

UNIVERSIDADE DO SAGRADO CORAÇÃO

KETLIN FABRI DOS SANTOS

**SISTEMA ESPECIALISTA PARA ANÁLISE DE
INTERAÇÕES MEDICAMENTOSAS PARA USO DE
ESTUDANTES E PROFISSIONAIS DE FARMÁCIA**

BAURU
2017

KETLIN FABRI DOS SANTOS

**SISTEMA ESPECIALISTA PARA ANÁLISE DE
INTERAÇÕES MEDICAMENTOSAS PARA USO DE
ESTUDANTES E PROFISSIONAIS DE FARMÁCIA**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para a obtenção do título de bacharel em Ciência da Computação, sob orientação do Prof. Dr. Elvio Gilberto da Silva.

BAURU
2017

Santos, Ketlin Fabri dos

S2375s

Sistema especialista para análise de interações medicamentosas para uso de estudantes e profissionais de farmácia / Ketlin Fabri dos Santos. -- 2017.

68f. : il.

Orientador: Prof. Dr. Élvio Gilberto da Silva.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade do Sagrado Coração - Bauru - SP

1. Aplicação. 2. Farmácia. 3. Interação. 4. Medicamentos.
I. Silva, Élvio Gilberto da. II. Título.

KETLIN FABRI DOS SANTOS

**SISTEMA ESPECIALISTA PARA ANÁLISE DE INTERAÇÕES
MEDICAMENTOSAS PARA USO DE ESTUDANTES E
PROFISSIONAIS DE FARMÁCIA**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para a obtenção do título de bacharel em Ciência da Computação, sob orientação do Prof. Dr. Elvio Gilberto da Silva.

Banca examinadora:

Prof. Dr. Elvio Gilberto da Silva
Universidade do Sagrado Coração

Prof. Me. Fernando Tozze Alves Neves
Universidade do Sagrado Coração

Prof. Me. Patrick Pedreira Silva
Universidade do Sagrado Coração

Bauru, 14 de novembro de 2017.

“Adoramos a perfeição, porque não a podemos ter; repugna-la-íamos se a tivéssemos. O perfeito é o desumano porque o humano é imperfeito. ”

(PESSOA, 1986)

RESUMO

Desenvolvimento de um Sistema Especialista para que estudantes e profissionais de farmácia da Universidade do Sagrado Coração possam fazer consultas à uma base de dados, através de dispositivos móveis, à uma lista de medicamentos que quando tomados juntos podem apresentar ou não interação. Para tais fins, foi criada uma aplicação híbrida, que atende ambos usuários dos sistemas operacionais Android e iOS, utilizando as linguagens HTML5 e JavaScript, além de uma API disponibilizada gratuitamente pela U.S. Natciona Library of Medicine, que possui uma rica base de dados com informações sobre interações desses medicamentos.

Palavras-chave: Aplicação. Farmácia. Interação. Medicamentos

ABSTRACT

Development of a Specialist System that allows students and pharmacy professionals from the Universidade do Sagrado Coração consulting a database of drugs-drugs interaction while using a mobile device. Thinking of the user facilities we have created a hybrid mobile application that gives not only Android but iOS users the opportunity of access our system, using the Ionic framework and programming in HTML5 and JavaScript. Also, we found a free API that is current available by the U.S. National Library of Medicine on internet that gives us a great web service for accessing a database where we can check drug-drug interactions.

Keywords: Application. Drugs. Interactions. Pharmacy.

LISTA DE ILUSTRAÇÕES

Figura 1 - Características de uma farmacoterapia ideal.....	16
Figura 2 - O progresso da Inteligência Artificial.....	21
Figura 3 - Esquema clássico de um Sistema Especialista	23
Figura 4 - Regras básicas de um mecanismo de inferência.....	25
Figura 5 - Visão geral do sistema.....	27
Figura 6 - Defeito x erro x falha	33
Figura 7 - Modelo V descrevendo o paralelismo entre as atividades de desenvolvimento e teste de software	37
Figura 8 - Início da aplicação desenvolvida em Ionic.....	48
Figura 9 - Exemplo de código desenvolvido.....	49
Figura 10 - Tela com função que chama a API	52
Figura 11 - Primeira tela do tutorial	54
Figura 12 - Segunda tela do tutorial	55
Figura 13 - Terceira tela do tutorial	55
Figura 14 - Última tela do tutorial	56
Figura 15 - Tela de boas-vindas do aplicativo.....	57
Figura 16 - Tela que apresenta a aba Itens	58
Figura 17 - Tela para inclusão de novos itens.....	58
Figura 18 - Tela de buscas de interações medicamentosas	59

SUMÁRIO

1	INTRODUÇÃO	7
2	OBJETIVOS	9
2.1	OBJETIVO GERAL	9
2.2	OBJETIVOS ESPECÍFICOS	9
3	POLIMEDICAÇÃO	10
3.1	INTERAÇÕES MEDICAMENTOSAS	10
3.1.1	Fatores ligados à administração medicamentosa	11
3.1.2	Fatores ligados aos pacientes	11
4	ATENÇÃO FARMACÊUTICA	13
4.1	PROCESSO GERAL DE ATENÇÃO FARMACÊUTICA DO PACIENTE	14
5	FARMACOTERAPIA	15
6	INTELIGÊNCIA ARTIFICIAL	18
6.1	INTELIGÊNCIA ARTIFICIAL X INTELIGÊNCIA HUMANA	19
6.2	SISTEMAS ESPECIALISTAS	22
6.2.1	Componentes de um Sistema Especialista	22
6.2.2	Mecanismos de Aprendizagem	23
6.2.2.1	Quadro Negro	24
6.2.2.2	Mecanismo De Inferência	24
6.2.3	Vantagens de um Sistema Especialista	25
6.2.4	Exemplos de aplicação	26
6.2.5	Custos	27
6.3	SISTEMAS DE PRODUÇÃO	27
7	ENGENHARIA DE SOFTWARE	29
7.1	INTERAÇÃO HUMANO-COMPUTADOR (IHC)	30
7.1.1	IHC e Engenharia de Software	30
7.1.2	Modelagem de software	31
7.1.3	Linguagem UML	31
7.1.4	Diagramas UML	32
7.2	TESTE DE SOFTWARE	33
7.2.1	Níveis de teste de software	36

7.2.2	Técnicas de teste de software	38
8	INFORMÁTICA APLICADA À SAÚDE	40
8.1	APLICAÇÕES DESENVOLVIDAS PARA A SAÚDE	40
9	TRABALHOS CORRELATOS	42
10	METODOLOGIA.....	43
10.1	SOFTWARES E TECNOLOGIAS UTILIZADOS	44
10.1.1	MODELAGEM DO BANCO	44
10.1.2	FERRAMENTAS PARA O DESENVOLVIMENTO	45
10.1.2.1	ANDROID.....	45
10.1.2.2	JAVA	46
10.1.2.3	IOS.....	46
10.1.2.4	APLICAÇÕES MÓVEIS HÍBRIDAS	47
10.1.2.5	API	47
11	DESENVOLVIMENTO.....	48
12	TESTES.....	53
12.1	TESTE DE INTERFACE DO SOFTWARE	54
13	CONCLUSÃO.....	60
14	REFERÊNCIAS	61

1 INTRODUÇÃO

Desde o século XIX a tecnologia tem dado importantes contribuições para a área médica no diagnóstico e tratamento de diferentes tipos de doenças, começando com a invenção do Raio X em 1895, passando pela invenção do Ultrassom (década de 1950), da Tomografia computadorizada (1961), e chegando à Ressonância Magnética e PET (Positron Emission Tomography), ambos na década de 1970. Através da análise de imagens cada vez mais precisas, é possível dizer de forma detalhada a situação de um determinado paciente.

Hoje, podemos dizer que a tecnologia já é condição indispensável para toda a área da saúde; ela é utilizada em exames simples e complexos, em locais como hospitais, clínicas e farmácias. Este último segmento, o de farmácias, à fim de assegurar qualidade e controle na prescrição de medicamentos, segue normas e padrões pré-estabelecidos por organizações internacionais como a Organização Mundial da Saúde.

Nesse aspecto, a fim de auxiliar esses profissionais e diminuir a possibilidade de erros na prescrição de medicamentos, o desenvolvimento de um aplicativo para dispositivos móveis (tecnologia) se justifica para a consulta, por parte de profissionais e alunos da área farmacêutica (saúde), das possíveis interações medicamentosas que podem ocorrer em caso da má administração de remédios, ou seja, à quais efeitos colaterais o paciente pode ou não estar sujeito se fizer o uso de dois medicamentos de composições químicas diferentes, ao mesmo tempo. Para que tudo isso seja possível, será construído um sistema especialista. A aplicação desses sistemas na área médica é de grande importância, já que esta apresenta problemas peculiares que requerem estudo complexo de suas causas.

Os erros que podem levar à interação medicamentosa durante a prescrição dos remédios pelos profissionais da área da saúde são muito comuns. É necessário conhecer de forma aprofundada que tipo de interações são mais frequentes e como elas se processam.

As opções de aplicativos nessa área ainda são poucos, somadas à real situação de que os médicos continuam consultando bulários para ver se existem interações entre medicamentos, o que leva tempo, fazendo com que muitas vezes eles até mesmo desistam de checar essas interações.

Este trabalho é motivado justamente pela dinâmica da descoberta e disponibilização, no mercado de consumo, de novos medicamentos, visando prover ao profissional prescritor de receita uma ferramenta de auxílio para a análise das interações medicamentosas em seus pacientes, interações essas muitas vezes sujeitas à identificação após a utilização dos medicamentos. Com base neste contexto, este trabalho se propõe a desenvolver um sistema especialista (SE) que faça a análise crítica bioquímica das drogas que compõem cada medicamento, e a forma com que eles podem interagir quando administrados num "coquetel" terapêutico.

2 OBJETIVOS

A seguir serão apresentados os objetivos geral e específicos desta proposta.

2.1 OBJETIVO GERAL

Desenvolver um sistema especialista para auxiliar o profissional da área de farmácia no acompanhamento das interações medicamentosas dos remédios administrados a seus pacientes.

2.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

- a) Realizar pesquisa bibliográfica sobre Inteligência Artificial na criação de Sistemas Especialistas, e sobre Polimedicação na área de Farmácia;
- b) Pesquisar sobre API e novas tecnologias de aplicações móveis;
- c) Analisar os dados informados pelo público-alvo;
- d) Estudar o sistema operacional Android, para qual a aplicação será desenvolvida;
- e) Desenvolver o software na linguagem HTML5, utilizando JavaScript, AngularJS e CSS, com uma interface que seja clara e de fácil utilização;
- f) Efetuar levantamento bibliográfico sobre interação de medicamentos;
- g) Analisar os resultados buscando aprimorar o software desenvolvido, e ao mesmo tempo, compreender de que maneira a tecnologia pode influenciar e colaborar com profissionais e estudantes da área de saúde, em específico, os de Farmácia;

3 POLIMEDICAÇÃO

Os medicamentos ocupam um papel importante nos sistemas sanitários, pois salvam vidas e melhoram a saúde (MARIN et al, 2003). Mas e quando há a necessidade de um indivíduo fazer o uso concomitante de dois ou mais medicamentos? À essa prática dá-se o nome de polimedicação.

Embora a Agência Nacional de Vigilância Sanitária (ANVISA) regule a venda de medicamentos sem a prescrição médica, com as facilidades oferecidas hoje não é difícil encontrar pacientes que procuram diversos médicos atrás de receitas para o tratamento de uma mesma doença, ou façam uso de internet para buscar informações sobre sua doença e remédios, acabando por se automedicar. Esta última prática representa um risco enorme à saúde, já que não passa pelo crivo de um profissional que faz prescrição de medicamentos. Esses pacientes ficam então sensíveis aos mais adversos efeitos colaterais. Medicamentos mal administrados podem causar desde alergias, envenenamentos, agravamento dos sintomas da doença, até reação com substâncias de outro medicamento, o que provoca interação medicamentosa e acaba por dar origem a um novo problema de saúde.

3.1 INTERAÇÕES MEDICAMENTOSAS

Interações medicamentosas são respostas farmacológicas ou clínicas resultantes da interferência da ação de um medicamento, alimento ou qualquer outra substância química quando administrada concomitantemente com um primeiro medicamento. Essas interações podem ser físico-químicas, farmacodinâmicas ou farmacocinéticas, onde:

- a) **Físico-químicas:** reações que podem resultar na anulação de efeitos biológicos de uma ou mais drogas, ou na formação de um composto novo com uma atividade diferente da esperada inicialmente;
- b) **Farmacodinâmicas:** Acontecem quando duas ou mais drogas produzem efeitos que causam interação ou quando esta ocorre a nível de receptores farmacológicos;
- c) **Farmacocinéticas:** são as mais frequentes, e se baseiam em dados como variações no tempo do processo de absorção, distribuição, biotransformação e excreção no organismo do indivíduo.

De acordo com Hansten (1984) as condições específicas, sob as quais se dão as duas medicações interativas, são frequentemente mais importantes do que a interação das próprias drogas em si. Por isso deve ser feita uma avaliação que leve em conta fatores ligados à administração das medicações, e as condições clínicas do paciente. Ainda de acordo com o autor, a avaliação pode ser subdividida entre fatores ligados à administração medicamentosa e fatores ligados ao paciente.

3.1.1 Fatores ligados à administração medicamentosa

De acordo com Hansten (1984), os fatores ligados à administração medicamentosa são:

- a) Sequência de administração: A ordem da ingestão dos medicamentos pode influir consideravelmente no quadro clínico;
- b) Via de administração: Algumas vezes as reações acontecem apenas quando dois medicamentos são administrados via oral, enquanto outras, muitas vezes mais graves, quando apenas uma ou ambas as drogas são administradas via parenteral;
- c) Tempo de administração: Absorções gastrointestinais têm maiores probabilidades de acontecer quando o tempo de interação dos medicamentos é curto;
- d) Duração de tratamento: São raras as manifestações que ocorrem de imediato; muitas delas variam de horas a semanas;
- e) Dose dos medicamentos: Quanto maior a dose maiores as chances de a interação ser significativa;
- f) Formulação: Comprimidos e líquidos provocam interações em tempos diferentes, já que depende da forma como são liberados no organismo do paciente.

3.1.2 Fatores ligados aos pacientes

De acordo com Hansten (1984), os fatores ligados aos pacientes são:

- a) Estados mórbidos: Pacientes com diabete, hipertireoidismo, hipotireoidismo, alcoolismo e desordens gastrointestinais podem responder diferente de pacientes sem esses distúrbios;
- b) Função renal: Deficiência na função renal pode resultar no aumento da possibilidade de reações à interação de medicamentos;
- c) Função hepática: Diminuição da função hepática pode ocasionar no aparecimento de reações indesejáveis de interação;
- d) Níveis séricos de proteínas: A hipoalbuminemia aumenta a gravidade da interação de drogas que para ser eliminadas envolvem conjugação proteica;
- e) pH urinário: A ionização de drogas que são ácidos ou bases fracas podem afetar o ph urinário, o que influencia na excreção renal das mesmas;
- f) Fatores dietéticos: Alimentos afetam a absorção gastrointestinal de alguns medicamentos, fazendo com que o nível sanguíneo seja alterado e potencializando interações;
- g) Fatores ambientais: Poluição do ar e fumo podem, por exemplo, influenciar a ação de medicamentos;
- h) Fatores farmacogenéticos: Estudos realizados mostraram que as condições genéticas de um indivíduo podem influenciar na eliminação das drogas pelo organismo;
- i) Idade: Pacientes mais velhos são mais sensíveis a interações medicamentosas, pois muitos já estão acometidos a outros fatores.

Considerando a complexidade de se prescrever medicamentos, é importante a atenção farmacêutica, para que se leve em consideração o nível de polifarmácia do indivíduo, ou seja, a quantidade de medicamentos da qual uma pessoa faz o uso.

4 ATENÇÃO FARMACÊUTICA

A profissão farmacêutica, como todas as outras profissões, vem sofrendo transformações ao longo do tempo. Essas transformações foram desencadeadas pelo desenvolvimento e mecanização da indústria farmacêutica, aliada à padronização de formulações para a produção de medicamentos em larga escala e à descoberta de novos fármacos, sempre considerados de eficácia superior pela indústria farmacêutica, resultado da pesquisa farmacêutica de alta complexidade. Adiciona-se a estes fatos a evolução das formas farmacêuticas, remodelando ações terapêuticas de fármacos. Estes avanços levaram à quase obsolescência os laboratórios magistrais das farmácias, até então atividade primária do farmacêutico, definida pela sociedade e pelo âmbito profissional (FREITAS et al., 2002).

Com base nas mudanças de funções que apareceram com o passar dos anos, a atuação do farmacêutico passou a ser norteada para as ações de atenção primária em saúde, tendo o medicamento como insumo estratégico e o paciente como foco principal (MIKE et al., 1975).

Por definição do Conselho Federal de Farmácia, atenção farmacêutica é um conceito de prática profissional no qual o paciente é o principal beneficiário das ações do farmacêutico. A atenção é o compêndio das atitudes, dos comportamentos, dos compromissos, das inquietudes, dos valores éticos, das funções, dos conhecimentos, das responsabilidades e das habilidades do farmacêutico na prestação da farmacoterapia, com objetivo de alcançar resultados terapêuticos definidos na saúde e na qualidade de vida do paciente.

Um modelo de prática farmacêutica, desenvolvida no contexto da Assistência Farmacêutica. Compreende atitudes, valores éticos, comportamentos, habilidades, compromissos e co-responsabilidades na prevenção de doenças, promoção e recuperação da saúde, de forma integrada à equipe de saúde. É a interação direta do farmacêutico com o usuário, visando uma farmacoterapia racional e a obtenção de resultados definidos e mensuráveis, voltados para a melhoria da qualidade de vida. Esta interação também deve envolver as concepções dos seus sujeitos, respeitadas as suas especificidades bio-psico-sociais, sob a ótica da integralidade das ações de saúde. (CONSENSO BRASILEIRO DE ATENÇÃO FARMACÊUTICA, 2002).

4.1 PROCESSO GERAL DE ATENÇÃO FARMACÊUTICA DO PACIENTE

O processo de atenção farmacêutica obedece a uma sequência de passos conhecida como método clínico. O método clínico inclui a coleta de dados, identificação de problemas, implantação de um plano de cuidado e seguimento do paciente. (CORRER e OTUKI, 2011). Ainda segundo os autores, o método clínico da atenção farmacêutica pode ser dividido em quatro etapas diferentes:

1. Os dados do paciente são coletados e organizados, levando em consideração:
 - a) O motivo da consulta
 - b) Dados específicos do paciente;
 - c) História clínica;
 - d) História de medicação.
2. É feita a identificação de problemas relacionados à farmacoterapia, como:
 - a) Análise situacional;
 - b) Revisão da farmacoterapia;
 - c) Identificação de problemas presentes e/ou potenciais.
3. Elaboração de um plano de cuidado em conjunto com o paciente, onde:
 - a) São definidas metas terapêuticas;
 - b) São feitas intervenções;
 - c) São criados agendamentos de retorno e de continuidade do tratamento.
4. Realização do acompanhamento individual do paciente:
 - a) Análise de resultados e progressos do paciente;
 - b) Acompanhamento de novos problemas;
 - c) Alcance de metas terapêuticas.

Através desse processo de acompanhamento é possível estar mais próximo do paciente e perceber a evolução do seu caso quadro a quadro, de forma a evitar e/ou prevenir futuras enfermidades deste mesmo paciente.

5 FARMACOTERAPIA

A carência por orientações sobre a saúde e a farmacoterapia é comum e ocorre em todo o mundo. Por esta razão, a Organização Mundial da Saúde, em 1993, ressaltou que a profissão farmacêutica é mais próxima e disponível para iniciar o cuidado aos pacientes. (NETO, BALDONI, GUIDONI, 2013).

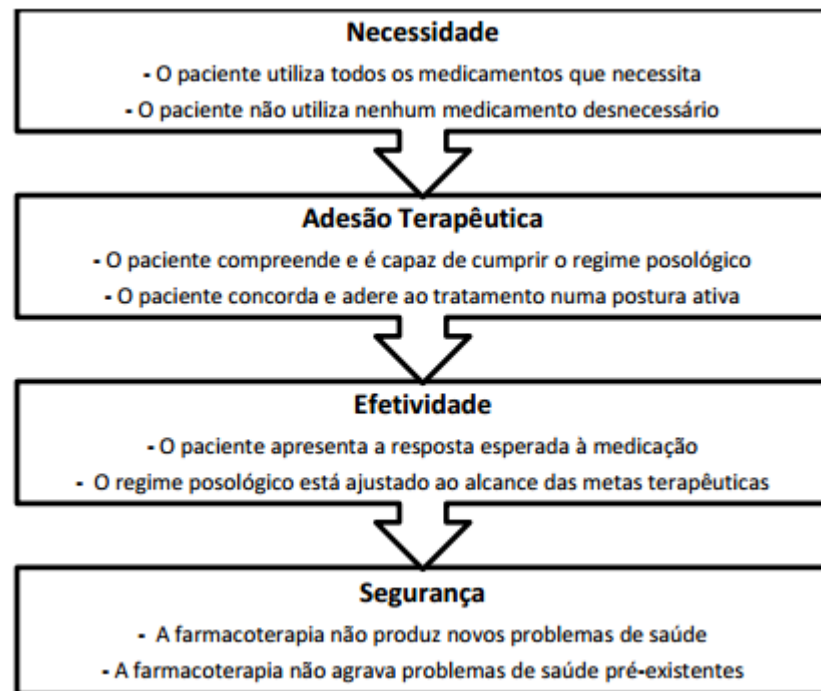
Segundo Correr e Otuki (2011) o trabalho pioneiro sobre problemas relacionados à farmacoterapia foi publicado no início dos anos 90 por STRAND et al. e segue sendo até hoje o conceito mais difundido sobre o tema. Atualmente, esses autores os definem como “qualquer evento indesejável vivenciado pelo paciente que envolva a farmacoterapia e que interfere com o alcance das metas terapêuticas do tratamento”.

Ainda segundos os autores, um problema do paciente relacionado à farmacoterapia sempre possui três componentes principais:

- a) Um evento indesejável ou o risco de um evento indesejável é vivenciado pelo paciente. O problema pode ter a forma de uma queixa clínica, sinal, sintoma, diagnóstico, doença, agravo, disfunção, exames laboratoriais alterados ou uma síndrome. Estes eventos podem ser resultado de condições fisiopatológicas, psicológicas, socioculturais ou mesmo econômicas;
- b) A farmacoterapia (produtos e/ou regime posológico) faz parte do problema;
- c) Há uma relação existente ou suspeita entre o evento e a farmacoterapia. O evento pode ser consequência da farmacoterapia, numa relação direta causa-efeito ou pode requerer a adição ou modificação da farmacoterapia para sua resolução ou prevenção.

A Figura 1 mostra quais as características ideais de uma farmacoterapia, levando em consideração tanto as necessidades do paciente quanto a segurança do processo em sua etapa final.

Figura 1 - Características de uma farmacoterapia ideal



Fonte: Correr e Otuki (2011). Baseado nos estudos de Strand et al. (1993).

De acordo com o livro Farmacoterapia: Guia terapêutico de doenças mais prevalentes, as doenças a seguir são algumas das mais apresentadas por pacientes:

- a) Hipertensão Arterial: Caracterizada por níveis elevados de pressão arterial, cujos valores altos podem alterar os níveis cardíaco, cerebral e renal. Alguns dos fatores que influenciam as causas da doença são: história familiar, dieta rica em sódio, sedentarismo, estresse, tabagismo, raça e obesidade.
O tratamento da hipertensão envolve a redução da pressão arterial do paciente através de medicamentos anti-hipertensivos, ou também através de terapia não farmacológica, como a mudança de estilo de vida (perda de peso, alimentação adequada, redução da ingestão de sal).
- b) Diabetes Mellitus: grupo de distúrbios metabólicos caracterizado por hiperglicemia crônica, podendo resultar em complicações microvasculares, macrovasculares e neuropáticas. Alguns dos fatores influenciadores da doença são dieta inadequada, sobrepeso, sedentarismo e controle glicêmico inadequado.

O tratamento farmacológico é feito através da aplicação de insulina, uma ou mais vezes por dia, e o tratamento não farmacológico deve ser feito através de dieta balanceada, prática de exercícios físicos e acompanhamento de profissionais de saúde.

c) Asma: doença crônica que obstrui as vias aéreas mais importantes. A dificuldade respiratória do paciente é o principal sintoma da doença.

A asma pode ser classificada em:

- Asma leve intermitente: com frequência inferior ou igual a duas vezes por semana;
- Asma persistente leve: sintomas com frequência superior a duas vezes por semana;
- Asma persistente moderada: acompanhada de sintomas diários e envolve o uso de inaladores diariamente;
- Asma grave persistente: os sintomas apresentados pelo paciente são praticamente contínuos.
- O tratamento é feito de forma dinâmica, podendo variar entre o controle dos sintomas até a prevenção de ataques da asma. Em casos graves, o tratamento deve ser através do uso de corticoides por inalação ou via oral.

6 INTELIGÊNCIA ARTIFICIAL

Inteligência Artificial (IA) é o nome dado a uma das áreas mais complexas e fascinantes da Ciência da Computação. A IA pode ser definida como a capacidade de uma máquina de racionar, agir, decidir, armazenar conhecimento e comunicar-se como um ser humano. (GONGORA, 2007).

A Inteligência Artificial (IA) é simplesmente uma maneira de fazer o computador pensar inteligentemente. Isto é conseguido estudando como as pessoas pensam quando estão tentando tomar decisões e resolver problemas, dividindo esses processos de pensamento em etapas básicas e desenhando um programa de computador que solucione problemas usando essas mesmas etapas. IA então fornece um método simples e estruturado de se projetar programas complexos de tomada de decisão. (LEVINE, DRANG, EDELSON, 1988).

Rich (1988) e Sato (2009), em definições mais práticas, definem a IA como sendo o estudo de como fazer com que as máquinas pensem e realizem tarefas em que serem humanos são melhores.

Em 1955, o matemático e cientista da computação americano John McCarthy foi o pioneiro no campo da inteligência artificial, criando a poderosa linguagem de programação LISP. Desde então, cientistas e engenheiros têm estudado e desenvolvido técnicas cada vez mais precisas para os mais diversos campos da ciência na qual a IA é aplicada.

De acordo com Whitson (2015), os cientistas da computação têm a inteligência artificial como meio de desenvolver programas que exibem um comportamento inteligente, que possam ser engajados em planos inteligentes como controladores de tempo das luzes dos semáforos, tradução de linguagens naturais (de um idioma para outro), agir como um especialista ao selecionar o melhor vinho para o jantar, por exemplo, e em diversas outras áreas.

No ano de 2014, que marcou 60 anos da morte de Alan Turing, pai da computação, um chatbot imitando um garoto de 13 anos, chamado Eugene Goostman, conseguiu passar no teste convencendo 33% dos juízes de que era humano. A prova aconteceu durante o Turing Test 2014, um evento anual organizado pela Universidade de Reading, na Inglaterra, em que máquinas e softwares tentam passar no teste de Turing. Por ser o primeiro software a ser aprovado, este passar a ser um marco na história da Inteligência Artificial. (ROHR, 2014).

E como tudo isso funciona? A IA funciona a partir do momento em que ela passa a imitar o processo do aprendizado humano. Para entendermos esse processo é preciso primeiramente conhecer um pouco de como funciona o processo de aprendizagem em humanos.

Começamos a aprender como utilizar os computadores para a solução de problemas, onde não temos algoritmos computáveis sistemáticos e eficientes. E agora sabemos, pelo menos numa área limitada, não só como programar computadores para executarem com êxito tais atividades de solução de problemas; sabemos também como programar os computadores para aprenderem a fazer essas coisas. (DREYFUS, 1975, pag. 29)

Nosso cérebro funciona como um sistema integrado, onde há uma rede neural que recebe informações continuamente, fazendo sua percepção sobre os fatos e então tomando uma decisão. Nesse sistema, os receptores convertem os estímulos do nosso corpo ou do ambiente a nossa volta em impulsos elétricos, que por sua vez convertem a informação para o cérebro. Esse mecanismo neural nos permite por exemplo, fazer o reconhecimento preciso de tamanhos de objetos, localização e orientação. Já a visão possibilitou que o homem, ao longo do processo evolutivo, desenvolvesse a habilidade de reconhecer padrões cada vez mais precisos. De imagens a palavras, o cérebro é capaz de fazer os mais diversos reconhecimentos de padrões a partir de um processo contínuo de treinamento e aprendizagem. Não diferente disso, dá-se o processo de aprendizado de máquinas.

6.1 INTELIGÊNCIA ARTIFICIAL X INTELIGÊNCIA HUMANA

Mas por que automatizar a inteligência humana?

Todos sabemos que a mente humana possui um amplo estoque de conhecimentos relacionados a uma incontável lista de objetos e ideias. Nossa sobrevivência depende de nossa habilidade em aplicar esses conhecimentos em qualquer situação que apareça e aprender continuamente com as novas experiências, para que sejamos capazes de responder a situações similares no futuro. (LEVINE, DRANG, EDELSON, 1988).

Durante a vida armazenamos em nossos cérebros infinitas regras que acabamos por utilizar todos os dias. Por exemplo, sabemos que um ferro de passar roupas quando ligado na tomada está quente; a partir desse fato podemos gerar a

seguinte regra: SE eu puser a minha mão em um ferro de passar ligado na tomada, ENTÃO eu vou me queimar. Mas com todo esse conjunto de fatos e regras, nossos cérebros acabam por ter uma vasta quantidade de informações que devem ser analisadas cuidadosamente para que a melhor decisão seja tomada, e para que ele não fique sobrecarregado de informações existe um sistema que seleciona a resposta mais adequada a uma situação específica, chamado de poda (pruning).

O processo de poda elimina os caminhos de pensamento que não são relevantes para o objetivo imediato de se alcançar uma meta. Assim como se corta os galhos mais baixos de uma árvore para que ela cresça com mais força, o mecanismo de poda em nossos cérebros nos afasta de qualquer fato e regra que não nos leve a nosso objetivo. Quando a mente se confronta com uma situação, o mecanismo de poda guia os processos de pensamento focalizando apenas aquelas regras relativas à solução do problema imediato. (LEVINE, DRANG, EDELSON, 1988).

Diante de todos esses fatos, criar sistemas inteligentes que armazenem uma quantidade enorme de informações sobre diferentes temas e ainda tomem decisões certas, faz com que sua utilização seja de grande valia para poupar nós, seres humanos, de processos que requerem muito tempo e energia, entretanto essa é uma decisão que ainda divide opiniões.

Desde muito tempo, os processos intelectivos foram estudados, tendo sempre sido uma curiosidade do ser humano. Com a Revolução Industrial, o advento da tecnologia e o avanço dos processos automáticos, vários estudos se fizeram presentes ao cotidiano dos que buscavam ampliar as estratégias de redução de problemas de controle de sistemas que apresentavam características não lineares, assim como as perspectivas de estruturas de máquinas inteligentes que pudessem substituir o homem em várias atividades. [...] os computadores baseados na máquina de Von Neumann deram ao mundo novas perspectivas. Mas foi o desenvolvimento das técnicas que imitam o comportamento do cérebro humano que fez muitos destes objetivos serem atingidos. (LUDWIG e COSTA, 2007).

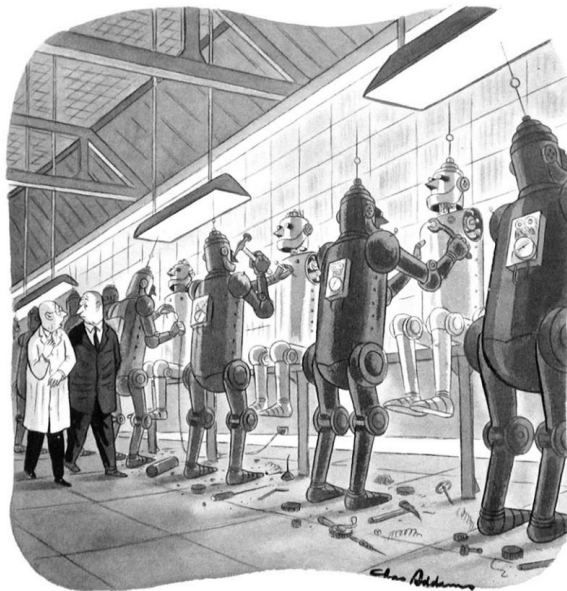
Muitos pesquisadores de Inteligência Artificial preferem trabalhar em projetos que visam construir ferramentas para ajudar o ser humano em tarefas difíceis, e não substituí-los em seus trabalhos, pois habilidades como visão, locomoção e manipulação de objetos, ainda são típicas de humanos. No entanto, outros pesquisadores entendem também que essas mesmas habilidades são pré-requisitos

para o conhecimento, o que poderia ser utilizado para treinar máquinas cada vez mais inteligentes.

De acordo com artigo publicado por Nis J. Nilsson, na *AI Magazine*, “humanos que desempenham diversos tipos de trabalho não nasceram com as habilidades que seus trabalhos requerem. Eles têm que aprendê-las, talvez através de cursos ou treinamentos no oferecidos pelo próprio emprego”. Tomando esta declaração como base, podemos utilizar de exemplo uma empresa que tenha dificuldades de encontrar funcionários aptos para determinados tipos de funções; esta mesma empresa poderia se beneficiar do uso de máquinas inteligentes, diminuindo os gastos com treinamentos de funcionários, contratações e tempo, e ainda otimizar diversos processos da indústria.

Na Figura 2 vemos a charge publicada no jornal *The New Yorker*, no ano de 1946, onde o autor preocupado com a ideia então nova de se automatizar a inteligência humana, se pergunta aonde todo esse processo de IA vai nos levar no futuro.

Figura 2 - O progresso da Inteligência Artificial



Fonte: Addams (1946).

De acordo com Levine, Drang, e Edelson (1988), todos os elementos presentes no processo de tomada de decisão do homem, como objetivos, fatos, regras e podas, devem ser reunidos em um programa de computador para que este possa ser realmente classificado como um programa inteligente, que possui IA. Partindo desse

princípio, o desenvolvimento de um sistema de IA baseado em regras é feito para permitir que processos de buscas feitas por usuários sejam otimizados.

6.2 SISTEMAS ESPECIALISTAS

Dentre as áreas de pesquisa da IA, encontram-se os Sistemas Especialistas, os quais atuam na forma de sistemas interativos, respondendo questões, solicitando ou fornecendo esclarecimentos e recomendações, podendo também auxiliar o usuário na tomada de decisões. De modo geral, podem simular o raciocínio humano ao fazer inferências, julgamentos ou projetar resultados (LINARES, 1997).

Sistemas especialistas são baseados em conhecimento e seu principal desafio é fazer com que máquinas consigam utilizar todo o conhecimento obtido através de treinamentos, da mesma forma como os humanos colocam em prática seus conhecimentos especializados em diversas áreas, para não só analisar um caso específico como também chegar a conclusões sobre ele.

Um sistema de inteligência artificial criado para resolver problemas em um determinado domínio é chamado de sistema especialista. Todo o conhecimento em um sistema especialista é fornecido por pessoas que são especialistas naquele domínio. (LEVINE, DRANG, EDELSON, 1988).

Visto que decisões especializadas são frutos do pensamento humano, que compara fatos levantados e localiza certa ligação entre os demais fatos que possuam situações idênticas, o sistema especialista também deve ter funcionamento embasado em experiências passadas para fornecer respostas especializadas ao usuário. (BARRETO e PREZOTO, 2010).

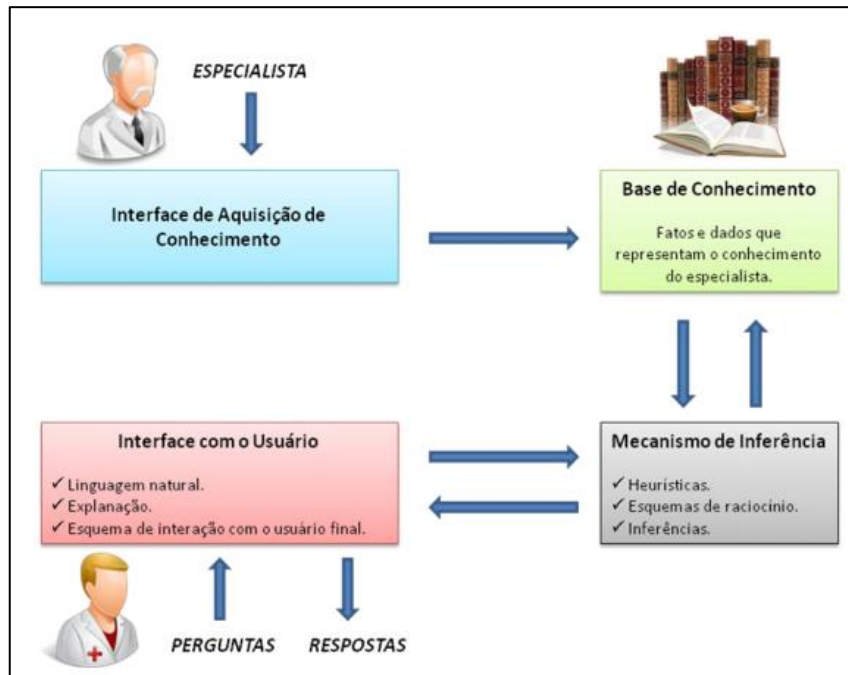
6.2.1 Componentes de um Sistema Especialista

De acordo com Cuet e Hirabara, 2010, de maneira geral, um Sistema Especialista é formado por quatro elementos básicos, como mostra a Figura 3:

- a) Base de Conhecimento: local onde fatos e regras estão armazenados;
- b) Interface de Aquisição: modifica ou adiciona conhecimento novo à base.
- c) Mecanismo de Inferência: parte do programa que vai interagir com o usuário a partir do momento que ele realizar uma consulta, acessando a base de conhecimento para fazer inferências;

- d) Interface com o usuário: linguagem natural, explicação. É acionado toda vez que o usuário solicita uma explicação sobre uma decisão particular tomada pelo sistema.

Figura 3 - Esquema clássico de um Sistema Especialista



Fonte: SOUZA e TALON [200?]. Adaptado de Mendes (1997)

6.2.2 Mecanismos de Aprendizagem

A especialização deste sistema se dá pelo “aprendizado” de novos conceitos e novas resoluções para problemas já apresentados anteriormente. Obviamente, uma máquina não aprende simplesmente o que fazer, pois lhe falta o senso, como descrito anteriormente, mas esta máquina poderá armazenar em sua base de dados, possíveis dados que possuam relação direta com uma determinada resposta que o usuário possa definir como sendo satisfatória. (BARRETO e PREZOTO, 2010).

Segundo Nilsson (1982) controlar as ações físicas de um robô não requer muita inteligência; no entanto, a mesma tarefa desempenhada por humanos, de forma praticamente inconsciente, ao ser realizada por uma máquina demanda a solução de um número maior de problemas intelectuais. A seguir, apresenta-se como funcionam muitos dos sistemas especialistas.

Muitos sistemas especialistas trabalham a técnica de Inteligência Artificial baseada na dedução de regras. Nesses sistemas, o conhecimento especializado é representado como um grande conjunto de regras simples, e essas regras são usadas para guiar o diálogo entre o sistema e o usuário, e para deduzir conclusões. (NILSSON, 1982, p. 4, tradução nossa).

6.2.2.1 *Quadro Negro*

O quadro-negro é um lugar dentro da memória do computador no qual as informações armazenadas em um sistema especialista são "afixadas" para que qualquer outro sistema especialista possa usá-lo se precisar das informações lá contidas para alcançar seus objetivos. (FÁVERO e SANTO, 2005).

As informações são mostradas aos leitores desses quadros, permitindo que a qualquer momento eles as alterem; assim, novos leitores podem compartilhar informações no mesmo quadro.

6.2.2.2 *Mecanismo De Inferência*

Mecanismo compartilhado entre sistemas especialistas para busca de dados na base de conhecimento. Tendo a premissa de que as bases de conhecimento são padronizadas e que os mecanismos de inferência sigam um padrão de busca, o mecanismo de inferência irá encontrar a melhor solução dentre um conjunto racional pré-determinado na base de conhecimento, chegando a resposta mais próxima do ideal esperado pelo usuário. (BARRETO e PREZOTO, 2010).

A Figura 3 mostra as tarefas básicas de um mecanismo de inferência. Após os passos nela descritos, o mecanismo de inferência faz a fixação dos dados mostrados no quadro-negro, fazendo com que as informações sejam empilhadas nele, ou seja, as regras mais novas são avaliadas antes das regras anteriores.

Segundo Barreto e Prezoto (2010) a regra é avaliada enquanto ter contexto verdadeiro, caso contrário, a regra é eliminada do quadro-negro, a meta estabelecida é desempilhada e uma nova regra para a meta estabelecida é criada, num processo recíproco até a localização da solução esperada. Logo, definiremos mecanismo de inferência o processo de avaliação das possíveis soluções das sub-tarefas de uma possível resposta de um problema. Este mecanismo também nos auxilia a localizar

erros nos fundamentos e objetivos lógicos, de forma a aprimorar nossas regras de localizar soluções a problemas.

Figura 4 - Regras básicas de um mecanismo de inferência



Fonte: (FÁVERO e SANTO, 2005)

6.2.3 Vantagens de um Sistema Especialista

Um sistema especialista apresenta, baseando-se em informações de (FÁVERO e SANTO, 2005), vantagens como as que seguem:

- Velocidade na determinação dos problemas;
- Segurança;
- Integração de ferramentas;
- A base de conhecimento está disponível a qualquer hora, dia e local;
- Resposta rápida para a solução de um problema específico;
- Sistemas especialistas são racionais, o que contribui para que as decisões tomadas sempre beneficiem o usuário;
- Um sistema inteligente pode ser construído com a finalidade de treinar novos usuários em diferentes áreas;

- h) Diminui o tempo de produção, e em contrapartida, aumenta os resultados e a qualidade;
- i) Ajuda na tomada de decisões de profissionais;
- j) Em alguns casos o uso de um sistema especialistas pode ser mais barato e preciso do que o serviço de um humano especialista.

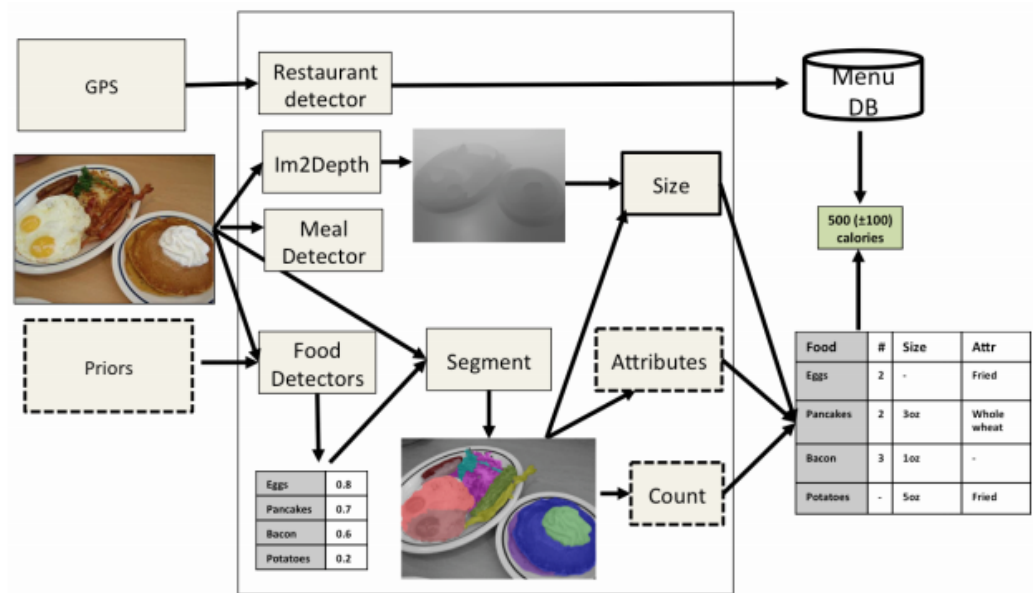
6.2.4 Exemplos de aplicação

Diferentes softwares e ferramentas fazem a utilização de sistemas especialistas em suas aplicações. Algumas aparecem a seguir:

- a) Google Cloud Platform: usada por Snapchat, Khan Academy, dentre outras companhias e organizações, esta plataforma do Google utiliza sistemas especialistas para oferecer serviços baseados em nuvem que permite ao cliente criar desde um simples website até aplicações mais complexas. De acordo com o site Google Cloud Platform são oferecidos serviços como Big Data, Machine Learning, Armazenamento de dados, Segurança, Ferramentas de Administração (login, monitoramento), Ferramentas de Desenvolvimento, dentre outras.
- b) Im2Calories: Sistema que pode reconhecer tudo o que está presente em sua refeição a partir de uma única imagem, assumindo que o usuário está comendo em um restaurante em que o menu é conhecido. Desta forma, é possível analisar as imagens e então descobrir as composições nutricionais correspondente de cada alimento.

Na Figura 5 temos uma visão geral do sistema abordado, onde as imagens de alimentos são os dados de entrada no sistema (input), e tem como saída (output) o então “diário da comida”, que é um contador que estima o total de calorias consumidos durante a refeição.

Figura 5 - Visão geral do sistema



Fonte: Meyers et al. (2015).

- c) Smartflow Traffic Management System: utilizado pela Administração Federal de Aviação (em inglês Federal Aviation Administration - FAA) dos Estados Unidos, como forma de melhor coordenar as atividades do tráfego aéreo.
- d) Hospitais: utilizam sistemas especialistas para interpretar diagnósticos de pacientes e também para o identificar interações medicamentosas e outros problemas relacionados à medicamentos.

6.2.5 Custos

Os custos de um sistema especialista variam de acordo com o tipo de sistema desenvolvido, e também devem ser levados em conta os valores que serão gastos pela companhia para o treinamento das pessoas que utilizarão o sistema, bem como sua manutenção ao longo dos anos.

6.3 SISTEMAS DE PRODUÇÃO

De acordo com Grzymala-Busse (1991), um sistema de produção consiste de uma base de conhecimento contendo regras de produção; uma base de dados

contendo fatos; e um interpretador de regras, também chamado de módulo aplicador de regras, para controlar por inteiro o sistema de produção.

Sistemas que exigem base de conhecimento atualizada muitas vezes encontram tarefas de difícil decisão, pois têm que analisar problemas que não podem ser expressos formalmente por matemática, estatística ou lógica. Desta forma, fazer o reconhecimento de padrões de espaço ou tempo torna-se um problema que em muitas das vezes não são solucionados. No entanto, sistemas adaptativos, que imitam a organização de um cérebro, possuem a capacidade de aprendizado; eles armazenam uma quantidade de fatos e regras que podem decifrar incógnitas até então só solucionadas por humanos.

7 ENGENHARIA DE SOFTWARE

Engenharia de Software é metodologia para desenvolvimento de soluções em software, ou seja, roteiro que pode utilizar diversas técnicas. A sequência de passos preestabelecidos permite optar e variar de técnicas e ferramentas nas suas diversas fases. (REZENDE, 2005).

Hoje, o software assume um duplo papel. Ele é um produto e, ao mesmo tempo, o veículo para distribuir um produto. [...] Independente de residir em um celular ou operar dentro de um mainframe, software é um transformador de informações – produzindo, gerenciando, adquirindo, modificando, exibindo ou transmitindo informações que poder ser tão simples quanto um único bit ou tão complexas quanto uma apresentação multimídia derivada de dados obtidos de dezenas de fontes independentes. (PRESSMAN, 2011).

Segundo Pressman (2011), os engenheiros de software encontram hoje sete diferentes categorias de software:

- a) Software de sistema: conjunto de programas feitos para atender a outros programas, como compiladores, editores e utilitários para gerenciamento de arquivos;
- b) Software de aplicação: programas que solucionam uma necessidade específica de negócio, como o controle de processos de fabricação em tempo real;
- c) Software científico/de engenharia: programas que necessitam processar uma quantidade enorme de números, como analisar a dinâmica orbital de ônibus espaciais;
- d) Software embutido: produto ou sistema que é utilizado para implementar e controlar características e funções para o usuário final e para o próprio sistema. Como exemplo, podemos citar funções digitais de automóveis e sistemas de freios.
- e) Software para linha de produtos: Utilizado por diferentes tipos de público. Esses softwares podem ser usados, dentre outras finalidades, para controle de estoques, ou planilhas eletrônicas;
- f) Aplicações para Web: Conjunto de arquivos hipertextos interconectados, oferecendo ao usuário, que se evoluiu a partir do aparecimento da Web 2.0.

- g) Software de inteligência artificial: utiliza-se de algoritmos números para solucionar problemas complexos, em áreas como robótica, sistemas especialistas, reconhecimento de padrões, redes neurais, etc.

Para o desenvolvimento do projeto proposto, será desenvolvido um software de inteligência artificial, explorando a área de sistemas especialistas para resolver um problema de interações medicamentosas.

7.1 INTERAÇÃO HUMANO-COMPUTADOR (IHC)

Humanos são limitados em suas capacidades de processar informação. Isso tem importantes implicações para o design. (DIX et al., 2003).

Ao desenvolver um software é importante que o desenvolvedor perceba as reais necessidades de um usuário. As informações que ficarão disponíveis na tela devem ser mais objetivas possível. Segundo DIX et al. (2003) a emoção influencia as capacidades humanas, e usuários compartilham de algumas capacitações comuns, mas são indivíduos com diferenças, as quais não podem ser ignoradas.

Desenvolver uma interface para usuário onde o modelo do programa coincide com o modelo do usuário não é fácil. Algumas vezes, seus usuários podem não ter uma ideia concreta de como o programa funciona e o que ele supostamente deve fazer. Neste caso, você vai ter que encontrar modos de dar para o usuário pistas de como as coisas funcionam. Com interface gráfica, uma maneira comum de resolver este problema é com metáforas. Mas nem todas as metáforas são criadas igualmente, e é importante entender como as metáforas funcionam para que você saiba se tem uma boa metáfora. A metáfora mais famosa é a "metáfora do desktop" utilizada no Windows e no Macintosh. [...] você pode arrastar um arquivo de uma pasta para outra para movê-lo. Para explicar como está metáfora funciona, é porque as pequenas figuras de pastas realmente lembram as pessoas de pastas, que faz com que elas percebam que podem colocar documentos dentro delas. (SPOLSKY, 2000).

7.1.1 IHC e Engenharia de Software

Segundo Dix et al. (2003), a engenharia de software apresenta uma forma de entendimento da estrutura do processo de design, promovendo o uso de critérios explícitos para julgar o sucesso de um produto em termos de usabilidade. Alguns processos de design são interativos, permitindo que o trabalho seja acompanhado por

consumidores que dão sugestões aos desenvolvedores, impactando nas decisões que vão afetar a usabilidade de um produto.

Ainda segundo os autores, os usuários têm diferentes requisições para suporte em diferentes momentos. O suporte ao usuário deve ser disponível, mas não obstrutivo, preciso e robusto, consistente e flexível; pode chegar através de comandos baseados em métodos, tutoriais, ajuda adaptativa, dentre outros.

O principal produto é um bom software capaz de satisfazer às necessidades de seus usuários e respectivos negócios. Tudo o mais é secundário. (BOOCH, RUMBAUGH, JACOBSON, 2005).

7.1.2 Modelagem de software

Projetos de softwares bem ou malsucedidos está diretamente ligado à modelagem de software.

A modelagem é uma parte central de todas as atividades que levam à implementação de um bom software. Construimos modelos para comunicar a estrutura e o comportamento desejados do sistema. Construimos modelos para visualizar e controlar a arquitetura do sistema. Construimos modelos para compreender melhor o sistema que estamos elaborando, muitas vezes expõe oportunidades de simplificação e reaproveitamento. Construimos modelos para gerenciar riscos. (BOOCH, RUMBAUGH, JACOBSON, 2005).

Segundo Booch, Rumbaugh e Jacobson (2005) através da modelagem é possível alcançar quatro objetivos:

- a) Os modelos ajudam a visualizar o sistema como ele é ou como desejamos que seja;
- b) Os modelos permitem especificar a estrutura ou o comportamento de um sistema;
- c) Os modelos proporcionam um guia para a construção do sistema;
- d) Os modelos documentam as decisões tomadas.

7.1.3 Linguagem UML

A UML - Linguagem de Modelar Unificada, é uma linguagem gráfica utilizada para visualizar, especificar, construir e documentar artefatos de sistemas complexos de software. (BOOCH, RUMBAUGH, JACOBSON, 2005).

Uma linguagem de modelagem, como a UML, é uma linguagem-padrão para a elaboração da estrutura de projetos de software. [...] A modelagem permite a compreensão de um sistema. Nenhum modelo é inteiramente suficiente. Sempre serão necessários vários modelos, conectados entre si, para tornar possível entender qualquer aspecto, ainda que seja o sistema mais trivial. No caso de sistemas que fazem uso intenso de software, torna-se necessária uma linguagem capaz de abranger as diferentes visões relacionadas à arquitetura do sistema, como essa arquitetura evolui ao longo do ciclo de vida de desenvolvimento do software. (BOOCH, RUMBAUGH, JACOBSON, 2005).

A UML pode ser utilizada na construção de sistemas simples ou complexos, como sistemas de informações corporativos, serviços bancários e financeiros, transportes, telecomunicações, defesa/espço aéreo, dentre outros. Ela está inserida na parte do desenvolvimento voltado à modelagem do software, permitindo a interação do ser humano com a interface do programa, criando modelo de dados e também funções que serão desempenhadas pelo software, bem como as telas que interligam os módulos. (BOOCH, RUMBAUGH, JACOBSON, 2005).

7.1.4 Diagramas UML

Diversos diagramas são utilizados em uma UML:

Um diagrama é a representação gráfica de um conjunto de elementos, geralmente representadas como gráficos de vértices (itens) e arcos (relacionamentos). [...] Em todos os sistemas, com exceção dos mais triviais, um diagrama representa uma visão parcial dos elementos que compõem o sistema. (BOOCH, RUMBAUGH, JACOBSON, 2005).

Segundo Booch, Rumbaugh e Jacobson (2005) uma UML possui 13 diagramas, e alguns deles podem ser vistos a seguir:

- a) **Diagrama de classes:** exhibe um conjunto de classes, interfaces e colaborações, sem encontrados de forma mais frequente em sistemas de modelagem orientados a objeto;
- b) **Diagrama de casos de uso:** abrange a visão estática de casos de uso do sistema; eles são importantes para a organização e modelagem de comportamentos do sistema;

- c) **Diagrama de seqüências:** exibe uma interação, consistindo de um conjunto de objetos ou papéis, incluindo as mensagens que podem ser trocadas entre eles;
- d) **Diagrama de gráficos de estados:** exibem uma máquina de estados, formada por estados, transições, eventos e atividades;

7.2 TESTE DE SOFTWARE

Antes de iniciarmos uma discussão sobre teste de software precisamos esclarecer alguns conceitos relacionados a essa atividade. Inicialmente, precisamos conhecer a diferença entre Defeitos, Erros e Falhas. As definições que iremos usar aqui seguem a terminologia padrão para Engenharia de Software do IEEE – Institute of Electrical and Electronics Engineers – (IEEE 610, 1990 citado por Claudio ([201-])).

- a) Defeito é um ato inconsistente cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta. Por exemplo, uma instrução ou comando incorreto.
- b) Erro é uma manifestação concreta de um defeito num artefato de software. Diferença entre o valor obtido e o valor esperado, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa constitui um erro.
- c) Falha é o comportamento operacional do software diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros e alguns erros podem nunca causar uma falha.

A Figura 6 expressa a diferença entre esses conceitos.

Figura 6 - Defeito x erro x falha



Fonte: Claudio ([201-]).

Como pode ser observado na Figura 6, defeitos fazem parte do universo físico (a aplicação propriamente dita) e são causados por pessoas, por exemplo, através do mal-uso de uma tecnologia, podendo ocasionar a manifestação de erros em um produto, ou seja, a construção de um software de forma diferente ao que foi especificado (universo de informação). Por fim, os erros geram falhas, que são comportamentos inesperados em um software, que afetam diretamente o usuário final da aplicação (universo do usuário), e podem inviabilizar a utilização de um software.

Segundo Claudio ([201-]), teste de software é o processo de execução de um produto para determinar se ele atingiu suas especificações e funcionou corretamente no ambiente para o qual foi projetado, ao mesmo tempo em que faz parte de todo o processo de engenharia de software. O seu objetivo é revelar falhas em um produto, para que as causas dessas falhas sejam identificadas e possam ser corrigidas pela equipe de desenvolvimento antes da entrega final. Por conta dessa característica das atividades de teste, dizemos que sua natureza é “destrutiva”, e não “construtiva”, pois visa ao aumento da confiança de um produto através da exposição de seus problemas, porém antes de sua entrega ao usuário final.

Ainda segundo o autor, o conceito de teste de software pode ser compreendido através de uma visão intuitiva ou mesmo de uma maneira formal. Existem atualmente várias definições para esse conceito. De uma forma simples, testar um software significa verificar através de uma execução controlada se o seu comportamento ocorre de acordo com o especificado. O objetivo principal desta tarefa é revelar o número máximo de falhas dispondo do mínimo de esforço, ou seja, mostrar aos que desenvolvem se os resultados estão ou não de acordo com os padrões estabelecidos.

Para Pressman et al. (2011), o processo de testes visa encontrar falhas no sistema para corrigi-las antes de distribuir o sistema, ou de atualizar o software com novos recursos, portanto o sistema deve ser projetado e implementado pensando na facilidade da realização dos testes. Por sua vez os testes devem ser feitos levando em consideração certas características para que nada seja deixado para trás.

Em conformidade com o autor, o conceito de testabilidade consiste em se medir o quão simples é o ato de testar um software. As características apresentadas seguir caracterizam um software a ser testável:

- a) Operabilidade: um sistema projetado e implementado tendo em mente a qualidade, terá poucas falhas quando os testes forem realizados.

- b) Observabilidade: quando é possível ver com clareza as entradas, saídas e variáveis do sistema fica mais fácil de detectar possíveis falhas.
- c) Controlabilidade: entradas geram saídas específicas, e para cada tipo de saída existirá um tipo de entrada específica. Se o engenheiro puder controlar essas entradas ficará mais fácil realizar os testes.
- d) Decomponibilidade: o sistema é construído a partir de módulos e pode ser testado em partes.
- e) Simplicidade: Quanto mais simples um sistema for atingindo o objetivo, mais simples serão os testes.
- f) Estabilidade: Quanto menos alterações o software tiver, menos testes precisarão ser feitos.
- g) Compreensibilidade: Quanto mais informações estiverem disponíveis para o entendimento do software, mais eficazes serão os testes, isso inclui manuais organizados, detalhados e especificados.

A atividade de teste é composta por alguns elementos essenciais que auxiliam na formalização desta atividade, os quais serão apresentados a seguir.

- a) Caso de Teste: descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado. (CRAIG; JASKIEL, 2002).
- b) Procedimento de Teste: é uma descrição dos passos necessários para executar um caso (ou um grupo de casos) de teste. (CRAIG; JASKIEL, 2002).
- c) Critério de Teste: serve para selecionar e avaliar casos de teste de forma a aumentar as possibilidades de provocar falhas ou, quando isso não ocorre, estabelecer um nível elevado de confiança na correção do produto. (ROCHA et al., 2001). Os critérios de teste podem ser utilizados como:
 - a. Critério de Cobertura dos Testes: permite a identificação de partes do programa que devem ser executadas para garantir a qualidade do software e indicar quando o mesmo foi suficientemente testado. (RAPPS; WEYUKER, 1982). Ou seja,

determinar o percentual de elementos necessários por um critério de teste que foram executados pelo conjunto de casos de teste.

- b. Critério de Adequação de Casos de Teste: Quando, a partir de um conjunto de casos de teste T qualquer, ele é utilizado para verificar se T satisfaz os requisitos de teste estabelecidos pelo critério. Ou seja, este critério avalia se os casos de teste definidos são suficientes ou não para avaliação de um produto ou uma função. (ROCHA et al., 2001).
- c. Critério de Geração de Casos de Teste: quando o critério é utilizado para gerar um conjunto de casos de teste T adequado para um produto ou função, ou seja, este critério define as regras e diretrizes para geração dos casos de teste de um produto que esteja de acordo com o critério de adequação definido anteriormente. (ROCHA et al., 2001).

7.2.1 Níveis de teste de software

O planejamento dos testes deve ocorrer em diferentes níveis e em paralelo ao desenvolvimento do software (Figura 4). Segundo Rocha et al. (2001) definimos que os principais níveis de teste de software são:

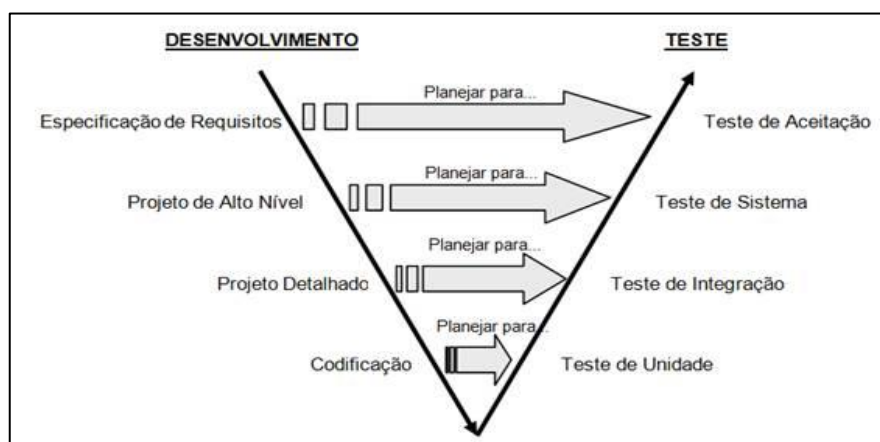
- a) Teste de Unidade: também conhecido como teste unitário. Tem por objetivo explorar a menor unidade do projeto, procurando provocar falhas ocasionadas por defeitos de lógica e de implementação em cada módulo, separadamente. O universo alvo desse tipo de teste são os métodos dos objetos ou mesmo pequenos trechos de código.
- b) Teste de Integração: visa provocar falhas associadas às interfaces entre os módulos quando esses são integrados para construir a estrutura do software que foi estabelecida na fase de projeto.
- c) Teste de Sistema: avalia o software em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final. Dessa maneira, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia de manipulação do software. Verifica se o produto satisfaz seus requisitos.

- d) Teste de Aceitação: são realizados geralmente por um restrito grupo de usuários finais do sistema. Esses simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.
- e) Teste de Regressão: Teste de regressão não corresponde a um nível de teste, mas é uma estratégia importante para redução de “efeitos colaterais”. Consiste em se aplicar, a cada nova versão do software ou a cada ciclo, todos os testes que já foram aplicados nas versões ou ciclos de teste anteriores do sistema. Pode ser aplicado em qualquer nível de teste.

Dessa forma, em conformidade com a Figura 4, o planejamento e projeto dos testes devem ocorrer de cima para baixo, ou seja:

1. Inicialmente é planejado o teste de aceitação a partir do documento de requisitos;
2. Após isso é planejado o teste de sistema a partir do projeto de alto nível do software;
3. Em seguida, ocorre o planejamento dos testes de integração a partir o projeto detalhado;
4. E por fim, o planejamento dos testes a partir da codificação.

Figura 7 - Modelo V descrevendo o paralelismo entre as atividades de desenvolvimento e teste de software



Fonte: (Craig e Jaskiel, 2002).

7.2.2 Técnicas de teste de software

Atualmente existem muitas maneiras de se testar um software. Mesmo assim, existem as técnicas que sempre foram muito utilizadas em sistemas desenvolvidos sobre linguagens estruturadas que ainda hoje têm grande valia para os sistemas orientados a objeto. Apesar dos paradigmas de desenvolvimento serem diferentes, o objetivo principal destas técnicas continua a ser o mesmo: encontrar falhas no software.

De acordo com Claudio ([201-]), as técnicas de teste são classificadas de acordo com a origem das informações utilizadas para estabelecer os requisitos de teste. Elas contemplam diferentes perspectivas do software, e impõe-se a necessidade de se estabelecer uma estratégia de teste que contemple as vantagens e os aspectos complementares dessas técnicas. As técnicas existentes são: técnica funcional e estrutural.

Segundo Lourenço (2013) todo software pode ser testado a partir de duas técnicas existentes:

- a) Teste de Caixa Branca: também conhecido como teste estrutural, é uma abordagem em que conhecendo o funcionamento interno do software (código), consegue-se garantir que todas as operações internas são executadas conforme as especificações;
- b) Teste de Caixa Preta: é um teste funcional, onde conhecendo-se a funcionalidade especificada para o software, demonstra-se que todas as funções especificadas para estão completamente operacionais enquanto simultaneamente é feita uma busca por erros nessas funções. O que difere esse teste do de caixa branca, é que esse traz a ideia de um teste conduzido na interface do software com o usuário, sem fazer a observação do código interno deste software.

Apesar de parecer que utilizando somente a abordagem de teste de caixa branca pode-se garantir software 100% livre de erros, é fácil demonstrar que isto é impraticável. Um teste de caixa branca exercita a estrutura de controle de um programa através da definição de todos os caminhos lógicos possíveis do programa, que passam pelos condicionais e pelos laços. Em uma aplicação de grande porte, o número de caminhos lógicos é extremamente alto e, portanto, o número de casos de teste para exercitar todos os caminhos seria muito grande. A execução de tantos casos de teste demandaria um enorme tempo e alto custo, o que seria impraticável. (LOURENÇO, 2013).

Segundo Lourenço (2013) tanto a técnica de caixa branca quanto a técnica de caixa preta devem ser aplicadas em uma atividade de teste de software, de forma que uma atue como complemento da outra.

Outras técnicas de teste que podem e devem ser utilizadas de acordo com necessidades de negócio ou restrições tecnológicas são: teste de desempenho, teste de usabilidade, teste de carga, teste de stress, teste de confiabilidade e teste de recuperação.

Segundo Claudio ([201-]), alguns autores chegam a definir uma técnica de teste caixa cinza, que seria um mesclado do uso das técnicas de caixa preta e caixa branca, mas, como toda execução de trabalho relacionado à atividade de teste utilizará simultaneamente mais de uma técnica de teste, é recomendável que se fixe os conceitos primários de cada técnica.

8 INFORMÁTICA APLICADA À SAÚDE

Sistemas Especialistas têm sido cada vez mais utilizados na área da saúde para ajudar no diagnóstico de pacientes.

Na Medicina, sistemas especialistas são utilizados para realização de diagnóstico em determinada área. Regras de inferência são aplicadas sobre uma grande base de conhecimento sobre sintomas e tratamentos possíveis, para que se possa identificar a doença e posteriormente oferecer um tratamento adequado. Estes sistemas são utilizados ainda como auxílio à tomada de decisão, e não como determinantes no diagnóstico definitivo. (Cuer e Hirabara, 1998).

Segundo Cuer e Hirabara (1998), esses sistemas podem ser classificados da seguinte forma:

- a) Sistemas com capacidade de decisão própria limitada ou ausente:
 - Recuperação de dados sobre pacientes;
 - Cálculos matemáticos assistido por computador;
 - Análise e interpretação primária de dados.
- b) Sistemas com capacidade de raciocínio automático e de inferência:
 - Sistemas de classificação de doenças;
 - Sistemas especialistas baseados em consulta;
 - Sistemas especialistas baseados em crítica.

O Sistema Especialista que será desenvolvido terá capacidade de raciocínio automático e inferência, pois através de consultas feitas em um banco de dados específico, ele será capaz de fazer a associação entre dois medicamentos e então dizer qual a interação medicamentosa causada.

8.1 APLICAÇÕES DESENVOLVIDAS PARA A SAÚDE

Hoje em dia há no mercado diversas aplicações que permitem ao usuário portador de um smartphone o acesso a tecnologias que vão desde o monitoramento de exercícios físicos até o de doenças crônicas. Confira a seguir alguns desses aplicativos:

- a) Docway é um aplicativo que leva o médico onde o usuário estiver! Sem filas, sem burocracia, sem esperas. Você entra no aplicativo, escolhe a especialidade e faz o agendamento. Se precisar de um atendimento de urgência, tem um médico de plantão para ajudar. (DOCWAY, 2016).
- b) DPOC oferece orientações sobre o diagnóstico, classificação e tratamento da Doença Pulmonar Obstrutiva Crônica (DPOC). As recomendações são baseadas nas diretrizes do Global Initiative for Chronic Obstructive Lung Disease (GOLD) e Caderno de Atenção Básica sobre Doenças Respiratórias Crônicas do Ministério da Saúde. (COFEN, 2016).
- c) Hydro é um aplicativo que ajuda o usuário a se lembrar e a monitorar a quantidade de água ingerida por dia. (EXAME, 2016).
- d) Fitbit auxilia no acompanhamento das atividades básicas do dia a dia. Com o aplicativo para ter uma visão completa das suas estatísticas diárias, incluindo passos, distância, calorias, sono, peso e monitoramento de exercícios físicos. (APLICATIVOS DE SAÚDE, 2016).
- e) O aplicativo Socorro permite que em casos de emergência, sua ficha médica, com informações vitais, ficará acessível as pessoas que lhe prestarem socorro. (iTUNES, 2016).

9 TRABALHOS CORRELATOS

Em breve pesquisa foram encontrados alguns trabalhos correlatos ao aplicativo de análise de interações medicamentosas que será desenvolvido:

Elie Chadarevian, mestrando EPUSP, publicou o trabalho intitulado como “Sistemas Inteligentes Aplicados à Medicina - Interações Medicamentosas” propondo um sistema especialista, baseado em regras de produção, com a finalidade de desenvolver um software responsável pelo “tratamento inteligente” dos dados, a fim de produzir respostas assertivas para os diferentes tipos de regras para a interação entre medicamentos.

O software APM (Avaliação de Prescrição de Medicamentos), desenvolvido Informática de Municípios Associados S/A (IMA), no final de novembro de 2011, tem como objetivo auxiliar os profissionais de farmácia a fazer uma análise sobre a ocorrência de casos de interações medicamentosas que incidem sobre as receitas prescritas.

Através da pesquisa realizada foi possível perceber que ainda há pouca produção e desenvolvimento de softwares que atinjam um nicho maior de profissionais da saúde no Brasil; portanto, a criação de um aplicativo que seja gratuito e permita que o usuário o execute através de um dispositivo móvel, como tablete ou celular, cria uma ferramenta de inclusão que pode ser acessada tanto por profissionais prescritores de medicamentos, quanto por pacientes, evitando que a administração de medicamentos seja feita de forma errônea e afete diretamente a saúde e bem-estar do paciente.

10 METODOLOGIA

Esse trabalho tem o propósito de pesquisas exploratórias que elucidam, através da produção de conhecimento, o caminho planejado que deverá ser traçado, e bem executado, para que o objetivo inicial da pesquisa seja atingido pelo pesquisador.

Nesse caso, objetivo do trabalho acadêmico é o desenvolvimento de uma aplicação para celulares com os sistemas operacionais Android e iOS, de forma híbrida através do framework Ionic, que atenda a estudantes de farmácia e profissionais prescritores de medicamentos, utilizando para tais fins bases de conhecimento relacionadas à área de Inteligência Artificial.

Diversas etapas foram necessárias para o desenvolvimento desse projeto, considerando que o objetivo é produzir resultados pré-estabelecidos. Neste caso devem ser feitas análises, através da API, que nos permitem dizer quando haverá ou não interação entre dois medicamentos administrados concomitantemente.

Para a obtenção das informações sobre as interações medicamentosas que podem ser encontradas na combinação de dois ou mais medicamentos, será utilizada uma API gratuita que permite acesso à uma base de dados também gratuita, com milhares de informações sobre as diferentes interações apresentadas por determinados medicamentos.

O trabalho começou com o estudo de dois segmentos diferentes: saúde e tecnologia. À fim de compreendê-los de maneira mais profunda e fazer conexões entre eles, foram visitadas literaturas em ambas as áreas. Ao longo do tempo, essas duas áreas evoluíram de forma com seus caminhos passaram a se cruzar de forma frequente, e partindo dessa premissa, foram pesquisadas formas de integrar Inteligência Artificial com a área de Farmacologia. Nesta primeira parte do trabalho foram pesquisados e lidos livros, artigos, revistas e trabalhos acadêmicos. Na área de Ciência da Computação fez-se o uso de materiais sobre Inteligência Artificial, Redes Neurais, Machine Learning e Sistemas Especialistas; já na área de Farmácia, dividiu-se a literatura entre Atenção Farmacêutica, Diretrizes Terapêuticas, Interações Medicamentosas e Farmacoterapia.

A segunda etapa do desenvolvimento desse trabalho consistiu no estudo e pesquisa sobre o público-alvo, ou seja, o usuário final do aplicativo, bem como as reais necessidades desse nicho de mercado que é voltado à prescritores de medicamentos.

A próxima etapa foi de análise para descobrir qual o melhor modelo de sistema para criação do software. Dentro da área de Inteligência Artificial chegou-se à conclusão de que os sistemas baseados em conhecimentos, ou sistemas especialistas, seriam ideais para esse trabalho em específico, pois as informações sobre os dados de entrada já são conhecidas. Como se trata de um protótipo relacionado à área médica onde se lida diretamente com a vida de muitas pessoas, os resultados desse projeto devem ser assertivos; por isso, nada melhor do que utilizar um sistema que através de uma API que permite acesso ao banco de dados com registro de milhares de interações entre medicamentos reproduza os resultados esperados sem cometer erros, agindo não como um sistema lógico, mas sim pensando como o homem.

O próximo passo foi a definição de um banco de dados confiável que tenha um número grande de medicamentos cadastrados; desta forma, a implementação de um banco de dados que possa abranger o maior número de informações possíveis permitirá ao algoritmo de buscas trazer os melhores resultados para possíveis interações medicamentosas.

A definição dos softwares que foram utilizados e o estudo de suas respectivas ferramentas foi a etapa seguinte do trabalho. Através deles foram escritos os códigos e desenvolvidas as telas que permitem a interação dos usuários. A seguir, alguns dos softwares já definidos.

10.1 SOFTWARES E TECNOLOGIAS UTILIZADOS

10.1.1 MODELAGEM DO BANCO

Devido à grande quantidade de medicamentos disponível, será feito um recorte no número de dados para que seja feita, num primeiro momento, a interação seja mostrada no sistema especialista.

Além de fazer deduções e chegar a conclusões, um Sistema Especialista deve ter a capacidade de expor cada passo de suas inferências, sendo esta uma facilidade obrigatória no desenvolvimento da confiabilidade atribuída ao sistema e, no caso da análise de interações medicamentosas [...] comparativamente a um Sistema Convencional, baseado em algoritmos que processam um volume de dados de maneira repetitiva e emitem um resultado final, um Sistema Especialista baseia-se em uma busca heurística e normalmente trabalha com questões para as quais não existe uma solução

organizada de forma algorítmica convencional. (CHADAREVIAN, 2004).

À princípio serão utilizados dados sobre medicamentos disponíveis na plataforma do RXNorm, que é a entidade Biblioteca Nacional de Medicina dos Estados Unidos, além de inferências baseadas nas bulas dos medicamentos, que permitem a análise bioquímica das drogas, e o levantamento de dados como nome do medicamento, indicação, classe, subclasse, farmacodinâmica, classe, subclasse, efeitos colaterais e interações.

Os dados serão extraídos de uma API, o que é excelente para o caso de aplicações menores, permitindo que com acesso à internet o usuário faça requisições cliente-servidor.

10.1.2 FERRAMENTAS PARA O DESENVOLVIMENTO

A seguir serão apresentadas algumas das tecnologias utilizadas para o desenvolvimento do projeto.

10.1.2.1 ANDROID

O sistema operacional Android foi desenvolvido pela Android, Inc., comprada pelo Google em 2005. Em dezembro de 2010, mais de 300.000 smartphones com Android eram ativados a cada dia. (DEITEL, 2012).

Aplicações que são executadas em um celular podem estar literalmente conectadas e online, sincronizando informações diretamente de um servidor confiável da empresa. Hoje, utilizamos esses tipos de serviços para pagar nossas contas em bancos, realizar compras com o celular, dentre outras facilidades. O que todos precisavam na verdade era de uma plataforma poderosa e flexível para tomar tudo isso mais viável e cada vez uma realidade para todos. (LECHETA, 2014).

Existem muitas plataformas móveis no mercado hoje, mas de acordo com BURNETTE (2009) o que faz o sistema operacional Android especial, é a combinação de fatores como:

- a) Plataforma baseada em Linux e com código aberto, permitindo ao desenvolvedor acesso total sem que ele tenha que pagar para utilizá-la;

- b) Arquitetura baseada em componentes, onde partes de uma aplicação podem ser reaproveitadas por desenvolvedores em outros projetos;
- c) Customização da experiência do usuário através de serviços baseados na localização utilizando GPS;
- d) A forma com a qual o Android oferece uma maneira inovadora de como as aplicações para dispositivos móveis interface com os usuários.

O desenvolvimento do projeto deste Trabalho de Conclusão de Curso será desenvolvido para dispositivos móveis que fazem uso desse sistema operacional, levando em consideração o rico material disponível sobre a ferramenta em livros e sites, além do acesso à códigos abertos para desenvolvedores.

10.1.2.2 JAVA

Java é a linguagem de programação mais utilizada mundo afora. (DEITEL, 2012).

Lançada em 1995 pela Sun Microsystems, essa linguagem permite que o usuário jogue online, converse em chats com pessoas do mundo todo, visualize imagens em 3D e muito mais. (ORACLE, 2016).

Java é orientada a objetos e permite que programadores desenvolvam para diferentes tipos de plataformas, de acordo com as suas necessidades.

Com o advento da Programação Orientada a Objetos houve um salto na organização dos sistemas. O mundo passou a ser modelado na forma de classes e objetos, e como tal, as classes de objetos puderam ser reutilizadas com maior facilidade. (RESENDE, 2005).

10.1.2.3 IOS

IOS é o sistema operacional móvel lançado em pela Apple em 2007, juntamente com o primeiro iPhone. (GIZMODO, 2016). Anualmente são criadas atualizações para reduzir os erros e melhorar o processamento dos produtos que utilizam o sistema operacional.

Aplicativos nativos para iOS podem ser desenvolvidos através das linguagens Objective-C ou Swift.

10.1.2.4 APLICAÇÕES MÓVEIS HÍBRIDAS

Aplicações baseadas em web e que utilizam HTML5 para se adaptarem aos diferentes tamanhos de telas disponíveis em aparelhos móveis. Uma ferramenta que permite utilizar essa tecnologia é o Ionic.

O Ionic é framework de código aberto, que nos permite desenvolver aplicações móveis híbridas utilizando HTML5, CSS, JavaScript, dentre outras linguagens de programação. A companhia de mesmo nome foi fundada em 2012, e tem a missão de prover ferramentas de código aberto e serviços que empoderam pessoas por todo o mundo participando do rápido crescimento econômico voltado à aplicações móveis (IONIC, 2017).

10.1.2.5 API

Para fazer integração com o framework Ionic, foi utilizada uma API que permite utilizar os dados de milhares de informações sobre a interação de dois ou mais medicamento.

API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseada na Web. A sigla API refere-se ao termo em inglês Application Programming Interface que significa em português “Interface de Programação de Aplicativos. (CANALTECH, 2017).

Uma API pode ser disponibilizada por uma empresa quando há a intenção de que outros desenvolvedores criem softwares ou aplicações associadas aos seus serviços. Através de uma API os aplicativos podem se comunicar uns com os outros sem que os usuários percebam a interação por trás da interface.

Para detectar as possíveis interações medicamentosas, será utilizada nesse trabalho a “Drug Interaction API”, disponível no site do RXNAV.

Antes de chegar à última etapa, foram realizados testes pelo desenvolvedor e também o aprimoramento das funcionalidades do sistema.

A última etapa foi a disponibilização e implementação do sistema em sua versão final para a avaliação prática do usuário, verificando se as necessidades foram atendidas e se o aplicativo realmente cumpriu seu objetivo inicial.

11 DESENVOLVIMENTO

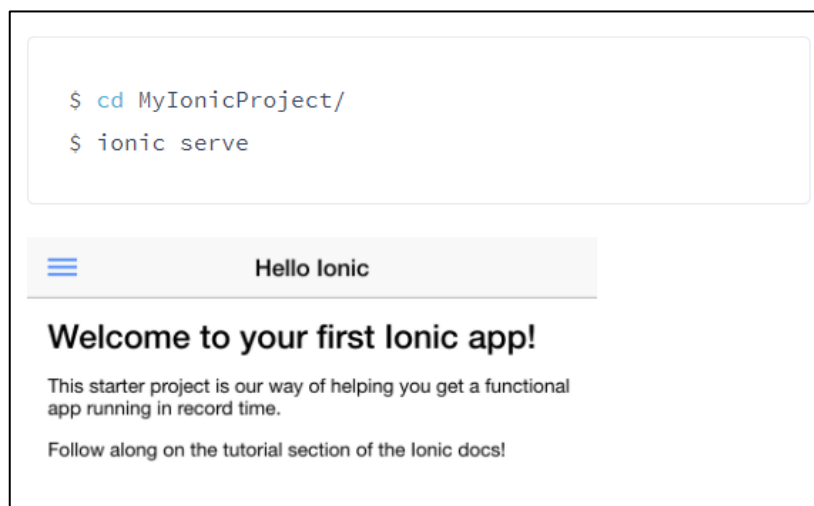
O desenvolvimento teve início com a escolha do template do framework Ionic para fazer a aplicação híbrida.

Para criar projetos no Ionic, foram necessárias as instalações do Cordova e da versão mais recente do Node.js. Depois de seguir o passo a passo da instalação apresentada no site do framework Ionic, através do terminal do Windows demos início a criação do projeto, escolhendo entre os templates oferecidos pela ferramenta, como:

- tabs (layout simples com 3 abas);
- sidemenu (layout que permite navegar entre o menu e qualquer uma das páginas);
- blank (página única).

A visualização do aplicativo pode ser feita de forma local. Na Figura 8, são mostrados os comandos que são colocados no terminal, e o resultado da aplicação desenvolvida mostrada no navegador.

Figura 8 - Início da aplicação desenvolvida em Ionic



Fonte: Ionic (2017).

A programação para as telas da aplicação foi feita utilizando as linguagens HTML5, CSS e JavaScript. Na imagem que segue é possível ver a estrutura das pastas onde são inseridos os códigos de programação:

Figura 9 - Exemplo de código desenvolvido

```

1 import { Component } from '@angular/core';
2 import { IonicPage, NavController } from 'ionic-angular';
3
4 /**
5  * The Welcome Page is a splash page that quickly describes the app,
6  * and then directs the user to create an account or log in.
7  * If you'd like to immediately put the user onto a login/signup page,
8  * we recommend not using the Welcome page.
9  */
10 @IonicPage()
11 @Component({
12   selector: 'page-welcome',
13   templateUrl: 'welcome.html'
14 })
15 export class WelcomePage {
16
17   constructor(public navCtrl: NavController) { }
18
19   login() {
20     this.navCtrl.push('LoginPage');
21   }
22
23   signup() {
24     this.navCtrl.push('SignupPage');
25   }
26 }
27

```

Fonte: Print da tela com o código welcome.ts, no editor de textos Sublime. (2017)

Para mostrar a interação dos medicamentos no Sistema Especialista, foi utilizada uma API que traz dados de diversas fontes sobre a interação medicamentosa.

A RXNorm RESTful webAPI é um serviço web desenvolvido na Biblioteca Nacional de Medicina (USA) através do acesso aos dados RXNorm. Representational state transfer (REST) é um estilo de arquitetura de software para sistemas de hipermídias distribuídas, como a World Wide Web. O estilo REST de arquiteturas consistem de clientes e servidores. Os clientes iniciam a requisição aos servidores; servidores processam as requisições e retornam as respostas apropriadas. (RXNAV, 2017, tradução nossa).

À princípio foram escolhidos cinco medicamentos populares para comprovar o risco de interação que pode ocorrer entre eles. São estes:

- **Ácido acetilsalicílico:** Segundo bula disponibilizada pela farmacêutica EMS no site da Anvisa, este medicamento contém a substância ativa ácido acetilsalicílico, que pertence ao grupo de substâncias inflamatórias não-esteróides, com propriedades anti-inflamatória (atua na inflamação), analgésica (atua na dor) e antitérmica (atua na febre);
- **Omeprazol:** De acordo com a bula disponibilizada no site da Anvisa, essa substância ativa atua na diminuição de quantidade de ácido produzida no estômago.

- Alprazolam: esse medicamento é indicado no tratamento de ansiedade;
- Captopril: medicamento indicado no tratamento de hipertensão;
- Ibuprofeno: indicado para febres e dores leves e moderadas.

A API disponibilizada pelo U.S. National Library of Medicine (NLM) permite que através do método REST sejam inseridos dois valores que serão buscados na base de conhecimento.

O layout foi criado para que fosse intuitivo ao usuário. Desta forma, foi utilizado um template baseado em abas, que permite a melhor organização de informação e fácil troca de páginas durante a utilização do aplicativo. Assim que iniciado, o aplicativo chamado InterMed, traz uma apresentação em formato de slides sobre as suas principais funcionalidades. Logo após, o usuário é direcionado à página que traz duas diferentes abas:

- Itens: página que permite ao usuário fazer o cadastro de medicamentos, no formato de lista, que contenham informações interessantes a ele, contendo nome, descrição e foto;
- Busca: página onde será mostrada a interação de medicamentos. O usuário pode selecionar dois diferentes medicamentos nas listas apresentadas e checar se há a existência de interação entre eles.

O código que segue, referente à página search.html, mostra um exemplo da estrutura de código das páginas da aplicação, utilizando comandos de HTML5 juntamente com tags do framework Ionic:

```
<ion-header>
  <ion-navbar>
    <ion-title>{{ 'SEARCH_TITLE' | translate }}</ion-title>
  </ion-navbar>
</ion-header>
<ion-content>
  <ion-item>
    <ion-label>Primeiro Medicamento</ion-label>
    <ion-select [(ngModel)]="medicamento1">
      <ion-option value="91101">AAS</ion-option>
      <ion-option value="7646">Omeprazol</ion-option>
      <ion-option value="596">Alprazolam</ion-option>
    </ion-select>
  </ion-item>
</ion-content>
```

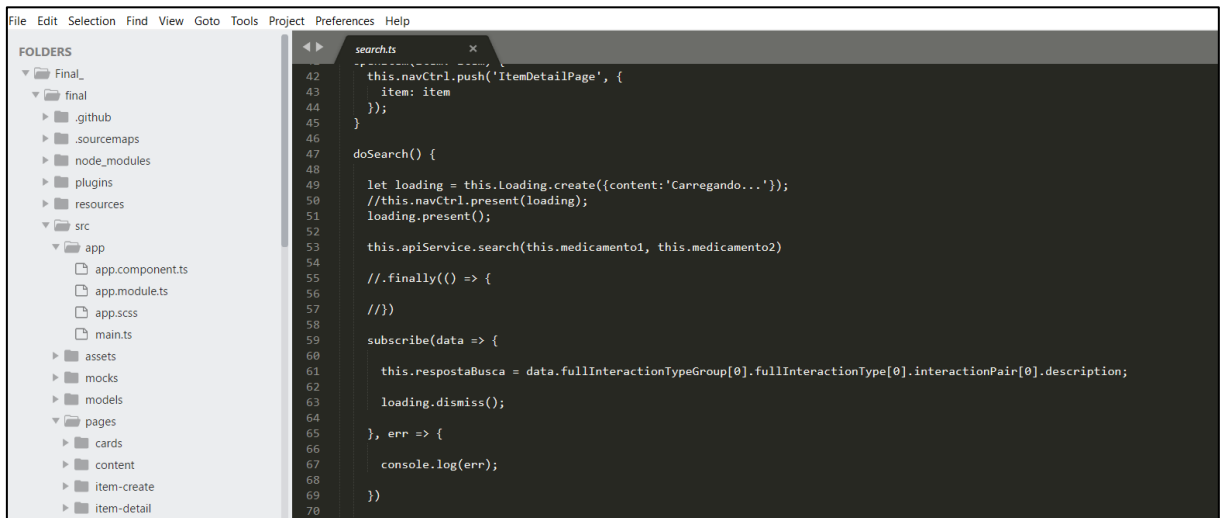
```

        <ion-option value="1998">Captopril</ion-option>
        <ion-option value="5640">Ibuprofeno</ion-option>
    </ion-select>
</ion-item>
<ion-item>
    <ion-label>Segundo Medicamento</ion-label>
    <ion-select [(ngModel)]="medicamento2">
        <ion-option value="91101">AAS</ion-option>
        <ion-option value="7646">Omeprazol</ion-option>
        <ion-option value="596">Alprazolam</ion-option>
        <ion-option value="1998">Captopril</ion-option>
        <ion-option value="5640">Ibuprofeno</ion-option>
    </ion-select>
</ion-item>
<div padding>
    <button ion-button color="primary" block (click)="doSearch();">{{
'DESCUBRA!' | translate }}</button>
</div>
<div *ngIf="respostaBusca">
    {{ respostaBusca | translate }}
</div>
</ion-content>

```

Para a integração com a API foi preciso criar funções e métodos para manipular os resultados obtidos nas interações. Na imagem à seguir, está um exemplo da função `doSearch()` que recebe dois parâmetros (`medicamento1` e `medicamento2`), armazena os valores de `rxci` (código universal dos medicamentos) em um array, e então retorna como resultado apenas a linha de descrição do Json, que será chamado na tela `search.html`, no clique do botão "Descobrir!".

Figura 10 - Tela com função que chama a API



```
42 this.navCtrl.push('ItemDetailPage', {
43   item: item
44 });
45 }
46
47 doSearch() {
48
49   let loading = this.Loading.create({content: 'Carregando...'});
50   //this.navCtrl.present(loading);
51   loading.present();
52
53   this.apiService.search(this.medicamento1, this.medicamento2)
54
55   // .finally(() => {
56   // })
57
58   subscribe(data => {
59
60     this.respostaBusca = data.fullInteractionTypeGroup[0].fullInteractionType[0].interactionPair[0].description;
61
62     loading.dismiss();
63
64   }, err => {
65
66     console.log(err);
67
68   })
69
70 }
```

Fonte: Print da tela search.ts, aberta no editor de textos Sublime. (2017)

12 TESTES

O sistema especialista foi testado por dois estudantes de Ciência da Computação da Universidade do Sagrado Coração.

O teste de produtos de software envolve basicamente quatro etapas: planejamento de testes, projeto de casos de teste, execução e avaliação dos resultados dos testes. (Maldonado et al, 2004).

O levantamento de requisitos é parte importante do desenvolvimento de um software. Segundo Paula Filho (2000), quando esse levantamento é bem feito, os requisitos implícitos são minimizados.

Há diversos tipos de testes que podem ser realizados para verificar o desempenho da aplicação e suas funções. Para esse trabalho foram escolhidos os testes nas seguintes categorias:

- Teste de Volume: teste a quantidade de dados envolvidos
- Teste de regressão: o sistema é testado toda vez que tenha uma mudança em seu desenvolvimento.
- Teste de aceitação do usuário: Verifica como o usuário reagiu à interface criada para a aplicação. No caso do InterMed, o tamanho dos botões, listas e navegação das páginas do aplicativo.

Para essa fase do desenvolvimento do aplicativo, foram extraídos e utilizados dados de cinco medicamentos diferentes da API, portanto, uma quantidade pequena de dados.

Em relação aos testes de aceitação do usuário, foi solicitado aos alunos que navegassem entre as todas as páginas disponíveis no aplicativo, de modo à avaliar a troca de uma para outra na tela da aplicação, bem como os elementos visuais e interativos, além das funções oferecidas pelo sistema.

O teste de regressão por duas vezes, para descobrir se houve alguma instabilidade na aplicação após pequenas alterações em seu layout.

12.1 TESTE DE INTERFACE DO SOFTWARE

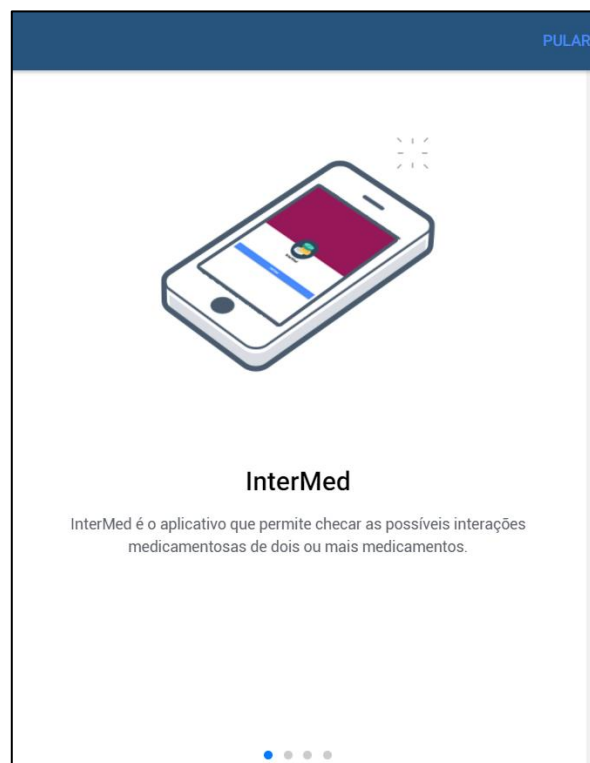
Os testes realizados nos permitem chegar aos seguintes resultados:

A interface permitiu que o usuário conhecesse um pouco mais sobre o aplicativo antes de procurar pelas interações. A seguir, são as telas que chamei de tutorial, contendo breves instruções daquilo que é visto na aplicação.

A tela de boas-vindas apresenta a logomarca do aplicativo e também o botão Entrar. O Objetivo é que com poucos movimentos o usuário possa chegar a principal função do aplicativo: checar interações entre medicamentos.

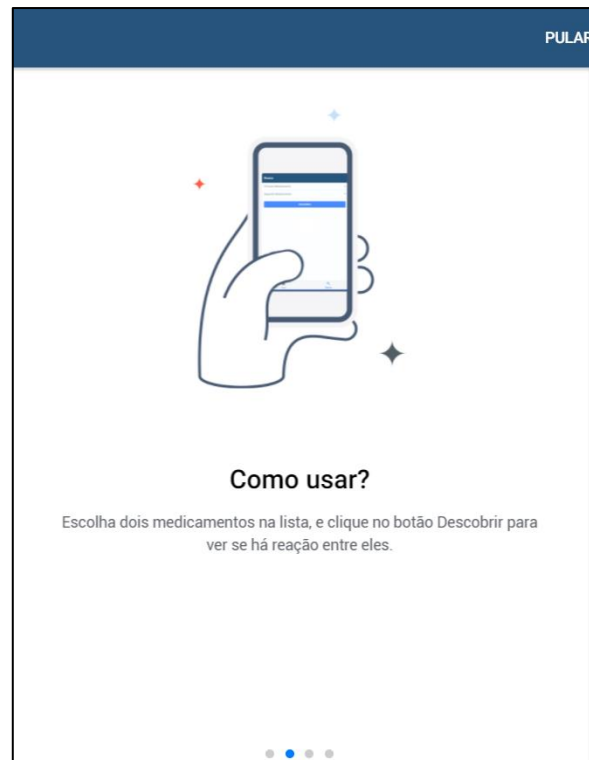
A Figura 11 mostra a primeira tela do tutorial disponibilizado no aplicativo, seguido pela Figura 12, onde é mostrado ao usuário o procedimento de consulta que deve ser realizado para verificar as interações, pela Figura 13 que mostra um exemplo de como as listas de medicamentos podem ser utilizadas, seja para cadastrar um medicamento já procurado, ou também para adicionar os nomes do que ainda serão procurados, e pela Figura 14, um convite ao usuário para acessar o aplicativo e descobrir todas as suas funcionalidades.

Figura 11 - Primeira tela do tutorial



Fonte: Print da primeira tela de tutorial. (2017)

Figura 12 - Segunda tela do tutorial



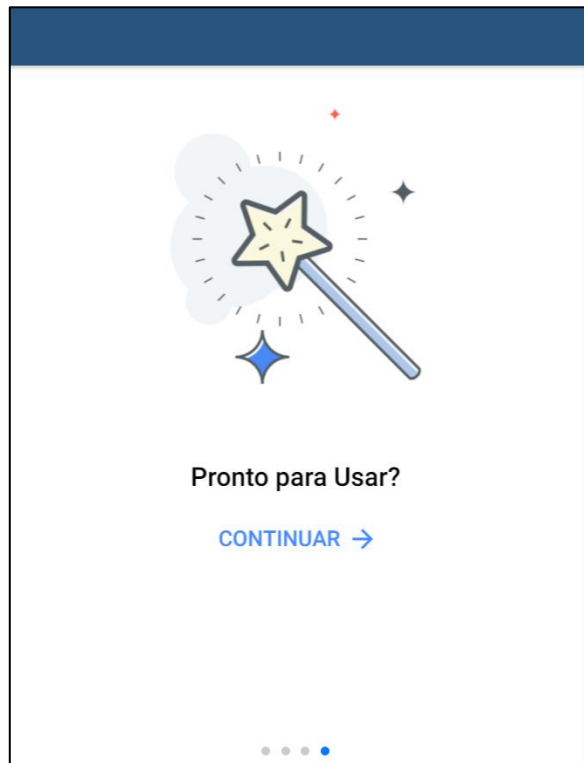
Fonte: Print da segunda tela de tutorial. (2017)

Figura 13 - Terceira tela do tutorial



Fonte: Print da terceira tela de tutorial. (2017)

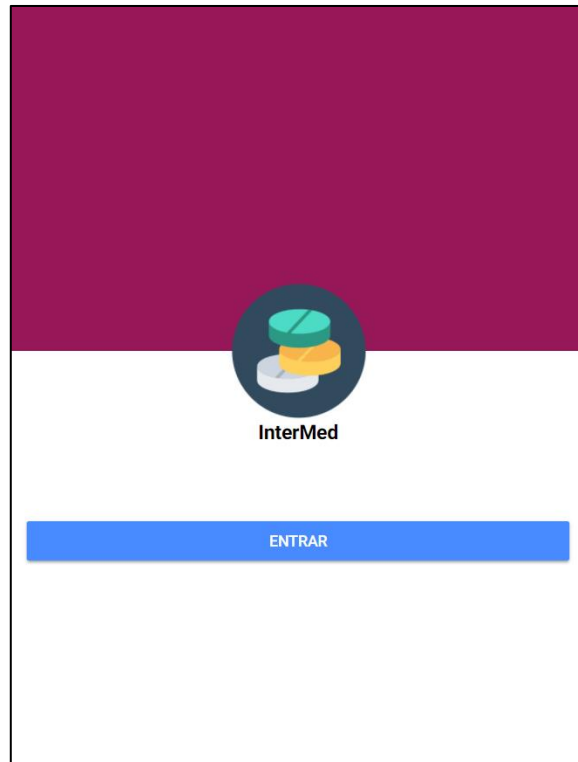
Figura 14 - Última tela do tutorial



Fonte: Print da última tela de tutorial. (2017)

Após a tutorial, há a tela de boas-vindas ao aplicativo, conforme mostrada na Figura 15.

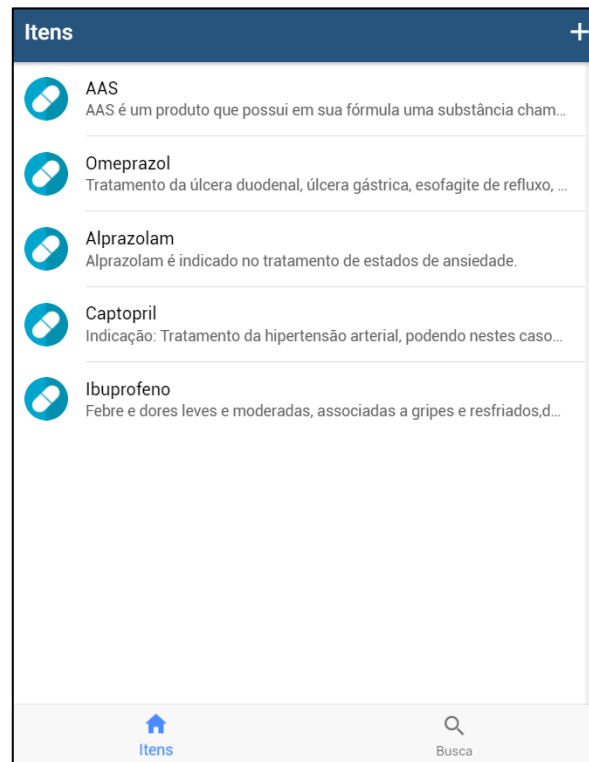
Figura 15 - Tela de boas-vindas do aplicativo



Fonte: Print da tela de boas-vindas do aplicativo. (2017)

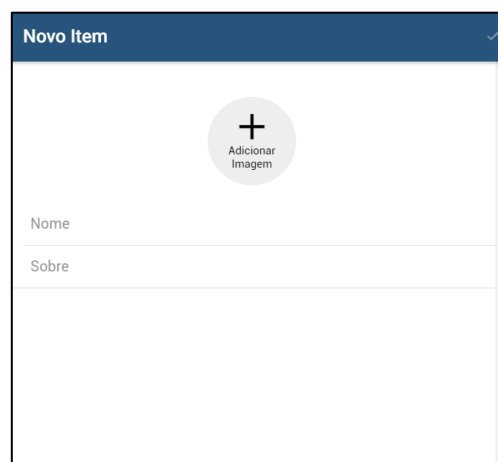
As figuras 16 e 17 mostram a página Itens, que permite criar uma lista pessoal com o nome, foto e descrição de alguns medicamentos. Para que o usuário faça o controle dos medicamentos administrados por ele, como horários, doses, e até fazer o upload de fotos da embalagem do medicamento.

Figura 16 - Tela que apresenta a aba Itens



Fonte: Print da tela Itens. (2017)

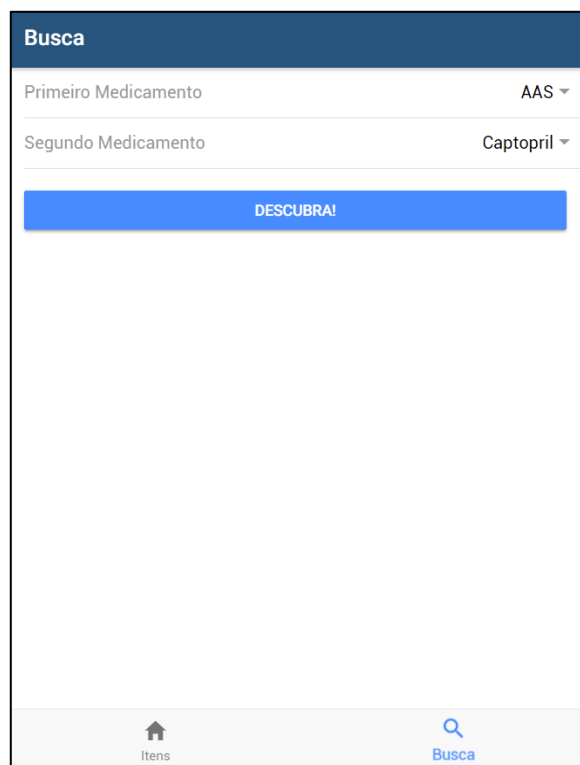
Figura 17 - Tela para inclusão de novos itens



Fonte: Print da tela de inclusão de novos itens. (2017)

A Busca cumpriu seu objetivo e sua tela pode ser vista na Figura 18. Através de duas listas com os medicamentos pré-definidos, o usuário escolhe dois distintos para testar se há ou não interação entre eles. Então, foi feita a requisição à API que em poucos segundos trouxe a mensagem retornada pelo json, onde confirma os efeitos colaterais da administração concomitante dos dois medicamentos escolhidos pelo usuário, conforme telas:

Figura 18 - Tela de buscas de interações medicamentosas



Fonte: Print da tela de buscas. (2017)

13 CONCLUSÃO

O aplicativo cumpriu a proposta inicial de demonstrar a interação de medicamentos através de um sistema especialista.

A principal dificuldade encontrada durante o desenvolvimento do aplicativo foi no retorno dos dados obtidos através da API de interações medicamentosas. Hoje, a maioria dos bancos de dados utilizados na área da saúde estão em inglês, o que inviabiliza o acesso de grandes massas ao aplicativo. Como uma ideia futura de melhoria, seria interessante pensar numa solução de tradução para o Português das respostas das interações retornadas pelo JSON.

Outras implementações podem ser feitas na aplicação, como fazer a interação entre a quantidade de medicamentos que o usuário achar necessário, além de permitir que através de uma barra de busca seja colocado o nome do medicamento, e então tenhamos o retorno do JSON com todas as informações compatíveis com o que foi digitado pelo usuário.

O layout foi eficaz e permitiu que o usuário chegasse intuitivamente à principal função do aplicativo, que é a detecção das interações medicamentosas. O tamanho dos botões, listas e outros objetos também atenderam às necessidades reais de navegação, bem como o fluxo de abrir uma nova página e voltar às anteriores sem ocasionar erros na utilização da aplicação, o que significou boa aceitação por parte do usuário.

Com a baixa quantidade de dados envolvidas nessa etapa do projeto, não houve lentidão da aplicação durante o tempo de requisição à API. O desempenho do sistema foi rápido e mostrou se houve interação, conforme inicialmente proposto.

14 REFERÊNCIAS

BARRETO, Luiz; PREZOTO, Marcelo. **INTRODUÇÃO A SISTEMAS ESPECIALISTAS.** , [S.L.], jan./dez. 2010. Disponível em: <http://www.ft.unicamp.br/liag/monografias/monografias/2010_IA_FT_UNICAMP_sistemasEspecialistas.pdf>. Acesso em: 21 mai. 2016.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **Uml: guia do usuário.** [S.L.]: Elsevier Brasil, 2006. 474 p. Disponível em: <https://books.google.com.br/books?id=ddWqxcDKGF8C&printsec=frontcover&dq=Uml:+guia+do+usu%C3%A1rio&hl=pt-BR&sa=X&ved=0ahUKEwifvWC5v_MAhUGPCYKHUzPA0kQ6AEINTAA>. Acesso em: 25 mai. 2016.

BRASIL PHARMA. **Automedicação: problema de saúde pública.** Disponível em: <<http://brasilpharma.com.br/site/bem-estar-brasil-pharma/automedicacao-problema-de-saude-publica/>>. Acesso em: 20 mar. 2016.

BURNETTE, Ed. **Hello, android:** Introducing google's mobile development platform. [S.L.]: The Pragmatic Bookshelf, 2008.

CLAUDIO, A. Introdução a teste de software. DEVMEDIA, Rio de Janeiro, 201-. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035>>. Acesso em: 26 mai. 2016.

COFEN. **Novos aplicativos disponíveis para android e ios.** Disponível em: <http://www.cofen.gov.br/novos-aplicativos-disponiveis-para-android-e-ios_36556.html>. Acesso em: 20 mai. 2016.

COPPIN, Ben. **Artificial intelligence illuminated.** 1 ed. Massachusetts: Jones and Bartlett Publishers, 2004.

CORRER, Cassyano J; OTUKI, Michel F. **MÉTODO CLÍNICO DE ATENÇÃO FARMACÊUTICA.** , [S.L.], mar. 2011. Disponível em: <<http://www.saude.sp.gov.br/resources/ipgg/assistencia-farmaceutica/otuki-metodoclinicoparaatencaofarmaceutica.pdf>>. Acesso em: 20 mai. 2016.

CRAIG, R. D., JASKIEL, S. P. **Systematic Software Testing.** Boston: Artech House Publishers, 2002.

DEITEL, P. J.; DEITEL, H. M.. **Java: HOW TO PROGRAM.** 9 ed. [S.L.: s.n.], 2012.

DREYFUS, H. L.. **O que os computadores não podem fazer:** Crítica da razão artificial. Rio de Janeiro: A Casa do Livro Eldorado S.A, 1976.

EDITORA MANDRUVÁ. **Sistemas inteligentes aplicados à medicina - "interações medicamentosas".** Disponível em: <<http://hottopos.com/videtur27/eli.htm>>. Acesso em: 09 abr. 2016.

EXAME. **10 aplicativos para cuidar da saúde**. Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/10-aplicativos-para-cuidar-da-saude#9>>. Acesso em: 21 mai. 2016.

FACELI, K. et al. **Inteligência artificial: uma abordagem de aprendizagem de máquina**. Rio de Janeiro: LTC, 2011.

GAZZANIGA, S. MICHAEL. **Cognitive neuroscience: a reader**. [S.L.: s.n.], 2000.

GRZYMALA-BUSSE, Jerzy W.. Managing Uncertainty in Expert Systems. **Managing Uncertainty in Expert Systems**, [S.L.], ./. undefined. Disponível em: <<http://www3.cs.stonybrook.edu/~cse352/busse.pdf>>. Acesso em: 20 abr. 2016.

HANSTEN, D. PHILIP. **Associação de medicamentos**. Rio de Janeiro - São Paulo: Livraria Atheneu, 1984.

HAYKIN, SIMON. **Neural networks and learning machines**. 3 ed. [S.L.: s.n.], 1999.

HCI BOOK. **Human-computer interaction**. Disponível em: <<http://www.hcibook.com/e3/plain/chaps/ch15>>. Acesso em: 21 mai. 2016.

INTERAÇÕES MEDICAMENTOSAS. **Informações para profissionais de saúde**. Disponível em: <<http://interacoesmedicamentosas.com.br>>. Acesso em: 15 mar. 2016.

JOEL ON SOFTWARE. **Joel on software**. Disponível em: <<http://brazil.joelonsoftware.com/uibook/chapters/1.html>>. Acesso em: 21 mai. 2016.

JR., O LUDWIG; COSTA, E. M. M. **Redes neurais: fundamentos e aplicações com programas em c**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2007.

JÚNIOR, F. P. et al. Potenciais interações medicamentosas em prescrições de um hospital-escola de Porto Alegre. **Revista da AMRIGS**, Porto Alegre, v. 53, n. 3, p. 251-256, jul./set. 2009. Disponível em: <http://www.amrigs.org.br/revista/53-03/12-420_potenciais-intera%C3%A7%C3%B5es.pdf>. Acesso em: 09 abr. 2016.

KREIBICH, J. **Using SQLite**. 2010. Disponível em: <https://books.google.com.br/books?id=HFIM47wp0X0C&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false>. Acesso em: 01 jun. 2016.

LECHETA, Ricardo R.. **Google android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3 ed. [S.L.]: Novatec Editora, 2013.

LEVINE, ROBERT I.; DRANG, DIANE E.; EDELSON, BARRY. **Inteligência artificial e sistemas especialista**. São Paulo: McGraw-Hill, 1988.

LEÃO, Danyllo Fábio Lessa; MOURA, Cristiano Soares De; MEDEIROS, Danielle Souto De. Avaliação de interações medicamentosas potenciais em prescrições da

atenção primária de Vitória da Conquista (BA), Brasil. **Ciência e Saúde Coletiva**, [S.L], nov./jan. 2013. Disponível em: <<http://www.scielo.org/pdf/csc/v19n1/1413-8123-csc-19-01-00311.pdf>>. Acesso em: 09 abr. 2016.

MATTOS, M. M., FERNANDES, A., LÓPEZ, O. C. Sistema Especialista para Apoio ao Aprendizado de Lógica de Programação. 1999. Disponível em: <http://www.inf.furb.br/~mattos/arquivos/publicacoes/sistema_especialista.doc>. Acesso em: 29 mai. 2016.

MENDES, Raquel Dias. Inteligência Artificial: Sistemas Especialistas No Gerenciamento Da Informação. **Ciência da Informação**, Brasília, v. 26, n. 1, jan./abr. 1997. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19651997000100006&lng=en&nrm=iso>. Acesso em: 09 mar. 2016.

MYERS, A. et al. Im2Calories: towards an automated mobile vision food diary. **Research at Google**, [S.L], ./ . undefined. Disponível em: <<http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/44321.pdf>>. Acesso em: 19 mar. 2016.

MYERS, A. et al. Im2Calories: towards an automated mobile vision food diary. **Research at Google**, [S.L], n. 44321, ./ . undefined. Disponível em: <<http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/44321.pdf>>. Acesso em: 19 mai. 2016.

NATIONAL LIBRARY OF MEDICINE. **RXNorm**. Disponível em: <<https://www.nlm.nih.gov/>>. Acesso em 10 Out. 2017

NETO, P. R. O. et al. **Farmacoterapia: guia terapêutico de doenças mais prevalentes**. São Paulo: Pharmabooks, 2013.

NILSSON, NEILS S. **Principles of artificial intelligence**. Berlin: Springer Verlag, 1982.

NILSSON, Nils J. Human-Level Artificial Intelligence? Be Serious!. **AI Magazine**, [S.L], n. 25, jan./dez. 2005. Disponível em: <<http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/General%20Essays/AIMag26-04-HLAI.pdf>>. Acesso em: 15 mar. 2016.

ORGANIZATION, World Health. The World Health Report 2014. **The World Health Report 2014**, [S.L], 200. undefined. Disponível em: <<http://apps.who.int/iris/bitstream/10665/42891/1/924156265X.pdf>>. Acesso em: 20 mai. 2016.

PESSOA, F. **Livro do Desassossego**, por Bernardo Soares. Vol. II. Mem Martins: Europa-América, 1986.

PRESSMAN, Roger S. **Engenharia de software**. 7 ed. [S.L.]: AMGH Editora, 2009.

RAPPS, S.; WEYUKER, E.J. **Data Flow analysis techniques for test data**

selection In: International Conference on Software Engineering, p. 272-278, Tokio, Sep. 1982.

RESENDE, Antônio Maria Pereira De; SILVA, Claudiney Calixto Da. **Java linguagem orientada ao objeto**. [S.L.]: Brasport, 2005.

REZENDE, Denis Alcides. **Engenharia de software e sistemas de informação**. 3 ed. Rio de Janeiro: Brasport, 2005.

ROCHA, A. R. C. et al. **Qualidade de software: teoria e prática**. São Paulo: Prentice Hall, 2001.

SCIENCEC MUSEUM. **Technology and medicine**. Disponível em: <<http://www.sciencemuseum.org.uk/broughttolife/themes/technologies>>. Acesso em: 13 mar. 2016.

SILVA, N. M. O. et al.; Avaliação de potenciais interações medicamentosas em prescrições de pacientes internadas, em hospital público universitário especializado em saúde da mulher, em Campinas-SP. **Revista de Ciências Farmacêuticas Básica e Aplicada**, [S.L], p. 171-176, dez./jun. 2010. Disponível em: <http://serv-bib.fcfar.unesp.br/seer/index.php/Cien_Farm/article/viewFile/1074/949>. Acesso em: 09 abr. 2016.

SISTEMAS ESPECIALISTAS. **Sistemas especialistas**. Disponível em: <<http://ws2.din.uem.br/~ia/especialistas/>>. Acesso em: 21 mai. 2016.

SOCIEDADE BRASILEIRA DE ENDOCRINOLOGIA E METABOLOGIA. **Os perigos da automedicação**. Disponível em: <<http://www.endocrino.org.br/os-perigos-da-automedicacao>>. Acesso em: 20 mar. 2016.

SOUZA, Ademar Rosa De; TALON, Anderson Francisco. Integração entre Banco de Dados e Redes de Bayes no Suporte à Medicina. **Fatec**, [S.L], v. 3, n. 26, ./ . undefined. Disponível em: <http://www.fatecgarca.edu.br/revista/Volume3/artigos_vol3/Artigo_26.pdf>. Acesso em: 08 abr. 2016.

WHITSON, G. M.. **Artificial intelligence**. [S.L.]: Salem Press Encyclopedia of Science, 2015.

WIDMAN, Lawrence E.. Sistemas Especialistas em Medicina. **Informática Médica**, [S.L], v. 1, n. 5, set./out. 1998. Disponível em: <<http://www.informaticamedica.org.br/informaticamedica/n0105/widman.htm>>. Acesso em: 09 mar. 2016.

