

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SUL-RIO-GRANDENSE - CÂMPUS PASSO FUNDO  
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET**

**MATEUS PIMENTEL DE OLIVEIRA**

**APLICATIVO PARA CONSULTA DE DISPONIBILIDADE DE  
MEDICAMENTOS DESENVOLVIDO COM XAMARIN.FORMS**

**Prof. Dr. João Mário Lopes Brezolin**

**PASSO FUNDO  
2020**

**MATEUS PIMENTEL DE OLIVEIRA**

**APLICATIVO PARA CONSULTA DE DISPONIBILIDADE DE  
MEDICAMENTOS DESENVOLVIDO COM XAMARIN.FORMS**

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador (a): Prof. Dr. João Mário Lopes  
Brezolin

**PASSO FUNDO**

**2020**

**MATEUS PIMENTEL DE OLIVEIRA**

**APLICATIVO PARA CONSULTA DE DISPONIBILIDADE DE  
MEDICAMENTOS DESENVOLVIDO COM XAMARIN.FORMS**

Trabalho de Conclusão de Curso aprovado em 09/12/2020 como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

---

João Mário Lopes Brezolin

---

Élder Francisco Fontana Bernardi

---

Vanessa Lago Machado

---

Rafael Marisco Bertei

**PASSO FUNDO  
ANO DE CONCLUSÃO**

## RESUMO

O presente estudo teve como objetivo o desenvolvimento de um protótipo de aplicativo que permitisse aos usuários realizar consultas sobre a disponibilidade de medicamentos na central de distribuição municipal através de um dispositivo móvel sem a necessidade de seguir uma fila física para tal e evitando deslocamento desnecessário, afinal muitas vezes os medicamentos podem estar em falta. O protótipo do aplicativo foi desenvolvido utilizando o framework Xamarin.Forms na linguagem C#, fazendo comunicação através do uso de serviços web. também desenvolvidos em C# com o framework .Net Core. Após a implementação do mesmo ocorreu o processo de validação junto aos usuários da farmácia. O aplicativo mostrou-se adequado fornecendo um método para que quando um medicamento não estivesse disponível os usuários pudessem compartilhar entre si os medicamentos ajudando aqueles que não teriam outra forma de adquirir ao mesmo. Quanto ao Xamarin.Forms mostrou-se eficiente, cumprindo o objetivo de facilitar a criação de um aplicativo mobile multiplataforma nativo.

Palavras-chave: .Net Core, C#, Entity Framework, Xamarin.Forms.

## **ABSTRACT**

The present study aimed to develop an application prototype that would allow users to make inquiries about the availability of medicines at the municipal distribution center through a mobile device without having to follow a physical queue for this and avoiding unnecessary displacement, after all medications can often be in short supply. The application prototype was developed using the Xamarin.Forms framework in the C# language, making communication through the use of web services, also developed in C# with the .Net Core framework. After its implementation, the validation process took place with users of the pharmacy. The application proved to be adequate by providing a method so that when a medication was not available, users could share the medications among themselves, helping those who would have no other way to acquire it. As for Xamarin.Forms, it proved to be efficient, fulfilling the objective of facilitating the creation of a native multiplatform mobile application.

Keywords: .Net Core, C#, Entity Framework, Xamarin.Forms.

## LISTA DE FIGURAS

Figura 1 - Esquema da plataforma Xamarin.	13
Figura 2 - Esquema da Arquitetura MVVM.	14
Figura 3 - Formato de String Json.	15
Figura 4 - Representação do Diagrama de Casos de Uso.	20
Figura 5 - Diagrama de Casos de Uso.	21
Figura 6 - Representação da estrutura da comunicação.	22
Figura 7 - Representação do Diagrama de Componentes.	23
Figura 8 - Representação do Diagrama de Casos de Uso do Protótipo.	24
Figura 9 - Representação do Diagrama de Caso de Uso do Serviço Web .	28
Figura 10 - Representação do Diagrama de Classes do protótipo.	30
Figura 11 - Estrutura do projeto PC2 App.	32
Figura 12 - Representação da Estrutura do Projeto Web Service.	33
Figura 13 - Tela de Login.	34
Figura 14 - Tela de Listagem de Medicamentos.	35
Figura 15 - Tela de Contatos.	36

## LISTA DE TABELAS

Tabela 1 – Caso de uso realizar login.	24
Tabela 2 – Caso de uso Consultar Disponibilidade.	25
Tabela 3 – Caso de uso Pedido de Doação.	26
Tabela 4 – Caso de uso Realizar Processo de Doação.	27
Tabela 5 – Caso de uso Capturar Dados.	28
Tabela 6 – Caso de uso Fornecer Dados.	29

## LISTA DE ABREVIATURAS E SIGLAS

API – *Application Programming Interface*

APP – Aplicativo

C# – C-Sharp

CLR – Common Language Runtime

DLL – Dynamic-link library

FCL – Framework Class Library

HTTP – Hypertext Transfer Protocol

IFSUL – Instituto Federal Sul-rio-grandense

MVVM – Model View ViewModel

PaaS – Platform as a service

PC2 – Projeto de Conclusão 2

PCL – Portable Class Library

PYPL – PopularitY of Programming Language Index

SAP – Shared Asset Project

SUS – Sistema Único de Saúde

TCC – Trabalho de Conclusão de Curso

TXT – Nome dado a extensão de arquivos de texto

XML – eXtensible Markup Language

WPF – Windows Presentation Foundation



## SUMÁRIO

<b>INTRODUÇÃO</b>	<b>8</b>
<b>OBJETIVOS</b>	<b>9</b>
<b>ESTRUTURA DA MONOGRAFIA</b>	<b>9</b>
<b>REFERENCIAL TEÓRICO</b>	<b>11</b>
<b>DISTRIBUIÇÃO DE MEDICAMENTOS</b>	<b>11</b>
<b>XAMARIN E XAMARIN.FORMS</b>	<b>11</b>
<b>MVVM</b>	<b>14</b>
<b>JSON</b>	<b>15</b>
<b>SERVIÇOS WEB</b>	<b>16</b>
<b>.NET E .NET CORE</b>	<b>16</b>
<b>C# (C-SHARP)</b>	<b>17</b>
<b>MICROSOFT SQL SERVER</b>	<b>18</b>
<b>TRABALHOS RELACIONADOS</b>	<b>18</b>
<b>IMPLEMENTAÇÃO DO SISTEMA</b>	<b>20</b>
<b>ESCOPO DE APLICAÇÃO DO APLICATIVO DESENVOLVIDO</b>	<b>20</b>
<b>REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS DO SISTEMA</b>	<b>21</b>
<b>DESCRIÇÃO DO DIAGRAMA DE CASOS DE USO DO PROTÓTIPO</b>	<b>23</b>
<b>DESCRIÇÃO DO DIAGRAMA DE CASOS DO SERVIÇO WEB</b>	<b>28</b>
<b>DIAGRAMA DE CLASSES</b>	<b>30</b>
<b>AMBIENTE DE DESENVOLVIMENTO</b>	<b>31</b>
<b>ESTRUTURA DO PROJETO DO PROTÓTIPO DE APLICATIVO</b>	<b>31</b>
<b>ESTRUTURA DO PROJETO DO SERVIÇO WEB</b>	<b>32</b>
<b>FUNCIONAMENTO DO SISTEMA DESENVOLVIDO</b>	<b>33</b>
<b>Processo de Login no PC2 App</b>	<b>33</b>
<b>Listagem de Medicamentos no PC2 App</b>	<b>34</b>
<b>Pedido Doação e Iniciar Processo de Doação no PC2 App</b>	<b>35</b>
<b>Comunicação entre Usuários pela Tela de Contatos no PC2 App</b>	<b>36</b>
<b>VALIDAÇÃO DE USABILIDADE</b>	<b>38</b>
<b>CONCLUSÃO</b>	<b>40</b>
<b>REFERÊNCIAS</b>	<b>41</b>

## 1 INTRODUÇÃO

Em Passo Fundo, a distribuição gratuita de medicamentos é feita pelas farmácias municipais. Hoje, porém, não existe nenhum sistema de acesso público para verificar se um determinado medicamento está ou não disponível para retirada. E caso o medicamento não esteja disponível, não há nenhuma outra forma de obtê-lo de forma gratuita. Nesse sentido, observou-se que ocorrem muitos deslocamentos ineficazes até a farmácia, afinal além de ter dificuldade em conseguir a informação seja por lista de esperas ou horários limitantes de atendimento, muitas vezes ao tentar retirar um medicamento o mesmo não consta em estoque. Nesse sentido, visando sanar esses problemas esse estudo propõe a implementação de protótipo de sistema *mobile* multiplataforma. Com a popularização dos smartphones e o avanço da tecnologia de rede da telefonia móvel esses sistemas têm se tornado acessíveis para muitos usuários. Segundo a Fundação Getúlio Vargas (2020) o número de smartphones no Brasil já é superior ao número de computadores. Além disso, já existem mais smartphones do que pessoas no Brasil, cerca de 1,116 aparelhos por pessoa, existindo cerca de 234 milhões de aparelhos (Fundação Getúlio Vargas, 2020). Entre os principais sistemas operacionais nos smartphones, se destacam no número de vendas o Android da empresa Google e o iOS da Apple (Statista, 2020). Cada um deles trabalha de maneira distinta com seus aplicativos e suas funcionalidades, assim cada uma das plataformas é feita através de uma linguagem de programação diferente (COSTA e FILHO, 2013).

Para que seja possível criar aplicações que funcionem em diversas plataformas, é abordada muitas vezes a programação híbrida, porém esta ocasiona uma perda no desempenho das aplicações (HOLZER e ONDRUS, 2012). Para resolver esse problema surgiu o framework Xamarin.Forms que tem como intuito principal tornar possível uma programação multiplataforma, que possa ter acesso às diversas APIs nativas nos diferentes sistemas operacionais e ainda garantir uma alta taxa de compartilhamento de código (MICROSOFT, 2020d). O Xamarin.Forms é um conjunto de ferramentas voltadas para o desenvolvimento de aplicativos móveis. Com ele é possível desenvolver aplicativos móveis nativos utilizando uma única linguagem de programação, neste caso o C#, e compartilhar o mesmo código através das diversas arquiteturas distintas.

Dessa forma, fazendo uso desse framework é possível fazer o desenvolvimento de um protótipo de aplicativo mobile nativo multiplataforma, compatível nos principais sistemas operacionais disponíveis nos smartphones (Android, iOS) e possua um acesso a diversas funções que o hardware do dispositivo disponibiliza. Assim, os usuários da farmácia poderão ter amplo acesso ao aplicativo, que pode ser instalado na maioria dos sistemas operacionais *mobile*.

Com esse sistema desenvolvido foi realizada uma validação com usuários que fazem uso da distribuição gratuita de medicamentos, nesse processo observou-se que o aplicativo está adequado às necessidades dos usuários.

## **1.1 OBJETIVOS**

Esse estudo tem como objetivo desenvolver um protótipo para auxiliar seus usuários a monitorar a disponibilidade de medicamentos da farmácia municipal e estreitar a comunicação entre os usuários visando criar um meio para que os mesmos possam auxiliar um ao outro quando o medicamento não estiver disponível.

### **2.1 OBJETIVOS ESPECÍFICOS**

- Realizar a revisão bibliográfica sobre o tema da pesquisa.
- Pesquisar os recursos da plataforma Xamarin.Forms.
- Implementação do aplicativo.
- Realizar a validação do aplicativo junto a usuários da farmácia municipal.

## **1.2 ESTRUTURA DA MONOGRAFIA**

Visando a melhor organização e estruturação, esta monografia foi dividida nos seguintes capítulos: Capítulo 2 (Referencial Teórico): apresenta as tecnologias estudadas que foram necessárias para realizar o desenvolvimento do protótipo do aplicativo proposto. Capítulo 3 (Implementação do sistema): Apresenta o desenvolvimento da solução proposta Serviço Web e APP. Capítulo 4 (Validação): Descreve os resultados obtidos a partir do retorno dos questionários de validação.

Capítulo 5 (Conclusão): Apresenta o resultado do projeto desenvolvido e propõe trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Essa seção apresenta a revisão das tecnologias que serão utilizadas no desenvolvimento da aplicação proposta.

### 2.1 DISTRIBUIÇÃO DE MEDICAMENTOS

A Política Nacional de Medicamentos (BRASIL, 2001) busca constituir os elementos necessários para a efetiva implementação de ações capazes de promover a melhoria das condições da assistência à saúde da população. Segundo essa política, é função das esferas federal, estadual e municipal, através do SUS, garantir a necessária segurança, eficácia e qualidade dos medicamentos, a promoção do uso racional e o acesso da população àqueles considerados essenciais. Nos municípios o acesso aos medicamentos é realizado pelas farmácias municipais. Uma vez diagnosticados, os usuários da farmácia podem retirar periodicamente os medicamentos essenciais na farmácia municipal. Entretanto, muitas vezes ocorre a falta de medicamentos decorrendo em filas e no deslocamento desnecessário de pessoas até a farmácia. Apesar da disponibilidade de medicamentos poder ser consultada a partir de consultas telefônicas, a demanda muitas vezes impede o correto acesso às informações causando transtornos desnecessários aos usuários. Nesse sentido, o aplicativo proposto busca estabelecer um meio para a melhor disponibilização de informações e evitar transtornos aos usuários da farmácia.

### 2.2 XAMARIN E XAMARIN.FORMS

Xamarin é uma suíte de desenvolvimento que possibilita o desenvolvimento de aplicativos multiplataforma através de um ambiente unificado. Ao invés de escrever códigos em linguagens específicas para cada sistema, ela permite que desenvolvedores escrevam códigos apenas na linguagem C#. A suíte gera soluções multiplataforma através de compilação cruzada, o que significa que serão geradas aplicações nativas para as plataformas Android, Windows Phone, iOS.

Para o desenvolvimento de aplicações através do Xamarin.Forms recomenda-se o uso da arquitetura MVVM (*Model-View-ViewModel*) que permite estabelecer a separação de códigos entre dados (*Model*) e interfaces de usuário

(*View*) e especifica a interface entre ambos (*ViewModel*). Dessa forma, torna-se possível separar o código em código específico da plataforma: *View*, que interage com a API da plataforma, e código independente: *Model* e *ViewModel*. O código específico precisa ter acesso a arquivos, rede, ou usar threads. Essas tarefas nas soluções totalmente nativas fazem parte de uma API do sistema operacional, porém no Xamarin elas podem também fazer parte da biblioteca .NET Framework (MICROSOFT, 2020d). Esta biblioteca também permite que sejam utilizadas funções *async*, *await* ou expressões *lambdas* (QUAIATO, 2016) que facilitam diversas tarefas no desenvolvimento, outra característica importante da biblioteca .NET é que ela está disponível em cada plataforma de maneira independente, sendo ela efetivamente multiplataforma dentro do Xamarin (DE LA TORRE, 2016).

As codificações independentes de plataforma, como as classes responsáveis pelas regras de negócio podem ser isoladas em um projeto que contém o código compartilhado entre as três plataformas. Isto pode ser realizado através de um *Shared Asset Project* (SAP), que consiste em códigos e outros arquivos acessíveis a outros projetos, ou um *Portable Class Library* (PCL), que transforma todo código comum em uma biblioteca de vínculo dinâmico (DLL) que pode então ser referenciada a partir de outros projetos (MACORATTI, 2015).

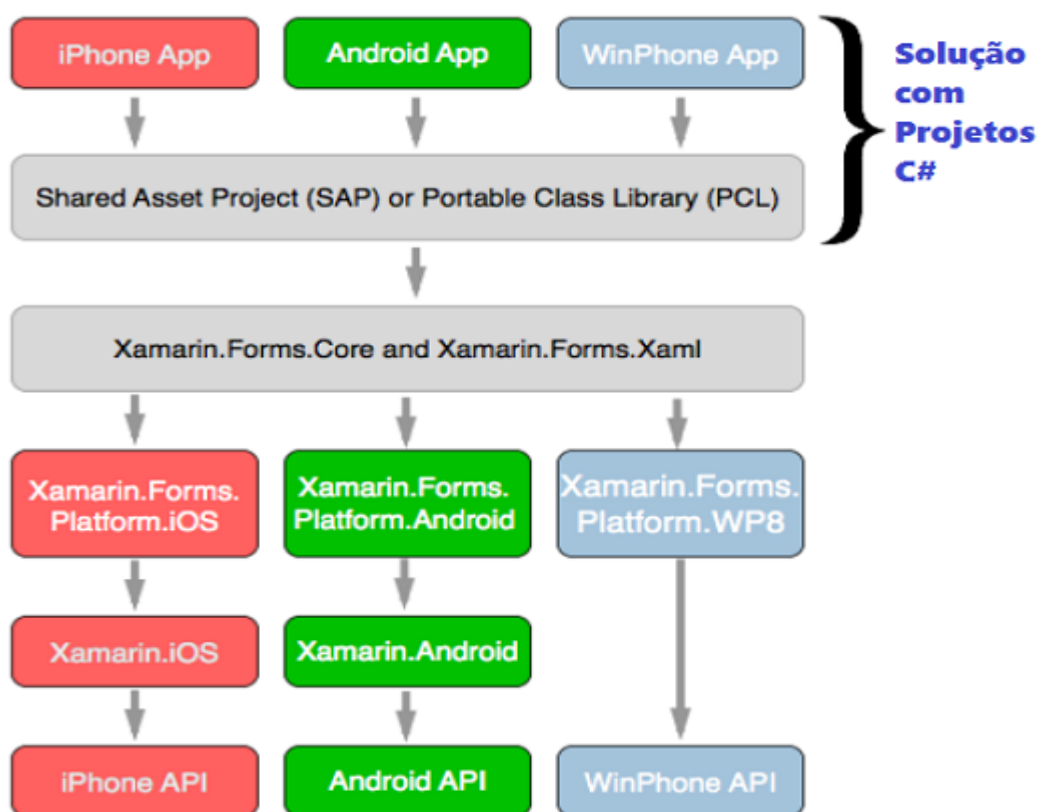
No Xamarin todos os controles e APIs usados quando compilados, são referenciados como os nativos da plataforma, sendo possível ter o acesso total às APIs da plataforma subjacente. Isto oferece uma usabilidade e um desempenho muito próximo daqueles desenvolvidos de forma completamente nativa (MICROSOFT, 2020d). A arquitetura das aplicações construídas com o Xamarin.Forms é muito semelhante às que outras aplicações multiplataformas utilizam (MACORATTI, 2015). Cerca de 75% do código de um projeto pode ser compartilhado entre os aplicativos utilizando essa tecnologia (MICROSOFT, 2020d). Segundo Barth (2014) isso proporciona satisfação em relação à usabilidade e a facilidade de acesso às funções e componentes. Provedi (2016) destaca que pode ser possível alcançar uma grande reutilização, fazendo com que a taxa de reaproveitamento chegue próximo ao 100%, utilizando os códigos das interfaces de usuários de maneira compartilhada entre os projetos com o Xamarin.Forms.

Xamarin.Forms permite escrever códigos de interface de usuários que podem ser compilados para iOS, Android e Windows Phone. De uma maneira geral, uma aplicação criada com essa tecnologia possui três projetos separados, um para cada

plataforma. Esses códigos, segundo Macoratti (2015), muitas vezes são apenas de código de inicialização, e um quarto projeto que contém o código compartilhado entre as plataformas. O projeto compartilhado contém a maior parte do aplicativo, inclusive as interfaces de usuários (PETZOLD, 2016).

O framework do Xamarin possui duas importantes bibliotecas, a Xamarin.Forms.Core e Xamarin.Forms.Xaml. De acordo com cada plataforma, o Xamarin.Forms.Core faz o uso de uma das bibliotecas Xamarin.Forms.Platform fazendo uma ligação entre as bibliotecas do framework com as bibliotecas dos sistemas operacionais específicos. Por exemplo, se a aplicação for compilada para o sistema Android, o framework fará uso das bibliotecas exclusivamente nativas desse sistema. Caso a aplicação seja destinada para a plataforma Windows, a mesma fará uso das bibliotecas específicas para o Windows Phone. Enquanto o Xamarin.Forms.Xaml cuida das bibliotecas que representam coleções de classes chamadas renderizadores, as quais transformam os objetos da interface do usuário Xamarin.Forms na interface do usuário específica do sistema a ser atendido, seja ele qual for (PETZOLD, 2016), como mostrado na Figura 1.

Figura 1: Esquema da plataforma Xamarin



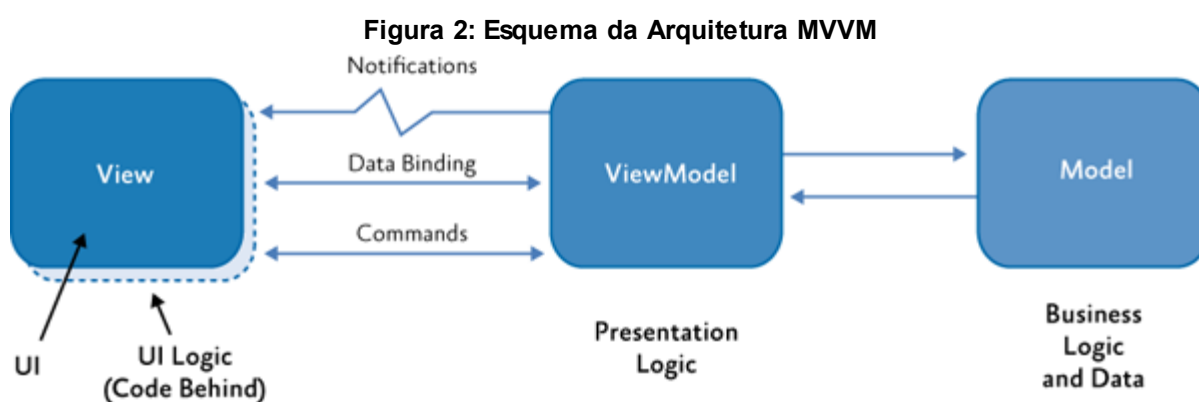
Fonte: Página do Macoratti na Web.

Essa implementação da interface do usuário traz todos os benefícios das implementações nativas e também suas falhas, ou seja, se a interface nativa contiver algum erro, essa API dentro Xamarin.Forms também vai ter esse problema (MICROSOFT, 2020d).

## 2.3 MVVM

O modelo MVVM (*Model-View-ViewModel*) é um padrão de arquitetura de software idealizado em 2005 por John Gossman, originalmente para ser utilizado junto da plataforma WPF (Windows Presentation Foundation). Smith (2009) apresenta essa arquitetura como uma versão mais moderna do modelo MVC (*Model-View-Controller*), em que o foco é ajudar a separar de forma limpa a lógica de negócios e apresentação de um aplicativo de sua interface de usuário (User Interface).

No padrão MVVM, a *View* encapsula a interface do usuário e qualquer lógica UI sendo assim ela é responsável pelo layout da tela, o *Model* encapsula a lógica de negócios e dados podendo incluir modelos de dados juntamente com as regras de negócio e suas validações, enquanto *ViewModel* isola a lógica de apresentação e o estado sendo uma intermediária entre a *View* e a *Model*. As interações entre as partes são ilustradas na Figura 2:



Fonte: Página da Microsoft

No modelo, *View* interage com o *ViewModel* através de *data binding*, comandos, e notificações de mudança. O *ViewModel* por sua vez consulta, observa e coordena atualizações para o modelo, convertendo, validando e agregando dados conforme necessário para exibição na *View* (SMITH, 2009).



## 2.4 JSON

O JSON (JavaScript Object Notation) “é um formato de intercâmbio de dados aberto, baseado em texto derivado do JavaScript” (LENNON, 2009, p. 87). Devido às essas características o formato é extremamente leve e também é independente de linguagem, praticamente qualquer uma das linguagens de programação existentes possuem uma biblioteca que faz a leitura e a interpretação da mesma. Esse tipo de dado é mais utilizado para intercâmbio de informações na web, porém pode ser usado como um formato de dados em qualquer situação em que a informação precisa ser armazenada como texto, como em comunicações HTTP.

Petkovic (2017) descreve que a estrutura do conteúdo JSON segue a estrutura de sintaxe para JavaScript onde, uma String neste formato contém uma matriz de valores ou um objeto, a matriz é formada de pares nome / valor. Esta é cercada por um par de colchetes que contém uma lista de valores separados por vírgulas. Enquanto um objeto é cercado por um par de chaves e contém uma lista separada por vírgulas de pares nome / valor. Um par de nome / valor consiste em um nome de campo (entre aspas duplas), seguido por dois pontos (:), seguido pelo valor do campo (com aspas duplas) (PETKOVIĆ, 2017), como demonstrado na Figura 3.

**Figura 3: Formato de String Json**

```
{“usuario”:{“cpf”:“00000000000”,“sus”:“0000000000”,“medicamentos”:[{“id”:“01”,“descricao”:“Paracetamol500mg”,“status”:“Disponivel”},{“id”:“02”,“descricao”:“Levotiroxina Sódica 50mg”,“status”: “Indisponivel”}]}}
```

**Fonte: (Autoria Própria, 2020).**

Petković (2017) também argumenta que o JSON pode ser organizado de maneira estruturada e relacional como também não relacional. Para que a aplicação se comunique com o serviço web será utilizado esse formato para o intercâmbio de dados.

## 2.5 SERVIÇOS WEB

Serviços Web são soluções utilizadas na integração entre sistemas, união e na comunicação entre aplicações diferentes e segundo Ferreira Filho (2009) garantem

que essa comunicação seja de forma interoperável e independente de plataforma. Ele ainda acrescenta que essa tecnologia representa uma abordagem muito promissora em relação a reutilização de software nos ambientes distribuídos, porque acabam por valorizar os softwares já existentes.

Essa tecnologia permite que o sistema possa acessar uma função ou até mesmo dados que estão dentro de outro sistema, a fim de aprimorar suas próprias funcionalidades ou até mesmo obter informações. Estas chamadas podem enviar ou receber informações em diversos formatos, sendo um deles o JSON (LECHETA, 2015). A utilização de serviços web permite um acesso aos dados independente do sistema operacional, da linguagem de programação, protocolos internos e processador, afinal a comunicação é através de protocolos HTTP (FERREIRA FILHO, 2009). Este conceito trouxe um grande avanço no modo pelo quais os sistemas são construídos, pois permite uma maior flexibilidade (LECHETA, 2015). De maneira resumida serviços web constituem-se em softwares de baixo acoplamento, reutilizáveis, com o intuito de serem facilmente acessados pela internet, e devido a essa versatilidade e a necessidade de acessar as informações disponibilizadas pela farmácia e enviá-las para a aplicação mobile, essa é a tecnologia proposta como o meio para realizar tal ação.

## **2.6 .NET E .NET CORE**

O .NET é uma plataforma de desenvolvimento de software livre e gratuita para a criação de vários tipos de aplicativos entre eles temos Aplicativos Web, APIs da Web e microservices, Funções sem servidor na nuvem, Aplicativos nativos da nuvem, Aplicativos móveis, Aplicativos da área de trabalho, WPF do Windows, Windows Forms, UWP (Plataforma Universal do Windows), Jogos, Internet das Coisas (IoT), Aprendizado de máquina, Aplicativos de console, Serviços do Windows (Microsoft 2020a). A MICROSOFT (2020a) destaca que qualquer aplicativo desenvolvido com o .NET Framework executa código gerenciado em um ambiente de tempo de execução conhecido como Common Language Runtime (CLR, traduzida como Ambiente de Execução Independente de Linguagem) interagindo com uma Framework Class Library (FCL, traduzido como Conjunto de Bibliotecas Unificadas). Ele permite executar diversas linguagens destacando-se o C#, o VB.NET (Visual Basic) e o F#,

dessa forma é possível escrever uma FCL em F# e utilizar ele em uma aplicação C# por exemplo, segundo a MICROSOFT (2020a).

O grande diferencial do .NET Core é que ele é totalmente multiplataforma sendo possível criar aplicativos para diversos sistemas operacionais como Windows, macOS, Linux, Android e iOS (MICROSOFT, 2020a) e também para diversos processadores ex: x64, x86, ARM32 e ARM64. Além disso, é afirmado que “que o .NET CORE permite que você use recursos específicos da plataforma, como APIs do sistema operacional. Os exemplos são Windows Forms e WPF no Windows e as ligações nativas para cada plataforma móvel do Xamarin” (MICROSOFT, 2020a).

## 2.7 C# (C-SHARP)

É uma linguagem de programação que faz parte da estrutura do framework .NET, baseada em C++ e influenciada por outras linguagens como Java, sendo uma linguagem orientada a objetos e fortemente tipada. Hoje é considerada a 4ª linguagem de programação mais popular segundo o PYPL (*Popularity of Programming Language Index*).

Segundo a MICROSOFT (2020b) o .NET framework fornece uma infraestrutura para os tipos de dados básicos do C# (int, double, string e assim por diante) e também oferece suporte para muitas tarefas comuns encontradas em muitos tipos diferentes de programação. Sendo elas: Operações Matemáticas, Depuração do código, Reflexões, Coleções, Globalização, Manipulação de arquivos, Rede, Segurança, Threading, Serviços Web, Manipulação de dados, XML e leitura e escrita JSON (PETZOLD, 2016).

A sintaxe da linguagem C# se assemelha há outras linguagens que tem sua origem no C, como C++ e Java (MICROSOFT, 2020b). Algumas características únicas da linguagem segundo a MICROSOFT (2020b) são:

- O ponto e vírgula (;) é usado para indicar o final de uma declaração ou instrução.
- As chaves ({ }) são usadas para declarações de grupo. As declarações são geralmente agrupadas em métodos (funções), métodos em classes e classes em namespaces.
- As variáveis são atribuídas usando um sinal de igual (=), mas a comparação é feita com dois sinais de iguais consecutivos (==).

- Os colchetes ([ ]) são usados em vetores e listas, tanto na declaração quanto para recuperar o valor em uma determinada posição/índice deles.

## 2.8 MICROSOFT SQL SERVER

O Microsoft SQL SERVER é um Sistema de Gerenciamento de Banco e Dados (SGBD) para banco de dados relacionais desenvolvido pela Microsoft. Um diferencial da versão 2019 da ferramenta é que hoje ela está disponível tanto Windows quanto para Linux (MICROSOFT, 2020c), ela assim como a maioria dos SGBD tem a função de armazenar e recuperar dados. Para desenvolvedores o SQL Server é completamente gratuito em sua versão Developer, para pequenas empresas existe a versão Express, enquanto para grandes corporações existem a versão Enterprise, cada uma trabalha de forma diferente ao que diz respeito a quantia ou carga de trabalho podendo ser algo básico como armazenamento de um sistema com apenas um usuário como também uma grande base de dados com milhares de usuários acessando tudo remotamente e simultaneamente (MICROSOFT, 2020c).

MS-SQL tem entre suas funcionalidades os Gatilhos, Funções, Stored Procedures e Extended Stored Procedures. Existe também uma ferramenta disponibilizada pela Microsoft como *Platform as a Service* (PaaS, traduzido como Plataforma como uma Oferta de Serviços) que é uma espécie de versão baseada em nuvem do Microsoft SQL Server no Microsoft Azure (MICROSOFT, 2020c).

## 2.9 TRABALHOS RELACIONADOS

Softwares e trabalhos acadêmicos com propostas semelhantes foram analisados para que fosse possível idealizar um diferencial para este projeto. Dentre os trabalhos analisados, encontra-se o “Aqui tem Remédio”, um software lançado pela Secretaria Municipal da Saúde de São Paulo (2015). Segundo ela, trata-se de uma ferramenta digital desenvolvida para auxiliar a população a encontrar em quais unidades do serviço público municipal de saúde da cidade se encontram os medicamentos por ela buscados. O sistema possui uma versão Web e uma versão Mobile, são descritos no manual de utilização que os sistemas contêm como principais funcionalidades:

- Busca de medicamentos na rede municipal;

- Quantidade do estoque desse medicamento;
- Localização do posto de saúde mais próxima do usuário;
- Condução a ser utilizada para chegar ao posto de saúde.

Outro trabalho analisado foi “Gestão de uma Farmácia Solidária a partir de um Aplicativo Pautado em Normas, Padrões e Requisitos preconizados pela Sociedade Brasileira de Informática em Saúde” de SADZINSKI et al. (2010), que teve como objetivo desenvolver um protótipo de um aplicativo para uma Farmácia Solidária. Este teve como resultado um protótipo de um aplicativo web que visava atender a demanda e o controle de medicamentos da farmácia solidária. Neste trabalho poucas informações sobre o protótipo resultante são mostradas e devido a isso pouco se pode analisar a respeito de suas funcionalidades.

Sobre as plataformas de uso, o trabalho da Secretaria Municipal da Saúde de São Paulo atende tanto no formato de website quanto como aplicação para dispositivos mobile, enquanto o trabalho de Sadzinski apresenta somente um aplicativo web.

Nos trabalhos estudados, nenhum deles se preocupou em ajudar o usuário caso não houvesse como obter o medicamento, que vem a ser o grande diferencial proposto neste trabalho.

### 3 IMPLEMENTAÇÃO DO SISTEMA

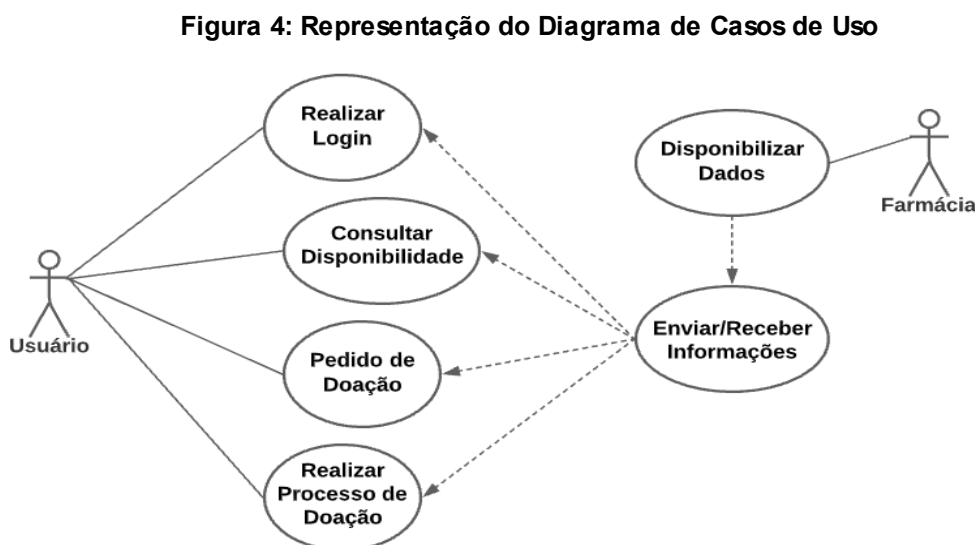
Esta seção descreve o sistema e também o processo de desenvolvimento do mesmo, abordando os diagramas de caso de uso, sua arquitetura, a descrição das classes e a estrutura do projeto, entre outros aspectos relevantes.

#### 3.1 ESCOPO DE APLICAÇÃO DO APLICATIVO DESENVOLVIDO

O estudo teve como base a implementação de um protótipo de aplicação de consulta de disponibilidade de medicamentos na farmácia municipal. O processo é basicamente uma interação entre o usuário e as informações disponibilizadas pela farmácia.

A farmácia deve disponibilizar os dados de medicamentos já retirados pelo usuário e se os mesmos estão disponíveis para retirada, a um serviço web. Os dados sempre que requisitados pelo protótipo devem ser enviados para o mesmo através de comunicações HTTP em formato de JSON, assim nutrindo as informações básicas que o protótipo necessita para o funcionamento.

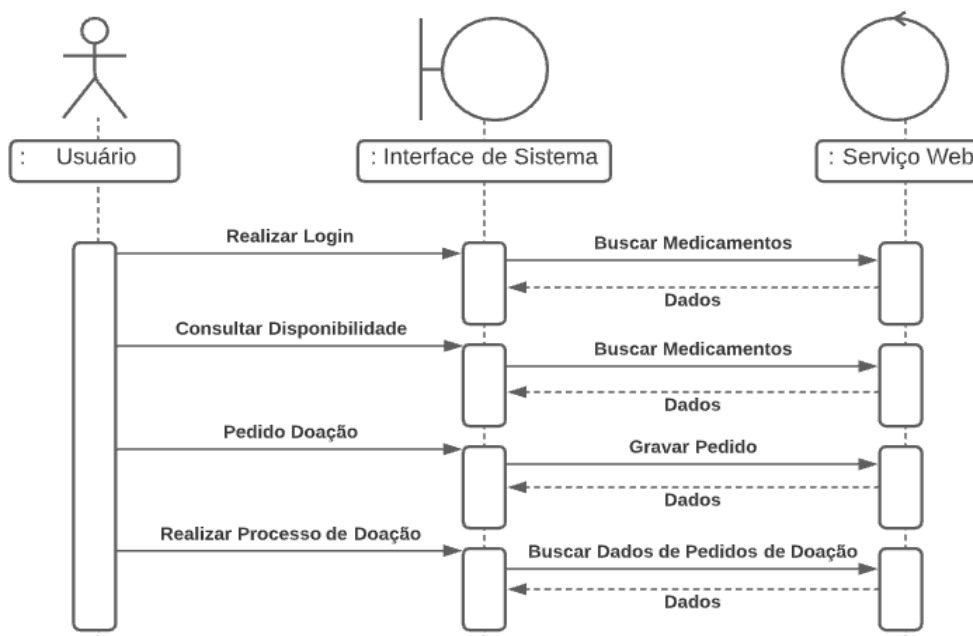
O usuário por sua vez, após informar seus dados pessoais terá acesso à listagem de medicamentos e suas disponibilidades, então poderá realizar um pedido de doação aos demais usuários de um determinado medicamento, como também responder às solicitações de outros usuários realizando um processo de doação. Abaixo segue a Figura 4 que representa o diagrama de casos de uso.



Fonte: (Autoria Própria, 2020).

Para um melhor entendimento do funcionamento do sistema proposto, é apresentado um diagrama de sequência que destaca os casos de uso da figura anterior, conforme Figura 5.

**Figura 5: Diagrama de Casos de Uso**



Fonte: (Autoria Própria, 2020).

- Inicialmente o usuário faz o login e a interface do sistema envia as informações que resultam em uma requisição ao Serviço Web. Nesse processo é feita uma consulta na base de dados e devolve somente as informações dos medicamentos associados aos dados informados;

- A interface então permite que o usuário realize as ações, consultar disponibilidade, dependendo dos dados ainda pode permitir solicitar doação e/ou iniciar processo de doação;

- Essas ações quando realizadas pelo usuário mandam uma requisição ao Serviço Web, que então responde para a interface e a mesma atualiza a informação sobre as ações disponíveis conforme suas regras de negócio.

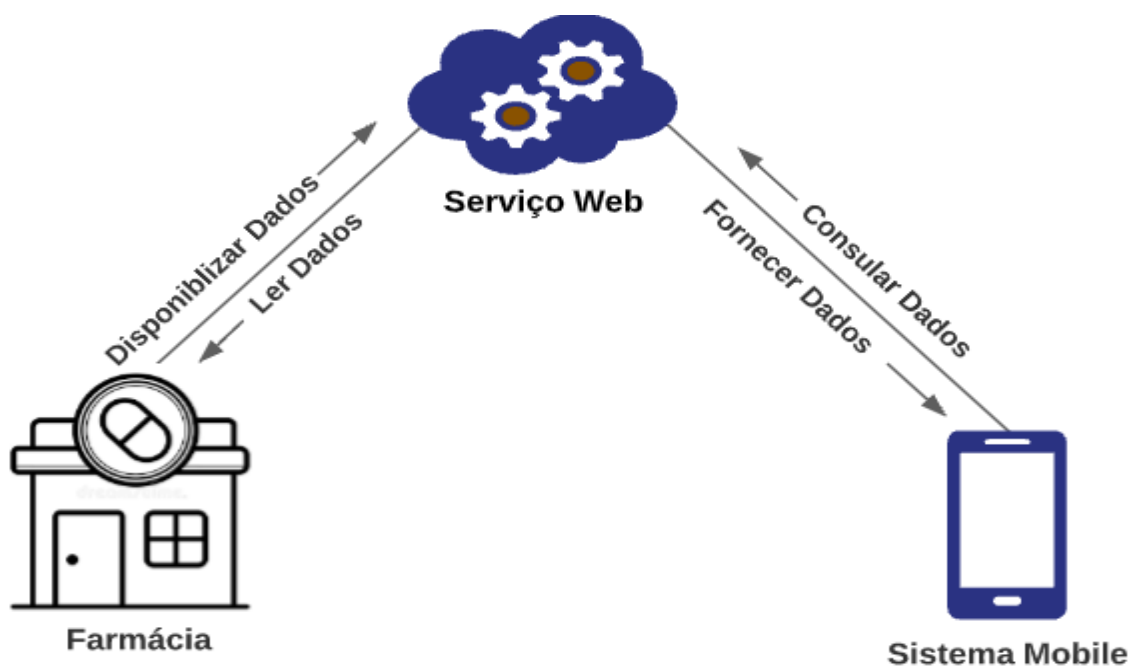
### 3.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS DO SISTEMA

Esse estudo de caso, como já mencionado, trata-se de um protótipo que deve informar a disponibilidade dos medicamentos que um usuário já fez a retirada na

farmácia municipal. Nele o usuário é responsável unicamente por registrar seus pedidos de doação e atender a pedidos de doação de outros usuários no aplicativo, e um serviço web é responsável por prover as informações disponibilizadas pela farmácia para que o protótipo da aplicação funcione.

A Figura 6 apresenta um esboço da comunicação entre os componentes propostos.

**Figura 6: Representação da estrutura da comunicação**



Fonte: (Autoria Própria, 2020).

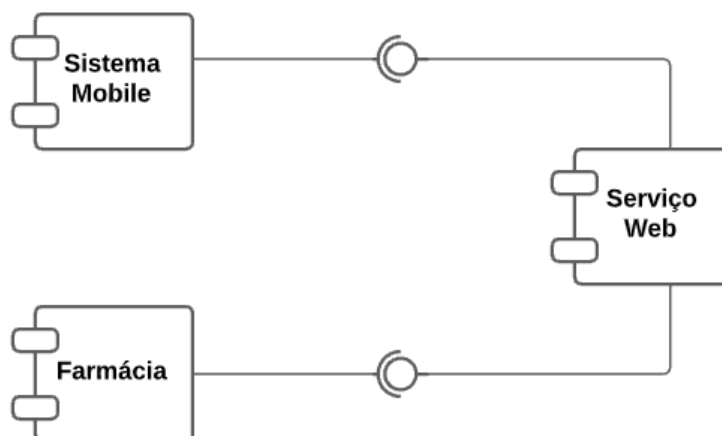
Os componentes do sistema são descritos a seguir:

- Sistema Mobile: É a representação do Protótipo responsável pelas interações com o usuário, também é ele que faz as requisições de dados para o Serviço Web;
- Serviço Web: Esse componente é o responsável por fazer a comunicação da fonte de dados disponibilizados pela farmácia com o protótipo mobile.
- Farmácia: Esse componente representa a fonte de dados que será disponibilizada pela farmácia, onde estão contidas as informações sobre os medicamentos e usuários do sistema.

A comunicação dos componentes também é demonstrada através de um diagrama de componentes na Figura 7.



**Figura 7: Representação do Diagrama de Componentes**



Fonte: (Autoria Própria, 2020).

A Farmácia deve conseguir disponibilizar os dados em qualquer formato compatível com os meios de leitura disponíveis no sistema do Serviço Web, seja um arquivo XML, TXT, uma conexão direta com o banco local entre outras. Os requisitos funcionais do Serviço Web são:

Consultar e obter dados da farmácia;

- Enviar dados essenciais para o funcionamento do protótipo.

Os requisitos funcionais para o protótipo são:

- Visualizar os dados recebidos do Serviço Web;
- Enviar solicitações para o Serviço Web a fim de armazenar registros de pedidos de doação e iniciar o processo de doação;
- Exibir informações sobre a disponibilidade dos medicamentos.
- Viabilizar a comunicação entre os usuários para o processo de doação.

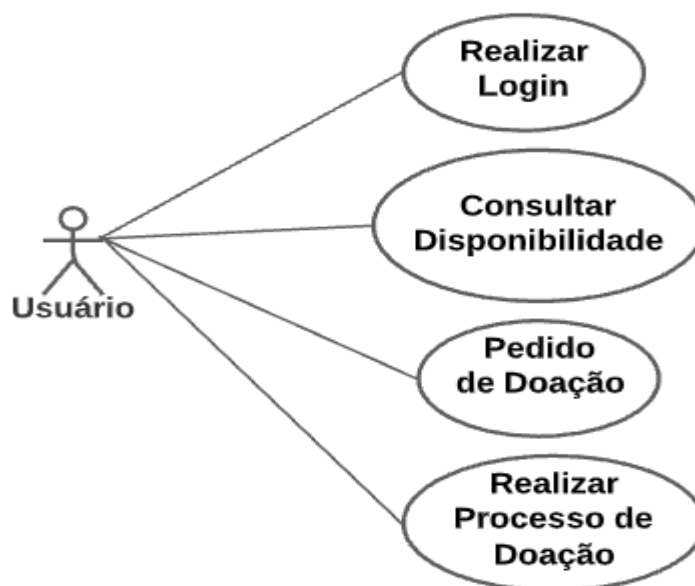
Para os requisitos não funcionais da solução temos os seguintes:

- O protótipo e serviço web devem ser multiplataforma;

### **3.3 DESCRIÇÃO DO DIAGRAMA DE CASOS DE USO DO PROTÓTIPO**

Nesta seção serão descritos as funcionalidades dos casos de uso propostos para o protótipo mobile. No diagrama de Casos de Uso do Protótipo são demonstradas as funcionalidades que o usuário terá com a utilização do sistema. Na Figura 8 está descrito este diagrama:

Figura 8: Representação do Diagrama de Casos de Uso do Protótipo



Fonte: (Autoria Própria, 2020).

Tabela 1 - Caso de uso realizar login

Nome do Caso de Uso	Realizar Login
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	Usuário
<b>Atores Secundários</b>	
<b>Resumo</b>	O usuário irá informar seu CPF e nº do Cartão do SUS no sistema e o mesmo consultará a validade dos dados informados.
<b>Pré-Condições</b>	Possuir um cadastro no sistema da farmácia municipal, conexão com a internet.
<b>Pós-Condições</b>	Listagem de remédios já retirados na farmácia municipal e dados do usuário
<b>Fluxo Principal</b>	

<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Inserir CPF e nº do Cartão do SUS no sistema	
	2. Validar dados e receber dados dos medicamentos já retirados pelo usuário, como também informações do usuário.
	3. Listar informações no aplicativo.

Fonte: (Autoria Própria, 2020).

Tabela 2 - Caso de uso Consultar Disponibilidade

<b>Nome do Caso de Uso</b>	<b>Consultar Disponibilidade</b>
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	Usuário
<b>Atores Secundários</b>	
<b>Resumo</b>	O sistema mostrará ou atualizará a lista de medicamentos que o usuário já retirou e sua disponibilidade, e nela poderão constar alguns botões para as ações de Pedido de Doação e Realizar Processo de Doação.
<b>Pré-Condições</b>	Ter informado feito o login e possuir medicamentos já retirados, conexão com a internet.
<b>Pós-Condições</b>	
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Realizar o Login ou acionar o botão	

de atualizar.	
	2. Carregar/atualizar a listagem dos medicamentos e os botões de ações disponíveis.

Fonte: (Autoria Própria, 2020).

Tabela 3 - Caso de uso Pedido de Doação

<b>Nome do Caso de Uso</b>	<b>Pedido de Doação</b>
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	Usuário
<b>Atores Secundários</b>	
<b>Resumo</b>	Na lista de medicamentos o usuário poderá pressionar o botão de solicitar para realizar um pedido de doação daquele medicamento a outros usuários.
<b>Pré-Condições</b>	Não ter feito a solicitação há menos de sete dias, conexão com a internet.
<b>Pós-Condições</b>	A opção fica indisponível pelos próximos sete.
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Acionar o botão com a descrição solicitar.	
	2. O sistema manda uma requisição ao serviço web que faz o cadastro da solicitação e então devolve a lista de medicamentos atualizada.

	3. O sistema desabilita o botão até que sete dias se passem contando a data da requisição.
--	--

Fonte: (Autoria Própria, 2020).

Tabela 4 - Caso de uso Realizar Processo de Doação

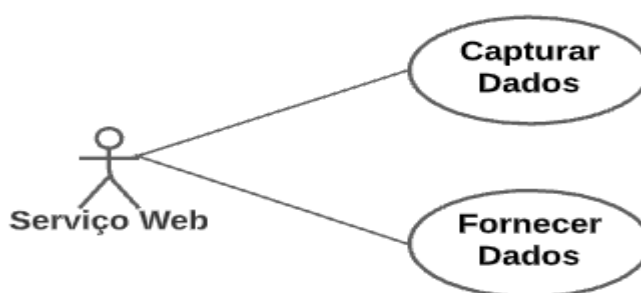
Nome do Caso de Uso	Realizar Processo de Doação
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	Usuário
<b>Atores Secundários</b>	
<b>Resumo</b>	Na listagem de medicamentos, caso outros usuários já tenham feito um pedido de doação para aquele medicamento da lista, um botão com a descrição Doar ficará habilitado, ao pressioná-lo uma listagem de solicitantes irá ser mostrada e o usuário poderá entrar em contato para realizar a doação.
<b>Pré-Condições</b>	Estar logado, possuir medicamentos já retirados, medicamento na lista precisa ter alguma solicitação de doação.
<b>Pós-Condições</b>	
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Acionar o botão com a descrição Doar.	
	2. O sistema solicita ao serviço web os dados de todos os usuários que solicitaram a doação e lista ao usuário para que ele possa realizar a doação.

Fonte: (Autoria Própria, 2020).

### 3.4 DESCRIÇÃO DO DIAGRAMA DE CASOS DO SERVIÇO WEB

Nesta seção serão descritos as funcionalidades dos casos de uso propostos para o Serviço Web. No diagrama de Casos de Uso do Serviço Web é abordado o fluxo de comunicação entre os dados fornecidos pela farmácia e o protótipo mobile. Na Figura 9 segue a apresentação deste diagrama:

Figura 9: Representação do Diagrama de Caso de Uso do Serviço Web



Fonte: (Autoria Própria, 2020).

Tabela 5 - Caso de uso Capturar Dados

Nome do Caso de Uso	Capturar Dados
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	Serviço Web
<b>Atores Secundários</b>	
<b>Resumo</b>	Quando o protótipo enviar uma requisição para obter os dados para seu funcionamento o Serviço Web vai consultar os dados disponibilizados pela farmácia.
<b>Pré-Condições</b>	Requisição do protótipo, conexão com a internet, farmácia disponibilizar os dados.
<b>Pós-Condições</b>	Dados para serem enviados para o protótipo.

<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Recebe requisição do protótipo.	
2. Comunicar com a fonte de dados disponibilizada pela farmácia.	
3. Captura informações conforme requisição recebida pelo protótipo.	
4. Manter informações em memória	

Fonte: (Autoria Própria, 2020).

Tabela 6 - Caso de uso Fornecer Dados

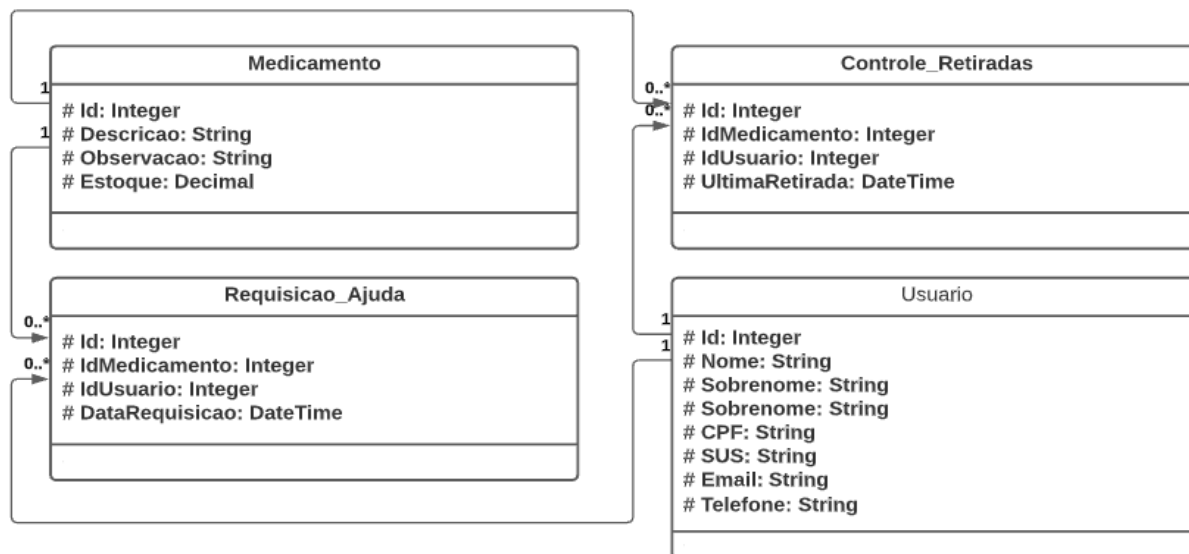
<b>Nome do Caso de Uso</b>	<b>Fornecer Dados</b>
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	Serviço Web
<b>Atores Secundários</b>	
<b>Resumo</b>	Com os dados fornecidos pela farmácia em memória, o Serviço Web devolve os mesmos para o protótipo que fez a requisição.
<b>Pré-Condições</b>	Conseguir obter os dados, conexão com a internet.
<b>Pós-Condições</b>	Envio de dados para o protótipo
<b>Fluxo Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Organizar informações em memória.	
2. Responder à requisição do protótipo com as informações em formato JSON.	

Fonte: (Autoria Própria, 2020).

### 3.5 DIAGRAMA DE CLASSES

O diagrama de classes apresentado na Figura 10 representa como se dispõem os elementos do protótipo, para que seja possível ver a ligação entre os mesmos.

Figura 10: Representação do Diagrama de Classes do protótipo



Fonte: (Autoria Própria, 2020).

Com o diagrama de classes pode ser visualizado as relações entre cada elemento do protótipo. Abaixo são detalhadas as classes pertencentes ao diagrama:

- **Medicamento**: Essa classe é responsável por manter os dados dos medicamentos e sua disponibilidade contendo o atributo *Id* como chave, os atributos *Descricao* e *Observacao* contém informações sobre o medicamento e o *Estoque* armazena a informação quanto a disponibilidade dele caso seja maior que zero está disponível, senão não;

- **Usuario**: Essa classe é responsável por manter os dados dos usuários, ela possui o *ID* como chave, o atributo *SUS* é referente ao número do cartão do SUS e o atributo *CPF* guarda a informação do CPF do usuários, ambos são usados para validar o login, a classe ainda contém o atributo *Nome*, *Sobrenome*, *Email*, e *Telefone*, que guardam informações do usuário na base de dados;

- **Controle\_retiradas**: Através desta classe temos as informações para avaliar os medicamentos já retirados pelo usuário. Tendo como chave o atributo *ID*, ainda temos o atributo *IdMedicamento* refere-se à chave da classe *Medicamento*, da mesma forma o atributo *IdUsuario* com a classe *Usuario*, o atributo *ultima\_retirada* por sua vez representa a data em que o medicamento em questão foi retirado pelo usuário;



- *Requisicao\_Ajuda*: Essa classe é responsável pelos dados cadastrados através do pedido de doação, temos como chave o atributo *ID*, o atributo *IdMedicamento* refere-se à chave da classe *Medicamento*, enquanto o atributo *IdUsuario* refere-se à chave da classe *Usuario*, o atributo *data\_requisicao* serve para controlar a quantia de requisições que podem ser feitas, afinal uma requisição de ajuda só pode ser feita a cada sete dias;

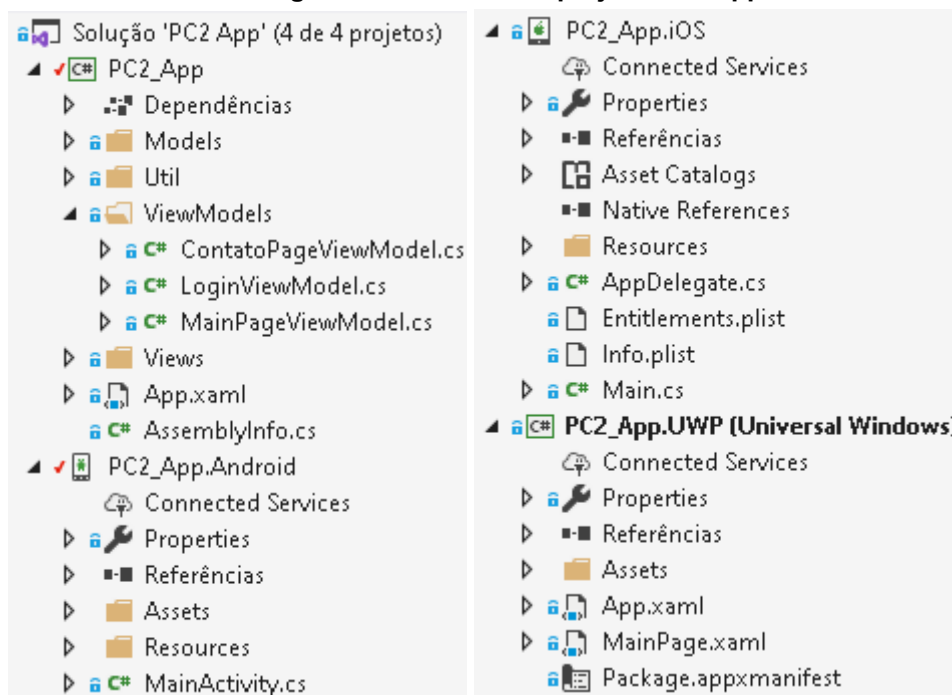
### 3.6 AMBIENTE DE DESENVOLVIMENTO

A IDE utilizada para o desenvolvimento tanto do protótipo do aplicativo móvel quanto para o serviço web foi o Visual Studio Community 2019. O projeto do protótipo foi do tipo Cross-Platform Xamarin.Forms. Enquanto o projeto do Serviço Web foi um Aplicativo Web ASP.NET Core, também foi criado para uma questão de simulação um banco de dados relacional com o Microsoft SQL Server. Devido a uma limitação de hardware não foi possível realizar o desenvolvimento completo do protótipo para iOS, afinal é necessário um computador MAC e não foi possível obter um exemplar do mesmo.

### 3.7 ESTRUTURA DO PROJETO DO PROTÓTIPO DE APLICATIVO

A solução de nome PC2 App é composta pelos projetos **PC2\_App**, **PC2\_App.Android**, **PC2\_App.iOS** e **PC2\_App.UWP**. O projeto **PC2\_App** contém todo o código que é compartilhado com os demais projetos da Solução. Os demais projetos são específicos de cada plataforma, Android, iOS e UWP (Universal Windows Platform), nesses projetos só deve ser implementado o código específico da plataforma, porém pelo método de desenvolvimento utilizado não foi desenvolvido nada neles, somente no projeto com código base (compartilhado). A Figura 11 exibe a estrutura dos projetos, vale ressaltar que o desenvolvimento do aplicativo segue o padrão MVVM.

**Figura 11: Estrutura do projeto PC2 App**

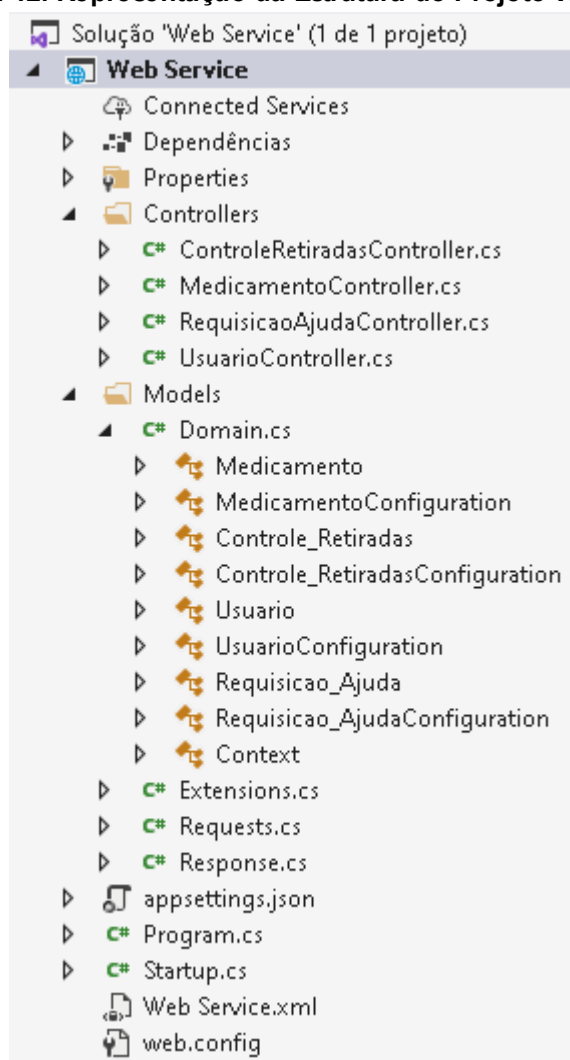


Fonte: (Autoria Própria, 2020).

### 3.8 ESTRUTURA DO PROJETO DO SERVIÇO WEB

A solução Web Service é a que contém o serviço web citado nas seções anteriores, é ele que faz a comunicação com a fonte de dados e nutre o aplicativo com as informações necessárias para seu funcionamento. A Figura 12 exibe sua estrutura, onde a pasta Models contém as classes que fazem a comunicação com o aplicativo, principalmente o arquivo domain.cs que nele estão as especificações dos tipos descritos no diagrama de classes, além das classes com as configurações de mapeamento objeto relacional (Configuration) feitos com o .Net Core. A pasta Controllers contém controladores que recebem as requisições do protótipo. Este projeto foi publicado no site de hospedagem [www.smarterasp.net](http://www.smarterasp.net) que possibilitou a realização dos testes com usuários reais, a fim de validar a usabilidade do protótipo, bastante apenas uma conexão com a internet. Para realizar a publicação foi necessário apenas baixar o arquivo de publicação disponibilizado pelo próprio site e então com o mínimo de configuração o serviço web já ficou disponível.

**Figura 12: Representação da Estrutura do Projeto Web Service**



Fonte: (Autoria Própria, 2020).

### 3.9 FUNCIONAMENTO DO SISTEMA DESENVOLVIDO

Nessa seção são descritas as funcionalidades do aplicativo e são ilustradas as suas telas. O item 3.9.1 descreve como funciona o login no protótipo; O item 3.9.2 apresenta a Tela Principal do aplicativo, que é a tela de listagem de medicamentos; O item 3.9.3 detalha o funcionamento das ações disponíveis na tela principal; O item 3.9.4 descreve o funcionamento do processo de doação através da tela de contatos.

#### 3.9.1 Processo de Login no PC2 App

O processo de login no aplicativo só pode ocorrer para usuários que tenham seus dados já cadastrados na base de dados da central de distribuição, logo não temos um cadastro de novos usuários, a tela de login é ilustrada na Figura 13. Basta

o usuário informar o número do seu CPF e o número do cartão do SUS e pressionar o botão de ENTRAR, então o APP envia uma requisição para o Serviço Web que valida os dados e se estiverem ok redireciona para tela principal, se não mostra uma mensagem de login inválido.

**Figura 13: Tela de Login**

Disponibilidade de Medicamentos

CPF

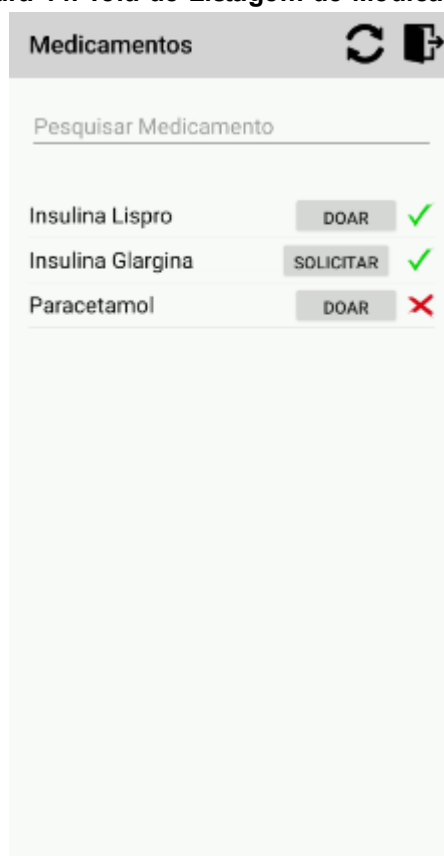
SUS

ENTRAR

Fonte: (Autoria Própria, 2020).

### **3.9.2 Listagem de Medicamentos no PC2 App**

A Figura 14 apresenta a tela principal onde temos a listagem dos medicamentos e suas disponibilidades, além de também termos uma entrada de texto que funciona como filtro da lista exibida, temos na barra superior botões de ação como atualizar a lista e fazer o logout da aplicação, também há alguns botões de ações disponíveis, que serão explicadas posteriormente também como suas disponibilidades dentro do APP. A disponibilidade dos medicamentos são confirmados pelo ícone verde, enquanto a falta do mesmo com o ícone vermelho.

**Figura 14: Tela de Listagem de Medicamentos**

Fonte: (Autoria Própria, 2020).

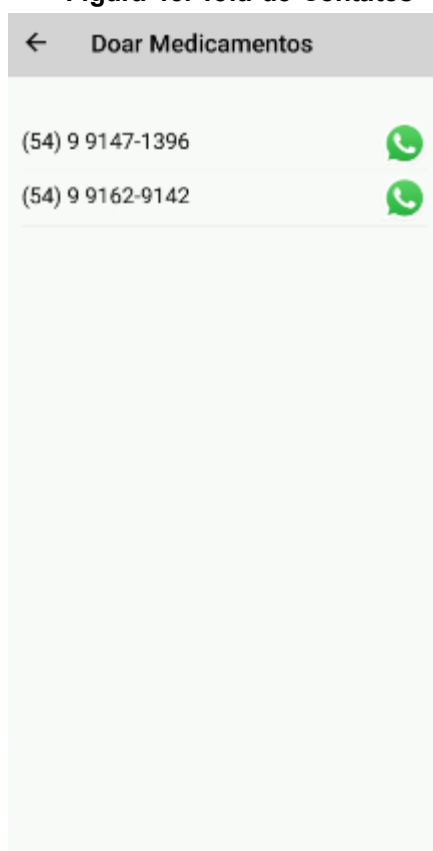
#### 3.9.4 Pedido Doação e Iniciar Processo de Doação no PC2 App

Como mencionado anteriormente na tela de listagem de medicamentos temos os botões de ação que possuem algumas regras específicas de funcionamento, são eles o Solicitar e o Doar. O botão Solicitar realiza ação de fazer um pedido de doação, que consiste em enviar uma requisição ao serviço web e registrar que o usuário logado precisa da doação daquele medicamento em questão, esse botão só fica visível se o usuário logado não fez uma solicitação nos últimos 7 dias, evitando múltiplos pedidos repetidos para um mesmo usuário. O botão Doar executa a ação de iniciar o processo de doação e só fica visível se outro usuário tenha feito um pedido de doação nos últimos 7 dias para aquele medicamento, sendo assim só pode-se doar medicamentos que tiver solicitações, ao pressionar o botão o usuário logado é redirecionado para a tela de contato.

### 3.9.5 Comunicação entre Usuários pela Tela de Contatos no PC2 App

A comunicação entre os usuários será feita através de uma integração entre o aplicativo desenvolvido e o WhatsApp de maneira muito simples, no banco de dados da farmácia municipal o usuário tem a informação de celular, então usando essa informação sem identificar de fato o nome do outro usuário para tentar manter de forma sigilosa até certo ponto, tanto a parte que quer receber os medicamentos quanto quem quer doá-los, os usuários poderão se comunicar e definir a melhor maneira de se auxiliar em caso de falta de medicamentos, clicando no ícone do WhatsApp. A lista consiste em usuários que possuem em seu registro um telefone celular cadastrado e fizeram um pedido de doação nos últimos sete dias como ilustrado pela Figura 15.

**Figura 15: Tela de Contatos**



Fonte: (Autoria Própria, 2020).

Ao clicar o ícone do Whatsapp uma mensagem personalizada é preparada e já deixa pronta para envio, segundo o seguinte formato: “Olá, vim através do APP de consulta de medicamentos e tenho disponível o medicamento \*\_Nome do Medicamento\*\_ que você solicitou.”.

#### 4 VALIDAÇÃO DE USABILIDADE

A validação do aplicativo foi realizada em duas etapas. Inicialmente foi disponibilizado um questionário para identificar o perfil do usuário e as suas expectativas em relação ao aplicativo de consulta de medicamentos. Após foi disponibilizado um link a partir do qual o usuário poderia fazer download do aplicativo e realizar os testes. Finalizando, então, o processo de avaliação, os usuários foram convidados a um questionário de pós-teste que buscou capturar as suas expectativas com relação ao uso aplicativo. No total foram realizados cinco testes com potenciais usuários da farmácia do município. Os questionários utilizados no pré-teste e pós-teste constam no apêndice deste relatório. Os resultados do processo de validação são apresentados a seguir:

O usuário A1 está na faixa-etária de dos 46 aos 50 anos e possui pouca experiência com o uso de aplicativos. Assinala a necessidade do aplicativo que pode evitar deslocamento desnecessários a farmácia. O usuário A1 considerou o aplicativo de fácil uso, porém assinalou que seria interessante do aplicativo disponibilizar um cadastro de doação para outros usuários que não estejam cadastrados na farmácia aplicando o espectro de potenciais doadores.

O usuário A2 está na faixa-etária acima dos 50 anos e faz uso constante de aplicativos de diversas naturezas. O usuário A2 considerou o aplicativo muito intuitivo assinalando que o mesmo atende plenamente às suas expectativas. O usuário A3, que está na faixa-etária dos 36 a 45 anos, também observou a facilidade de uso do aplicativo, porém considerou que o mesmo não traz informações sobre a previsão de disponibilidade de medicamentos o que seria uma informação importante para os usuários da farmácia. As mesmas considerações foram realizadas pelo usuário A4, que encontra-se na faixa-etária dos 26 aos 35 anos. Finalmente, o usuário A5, que está na faixa-etária de dos 18 aos 25 anos, considerou que o aplicativo atendeu as necessidades de um usuário da farmácia municipal.

Os resultados dos testes permitiram observar que independentemente da faixa-etária, os usuários assinalaram que o aplicativo desenvolvido possui uma interface intuitiva e que o seu uso pode ser facilmente compreendido. Os usuários consideram que as informações são importantes e atendem ao objetivo de tornar o processo de retirada de medicamentos menos oneroso para os usuários da farmácia. Os usuários assinalaram, porém, que a disponibilização de outras informações como a

previsão de disponibilidade de medicamentos é necessária para que o usuário possa planejar quando realizará o seu deslocamento para a farmácia municipal.

Os resultados dos testes realizados serão utilizados para planejar a implementação de melhoramentos no aplicativo desenvolvido em versões futuras do mesmo.



## 5 CONCLUSÃO

O presente trabalho buscou criar um recurso informatizado que auxiliasse os usuários da farmácia municipal. Muitas vezes, esses usuários têm que se deslocar até a farmácia sem saber ao certo se o medicamento que buscam está disponível ou não. O aplicativo desenvolvido buscou criar uma alternativa para minimizar esse problema e disponibilizar uma solução quando o mesmo não estiver disponível, criando uma rede de cooperação entre os usuários.

O aplicativo foi criado utilizando a plataforma Xamarin.Forms. Essa suíte de possibilitar o desenvolvimento de aplicativos multiplataforma através de um ambiente unificado utilizando a linguagem de programação C#. Dessa forma, foi possível criar versões da aplicação que rodam nos Sistemas Operacionais Android, Windows Phone e futuramente também será disponibilizada uma versão para iOS. Além de facilitar o desenvolvimento, o uso dessa metodologia também facilita a aceitação do aplicativo pelos usuários da farmácia que poderão utilizar o mesmo no sistema operacional de sua preferência.

Os testes realizados com cinco usuários da farmácia indicaram que a interface desenvolvida para aplicativo é adequada independentemente da faixa-etária dos usuários, porém informações como a previsão de disponibilidade de medicamentos deve ser adicionada para facilitar o planejamento dos usuários da farmácia. Os melhoramentos sugeridos pelos usuários serão adicionados em versões futuras do aplicativo.

Nessa etapa de desenvolvimento foram utilizados dados simulados para testar e demonstrar o uso do aplicativo. Espera-se que futuramente o aplicativo possa ser utilizado em um contexto real.

## 6 REFERÊNCIAS

BARTH, Nikson. **ANÁLISE COMPARATIVA DE FERRAMENTAS DE DESENVOLVIMENTO DE APLICATIVOS MÓVEIS MULTIPLATAFORMA**. 2014. Trabalho de Conclusão de Curso II (Bacharelado em Ciência da Computação) – Universidade Regional de Blumenau Centro de Ciências Exatas e Naturais, Blumenau.

Brasil, Secretaria de Políticas de Saúde Departamento de Formulação de Políticas de Saúde, **POLÍTICA NACIONAL DE MEDICAMENTOS**, 2001. Disponível em: <[http://bvsms.saude.gov.br/bvs/publicacoes/politica\\_medicamentos.pdf](http://bvsms.saude.gov.br/bvs/publicacoes/politica_medicamentos.pdf)>. Acesso em: 01 dezembro de 2020.

DE LA TORRE, Cesar. .NET Core, .NET Framework, Xamarin – **The “WHAT and WHEN to use it”**, 2016. Disponível em: <<https://devblogs.microsoft.com/cesardelatorre/net-core-1-0-net-framework-xamarin-the-whatand-when-to-use-it/>>. Acesso em: 20 de novembro de 2020.

FERREIRA FILHO, Otávio Freitas. **Serviços semânticos: uma abordagem RESTful**. 2009. Dissertação (Mestrado) (Bacharelado em Ciência da Computação) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais, São Paulo.

Fundação Getulio Vargas. **Brasil tem 424 milhões de dispositivos digitais em uso, revela a 31ª Pesquisa Anual do FGVcia**. Disponível em: <<https://portal.fgv.br/noticias/brasil-tem-424-milhoes-dispositivos-digitais-uso-revela-31a-pesquisa-anual-fgvcia>>. Acesso em: 24 de novembro de 2020.

STADISTA. **Mobile operating systems' market share worldwide from January 2012 to July 2020**. 2020. Disponível em: <<https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>>. Acesso em: 25 de novembro de 2020.

SMITH, John. Patterns - **WPF Apps With The Model-View-ViewModel Design Pattern**, 2009. Disponível em: <<https://docs.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern>>. Acesso em: 21 de novembro de 2020.

LECHETA, Ricardo R. **Web Services RESTful: Aprenda a criar web services RESTful em Java na nuvem do Google**. São Paulo, Novatec Editora, 2015.

LENNON, Joe. **Beginning CouchDB**. 2009. E-Book. ISBN 978-1-4302-7237-3 Disponível em: <<https://link.springer.com/book/10.1007/978-1-4302-7236-6>>. Acesso em: 28 novembro 2020.

MACORATTI, José Carlos. **Xamarin Studio - Apresentando Xamarin.Forms**, 2016. Disponível em: <[http://www.macoratti.net/15/12/xam\\_forms1.htm](http://www.macoratti.net/15/12/xam_forms1.htm)>. Acesso em:

29 de novembro de 2020.

MICROSOFT. **Documentação do C#**, 2020a. Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/csharp/>>. Acesso em: 28 de novembro de 2020.

MICROSOFT. **Introdução ao .NET**, 2020b Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/core/introduction>>. Acesso em: 28 de novembro de 2020.

MICROSOFT. **Documentação Técnica do SQL Server**, 2020c Disponível em: <<https://docs.microsoft.com/pt-br/sql/sql-server/?view=sql-server-ver15>>. Acesso em: 28 de novembro de 2020.

MICROSOFT., **O que é o Xamarin?**, 2020d. Disponível em: <<https://docs.microsoft.com/pt-br/xamarin/get-started/what-is-xamarin>>. Acesso em: 28 novembro 2020

COSTA, Norben P. O.; FILHO, Nemésio F. Duarte. **Análise e Avaliação Funcional de Sistemas Operacionais Móveis: Vantagens e Desvantagens**. Revista de Sistemas e Computação, Salvador, v.3, nº 1, p. 66-77, jan./jun. 2013.

HOLZER, Adrian; ONDRUS Jan. **Mobile app development: Native or Web?**.Proc. Workshop eBus.(WeB). 2012.

PETKOVIĆ, Dušan. **JSON Integration in Relational Database Systems**, 2017. Disponível em: <<https://www.ijcaonline.org/archives/volume168/number5/petkovi%20c4%87-2017-ijca-914389.pdf>>. Acesso em: Acesso em 25 novembro 2020.

PETZOLD, Charles. **Creating Mobile Apps with Xamarin.Forms**. 2. ed. atual. e rev. Novembro 2016. E-Book. ISBN: 978-1509302987 Disponível em: <<https://github.com/xamarin/xamarin-forms-book-samples>>. Acesso em: 26 novembro 2020.

PROCEDI, Lisandro. **Avaliação do framework Xamarin.Forms para desenvolvimento de aplicativos móveis multiplataforma, criando uma aplicação real**. 2016. Monografia de Curso (Bacharelado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre.

QUAIATO, Vinicius. **O que é Xamarin?**, 2016. Disponível em: <<http://www.lambda3.com.br/2016/10/o-que-e-xamarin/>>. Acesso em: 25 de novembro de 2020.

SADZINSKI Osmani José et al., **“Gestão de uma Farmácia Solidária a partir de um Aplicativo Pautado em Normas, Padrões e Requisitos preconizados pela Sociedade Brasileira de Informática em Saúde”**, 2010. Disponível em: <<http://periodicos.unesc.net/sulcomp/article/view/258/376>>. Acesso em: 22 novembro de 2017.

Secretaria Municipal da Saúde de São Paulo, **“MANUAL DE INSTRUÇÕES DO SITE/APLICATIVO AQUI TEM REMÉDIO”**, 2015. Disponível em: <[https://www.prefeitura.sp.gov.br/cidade/secretarias/upload/saude/2015\\_12\\_16\\_MANUAL%20AQUI%20TEM%20REMEDI0%20\(1\).pdf](https://www.prefeitura.sp.gov.br/cidade/secretarias/upload/saude/2015_12_16_MANUAL%20AQUI%20TEM%20REMEDI0%20(1).pdf)>. Acesso em: 22 novembro de 2020.

## APÊNDICES

### APÊNDICE A – Questionário pré-teste:

- 1) Faixa-etária:
  - a) 18 a 25 anos
  - b) 26 a 35 anos
  - c) 36 a 45 anos
  - d) 46 a 50 anos
  - e) mais de 50 anos
- 2) Qual a sua experiência no uso de aplicativos de smartphone?
- 3) Como você acha que o aplicativo de consulta de disponibilidade de medicamentos pode auxiliar na sua rotina ou tarefas?
- 4) Você já fez uso de outros sistemas de consulta de disponibilidade de medicamentos? Em caso de positivo qual a sua percepção desta experiência?
- 5) Que tipo de informações você espera que o aplicativo possa lhe fornecer?

### APÊNDICE B – Questionário após o teste:

- 1) Você considera que o aplicativo é fácil de utilizar? Se você teve dificuldades, quais foram?
- 2) As informações recuperadas estão de acordo com as suas expectativas?
- 3) Teria sugestões para melhorar o aplicativo?