

Um Sistema Pericial para Dispositivos Móveis com Base em Computação na Nuvem

An Expert System for mobile devices based on cloud computing

Carlos Carreto, Mário Baltazar

Unidade de Investigação para o Desenvolvimento do Interior
Instituto Politécnico da Guarda
Guarda, Portugal
ccarreto@ipg.pt, mariombaltazar@gmail.com

Resumo — Este trabalho descreve a implementação de um Sistema Pericial para dispositivos móveis Android, dirigido ao utilizador comum e com a capacidade de usar diferentes bases de conhecimento, selecionáveis pelo utilizador. O sistema usa uma arquitetura com base em computação na nuvem para facilitar a criação e distribuição de diferentes bases de conhecimento.

Palavras chave - Sistema Pericial; Android; Computação na Nuvem

Abstract — This paper describes the implementation of an Expert System for Android mobile devices, directed to the common user and the ability to use different knowledge bases, selectable by the user. The system uses a cloud computing-based architecture to facilitate the creation and distribution of different knowledge bases.

Keywords-Expert System; Android; Cloud computing;

I. INTRODUÇÃO

Um Sistema Pericial (SP) é um sistema computacional que procura simular processos de raciocínio e decisão de um perito humano. Estes sistemas são utilizados nas mais diversas áreas, embora seja na engenharia, na medicina e no mercado financeiro, onde se encontram mais exemplos.

Um SP típico é constituído por uma *interface com o utilizador*, através da qual são colocadas questões e obtidas respostas; uma *base de conhecimento* que contém a representação do conhecimento sobre determinada área; e uma *máquina de inferências*, capaz de analisar a base de conhecimento e juntamente com as questões e as respostas, obter conclusões. [1]

A representação do conhecimento pode ser realizada através de diferentes métodos, como por exemplo, regras de sistemas de produção, redes semânticas ou árvores de decisão [2]. O conhecimento deve ser suficiente para simular o processo de decisão do perito humano e estar organizado de forma a facilitar a sua utilização para consulta e resolução de problemas por parte de um utilizador não perito. A base de

conhecimento é normalmente gerada a partir de um *módulo de aquisição de conhecimento* [3].

A máquina de inferência é implementada com base no método escolhido para representar o conhecimento, permitindo que a mesma máquina de inferência possa processar diferentes bases de conhecimento desde que usem o mesmo método de representação.

A proliferação e evolução de dispositivos móveis tipo smartphone e tablet, faz destes dispositivos plataformas muito interessantes para a implementação de um SP, dando a oportunidade de massificar a utilização destes sistemas por parte do utilizador comum que passará a ter acesso a bases de conhecimento que poderá usar na resolução de problemas e tomada de decisão em atividades do dia-a-dia. Este trabalho é motivado por essa oportunidade e descreve a implementação de um SP para a plataforma Android, dirigido ao utilizador comum e com a capacidade de usar diferentes bases de conhecimento, selecionáveis pelo utilizador. O sistema é implementado com base em computação na nuvem, para facilitar o desenvolvimento do sistema, especialmente no que se refere à criação e distribuição de diferentes bases de conhecimento.

Ao contrário de outros SP para dispositivos móveis que são desenvolvidos com o intuito de usarem uma única base de conhecimento, o sistema proposto permite que seja o utilizador final a escolher, dentro das bases de conhecimento disponíveis, as que pretende descarregar e usar no SP instalado no seu dispositivo móvel.

O resto do artigo está organizado como se descreve a seguir. Na secção II são descritos alguns SP conhecidos para dispositivos móveis e as suas principais diferenças em relação ao sistema desenvolvido. Na secção III é apresentada a arquitetura do sistema desenvolvido com ênfase na plataforma de computação na nuvem que foi utilizada. A secção IV descreve os principais componentes do SP implementado. A secção V apresenta alguns testes e resultados e a secção VI termina o artigo com as conclusões.

II. TRABALHO RELACIONADO

Existem atualmente vários SP desenvolvidos para smartphones. Descrevem-se a seguir alguns deles semelhantes ao sistema proposto.

O Nutriton UCR [4] é um SP para monitorizar e gerir a dieta nutricional de um indivíduo. Obtém as características físicas indicadas pelo utilizador e avalia o estado nutricional deste, apresentando recomendações para este alcançar os requisitos nutricionais de uma dieta equilibrada.

O sistema usa uma arquitetura baseada em serviços formada por uma aplicação cliente e uma aplicação servidor. A aplicação cliente foi desenvolvida para a plataforma Android e é através dela que o utilizador envia as suas características físicas para a aplicação servidor onde se encontra o SP. Este foi implementado na linguagem Java com recurso à biblioteca JESS (JAVA Shell Expert System) [5].

O sistema contém uma única base de conhecimento representada por um script em regras JESS que é armazenada num ficheiro separado da aplicação para permitir a sua atualização independente do código da aplicação.

A aplicação Expert Advisor [6] é outra aplicação para a plataforma Android que implementa um SP que permite usar bases de conhecimento (baseadas em árvores de decisão), criadas com a aplicação XpertRule's Knowledge Builder e localizadas no servidor da empresa proprietária do sistema. Nesta aplicação a máquina de inferência está localizada na aplicação cliente e as bases de conhecimento podem ser descarregadas do servidor da empresa ficando armazenadas na aplicação cliente.

O Expertide2Go [7] é um SP que inclui uma ferramenta de criação de bases de conhecimento baseada em regras. A criação de bases de conhecimento para este sistema é feita através de uma applet disponibilizada para o efeito. Neste sistema, apesar da máquina de inferência estar localizada na aplicação móvel, é necessário uma ligação de dados pois não é possível guardar as bases de conhecimento na aplicação móvel. O sistema está implementado numa aplicação para a plataforma Android, a qual permite seleccionar uma base de conhecimento que deve estar acessível através de um endereço web. Nesta aplicação as bases de conhecimento são guardadas num ficheiro de texto na forma de regras.

O SESMED [8] é um SP para apoio a paramédico. O sistema foi desenvolvido com base na plataforma J2ME numa arquitetura cliente-servidor usada para a aplicação do dispositivo móvel comunicar com um servidor onde está a máquina de inferência. O utilizador usa a aplicação cliente para enviar os dados dos pacientes para o SP, o qual retorna o diagnóstico com base na base de conhecimento. Para representar o conhecimento o sistema utiliza linguagem Frame Logic e a base de conhecimento está armazenada numa base de dados relacional Oracle.

Os SP descritos apresentam uma arquitetura baseada em serviços, onde uma aplicação cliente necessita de estar ligada a uma aplicação servidor, em alguns casos para enviar dados e receber a resposta do SP e noutros para aceder a bases de conhecimento. Esta arquitetura tem o grande inconveniente de

ser necessário haver uma ligação de dados permanente, entre o cliente e o servidor para se poder usar o sistema.

O sistema proposto também usa uma arquitetura baseada em serviços, mas com a diferença de não ser necessária uma ligação de dados permanente entre a aplicação cliente e a aplicação servidor, pois o SP está implementado na aplicação móvel e a ligação ao servidor é feita pontualmente apenas para atualizar ou descarregar novas bases de conhecimento.

Outra característica importante do sistema proposto é que permite a seleção de bases de conhecimento de diferentes áreas, enquanto que a maioria dos sistemas analisados usam uma única base de conhecimentos de uma área específica.

III. ARQUITECTURA DO SISTEMA

O SP proposto foi implementado para a plataforma Android através de uma arquitetura baseado em serviços, utilizando a infraestrutura de computação na nuvem Google App Engine [9]. Esta infraestrutura do tipo PaaS (Platform as a Service) [10], proporciona uma plataforma e um conjunto de recursos que facilitam significativamente o desenvolvimento do sistema, nomeadamente no que se refere à implementação e manutenção do servidor, assim como ao processo de comunicação entre a aplicação cliente e o servidor. Para além disso, proporcionam um modelo de negócio potencialmente mais económico e escalável do que outras soluções comuns como as baseadas em Web Services, por exemplo.

Um elemento essencial no uso da plataforma Google App Engine é o Google Cloud Endpoints [11] que disponibiliza ferramentas, bibliotecas e recursos que permitem gerar APIs e bibliotecas cliente, com o intuito de simplificar o acesso aos dados de outras aplicações.

O Google Cloud Endpoints permite que do ponto de vista de desenvolvimento, a ligação cliente-servidor seja quase transparente, não sendo necessário fazer qualquer gestão da ligação, bastando apenas fazer chamadas de funções previamente programadas no backend da plataforma. Para além disso, proporciona recursos para a implementação de infraestruturas críticas como é o caso da autenticação, reduzindo assim o esforço de desenvolvimento.

A utilização da plataforma de computação na nuvem diminui significativamente o trabalho administrativo do sistema, como é o caso do balanceamento de carga, escalonamento e manutenção do servidor, pois todos esses serviços e recursos já estão implementados na plataforma Google App Engine.

A figura 1 apresenta uma visão geral da arquitetura do sistema proposto.

A arquitetura pode ser dividida em três componentes principais: a aplicação MyExpert Manager para a plataforma Android, a qual permite a criação e gestão de bases de conhecimento; a aplicação MyExpert, também para a plataforma Android (aplicação para o utilizador final onde está implementada a máquina de inferência do SP); e a coleção de bases de conhecimento do sistema, armazenadas na Google Cloud SQL [12] que é a infraestrutura de base de dados relacional usada na plataforma Google App Engine.

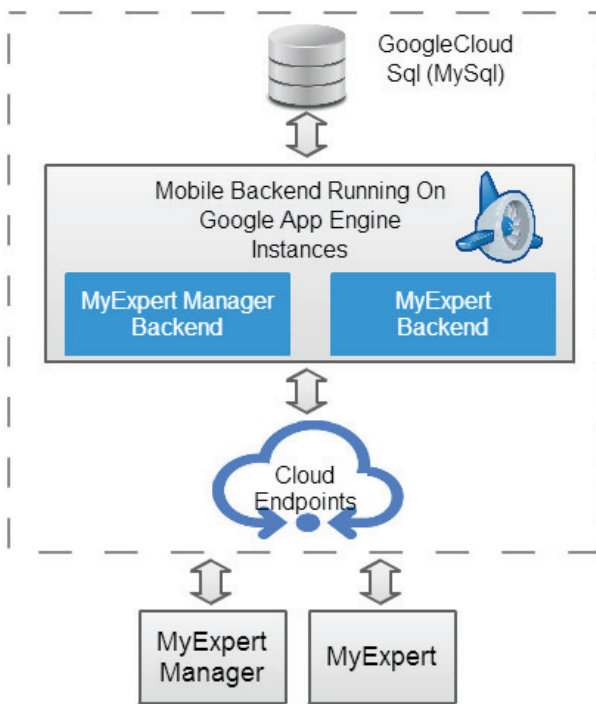


Figura 1. Arquitetura do Sistema

O acesso à base de dados Google Cloud SQL é realizado através do Mobile Backend do sistema, formado por duas aplicações desenvolvidas para o efeito e com as quais as aplicações MyExpert e MyExpert comunicam através de Cloud Endpoints.

A motivação para o Mobile Backend do sistema ser formado por duas aplicações distintas prende-se com aspetos de segurança. Para garantir a segurança nas aplicações de backend, é apenas dado o acesso à aplicação correspondente, permissão essa que é definida na configuração da plataforma, onde são especificadas as definições das aplicações backend. Para isso é gerado um Client ID OAuth 2.0 [10], gerado a partir do nome do package e do certificado de assinatura digital da aplicação cliente. Esse ID é inserido na aplicação cliente para que se possa autenticar na correspondente aplicação do Mobile Backend. Outro mecanismo extra de segurança, utilizado apenas na aplicação do Mobile Backend com a qual a aplicação MyExpert Manager comunica, foi a de associar a permissão de execução de funções a uma conta Google. Assim, o utilizador da aplicação MyExpert Manager terá que fazer primeiro o login na conta para poder aceder à base de dados da plataforma.

A figura 2 apresenta os dois tipos de atores intervenientes no sistema e os respetivos casos de uso. Os atores são o utilizador final e o gestor do sistema. O End User (utilizador final) é a pessoa que vai seleccionar e descarregar bases de conhecimento, verificar se existem atualizações de bases de conhecimento já descarregadas e usar essas mesmas bases de conhecimento no SP da aplicação. O System Manager (gestor do sistema) é a pessoa responsável pela criação das bases de conhecimento, atualizar as mesmas e publicá-las, deixando-as disponíveis para o utilizador final as poder descarregar.

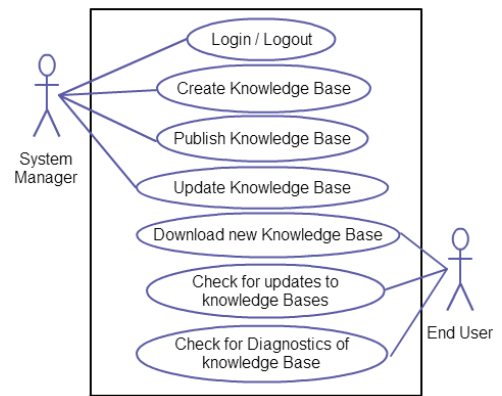


Figura 2. Atores do Sistema e respetivos casos de uso

IV. SISTEMA PERICIAL

O SP foi implementado com base na aplicação MyExpert e na aplicação MyExpert Manager, apresentadas na secção anterior. Enquanto a primeira permite ao utilizador seleccionar a base de conhecimentos a usar e implementa a máquina de inferência, assim como o módulo de explicação do SP, a segunda implementa o módulo de aquisição de conhecimento.

A. Aplicação MyExpert

A aplicação MyExpert é constituída pelos módulos mostrados na figura 3.

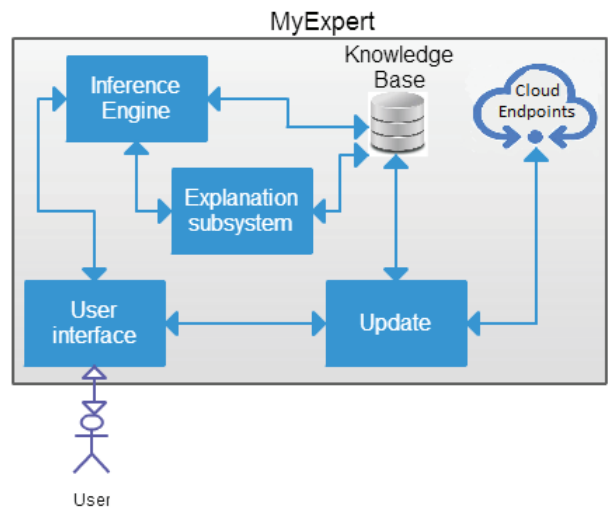


Figura 3. Módulos da aplicação MyExpert

Segue-se uma descrição de cada um dos módulos.

O módulo Cloud Endpoints refere-se à parte do sistema que comunica com a respetiva aplicação no Mobile Backend. Este módulo faz a gestão da ligação e consiste num conjunto de bibliotecas de funções, com o intuito de simplificar o acesso aos dados de outras aplicações.

O módulo Knowledge Base representa as bases de conhecimento descarregadas do servidor e armazenadas na

aplicação. As bases de conhecimento são armazenadas numa base de dados SQLite. Estas bases de conhecimento são transferidas da Google Cloud SQL onde estão armazenadas numa base de dados MySQL. As bases de dados têm a mesma estrutura, pelo que a transferência destas é realizada de forma transparente recorrendo aos recursos de Cloud Endpoints.

Uma base de conhecimentos é formada por um conjunto de regras com o seguinte formato

SE C_i OpB C_{i+1} OpB C_{i+2} ...
ENTÃO Conclusão

Onde:

C_i = Condição i (do tipo V_i OpR Valor)

V_i = Variável i

OpR = Operador Relacional (<, ≤, =, ≠, >, ≥)

OpB = Operador Booleano (E e OU)

A parte “SE” da regra é constituída por cláusulas. Estas cláusulas são condições que juntas formarão uma decisão que leva a uma conclusão. Cada cláusula pode ser formada por várias condições, condições estas que estão ligadas por operadores booleanos. Cada condição é formada por uma variável, um operador relacional e um valor.

A parte “ENTÃO” da regra constitui a conclusão a que se chega se a combinação das condições for verdadeira.

A figura 4 mostra o diagrama de Entidade-Relacionamento correspondente ao armazenamento das bases de conhecimento.

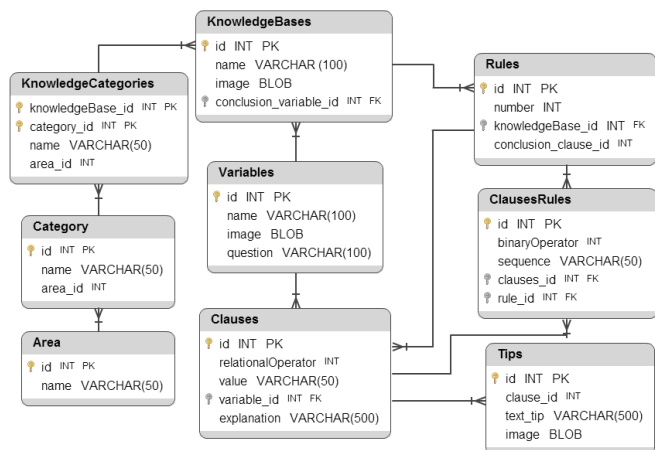


Figura 4. Diagrama Entidade-Relacionamento das bases de conhecimento

Este diagrama é igualmente válido para a base de dados da aplicação móvel. Como se pode ver no diagrama, para cada base de conhecimento é guardada, além de um nome e de uma imagem, a variável de conclusão da base de conhecimento. Cada regra (armazenada na entidade Rules) tem várias cláusulas (entidade ClausesRules) e cada cláusula tem uma variável. Existem ainda as entidades Area e Category que permitem separar as bases de conhecimento por diferentes áreas e categorias, o que permite aos utilizadores finais localizarem mais facilmente as bases de conhecimento do seu interesse.

O módulo Inference Engine é responsável pelo processamento da base de conhecimentos recorrendo a raciocínio e planeamento lógico. Existem dois mecanismos possíveis para implementar uma máquina de inferência, o encadeamento para frente e o encadeamento para trás (sendo também possível combinar os dois). Neste caso foi implementado o mecanismo de encadeamento para trás com base no algoritmo básico descrito em [3]. O mecanismo consiste em começar por escolher hipoteticamente uma conclusão para o raciocínio e a partir daí solicitar mais informações ao utilizador para verificar se as cláusulas que levam a essa conclusão se verificam ou não, de modo a poder confirmar a conclusão. Se a hipótese não se confirmar, a máquina de inferência escolhe nova conclusão, recorrendo aos dados recolhidos até ao momento, e repete o processo até que uma conclusão seja confirmada ou até não haver mais conclusões possíveis.

O módulo Explanation gera a linha de raciocínio seguida pela máquina de inferência para chegar à conclusão. Caso o utilizador o solicite, a linha de raciocínio é apresentada na interface.

O módulo User Interface é responsável pela interação entre o utilizador e a máquina de inferência, com base em perguntas e respostas. O módulo também permite ao utilizador interagir com a plataforma de computação na nuvem para atualizar e descarregar novas bases de conhecimento. Permite ainda visualizar as explicações geradas pelo módulo de explicação.

A figura 5 apresenta dois ecrãs exemplificativos da interação entre o utilizador e a máquina de inferência. O ecrã da esquerda corresponde a uma questão colocada ao utilizador à qual deve responder com Sim ou Não, para continuar o processo de inferência. O ecrã da direita mostra a apresentação da conclusão obtida pela máquina de inferência. Algumas conclusões poderão ter associada informação extra a que o utilizador pode aceder clicando no botão correspondente.

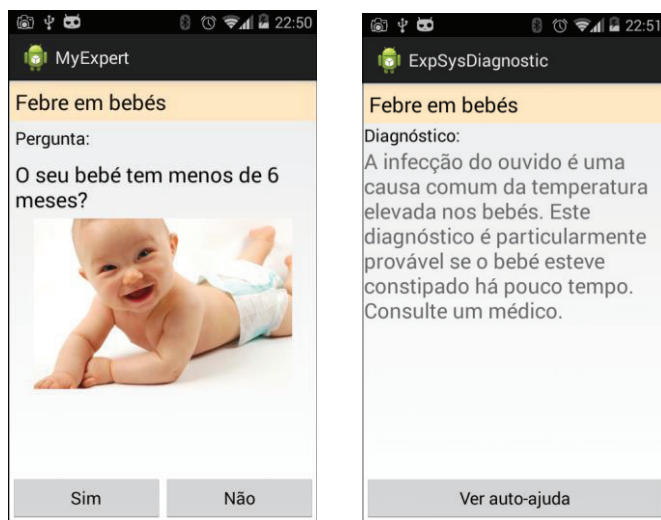


Figura 5. Ecrãs exemplificativos da interação com a máquina de inferência

O módulo Update é responsável pelo processo de atualizações das bases de conhecimento. Este módulo recebe

notificações por parte da aplicação de backend, com a informação das bases de conhecimento que têm actualizações disponíveis e, caso a aplicação MyExpert possua essa base de conhecimento armazenada, notifica o utilizador que pode proceder à actualização. É também através deste módulo que podem ser seleccionadas e descarregadas novas bases de conhecimento entretanto criadas e armazenadas na plataforma do sistema.

B. Aplicação MyExpert Manager

A figura 6 apresenta os 3 módulos que forma a aplicação MyExpert Manager.

O módulo User Interface é responsável pela interação entre o utilizador e o módulo Knowledge Acquisition que é responsável pela criação e actualização das bases de conhecimento. O módulo Update é o módulo responsável pelo armazenamento das bases de conhecimento na Google Cloud SQL. Tal como no caso a aplicação MyExpert, também aqui se usam os recursos Cloud Endpoints para a comunicação com a respetiva aplicação no backend do sistema.

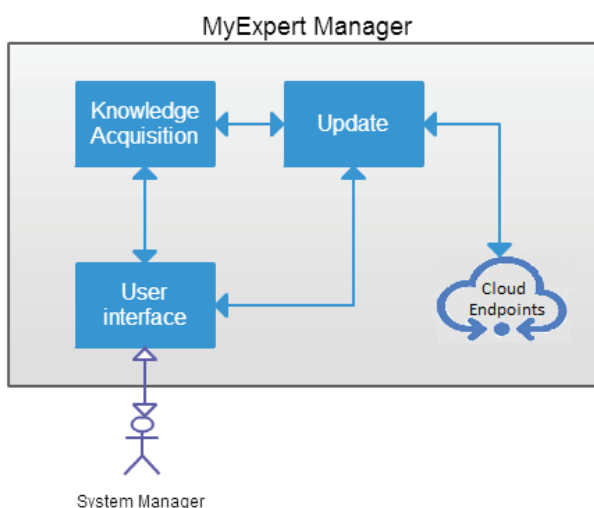


Figura 6. Módulos da aplicação MyExpert Manager

A aplicação foi também desenvolvida para a plataforma Android, mas foi otimizada para ecrãs com dimensão de 10”.

A figura 7 mostra um ecrã da aplicação que ilustra o processo de criação das regras de uma base de conhecimento. Do lado esquerdo podem-se ver as regras já criadas e do lado direito os dados inseridos para criar a próxima regra.

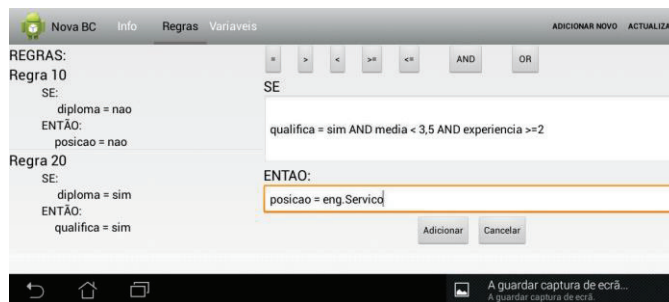


Figura 7. Ecrã exemplificativo da criação das regras de uma base de conhecimento

V. TESTES E RESULTADOS

Os testes realizados até ao momento testaram a comunicação e interoperabilidade entre os diferentes componentes da arquitetura implementada, tendo-se obtido resultados muito satisfatórios. A comunicação entre as aplicações cliente e o backend do sistema demonstrou ser rápida e estável, não tendo sido detetado qualquer problema. No que diz respeito ao serviço de notificações sobre as actualizações de bases de conhecimento disponíveis, considera-se a performance do sistema excelente, tendo em conta que nos testes realizados, a notificação é no geral instantânea. Os testes foram realizados através de uma rede académica Wi-Fi Eduroam com acesso à Internet.

Foi também testado o funcionamento da máquina de inferência. Para isso foi criada uma base de conhecimento para reconhecer situações de emergência médica em bebés. A base de conhecimento baseada em regras foi criada a partir de 5 quadros de diagnóstico, apresentados no livro “Guia Médico da Família” [13]. A transformação dos quadros de diagnóstico em regras foi realizada com base no algoritmo descrito em [3].

O teste realizado consistiu em simular 10 situações de emergência e comparar os diagnósticos obtidos utilizando os quadros de diagnóstico originais e utilizando o SP desenvolvido. A comparação dos diagnósticos foi realizada tendo em conta dois fatores: se o diagnóstico obtido é o correto, e a quantidade de questões que tiveram que ser respondidas para obter o diagnóstico. A tabela 1 apresenta os resultados obtidos.

A máquina de inferência encontrou sempre o diagnóstico esperado, apesar de ter que colocar mais questões (em 8 das 10 situações), do que no caso do uso direto dos quadros de diagnóstico. Esta diferença deve-se a que os quadros de diagnóstico têm uma estrutura em árvore binária que os torna mais eficientes na escolha das questões a colocar. Por sua vez, o algoritmo de conversão dos quadros em regras, simplesmente cria uma regra para cada nó folha e a máquina de inferência escolhe as regras na ordem pela qual se encontram na base de conhecimento, levando a que possam ser colocadas questões de outras conclusões que são testadas em primeiro lugar. Um algoritmo de conversão mais eficiente com base em conclusões intermédias poderá melhorar a performance da máquina de inferência.

TABELA 1. RESULTADOS OBTIDOS

	Especialista (Q. de Diagnóstico)		MyExpert	
	Conclusão	Questões	Conclusão	Questões
1	✓	2	✓	2
2	✓	6	✓	7
3	✓	4	✓	4
4	✓	7	✓	8
5	✓	7	✓	8
6	✓	3	✓	4
7	✓	5	✓	6
8	✓	7	✓	8
9	✓	5	✓	12
10	✓	4	✓	12

VI. CONCLUSÃO

O sistema implementado permite colocar ao alcance de um utilizador comum, um SP para a plataforma Android, fácil de usar e com a capacidade de selecionar diferentes bases de conhecimento das áreas de interesse de cada utilizador. O sistema foi implementado com base em computação na nuvem para facilitar o desenvolvimento do sistema e a criação e distribuição de diferentes bases de conhecimento.

A arquitetura do sistema baseada na plataforma Google App Engine mostrou-se adequada tanto ao nível do processo de desenvolvimento como ao nível de utilização e administração do sistema. A plataforma é rica em recursos, fácil de utilizar e tem uma performance muito satisfatória.

Como trabalho futuro aponta-se a possibilidade de melhorar a máquina de inferência e adicionar a possibilidade de criar bases de conhecimento baseado em árvores de decisão, pois muitas bases de conhecimento que poderão ser úteis para o utilizador comum encontram-se estruturadas nesta forma, como por exemplo, quadros de diagnóstico médico, manuais de reparação de avarias, quadros de classificação de espécimes, entre outros.

AGRADECIMENTOS

Publicação apoiada pelo projeto PEst-OE/EGE/UI4056/2014 UDI/IPG, financiado pela Fundação para a Ciência e Tecnologia.

REFERÊNCIAS

[1] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3th edition. Prentice Hall, 2009.

- [2] F. Hayes-Roth and D. B. Lenat, *Building Expert Systems*. Addison-Wesley, 1983.
- [3] R. I. Levine, D. E. Drang, and B. Edelson, *Inteligência Artificial e Sistemas Especialistas*. São Paulo: McGraw-Hill, 1988.
- [4] C. Quesada and M. Jenkins, “A Prototype Mobile Expert System for Nutritional Diagnosis,” no. Academy 2012, pp. 118–123, 2013.
- [5] “Rule Engine for the Java Platform.” [Online]. Available: <http://herzberg.ca.sandia.gov/>.
- [6] “xpertrule.” [Online]. Available: <http://www.xpertrule.com/>.
- [7] “expertise2go.” [Online]. Available: <http://expertise2go.com/>.
- [8] S. J. Vidaletti and S. R. Silveira, “SESMED (Smart Emergency Medical System) - Sistema Especialista para Apoio a Paramédicos.”
- [9] “Google App Engine.” [Online]. Available: <https://developers.google.com/appengine/docs/whatiscgogleappengine>.
- [10] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” *Communications of the ACM*, New York, pp. 50–58, 2010.
- [11] “Overview of Google Cloud Endpoints,” 2013. [Online]. Available: <https://developers.google.com/appengine/docs/java/endpoints/>.
- [12] “Google Cloud SQL: Relational Databases in Google’s Cloud.” [Online]. Available: <https://developers.google.com/cloud-sql/>.
- [13] Associação Paulista de Medicina, *Guia Médico Da Família*. São Paulo: Editora Nova Cultura, 1994.