UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
FACULDADE DE CIÊNCIAS EXATAS E NATURAIS
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JORGE CHRYSTIANN GUIMARÃES DA CUNHA NUNES

CONSFARM: Sistema de Consulta à Disponibilidade de Produtos Farmacêuticos

SANTA CRUZ
2017

JORGE CHRYSTIANN GUIMARÃES DA CUNHA NUNES

CONSFARM: Sistema de Consulta à Disponibilidade de Produtos Farmacêuticos

Trabalho de Conclusão de Curso submetido à Universidade do Estado do Rio Grande do Norte como requisito para obtenção do grau de Bacharel em Ciência da Computação.

ORIENTADOR – Prof. Dr. Felipe Denis Mendonça de Oliveira.
CO-ORIENTADOR - Prof. Dr. Carlos Alberto de Albuquerque Silva.

SANTA CRUZ
2017

JORGE CHRYSTIANN GUIMARÃES DA CUNHA NUNES

CONSFARM: Sistema de Consulta à Disponibilidade de Produtos Farmacêuticos

Trabalho de Conclusão de Curso submetido à Universidade do Estado do Rio Grande do Norte como requisito para obtenção do grau de Bacharel em Ciência da Computação

Aprovado em: _____/_____/_____.

BANCA EXAMINADORA

_____

Prof. Dr. Felipe Denis Mendonça de Oliveira – Orientador
Universidade do Estado do Rio Grande do Norte

_____

Prof. Dr. Carlos Alberto de Albuquerque Silva - Co-Orientador
Universidade do Estado do Rio Grande do Norte

_____

Profª. MSc. Lisa Cristina Silva de França Oliveira
Universidade do Estado do Rio Grande do Norte

_____

Prof. Noély Gomes de Araújo e Mello
Universidade do Estado do Rio Grande do Norte

Dedico esse trabalho aos meus Avós,
aos meus pais, e a minha noiva Husllanya Lany.

AGRADECIMENTOS

Agradecer primeiramente a Deus, por proteger todos os meus passos nessa trajetória.

Agradecer aos professores por toda dedicação e ensinamento me passados durante todo o curso, principalmente a Carlos Albuquerque e Felipe Denis, que me ajudaram para que uma ideia se torne realidade.

Agradecer aos meus Familiares em especial aqueles que estiveram ao meu lado para me dar força e determinação para continuar a cada dia, não esquecendo dos meus Avós, mesmo não estando presentes fisicamente, sei que estavam ao meu lado torcendo por mim.

Agradecer a minha Noiva, Husllanya Lany, por todo amor, carinho, paciência e conselhos para vencer os desafios, sempre com pensamento positivo me levando a nem cogitar a possibilidade de desistir.

Agradecer ao amigo, Erivaldo Pinheiro, que mesmo não sendo pai, prestou o papel de forma importante para ensinar muitos caminhos para que eu me torne uma pessoa melhor.

Por fim agradecer a todos os amigos que colaboraram de forma direta e indireta para que este trabalho se tornasse realidade, dedicando um pouco de seu tempo para ajudar.

Dificuldades preparam pessoas comuns para destinos extraordinários.

(C.S Lewis)

RESUMO

A tecnologia de Informação vem, a cada dia, firmando mais espaço no cotidiano das pessoas e das empresas. Isto se dá em virtude do avanço tecnológico, consequência do barateamento dos *hardwares*, dos *softwares* e dos avanços em pesquisas. Um exemplo destes avanços tecnológicos que se pode citar é a popularização da telefonia móvel por meio das tecnologias envolvidas, por exemplo: IPV6. O outro exemplo de tecnologia que revolucionou o mercado de telefonia foi o desenvolvimento dos *smartphones*, os quais fazem uso de aplicativos que possibilitam a utilização destes aparelhos. Mas o que mais marcou a área de telefonia foi a possibilidade de acessar a internet por meio destes aparelhos. Entretanto, a comunicação entre alguns sistemas, sejam eles voltados para *desktop* ou para *WEB* com os aplicativos dos *smartphones* só é possível por meio de um serviço disponibilizado pelo servidor de aplicação. A esse serviço disponibilizado no servidor dá-se o nome de *webservice,* o qual permitirá a comunicação do cliente móvel com a estação servidora. Podendo ter várias aplicações envolvendo *webservices*. Tendo isso em vista, o presente trabalho desenvolveu um aplicativo para os *smartphones* que usam o sistema *Android* que permita a comunicação com as farmácias por meio de um *webservice.*

Palavras Chave: Engenharia de *Software*. *Mobile. WebService*. Tecnologia da Informação.

ABSTRACT

The Information technology comes, every day, establishing more space in the daily life of people and companies. This is due to the technological advance, resulting from the cheapening of hardware, software and research advances. An example of such technological advances is the popularization of mobile telephony through the technologies involved, for example: IPV6. The other example of technology that revolutionized the telephony market was the development of smartphones, which make use of applications that enable the use of these handsets. But what most marked the area of telephony was the possibility of accessing the internet through these devices. However, communication between some systems, whether they are desktop or WEB-oriented with smartphone applications, is only possible through a service provided by the application server. This service provided on the server is called webservice, which will allow the mobile client to communicate with the server station. It may have several applications involving webservices. With this in mind, this work has developed an application for smartphones that use the Android system that allows communication with pharmacies through a webservice.

Keywords: Software Engineering. Mobile. WebService. Information Technology.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

RUP	*Rational unified process* - Processo Racional Unificado
XP	*Extreme programming* - Programação extrema
SOAP	*Simple Object Acess Protocol* - Protocolo de Acessos de Objeto Simples
UML	*Unified Modeling Language*
REST	*Representational State Transfer* - Transferência Estatal Representativa
W3C	*World Wide Web* - Rede mundial de computadores
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
XML	*eXtensible Markup Language* - Linguagem de marcação extensível
MIMO	*Multiple-Input Multiple-Output*- Saída múltipla de múltiplas entradas
XSD	XML Schema Definition - Definição de Esquema XML

SUMÁRIO

1. INTRODUÇÃO

A tecnologia de informação veio para melhorar a vida e trazer novas soluções para a correria do dia-a-dia. Nesse sentido a todo momento aparecem novidades que ajudam a todos a diminuir as dificuldades.

O caso do celular não foi diferente, começando apenas como um meio de comunicação, hoje podemos fazer quase tudo na palma da mão, seja o uso de um despertador ou mesmo transações bancárias. E por que não pensar nessa tecnologia para nos auxiliar naqueles dias em que a saúde pede uma parada na correria para buscar seu medicamento de forma rápida e segura.

Foi baseado nessa ideia, de uma aplicação para resolver essa questão da comunicação rápida entre os clientes e as farmácias que se desenvolveu o presente trabalho

1.1 PROBLEMÁTICA E JUSTIFICATIVA

Um dos problemas são as farmácias não é obrigada a serem credenciadas para venda de medicamento psicotrópicos de controle especial. Segundo a Anvisa (2017) é necessário tomar algumas medidas antes de começar a vender os psicotrópicos, como cita a resolução (rdc nº. 27, de 30 de março de 2007), desde já, esta resolução criada pela Anvisa conclui que as farmácias não são obrigadas a este tipo de venda. Por não serem obrigadas a venderem medicamentos psicotrópicos, constata-se que não há uma comunicação entre os profissionais da área de saúde com as farmácias locais, que atuam nos municípios. Eles não têm o conhecimento se o município tem farmácias autorizadas a esta venda. Sendo um problema para a população ter que comprar em outro município a medicação.

Também um dos problemas que pode vir a acontecer é os hospitais não dispuserem de certos tipos de medicamentos ou até mesmo por algum problema, deixarem de oferecer aos pacientes os medicamentos básicos. Como exemplo, pode-se citar a reportagem do RNTV (2016), que mostrou os problemas enfrentados por um hospital, levando a acreditar que os familiares dos pacientes tiveram que comprar estes medicamentos para evitar a morte dos mesmos.

Diante disso, surgiu a ideia de fazer um *software* de ligação entre as pessoas e as farmácias, sejam elas médicos ou pacientes. Por meio dele as farmácias poderão fornecer uma listagem de medicamentos disponíveis. Com isso, todas as pessoas que possuam um PC ou um celular com Sistema Operacional (SO) *Android (Android*, 2017) conectados à internet poderão acessar uma lista dos medicamentos existentes nas farmácias por meio um sistema *WEB* que possibilitará a comunicação do banco de dados das farmácias com os *hardwares* dos clientes.

1.2 OBJETIVO

O objetivo principal é desenvolver uma aplicação *Android*, que se comunique com as farmácias por meio de um *WebService*, o qual disponibilizará os serviços a partir do protocolo REST *(Representational State Transfer* - Transferência Estatal Representativa), para toda a população que dispõe de um dispositivo *Android* e *Internet*, fazendo busca em tempo real, tendo assim maior disponibilidade sobre a real situação do estoque farmacêutico.

Na primeira seção foi apresentada toda a proposta do aplicativo, incluindo qual foi a motivação e justificativa para esta ideia, incluindo ao final o objetivo para ser alcançado durante esse desenvolvimento.

Já na segunda seção, será mostrado toda a teoria existente para esse projeto, buscando mostrar a tecnologia *mobile*, junto com os conceitos de engenharia de *software* e *webservice* e, ainda, trabalhos relacionados com esses temas.

Na terceira seção serão apresentados os requisitos funcionais e não funcionais da aplicação, haverá também informações sobre *WebService*, aplicação *mobile* e o desenvolvimento da aplicação.

Na quarta seção serão mostrados todos os resultados conquistados pelo que foi mencionado nas sessões anteriores, apresentando os resultados e discussões abordadas.

Na quinta seção, para finalizar o trabalho, será apresentada a conclusão.

2. REFERENCIAL TEÓRICO

No referencial teórico será mostrada toda a base de conhecimento usada para o desenvolvimento do projeto, contendo assuntos relacionados também para criação e aprimoramento do aplicativo.

2.1. ENGENHARIA DE *SOFTWARE*

Falando do termo engenharia e *software* separados tem-se que:

Engenharia: É a arte de aplicar conhecimentos científicos e certas habilitações especificas a criação de estruturas, dispositivos e processos que se utilizam para converter recursos naturais em formas adequadas ao atendimento das necessidades humanas (Aurelio,2016).

Software: Conjunto de componentes lógicos de um computador ou sistema de processamento de dados; programa, rotina ou conjunto de instruções que controlam o funcionamento de um computador, suporte lógico. (Aurelio,2016).

Tendo essas visões separadas dos dois termos, pode-se tirar as conclusões sobre o que seria engenharia de *software*:

Para Sommerville (2000), a engenharia de *software* é uma matéria que se dedica a construção de *software,* onde esses engenheiros teriam que desenvolver de forma organizada e utilizar técnicas e ferramentas apropriadas.

Já para Fritz Bauer (1968) em uma conferência em 1968, a engenharia de *Software* é: “A criação de sólidos princípios de engenharia a fim de obter *software* de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais”.

2.1.1. Metodologias Ágeis:

As Metodologias Ágeis foram criadas em 2001, quando vários especialistas em desenvolvimento de *software,* representaram os métodos *Scrum* e *Extreme programming*, onde ficaram determinados alguns princípios e conceitos desses métodos. Com isso foi criada uma fundação chamada “manifesto Ágil”.

Segundo Fonseca (2015), os principais conceitos do manifesto ágil são:

- Pessoas e interações, ao contrário de processos e ferramentas.
- *Software* executável, ao contrário de documentação extensa e confusa.
- Colaboração do cliente, ao contrário de constantes negociações de contratos.
- Respostas rápidas para as mudanças, ao contrário de seguir planos previamente definidos.

2.1.2. Diferença entre RUP, XP e Iconix:

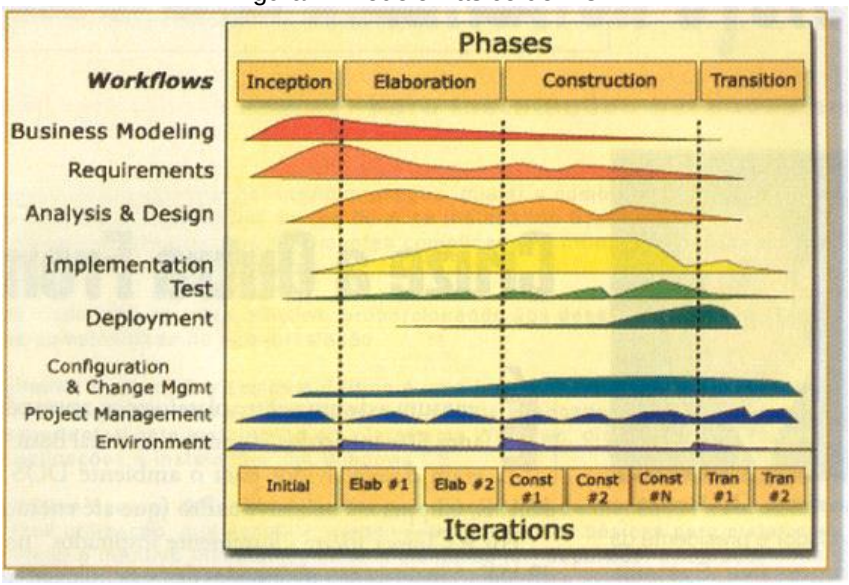
2.1.2.1. *Rational unified process* - Processo Racional Unificado (RUP):

O RUP, além de uma metodologia, é comercializado por uma empresa privada, nada mais é que uma grande documentação desenvolvida em hipertexto. Diante disso pode-se citar alguns conteúdos, caso fosse utilizada esta metodologia. Em Destaque: *WorkFlows*, onde cada um é descrito detalhadamente. Além desse conteúdo, tem-se também as tarefas, os modelos de equipe e os modelos de equipamentos, tornando, portanto, como citado anteriormente a metodologia RUP uma metodologia muito burocrática.

Segundo Viana (2016), a Figura 1(página 17) apresenta os elementos básicos do RUP. Nesta metodologia, o projeto passa por 4 fases básicas. Estas fases são:

- *Inception* - entendimento da necessidade e visão do projeto;
- *Elaboration* - especificação e abordagem dos pontos de maior risco;
- *Construction* - desenvolvimento principal do sistema;
- *Transition* - ajustes, implantação e transferência de propriedade do sistema.

Figura 1: Modelo Básico do RUP



Fonte: (VIANA, 2016).

2.1.2.2. *Extreme programming* - Programação extrema (XP):

O XP já passa a ser uma Metodologia Ágil, muitas vezes usada por pequenas e médias empresas para o desenvolvimento de *softwares* que se modificam rapidamente. Um dos principais objetivos do XP é dá agilidade e satisfação ao cliente, deixando muitas vezes a desejar na Analise e Projeto, tendo 4 princípios básicos: Princípio de Comunicação, princípio da Simplicidade, Princípio do *FeedBack* e o princípio da coragem.

2.1.2.3. Iconix:

O ICONIX é considerado uma metodologia simples, prática, mas que é considerada forte devido não ser tão burocrática como a RUP, sendo ágil como a XP, tendo então o meio termo entre as outras duas metodologias. O Iconix é composto por cinco fases, que são: Modelo de Domínio, modelo de Caso de Uso, Analise Robusta, Diagrama de Sequência e Diagrama de Classe. Esta metodologia apresenta ainda três características fundamentais: interativo e Incremental, rastreabilidade e aerodinâmica da UML.

Além disso, o trabalho publicado por (MEIRELES, BONIFACIO,2015) esclarece que o aparecimento constante de novas tecnologias de *software* tem alterado a forma como as tecnologias de *software* são desenvolvidas. Para acompanhar essa transformação o processo de ensino em Engenharia de *Software* também tem evoluído.

Outro estudo que fala de aplicações móveis é o de Constantinou, Camilleri, e Kapetanaki, (2013). Neste estudo os autores relatam que os atuais *smartphones* estão sendo cada vez mais utilizados, aumentando consideravelmente sua utilização e, com isto, possibilitando o surgimento de novas funcionalidades para o mesmo. O sistema operacional *Android* vem se popularizando e se tornando o principal sistema operacional para *smartphones*.

2.2. WEB SERVICES

Web Services são serviços em uma rede que serve para a troca de informações entre vários dispositivos eletrônicos, podendo enviar e processar os dados de acordo com que é requisitado. Segundo Fidel (2017) “Entre as abordagens existentes para a implementação de *webservices*, os protocolos *SOAP (Simple Object Acess Protocol* – Protocolo de Acesso Simples*) e REST* são as opções de maior destaque nos dias de hoje, estando presentes em grande parte das discussões relacionadas as arquiteturas orientadas a serviços na *web.*”.

Logo, o sítio da *DevMedia,* os *Web Services* estão cada vez mais se tornando popular entre os programadores, pois permitem a troca de informações entre várias plataformas sem nenhum problema, sendo elas *Android* com *Linux,* por exemplo. Com base nisso, pode-se citar que são os dois mais utilizados e neste presente trabalho serão citados os dois, mas no aplicativo será usado o *REST*.

2.2.1. *Simple Object Acess Protocol* (SOAP)

O protocolo *SOAP* serve especificamente para a troca de informações entre dispositivos, ou seja, um tipo de formato de dados para troca de dados entre serviços, permitindo assim a interoperabilidade entre os mesmos.

O protocolo *SOAP* foi adotado pelo *World Wide Web Consortion* (W3C) sendo que a mensagem será levada através do protocolo HTTP, com isso, acaba sendo beneficiado pela infraestrutura criada para o HTTP para passar pelos *firewalls* e roteadores, tornando assim uma fácil comunicação.

Segundo Carlos (2012) as características do SOAP são:

- Definido pelo consórcio W3C.
- Protocolo baseado em XML(*eXtensible Markup Language* - Linguagem de marcação extensível) para troca de informações em ambientes distribuídos
- Padrão de utilização para *Web Services*.
- Normalmente utiliza HTTP como protocolo de transporte.

Segundo Carlos (2012), uma mensagem SOAP contém basicamente os elementos:

- Envelope: É o elemento principal do XML que representa a mensagem.
- *Header:* É um mecanismo genérico que permite a adição de características à mensagem SOAP de forma descentralizada, sem acordo anterior entre as partes comunicantes.
- *Body*: Este elemento contém a informação a ser transportada.

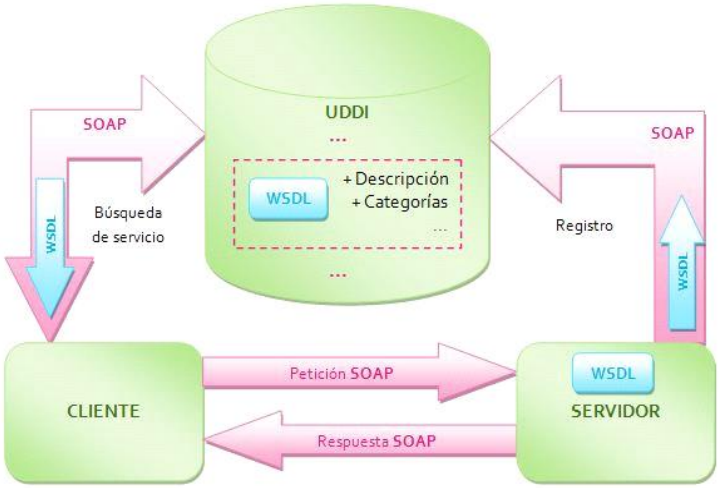
Em se tratando dos elementos da mensagem SOAP, o envelope é uma das partes obrigatórias, funcionando como uma caixa que recebe os outros elementos, precisando conter o cabeçalho para que possa entregar corretamente no local desejado e o corpo da mensagem para que a entrega seja feita no local certo.

Já, o corpo (*Body*) serve para guardar os dados contidos em um método, contendo o nome do método e os parâmetros de entrada e saída, retornando assim o resultado do que foi produzido pelo método, esse elemento deve vir logo após o cabeçalho, sendo que será necessário receber uma resposta ou requisição, se for uma resposta irá trazer os resultados obtidos quando chamou o método.

Segundo Carlos (2012), uma grande vantagem de usar o protocolo SOAP é que o mesmo aceita todos os tipos de dados que contem no *XSD*(XML **Schema** Definition-Definição de Esquema XML) *Schema*.

Como pode-se observar na Figura 2(página 20), existe uma comunicação entre cliente servidor através de uma mensagem SOAP.

Figura 2: Relação SOAP,WSDL e UDDI



Fonte:(Txikiboo, 2013)

2.2.2. *Representational State Transfer* (REST)

O protocolo REST foi apresentado pela primeira vez por Roy Fielding, em sua tese de doutorado, no ano de 2000, mostrando que esse tipo de arquitetura de *Software* para Sistemas Distribuídos, não seria utilizado apenas no desenvolvimento de *Web Services*. Segundo (DAL, FRIENDRICH, TRINDADE, 2009), o termo geralmente é usado para descrever qualquer interface que transmita dados de um domínio especifico sobre HTTP mesmo sem a necessidade de uma outra camada de mensagem como no SOAP.

2.2.2.1. Princípios do estilo REST:

Segundo Fielding (2000), para que esses princípios do estilo sejam respeitados, existe algumas restrições que devem ser seguidas. Abaixo algumas delas:

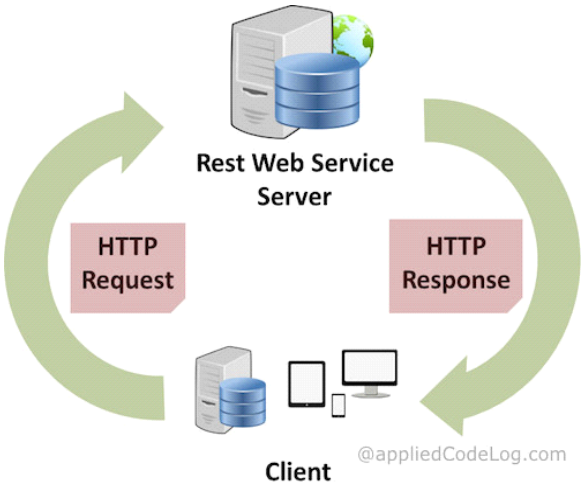
- Cliente-Servidor:

Essa restrição é comumente encontrada em aplicações *web.* Sendo que se tem um Servidor com vários serviços disponíveis e aguarda algumas requisições enviadas pelo cliente. Já o cliente necessita que o servidor esteja disponível para que possa enviar

uma requisição, este servidor poderá aceitar e executar o serviço como também rejeitar e enviar a decisão que foi tomada para o cliente.

O resultado disso é uma separação da estrutura do Usuário da estrutura do servidor, servindo para melhorar o acesso do cliente através de várias plataformas e também conseguir outros benefícios como a escalabilidade, simplificando assim o servidor, como mostrado na figura abaixo, O cliente manda uma requisição para o servidor e o Servidor manda uma resposta para o Cliente.

Figura 3: Troca de informações Cliente-Servidor



Fonte: (stackOverflow, 2017)

- *Cache*:

O *cache* veio com a ideia de diminuir os problemas com a diminuição do desempenho. Fazendo que seja obrigatório na hora de usar o REST, os dados devem conter uma marcação chamada *cacheable* ou *noncacheable*, ou seja passível ou não da utilização da *cache*. Se uma resposta for colocada como *cacheable*, então se for feita a mesma solicitação no futuro, será utilizada a mesma resposta. Com isso, em algumas operações não serão necessárias as interações com o servidor, conseguindo com isso ótimo desempenho e eficiência.

- *Stateless* (Sem estado):

Essa Restrição diz respeito a interação entre cliente e servidor. A comunicação só poderá ser feita se não houver nenhum tipo de estado no servidor, ou seja, todas as

requisições enviadas para o servidor deverão conter todas as informações necessárias para que ela seja entendida. Com isso, tudo que diz respeito a estados de seção deve ser totalmente mantido no cliente.

- *Interface* Uniforme:

A principal característica que diferencia o REST de outros estilos baseados em rede é sua eficiência em uma *interface* uniforme entre os componentes (Clientes e Servidor). Sendo definido 4 requisitos para essa interface uniforme: Manipulação de recursos através das representações; identificação dos recursos; mensagem auto descritivas e hipermídia como mecanismos para o estado da aplicação. Atualmente, o protocolo HTTP é mais indicado para trabalhar com *Web Service* REST, e com isso conseguindo ter uma interface uniforme utilizando os quatro métodos básicos para serem feitas operações comuns, sendo elas:

- *Get* – Utilizada para recuperar uma representação de um recurso;
- *Put* – Utilizada para criar novo recurso ou modificar um existente;
- Delete – Serve para remover um Recurso;
- *POST* – Servindo para criar um novo recurso.

- Multicamada:

Pensando em melhorar o requisito de escalabilidade da Internet, foi introduzido ao estilo REST o atributo de divisão em camadas. Esses sistemas para separar diferentes unidades de funcionalidade utilizam camadas, tendo como desvantagem, em alguns momentos, a latência com os dados processados e uma adição de *overhead*.

- *Code-On-Demand*:

Essa última característica não é tão importante por se caracterizar opcional, ou seja, do que depender do aplicativo, o mesmo poderá ser utilizada ou não. Permitindo, portanto, que os usuários possam baixar e executar processos diretamente no lado do cliente, vendo isso, o REST conseguiu manter extensibilidade.

2.3. TECNOLOGIAS PARA DESENVOLVIMENTO MOBILE:

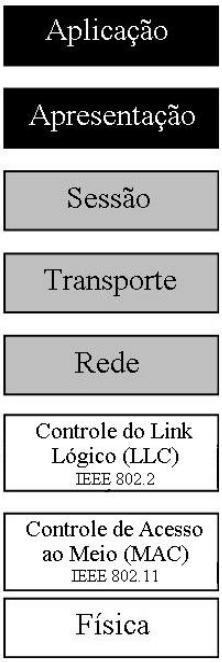
Nesse capitulo serão citadas as tecnologias utilizadas para o desenvolvimento do aplicativo, tendo também as tecnologias que serão usadas no aplicativo.

2.3.1. Wifi

O termo Wi-fi surgiu devido ser uma marca registrada pela *Wi-Fi Alliance*, o qual se tornou um sinônimo para a tecnologia IEEE 802.11, permitindo assim uma conexão entre vários dispositivos que possuem essa tecnologia.

Com o intuito de diminuir a falta de compatibilidade entre a transmissão de dados através de ondas de rádio entre equipamentos de diferentes fabricantes, o que era bastante comum antigamente, o IEEE lançou o padrão 802.11 (Figura 4), define a camada de Controle de Acesso ao Meio para transmissão sem fio (TORRES, 2001).

Figura 4: Modelo de Camadas do padrão 802.11



Fonte: (TORRES, Gabriel, 2001)

As redes *Wifi* ganharam o mercado, substituindo e, até mesmo, chegando a ter velocidades superiores ás redes cabeadas, fazendo com que cada vez mais houvesse necessidade de criar variações de padrão para que se adequasse a realidade (Figura 5, página 24). Os padrões criados foram:

802.11a: Esse padrão ficou disponível em 1999, tornando sua principal característica, as velocidades que variavam entre 6, 9, 12, 18, 24, 36, 48 e 54 Mb/s, alcançando geograficamente uma média de transmissão de 50 metros.

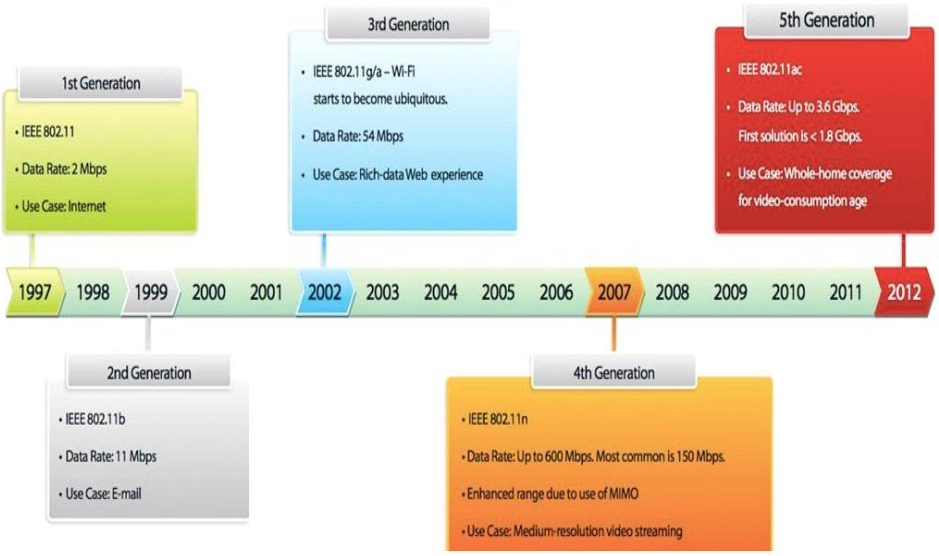
802.11b: No início de 1999, foi lançada a primeira atualização do padrão 802.11, conhecida como 802.11b, a qual apresentava como principais características chegar a velocidades de 1, 2, 5,5, e 11Mb/s, sendo que sua área de transmissão poderia chegar até 400 metros em locais abertos e 50 metros em lugares fechados, podendo sofrer variações desde o tempo como obstáculo no caminho da transmissão.

802.11g: Este padrão ficou disponível a partir de 2003 e veio para ser o substituto do padrão 802.11b, sendo que o mesmo é totalmente compatível entre eles, o 802.11g possui taxa de transmissão igual à do padrão 802.11a, e sua diferença ficou nas frequências utilizadas que passaram a ser 2,4Ghz e os canais de 20Mhz, possuindo a mesma cobertura de seu antecessor, passando também a surgir mais uma técnica de transmissão chamada de DSSS.

802.11n: O padrão 802.11n teve seu início de desenvolvimento em 2004 mas foi finalizado apenas em setembro de 2009. Tendo nesse intervalo de tempo vários dispositivos utilizando esse padrão, mesmo ainda em estado de desenvolvimento, sendo o sucessor do padrão 802.11g, tendo como principal diferença um esquema chamado de *Multiple-Input Multiple-Output(MIMO)*, aumentando assim consideravelmente as taxas de transmissão, levando a taxas entre 300Mb/s até podendo atingir taxas de 600Mb/s.

802.11ac: O padrão 802.11ac teve sua finalização e seu lançamento em 2012, tendo como principal mudança as taxas de transmissão que chegam até 6GB/s, trabalhando na frequência de 5Ghz, utilizando várias antenas, (no máximo 8).

Figura 5: Gerações dos Padrão 802.11



Fonte:(Reprodução/AnandTech, 2017)

2.4. METODOLOGIA DE DESENVOLVIMENTO:

Para desenvolver um aplicativo *Web* que faça uso dos conceitos de *Web Services,* deve-se levar em conta alguns pontos que serão necessários, tais como: reduzir os erros, fazer um levantamento de custos, levantamento de tarifas no mercado, verificar os riscos e impactos durante e depois do seu desenvolvimento, para que se possa alcançar objetivos desejáveis de qualidade, acompanhando também o seu ciclo de vida. Pensando nesses pontos deve-se procurar sempre uma boa qualidade de *software e* também muitas metodologias e processos que já foram pesquisadas e citadas no desenvolvimento do *software.*

Metodologia de Desenvolvimento é um conjunto de práticas recomendadas para o Desenvolvimento de *Softwares*, sendo que essas práticas, geralmente, passam por fases ou passos, que são subdivisões do processo para ordená-lo e melhor gerenciá-lo (SOMMERVILLE, 2000).

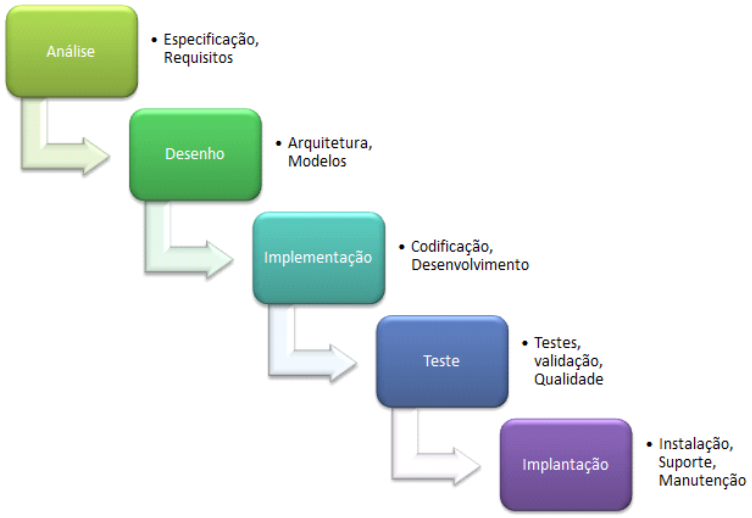
2.5. MODELOS DE CICLO DE VIDA

Em Engenharia de *Software,* os processos podem ser definidos para atividades como manutenção, aquisição, desenvolvimento e contradição. Vendo isso em um processo de desenvolvimento de *software*, o ponto iniciar a arquitetura de um processo é a escolha de um modelo de ciclo de vida.

2.5.1. Ciclo de Vida em Cascata

No modelo de Ciclo de Vida em Cascata (Figura 6), os processos são feitos e executados em extrema sequência, o que facilita marcá-los com pontos de controle bem definidos, tornando esse projeto confiável e utilizável em projetos de alto nível. Mas como todo ciclo tem seu lado bom e ruim, se for interpretado literalmente o mesmo irá se tornar um processo rígido e burocrático, em que a análise tem que ser bastante dominada, devido o processo não possibilitar correções posteriores nas fases anteriores, tornando assim o modelo de cascata uma baixa visibilidade para o cliente, que apenas no final recebe o resultado do projeto. (DE PÁDUA PAULA FILHO, 2013)

Figura 6: Ciclo de Vida em Cascata

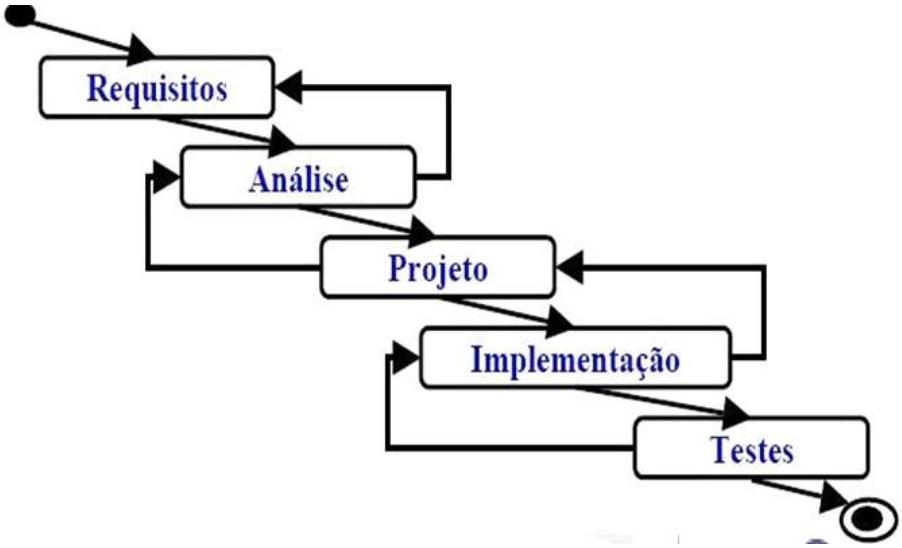


Fonte (SILVA, Mauro, 2016)

Analisando isso, tem-se que na prática será necessário corrigir nas fases posteriores, alguns erros ou mudanças ocorridas nas fases anteriores. Como exemplo um desenho que faltou algo e só foi percebido quando estava na fase de implementação,

à medida que a necessidade de algo novo, não pensado anteriormente e que vai sendo descoberto no decorrer do processo de implementação. Pensando nisso surgiu uma exceção que permite superposição entre fases chamada de Cascata com realimentação (Figura 7).

Figura 7: Ciclo de Vida em cascata com realimentação



Fonte: (VILAÇA, 2016)

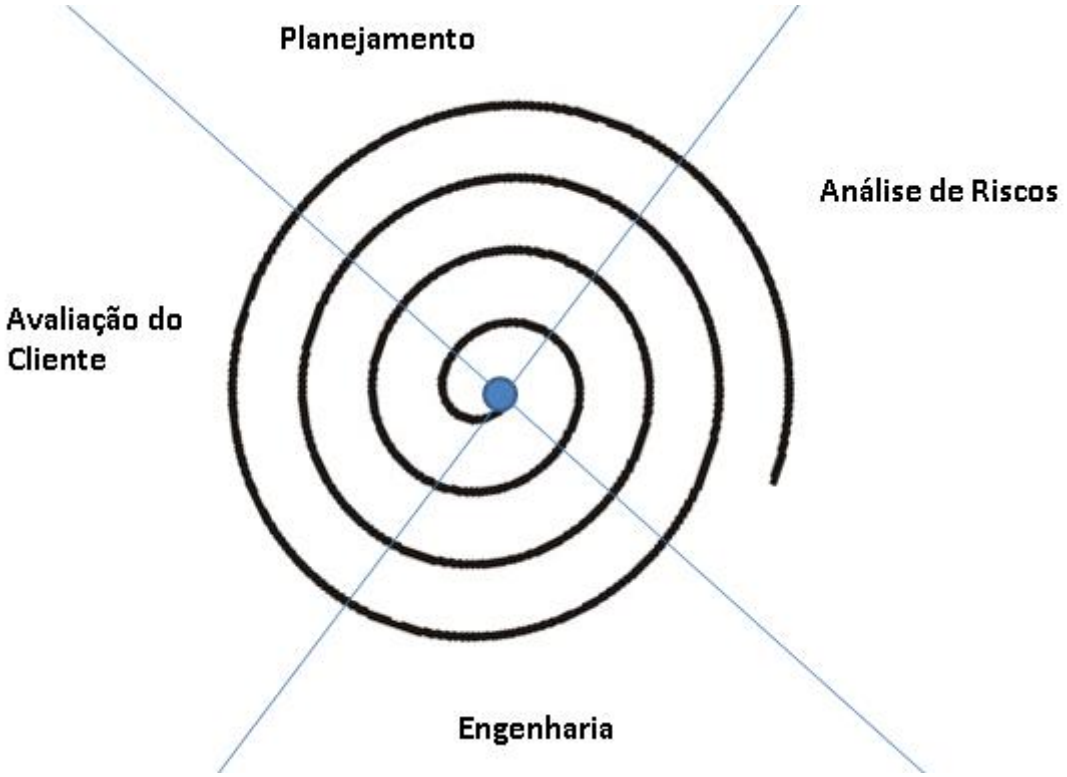
2.5.2. Ciclo de Vida em Espiral

O modelo em Espiral (Figura 8, página 28) é totalmente diferente do citado anteriormente. O modelo em Espiral é desenvolvido em uma serie de interações. Fazendo com que cada nova interação seja uma volta no Espiral, se for usado por liberação entende-se que é um estágio parcialmente operacional de um produto, que posteriormente é avaliado pelos usuários em um determinado ponto do projeto. As liberações escolhidas podem se tornar como versões oficiais do produto, e com isso já ser colocadas em uso, tornando assim novos produtos em curto prazo e à medida que vai se liberando, virá novos produtos com mais características e recursos que são agregados à medida que se descobre a sua necessidade. (DE PÁDUA PAULA FILHO, 2013)

O ciclo de vida Espiral permite que os requisitos se tornem definidos progressivamente, apresentando assim visibilidade e alta flexibilidade, sendo assim em um determinado ponto especifico, os usuários podem avaliar fases do produto e fornecer realimentação quando as decisões forem tomadas, facilitando assim o acompanhamento do progresso de cada projeto, tanto pelo cliente como pelos seus gerentes.

Esse modelo é aplicado principalmente em processos que se dizem ágeis.

Figura 8: Modelo Espiral



Fonte: (AUDY, PRIKLADNICKI, 2007)

2.6. REVISÃO DE LITERATURA

Segundo a Anvisa (2016) existe algumas diferenças entre Farmácia, Drogaria e Posto de Medicamento:

- Farmácia - estabelecimento de manipulação de fórmulas magistrais e oficinais, de comércio de drogas, medicamentos, insumos farmacêuticos e correlatos, compreendendo o de dispensação e o de atendimento privativo de unidade hospitalar ou de qualquer outra equivalente de assistência médica;
- Drogaria - estabelecimento de dispensação e comércio de drogas, medicamentos, insumos farmacêuticos e correlatos em suas embalagens originais.

A Farmácia Hospitalar é responsável pela armazenagem de estoque dos medicamentos que contem no hospital.

Segundo Maia (2005), dentre os principais objetivos da Farmácia Hospitalar, podemos destacar pela sua importância e abrangência:

1. Planejamento, aquisição, análise, armazenamento, distribuição e controle de medicamentos e correlatos (Logística);
2. Desenvolvimento e/ou manipulação de fórmulas magistrais e ou/ oficinais;
3. Produção de medicamentos e correlatos;
4. Desenvolvimento de pesquisas e trabalhos próprios ou em colaboração com profissionais de outros serviços;
5. Desenvolver atividades didáticas;
6. Adequar-se aos problemas políticos, sociais, econômicos, financeiros e culturais do hospital;
7. Estimular a implantação e o desenvolvimento da Farmácia Clínica.

Ainda, segundo MAIA (2005): A logística é a atividade responsável pelo desenvolvimento de ações que integrem o suprimento físico de recursos materiais na relação fornecedor/empresa; fluxo físico e movimentação de medicamentos e materiais no ambiente de produção/ transformação/estocagem e fluxo dos produtos acabados que são dispensados ou utilizados pelos pacientes do hospital. Essas ações devem conduzir a empresa/instituição a um patamar de custos, prazos e qualidade, constituindo um nível

aceitável de agregação de valor dos produtos (medicamentos, materiais médico-hospitalares, etc).

3. ETAPAS DE DESENVOLVIMENTO DO CONSFARM

Nesta seção será citado todos as metodologias utilizadas para o desenvolvimento do aplicativo.

3.1. LEVANTAMENTO DE REQUISITOS:

Esta seção irá tratar de todos os requisitos funcionais e não funcionais do aplicativo.

3.1.1. Requisitos Funcionais:

O aplicativo deverá fornecer uma página de consulta tendo nessa página a opção de consultar pelo o que o usuário deseja, se for medicamento o mesmo poderá pesquisar pelo código de barra, pelo nome do medicamento ou até mesmo pelo seu princípio ativo, sendo possível filtrar a pesquisa apenas na cidade que deseja obter o medicamento.

O aplicativo terá uma tela de login para que no futuro possa cadastrar os usuários e para que os mesmos possam utilizar o sistema para entrega a domicílio, tornando-se assim um aplicativo de consulta e entrega de medicamentos.

3.1.2. Requisitos não funcionais:

O aplicativo possuirá uma classe de adaptação vendo que de acordo com a lista ele irá criar linhas com seus respectivos conteúdos para cada medicamento encontrado.

3.1.3. Tópicos e suas Prioridades: Essencial, Importante, Desejável.

Requisitos Funcionais:

- Login e Senha para Acesso:
- Descrição: O login e senha terá prioridade média, pois trata-se apenas de um registro, para que no futuro possa ser colocado um controle de quantos usuários acessam o sistema.
- Cadastro de Usuário:
- Descrição: O Cadastro de Usuário será necessário para ter um login e senha para acesso ao sistema, como foi citado acima sem esse cadastro não terá como ter acesso ao sistema, tornando assim o cadastro de usuário um item essencial para o sistema.
- Medicamento:

Descrição: O Cadastro de Medicamento não terá prioridade, pois o banco de dados já irá trazer vários medicamentos cadastrados, tornando assim o cadastro de novos medicamentos importante para o sistema quando necessário.

3.2. PROPOSTA DO APLICATIVO:

3.2.1. Aplicação em *Android:*

Utilizar a aplicação em *Android* tornou mais fácil a comunicação de dados com o *Web Service*, fazendo com que o aplicativo faça seu papel e se comunique com o *Web Service* de forma rápida utilizando o protocolo REST.

Sendo assim o usuário, por exemplo, pede para pesquisar um medicamento com o princípio ativo “Dipirona”, o aplicativo irá solicitar o serviço através do *Web Service*, o mesmo irá pesquisar nos bancos de dados das farmácias e irá retornar as informações de nome de medicamento, o seu código de barras, laboratório e onde está disponível, de acordo com a cidade que o mesmo escolheu. Caso não seja encontrado o medicamento ou o usuário estiver digitado errado, o aplicativo o notificará que o medicamento não foi localizado ou não possui disponibilidade nas farmácias da região.

O aplicativo necessita do uso de *internet*, pois todo os dados serão armazenados no *Web Service* e os medicamentos permanecerão no banco de dados das farmácias, sendo assim possível apenas a consulta a medicamento em tempo real, tendo como ponto positivo, saber se naquele momento ainda tem o medicamento e como ponto negativo a interrupção do serviço quando o mesmo não encontrar acesso a internet, informando assim ao usuário a falta de comunicação.

3.2.2. Web Service:

O Web Service se tornou uma das maneiras mais fáceis de concentrar todos os serviços de uma aplicação. Sua utilização fazendo o meio de campo entre a aplicação e o banco de dados se tornou mais fácil, pois a aplicação será responsável por acessar os serviços e o *Web Service* busca a resposta do banco e retorna o que foi pedido pela aplicação.

Os serviços realizados pelo *Web Service,* deverá consultar os bancos de dados para obter a informação se os medicamentos estão disponíveis e em quais farmácias os mesmos se encontram, retornando assim a resposta para que o aplicativo possa tratar das informações e mostrar o melhor resultado para o usuário, tornando possível a utilização do *Web Services* a se comunicar com qualquer banco, fazendo com que na aplicação seja utilizado o uso do *Web Service* em 2 tipos de bancos: Mysql e PostgreSql.

3.2.3. Simulação:

Como proposta para o aplicativo irá ser realizada uma simulação para que se torne válido o projeto. Na simulação serão criados 2 dispositivos utilizando um o Windows 7 e o outro o Linux Ubuntu, fazendo com que seja possível mostrar que a comunicação *Web Service* se comunica normalmente entre esses dois sistemas operacionais. Dentro desse sistema operacional, em cada um irá ter um banco de dados, sendo um o Mysql e o outro *Postgresql.* Como citado anteriormente, nesses bancos de dados foram cadastrados 5 produtos reais, que existem no mercado, sendo feito 3 categorias, medicamento, perfumaria e suplemento. Após o término do cadastro irá simular o uso do

programa com dois usuários já cadastrados, sendo um dos usuários com o nome Jorge e o outro UERN.

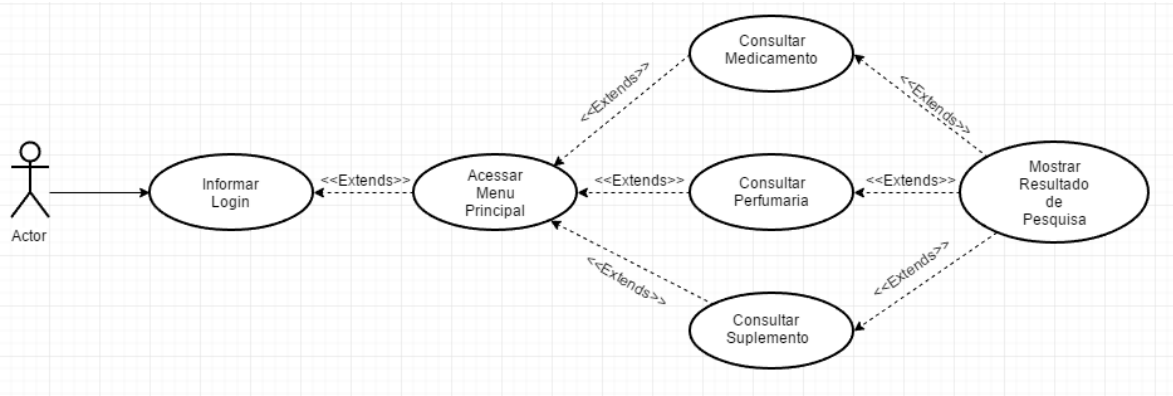
3.3. DESENVOLVIMENTO DO APLICATIVO

O aplicativo se iniciou com o pensamento da forma que seria mais prática e visual para que todos os tipos de usuários pudessem utilizar sem problema. No início já foi feito com o intuito de algo inédito e que seria inovador para o ramo de Tecnologia e Saúde. Iniciando o desenvolvimento do aplicativo resolveu-se implementá-lo no *Android Studio*. Abaixo será mostrado o passo a passo realizado até o final da implementação do aplicativo e o Servidor *Web Service*.

3.4.1. Fases de Desenvolvimento

O digrama de uso (Figura 9), mostra os passos dados pelo usuário até chegar no resultado de pesquisa, estes quadrados demonstram as telas com as quais os usuários irão se deparar ao selecionar as informações. A Figura 9 consiste neste esquema:

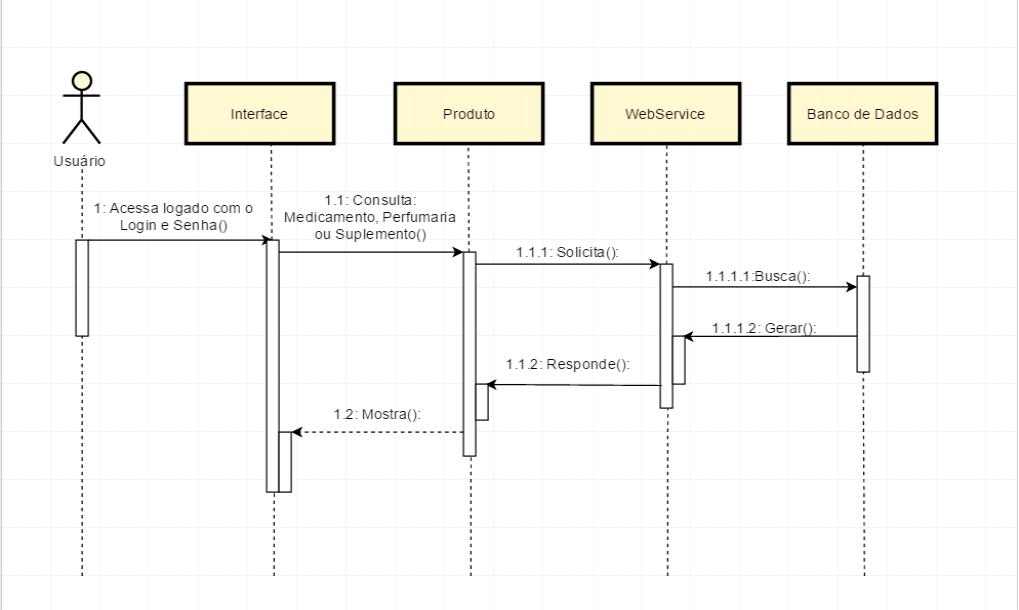
Figura 9: Diagrama de Caso de Uso



Fonte: (Próprio autor)

O diagrama de sequência (Figura 10, página 34) faz parte do diagrama da UML que representa as mensagens trocadas entre os objetos de um sistema. A figura abaixo um exemplo de diagrama de sequência quando um usuário abre o programa até chegar na consulta.

Figura 10: Diagrama de Sequencia



Fonte: (Próprio Autor)

E este diagrama irá mostrar que ao acessar o aplicativo o usuário necessita logar através do login e senha. Quando digitado corretamente o mesmo é levado ao menu principal, selecionando qual produto deseja pesquisar. Por exemplo, se ele deseja pesquisar um medicamento, o mesmo irá selecionar medicamento, clicar qual meio de pesquisa deseja utilizar, se selecionado código de barra, ele deverá digitar o código e solicitar (clicando no botão de pesquisa), ao WebService que busque no banco de dados se o medicamento se encontra em alguma farmácia. O banco de dados irá retornar uma resposta, se for um sim mostra os medicamentos disponíveis e em quais farmácias o usuário irá encontrar, se for um não, mostra uma mensagem de “Medicamento não encontrado”.

4. RESULTADO E DISCUSSÕES

Nesta seção irá ser discutido e apresentado todos os resultados do aplicativo desde sua criação até sua conclusão para o presente trabalho.

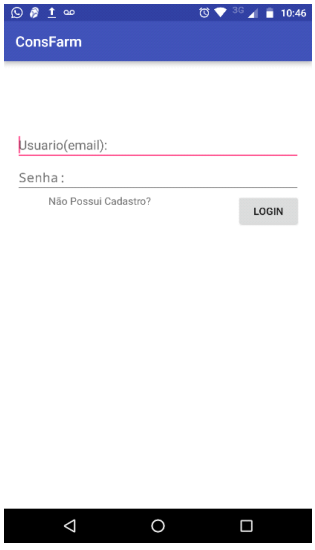
4.1. PASSO A PASSO DO APLICATIVO

- Tela Login:

Inicialmente foi analisado implementar um login, para que no futuro, se desejar levar o aplicativo para a realidade, seja necessária uma confiabilidade, visando que para um próximo passo do aplicativo será a encomenda do produto através do aplicativo e a Drogaria poderá fazer entregas a domicílio.

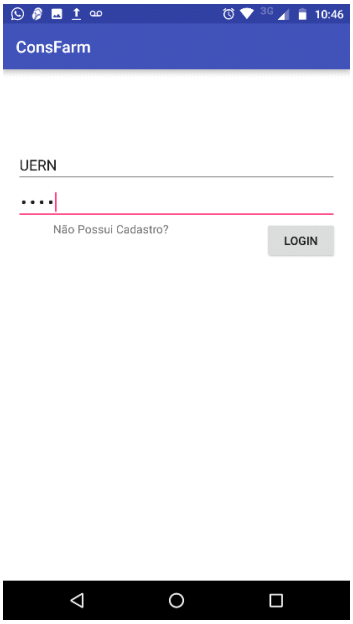
A tela Login possui dois *editText* que o usuário terá que preencher com login e senha (Padrão da Aplicação: Login UERN, senha 1234) para validação dos dados. Logo que digitado o mesmo deverá apertar no botão login. Caso a pessoa não possua cadastro a mesma poderá clicar na mensagem de “não possui cadastro” e realizar seu cadastro na hora, sendo que para este trabalho não será necessário, pois será utilizado o padrão já criado anteriormente. A Figura 11, demonstra os dados sem estarem preenchidos e na Figura 12(página 36), os dados já preenchidos pelo usuário pronto para ser verificado no *WebService.*

Figura 11: Tela Consultar medicamento



Fonte: (Próprio Autor)

Figura 12: Tela Login do Aplicativo, Preenchido pelo Usuário



Fonte: (Próprio Autor)

Logo após ser preenchido, como mostra a imagem acima, o usuário clica em login e se tudo estiver certo, ele será direcionado para a tela do menu principal. Caso algum dado esteja errado o mesmo será notificado através do tratamento de erro, mostrando para ele uma notificação de “Login ou senha Incorreta”, caso algum dos campos seja deixado em branco, o software informará quais deverão ser preenchidos.

- Tela Menu Principal

Na tela do menu principal (Figura 13, página 37), o usuário ira dispor de três opções colocadas como imagemButton, e uma como texto de edição, onde o mesmo poderá procurar por medicamentos, perfumaria, suplementos ou até mesmo consultar os produtos que as farmácias contêm. Para isso ele terá disponível um *EditText* para digitar o nome da Farmácia que deseja consultar os produtos e clicar no menu de pesquisa. Abaixo tem-se a imagem do menu principal.

Figura 13: Menu principal Aplicativo

Fonte: (Próprio Autor)

Logo após chegar na tela acima, o usuário deverá clicar nos ícones referentes ao que deseja pesquisar, ou digitar onde está escrito “nome do estabelecimento” o local onde deseja ver todos os produtos.

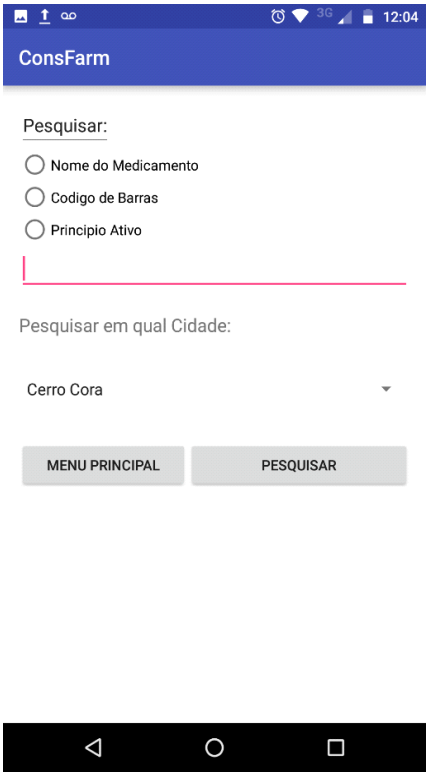
Dando continuidade, será demonstrado abaixo todas as telas que o usuário irá ver após este *menu*, dependendo do botão no qual o mesmo apertar. Se apertar em medicamento, ele será movido para a tela da consulta de medicamento, caso ele decida optar pela perfumaria, será direcionado para a tela de perfuraria e o mesmo acontece com a tela de suplemento.

- Tela Consulta de Medicamento

Na tela de consultar medicamento o usuário terá 3 opções de consulta, podendo consultar por nome do medicamento, código de barras e princípio ativo. Após selecionar um dos *RadioButton,* ele irá digitar o que se refere a sua pesquisa, em seguida o mesmo terá que filtrar a cidade onde deseja pesquisar fazendo com que isso facilite sua pesquisa

e o direcione para as farmácias do local que ele está. Na Figura 14 vemos a tela de consultar medicamentos sem o usuário ter digitado nenhuma informação:

Figura 14: Tela Consultar medicamento



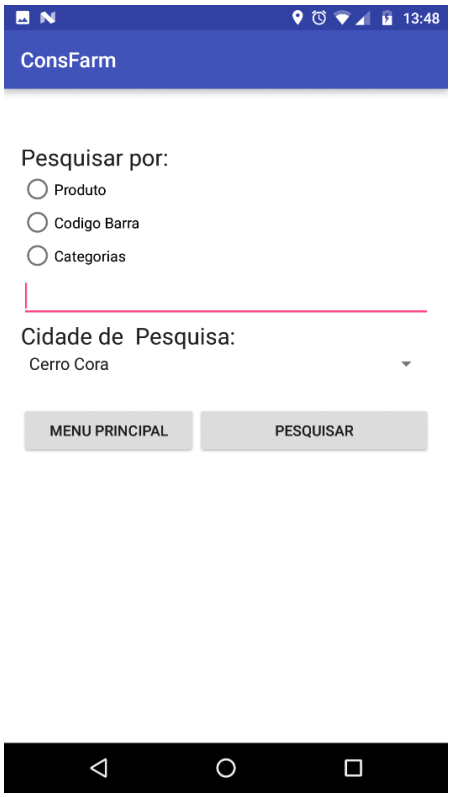
Fonte: (Próprio Autor)

Após preencher todos os dados, o mesmo receberá uma resposta fornecida pelo *Web Service*, que quando enviado a solicitação irá iniciar os serviços de buscar nos bancos de dados das referidas farmácias do local indicado para procura. A próxima tela é o objetivo desejado pelo nosso aplicativo.

- Tela Consulta a Perfumaria e suplemento

A tela de consulta a perfumaria ou suplemento (Figura 15, página 39) contém os mesmos critérios da tela de medicamentos, mudando apenas os *RadioButton* para algumas informações em relação a perfumaria. Nesses *RadioButton* poderá ser selecionado produto, código de barra e categoria, onde categoria é o tipo do produto por exemplo, perfume, ou desodorante entre outros. Abaixo segue uma imagem da tela de perfumaria:

Figura 15: Tela Pesquisar Perfumaria ou suplemento



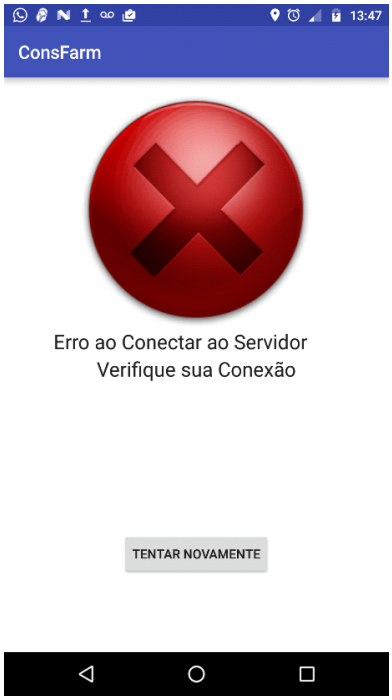
Fonte: (Próprio Autor)

- TELA ERRO, SEM CONEXÃO COM INTERNET

Essa tela (Figura 16, página 40) foi colocada dentro da exceção do *catch* referente a conexão com o banco, caso o aplicativo tente acessar o *Web Service* e o mesmo esteja fora do ar, sendo conduzido para a exceção de erro.

O aplicativo foi feito idealizado para que em nenhum momento pare de funcionar dando erros, e sim demonstrando para o usuário que ele tem algum problema relativo a uma questão externa do aplicativo, tendo que verificar a internet, por exemplo. Ou se o problema não for internet, poderá ser alguma manutenção no servidor. Isso possibilitará ao usuário confiabilidade no aplicativo. Dentro da tela terá um botão que o mesmo poderá usar para retornar ao login, e com isso não haverá necessidade de fechar o aplicativo após a conexão voltar ativa.

Figura 16: Tela, Erro de Comunicação com Servidor

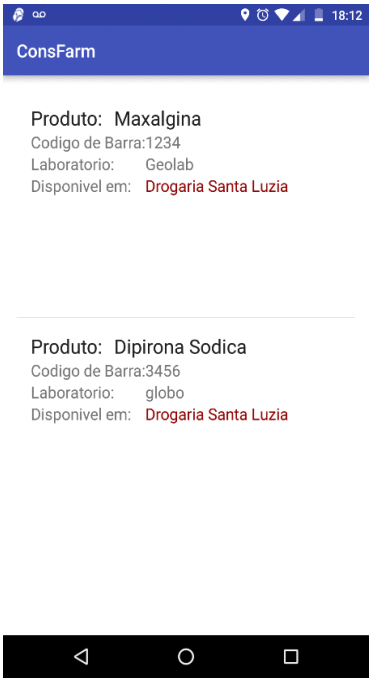


Fonte: (Próprio Autor)

- Tela Resultado da Busca

Essa tela (Figura 17, pagina 41) é o resultado final do trabalho, ela irá mostrar em uma lista, todos os medicamentos disponíveis em relação a consulta feita pelo usuário. No exemplo da foto foi procurado o medicamento com o princípio ativo chamado dipirona, e foi retornada uma lista na qual se encontrava esse remédio.

Figura 17: Tela Resultado da Busca



Fonte: (Próprio Autor)

Nesta Seção foram vistas todas as telas do resultado obtido pelo CONSFARM, durante a inicialização do aplicativo.: tem-se a tela de login, para que o usuário possa se credenciar para uso do aplicativo, lembrando que se o login ou senha estiver incorreto, o mesmo será notificado através de mensagem que irá aparecer em sua tela. Se tudo estiver correto o aplicativo irá abrir a tela do menu principal, onde o usuário terá quatro opções de livre escolha, tendo que ser escolhida apenas uma. Ao escolher buscar itens de uma farmácia, o mesmo terá que digitar o nome da farmácia e a cidade onde ela se encontra. Caso ele erre o nome da farmácia, será novamente notificado com uma mensagem na tela, mostrando que ele errou este nome, caso coloque o nome da farmácia corretamente será listado na tela de resultado todos os produtos e medicamentos localizados nesta farmácia. Tornando assim o aplicativo com um fácil acesso ao que o usuário deseja de forma direta e clara.

5. CONCLUSÃO

A utilização de um aplicativo, como o CONSFARM, gera um benefício enorme para a população, trazendo comodidade, eficiência, e agilidade na compra de medicamentos tendo, com isso, um ganho para a população em termo de tempo e os comerciantes deste ramo.

O desenvolvimento deste sistema foi um pouco complexo, devido à necessidade de fazer uma comunicação entre três programas, o que se traduziu no final, em maior segurança e eficiência, utilizando o protocolo REST, gerando a possibilidade de utilizar qualquer banco de dados para se obter dados através do WebService.

Ao final foram atingidos os seguintes objetivos:

- Objetivo 1: Desenvolver uma aplicação *Android:*

  A aplicação foi desenvolvida através do Android Studio, tendo como resultado final as imagens geradas neste TCC. Todas as imagens foram capturadas utilizando o botão de *PrintScreen* do *Smartfone*, com o aplicativo funcionando no sistema Operacional (SO) Android.

- Objetivo 2: A aplicação se comunicar com as farmácias por meio de um *WebService.*

  Objetivo conquistado com sucesso. Para que esse objetivo fosse alcançado, foram criadas duas maquinas virtuais simulando as farmácias, tendo como resultado a comunicação e retorno da resposta através da lista, como foi ilustrado na figura 17 da tela de resultados.

Sendo assim, o sistema CONSFARM comprovou sua eficiência, atendo aos propósitos desejados nesse trabalho.

5.1. TRABALHOS FUTUROS

Para trabalhos futuros, irá ser implementado a parte de venda e a inclusão de preço, pois não foi feito antes, para que não houvesse problema com o credenciamento das farmácias por parte dos comerciantes que não desejam divulgação. Como citado anteriormente no trabalho, será criado o cadastro de usuários para ter maior segurança na compra dos medicamentos.

5.2. CONTRIBUIÇÕES POSITIVAS E DIFICULDADES ENCONTRADAS

O CONSFARM vem para mudar a rotina das pessoas, diminuir a busca por medicamentos em muitas farmácias e assim consequentemente tornar o dia das pessoas que sofrem por algum problema ou precisam da ajuda de outra pessoa que encontre logo a solução.

Houve algumas dificuldades encontrada, como exemplo, como seria para obter os dados da farmácia. Para isso será criado um programa que será instalado na farmácia e assim se obter os dados e enviar para um banco nas nuvens.

REFERÊNCIAS BIBLIOGRÁFICAS

ALECRIM, Emerson, **O que é Wi-Fi (IEEE 802.11)?**, disponível em: http://www.infowester.com/wifi.php, acesso em 12. Out.2016.

ANDROID**, pagina oficial,** disponível em: <https://www.android.com/intl/pt-BR_br/>, acesso em: 07. Mar. 2017

ANVISA: **Conceitos**, disponível no site: <http://www.anvisa.gov.br/medicamentos/conceito.htm#1.10>, acesso em 15. Fev.2017

ANVISA, **RESOLUÇÃO DA DIRETORIA COLEGIADA - RDC Nº. 27, DE 30 DE MARÇO DE 2007,** disponível em: <http://www.anvisa.gov.br/sngpc/Documentos2012/RDC%2027%202007.pdf?id=26280 HYPERLINK =>, acesso em: 07.abr.2017.

AUDY, Jorge, PRIKLADNICKI, Rafael. **Desenvolvimento distribuído de Software**. Campus: São Paulo, 2007.

CARLOS, Jean, **Web SERVICE (SOAP X REST**), disponível em: <http://www.fatecsp.br/dti/tcc/tcc00056.pdf>, acesso em 18.Mar.2017.

CONSTANTINOU, A., CAMILLERI, E. and KAPETANAKIS, M. (2013) **Developer Economics Q3 2013: State of the Developer Nation**. Londres: Visionmobile, 2013.

DAL MORO, Tharcis, Carina Friedrich Dorneles, and Marcelo Trindade. **Web services WS-* versus Web Services REST.** Revista de Iniciação Científica 11.1 (2009).

DE PÁDUA PAULA FILHO, Wilson. **Engenharia de software**. LTC, 2003.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. Tese (Doutorado em Informação e Ciência da Computação) – Universidade da Califórnia, Califórnia, 2000.

FIDDEL, Brunno, **Web Services REST versus SOAP,** disponível em: <http://www.devmedia.com.br/web-services-rest-versus-soap/32451>, acesso: 28.Março.2017

FONSECA, Daniel, **Conceitos básicos sobre Metodologias Ágeis para Desenvolvimento de Software (Metodologias Clássicas x ExtremeProgramming)**, disponível em: <http://www.devmedia.com.br/conceitos-basicos-sobre-metodologias-ageis-para-desenvolvimento-de-software-metodologias-classicas-x-extreme-programming/10596>, acesso em: 06.Jul.2016

LUIZ, Valéria Caroline; DE CASTRO, Adriane Belluci Belório; DE ALMEIDA, Osvaldo César Pinheiro**. Produção textual no ensino médio: proposta de desenvolvimento de um aplicativo utilizando web service.** Tekhne e Logos, v. 7, n. 2, p. 73-87, 2016

MAIA NETO, J. F. Logística e gestão de estoque. In: MAIA NETO, J.F. **Farmácia Hospitalar e suas interfaces com a saúde**. 1ª ed. São Paulo: Rx, 2005. cap. 3, p. 47 – 70. MALONE, P.M.; KIER, K.L. **Pharmacy and therapeutitics commitee**. In MALONE, P.M.; KIER, K.L.; STANOVICH, J.E. Drug information. A guide for pharmacists. Stamford: Appleton e Lange, 1994. p. 227-281.

MEIRELES, Maria Costa; BONIFÁCIO, Bruno. **Uso de métodos ágeis e aprendizagem baseada em problema no ensino de engenharia de software: Um relato de experiência**. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2015. p. 180.

PLAZA, Willian, **Entendendo o Wi-Fi 802.11ac**, disponível em: http://www.hardware.com.br/artigos/entendendo-wifi-802.11-ac/, acesso em 12.Out.2016.

RNTV, **Médicos reclamam de falta de medicamentos no Hospital Walfredo Gurgel G1 Rio Grande do Norte RN,** disponível em <https://www.youtube.com/watch?v=eRRV6x-Tsuo>, acesso em 07.Abr. 2017

SILVA, Mauro, **Ciclo de Vida de um Projeto**, Disponível em: <http://docplayer.com.br/10204764-Ciclo-de-vida-de-um-projeto.html>, acesso em: 11.Out.2016

SOMMERVILLE, I., "**Software Engineering**", 6th edition, Addison-Wesley, 2000.

TORRES, Gabriel**. Redes de Computadores Curso Completo**. Rio de Janeiro: Axcel Books, 2001.

VIANA, Mauro, **Conheça o Rational Unified Process (RUP),** Disponível em: <http://www.linhadecodigo.com.br/artigo/79/conheca-o-rational-unified-process-rup.aspx>, acesso em 05. Jul.2016.

VILAÇA, Alberto, **Engenharia de Software**, Disponível em: <http://slideplayer.com.br/slide/3679832/>, Acesso em: 11.Out.2016.

APÊNDICES

APÊNDICE A: IMAGENS DA CONEXÃO DOS BANCOS COM O WEBSERVICE:

Figura 18: Tela Máquina Virtual Drogaria Sta Luzia

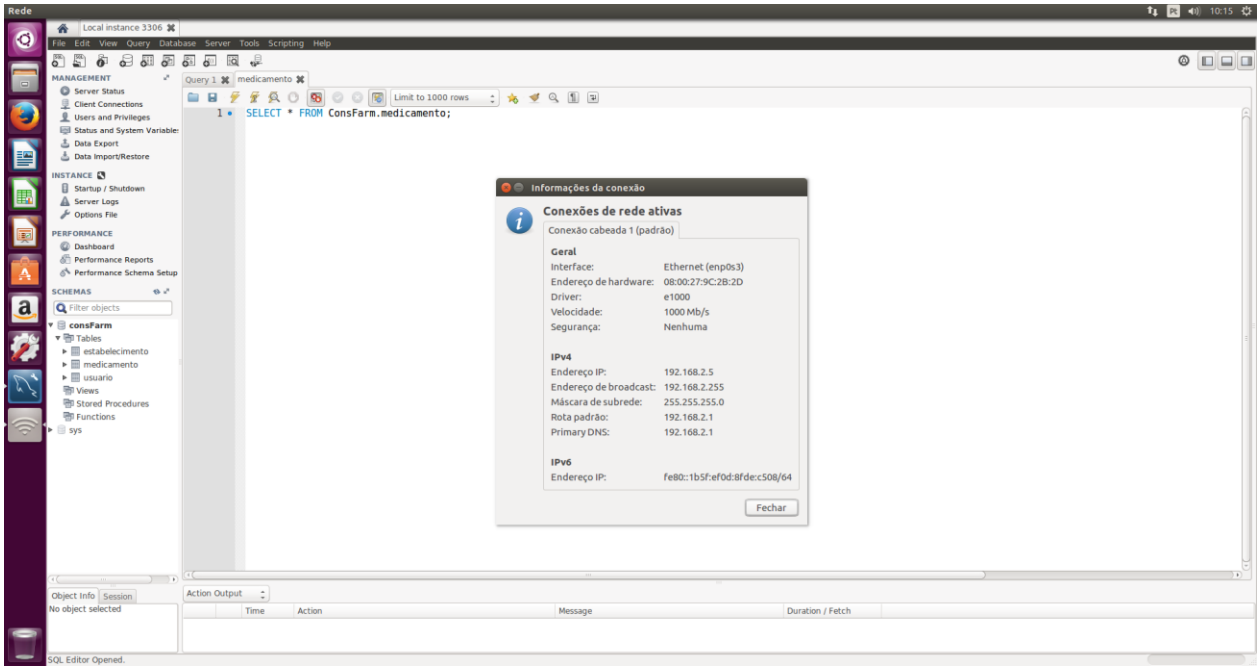


Figura 19: Tela Máquina Virtual Drogaria Seridó

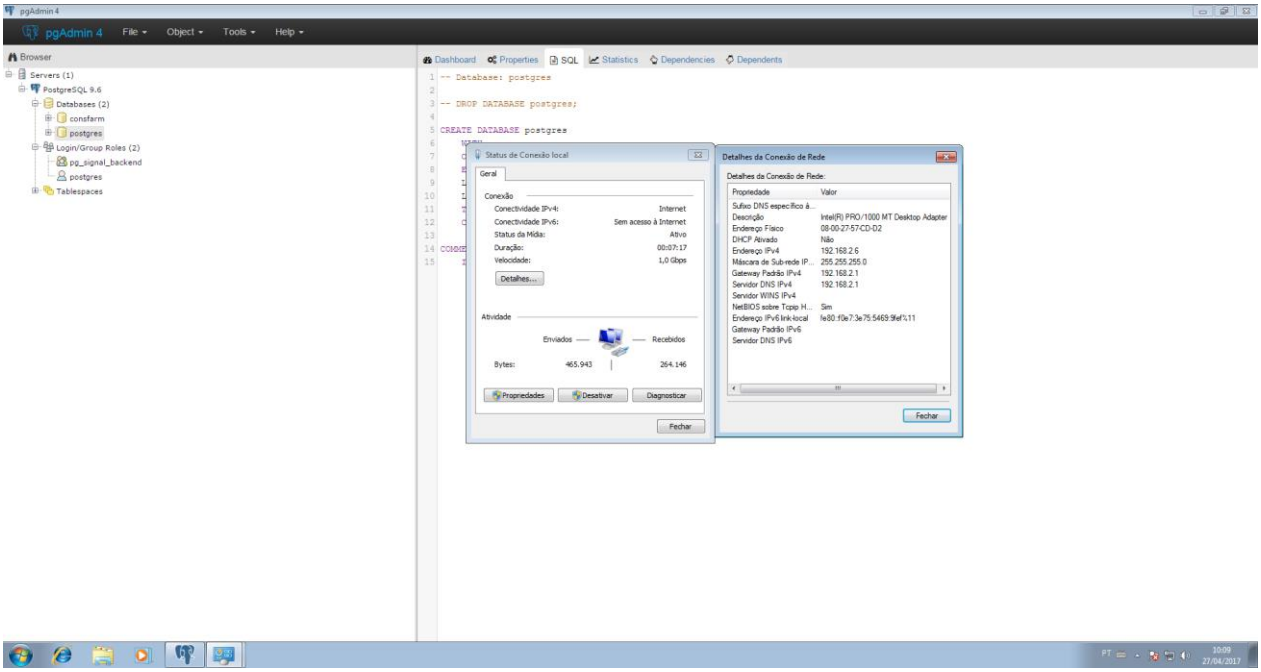
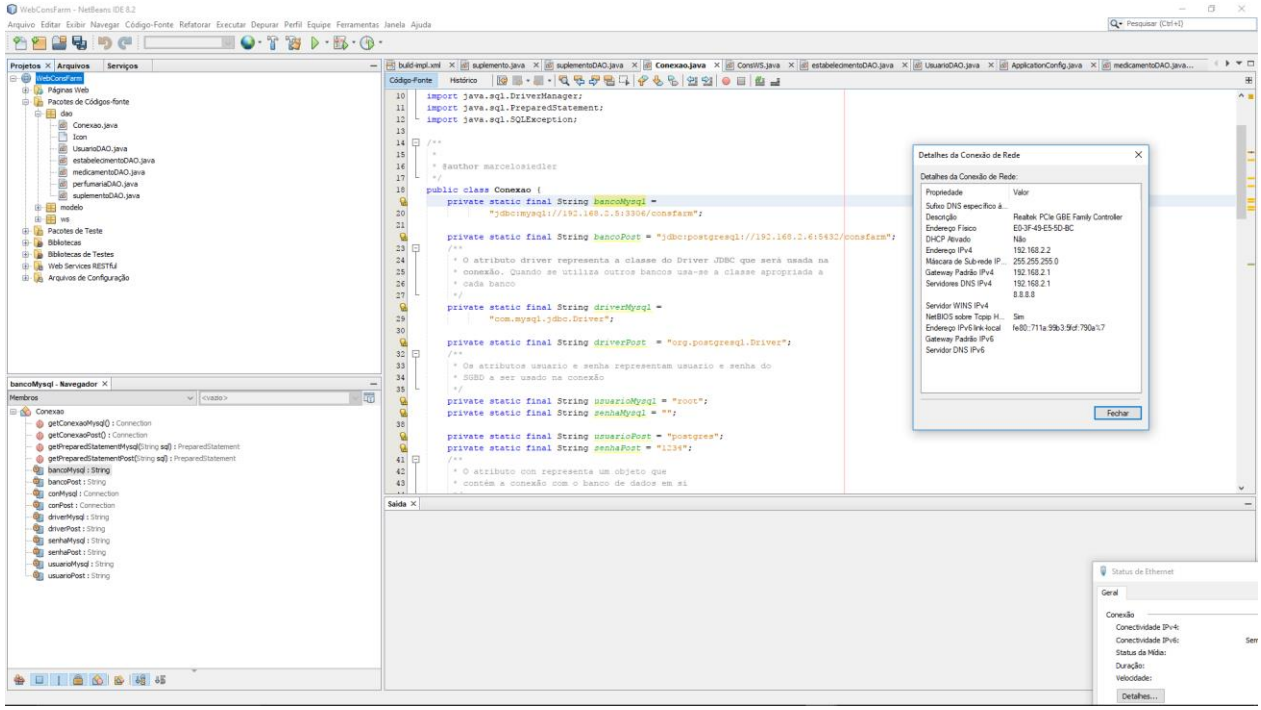


Figura 20: Tela Maquina Real, Servidor WebService



APÊNDICE B: IMAGENS DE ALGUMAS CLASSES DO WEBSERVICE:

Figura 21: Obtendo dados Perfumaria: WebService

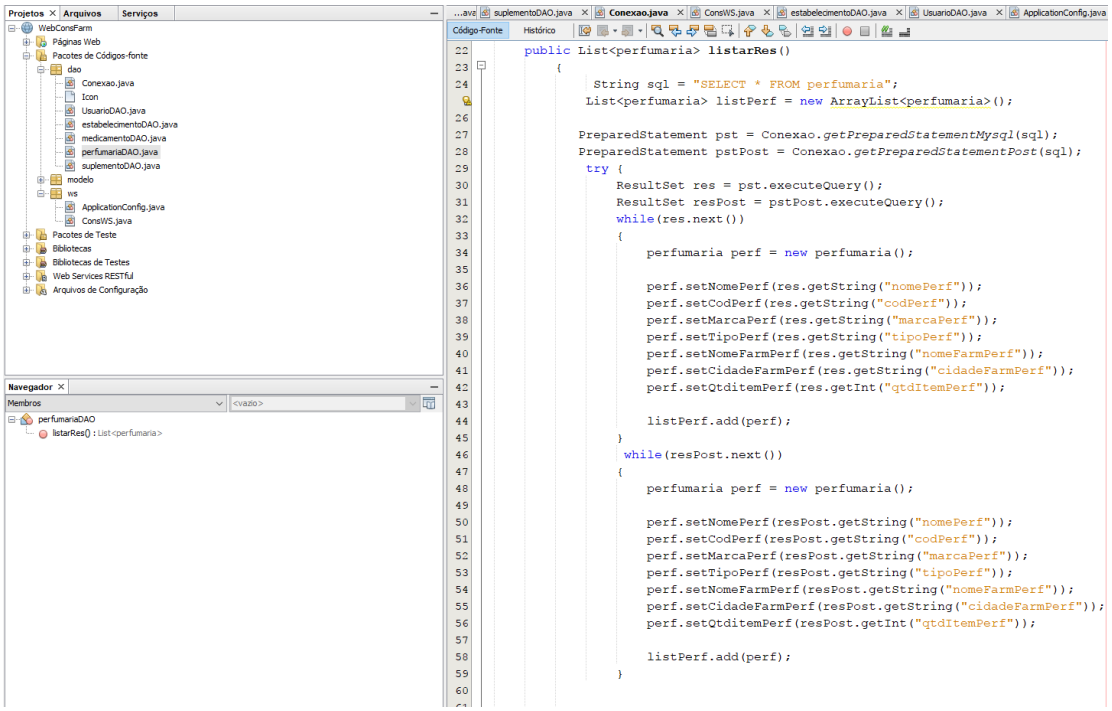
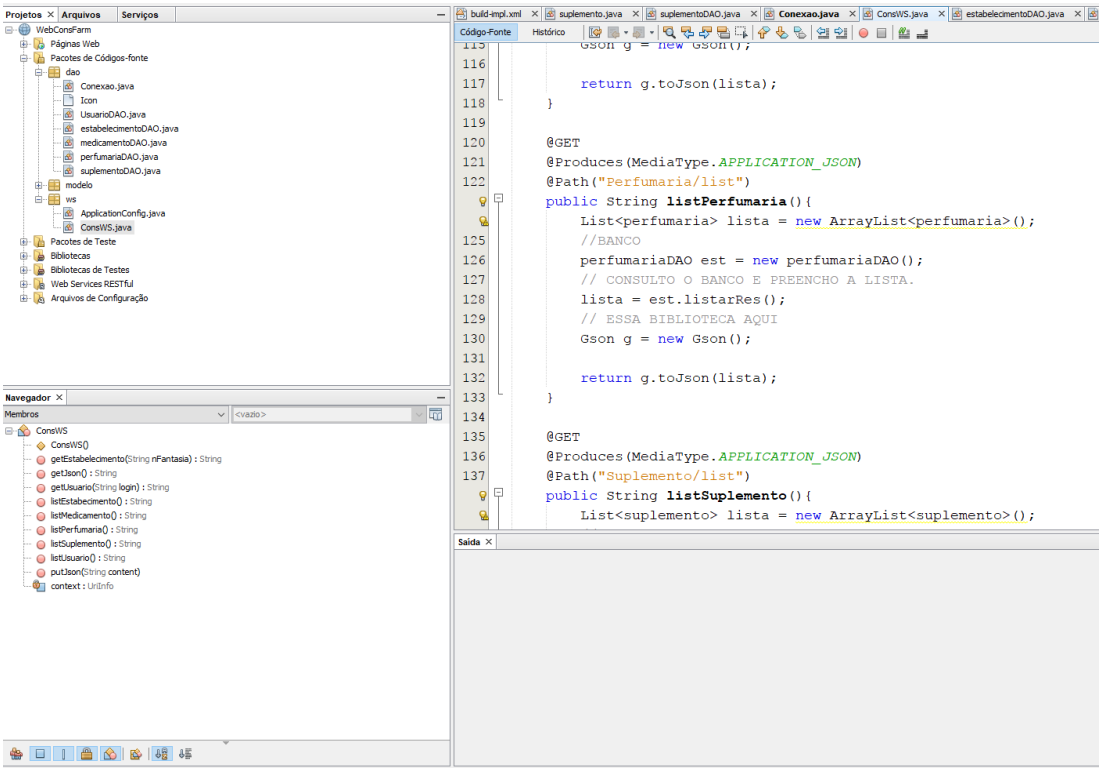


Figura 22: Consultando Perfumaria no Banco e Guardando em Lista.



APÊNDICE C – IMAGENS DE ALGUMAS CLASSES DO APLICATIVO ANDROID:

Figura 23: Implementação da Classe de Conexão com WebService

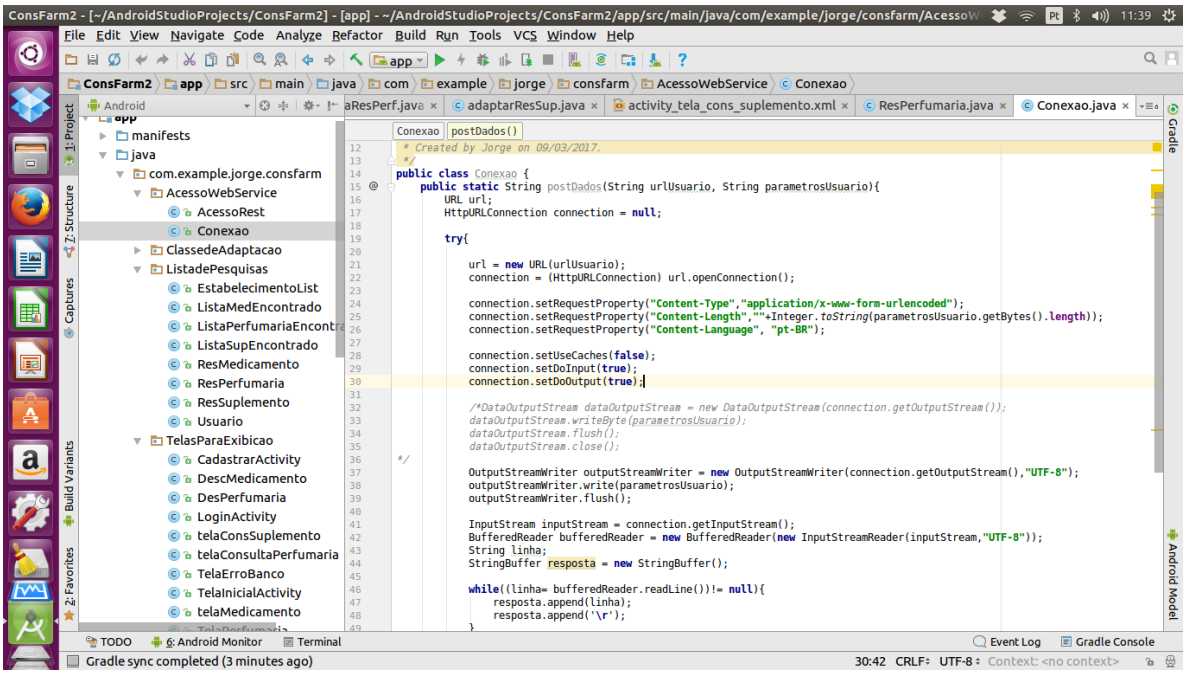


Figura 24:Implementação da Classe de adaptação dos Valores das Linhas

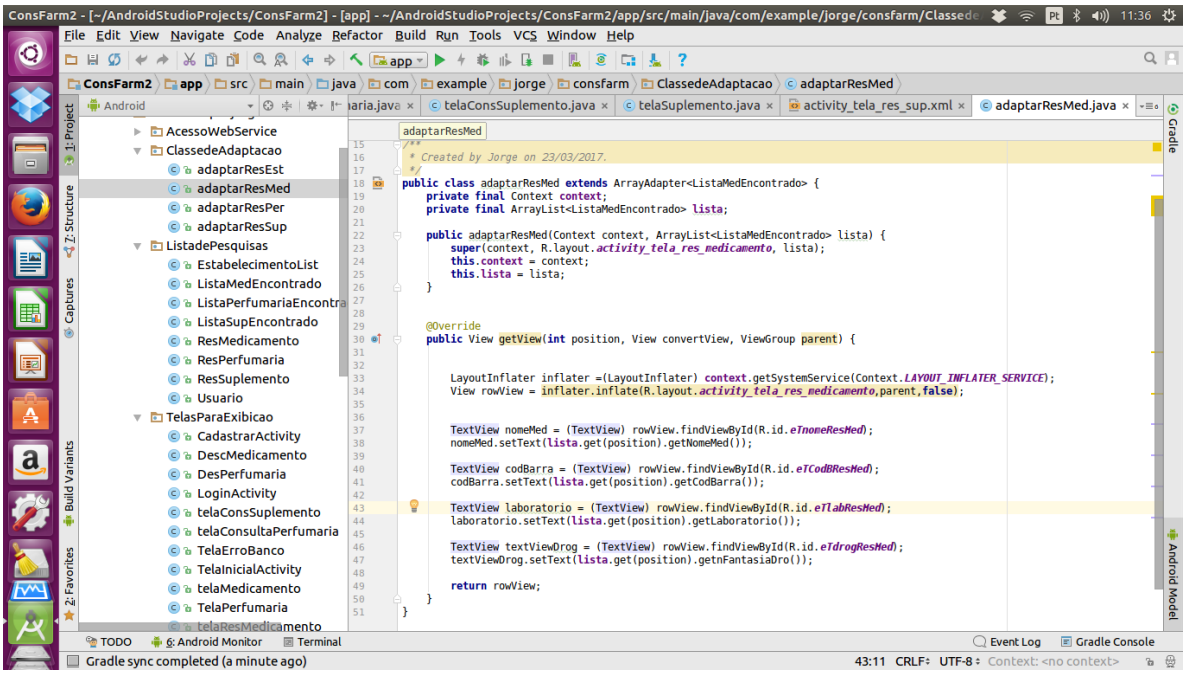


Figura 25: Obtenção dos valores informados pelos usuário e Consulta

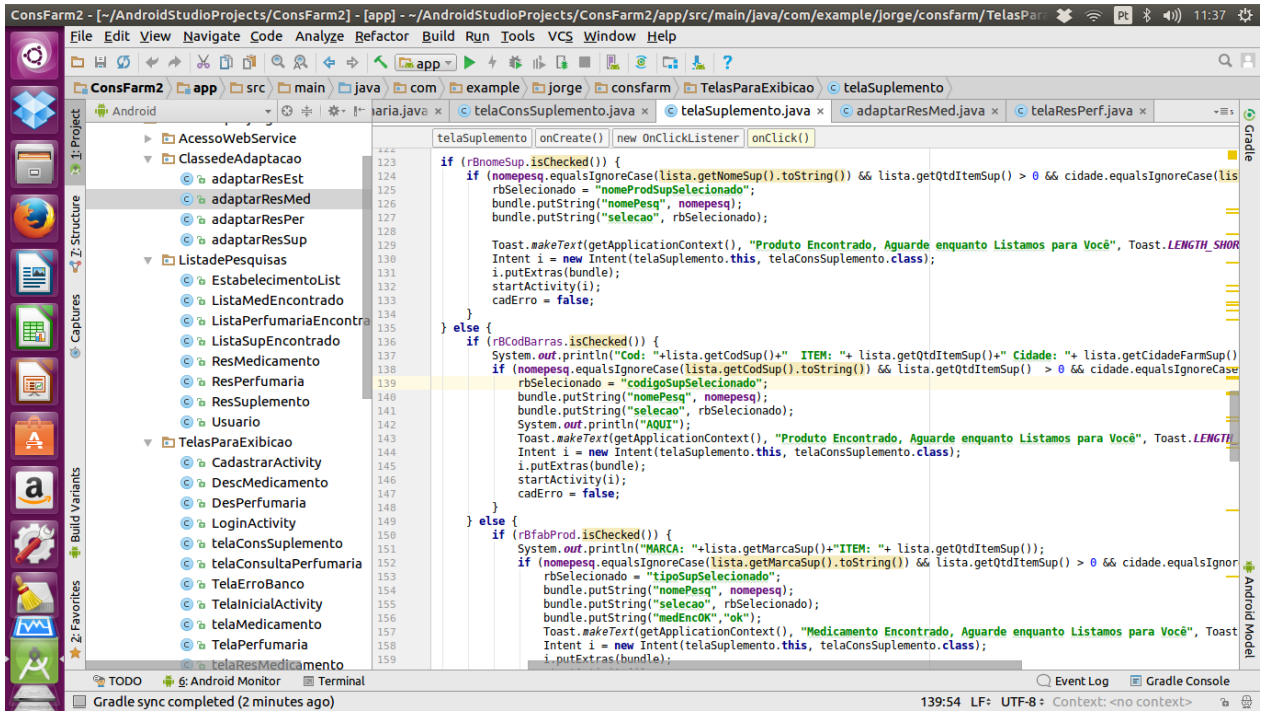


Figura 26: Criação das Variáveis para Guardar os valores vindo do WebService