

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

**ANDRÉ FELIPE WASEM**

**COMÉRCIO ELETRÔNICO – SISTEMA MELHOR PREÇO PARA  
FARMÁCIAS**

**FERNANDO ABRAHÃO AFONSO**

**PASSO FUNDO, 2014**

**ANDRÉ FELIPE WASEM**

**COMÉRCIO ELETRÔNICO – SISTEMA MELHOR PREÇO PARA  
FARMÁCIAS**

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-Rio-Grandense, *Campus* Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador (a): Fernando Abrahão Afonso

**PASSO FUNDO, 2014**

**ANDRÉ FELIPE WASEM**

**COMÉRCIO ELETRÔNICO – SISTEMA MELHOR PREÇO PARA FARMÁCIAS**

Trabalho de Conclusão de Curso aprovado em \_\_\_\_/\_\_\_\_/\_\_\_\_ como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

---

Nome do Professor(a) Orientador(a)

---

Nome do Professor(a) Convidado(a)

---

Nome do Professor(a) Convidado(a)

---

Coordenação do Curso

**PASSO FUNDO, 2014**

## **AGRADECIMENTOS**

Quero agradecer a todos os meus professores que me incentivaram ao longo dos anos de curso a não desistir e sempre mostrando motivos para continuar.

Gostaria de agradecer em especial a dois professores que são o Fernando Afonso e a Silvani Lopes Lima. Ao Fernando meu muito obrigado pela paciência e compreensão ao longo do desenvolvimento do trabalho e à professora Silvani pela compreensão e a ajuda nas correções que foram muito úteis.

## RESUMO

Este trabalho de pesquisa tem como tema o desenvolvimento de um software para aplicativos móveis utilizando o SDK do Android capaz de auxiliar os consumidores na consulta de preços de medicamento farmacêuticos, permitindo que ele possa optar pelo melhor preço disponível no mercado. A motivação para este estudo parte do princípio de que os consumidores de produtos farmacêuticos nem sempre dispõem de tempo para percorrer as farmácias e realizar a pesquisa de preços dos produtos que estão adquirindo. Para este estudo, além do uso do SDK do Android para o aplicativo móvel, optou-se pela utilização da linguagem Java em conjunto de outras APIs para o desenvolvimento do *web services*.

Palavras-chave: Produtos farmacêuticos; Software para pesquisa de preços; Aplicativos móveis.

## **ABSTRACT**

This research theme is the development of software for mobile applications using the Android SDK can help consumers in outpatient pharmaceutical drug prices, allowing it to opt for the best available price on the market. The motivation for this study assumes that consumers of pharmaceutical products do not always have time to go through pharmacies and conduct research on the prices of products they are buying. For this study, besides the use of the Android SDK for mobile application, we chose to use the Java language together with other APIs for the development of web services.

**Keywords:** Products Pharmaceuticals; Software to research prices; Mobile applications.

## LISTA DE FIGURAS

Figura 1: Mercado Mundial de Smartphones .....	18
Figura 2: Diagrama de Atividades.....	33
Figura 3: Representação Geral do Sistema.....	34
Figura 4: Caso de uso sistema de cadastro de medicamentos .....	37
Figura 5: Diagrama de classes de cadastro de medicamentos.....	40
Figura 6: Formulário de cadastro de medicamentos.....	40
Figura 7: Formulário de listagem de medicamentos .....	40
Figura 8: Caso de uso sistema de cadastro de <i>web services</i> .....	42
Figura 9: Diagrama de classes do sistema de cadastro de <i>web services</i> .....	45
Figura 10: Formulário que lista os <i>web services</i> .....	45
Figura 11: Formulário de cadastro de <i>web services</i> .....	45
Figura 12: Caso de uso do aplicativo móvel.....	49
Figura 13: Formulário de pesquisa de medicamentos .....	50
Figura 14: Tela com a mensagem "GPS deve ser ligado" .....	51
Figura 15: Tela com lista de medicamentos ordenada pela distância.....	51
Figura 16: Tela com lista de medicamentos ordenado por valor.....	52
Figura 17: Tela de onde é selecionado o tipo de ordenação da lista de medicamentos.....	53
Figura 18: Tela com informações do medicamento selecionado na lista .....	53

## LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i> , p. 9
CMED	Câmara de Regulação do Mercado de Medicamentos, p. 16
CRF-RJ	Conselho Regional de Farmácia do Estado do Rio de Janeiro, p. 13
DB	<i>Data Base</i> , p. 10
DDL	<i>Data Definition Language</i> , p. 28
DML	<i>Data Manipulation Language</i> , p. 28
GSON	<i>GoogleScript Obejct Notation</i> , p. 9
GPS	<i>Global Positioning System</i> , p. 26
HTML	<i>Hypertext Markup Language</i> , p. 25
IDE	<i>Integrated Development Environment</i> , p. 10
JDK	<i>Java Development Kit</i> , p. 20
JSON	<i>JavaScript Object Notation</i> , p. 9
PL/SQL	<i>Procedural Language extensions to SQL</i> , p. 28
SDK	<i>Software Development Kit</i> , p. 10
SGBD	Sistema de Gerenciador de Banco de Dados, p. 27
SO	Sistema Operacional, p. 17
SOA	<i>Service Oriented Architeteture</i> , p. 24
SOAP	<i>Simple Object Access Protocol</i> , p. 24
SQL	<i>Structured Query Language</i> , p. 28
WM	<i>Windows Mobile</i> , p. 21
WP	<i>Windows Phone</i> , p. 22
XML	<i>eXtensible Markup Language</i> , p. 9

## SUMÁRIO

1	INTRODUÇÃO .....	9
1.1	MOTIVAÇÃO .....	10
1.2	OBJETIVOS .....	10
1.2.1	Objetivo Geral .....	10
1.2.2	Objetivos específicos .....	11
2	REFERENCIAL TEÓRICO .....	12
2.1	O COMÉRCIO FARMACÊUTICO .....	12
2.1.1	Definição de tecnologia .....	14
2.1.2	Delimitação do tema .....	15
2.2	CRESCIMENTO NA VENDA DE SMARTPHONES .....	17
2.3	DESENVOLVIMENTO DE APLICATIVOS PARA SMARTPHONES.....	18
2.4	WEB SERVICES.....	23
2.4.1	Consumindo <i>web services</i> com <i>android</i> .....	24
2.4.2	JSON .....	24
2.4.5	XML.....	25
2.5	ANDROID GEOLOCALIZAÇÃO.....	25
2.6	BANCO DE DADOS RELACIONAIS .....	27
2.2	RECURSOS UTILIZADOS .....	28
2.2.1	Linguagem de programação utilizada.....	29
2.2.2	Gson.....	29
2.2.3	Jersey e arquitetura <i>restfull</i> .....	30
2.2.4	Hibernate .....	31
2.2.5	Servidor de aplicação Apache Tomcat .....	32
3	DESENVOLVIMENTO METODOLÓGICO.....	33
3.1	SISTEMA DE CADASTRO DE MEDICAMENTOS .....	36
3.1.1	Diagrama de caso de uso .....	36
3.1.2	Diagrama de Classes do sistema de cadastro de medicamentos.....	39
3.2	SISTEMA DE CADASTRO DE WEB SERVICES .....	41
3.1.1	Diagrama de caso de uso .....	41
3.1.2	Diagrama de classes do Sistema de Cadastro de <i>Web Services</i> .....	44
3.3	SERVIÇO WEB DAS FARMÁCIAS .....	46

3.4	WEB SERVICES CENTRAL .....	46
3.4.1	Usando <i>web services</i> do Google para determinar cidade .....	47
3.4.2	Determinar distância e duração da viagem em modo <i>driving</i> .....	47
3.4.3	Enviando resultado para o dispositivo móvel.....	48
3.5	APLICATIVO ANDROID .....	48
3.5.1	Funcionalidades realizadas pelo aplicativo .....	49
3.5.2	Apresentação do sistema .....	50
4	CONSIDERAÇÕES FINAIS.....	54
5	REFERÊNCIAS .....	56

## 1 INTRODUÇÃO

Normalmente, pessoas que precisam de medicamento, algum produto de higiene, ou qualquer outro produto vendido em farmácia vão logo até a farmácia mais próxima e acabam muitas vezes não pesquisando aquela que oferece o preço mais vantajoso. Preços estes que muitas vezes variam de centavos para reais, isso dependendo da quantidade e valores dos produtos adquiridos. Também existe a locomoção que é feita pelo cliente até a farmácia e a possibilidade de o produto não estar disponível na farmácia

Surgiu ai uma oportunidade para o desenvolvimento de um aplicativo que realize a pesquisa dos preços de medicamentos, fazendo assim com que o consumidor não precise se deslocar para todas as farmácias para conseguir avaliar o preço dos medicamentos. Evitando assim gasto de tempo e dinheiro desnecessário.

Este projeto tem como propósito desenvolver uma aplicação que atenda a um público específico, no caso os consumidores de produtos farmacêuticos e que façam uso de um dispositivo com Sistema Operacional Android. A escolha pelo desenvolvimento de aplicação para Android foi adotada, devido à grande parcela que o Android tem distribuído pelo mundo.

Para que fosse possível realizar a conclusão do projeto, foi realizado um estudo da linguagem de programação Java. Esta linguagem foi adotada devido à documentação de fácil acesso, comunidades de respostas rápidas, além da grande quantidade de recursos por ela oferecido.

Também foi necessário realizar um estudo sobre as diversas APIs (*Application Programming Interface*) disponíveis para facilitar o desenvolvimento dos projetos. Através do estudo, foi possível constatar, que se o uso de APIs já existentes não for empregado de forma incorreta, acaba por gerar muito trabalho desnecessário, por este motivo procurou-se estudar ao máximo a implementação de cada API aplicada nos projetos.

O sistema foi desenvolvido utilizando a linguagem de programação Java para Android, com XML (*eXtensible Markup Language*) para a camada *View* (Visão) da aplicação. A aplicação realiza a busca do produto pesquisado no *web services*, retornando para o usuário uma lista de produtos ordenada pelos produtos de menor preço e pela farmácia que está mais acessível para o usuário no momento da pesquisa.

O formato de comunicação entre o *web services* e a aplicação Android é realizado no formato JSON (*JavaScript Object Notation*), utilizando a API GSON (*GoogleScript Obejct Notation*).

No que se refere às ferramentas de desenvolvimento, foi utilizado a IDE (*Integrated Development Environment*) Eclipse, Netbeans, na parte de DB (*Data Base*) PostgreSQL e Android SDK (*Software Development Kit*) para desenvolvimento da aplicação Android.

Outras APIs também são utilizadas, como a API Jersey, JSF (*Java Server Faces*), *Prime faces* e *Hibernate*. Cada API tem sua particularidade, o que a torna extremamente importante para uma rápida solução e de fácil controle.

A API Gson é mantida por um grupo no Google e tem como objetivo minimizar o trabalho do desenvolvedor na hora de realizar a conversão de strings para objetos.

Já a API JSON e Jersey são mantidas pela Oracle, porém cada uma tem um uso diferente, no caso da API JSON é desenvolvida para realizar a conversão de strings para objetos e vice-versa e API Jersey já tem como objetivo facilitar o desenvolvimento de *web services*.

O Hibernate foi utilizado na camada de persistência dos dados e o JSF e PrimeFaces para a camada de apresentação dos dados.

## **1.1 MOTIVAÇÃO**

O elemento que motivou a realização deste trabalho foi a análise do contexto atual dos preços de medicamentos e demais produtos vendidos nas farmácias. É possível identificar diferenças de valores, sendo essas diferenças de estado para estado, de cidade para cidade e de bairro para bairro, variando de centavos a reais.

## **1.2 OBJETIVOS**

Este trabalho tem como objetivo principal desenvolver uma aplicação para dispositivos móveis baseada na tecnologia Android que faça a pesquisa do melhor preço entre produtos vendidos nas farmácias.

### **1.2.1 Objetivo Geral**

Desenvolver uma aplicação Android que faça a pesquisa de produtos de farmácias apresentando o preço de cada produto e a localização da farmácia para o consumidor que está pesquisando.

### 1.2.2 Objetivos específicos

- Realizar pesquisa a respeito das diferenças de preços entre produtos de farmácias.
- Estudar a comunicação *web services* com dispositivos móveis, utilizando a linguagem Java.
- Desenvolver uma aplicação Android que faça a pesquisa dos produtos nos *web services* das farmácias.
- Desenvolver um sistema que possibilite simular o cadastro de medicamentos utilizado pelas farmácias e um *web services* para disponibilização dos dados para consulta do aplicativo móvel.
- Desenvolver uma aplicação para realizar o cadastro de *web services*.
- Desenvolver o *web services* para listagem das farmácias.

## 2 REFERENCIAL TEÓRICO

A evolução no desenvolvimento de aplicações para dispositivo móvel vem aumentando diariamente. Este constante desenvolvimento demanda cada vez mais mão-de-obra e também novos tipos de sistemas. Pensando nisso, este projeto busca sanar a necessidade de uma aplicação que possibilite que consumidores de produtos farmacêuticos consigam facilmente fazer pesquisa de produtos fornecidos pelas farmácias espalhadas pela região onde vivem.

### 2.1 O COMÉRCIO FARMACÊUTICO

A indústria farmacêutica instalada no Brasil só cresce e nos últimos anos vem tendo um aumento significativo em suas vendas. No ano de 2011, o setor teve um crescimento de 10% em suas vendas, comparado ao ano de 2010. Já no ano de 2012, estimava-se que o mercado farmacêutico tivesse um crescimento de 14% (FRIAS, 2012); algo que já pôde ser confirmado no ano de 2013. Esta porcentagem equivale a uma montante de R\$ 49 bilhões.

E o setor farmacêutico promete continuar crescendo e a obter resultados ainda mais expressivos do que o comércio em geral. Atualmente, a taxa de crescimento de vendas de medicamentos no Brasil tem sido seis vezes maior do que o desempenho dos países desenvolvidos, girando em torno de 13% ao ano, o que em países ricos não chega a 2%. Segundo Favoretto (2013), “As estimativas revelam ainda que o mercado farmacêutico deve dobrar em cinco anos, movimentando R\$ 87 bilhões em 2017”.

O motivo do crescimento do setor farmacêutico deve-se ao aumento das pessoas que subiram para a classe média. Este aumento não deve ser somente atrelado ao maior acesso da população, mas também aos diversos produtos que as farmácias oferecem. Quando uma pessoa vai a uma farmácia não pensa somente em adquirir um medicamento, mas também produtos de beleza, dermocosméticos, higiene, alimentos, fitness, ortopedia e fisioterapia, entre outras linhas de produtos oferecidos.

O comércio farmacêutico cresce diariamente e a quantidade de produtos fornecidos pelas farmácias é enorme. Com isto começam a surgir diferenças nos preços, estas diferenças podem variar de farmácia para farmácia e de bairro para bairro. Algumas farmácias podem oferecer produtos com maior preço que outras farmácias e dependendo do produto um preço menor que suas concorrentes.

Dados demonstram que somente o fato de se atravessar de uma rua para outra para pesquisar preços de produtos farmacêuticos pode render ao consumidor uma compra bem mais econômica em farmácias do Rio de Janeiro. De acordo com reportagem publicada pelo Jornal *O globo* (JARDIM, 2012), uma pesquisa sobre as farmácias na cidade do Rio de Janeiro revela que “Um simples ato, como atravessar a rua ou dobrar a esquina, pode render ao consumidor uma compra bem mais barata nas farmácias do Rio. Um levantamento de *O Globo*, realizado no dia 7 de novembro, constatou diferenças de preços de quase 70% em estabelecimentos situados no mesmo quarteirão”.

Segundo dados do CRF-RJ (Conselho Regional de Farmácia do Estado do Rio de Janeiro), hoje a cidade capital do estado do Rio de Janeiro acumula um total de 1.631 drogarias e farmácias espalhadas na cidade. Na pesquisa feita por *O Globo*, foram percorridas 12 lojas na Tijuca, Centro, Leblon e Copacabana. Verificou-se que para gastar menos, o consumidor tem de pesquisar muito, pois uma mesma farmácia pode oferecer produtos com preços acima de todas as outras farmácias e ao mesmo tempo oferecer produtos com preços menores que outros concorrentes. (JARDIM, 2012)

Na tabela 1, pode ser visualizada a diferença de preço entre produtos farmacêuticos pesquisados no Rio de Janeiro, sendo mostrada a diferença de preços entre os medicamentos de uma farmácia para outra. Depois é demonstrada a diferença entre produtos de higiene na tabela 2, a referida pesquisa não se restringe somente a medicamentos e higiene, porém aqui o objetivo é apenas colocar uma amostra destas diferenças que ocorre entre as farmácias.

**Tabela 1:** Tabela de diferença entre valores de medicamentos

Descrição	Rua	Bairro	Farmácia	Valor	Diferença em %
Energético Targifor C, com 16 comprimidos	Rua Ataulfo de Paiva	Leblon	Droga Raia (número 341) Pacheco (número 319)	R\$ 35,25 R\$ 22,45	60%
Polvitamínico Centrum 130 comprimidos	Rua Ataulfo de Paiva	Leblon	Droga Raia (número 341) Farmalife (número 285)	R\$ 135,00 R\$ 157,60	16,7%
Uma caixa com 30 adesivos de cinco centímetros de Exelon	General Roca, 836	Tijuca	Venâncio Drogasmil	R\$ 329,89 R\$ 442,00	34%
Xenical com 84 cápsulas (produto usado para eliminar gordura)	General Roca, 932	Tijuca	Drogasmil Venâncio	R\$ 339,62 R\$ 267,99	26,7%
Claritin de seis comprimidos	General Roca, 818	Tijuca	Venâncio Drogaria São Paulo	R\$ 11,95 R\$ 14,21	20%

Fonte: adaptado de JARDIM, 2012

**Tabela 2:** Tabela de diferença entre valores de produtos de higiene

Descrição produto	Rua	Bairro	Farmácia	Valor	Diferença em %
Sundown fator 30, de 250 ml	Rua Ataulfo de Paiva	Leblon	Farmalife Pacheco	R\$ 28,05 R\$ 34,49	23%
Colgate Luminous White	Nossa Senhora de Copacabana	Copacabana	Rio Farma	R\$ 3,49	67%

	número 500				
	Nossa Senhora de Copacabana número 479		Pague Menos	R\$ 5,85	
Condicionador Seda CoCriações	Nossa Senhora de Copacabana número 500	Copacabana	Rio Farma	R\$ 5,49	65%
	Nossa Senhora de Copacabana número 479		Pague Menos	R\$ 8,99	
Fralda da Turma da Mônica soft touch, P, com 28 unidades	Nossa Senhora de Copacabana número 500	Copacabana	Rio Farma	R\$ 16,49	29%
	Nossa Senhora de Copacabana número 479		Pague Menos	R\$ 21,29	

**Fonte:** adaptado de JARDIM, 2012.

### 2.1.1 Definição de tecnologia

Pensando em uma solução para atender este problema, foi pesquisado o desenvolvimento de uma aplicação que forneça ao consumidor a farmácia que está com menor preço e que seja mais acessível. Para o desenvolvimento desta solução foi realizado o estudo de algumas tecnologias, priorizando a tecnologia que está em destaque. Considerando assim a tecnologia voltada para dispositivos móveis Android, sendo desenvolvida uma aplicação que consome um *web services*, o qual lista todos os *web services* das farmácias.

As farmácias irão disponibilizar informações referentes aos produtos que vendem, para isso será fornecido um sistema onde as farmácias possam enviar seus dados através de *web services*. Estes dados podem ser transferidos via processo manual (um usuário digitando todos os produtos que deseja postar no *web services* via aplicação Web) ou automatizado (farmácia cria um algoritmo que faz este processo após cada inserção, atualização ou deleção no seu banco de dados), assim após cada atualização no banco das farmácias será disparado uma atualização para o *web services*, isso mediante validação da farmácia que está enviando os dados e se o item enviado está em acordo com as regras do sistema. Quando o produto já se encontra cadastrado no *web services*, o mesmo somente será atualizado.

Após estas informações serem inseridos no *web services*, os usuários, através de um aplicativo inicialmente somente desenvolvido para a plataforma Android, irão acessar e verificar a farmácia que está oferecendo o produto que seja mais vantajoso e que está mais acessível para o usuário no momento da pesquisa.

A comunicação com o *web services* poderá ser realizada em vários formatos de comunicação, inicialmente procura-se estudar a comunicação que ofereça mais recursos e vantagens, deixando o desenvolvimento mais prático, e que atenda às necessidades da aplicação. Nesse sentido, está sendo pesquisada a comunicação com *web services* através do formato JSON com a utilização da API GSON.

O formato JSON oferece a vantagem de ter uma compactação com tamanho menor dos dados, tornando-se assim o formato ideal para o desenvolvimento de aplicações móveis que não tem uma capacidade processamento tão elevada.

Existe também a possibilidade de fazer a comunicação via XML, porém esta comunicação apresenta-se mais lenta, pois a compactação dos dados é realizada no formato mais extenso de texto, o que prejudica a velocidade com que as informações irão trafegar na rede, aumentando o tempo e diminuindo a qualidade de entrega dos dados para os usuários.

É possível utilizar a API do JSON e GSON juntas em uma mesma aplicação, pois estes só fazem a conversão de objetos para formato texto e vice-versa. Com relação à integração entre as duas APIs, ainda não foi definido se isso se procederá, pois o objetivo é ver se há real necessidade e o que cada uma atende para esta aplicação.

### **2.1.2 Delimitação do tema**

Como este projeto está voltado inicialmente para a elaboração de um sistema que faça apenas o comparativo do módulo de medicamentos, os demais módulos não serão abordados no decorrer do texto. Em um primeiro momento, é estudado apenas o contexto para o desenvolvimento de uma aplicação que faça pesquisa de medicamento por categorias, melhor preço entre as categorias de medicamentos e melhor preço de medicamentos entre as farmácias.

Quase todos os medicamentos são distribuídos nas diferentes categorias, sendo elas ético/de marca, genérico ou similar. Para que um medicamento passe a existir nestas três categorias demora-se 20 anos, pois após este tempo o medicamento de marca/ético perde a patente, assim os laboratórios podem desenvolvê-lo. O medicamento genérico tem a mesma composição do medicamento ético, porém ele não teve um custo em pesquisa, devido a isso o seu preço é bem mais acessível.

“O mercado farmacêutico do país conta desde o dia 3 de fevereiro com três categorias de medicamentos: os remédios de marca, os genéricos e os similares” (ANVISA, 2000). Para que estes medicamentos fossem disponibilizados de forma correta e que fosse possível identificar o que é um medicamento ético, genérico e similar, o Ministério da Saúde instituiu um diferencial gráfico, assim é possível que o consumidor identifique facilmente a categoria do medicamento.

Assim, na embalagem do medicamento vem descrito “abaixo do nome do princípio ativo que identifica o produto, a frase ‘medicamento genérico – Lei 9.787/99’”. Esta descrição serve para identificar se o medicamento é genérico.

“Os remédios similares que até o dia 23 de janeiro eram comercializados somente pelo nome do princípio ativo estão obrigados a partir daquela data a adotar uma marca comercial ou agregar à denominação do princípio ativo o nome do laboratório fabricante.” (ANVISA, 2000).

Segundo a Anvisa, “As embalagens dos similares não têm nem terão a frase ‘medicamento genérico – Lei 9.787/99’” (2000). Então para que seja possível identificar se um medicamento é genérico ou similar, basta verificar se existe a mensagem, caso não exista, é similar.

Já para os medicamentos de marca, a mudança consiste na descrição do medicamento que, até o dia 23 de janeiro de 2000, tinha apenas 20% do nome da marca, teve de ser aumentado para 50%. Segundo a Anvisa (2000), “A modificação busca dar maior visibilidade ao nome do princípio ativo também nos medicamentos de marca.”

Atualmente, a Anvisa é responsável por todos os registros de medicamentos, autorização de funcionamento de laboratórios farmacêuticos e empresas da cadeia farmacêutica. A Anvisa também se responsabiliza pela regulação de ensaios clínicos, preços de medicamentos, por meio da Câmara de Regulação do Mercado de Medicamentos (CMED).

O texto a seguir tenta especificar cada categoria de medicamento demonstrando o que são medicamentos genéricos, similares e éticos/de marca.

#### ***a) Éticos/De Marca***

Segundo a Anvisa (2000), “É o produto inovador, cuja eficácia, segurança e qualidade foram comprovadas cientificamente por ocasião do registro”, este remédio surgiu inicialmente para curar uma determinada doença. Quando o remédio é ético, o laboratório que desenvolveu tem direitos sobre a patente do produto, sendo assim, atualmente no Brasil, o laboratório que desenvolveu tem direito a vinte anos de comércio deste medicamento sem concorrentes.

#### ***b) Genéricos***

O remédio genérico é um produto que possui as mesmas características e efeitos do produto ético. Para que o produto seja considerado um remédio genérico, o

Ministério da Saúde exige que “testes de bioequivalência farmacêutica para aprovar os genéricos.” (ANVISA, 2000).

Os testes de bioequivalência servem para verificar se dois produtos possuem “a mesma fórmula farmacêutica, composição, qualitativa e quantitativa, princípio ativo, são absorvidos na mesma proporção e velocidade pelo organismo.” (ANVISA, 2000).

*c) Similares*

Segundo definição da Anvisa,

medicamento similar é aquele que contém o mesmo ou os mesmos princípios ativos, apresenta mesma concentração, forma farmacêutica, via de administração, posologia e indicação terapêutica, e que é equivalente ao medicamento registrado no órgão federal responsável pela vigilância sanitária, podendo diferir somente em características relativas ao tamanho e forma do produto, prazo de validade, embalagem, rotulagem, excipientes e veículo, devendo sempre ser identificado por nome comercial ou marca. (2013)

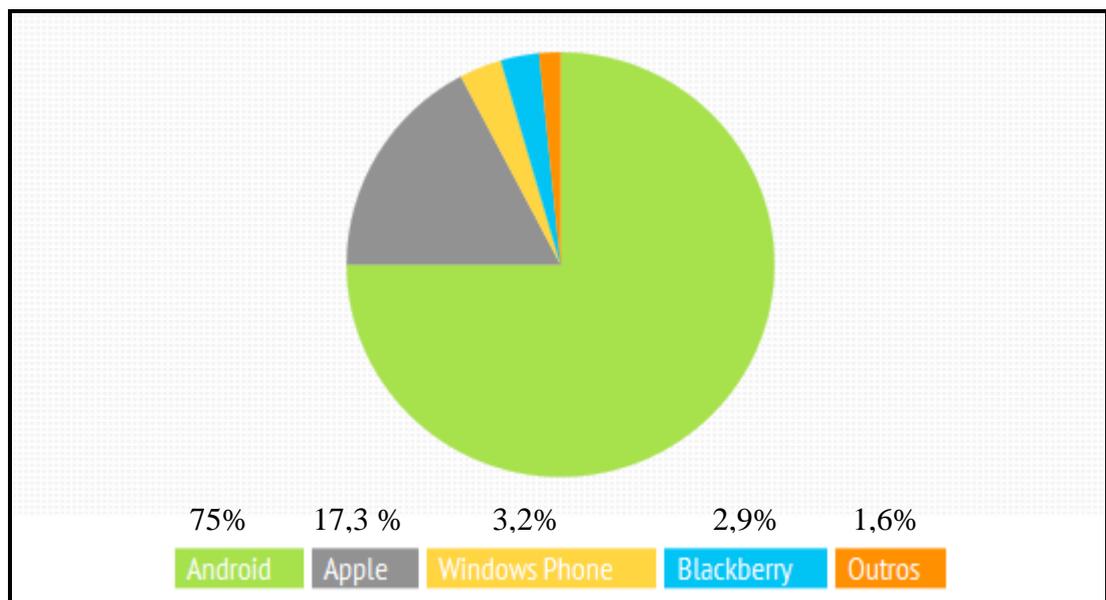
## **2.2 CRESCIMENTO NA VENDA DE SMARTPHONES**

A venda de dispositivos com Android vem aumentando diariamente. Porém a aquisição do dispositivo não está somente relacionada com os aparelhos em si, mas também ao SO (Sistema operacional) que ele possui instalado. Atualmente a Google com seu SO Android vem dominando o mercado, “O gigante da internet Google e seu sistema operacional Android confirmaram seu domínio no mercado dos smartphones no primeiro trimestre do ano, enquanto o Windows da Microsoft conquistou o terceiro posto atrás da Apple” (ANDROID, 2013). Nas pesquisas realizadas também foi constatado que o SO Android da Google estava presente em 3 a cada 4 aparelhos vendidos no mundo no primeiro trimestre de 2013.

Atualmente o SO Android já vem instalado em diversos dispositivos móveis, como smartphones e tablets. Pode-se destacar como principal fabricante e parceiro do Google a Samsung, empresa sul-coreana. A presença de smartphones com SO Android no mercado constitui um total de 75 % do total das vendas do primeiro trimestre de 2013. Segundo reportagem do jornal Zero Hora, referente à pesquisa junto a IDC (2013), “isso aumenta a distância entre seu principal rival, o grupo norte-americano Apple, que possui 17,3% do mercado com seu próprio sistema operacional iOS, utilizado no iPhone e iPad” (ANDROID..., 2013).

E a concorrência não para por aí, a Microsoft também está na briga e assumiu a terceira posição no mercado de smartphones, ficando assim com um total de 3,2 % do total das vendas do primeiro trimestre de 2013, isso graças aos aparelhos fornecidos por sua parceria, a empresa finlandesa Nokia. Por outro lado, a empresa canadense BlackBerry caiu um posto no primeiro trimestre de 2013, ficando com um total de 2,9 % das vendas. A figura 1 demonstra o mercado atual de smartphones.

**Figura 1:** Mercado Mundial de Smartphones



**Fonte:** do Autor

### 2.3 DESENVOLVIMENTO DE APLICATIVOS PARA SMARTPHONES

O desenvolvimento para aplicativos móveis pode ser realizado em diferentes linguagens de programação, cada linguagem oferece suas vantagens e desvantagens (MORANDI; RODOLPHO, 2013). As vantagens podem ser com relação ao desenvolvimento, custo inicial do desenvolvimento, maior público, capacidade de processamento do aparelho, layout e custo do aparelho móvel para os usuários.

- **Desenvolvimento:** na questão de desenvolvimento deve ser levada em consideração qual a linguagem que a plataforma fornece para desenvolver, APIs que se pode utilizar para facilitar o desenvolvimento e a IDE ferramenta que auxilia o desenvolvedor (MORANDI; RODOLPHO, 2013).

- **Custo inicial do desenvolvimento:** nesta parte entra a questão como: deve-se adquirir algum dispositivo específico para o desenvolvimento; o custo deste dispositivo é viável; sua aceitação e público são relevantes para que seja feita a aquisição. No que diz respeito à aquisição, entram os tablets e smartphones, público ao qual se deseja atingir com este projeto (MORANDI; RODOLPHO, 2013).
- **Maior público:** quando se fala no desenvolvimento para um maior público, fala-se de desenvolver aplicações para dispositivos móveis que tenham mais aceitação no mercado. Neste momento, o Google, com o SO Android, detém 75% do mercado, o que torna este sistema o que possui maior público no momento (ANDROID..., 2013).
- **Capacidade de processamento:** os dispositivos móveis hoje se comparam aos computadores, no momento em que se compra um aparelho já é analisado quanto de memória para armazenamento se tem, que processador possui, qual SO está instalado, quanto possui de memória interna, entre outros recursos oferecidos pelos dispositivos vendidos no mercado.
- **Custo do dispositivo móvel para o usuário:** desenvolver uma aplicação com grande capacidade de processamento pode exigir que o usuário faça a aquisição de um aparelho com um custo mais alto. Pensando neste problema, deve-se levar em consideração que o aparelho deve apenas consumir serviço, toda a parte de processamento e armazenamento de dados deve ser feito pelo servidor. Desta forma, pode-se ter uma aplicação rodando em diversos aparelhos e com uma capacidade de processamento mais baixa, lembrando que isto não é válido quando se fala de desenvolvimento para games, pois neste ponto exige-se alto desempenho e capacidade de armazenamento (MORANDI; RODOLPHO, 2013).

#### a) Android

O desenvolvimento para Android tem um custo menor, a plataforma é estável e, ao mesmo tempo, está em constante evolução, é mantida pela Google, além disso, o desenvolvimento para Android tem a facilidade de ser desenvolvido em Java e a própria Google já fornece um kit de desenvolvimento e tutoriais que ajudam quem está iniciando no desenvolvimento. Isso faz com que aplicativos sejam desenvolvidos em grande escala (MORANDI; RODOLPHO, 2013).

O Google oferece a facilidade de logo que as aplicações são desenvolvidas poderem ser postadas sem custo algum para o desenvolvedor ou dono do produto. Para isso, o Google criou o *Play Store*, onde as aplicações são postadas. Assim, quando alguém for postar uma aplicação, terá de assinar o contrato do Google, prometendo que o software está de acordo com as regras, caso alguma aplicação não esteja em pleno acordo com as regras do contrato, a mesma terá um prazo para se adequar, se isso não for atendido, a aplicação é descartada. Este controle é feito manualmente, sendo que desenvolvedores do Google avaliam os conteúdos, a Google mantém este controle para que as aplicações que não atendam aos termos sejam descartadas.

Quando um aplicativo é postado, tem de ser marcado o público para o qual foi desenvolvido, além de estar em comum acordo com as normas de desenvolvimento para que não seja retirado do *Play Store*.

Para se tornar um desenvolvedor Android basta ter uma máquina comum com um SO Windows ou Linux, baixar a IDE de desenvolvimento Eclipse (de preferência a mais atual), baixar o SDK do Android, ter o JDK (*Java Development Kit*) do Java mais atualizado e fazer alguns procedimentos para configurar o Android corretamente no Eclipse. Os procedimentos de configuração podem ser facilmente realizados, para isso basta seguir o manual de instalação e configuração oferecido pelo Google. Logo abaixo, segue uma lista de alguns itens que compõem as últimas versões do Android:

- **Multitarefa ou *multitasking***: se for selecionando o “*Running Apps*” irá aparecer uma lista com os aplicativos que foram usados recentemente, sendo possível que o usuário alterne entre as aplicações que está usando. Isso faz com que o usuário possa deixar suas aplicações rodando em segundo plano, assim o Android se assemelha ainda mais aos PCs (BOLTON, 2012).
- **Personalização: *Widgets*** redimensionáveis, sendo possível mexer no tamanho dos ícones que estão na home ou em outras divisões criadas pelo usuário (BOLTON, 2012).
- **Suporte *Adobe Flash***: o suporte para este *plugin* é uma importante escolha do Google, já que ainda existem muitos *websites* desenvolvidos utilizando *flash* para animação de páginas (BOLTON, 2012).

#### b) *iOS*

No desenvolvimento para *iOS*, o desenvolvedor tem de adquirir uma máquina MAC, e os dispositivos ao qual deseja desenvolver sua aplicação. Por exemplo, se o desenvolver optar por desenvolver um jogo, ele tem de adquirir um

iPhone ou iPad. Este custo a mais para o desenvolvimento pode se tornar um problema, pois no desenvolvimento para Android não se fica restrito a somente um dispositivo, não se vai estar apenas desenvolvendo para o dispositivo da Samsung, por exemplo, como é o caso do MAC.

O desenvolvedor também fica limitado para usar recursos do aparelho, um exemplo disso é o problema para poder acessar um recurso simples como o Bluetooth, dentre outros recursos que são facilmente manipulados com o Android.

O custo de um computador da Apple é mais elevado, e para testar sua aplicação, o desenvolvedor deve possuir o aparelho para o qual está desenvolvendo seu aplicativo. Neste ponto, o Android sai na frente, pois possibilita que o emulador Android consiga realizar testes em diferentes versões do aparelho e que possa emular tanto smartphones como tablets.

O *iOS 6* traz uma série de recursos que são usados diariamente, estes recursos podem ser baixados e instalados facilmente (BOLTON, 2012). Abaixo segue a descrição de alguns desses itens.

- **Mapas:** é possível visualizar os gráficos 3D, recurso possível graças ao Flyover que faz com que o usuário tenha uma visão de cima. Para acessar determinada área, basta um simples toque no local e começar a navegar, a visão 3D é uma forma de navegação mais atraente, o que deixa muitos usuários fascinados (BOLTON, 2012).
- **Mensagens:** As mensagens podem ser trocadas via aplicativo Messenger ou via aplicativos de comunicação com redes sociais, como o Facebook (BOLTON, 2012).
- **Multi-threads:** o iOS possibilita o funcionamento *multi-threads*, porém quando uma *thread* está executando, outras são colocadas para dormir, isso também ocorre em algumas versões mais antiga do Android, porém nas versões atuais o usuário pode acessar um aplicativo e ficar com ele rodando em *background* (segundo plano), enquanto navega no aparelho (BOLTON, 2012).

No momento, o desenvolvimento para IOs se torna inviável, pois o orçamento é grande, havendo possibilidade de não se ter retorno.

### c) **Windows Phone**

Trata-se de um sistema operacional móvel, que foi desenvolvido pela Microsoft, também conhecido como sucessor da plataforma WM (Windows Mobile).

Este novo produto é voltado para o mercado consumidor, o que o diferencia do seu predecessor que era voltado para o mercado empresarial.

O lançamento ocorreu na Europa, Austrália e Singapura no dia 21 de outubro de 2010, EUA e Canadá tiveram o lançamento no dia 8 de novembro de 2010, no dia 24 deste mesmo mês e ano também ocorreu o lançamento no México, no início de 2011, na Ásia. O WP (Windows *Phone*) apresenta uma nova interface gráfica, totalmente baseada nas versões do Windows atual (WINDOWS, 2013).

Uma das grandes dificuldades que foi encontrada nas versões atuais do WP é que os programas que rodavam anteriormente no WM não funcionam nesta nova plataforma. Este problema ocorreu porque o WP foi desenvolvido rapidamente e as mudanças de hardware para dispositivos móveis foram grandes, não houve condições de se manter compatibilidade com as antigas aplicações do WM. Abaixo seguem descritos alguns itens referentes às funcionalidades atualizadas no WP, que atualmente se encontra na versão 7.8 (KOLAKOWSKI, 2013).

- **Interface:** é possível criar e remover atalhos, mudar de posição, além de estar com uma interface totalmente voltada ao *Touch Screen*. Existe a possibilidade de baixar e instalar aplicativos como Facebook, Twitter, entre outras redes sociais e aplicativos que podem ser baixados e atualizados (DISPONIBILIDADE, 2014).
- **Textos:** quando se deseja utilizar a escrita de texto, o sistema oferece um teclado virtual, inclui botão para diversos *emotions*, além de previsão de escrita, o que pode auxiliar muito na escrita de um texto.
- **Mensagens:** é possível utilizar várias ferramentas de comunicação integradas, é possível utilizar o Windows Live Messenger e *chat* Facebook, tudo isso integrado com as mensagens enviadas pelo aparelho. Esta é uma opção que já é usada por todos os SOs mais novos.
- **Navegação na internet:** O WP atualmente tem embutido o navegador IE 9, no navegador é possível adicionar sites aos favoritos, excluir histórico de navegação, configurar Proxy, entre outras configurações que são utilizadas na versão para computadores.
- **Contatos:** os contatos podem ser importados ou inseridos manualmente, centralizando assim a lista de contatos existentes. Também existe a

possibilidade de se criar grupos, onde as informações dos contatos são combinadas.

- **E-mail:** email do hotmail, Yahoo! Mail e outros serviços de email conhecidos no mercado podem ser configurados e acessados facilmente.
- **Multimídia:** músicas e vídeos podem ser armazenados e manipulados facilmente no aparelho. Além de ter a possibilidade baixar aplicativos onde o usuário pode editar imagens tiradas da câmera do aparelho, esta edição pode acontecer em tempo real, no caso, na hora em que se está tirando a foto, ou após a foto ser tirada. Claro que existem as limitações de processamento do aparelho, e a complexidade do *software* que está operando sobre o SO.
- **Áudio:** Os formatos de ficheiro de áudio que são suportados incluem AVI, MP4/M4V, WAV, AMR, AAC/MP4/M4A/M4B e 3GP/3G2.
- **Vídeo:** os formatos de ficheiros de vídeo são WMV, AVI, MP4/M4V, 3GP/3G2 e MOV (QuickTime). Os suportes de vídeos descritos aqui são dependente de *codecs* DivX e Xvid.
- **Imagem:** no que diz respeito às imagens, existem os JPG/JPEG, PNG, GIF, TIF e Bitmap.
- **Reconhecimento de voz:** o sistema permite o reconhecimento de voz, sendo assim o usuário pode acessar o buscador Bing e falar o tipo de pesquisa que deseja realizar.
- **Suite Microsoft Office:** é possível criar documentos formatados assim como se estivesse usando o Word, porém com limitações, ainda existe possibilidade de se armazenar todos os documentos criados no SkyDrive, desta forma os documentos Word, Power Point, entre outros tipos de documentos que podem ser criados, não ficam acumulando espaço em disco no aparelho e ainda possibilita o controle de versão dos arquivos criados (DISPONIBILIDADE, 2014).

## 2.4 WEB SERVICES

*Web Services* tem sido utilizado em aplicações comerciais, permitindo assim que vários sistemas escritos em linguagens distintas possam se comunicar por meio de serviços

(métodos) que ficam expostos para que outros sistemas ou módulos possam acessá-los (MEDEIROS, s.d.). Para isso, a utilização da tecnologia XML associada ao protocolo SOAP (*Simple Object Access Protocol*) constitui uma poderosa ferramenta de comunicação *web services*. Esta ferramenta é utilizada como um canal comum de comunicação, assim as duas juntas (XML e SOAP) constituem a arquitetura SOA (*Service Oriented Architecture*) (WEB).

#### **2.4.1 Consumindo *web services* com *android***

Existem várias formas de se consumir um *web services*, porém quando se fala em aplicações para dispositivos móveis, alguns padrões de formato de dados não são levadas em consideração. Isso devido ao fato de que os dispositivos móveis possuem menor processamento e espaço de armazenamento que PCs.

No momento, o padrão de envio de dados JSON tem sido muito utilizado para o desenvolvimento de aplicativo móvel que consomem *web services*, pois possuem uma compactação maior que a de outros formatos (MEDEIROS, s.d.). O XML, por exemplo, é todo baseado em abrir e fechar *tags*, isso faz com que o texto gerado seja maior, o que torna inviável o seu uso quando se deseja consumir um *web services* com aplicativos móveis, pois demandam uma taxa de transferência de dados maior do que a usada para enviar no formato JSON.

#### **2.4.2 JSON**

JSON pode ser descrito como um modelo para armazenamento de informações e troca das mesmas no formato de texto. Mesmo sendo mais simples que o XML, tem sido frequentemente usado devido à capacidade de compactar os dados em menor tamanho que outros padrões, como XML. Algumas empresas como Google e Yahoo já fazem uso de JSON em seus projetos (MACEDO, s. d).

A sintaxe é derivada do *JavaScript* na representação de informações, onde se utiliza um par nome/valor, o nome deve ser representado entre aspas duplas e depois, seguido por dois pontos, o valor do nome (MACEDO, s.d.).

Os valores podem possuir apenas três tipos: booleano, string e numérico (inteiro ou real). A partir dos tipos simples é possível chegar a tipos mais complexos, como *arrays* e objetos. Os *arrays* são delimitados por colchetes e seus elementos/objetos são separados por colchetes (MEDEIROS, s. d.).

Já os objetos são especificados entre chaves, podem ser compostos por múltiplos pares nome/valor, por outros objetos e por *arrays*. Desta forma, um objeto JSON pode representar, virtualmente, qualquer tipo de informação (MACEDO, s.d).

O JSON foi construído com objetivo principal de conseguir trabalhar de forma mais simples com os dados e de ser mais compacto. Para o desenvolvimento de aplicações para dispositivos móveis, ele vem a calhar, pois estes aparelhos, em sua maioria, não possuem grande processamento e têm pouco espaço para o armazenamento de dados (MACEDO, s.d.).

#### **2.4.5 XML**

Na década de 1990 surge a *World Wide Web Consortium* (W3C), e no ano de 1996, a W3C desenvolveu a linguagem XML, que combinava a flexibilidade da *Standard Generalized Markup Language* (SGML) com a simplicidade do *Hypertext Markup Language* (HTML). No início, o projeto objetivava criar uma linguagem que fosse possível de ser lida por software, interagindo com outras aplicações (LIMA; CARVALHO, 2005, p. 3).

É uma linguagem padronizada para descrever diversos tipos de dados, tendo como objetivo facilitar o compartilhamento de informações na internet. Sendo possível de se organizar os documentos de forma hierárquica, sendo independente do *software* ou *hardware* que manipula o documento XML (CONSTANTINO FILHO, 2002).

### **2.5 ANDROID GEOLOCALIZAÇÃO**

Com a API de geolocalização do Android é possível localizar um dispositivo móvel, este recurso é usado para que as pessoas possam se localizar e localizar um determinado ponto (DRABACK). Neste projeto, será usado a geolocalização para que usuário possa identificar qual o ponto em que ele está, e qual a farmácia que está mais acessível.

Desta forma, o usuário irá fazer a pesquisa pelo produto, sendo apresentado o produto e a localização das farmácias e ainda a farmácia mais perto do ponto onde o dispositivo está executando a pesquisa. A seguir, encontram-se algumas APIs que são utilizadas para obter a geolocalização no Android.

### a) **Android Location**

Esta API contém classes que definem a localização do aparelho (DRABACK). A API *Location Services* do *Google Play Services* fornece este *framework* de alto nível que faz a pesquisa do provedor (ANDROID.LOCATION).

Lista de interfaces que fazem parte:

- **GpsStatus.Listener**: usado para receber notificações, usando o GPS (*Global Positioning System*) para ir atualizando o status.
- **GpsStatus.NmeaListener**: usado para receber sentenças NMEA do GPS.
- **LocationListener**: utilizado para receber notificações quando a *LocationManager* for alterada.

Lista de classes que compõem esta API:

- **Address**: que representa um endereço ou pode ser descrito como um conjunto de informações que descrevem um local.
- **Criteria**: indica os critérios da aplicação para a seleção de um provedor de localização.
- **Geocoder**: para manipulação de geocodificação e reverter a geocodificação.
- **GpsSatellite**: representa o estado atual GPS satélite.
- **GpsStatus**: representa o estado atual do motor do GPS.
- **Location**: dados que representam a localização geográfica.
- **Location Manager**: fornece acesso de localização de serviços do sistema.
- **Location Provider**: superclasse abstrata para provedores de localização.

### b) **Class LocationManager**

Esta classe fornece acesso aos serviços de localização do sistema (ANDROID.LOCATION). Desta forma, este serviço permite que aplicativos obtenham atualizações periódicas da localização geográfica do dispositivo, ou possam disparar/enviar algum dado para a Internet, sendo utilizado pelos aplicativos quando o dispositivo entra em uma determinada localização geográfica (DRABACK, 2012).

Para fazer uso, não existe a necessidade de instanciar diretamente, pode ser apenas recuperá-la com o comando *Context.getSystemService* (*Context.LOCATION\_SERVICE*).

c) ***Class LocationProviders***

Uma superclasse abstrata para os provedores de localização que tem como funcionalidade fornece relatórios periódicos a respeito da localização geográfica do dispositivo (DRABACK, 2012).

Cada provedor/fornecedor tem um conjunto de critérios, o qual para ser possível utilizar exige-se que o usuário disponha de alguns recursos de hardware como GPS, que esteja visível para os satélites (DRABACK, 2012). Outros provedores necessitam que o dispositivo esteja conectado a uma rede de transporte específico, ou para a internet.

d) **Permissões no Android**

Para que um aplicativo possa acessar qualquer que seja o recurso do sistema, o aparelho deve solicitar acesso através da classe *Manifest.permission* (DRABACK, 2012), aqui estão contidos todas as constantes necessárias para solicitar permissão especial de recursos do *hardware* do dispositivo. Os recursos oferecidos são o controle de bateria, áudio, vídeo, entre outros recursos de hardware que são oferecidos (ANDROID.LOCATION).

## 2.6 BANCO DE DADOS RELACIONAIS

Os bancos de dados começaram a ser utilizados comercialmente nos anos 60. Nestes primeiros anos, as informações eram guardadas em ficheiros e a manipulação era muito pouco utilizada (Osvaldo Kotaro Takai e col., 2014). Já no início dos anos 70, surgiram os Sistemas de Gerenciamento de Banco de Dados (SGBD) relacionais, sua popularidade não para de crescer até hoje. Este sucesso é devido à simplicidade do modelo relacional, este modelo é basicamente constituído por relações; e também ao fato de ter surgido uma linguagem de manipulação simples e eficiente, o SQL (*Structured Query Language*) (Osvaldo Kotaro Takai e col., 2014). Na figura 2, pode ser visualizado como funciona um BD.

Abaixo estão descritos alguns itens que compõem um banco de dados.

- **Dado:** é o valor do campo que é armazenado no banco de dados (COSTA, 2011, p. 10).
- **Tabelas Lógicas:** Estruturas de armazenamento de dados, arquivos de sistemas.

- **SGBD:** Software responsável pelo gerenciamento dos dados no Banco de Dados, permitindo que usuários possam definir, manipular e construir uma base de dados (Osvaldo Kotaro Takai e col., 2014).
- **Conteúdo de campo:** Valor do campo que é armazenado no banco de dados (COSTA, 2011, p. 11).
- **Modelo de Banco de Dados:** Modelo Relacional, Modelo Hierárquico e Modelo em Rede. Representa a estrutura física na qual o armazenamento dos dados foi projetado. O modelo identifica a estrutura interna de recuperação e armazenamento dos dados na qual o SGBD foi projetado.
- **DDL (*Data Definition Language*):** Linguagem que define as aplicações, campos e tabelas que irão compor o banco de dados (COSTA, 2011, p. 39).
- **DML (*Data Manipulation Language*):** Comandos de manipulação do banco de dados e que definem operações que devem ser executadas pelo banco de dados (COSTA, 2011, p. 39).
- **SQL:** É a linguagem padrão ANSI (American National Standards Institute) para a manipulação de banco de dados relacionais (COSTA, 2011, p. 39).
- **PL/SQL (*Procedural Language extensions to SQL*):** Até hoje entendida como uma extensão da linguagem SQL, com funcionalidades que a tornam uma linguagem de programação completa: tratamento de exceções, controle de fluxo, entre outras, orientação a objetos. (RIELAU, 2013) Com a PL/SQL, pode-se escrever programas inteiros, desde os programas mais simples até os mais sofisticados. Esta linguagem foi criada com o propósito de oferecer soluções de programação para os usuários que precisavam escrever aplicações de missão crítica executadas no SGBD (PL/SQL).

## 2.2 RECURSOS UTILIZADOS

Através do estudo aplicado em tecnologias atuais, optou-se pelo uso dos seguintes recursos no desenvolvimento dos aplicativos

- Linguagens: JAVA, HTML, CSS (*Cascading Style Sheets*) e JavaScript;
- Bibliotecas: GSON, SDK Android, Jersey e Hibernate;
- IDEs: Eclipse, Netbeans;
- Servidor: Apache.

### 2.2.1 Linguagem de programação utilizada

Linguagem de programação é um método padronizado para comunicar instruções de um programa para um computador, formada por regras sintáticas e semânticas que são utilizadas para definir um programa. Através da linguagem de programação, um programador pode definir sob qual plataforma o programa gerado pode atuar, transmitindo e recebendo dados, local de armazenamento de informações e quais ações devem ser tomadas (WILLRICH, 2014).

No desenvolvimento deste projeto, optou-se pela linguagem de programação Java, pois esta linguagem oferece diversas APIs que auxiliam o desenvolvedor na construção de um projeto.

Esta linguagem é orientada a objetos, desenvolvida na década de 90, por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. (APOSTILA, 2013). Participaram como mentores do projeto Patrick Naughton, Mike Sheridan e James Gosling.

Algumas das principais características que diferencia a linguagem Java de outras são:

- Linguagem não compilada para o código nativo;
- Código Java é compilado para *bytecodes*;
- Os *bytecodes* são executados por uma máquina virtual;
- Linguagem orientada a objetos;
- Portabilidade – independência de plataforma, “escreva uma vez, execute em qualquer lugar” (“*write once, run anywhere*”);
- Recursos de Rede – possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP;
- Segurança – pode executar programas via rede com restrições de execução.

Para o desenvolvimento deste projeto, a linguagem Java é essencial, tendo em vista que o desenvolvimento para Android utiliza, na parte de processamento, esta linguagem.

### 2.2.2 Gson

GSON é uma biblioteca desenvolvida em java que realiza a conversão de objetos java para a forma de texto no padrão JSON. Esta biblioteca fornece envio de dados na forma

de texto com menor tamanho que o XML. Desta forma tem melhor desempenho no envio dados se comparado com XML.

É uma API que busca facilitar o trabalho do desenvolvedor, fazendo o *casting* de Objetos Java para o formato JSON (PARSING, 2013). Também é possível converter JSON em formato de texto para um Objeto Java. (MEDEIROS, s.d.).

O código fonte desta API é aberto, sendo assim o projeto vive em constante evolução. Esta API é um projeto que foi desenvolvido com o objetivo de fazer *casting* de objetos para string em um formato simples, auxiliando no desenvolvimento das aplicações para dispositivos móveis (MEDEIROS, s.d.).

Funcionalidades API GSON:

- converter string em modo texto no formato JSON para objetos java;
- possuir amplo suporte de genéricos do Java;
- permitir representação customizada para objetos.

O que justifica a escolha pelo padrão do formato JSON é a taxa de compactação, pois é mais elevada do que outros padrões e ocupa um menor espaço em disco, como é o caso do XML. A escolha da API GSON se deve justamente ao fato de fazer a formatação de dados do arquivo em um formato menor.

O seu uso no projeto foi adicionado em dois cenários, o primeiro foi no *web services* que realiza a leitura de todos os *web services* das farmácias disponíveis e envia os mesmo para o cliente Android; o segundo é nos *web services* das farmácias que estão fornecendo serviço para o acesso do aplicativo aos medicamentos.

Os principais métodos utilizados no desenvolvimento são:

- **toJson(Object):** Este método é responsável por realizar a conversão de um objeto java para uma string no padrão JSON.
- **fromJson(String, Class):** Neste método acontece o inverso do método toJson(Object). Este método recebe dois parâmetros, o primeiro é a string que deve ser passada no formato Json, o segundo é o tipo de objeto ao qual se deseja converter a string.

### 2.2.3 Jersey e arquitetura *restfull*

O desenvolvimento de *web services* é uma das formas de integrar diferentes aplicações. Para realizar o desenvolvimento de um serviço web, existem diferentes tipos de arquiteturas e o *RESTfull* é uma das mais simples se comparada com arquiteturas de *web services* que utilizam SOA. (MACEDO, 2014).

*RESTfull* é uma implementação de *web services* simples utilizando o HTTP e os princípios de *REST*. Não necessariamente devem ser utilizados todos os métodos oferecidos pela arquitetura REST, mas o melhor que ele pode oferecer.

Para o desenvolvimento dos *web services* neste projeto, optou-se pela utilização do Jersey, que é uma API que implementa JAX-RS (*Java API for RESTful Web Services*), e que é a especificação (a JSR-311) que define a criação de um *WebService* com arquitetura *RESTfull*. (RONDON, 2014).

Algumas justificativas para a escolha da API Jersey:

- *Open Source*;
- É uma implementação de referência do JAX-RS;
- Faz uso de anotações para identificação de classes, operações e atributos;
- Simplificação das configurações;
- Mantida pela *Oracle*;
- Documentação sempre acessível;
- Uso de comunicação em formato JSON;

#### 2.2.4 Hibernate

Hibernate é um framework desenvolvido para realizar a persistência de forma Objeto/Relacional de Mapeamento (ORM) por meio do Java. O termo ORM refere-se à técnica de mapeamento de dados, representada desde o objeto modelo aos dados relacionais modelo com um esquema baseado na SQL. (Gavin King e col., 2014)

O Hibernate não fica apenas no controle do mapeamento das classes e seus atributos, mas também na inserção, alteração, exclusão e recuperação dos dados.

Segundo Gavin King e col. (2014), “A meta de Hibernate é aliviar o desenvolvedor em 95% de dados comuns de persistência relacionados às tarefas de programação.”

Neste projeto o Hibernate teve um papel importante para o rápido desenvolvimento dos projetos, além da segurança fornecida por ele na persistência. Seu uso foi aplicado no desenvolvimento das seguintes aplicações: Sistema de Cadastro de Medicamentos, Sistema de Cadastro de *Web Services*, *Web Services* das Farmácias e *Web Services* que listam as farmácias.

### 2.2.5 Servidor de aplicação Apache Tomcat

O Tomcat é um servidor web Java, mais especificamente um container de *servlets* (APACHE, 2014).

Seu uso foi aplicado para o desenvolvimento nas aplicações de cadastro de *web services*, cadastro de medicamentos e também para o desenvolvimento dos *web services* que listam as farmácias e o *web services* que representa um *web services* de farmácias (MEDEIROS, 2014).

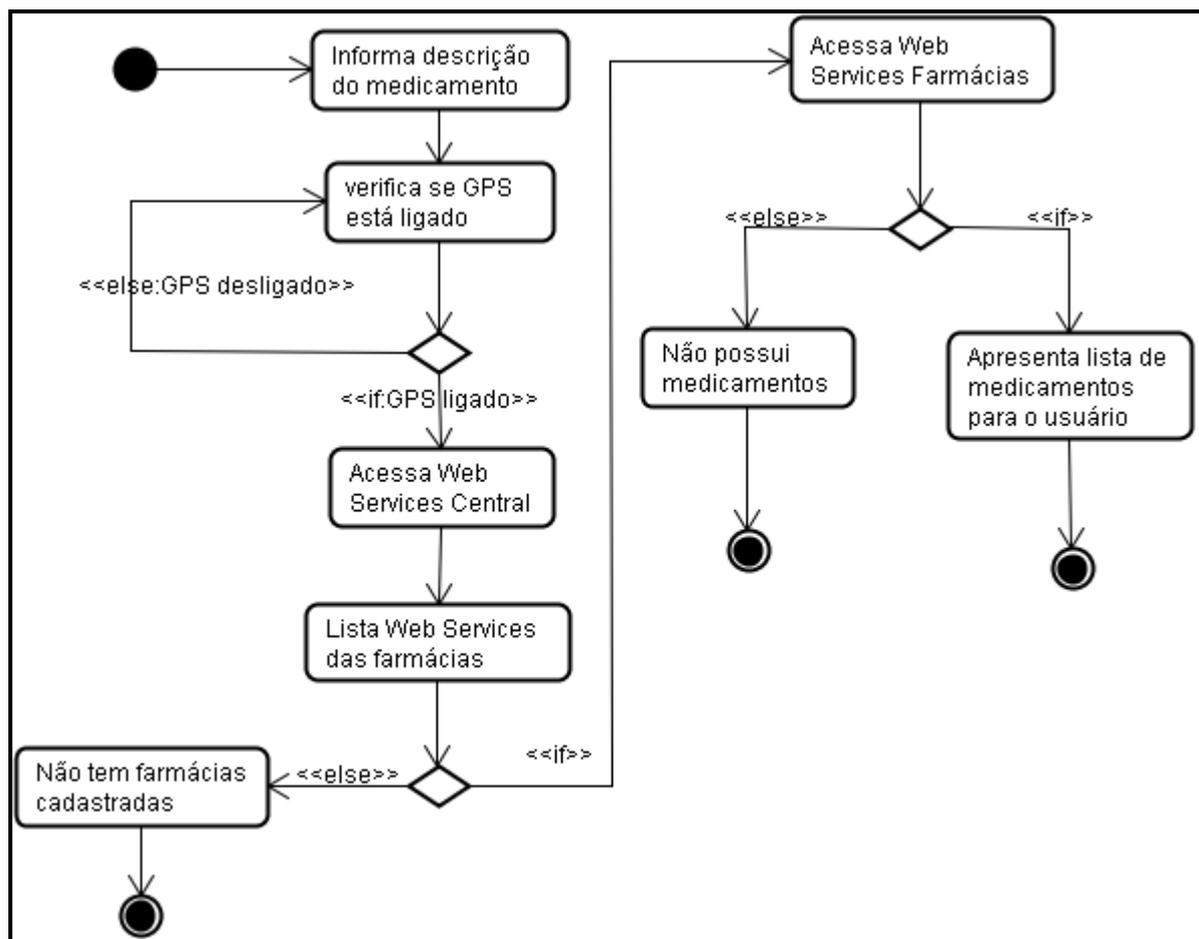
A escolha pelo desenvolvimento das aplicações utilizando o *container* Tomcat se deve à quantidade de recursos oferecidos. Estes recursos são:

- Segurança;
- Facilidade de configurar;
- Comunidade sempre ativa;
- Ferramenta sempre atualizada.

### 3 DESENVOLVIMENTO METODOLÓGICO

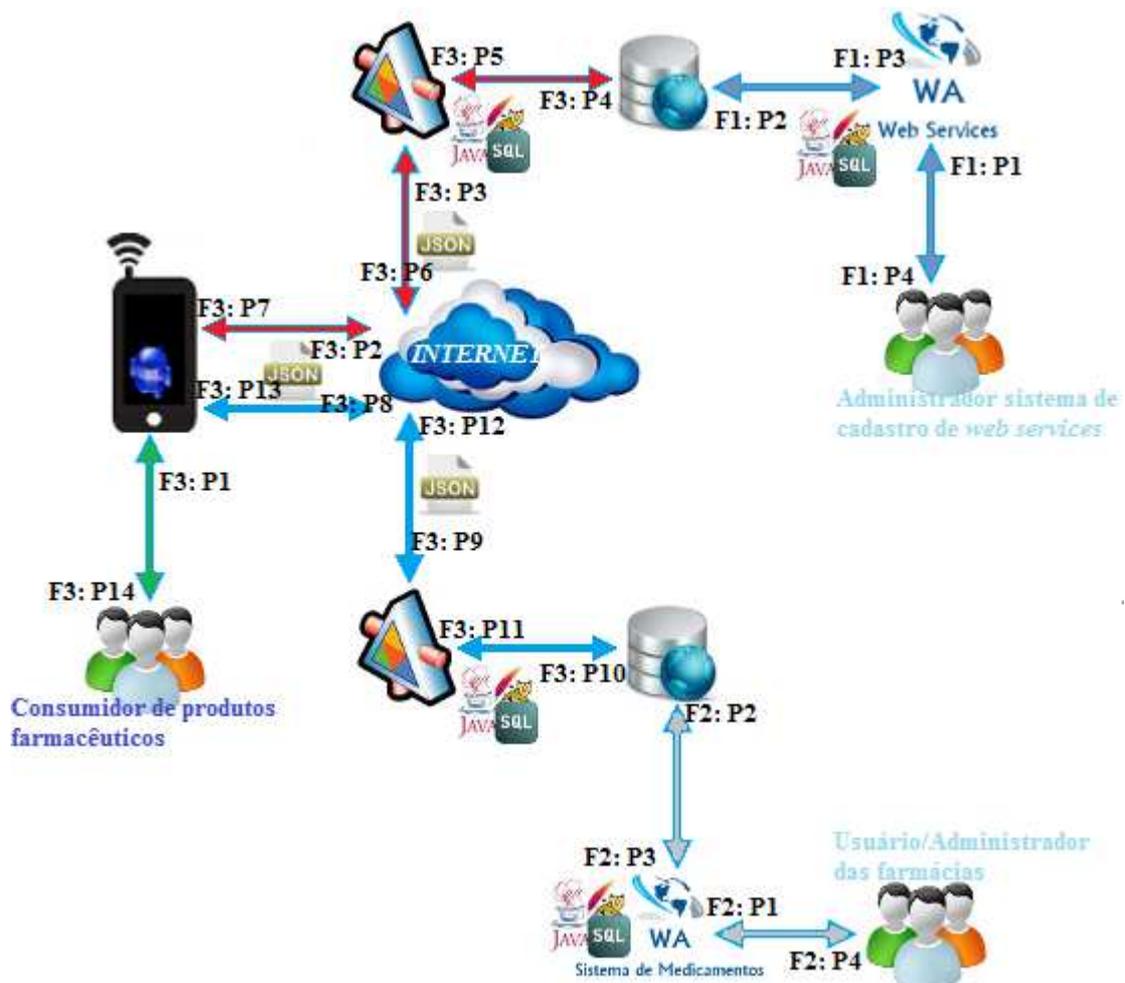
O sistema desenvolvido é composto por cinco módulos, sendo eles: um aplicativo desenvolvido em Android, este aplicativo realiza o acesso ao Serviço Web Central para a leitura de todos os *web services* das farmácias disponíveis. O segundo módulo é a aplicação que realiza o cadastro de *web services*, a terceira aplicação é o sistema de cadastro de medicamentos representando um sistema de cadastro das farmácias, e os outros módulos são dois *web services*, sendo um deles para ser acessado em primeiro momento pelo aplicativo e outro sendo a representação de um *web services* das farmácias. Na figura 2 pode ser visualizado o diagrama de atividades e na figura 3 uma visão geral do sistema. E nas próximas seções será apresentado cada um dos módulos.

**Figura 2:** Diagrama de Atividades



Fonte: do Autor

Figura 3: Representação Geral do Sistema



Fonte: do Autor

Fluxo de cadastro de *web services*:

- F1:P1 - Usuário visualiza o formulário de listagem de *web services* e escolhe procedimento a ser realizado, alterar/cadastrar/excluir;
- F1:P2 - Depois de realizada a confirmação pelo usuário, os dados são enviados para o banco de dados;
- F1:P3 - Sistema mostra mensagem informando se o procedimento foi realizado com sucesso ou não;
- F1:P4 - Usuário confirma se operação está de acordo com o que foi realizado.

Fluxo de cadastro de medicamentos:

- F2:P1 - Usuário informa função a ser realiza pelo sistema, alterar/excluir/cadastrar;
- F2:P2 - Sistema envia os dados para a base de dados;

- F2:P3 - Sistema mostra se o procedimento realizado pelo usuário foi realizado com sucesso ou não;
- F2:P4 - Usuário visualiza mensagem e apresentada pelo sistema e confirma se está tudo certo.

Fluxo de acesso do usuário ao aplicativo Android e a comunicação com *web services*:

- F3:P1 - Usuário informa descrição do medicamento a ser pesquisado. Aplicativo verifica se GPS está ligado, caso não esteja, apresenta mensagem para que o usuário ligue-o;
- F3:P2 - Aplicativo envia as coordenadas de onde se encontra o dispositivo para o *web services*;
- F3:P3 - *Web services* recebe os dados enviados pelo dispositivo;
- F3:P4 - Serviço realiza pesquisa das farmácias cadastradas naquela região em que se encontra o dispositivo;
- F3:P5 - Serviço recebe os dados do banco e realiza formatação para retorno dos dados ao aplicativo;
- F3:P6 - *Web services* envia dados em formato JSON;
- F3:P7 - Aplicativo recebe os dados em formato JSON e realiza processamento e consulta do medicamento em cada farmácia;
- F3:P8 - Aplicativo acessa os *web services* de cada farmácia, enviando a descrição do medicamento a ser pesquisado;
- F3:P9 - *Web services* das farmácias recebe a descrição do medicamento a realiza processamento dos dados;
- F3:P10 - *Web services* se comunica com o banco de dados;
- F3:P11 - *Web services* processa informações retornada da base de dados;
- F3:P12 - Serviço realiza envio de dados em formato JSON;
- F3:P13 - Aplicativo recebe lista de medicamentos de cada farmácia e processa as informações para apresentação;
- F3:P14 - Aplicativo mostra para o usuário a lista de medicamentos.

### **3.1 SISTEMA DE CADASTRO DE MEDICAMENTOS**

Este sistema foi desenvolvido para poder realizar o cadastro de medicamentos das farmácias, este software tem por objetivo ser um modelo que represente o cadastro de medicamentos de uma farmácia. Não sendo um sistema de farmácia em sua originalidade. Desta forma o sistema tem por objetivo apresentar as funcionalidades necessárias para que seja possível inserir, alterar, deletar e listar os medicamentos.

Para realizar a inserção de um medicamento o usuário deve inserir os seguintes dados: descrição, concentração, forma farmacêutica, conteúdo, disponibilidade, valor.

Quando o usuário for inserir os dados ele deve estar ciente que as informações por ele fornecidas devem estar em total acordo com o medicamento cadastrado.

#### **3.1.1 Diagrama de caso de uso**

Caso de uso é a descrição de uma sequência de eventos que envolvem atores e requisitos funcionais, com o objetivo de apresentar as atividades que são realizadas pelo sistema, facilitando assim o entendimento dos processos realizados pelos sistemas.

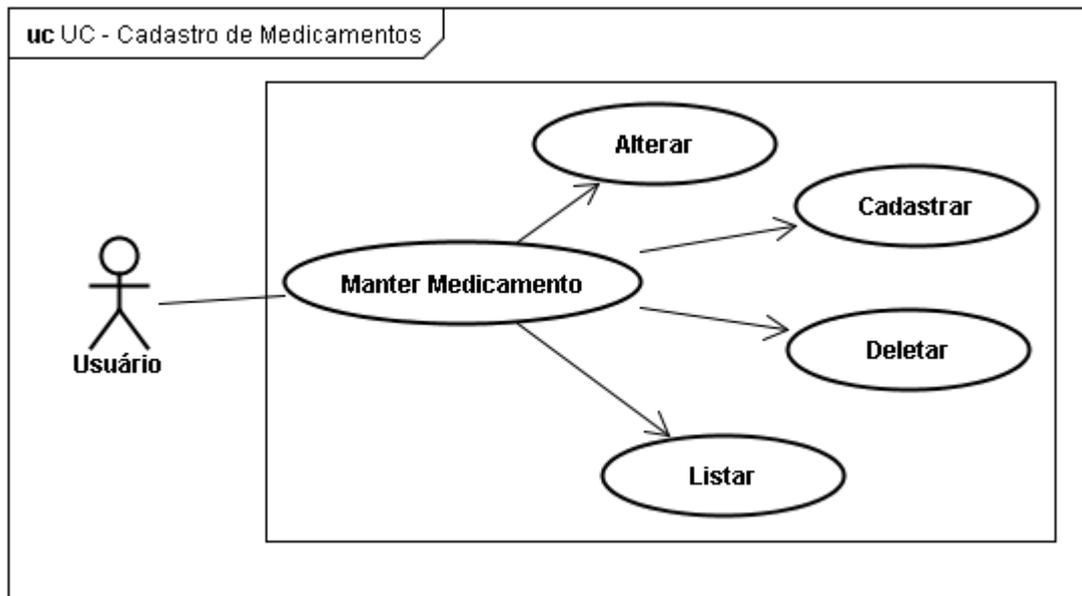
Neste sistema o ator é um usuário qualquer da farmácia que seja responsável por realizar as atividades, não havendo restrição por tipo de usuário.

Interações realizadas pelo usuário após cadastro ser aprovado pelo administrador:

- Cadastrar medicamento;
- Listar medicamentos;
- Alterar medicamento;
- Excluir Medicamento.

Na figura 4 ser visualizado o caso de uso do sistema. A ideia aqui é proporcionar uma visão de como ocorre a comunicação entre o usuário e o sistema de cadastro de medicamentos.

**Figura 4:** Caso de uso sistema de cadastro de medicamentos



Fonte: do Autor

A descrição do Caso de Uso do sistema de cadastro de medicamentos tem como finalidade descrever o cenário de interação entre o ator e o sistema.

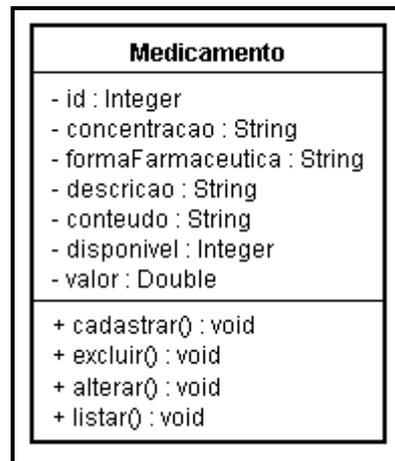
<b>Nome do Caso de Uso</b>	Manter Medicamento
<b>Sumário</b>	Usuário utiliza o sistema para realizar o cadastro de medicamentos.
<b>Ator primário</b>	Usuário de farmácia
<b>Ator secundário</b>	Não existe ator secundário.
<b>Precondições</b>	Apenas ter acesso ao sistema.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema apresenta formulário de listagem de medicamentos.</li> <li>2. Usuário seleciona ação cadastrar medicamento.</li> <li>3. Usuário seleciona ação de alterar medicamento. Para realizar este procedimento o usuário deve selecionar um medicamento no formulário de listagem de medicamentos.</li> <li>4. Usuário seleciona ação de excluir medicamento.</li> </ol>
<b>Fluxo Alternativo</b>	<p><b>(2) Cadastrar Medicamento</b></p> <ol style="list-style-type: none"> <li>a. Sistema apresenta formulário de cadastro de medicamentos.</li> <li>b. Usuário clica no botão gravar, sistema valida os dados do</li> </ol>

	<p>medicamento conforme RN01, RN02, RN03 RN04, RN05 e RN06. Se os dados estiverem corretos o sistema volta para o formulário de listagem e apresenta a MSG01.</p> <p>c. Usuário clica no botão cancelar, sistema apresenta janela questionando se realmente o usuário deseja cancelar a operação. Caso o usuário clicar em sim, o sistema retorna o formulário listagem de medicamentos, caso não, o sistema retorna para o formulário de cadastro.</p> <p><b>(3) Alterar medicamento</b></p> <p>a. Usuário seleciona o medicamento a ser alterado na janela de listagem de medicamentos.</p> <p>b. Sistema apresenta os dados do medicamento no formulário de cadastro de medicamento.</p> <p>c. Usuário altera os dados do medicamento e clica no botão gravar. O sistema valida os dados conforme RN01, RN02, RN03, RN04, RN05, RN06. Se obtiver êxito, o sistema retorna o usuário para lista de medicamentos e apresenta a MSG02.</p> <p>d. Usuário seleciona opção cancelar, sistema apresenta janela para o usuário selecionar se realmente deseja cancelar a operação, se sim o sistema volta para o formulário de listagem de medicamentos, senão o sistema retorna para o formulário de alteração de medicamento.</p> <p><b>(4) Excluir medicamento</b></p> <p>a. Usuário seleciona o medicamento no formulário de listagem de medicamento, clicando na coluna excluir. Sistema apresenta janela questionando se realmente o usuário deseja excluir o medicamento. Se o usuário clicar em sim, o sistema realiza a exclusão do medicamento e apresenta a MSG08 e continua no formulário de listagem de medicamento.</p> <p>b. Usuário clica no botão cancelar. Sistema retorna para o formulário de listagem de medicamento</p>
<b>Pós-condições</b>	O medicamento foi incluído, alterado ou excluído da base de dados.
<b>[RN] Regras de negócio</b>	RN01 – A descrição do medicamento deve ser informada, caso não informado o sistema apresenta a MSG03.

	<p>RN02 – A concentração do medicamento deve ser informada, caso não informada o sistema apresenta a MSG04.</p> <p>RN03 – A forma farmacêutica deve ser informada, caso não informada o sistema apresenta a MSG05.</p> <p>RN04 – O conteúdo deve ser preenchido, caso não informado o sistema apresenta a MSG06.</p> <p>RN05 – O valor deve ser preenchido, caso não informado o sistema apresenta a MSG07.</p> <p>RN06 – Disponibilidade do medicamento deve vir pré definida como sim.</p>
<b>Requisitos funcionais</b>	<b>não</b> Não definidos.
<b>[MSG] Mensagens</b>	<p>MSG01 – Medicamento cadastrado com sucesso.</p> <p>MSG02 – Medicamento alterado com sucesso.</p> <p>MSG03 – A descrição deve ser informada.</p> <p>MSG04 – A concentração deve ser informada.</p> <p>MSG05 – A forma farmacêutica deve ser preenchida.</p> <p>MSG06 – O conteúdo deve ser informado.</p> <p>MSG07 – O valor deve ser preenchido.</p> <p>MSG08 – Medicamento excluído com sucesso.</p>
<b>Autor</b>	do Autor
<b>Data</b>	10/07/2014

### 3.1.2 Diagrama de Classes do sistema de cadastro de medicamentos

O diagrama de classes do sistema de cadastro de medicamentos é simples, pois trata-se de apenas uma classe que foi desenvolvida para representar os dados referentes aos medicamentos das farmácias. Na figura 5, pode ser visualizado os atributos e as operações realizadas pela classe medicamento.

**Figura 5:** Diagrama de classes de cadastro de medicamentos

**Fonte:** do Autor

Na figura 6, pode ser visualizado formulário de cadastro de medicamentos e na figura 7 o formulário que apresenta a lista de medicamentos.

**Figura 6:** Formulário de cadastro de medicamentos

The screenshot shows a web form titled "Edição de Medicamento". It contains several input fields: "ID", "Descrição", "Concentração", "Forma Farmacêutica", "Conteúdo", and "Valor \*". The "Valor \*" field has a dropdown menu with "Obrigatório" selected. There is also a "Disponível" section with radio buttons for "Sim" and "Não". At the bottom, there are two buttons: "Cancelar" and "Gravar".

**Fonte:** do Autor

**Figura 7:** Formulário de listagem de medicamentos

Listagem de Medicamentos <span style="float: right;">Novo</span>								
ID	Descrição	Concentração	Forma Farmacêutica	Conteúdo	Valor	Disponível	Alterar	Excluir
1	Paracetamol	750 mg	Comp. Rev	Cx. c 12	10.0	1	<a href="#">Alterar</a>	<a href="#">Excluir</a>
2	Paracetamol	500 mg	Comp.	Cx. c 10	7.2	1	<a href="#">Alterar</a>	<a href="#">Excluir</a>
3	Paracetamol	250 mg	Comp.	Cx. c 10	6.3	1	<a href="#">Alterar</a>	<a href="#">Excluir</a>
4	Paracetamol	300 mg	Comp.	Cx. c 10	6.12	1	<a href="#">Alterar</a>	<a href="#">Excluir</a>

**Fonte:** do Autor

## 3.2 SISTEMA DE CADASTRO DE *WEB SERVICES*

Este sistema foi desenvolvido para realizar o cadastro dos *web services* das farmácias. Este sistema não exige uma autenticação de usuário para realizar o cadastro das farmácias.

Após o cadastro do serviço web, a farmácia já fica disponível para consulta dos medicamentos. Sendo assim, quando o aplicativo Android realizar uma consulta, o mesmo já passará a ter acesso ao serviço da farmácia cadastrada.

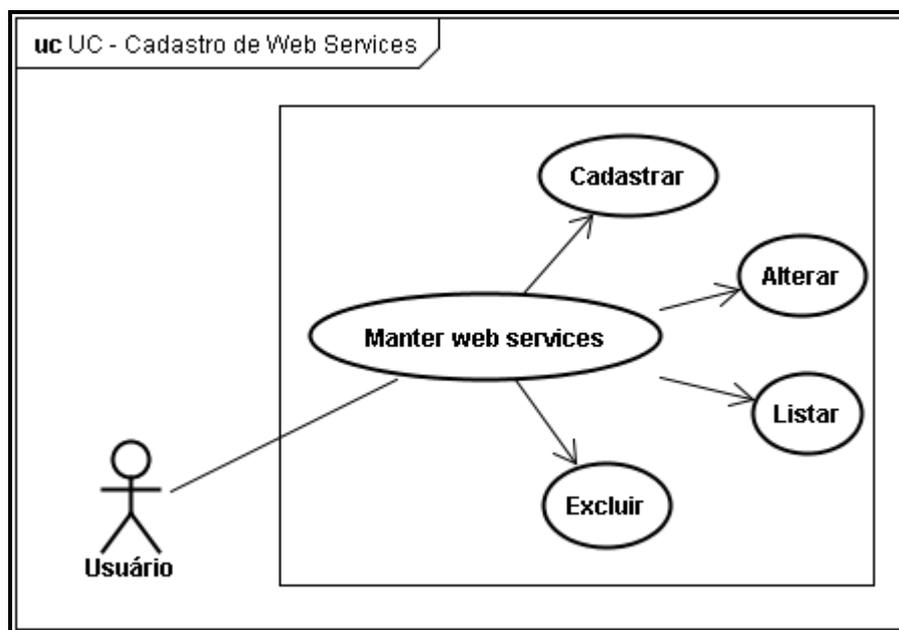
### 3.1.1 Diagrama de caso de uso

O caso de uso do sistema de cadastro de *web services* tem por objetivo demonstrar uma visão geral do sistema. Neste caso de uso, o ator é um usuário que tem permissão de realizar as seguintes interações no sistema:

- Cadastrar farmácia;
- Excluir farmácia;
- Listar farmácia;
- Alterar farmácia.

Na figura 8 é apresentado o caso de uso do sistema de cadastro de farmácias. Aqui vai existir um usuário responsável por realizar a manutenção de farmácias, cidades e estados.

**Figura 8:** Caso de uso sistema de cadastro de *web services*



Fonte: do Autor

A descrição do caso de uso do sistema de cadastro de *web services* procura demonstrar como que funciona a iteração entre os atores e o sistema.

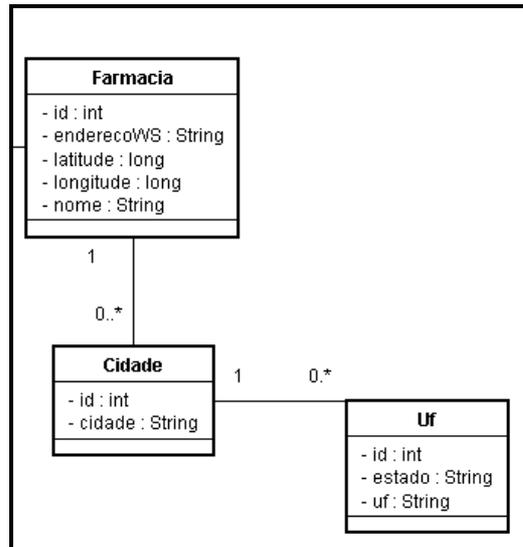
<b>Nome do Caso de Uso</b>	Manter <i>web services</i>
<b>Sumário</b>	Usuário utiliza o sistema para realizar o cadastro de <i>web services</i> .
<b>Ator primário</b>	Usuário responsável por cadastrar os <i>web services</i> .
<b>Ator secundário</b>	Não existe ator secundário.
<b>Precondições</b>	Apenas ter acesso ao sistema.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema apresenta a tela de listagem de <i>web services</i>.</li> <li>2. Usuário seleciona ação cadastrar <i>web services</i>.</li> <li>3. Usuário seleciona ação de alterar <i>web services</i>, selecionando a farmácia no formulário de listagem.</li> <li>4. Usuário seleciona ação de excluir um <i>web services</i>.</li> </ol>
<b>Fluxo Alternativo</b>	<p><b>(2) Cadastrar <i>web services</i></b></p> <ol style="list-style-type: none"> <li>a. Usuário seleciona ação de cadastrar um <i>web services</i>.</li> <li>b. Usuário clica no botão gravar, sistema valida os dados do medicamento conforme RN01, RN02, RN03 RN04 e RN05. Se</li> </ol>

	<p>os dados estiverem corretos o sistema volta para o formulário de listagem e apresenta a MSG01.</p> <p>c. Usuário clica no botão cancelar, sistema apresenta janela questionando se realmente o usuário deseja cancelar a operação. Caso o usuário clicar em sim, o sistema retorna o formulário listagem de <i>web services</i>, caso não, o sistema continua no formulário de cadastro.</p> <p><b>(3) Alterar <i>web services</i></b></p> <p>d. Usuário seleciona o <i>web services</i> a ser alterado na janela de listagem de <i>web services</i>.</p> <p>e. Sistema apresenta os dados do <i>web services</i> no formulário de cadastro.</p> <p>f. Usuário altera os dados referente ao serviço e clica no botão gravar. O sistema valida os dados conforme RN01, RN02, RN03, RN04 e RN05. Se obtiver êxito, o sistema retorna o usuário para lista de <i>web services</i> e apresenta a MSG02, caso contrário o sistema se mantém no formulário de cadastro.</p> <p>e. Usuário seleciona opção cancelar, sistema apresenta janela para o usuário selecionar se realmente deseja cancelar a operação, se sim o sistema volta para o formulário de listagem de <i>web services</i>, senão o sistema mantém o usuário no formulário de edição de <i>web services</i>.</p> <p><b>(4) Excluir <i>web services</i></b></p> <p>c. Usuário seleciona o <i>web services</i> no formulário de listagem de <i>web services</i>, clicando na coluna excluir. Sistema apresenta janela questionando se realmente o usuário deseja excluir o serviço. Se o usuário clicar em sim, o sistema realiza a exclusão do medicamento e apresenta a MSG03, atualiza a lista e continua no formulário de listagem.</p> <p>d. Usuário clica no botão cancelar. Sistema retorna para o formulário de listagem e não exclui o serviço.</p>
<b>Pós-condições</b>	A farmácia se encontrará cadastrada ou excluída da base de dados.
<b>[RN] Regras de negócio</b>	RN01 – O nome da farmácia deve ser informado, caso não informado o sistema apresenta a mensagem MSG04.

	<p>RN02 – A cidade da farmácia deve ser selecionada, caso não informado o sistema apresenta a MSG05.</p> <p>RN03 – A latitude deve ser informada, caso não informada o sistema apresenta a MSG06.</p> <p>RN04 – A longitude da farmácia deve ser informada, caso ao informada o sistema apresenta a MSG07.</p> <p>RN05 – O endereço do <i>web services</i> da farmácia deve ser preenchido, caso não tenha sido preenchido o sistema apresenta a MSG08..</p>
<b>Requisitos funcionais</b>	<b>não</b> Não definidos.
<b>[MSG] Mensagens</b>	<p>MSG01 – <i>Web Services</i> cadastrado com sucesso.</p> <p>MSG02 – <i>Web Services</i> alterado com sucesso.</p> <p>MSG03 – <i>Web Services</i> excluído com sucesso.</p> <p>MSG04 – O nome da farmácia deve ser preenchido.</p> <p>MSG05 – A cidade da farmácia deve ser preenchida.</p> <p>MSG06 – A latitude deve ser preenchida.</p> <p>MSG07 – A longitude deve ser preenchida.</p> <p>MSG08 – O endereço do <i>web services</i> deve ser preenchido.</p>
<b>Autor</b>	do Autor
<b>Data</b>	10/07/2014

### 3.1.2 Diagrama de classes do Sistema de Cadastro de *Web Services*

O diagrama de classes do sistema de cadastro de web service tem como objetivo representar a forma como ocorre o cadastro de web service. Na figura 9, pode ser visualizado como isto ocorre.

**Figura 9:** Diagrama de classes do sistema de cadastro de *web services***Fonte:** do Autor

Na figura 10, pode ser visualizado o formulário que lista as farmácias cadastradas, e na figura 11 é apresentado o formulário de cadastro de farmácias.

**Figura 10:** Formulário que lista os *web services*

ID	Nome	Endereço WS	Latitude	Longitude	Cidade	UF	Alterar	Excluir
1	Farmacia 2	http://10.0.2.2:8080/WSF_1/medicamento/findAll?descricao=-	-28.263475	-52.416175	Passo Fundo	RS	<a href="#">Alterar</a>	<a href="#">Excluir</a>
24	Farmacia 1	http://10.0.2.2:8080/WSF_2/medicamento/findAll?descricao=-	-28.26567	-52.417687	Passo Fundo	RS	<a href="#">Alterar</a>	<a href="#">Excluir</a>

Lista de Web Services

Listando de 1 até 2 de 2 registros

**Fonte1:** do Autor**Figura 11:** Formulário de cadastro de *web services*

Edição de Web Service	
ID	<input type="text"/>
Nome	<input type="text"/>
Cidade:	<input type="text"/>
Latitude	<input type="text" value="0.0"/>
Longitude	<input type="text" value="0.0"/>
Endereço WS	<input type="text"/>
<input type="button" value="Cancelar"/> <input type="button" value="Gravar"/>	

**Fonte:** do Autor

### 3.3 SERVIÇO WEB DAS FARMÁCIAS

Os *web services* das farmácias são mantidos pelas próprias farmácias, cada farmácia poderá desenvolver o seu, porém existe um padrão necessário para a publicação do serviço.

Este padrão inclui que serviço irá receber a descrição do medicamento e retornar uma lista de medicamentos referente à descrição pesquisada para o aplicativo que está consumindo este serviço.

Procedimentos necessários para publicar o serviço:

- Realizar o cadastro do *web services*, este cadastro será solicitado via e-mail e após autenticação dos dados pelo administrador, o serviço se encontrará disponível.
- O serviço deve ser publicado no padrão JSON, sendo assim a farmácia fica livre para escolher a forma como irá programar seu serviço, porém tem de fornecer os dados no padrão especificado.
- O *web services* cadastrado deverá aceitar um parâmetro de entrada, este parâmetro será a descrição do medicamento que está sendo pesquisado pelo usuário.
- Os dados devem ser fornecidos em forma de uma lista de objetos.
- A lista de retorno dos serviços das farmácias deve conter objetos no seguinte formato:
  - id: código de barras do medicamento;
  - desc: descrição do medicamento, ex.: Dipirona, Paracetamol;
  - con: concentração do medicamento, ex.: 500 mg, 250mg;
  - forma: forma farmacêutica do medicamento, ex.: Injetável, cápsulas;
  - cont: conteúdo do medicamento, ex.: Cx c 10, Cx c 20;
  - disp: disponível ou indisponível, padrão é 1 para disponível e 0 para indisponível;
  - valor: valor do medicamento, ex.: R\$ 10,00;

### 3.4 WEB SERVICES CENTRAL

O aplicativo desenvolvido em Android irá acessar diretamente o *web services* Central. Este *web services* irá retornar para o aplicativo uma lista de objetos contendo todos os *web services* da região que se encontra o dispositivo.

No desenvolvimento deste *web services* foram utilizados alguns recursos, como: Apache Tomcat, Java, Hibernate, GSON e Jersey.

O Apache Tomcat é container onde está publicado o serviço e Java é linguagem de programação utilizada para o desenvolvimento da aplicação.

O Hibernate foi utilizado para fazer a comunicação com a base de dados de forma *Object-Relational Model* (ORM).

Jersey teve seu uso aplicado no desenvolvimento da arquitetura do serviço e a API GSON teve seu uso aplicado para realizar a conversão objetos para strings no padrão JSON.

Nas próximas seções serão apresentadas as funcionalidades desse *web services*.

### **3.4.1 Usando *web services* do Google para determinar cidade**

Para realizar este tipo de serviço, fez-se necessário o uso de um *web services* do Google.

Para obter a cidade onde se encontra o dispositivo, fez-se necessário o uso de um serviço da Google. Para o uso desta API é necessário informar a latitude e longitude onde se encontra o dispositivo.

Com uso *web services* do Google foi possível obter a cidade e estado onde se encontra o dispositivo móvel no momento da pesquisa.

O resultado desta API pode ser obtido em formato XML ou JSON, cada formato tem a sua forma de trabalhar, no desenvolvimento do *web services* foi utilizado o formato JSON, por ser um formato que já está sendo trabalhado em todo o projeto.

### **3.4.2 Determinar distância e duração da viagem em modo *driving***

Na montagem da lista de *web services* de farmácias disponíveis para a cidade na qual se encontra o usuário, é necessário determinar a distância e a duração de viagem entre o dispositivo e a farmácia.

Para realização desta função, fez-se novamente necessário o acesso a outra API do Google, esta API tem como objetivo determinar a distância e tempo de viagem do ponto onde se encontra o dispositivo e a farmácia.

Para obter a distância e a duração da viagem é necessário ser passados os parâmetros de latitude e longitude do local de origem, no caso onde se encontra o dispositivo móvel e latitude, longitude onde se encontra a farmácia.

Após é obtido o resultado, que pode ser também selecionado para aparecer no formato JSON ou no formato XML, neste projeto o formato foi JSON. Depois de obtido o resultado em formato JSON, o mesmo deve ser trabalhado de forma a se determinar a distância e duração de viagem. A respeito da viagem, neste sistema foi optado por utilizar o formato de viagem em modo *driving* (locomoção de carro).

### **3.4.3 Enviando resultado para o dispositivo móvel**

O resultado enviado para o dispositivo móvel é uma lista de serviços contendo todas as farmácias disponíveis na cidade onde se encontra o usuário.

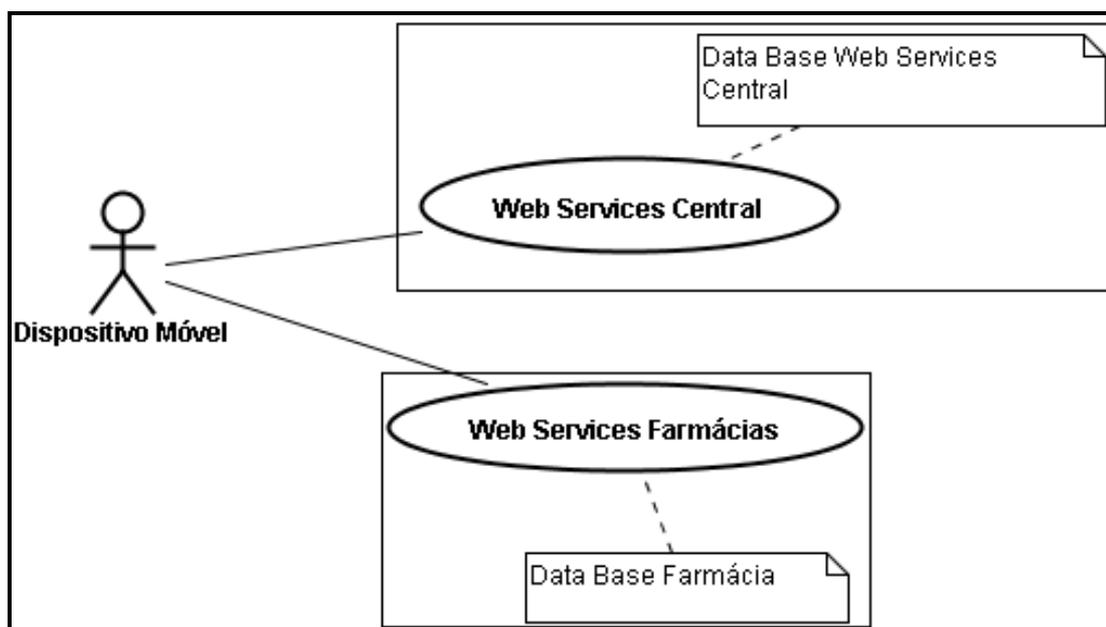
Os dados contidos na lista são descrição da farmácia, localização, distância e tempo entre cada farmácia e o dispositivo.

## **3.5 APLICATIVO ANDROID**

Este aplicativo desenvolvido em Android realiza a pesquisa dos medicamentos entre as farmácias da cidade onde se encontra o dispositivo. Para que seja possível realizar a pesquisa, o sistema solicita acesso ao GPS do aparelho para obtenção das coordenadas de latitude e longitude. Através destas coordenadas é possível obter a cidade onde se encontra o dispositivo, obtendo a cidade, o sistema acessa todos os *web services* disponibilizados pelas farmácias.

Após o usuário informar a descrição do medicamento que deseja pesquisar, o sistema realiza o acesso a todas as farmácias disponíveis na região. Na figura 12 pode ser visualizado o caso de uso do acesso Android ao *Web Services Central* e aos *web services* das farmácias

**Figura 12:** Caso de uso do aplicativo móvel



Fonte: do Autor

### 3.5.1 Funcionalidades realizadas pelo aplicativo

Este aplicativo realiza diversas funcionalidades, entre elas podem ser citadas: acessar *web services* central, acessar GPS do aplicativo, consumindo *web services* das farmácias.

Quando o usuário acessa o aplicativo, a primeira tela a ser apresentada é a tela de pesquisa de medicamentos, esta tela pode ser visualizada na figura 12. Nesta tela, o usuário irá informar a descrição do medicamento desejado, após o aplicativo irá realizar a pesquisa, caso o GPS do aplicativo não esteja ligado, o aplicativo irá mostrar uma informação com a seguinte mensagem: “GPS deve ser ligado”.

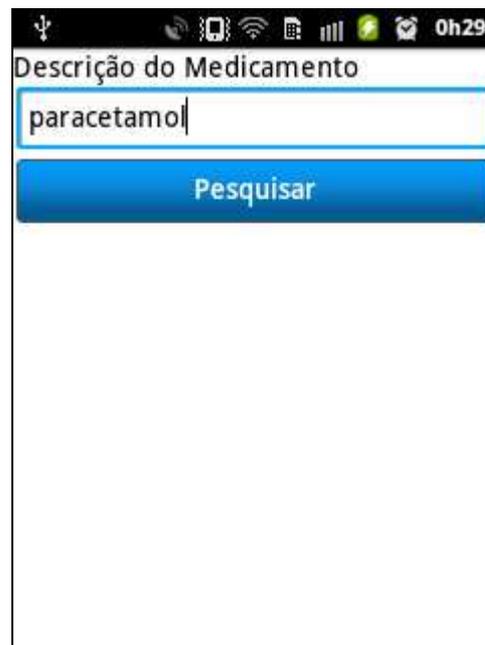
Se o usuário já está com o GPS ligado, o aplicativo irá obter as coordenadas (latitude e longitude) do aparelho. Com as coordenadas obtidas, o aplicativo irá realizar o acesso ao *web services* central passando como parâmetros a latitude e longitude do dispositivo. Com as coordenadas, o *web services* irá realizar acesso a API do Google para obter a cidade.

Com a cidade obtida, o *web services* irá realizar a pesquisa das farmácias/*web services* disponíveis para o local do dispositivo. Obtendo a lista de farmácias disponíveis, o *web services* central vai realizar novamente uma consulta a uma API do Google para obter o tempo e a distância até a farmácia. Após o *web services* irá enviar uma lista de farmácias, onde cada uma contém sua descrição, seu *web services* e suas coordenadas.

### 3.5.2 Apresentação do sistema

No desenvolvimento do aplicativo foram criadas algumas telas. Cada tela tem como finalidade fornecer um *designer* simples para o usuário, e o aproveitamento do espaço fornecido pela tela do dispositivo. Na figura 13, é apresentado o formulário de pesquisa de medicamentos.

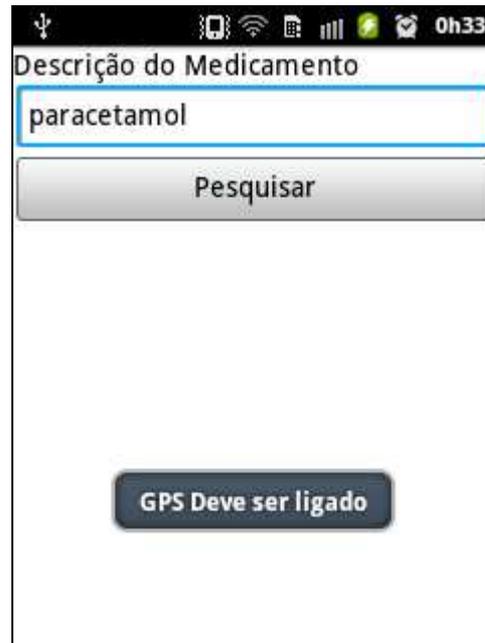
**Figura 13:** Formulário de pesquisa de medicamentos



**Fonte:** do Autor

Na figura 14 é apresentado o formulário de pesquisa de medicamentos com a mensagem “GPS deve ser ligado”, esta mensagem é apresentada quando o aplicativo não consegue obter as coordenadas para realizar a pesquisa. Após é apresentado, na figura 15, a tela de lista de medicamentos ordenada pela distância entre o dispositivo e o medicamento. Já na figura 16 é apresentada a tela com a lista de medicamentos ordenada pelo valor do medicamento.

**Figura 14:** Tela com a mensagem "GPS deve ser ligado"



**Fonte:** do Autor

**Figura 15:** Tela com lista de medicamentos ordenada pela distância



**Fonte:** do Autor

**Figura 16:** Tela com lista de medicamentos ordenado por valor



**Fonte:** do Autor

Na figura 17 é apresentada a tela onde o usuário pode selecionar o tipo de ordenação a ser realizada na lista de medicamentos. E na figura 18 é apresentada a tela com informações a respeito do medicamento selecionado na lista.

As informações apresentadas têm por objetivo informar o nome da farmácia, descrição do medicamento, valor, se está disponível e informações referente à localização da farmácia, nas informações de localização constam os seguintes dados: rua, CEP, bairro, cidade, estado e país.

**Figura 17:** Tela de onde é selecionado o tipo de ordenação da lista de medicamentos



Fonte: do Autor

**Figura 18:** Tela com informações do medicamento selecionado na lista



Fonte: do Autor

## 4 CONSIDERAÇÕES FINAIS

Com o amplo dinamismo das tecnologias e a crescente utilização pela sociedade, torna-se necessário o desenvolvimento de projetos/aplicativos voltados para dispositivos móveis. Mas para o correto desenvolvimento se faz necessário buscar o conhecimento das tecnologias disponíveis.

Pensando nisso, este trabalho fez o estudo de algumas tecnologias disponíveis para o desenvolvimento, estas tecnologias não se restringem somente a questões de dispositivos, mas também em relação a linguagens e bibliotecas presente no mercado atual.

O estudo aqui desenvolvido teve como principal objetivo o desenvolvimento de um aplicativo para consultar preço de medicamentos entre as farmácias. No decorrer do desenvolvimento procurou-se estudar tecnologias disponíveis no mercado e como funciona o cadastro de medicamentos, procurando assim uma representação de como estes dados são armazenados.

Também foi possível ampliar o conhecimento tecnológico, identificar a importância da pesquisa e do aprendizado adquirido sobre o desenvolvimento para aplicativos móveis e o desenvolvimento de *web services*. Outro ponto que também pode ser colocado como importante foi o conhecimento no negócio, com relação ao cadastro de medicamentos e como funcionam as especificações determinadas pela Anvisa para o controle de medicamentos.

O conhecimento adquirido proporcionou que fosse possível identificar tecnologias que ajudaram muito para conclusão do projeto, como *JSON*, *GSON*, *Jersey*, *Hibernate* e *Apache Tomcat*. Estas tecnologias foram muito importantes para o desenvolvimento, agilizando as tarefas que teriam um tempo de desenvolvimento maior se estes recursos não estivessem disponíveis. Sua escolha se deve ao fato de diminuir o custo do projeto, pois todas as tecnologias aqui utilizadas são *open source* e sem custo de licença para o desenvolvimento.

Com o desenvolvimento do aplicativo foi possível obter alguns resultados como: desenvolver para Android se torna complexo, uma vez que a forma de desenvolvimento muda de versão para versão, o lado bom é que está sempre atualizado e com documentação, o lado ruim é que para cada versão de aplicativo terá de ser desenvolvido de uma forma.

Após o projeto ter sido concluído, foram realizados alguns testes, como a facilidade de navegação entre as janelas do aplicativo, utilizando recursos disponíveis no aparelho como botão voltar do dispositivo, dessa forma aproveitando ao máximo espaço das telas do smartphones. O sistema não se encontra em produção, pois ainda necessitará de alterações futuras para sua utilização no mercado.

Por fim, esse projeto somente foi possível de ser realizado através do conhecimento adquirido no curso Superior de Tecnologia em Sistemas para Internet. Com a finalização deste projeto, espera-se que o mesmo sirva de base para futuros projetos a serem desenvolvidos.

## 5 REFERÊNCIAS

ANDROID confirma liderança dos *smartphones*. *Jornal Zero Hora*. Publicado em 16/05/13. Disponível em:

<<http://zerohora.clicrbs.com.br/rs/economia/tecnologia/noticia/2013/05/android-confirma-lideranca-dos-smartphones-4139776.html>>. Acesso em: 16 mai. 2013.

ANDROID.LOCATION. *Google Developers References*. Disponível em:

<<http://developer.android.com/reference/android/location/package-summary.html>>. Acesso em: 24 jun. 2013.

ANVISA. Diferenças entre embalagem de genérico e similar, publicado em 10/02/2000.

Disponível em: <<http://www.anvisa.gov.br/divulga/noticias/100300.htm>>. Acesso em: 01 jun 2013.

\_\_\_\_\_. Medicamento Similar. Disponível em: <<http://s.anvisa.gov.br/wps/s/r/ft>> Acesso em: 18 jun. 2013.

APACHE Tomcat. Disponível em: <<http://tomcat.apache.org/index.html>>. Acesso em: 02 jun. 2014.

APOSTILA *Java e Orientação a Objetos*, Capítulo 2. Disponível em:

<<http://www.caelum.com.br/apostila-java-orientacao-objetos/o-que-e-java/>>. Acesso em: 18 jun. 2013.

BOLTON, Matt. iOS e Android são colocados frente a frente: acompanhe comparativo dos sistemas móveis. Publicado em 19/06/2012. Disponível em:

<<http://tecnologia.uol.com.br/noticias/redacao/2012/06/19/ios-e-android-sao-colocados-frente-a-frente-acompanhe-comparativo-entre-os-sistemas.htm>>. Acesso em: 13 jun. 2013.

CONSTANTINO FILHO, Ademir. Entendendo o XML. Publicado em 30/09/2002.

Disponível em: <<http://www.guj.com.br/articles/19>>. Acesso em: 10 jun. 2013.

COSTA, Elisângela Rocha da. *Bancos de Dados Relacionais*. 2011. Trabalho de conclusão de Curso (Tecnólogo em Processamento de Dados). – Faculdade de Tecnologia de São Paulo, São Paulo, 2011. Disponível em: <<http://www.fatecsp.br/dti/tcc/tcc0025.pdf>>. Acesso em: 08 mai. 2013.

DISPONIBILIDADE de recursos e serviços.

Disponível em: <<http://www.windowsphone.com/pt-br/how-to/wp8/basics/feature-and-service-availability>>. Acesso em: 10 mar. 2014.

DRABACK, Renan. Android – Geolocalização. Publicado em: 20/09/2012. Disponível em:

<<http://www.mobiltec.com.br/blog/index.php/tutorial-android-geolocalizacao/>>. Acesso em: 10 jun. 2013.

FRIAS, Maria Cristina. Setor farmacêutico espera crescer 14% neste ano. *Folha de S. Paulo*.

19/12/2012. Disponível em: <<http://www1.folha.uol.com.br/colunas/mercadoaberto/1203444-setor-farmaceutico-espera-crescer-14-neste-ano.shtml>>. Acesso em: 11 jun. 2013.

JARDIM, Carolina. Preços nas farmácias podem variar quase 70 % no mesmo bairro no Rio. *Jornal O Globo*. Publicado em 09/11/12. Disponível em: <<http://oglobo.globo.com/economia/precos-nas-farmacias-podem-variatar-quase-70-no-mesmo-bairro-no-rio-6679734>>. Acesso em: 23 mai. 2013.

KOLAKOWSKI, Nicholas. Microsoft Explains Windows Phone 7 Lack of Compatibility. Publicado em 15/03/2010. Disponível em: <<http://www.eweek.com/c/a/Mobile-and-Wireless/Microsoft-Explains-Windows-Phone-7-Lack-of-Compatibility-588900/>>. Acesso em: 01 jun. 2013.

KING, Gavin.; BAUER, Christian.; ANDERSEN , Max Rydahl.; BERNARD, Emmanuel.; e EBERSOLE, Steve. HIBERNATE – Persistência Relacional para Java Idiomático (Documentação de Referência Hibernate 3.5.6-Final). Disponível em: <[https://docs.jboss.org/hibernate/core/3.5/reference/pt-BR/pdf/hibernate\\_reference.pdf](https://docs.jboss.org/hibernate/core/3.5/reference/pt-BR/pdf/hibernate_reference.pdf)>. Acesso em: 02 jun. 2014.

LIMA, Júnio César de.; CARVALHO, Cedric Luiz de. *Extensible Markup Language (XML)*. Relatório Técnico. Instituto de Informática. Universidade Federal de Goiás. Goiás, Junho/2005. Disponível em: <[http://www.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF\\_002-05.pdf](http://www.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_002-05.pdf)>. Acesso em: 04 jun. 2013.

MACEDO, Alexandre. *WebService Restfull* utilizando JSON. Disponível <<http://www.k19.com.br/artigos/web-service-restful-utilizando-json/>>. Acesso em: 16 mai. 2013.

\_\_\_\_\_, Alexandre. Criando um Webservice Restful em Java. Publicado 12 nov. 2010. Disponível em: <<http://www.k19.com.br/artigos/criando-um-webservice-restful-em-java/>>. Acesso em: 03 jun. 2014.

MEDEIROS, Wesley Nepomuceno de. Consumindo *WebService* em aplicações *Android*. Disponível em: <<http://www.devmedia.com.br/consumindo-webservices-em-aplicacoes-android/26866>>. Acesso em: 14 mai. 2013.

\_\_\_\_\_. Consumindo JSON em Aplicações Android. Disponível em <<http://www.devmedia.com.br/consumindo-json-em-aplicacoes-android/27589>>. Acesso em: 15 mai. 2013.

MEDEIROS, Higor. Introduzindo o servidor de aplicação Apache Tomcat <<http://www.devmedia.com.br/introduzindo-o-servidor-de-aplicacao-apache-tomcat/27939>>. Acesso em: 02 jun. 2014.

MORANDI, Paulo Sérgio; RODOLPHO, Luiz Roberto Lethang. Desenvolvimento Mobile nas Plataformas Android e iOS. Disponível em: <<http://www.mobiltec.com.br/blog/index.php/desenvolvimento-mobile-nas-plataformas-android-e-ios/>>. Acesso em: 15 jun. 2013.

PARSING JSON using GSON in Android. Publicado em: 05/06/2010. Disponível em: <<http://primalpop.wordpress.com/2010/06/05/parsing-json-using-gson-in-android/>>. Acesso em: 10 jun. 2013.

RIELAU, Serge. DB2 9.7: Execute aplicativos Oracle em DB2 9.7 para Linux, UNIX e Windows. Publicado em: 08/05/2013. Disponível em: <<http://www.ibm.com/developerworks/data/library/techarticle/dm-0907oracleappsondb2/index.html>>. Acesso em: 10 mai. 2013.

RONDON, Thiago. Arquitetura REST e o serviço web '*RESTful*'. <<http://sao-paulo.pm.org/artigo/2010/RESTful>>. Acesso em: 02 jun. 2014.

TAKAI, Osvaldo Kotaro.;ITALIANO, Isabel Cristina.;FERREIRA, João Eduardo. Introdução a banco de dados. Disponível em: <<http://www.ime.usp.br/~jef/apostila.pdf>>. Acesso em: 10 mar. 2014.

WILLRICH, Roberto. Linguagens de Programação. 4.1 Programação de Computadores. Disponível em: <[http://algoritmo.dcc.ufla.br/~monserrat/icc/Introducao\\_linguagens.pdf](http://algoritmo.dcc.ufla.br/~monserrat/icc/Introducao_linguagens.pdf)>. Acesso em: 05 mai. 2014.

WINDOWS Phone 8.1 é lançado para desenvolvedores, mas você já pode baixá-lo. Publicado em 14/04/2014. Disponível em: <<http://canaltech.com.br/noticia/windows-phone/Windows-Phone-81-e-lancado-para-desenvolvedores-mas-voce-ja-pode-baixa-lo/>>. Acesso em: 01 jun. 2013.