

**MINISTÉRIO DA EDUCAÇÃO**  
**SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**  
**INSTITUTO FEDERAL GOIANO – CAMPUS IPORÁ**  
**CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**ALCÍDIA CRISTINA RODRIGUES OLIVEIRA**

**ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA MOBILE E WEB**  
**UTILIZANDO API PARA COMPARAÇÃO DE PREÇOS DE**  
**MEDICAMENTOS DAS FARMÁCIAS DE IPORÁ - GO**

Iporá-GO  
Novembro - 2019

**MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL GOIANO – CAMPUS IPORÁ  
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**ALCÍDIA CRISTINA RODRIGUES OLIVEIRA**

**ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA MOBILE E WEB  
UTILIZANDO API PARA COMPARAÇÃO DE PREÇOS DE  
MEDICAMENTOS DAS FARMÁCIAS DE IPORÁ - GO**

Trabalho de curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia Goiano – Campus Iporá, como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, sob orientação da Professora Ma. Luciana Recart Cardoso.

Iporá-GO  
Novembro – 2019

Sistema desenvolvido pelo ICMC/USP  
Dados Internacionais de Catalogação na Publicação (CIP)  
**Sistema Integrado de Bibliotecas - Instituto Federal Goiano**

Oliveira, Alcídia Cristina

OOL48a      Análise e Desenvolvimento de um sistema mobile e web  
utilizando API para comparação de preços de medicamentos  
das Farmácias de Iporá-GO / Alcídia Cristina  
Oliveira; orientadora Luciana                  Recart Cardoso;  
co-orientador Jefferson Teixeira Oliveira. -  
- Iporá, 2019.  
52 p.

Monografia ( em Tecnologia em Análise e  
Desenvolvimento de Sistemas) -- Instituto Federal  
Goiano, Campus Iporá,2019.

1. medicamentos. 2. sistema mobile. 3. sistema  
web. 4. API. I. Recart Cardoso, Luciana , orient.  
II. Teixeira Oliveira, Jefferson, co-orient. III.  
Título.

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO**

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

**Identificação da Produção Técnico-Científica**

- Tese  Artigo Científico  
 Dissertação  Capítulo de Livro  
 Monografia - Especialização  Livro  
 TCC - Graduação  Trabalho Apresentado em Evento  
 Produto Técnico e Educacional - Tipo:

Nome Completo do Autor: Alcídia Cristina Rodrigues Oliveira

Matrícula: 2017105210430198

Título do Trabalho: Análise e Desenvolvimento de um sistema mobile e web utilizando API para comparação de preços de medicamentos das Farmácias de Iporá-GO.

**Restrições de Acesso ao Documento**

Documento confidencial:  Não  Sim, justifique: \_\_\_\_\_

Informe a data que poderá ser disponibilizado no RIIF Goiano: 20/02/2020

O documento está sujeito a registro de patente?  Sim  Não

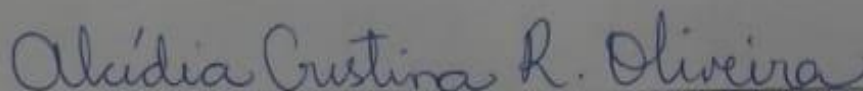
O documento pode vir a ser publicado como livro?  Sim  Não

**DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA**

O/A referido/a autor/a declara que:

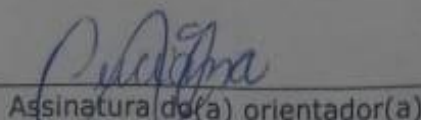
- o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Iporá, 20/02/2020.  
Local Data



Assinatura do Autor e/ou Detentor dos Direitos Autorais

Ciente e de acordo:

  
Assinatura do(a) orientador(a)



ATA DA SESSÃO DE JULGAMENTO DO TRABALHO DE CURSO  
DE ALCÍDIA CRISTINA RODRIGUES OLIVEIRA

Aos onze dias do mês de dezembro de dois mil e dezenove, às vinte horas e trinta minutos, no Laboratório de Informática III do Instituto Federal Goiano – Câmpus Iporá, reuniu-se, em sessão pública, a banca examinadora designada na forma regimental pela Coordenação do Curso para julgar o trabalho de curso intitulado “ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA MOBILE E WEB UTILIZANDO API PARA COMPARAÇÃO DE PREÇOS DE MEDICAMENTOS DAS FARMÁCIAS DE IPORÁ -GO”, apresentado pela acadêmica **Alcídia Cristina Rodrigues Oliveira** como parte dos requisitos necessários à obtenção do grau de Tecnóloga em Análise e Desenvolvimento de Sistemas. A banca examinadora foi presidida pela orientadora do trabalho de curso, a professora mestre, **Luciana Recart Cardoso**, e pelo coorientador graduado **Jefferson Teixeira Oliveira**, tendo como membros a professora especialista Livia Mancine Coelho de Campos e pelo o professor mestre, Cleon Xavier e o professor. Aberta a sessão, a acadêmica expôs seu trabalho. Em seguida, foi arguida pelos membros da banca e:

(  ) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **aprovação** da acadêmica, sem restrições.

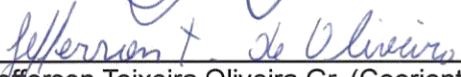
(  ) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **aprovação** da acadêmica, **condicionada a satisfazer as exigências** listadas na Folha de Modificação de Trabalho de Curso anexa à presente ata, no prazo máximo de 80 (oitenta) dias, a contar da presente data, ficando o professor orientador responsável por atestar o cumprimento dessas exigências.

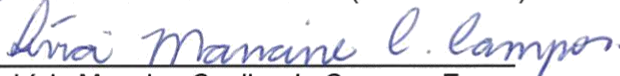
(  ) não tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **reprovação** da acadêmica.

Conforme avaliação individual de cada membro da banca, será atribuída a nota ..... 9,0 .....  
( ..... nove ..... ) para fins de registro em histórico acadêmico.

Os trabalhos foram encerrados às ..... vinte e uma ..... horas e ..... trinta ..... minutos. Nos termos do Regulamento do Trabalho de Curso do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal Goiano – Câmpus Iporá, lavrou-se a presente ata que, lida e julgada conforme, segue assinada pelos membros da banca examinadora.

  
\_\_\_\_\_  
Luciana Recart Cardoso, Ma. (Orientadora)

  
\_\_\_\_\_  
Jefferson Teixeira Oliveira Gr. (Coorientador)

  
\_\_\_\_\_  
Livia Mancine Coelho de Campos, Esp.

  
\_\_\_\_\_  
Cleon Xavier Júnior, Me.

## **AGRADECIMENTOS**

Primeiramente a Deus por ter me dado forças para desenvolver esse projeto, e por ter colocado diversas pessoas para me ajudar nesse caminho.

Ao meu querido esposo, Uender Carlos Barbosa pelo amor e carinho, mesmo nos momentos mais difíceis esteve ao meu lado, me auxiliando e dizendo diversas vezes “calma, você vai conseguir”.

A minha mãe, Márcia Rodrigues de Souza Bergland, por nunca medir esforços para ajudar, sempre dando o apoio e incentivo. A minha querida sogra, Maria Eva Barbosa, que apesar da idade nunca mediu esforços para ajudar nas tarefas do dia-a-dia para que eu pudesse ter mais tempo para estudar.

A instituição e todos os docentes que contribuíram na minha jornada acadêmica, servindo de base para meu aprendizado.

A minha orientadora Luciana Recart Cardoso e coorientador Jefferson Teixeira de Oliveira, por terem sempre me incentivado, sem esta ajuda acredito que eu não teria chegado até aqui.

Enfim, a todos que, de forma direta ou indireta contribuíram para meu crescimento e sempre estiveram presentes no decorrer da minha caminhada, ajudando em minha formação acadêmica, deixo aqui, meu mais sincero muito obrigada.

## EPÍGRAFE

*“O pessimista vê dificuldade em cada oportunidade; o otimista vê oportunidade em cada dificuldade.”*  
(Winston Churchill)

## RESUMO

Através de pesquisas verifica-se que há uma grande variação de preços de medicamentos de uma farmácia para outra dentro de uma mesma cidade. Observa-se também que o número de *smartphones* utilizados no Brasil ultrapassa a quantidade total da população. Tais constatações motivaram o desenvolvimento deste trabalho que teve como objetivo principal desenvolver um sistema *mobile* para auxiliar os usuários de fármacos a encontrar o menor preço de medicamentos nas farmácias de Iporá – GO. Os clientes das farmácias pesquisarão o menor preço de medicamentos por meio de comparação efetuada via sistema *mobile*. Por meio do sistema web, serão fornecidos pelas farmácias, os dados relevantes para quem deseja pesquisar medicamentos. A integração entre os sistemas *mobile* e web acontecerá através da API, que fará o processamento de dados no *back-end*. Espera-se que através do sistema gerado os consumidores de fármacos possam economizar em seus tratamentos de saúde de forma cômoda e ágil.

Palavras-chave: medicamentos; sistema *mobile*; sistema web; API.



## **ABSTRACT**

Research shows that drug prices vary greatly from one pharmacy to another within the same city. It is also observed that the number of smartphones used in Brazil exceeds the total amount of the population. These findings motivated the development of this project whose main objective was to develop a mobile system to help drug users find the lowest price of medicines in the pharmacies of Iporá - GO. Pharmacy customers will search for the lowest drug prices by comparing them via the mobile system. Through the web system, will be provided by pharmacies, the relevant data for those who want to search medicines. The integration between mobile and web systems will happen through the API, which will process data in the backend. It is hoped that through the generated system consumers of drugs can save on their health care in a comfortable and agile way.

Keywords: medicines; mobile system; web system; API.

## **LISTA DE ABREVIATURA E DE SIGLAS**

ANVISA - Agência Nacional de Vigilância Sanitária

API – Application Programming Interface

GPS - Global Positioning System

IDE – Integrated Development Environment

REST - Representational State Transfer

CNPJ - Cadastro Nacional da Pessoa Jurídica

HTTP - Hypertext Transfer Protocol

URL – Uniform Resource Locator

JSON – JavaScript Object Notation

## LISTA DE FIGURAS

Figura 1. Diagrama Geral de casos de uso	23
Figura 2. Diagrama de casos de uso específico manter farmácia	24
Figura 3. Diagrama de caso de uso específico manter medicamento	25
Figura 4. Tela inicial de busca de medicamentos	27
Figura 5. Tela de busca de medicamentos por categoria	28
Figura 6. Tela de busca de medicamentos filtrados por categoria	28
Figura 7. Tela de medicamentos com menor preço	29
Figura 8. Tela de exibição de detalhes do medicamento com localização da farmácia	29
Figura 9. Tela de Login	30
Figura 10. Tela de cadastro da farmácia	30
Figura 11. Tela de consulta, alteração e exclusão da farmácia	31
Figura 12. Tela de cadastro de medicamento	32
Figura 13. Tela de consulta de medicamento	32
Figura 14. Tela de alteração de medicamento	33
Figura 15. Tela de recuperação de senha	33
Figura 16. Diagrama de Classe	34
Figura 17 – Arquitetura do Sistema	36
Figura 18. Diagrama Entidade Relacionamento	36
Figura 19. Diagrama Modelo Entidade Relacionamento	37
Figura 20. Diagrama de Implantação	37

## **LISTA DE TABELAS**

Tabela 1: Regras de negócio	22
Tabela 2: Caso de uso específico manter farmácia	24
Tabela 3: Caso de uso específico manter medicamento	26
Tabela 4: Operações REST	35

## **LISTA DE CÓDIGOS-FONTE**

Código-Fonte 1: Classe Medicamentos.java	38
Código-Fonte 2: Classe MedicamentosRepository	41
Código-Fonte 3: Classe MedicamentosService	41
Código-Fonte 4: Classe MedicamentosController	30

## SUMÁRIO

<b>1 IDENTIFICAÇÃO DO PROBLEMA E REVISÃO DA LITERATURA .....</b>	<b>15</b>
<b>2 JUSTIFICATIVA .....</b>	<b>16</b>
<b>3 OBJETIVOS.....</b>	<b>17</b>
<b>3.1 OBJETIVO GERAL.....</b>	<b>17</b>
<b>3.1 OBJETIVOS ESPECÍFICOS.....</b>	<b>17</b>
<b>4 MATERIAIS E MÉTODOS.....</b>	<b>18</b>
<b>5 RESULTADOS E DISCUSSÃO.....</b>	<b>20</b>
<b>5.1 ESCOPO.....</b>	<b>20</b>
<b>5.2 REQUISITOS FUNCIONAIS (CASOS DE USO).....</b>	<b>20</b>
<b>5.3 REQUISITOS NÃO FUNCIONAIS.....</b>	<b>24</b>
<b>5.4 REGRAS DE NEGÓCIO .....</b>	<b>27</b>
<b>TABELA 1 - REGRAS DE NEGÓCIO .....</b>	<b>27</b>
<b>5.5 DIAGRAMAS DE CASOS DE USO .....</b>	<b>27</b>
<b>5.5.1 DIAGRAMA DE CASOS DE USO GERAL.....</b>	<b>27</b>
<b>5.5.2 DIAGRAMAS DE CASOS DE USO ESPECÍFICOS .....</b>	<b>28</b>
<b>5.6 DESCRIÇÃO DA INTERFACE COM O USUÁRIO .....</b>	<b>31</b>
<b>5.7 DIAGRAMAS DE CLASSE.....</b>	<b>38</b>
<b>5.8 ARQUITETURA DO SISTEMA .....</b>	<b>38</b>
<b>5.8.1 UNIFORM RESOURCE LOCATOR (URL).....</b>	<b>39</b>
<b>5.8.2 MÉTODOS HTTP .....</b>	<b>39</b>
<b>5.8.3 JSON .....</b>	<b>39</b>
<b>5.9 DIAGRAMA DE ENTIDADES-RELACIONAMENTO .....</b>	<b>40</b>
<b>5.10 DIAGRAMA MODELO ENTIDADES-RELACIONAMENTO.....</b>	<b>41</b>
<b>5.11 DIAGRAMA DE IMPLANTAÇÃO.....</b>	<b>41</b>
<b>5.12 IMPLEMENTAÇÃO DO SISTEMA PROPOSTO .....</b>	<b>42</b>
<b>6 CONSIDERAÇÕES FINAIS .....</b>	<b>49</b>
<b>REFERÊNCIAS .....</b>	<b>50</b>

## **1 IDENTIFICAÇÃO DO PROBLEMA E REVISÃO DA LITERATURA**

O mundo passa por uma evolução tecnológica em todos os aspectos e no âmbito da saúde não é diferente. Desde os primórdios a humanidade busca pela cura de suas enfermidades. Os povos antigos tinham uma medicina rústica e culpavam o sobrenatural por suas doenças. Com o decorrer do tempo surgiu o método científico e tal ideia foi mudada. Descobriu-se que as doenças eram causadas por bactérias, vírus e logo depois por fatores genéticos (VIEIRA, 2016).

No século XVIII, a “revolução química” possibilitou o descobrimento de novos princípios ativos para o desenvolvimento de medicamentos em laboratório. Porém, só nos próximos séculos o mercado de medicamentos se consolida com a fabricação em larga escala pela indústria, facilitando assim sua obtenção pela população (VIEIRA, 2016).

Nos dias atuais um dos desafios encontrados pelo consumidor de medicamentos no Brasil, são os preços, que às vezes são imputados a excessiva tributação do país (Revista Abrale, 2019). Uma pesquisa realizada pelo PROCON de Goiás verificou uma variação de preços de até 972,26% em medicamentos nas farmácias de Goiânia (PROCON, 2017). E em Iporá - GO não é diferente, podendo ser detectada também uma grande variação de preços de uma farmácia para a outra.

O objetivo deste trabalho foi desenvolver um software que auxilie o usuário a encontrar o menor preço de medicamentos em farmácias por meio de comparação. O aplicativo tem o nome de IporáFarma e envolve os usuários de medicamentos e farmácias da cidade de Iporá- GO.

## 2 JUSTIFICATIVA

Nos últimos anos tem-se notado um crescimento enorme de dispositivos móveis no Brasil. As pesquisas revelam que o número de *smartphones* no país já ultrapassa a quantidade total da população (DEMARTINI, 2018). Dentre os sistemas operacionais mais utilizados no Brasil está o Android com 95,5 % do mercado (HIGA, 2016). Nota-se também a variação de preços de medicamentos de uma farmácia para outra dentro da mesma cidade (PROCON, 2017).

Observa-se o quão propício torna-se o desenvolvimento de um aplicativo para dispositivos móveis que possibilite aos seus usuários a comparação de preços de medicamentos.

Alguns aplicativos disponíveis no mercado como: o Consulta Remédios e o MediPreço realizam estas funções. Ambos são gratuitos, desenvolvidos para as plataformas Android e iPhone (IOS) e auxiliam seus usuários a encontrar o menor preço de medicamentos em todo o país.

O Consulta Remédios apresenta falhas no filtro por região e no leitor do código de barras que quando testado não encontrou os remédios scaneados (Freire, 2017). No MediPreço o usuário fotografa o código de barras do medicamento e o sistema busca o valor máximo permitido pelo Ministério da Saúde para aquele medicamento e o menor valor encontrado na região, através da geolocalização (RPA, 2018). Através de buscas foi verificado que o aplicativo apresenta em sua maioria os valores de medicamentos das grandes redes de farmácia deixando de fora outros estabelecimentos.

O diferencial do aplicativo proposto é que será local, destinado especificadamente para usuários de medicamentos e farmácias de Iporá-GO. A vantagem deste trabalho é o estímulo de concorrência entre as farmácias com a intenção de manter ou aumentar sua gama de clientes.



### **3 OBJETIVOS**

#### **3.1 Objetivo geral**

Desenvolver um aplicativo móvel compatível com Android que contribua para a comparação de preços de medicamentos das farmácias de Iporá - GO.

#### **3.1 Objetivos específicos**

- Definir módulos a serem desenvolvidos;
- Definir o método de comparação de preços;
- Apresentar localização da farmácia no mapa.

## 4 MATERIAIS E MÉTODOS

Neste trabalho foi utilizada como técnica de elicitação de requisitos a entrevista. Esta técnica permite que entrevistador e entrevistado possam ter um diálogo de maneira mais completa. O entrevistador pode identificar e compreender as necessidades do entrevistado (CHAVES, 2017).

A linguagem de programação utilizada para desenvolver os sistemas *mobile* e API foi Java. É uma linguagem orientada a objetos que está entre as linguagens mais utilizadas do mundo. Além disso, está presente em toda a parte e é a mais usada entre outras para o desenvolvimento de aplicativos móveis para Android, aplicações web, desktop, etc (GUEDES, 2019).

Para desenvolver o sistema web foi utilizada a linguagem de programação PHP, uma linguagem de *scripts* muito utilizada e adequada para o desenvolvimento web (ESTRELLA, 2019). Foi utilizado também o Bootstrap, um *framework* de código aberto que proporciona configurações padronizadas para estilização de páginas web sem exigir do desenvolvedor um conhecimento robusto de métodos de design (FERREIRA, 2015).

Foi utilizada a *Integrated development environment* (IDE) Android Studio para desenvolvimento do aplicativo. Esta IDE é bastante completa, além de possuir um editor de códigos inteligente, possui emuladores para se testar os aplicativos desenvolvidos, o que propicia o teste simulado em vários modelos de aparelhos (CORDEIRO, 2017).

Para o desenvolvimento dos sistemas web e API foi usada a IDE Netbeans, é completa, com código fonte aberto, suporte a várias linguagens e permite a edição de códigos com rapidez e inteligência (OFICINA, 2008).

Foram utilizados para o desenvolvimento da API os *frameworks* Spring Boot e Hibernate. O Spring Boot é um projeto da Spring que simplifica os procedimentos de configuração e publicação dos sistemas desenvolvidos (AFONSO, 2017). Já o Hibernate é um *framework* que faz o mapeamento objeto-relacional e é utilizado para armazenar, manipular e acessar dados do banco de dados (MARINHO, 2014).

Foi utilizado o Sistema de gerenciamento de banco de dados (SGBD) MariaDB que concentra seu principal objetivo na segurança dos dados (CARDOSO, 2015). O modelo de processo adotado para projeto foi o *Scrum Solo*, pois se trata de uma adaptação do *Scrum* que se destina a desenvolvimento individual de *softwares*. Nesse tipo de modelagem de processo os *sprints* têm duração de uma semana, que é quando é entregue pelo desenvolvedor um protótipo com novas funções (PAGOTTO et al., 2016).

## 5 RESULTADOS E DISCUSSÃO

### 5.1 Escopo

O escopo deste trabalho envolveu a análise, a modelagem e a implementação dos sistemas móvel e web utilizando API para comparação de preços de medicamentos de Iporá - GO.

Através do sistema móvel o cliente é capaz de consultar medicamentos com menor preço. O cliente poderá consultar preços sem se cadastrar no sistema. No sistema web a farmácia deve manter seu cadastro e manter o cadastro dos medicamentos.

A localização das farmácias cadastradas será informada pelo sistema GPS ao sistema móvel. O sistema API será responsável por realizar o recebimento e o envio de dados através de requisições HTTP realizadas tanto pelo sistema web quanto pelo sistema móvel.

Não será possível a realização de compras de medicamentos através do aplicativo. O sistema como um todo, não fará o controle de estoque das farmácias.

### 5.2 Requisitos funcionais (casos de uso)

Os requisitos apresentados nesta seção estão relacionados com as funcionalidades dos sistemas mobile e web.

#### [RF001] Consultar Medicamento Mobile

O caso de uso consultar medicamento tem por função consultar os dados do medicamento.

**Ator:** Cliente

**Prioridade:**  Essencial  Importante  Desejável

**Entradas e pré-condições:** Informar o nome, princípio ativo ou categoria do medicamento.

**Saídas e pós-condições:** O sistema apresenta todos os medicamentos que corresponderem ao nome do medicamento, princípio(s) ativo(s) ou categoria

informada ordenados pelo menor preço. O sistema informa que o medicamento não consta no cadastro.

#### **[RF002] Efetuar Login usuário web**

O caso de uso efetuar login usuário web tem por objetivo autenticar o usuário no sistema através do nome de usuário e senha previamente cadastrados.

**Ator:** Farmácia

**Prioridade:**  Essencial  Importante  Desejável

**Entradas e pré-condições:** Nome de usuário e senha. A farmácia deve estar cadastrada.

**Saídas e pós-condições:** O sistema dá acesso às funcionalidades ao usuário.

#### **[RF003] Lembrar senha usuário web**

O caso de uso lembrar senha usuário web tem por objetivo guardar a senha do usuário.

**Ator:** Farmácia

**Prioridade:**  Essencial  Importante  Desejável

**Entradas e pré-condições:** A Farmácia ter informado login e senha válidos.

**Saídas e pós-condições:** A senha informada fica armazenada.

#### **[RF004] Recuperar senha usuário Web**

O caso de uso recuperar senha usuário web tem por objetivo recuperar a senha do usuário.

**Ator:** Farmácia

**Prioridade:**  Essencial  Importante  Desejável

**Entradas e pré-condições:** E-mail. Usuário cadastrado.

**Saídas e pós-condições:** O sistema envia um código de verificação para o e-mail cadastrado.

#### **[RF005] Cadastrar Farmácia**

O caso de uso cadastrar farmácia tem por função armazenar os dados da farmácia tais como nome de usuário, senha, nome da farmácia, CNPJ, e-mail, telefone e endereço.

**Ator:** Ator: Farmácia

**Prioridade:**  Essencial  Importante  Desejável

**Entradas e pré-condições:** Não há.

**Saídas e pós-condições:** O sistema informa que o cadastro foi criado.

#### [RF006] Alterar Farmácia

O caso de uso alterar farmácia tem por função a alteração dos dados da farmácia.

**Ator:** Ator: Farmácia

**Prioridade:**  Essencial  Importante  Desejável

**Entradas e pré-condições:** A farmácia deve estar cadastrada.

**Saídas e pós-condições:** O sistema informa que o cadastro foi alterado. O sistema informa que não foi possível alterar os dados.

#### [RF007] Consultar Farmácia

O caso de uso consultar farmácia tem por função consultar os dados da farmácia.

**Ator:** Ator: Farmácia

**Prioridade:**  Essencial  Importante  Desejável

**Entradas e pré-condições:** A farmácia deve estar cadastrada.

**Saídas e pós-condições:** O sistema mostra os dados de cadastro da farmácia.

#### [RF008] Desabilitar Farmácia

O caso de uso desabilitar farmácia tem por função desabilitar as alterações e consultas ligadas a determinada farmácia por certo tempo desejado.

**Ator:** Ator: Farmácia

**Prioridade:**  Essencial  Importante  Desejável

**Entradas e pré-condições:** A farmácia deve estar cadastrada.

**Saídas e pós-condições:** O sistema informa que o cadastro foi desabilitado.

#### [RF009] Cadastrar Medicamento

O caso de uso cadastrar medicamento tem por função armazenar os dados dos medicamentos tais como: nome comercial do produto, princípio(s) ativo(s), apresentação do medicamento, incluindo a concentração, forma farmacêutica, quantidade, número de registro na ANVISA, imagem, categoria, nome do detentor do registro e preço.

**Ator:** Farmácia

**Prioridade:**  Essencial  Importante  Desejável

**Entradas e pré-condições:** A farmácia deve estar cadastrada.

**Saídas e pós-condições:** O sistema informa que o medicamento foi cadastrado. O sistema informa que não foi possível cadastrar o medicamento.

#### **[RF0010] Alterar Medicamento**

O caso de uso alterar medicamento tem por objetivo alterar os dados do medicamento.

**Ator:** Farmácia

**Prioridade:**      Essencial             Importante             Desejável

**Entradas e pré-condições:** O medicamento deve estar cadastrado.

**Saídas e pós-condições:** O sistema informa que a alteração foi realizada. O sistema informa que não foi possível alterar os dados do medicamento.

#### **[RF0011] Desabilitar Medicamento**

O caso de uso desabilitar medicamento tem por objetivo desabilitar as alterações e consultas ligadas a determinado medicamento por certo tempo desejado.

**Ator:** Farmácia

**Prioridade:**      Essencial             Importante             Desejável

**Entradas e pré-condições:** O medicamento deve estar cadastrado.

**Saídas e pós-condições:** O sistema informa que o medicamento foi desabilitado. O sistema informa que não foi possível desabilitar o medicamento.

#### **[RF0012] Consultar Medicamento Web**

O caso de uso consultar medicamento tem por função consultar os dados do medicamento.

**Ator:** Farmácia

**Prioridade:**             Essencial             Importante             Desejável

**Entradas e pré-condições:** O medicamento deve estar previamente cadastrado.

**Saídas e pós-condições:** O sistema apresenta todos os medicamentos, incluindo habilitados e desabilitados. O sistema informa que o medicamento não consta no cadastro.

### 5.3 Requisitos não funcionais

#### Usabilidade

Este tópico descreve os requisitos não funcionais associados à facilidade de uso da interface com o usuário, material de treinamento e documentação do sistema.

#### [NF001] Inteligibilidade

A interface gráfica dos sistemas oferece clareza na exibição das informações, fluidez durante a navegação.

**Prioridade:**     Essencial             Importante             Desejável

#### [NF002] Apreensibilidade

Uso de ícones intuitivos. Os sistemas utilizam ícones intuitivos que facilitam o reconhecimento de suas funcionalidades.

**Prioridade:**     Essencial             Importante             Desejável

#### [NF003] Operacionalidade

O sistema móvel oferece facilidade na realização da busca de medicamentos, com apenas três cliques o cliente obtém as informações que deseja.

**Prioridade:**     Essencial             Importante             Desejável

#### Confiabilidade

Este tópico descreve os requisitos não funcionais associados à frequência, severidade de falhas do sistema e habilidade de recuperação, bem como à correção do sistema.

#### [NF004] Disponibilidade

Os sistemas estão disponíveis aos usuários vinte e quatro horas por dia, sete dias por semana.

**Prioridade:**     Essencial             Importante             Desejável

#### [NF005] Recuperabilidade

Quando os formulários apresentarem alguma falha, o sistema retorna ao seu *status* anterior no banco de dados.



**Prioridade:**     Essencial             Importante             Desejável

#### **[NF006] Prevenção de erros**

A validação de dados restringe o tipo de dados que os usuários inserem nos campos.

**Prioridade:**     Essencial             Importante             Desejável

#### **Desempenho**

Este tópico descreve os requisitos não funcionais associados à eficiência, uso de recursos e tempo de resposta do sistema.

#### **[NF007] Interoperabilidade**

O sistema utiliza uma *Application Programming Interface* (API) para enviar e consumir dados para o aplicativo através de requisições *Hypertext Transfer Protocol* (HTTP). Assim, grande parte do processamento fica a cargo do servidor e não dos sistemas móvel e web.

**Prioridade:**     Essencial             Importante             Desejável

#### **[NF008] Tempo de Resposta**

Através de teste de estresse realizado com o *Webserver Stress Tool* constatou-se que o sistema é capaz de suportar o total de 1300 usuários realizando simultaneamente 3 requisições cada um, com o tempo de resposta de 6 segundos.

**Prioridade:**     Essencial             Importante             Desejável

#### **Segurança**

Este tópico descreve os requisitos não funcionais associados à integridade, privacidade e autenticidade dos dados do sistema.

#### **[NF009] Tipos de acessos**

A Farmácia tem acesso restrito às funcionalidades destinadas à sua categoria.

**Prioridade:**     Essencial             Importante             Desejável

## **Distribuição**

Este tópico descreve os requisitos não funcionais associados à distribuição da versão executável do sistema.

### **[NF0010] Requisito de sistema**

O aplicativo será distribuído para o sistema operacional Android a partir da versão 4.0.3 (Ice Cream Sandwich) até a 8.0 (Oreo).

**Prioridade:**     Essencial             Importante             Desejável

## **Padrões**

Este tópico descreve os requisitos não funcionais associados a padrões ou normas que devem ser seguidos pelo sistema ou pelo seu processo de desenvolvimento.

### **[NF0011] Conformidade**

O aplicativo segue as normas determinadas pela ANVISA para divulgação dos preços dos medicamentos disponíveis nas farmácias.

**Prioridade:**     Essencial             Importante             Desejável

## **Hardware e software**

Este tópico descreve os requisitos não funcionais associados ao hardware e software usados para desenvolver ou para executar o sistema.

### **[NF0012] Conexão com a Internet**

Necessita de dispositivo conectado à *Internet* para utilizar o sistema.

**Prioridade:**     Essencial             Importante             Desejável

## 5.4 Regras de negócio

Tabela 1 - Regras de negócio

<b>RN001</b>	Para efetuar consultas, cliente mobile não necessita estar cadastrado.
<b>RN002</b>	O sistema web permite o acesso dos usuários somente pela tela de login.
<b>RN003</b>	O preenchimento dos campos indicados para medicamentos contidos nos sistemas mobile e web deverão conter obrigatoriamente: Nome comercial do produto; Princípio(s) ativo(s), conforme Denominação Comum Brasileira; Apresentação do medicamento, incluindo a concentração, forma farmacêutica e a quantidade; número de registro na ANVISA; imagem, categoria, nome do detentor do registro e preço.
<b>RN004</b>	As listas de preços do sistema móvel não poderão utilizar designações, símbolos, figuras, imagens, desenhos, marcas figurativas ou mistas, slogans e quaisquer argumentos de cunho publicitário em relação aos medicamentos.
<b>RN005</b>	Somente o usuário web cadastrado poderá consultar, alterar, e desabilitar a farmácia.
<b>RN006</b>	Somente o usuário web cadastrado poderá cadastrar, consultar, alterar, habilitar e desabilitar medicamentos.
<b>RN007</b>	A localização das farmácias cadastradas será informada pelo sistema GPS.
<b>RN008</b>	Forma farmacêutica é a maneira como os medicamentos são preparados, apresentados e conseqüentemente comercializados e utilizados, ou seja, comprimido, xarope, suspensão e outros.

Fonte: Criada pelos autores

## 5.5 Diagramas de casos de uso

### 5.5.1 Diagrama de casos de uso geral

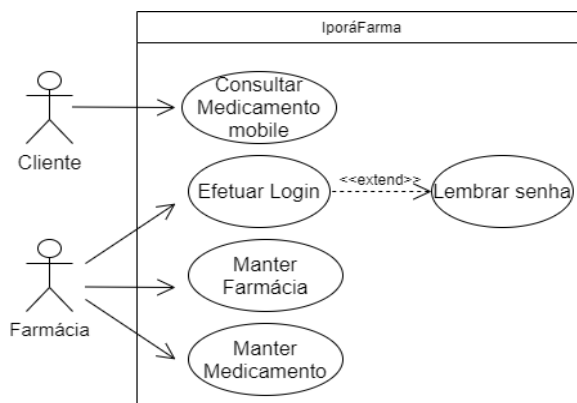


Figura 1. Diagrama Geral de casos de uso

Fonte: Criada pelos autores

## 5.5.2 Diagramas de casos de uso específicos

### 5.5.2.1 Manter Farmácia

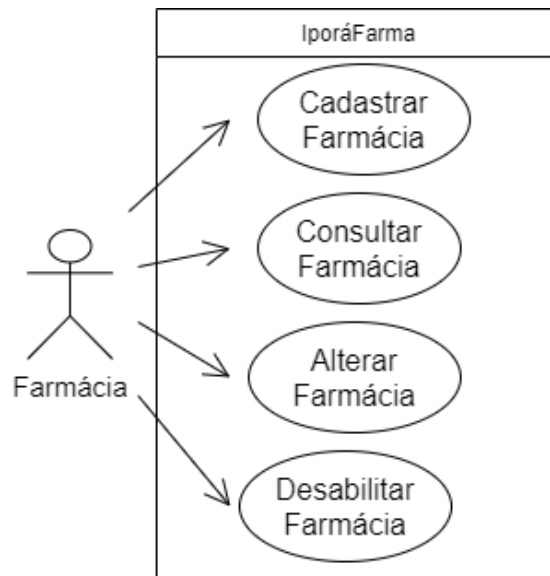


Figura 2. Diagrama de casos de uso específico manter farmácia

Fonte: Criada pelos autores

Tabela 2-Caso de uso específico manter farmácia

<b>Caso de Uso</b>	<b>Manter Farmácia</b>
<b>Ator</b>	Farmácia
<b>Pré-Condições</b>	Não há
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	O sistema apresenta a tela principal com as opções de logar e cadastrar.
O ator seleciona a opção cadastrar.	O sistema carrega a tela de cadastro.
O ator realiza seu cadastrado inserindo o nome de usuário, senha, nome da farmácia, CNPJ, e-mail, telefone, endereço e aciona a opção salvar	O sistema salva os dados de cadastro no banco e informa que o cadastro foi realizado com sucesso.
<b>Fluxos Alternativos</b>	
<b>Fluxo Alternativo I</b>	Consultar Farmácia

<b>Ações do Ator</b>	<b>Ações do Sistema</b>
O ator seleciona a opção Consultar Farmácia.	O sistema carrega a tela de consulta verifica se os dados informados existem no banco e os exibe na tela.
<b>Fluxo Alternativo II</b>	Alterar Farmácia
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
O ator seleciona a opção Alterar Farmácia.	O sistema carrega a tela de alteração.
O ator altera os dados desejados em campos específicos e aciona a opção salvar.	O sistema exibe uma mensagem de alteração realizada com sucesso.
<b>Fluxo Alternativo III</b>	Desabilitar Farmácia
O ator seleciona a opção Desabilitar Farmácia.	O sistema exibe a tela de desabilitar.
O ator desabilita o cadastro da Farmácia.	O sistema exibe uma tela dizendo que o cadastro foi desabilitado com sucesso.
<b>Pós-Condições</b>	Os dados foram cadastrados, ou alterados ou visualizados, ou desabilitados no sistema.
<b>Fluxo de exceção</b>	
	O sistema encontrou algum dado inconsistente (faltando ou inválido) e emite mensagem relatando o erro verificado.
<b>Regras de Negócios</b>	<b>RN002, RN005</b>

Fonte: Criada pelos autores

### 5.5.2.3 Manter Medicamento

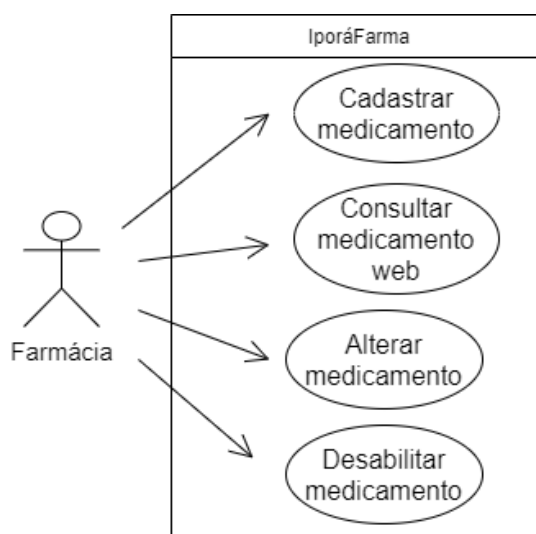


Figura 3. Diagrama de caso de uso específico manter medicamento

Fonte: Criada pelos autores

**Tabela 3** - Caso de uso específico manter medicamento

<b>Caso de Uso</b>		<b>Manter Medicamento</b>
<b>Ator</b>	Farmácia	
<b>Pré-Condições</b>	O ator deve estar logado no sistema.	
<b>Fluxo Principal</b>		
<b>Ações do Ator</b>	<b>Ações do Sistema</b>	
	O sistema apresenta a tela principal com as opções cadastrar, consultar, alterar desabilitar e habilitar.	
O ator seleciona a opção cadastrar.	O sistema carrega a tela de cadastro.	
O ator realiza o cadastro do medicamento inserindo: nome, princípio (s) ativo(s), concentração, forma farmacêutica, quantidade, número de registro na ANVISA, nome do detentor do registro, preço, imagem e categoria.	O sistema verifica se o medicamento ainda não é cadastrado.	
O ator aciona a opção salvar.	O sistema salva os dados do medicamento no banco e informa que o cadastro foi realizado com sucesso.	
<b>Fluxos Alternativos</b>		
<b>Fluxo Alternativo I</b>	Consultar Medicamento web	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>	
O ator seleciona a opção Consultar Medicamento.	O sistema carrega a tela de consulta.	
O ator insere o nome do medicamento, princípio(s) ativo(s) ou categoria.	O sistema verifica se os dados informados existem no banco e os exibe na tela.	
<b>Fluxo Alternativo II</b>	Alterar Medicamento	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>	
O ator seleciona a opção Alterar Medicamento.	O sistema carrega a tela de alteração.	
O ator insere o nome do medicamento ou princípio(s) ativo(s).	O sistema verifica se o medicamento ou princípio(s) ativo(s) está cadastrado no banco de dados e carrega os dados do medicamento.	
O ator altera os dados desejados em campos específicos e aciona a opção salvar.	O sistema exibe uma mensagem de alteração realizada com sucesso.	
<b>Fluxo Alternativo III</b>	Desabilitar Medicamento	
O ator seleciona a opção Desabilitar Medicamento.	O sistema exibe a tela de desabilitar.	

O ator insere o nome do medicamento ou princípio(s) ativo(s).	O sistema verifica se o medicamento ou princípio(s) ativo(s) está cadastrado no banco de dados e carrega os dados do medicamento.
O ator desabilita o cadastro do medicamento.	O sistema exibe uma mensagem que o medicamento foi desabilitado com sucesso.
<b>Fluxo Alternativo IV</b>	
<b>Habilitar Medicamento</b>	
O ator seleciona a opção Habilitar Medicamento.	O sistema exibe a tela de habilitar.
O ator insere o nome do medicamento ou princípio(s) ativo(s).	O sistema verifica se o medicamento ou princípio(s) ativo(s) está cadastrado no banco de dados e carrega os dados do medicamento.
O ator habilita o medicamento.	O sistema exibe uma mensagem que o medicamento foi habilitado com sucesso.
<b>Pós-Condições</b>	
	Os dados foram cadastrados, ou visualizados, ou alterados, ou desabilitados, ou habilitados no sistema.
<b>Fluxo de exceção</b>	
	O sistema encontrou algum dado inconsistente (faltando ou inválido) e emite mensagem relatando o erro verificado.
<b>Regras de Negócios</b>	
	<b>RN002, RN003, RN006, RN008</b>

Fonte: Criada pelos autores

## 5.6 Descrição da interface com o usuário

### 5.6.1 Tela inicial do sistema mobile

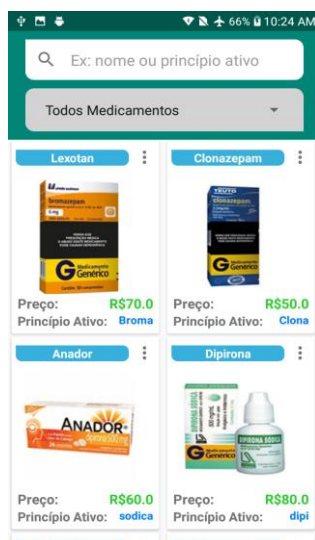


Figura 4. Tela inicial de busca de medicamentos

Através da tela inicial o cliente poderá optar por buscar o medicamento por nome ou princípio ativo. Terá também a opção de buscar o medicamento por categoria ou visualizar todos os medicamentos.

### 5.6.2 Tela de busca de medicamentos por categoria



Figura 5. Tela de busca de medicamentos por categoria

### 5.6.3 Tela de busca de medicamentos filtrados por categoria



Figura 6. Tela de busca de medicamentos filtrados por categoria



### 5.6.4 Tela de exibição de medicamentos com menor preço

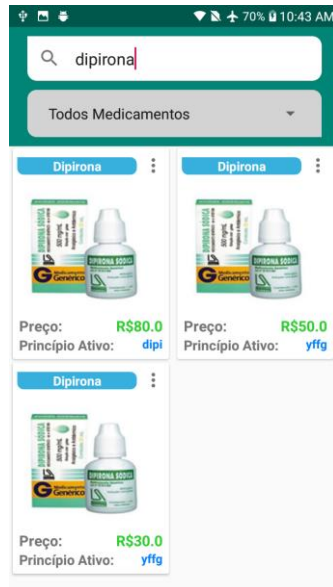


Figura 7. Tela de medicamentos com menor preço

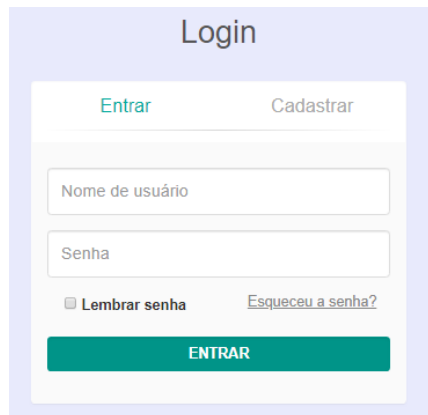
### 5.6.4 Tela de exibição de detalhes do medicamento com localização da farmácia



Figura 8. Tela de detalhes de medicamentos com localização da farmácia

## 5.6.5 Telas do sistema web

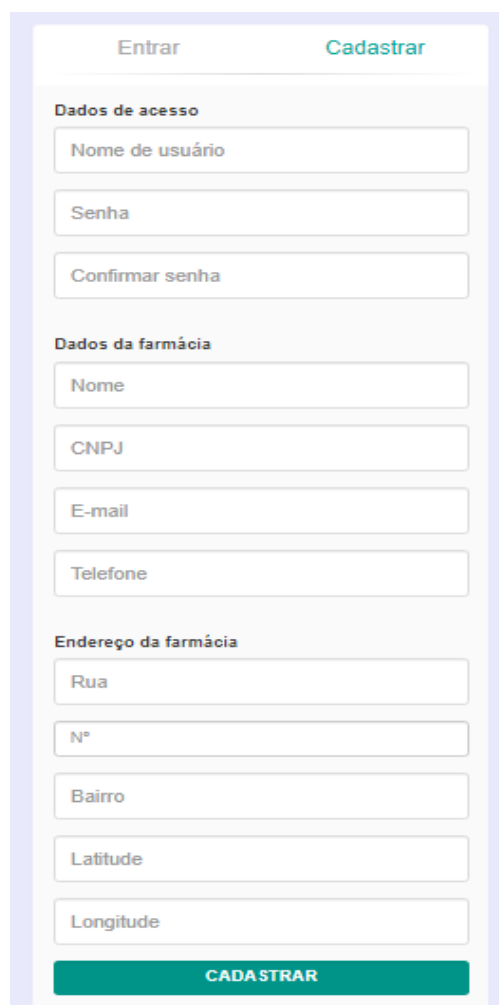
### 5.6.6 Tela de Login da Farmácia



The screenshot shows a login form titled "Login". At the top, there are two tabs: "Entrar" (selected) and "Cadastrar". Below the tabs are two input fields: "Nome de usuário" and "Senha". Under the "Senha" field, there is a checkbox labeled "Lembrar senha" and a link "Esqueceu a senha?". At the bottom of the form is a large green button labeled "ENTRAR".

Figura 9. Tela de Login

### 5.6.7 Tela de Cadastro da Farmácia



The screenshot shows a registration form titled "Cadastrar". At the top, there are two tabs: "Entrar" and "Cadastrar" (selected). The form is divided into three sections: "Dados de acesso", "Dados da farmácia", and "Endereço da farmácia".

- Dados de acesso:** Three input fields for "Nome de usuário", "Senha", and "Confirmar senha".
- Dados da farmácia:** Four input fields for "Nome", "CNPJ", "E-mail", and "Telefone".
- Endereço da farmácia:** Five input fields for "Rua", "Nº", "Bairro", "Latitude", and "Longitude".

At the bottom of the form is a large green button labeled "CADASTRAR".

Figura 10. Tela de cadastro da farmácia

## 5.6.8 Tela de Consulta, Alteração e exclusão da Farmácia

**Minha Conta**

Informações sobre a farmácia

**Dados da farmácia**

Farmácia Tocantins

22.222.222/2222-22

(64) 3674-8920

**Endereço da farmácia**

esmerindo pereira

767

central

-16.444121

-51.120541

**Configurações Adicionais**

**Alterar Conta**

Para alterar os dados da farmácia, siga as instruções abaixo.

Para alterar sua conta click em "Alterar" **Alterar**

**Excluir Conta**

Atenção, ao realizar este passo você perderá acesso imediato a sua conta, bem como a todos os produtos já cadastrados! Se mesmo assim deseja prosseguir, siga as instruções abaixo.

Para excluir sua conta click em "Excluir" **Excluir**

Figura 11. Tela de consulta, alteração e exclusão da farmácia

## 5.6.9 Tela cadastro de medicamento

Novo Medicamento

Todos os itens com ' \* ' posterior ao nome do campo são de preenchimento obrigatório!

**Nome do medicamento\***

**Princípio Ativo\***

**Registro da Anvisa\***  **Detentor do Registro\***

**Concentração\***

**Preço\***  **Quantidade\***

**Categorias\***  
 Seleccione uma ou mais categorias  
 Ansiolítico  
 Analgésico  
 Anti-inflamatório  
 ...

**Forma Farmacêutica\***

**Imagem**  
 Nenhum arquivo selecionado

Figura 12. Tela de cadastro de medicamento

## 5.6.10 Tela Consulta de medicamento

Todos os medicamentos

Buscar Medicamento

Habilitados Desabilitados

Imagem	Nome	Forma Farmacêutica	Concentração	Detentor do Registro	Registro Anvisa	Princípio Ativo	Preço	Quantidade	Action
	Azitromicina	comprimido	6565	hgfbn v	6576	hgfgff	40,00	66544	<input type="button" value="Alterar"/> <input type="button" value="Desabilitar"/>
	Nimesulida	comprimido	8798	jujgj	6576	jghgj	90,00	654	<input type="button" value="Alterar"/> <input type="button" value="Desabilitar"/>
	Diclofenaco	comprimido	8767	ghgh	7657	jhgj	40,00	655	<input type="button" value="Alterar"/> <input type="button" value="Desabilitar"/>
	Dipirona	comprimido	8776	hgh	6565	yfig	80,00	66	<input type="button" value="Alterar"/> <input type="button" value="Desabilitar"/>
	Anador	comprimido	4556	ghghg	87878	kjhj	60,00	66	<input type="button" value="Alterar"/> <input type="button" value="Desabilitar"/>

Figura 13. Tela de consulta de medicamento

Através desta tela a farmácia consulta o medicamento por nome ou princípio ativo assim como também visualiza todos os medicamentos (habilitados e desabilitados).

### 5.6.11 Tela Alteração de medicamento

Figura 14. Tela de alteração de medicamento

### 5.6.12 Tela de recuperação de senha

Figura 15. Tela de recuperação de senha

## 5.7 Diagramas de classe

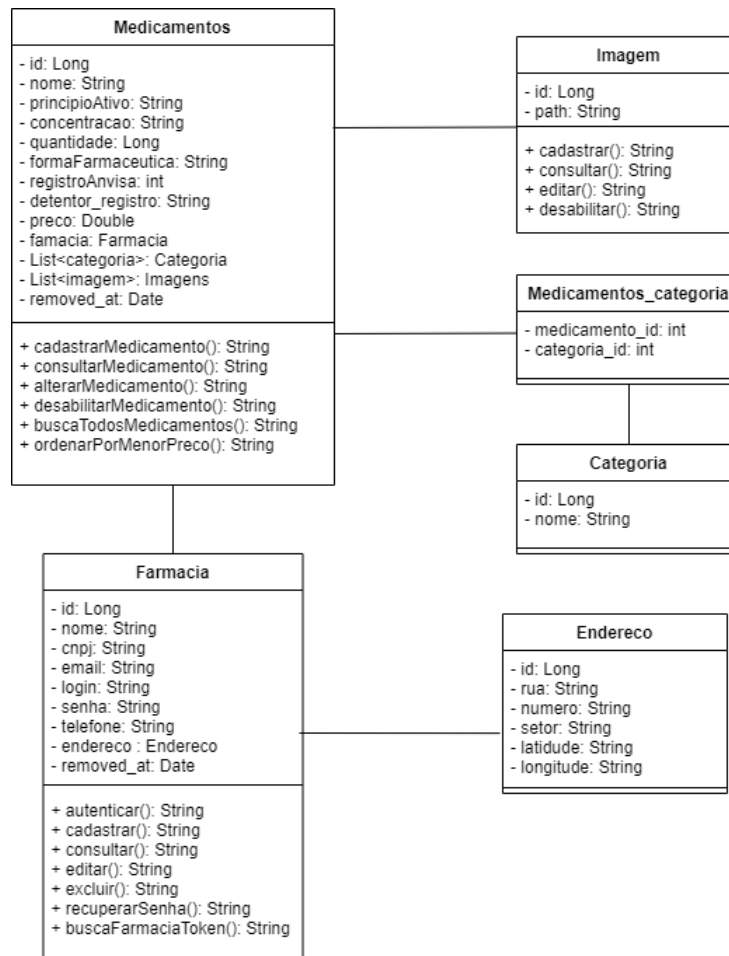


Figura 16. Diagrama de Classe

Fonte: Criada pelos autores

## 5.8 Arquitetura do Sistema

A *Application Programming Interface* (API) desenvolvida foi fundamentada na arquitetura *Representational State Transfer* (REST), um modelo baseado nos serviços web que é composto de princípios e regras que propiciam a elaboração de um projeto com interfaces bem definidas. Além de permitir a interoperabilidade entre aplicações. (FERREIRA, 2017).

O conceito principal de REST está atrelado a recursos, ou seja, dados e funcionalidades são tidos como recursos que são acessados por meio de *Uniform Resource*

*Locator* (URL) e manipulados através de métodos *Hypertext Transfer Protocol* (HTTP) (FERREIRA, 2017).

### 5.8.1 Uniform Resource Locator (URL)

No meio computacional a URL é um localizador de recursos universal, ou seja, define a localização de recursos na Internet, usualmente através de links (FERREIRA, 2017). Também é denominado como endereço da web e é dividido em várias partes que incluem um protocolo e um nome de domínio, responsáveis por informar ao navegador da Web como e onde recuperar um recurso (ROUSE, 2019).

### 5.8.2 Métodos HTTP

O HTTP é o protocolo de comunicação de dados da internet (Web). Os métodos ou verbos HTTP são utilizados para dizer ao servidor qual ação tomar quando receber uma requisição ou “mensagem” via cliente (FERREIRA, 2017). A tabela a seguir apresenta os principais métodos:

Tabela 4 – Operações REST

POST	Cria um novo recurso
GET	Consulta um recurso
PUT	Edita um recurso
DELETE	Desabilita ou exclui um recurso

Fonte: Criada pelos autores

### 5.8.3 JSON

Quando um cliente realiza uma solicitação a um recurso, o servidor faz várias atividades e retorna uma mensagem. A mensagem retornada é a representação do recurso determinado. Normalmente o formato da mensagem é identificado no cabeçalho da requisição por um campo chamado *Content-type*. Em se tratando da arquitetura REST podem ser utilizados outros formatos, porém neste trabalho foi utilizado o *JavaScript Object Notation* (JSON) por se tratar de um formato leve para troca de informações entre sistemas e por ser simples de ser lido (FERREIRA, 2017).

A comunicação entre os sistemas web, aplicativo móvel e o back-end ocorre através de requisições HTTP, utilizando os verbos: GET, POST, PUT e DELETE. O tipo do dado

trafegado é no formato JSON. Existem outros sistemas que interagem com a API, um servidor de e-mail e um banco de dados, e com o aplicativo mobile, uma API do Serviço de Localização Global (GPS), como ilustrado na figura 17:

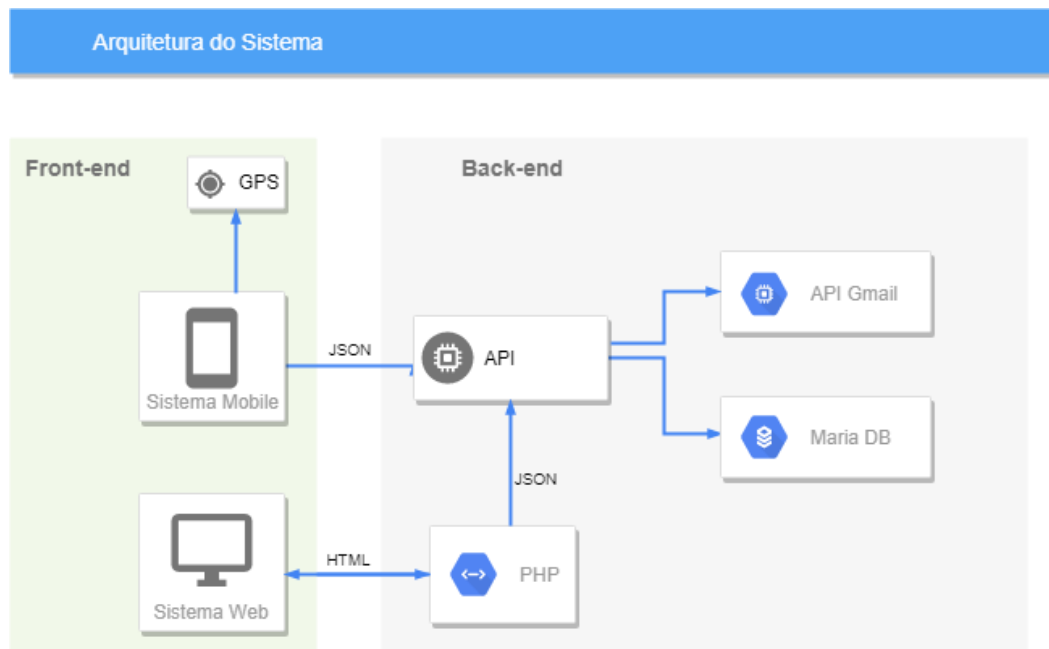


Figura 17. Arquitetura do Sistema

Fonte: Criada pelos autores

## 5.9 Diagrama de Entidades-Relacionamento

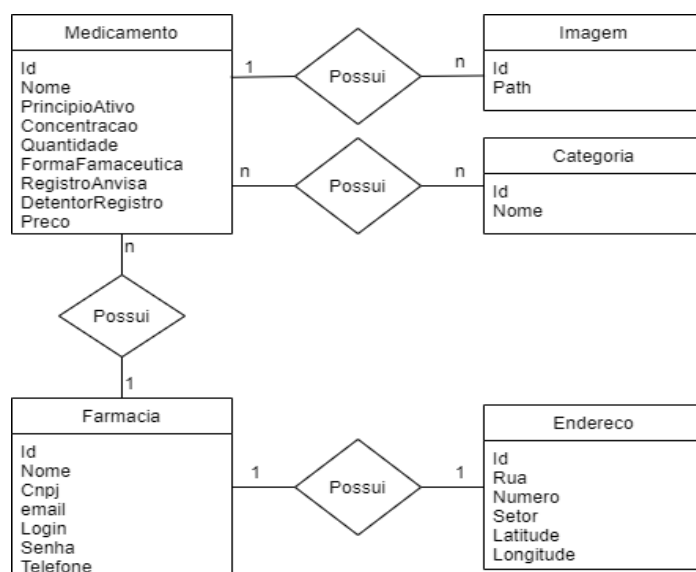


Figura 18. Diagrama Entidade Relacionamento

Fonte: Criada pelos autores



## 5.10 Diagrama Modelo Entidades-Relacionamento

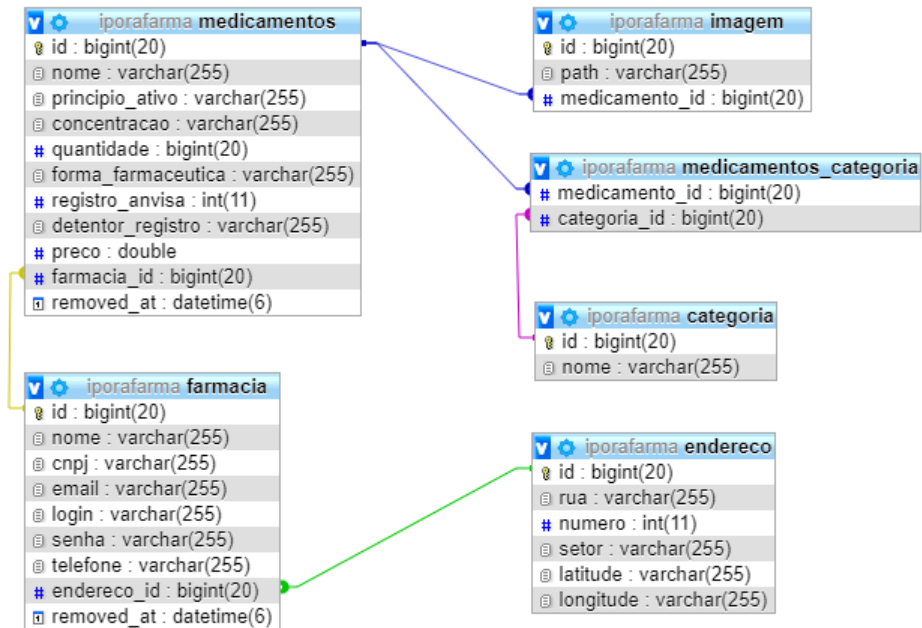


Figura 19. Diagrama Modelo Entidade Relacionamento

Fonte: Criada pelos próprios autores

## 5.11 Diagrama de Implantação

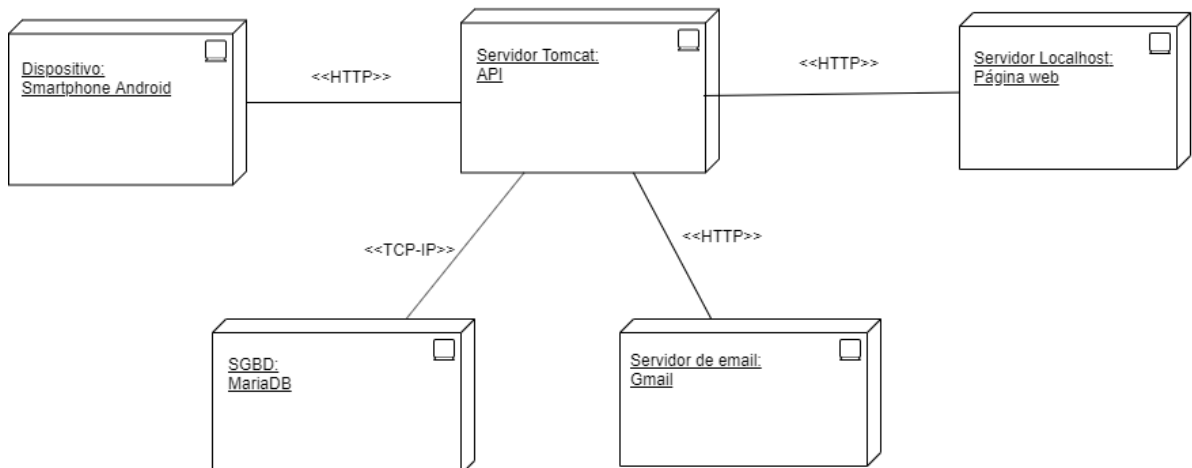


Figura 20. Diagrama de Implantação

Fonte: Criada pelos próprios autores

## 5.12 Implementação do Sistema Proposto

A API desenvolvida tem a estrutura de pacotes: model (classes que contém os pojos), repository (classes responsáveis por realizar as funções de persistência (CRUD e outras operações)), service (classes que realizam um ou mais serviços para outras classes, validações, inclusões das regras de negócios) e controller (classes responsáveis por responder as requisições HTTP).

```
package com.example.demo.model;

import com.fasterxml.jackson.annotation.JsonIgnore;
import java.io.Serializable;
import java.util.Date;
import java.util.List;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToMany;
import javax.persistence.OneToOne;
import javax.persistence.OneToMany;

@Entity // A classe que será persistida no banco de dados recebe a anotação
@Entity
public class Medicamentos implements Serializable{

    private Long id;
    private String nome;
    private String principioAtivo;
    private String concentracao;
    private String formaFarmaceutica;
    private int registroAnvisa;
    private String detentorRegistro;
    private Farmacia farmacia;
    private List<Categoria> categoria;
    private double preco;
    private Long quantidade;
    private List<Imagem> imagens;
    private Date removedAt;

    @Id //Define que o atributo é primary key
    @GeneratedValue(strategy=GenerationType.IDENTITY) //define qual será a estratégia de criação de chave
    primária, neste caso (IDENTITY) será automática, de responsabilidade do banco de dados.
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }
}
```

```

}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public double getPreco() {
    return preco;
}

public void setPreco(double preco) {
    this.preco = preco;
}

public Long getQuantidade() {
    return quantidade;
}

public void setQuantidade(Long quantidade) {
    this.quantidade = quantidade;
}

public String getPrincipioAtivo() {
    return principioAtivo;
}

public void setPrincipioAtivo(String principioAtivo) {
    this.principioAtivo = principioAtivo;
}

public String getConcentracao() {
    return concentracao;
}

public void setConcentracao(String concentracao) {
    this.concentracao = concentracao;
}

public String getFormaFarmaceutica() {
    return formaFarmaceutica;
}

public void setFormaFarmaceutica(String formaFarmaceutica) {
    this.formaFarmaceutica = formaFarmaceutica;
}

public int getRegistroAnvisa() {
    return registroAnvisa;
}

public void setRegistroAnvisa(int registroAnvisa) {
    this.registroAnvisa = registroAnvisa;
}

public String getDetentorRegistro() {

```

```

return detentorRegistro;
}

public void setDetentorRegistro(String detentorRegistro) {
    this.detentorRegistro = detentorRegistro;
}

// A notação @OneToMany especifica que um medicamento possui uma lista de imagens.
// A notação cascade, salva em cascata, altera pai e filho em cascata, remove em cascata.
@OneToMany(cascade = {CascadeType.PERSIST, CascadeType.MERGE, CascadeType.REMOVE})
@JoinColumn(name = "medicamento_id") // cria a coluna medicamento_id que fará a ligação entre as tabelas
public List<Imagem> getImagens() {
    return imagens;
}

public void setImagens(List<Imagem> imagens) {
    this.imagens = imagens;
}

//A notação @ManyToMany especifica que um medicamento possui várias categorias e que uma categoria
possui vários medicamentos.
// A notação cascade, salva em cascata, altera pai e filho em cascata, remove em cascata.

@ManyToMany(cascade = {CascadeType.DETACH, CascadeType.MERGE, CascadeType.REMOVE})
public List<Categoria> getCategoria() {
    return categoria;
}
public void setCategoria(List<Categoria> categoria) {
    this.categoria = categoria;
}

// A notação @ OneToOne especifica que um medicamento está contido em uma farmácia e que uma farmácia
contém um medicamento.
// A notação cascade, salva em cascata, altera pai e filho em cascata, remove em cascata.

@OneToOne(cascade={ CascadeType.REFRESH,CascadeType.REMOVE })
public Farmacia getFarmacia() {
    return farmacia;
}

public void setFarmacia(Farmacia farmacia) {
    this.farmacia = farmacia;
}

@Column(name = "removed_at") // cria a coluna removed_at
public Date getRemovedAt() {
    return removedAt;
}

public void setRemovedAt(Date removedAt) {
    this.removedAt = removedAt;
}
}

```

Código-fonte 1. Classe Medicamentos.java

```

package com.example.demo.repository;

import com.example.demo.model.Categoria;
import com.example.demo.model.Medicamentos;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.yuequan.jpa.soft.delete.repository.SoftDelete;

@SoftDelete //sinaliza registros como desabilitados.
public interface MedicamentosRepository extends JpaRepository<Medicamentos, Long> {

public List<Medicamentos> findByCategoria(Categoria categoria);

public List<Medicamentos> findByNomeOrPrincipioAtivoOrderByPrecoAsc(@Param("nome")String
nome,@Param("principioAtivo")String principioAtivo);

@Query("select m from Medicamentos m where m.farmacia.id= :idFarmacia")
public List<Medicamentos> FindAllEvenRemoved(@Param ("idFarmacia")Long idFarmacia);
}

```

Código-fonte 2. Classe MedicamentosRepository

```

package com.example.demo.service;

import com.example.demo.model.Medicamentos;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.example.demo.repository.MedicamentosRepository;

@Service
public class MedicamentosService {

    @Autowired
    MedicamentosRepository produtoRepository;

    public Medicamentos cadastrarMedicamento(Medicamentos pro) {
        return produtoRepository.save(pro);
    }

    public Medicamentos alterarMedicamento(Medicamentos pro) {
        return produtoRepository.save(pro);
    }

    public void desabilitarMedicamento(Long id) {
        produtoRepository.deleteById(id);
    }

    public Medicamentos consultarMedicamento(Long id) {
        return produtoRepository.findById(id).get();
    }

    public List<Medicamentos> buscaTodosProdutos(){
        return produtoRepository.findAll();
    }
}

```

```

    }

    public List<Medicamentos> ordenarPorMenorPreco(String nome, String principioAtivo){

        return produtoRepository.findByNomeOrPrincipioAtivoOrderByPrecoAsc(nome, principioAtivo);

    }

}

```

Código-fonte 3. Classe MedicamentosService

```

package com.example.demo.controller;

package com.example.demo.controller;

import ch.qos.logback.core.util.OptionHelper;
import com.example.demo.model.Categoria;
import com.example.demo.model.Farmacia;
import com.example.demo.model.Imagem;
import com.example.demo.model.Medicamentos;
import com.example.demo.service.CategoriaService;
import com.example.demo.service.FarmaciaService;
import com.example.demo.service.MedicamentosService;
import java.util.ArrayList;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestHeader;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

/*
 @author Alcídia Cristina
 */

@RestController
@RequestMapping(value = "/")
public class MedicamentosController {

    @Autowired
    private MedicamentosService produtoService;

    @Autowired
    private FarmaciaService farmaciaService;

    @Autowired
    private CategoriaService categoriaService;

    @RequestMapping(method = RequestMethod.POST, consumes =
    MediaType.APPLICATION_JSON_VALUE, value = "/adminAut/produto")

```

```

public ResponseEntity cadastrarProduto(@RequestHeader HttpHeaders headers, @RequestBody
Medicamentos produto){

    //Obtém através do token a farmácia que está cadastrando.

    List<Categoria> categorias;

    try {
        produto.setFarmacia(farmaciaService.buscaFarmaciaToken(headers));

        categorias = produto.getCategoria();

        List<Categoria> listaCategorias = new ArrayList<>();
        for(int i = 0; i < categorias.size(); i++) {
            listaCategorias.add( categoriaService.buscarCategoriaId(categorias.get(i).getId()) );
        }

        produto.setCategoria(listaCategorias);

    } catch (Exception ex) {
        return new ResponseEntity(HttpStatus.INTERNAL_SERVER_ERROR);
    }

    produtoService.cadastrarMedicamento(produto);

    return new ResponseEntity(HttpStatus.CREATED);
}

@RequestMapping(method = RequestMethod.GET, value = "/produto/{id}", produces =
MediaType.APPLICATION_JSON_VALUE)
public ResponseEntity<Medicamentos> mostrarProduto(@PathVariable Long id){

    Medicamentos produto = produtoService.consultarMedicamento(id);

    return new ResponseEntity(produto, HttpStatus.OK);
}

@RequestMapping(method = RequestMethod.GET, value = "/produto", produces =
MediaType.APPLICATION_JSON_VALUE)
public ResponseEntity<Medicamentos> mostrarTodosProduto(){
    List<Medicamentos> produtos = produtoService.buscaTodosProdutos();

    return new ResponseEntity(produtos, HttpStatus.OK);
}

@RequestMapping(method = RequestMethod.GET, value = "/menor/produto", produces =
MediaType.APPLICATION_JSON_VALUE)
public ResponseEntity<Medicamentos> ordenarPorMenorPreco(String nome, String principioAtivo){
    List<Medicamentos> produtos = produtoService.ordenarPorMenorPreco(nome, principioAtivo);

    return new ResponseEntity(produtos, HttpStatus.OK);
}

@RequestMapping(method = RequestMethod.PUT, consumes =
MediaType.APPLICATION_JSON_VALUE, value = "/adminAut/produto")
public ResponseEntity editaProduto(@RequestBody Medicamentos produto, @RequestHeader HttpHeaders
headers){

```

```

Medicamentos medicamento;
try{
    produto.setFarmacia(farmaciaService.buscaFarmaciaToken(headers));

    medicamento = produtoService.consultarMedicamento(produto.getId());
    List<Imagem> imagens = new ArrayList<>();
    imagens = medicamento.getImagens();

    if(produto.getImagens() == null || produto.getImagens().isEmpty()) {

        if(imagens != null && !imagens.isEmpty()) {
            produto.setImagens(imagens);
        }
    }
} catch(Exception ex) {
    return new ResponseEntity(HttpStatus.INTERNAL_SERVER_ERROR);
}

produtoService.alterarMedicamento(produto);

return new ResponseEntity(HttpStatus.OK);
}
@RequestMapping(method = RequestMethod.PUT, consumes =
MediaType.APPLICATION_JSON_VALUE, value = "/adminAut/ativarproduto")
public ResponseEntity ativarMedicamento(@RequestBody Medicamentos medicamento,@RequestHeader
HttpHeaders headers){

    Farmacia farmacia;
    try{

        farmacia = farmaciaService.buscaFarmaciaToken(headers);

    } catch(Exception ex) {
        return new ResponseEntity(HttpStatus.INTERNAL_SERVER_ERROR);
    }

    produtoService.alterarMedicamento(medicamento);

    return new ResponseEntity(HttpStatus.OK);
}

@RequestMapping(method = RequestMethod.DELETE,consumes =
MediaType.APPLICATION_JSON_VALUE, value = "/adminAut/produto/{id}")
public ResponseEntity excluirProduto(@PathVariable Long id, @RequestHeader HttpHeaders headers){

    Farmacia adm;
    try {

        adm = farmaciaService.buscaFarmaciaToken(headers);
    } catch (Exception ex) {
        return new ResponseEntity(HttpStatus.INTERNAL_SERVER_ERROR);
    }

    produtoService.desabilitarMedicamento(id);

    return new ResponseEntity(HttpStatus.OK);}
}

```



## **6 CONSIDERAÇÕES FINAIS**

O objetivo geral deste projeto que foi desenvolver um sistema mobile compatível com Android que contribuísse para a comparação de preços de medicamentos das farmácias de Iporá – GO foi atingido, resultando em um aplicativo que poderá contribuir de forma significativa nesse processo.

Para que o desenvolvimento deste projeto fosse possível fez-se necessário um conhecimento multidisciplinar, envolvendo disciplinas como: Análise e Modelagem de Sistemas, Programação Orientada a Objetos I e II, Desenvolvimento para Dispositivos Móveis I e II, Programação para Web I e II e Desenvolvimento de Sistemas com Framework. Os conhecimentos adquiridos em tais disciplinas por si só possibilitaram grande parte do desenvolvimento, cabendo apenas buscar alguns conhecimentos específicos quanto ao uso de algumas bibliotecas e ou componentes mais específicos utilizados pelo sistema.

Através de pesquisas foi possível elicitare requisitos e identificar tecnologias que serviram de apoio no desenvolvimento deste projeto, como, definir a plataforma que melhor atenderá ao usuário escolhido, no caso a plataforma para dispositivos móveis, observando que existe um crescimento constante de usuários.

Inicialmente a proposta do trabalho era composta por uma avaliação realizada pelo usuário do sistema mobile às respectivas farmácias onde encontrasse o menor preço. Porém no decorrer do trabalho foram encontradas algumas dificuldades em como vincular o cliente à farmácia, já que o cliente apenas realiza consultas de medicamentos.

Como sugestão para trabalhos futuros, propõe-se que o sistema mobile contenha um botão que possibilite aos clientes realizarem ligações para as farmácias, assim como também o desenvolvimento de um método de avaliação das farmácias pelos clientes. Uma integração do sistema web ao sistema da farmácia, facilitando assim o cadastro e a atualização de preços dos medicamentos, além também de uma integração do sistema web com uma API da Receita Federal para validação de CNPJ.

## REFERÊNCIAS

AFONSO, Alexandre. O que é Spring Boot? [S. l.], 2017. Disponível em: <https://blog.algaworks.com/spring-boot/>. Acesso em: 20 jan. 2020.

CARDOSO, Andrey. O que é o MariaDB ? [S. l.], 23 dez. 2015. Disponível em: <https://www.bravulink.com.br/o-que-e-o-mariadb/>. Acesso em: 15 ago. 2019.

CARVALHO, Lucas. Android cresce no Brasil e aumenta distância para iOS e Windows Phone. [S. l.], 2016 ou 2017. Disponível em: <https://olhardigital.com.br/noticia/Android-cresce-no-brasil-e-aumenta-distancia-para-ios-e-windows-phone/68023>. Acesso em: 10 mar. 2019.

CHAVES, Ana Maria Siqueira. Técnicas de extração de requisitos. [S. l.], 1 mar. 2017. Disponível em: <https://www.tiespecialistas.com.br/tecnicas-de-extracao-de-requisitos>. Acesso em: 10 mar. 2019.

CORDEIRO, Fellipe. 06 passos essenciais para começar com Android Studio. [S. l.], 2017. Disponível em: [https://www.androidpro.com.br/blog/android-studio/android-studio-passos-essenciais-para-comecar/#disqus\\_thread](https://www.androidpro.com.br/blog/android-studio/android-studio-passos-essenciais-para-comecar/#disqus_thread). Acesso em: 20 jan. 2020.

DEMARTINI, Felipe. Brasil já tem mais de um smartphone ativo por habitante. [S. l.], 20 abr. 2018. Disponível em: <https://canaltech.com.br/produtos/brasil-ja-tem-mais-de-um-smartphone-ativo-por-habitante-112294/>. Acesso em: 5 mar. 2019.

ESTRELLA, Carlos. O Que é PHP? Guia Básico de Programação PHP. [S. l.], 27 jun. 2019. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-php-guia-basico/>. Acesso em: 15 ago. 2019.

FERNANDES, André. O que é API? Entenda de uma maneira simples. Rio de Janeiro, 1 mar. 2018. Disponível em: <https://vertigo.com.br/o-que-e-api-entenda-de-uma-maneira-simples/>. Acesso em: 14 ago. 2019.

FERREIRA, Débora Cristina. Web Service: Arquitetura REST. *In*: FERREIRA, Débora Cristina. APISIM - Uma API RESTful para o gerenciamento de recursos de sistemas operacionais. Orientador: Odilon Correa da Silva. 2017. Monografia (Bacharel em Engenharia da Computação) - Centro Federal de Educação Tecnológica de Minas Gerais, Timóteo-MG, 2017. p. 58. Disponível em: [http://sistemas.timoteo.cefetmg.br/nos/media/bd:tcc:ec:2017:2017\\_ferreira.pdf](http://sistemas.timoteo.cefetmg.br/nos/media/bd:tcc:ec:2017:2017_ferreira.pdf). Acesso em: 22 set. 2019.

FERREIRA, Érico Dias. Framework CSS Bootstrap. In: FERREIRA, Érico Dias. Desenvolvimento de um sistema para o gerenciamento do processo de trabalho de conclusão de curso do curso de Tecnologia em Sistemas para Internet da UTFPR Campus Guarapuava. Orientador: Prof. Dr. Diego Marczal. 2015. Monografia (Superior em Tecnologia em Sistemas para Internet) - Universidade Tecnológica Federal do Paraná - UTFPR - Câmpus Guarapuava, Guarapuava, 2015. p. 55. Disponível em: [https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/28/GP\\_COINT\\_2015\\_1\\_10.pdf?1455720007](https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/28/GP_COINT_2015_1_10.pdf?1455720007). Acesso em: 7 out. 2019.

FREIRE, Raquel. Compare os preços de remédios e economize na conta da farmácia. [S. l.], 3 nov. 2017. Disponível em: <https://www.techtudo.com.br/tudo-sobre/consulta-remedios.html>. Acesso em: 5 mar. 2019.

GUEDES, Marylene. 5 motivos para estudar Java. São Paulo, 25 set. 2019. Disponível em: <https://www.treinaweb.com.br/blog/5-motivos-para-estudar-java/>. Acesso em: 20 jan. 2020.

HIGA, Paulo. 95,5% dos smartphones vendidos no Brasil são Androids. [S. l.], 2016. Disponível em: <https://tecnoblog.net/203749/Android-ios-market-share-brasil-3t-2016/>. Acesso em: 13 mar. 2019.

IMPRESA MERCADO & CONSUMO. Aplicativo realiza comparação de preços de medicamentos. [S. l.], 13 abr. 2018. Disponível em: <https://www.mercadoeconsumo.com.br/2018/04/13/aplicativo-realiza-comparacao-de-precos-de-medicamentos/>. Acesso em: 5 mar. 2019.

MARINHO, Paulo R. Como funciona o Hibernate. [S. l.], 7 jan. 2014. Disponível em: <http://www.webcodefree.com.br/blog/?p=1050>. Acesso em: 20 jan. 2020.

OFICINA, Redação. O que é o NetBeans?. [S. l.], 2008. Disponível em: [https://www.oficinadanet.com.br/artigo/1061/o\\_que\\_e\\_o\\_netbeans](https://www.oficinadanet.com.br/artigo/1061/o_que_e_o_netbeans). Acesso em: 20 jan. 2020.

PAGOTTO, Tiago et al. Scrum solo: Processo de software para desenvolvimento individual. In: 11ª CONFERÊNCIA IBÉRICA DE SISTEMAS E TECNOLOGIAS DA INFORMAÇÃO (CISTI), 2016, Las Palmas. Scrum solo: Processo de software para desenvolvimento individual [...]. Espanha: IEEE, 2016. Disponível em: <https://ieeexplore.ieee.org/document/7521555>. Acesso em: 21 mar. 2019.

PIRES, Jackson. O que é API? REST e RESTful? Conheça as definições e diferenças!. [S. l.], 2017. Disponível em: <https://becode.com.br/o-que-e-api-rest-e-restful/>. Acesso em: 14 ago. 2019.

PROCON Goiás divulga pesquisa de preços de medicamentos de referência e genéricos. Goiânia, 2017. Disponível em: <https://www.procon.go.gov.br/noticias/procon-goias-divulga-pesquisa-de-precos-de-medicamentos-de-referencia-e-genericos.html>. Acesso em: 4 mar. 2019.

ROUSE, Margaret. URL (Uniform Resource Locator). [S. l.], 2019. Disponível em: <https://searchnetworking.techtarget.com/definition/URL>. Acesso em: 20 jan. 2020.

RPA, Redação. Aplicativo gratuito ajuda a encontrar os melhores preços em medicamentos. [S. l.], 27 jul. 2018. Disponível em: <https://razoesparaacreditar.com/aplicativo-melhores-precos-medicamentos/>. Acesso em: 20 jan. 2020.

TUDO o que você precisa saber sobre o Spring Boot. [S. l.], 24 jun. 2019. Disponível em: <https://blog.geekhunter.com.br/tudo-o-que-voce-precisa-saber-sobre-o-spring-boot/>. Acesso em: 15 ago. 2019.

VIEIRA, Marcelo de Mello. Aspectos Jurídicos dos medicamentos: um diálogo entre o direito à saúde e o direito do consumidor. In: XXV ENCONTRO NACIONAL DO CONPEDI, 2016, BRASÍLIA/DF. Direito Civil Constitucional [...]. 2016: [s. n.], 2016. Disponível em: <https://www.conpedi.org.br/publicacoes/y0ii48h0/kvg8f9o7/4KMiedtITeLp6Rjx.pdf>. Acesso em: 4 mar. 2019.