



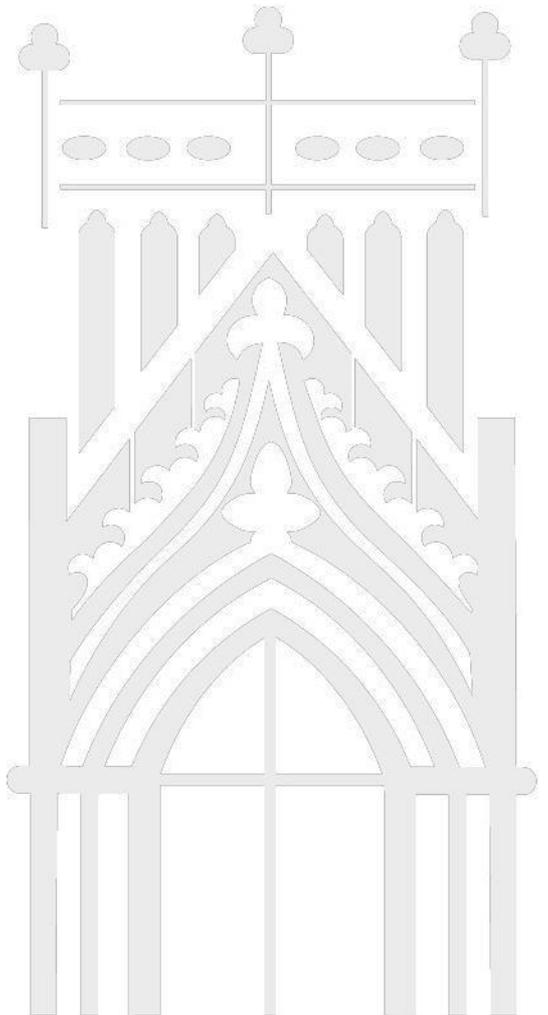
IPG Politécnico
|da|Guarda
Polytechnic
of Guarda

Mestrado em Computação Móvel

Sistema Pericial para Dispositivos Móveis com Base em
Computação na Nuvem

Mário Miguel Rosa Baltazar

julho | 2014



Escola Superior
de Tecnologia e Gestão



Escola Superior de Tecnologia e Gestão

Instituto Politécnico da Guarda

Sistema Pericial para Dispositivos Móveis com Base em Computação na Nuvem

Relatório de Projeto Aplicado submetido como requisito parcial para
obtenção do grau de Mestre em Computação Móvel

Orientador: Prof. Doutor Carlos Carreto

Mário Miguel Rosa Baltazar

Julho | 2014

Agradecimentos

O presente trabalho só teria sido possível graças à colaboração e apoio de algumas pessoas, às quais gostaria de deixar o meu reconhecimento:

- Ao meu orientador Prof. Doutor Carlos Carreto pelo apoio prestado durante todo o tempo de realização do trabalho.

- À empresa Wuso pelo tempo e apoio dispensados.

- À Rita, aos meus pais, pela muita paciência durante este tempo.

- À Unidade de Investigação para o Desenvolvimento do Interior que apoiou o trabalho no âmbito do projeto PEst-OE/EGE/UI4056/2014 UDI/IPG, financiado pela Fundação para a Ciência e Tecnologia.

Resumo

Este trabalho descreve a implementação de um Sistema Pericial para dispositivos móveis, dirigido ao utilizador comum e com a capacidade de usar diferentes bases de conhecimento de diferentes áreas, seleccionáveis pelo utilizador.

O sistema é composto por duas aplicações, uma para o administrador, e outra para o utilizador comum. A primeira tem como objetivo permitir ao administrador “alimentar” o sistemas com novas Bases de conhecimento assim como de fazer alterações às já existentes e ainda de notificar os utilizadores das novas alterações que foram feitas. Já a segunda aplicação permite ao utilizador descarregar para o seu smartphone as bases de conhecimento disponíveis, consultá-las, e receber as notificações das alterações disponíveis.

O sistema usa uma arquitetura com base em computação na nuvem para facilitar a criação e distribuição de diferentes bases de conhecimento. A plataforma de computação na nuvem utilizada foi a plataforma da Google – Google App Engine, que é uma plataforma do tipo Plataforma como serviço.

O sistema usa dois tipos de bases de conhecimento: bases de conhecimento baseado em regras e bases de conhecimento baseado em árvores de decisão binária. Estas estão organizadas por áreas e, por cada área, poderão ser organizadas em categorias e subcategorias. O motor de inferência implementado tem assim um comportamento diferente de acordo com o tipo de base de conhecimento escolhido pelo utilizador.

Nos testes efetuados conclui-se que a arquitetura da aplicação foi bem escolhida pois a comunicação entre aplicações de backend e aplicação cliente é bastante eficaz, tendo-se medido tempos de atualização muito reduzidos. No que diz respeito à usabilidade a aplicação obteve uma pontuação de 81, numa escala de 0 a 100, utilizando a escala SUS (System Usability Scale) proposta por John Brooke.

Palavras-chave: Sistema Pericial, Android; Google App Engine, Computação na Nuvem.

Abstract

This paper describes the implementation of an Expert System for mobile devices, and directed to the common user with the ability to use different knowledge bases of different areas, selectable by the user.

The system consists of two applications, one for the administrator and another for the average user. The first objective is to allow the "feeder" systems administrator with new knowledge bases as well as making changes to existing and yet to notify users of new changes that were made. The second application allows the user to download to your smartphone knowledge bases available, consult them, and receive notifications of updates available.

The system uses an architecture based on cloud computing to facilitate the creation and distribution of different knowledge bases. The cloud computing platform used was the platform of Google - Google App Engine, which is a platform-type platform as a service.

The system uses two types of knowledge bases: knowledge bases and rule-based knowledge bases based on binary decision trees. These are organized into areas, and each area can be organized into categories and subcategories. The inference engine is implemented so different according to the type of knowledge base selected by the user behavior.

In the performed tests it is concluded that the application architecture is well chosen because communication between the client application and back-end applications is quite effective, and it was measured very low refresh times. With respect to the usability application obtained a score of 81 on a scale of 0 to 100 scale using SUS (System Usability Scale) proposed by John Brooke.

Keywords: Expert System, Android; mobile devices, Google App Engine, Cloud Computing.

Índice

Índice de Figuras	vi
Índice de Tabelas	viii
CAPÍTULO 1 - INTRODUÇÃO.....	9
1.1 Objetivos.....	10
1.2 Estrutura do Relatório.....	10
CAPÍTULO 2 - TRABALHO RELACIONADO	12
2.1 Sistemas Periciais	12
2.2 Componentes de um Sistema Pericial.....	13
2.3 Representação do conhecimento.....	15
2.4 Plataforma Android.....	16
2.5 Exemplos de Sistemas Periciais.....	18
2.6 Plataformas de Computação na Nuvem.....	20
CAPÍTULO 3 - TRABALHO DESENVOLVIDO	23
3.1 Análise dos Requisitos.....	23
3.2 Ferramentas e Serviços	24
3.3 Serviços da plataforma Google App Engine.....	26
3.3.1 Google Cloud SQL	28
3.3.2 Google Cloud Endpoints	28
3.3.3 Google Cloud Messaging	30
3.4 Casos de uso.....	30
3.5 Representação das Bases de Conhecimento	32
3.6 Armazenamento na Base de Dados	35
3.6.1 Base de dados alojada no Google Cloud SQL.....	35
3.6.2 Base de dados mobile	41
3.7 Algoritmos do Motor de Inferência	42
3.8 Arquitetura do Sistema	44
3.9 Aplicação MyExpert Manager.....	45
3.9.1 Módulo User Interface.....	46
3.9.2 Módulo Knowledge Acquisition	46
3.9.3 Módulo Update	47
3.9.4 Módulo Update Database	47
3.9.5 Módulo Notification	47
3.10 Aplicação MyExpert	48
3.10.1 Módulo User interface	49
3.10.2 Módulo Inference Engine	49
3.10.3 Módulo Explanation Subsystem.....	49
3.10.4 Módulo Update	49
3.10.5 Módulo KnowledgeBase	50
3.10.6 Módulo Registration	50
3.10.7 Módulo GetUpdate	51
3.11 Mecanismo de atualização	51
3.12 Publicação da aplicação na loja de aplicações Google Play	53
Capítulo 4 - Testes e resultados.....	54

Capítulo 5 - Conclusões.....	61
Referências bibliográficas	63
Anexos.....	65
Anexo A - Artigo publicado na conferência CISTI2014.....	65
Anexo B - Formulário para aplicação do questionário SUS	73
Anexo C - Funcionamento da aplicação MyExpert.....	75
Anexo D - Funcionamento da aplicação MyExpert Manager	80

ÍNDICE DE FIGURAS

Figura 1 - Componentes de um Sistema Pericial.....	13
Figura 2 - Exemplo de uma Base de Conhecimento baseada em árvores de decisão [8].....	16
Figura 3 - Arquitetura da plataforma Android [5].....	17
Figura 4 - Número de downloads mensais de aplicações.....	18
Figura 5 - Número de downloads mensais de aplicações.....	18
Figura 6 - Adicionar Repositório do plugin da Google para o Eclipse	25
Figura 7 - Dashboard de um projeto Google App Engine.....	27
Figura 8 - Funcionamento do Google Cloud Endpoints.....	28
Figura 9 - Criação da classe Endpoint	29
Figura 10 -Atores e respetivos casos de uso do sistema desenvolvido	31
Figura 11 - Exemplo de Base de conhecimento em Árvore de Decisão	34
Figura 12 – Diagrama ER da base de dados presente no serviço Google Cloud SQL... ..	36
Figura 13 - ER da Base de dados Mobile	41
Figura 14 - Arquitetura Geral do sistema	45
Figura 15 - Arquitetura da Aplicação MyExpert Manager.....	46
Figura 16 - Arquitetura da aplicação MyExpert.....	48
Figura 17 - Funcionamento do Módulo Registration	50
Figura 18 - Funcionamento do módulo de atualizações.....	52
Figura 19 – Página da Aplicação no Google Play	53
Figura 20 - Testes de velocidade de inserção base de conhecimento.....	56
Figura 21 - Média SUS dos resultados para cada questão.....	59
Figura 22 - Análise das características de usabilidade da norma 9126	60
Figura 23. Arquitetura do Sistema.....	67
Figura 24. Atores do Sistema e respetivos casos de uso	68
Figura 25. Módulos da aplicação MyExpert	68
Figura 26. Diagrama Entidade-Relacionamento das bases de conhecimento	69
Figura 27. Ecrãs exemplificativos da interação com a máquina de inferência.....	70
Figura 28. Módulos da aplicação MyExpert Manager	70
Figura 29. Ecrã exemplificativo da criação das regras de uma base de conhecimento..	70
Figura 30 - Login na aplicação MyExper Manager.....	75
Figura 31 - Home da aplicação MyExpert Manager	75
Figura 32 - Escolha do tipo de Base de Conhecimento a adicionar	76
Figura 33 - Adicionar informação da Base de conhecimento	77
Figura 34 - Criação de regras de uma Base de Conhecimentos	77
Figura 35- Adicionar informação da nova Base de Conhecimento.....	77
Figura 36 -Descrição das variáveis das bases de conhecimento	78
Figura 37 - Criação dos nós que compõem uma base de conhecimentos.....	79
Figura 38 - Ecrã inicial da aplicação	80
Figura 39 – a) Lista de áreas de Bases de conhecimento b) Lista de Caterogias de Bases de Conhecimento c) Download de uma Categoria	81
Figura 40 – a) Subcategorias de Bases de Conhecimento b) Bases de Conhecimento ..	81

Figura 41 – a) Exemplo de uma questão colocada ao utilizador b)Exemplo de uma
conclusão encontrada..... 82

ÍNDICE DE TABELAS

Tabela 1 - Comparação entre plataformas de computação na nuvem [6]	22
Tabela 2 - Descrição da tabela Area	36
Tabela 3 - Descrição da tabela Category	36
Tabela 4 - Descrição da tabela SubCategory	37
Tabela 5 - Descrição da tabela KnowledgeBases	37
Tabela 6 - Descrição da tabela Rules	38
Tabela 7 - Descrição da tabela ClausesRules	38
Tabela 8 - Descrição da tabela Clauses	39
Tabela 9 - Descrição da tabela Variables	39
Tabela 10 - Descrição da tabela Node	39
Tabela 11 - Descrição da tabela Tip	40
Tabela 12 - Descrição da tabela Devices	40
Tabela 13 - Descrição da tabela UpdateMessage	41
Tabela 14 - Resultados do teste do motor de inferência	55
Tabela 15 - Tempos de atualização da base de conhecimento com 10 nós	57
Tabela 16 - Tempos de atualização da base de conhecimento com 20 nós	57
Tabela 17 - Tempo desde que a ligação á internet é estabelecida até que o utilizador recebe a notificação	58

CAPÍTULO 1

INTRODUÇÃO

Este relatório descreve o trabalho realizado pelo autor no âmbito do projeto aplicado do Mestrado em Computação Móvel da Escola Superior de Tecnologia e Gestão do Instituto Politécnico da Guarda.

O projeto consistiu no desenvolvimento de um sistema pericial para dispositivos móveis (smartphones e tablets) com o sistema operativo Android, desenvolvido com base em computação na nuvem.

Os SP pretendem resolver problemas em um determinado domínio onde o conhecimento é muito específico e fortemente limitado aos peritos da área. Estes sistemas pretendem simular o raciocínio destes peritos na resolução de determinados problemas.

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.

O SP desenvolvido tem como principal objetivo ser um sistema dirigido ao utilizador comum e com a capacidade de usar diferentes bases de conhecimento, seleccionáveis pelo utilizador. O sistema foi desenvolvido utilizando a plataforma de Computação na Nuvem da Google – Google App Engine. Este facilitou o desenvolvimento de todo o sistema, especialmente no que se refere à criação e distribuição de diferentes bases de conhecimento, pois algumas aplicações do sistema ficam alojadas nos servidores da Google. Com a utilização desta plataforma é possível utilizar os vários serviços disponibilizados por esta, como é o caso

da gestão de base de dados, das aplicações de backend, da troca de informação entre a aplicação móvel e as aplicações de backend e do envio de notificações.

O sistema desenvolvido foi designado de MyExpert. A razão da escolha deste nome prende-se com o intuito de transmitir a ideia de que a aplicação dá a possibilidade de cada utilizador possuir “na palma da mão” e ter sempre disponível para consulta, um perito das suas áreas de interesse.

O sistema desenvolvido compreende 2 aplicações distintas. A primeira designada por MyExpert Manager, desenvolvida apenas para o administrador do sistema, tem como objetivo fazer a gestão do sistema, o que implica, inserção e atualização das bases de conhecimento que compõem o sistema pericial. A segunda designada apenas por MyExpert, é a aplicação desenvolvida para o utilizador final, que vai usar as bases de conhecimento disponíveis no sistema.

1.1 Objetivos

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.

1.2 Estrutura do Relatório

O presente documento encontra-se estruturado em 5 capítulos. No primeiro capítulo faz-se uma apresentação do trabalho desenvolvido, assim como os objetivos que foram traçados para o projeto, e motivação para o mesmo.

O segundo capítulo inicia-se com uma apresentação sobre o que são os sistemas periciais e quais são os componentes que podem ser considerados de um sistema pericial assim como as formas de representação do conhecimento e do funcionamento do motor de inferência de um sistema pericial. De seguida é apresentada a plataforma de dispositivos móveis utilizada, a plataforma Android, e a sua arquitetura.

Para terminar este capítulo são apresentados alguns SP conhecidos e as suas características comparando com o sistema desenvolvido. No final do capítulo é feita uma apresentação das plataformas de computação na nuvem.

No terceiro capítulo é apresentado o sistema desenvolvido, é feita a análise dos requisitos do sistema, descrito o ambiente de desenvolvimento, serviços utilizados e ferramentas, assim como a arquitetura de cada aplicação desenvolvida.

O quarto capítulo apresenta os testes a que foi sujeito o sistema desenvolvido para analisar a sua consistência, respostas por parte das solicitações dos utilizadores assim como da usabilidade deste.

No quinto e último capítulo faz a conclusão do trabalho realizado assim como apresenta trabalho futuro de melhoramento do sistema.

CAPÍTULO 2

TRABALHO RELACIONADO

No presente capítulo será descrito o que são sistemas periciais. São descritos os componentes típicos de um SP, com destaque para motor de inferência. São referidas as principais formas de representação do conhecimento e dados exemplos de cada uma delas.

A seguir o Capítulo apresentada uma visão geral da plataforma móvel Android e da sua arquitetura.

O Capítulo continua com a descrição de alguns exemplos de SP semelhantes como sistema desenvolvido.

Termina-se com uma descrição das principais plataformas de computação na nuvem existentes no mercado.

2.1 Sistemas Periciais

Os Sistemas Periciais (SP) são programas que derivam de um ramo da ciência da computação chamado Inteligência Artificial (IA). A IA tenta dotar as máquinas de inteligência, tentando simular a inteligência humana através de programas de computador.

Um 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 a sua utilização é mais expressiva.

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 [5]. A secção seguinte descreve com detalhe os componentes de um SP.

2.2 Componentes de um Sistema Pericial

Um SP típico pode ser dividido em vários componentes, tendo cada um destes um papel fundamental no sistema. No desenvolvimento de um SP deve-se assegurar que cada componente deverá ser o mais independente do outro para garantir que uma alteração num módulo não implique alteração em todo o sistema. Os componentes de um SP são apresentados no diagrama da Figura 1.

Perito – Pessoa que detém o conhecimento de uma determinada área. É este que fornece o conhecimento para criar a base de conhecimento do SP.

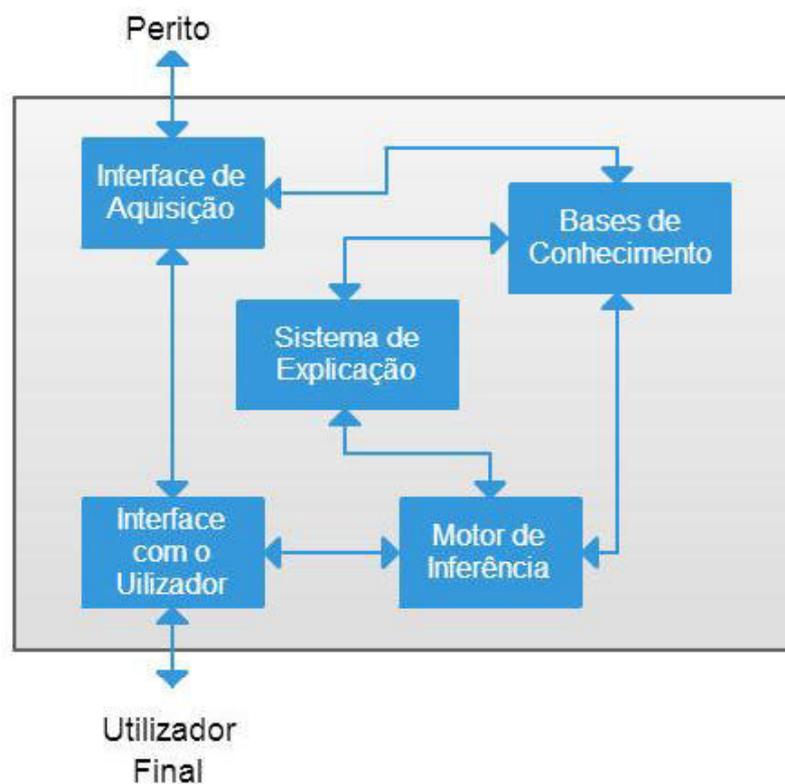


Figura 1 - Componentes de um Sistema Pericial

Interface de Aquisição – Este poderá ser considerado componente do SP mais completo. A aquisição do conhecimento poderá ser feita através de uma interface de software onde o conhecimento do perito é convertido numa base de conhecimento através de entrevistas feitas por um engenheiro do conhecimento ao perito, ou por ferramentas informáticas que fazem a aquisição dos dados de uma forma automática.

Bases de Conhecimento - Parte do sistema onde devem ser armazenados os novos conhecimentos. Onde o conhecimento antigo pode ser alterado, e conseqüentemente a base de conhecimentos atualizada. É aqui onde factos e regras ficam armazenadas. O conhecimento contido na base de conhecimentos deve ser de fácil manutenção para futuras modificações. Este deverá ser contruído independente da máquina de inferência para que esta seja capaz de o interpretar independentemente da área a que pertencer.

Interface com o utilizador - Parte que oferece comunicação e interação com o utilizador final. Esta irá exhibir perguntas ao utilizador e permitir que este introduza respostas. É através deste módulo que o sistema irá comunicar ao utilizador final as conclusões. Este é considerada a “ponte” entre a máquina de inferência e o utilizador. Esta deve ser simples e intuitiva para que um utilizador sem grande conhecimento na área em questão da base de conhecimento consiga fazer uso do SP, e sem a ajuda de um perito conseguir obter uma resposta para a sua dúvida.

Motor de inferência – Parte do sistema que efetua consultas na base de conhecimento à procura de respostas, decidindo a ordem em que as inferências serão consideradas. Esta compara as informações fornecidas pelo utilizador, com os factos e regras da base de conhecimentos, em busca de resposta para o problema em causa. Existem dois métodos de fazer a análise das regras, o encadeamento para a frente, e encadeamento para trás. Na secção 2.3 são descritos com mais detalhe cada processo de encadeamento.

Sistema de Explicação – Parte responsável por explicar ao utilizador final os factos que levaram o sistema à conclusão alcançada.

Utilizador Final – Pessoa que vai responder às perguntas feitas pelo sistema e ao qual vai ser apresentada a conclusão.

2.3 Representação do conhecimento

A representação do conhecimento pode ser feita através de diferentes métodos como por exemplo: regras de sistemas de produção, redes semânticas ou árvores de decisão [6]. O método usado 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 do módulo de Aquisição de Conhecimento [7].

Para o sistema desenvolvido teve-se em consideração duas formas de representação: Árvore de Decisão e Regras de Produção.

Numa base de conhecimento baseada em regras, 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. De seguida é apresentada um exemplo de uma Base de Conhecimentos baseada em regras.

10	SE	TAXA DE JUROS = BAIXA
	ENTÃO	BOLSA DE VALORES = ALTA
20	SE	TAXA DE JUROS = ALTA
	ENTÃO	BOLSA DE VAORES = BAIXA
30	SE	COTAÇÃO DO DOLÁR
	ENTÃO	TAXA DE JUROS = ALTA
40	SE	COTAÇÃO DO DOLAR = ALTA
	ENTÃO	TAXA DE JUROS = BAIXA

No caso do método de Árvore de Decisão esta tem este nome pois ramifica-se exatamente como uma árvore e no final de cada ramo existe uma conclusão. Com esta forma de representação o problema a resolver fica mais claro de entender, pois é de fácil leitura.

A Figura 2 representa um exemplo de uma árvore de decisão para analisar se um candidato se qualifica para uma posição dentro de uma determinada empresa e qual o cargo a oferecer.

Uma Árvore de Decisão pode ser representada por um grafo, onde os nós representam questões ou conclusões e os arcos dirigidos representam possíveis respostas às questões. Os arcos determinam a direção do diagrama, sendo a direção também influenciada pela resposta que o utilizador dá a cada nó. No caso do exemplo da Figura 2, os círculos que contêm as perguntas são os nós de decisão, enquanto que os retângulos representam nós de conclusões.

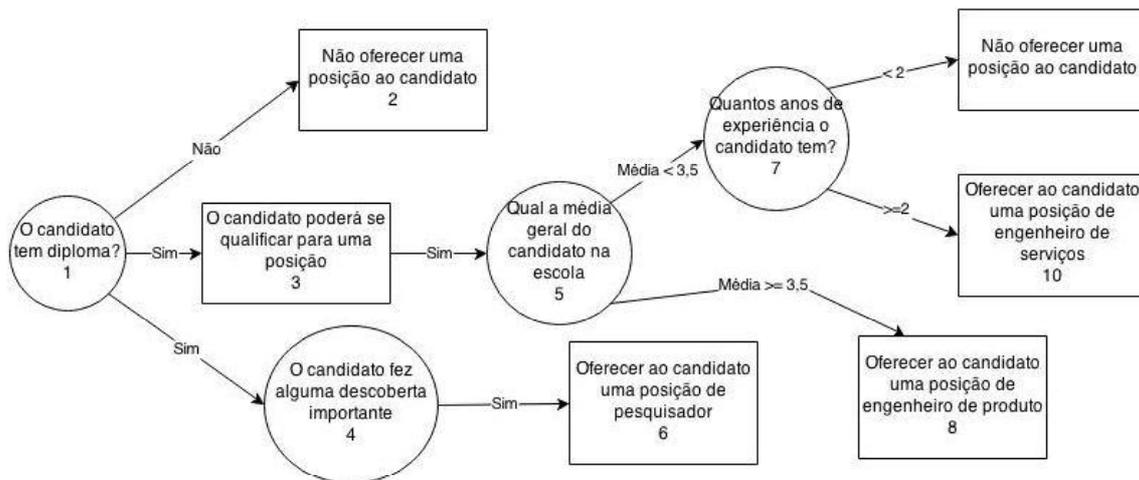


Figura 2 - Exemplo de uma Base de Conhecimento baseada em árvores de decisão [8]

Existem formas de converter uma árvore de decisão em regras do tipo das representadas no exemplo da Figura 1, mas estas não foram consideradas para o projeto pois com os testes realizados que são apresentados no Capítulo 4 o número de perguntas a ser feito ao utilizador no caso das árvores de decisão é menor para se alcançar a respostas pretendida. Assim optou-se por utilizar as 2 formas de representação de bases de conhecimento. O sistema implementado permite a representação dos dados em árvore de decisão e regras. Deste modo o motor de inferência vai atuar de acordo com a forma como a informação está representada.

2.4 Plataforma Android

A plataforma Android caracteriza-se por ser uma plataforma aberta para dispositivos móveis, baseada no núcleo do Linux,. Esta teve o seu lançamento em 12 de Novembro de 2007 [9], por uma associação de empresas liderada pela Google, com o nome Open Handset Alliance.

Antes de iniciar o desenvolvimento sob uma plataforma é importante ter o completo conhecimento da arquitetura do sistema que se vai utilizar. A Figura 3 apresenta a arquitetura da plataforma Android.

A camada mais baixa da plataforma, que faz a ligação com o hardware, como já foi referido, é baseada no núcleo do Linux, fornecendo uma camada de abstração entre o software e o hardware. Esta contém os drivers essenciais. Android utiliza assim esta camada para fazer as tarefas essenciais de um sistema operativo como é o caso da gestão de memória, gestão de processos, rede e gestão das definições de segurança.

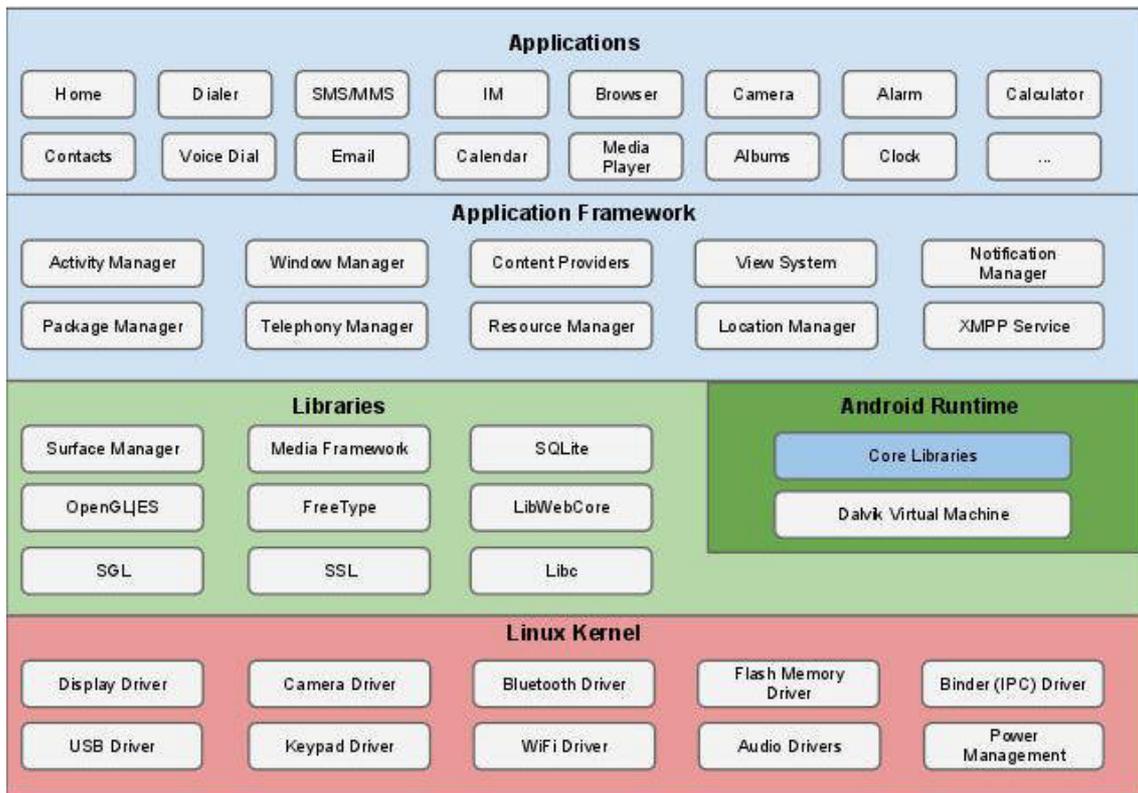


Figura 3 - Arquitetura da plataforma Android [5]

A camada Bibliotecas (Libraries) nativas da plataforma. Estas fornecem funcionalidades nativas da plataforma e são específicas para cada tipo de hardware que o dispositivo possua. Algumas bibliotecas importantes são por exemplo a Surface Manager, que é a responsável por fazer a gestão das janelas, SQLite, que é o motor de base de dados usado para fazer a gestão do armazenamento dos dados, Webkit, que é o motor do browser para mostrar o conteúdo HTML, e OpenGL que é responsável por fazer o renderização dos gráficos 2D e 3D no ecrã.

A camada seguinte é a camada Android Runtime, que permite que cada thread “corra” a sua própria máquina virtual.

A camada Application Framework contém os “blocos” com o qual as aplicações nativas interagem diretamente. Estes gerem as funcionalidades básicas dos dispositivos como é o caso da gestão das chamadas, gestão de recursos, localização e gestão de notificações.

A camada Applications constitui a camada do topo da arquitetura. Aqui inserem-se as aplicações que interagem diretamente com o utilizador do dispositivo.

Quanto a números a plataforma Android tem vindo a aumentar não só o número de dispositivos com este sistema como também o número de downloads de aplicações das respetivas lojas de aplicações. Como se pode ver na Figura 4 o número de downloads de aplicações .

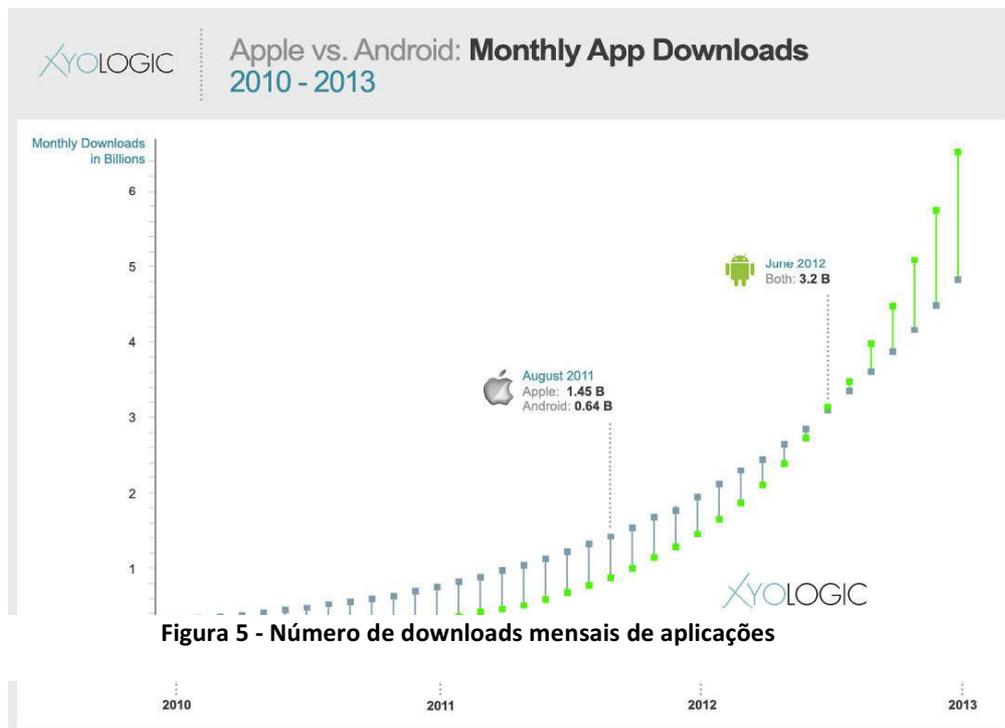


Figura 4 - Número de downloads mensais de aplicações

Quanto ao número de dispositivos, segundo o site da plataforma cada dia são ativados mais de 1 milhão de dispositivos com esta plataforma em todo o mundo [10].

A juntar ao facto de a plataforma ser aberta, ao aumento de instalações, nos últimos tempos, a justificação para a escolha desta plataforma também se prende com o facto de o investimento inicial para começar a desenvolver e publicar na loja de aplicações ser substancialmente menor para o caso do Android, quando comparado com outras plataformas.

2.5 Exemplos de Sistemas Periciais

Existem atualmente vários SP desenvolvidos para smartphones e para a Web. Descrevem-se a seguir alguns exemplos de sistemas semelhantes ao sistema desenvolvido.

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. Esta base de conhecimento é constituída por 60 regras e 10 funções que fazem o diagnóstico nutricional da pessoa em causa com base na informação fornecida pelo utilizador.

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 localizado na aplicação cliente, e as bases de conhecimento podem ser descarregadas do servidor da empresa, ficando localizadas 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 selecionar 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 sob a plataforma J2ME, em uma arquitetura cliente-servidor que é usada para comunicar com uma aplicação servidor onde está a máquina de inferência. O utilizador usa a aplicação cliente para enviar os dados dos pacientes para o sistema especialista, o qual retorna o diagnóstico. Para representar o conhecimento o sistema utiliza Frames e estas são guardadas numa base de dados relacional Oracle. O algoritmo de inferência utilizado foi definido pelos autores da aplicação em

parceria com os especialistas da área (médicos. Tal como na aplicação Nutrition UCR neste sistemas a aplicação cliente apenas tem como função fazer a comunicação com o servidor, onde está localizada a máquina de inferência, que após receber as informações dos pacientes através da aplicação móvel, retorna os possíveis diagnósticos.

O DxPlain é um sistema um pouco diferentes dos outros analisados até agora, este é um sistema especialista desenvolvido para sugerir uma lista de doenças que estão associadas a um conjunto de estados clínicos. Esta Ferramenta foi desenvolvida no Massachusetts General Hospital com o objetivo de dar apoio à decisão. A base de conhecimentos deste sistema está implementado baseado em regras. Este sistema apenas está disponível online, contando com um uso de quase 20 anos.

Os SP descritos apresentam uma arquitetura baseada em serviços, em que uma aplicação cliente comunica com uma aplicação servidor, em alguns casos para enviar dados e receber a resposta do SP e em outros para aceder às bases de conhecimento. Esta arquitetura tem o grande inconveniente de necessitar de uma ligação de dados permanente entre cliente e servidor.

O sistema proposto também usa uma arquitetura baseada em serviços, mas com a diferença de não ser necessária uma ligação de dados permanente entre cliente e 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 maior parte dos sistemas analisados usam uma única base de conhecimentos de uma área específica.

2.6 Plataformas de Computação na Nuvem

A Computação na Nuvem tem como base a partilha de recursos de computação. Em vez de servidores locais pessoais para alojar as aplicações existem servidores partilhados que partilham recursos e processamento.

Os serviços de Computação na Nuvem permitem criar, testar e implementar aplicações em infraestruturas altamente escaláveis e confiáveis.

Nestes tipos de serviços é possível ter disponível a baixo custo máquinas virtuais de alto desempenho assim como balanceamento de carga no caso de ser necessário grande resposta por parte dos servidores.

Resumidamente poderemos enumerar as seguintes grandes vantagens:

- Escalabilidade;
- Confiabilidade;
- Disponibilidade;
- Apenas é necessário pagar pelo tempo que se utiliza o serviço;
- Não é necessário investimento inicial em hardware, nem compra de licenças ou pagar por atualizações necessários.
- Não é necessária preocupação de monitorização. Se o servidor deixar de funcionar, essa tarefa fica a cargo do fornecedor do serviço.
- Não é necessário armazenamento extra para as cópias de segurança.
- Aplicações têm níveis de segurança implementados.

Devido às inúmeras vantagens destes tipos de serviços, tem se verificado um grande aumento de adesão nos últimos tempos, sendo apontados como substitutos aos serviços tradicionais de servidores dedicados mesmo em pequenas organizações.

Podemos distinguir 3 tipos de serviços de Computação na Nuvem: Software como serviço (Software as a Service, SaaS), Plataforma como serviço (Platform as a Service, PaaS), Infraestrutura como serviço (Infrastructure as a Service, IaaS).

No primeiro (SaaS) trata-se de um modelo de distribuição de software em que este não é instalado localmente, mas sim disponibilizado através de um acesso pela internet.

No IaaS são disponibilizadas as infraestruturas computacionais como um serviço. Sempre que é necessário maiores capacidades de armazenamento ou de computação, o serviço aumenta as capacidades da máquina virtual em vez de se ter que adquirir novos equipamentos.

Um serviço do tipo PaaS 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. PaaS permite que o desenvolvimento seja totalmente focado no negócio, pois a plataforma já administra a arquitetura física de forma transparente.

Na Tabela 1 são apresentadas a comparação entre diferentes plataformas

Tabela 1 - Comparação entre plataformas de computação na nuvem [6]

Property	Different Platforms				
	Amazon Elastic Compute Cloud (EC2)	Microsoft Azure	Google App Engine	Sun Network.com (Sun Grid)	GRIDS Lab Aneka
Focus	Infrastructure	Platform	Platform	Infra-structure	Enterprise clouds
Service Type	Compute, Storage (Amazon S3)	Web and nonweb application	Web Application	Computing	Computing
User Access interface	Amazon EC2 command-line tools	Microsoft windows azure portal	Web-based administration	scripts, Sun Grid web portal	Work-bench, web-based portal
Value-added service providers	Yes	Yes	No	Yes	No
Virtualization	OS level running on a Xen hypervisor	OS level through fabric controller	Application container	Job management system (Sun Grid Engine)	Resource manager and scheduler
Web APIs	Yes	Yes	Yes	Yes	Yes
Dynamic negotiation of QoS	None	None	None	None	SLA-base Resources reservation
Programming frameworks	Amazon Machine Images (AMI)	Microsoft.NET	Python, Go, java, php	Solaris OS. Java, C, C++, FORTRAN	APIs supporting models in c#.Net

Pela análise da tabela apresentada podemos concluir que as características apresentadas pelas plataformas diferem em alguns detalhes que poderão ser determinantes no momento de optar por um deles. Para o presente projeto foi escolhida a plataforma Google App Engine pela fácil integração com a plataforma Android assim como pelos serviços oferecidos. Na secção 0 serão apresentados com mais detalhe os serviços da plataforma.

CAPÍTULO 3

TRABALHO DESENVOLVIDO

No presente capítulo será descrito todo o trabalho desenvolvido para implementar o sistema pretendido. Em primeiro lugar é feito um levantamento dos requisitos assim com as ferramentas utilizadas durante o processo de desenvolvimento e os serviços utilizados da plataforma de Computação na Nuvem Google App Engine.

A seguir é descrito como cada tipo de bases de conhecimento é guardada na base de dados e quais os algoritmos do motor de inferências utilizados para cada tipo de bases de conhecimento.

Por último são descritas as arquiteturas de cada aplicação e os seus modos de funcionamento do ponto de vista do utilizador final.

Os diagramas das aplicações apresentados encontram-se em Inglês para que pudessem ser utilizado em futuras publicações.

3.1 Análise dos Requisitos

O sistema desenvolvido pretende ser um SP que possa ser utilizado por qualquer utilizador sem conhecimento da área da base de conhecimento em questão. Este pretende ser de fácil utilização, e intuitivo para o utilizador final.

Este terá disponíveis várias bases de conhecimento agrupadas por áreas, podendo o utilizador fazer o download das bases de conhecimento do seu interesse. Todas as bases de conhecimento descarregadas podem ser utilizadas sem necessidade de qualquer tipo de ligação de dados, pois o motor de inferência está localizado no dispositivo móvel, e irá utilizar as bases de conhecimento descarregadas.

Para fazer o armazenamento das bases de conhecimento utilizou-se uma base de dados relacional, que está alojada num servidor e que, através de uma aplicação de backend, permite que as aplicações móveis acessem aos dados armazenados.

A plataforma móvel escolhida para o desenvolvimento das aplicações foi a plataforma Android, e a plataforma de Computação na Nuvem escolhida por razões de compatibilidade e de funcionalidades disponibilizadas foi a plataforma Google App Engine. A utilização desta no sistema foi para alojar a aplicação de backend de cada aplicação móvel desenvolvida, alojamento da base de dados, e envio de notificações de atualizações de bases de conhecimento.

Foram desenvolvidas duas aplicações, tendo sido ambas desenvolvidas para a plataforma Android. A aplicação MyExpert foi desenvolvida e otimizada para smartphone. Esta é a aplicação que será disponibilizada para os utilizadores finais, para consultar as bases de conhecimento.

A aplicação MyExpert Manager foi desenvolvida e otimizada para Tablet de 10". Esta tem a função de gerir as bases de conhecimento que ficam acessíveis aos utilizadores finais.

Para cada uma das aplicações existe uma aplicação de backend que permite fazer a “ponte” entre a aplicação móvel e a base de dados.

3.2 Ferramentas e Serviços

Para o desenvolvimento do sistema descrito foram utilizadas várias ferramentas de desenvolvimento, assim como vários serviços disponibilizados pela plataforma de Computação na Nuvem Google App Engine. Esta secção descreve as principais ferramentas e serviços utilizados no desenvolvimento do projeto.

Começamos por descrever a ferramenta base de desenvolvimento das aplicações móveis e das aplicações de backend respetivas. A ferramenta utilizada foi o ambiente de desenvolvimento Eclipse, com o plugin ADT (Android Developer Tools), e o Android SDK Tools. Para facilitar a instalação destes plugins foi aceite a sugestão da página de developers da Google [11] e feito o download do pacote de instalação que contem estas ferramentas todas já instaladas assim como uma emulador de dispositivos móveis para poder testar a aplicação sem que para isso seja necessária a utilização de um dispositivo físico.

A versão do Eclipse utilizada foi a 3.2.1, sendo esta a última versão no momento do início do desenvolvimento do projeto.

Após o download destes pacotes de ferramentas, o próximo passo é deixar o eclipse com a possibilidade de desenvolver aplicações para a plataforma de computação na nuvem da Google. Para permitir isto é necessário a instalação do plugin da Google para o Eclipse. A

escolha da versão do plugin depende da versão do Eclipse utilizada [12]. Para a instalação começa-se por adicionar um novo repositório ao centro de repositórios de download do Eclipse:

Help -> Install New Software... -> Add

E preenchemos os campos como indicado na Figura 6.

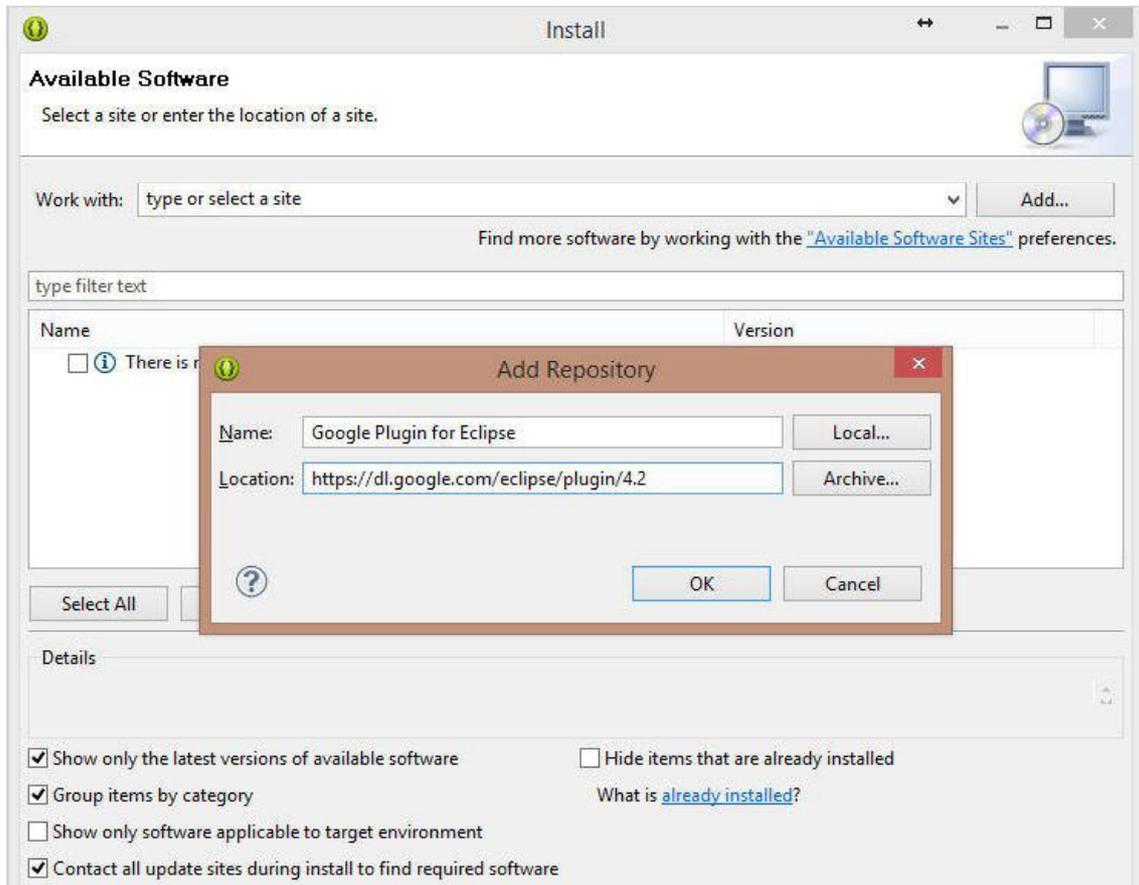


Figura 6 - Adicionar Repositório do plugin da Google para o Eclipse

Após o eclipse reconhecer o repositório apenas é necessário escolher todas as instalações disponíveis e proceder á instalação.

Para criar um projeto Android com uma aplicação de backend alojada na Google App Engine é necessário executar as seguintes ações e seguir os passos do assistente de criação de projeto:

File->New -> Other -> Android -> App Engine Connected Android Project.

A versão do SDK para o Google App Engine foi a versão 1.8.7, sendo esta a versão mais recente do SDK no início do projeto.

Para visualizar os logs gerados pelas aplicações de backend foi utilizado a aplicação web da Google, Google App Engine Console.

Já para fazer a gestão da base de dados alojada na Google cloud sql foi utilizada a ferramenta Squirrel [7].

3.3 Serviços da plataforma Google App Engine

O Google App Engine permite executar aplicações numa plataforma PaaS, ficando a gestão desta acessível através de uma página web, como pode ser visto na Figura 7. Este permite que qualquer programador possa executar as suas aplicações web nas infraestruturas da plataforma Google, ficando a beneficiar da experiencia dos técnicos desta empresa. As aplicações alojadas neste serviço são fáceis de criar, manter e escalonar à medida que o tráfego e armazenamento de dados necessita de crescer.

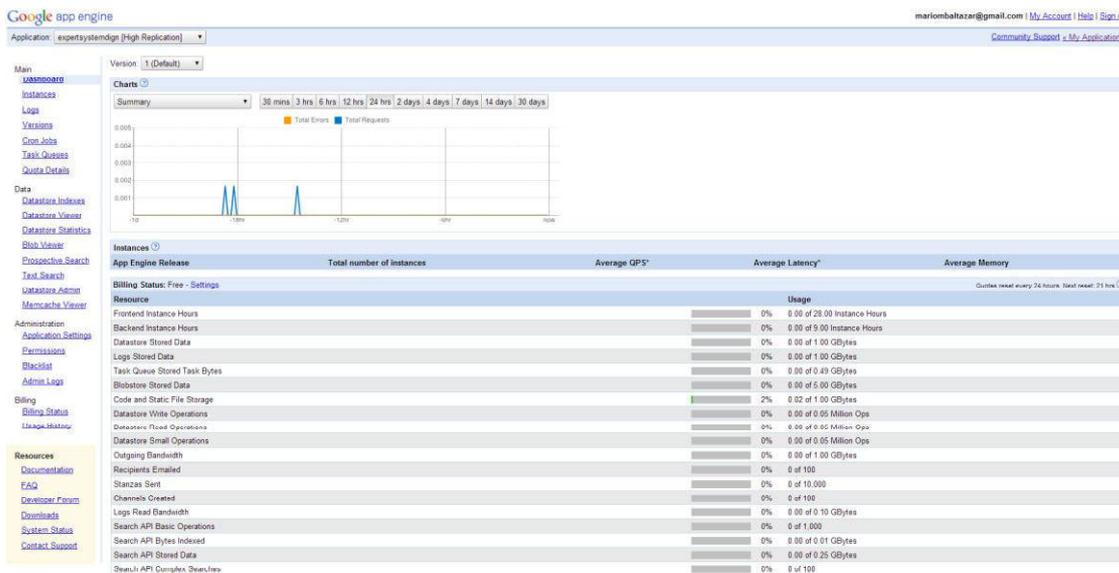


Figura 7 - Dashboard de um projeto Google App Engine

Além destas vantagens temos ainda disponíveis vários serviços que podemos integrar nas aplicações.

Como já foi referido na secção 3, foram desenvolvidas duas aplicações distintas, sendo que ambas comunicam com a mesma base de dados.

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 esta que é definida na plataforma de configuração da cloud da Goole, onde são especificadas as definições das aplicações backend. Para isso é gerado um Cliend ID OAuth 2.0 [10], gerado a partir do nome do package e do certificado de assinatura digital da aplicação cliente. Esse ID é usado 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.

Nas secções seguintes são analisados com mais algum detalhe alguns destes serviços.

3.3.1 Google Cloud SQL

O serviço Google Cloud Sql permite ter bases de dados relacionais MySQL na cloud da Google, podendo fazer a gestão destes totalmente através de uma página web. A gestão da replicação, criptografia e backups fica da responsabilidade da plataforma Google App Engine, assim permite libertar o programador destas preocupações e manter o foco apenas no desenvolvimento da aplicação.

Esta tem fortes mecanismos de seguranças implementados e tem integração total com todos os serviços disponibilizados pela Google App Engine.

Uma das grandes vantagens deste serviço é que podemos escolher várias modalidades de pagamento como por exemplo pagar apenas pelas horas em que a instância de base de dados é utilizada ou escolher um pacote dependendo da RAM e espaço necessário para a base de dados. Para o presente projeto a escolha foi de pagamento pelas horas de utilização.

3.3.2 Google Cloud Endpoints

Um elemento importante 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. Na Figura 8 é pode-se ver o esquema do funcionamento deste serviço. Para fazer a ligação entre de um cliente Android, IOS ou de uma aplicação Chrome pode-se utilizar estas ferramentas e bibliotecas.

O Google Cloud Endpoints permite que, do ponto de vista do programador, a ligação cliente-servidor seja quase transparente, não sendo necessário fazer qualquer gestão da ligação, apenas fazer chamadas de funções previamente programadas no backend web da plataforma.

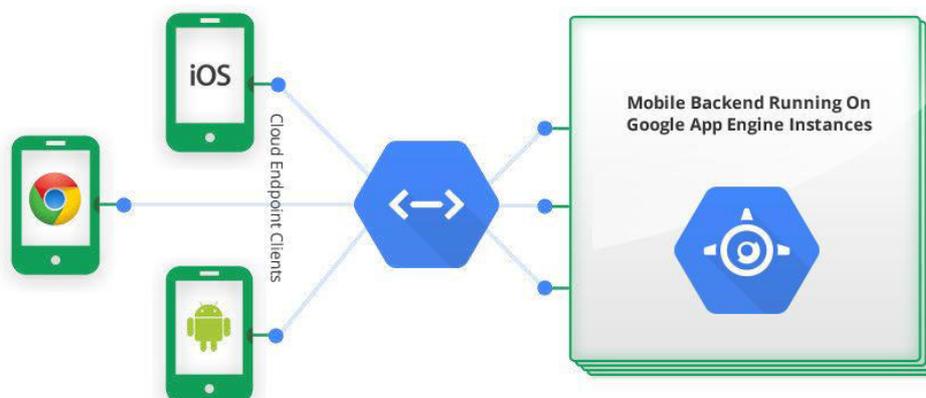


Figura 8 - Funcionamento do Google Cloud Endpoints

Além disso, proporciona ainda recursos para a implementação de infraestruturas críticas como é o caso da autenticação, reduzindo assim o esforço de desenvolvimento.

Com este serviço, para cada classe que necessite de aceder à aplicação de backend, é necessário gerar a classe Endpoint correspondente. Como se pode ver pela Figura 9 é fácil criar uma classe Endpoint. Depois de criada a classe apenas é necessário criar os métodos a disponibilizar para a aplicação cliente.

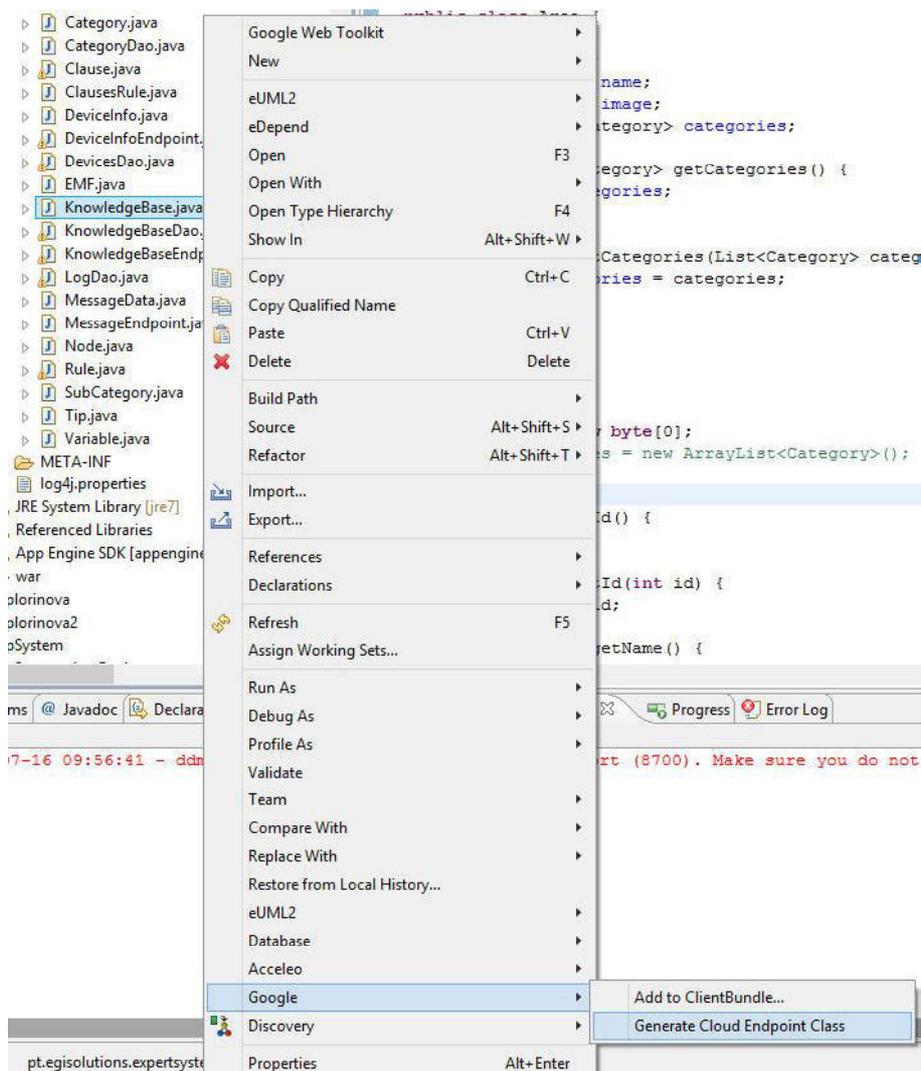


Figura 9 - Criação da classe Endpoint

Este serviço é utilizado por ambas as aplicações é através das bibliotecas geradas pelo Google Cloud Endpoints que estas acedem às respetivas aplicações de backend.

3.3.3 Google Cloud Messaging

O GCM (Google Cloud Messaging) é um serviço que permite enviar dados de um servidor para dispositivos Android, assim como para receber mensagem a partir de dispositivos na mesma ligação. O serviço GCM lida com todos os aspetos de enfileiramento de mensagens e encarrega-se de fazer a entrega da mensagem na respetiva aplicação no dispositivo de destino. Este serviço é completamente livre, não importando o número de mensagens que é necessário enviar e não tem quotas.

O funcionamento deste serviço na aplicação desenvolvida é explicado com mais detalhe na secção 3.9.5.

3.4 Casos de uso

A Figura 10 apresenta os dois tipos de atores intervenientes no sistema e os respetivos casos de uso. Os atores são o End User (Utilizador final) e o System Manager (Administrador do sistema).

O End User é a pessoa que vai fazer o registo do seu equipamento no sistema, seleciona 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 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.

As ações do Utilizador Final foram implementadas na aplicação My Expert, enquanto que as funções de gestão das bases de conhecimento foram implementadas na aplicação My Expert Manager.

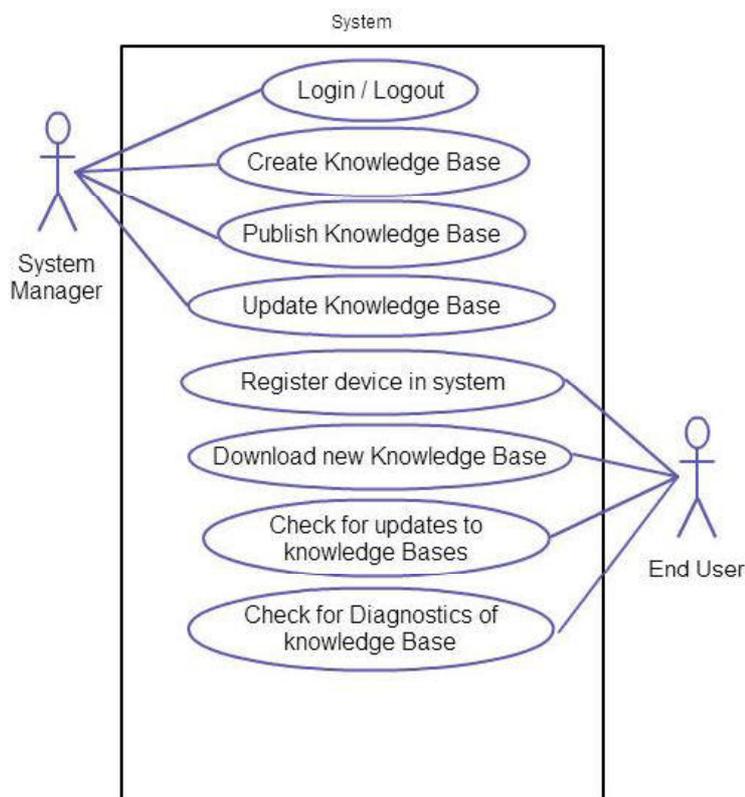


Figura 10 -Atores e respetivos casos de uso do sistema desenvolvido

Casos de uso do Ator System Manager:

- Login/logout: Para o administrador do sistema conseguir aceder ao menu para fazer a gestão de todas as bases de conhecimento, este terá que fazer login na aplicação, login este que é confirmado na aplicação de backend.
- Create Knowledge Base: A aplicação MyExpert Manager permite criar novas bases de conhecimento que são inseridas na base de dados através da aplicação de backend.
- Publish Knowledge Base: Após a criação da base de conhecimento esta não fica automaticamente disponível para os utilizadores finais da aplicação, é necessário publicá-la para que os utilizadores a consigam “ver” e fazer o download.
- Update knowledge Base: Através da aplicação MyEspert Manager o administrador pode fazer o download da base de conhecimento e proceder à alteração desta, e no final fazer o upload para a base de dados, processo este que no final desencadeia o envio de notificações para os clientes a informar da atualização efetuada para que a possam efectuar nos seus dispositivos.

Casos de uso do Ator End User:

- Register Device in system: Para que o dispositivo receba notificações este terá que efetuar o registo do dispositivo através do serviço Google Cloud Messaging. Este processo é descrito com mais detalhe na secção 3.10.6.
- Download new Knowledge Base: O utilizador final consegue ver todas as bases de conhecimento disponibilizadas pelo administrador do sistema, mas apenas descarrega para o seu dispositivo aquelas que forem do seu interesse para que as possa consultar.
- Update knowledge Bases: Quando o administrador do sistema efetua alterações em uma base de conhecimento, o utilizador final recebe uma notificação, caso o utilizador já tenha feito o download da Base de Conhecimento, este poderá fazer a sua actualização.
- Check for Diagnostics of knowledge Base: Esta é a função base de todo o sistema, em que o utilizador consulta uma base de conhecimento já descarregada. Aqui o sistema vai executar o motor de inferência para a base de conhecimento selecionada pelo utilizador, faz perguntas ao utilizador com o objetivo de chegar a um conclusão e apresenta-la ao utilizador.

3.5 Representação das Bases de Conhecimento

Com já foi referido na secção 2.3 as bases de conhecimentos no sistema podem ser representadas com base em regras ou árvores de decisão.

Para o caso das bases de conhecimento baseadas em regras, vamos assumir como exemplo a base de conhecimento representada a seguir, que serve para determinar se um determinado candidato se qualifica para uma posição em uma determinada empresa e qual a posição a oferecer.

Regra 10	SE	DIPLOMA = NÃO
	ENTÃO	POSIÇÃO = NÃO
Regra 20	SE	DIPLOMA = SIM
	ENTÃO	QUALIFICA = SIM
Regra 30	SE	DIPLOMA = SIM E
		DESCOBERTA = SIM
	ENTÃO	POSIÇÃO = PESQUISA
Regra 40	SE	QUALIFICA = SIM E

		MÉDIA < 3,5 E EXPERIÊNCIA >=2
	ENTÃO	POSIÇÃO = ENGENHEIRO DE SERVIÇO
Regra 50	SE	QUALIFICA = SIM E MÉDIA < 3,5 E EXPERIÊNCIA < 2
	ENTÃO	POSIÇÃO = NÃO
Regra 60	SE	QUALIFICA = SIM E MÉDIA >= 3,5
	ENTÃO	POSIÇÃO = ENGENHEIRO DE PRODUTO

Analisando cada uma das regras, chegamos à conclusão que podemos representar uma regra no 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 de 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.

Após esta análise tiramos as seguintes conclusões:

- Para cada base de conhecimento necessitamos de guardar qual a variável de conclusão, ou seja a resposta que pretendemos alcançar (no exemplo referido seria a variável POSIÇÃO).
- Cada knowledgeBase tem várias regras.
- Cada Regra é composta por uma parte “IF” e por uma parte “THEN”
- A parte “IF” pode ser composta por uma ou várias Cláusulas (tabela “Clauses”), ligadas por um operador Booleano.

- A parte “THEN” é composta por uma única cláusula, a cláusula de conclusão.

Após esta análise chegou-se ao diagrama ER para guardar a Base de conhecimento que será apresentação na secção seguinte.

No que diz respeito a bases de conhecimento com base em árvores de decisão vamos assumir como exemplo a árvore de decisão da Figura 11, retirada de [13].

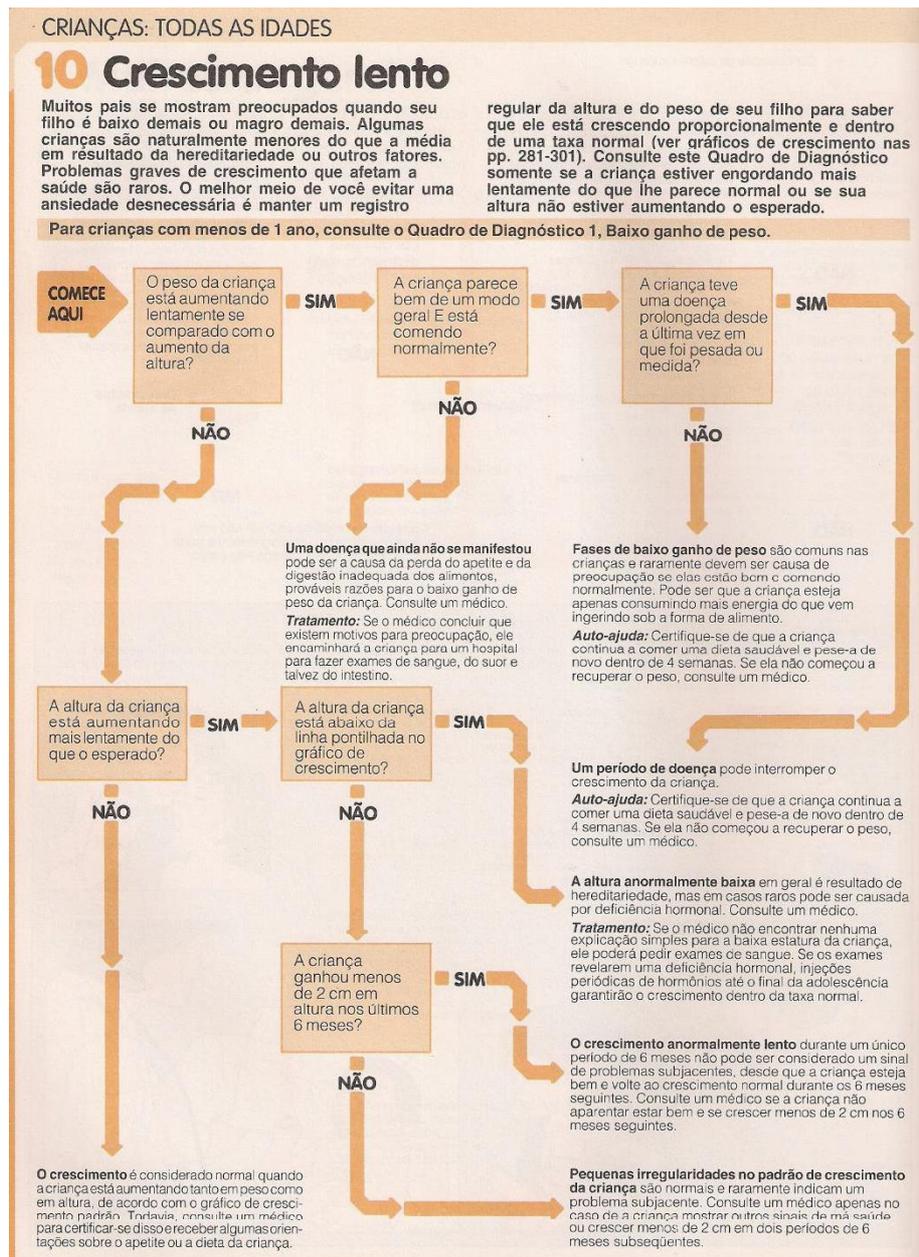


Figura 11 - Exemplo de Base de conhecimento em Árvore de Decisão

Para uma base de conhecimentos baseadas em árvores de decisão, necessitamos de guardar a informação relativa ao nó, e para cada um destes qual o nó que nos leva respondendo “Sim” a esse nó e qual o nó respondendo “Não”. Para além desta informação necessitamos de guardar se o nó é um nó raiz e se o nó é um nó folha, ou seja se é o primeiro ser analisado ou se depois deste não existe mais nós, ou seja encontrou-se a respostas. Como se pode ver no exemplo da Figura 11, alguns nós além de uma conclusão têm associada informação complementar (auto-ajuda ou tratamento) que necessita de ser guardada.

3.6 Armazenamento na Base de Dados

Para organizar as Bases de Conhecimento optou-se por separá-las em Áreas (Saúde, informática, etc.), Categorias (área de Saúde: Crianças, Adultos, Idosos), e SubCategorias (Categoria Crianças: Bebés com menos de um ano, crianças de todas as idades, Adolescentes). Esta organização foi escolhida para que o utilizador possa selecionar apenas as bases de conhecimento do seu interesse.

Para que o número de downloads das bases de conhecimento não fosse muito grande, optou-se por permitir que o utilizador final fizesse o download ao nível das Categorias. Assim, quando um utilizador faz um download de uma Categoria este descarrega todas as Bases de Conhecimento que estão associadas a todas as subcategorias da Categoria escolhida.

A seguir são descritos o diagrama Entidade Relacionamento da Base de dados alojada no servidor (Google Cloud SQL) e Base de Dados da aplicações cliente (aplicação MyExpert).

3.6.1 Base de dados alojada no Google Cloud SQL

No que se refere à base de dados alojada no serviço Google Cloud SQL, da análise anterior, chegamos ao diagrama ER apresentado na Figura 12. Assim temos a tabela Area, Category e SubCategory, que serve para fazer a organização das Bases de conhecimento, temos a tabela KnowledgeBases que é a onde são guardadas as informações da base de conhecimento, o tipo e se é pública. As tabelas Rules, ClausulesRules, Clauses e Variables são as tabelas para guardar a informação das bases de conhecimento baseadas em regras, enquanto que as Tabelas Node e Tip são as tabelas onde são guardadas as informações das bases de conhecimento baseadas em árvores de decisão.

Na tabela Devices são guardados todos os dispositivos que se registam para que possa ser possível enviar notificações de atualizações ou novas bases de conhecimento disponíveis.

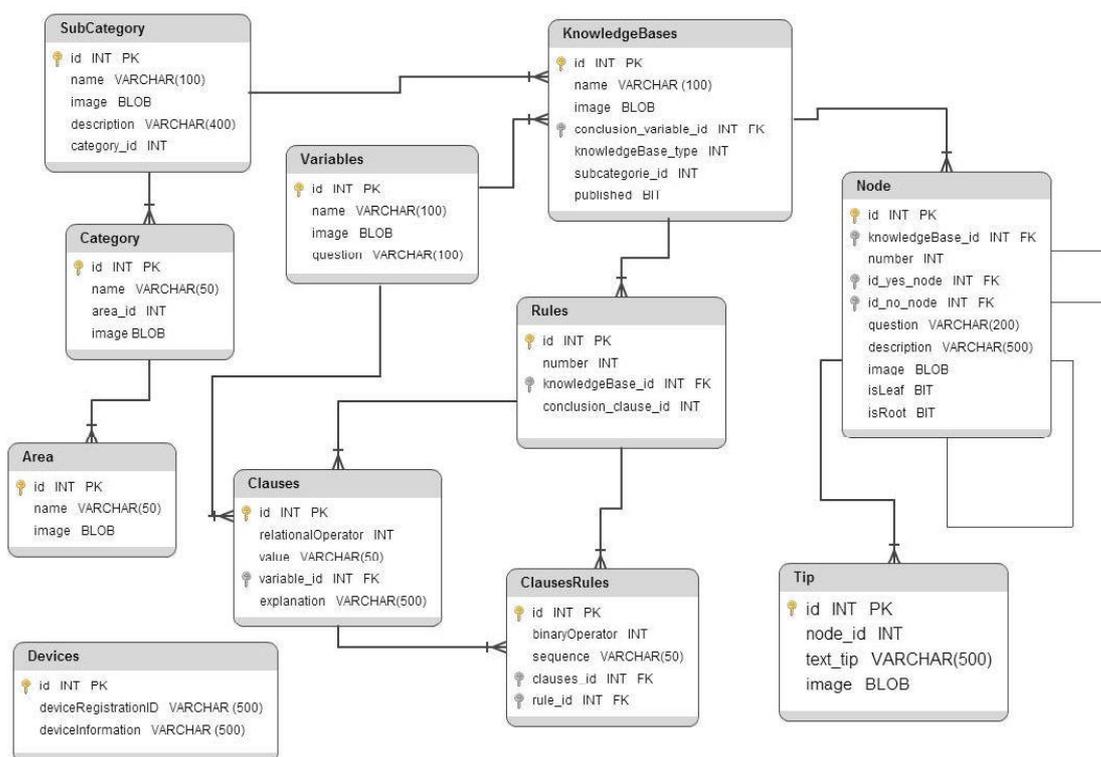


Figura 12 – Diagrama ER da base de dados presente no serviço Google Cloud SQL

De seguida é apresentada a estrutura de cada Tabela.

Tabela 2 - Descrição da tabela Area

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação da área	Numérico	Primária	X
name	Nome da área	Texto		X
image	Imagem descritiva da área.	Imagem		

Tabela 3 - Descrição da tabela Category

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação da categoria	Numérico	Primária	X
name	Nome da categoria	Texto		X
image	Imagem descritiva da categoria.	Imagem		
area_id	Código de	Numérico	Estrangeira	X

	identificação da área a que pertence a categoria			
--	--	--	--	--

Tabela 4 - Descrição da tabela SubCategory

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação da subcategoria	Numérico	Primária	X
name	Nome da Subcategoria	Texto		X
image	Imagem descritiva da subcategoria.	Imagem		
category_id	Código de identificação da categoria a que pertence a categoria	Numérico	Estrangeira	X
description	Descrição da subcategoria	Texto		

Tabela 5 - Descrição da tabela KnowledgeBases

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação da Base de Conhecimento	Numérico	Primária	X
name	Nome da Base de Conhecimento	Texto		X
image	Imagem descritiva da Base de Conhecimento.	Imagem		
conclusion_variable_id	Código da variável de conclusão da base de conhecimento (apenas para bases de conhecimento baseada em regras)	Numérico	Estrangeira	
knowledgeBase_type	Tipo de Base de Conhecimento, caso seja do tipo 1 indica que é baseada em regras, ou do tipo 2 do tipo baseada em	Numérico		X

	árvore de decisão.			
Subcategory_id	Código de identificação da subcategoria a que pertence a Base de Conhecimento	Numérico	Estrangeira	X

Tabela Rules

Tabela 6 - Descrição da tabela Rules

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação da regra	Numérico	Primária	X
number	Número da regra	Numérico		X
knowledgeBase_id	Código de identificação da Base de Conhecimento a que pertence	Numérico	Estrangeira	X
conclusion_clause_id	Código de identificação da cláusula de conclusão	Numérico	Estrangeira	X

Tabela 7 - Descrição da tabela ClausesRules

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação da cláusula da regra	Numérico	Primária	X
binaryOperator	Operador binário	Numérico		X
sequence	Código de identificação da Base de Conhecimento a que pertence	Texto		X
clause_id	Código de identificação da cláusula	Numérico	Estrangeira	X
rule_id	Código de identificação da regra	Numérico	Estrangeira	X

Tabela 8 - Descrição da tabela Clauses

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação da cláusula	Numérico	Primária	X
relationalOperator	Operador relacional	Numérico		X
value	Valor a comparar com o operador relacional	Texto		X
variable_id	Código de identificação da variável	Numérico	Estrangeira	X
explanation	Explicação da cláusula	Texto		

Tabela 9 - Descrição da tabela Variables

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação da variável	Numérico	Primária	X
name	Nome da variável	Texto		X
image	Imagem descritiva da pergunta a fazer ao utilizador	Imagem		
question	Questão a colocar ao utilizador	Texto		X

Tabela 10 - Descrição da tabela Node

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação do nó	Numérico	Primária	X
knowledgeBase_id	Código de identificação da base de conhecimento que pertence o nó	Numérico	Estrangeira	X
number	Número do nó	Numérico		X
id_yes_node	Nó que está ligado á resposta “Sim” do utilizador	Numérico	Estrangeira	

id_no_node	Nó que está ligado á resposta “Não” do utilizador	Numérico	Estrangeira	
question	Pergunta a colocar ao utilizador	Texto		X
description	Informação complementar da pergunta	Texto		
image	Imagem descritiva da pergunta a fazer ao utilizador	Imagem		
isLeaf	Indicação se o nó é um nó folha	Booleano		X
isRoot	Indicação se o nó é um nó raiz.	Booleano		X

Tabela 11 - Descrição da tabela Tip

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação da dica	Numérico	Primária	X
nod_id	Código de identificação do nó	Numérico	Estrangeira	X
texto_tip	Informação extra a apresentar ao utilizador	Texto		X
imagem	Imagem descritiva da dica	Texto		

Tabela 12 - Descrição da tabela Devices

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação dispositivo	Numérico	Primária	X
deciceRegistrationId	Código de registo no servido GCM	Texto		X
deviceInformation	Informação do dispositivo como marca e modelo	Texto		X

3.6.2 Base de dados mobile

A base de dados que fica no dispositivo mobile tem a estrutura muito idêntica à do servidor, com a diferença nos seguintes pontos:

- Tipo de dados em todas as tabelas;
- Na tabela “Category” uma campo que indique se foi feito o download do servidor ou não.
- Tabela “UpdateMessage” onde são guardas todas as notificações de atualizações recebidas para processamento.

O ER da Bases de Dados Mobile é apresentado na Figura 13.

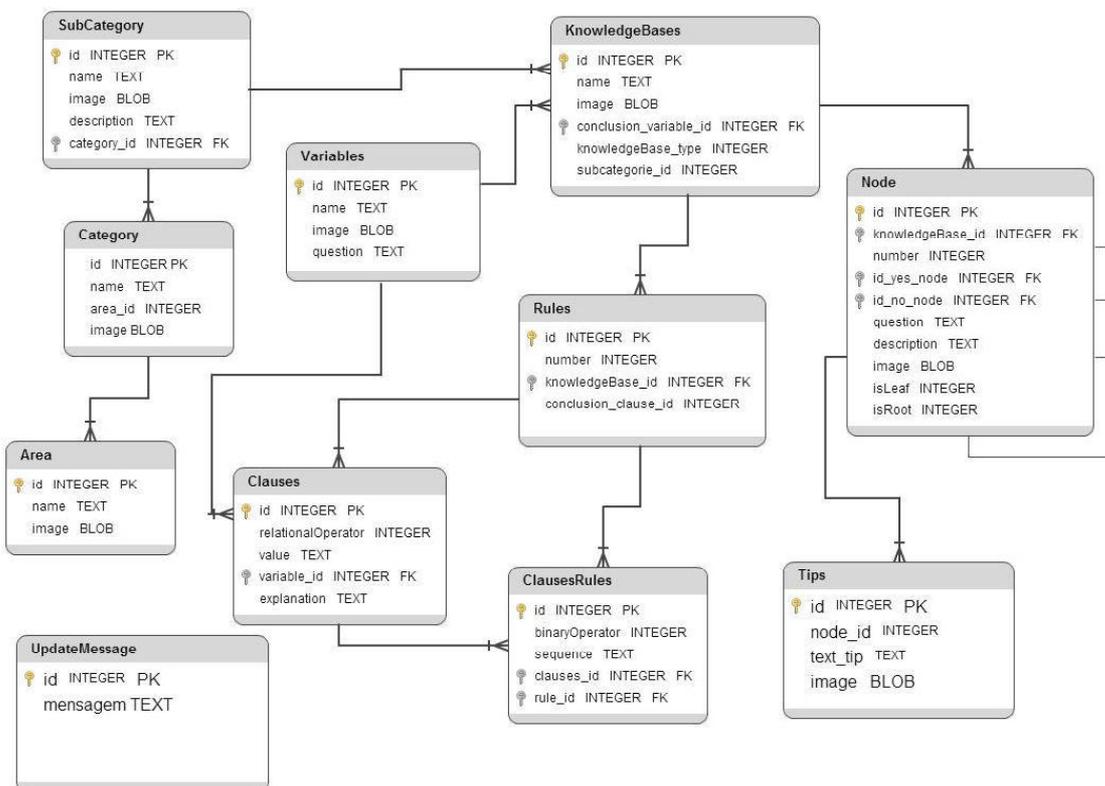


Figura 13 - ER da Base de dados Mobile

A estrutura da tabela UpdateMessage é a seguinte:

Tabela 13 - Descrição da tabela UpdateMessage

Campo	Descrição	Tipo	Chave	Obrigatório
id	Código de identificação	Numérico	Primária	X

	dispositivo			
message	Mensagem recebida para atualização	Texto		X

3.7 Algoritmos do Motor de Inferência

Como já foi referido o motor de inferência vai atuar de acordo com o tipo de base de conhecimento escolhida pelo utilizador.

Para o caso de bases de conhecimento baseado em regras 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 que 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, o motor 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 algoritmo do processo de encadeamento para trás utilizado foi o descrito em[8] e será apresentado de seguida.

Vamos tomar como exemplo a base de conhecimento apresentada na **Erro! A origem da referência não foi encontrada.**

Para proceder ao encadeamento para trás, o algoritmo utilizado necessita de construir um conjunto de estruturas de apoio ao processo de inferência, esta estrutura deriva diretamente da base de conhecimento.

Após se ter construído estas estruturas podemos utilizar o seguinte algoritmo:

Algoritmo 1 – Algoritmo utilizado para o encadeamento para trás

1. *Identificar Conclusão*
 2. *Procurar na lista de conclusões a 1ª ocorrência variável de conclusão. Colocar o número da regra e 1 para representar o número da cláusula.*
 3. *Instanciar a cláusula SE.*
Para cada variável da regra:
 Se variável não for variável de conclusão e não tiver na lista de conclusão
 Pedir ao utilizador para inserir um valor
 Se não, se for variável de conclusão
 Colocar o número da regra da variável de conclusão no topo da pilha, e voltar ao passo 3.
 4. *Se unidade no topo da pilha não poder ser instanciada*
 Remover a unidade do topo da pilha
 Se existir outra ocorrência do nome da variável de conclusão na lista de conclusões:
 Voltar ao passo 3
 Se não
 Regra da conclusão anterior é falsa
 Se não existir uma conclusão anterior
 Notificar o utilizador que a resposta não pode ser encontrada.
 Se não
 Voltar ao passo 4.
 Se não Se regra poder ser instanciada
 Removê-la da pilha
 Se existir outra variável abaixo dela
 Incrementar número da cláusula
 Para as cláusulas restantes voltar ao passo 3
 Se não
 Respondemos à nossa pergunta, o utilizador chegou a uma conclusão.
-

Para o caso das bases de conhecimento baseadas em árvores o algoritmo aplicado pelo motor de inferência para procurar respostas para o utilizador é o seguinte:

Algoritmo 2 – Algoritmo de procura dos nós:

GET Nó Root

Enquanto Novo Nó não for nó Leaf

Lê resposta Utilizador

Se resposta == SIM

Get Nó id = id_yes_node

Se não Se resposta == NÃO

Get Nó id = id_yes_node

Apresentar a resposta encontrada ao utilizar.

3.8 Arquitetura do Sistema

Com já foi referido o SP foi implementado para a plataforma Android através de uma arquitetura baseado em serviços, utilizando a infraestrutura de computação na nuvem da Google App Engine. A utilização desta permitiu a abstração em relação ao trabalho administrativo do sistema, como é o caso do balanceamento de carga, escalonamento e manutenção do servidor, pois a infraestrutura da cloud da Google já tem estes serviços implementados, sendo estas as vantagens da utilização deste tipo de serviços.

A Figura 14 apresenta uma visão geral da arquitetura do sistema proposto.

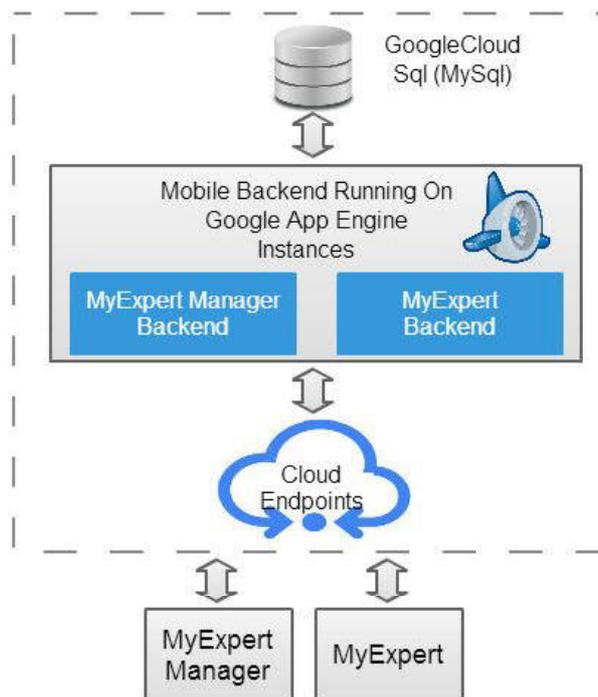


Figura 14 - Arquitetura Geral do sistema

A arquitetura pode ser dividida em três componentes principais: a aplicação MyExpert Manager, MyExpert e Base de dados alojada no Google Cloud Sql.

A aplicação MyExpert Manager, que engloba a aplicação desenvolvida para a plataforma Android, permite a criação e gestão de bases de conhecimento. Esta engloba também a aplicação MyExpert Manager Backend, aplicação de backend alojada na plataforma de Computação na Nuvem da Google, Google App Engine, que faz a “ponte” entre a aplicação cliente e base de dados;

A aplicação MyExpert, igualmente desenvolvida para a plataforma Android, é a aplicação para o utilizador final, sendo nesta onde está implementada a máquina de inferência do SP. Neste considere-se de igual forma a respectiva aplicação de Backend – MyExpert Backend.

A Base de Dados é o que contem a coleção de bases de conhecimento do sistema e respetiva estrutura, armazenadas na Google Cloud SQL.

Nos capítulos seguintes serão apresentados com mais algum detalhe estas aplicações.

3.9 Aplicação MyExpert Manager

A Figura 15 apresenta os módulos que formam a aplicação MyExpert Manager. Na figura estão representados os módulos que compõem a aplicação desenvolvida para tablet assim como os módulos da aplicação de backend e respetivas ligações. No anexo D é apresentado o funcionamento desta aplicação.

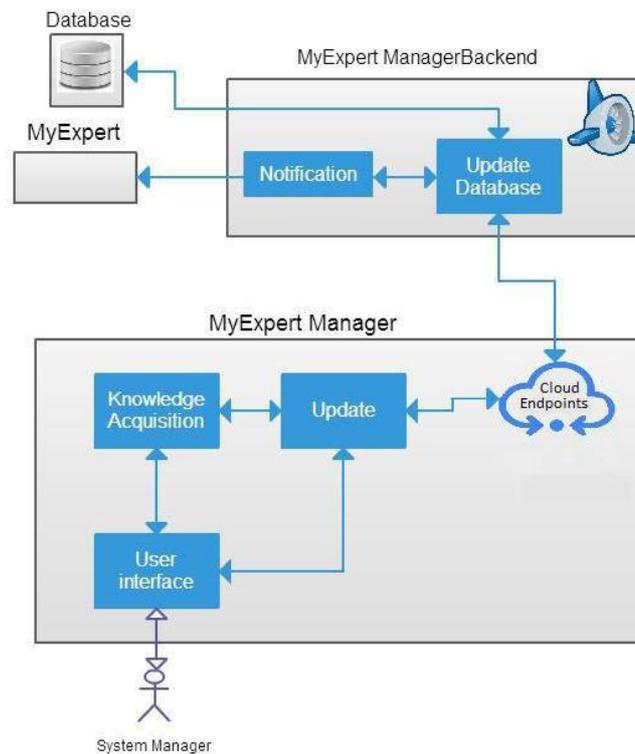


Figura 15 - Arquitetura da Aplicação MyExpert Manager

3.9.1 Módulo User Interface

O módulo User Interface é responsável pela interação entre o utilizador e o módulo Knowledge Acquisition. Este representa todos os ecrãs da aplicação MyExpert Manager, que permitem ao utilizador fazer toda a gestão das bases de conhecimento disponíveis para o utilizador final.

3.9.2 Módulo Knowledge Acquisition

Este módulo é responsável pela criação e atualização das bases de conhecimento. Este comunica com o módulo User Interface, onde vai recolher a informação introduzida sobre as bases de conhecimento, constrói a base de conhecimento e envia para o módulo Update que faz com que esta seja introduzida/atualizada na base de dados.

Para cada método de introdução, este módulo após o utilizador introduzir todas as informações da base de conhecimento, constrói a base de conhecimento no formato especificado, e passa a base de conhecimento para o módulo Update.

3.9.3 Módulo Update

O módulo Update é o módulo responsável pela inserção e atualização das bases de conhecimento na Google Cloud SQL.

Os detalhes deste módulo serão apresentados na secção 3.11, pois é difícil fazer a separação entre o módulo Update da aplicação MyExpert Manager da aplicação MyExpert.

3.9.4 Módulo Update Database

O módulo Update Database é responsável pela inserção e atualização de bases de conhecimento na base de dados MySql localizada na Google Cloud Sql assim como de enviar para o módulo de notificações o tipo de atualização que foi feita no sistema.

3.9.5 Módulo Notification

Este módulo faz uso do serviço da Google - Google Cloud Messaging para enviar notificações para os clientes registados com a informação de que existem atualizações disponíveis no servidor.

Para permitir informar que tipo de atualização foi feita e o que foi atualizado ou foi criado um protocolo simples de comunicação baseado em mensagens de texto. O formato das mensagens é explicado a seguir.

Cada mensagem é formada por um conjunto de caracteres e pode ser dividida em 4 partes. Esta terá o seguinte especto:

A # B # C # D

Onde:

A-> tipo de atualização

1 - Adiciona

2 - Elimina

3 - Update

B -> o que atualiza

1 - Área

2 - Categoria

3 - Subcategoria

4 - Base de Conhecimento

5 - Menu

C -> id do que atualiza

D -> Mensagem

Com a definição deste protocolo não será necessário atualizar toda a informação da base de dados apenas quando a atualização que se quer publicar é apenas um erro ortográfico por exemplo.

3.10 Aplicação MyExpert

A Figura 16 apresenta a arquitetura da aplicação MyExpert. Esta compreende a aplicação desenvolvida para smartphone, aplicação de Backend e respetivos serviços utilizados. Nas secções seguintes irá ser apresentado com mais algum detalhe cada módulo da aplicação. No anexo C é apresentado o funcionamento desta aplicação.

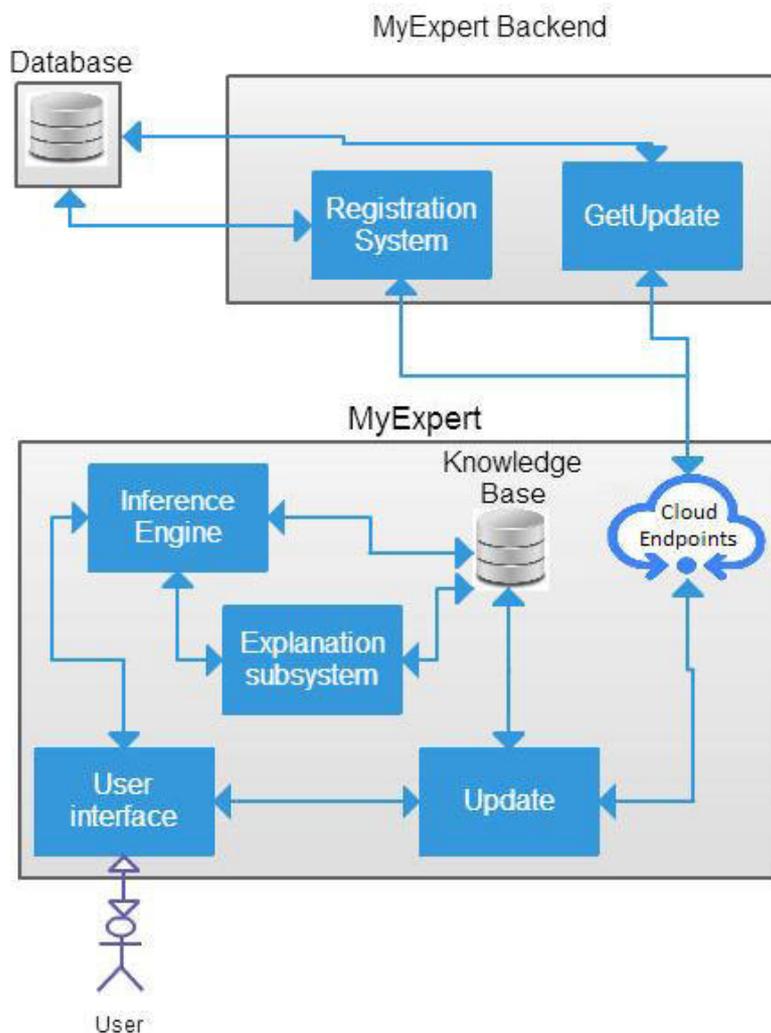


Figura 16 - Arquitetura da aplicação MyExpert

3.10.1 Módulo User 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.

3.10.2 Módulo Inference Engine

O módulo Inference Engine (máquina de inferência) é o responsável pelo processamento da base de conhecimentos recorrendo a raciocínio e planeamento lógico.

Com já foi descrito, para cada tipo de base de conhecimento é desencadeado seu processo de encadeamento correspondente. Estes processos foram descritos na Secção 3.7. Para o utilizador final este processo é totalmente transparente.

3.10.3 Módulo Explanation Subsystem

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.

Para o caso da base de conhecimento baseado em regras a explicação para a solução encontrada é a parte SE na regra que foi utilizada para chegar à conclusão, enquanto para o caso da Base de Conhecimento baseada em árvores de decisão a explicação consiste no caminho que leva do nó Root ao nó Leaf.

3.10.4 Módulo Update

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 atualizações disponíveis e, caso a aplicação MyExpert possua essa base de conhecimento armazenada, notifica o utilizador através da interface que pode proceder à atualização. É também através deste módulo que podem ser selecionadas e descarregadas novas bases de conhecimento entretanto criadas e armazenadas na plataforma Cloud do sistema.

Na secção 3.11 é mostrado com mais detalhe como funciona todo o processo de atualização do sistema.

3.10.5 Módulo KnowledgeBase

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 do Google Cloud SQL onde estão armazenadas numa base de dados MySQL. As bases de dados têm a mesma estrutura no que se refere ao armazenamento das bases de conhecimento pelo que a transferência destas é realizada de forma transparente recorrendo ao cloud endpoints.

3.10.6 Módulo Registration

Este módulo é responsável por fazer o registo do dispositivo móvel na base de dados do sistema, para que quando existir uma atualização do sistema este possa ser notificado da existência da atualização. A Figura 17 representa o funcionamento do módulo.

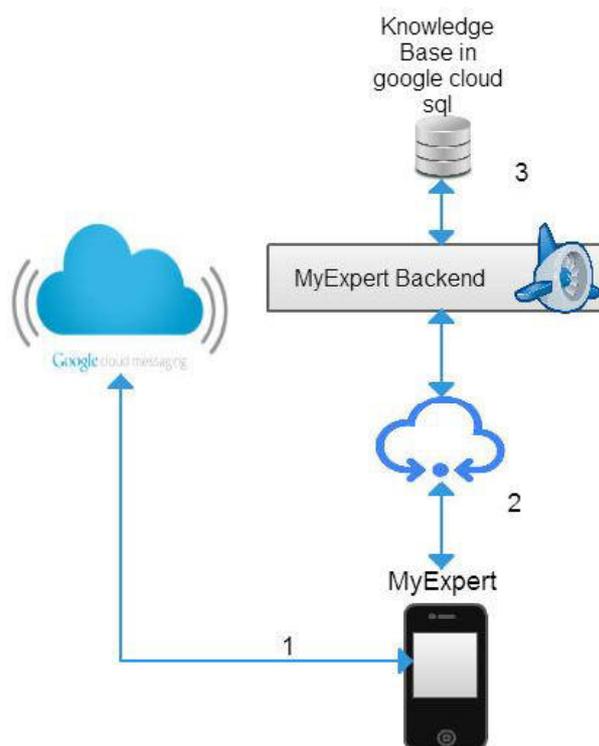


Figura 17 - Funcionamento do Módulo Registration

No arranque da aplicação é verificado se esta já se tem um “registration_id”, ou seja se já foi efetuado o registo da aplicação no servidor, caso este registo ainda não tenha sido efetuado é desencadeado o seguinte processo:

- 1- Aplicação MyExpert comunica com o serviço Google Cloud Messaging para fazer o registo da aplicação neste serviço. Deste modo quando a aplicação de backend pedir a este serviço, para enviar uma mensagem para o cliente este sabe qual o dispositivo a enviar.
Este devolve o Registration-Id deste cliente.
- 2- A aplicação comunica com a aplicação MyExpert Manager, através do Google Cloud Endpoints e passa-lhe o “Registration-Id” devolvido pela Google Cloud Messaging.
- 3- A aplicação MyExpert Backend vai procurar na base de dados se o “Registration-Id” existe na Base de dados, caso não exista insere-o na Tabela “Devices”. Esta verificação torna-se importante para assegurar que um dispositivo não irá receber várias notificações para a mesma atualização.

3.10.7 Módulo GetUpdate

Este é o módulo responsável por devolver as atualizações solicitadas pela aplicação MyExpert a pedido do utilizador ou automaticamente, caso o utilizador tenha atualizações automáticas.

Do lado do servidor não é guardada qualquer informação que bases de conhecimento cada utilizador fez download, assim quando existe uma atualização de uma determinada base de conhecimento é enviada uma notificação para todos os dispositivos registados, ficando a responsabilidade do lado da aplicação cliente de descartar a notificação, e não a apresentar ao utilizador caso não se aplique.

3.11 Mecanismo de atualização

A Figura 18 representa o módulo de atualizações do sistema. De seguida é apresentado o processo de atualização.

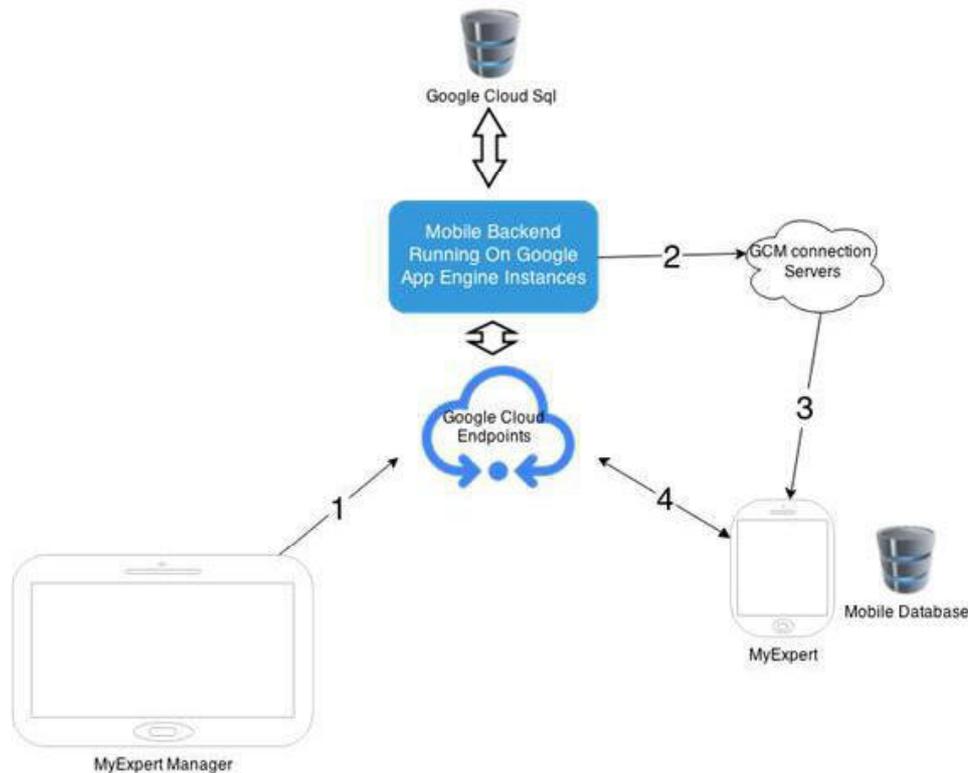


Figura 18 - Funcionamento do módulo de atualizações

- 1- Aplicação MyExpert faz uma atualização/inserção de uma base de conhecimentos no servidor.
- 2- Aplicação de backend envia uma mensagem através do serviço de envio de dados da Google – Google Cloud Messaging – para todos os dispositivos.
- 3- Quando existir uma ligação de dados com o dispositivo, é enviada a mensagem para todos os dispositivos registados. O utilizador recebe uma notificação de que existe uma atualização de uma base de conhecimento que já fez download ou que está disponível uma nova base de conhecimento para download.

No caso de o utilizador ter a opção de atualização automática, o dispositivo efetua uma ligação com a aplicação de backend para transferir essa atualização, caso contrário a atualização é efetuada a pedido do utilizador.

3.12 Publicação da aplicação na loja de aplicações Google Play

Para a distribuição da aplicação optou-se pela colocação da aplicação na loja de aplicações da Google – Google Play.

A Figura 19 mostra a aplicação publicada e disponível para download.

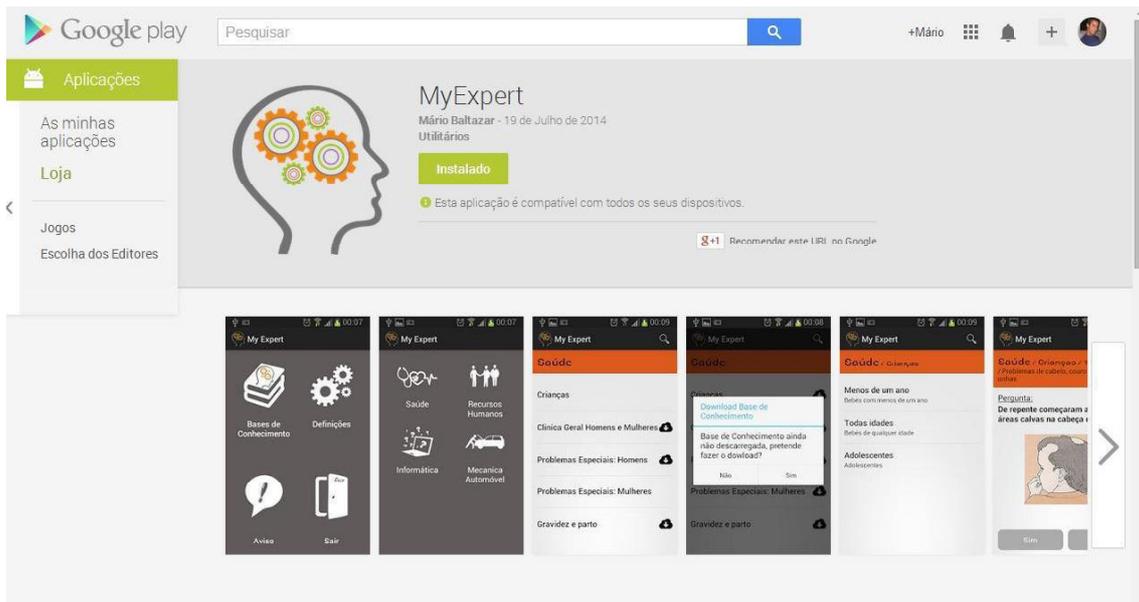


Figura 19 – Página da Aplicação no Google Play

No que diz respeito aos modelos de negócio da aplicação, as opções mais comuns são a publicação da aplicação por um valor nas lojas de aplicações, ou a colocação de banners com publicidade, em que é pago por cliques nos respectivos banners. A primeira forma é aconselhada quando a aplicação é muito específica nas suas funções e/ou o seu público-alvo se considera muito restrito, enquanto que a segunda é aconselhada quando o destinatário é o público em geral, ou seja, é tentado lucrar o investimento na aplicação pela quantidade de utilização que os utilizadores lhe irão dar. Outra alternativa é combinar estas duas formas, ter uma aplicação paga, e ter outra com recursos limitados com publicidade.

A aplicação desenvolvida é considerada de uso geral, logo o método escolhido foi a colocação de banners de publicidade, através do serviço de publicidade Google AdMob. Com esta opção os ganhos com a aplicação ficam a dever-se aos cliques nos banners publicitários que irão aparecer na aplicação.

CAPÍTULO 4

TESTES E RESULTADOS

Para testar todo o sistema desenvolvido fizeram-se alguns testes com a aplicação de modo a poder retirar algumas conclusões em relação a todo o trabalho desenvolvido. Foram efetuados alguns testes, nomeadamente no que se refere á capacidade/velocidade com que a aplicação de backend responde a um pedido de inserção de uma base de conhecimento, ao tempo desde que o administrador clicou para publicar uma base de conhecimento e o tempo em que os utilizadores recebiam a notificação da nova base de conhecimento, testes ao funcionamento da máquina de inferência, assim como testes em relação à usabilidade do sistema.

O primeiro teste a ser realizado foi o teste ao funcionamento do motor de inferência. Para este teste foi utilizada uma base de conhecimento na área da saúde, mais precisamente sobre diagnóstico de doenças em bebés. A base de conhecimento baseada em regras foi criada a partir de 5 quadros de diagnóstico apresentados em [13]. Como no momento em que foram efetuados estes testes o sistema não permitia a representação de bases de conhecimento baseado em árvores de decisão foi utilizado o algoritmo descrito em [8] para fazer a passagem de árvores de decisão para regras.

O teste realizado consistiu em simular 10 situações de mal-estar para um bebé 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 realizadas para obter o diagnóstico.

A Tabela 14 apresenta os resultados obtidos.

Tabela 14 - Resultados do teste do motor de inferência

	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

Após analisar os resultados podemos concluir que o SP encontra sempre o diagnóstico esperado, apesar de ter que colocar mais questões (em 6 dos 10 casos), do que no caso do uso direto dos quadros de diagnóstico. Esta diferença deve-se a que o mecanismo de encadeamento para trás, usado pelo motor de inferência, utiliza a ordem pela qual as regras se encontram na base de conhecimento para escolher a próxima regra a analisar.

Uma possível solução pode ser implementar um método de escolha mais eficiente, sensível ao contexto, que dê preferência às regras que envolvam dados obtidos mais recentemente.

O próximo teste a ser realizado foi o teste dos tempos de resposta da aplicação de backend a pedidos de inserção de bases de conhecimento na base de dados. Para esse teste foram inseridas 2 base de conhecimentos do tipo árvores de decisão com dimensões de 10 e 20 nós, sendo repetida a inserção de cada uma delas 5 vezes, em diferentes intervalos de tempo e foi feita uma média dos tempos de resposta para retirar a hipótese de que a utilização da rede poderia influencia o resultado. Para isso foi feito o depuramento mandando escrever nas diferentes aplicações o momento no tempo em que as ações foram executadas. A Figura 20 representa o teste efetuado. Os momentos onde foram marcados os tempos são:

- t1: momento em que o administrador do sistema pede para adicionar a base de conhecimento no servidor.
- t2: momento em que a aplicação de backend recebe o pedido para adicionar a base de conhecimento na base de dados.

- t3: momento em que a nova base de conhecimentos é completamente adicionada na base de dados.
- t4: momento em que todas as notificações são enviadas.
- t5: momento em que a aplicação MyExpert Manager recebe a confirmação de todo o processo de atualização.
- t6: momento em que os dispositivos recebem a notificações de que existe uma nova base de conhecimentos para fazer donwload. Como a hora dos dispositivos não está sincronizada, estes tempos não foram considerados, pois não era possível compará-lo. Mas pela observação efetuada podemos concluir que o tempo que a receção da notificação era quase instantânea.

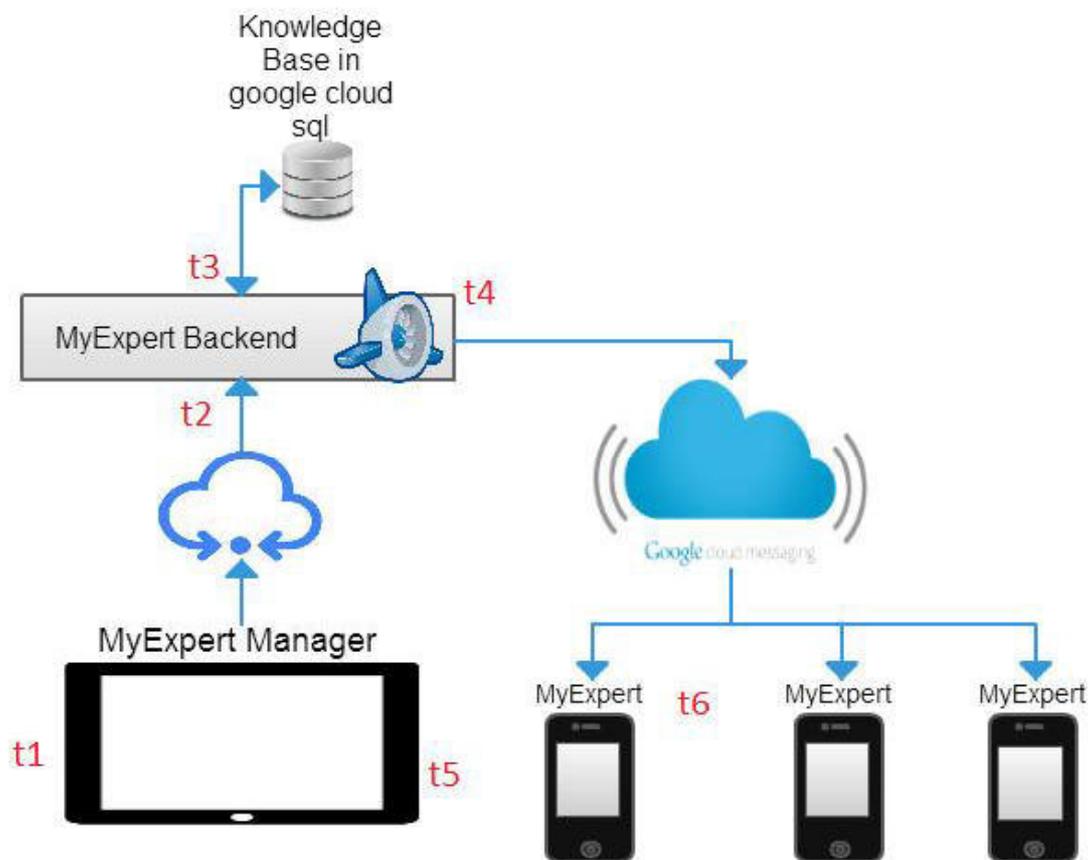


Figura 20 - Testes de velocidade de inserção base de conhecimento

A Tabela 15 e a Tabela 16 mostram os tempos medidos durante o teste efetuado. Como o teste foi efetuado com diferentes dispositivos e os relógios destes não estavam totalmente sincronizados não foi possível comparar todos os tempos. Assim para o tempo total da atualização foram comparados os tempos t1 com o tempo t5, da aplicação MyExpert Manager, enquanto que, para o tempo da atualização no servidor, foram comparados os tempos t2 e t4. Da análise dos tempos podemos concluir que a plataforma Google App Engine tem um opima resposta aos pedidos das aplicações clientes, podemos ainda concluir que o tempo da atualização é quase que da inteira responsabilidade da ligação que tem que ser estabelecida da aplicação MyExpert Manager com a respetiva aplicação de backend. Como se pode ver pelos resultados obtidos, a diferença do número de nós na base de conhecimento não tem impacto no tempo de atualização ou este é insignificante.

Tabela 15 - Tempos de atualização da base de conhecimento com 10 nós

		Exp1	Exp1	Exp3	Exp4	Exp5
Tempo total de atualização	Tempo	24s	9s	10s	10s	10s
	Média	12s				
Tempo atualização no servidor	Tempo	9s	7s	0s	1s	0s
	Média	3s				

Tabela 16 - Tempos de atualização da base de conhecimento com 20 nós

		Exp1	Exp1	Exp3	Exp4	Exp5
Tempo total de atualização	Tempo	21s	16s	3s	10s	8s
	Média	12s				
Tempo atualização no servidor	Tempo	1s	1s	1s	1s	2s
	Média	1s				

De seguida foi testado o tempo de espera entre o momento em que o administrador do sistema publicava uma base de conhecimento já criada e o momento em que os dispositivos recebiam a notificação. Para este teste foram utilizados 3 smartphones diferentes e foram efetuadas medições com 5 notificações. Mais uma vez, neste teste como a hora dos dispositivos não estava sincronizada, não foi feito o depuramento com a hora da atualização. Foi feita a tentativa de medir o tempo com um cronómetro, mas por este intervalo de tempo ser tão curto o intervalo de tempo que foi possível medir em um dos testes foi de apenas 3 segundos. Ou seja

nos testes efetuados, no máximo, o último smartphone a receber a notificação passado 3 segundos.

No que diz respeito a tempos foi ainda testado o envio de uma notificação estando os smartphones sem qualquer ligação de dados, ou seja, estes estavam impedidos de receber as notificações, e foi medido o tempo desde que a ligação à Internet foi estabelecida até que o utilizador recebe a notificação. Com o uso do serviço GCM não é necessário estar a verificar periodicamente por atualizações no servidor, com este serviço assim que existir uma ligação é feito a troca das mensagens. Estes tempos estão representados na Tabela 17. Com os testes efetuados concluímos que o serviço é muito eficiente, sendo quase instantâneo a troca de mensagens assim que existe uma ligação.

Tabela 17 - Tempo desde que a ligação á internet é estabelecida até que o utilizador recebe a notificação

	Exp 1	Exp2
disp 1	00:00:03:17	00:00:03:80
disp 2	00:00:03:36	00:00:03:28

O último teste efetuado foi o teste da usabilidade da aplicação MyExpert, pois considera-se que o sucesso da aplicação depende da aceitação da mesma por parte dos utilizadores finais

Para avaliar a usabilidade da aplicação foi usado o teste SUS (System Usability Scale) baseado num questionário que os utilizadores devem responder. O teste foi desenvolvido por John Brooke, no laboratório da Digital Equipment Corporation no Reino Unido e permite representar a usabilidade da aplicação numa escala de 0 a 100. O questionário é composto por 10 questões com 5 opções de resposta. Cada resposta ao questionário é assinalada numa escala Likert, que varia entre “Discordo totalmente” e “Concordo totalmente”. Para disponibilizar o questionário aos utilizadores foi usado o serviço Google Docs. O formulário utilizado está presente no Anexo B.

O resultado deste teste é obtido somando a contribuição de cada item. Para os itens 1,3,5,7 e 9 a contribuição é a posição respondida menos 1. Para os itens 2,4,6,8 e 10 a contribuição é 5- posição respondida. Multiplicando a soma da pontuação por 2.5 obtém-se o valor do SUS [14].

Para o presente teste foram efetuados 15 inquéritos do inquérito presente no anexo B. A pontuação média da SUS dos testes realizados foi de 81. Tendo em conta que os valores de referência indicados pelo autor da escala é de 68 considera-se que o sistema apresenta um bom índice de usabilidade.

O gráfico da Figura 21 apresenta a média dos resultados para cada pergunta depois de feita a análise das respostas. Como podemos verificar pelo gráfico a questão que obteve menor pontuação foi a pergunta “Senti-me muito confiante com o sistema” o que pode ser explicado pelas bases de conhecimento não transmitirem muita confiança ao utilizador.

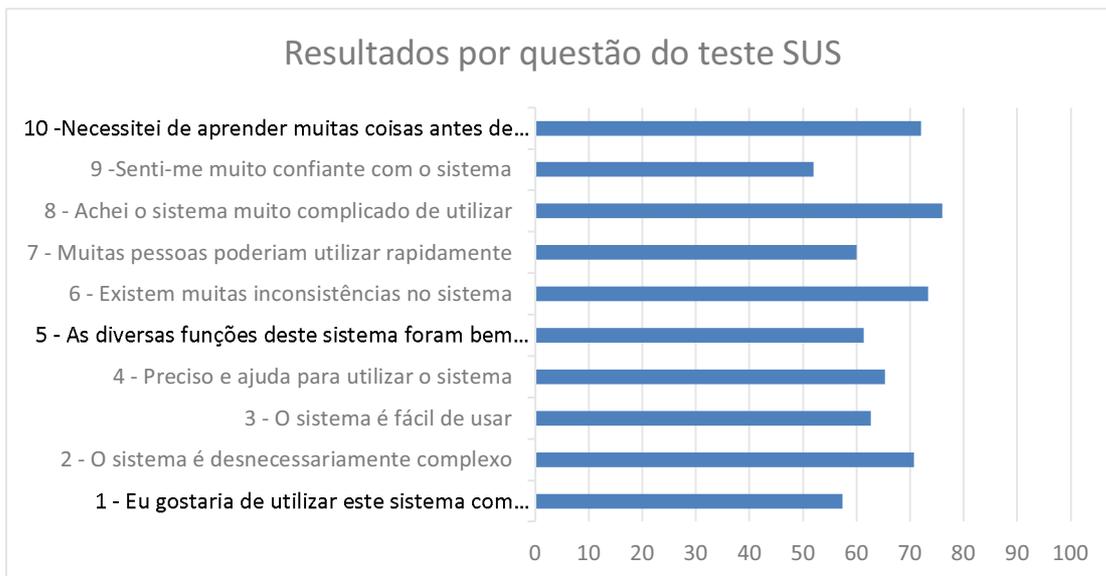


Figura 21 - Média SUS dos resultados para cada questão

Segundo referido no artigo [15] através do modelo SUS é possível avaliar os seguintes aspetos nas perguntas do questionário:

- Facilidade de Aprendizagem: questões 3, 4, 7 e 10;
- Eficiência: 5, 6 e 8;
- Facilidade de memorização: 2;
- Minimização dos erros: 6;
- Satisfação: 1, 4, 9.

Aplicando este cálculo para os resultados obtidos obtemos o seguinte gráfico.

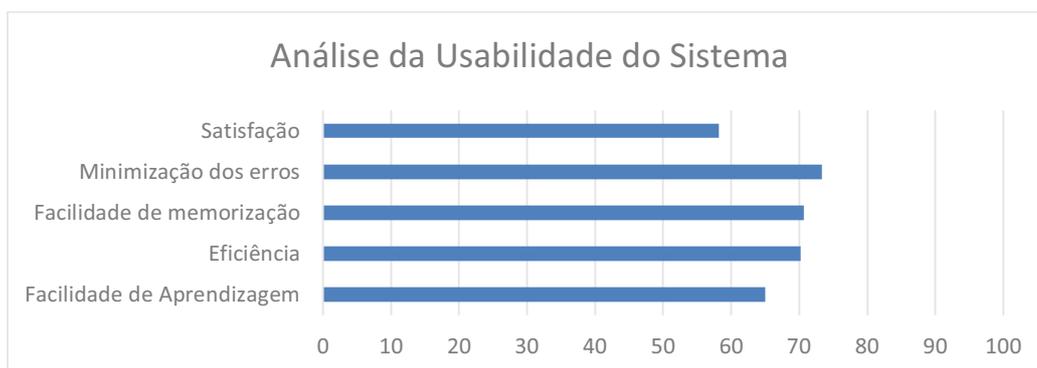


Figura 22 - Análise das características de usabilidade da norma 9126

Pela análise do gráfico anterior podemos concluir que o sistema apresenta boas características de usabilidade, existindo alguns pontos como é a questão da satisfação que poderá ser melhorada.

CAPÍTULO 5

CONCLUSÕES

Neste trabalho foi feito o desenvolvimento de um sistema pericial para dispositivos móveis com base em Computação na Nuvem Google App Engine, o que permite colocar ao alcance de um utilizador comum diferentes bases de conhecimento de diferentes áreas. 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.

Os objetivos propostos foram alcançados e o sistema encontra-se em funcionamento estando acessível ao público em geral através da aplicação MyExpert disponível na loja de aplicações da plataforma Android.

O trabalho desenvolvido foi apresentado recentemente na conferência CISTI 2014 - 9ª Conferência Ibérica de Sistemas e Tecnologias de Informação que teve lugar em Barcelona, Espanha. O artigo apresentado na conferência encontra-se no Anexo A.

Pelos resultados dos testes efetuados, a arquitetura desenvolvida para o sistema revelou-se bastante eficiente. Os tempos medidos no processo de atualização de uma base de conhecimentos consideram-se bastante satisfatórios.

A nível de usabilidade da aplicação MyExpert, nos testes efetuados podemos considerar que a aplicação tem um bom nível de usabilidade, no entanto existem alguns pontos que poderão ser melhorados como o resultado das respostas à questão 1 do teste SUS, “Eu gostaria de utilizar este sistema com frequência”. Para melhorar o resultado obtido a esta questão será necessário disponibilizar mais bases de conhecimento de mais áreas para deste modo aumentar o interesse pela aplicação. Outra questão cujo resultado poderia ser melhorado é a questão 9 “Senti-me muito confiante com o sistema”, melhorando a qualidade das bases de conhecimento.

Como trabalho futuro aponta-se a criação de mais bases de conhecimento de diferentes áreas, o que poderá implicar a necessidade de alterar o modelo de negócio para uma versão paga, com acesso a mais bases de conhecimento. A disponibilização de bases de conhecimento em diferentes línguas também pode ser uma funcionalidade interessante a implementar.

Outro trabalho futuro que se aponta, que poderá aumentar a adesão á utilização do sistema, é a disponibilização para o utilizador final de uma ferramenta para criação de bases de conhecimento e permitir que estes contribuam para o crescimento do número de bases de conhecimento disponíveis.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] S. Russell e P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd edition, Prentice Hall, 2009.
- [2] F. Hayes-Roth, D. Waterman e D. B. Lenat, *Building Expert Systems*, Addison-Wesley, 1983.
- [3] R. I. Levine, D. E. Drang e B. Edelson, *Inteligência Artificial e Sistemas Especialistas*, São Paulo: McGraw-Hill, 1988.
- [4] R. I. Levine, D. E. Drang e B. Edelson, *Inteligência Artificial e Sistemas Especialistas*.
- [5] Android, “Android Security Overview,” [Online]. Available: <https://source.android.com/devices/tech/security/>. [Acedido em 16 Julho 2014].
- [6] N. Khan, A. Noraziah e E. I. Ismail, “Cloud Computing: Analysis of Various Platforms”.
- [7] Google Developers, “Google Cloud SQL - Admin and Reporting Tools,” [Online]. Available: <https://developers.google.com/cloud-sql/docs/admin-tools>. [Acedido em 16 Julho 2014].
- [8] Associação Paulista de Medicina, *Guia Médico da Família*, Nova Cultura.
- [9] J. Brooke, “SUS - A quick and dirty usability scale,” 1986.
- [10] R. M. Boucinha e L. M. Tarouco, “Avaliação de Ambiente Virtual de Aprendizagem com o uso do”.
- [11] R. Kurzweil, *The Age of Intelligent Machines*, The MIT Press, 1990.
- [12] R. J. Schalkoff, *Artificial Intelligence: An Engineering Approach*, New York: McGraw-Hill, 1990.
- [13] G. F. Luger , *Artificial Intelligence : Structures and Strategies for Complex Problem Solving*, Addison-Wesley, 2008.
- [14] A. Barr e E. A. Feigenbaum, *The Handbook of Artificial Intelligence*, Heuristech Press, 1981.
- [15] Open Handset Alliance, “Open Handset Alliance - Alliance Announcements,” [Online]. Available: http://www.openhandsetalliance.com/press_releases.html. [Acedido em 15 Julho 2014].
- [16] Android, “Android Devoper,” [Online]. Available: <https://developer.android.com/about/index.html>. [Acedido em 15 Julho 2014].
- [17] Android, “Android SDK - Download,” [Online]. Available:

- <http://developer.android.com/sdk/index.html>. [Acedido em 15 Julho 2014].
- [18] Google Developers, “Google Plugin for Eclipse,” [Online]. Available: <https://developers.google.com/eclipse/docs/download>. [Acedido em 16 Julho 2014].

ANEXOS

Anexo A - Artigo publicado na conferência CISTI2014

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.

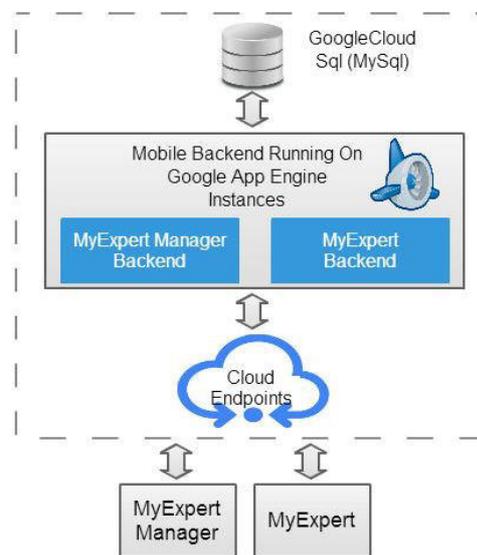


Figura 23. Arquitetura do Sistema

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.

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 selecionar 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.

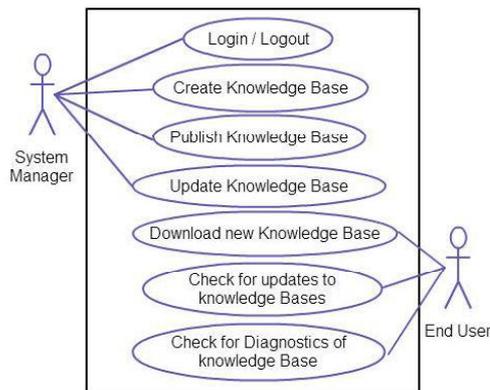


Figura 24. Atores do Sistema e respetivos casos de uso

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 selecionar 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.

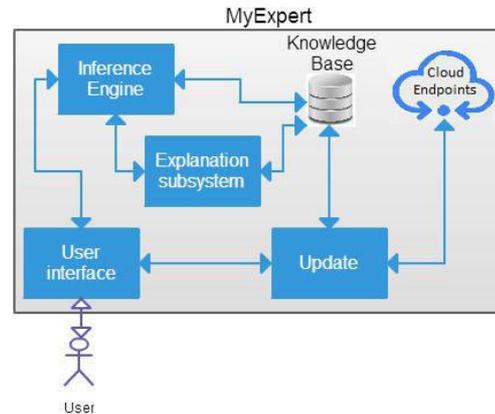


Figura 25. Módulos da aplicação MyExpert

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

$$SE C_i OpB C_{i+1} OpB C_{i+2} \dots \\ ENTÃO Conclusão$$

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.

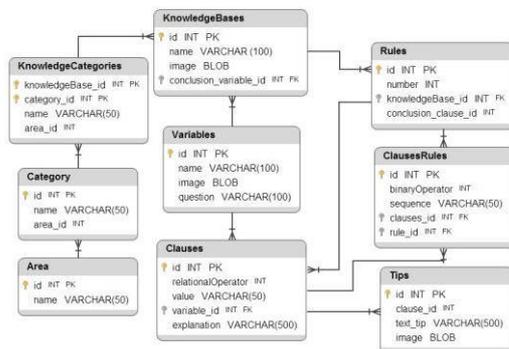


Figura 26. Diagrama Entidade-Relacionamento das bases de conhecimento

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 27. 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 atualização. 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 atualização 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.

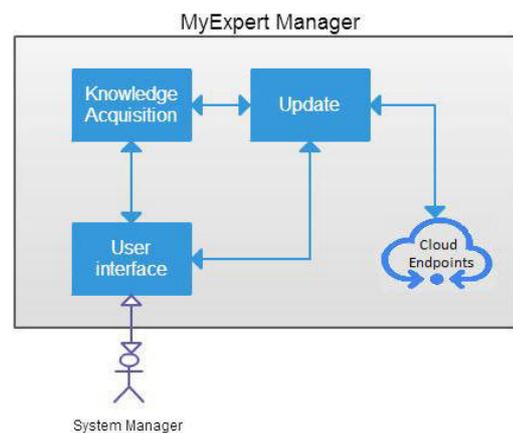


Figura 28. Módulos da aplicação MyExpert Manager

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.

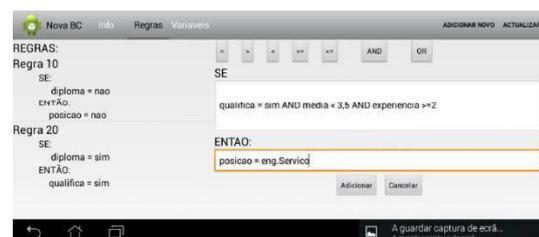


Figura 29. Ecrã exemplificativo da criação das regras de uma base de conhecimento

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 atualizações 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.

TABELA 18. 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

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.

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 a 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/whatisgoogleappengine>.
- [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.

Anexo B - Formulário para aplicação do questionário SUS

My Expert - System Usability Scale

Este formulário destina-se a avaliar a usabilidade da aplicação MyExpert, aplicação desenvolvida para obtenção do grau de Mestre em Computação Móvel da Escola Superior de Tecnologia e Gestão, do Instituto Politécnico da Guarda.

***Obrigatório**

1 -Eu gostaria de utilizar este sistema com frequência *

1 2 3 4 5

Discordo Totalmente Concordo Totalmente

2 -O sistema é desnecessariamente complexo *

1 2 3 4 5

Discordo Totalmente Concordo Totalmente

3 - O sistema é fácil de utilizar *

1 2 3 4 5

Discordo Totalmente Concordo Totalmente

4 - Preciso de ajuda para utilizar o sistema *

1 2 3 4 5

Discordo Totalmente Concordo Totalmente

5- As diversas funções deste sistema foram bem integradas *

1 2 3 4 5

Discordo Totalmente Concordo Totalmente

6 - Existem muitas inconsistências no sistema *

1 2 3 4 5

Discordo Totalmente Concordo Totalmente

7- Muitas pessoas poderiam utilizar o sistema rapidamente *

1 2 3 4 5

Discordo Totalmente Concordo Totalmente

8 - Achei o sistema muito complicado de utilizar *

1 2 3 4 5
Discordo Totalmente Concordo Totalmente

9 - Senti-me muito confiante com o sistema *

1 2 3 4 5
Discordo Totalmente Concordo Totalmente

10 - Necessitei de aprender muitas coisas antes de poder utilizar o sistema *

1 2 3 4 5
Discordo Totalmente Concordo Totalmente

Enviar

Nunca envie palavras-passe através dos Formulários do Google.
Tecnologia do [Documentos do Google](#)

Anexo C - Funcionamento da aplicação MyExpert

Para aceder a este módulo é necessário login com uma conta de email do Gmail, tal como pode ser visto na Figura 30 e apenas é permitido o acesso ás contas de email configuradas na aplicação de backend, pois é nesta que o login é confirmado.

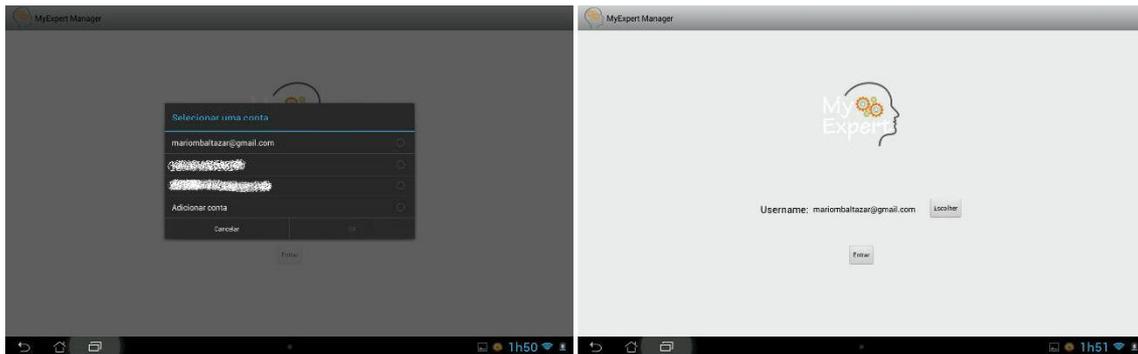


Figura 30 - Login na aplicação MyExper Manager

Após fazer login o administrador tem acesso a um ecrã onde poderá visualizar todas as bases de conhecimento, como se pode ver na Figura 31.

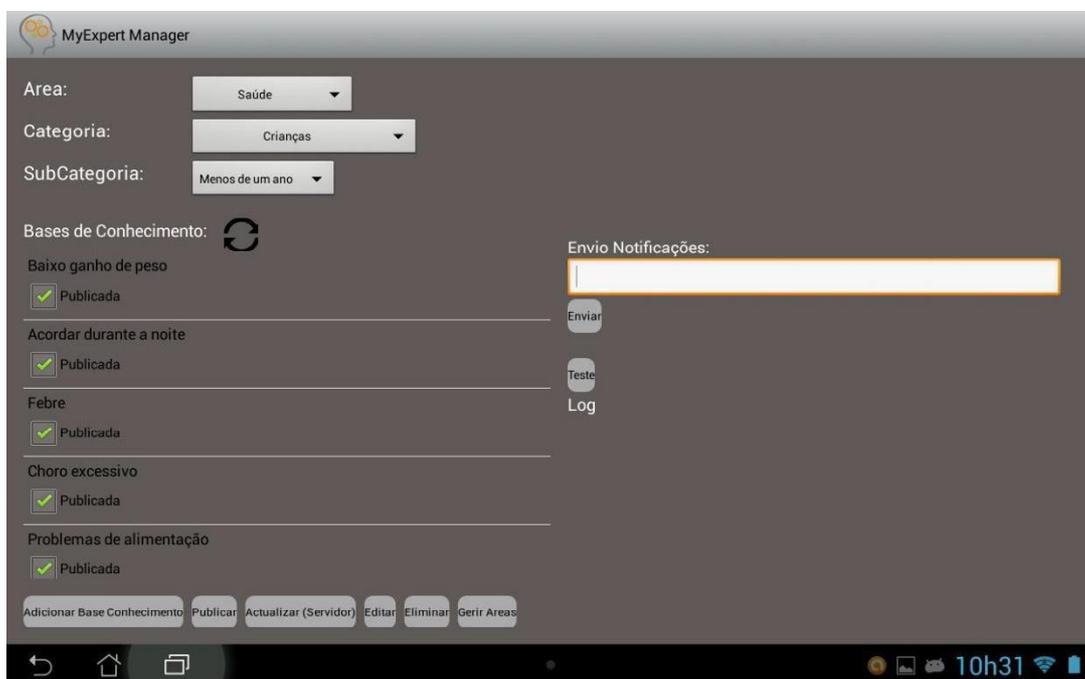


Figura 31 - Home da aplicação MyExpert Manager

Para o administrador adicionar uma nova Base de conhecimento este terá que clicar no botão que está no canto inferior esquerdo, e de seguida indicar qual o tipo de Base de Conhecimento que pretende adicionar (Figura 32).

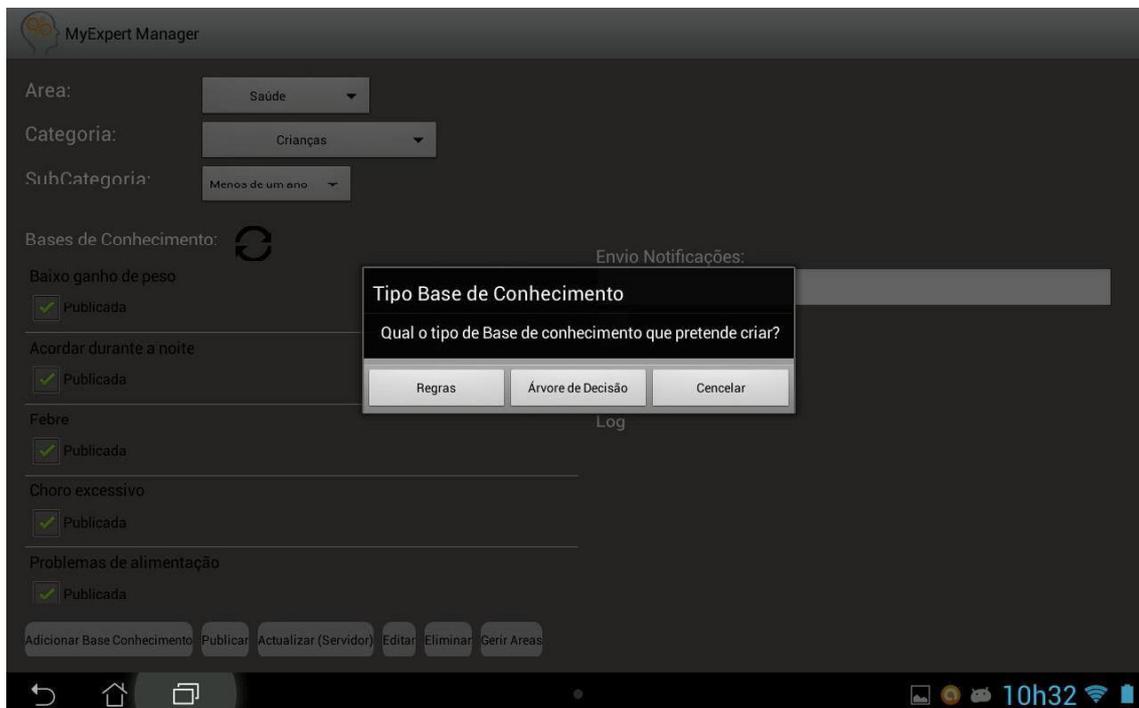


Figura 32 - Escolha do tipo de Base de Conhecimento a adicionar

Para o caso de se escolher adicionar uma base de conhecimento baseada em regras a primeira tarefa é indicar qual o nome da base de conhecimento e de seguida escolher qual a subcategoria a que irá pertencer a nova base de conhecimento (Figura 33).

The screenshot shows a mobile application interface for managing a knowledge base. At the top, there are tabs for 'Nova BC', 'Info', 'Regras', and 'Variáveis'. On the right side of the header, there are buttons for 'ADICIONAR NOVO' and 'ACTUALIZAR'. The main form area includes a 'Nome:' text input field, an 'Imagem:' section with an 'Escolher' button and a green Android robot icon, a checked 'Publicada' checkbox, an 'Area:' dropdown menu with the text '- Escolha uma Area -', and two more dropdown menus for 'Categoria:' and 'SubCategoria:'. At the bottom left, there is a 'Logs:' label.

Figura 33 - Adicionar informação da Base de conhecimento

O próximo passo será criar as regras que irão constituir a base de conhecimentos. Para isso é necessário escrever as regras no lado esquerdo do ecrã e clicar em adicionar (Figura 34).

The screenshot shows the 'Regras' tab in the 'Nova BC' application. On the left, under 'REGRAS:', there is an example rule: 'Regra 10' with 'SE: diploma = sim AND descoberta = sim' and 'ENTÃO: posicao = pesquisa'. On the right, there is a rule editor with a toolbar containing logical operators: '=', '>', '<', '>=', '<=', 'AND', and 'OR'. Below the toolbar, there are two text input fields: one for the condition (labeled 'SE:') and one for the consequence (labeled 'ENTÃO:'). At the bottom right of the editor, there are 'Adicionar' and 'Cancelar' buttons. The bottom of the screen shows a mobile OS navigation bar with icons for back, home, and recent apps, along with the time '10h54' and battery status.

Figura 34 - Criação de regras de uma Base de Conhecimentos

Cada da regra tem que ter o formato exemplificado na imagem, assim como a conclusão:

VARIAVEL = CONDIÇÃO AND/OR VARIAVEL = CONDIÇÃO

Após adicionar uma regra esta passa para o lado esquerdo do ecrã ficando no formato em que foi introduzida.

No final de introduzidas todas as regras é necessário passar para o separador regras (Figura 36), atualizar a lista das regras e para cada regra indicar qual a pergunta que será necessário fazer ao utilizador para poder instanciar a variável e a imagem ilustrativa da pergunta. Além disso será necessário indicar qual é a variável de conclusão.

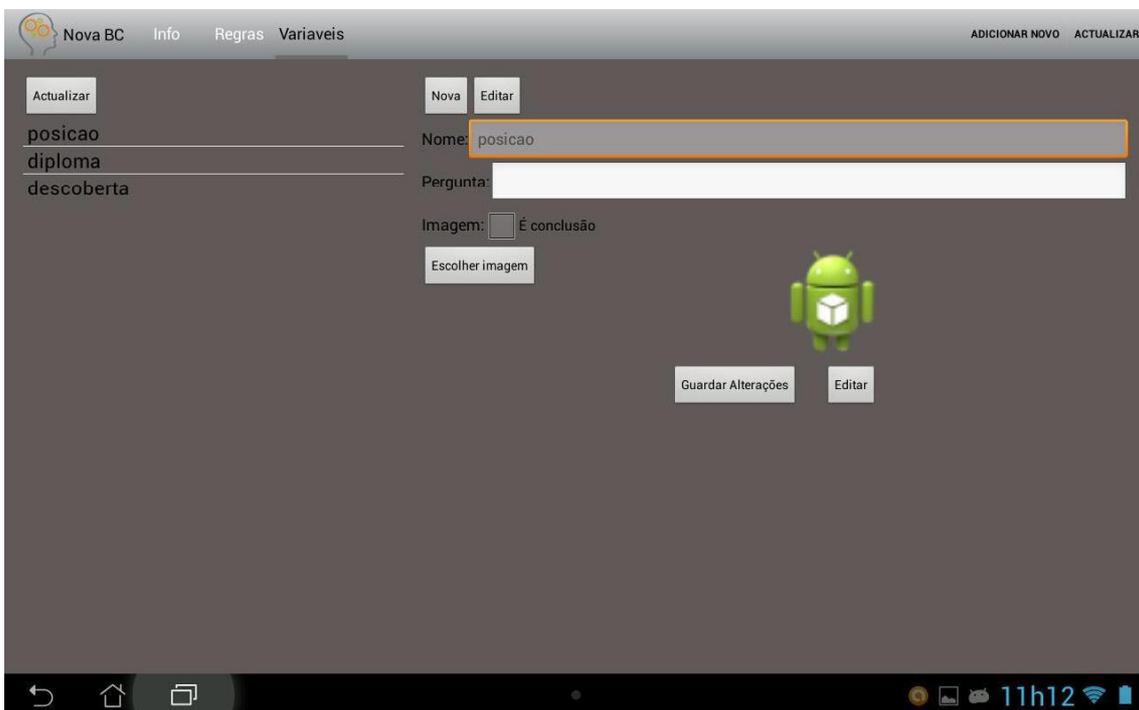


Figura 36 -Descrição das variáveis das bases de conhecimento

No caso de o administrador querer adicionar uma base de conhecimento baseada em árvores de decisão, o primeiro passo é igual ao anterior, ou seja é necessário indicar qual o nome, imagem e subcategoria da base de conhecimento.

No passo seguinte é onde vamos adicionar os nós que constituem a base de conhecimento. Para cada nó é necessário indicar qual o número do nó, pergunta a fazer ao utilizador, imagem, e indicar se o nó é um nó Raiz ou um nó folha, tal como pode ser visto na Figura 37.



The screenshot shows a mobile application interface titled "Nova BC" with tabs for "Info" and "Nodes". In the top right corner, there are buttons for "ADICIONAR NOVO" and "ACTUALIZAR". On the left side, there are two checked checkboxes labeled "NODES" and "LEAFS". The main form area contains the following fields and controls:

- "Number:" followed by a text input field.
- "Description:" followed by a larger text input area.
- "Nº YES Node:" followed by a text input field.
- "Nº No Node:" followed by a text input field.
- "Leaf:" with an unchecked checkbox and "Root:" with an unchecked checkbox.
- "Image:" followed by an "Escolher" button and a green Android robot icon.
- At the bottom of the form area are two buttons: "Adicionar" and "Cancelar".

The bottom of the screen shows a standard Android navigation bar with back, home, and recent apps icons, and a status bar displaying the time as 11h49.

Figura 37 - Criação dos nós que compõem uma base de conhecimentos

No final de criada a base de conhecimento apenas é necessário clicar no botão que está no canto superior direito e, através da aplicação de backend, a base de conhecimento será adicionada na base de dados alojada na Google Cloud SQL.

Anexo D - Funcionamento da aplicação MyExpert Manager

A Figura 38 representa o primeiro ecrã da aplicação. A primeira opção é onde podemos consultar as bases de conhecimento disponíveis e aquelas que já se fez download. Em definições é onde podemos definir se queremos receber notificações de novas bases de conhecimento e se as atualizações são feitas de forma automática ou manual. Na opção aviso é onde é alertado o utilizador para a não responsabilização pelos resultados das conclusões das bases de conhecimento.



Figura 38 - Ecrã inicial da aplicação

Clicando na opção “Bases de Conhecimento” o ecrã seguinte lista ao utilizador as diferentes áreas que têm bases de conhecimento (Figura 39 a)). Escolhendo uma área é listado as Categorias (Figura 39 b)). É aqui que é possível fazer o download das categorias que o utilizador desejar utilizar, como é mostrado na Figura 39 c)).

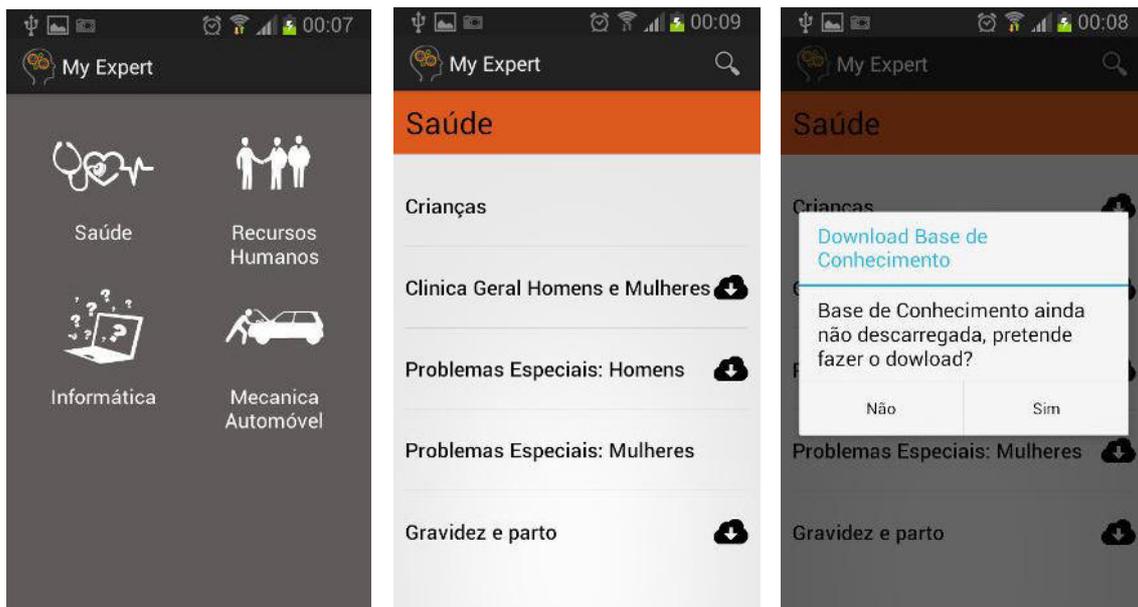


Figura 39 - a) Lista de áreas de Bases de conhecimento b) Lista de Categorias de Bases de Conhecimento c) Download de uma Categoria

Para as Categorias que é feito o download é possível ver as Subcategorias (Figura 40 a)), e escolhendo uma subcategoria é possível ver as Bases de conhecimento (Figura 40 b)).

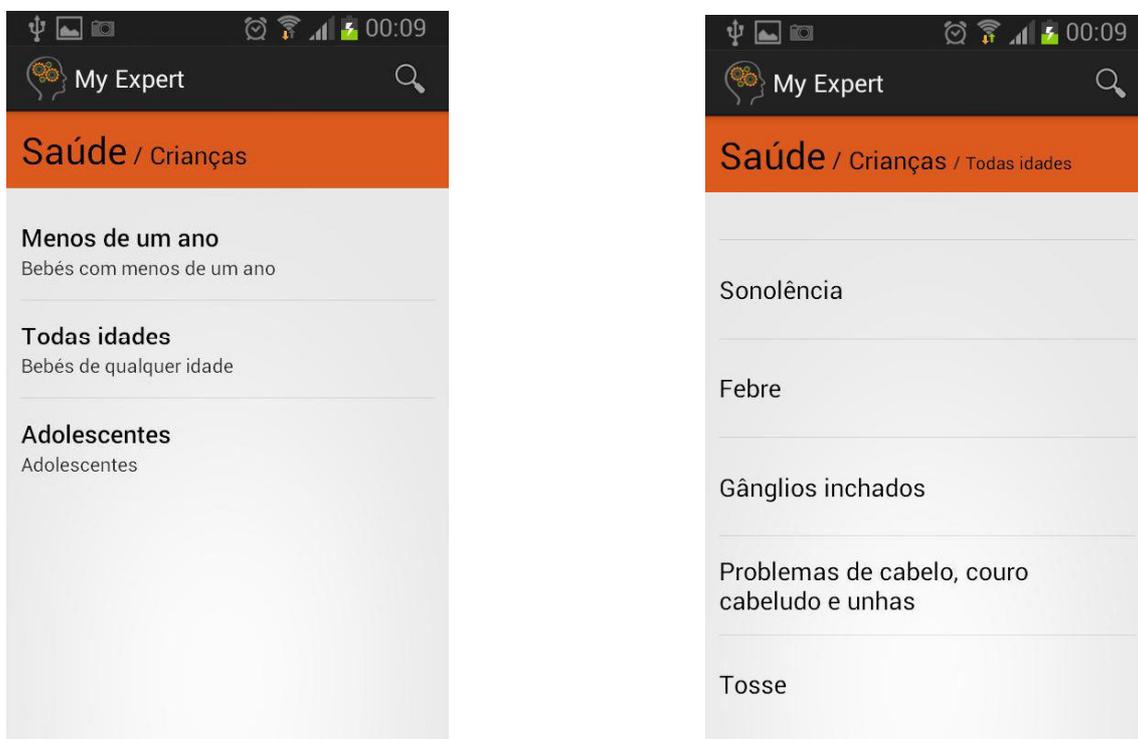


Figura 40 - a) Subcategorias de Bases de Conhecimento b) Bases de Conhecimento

Nas figuras seguintes são apresentados dois ecrãs exemplificativos da interação entre o utilizador e a máquina de inferência. A Figura 41 a) corresponde a uma questão colocada ao utilizador à qual deve responder com Sim ou Não, para continuar o processo de inferência. A Figura 41 b) 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.

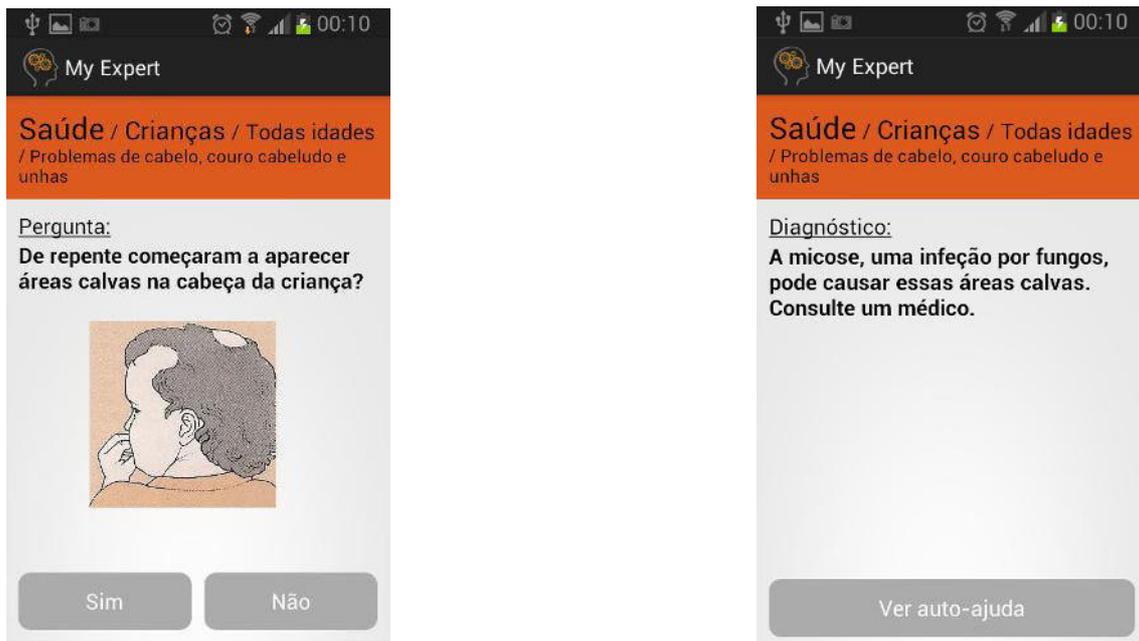


Figura 41 – a) Exemplo de uma questão colocada ao utilizador b) Exemplo de uma conclusão encontrada