para a API Rest

Na arquitectura básica e funcional de todas às páginas da web, a arquitectura Client/Servidor é a predominante. Tendo isso em conta o NodeJs dá-nos a facilidade de criar servidores em diversos frameworks para que às nossas APIS possam servir os clientes.

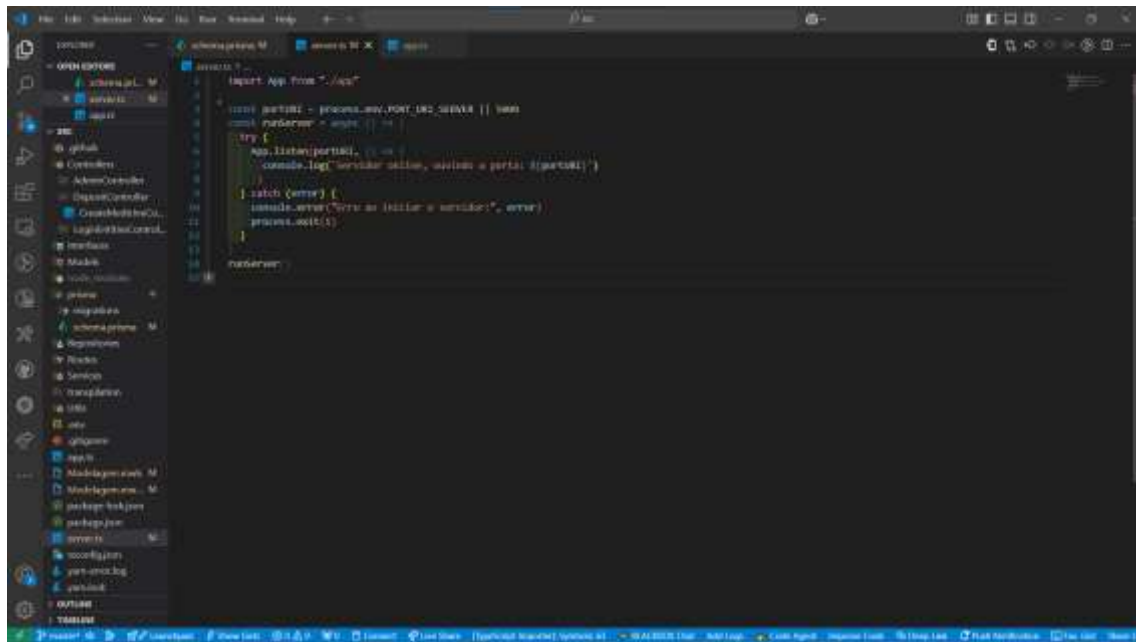


Figura 16. Código fonte da inicialização do Servidor da aplicação com ExpressJS (fonte: Autor)

### 3.5. Análise e Discussão dos Resultados Obtidos

Durante o desenvolvimento do software foram retidas diversas observações no que concerne aos seguintes pontos.

### **3.5.1 Eficiência do Software da Aplicação**

Foi desenvolvido um software para uma aplicação Web com capacidade de entregar aos usuários medicamentos pertencente à depósitos mais próximos da sua geolocalização de forma eficiente, para isso o software da aplicação usa recursos de geolocalização por meio de uma API de geolocalização, o que faz com que o nosso software optimize significativamente este mesmo processo nos próximos dias, meses e anos. Desenvolvemos o software desta aplicação com boas práticas para ter uma aplicação eficiente e menos frustrante para os usuários, o nosso software presa muito por rapidez e eficiência, uma vez que estamos desenvolvendo uma solução para o mercado farmacêutico, para isso a nossa aplicação foi hospedada a nível da Web para uma melhor performance. A escolha de desenvolver uma aplicação Web facilita o acesso a nossa aplicação pelo facto de não ser preciso instalar a aplicação localmente, sendo assim uma opção viável, uma vez que para além de não ser necessário instalar a aplicação, o usuário poderá acessar a aplicação em qualquer lugar, desde que tenha acesso a um navegador.

### **3.5.2 Eficiência da Aplicação**

Foram analisados um conjunto de relações métricas entre o que foi traçado e o que foi obtido, sendo que, teve-se como resultado um conjunto satisfatório de elementos que serão em seguida referidos.

A eficiência e desempenho esperado do software da aplicação corresponde ao que nós como equipa traçamos desde a modelagem deste mesmo software, a nossa aplicação otimiza de forma eficiente o processo de aquisição de medicamentos, desenvolvemos uma solução digital que com certeza irá mitigar este fenómeno no mercado farmacêutico.

### **3.6 CONCLUSÕES**

A nossa proposta apresentou soluções inovadoras e eficientes, destacando-se pela facilidade de uso e interação. A implementação desta aplicação na sociedade angolana tem o potencial de otimizar significativamente o processo de aquisição de medicamentos pelas farmácias nos depósitos de medicamentos em Angola. Com essa aplicação, farmácias poderão realizar a aquisição de seus medicamentos de forma instantânea, otimizada e organizada. O sistema foi projetado para descentralizar e desacoplar responsabilidades, promovendo um ambiente digital confiável e de fácil gerenciamento. Além disso, a integração de tecnologias modernas garante segurança, escalabilidade e adaptabilidade, tornando a solução viável a longo prazo. Ao longo do desenvolvimento, os objetivos traçados foram plenamente cumpridos, consolidando a proposta como uma solução viável e funcional.

Por fim, acreditamos que este projeto não apenas resolve um problema crítico, mas também serve como um modelo para futuras inovações tecnológicas voltadas para o setor da farmacêutico em Angola. Esperamos que essa proposta inspire outras iniciativas e contribua para a modernização do setor farmacêutico no país.

### **3.7 RECOMENDAÇÕES**

O projecto apresentado, foi terminado, e concluído de acordo à expectativa tida desdeo início da construção do projecto, mas como qualquer software, ele pode sim ser melhorado, sendo que, com o aumento de requisições uma melhoria seria hospedar a nossa aplicação em um serviço de hospedagem mais dedicado, outra melhora para uma melhor performance em geolocalização a aplicação poderia utilizar serviços de geolocalização mais poderoso como o Google Maps.

## REFERÊNCIAS

André, L. (30 de Março de 2021). O Que é MySQL – Guia para Iniciantes. Obtido de Weblink: <https://www.weblink.com.br/blog/o-que-e-mysql/>

Candido, C. H. (3 de Fevereiro de 2022). brModelo 3.2\_ Projeto brModelo. Obtido de sourceforce: <https://sourceforge.net/p/brmodelo/ /Projeto%20brModelo/>

Cortes, A. (23 de Julho de 2021). MER e DER\_ o que é, as principais diferenças e como usar. Obtido de Remessa online: <https://www.remessaonline.com.br/blog/mer-e-der-o-que-e-as-principais-diferencas-e-como-usar/>

Dev.(13 de Agosto de 2021). Node.js. Obtido de devmedia: <https://www.devmedia.com.br/guia/node-js/40312>

Dev.(s.d). Visão geral da API Geolocation. Obtido de :<https://developers.google.com/maps/documentation/geolocation/overview?hl=pt-br>

IBM (s.d). Oque é uma API Rest. Obtido de: <https://www.ibm.com/br-pt/topics/rest-apis>.

Jorge Borges. (6 de Janeiro de 2022). JavaScript. Obtido de Developers Plus:<https://developerplus.com.br/program/JavaScript>.

King Host. (08 de Outubro de 2024). Aplicação Web. Obtido de: <https://king.host/blog/tecnologia/aplicacao-web/>

Longen, A. (15 de Julho de 2021). Banco de Dados. Obtido: database:<https://www.ti-enxame.com/pt/database/operador-sql-no-cloud-firestore/833844445/>

Macedo, E. (2019). Diagramas UML. SP de Brasil: SP.

Node js. Obtido de: <https://en.wikipedia.org/wiki/Node.js>

Jean Claude.(21 dezembro 2022). React Js. Obitido de: <https://www.letecode.com/quest-ce-que-react-et-comment-fonctionne-t-il-reellement>

O Que é React e Como Funciona? Obtido de Hostinger Tutoriais: <https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript>.

Santos, J. (2020). Diagramas. São Paulo: Casa do Código

Wattson, E. (14 de Agosto de 2021). Sistema de gerenciamento de banco de dados.  
Obtido de Developers Plus:  
[https://developerplus.com.br/Sistema\\_de\\_gerencia\\_mento\\_de\\_banco\\_de\\_dados](https://developerplus.com.br/Sistema_de_gerencia_mento_de_banco_de_dados)

Wikipedia. (n.d.). Express.js. Obtido de: <https://en.wikipedia.org/wiki/Express.js>

Wikipedia. (n.d.). TypeScript. Obtido de: <https://en.wikipedia.org/wiki/TypeScript>

Willrich, R. (2020). Introducao\_linguagens.pdf.