

CURSO DE CIÊNCIA DA COMPUTAÇÃO

Karoline Hoffmann

**DESENVOLVIMENTO DE APLICAÇÃO MÓVEL PARA AUXÍLIO NA TOMADA
DE DECISÃO PARA A ÁREA DE SAÚDE ANIMAL**

Santa Cruz do Sul, Junho de 2015.

Karoline Hoffmann

**DESENVOLVIMENTO DE APLICAÇÃO MÓVEL PARA AUXÍLIO NA TOMADA
DE DECISÃO PARA A ÁREA DE SAÚDE ANIMAL**

Trabalho de Conclusão apresentado ao Curso de
Ciência da Computação da Universidade de Santa Cruz
do Sul, para obtenção do título de Bacharel em Ciência
da Computação.

Orientadora: Prof^a. Dra. Andréa Aparecida Konzen.

Santa Cruz do Sul, Junho de 2015.

AGRADECIMENTOS

Não poderia deixar de agradecer a todos que de alguma maneira contribuirão para essa conquista.

Agradeço á meus pais que sempre me apoiaram, ajudaram e entenderam meus finais de semana em casa, sem vocês não teria chegado aonde cheguei e espero deixa-los orgulhosos.

Agradeço ao meu namorado que ao longo de todos esses anos esteve presente e que por muitas vezes foi deixado de lado por motivos de estudo, eu te amo e essa conquista é tanto minha quanto sua.

Agradeço aos amigos que muitas vezes reclamaram de minha ausência, mas sempre compreenderão o quanto os estudos exigiam de mim e o quanto isso era importante.

Agradeço de coração a minha orientadora Prof^a Dra. Andréa Aparecida Konzen que me deu grande apoio na realização deste trabalho, contribuindo com dicas, conselhos e puxadas de orelha.

Por fim agradeço a todos que fizeram parte disto de alguma maneira, amigos, familiares, colegas e professores, essa conquista é um pouco de cada um.

*“Aonde fica a saída?”, Perguntou Alice ao gato que ria.
”Depende”, respondeu o gato.
”De quê?”, replicou Alice;
”Depende de para onde você quer ir...”
“Alice no país das maravilhas” - Lewis Carroll*

RESUMO

A Inteligência Artificial, ao longo dos anos, se tornou uma ferramenta de grande valia para a execução de tarefas do cotidiano em inúmeras áreas específicas. Uma dessas técnicas, o Raciocínio Baseado em Casos, possibilita de forma mais efetiva a utilização dos dados coletados dos usuários, ajudando a refinar consultas e tornando os resultados de buscas mais eficientes, através da criação e armazenamento de casos novos e uso de casos já existentes. Além disso, o avanço crescente da tecnologia aliado a imensa disponibilidade de ferramentas para o desenvolvimento de aplicativos usados em dispositivos móveis tem permitido a criação de aplicações que façam parte do cotidiano do usuário, possibilitando a união da mobilidade dos dispositivos com a utilidade de um computador pessoal para o auxílio nas atividades pessoais e profissionais. Com esse crescimento exponencial e a grande aceitação dos dispositivos móveis, o desenvolvimento de aplicações específicas já se tornou um padrão em relação a profissionais e empresas da área de tecnologia e, tanto para desenvolvedores quanto para os usuários, que estão cada vez mais exigentes em relação à utilização de aplicações para estes tipos de dispositivos, torna-se indispensável para o dia a dia. Unindo a técnica de Inteligência Artificial com uma aplicação móvel, consegue-se obter diferenciais que permitem ao usuário fazer a atividade necessária de maneira automatizada e inteligente, além de tornar a utilização agradável e de boa usabilidade. Assim, este trabalho aborda a criação de um aplicativo para dispositivos móveis utilizando as ferramentas para desenvolvimento disponíveis da plataforma Android. A proposta é auxiliar os profissionais da área de saúde animal, focando em animais de grande porte, no diagnóstico e tratamento, que necessitam de mobilidade para atender onde for preciso e integrar com uma base de dados que possa auxiliar na sua tomada de decisão em relação a diagnósticos e o acompanhamento de animais, utilizando uma técnica inteligente.

Palavras chave: Inteligência Artificial, PhoneGap, Android, Raciocínio Baseado em Casos, Dispositivos Móveis, Saúde Animal.

ABSTRACT

The Artificial Intelligence, over the years, has become a valuable tool for performing everyday tasks in several specific areas. One such technique, the Case Based Reasoning, enables more effectively the use of data collected from users, helping to refine queries and making the results more efficient searches, by creating and storing new cases and use of existing cases. In addition, the increasing advancement of technology combined with immense availability of tools for developing applications used on mobile devices has allowed the creation of applications that are part of the everyday user, enabling the Union mobility devices with the use of a computer staff for assistance in personal and professional activities. With this exponential growth and wide acceptance of mobile devices, the development of specific applications has become a standard in relation to professionals and companies in the technology area, both for developers and for users, who are increasingly demanding in relation use applications for these types of devices, it is essential for day to day. Joining the Artificial Intelligence technique with a mobile application, it is possible to obtain advantages that allow the user to make the necessary activity of automated and intelligent way, and make nice use and good usability. This work deals with creating an application for mobile devices using the tools available to develop the Android platform. The proposal is to assist animal health professionals, focusing on large animals, diagnosis and treatment, who need mobility to serve where needed and integrate with a database that can aid in your decision-making concerning the diagnosis and monitoring of animals, using an intelligent technique.

Keywords: Artificial Intelligence, PhoneGap, Android, Case Based Reasoning, Mobiles, Animal Health.

LISTA DE FIGURAS

Figura 1 - Modelo básico do enfoque RBC	18
Figura 2 - Ciclo do RBC.....	19
Figura 3 - Exemplo simplificado de uma base de casos	20
Figura 4 - Componentes básicos na representação dos casos	22
Figura 5 - Visão Geral do modelo RBC no projeto IACVIRTUAL.....	27
Figura 6 - Estrutura de um Sistema de recomendação	29
Figura 7 - Estrutura de SR baseado em técnicas de filtragem de conteúdo.....	30
Figura 8 - Estrutura de SR com aplicação de Mineração de Dados	31
Figura 9 - Taxonomia de um SR.....	32
Figura 10 - Usuários de telefones celular com 16 anos ou mais	42
Figura 11 - Os sistemas operacionais mais utilizados nos últimos 5 anos	44
Figura 12 - Arquitetura básica Android	46
Figura 13 – Ciclo de vida de uma atividade	48
Figura 14 - Arquitetura Apache Cordova / Phonegap	52
Figura 15 - Fluxo de criação com o PhoneGap e utilização de Plugins	53
Figura 16 - Diagrama de Atores	57
Figura 17 - Diagrama de Casos de Uso	59
Figura 18 - Diagrama ER Processo de cadastro	63
Figura 19 - Diagrama ER Processo RBC	64
Figura 20 - Diagrama de sequência	66
Figura 21 - Arquitetura base da aplicação proposta	69
Figura 22 - Tela inicial da aplicação.....	70
Figura 23 - Menu da aplicação	71
Figura 24 - Tela de listagem de registros.....	72
Figura 25 - Cadastro de Espécie	72
Figura 26 - Cadastro de Raça.....	73
Figura 27 - Cadastro de Animal.....	74
Figura 28 - Tela inicial do diagnóstico	75
Figura 29 - Diagnóstico: Seleção de sintomas.....	75
Figura 30 - Diagnóstico: Seleção de Patologia.....	76
Figura 31 - Tratamento Manual	77
Figura 32 - Diagnóstico: exibição de tratamento.....	78
Figura 33 - Fluxo do processamento RBC na aplicação.....	79
Figura 34 - Fórmula para determinar o grau de similaridade	80
Figura 35 - Ficha do especialista	82

LISTA DE TABELAS

Tabela 1 - Principais diferenças entre os sistemas convencionais e os sistemas baseados em conhecimento	26
Tabela 2 - Grau de personalização e seus métodos de recomendação.....	34
Tabela 3 - Comparação entre trabalhos relacionados.	39
Tabela 4 - Membros da OHA	45
Tabela 5 - Métodos que definem o ciclo de vida de uma atividade	49
Tabela 6 – Análise dos Trabalhos relacionados	54
Tabela 7 - Requisitos Funcionais.....	56
Tabela 8 - Regras de Negócio.....	58
Tabela 9 - Caso de Uso UC1: Cadastro de Espécie.....	60
Tabela 10- Caso de Uso UC2: Cadastro de Raça	60
Tabela 11 - Caso de Uso UC3: Cadastro de Animal	61
Tabela 12 - Caso de Uso UC4: Diagnóstico	61
Tabela 13 - Caso de Uso UC5: Listar Sintomas	61
Tabela 14 - Caso de Uso UC6: Listar Tratamento	62
Tabela 15 - Caso de Uso UC7: Confirmar Tratamento	62
Tabela 16 - Exemplo de pesquisa a ser feita.....	81

LISTA DE ABREVIATURAS

API – *Application Programming Interface*
BC – Base de Casos
BD – Base de Dados
IA – Inteligência Artificial
IDE – Ambiente Integrado de Desenvolvimento
JDK – *Java Development Kit*
JVM – *Java Virtual Machine* ou Máquina Virtual Java
MLE – *Mobile Learning Engine*
OHA – Open Handset Alliance
RBC – Raciocínio Baseado em Casos
SBC – Sistemas Baseados em Conhecimento
SDK – *Software Development Kit*
SE – Sistemas Especialistas
SI – Sistemas Inteligentes
SO – Sistema Operacional
SR – Sistemas de Recomendação
TCI – Trabalho de Conclusão I
TCII – Trabalho de Conclusão II
VM – Máquina Virtual
VS2012 – *Visual Studio 2012*

SUMÁRIO

1. INTRODUÇÃO	11
1.1. Objetivo Geral	12
1.2. Objetivos Específicos	13
1.3. Motivação	13
1.4. Organização do texto	13
2. IA E AS TECNICAS ESTUDADAS	15
2.1. Raciocínio Baseado em Casos	16
2.1.1. RBC e seus elementos básicos	17
2.1.2. Ciclo de RBC	18
2.1.3. Base de Casos	20
2.1.4. Desenvolvimento de um sistema RBC	20
2.1.5. Aquisição e representação dos casos	22
2.1.6. Indexação dos casos	23
2.1.7. Recuperação dos casos	23
2.1.8. Reutilização ou adaptação dos casos	24
2.1.9. Vantagens e Limitações	25
2.1.10. Trabalhos relacionados	27
2.2. Sistemas de Recomendação	28
2.2.1. Arquitetura e estrutura	30
2.2.2. Taxonomia	31
2.2.3. Métodos de recomendação	32
2.2.4. Entrada do Usuário Alvo e Entradas da Comunidade	33
2.2.5. Saída	34
2.2.6. Entrega	34
2.2.7. Grau de personalização	34
2.2.8. Técnicas de Sistemas de Recomendação	35
2.2.9. Vantagens e Desvantagens	36
2.2.10. Trabalhos relacionados	37
2.3. Comparação RBC X SR e aplicação no trabalho proposto	38
2.4. Considerações	40
3. DISPOSITIVOS MÓVEIS	41
3.1. Android	42

3.1.1. A plataforma.....	43
3.1.2. Arquitetura	46
3.1.3. Ciclo de Vida.....	48
3.1.4. Android SDK.....	49
3.1.5. Ferramentas de desenvolvimento	50
3.2. Apache Cordova / Phonegap	51
3.3. Trabalhos relacionados.....	53
3.4. Considerações	55
4. LEVANTAMENTO DE REQUISITOS PARA A APLICAÇÃO	56
4.1. Requisitos Funcionais	56
4.2. Requisitos não funcionais	56
4.3. Descrição dos atores.....	57
4.4. Regras de negócio	57
4.5. Casos de Uso	59
4.5.1. Diagrama de Casos de Uso	59
4.5.2. Descrição dos Casos de uso	60
4.6. Diagrama de Entidade e Relacionamento	62
4.7. Diagrama de Sequências	65
4.8. Considerações	67
5. APLICAÇÃO MÓVEL DESENVOLVIDA	68
5.1. Arquitetura do sistema	68
5.2. Tecnologias utilizadas	69
5.3. Metodologia e Desenvolvimento	70
5.3.1. Aplicação do RBC na aplicação proposta.....	78
5.3.2. Avaliação do especialista.....	80
5.3.3. Considerações	82
6. CONSIDERAÇÕES FINAIS	83
6.1. Trabalhos Futuros.....	84
7. REFERÊNCIAS.....	85

1. INTRODUÇÃO

“O que é a ciência na Inteligência Artificial”, perguntou em 1977, Allen Newell, e sua resposta foi clara: “conhecimento que descreve os meios para alcançar uma classe de fins desejados, dadas certas estruturas e situações. A ciência vai além da situação da sua geração, e, torna-se uma fonte de conhecimento para futuros cientistas e tecnólogos – investimento, em vez de consumo” (COELHO, 1995).

Com o avanço da tecnologia, nos últimos anos, a Inteligência Artificial é reconhecida como um importante instrumento no processo de tomada de decisão para auxiliar em diferentes áreas. Atualmente, devido a esse reconhecimento, diversas propostas de ambientes virtuais estão sendo desenvolvidas para a integração entre o virtual e o real, agregando inteligência.

Na Inteligência Artificial, ao longo dos anos, foram propostos modelos computacionais que permitem auxiliar e descrever o comportamento humano em campos especializados. Tais sistemas estão em todas as áreas podendo atuar em praticamente todos os campos onde se tenha a necessidade de tomadas de decisão em sistemas complexos que necessitem de um grau de segurança alto. Assim, diversas são as técnicas que podem ser usadas para o auxílio na tomada de decisão. Entre elas foram escolhidas, para estudo, os Sistemas de Raciocínio Baseado em Casos (RBC) e os Sistemas de Recomendação (SR).

Os sistemas de Raciocínio Baseado em Casos são compostos de uma técnica que busca solucionar novos problemas adaptando soluções utilizadas para resolver problemas anteriores (WANGENHEIM e WANGENHEIM, 2003). A qualidade do sistema vai depender da experiência e utilização que este já obteve, dependendo do número de casos relevantes e com boa solução que faz parte da base de casos. Uma vez que existe uma base de dados é possível extrair o conhecimento necessário a ser aplicado na tomada de novas decisões.

Os Sistemas de Recomendação também foram estudados com o propósito de verificar sua aplicabilidade para o objeto de estudo. Estes sistemas têm como ideia principal a criação de um ambiente onde cada usuário tem seu espaço personalizado. A maior utilização desse tipo de ferramenta é para comércio eletrônico, mas não sendo a única. Podem ser classificados em dois grupos: colaborativos, onde há relação com o usuário que possui interesse em comum, gerando recomendação a grupos de usuários semelhantes, e baseados em conteúdo, onde há recomendação com base em itens semelhantes àqueles relacionados com um

determinado usuário ou pesquisa, podendo recomendar resultados individualmente. Estes tipos de sistemas podem ser de grande valia na automatização parcial ou total de busca por informações relevantes ao usuário, quando este colabora de alguma maneira com as informações do sistema, possibilitando a escolha de produtos, serviços, conteúdos relevantes dentre uma rede complexa de informações.

Aliando todos os benefícios da Inteligência Artificial com a utilidade e popularização dos dispositivos móveis para o auxílio de profissionais, cria-se uma nova maneira de interagir com os sistemas de computador, sendo possível adicionar a pequenos dispositivos as mesmas funcionalidades que antes se permitiam somente a computadores pessoais. Com a ascensão destes dispositivos, veio a necessidade do desenvolvimento de bons sistemas que deem suporte às diversas necessidades dos usuários e que permitam que os mesmos interajam de maneira simples e objetiva com as aplicações, e, que também, seja permitido que empresas ou usuários desenvolvam suas próprias aplicações para venda ou para própria utilização.

Várias são as plataformas para desenvolvimento de aplicações para dispositivos móveis. Dentre as mais populares está o sistema Android, que é formado por diversas empresas de áreas específicas, sendo este de código aberto, desenvolvimento acessível e que devido à utilização dele em aparelhos muito mais baratos, teve um alcance muito superior em pouco tempo. Essa flexibilidade de desenvolvimento permite que a plataforma Android esteja sempre se adaptando para inclusão de novas tecnologias, já que a comunidade de desenvolvedores está sempre trabalhando em conjunto para disponibilizar aplicações móveis inovadoras. O desenvolvimento de aplicações para dispositivos móveis tem como ideia inicial que cada um pode tornar seu dispositivo o seu computador pessoal, e para isso, surge a necessidade de aplicativos com maior usabilidade, para atenderem desde o usuário leigo até o usuário com mais conhecimento.

1.1. Objetivo Geral

Tem-se como objetivo principal deste trabalho criar uma aplicação, desenvolvida para aplicativos móveis, que se utilize de uma das técnicas da Inteligência Artificial, Raciocínio Baseado em Casos, para auxiliar os profissionais na área de saúde animal no diagnóstico de doenças e a tomar decisões referentes ao animal, baseando-se em dados pré-cadastrados. O foco da aplicação será em relação a diagnósticos para animais de grande porte, localizados em

fazendas ou lugares mais remotos, onde um dispositivo móvel pode ser utilizado de maneira prática pelo profissional veterinário.

1.2. Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Pesquisar e analisar a fundamentação teórica necessária para o desenvolvimento de uma aplicação móvel utilizando a técnica de Raciocínio Baseado em Casos da Inteligência Artificial;
- Realizar o levantamento de requisitos para a criação do sistema;
- Analisar os requisitos coletados;
- Criar o projeto do software como base para a implementação da aplicação proposta;
- Desenvolver a aplicação móvel.
- Realizar testes junto a um especialista.

1.3. Motivação

O propósito principal é desenvolver e oferecer uma aplicação móvel inteligente capaz de auxiliar um profissional na tomada de decisão de diagnósticos e, assim, melhorar e tornar mais eficiente as suas decisões profissionais. Levando em conta que na medicina veterinária muitos profissionais possam vir a passar por dificuldades ao realizar o diagnóstico de animais, onde os sintomas encontrados são relacionados a mais de uma patologia existente, e onde um sintoma pode ser o diferencial para a identificação da patologia mais provável e realização do diagnóstico correto.

1.4. Organização do texto

O trabalho está organizado da seguinte forma: o capítulo 2 aborda a fundamentação teórica relacionada às técnicas da Inteligência Artificial estudadas para a elaboração deste trabalho (Raciocínio Baseado em Casos e Sistemas de Recomendação) agregando seus fundamentos teóricos com a finalidade de determinar qual das técnicas deveria ser aplicada a

este Trabalho de Conclusão seguido, no capítulo 3, da descrição referente a dispositivos móveis que faz a introdução para o estudo sobre a plataforma Android para desenvolvimento de aplicações para dispositivos móveis e desenvolvimento com o *FrameWork PhoneGap / Apache Cordova* que possibilita o desenvolvimento multiplataforma. O capítulo 4 é composto do levantamento de requisitos necessários para o desenvolvimento e entendimento da aplicação e no capítulo 5 é apresentado o software desenvolvido. E por fim, são apresentadas as conclusões, trabalhos futuros e as referências bibliográficas.

2. IA E AS TECNICAS ESTUDADAS

Não há como negar que os computadores têm algumas significativas vantagens em relação aos seres humanos, principalmente se tratando da velocidade de processamento de uma situação ou a consistência com a qual se executar determinadas funções. Mas para conseguir fazer um computador manipular e retornar informações de forma semelhante a um especialista humano é preciso que este tenha as ferramentas e o conhecimento necessário, para que estas sejam muito bem manipuladas e que sua base de dados seja relevante e consistente.

Muitos modelos computacionais vêm sendo propostos na área da Inteligência Artificial (IA) no decorrer das últimas décadas, que descrevem o comportamento humano em campos especializados. Tais sistemas estão em todas as áreas possíveis, podendo atuar em praticamente todos os campos onde se tenha a necessidade de tomadas de decisão e de sistemas complexos que necessitem de certo grau de segurança. A IA tem o propósito de sistematizar e automatizar tarefas de cunho intelectual, por isso é relevante para qualquer área que requer certo grau de inteligência, podendo ser aplicada em qualquer campo do universo (RUSSEL e NORVING, 2004).

Os pontos-chaves da IA para a elaboração de sistemas inteligentes são a habilidade de usar conhecimento para desempenhar tarefas ou resolver problemas e a capacidade para aproveitar associações e inferências para trabalhar com problemas complexos que se assemelham a problemas reais (REZENDE, 2003). Dessa forma se torna eficiente o armazenamento e recuperação das informações relevantes, onde a solução de problemas ou tomadas de decisões contribuem para o desenvolvimento de sistemas computacionais inteligentes, que se utilizam de técnicas da IA, sendo que estas podem estar sendo utilizadas individualmente ou em conjunto para o auxílio do processo decisório.

Na Inteligência Artificial, diversos podem ser os modelos desenvolvidos. Entre as áreas de aplicação estão os sistemas de Raciocínio Baseado em Casos e os Sistemas de Recomendação, que são as técnicas estudadas na próxima seção.

2.1. Raciocínio Baseado em Casos

Segundo Campos e Saito (2004), o ser humano adquire conhecimento da sociedade a sua volta, e estes conhecimentos ficam armazenados na memória e são processados quando necessário. Dessa maneira, o ser humano tem a capacidade de relembrar atividades já realizadas para solucionar questões atuais. Da vontade de entender e reproduzir a forma como os homens pensavam e resolviam os problemas surgiram os estudos do Raciocínio Baseado em Casos.

A técnica de RBC para a Inteligência Artificial foi baseada nos trabalhos de Roger Schank e seus estudantes na universidade de Yale, na década de 80. O modelo de memória dinâmica de Schank foi a base para o desenvolvimento dos primeiros sistemas de RBC: CYRUS, de Janet Kolodner, e IPP, de Michael Lebowitz. Essa técnica tornou-se uma ferramenta importante e muito utilizada para auxílio no processo decisório, resolução de problemas complexos, e capaz de atribuir maior agilidade e velocidade ao processo de decisão.

De acordo com Riesbeck e Schank (1989) o objetivo do RBC é resolver um novo problema relembrando uma situação anterior similar e, então, reutilizar a informação e conhecimento daquela situação. Ou seja, possibilita a extração do conhecimento a partir de casos ou experiências que se encontram no próprio sistema, identificando as características mais significativas do caso apresentado a fim de encontrar a melhor solução para o problema.

Ainda é possível afirmar que são sistemas que manipulam o conhecimento e as informações necessárias de maneira inteligente e tem a função de auxiliar em resoluções de questões que necessitam de conhecimento humano considerável (REZENDE, 2003).

Segundo WANGENHEIM e WANGENHEIM (2003) o RBC é aplicável de forma simples e direta a uma quantidade extremamente grande de problemas e situações do cotidiano, o que justifica a grande aceitação por quem desenvolve sistemas inteligentes, e foi influenciado por diversas outras áreas, como por exemplo, Ciências Cognitivas, Sistemas Baseados em Conhecimento (SBC), aprendizado de máquina, base de dados, recuperação de informações, redes neurais, entre outros. Também existe relação com reconhecimento de padrões, incerteza e estatística.

2.1.1. RBC e seus elementos básicos

Para explicar o funcionamento do RBC é possível utilizar várias situações do cotidiano, onde é notável como as pessoas se utilizam de casos conhecidos e já vividos para encontrar a melhor solução para um problema de forma natural e até mesmo automática.

WANGENHEIM e WANGENHEIM (2003) cita como um exemplo básico que mostra como pode ser obtida uma solução para o problema através de vivências já conhecidas e que foram utilizadas para guiar a então solução na situação atual: no atendimento de um novo paciente, ao escutar seus problemas, o médico relembra o histórico da doença de outro paciente, devido ao conjunto análogo de sintomas, e repassa o tratamento semelhante ao que foi administrado ao paciente que está sendo consultado.

A tecnologia RBC pode ser considerada como uma metodologia para moldar o pensamento e raciocínio humano e, também, para a construção de sistemas computacionais inteligentes para auxílio na tomada de decisões do dia-a-dia.

A situação elucidada anteriormente representa o característico caso de RBC, porém para se criar um sistema completo RBC é necessário que sejam definidos alguns de seus elementos básicos (WANGENHEIM e WANGENHEIM, 2003; MODRO, 2000), representados na Figura 1 e detalhados a seguir:

- **Representação do Conhecimento:** onde o conhecimento é representado principalmente através de casos que narram experiências definidas;
- **Medida de Similaridade:** consiste na capacidade de se encontrar um caso a partir da descrição de um problema e tem como objetivo principal definir o caso mais similar para o problema em questão, onde para se definir qual dos casos armazenados é similar a este é necessário medir a similaridade entre ambos;
- **Adaptação:** necessário haver previsão de que as situações já cadastradas e representadas como casos improvavelmente serão idênticas as do problema atual, por mais parecidos que eles possam ser. Dessa premissa surge a necessidade de uma adaptação da solução similar para que seja satisfatória para os requisitos da situação em questão;
- **Aprendizado:** este pode ser considerado o fator de diferenciação de um sistema RBC. O aprendizado ocorre toda vez que uma solução é aplicada e produz a solução para o problema, sendo incorporada a base de casos, para ser utilizada em questões futuras.

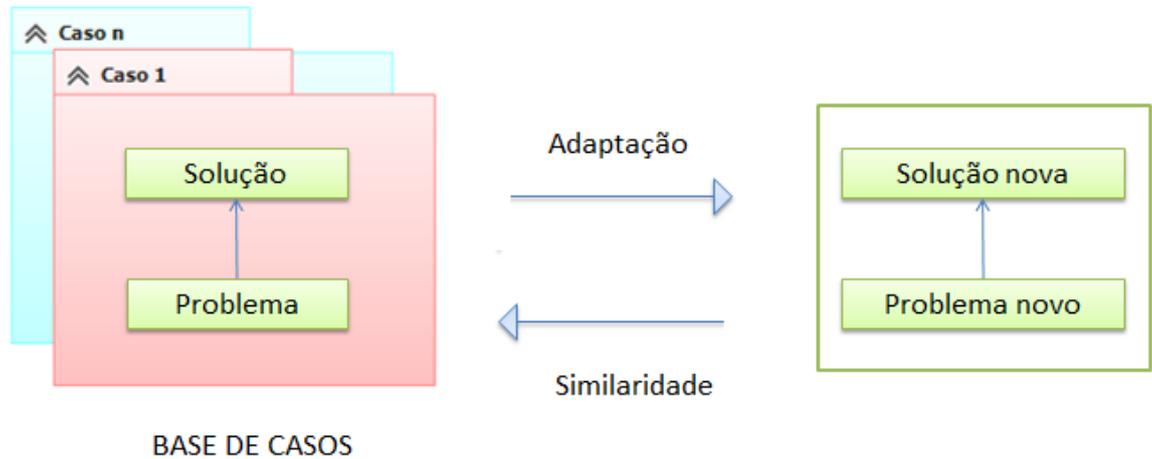


Figura 1 - Modelo básico do enfoque RBC
 Fonte: Adaptado de WANGENHEIM e WANGENHEIM (2003)

2.1.2. Ciclo de RBC

Um sistema RBC deve possuir a capacidade de indagar seu usuário, usando uma linguagem de fácil entendimento, possibilitando desenvolver uma linha de raciocínio a partir das informações repassadas e do conhecimento nele implantado.

O padrão mais aceito para o processo de RBC é o Ciclo de RBC proposto por Aamondt e Plaza (1994), que engloba um ciclo de raciocínio contínuo composto por quatro tarefas principais:

- **Recuperação:** na fase de recuperação, procuram-se, na base de casos, os casos mais semelhantes com o problema em questão, sendo necessário, para esse tipo de pesquisa, um método de avaliação de similaridade entre os casos da base de casos;
- **Reutilização:** sendo identificado o caso que possua mais similaridade, é dado que a solução encontrada pode ser útil para resolver o problema em questão;
- **Revisão:** nesta fase, são feitos ajustes em relação às eventuais diferenças que possam existir entre a solução encontrada e o problema. Sendo necessária uma adaptação da solução encontrada, esta é realizada através de simulações e assemelha-se com o processo cognitivo humano, procurando confirmação da solução encontrada.
- **Retenção:** é feito o armazenamento do caso ou partes interessantes do novo problema para reutilização, sendo, através da retenção, que se têm os dados para a ampliação da base de conhecimento, onde já se tem a contemplação das adaptações realizadas.

Este ciclo é representado na Figura 2. O problema é pesquisado na base de casos, buscando-se as tarefas executadas anteriormente (que foram resolvidas com sucesso), cujas descrições sejam similares á do problema em questão. Uma vez tendo o caso recuperado, é realizada uma tentativa de reutilizá-lo para resolver o problema proposto. Normalmente o caso não é igual ao problema em questão, podendo ser necessário realizar uma adaptação do mesmo.

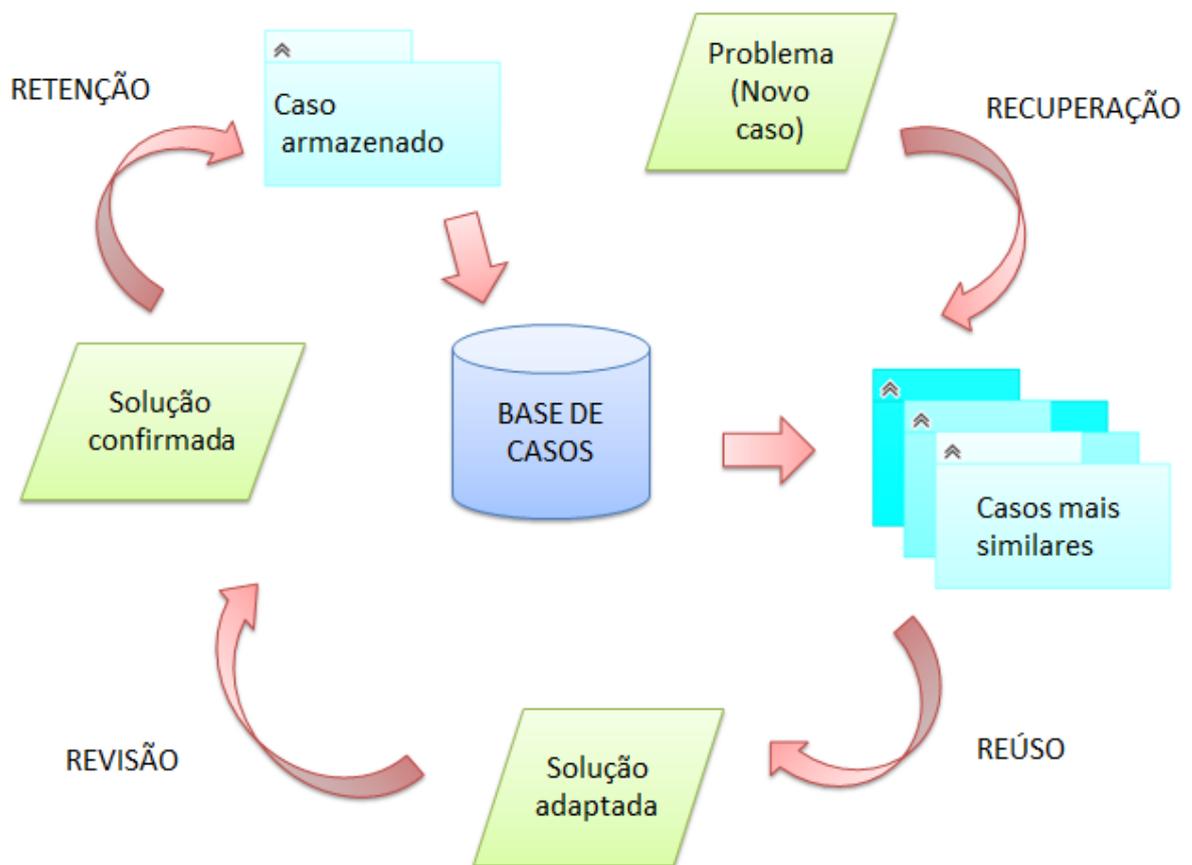


Figura 2 - Ciclo do RBC

Fonte: Adaptado de WANGENHEIM e WANGENHEIM (2003).

Após, é necessário realizar uma validação (fase de revisão), onde é feita uma análise para verificar se a solução proposta não possui falhas. Não havendo falhas, passa para a fase de retenção, onde o caso com solução positiva é armazenado na base de casos e ocorre o aprendizado do sistema. Cada caso é composto por: uma descrição dos aspectos relevantes do problema que caracterizam a situação particular a ser resolvida, o contexto no qual o caso se insere, a descrição da solução associada ao problema na forma de um diagnóstico, uma classificação e uma sequência de ações e uma avaliação da solução empregada ao problema em questão.

2.1.3. Base de Casos

Para os casos serem reutilizados é necessário que estes estejam à disposição em uma Base de Casos (BC), que é um conjunto de casos organizados, onde geralmente é formada por experiências que obtiveram resultados positivos e que descrevem as estratégias de solução que contribuem com informações para resolver o problema descrito com o maior nível de sucesso, podendo ser reutilizadas no futuro em outro problema semelhante. Experiências com resultados negativos podem também serem armazenadas, com o intuito de sinalizar potenciais problemas e prevenir a repetição de erros passados (VITORINO, 2009).

Na Figura 3 é dado um exemplo simplificado de uma base de casos na qual estão armazenados somente dois exemplos bem sucedidos referentes a reparos de impressoras no passado, onde são descritos os requisitos, o problema e a solução.

Caso 1		
Problema / Sintomas	Problema:	Impressora não funciona
	Modelo:	Robotron Matrix 600
	Luz de estado do papel:	apagada
	Luz de estado da tinta colorida:	apagada
	Luz de estado da tinta preta:	apagada
Solução	Diagnóstico	Curto-circuito
	Ação:	Troca de fonte de alimentação

Caso 2		
Problema / Sintomas	Problema:	Não imprime em preto
	Modelo:	Robotron Matrix 800
	Luz de estado do papel:	apagada
	Luz de estado da tinta colorida:	apagada
	Luz de estado da tinta preta:	aceso
Solução	Diagnóstico:	Cartucho de tinta preta vazio
	Ação:	Trocar cartucho de tinta preta

Figura 3 - Exemplo simplificado de uma base de casos
Fonte: Adaptado de WANGENHEIM e WANGENHEIM (2003).

2.1.4. Desenvolvimento de um sistema RBC

Tendo disponível uma base de dados para ser investigada, torna-se possível extrair o conhecimento necessário para ser aplicado na tomada de novas decisões. O desenvolvimento

de um sistema RBC a partir da base de casos passa pelas definições de técnicas e das formas da implementação de cada componente do sistema.

De forma geral um sistema baseado em conhecimento, como o RBC, possui dois componentes essenciais para seu desenvolvimento: uma base de casos que contenha o conhecimento específico do domínio e um motor de inferência consistindo de um ou mais algoritmos responsáveis por manipular o conhecimento da BC.

Um sistema de RBC de qualidade depende de diversos fatores, entre eles pode-se citar o número de casos relevantes, contidos na base de casos, onde quanto maior o número de casos maior será o conhecimento ou experiência do sistema e também deve ser considerada a capacidade do sistema compreender e aprender com novos casos, levando em conta a capacidade de avaliação e adaptação da solução.

De acordo com Lorenzi (1998), a construção de um sistema RBC passa pelos seguintes passos:

- Seleção das informações que farão parte da base de casos; quanto mais completa e correta forem as informações contidas na base de casos, o sistema se tornará mais confiável e mais facilmente desenvolvido;
- Definições de quais atributos das informações contidas na BC serão mais relevantes e que devam ser utilizadas para a solução do problema;
- Definição dos índices para os casos, tornando possível a recuperação quando necessário. O passo de indexação é muito importante para a construção de um bom sistema RBC, onde deve ser decidido o que armazenar em um novo caso, encontrando uma estrutura apropriada para a descrição dos conteúdos dos casos e decidir como organizar o indexador da memória de casos;
- Definição dos métodos de recuperação de casos para verificação da similaridade entre os casos da base e o novo problema;
- Definição da forma como os casos recuperados serão adaptados para solucionar o novo problema;
- Definição do processo de aprendizagem do sistema, que indica se a solução adaptada teve sucesso, sendo necessário que o sistema avalie se o caso deve ou não ser armazenado. Normalmente, somente casos que tenham relevância para a solução são armazenados,

evitando redundância de informações, mas pode haver o armazenamento de casos não resolvidos que ajudam na aprendizagem do sistema.

2.1.5. Aquisição e representação dos casos

A representação do conhecimento é essencial em sistemas RBC e esse conhecimento é, principalmente, armazenado no formato de casos. Um caso é uma peça de conhecimento contextualizado representando uma experiência ou episódio concretos, e contém a lição passada, que é o conteúdo do caso e o contexto em que a lição pode ser usada (WANGENHEIM e WANGENHEIM, 2003). Um caso representa um conhecimento específico que tem relação direta com uma situação em determinado contexto.

Mais especificamente, um caso é a definição de quais atributos da informação contida são relevantes e podem ser utilizados para a resolução de questões específicas. Geralmente, a representação de casos consiste de três componentes básicos (Figura 4) que são uma situação (problema), que contém a descrição dos aspectos relevantes do problema em questão e que caracterizam uma situação particular, em conjunto com as experiências adquiridas (solução) e uma avaliação da solução utilizada para determinado problema (conclusão).

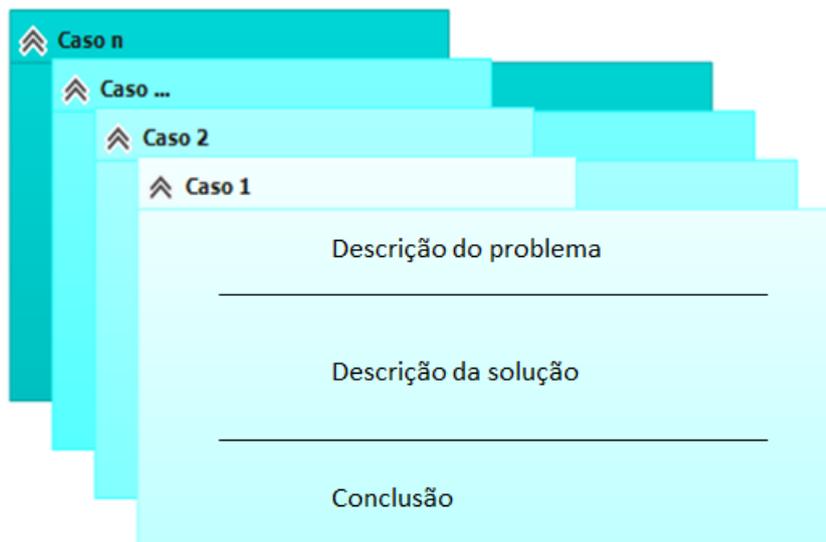


Figura 4 - Componentes básicos na representação dos casos
Fonte: Adaptado de WANGENHEIM e WANGENHEIM (2003).

2.1.6. Indexação dos casos

Como sistemas baseados em RBC dependem de sua habilidade de recuperação de casos relevantes, em uma biblioteca de casos, é imprescindível saber como essa biblioteca deve ser indexada de maneira que o processo de recuperação seja o mais preciso e eficiente.

Indexar casos corresponde a atribuir índices aos casos de forma a facilitar sua recuperação. Isso inclui colocar os devidos rótulos nos casos, quando estes forem acrescentados a base de casos, para que possam ser posteriormente recuperados. A organização dos casos para facilitar a sua busca e recuperação de maneira eficaz vai depender do algoritmo de recuperação definido.

Segundo Kolodner (1993) as qualidades necessárias para se ter bons índices são:

- Prever a futura utilização da informação para solução de diferentes problemas;
- Endereçar similaridades úteis entre os casos;
- Ser abstrato o suficiente para se tornar um caso útil em uma variedade diferente de situações;
- Ser concreto o suficiente para ser facilmente reconhecido em futuras situações.

Sendo possível a seleção de índices tanto manualmente quanto automaticamente. A diferença entre as duas formas está, respectivamente, em analisar caso a caso para determinar quais as características descritas que determinam as variações sobre as conclusões e quantificar as diferenças entre os casos e os relacionamentos entre feições do problema e soluções adotadas.

Segundo Kolodner (1993), mesmo que os métodos automatizados auxiliem na escolha de bons índices, na prática, os sistemas cujos índices forem definidos como manuais tendem a ter melhor desempenho, já que o processo de indexação dos casos sempre procura retornar os casos com melhores resultados.

2.1.7. Recuperação dos casos

O objetivo da recuperação de casos em um sistema RBC seria de que, dada uma descrição de um problema, um algoritmo deve recuperar os casos mais similares, ou um pequeno conjunto de casos, que se adequem à solução da situação atual.

Esta etapa talvez seja a etapa mais importante e mais custosa do RBC, pois, para que um novo problema seja solucionado, primeiramente deve ser encontrado um conjunto de casos relevantes que possam obter uma solução para o problema. Isso significa que, além da questão da qualidade da solução encontrada, a eficiência do processo é um ponto chave em um bom sistema de RBC.

A seguir são descritos alguns métodos para recuperação de casos, que foram propostos para o desenvolvimento de sistemas RBC e que podem ser utilizados sozinhos ou combinados, citado por ABEL (1996):

- **Algoritmo de vizinhança (Nearest Neighbour):** baseia-se na comparação entre um novo caso e aqueles já armazenados, baseado no somatório de pesos das características do novo problema. É necessário atribuir um peso a cada uma das feições que descrevem o caso e que serão utilizadas para a recuperação. Este método se torna mais eficaz em base de casos relativamente pequenos, pois o tempo de recuperação está ligado ao número de casos que compõem a base de casos;
- **Algoritmo de indução:** determina quais as feições são mais eficazes em diferenciar casos e se utiliza dessas feições para gerar uma árvore de decisão para organização da memória de casos. A árvore resultante possui tempo de recuperação mais rápido do que o método de recuperação do algoritmo de vizinhança e torna-se o melhor método quando a meta da recuperação é bem definida;
- **Algoritmo de recuperação baseada em conhecimento:** utiliza o conhecimento da aplicação para identificar manualmente as características dos casos que são conhecidas. Semelhante a sistemas especialistas baseados em regras, onde o conhecimento não precisa ser completo e, por isso, normalmente é utilizado em conjunto com outros métodos;
- **Recuperação de padrões:** utiliza métodos similares às consultas SQL de banco de dados, recuperando casos que se assemelham em certas características pré-definidas. São muito eficientes, até mesmo com grandes volumes de dados, mas muito pobres em relação à seleção do melhor caso, sendo necessária a utilização em conjunto com outros métodos de recuperação.

2.1.8. Reutilização ou adaptação dos casos

A reutilização dos casos também é chamada de adaptação de casos e é a tarefa final de um sistema RBC. Essa etapa prevê utilizar casos recuperados na solução do novo

problema. Como casos já existentes dificilmente serão idênticos aos do problema atual, assim é necessária a adaptação, presente em sistemas RBC mais avançados e que possuem mecanismos e conhecimento necessário para adaptar os casos recuperados e verificar se eles satisfazem as características necessárias para a situação presente.

Para a reutilização de casos, onde há basicamente a cópia do caso inserido na base de casos, é necessário que as similaridades sejam consideradas relevantes e, assim, a solução do caso recuperado é transferida para o novo caso como sua solução. Já a adaptação consiste de reutilização do caso mais similar com a correção de problemas, uma vez que o caso possa não satisfazer completamente os requisitos dados pela nova situação.

Conforme Vitorino (2009) são comuns na adaptação de casos:

- **Adaptação estrutural:** onde as regras de adaptação são aplicadas sobre a solução armazenada junto aos casos, reutilizando situações de casos passados;
- **Adaptação derivacional:** reutiliza os algoritmos, métodos ou regras que deram origem a solução presente na base de casos para gerar uma nova solução para o problema corrente;
- **Adaptação nula** também é bastante utilizada e ela basicamente apenas aplica a solução de um caso recuperado para uma nova situação.

2.1.9. Vantagens e Limitações

Diferentemente das demais técnicas da IA, os sistemas RBC, ao invés de basearem-se unicamente no conhecimento de domínio geral de um problema dado, é capaz de utilizar formas explícitas de conhecimento específico das situações concretas que foram executadas anteriormente na forma de casos.

O uso dessa metodologia em ambientes de aprendizado tem a fundamentação de uma ampla teoria cognitiva, como o processo de lembrar, onde o processo de reutilizar situações já vividas corresponde diretamente à forma de raciocínio humano.

As principais vantagens na forma do desenvolvimento de sistemas RBC para ambientes de apoio à decisão são:

- não é necessária uma modelagem explícita do domínio;
- a implementação é reduzida a identificar as características significantes que descrevem um caso dado;

- pode ser gerenciado grande volume de informação de forma eficaz, bastando ter um banco de dados;
- a atualização do conhecimento é feita de maneira automática, de acordo com a experiência obtida através da base de casos, surgindo daí a forma de aprendizagem do sistema;
- como a BC é gerada a partir das experiências passadas, os casos retornados tendem a ser consistentes com as soluções obtidas, o que representa uma proximidade bastante significativa com o comportamento humano;
- a difícil extração do conhecimento especialista é facilitada em sistemas RBC, pois a representação do conhecimento pode ser feita com o preenchimento direto dos fatos que descrevem uma experiência.

Sistemas baseados em conhecimento se diferem dos sistemas convencionais em como são organizados, da maneira como incorporam o conhecimento, como são executados e na impressão que causam ao usuário com os quais interagem, conforme a Tabela 1 (REZENDE, 2003):

Tabela 1 - Principais diferenças entre os sistemas convencionais e os sistemas baseados em conhecimento

Sistemas convencionais	Sistemas baseados em conhecimento
Estrutura de dados	Representação do conhecimento
Dados e relações entre dados	Conceitos, relações entre conceitos e regras.
Tipicamente usa algoritmos determinísticos	Busca heurística
Conhecimento embutido no código do programa	Conhecimento representado explicitamente e separado do programa que o manipula e o interpreta
Explicação do raciocínio é difícil	Podem e devem explicar o raciocínio.

Fonte: Adaptado de REZENDE (2003)

A limitação da utilização dessa técnica fica apenas na questão ao acesso a bases de dados completas e confiáveis, que possuam entre as informações cadastradas, as descrições completas do problema e as soluções obtidas que foram aplicadas em algum momento.

2.1.10. Trabalhos relacionados

Não são poucas as aplicações que utilizam sistemas RBC. Lorenzi (1998) criou uma aplicação para controle de autorizações de Internação Hospitalar, onde o sistema de RBC tem a capacidade de bloquear ou não o pagamento das internações hospitalares de pacientes que contenham múltiplas internações no período do relatório emitido. Partindo da ideia de que múltiplas internações muitas vezes escondem irregularidades ou falhas do atendimento médico, identificando setores que talvez devam ser investigados.

Já Pilger e Hugo (2012) desenvolveram uma aplicação de RBC no suporte a decisão de um sistema *web* de *help desk*, que foi aplicado no seu domínio técnico em uma empresa desenvolvedora de aplicações de automação comercial e foi desenvolvido com o foco de auxiliar no suporte a decisão dos técnicos envolvidos com o atendimento aos clientes.

O Projeto IACVIRTUAL, proposto por Silva et al. (2005) tem a ideia de um sistema RBC que simula um consultório virtual via *web*, concebido para atender pacientes que possam ter interesse em acompanhar seu histórico médico, especialistas que Figura 5 é possível observar a visão geral do modelo RBC aplicado no projeto IACVIRTUAL.

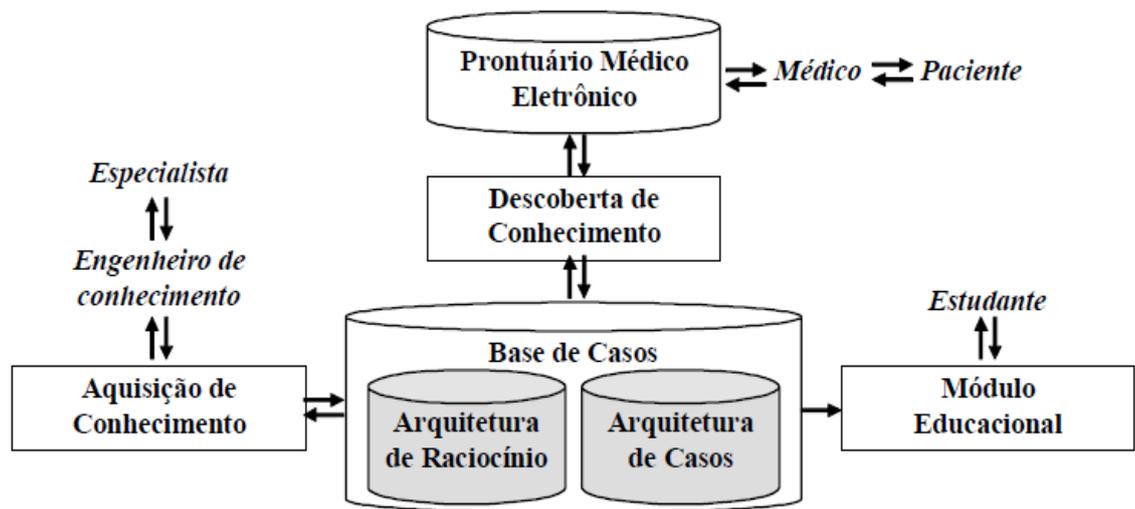


Figura 5 - Visão Geral do modelo RBC no projeto IACVIRTUAL

Fonte: SILVA et al. (2005).

NEWTON, SCHUTZ e SPIEKER (2004) desenvolveram um sistema especialista baseado em regras que permite auxiliar o profissional veterinário no atendimento de animais com dermatose, no processo de reunir o máximo de informações. Essas informações permitem criar um quadro patológico no qual o profissional baseia-se para efetuar o

diagnóstico da doença e assim poder identificar e elaborar o tratamento mais adequado, levando em consideração os tratamentos contidos no receituário.

Analisando os trabalhos citados, pode-se concluir que existem vantagens e desvantagens na utilização da técnica de RBC. Em alguns casos a utilização de experiências passadas do RBC pode ser o melhor e único caminho para a aquisição e representação de conhecimento desses domínios. Por outro lado a utilização em domínios muito complexos torna a tarefa de extração de informação relevante bastante difícil.

2.2. Sistemas de Recomendação

Entre as diversas áreas da IA estão os Sistemas de Recomendação, que têm como foco a identificação de usuários, armazenamento de suas preferencias e recomendação de itens que podem ser úteis a eles, tal como produtos, serviços e/ou conteúdos relevantes aos seus interesses e necessidades.

Esse tipo de sistema surgiu para auxiliar o usuário a obter a melhor escolha de compra de produtos e para ajudar as empresas de comércio eletrônico a sugerir produtos específicos a cada usuário de forma personalizada. Segundo Torres (2004), a utilização de SR gera um grande contentamento perante os usuários do sistema, já que o usuário sente-se único e reconhecido.

Sistemas de Recomendação consideram basicamente o perfil do usuário, as informações de outros usuários com informações similares e o histórico resultante das ações do grupo de usuários. A importância desse tipo de sistema fica clara principalmente nas situações a seguir:

- Quando o número de informação e resultados é grande demais ou desconhecido;
- Quando o usuário não tem o conhecimento necessário ou experiência suficiente para efetuar as buscas e escolher entre as informações disponíveis;
- Quando se tem acesso a registros de comportamento de outros usuários com perfis semelhantes.

Sistemas de Recomendação têm o poder de automatizar total ou parcialmente a busca pela informação necessária e relevante àquele usuário que requisita, com a característica de

poder aplicar a personalização de acordo com o usuário que maneja o sistema, podendo recomendar serviços, produtos e informações relevantes.

Sendo uma das áreas de estudo mais importantes da IA, se tornou muito usual com a popularização da *internet*, onde suas técnicas de recomendação se tornaram altamente eficazes, já que nesse meio se tem uma quantidade gigantesca de dados para a pesquisa.

Segundo SCHAFFER e KONSTAN (2000), a estrutura dos sistemas de recomendação é dividida em quatro processos: identificação do usuário, coleta de informações, estratégias de recomendação e visualização das recomendações obtidas. A Figura 6 mostra este modelo.

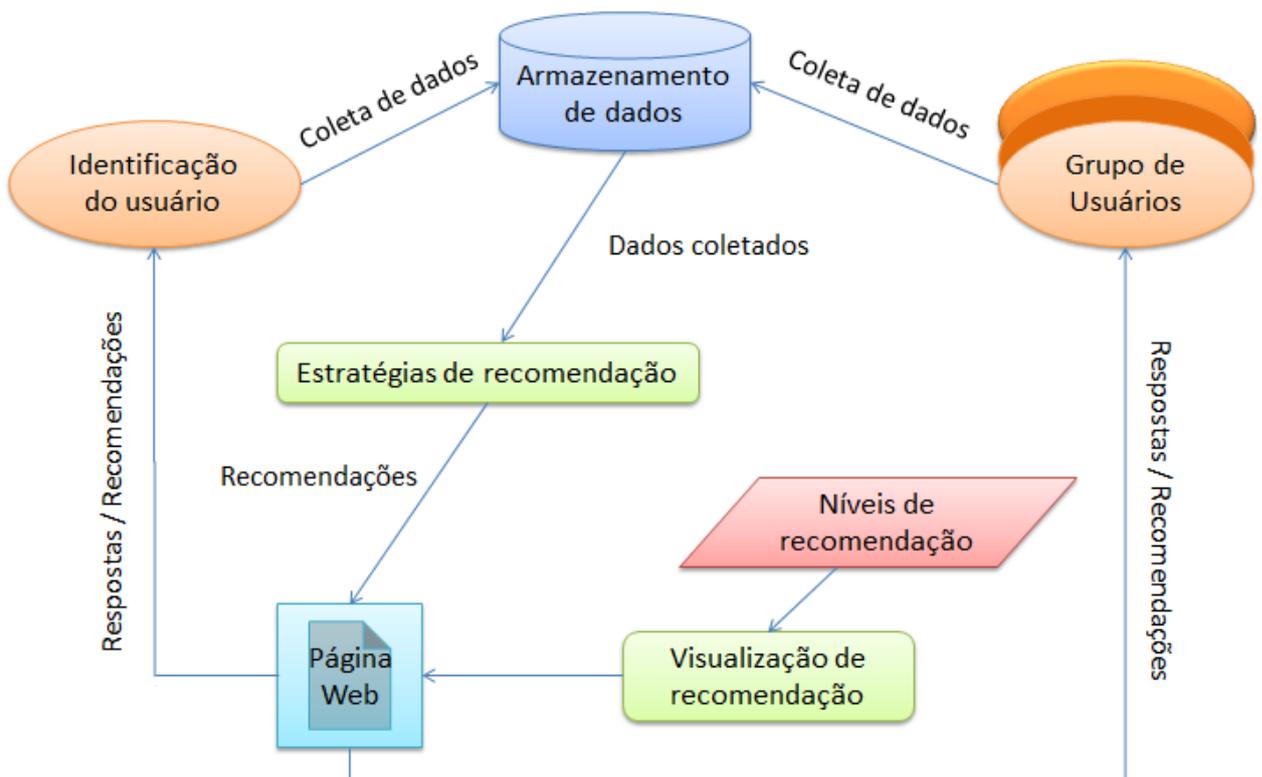


Figura 6 - Estrutura de um Sistema de recomendação

Fonte: Adaptado de SCHAFFER e KONSTAN (2000).

A identificação de um usuário em um sistema deste tipo não necessariamente é obrigatória, mas quando se utiliza um processo de identificação torna-se altamente eficaz e caracteriza a primeira etapa de coleta de dado. Uma vez coletados os dados, estes são armazenados em uma base de dados, para posteriormente serem utilizados nas pesquisas envolvendo o usuário e criar as recomendações de acordo com os dados retidos no BC.

Para ser possível tornar o conteúdo personalizável de acordo com o usuário existem basicamente duas formas de identificação de usuário, por meio de *cookie*, quando se tratando

de páginas *web* ou o *login*. Quando a identificação é feita por *cookie*, há um reconhecimento automático do usuário, e quando é feita através de um *login* é necessário que se informe os dados de *login* e é feita uma autenticação no BD.

2.2.1. Arquitetura e estrutura

Sabe-se que um SR deve ter a capacidade de recuperar as preferências informadas e também as subentendidas, conteúdos ou serviços relevantes para cada usuário, que possuam relação a itens que já se tenha mostrado interesse no passado. Esse tipo de sistema deve permanecer em comunicação com o servidor, que exhibe os resultados de recomendação para o usuário.

Para a execução de sistemas de recomendação existem dois principais tipos de arquiteturas responsáveis por seu funcionamento (REATEGUI e CAZELLA, 2005).

O primeiro é baseado na técnica de filtragem de conteúdo. O sistema executa a pesquisa através da análise direta dos dados ao usuário e diferentes estratégias de recomendação são armazenados para permitir que diferentes tipos de sugestões sejam apresentados. O servidor faz a requisição ao SR, que retorna a recomendação mais relevante, como pode ser visto na Figura 7.

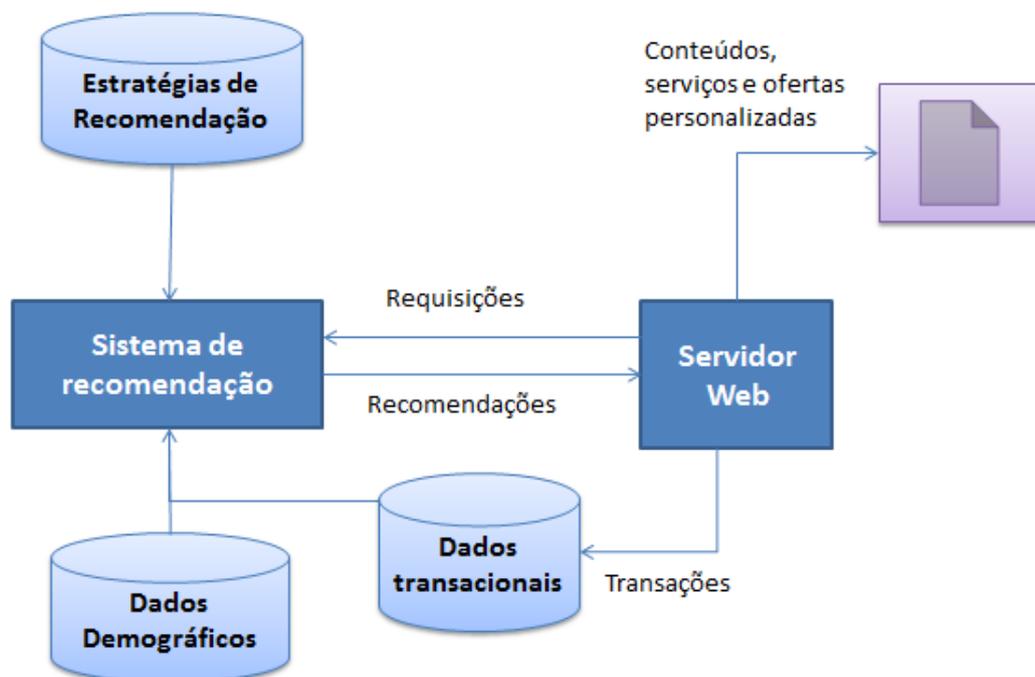


Figura 7 - Estrutura de SR baseado em técnicas de filtragem de conteúdo
 Fonte: Adaptado de REATEGUI e CAZELLA (2005)

Já no segundo tipo de arquitetura, existe a mineração de dados que se torna responsável pela localização de padrões dos dados do usuário. Depois de identificados, os padrões são armazenados em uma base de conhecimento, onde também é possível o armazenamento de diferentes estratégias de recomendação. O servidor *web* é responsável por requisitar ao SR, que retorna as recomendações relevantes sem a inferência direta do sistema de mineração, conforme a Figura 8.

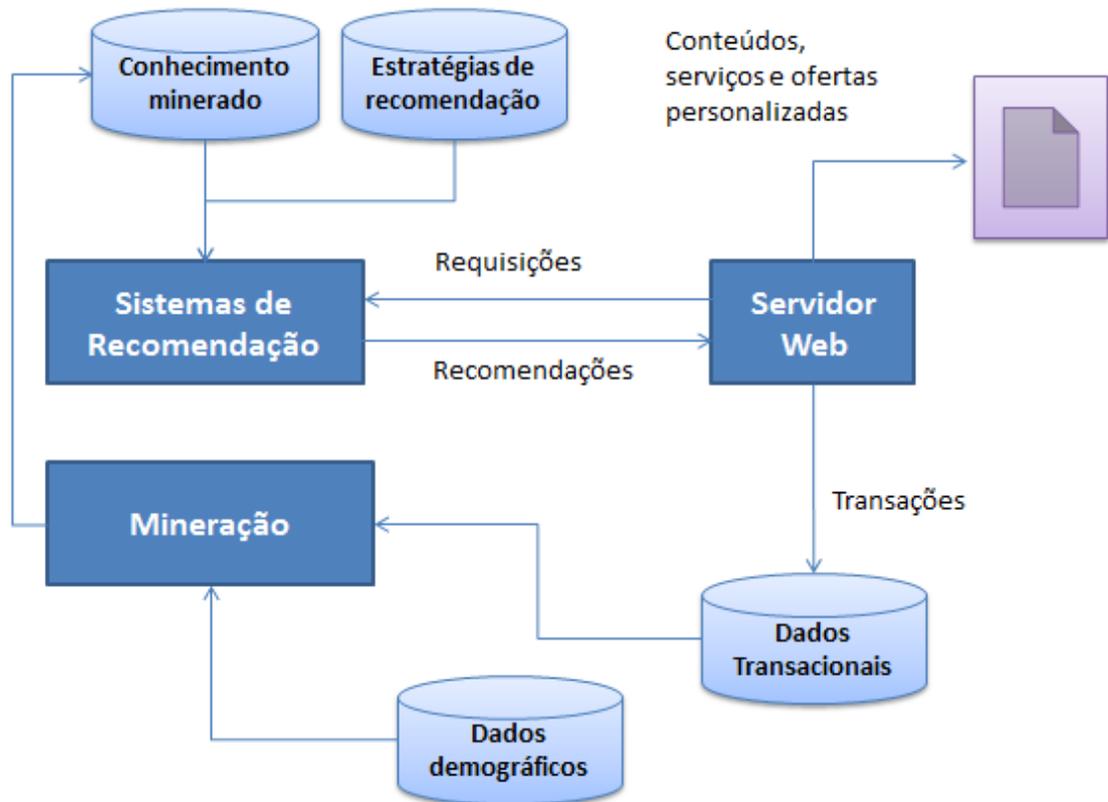


Figura 8 - Estrutura de SR com aplicação de Mineração de Dados

Fonte: Adaptado de REATEGUI e CAZELLA (2005)

2.2.2. Taxonomia

SCHAFFER e KONSTAN (2000) desenvolveu uma taxonomia, que pode ser visualizada na Figura 9, para classificação da arquitetura de um SR onde a arquitetura base é envolta em basicamente três blocos bem distintos e complexos, havendo uma divisão necessária, composta por: Entrada, Saída, Entrega, Grau de Personalização e os Métodos de Recomendação.

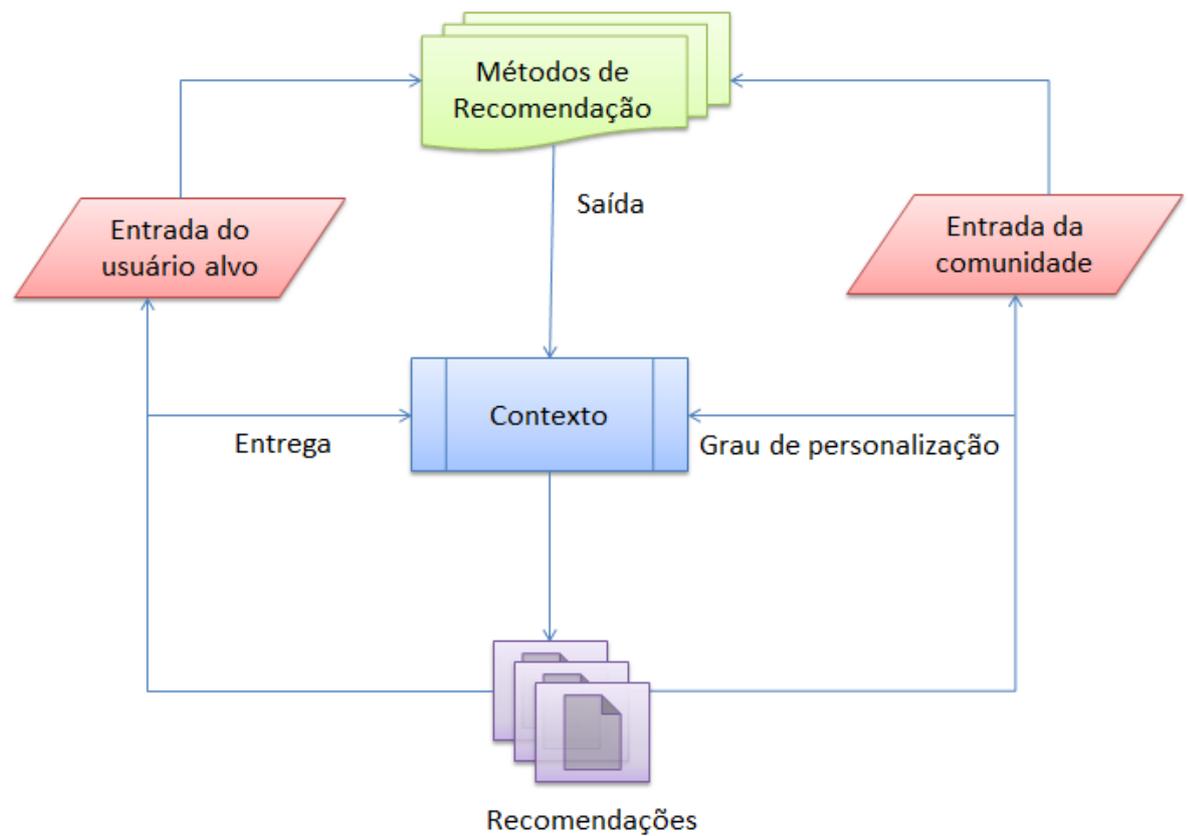


Figura 9 - Taxonomia de um SR
 Fonte: Adaptado de FILHO (2006)

2.2.3. Métodos de recomendação

Métodos de recomendação utilizam diferentes formas para retornar recomendações aos usuários através das informações retidas na base coletadas através de ações de usuários. Entre elas estão: seleção manual, resumos estatísticos, recuperação bruta, correlação usuário para usuário, baseado em atributos e correlação item para item:

- **Seleção manual:** editores ou especialistas que têm conhecimento do contexto e seu público alvo recomendam manualmente. Método bastante simples e muito trabalhoso e é necessário conhecimento especialista;
- **Resumos estatísticos:** para complementar o método de seleção manual, onde existe a necessidade de um especialista, que nem sempre tem o conhecimento do contexto, existe o método de resumos estatísticos que geram índices de satisfação do usuário;
- **Recuperação bruta:** sistemas que utilizam esse método têm a necessidade de uma interface simples com busca por palavras chaves, onde é retornada uma lista com todos os itens relacionados com a palavra pesquisada;

- **Correlação usuário para usuário:** esse método utiliza todos os dados coletados dos usuários que têm gostos similares e recomenda serviços/produtos de acordo com as preferências em comum do usuário alvo com outros usuários;
- **Baseado em atributos:** recomenda itens baseados em suas características;
- **Correlação item para item:** faz tentativas de identificar itens que possuam relação com itens que os usuários demonstraram interesse no seu histórico de ações.

2.2.4. Entrada do Usuário Alvo e Entradas da Comunidade

Para haver a personalização do usuário e de seus resultados, é necessário que seja informado os dados do usuário. Essa é a base de um sistema de recomendação, onde quanto mais informações coletadas sobre o usuário, mais será efetivo a recomendação dada pelo sistema em relação ao seu contexto.

As formas de coleta do usuário alvo são implícita e explícita, onde as informações implícitas são recuperadas de forma que o usuário em questão não note que suas informações estão sendo coletadas, principalmente através da navegação do usuário em conjunto com ações realizadas no ambiente ou o histórico de ações do usuário, como por exemplo, o seu histórico de compras em uma loja virtual. Já, na forma explícita, as informações são dadas diretamente pelo usuário, que informa o quão útil foi o serviço/produto recomendado e o quão satisfeito ficou, sendo uma forma mais confiável de coleta de dados já que o próprio usuário informa os dados.

Grande parte dos SR se utiliza da combinação de coleta de dados implícita e explícita para gerar dados mais confiáveis, já que implicitamente nem sempre se tem os dados confiáveis e exatos e da forma explícita nem sempre o usuário está disposto a colaborar de forma eficaz com o sistema.

A forma de coleta da comunidade consiste da coleta de informações de todos os usuários que interagem de alguma forma com o ambiente, através de comentários sobre serviços/produtos relacionados ao ambiente, avaliações diretas através de notas numéricas ou através da popularidade de um serviço/produtos, sendo que esta última é muito utilizada quando ainda não se tem informações suficientes de um usuário.

2.2.5. Saída

Representa a resposta que o sistema retorna ao usuário, onde geralmente o sistema recomenda um serviço/produto ao usuário retonados através dos métodos de recomendação. No exemplo de lojas virtuais, será recomendado um produto de interesse do usuário.

2.2.6. Entrega

A entrega consiste na estratégia de personalização definida e pode ser classificada de três maneiras: *push*, *pull* e orgânica. Por *Push* é feita recomendação de serviços/produtos ao usuário sem a requisição do usuário por recomendação, sendo que ele não necessita estar interagindo com o contexto para que seja dada uma recomendação a ele. O *pull* é capaz de conceder certo controle ao usuário para escolha de quando será oferecida uma recomendação a ele. A forma orgânica informa as recomendações ao usuário de forma não tão visível, sem que o usuário receba avisos e o mesmo não possui controle sobre a requisição da recomendação.

2.2.7. Grau de personalização

O grande diferencial de sistemas de recomendação na visão do usuário é que ele se sente exclusivo e como se houvesse um tratamento exclusivo somente para ele. O grau de personalização, Tabela 2, de um SR pode ser definido em três níveis: não personalizado, quando o sistema gera recomendações iguais para todos os usuários não levando em conta as informações pessoais do usuário; sistema efêmero, quando é levado em consideração as necessidades e ações momentâneas do usuário; e, ainda, o sistema personalizado que leva em consideração todas as informações retidas do usuário para criar um ambiente e recomendações totalmente personalizadas.

Tabela 2 - Grau de personalização e seus métodos de recomendação

Grau de personalização	Método de recomendação
Não personalizada	Seleção manual, resumos estatísticos.
Efêmera	Item para item, baseada em atributo.
Personalizada	Usuário para usuário

Fonte: Adaptado de FILHO (2006)

2.2.8. Técnicas de Sistemas de Recomendação

Ao longo dos estudos na área da Inteligência Artificial, muitas tecnologias surgiram para auxiliar na identificação de padrões de comportamento e a utilização desses padrões é aplicada na personalização do relacionamento direto com o usuário. Muitas técnicas se fundamentaram para o funcionamento dos sistemas de recomendação voltados para a filtragem de informação, entre as principais pode-se citar a filtragem colaborativa, a filtragem baseada em conteúdo e a filtragem híbrida.

- ***Filtragem Colaborativa***

A técnica de filtragem colaborativa parte da ideia de que usuários requerem recomendações sobre determinado produto ou serviço de seu interesse a outros usuários que já tenham utilizado esses produtos ou serviço e tenham gostos similares, e pode-se dizer que desse modo há uma troca de experiências entre os usuários que têm interesse em comum.

Nestes sistemas, os itens são filtrados com base nas avaliações feitas pelos usuários. Sistemas de filtragem colaborativa menos sofisticados apresentam para o usuário uma média de pontuação para cada item que tem interesse em potencial de aquisição, onde a pontuação dada vai permitir que o usuário identifique itens que são considerados de grande interesse pelo grupo e evitar a exibição de itens que não provoquem interesse. Já sistemas mais arrojados permitem que a descoberta de relações entre usuários e seus gostos similares seja automática, encontrando padrões comuns de comportamento.

Essa técnica possui algumas vantagens, sendo uma delas a possibilidade de apresentar ao usuário recomendações inesperadas, onde o usuário recebe recomendação de algum item que não estava sendo pesquisado. Outra vantagem importante na utilização da filtragem colaborativa é referente à possibilidade de formação de comunidades de usuários, ou grupos de usuários, pela identificação de seus interesses em comum.

- ***Filtragem Baseada em Conteúdo***

Essa técnica, ao contrário da filtragem colaborativa, tem a capacidade de fazer comparações de similaridade entre o usuário e o item pesquisado, sendo possível a recomendação mais similar com o perfil do usuário.

Sistemas que utilizam a técnica de filtragem baseada em conteúdo se tornam mais complexos aos sistemas de recuperação de informação. Se dá através de uma consulta por determinada palavra chave, que efetua uma busca na base de dados, e o sistema apresenta como resposta para o usuário os itens que contenham em seu conteúdo a palavra chave buscada.

Como principais vantagens da utilização dessa técnica estão a capacidade de recomendação de todos os itens, já que todos os produtos da base são comparados com o perfil do usuário, e que não há necessidade de que o usuário avalie o item para gerar futuras recomendações, não gerando o problema do primeiro avaliador.

Essa técnica possui duas desvantagens que devem ser levadas em consideração na elaboração de sistemas de recomendação. Uma delas é de que não há como saber se o resultado da pesquisa e a recomendação gerada foi útil para o usuário, pois, ao contrário da filtragem colaborativa o usuário não deu sua opinião sobre. O outro grande problema parte do fato de que é impossível a recomendação de novos produtos, já que essa técnica se baseia nos produtos já consumidos pelo usuário e da comparação das palavras chaves dos produtos e encontradas no perfil do usuário.

- ***Filtragem Híbrida***

Na filtragem híbrida é necessária a combinação das duas técnicas de filtragem de informação citadas anteriormente, onde uma supre a desvantagem da outra para minimizar as limitações individuais de cada técnica.

O único problema que pode ser encontrado é que, com a complementação da filtragem colaborativa, não se consegue sanar o chamado *start-up*, ou seja, tempo que o sistema leva para encontrar o conteúdo relevante à pesquisa do usuário e gerar boas recomendações.

2.2.9. Vantagens e Desvantagens

Grande parte das aplicações desenvolvidas com SR se referem à personalização de *web sites* e a possibilidade de tornar a recomendação de produtos ou serviços extremamente otimizada. No entanto os sistemas de recomendação deixaram de ser utilizados apenas em páginas *web* e comércio eletrônico para ganhar mercado em outras áreas, como pode ser visto nos trabalhos relacionados na seção 2.2.10.

As principais vantagens da utilização de aplicações com personalização de acordo com a utilização do usuário vêm desde a redução do tempo de atendimento e pesquisa até o aumento na satisfação do cliente, já que é possível fazer com que o cliente sinta-se único. Esse tipo de sistema tem o poder de automatizar o processo de busca de informação relevante ao que o usuário necessita. E pode-se citar como maior vantagem a facilidade de implementação, bastando-se apenas uma base de dados para efetuar a pesquisa dos dados.

A maior desvantagem e o grande problema da utilização de SR ainda é referente aos usuários, já que são estes que são os maiores responsáveis pelos dados contidos na base, que serão utilizados para se efetuar as recomendações, de acordo com interesses relevantes ao grupo de usuários utilizadores do ambiente.

2.2.10. Trabalhos relacionados

Diversos são os trabalhos em que se utilizam os Sistemas de Recomendação. Alguns deles são bem conhecidos na área da Inteligência Artificial e são descritos a seguir.

O sistema e-Recommender de Macário (2006), tem como ideia principal a recomendação de produtos de uma loja virtual baseando-se nos produtos comprados pelo usuário. É um sistema de Filtragem Híbrida que possui a filtragem baseada em conteúdo para entrada no algoritmo de filtragem colaborativa.

Já Cazella, Chagas e Barbosa (2008) desenvolveram um modelo para recomendação de artigos acadêmicos baseado em filtragem colaborativa em ambientes móveis, aplicado em um protótipo visando a recomendação relevante de artigos científicos para seu aprendizado, onde eram recebidas avaliações dos usuários pelas recomendações geradas.

Sibaldo, Sales e Calado (2007) desenvolveram um sistema para aplicativos móveis chamado Mobile GraW, que é utilizado pela Universidade Federal de Alagoas para aumentar e melhorar a interação na área acadêmica através de ferramentas. Este sistema é uma extensão para dispositivos móveis do sistema baseado em comunidades virtuais GraW – Graduação na Web. Tal aplicação faz uso de sistema de recomendação de usuários devido à necessidade de encontrar potenciais usuários que auxiliem no processo de ensino/aprendizagem.

Utilizando sistemas de recomendação sensível ao contexto, Silva, Neto e Junior (2011) desenvolveram uma aplicação chamada *MobiLE*, que tem como ideia um ambiente multiagente de aprendizagem móvel para apoiar a recomendação sensível ao contexto de

objetos de aprendizagem. É proposto um ambiente de aprendizagem móvel, que através de objetos de aprendizagem, ontologias e agentes de *software*, se adequa as necessidades dos estudantes, de acordo com as características do contexto onde se encontram. O *software* foi desenvolvido em Java, com o *framework* de desenvolvimento de aplicações MLE (*Mobile Learning Engine*) para a construção do ambiente.

Como se pôde observar nos trabalhos descritos, o perfil do usuário é um fator relevante para o processo de recomendação, já que através das informações coletadas das ações deste tem-se a possibilidade de gerar recomendações de seu interesse, e não apenas informações não relevantes.

2.3. Comparação RBC X SR e aplicação no trabalho proposto

Segundo Rezende (2003) na área da Inteligência Artificial os pontos chave e decisivos para a boa elaboração de sistemas inteligentes estão na habilidade de utilizar o conhecimento adquirido para desempenhar tarefas e/ou resolver problemas e a capacidade para aproveitar as associações e inferências para trabalhar com problemas que se assemelham a vida real.

Tanto sistemas RBC quanto SR têm a capacidade de retornar ao usuário o conhecimento necessário para a ação que o mesmo requisita, desde que se tenha o desenvolvimento de acordo com as regras necessárias e, o mais importante, que contenham dados redundantes em sua base de dados.

A vantagem em RBC é que os problemas solucionados uma vez estão armazenados para poderem ser reutilizados em casos futuros, onde pode se ter a utilização ou adaptação da solução já existente na base de casos, mas essa técnica torna-se empobrecida quando a base de casos não possui dados significantes que justifiquem sua utilização.

Já em SR, o retorno para o usuário vem através das informações passadas pelo próprio usuário, direta ou indiretamente, possibilitando-lhe recomendar informações relevantes ao seu interesse atual, mesmo que a base de dados não esteja com um número significativo de dados.

Quando se trata de saúde não se pode ter apenas recomendações baseadas e informações do usuário, já que nem toda pesquisa será igual e nem toda doença pesquisada pode ser diagnosticada através de similaridade. Nestes casos, o RBC se tornaria mais confiável, já que apenas casos bem sucedidos estarão contidos na base.

Tabela 3 - Comparação entre trabalhos relacionados.

Trabalhos Relacionados	Técnica	Contempla proposta do trabalho	Ambiente	Usabilidade da aplicação	Principais características
T1	RBC	Não	Web	Sim	Identificar irregularidades ou falhas em atendimento médico
T2	RBC	Não	Web	Sim	Suporte a decisão de técnicos de Help Desk
T3	RBC	Em partes	Web	Sim	Acompanhamento de histórico médico, apoio a decisão de profissionais e ambiente de aprendizado.
T4	SR	Não	Web	Sim	Recomendação de produtos através da filtragem híbrida.
T5	SR	Não	Móvel	Sim	Recomenda artigos acadêmicos para aprendizagem.
T6	SR	Não	Móvel	Sim	Auxílio de recomendação de usuários para ajudar no processo de ensino.
T7	SR	Em partes	Móvel	Sim	Ambiente de aprendizagem móvel para auxiliar estudantes
T8	Regras	Em partes	Desktop	Sim	Sistema Auxiliar de Diagnóstico e Tratamento Dermatológico

Fonte: do Autor

Legenda:

T1: Aplicação para controle de autorizações de Internação Hospitalar

T2: Aplicação de RBC no suporte a decisão de um sistema WEB de *Help Desk*

T3: *IACVIRTUAL*

T4: e-Recommend

T5: Modelo para recomendação de artigos acadêmicos baseado em filtragem colaborativa aplicado a ambientes móveis.

T6: Mobile GraW

T7: MobiLE

T8: Dermapet

A Tabela 3 faz referências aos trabalhos relacionados sobre RBC e SR citados nesse trabalho, e sua aplicabilidade no trabalho proposto.

2.4. Considerações

Nesse capítulo pode-se ter uma noção geral da aplicabilidade das técnicas de RBC e de SR da Inteligência Artificial, onde cada uma possui sua vantagem e sua desvantagem.

Pode-se levar em conta que a utilização da técnica RBC em bases que tenham muitos dados pode ser mais interessante do que a utilização da técnica de SR, já que, uma vez que um caso foi registrado, ele estará disponível para reutilização em outro momento, não havendo necessidade de sempre fazer uma nova pesquisa do zero.

A técnica de SR é altamente aconselhável quando se tem a colaboração do usuário para com o sistema, já que as informações geradas pelo próprio usuário podem ser muito mais relevantes do que informações coletadas implicitamente.

Dessa maneira, determinou-se a utilização da técnica de RBC para o desenvolvimento da aplicação proposta nesse trabalho, considerando que a quantidade de dados para pesquisa pode ser imensa e a utilização de situações já vivenciadas pode ser relevante na resolução do problema proposto para esse trabalho de conclusão.

3. DISPOSITIVOS MÓVEIS

Na década de 90 os dispositivos móveis não possuíam as ferramentas multifuncionais e os usuários não viam a necessidade de nada além de efetuar e receber chamadas telefônicas e eventualmente para o envio mensagens de texto. Não era imaginável a navegação na internet, possibilidade de ouvir música, assistir vídeos ou qualquer uma das funções básicas que os celulares possuem hoje, tornando os aparelhos celulares os principais veículos de comunicação ao redor do mundo.

Mas com o avanço da tecnologia permitiu-se que o *hardware* fosse melhorado e fosse desenvolvido em tamanho incrivelmente reduzido. Isso propiciou que os dispositivos móveis começassem a ganhar funções além das comumente utilizadas, caindo no gosto dos usuários, sendo necessário que sistema operacional e programas que gerenciavam essas funções acompanhassem esse avanço.

Segundo CINDRAL (2013), um dos grandes avanços ocorreu com o surgimento do *Symbian*, um Sistema Operacional (SO) para dispositivos móveis de código aberto que foi adotado pela Nokia, que permitia que desenvolvedores pudessem criar aplicativos baseados nos novos recursos. Outros SO para dispositivos móveis surgiram e lideravam, entre eles o iOS da Apple, Windows Mobile da Microsoft e o BlackBerryOS da BlackBerry e até a Google lançar o SO Android.

Hoje, os SO e recursos nos aparelhos existentes permitem aos usuários mais acesso a conteúdos e recursos multimídia. Os *smartphones*¹ estão transformando os hábitos de uso de celular e estão deixando de servir apenas para chamadas, como no primeiro celular apresentado em 1973 (NIELSEN, 2013), servindo para entretenimento e trabalho e sendo utilizado por uma massa de usuários, como mostra a Figura 10.

O desenvolvimento de aplicações para dispositivos móveis tem como ideia inicial de que cada um pode tornar seu dispositivo o seu computador pessoal e, para isso, surge a necessidade de aplicativos com maior usabilidade, para atender desde o usuário leigo até o usuário com mais conhecimento.

¹ Smartphone (telefone inteligente) parte da ideia de um telefone com funcionalidades avançadas e que podem ser estendidas através de programas a serem instalados em seu sistema operacional.

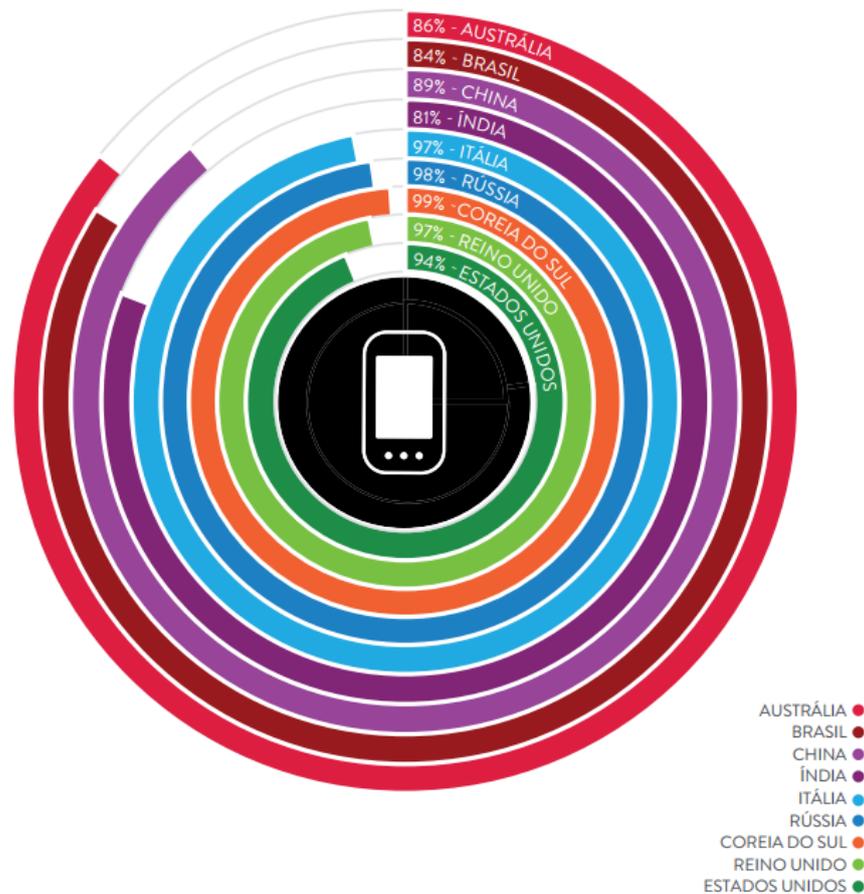


Figura 10 - Usuários de telefones celular com 16 anos ou mais

Fonte: NIELSEN (2013)

Com o lançamento da plataforma Android para desenvolvimento, foi introduzido o conceito de integração e flexibilidade, onde os aplicativos fornecidos por padrão com o sistema poderiam ser substituídos por aplicações customizadas, tornando muito interessante, principalmente, para os fabricantes de aparelhos, como o código fonte é *open source*, tem-se a liberdade de alterar de acordo com as necessidades, inserindo *marketing* favorável a cada fabricante.

Sendo assim, devido a plataforma Android possuir uma arquitetura simples, flexível e ao mesmo tempo incrivelmente poderosa foi escolhida para o desenvolvimento da aplicação proposta neste Trabalho de Conclusão e será mais detalhado na seção 3.1.

3.1. Android

Com a popularização dos dispositivos móveis foi criada uma nova maneira de interagir com os sistemas de computador, sendo possível adicionar a pequenos dispositivos as mesmas funcionalidades que antes se permitiam somente a computadores pessoais. Com a

ascensão destes dispositivos, veio a necessidade do desenvolvimento de bons sistemas operacionais que deem suporte às diversas necessidades dos usuários e que permitam que os mesmos interajam de maneira simples e objetiva com as aplicações, e que também seja permitido que empresas ou usuários desenvolvam suas próprias aplicações para venda ou para própria utilização.

Como já citado, várias são as plataformas para desenvolvimento de aplicações para dispositivos móveis, entre elas: *Symbian*, *JavaME*, *.NET Compact Framework*, *Android* e *IOS*. E essa evolução de dispositivos permitiu o desenvolvimento de sistemas operacionais específicos, surgindo um dos principais sistemas operacionais do mercado atual o sistema *Android*, hoje mantido pela empresa *Google* em parceria com o grupo *OHA*. Este é de código aberto, desenvolvimento acessível e que devido a utilização dele em aparelhos muito mais baratos, teve um alcance muito superior em pouco tempo, batendo de frente com o concorrente, a *Apple*.

O *SO Android* tem boa parte de sua estratégia centrada no desenvolvimento de aplicativos por parte de outras empresas e programadores e proporciona, de acordo com o próprio site do desenvolvedor (*ANDROID*, 2013), facilidade de navegação e vários recursos além da agilidade e a flexibilidade para o desenvolvimento de novas aplicações. O seu código fonte é disponibilizado para os usuários, permitindo a modificação do mesmo para adaptar-se as diferentes necessidades e possui o diferencial de poder ser instalado em aparelhos de diversos fabricantes, o que permitiu que já alcançasse o número de mais de 900 milhões de dispositivos em uso e o desenvolvimento de mais de 975 mil aplicativos (*ANDROID*, 2013).

Essa flexibilidade de desenvolvimento permite que a plataforma *Android* seja referenciada como o sistema operacional para smartphones que mais cresce no mundo e esteja sempre se adaptando para inclusão de novas tecnologias, já que a comunidade de desenvolvedores está sempre trabalhando em conjunto para disponibilizar aplicações móveis inovadoras.

3.1.1. A plataforma

Segundo *Smith e Friesen* (2012), ao contrário do que se fala na maioria das fontes bibliográficas, o *Android* não se originou pela empresa *Google*, mas sim foi desenvolvido pela companhia *Android Inc*, que foi comprada pela gigante *Google* em 2005. Em 2007 foi lançado com o objetivo inicial e principal de criar uma plataforma única e aberta, totalmente

código livre e flexível para o desenvolvimento para celulares e aplicativos móveis. O projeto, hoje mantido em parceria com o *Open Handset Alliance* (OHA), um grupo formado por mais de 50 empresas (Tabela 4) das áreas de tecnologias de dispositivos móveis, provedoras de serviços, fabricantes de aparelhos, surgiu com o objetivo de unir forças para inovar e acelerar o desenvolvimento de aplicações e serviços (RABELLO, 2008).

O Android foi construído para permitir que os desenvolvedores criem aplicativos móveis atraentes, que tiram o máximo proveito de tudo o que um aparelho móvel possa oferecer (OHA, 2013). Pode-se dizer que a plataforma Android é a primeira plataforma móvel completa, aberta e livre, além de poder ser aplicado em diversos aparelhos móveis de diversos fabricantes, o que possibilita um custo para o consumidor final mais barato, o que gerou um gosto popular e em poucos anos se tornou o sistema operacional para dispositivo móvel mais utilizado no mundo, como pode ser visto na Figura 11.

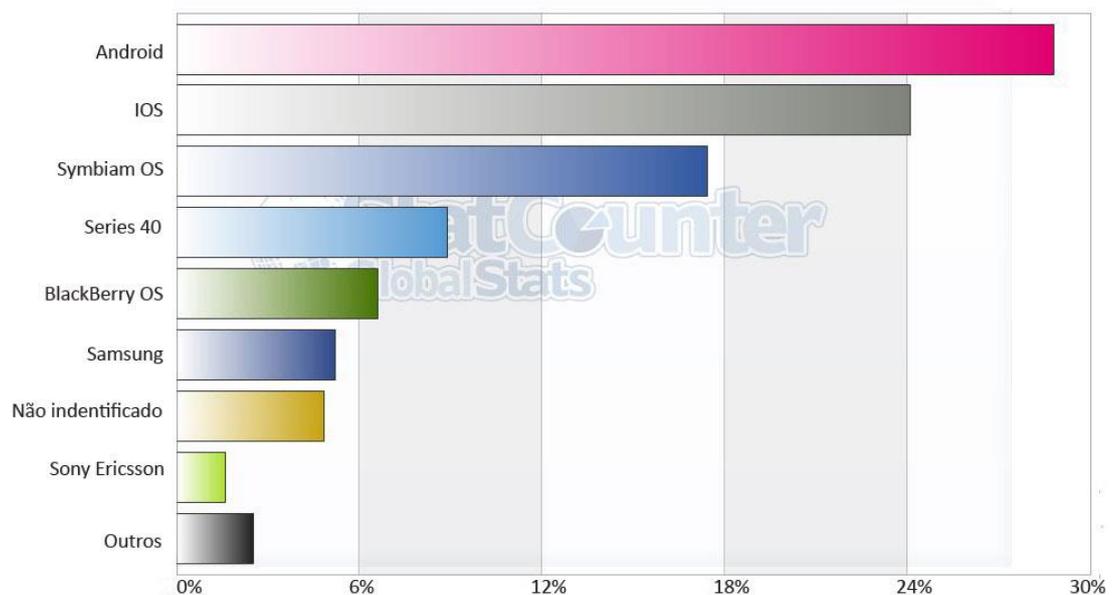


Figura 11 - Os sistemas operacionais mais utilizados nos últimos 5 anos

Fonte: STATCOUNTER² (2013)

² StatCounter é uma ferramenta de captura de informação através de acesso a páginas *web* que contenham código de configuração, retornando assim os dados referente aos usuários que acessam tais páginas.

Tabela 4 - Membros da OHA

Operadoras Móveis	Empresas de Software	Empresas de comercialização	Indústria de semicondutores	Fabricantes de celulares
<ul style="list-style-type: none"> - Bouygues Telecom; - China Mobile; - China Telecommunications; - China United; - KDDI CORPORATION; - LG Uplus; - NTT DOCOMO; - SOFTBANK MOBILE; - Sprint Nextel; - T-Mobile; - Telecom Italia; - Telefónica; - TELUS; - Vodafone. 	<ul style="list-style-type: none"> - Ándago Ingeniería S.L.; - ACCESS; - Ascender; - Cooliris; - eBay; - Google; - LivingImage; - Myriad; - MOTOYA; - Nuance Communications; - NXP Software; - OMRON SOFTWARE; - PacketVideo (PV); - SkyPop; - SONiVOX; - SVOX; - VisualOn. 	<ul style="list-style-type: none"> - Accenture; - Aplix Corporation; - Borqs; - Intrinsic Software International; - L&T Infotech; - Noser; - Sasken; - SQLStar International; - TAT; - Teleca AB; - Wind River; - Wipro Technologies. 	<ul style="list-style-type: none"> - AKM Semiconductor; - Audience; - ARM; - Atheros; - Broadcom; - CSR Pic.; - Cypress; - Freescale; - Gemalto; - Imagination; - Intel; - Marvell; - MediaTek; - MIPS; - NVIDIA; - Qualcomm; - Renesas; - ST – Ericsson; - Synaptics; - Texas Instruments; - Via Telecom. 	<ul style="list-style-type: none"> - Acer; - Alcatel; - AsusTek; - CCI; - Dell; - FoxCnn; - Fujitsu; - Garmin; - Haier; - Huawei; - Kyocera; - Lenovo; - LG; - Motorola; - NEC; - OPPO; - Pantech; - Samsung; - Sharp; - Sony Ericsson; - Toshiba; - ZTE.

Fonte: Adaptado de OHA (2013)

Cada empresa do grupo OHA tem seu papel na construção da plataforma Android e, levando em conta que a plataforma tem seu código aberto, não somente as empresas pertencentes ao grupo fazem parte de seu desenvolvimento, permitindo que desenvolvedores de qualquer parte do mundo possam fazer alterações no código e aplicativos para a plataforma.

A plataforma Android inclui um sistema operacional que é executado no *Kernel 2.6* do Linux. Sendo assim, todas as características intrínsecas deste sistema foram incorporadas, como sistemas de arquivos, o *Kernel*, os servidores de terminais, etc (RABELLO, 2008), permitindo ser um sistema muito leve que possui inúmeros recursos, além do *middleware*³ que é baseado em Java e aplicações-chave (SMITH e FRIESEN, 2012).

³ Middleware pode ser definido como o software que interliga o SO de baixo nível com o alto nível das aplicações.

No Android uma versão do sistema operacional é conhecido como plataforma, então pode-se dizer que existem diversas plataformas diferentes do Android (LECHETA, 2010). Atualmente a plataforma Android encontra-se na sua versão 4.4 ou denominada *kit-kat*, que foi lançada em outubro de 2013 e possui *Api Level 17* sendo sucessora da versão 4.3 denominada *Jelly Bean* e *Api Level 16*.

Os aplicativos desenvolvidos para Android também se utilizam do Java e podem ser portados com bastante facilidade para permitir o desenvolvimento das aplicações voltadas para a plataforma Android.

3.1.2. Arquitetura

A arquitetura da plataforma Android é dividida em quatro camadas: o Kernel GNU Linux, bibliotecas, *framework* para aplicações e as próprias aplicações, além do *Runtime*, necessário para a execução dos aplicativos no dispositivo. A Figura 12 ilustra a arquitetura básica do Android e mostra uma visão geral do sistema, com intuito de fornecer as bibliotecas do SO, de forma sucinta e segura, para as aplicações.

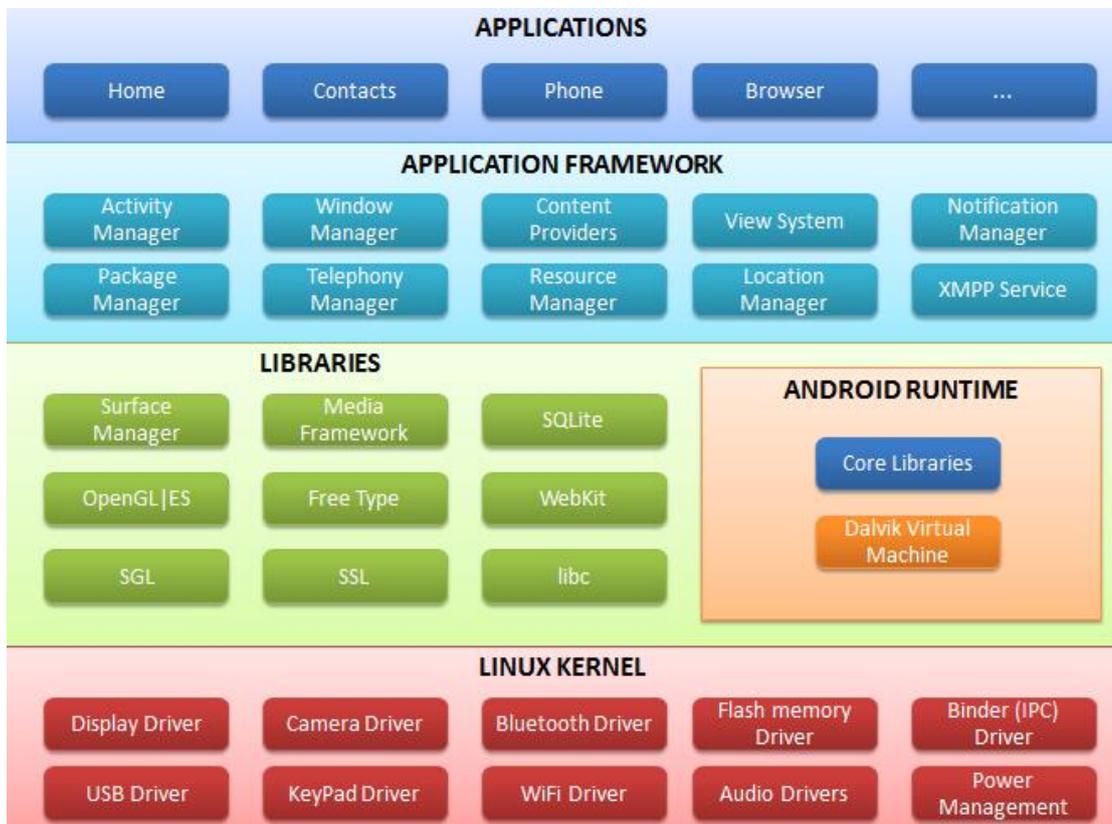


Figura 12 - Arquitetura básica Android
Fonte: Adaptado de Android Developers (2013)

Esta arquitetura é composta dos seguintes módulos principais:

- **Applications:** seria a camada de mais alto nível da arquitetura e é a camada de aplicações é composta pelo conjunto de aplicações nativas do sistema Android, que seriam as aplicações básicas do sistema operacional, como cliente de e-mail, calculadora, calendário, navegador entre outros que podem ser desenvolvidos pela comunidade. Uma vantagem da plataforma Android é que não há diferença entre as aplicações básicas do sistema, que vêm por padrão, e aplicações instaladas pelo usuário, podendo o usuário substituir um aplicativo pré-instalado por outro de sua preferência;
- **Application Framework:** nessa camada se localizam os componentes que permitirão que novas estruturas sejam utilizadas para aplicações futuras. Permite que todos os desenvolvedores tenham acesso às mesmas APIs utilizadas pelas aplicações pré-instaladas no SO, e foi desenvolvida para simplificar a reutilização de código e componentes. O gerenciador de Atividades contido na camada de *Application Framework* do Android pode ser considerado uma das APIs mais importantes da plataforma, sendo responsável pelo gerenciamento do ciclo de vida de todas as atividades e é imprescindível para o desenvolvimento de aplicações Android;
- **Libraries:** a camada de bibliotecas possibilita a utilização de um conjunto de bibliotecas C/C++ consumida por inúmeros componentes do sistema, que são expostas para os desenvolvedores através da camada de estrutura de Aplicação do Android;
- **Runtime:** seria a camada de execução e se encontra no mesmo nível da camada de bibliotecas, sendo composta pela Máquina Virtual (VM) Dalvik. Toda e qualquer aplicação executa dentro de seu próprio processo (RABELLO, 2008), podendo ser considerada um dos principais componentes da arquitetura do SO Android, porque a base de desenvolvimento de aplicações para Android é o Java, sendo necessário que haja a VM para que exista uma forma de interpretar os códigos JAVA para dispositivos móveis;
- **Linux Kernel:** o Android utiliza o Linux versão 2.6 para os serviços essenciais do sistema, como segurança, gerenciamento de memória, gerenciamento de processos, pilhas de redes, etc. É a camada de mais baixo nível e é responsável pela abstração entre o *hardware* e os aplicativos (CRUZ, 2010).

3.1.3. Ciclo de Vida

Cada tela do Android é tida como uma atividade e cada atividade possui o seu próprio ciclo de vida que é gerenciado pelo gerenciador de Atividades da camada de *Framework* de Aplicação.

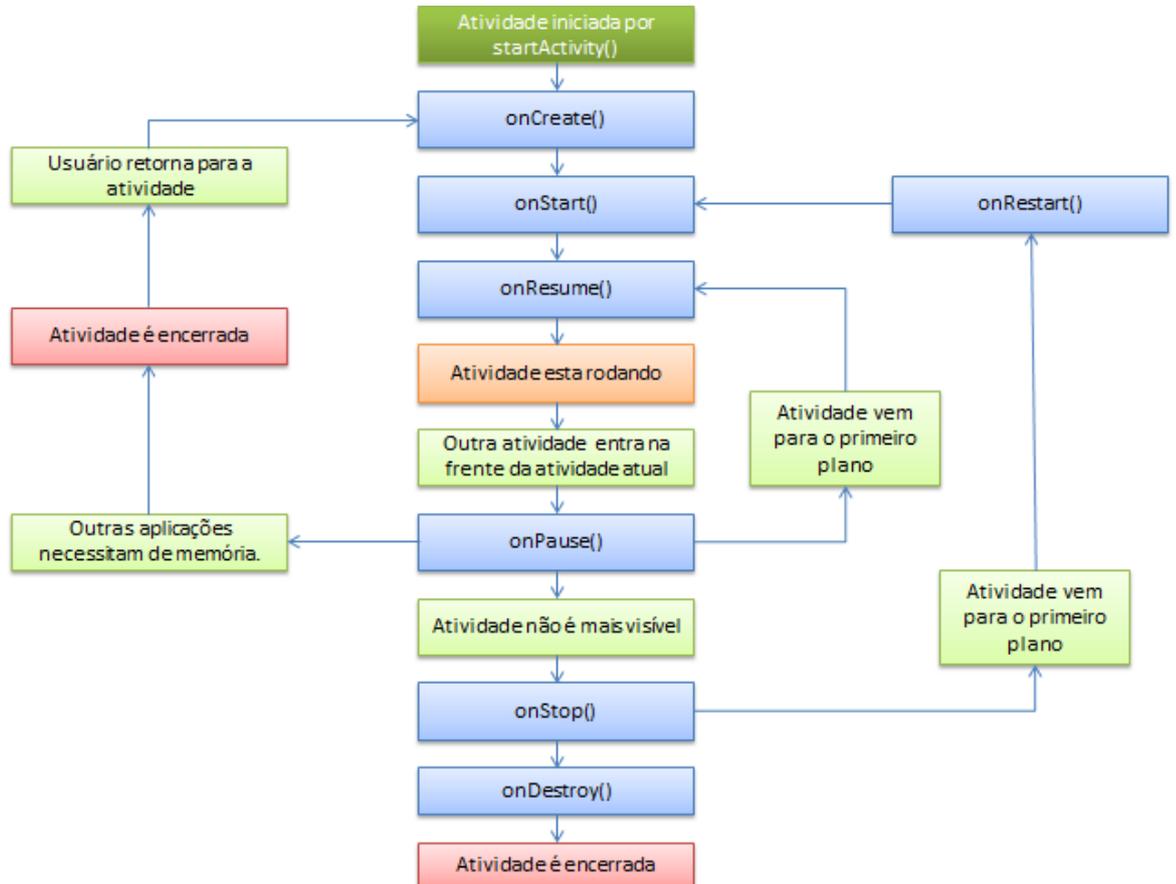


Figura 13 – Ciclo de vida de uma atividade
 Fonte: Adaptado de SMITH e FRIESEN (2012)

Quando uma nova atividade é criada, esta é passada para o topo da fila, tornando-se a atividade corrente. A atividade anterior permanece abaixo na pilha e não será mostrada enquanto a atividade corrente não terminar. A Figura 13 ilustra o ciclo de vida de uma atividade, com o fluxograma detalhado dos estados que podem ser assumidos pela atividade dentro dos sete métodos que são descritos na Tabela 5.

Tabela 5 - Métodos que definem o ciclo de vida de uma atividade

Método	Descrição	Próximo
onCreate()	É chamado quando a atividade é criada pela primeira vez. Este método é usado para criar interface da atividade do usuário, criar tópicos de fundo quando necessário e realizar demais inicializações globais.	onStart()
onRestart()	É chamado depois que a atividade terminou, logo antes de ser iniciado novamente.	onStart()
onStart()	É chamado logo após a atividade tornar-se visível para o usuário.	onResume()ou onStop()
onResume()	É chamado logo após a atividade passar a interação com o usuário. Neste ponto a atividade tem o foco, e, a entrada do usuário é direcionada para a atividade.	onPause()
onPause()	É chamado quando o Android está prestes a retomar a outra atividade. Este método geralmente é utilizado para persistir as alterações não salvas, para animações que podem consumir ciclos de processamento, e assim por diante e deve executar o seu trabalho rapidamente, pois a próxima atividade não será retomada até que ele tenha retorno.	onResume()ou onStop()
onStop()	É chamado quando a atividade já não é visível para o usuário. Isso pode acontecer por que a atividade está sendo destruída, ou por que outra atividade, seja uma existente ou uma nova atividade, foi retomada e está abrangendo a atividade.	onRestart()ou onDestroy()
onDestroy()	É chamado antes de a atividade ser destruída, a menos que a memória seja pequena e o Android seja forçado a matar o processo da atividade.	Nenhum

Fonte: Adaptado de SMITH e FRIESEN (2012)

3.1.4. Android SDK

Para iniciar o desenvolvimento de aplicações para o Android, é necessários instalar o Android SDK, ou *Kit* de desenvolvimento de *softwares*, que é um software disponibilizado no site do desenvolvedor (ANDROID, 2013) e que permite desenvolver o Android OS. Fazem parte desse *Kit* um emulador capaz de executar o Android OS e capaz de simular um dispositivo móvel, uma biblioteca completa para a linguagem JAVA, contendo as classes necessárias para desenvolver as aplicações para Android e outras ferramentas como compilador, debugador, exemplos de código e documentação de apoio.

O *Java Development Kit* (JDK), versão 6 ou superior, é um requisito para a instalação do Android SDK, e consiste de bibliotecas de classes Java. Para o desenvolvimento o Android SDK fornece a ferramenta Eclipse IDE.

3.1.5. Ferramentas de desenvolvimento

Além do pacote padrão Eclipse IDE que é disponibilizado pelo site do desenvolvedor (ANDROID, 2013), existem outras ferramentas no mercado que possibilitam a criação de aplicações destinadas ao Android:

- **Mono:** para desenvolver para Android, é necessário utilizar a linguagem Java, mas, com a criação do projeto *MonoDroid*, é possível também desenvolver aplicativos com a linguagem de programação C#. Através do Mono tem-se a possibilidade de utilizar todas as bibliotecas .NET e as nativas do Google Android. A própria ferramenta se encarrega de transformar o código C# para os formatos suportados pela Máquina Virtual do Android. A IDE Mono é gratuita, mas para se utilizar a API da *Xamarin* necessita de uma licença de desenvolvimento. Disponibilizada pela empresa *Xamarin*⁴ e pode ser utilizada com o Visual Studio disponibilizado pela Microsoft (MONO DEVELOPED, 2013).
- **Android Studio:** uma IDE com base na versão comunitária do IntelliJ IDEA. O Android Studio vem com todos os pacotes para desenvolvimento na plataforma Android, incluindo o SDK, as ferramentas e os emuladores. A programação é feita em Java, semelhante ao Eclipse, e fornece ferramentas de desenvolvimento integradas Android para desenvolvimento e depuração. É uma ferramenta gratuita que pode ser baixada diretamente do site do fabricante, mas está em uma versão *Preview* sendo desenvolvida pelo Google, então não é recomendado fazer a migração de projetos já existentes, pois a ferramenta ainda não possui sua versão final. (ANDROID STUDIO, 2013).
- **PhoneGap/ApacheCordova:** é um *framework* gratuito e *open source* que permite a criação de aplicações para dispositivos móveis usando APIs padronizadas da *web* baseadas em HTML5, *Javascript* e CSS. Permite o desenvolvimento para dispositivos móveis multiplataforma, podendo ser aplicado a diversos SO *mobile* existentes no mercado. Com uma interface baseada em telas de HTML, a sua lógica de programação e os seus recursos de dispositivos são tratados pelo *Javascript* tendo a possibilidade de realizar combinações com diversos *scripts*. (PHONEGAP, 2013).

Devido ao conhecimento com a programação *Javascript* e HTML, além de ser uma ferramenta totalmente gratuita, foi escolhido para o desenvolvimento da aplicação a ferramenta *PhoneGap*, que permite transformar o código desenvolvido para o padrão do

⁴ <http://xamarin.com/>

AndroidOS, ao invés da IDE disponibilizada pelo Android, em que seu desenvolvimento é feito em Java.

3.2. Apache Cordova / Phonegap

Não há como negar que o mercado de aplicações móveis está em constante evolução, em todas as áreas e dispositivos, desde *smartphones* a tablets e relógios inteligentes, gerando um leque considerável de plataformas e sistemas operacionais. No entanto, o desenvolvimento para cada uma dessas plataformas em separado pode ser uma tarefa árdua e maçante, especialmente quando os recursos para o desenvolvimento e conhecimento são limitados. Devido a essa necessidade de mercado que em 2009 o Phonegap foi criado. Este consiste de uma API de código aberto para acesso a recursos nativos dos dispositivos móveis, com o objetivo de permitir que os desenvolvedores criem as aplicações móveis usando tecnologias padrão de *web*.

Tecnicamente, o desenvolvimento utilizando Phonegap não possui limitações em relação a aplicações nativamente desenvolvidas, pois se trata de uma API *Javascript*, que serve como um tradutor do *Javascript* para a linguagem nativa do SO utilizado. Isso significa que o desenvolvimento para iOS irá gerar um arquivo IPA, para o Android um arquivo APF, e Windows Phone um XAP, pronto para fazer o envio da aplicação para as lojas de cada um desses SO, como pode ser visto na Figura 14.

Para desenvolver uma aplicação móvel com o *Apache Cordova / Phonegap* é possível optar por dois caminhos básicos:

- Desenvolver aplicações multiplataforma, sem focar em uma plataforma específica, com o auxílio de ferramentas do próprio *Apache Cordova / Phonegap*, como o Cordova Command-Line Interface (CLI), que serve como uma abstração de nível mais elevado para configurar e construir a sua aplicação para diferentes plataformas. Este é o caminho de desenvolvimento mais comumente usado;
- E existe a possibilidade de desenvolvimento de uma aplicação focada em uma plataforma específica, onde é utilizado o fluxo de trabalho centrado nessa plataforma. Dessa forma, é possível ajustar e modificar o código em um nível

inferior, fazendo uma mescla de componentes nativos com os componentes disponíveis no *Apache Cordova / Phonegap*.

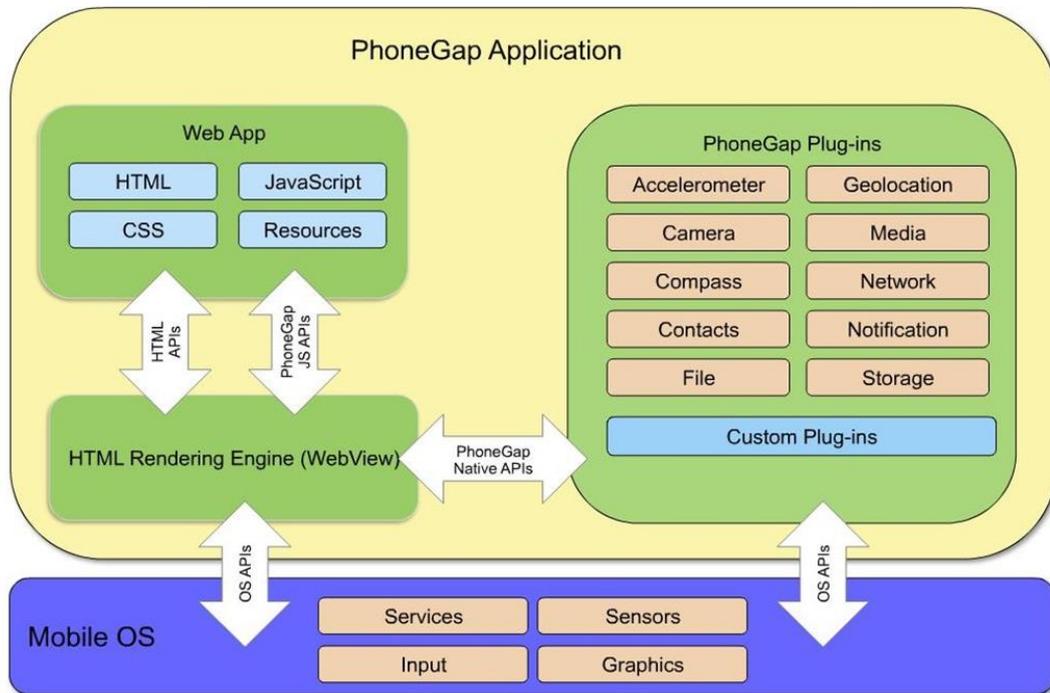


Figura 14 - Arquitetura Apache Cordova / Phonegap
Fonte: Google Images (2013)

É possível acessar componentes nativos graças a uma interface de *plug-ins* existentes no *Apache Cordova / Phonegap* que permite invocar código nativo da plataforma para a qual se está trabalhando a partir do *Javascript*, podendo utilizar componentes nativos do *Apache Cordova / Phonegap* ou desenvolvidos por terceiros, na Figura 15 tem-se uma visão da utilização de *plug-ins* com o *Apache Cordova / Phonegap*.

As aplicações desenvolvidas com *Apache Cordova / Phonegap* possuem um arquivo de configuração em comum: “*config.xml*”, sendo que a partir desse é possível adquirir informações sobre a aplicação e os parâmetros de configuração. A aplicação é implementada utilizando HTML5, como uma página *web*, e executa como uma *WebView* dentro da plataforma nativa que está sendo executada, podendo ser distribuída para as lojas de *download* de cada sistema operacional.

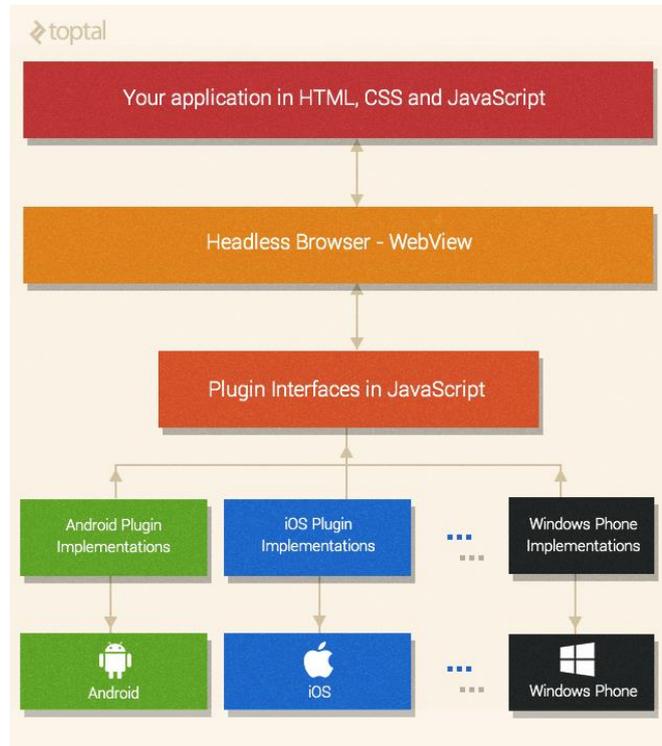


Figura 15 - Fluxo de criação com o PhoneGap e utilização de Plugins
 Fonte: Google Imagens (2013)

3.3. Trabalhos relacionados

Com a evolução da tecnologia para dispositivos móveis o desenvolvimento para esse tipo de plataforma acelerou e tomou proporções gigantescas, sendo inúmeras as aplicações desenvolvidas especialmente para aparelhos portáteis hoje.

SIBALDO, SALES e CALADO (2007) desenvolveu uma aplicação para dispositivos móveis baseado em comunidades virtuais, chamada *Mobile GraW*. Tal aplicação faz uso de sistema de recomendação de usuários devido a necessidade de encontrar potenciais usuários que auxiliem no processo de ensino/aprendizagem. Esse sistema pode ser acessado através de inúmeros dispositivos móveis e apoia o processo de aprendizagem fora da sala de aula através de ferramentas, tais como agenda, fórum e mensagem.

Morais et al (2012) construiu um sistema móvel de suporte à decisão para apoio ao diagnóstico médico inicial denominado *InteliMED*, com foco em atenção básica à saúde, com ênfase ao diagnóstico de asma devido à importância da doença para o Sistema Único de Saúde, utilizando inteligência computacional e desenvolvido utilizando a tecnologia Java e a plataforma Android.

Pacheco et al. (2012) apresenta uma solução para apoio ao diagnóstico clínico, através da geração de indicadores, processados em tempo real, gerados através de soluções móveis, compatíveis com a plataforma Android, denominado de Sistema de Auxílio de Diagnóstico em tempo real ou SADTR.

Paiva et al. (2013) criou um projeto de pesquisa, cujo objetivo geral é o desenvolvimento de um sistema para apoio à aprendizagem da anatomia humana, unindo as áreas da saúde e educação. Foram utilizados os recursos da Realidade Virtual e Aumentada (RVA) que possibilitam a visualização, interação e simulação tridimensional de imagens em conjunto com o desenvolvimento de um aplicativo em Android.

Vet Smart (2015) é um aplicativo para os sistemas operacionais Android e IOS para o apoio clínico e para estudantes de medicina veterinária. O Aplicativo possui duas versões, um para Cães e Gatos e um para Bovinos e Equinos. A base de dados funciona mais como uma base de dados para bulas de medicamentos direcionadas a animais, onde a pesquisa é feita pelo medicamento.

Na Tabela 6 se faz a análise e comparação entre os trabalhos relacionados citados nessa seção.

Tabela 6 – Análise dos Trabalhos relacionados

Trabalhos Relacionados	Comtempla proposta do trabalho	Tecnologia	Usabilidade da aplicação	Principais características
Mobile GraW	Não	J2ME	Sim	Sistema de recomendação de usuários devido a necessidade de encontrar potenciais usuários que auxiliem no processo de ensino/aprendizagem.
SADTR	Em partes	Android	Sim	Auxílio de diagnóstico clínico em tempo real
InteliMED	Em partes	Android	Sim	Sistema de apoio à decisão ao diagnóstico de asma.
Sistema para apoio à aprendizagem da anatomia humana	Não	Android	Sim	Sistema para apoio à aprendizagem da anatomia humana, unindo as áreas da saúde e educação.
Vet Smart	Em partes	Android e IOS	Sim	O Vet Smart Bovinos e Equinos é o primeiro aplicativo brasileiro para apoio ao clínico e estudante de Medicina Veterinária de grandes animais.

Fonte: do Autor

Como se pode observar nos trabalhos relacionados, o desenvolvimento de aplicações para dispositivos móveis pode ser muito útil em qualquer área de atuação, desde aplicativos que auxiliem a aprendizagem até aqueles que provêm o auxílio de diagnósticos de profissionais na área de saúde também mostra a geração de aplicações em que existe a descentralização do serviço oferecido, permitindo que seja acessado em qualquer local, sendo funcional inclusive sem o acesso à internet. Além disso a utilização de aparelhos móveis é de muita valia para quaisquer profissionais, principalmente com a ausência de um especialista da área ou até mesmo onde a tecnologia é escassa e são necessários recursos que auxiliam no processo decisório.

3.4. Considerações

A facilidade de acesso e desenvolvimento aliada a grande robustez de documentações para o desenvolvimento de uma aplicação para dispositivo móvel na plataforma Android foi o fator decisivo para escolher esse SO para o desenvolvimento da aplicação proposta neste trabalho.

Visando a descentralização do serviço a ser oferecido, permitindo a aplicabilidade em qualquer área de atuação que seja necessário ter um auxiliar para aprendizagem ao alcance, em qualquer lugar, possibilitam que a criação de aplicações móveis que permitam tal ação seja mais requisitada.

Neste capítulo verificou-se as vantagens da utilização da plataforma Android para a criação de aplicações para dispositivos móveis, que está fortemente ligado à facilidade na implementação. Mesmo sendo uma tecnologia recente, já tem muitos adeptos tornando-se uma plataforma em constante avanço tecnológico, o que faz a tecnologia muito atrativa tanto para desenvolvedores de aplicação quanto para usuários dessas aplicações, tornando o desenvolvimento de aplicações para os dispositivos móveis um mercado forte e crescente.

4. LEVANTAMENTO DE REQUISITOS PARA A APLICAÇÃO

Neste capítulo é apresentado o levantamento de requisitos, que é responsável por assimilar o problema e definir as funcionalidades e restrições necessárias para a implementação da aplicação.

4.1. Requisitos Funcionais

As funcionalidades que o sistema deve possuir são descritas através de requisitos funcionais, que são apresentados na Tabela 7. Cada cadastro e funcionalidade do sistema está relacionada com a descrição das ações que o usuário pode realizar.

Tabela 7 - Requisitos Funcionais

Requisito	Nome	Descrição
RF1	Cadastro de Espécies	Opções para o cadastro, a manutenção e listagem de espécies na aplicação.
RF2	Cadastro de Raças	Opções para o cadastro, a manutenção e listagem de raças na aplicação.
RF3	Cadastro de Animal	Opções para o cadastro, a manutenção e listagem de animais na aplicação.
RF4	Configurações	Opções de configurações relevantes ao cálculo de RBC.
RF5	Diagnóstico	Usuário inicia o processo de diagnóstico, onde fará as seleções de sintomas e patologias pertinentes ao tratamento que está à procura.

Fonte: Do autor

4.2. Requisitos não funcionais

Os requisitos não funcionais consistem de características e aspectos internos da aplicação, com informações mais técnicas. São basicamente restrições do sistema definidas através dos requisitos não funcionais, que são descritos a seguir:

- para o correto funcionamento do sistema o usuário deve estar conectado a internet na hora da instalação e primeiro uso;
- a aplicação usa a implementação Web SQL, do HTML 5, para criar bancos de dados locais;

- os dados iniciais de sintomas, patologias e tratamentos serão alimentados através do *script* de dados que será acessado no primeiro uso;
- a aplicação funciona exclusivamente em aparelhos móveis, sendo assim, é necessário um dispositivo dessa categoria para sua utilização.

4.3. Descrição dos atores

Em UML (*Unified Modeling Language*), define-se para “ator” qualquer elemento externo que interage com o sistema. A Figura 16 mostra o Diagrama de Atores que apresenta o “Profissional” como elemento que faz uso da aplicação. Este ator tem a função de efetuar os cadastros necessários para o funcionamento e alimentação do sistema, já que sistemas RBC dependem da manutenção e aquisição de novos casos para manter uma base de casos o mais completa possível.

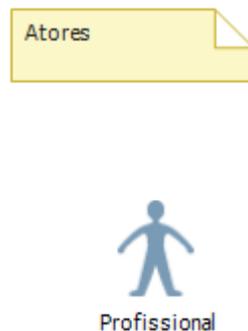


Figura 16 - Diagrama de Atores

Fonte: do autor

4.4. Regras de negócio

As regras de negócio são basicamente as definições de como o negócio funciona, evidenciando as restrições existentes para o funcionamento de determinado negócio, de acordo com a Tabela 8.

Tabela 8 - Regras de Negócio

Regra	Nome	Descrição
RN1	Acesso à aplicação	Usuário deve instalar a aplicação
RN2	Cadastro de Espécies	Campo de descrição é obrigatório e não pode ultrapassar 150 caracteres.
RN3	Cadastro de raças	Deve haver espécies cadastradas para obter sucesso no cadastro de Raças. Todos os campos são obrigatórios. Campo de descrição não pode ultrapassar 150 caracteres.
RN4	Cadastro de Animal	Deve haver espécies e raças cadastradas para obter sucesso no cadastro de animal. Todos os campos são obrigatórios. Campo de Descrição não pode ultrapassar 150 caracteres. Campo “Proprietário” não pode ultrapassar 150 caracteres. Campo de telefone não pode ultrapassar 15 caracteres.
RN5	Diagnóstico	Deve haver animais cadastrados para obter sucesso no diagnóstico. Todos os campos são obrigatórios.
RN6	Seleção de Sintomas	A seleção de, ao menos, um sintoma é obrigatória. Caso não existam sintomas cadastrados, direcionar para a tela de tratamento manual.
RN7	Seleção de Patologia	A seleção de uma patologia é obrigatória para dar continuidade ao processo de diagnóstico.
RN8	Visualização de Tratamento e adição de tratamento à base de casos.	Após selecionar a patologia deve-se salvar o processo de diagnóstico para persistir o novo tratamento, relacionado ao animal, à base de casos.
RN9	Tratamento Manual	Quando não existem sintomas cadastrados é automaticamente redirecionado à tela de Tratamento Manual, onde serão adicionados tratamentos de forma manual. Todos os campos são obrigatórios. É obrigatória a seleção de ao menos 1 sintoma.

Fonte: do autor

4.5. Casos de Uso

Pode-se definir um caso de uso como a especificação de uma sequência de iterações entre um sistema e o usuário que o está operando (BEZERRA, 2006). Nessa seção são apresentados o Diagrama de Casos de Uso e a descrição dos casos relevantes à aplicação desenvolvida.

4.5.1. Diagrama de Casos de Uso

A Figura 17 demonstra os casos de uso do sistema, apresentando o mapeamento das ações que o profissional pode realizar na aplicação desenvolvida, já que os diagramas de casos de uso nos fornecem uma descrição do sistema e suas interações, podendo representar uma visão de alto nível das funcionalidades da aplicação (FURLAN, 1998).

O processo de utilização da aplicação consiste dos cadastros de Espécies, de Raças, Cadastro de Animais e, por fim, o Diagnóstico e tratamento que irão depender exclusivamente dos sintomas selecionados.

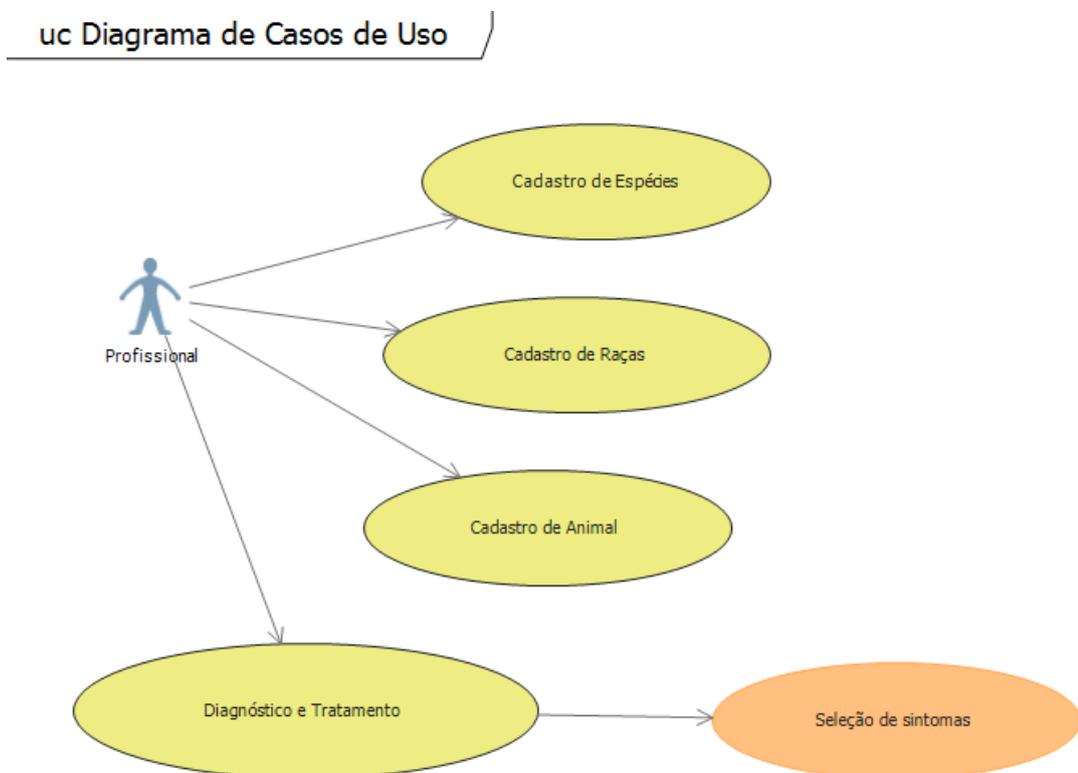


Figura 17 - Diagrama de Casos de Uso

Fonte: do autor

4.5.2. Descrição dos Casos de uso

Partindo do Diagrama de Casos de Uso pôde-se elaborar uma documentação sobre cada caso de uso relevante ao sistema, visando descrever cada funcionalidade e o que se espera de cada. As Tabelas 9, 10, 11, 12, 13, 14 e 15 descrevem os casos de uso resultantes dessa análise.

Tabela 9 - Caso de Uso UC1: Cadastro de Espécie

UC1: Cadastro de Espécie
Permite o cadastro, manutenção e listagem das espécies cadastradas na aplicação.
Atores: Profissional
Pré-condição: Esse caso de uso não possui pré-condição.
Pós-condição: Exibição das espécies cadastradas na base.
Fluxo principal:
<ol style="list-style-type: none"> 1. Esse caso de uso se inicia quando o ator acessa a tela de Cadastros de Espécies. 2. O ator cadastra um novo registro ou fará manutenção em um registro já existente. 3. O ator salva o registro. 4. O ator volta para a tela de listagem. 5. O ator cadastra/altera novos registros ou esse caso de uso é encerrado.

Fonte: do autor

Tabela 10- Caso de Uso UC2: Cadastro de Raça

UC2: Cadastro de Raça
Permite o cadastro, manutenção e listagem das raças cadastradas na aplicação.
Atores: Profissional
Pré-condição: Para efetuar o cadastro será necessário haver espécies listadas no sistema.
Pós-condição: Exibição das raças cadastradas na base.
Fluxo principal:
<ol style="list-style-type: none"> 1. Esse caso de uso se inicia quando o ator acessa a tela de Cadastros de Raças. 2. O ator cadastra um novo registro ou fará manutenção em um registro já existente. 3. O ator salva o registro. 4. O ator volta para a tela de listagem. 5. O ator cadastra/altera novos registros ou esse caso de uso é encerrado.

Fonte: do autor

Tabela 11 - Caso de Uso UC3: Cadastro de Animal

UC3: Cadastro de Animal Permite o cadastro, manutenção e listagem dos animais cadastrados na aplicação.
Atores: Profissional
Pré-condição: Para efetuar o cadastro será necessário haver espécies e raças listadas no sistema.
Pós-condição: Exibição dos animais cadastrados na base.
Fluxo principal: <ol style="list-style-type: none"> 1. Esse caso de uso se inicia quando o ator acessa a tela de Cadastros de Animais. 2. O ator cadastra um novo registro ou fará manutenção em um registro já existente. 3. O ator salva o registro. 4. O ator volta para a tela de listagem. 5. O ator cadastra/altera novos registros ou esse caso de uso é encerrado.

Fonte: do autor

Tabela 12 - Caso de Uso UC4: Diagnóstico

UC4: Diagnóstico Esse caso de uso dará início a execução do ciclo de vida do RBC na aplicação, que fará a busca de sintomas e os tratamentos relacionados a esses sintomas.
Atores: Profissional
Pré-condição: Deve haver pelo menos um animal cadastrado no sistema para dar continuidade ao processo de diagnóstico.
Pós-condição: Exibição de sintomas listados na base.
Fluxo principal: <ol style="list-style-type: none"> 1. Esse caso de uso se inicia quando o ator iniciar um novo processo de diagnóstico. 2. Esse caso de uso aciona o Caso de uso “Listar Sintomas”. 3. Esse caso de uso faz a ligação para o CSU2.

Fonte: do autor

Tabela 13 - Caso de Uso UC5: Listar Sintomas

UC5: Listar Sintomas Esse caso de uso listará os sintomas adicionados na base de dados.
Atores: Profissional
Pré-condição: O Ator deve ter iniciado um novo processo de diagnóstico.
Pós-condição: Cria-se uma relação para base de casos no sistema.
Fluxo principal: <ol style="list-style-type: none"> 1. Esse caso de uso se inicia após o CSU1, utilizando informações já repassadas para a seleção de sintomas. 2. Cria-se uma relação para base de casos no sistema. 3. Esse caso de uso faz a ligação para o CSU3. 4. Esse caso de uso se encerra.

Fonte: do autor

Tabela 14 - Caso de Uso UC6: Listar Tratamento

UC6: Listar Tratamentos Esse caso de uso listará os tratamentos adicionados na base de casos fazendo a consulta na base de casos e consultando a similaridade dos casos recuperando aquele com maior similaridade.
Atores: Profissional
Pré-condição: O Ator deve ter iniciado um novo processo de diagnóstico. O CSU2 deve ter sido executado.
Pós-condição: Retorna uma lista de casos recuperados.
Fluxo principal: <ol style="list-style-type: none"> 1. Esse caso de uso se inicia após o CSU2, realizando a busca na base de casos. 2. Neste caso será utilizado o método de recuperação do RBC. 3. Esse caso de uso faz a ligação para o CSU4. 4. Esse caso de uso se encerra.

Fonte: do autor

Tabela 15 - Caso de Uso UC7: Confirmar Tratamento

UC7: Confirmar Tratamento Os casos revisados são armazenados na base de casos do sistema, onde os casos de sucesso são armazenados na base de casos.
Atores: Profissional
Pré-condição: O CSU3 deve ter sido executado.
Pós-condição: Os casos de sucesso serão salvos na base de casos.
Fluxo principal: <ol style="list-style-type: none"> 1. Os casos de sucesso são armazenados na base de casos. 2. Neste caso é utilizado o método de recuperação do RBC. 3. Esse caso de uso se encerra.

Fonte: do autor

Partindo dos casos de uso relacionados nas tabelas acima, pode-se entender o fluxo principal do processo de diagnóstico começando no Caso de Uso UC1, onde se dá início aos cadastros base da aplicação (Cadastro de Espécies, Cadastro de Raças e Cadastro de Animal) até chegar ao objetivo de realização de diagnóstico e confirmação do tratamento.

4.6. Diagrama de Entidade e Relacionamento

O Diagrama de Entidade e Relacionamento (ER) do banco de dados permite descrever o modelo de dados da aplicação. Sua principal aplicação é visualizar graficamente o relacionamento entre as tabelas e facilitar o entendimento de como ocorrerá a persistência no banco de dados.

Na Figura 18 é apresentado o diagrama ER associado ao processo básico de cadastro do sistema, dados que têm relação com a interação do usuário.

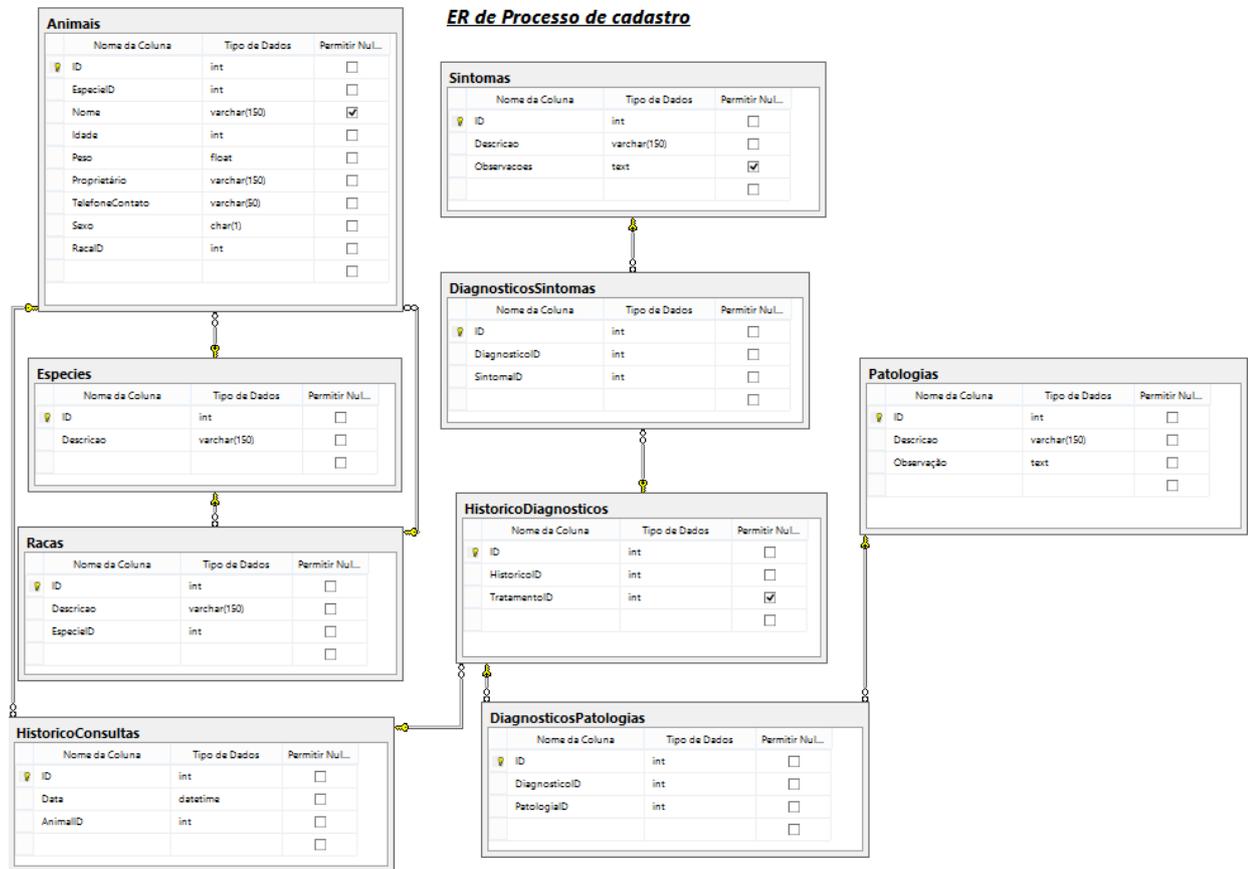


Figura 18 - Diagrama ER Processo de cadastro

Fonte: do autor

A seguir é apresentada a descrição de cada tabela do Diagrama ER Processo de Cadastro e como ela será utilizada na aplicação:

- **Especies:** tabela referente ao cadastro da tela Espécies da aplicação;
- **Racas:** tabela referente ao cadastro da tela Raça da aplicação;
- **Animais:** tabela referente ao cadastro da tela Animal da aplicação;
- **Sintomas:** tabela carregada automaticamente através de *script* inicial e com manutenção através da tela de Tratamento Manual acessado no menu configurações;

- **Patologias:** tabela carregada automaticamente através de *script* inicial e com manutenção através da tela de Tratamento Manual acessado no menu configurações;
- **HistoricoConsultas:** tabela referente a tela de diagnóstico onde se inicia o processo de consulta e diagnóstico relacionado ao animal. Faz relacionamento direto com a tabela **HistoricoDiagnosticos**;
- **DiagnosticoSintomas:** tabela referente a tela de diagnóstico no momento de seleção de sintomas;
- **DiagnosticosPatologias:** tabela referente a tela de diagnóstico no momento de seleção de patologia associada aos sintomas selecionados na tela anterior.

Na Figura 19 é apresentado o diagrama ER associado ao processo RBC do sistema, dados que têm relação com a interação ao método RBC e manutenção dos casos.

ER Processo RBC

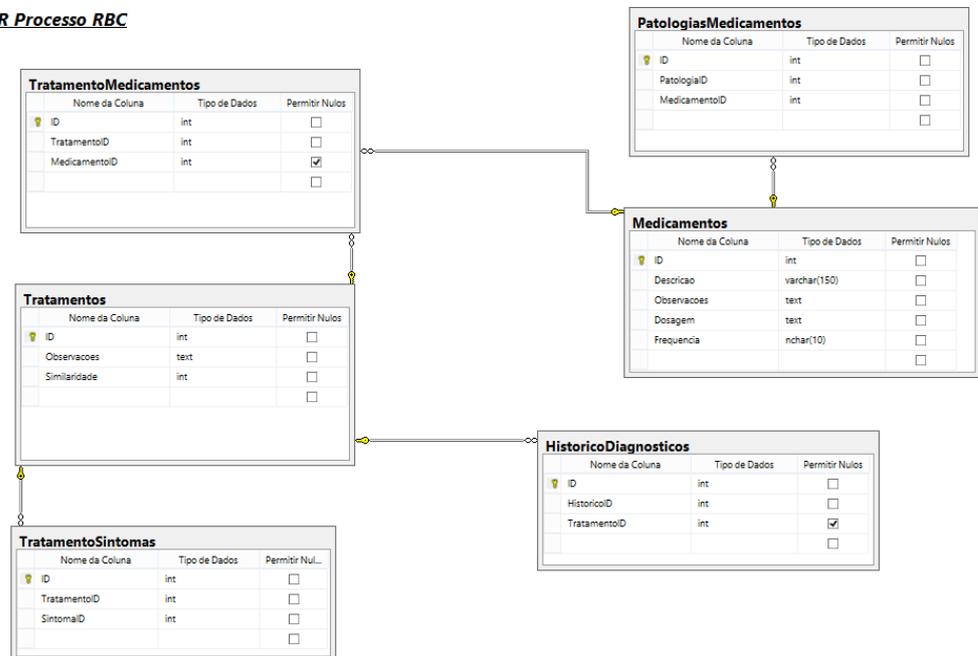


Figura 19 - Diagrama ER Processo RBC

Fonte: do autor

Abaixo é apresentada a descrição de cada tabela do Diagrama ER e como elas são utilizadas na aplicação:

- **Medicamentos:** tabela carregada automaticamente através de *script* inicial e com manutenção através da tela de Tratamento Manual acessado no menu configurações;
- **PatologiasMedicamentos:** tabela que faz relacionamento das patologias associadas ao Medicamento;
- **TratamentoMedicamentos:** tabela que faz relacionamento dos medicamentos relacionados ao tratamento específico;
- **Tratamentos:** tabela que contém as observações relacionadas ao tratamento com a similaridade resultante do método RBC definido na aplicação e retorna o tratamento gerado para a tabela **HistoricoDiagnosticos**;
- **TratamentoSintomas:** tabela que faz relacionamento dos sintomas selecionados ao tratamento específico.

4.7. Diagrama de Sequências

O diagrama de sequência tem como objetivo esclarecer as interações existentes no sistema de acordo com a sua ordem correta de execução. Na Figura 18 é apresentado o diagrama de sequência dos casos de uso relevantes ao sucesso do objetivo principal deste trabalho.

Ainda na Figura 20 pode-se observar que, para efetuar o Diagnóstico é necessário o retorno do cadastro de Animal e, este conseqüentemente precisa dos cadastros de Raça e Espécies. Seguido pelo processo de diagnóstico com a seleção dos sintomas, as patologias listadas por sintomas associados e, para finalmente obter o retorno do tratamento.

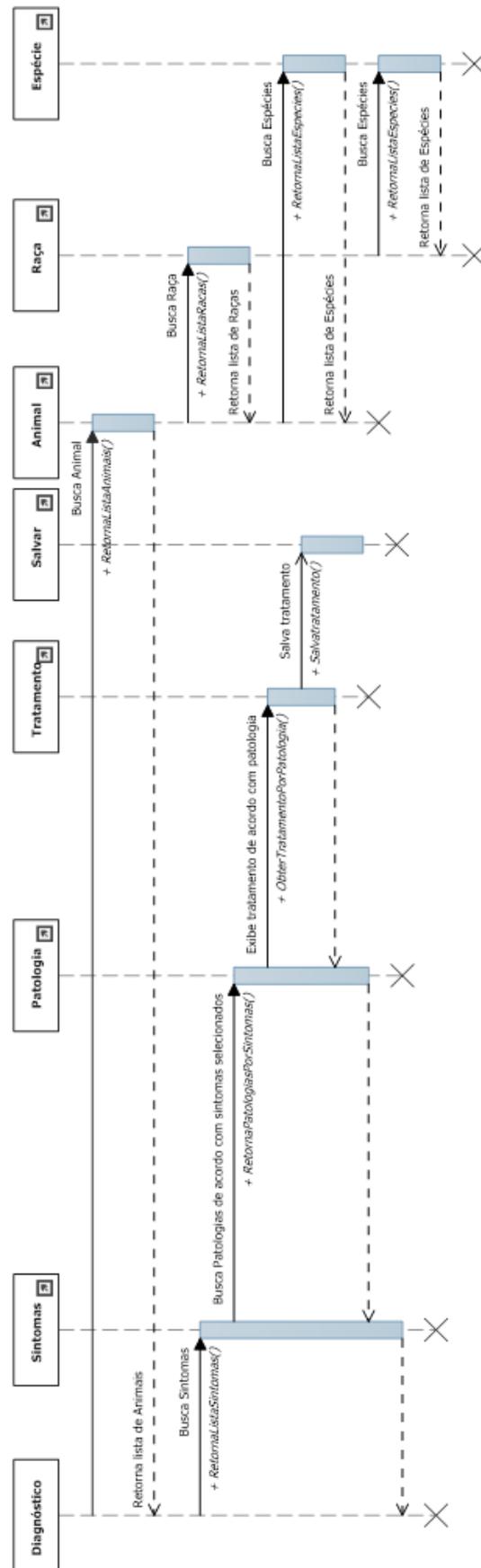


Figura 20 - Diagrama de seqüência
 Fonte: do autor

4.8. Considerações

A partir de todo o levantamento dos requisitos e regras necessárias para o entendimento do fluxo de desenvolvimento e do processo para realização de um diagnóstico pode-se dar início ao trabalho de desenvolvimento da aplicação que será vista no capítulo 5.

5. APLICAÇÃO MÓVEL DESENVOLVIDA

Este capítulo tem como objetivo apresentar a aplicação desenvolvida com base na incorporação da técnica do RBC selecionada. Assim, este aplicativo móvel possibilitará o auxílio de diagnósticos aos profissionais da área de medicina animal, com a utilização dessa técnica, permitindo a definição de casos já existentes para a pesquisa de uma solução para um problema específico.

O domínio escolhido para a utilização da aplicação foi de animais de grande porte, que habitualmente se encontram em lugares remotos onde a disponibilidade de tecnologia pode ser escassa e até mesmo nula. Dessa forma, esta aplicação pode ser considerada de grande valia aos profissionais veterinários. Além disso, quando a medicina veterinária entra em casos de prevenção, controle e tratamento de doenças surgem dificuldades no que diz respeito ao diagnóstico de determinadas patologias, já que existem casos onde se apresentam sintomas que podem por em dúvidas o real diagnóstico elaborado pelos profissionais.

5.1. Arquitetura do sistema

Inicialmente fez-se necessário a definição e coleta de dados para a montagem de uma base de dados onde é feita toda a extração de informação. E, para isso, foram coletados dados em relação a medicamentos específicos para a realização deste trabalho com auxílio de um especialista humano.

A Figura 21 apresenta a arquitetura proposta para o tráfego de informações na aplicação móvel, sendo dividida basicamente em três módulos: o módulo de configurações do sistema; o módulo de interface com o usuário e o módulo que é o foco principal da aplicação, consistindo da inteligência baseada na técnica de RBC.

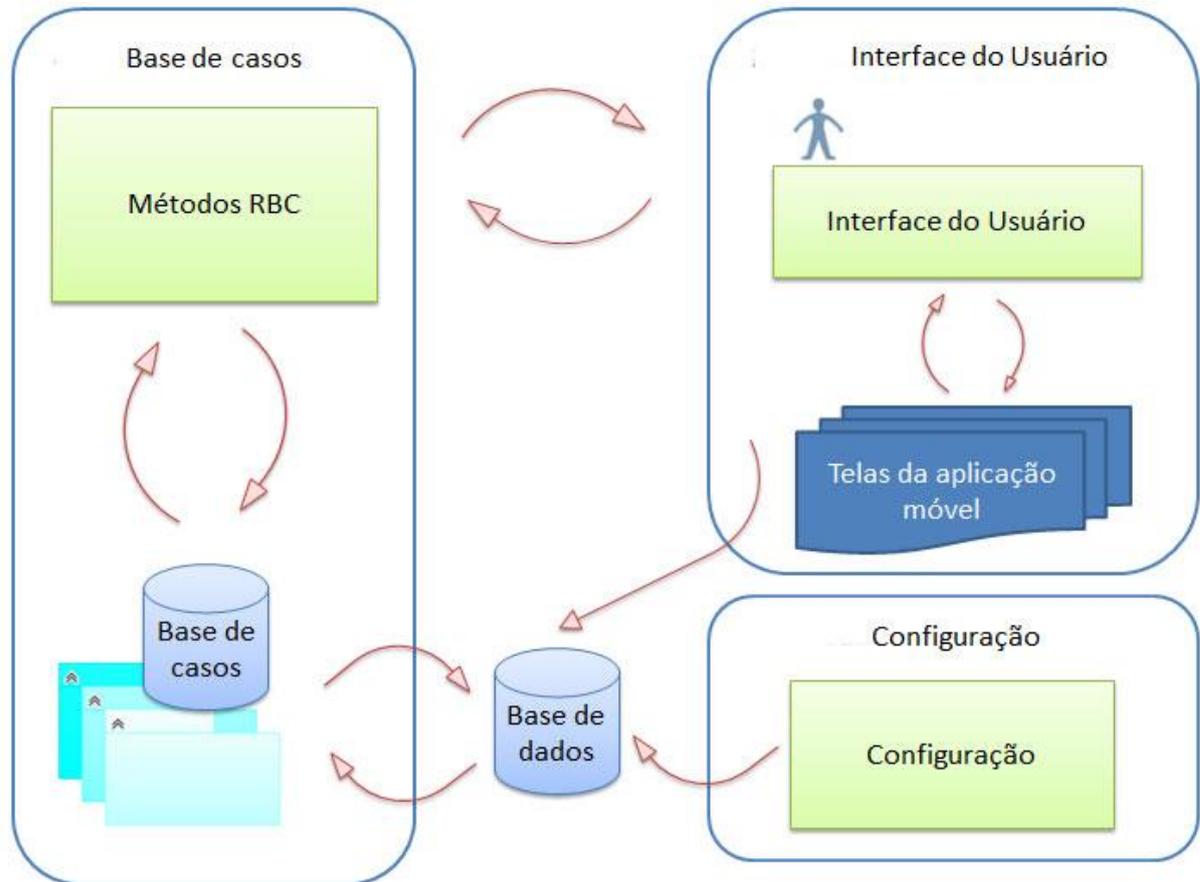


Figura 21 - Arquitetura base da aplicação proposta

Fonte: do Autor

O módulo da Base de Casos se encarregará da aplicação das técnicas de RBC sobre a base de casos e os dados contidos na base de casos e para a geração de novos casos ou reutilização de casos já existentes. Esta base de casos deverá conter dados relevantes para a pesquisa sobre casos contidos na Base de Casos. A sessão de configuração permitirá o cadastro de informações relevantes (Sintomas e tratamento de forma manual) para o funcionamento da aplicação. A interface do usuário fará comunicação com o módulo da Base de Casos para as pesquisas referentes aos dados para o diagnóstico, com a base de dados para o cadastro e visualização de informações relevantes para o usuário (cadastros de Espécies, Cadastro de Raças, Cadastro de Animais e, por fim, a tela de diagnóstico), além disso, fará a tradução para as telas da aplicação móvel.

5.2. Tecnologias utilizadas

Para o desenvolvimento deste trabalho foram utilizadas as seguintes tecnologias:

- **Apache Cordova / Phonegap:** *Framework* para desenvolvimento de aplicações multiplataforma. A aplicação consiste em páginas HTML, recursos (CSS, imagens etc) e código Javascript;
- **Web SQL:** o PhoneGap Storage usa a implementação Web SQL, do HTML 5, para criar bancos de dados locais;
- **Visual Studio 2013:** a IDE Visual Studio 2013 (VS2013) para o desenvolvimento e compilação do código para o padrão suportado para o SO Android e criação dos diagramas utilizados no Capítulo 4;
- Desenvolvimento focado para plataforma Android.

5.3. Metodologia e Desenvolvimento

Para o desenvolvimento da aplicação, inicialmente foi realizado um estudo baseado na fundamentação teórica (Capítulo 2), com o foco de adquirir conhecimento sobre os conceitos e fundamentos do RBC aplicado ao desenvolvimento de aplicações.

A partir disso, foram levantados os requisitos para a aplicação (Capítulo 4) e, com base nisso, a aplicação foi desenvolvida. As telas apresentadas nesta seção mostram e descrevem o funcionamento da aplicação como um todo.

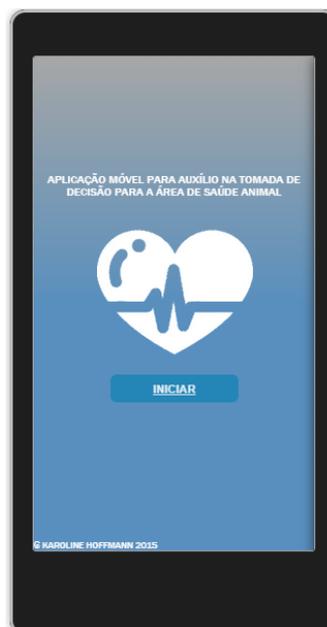


Figura 22 - Tela inicial da aplicação

Fonte: do autor

Na Figura 22 é apresentada a tela inicial da aplicação, onde é feita a execução de *scripts*. Já na Figura 23 é mostrada a tela de Menu da aplicação, contendo os botões que dão acesso a cada funcionalidade existente na aplicação, sendo: Cadastro de Espécie, Cadastro de Raça, Cadastro de Animal, Configuração, Diagnóstico e Sobre (que contém informações sobre a aplicação).



Figura 23 - Menu da aplicação

Fonte: do autor

Na Figura 24 é exibida a tela de filtro do cadastro de espécie que vai servir de padrão para todas as telas do sistema, contendo um botão para redirecionar ao menu, um botão para atualizar lista de registros e um botão para adicionar novo registro.

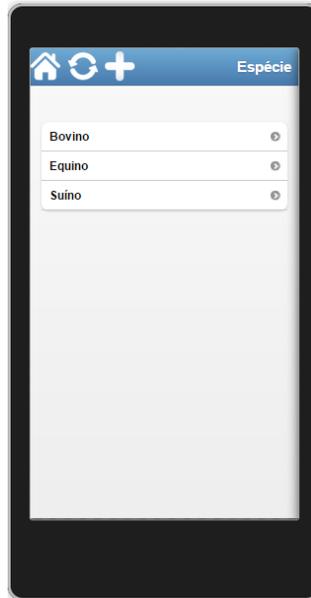


Figura 24 - Tela de listagem de registros
Fonte: do autor

Na Figura 25 é mostrada o a tela de cadastro de espécie que contém a descrição, que é um campo de texto para preenchimento da descrição da raça, seguido pelo botão “Salvar”. O botão indicado por uma seta retorna para a tela de filtro cancelando a operação.



Figura 25 - Cadastro de Espécie
Fonte: do autor

Na Figura 26 é exibida a tela de cadastro de Raças que contém os campos de Espécie (*combobox* onde será feita a seleção da espécie da raça) e descrição (campo de texto para preenchimento da descrição da raça). Seguidos pelo botão “Salvar”. O botão indicado por uma seta retorna para a tela de filtro cancelando a operação.



Figura 26 - Cadastro de Raça

Fonte: do autor

Na Figura 27 é exibida a tela de cadastro de Animal que contém os seguintes campos:

- Espécie: *combobox* onde será feita a seleção da espécie do animal;
- Raça: *combobox* onde será feita a seleção da raça do animal;
- Descrição: campo de texto para preenchimento da descrição do animal;
- Idade (em meses): campo de texto para preenchimento do tempo de vida do animal;
- Peso (em gramas): campo de texto para preenchimento do peso do animal;
- Proprietário: campo de texto para preenchimento do proprietário do animal;
- Telefone: campo de texto para preenchimento do telefone de contato do responsável pelo animal;

- Sexo: seleção do gênero do animal.

Além disso, possui o botão “Salvar” e o botão indicado por uma seta, que retorna para a tela de filtro cancelando a operação.

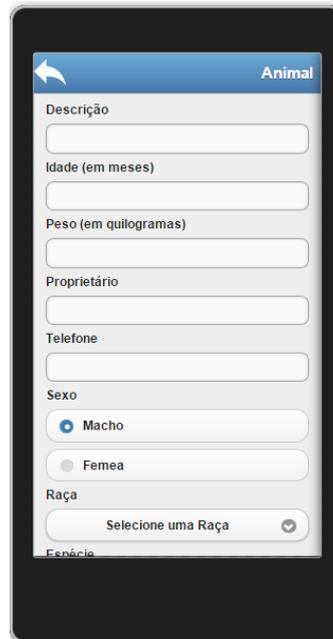


Figura 27 - Cadastro de Animal

Fonte: do autor

Na Figura 28, é exibida a tela inicial do diagnóstico do animal que contém os campos:

- Data: campo de data para selecionar a data da consulta;
- Animal: *combobox* onde será feita a seleção do animal;
- Observação: campo de texto para preenchimento de observações referentes a consulta.

Possui também “Pesquisar Sintomas”, que redireciona para a tela de seleção de sintomas. E o botão indicado por uma seta retorna para a tela de filtro cancelando a operação.



Figura 28 - Tela inicial do diagnóstico
Fonte: do autor

Na Figura 29 é exibida a tela de Sintomas que contém uma lista selecionável de sintomas relacionados ao diagnóstico. Possui o botão “Pesquisar Patologias”, que redireciona para a tela de seleção de patologias. O botão indicado por uma seta retorna para a tela anterior.

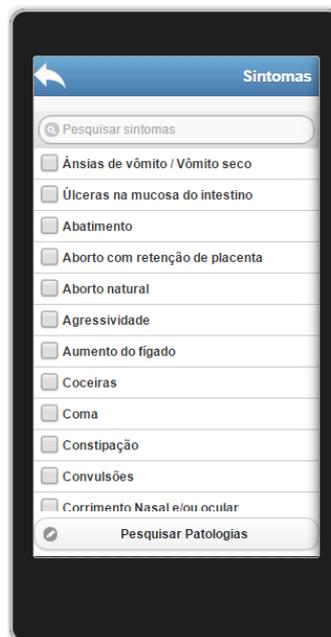


Figura 29 - Diagnóstico: Seleção de sintomas
Fonte: do autor

A Figura 30 apresenta a tela de Patologias que contém uma lista selecionável de patologias relacionadas aos sintomas selecionados na tela anterior, onde o item escolhido na lista redirecionará para a tela de tratamento. Possui o botão “Adicionar Tratamento Manual”, que redireciona para a tela de manutenção de tratamento manualmente. O botão indicado por uma seta retorna para a tela anterior.

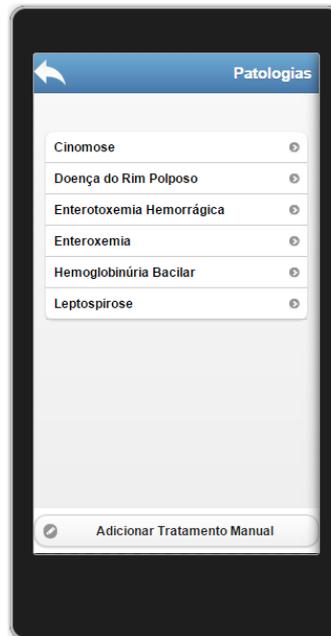


Figura 30 - Diagnóstico: Seleção de Patologia

Fonte: do autor

A Figura 31 apresenta a tela de Tratamento Manual que contém os campos:

- Data: campo que informa a data do diagnóstico;
- Animal: campo que informa o animal selecionado durante o diagnóstico;
- Sintomas: campo que informa os sintomas selecionados durante o diagnóstico;
- Patologia: campo de texto para preenchimento da descrição da Patologia;
- Medicação: campo de texto para preenchimento da descrição do Medicamento;
- Dosagem: campo de texto para preenchimento da dosagem do Medicamento;
- Frequência: campo de texto para preenchimento da frequência de aplicação do medicamento;

- Observação: campo de texto para preenchimento de observações referentes ao tratamento.

Possui o botão “Salvar”, que redireciona para a tela de Diagnóstico e exibe o tratamento adicionado. Essa tela serve de auxílio para quando não existem casos que se tornem adequados ao diagnóstico resultante. O botão indicado por uma casa retorna para a tela de menu.



A imagem mostra a interface de usuário de um aplicativo móvel, especificamente a tela de "Tratamento Manual". No topo, há uma barra azul com um ícone de casa à esquerda e o título "Tratamento Manual" à direita. Abaixo, o formulário contém os seguintes campos:

- Data: 13/07/2015
- Animal: 78541254
- Sintomas: Úlceras na mucosa do intestino, Abatimento, Convulsões, Decúbito lateral
- Patologia: [campo de texto vazio]
- Medicação: [campo de texto vazio]
- Dosagem: [campo de texto vazio]
- Frequência: [campo de texto vazio]
- Observação: [campo de texto vazio]

Na base da tela, há um botão branco com o texto "Confirmar Tratamento" e um ícone de seta para a esquerda.

Figura 31 - Tratamento Manual

Fonte: do autor

E por fim na Figura 32, é mostrada a tela de Tratamento relacionada a patologia. Possui também, como padrão em todas as telas, o botão “Salvar Tratamento” que salva o tratamento na base. O botão indicado por uma seta retorna para a tela anterior.

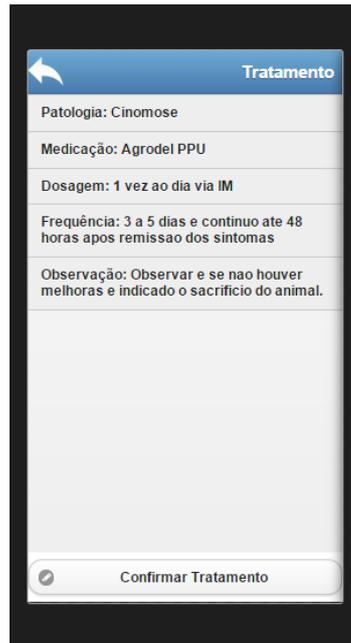


Figura 32 - Diagnóstico: exibição de tratamento

Fonte: do autor

5.3.1. Aplicação do RBC na aplicação proposta

Para elaborar a solução proposta neste trabalho, inicialmente se fez necessário um estudo sobre todas as técnicas disponíveis do Raciocínio Baseado em casos, com a finalidade de entender os conceitos e fundamentos, além de um estudo sobre desenvolvimento de aplicações móvel, como já citado.

Pode-se definir que a aplicação foi dividida em três partes: a primeira faz relação aos cadastros básicos, onde o usuário interage com a aplicação para alimentar as primeiras informações relevantes à aplicação. A segunda caracteriza a criação da definição de casos da aplicação. E, a terceira define o tratamento, recuperado a partir do ciclo do RBC, novos casos que são criados na aplicação, a partir de novos diagnósticos.

A Figura 33 mostra o fluxo do ciclo da aplicação agora direcionado para o ciclo RBC.

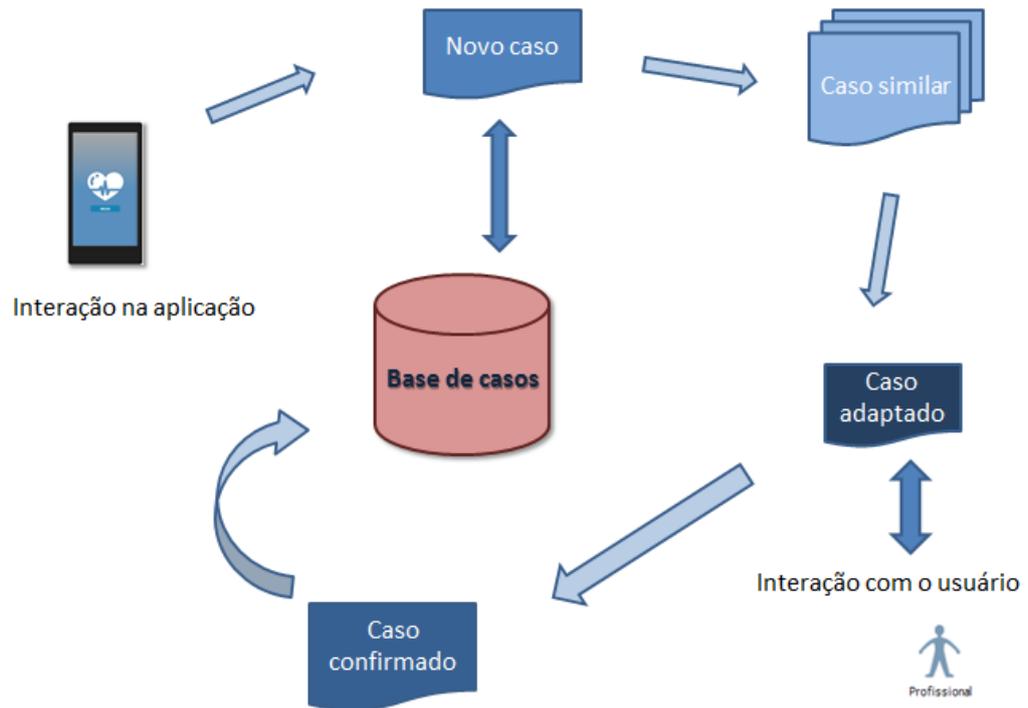


Figura 33 - Fluxo do processamento RBC na aplicação

Fonte: do autor

Neste trabalho, a definição de um caso parte do início e do cadastro de um diagnóstico, onde ambos ficam armazenados no banco de dados da aplicação. Na medida em que, são criados novos diagnósticos na aplicação, uma nova instância de caso surge, através do ciclo do RBC.

Na etapa de criação de diagnóstico e adaptação de um caso é quando existe a interação do usuário com o caso, confirmando o caso retornado. Por fim, o novo caso (que é definido a partir do diagnóstico) é armazenado na base do sistema para que seja retornado em um momento posterior. Quando os casos retornados não fazem jus à decisão que o profissional tomaria, existe a opção da criação do diagnóstico de forma manual, onde será adicionado um novo caso à base de casos que não terá sido relacionado a nenhum caso existente.

Para avaliar a similaridade dos tratamentos já existentes levou-se em conta a similaridade gravada no registro e os sintomas relacionados a ela, onde cada sintoma recebe um peso um na finalização do diagnóstico, onde se leva em conta o número de sintomas similares entre os casos.

A similaridade é uma questão bastante especial e requer que um estudo específico seja realizado para cada domínio onde o RBC é aplicado, com o enfoque podendo ser dado através de metas, restrições ou classificações.

A técnica de métrica de similaridade mais usual ainda é a do Vizinho mais próximo, que basicamente consiste de um método que faz a comparação do novo caso com os casos já existentes na base de casos, verificando cada atributo, onde a medida de similaridade é multiplicada por um fator denominado peso definido.

A representação da fórmula que pode determinar o grau de similaridade pode ser vista na Figura 34.

$$\text{similaridade}(N, F) = \sum_{i=1}^n f(N_i, F_i) * W_i$$

Figura 34 - Fórmula para determinar o grau de similaridade

Fonte: URNAU, 2011

Legenda:

N – Novo Caso;

F – Caso existente na Base de Casos;

n – Número de atributos;

i – Atributo individual, que varia de 1 a n;

f – Função de similaridade para o atributo i nos casos N e F;

w – Peso dado ao atributo i;

Esta função é aplicada a todos os casos existentes na BC para ser possível obter a classificação da similaridade. A utilização de pesos auxilia a recuperação de casos por similaridade entre os atributos comuns. E a outra contribuição está na possibilidade de desconsiderar aqueles que não possuem peso no momento da recuperação, o que contribui para a definição da similaridade entre dois casos (URNAU, 2011).

5.3.2. Avaliação do especialista

A etapa de validação junto ao especialista pode ser considerada uma das principais partes para o desenvolvimento do trabalho, considerando que, neste momento que se tem uma avaliação e validação de um profissional com conhecimento específico da aplicação, e entende-se que a contribuição de um especialista é essencial para compor os resultados atingidos com o desenvolvimento da aplicação.

Pode-se perceber junto ao especialista que ele não obteve problemas em interagir com a aplicação, definindo como de boa usabilidade.

A aplicação foi disponibilizada ao especialista humano e este utilizou algumas situações já vivenciadas durante a experiência profissional.

Como pode ser visto na seção 5.3.1, de acordo com os valores definidos para o cálculo de similaridade, como exemplo, pode ser obtido o seguinte resultado a partir do uso da aplicação móvel pelo especialista, como pode ser visto na Tabela 16. A pesquisa foi feita por alguns sintomas específicos, onde se entende que a importância (Peso) seja Alta, trazendo como resultado as patologias associadas aos sintomas e, conseqüentemente, o tratamento para cada patologia.

Tabela 16 - Exemplo de pesquisa a ser feita

Problema	
Sintomas – Atributos	Importância – Peso
Temperatura acima 41°	Alta
Diarreia	Alta
Corrimento nasal e/ou ocular	Alta
Patologias resultantes da seleção de Sintomas	
Cinomose, Leptospirose	
Solução: Tratamento com maior similaridade com a validação do especialista	
Agrodel PPU Frequência: 1 vez ao dia via IM Duração: 3 a 5 dias e continuo até 48 horas após remissão dos sintomas Dosagem: 0.03 - 0.06 mL/kg	

Fonte: do autor

De acordo com a Figura 35, pode-se ter uma perspectiva da avaliação do especialista, através de um questionário, perante a utilização do sistema. Pode ser observado que o tratamento resultante do processo de diagnóstico da aplicação assemelhou-se ao processo decisório que o especialista teria em relação aos sintomas observados, onde o mesmo aprovou os resultados e a usabilidade da aplicação para auxílio de diagnóstico e tratamento.

UNISC		Ficha de Avaliação do Especialista		20/06/2015
Trabalho de Conclusão de Curso Ciência da Computação Universidade de Santa Cruz		Aluna: Karoline Hoffmann		
Avaliação	Patologia / Tratamento profissional	Sintomas	Patologia / Tratamento Resultante	
Aprovado	Cinomose	Temperatura acima 41º, Diarréia, Corrimento nasal e/ou ocular, Sem apetite, Abatimento	Cinomose, Leptospirose	
Aprovado	Leptospirose	Temperatura até 41º, Sem apetite, Diarréia, Sangue na Urina, Abatimento	Cinomose, Leptospirose	
Aprovado	Raiva	Salivação intensa. Problemas de locomoção. Fezes secas. Sangue nas Fezes. Agressividade ou inuietação	Raiva	
Aprovado	Tétano	Narinas dilatadas, Tremores musculares, Constipação, prolapso da terceira pálpebra	Tétano	
Aprovado	Hemoglobinúria Bacilar	Ictericia, Dor abdominal, Abatimento, Falta de apetite, Febre,	Hemoglobinúria Bacilar	
Aprovado, procurar ser mais específico	Botulismo	Paralisia, problemas de locomoção e Abatimento	Botulismo, Hepatite Necrótica Infecciosa	
Considerações	O sistema funciona, deveria ter uma forma de adicionar mais itens, já que em alguns casos não foram encontrados o tratamento. O acesso aos cadastros é fácil e a utilização foi feita de maneira que retornasse o tratamento de acordo com os sintomas.			

Figura 35 - Ficha do especialista

Fonte: do autor

A especialista, ao final de sua experiência na utilização da aplicação, deu um parecer final destacando a boa usabilidade da aplicação e o fato de ter uma ferramenta que possa trazer eficiência e certeza para o seu desempenho profissional, auxiliando no processo decisório, foi considerado muito positivo e atraente.

5.3.3. Considerações

A partir da base de casos e dos dados mencionados, foi possível executar aplicação e geração de casos a partir de casos já solucionados, mas durante o desenvolvimento notou-se que será necessário ampliar a base de casos de maneira que garanta que o sistema seja capaz de cobrir todo o domínio utilizando um maior número de registros.

Durante o desenvolvimento pode-se constatar que a aquisição do conhecimento, mesmo sendo uma tarefa difícil e demorada na maioria dos sistemas inteligentes, com o RBC torna-se mais simplificado, já que o conhecimento é adquirido através do especialista humano e o diagnóstico é definido a partir desse conhecimento e com casos similares já armazenados, diferente de muitos outros sistemas que utilizam regras específicas para compor uma aplicação.

A fase de validação da aplicação pode-se qualificar como de suma importância para a execução e finalização deste trabalho, pois a partir daí pode-se comprovar a aplicabilidade e usabilidade do sistema desenvolvido.

6. CONSIDERAÇÕES FINAIS

Este trabalho de conclusão teve início na concepção de uma aplicação que pudesse auxiliar os profissionais da área de saúde animal que trabalham em locais remotos ou até mesmo na tarefa de elucidar um diagnóstico, levando em conta que muitas vezes o tratamento e o diagnóstico de patologias podem se tornar duvidosos. Quando a medicina veterinária entra em casos de prevenção, controle e tratamento de doenças, surgem dificuldades e dúvidas em relação ao correto diagnóstico de determinadas patologias, visto que existem casos em que se apresentam sintomas semelhantes que podem por o diagnóstico em suspeita, onde uma aplicação que auxilia nessa decisão é de grande valia.

Para isso, foram necessários estudos relacionados a área da Inteligência Artificial, buscando técnicas que mais se adequariam a resolução da proposta deste Trabalho de Conclusão. Com base nos estudos, pode-se verificar que a técnica de Raciocínio Baseado em Casos se adequaria melhor ao problema proposto, além de uma ampla pesquisa na melhor forma de desenvolver essa aplicação, já que hoje o mercado de desenvolvimento de aplicativos móveis está em evolução.

Além da definição da parte teórica da Inteligência Artificial fez-se necessário uma ampla pesquisa em relação à parte de desenvolvimento móvel, com o objetivo de definir a melhor linguagem para o desenvolvimento levando em consideração o tempo de aprendizado para o mesmo. A escolha do desenvolvimento com o *FrameWork PhoneGap / Apache Cordova* se deu não somente pela facilidade de utilização, instalação e curva de aprendizado, mas também pela sua facilidade e disponibilidade de documentação para pesquisa.

Durante o desenvolvimento e após várias pesquisas, perceberam-se várias vantagens e desvantagens na sua utilização e desenvolvimento, tanto de parte técnica como de usabilidade. Uma das vantagens, sem dúvida, vem da relação que em domínios muito complexos, a tarefa de extrair o conhecimento, fazendo uso de experiências passadas, torna a utilização da técnica do RBC muito mais simples para a atualização da base de casos e a aquisição de conhecimento devido a interação do usuário, o especialista humano.

Como contribuição para a área de Ciência da Computação, este trabalho mostra a possibilidade de utilizar a facilidade de desenvolvimento para dispositivos móveis em

conjunto com uma técnica da Inteligência Artificial, que permite aproveitar a mobilidade de uma aplicação especialista que permite acompanhar o profissional em qualquer lugar.

Fazendo uma comparação breve em relação a todos os aplicativos pesquisados para o auxílio do desenvolvimento deste trabalho, pode-se observar que o principal diferencial da aplicação, além de permitir a mobilidade, é a possibilidade de fazer a pesquisa diretamente pelo sintoma que está sendo diagnosticado, onde em demais aplicações já estão relacionadas com o medicamento a ser utilizado ou a patologia que foi diagnosticada, tornando a busca por uma solução muito mais dinâmica e eficaz.

Pode-se concluir que os objetivos gerais deste trabalho foram alcançados, visto que foi possível desenvolver uma aplicação para dispositivos móveis que permitisse o diagnóstico de animais levando em consideração os sintomas apresentados comparados a diagnósticos já existentes e a avaliação de um profissional especialista (Médico Veterinários), que interagindo com a aplicação conseguiu com sucesso resolver os problemas propostos.

Não se pode negar que fazer a pesquisa dos dados e obter dados junto ao especialista humano, foi um desafio e este teve que ser feito quase que todo manualmente, até por isso a quantidade de dados não foi tão significativa para os testes.

A aplicação desenvolvida é sem dúvidas de pequeno porte, mas com a utilização da técnica do RBC torna-a uma ferramenta eficaz e eficiente no auxílio dos profissionais da área específica.

6.1. Trabalhos Futuros

Ao finalizar este trabalho pode-se definir melhorias relevantes à aplicação que pode tornar sua utilização mais ampla, sendo uma delas a usabilidade da aplicação em diversos sistemas operacionais de dispositivos móveis, já que o teste desta aplicação se baseou no Sistema Operacional Android. Além disso, a criação de uma plataforma *Web* também seria de grande valia para os profissionais poderem utilizar diretamente em seus consultórios.

Mas a sugestão que seria mais significativa para a usabilidade desta aplicação, é a integração da aplicação móvel com *WebService*, onde a comunicação se faria através de serviços e esses dados estariam na nuvem para a utilização através de qualquer dispositivo ou computador, tornando a base de casos muito mais ampla, considerando que hoje os dados ficam somente em cada dispositivo.

7. REFERÊNCIAS

- AAMONDT, A. PLAZA, E. *Case-based Reasoning: Foundational Issues, Methodological variations, and System Approaches*. AICOM, 1994.
- ABEL, Mara. *Um Estudo sobre Raciocínio Baseado em Casos*. Trabalho Individual (Pós Graduação em Ciência da Computação). UFRGS: Porto Alegre, 1996.
- ANDROID. Desenvolvido pela empresa Google. [2013?]. Apresenta informações gerais sobre o Android. Disponível em: <<http://www.android.com/>>. Acesso em 20 ago. 2013.
- ANDROID DEVELOPERS. *Desenvolvido pela empresa Google*. [2013?]. Disponibiliza ferramentas para desenvolvedores com interesse no Android. Disponível em: <<http://developer.android.com/sdk/index.html>>. Acesso em 20 ago. 2013.
- ANDROID STUDIO. *Site destinado a desenvolvedores Android*. Disponível em: <<http://developer.android.com/sdk/installing/studio.html>>. Acesso em 02 nov. 2013.
- BARCELLOS, Carla D. et al. *Sistema de Recomendação Acadêmico para Apoio a Aprendizagem*. CINTED-UFRGS, Novas Tecnologias na Educação. V. 5 N° 2, Dezembro, 2007.
- BEZERRA, E. *Princípios de Análise e Projeto de Sistemas com UML*. Rio de Janeiro: Editora Campus, 2006.
- BITTENCOURT, Luís F.; LORENZI, Fabiana. *Um sistema de recomendação de compras baseado em casos*. Canoas, 2010.
- CAMPOS, M. M. de, SAITO, K. *Sistemas inteligentes em controle e automação de processos*. Rio de Janeiro: Editora Ciência Moderna Ltda., 2004.
- CARVALHO, Alexey. *Raciocínio baseado em casos aplicado ao processo decisório*. Revista de Ciências Gerenciais. 12(16); 25-35; 2008.
- CAZELLA, Sílvio C.; CHAGAS, Irismar C. das; BARBOSA, Jorge L. V. *Um modelo para recomendação de artigos acadêmicos baseado em filtragem colaborativa aplicado a ambientes moveis*. UFRGS: Novas tecnologias na Educação. Porto Alegre: Revista UFRGS, 2008.
- CAZELLA, Sílvio C.; NUNES, Maria A. S. N.; REATEGUI, Eliseo B. *A Ciência da Opinião: Estado da arte em Sistemas de Recomendação*. JAI: Jornada de Atualização em Informática da SBC. Rio de Janeiro: Editora da PUC Rio, 2010.
- CINDRAL, BELINE. *Sistemas operacionais para celulares*. 2011. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2011/01/afinal-o-que-e-android.html>>. Acesso em: 30 de outubro de 2013.
- COELHO, Helder. *Inteligência Artificial em 25 lições*. Lisboa: Fundação Valouste Gulberkian, 1995.

CRUZ, Bruno H. A.. *Desenvolvimento de uma aplicação embarcada em celular visando controle de robô via wi-fi*. Trabalho de conclusão de curso. UNISC: Santa Cruz do Sul, 2010.

DANTAS, E. R et al. *O Uso da Descoberta de Conhecimento em Base de Dados para apoiar a Tomada de Decisões*. SEGeT – Simpósio de Excelência em Gestão e Tecnologia. UNIPÊ: Pernambuco, 2008.

DEVMEDIA. Conteúdo para desenvolvedores Java, asp.net, mobile, e outras linguagens. Disponível em: <<http://www.devmedia.com.br/mono-for-android-revista-net-magazine-94/23665>>. Acesso em 02 nov. 2013.

FILHO, Valmir Macário. *E-Recomemender: Sistema Inteligente de Recomendação para Comércio Eletrônico*. 2006. Dissertação de Mestrado (Engenharia da Computação) Recife: EPP da UP, 2006.

FURLAN, José Davi. *Modelagem de Objetos através da UML*. São Paulo: Makron Books, 1998.

JESUS, R. P.; ESCOBAR, M. *Desenvolvimento de um Sistema de Recomendação de eventos com uso de Geolocalização*. 2011. Projeto Final (Análise e Desenvolvimento de Sistemas). Ulbra: Canoas, 2011.

KOLODNER, Janet. *Case Basead Reasining*. San Mateo: Morgam Kaufman Publishes, 1993.

LECHETA, Ricardo R.. *Google Android: Aprenda a desenvolver aplicações para dispositivos móveis com o Android SDK*, São Paulo: Novatec, 2010. ISBN 978-85-7522-244-7.

LECHETA, Ricardo R.. *Google Android para Tablets: Aprenda a desenvolver aplicações para o Android De smartphones a tablets*, São Paulo: Novatec, 2012. ISBN 978-85-7522-292-8.

LICHTNOW D. et al. *O Uso de Técnicas de Recomendação em um Sistema para Apoio à Aprendizagem Colaborativa*. Revista Brasileira de informática na educação (RBIE), 14(3):49–59.

LORENZI, F. *Uso da metodologia de raciocínio baseado em casos na investigação de irregularidades nas internações hospitalares*. 1998. Dissertação de Mestrado (Curso de Pós-Graduação em Ciência da Computação). Porto Alegre: CPGCC da UFRGS, 1998.

MODRO, Nilson Ribeiro. *Sistema Inteligente de Monitoramento e Gerenciamento Financeiro para Micro e Pequenas Empresas*. 2000. Dissertação (Mestrado em Engenharia de Produção) – UFSC. Florianópolis. P. 28-37.

MONO DEVELOPED. Ferramenta Mono para desenvolvimento de aplicações Android. Disponível em: <http://monodevelop.com/Download/Mono_For_Android>. Acesso em 08 nov. 2013.

MORAIS, Dyego C. S. et al. *Sistema Móvel de Apoio a Decisão Médica Aplicado ao Diagnóstico de Asma - InteliMED*. VIII Simpósio Brasileiro de Sistemas de Informação. UFPE: Recife, 2012.

NEWTON, Fernando; SCHUTZ, Sérgio Murilo; SPIEKER, Sandra Schmidt. *Sistema Auxiliar de Diagnósticos e Tratamento Dermatológico - Dermapet*. CBCComp – IV Congresso Brasileiro de Computação. UFRGS: Porto Alegre, 2009.

NIELSEN. *O consumidor móvel*. 2013. Disponível em: <<http://www.nielsen.com/content/dam/corporate/Brasil/reports/2013/Estudo-Consumidor-Mobile-Jun13.pdf>>. Acesso em: 30 de outubro de 2013.

OHA. *Site institucional da empresa Open Hantset Alliance*. Disponível em: <<http://www.openhandsetalliance.com>>. Acesso em 30 out. 2013.

PACHECO, Edson J. et al. *SADTR – Sistema de Auxílio de Diagnóstico em Tempo Real*. XIII Congresso Brasileiro em Informática em Saúde. PUCPR-UFPR: Curitiba, 2012.

PAIVA, Lilian R. M. et. Al. *Aplicação para dispositivos móveis utilizando tecnologias interativas. A Realidade Virtual e Aumentada aplicada ao estudo da anatomia humana*. X Encontro Anual de Computação – EnAComp. Uberlândia, 2013.

PILGER, Diego J.; HUGO, Marcel. *Aplicação de raciocínio baseado em casos no suporte a decisão de um sistema web de help desk*. Revista eletrônica do Alto do Vale Itajaí. LUGAR, 2012.

PHONEGAP. Conteúdo para desenvolvedores de aplicações móveis. Disponível em: <<http://phonegap.com>>. Acesso em 05 nov. 2013.

PLOTEGHER, Silvio L.; FERNANDES, Marcio M. *Raciocínio Baseado em casos aplicado a um sistema de diagnostico remoto*. XXV Congresso da Sociedade Brasileira de Computação. Unisinos: São Leopoldo, 2005.

RABELLO, Ramon R.. *Android: um novo paradigma de desenvolvimento móvel*. Pernanbuco, 2008

REATEGUI, E. B; CAZELLA, S. C. *Sistemas de Recomendação*. XXV Congresso da Sociedade Brasileira de Computação. Unisinos: São Leopoldo, 2005.

REZENDE, S. O. *Sistemas Inteligentes – Fundamentos e Aplicações*. Barueri: Editora Manole, 2003.

RIESBECK, C.K. SCHANK, R. C. *Inside Case-Based Reasoning*. Lawrence Erlbaum, 1989.

RUSSEL, Stuart; NORVING, Peter. *Inteligência Artificial*. Rio de Janeiro, RJ: Campus, 2004.

SCHAFER, J. Ben; KONSTAN, Joseph; *Recommender Systems*. Conference on electronic Commerce. Minneapolis: Proceedings, 2000.

SERRANO, Maurício. *Um Sistema de Recomendação para Mídias Baseado em Conteúdo Nebuloso*. 2003. Dissertação (Pós Graduação em Ciência da Computação) - UFSCar, São Paulo, 2003.

SIBALDO, Maria A. A.; SALES, Thiago B. M.; CALADO, Ivo A. A. R. *MobileGraW: uma aplicação para dispositivos moveis baseado em comunidades virtuais de aprendizado com suporte a recomendação*. XVIII Simpósio Brasileiro de Informática na Educação. LUGAR, 2007.

SILVA, Luiz C. N. da; NETO, Francisco M. M.; JUNIOR, Luiz J.. *MobiLE: Um Ambiente Multiagente de Aprendizagem Móvel para Apoiar a Recomendação Sensível ao Contexto de Objetos de Aprendizagem*. XXII Simpósio Brasileiro de Informática na Educação. Aracajú, 2011.

SILVA, Renata de P. et al. *Proposta de um Modelo RBC para Construção de um Sistema de Apoio ao Diagnóstico Médico*. IV SBQS – V Workshop de Informática Médica. Brasília, 2005.

SMITH, Dave; FRIESEN, Jeff. *Receitas Android: Uma abordagem para resolução de problemas*, Rio de Janeiro: Ciência Moderna, 2012. ISBN 978-85-399-0264-4.

STATCOUNTER. *Página desenvolvida para retorno de resultados de visitantes*. 2013. Disponível em: <http://gs.statcounter.com/#mobile_os-ww-yearly-2009-2013>. Acesso em: 31 de outubro de 2013.

THÉ, M.A.L. *Raciocínio Baseado em Casos. Uma Abordagem Fuzzy para Diagnóstico Nutricional*. 2001. Dissertação (Mestrado em Engenharia de Produção). UFSC, Joinville, 2001.

TORRES, Roberto. *Personalização na Internet*. São Paulo, SP: Novatec, 2004.

URNAU, Eduardo. *Sistema de Raciocínio Baseado em Casos como Auxílio ao Processo de Tomada de Decisão Estratégica*. 2011. 131f. Dissertação de Mestrado (Programa de Pós-Graduação em Sistemas e Processos Industriais - Mestrado) – Universidade de Santa Cruz do Sul, Santa Cruz do Sul, 2011.

VET SMART. *Aplicativo para apoio técnico ao veterinário de Grandes Animais*. Disponível em: <<https://www.vetsmart.com.br/>>. Acesso em 26 jul. 2015.

VITORINO, Thiago Arreguy Silva. *Raciocínio Baseado em Casos: Conceitos e aplicações*. 2009. Dissertação (Mestrado em Ciência da Computação) – UFMG. Belo Horizonte.

VON WANGENHEIM, Cristiane Grese; VON WANGENHEIM, Aldo. *Raciocínio Baseado em Casos*. Barueri, SP: Manole, 2003. ISBN 85-204-1457-1.