

UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA - CCET

Uma Aplicação Colaborativa de Educação em Saúde para Ambientes  
Computacionais Móveis

Thiago Silva Mendes

São Luís  
2013

Thiago Silva Mendes

Uma Aplicação Colaborativa de Educação em Saúde para Ambientes  
Computacionais Móveis

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte de requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Francisco José da Silva e Silva

São Luís

2013

Mendes, Thiago Silva.

Uma aplicação colaborativa de educação em saúde para ambientes computacionais móveis/ Thiago Silva Mendes. – São Luís, 2013.

61f.

Impresso por computador (fotocópia).

Orientador: Francisco José da Silva e Silva.

Monografia (Graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2013.

1. Computação móvel – redes sociais. 2. Saúde. I Título.

CDU 004.057.6: 316.3

Thiago Silva Mendes

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte de requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

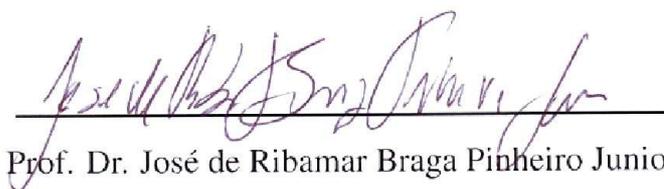
Orientador: Prof. Dr. Francisco José da Silva e Silva

Aprovada em: 26/07/2013

BANCA EXAMINADORA:



Prof. Dr. Francisco José da Silva e Silva  
Universidade Federal do Maranhão



Prof. Dr. José de Ribamar Braga Pinheiro Junior  
Universidade Federal do Maranhão



Prof. Dr. Carlos de Salles Soares Neto  
Universidade Federal do Maranhão

## **Resumo**

Uma rede social móvel é uma estrutura social em que seus membros se relacionam em grupos e a interação entre eles é feita através de tecnologias de informação e comunicação com os dispositivos de computação portáteis com acesso à tecnologias sem fio. Uma área promissora para a aplicação das redes sociais móveis é a saúde, permitindo promover a troca de informações, colaboração e integração entre pacientes e profissionais de pontos de atendimento à saúde.

Rede social móvel na área da saúde compreende o uso destas tecnologias com o propósito de conduzir coletivamente ações relacionadas à assistência médica e educação, independente do espaço. Este trabalho de monografia tem por objetivo o desenvolvimento de uma aplicação móvel para educação de pacientes quanto a manipulação de aparelhos ou uso de medicamentos, e de profissionais da saúde a partir da colaboração de informações entre os mesmos, que acontece através do compartilhamento, comentários e avaliações de mídias educacionais. Esta aplicação faz uso do middleware MobileHealthNet, que permite a criação de aplicações para dispositivos móveis utilizando serviços que foram desenvolvidos para área da saúde, como serviços de conteúdos, contexto, usuários e grupos, dentre outros.

**Palavras-chave:** computação móvel, redes sociais, saúde.

## **Abstract**

A mobile social network is a social structure in which members relate in groups and their interaction is done through information and communication technologies with portable computing devices with access to wireless technologies. A promising area for the application of mobile social networking is health, allowing promote the exchange of information, collaboration and integration between patients and professional points of health care.

Mobile social networking in healthcare includes the use of these technologies for the purpose of conducting collective actions related to health care and education, independent of space. This thesis work aims at developing a mobile application for patient education regarding the handling of equipment or medication use, and health professionals from the collaboration of information between them, which happens through sharing, comments and reviews of educational media. This application makes use of MobileHealthNet middleware, which enables the creation of applications for mobile devices using services that were developed for the health sector, as content services, context, users and groups, among others.

**Keywords:** mobile computing, social networking, health.

# *Sumário*

## **Lista de Figuras**

<b>1</b>	<b>Introdução</b>	p. 11
1.1	MobileHealthNet . . . . .	p. 12
1.2	Objetivos . . . . .	p. 13
1.3	Organização do trabalho . . . . .	p. 13
<b>2</b>	<b>Fundamentação Teórica</b>	p. 14
2.1	Redes Sociais na Área da Saúde . . . . .	p. 15
2.2	O Middleware MobileHealthNet . . . . .	p. 16
2.2.1	Communication Layer . . . . .	p. 17
2.2.2	Core Services . . . . .	p. 18
2.2.3	Security Services . . . . .	p. 19
2.2.4	Application Services . . . . .	p. 19
<b>3</b>	<b>Programação com Google Android e o Middleware MobileHealthNet</b>	p. 21
3.1	Google Android API . . . . .	p. 21
3.1.1	Arquitetura do Google Android . . . . .	p. 22
3.1.2	Classe Activity . . . . .	p. 23
3.1.3	Classe Intent . . . . .	p. 25
3.1.4	Classe View . . . . .	p. 26
3.1.5	Classe Adapter . . . . .	p. 26
3.1.6	Android.R . . . . .	p. 28

3.1.7	AndroidManifest.xml . . . . .	p. 28
3.1.8	Classe Dialog . . . . .	p. 29
3.2	Programando com o Middleware MobileHealthNet . . . . .	p. 29
3.2.1	MOBHAAuthentication . . . . .	p. 29
3.2.2	MOBHAContent . . . . .	p. 30
3.2.3	MOBHAContext . . . . .	p. 31
3.2.4	MOBHAAUser . . . . .	p. 32
<b>4</b>	<b>Aplicação Colaborativa na Área da Educação Médica</b>	<b>p. 34</b>
4.1	Requisitos . . . . .	p. 36
4.2	Funcionalidades e Layout de Telas . . . . .	p. 36
4.2.1	Tela de abertura e login . . . . .	p. 37
4.2.2	Classificação das mídias . . . . .	p. 37
4.2.3	Avaliação das mídias pelos usuários . . . . .	p. 38
4.2.4	Buscando mídias . . . . .	p. 40
4.2.5	Mídias Recentes . . . . .	p. 40
4.3	Aspectos de Implementação . . . . .	p. 41
4.3.1	Organização das telas e conteúdos . . . . .	p. 41
4.3.2	Diagrama de classes . . . . .	p. 43
4.3.3	Diagrama da classe Media . . . . .	p. 46
4.3.4	Relação entre a classe Comment e CommentListAdapter . . . . .	p. 46
4.3.5	Diagrama das classes MediaListActivity, ListAdapter e MediaViewActivity . . . . .	p. 48
4.3.6	Utilizando o MOBHAAuthentication . . . . .	p. 50
4.3.7	Utilizando o MOBHAAUser para obter os dados dos pacientes ou dos profissionais da saúde . . . . .	p. 51
4.3.8	Utilizando o MOBHAContent para obter conteúdos educacionais . . . . .	p. 51

<b>5 Considerações finais</b>	p. 56
5.1 Trabalhos futuros . . . . .	p. 57
<b>Referências Bibliográficas</b>	p. 58

# *Lista de Figuras*

2.1	Representação de uma rede social móvel. Fonte:[Teles et al. 2013]. . . . .	p. 14
2.2	Arquitetura Geral do MobileHealthNet. Fonte: [Teles et al. 2012] . . . . .	p. 17
3.1	Arquitetura do sistema Android. Fonte: [Google 2013] . . . . .	p. 23
3.2	Ciclo de vida de uma Activity (Fonte: Google, 2013) . . . . .	p. 24
3.3	Métodos correspondentes aos estados de uma Activity (Fonte: Google, 2013)	p. 25
3.4	Exemplo do uso da classe Intent . . . . .	p. 25
3.5	Declaração de uma classe serializável . . . . .	p. 26
3.6	Exemplo de uma interface Android em xml (Google, 2013) . . . . .	p. 27
3.7	Exemplo de uma interface xml vista pelo usuário . . . . .	p. 27
3.8	Exemplo de um Adapter . . . . .	p. 28
3.9	Exemplo Dialog (Fonte: marakana.com) . . . . .	p. 29
3.10	Exemplo de interface MOBHA com o MOBHAAuthentication . . . . .	p. 30
3.11	Tópico UserAdd . . . . .	p. 32
3.12	Exemplo de listener para ouvir a chegada dos tópicos no lado do cliente . . .	p. 33
4.1	Imagem da tela de login . . . . .	p. 37
4.2	Imagens das telas Categorias e Avaliar . . . . .	p. 38
4.3	Imagem da tela “Mais Votados” . . . . .	p. 39
4.4	Imagens das telas Buscar e Mídias Recentes . . . . .	p. 39
4.5	Imagens das telas Buscar e Mídias Recentes . . . . .	p. 40
4.6	Construtor da classe Media . . . . .	p. 42
4.7	Método writeByteToFile() . . . . .	p. 43
4.8	Permissão para escrever no cartão de memória . . . . .	p. 43

4.9	Diagrama de classes de visualização geral . . . . .	p. 44
4.10	Diagrama da classe Media . . . . .	p. 47
4.11	Diagramas das classe Comment e CommentListAdapter . . . . .	p. 48
4.12	Diagramas das classe MediaListActivity e ListAdapter . . . . .	p. 49
4.13	Diagrama da classe MediaViewActivity . . . . .	p. 49
4.14	Estrutura do tópico Login . . . . .	p. 50
4.15	Informações trafegadas através do tópico User . . . . .	p. 52
4.16	Exemplo de listener para ouvir a chegada dos tópicos no lado do cliente . . .	p. 53
4.17	Tópico ContentDownloadResponse . . . . .	p. 53
4.18	Tópico ContentInfoByIdResponse . . . . .	p. 54
4.19	Tópico ContentMetadata . . . . .	p. 55

# *1 Introdução*

Nos últimos anos, as recentes mudanças nas tecnologias de informação e comunicação (TIC) se difundiram em meio a sociedade influenciando desde os aspectos de produção de uma indústria até a forma em como as pessoas pudessem se comunicar, afetando diretamente na maneira como se relacionam.

O crescimento demasiado da Internet, por exemplo, proporcionou o surgimento de ferramentas capazes de oferecer às pessoas a possibilidade de publicarem conteúdos, tais como fotos e vídeos, bem como de exporem opiniões em torno de qualquer assunto na rede para serem compartilhadas entre diversos usuários.

Estas ferramentas são chamadas de Mídias Sociais e podem ser definidas, segundo Kaplan e Haenlein (2010), como um grupo de aplicações baseadas na Internet construídas sobre os fundamentos da Web 2.0, que permite a criação e troca de conteúdo gerado pelo usuário. Desta forma, mídias sociais são centradas no usuário e seu objetivo principal é permitir aos seus usuários uma interação e colaboração entre si como criadores de conteúdos dentro da comunidade virtual estabelecida a partir da mídia social. Blogs, Youtube e Twitter são claros exemplos do que são essas ferramentas.

Um dos tipos de mídia social com maior popularidade é a rede social, uma estrutura social cujos membros se relacionam em grupos [Lin and Kede 2011] e a interação é realizada através de tecnologias da informação e comunicação. Esta interação permite a quebra de barreiras geográficas e temporais devido ao acesso e interações simultâneas entre pessoas permitindo, por exemplo, o contato direto com familiares e amigos localizados em qualquer lugar do mundo. Redes sociais podem se focar no aprimoramento de idiomas, na troca de experiência acadêmica, na música, no estabelecimento de relacionamentos profissionais, entre outros interesses. Assim, segundo Weider e Asim (2009), uma rede social pode ser considerada como um meio de comunicação em uma estrutura social definida.

Acompanhando a evolução dos dispositivos de computação portáteis com acesso a tecnologias de comunicação sem fio (rede celular, rede local sem fio, Bluetooth, etc), as redes sociais estão cada vez mais presentes nesses tipos de dispositivos, aumentando a capacidade dos usuários de se comunicarem com as pessoas de seus interesses a qualquer hora do dia e em qualquer lugar [Rana et al. 2009]. Estas redes sociais são chamadas de Redes Sociais Móveis (RSMs)

e permitem explorar uma rica variedade de informações de contexto, tais como a localização corrente do usuário, o clima e temperatura local, as pessoas na vizinhança, o estado do usuário (como sua disponibilidade e humor), a ação por ele sendo realizada e sua intenção em realizá-la.

Uma área fértil para a aplicação dos conceitos relacionados as redes sociais é a saúde. Segundo Varshney (2009), e-health é definido como a aplicação das tecnologias de informação e comunicação através de todo escopo das funções envolvidas na prática e entrega de cuidados médicos. Neste contexto, RSMs podem ser utilizadas para promover o intercâmbio de informações, colaboração e integração social entre os diversos agentes envolvidos no processo de atendimento à saúde [Jain 2011].

## 1.1 MobileHealthNet

O MobileHealthNet é um projeto desenvolvido em parceria pelo Laboratório de Sistemas Distribuídos da UFMA e o *Laboratory for Advanced Collaboration* da PUC-Rio, tendo por objetivo geral avançar o estado da arte em sistemas de *middleware* para RSMs, utilizando-o para a criação de novos serviços e aplicações de RSMs voltados para a área da saúde. Este projeto conta com apoio institucional do Hospital Universitário da UFMA (HUUFMA). Em particular, duas unidades do HUUFMA estão diretamente envolvidas com o desenvolvimento do projeto: o Programa de Assistência a Pacientes Asmáticos (PAPA) e a Casa da Dor. O PAPA possui profissionais com especialidade em pneumologia e tem como objetivo principal o tratamento de pacientes portadores desta doença crônica. A Casa da Dor, por sua vez, possui profissionais especializados no tratamento de pacientes que sofrem com dores agudas, independentemente de sua etiologia [Teles et al. 2012].

O projeto MobileHealthNet foi concebido para ser aplicado em especial a comunidades carentes e distantes. Nesse contexto, o termo “distante” refere-se não somente a distância física entre a comunidade e os centros de atendimento a saúde, mas também a dificuldade em realizar o deslocamento entre eles, devido a precariedade do acesso ou a indisponibilidade de meios de transporte adequados.

Este trabalho de monografia está inserido no contexto do projeto MobileHealthNet, visando a utilização dos seus principais serviços em uma aplicação colaborativa para dispositivos móveis que foi desenvolvida, como proposta deste trabalho, de acordo com os objetivos apresentados na seção a seguir.

## 1.2 Objetivos

Esta monografia visa contribuir com o projeto “MobileHealthNet - Redes Sociais Móveis Voltadas ao Atendimento à Saúde”, por meio da especificação de requisitos, projeto e implementação de uma aplicação de redes sociais móveis voltada ao domínio da saúde. Os objetivos específicos que devem ser seguidos para a consolidação deste trabalho são os seguintes:

- Análise do estado da arte sobre as redes sociais móveis;
- Estudo da plataforma Google Android e do middleware MobileHealthNet, que serão utilizadas como base para o desenvolvimento da aplicação proposta neste trabalho;
- Elicitação dos requisitos da aplicação a ser desenvolvida, em uníssono com os profissionais da saúde do Hospital Universitário Presidente Dutra;
- Implementação e testes funcionais da aplicação;
- Avaliação da aplicação desenvolvida.

## 1.3 Organização do trabalho

Este documento está organizado da seguinte forma: o Capítulo 2 trata dos fundamentos de redes sociais móveis, descreve o uso de redes sociais móveis na área da saúde e o middleware MobileHealthNet, mostrando sua arquitetura e as funcionalidades disponibilizadas por cada serviço; O Capítulo 3 mostra uma noção de básica de implementação com o Google Android API e o middleware MobileHealthNet; o Capítulo 4 aborda os aspectos de implementação da aplicação colaborativa educacional, envolvendo o seu desenvolvimento sobre a plataforma Android e a sua integração ao *middleware* MobileHealthNet, visando a utilização dos seus serviços, tais como o de conteúdo, usuários e outros; Neste mesmo capítulo são mostrados os requisitos funcionais, não- funcionais, e as funcionalidades da aplicação desenvolvida; O Capítulo 5 relata as considerações finais sobre o trabalho.

## 2 *Fundamentação Teórica*

Redes sociais móveis são formadas por três áreas, sendo elas as redes sociais, computação móvel e sistemas de ciência de contexto [Batista 2012]. A união destas três áreas é mostrada na figura 2.1. Um ambiente de uma rede social permite aos seus membros estabelecerem vínculos virtuais que podem transpassar um sistema de computação para o mundo real. Ou o inverso, em que pessoas que possuam estreitas relações pessoais comecem a utilizar as redes sociais para manterem contato, por exemplo, na situação de passarem a morar em localidades distintas. A computação móvel, por sua vez, estabelece ao usuário uma maneira de realizar suas atividades a qualquer hora e em qualquer lugar e, em muitos casos, de forma *online*. E, por fim, os sistemas de ciência de contexto permite ao usuário adquirir e compartilhar informações de contexto, tais como temperatura local, localização geográfica, entre outros.

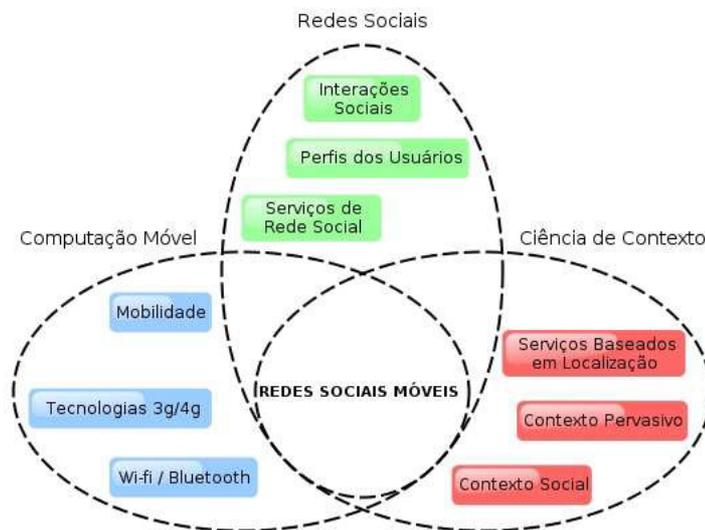


Figura 2.1: Representação de uma rede social móvel. Fonte:[Teles et al. 2013].

O desenvolvimento de aplicações para ambientes virtuais móveis traz em si muitos desafios, devido as limitações que os dispositivos móveis possuem, tais como baixa disponibilidade de processamento, memória, entre outros. Além disso, quando a aplicação necessita de uma conexão e esta pode envolver uma rede sem fio, toda a operação a ser realizada para o usuário pode ser afetada por que este tipo de conexão está mais propensa a problemas de desconexão, conectividade intermitente e flutuações de sua condição operacional, como largura de banda e

taxa de erros de comunicação [Batista 2012].

Para que ocorra uma total interação entre pessoas com diferentes dispositivos utilizando diferentes tecnologias, diversos middlewares são desenvolvidos a fim de facilitar a implementação de aplicações. Estes sistemas promovem um conjunto de ferramentas que facilitam o desenvolvimento das aplicações através do uso de paradigmas de programação e escondendo, por exemplo, a complexidade proveniente de componentes internos [Adelstein et al. 2005]. Para a realização deste trabalho de monografia, foi escolhido o middleware MobileHealthNet, que será apresentado na seção a seguir.

## 2.1 Redes Sociais na Área da Saúde

O uso mais intensivo das redes sociais aconteceu a partir do surgimento da Web 2.0, em que o compartilhamento de informações midiáticas, como áudio, vídeo, texto, animação ou imagem, passaram a ser realizadas entre os mais diversos tipos de usuários de maneira mais interativa.

Pesquisas recentes em torno da Web 2.0 também auxiliaram em grandes avanços, orientando a rede a ter suporte para informações de fenômenos físicos que ocorrem no organismo do usuário, tais como taxa de batimentos cardíacos, pressão sanguínea ou nível de glicose. Além disso, fenômenos que ocorrem no ambiente em que o indivíduo está localizado, também é abstraído como informação contundente: umidade, temperatura e localização, são grandes exemplos.

Obter informações como batimentos cardíacos ou a pressão sanguínea, por exemplo, no caso do setor da saúde, pode auxiliar os médicos no monitoramento de pacientes que estão sendo submetidos a um tratamento específico, permitindo avaliar com maior precisão o estado de saúde dos mesmos. Além disso, estas informações podem ser compartilhadas entre outros profissionais da saúde envolvidos no tratamento do paciente, favorecendo uma avaliação coletiva dos dados obtidos. Neste sentido, uma Rede Social na área da saúde pode ser definida como um grupo de pessoas (e a estrutura social que elas coletivamente constroem) que utilizam tecnologias da informação e comunicação com o propósito de conduzir coletivamente ações relacionadas à assistência médica e sua educação [Demiris 2006].

Redes sociais na área da saúde podem haver uma combinação dos diversos agentes envolvidos no processo de atenção à saúde, incluindo profissionais da saúde (médicos, enfermeiros, fisioterapeutas, terapeutas ocupacionais, etc), pesquisadores da saúde (professores e alunos de graduação e pós-graduação), pacientes e seus familiares, bem como membros da comunidade em geral. Pesquisadores da área da saúde, por exemplo, podem formar uma rede social para compartilharem informações importantes, que vão desde opiniões sobre um caso específico, até estudos acadêmicos que podem ser escritos colaborativamente por várias pessoas envolvidas na

pesquisa.

Redes sociais que envolvam pacientes e seus familiares podem ajudar indivíduos diagnosticados com a mesma condição médica e que estejam realizando o mesmo tratamento. Nesta situação, familiares de pacientes que estão submetidos a um tratamento específico podem estabelecer uma relação com familiares de outros pacientes que estão sob o mesmo tratamento, permitindo trocar experiências e opiniões a respeito do tratamento em questão.

Entre os profissionais da saúde e pacientes, as redes sociais podem ser utilizadas em aplicações que facilitem o monitoramento de doenças através, por exemplo, de sensores que transmitam as informações do estado de saúde do paciente para os profissionais da saúde e que possibilitem alternativas de comunicação e interação entre estes mesmos profissionais e os seus pacientes, permitindo que estabeleçam contato além de encontros presenciais [Teles et al. 2012].

Para os profissionais da saúde, as redes sociais favorecem o compartilhamento de informações e melhorias na coordenação entre os integrantes de equipes de atenção básica e os profissionais dos demais níveis de atendimento à saúde .

A motivação para o uso da computação móvel em conjunto com as redes sociais voltadas para área da saúde se encontra na necessidade dos profissionais da área estarem em constante locomoção de um ponto a outro dentro do próprio hospital, ou no caso de trabalharem em mais de um lugar, como acontece com os agentes de saúde responsáveis pela atenção básica, do Programa Saúde da Família [da Saúde 2006].

Por outro lado, o uso dos dispositivos móveis também favorece a comunicação entre o paciente e seus familiares, independente da localização física, já que àqueles também possuem a necessidade de se deslocarem ao longo do seu tratamento, para a realização de exames, consultas ou procedimentos. Além disso, o uso de dispositivos e sensores móveis permite o monitoramento contínuo como, batimento cardíaco, pressão arterial, taxa de glicemia, e outros. Estes dados podem ser disponibilizados em tempo real ao profissional da saúde, facilitando o acesso às informações relativas ao estado de saúde do paciente.

## 2.2 O Middleware MobileHealthNet

No âmbito do projeto “MobileHealthNet - Redes Sociais Móveis Voltadas ao Atendimento à Saúde” está sendo desenvolvido um middleware cujo principal objetivo é simplificar o desenvolvimento de aplicações de redes sociais móveis para a área da saúde.

Quatro aplicações foram previstas no projeto MobileHealthNet, descritas a seguir:

a) *HealthEducation*, que possui ferramentas para o compartilhamento de arquivos multimídia e seu objetivo é aprimorar a educação de pacientes e profissionais da saúde através de

mídias com conteúdo educacional;

b) *Professional Collaboration*, que visa diminuir a distância entre os profissionais da saúde, através de recursos como chat multiusuário, fórum de discussão e serviço de notificação que permita informar sobre a urgência de se obter o resultado de um exame, discutir a respeito do tratamento e acompanhamento de um determinado paciente;

c) *HUPD Care*, que tem por objetivo explorar os conceitos das RSMs para promover a assistência prestada por especialistas responsáveis pelo atendimento de alta complexidade a profissionais da atenção básica à saúde no atendimento a casos específicos;

d) *Patient-Buddy-Build*, que possibilita a criação de questionários simples e práticos a serem respondidos por pacientes, para informarem periodicamente o estado de sua doença. Dessa forma, mantém-se a interação médico-paciente além dos encontros presenciais (consultas), reduzindo-se a necessidade de visitas pós-diagnóstico e melhorando a efetividade no acompanhamento dos pacientes.

A arquitetura do MobileHealthNet é organizada em quatro camadas, ilustradas na Figura 2.2.

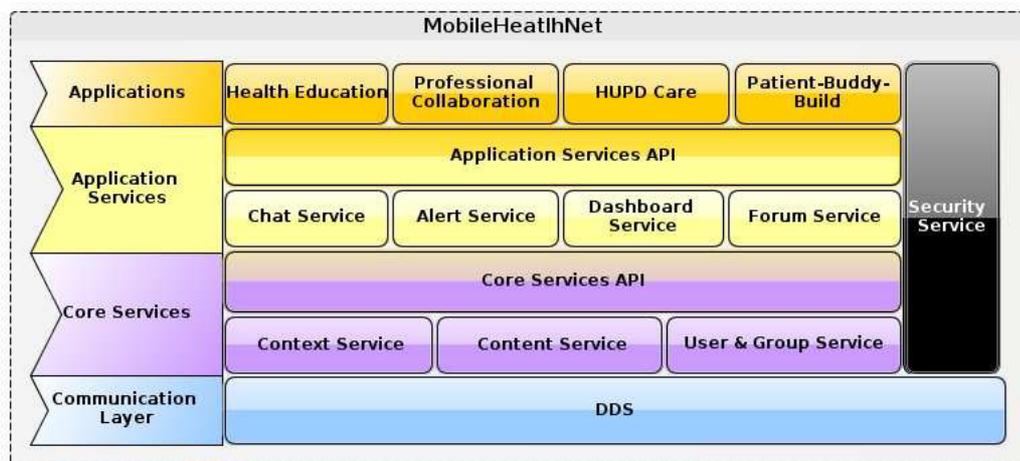


Figura 2.2: Arquitetura Geral do MobileHealthNet. Fonte: [Teles et al. 2012]

### 2.2.1 Communication Layer

A camada *Communication Layer* é baseada na especificação Data Distribution Service (DDS) da Object Management Group [Group 2001], um padrão para comunicação publish/subscribe com alta qualidade de serviço que visa à distribuição crítica de informações em sistemas distribuídos em tempo real. Nesta abordagem, existe um espaço global de dados para a troca de informações na rede, onde a geração de informação consiste em publicar (escrever) no espaço de dados enquanto que o consumo da informação (subscrição) se realiza lendo dados deste espaço [Vega 2008].

O DDS foi idealizado sobre um modelo Data-Centric Publish-Subscribe (DCPS) baseado em tópicos. Cada tópico representa um conjunto de informações acerca de uma entidade ou uma ação (solicitar informações sobre um usuário, efetuar seu login no sistema, entre outras) na rede social. Aplicações que necessitam compartilhar informações com outras aplicações pode utilizar o espaço global de dados informando suas intenções de publicar ou subscrever dados que são relacionados a um ou mais tópicos de interesses dos participantes [Batista 2012].

A escolha de se adotar o DDS como base para a comunicação no MobileHealthNet se deve ao fato dele ser implementado através do paradigma de interação publish/subscribe para enviar e receber informações em tempo real. Este paradigma é mais eficiente em ambientes móveis do que o request/response, pois não necessita de uma constante conexão com um servidor, uma característica útil em ambientes móveis cuja comunicação sem fio está sujeita a eventuais desconexões e conectividade intermitente.

No cliente móvel, um componente denominado MobileHealthNet on Android (MOBHA) provê a interface de comunicação entre as aplicações em execução no dispositivo e os serviços da rede social providos por um servidor.

O servidor e o cliente possuem uma camada em comum, o MobileHealthNet Framework. Esta camada é o centro da comunicação e contém o conjunto de tópicos, publicadores, subscritores e as políticas de QoS utilizados para efetuar a comunicação no middleware MobileHealthNet.

### 2.2.2 Core Services

A camada *Core Services* disponibiliza serviços básicos que são compartilhados pelos demais serviços e aplicações do MobileHealthNet. O Serviço de Contexto (Context Service) é o responsável pelo armazenamento e disponibilização de informações de contexto. Diversos tipos de contexto podem ser explorados em aplicações de RSMs para a área da saúde, como a informação de localização, por exemplo, que pode ser utilizada para determinar os membros da rede presentes na vizinhança com os quais o usuário costuma interagir.

O Serviço de Conteúdo (Content Service) tem como principal tarefa o compartilhamento de mídia (texto, fotos, áudio, filmes, etc) entre usuários da rede social. Este serviço permite rotular cada elemento de mídia com metainformações definidas pela aplicação como, por exemplo, a posição geográfica na qual a mídia foi obtida ou o modelo de uma bomba de asma ao qual um tutorial se refere [Batista 2012]. Diversas funcionalidades podem ser obtidas com esse serviço, como *upload* ou *download* de mídias, deletar um conteúdo, renomeá-lo, alterar as metainformações (metadados) associados, buscar as mídias pelo nome, por metadados ou pelo seu identificador na base de dados, e entre outros. Neste serviço o usuário é livre para associar metainformações no momento do *upload* da mídia, permitindo que outros usuários possam ter

acesso ao conteúdo mais facilmente e de acordo com o seu interesse.

O Serviço de Usuários e Grupos (User & Group Services) gerencia a criação de contas de usuários e grupos, bem como o gerenciamento de relacionamentos (amizades) entre usuários da rede. Este serviço permite que o usuário possa se autenticar utilizando uma conta cadastrada na base de dados do MobileHealthNet. Uma vez autenticado, o usuário poderá usufruir com uma única credencial todas as aplicações e todos os serviços disponibilizados pela rede social.

No *middleware* MobileHealthNet todos as informações do usuário são criptografadas para garantir a segurança destes tipos de dados trafegados na rede. As funcionalidades presentes nesse serviço é a adição de usuário, remoção, solicitação das informação de um usuário e edição de suas informações. Os usuários a serem cadastrados no MobileHealthNet precisam ser definidos como um paciente ou como um profissional da saúde, porque algumas das informações associadas a cada um deles no momento da inclusão do seus perfis no MobileHealthNet são diferentes.

### 2.2.3 Security Services

A camada *Security Services* é transversal a todas as camadas da arquitetura do middleware e disponibiliza os mecanismos que implementam o modelo de privacidade e segurança, necessários ao middleware MobileHealthNet.

A segurança e a privacidade dos dados são requisitos essenciais no contexto das redes sociais, uma vez que seus usuários estão constantemente publicando informações, muitas consideradas privadas e destinadas apenas a um usuário específico ou a um grupo da rede. Informações referentes às interações sociais podem também ser alvo de ataques e serem utilizadas para fins maliciosos [Gao et al. 2011]. No middleware MobileHealthNet, todas as operações devem ser realizadas por usuários autenticados, procurando garantir as informações relacionadas aos pacientes. É permitido também aos usuários acessarem todos os serviços e qualquer aplicação dentro do mesmo ambiente.

Além disso, deve-se prover mecanismos para o estabelecimento de canais seguros de comunicação no DDS, garantindo-se ainda as propriedades básicas de autenticidade, integridade e confidencialidade na distribuição dos dados.

### 2.2.4 Application Services

A camada *Application Services* disponibiliza serviços típicos presentes em redes sociais, tais como o serviço de chat (Chat Service), no qual duas ou mais pessoas podem se comunicar pela web; serviço de mural (Dashboard Service), onde se encontram as publicações mais recentes realizadas pelos usuários e que podem ser vistas por algumas pessoas (definidas por

---

quem publica) ou por todos os seus contatos; serviço de alerta (Alert Service), que realiza envios de notificações aos usuários em caso de, por exemplo, novas publicações criadas por seus contatos ou novas mensagens que chegam através de chat; e serviço de fórum (Forum Service), no qual permite aos usuários criarem tópicos para debaterem sobre um determinado assunto.

## 3 *Programação com Google Android e o Middleware MobileHealthNet*

Este capítulo abordará uma noção básica de desenvolvimento de aplicações com a Google Android API, bem o uso do componente MOBHA, desenvolvido para que clientes possam usar os serviços do middleware MobileHealthNet.

### 3.1 Google Android API

O sistema operacional Android é uma plataforma do Google voltada para dispositivos móveis. É mantida pela OHA (Open Handset Alliance), um grupo com mais de 30 empresas que se uniram com o propósito de inovar e acelerar o desenvolvimento de aplicações e serviços.

Atualmente grande parte dos dispositivos móveis possuem um dos quatro sistemas operacionais seguintes, como o *Symbian OS*, *Android*, *IOS* e o *Windows Mobile*. Todos eles são excelentes sistemas operacionais móveis, e permitem ao usuário uma boa experiência com acesso a suas contas de email, inúmeros aplicativos que podem ser baixados da rede e que ampliam ainda mais o uso do aparelho, agendamento de compromissos onde, em alguns casos, são compartilhados entre todos que estarão presentes em uma reunião, por exemplo, comunicação com outras pessoas ao redor do globo, não somente através da linha de telefone, mas pela utilização da rede de internet com a utilização de aplicações de chat, e entre outros benefícios.

O sistema operacional móvel *Android* destaca-se nesse sentido porque é uma plataforma que fornece um kit gratuito para implementação das aplicações dando maior liberdade e autonomia aos desenvolvedores, além de ser código-aberto, permitindo às grandes empresas de tecnologia móvel personalizarem o sistema de acordo com seus projetos. O sistema *Android* envolve as aplicações baseadas em serviços nas nuvens, como Google Calendar, Gmail, entre outros, em uma única conta da Google, permitindo sincronizar os dados do usuário no dispositivo.

A popularidade do *Android* cresceu muito rapidamente ao longo dos últimos anos, chegando a ter por dia 1 milhão e 300 mil aparelhos ativados com este sistema. Esta popularidade é devido às razões citadas anteriormente, como a liberdade e autonomia aos desenvolvedores de criarem

novas aplicações, isto fornece ao usuário maior flexibilidade em buscar aplicativos que atendam suas necessidades. O fato de ser nativamente integrado as redes sociais e aos principais serviços nas nuvens também favorece o alto índice de ativação deste sistema nos aparelhos.

Para o desenvolvimento de aplicações para o sistema Android, a empresa Google que gerencia todo o projeto, disponibiliza um pacote de desenvolvimento chamado Android SDK que possui um conjunto de ferramentas e APIs que favorecem a criação de aplicativos para o sistema Android utilizando a linguagem Java. Os aplicativos desenvolvidos são compilados em *bytecodes* e executados em uma Máquina Virtual Dalvik<sup>1</sup>, que possibilita rodar aplicações distribuídas em formato binário em qualquer dispositivo móvel independente do processador.

Devido as razões mencionadas, inclusive as de que existem aparelhos com este sistema móvel com um custo acessível, é que o sistema Android foi a plataforma escolhida para rodar a aplicação *HealthEducation*. O *middleware* MobileHealthNet tem uma melhor integração com o Android SDK, reforçando ainda mais esta decisão. Neste trabalho, foram utilizados o Android SDK 21.0.1 e o Google API 10, este último corresponde a versão 2.3 do sistema Android, no qual foi desenvolvida a aplicação.

A próxima seção mostrará as camadas e os componentes que formam a arquitetura do sistema Android.

### 3.1.1 Arquitetura do Google Android

A arquitetura do sistema Android, também chamado de Google Android, é mostrada na figura 3.1. O Linux Kernel é a base do sistema, que possui a versão 2.6 do Linux e inclui os serviços essenciais do Android, como o gerenciamento de memória, gerenciamento de energia, drivers, entre outros. A camada Android Runtime possui bibliotecas necessárias para as principais funcionalidades do sistema e uma Máquina Virtual Dalvik. Esta máquina virtual foi projetada para requerer pouca memória e permitir que seja instanciada várias vezes. Cada processo de uma aplicação Android roda em uma instância da Máquina Virtual Dalvik, auxiliando o sistema numa melhor gerência de memória. A camada Libraries possui outros tipos de bibliotecas, dentre as quais são bastante requisitadas para as funcionalidades da aplicação desenvolvida neste trabalho, como o Media Librarie, elaborada para suportar a reprodução de arquivos multimídia do tipo mp4, mp3, png e outros. A camada Application é onde se encontra todas as aplicações Android, como a agenda telefônica e navegador web.

---

<sup>1</sup>[http://pt.wikipedia.org/wiki/Dalvik\\_virtual\\_machine](http://pt.wikipedia.org/wiki/Dalvik_virtual_machine), acessado em 14/07/13

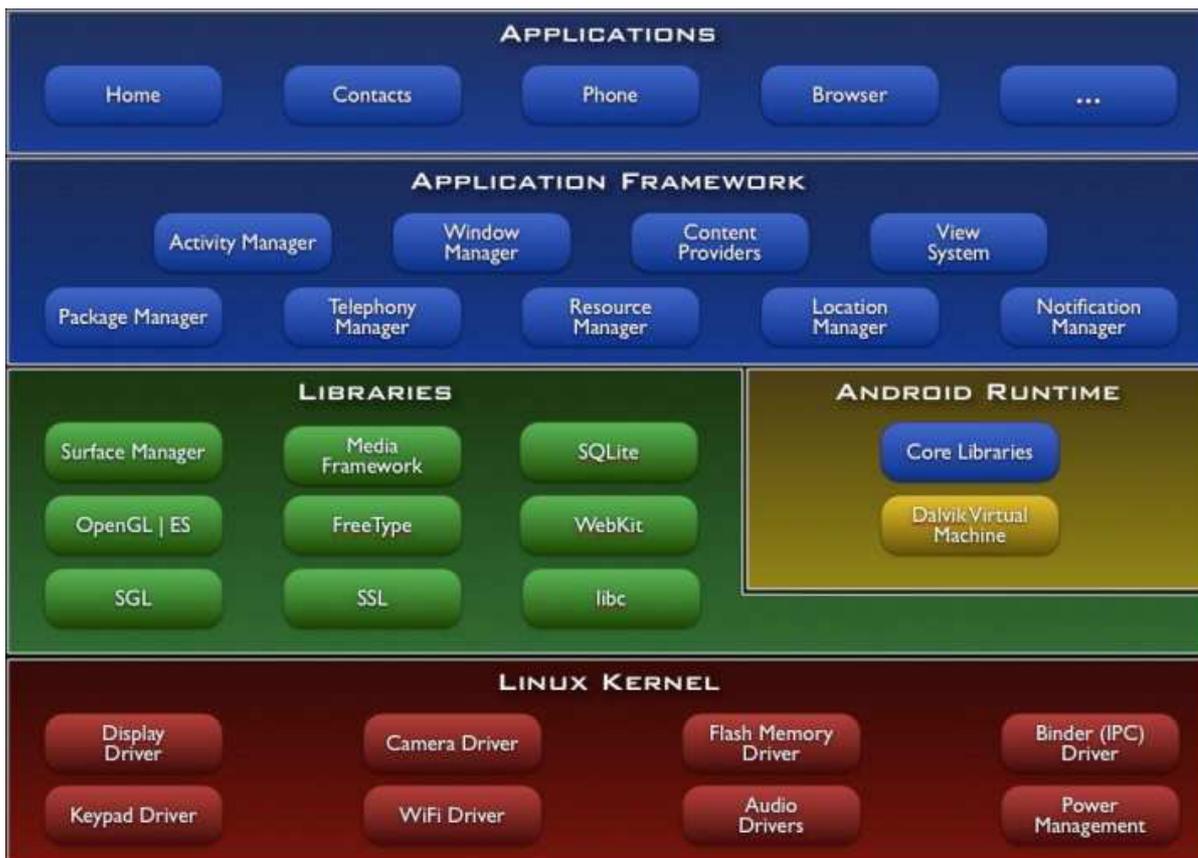


Figura 3.1: Arquitetura do sistema Android. Fonte: [Google 2013]

### 3.1.2 Classe Activity

Uma aplicação Android possui um componente muito importante para o desenvolvimento da mesma, a *Activity*. As *activities* definem como as informações serão apresentadas aos usuários. Quando um usuário vê uma tela no Android, está na realidade observando o resultado de uma interação entre a interface visual e a instância da *activity* que invocou esta interface. As *activities* podem se relacionar enviando informações umas as outras através de informações serializadas. Elas possuem um ciclo de vida dividido em seis estados (Figura 3.2), em que cada estado é considerado um método (Figura 3.3).

O estado `onCreate` é assumido quando uma *activity* é criada. Nele podem ser definidos parâmetros iniciais que são de suma importância para todo o ciclo de vida da *activity*. Assemelha-se bastante com um construtor, pela diferença de não permitir passagem de parâmetros. O `onStart` é chamada quando a *activity* está aparecendo ao usuário. Também pode ser utilizada para definir parâmetros iniciais. Porém, geralmente não é necessário ser chamado já que o seu uso pode ser substituído pelo `onCreate`, evitando redundância. O estado `onResume` ocorre quando a *activity* está saindo de uma paralisação temporária e está retornando à execução. O `onPause` retira a *activity* de sua execução salvando o seu estado corrente. O estado `onStop`



```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

Figura 3.3: Métodos correspondentes aos estados de uma Activity (Fonte: Google, 2013)

### 3.1.3 Classe Intent

Para uma activity ser alterada para outra e até mesmo se comunicarem, é necessário instanciar uma classe chamada Intent, representada por `android.content.Intent`, que permite que a activity A possa invocar a activity B em tempo de execução. A forma como esta classe é instanciada e como uma activity A invoca uma activity B, pode ser visto na figura 3.4.

```
Intent it = new Intent().setClass(this, Media.class);
int number = 0;
it.putExtra("example", number);
startActivity(it);
```

Figura 3.4: Exemplo do uso da classe Intent

Uma das formas de realizar a comunicação entre duas activities é fazendo que a activity A ao invocar a criação da activity B, por exemplo, envie um objeto serializado através da instância da classe `Intent` ou algum outro valor, como também pode ser visto na figura 3.4, quando é passado junto a instância da classe a variável inteira `number` que possui o número zero. No momento do envio de um objeto, este é associado a um identificador permitindo que outra activity possa recebê-lo. Para que o objeto seja enviado de forma serializada de uma activity a outra, é necessário especificar que o mesmo implementa a classe `Serializable`, como mostrado na figura 3.5.

```
public class Media implements Serializable {  
  
    /**  
     *  
     */  
  
}
```

Figura 3.5: Declaração de uma classe serializável

Além disso, a classe `Intent` pode ser invocada para solicitar ao sistema Android que faça chamadas telefônicas, abra o browser em algum endereço, entre outros.

### 3.1.4 Classe View

Uma activity, sendo ela uma instância da classe `android.app.Activity`, possui vários elementos visuais representados pela classe `android.view.View`. É nesta classe onde são encontrados muitos elementos comuns presentes numa tela de aplicação Android, tais como botões, caixa para inserção de texto (`TextView`), menus e outros, além de gerenciar layouts. Um dos métodos presentes nesta classe e muito importante é o `setContentView()`, que estabelece um elo de ligação entre a interface visual e a activity.

Uma interface visual, também chamada de *view*, é produzida para interagir diretamente com uma activity. Uma *view* pode ser implementada através de um `xml`, em que pode ser inseridos botões, campos de texto, um componente para exibição de imagens, entre outros. Além disso, a organização da interface pode ser feita utilizando vários tipos de layouts, como o `FrameLayout`, `LinearLayout`, e outros. A figura 3.6 mostra uma *view* em `xml` e a 3.7 exhibe como essa interface `xml` é vista pelo usuário.

### 3.1.5 Classe Adapter

Os objetos de uma activity a serem exibidos em uma tela e a *view* possuem “origens diferentes” e por conta disso são incompatíveis. Para solucionar tal problema, o Android possui um padrão de projeto chamado `Adapter`, que permite adaptar uma lista de objetos, por exemplo,

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

Figura 3.6: Exemplo de uma interface Android em xml (Google, 2013)



Figura 3.7: Exemplo de uma interface xml vista pelo usuário

para uma lista de elementos de uma interface gráfica. A figura 3.8 mostra um exemplo de uma classe que herda características de um Adapter.

```
public class ExampleAdapter extends BaseAdapter {

    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public Object getItem(int arg0) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public long getItemId(int arg0) {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public View getView(int arg0, View arg1, ViewGroup arg2) {
        // TODO Auto-generated method stub
        return null;
    }
}
```

Figura 3.8: Exemplo de um Adapter

### 3.1.6 Android.R

Quando uma interface é criada utilizando um arquivo xml, uma constante do tipo inteiro e que possui o nome do arquivo é criada na classe R. A classe R é gerada automaticamente pelo plugin Android que conecta o ambiente de desenvolvimento ao Android SDK, o ADT (Android Development Tools). A classe R é necessária para que qualquer arquivo presente no projeto da aplicação, como o xml, seja encontrado no momento da compilação do código. É esta referência do arquivo xml gerada automaticamente na classe R a ser passada como parâmetro no método setContentView(), no momento da criação da Activity, associando assim a activity com a interface.

### 3.1.7 AndroidManifest.xml

Informações como o nome da aplicação, qual é a primeira activity a ser iniciada com a aplicação, por exemplo, são declaradas em um arquivo xml, chamado AndroidManifest.xml. Toda classe que herda da classe android.app.activity deve ser declaradas no AndroidManifest.xml.

### 3.1.8 Classe Dialog

Muitas vezes é necessário exibir mensagens de alerta ou operações com respostas binárias do usuário, como “sim” e “não”; “ok” e “cancelar”, dentre outros. Para tal, o Android disponibiliza uma classe chamada `Dialog`, que consiste numa pequena janela onde o usuário pode realizar uma alguma decisão ou entrar com alguma informação. A figura 3.9 mostra um exemplo da classe `Dialog` instanciada.



Figura 3.9: Exemplo Dialog (Fonte: marakana.com)

## 3.2 Programando com o Middleware MobileHealthNet

O middleware MobileHealthNet disponibiliza um componente para o dispositivo móvel que permite à aplicação utilizar os seus principais serviços, como o contexto, conteúdo, entre outros. Este componente é chamado de MOBHA, `MobileHealthNetOnAndroid`, e foi implementado com o objetivo de diminuir o esforço do desenvolvedor com relação a implementação de métodos que realizam a publicação ou subscrição dos tópicos necessários para o estabelecimento da comunicação do cliente com o servidor. Para que ocorra esta comunicação, o Mobha possui uma biblioteca chamada “communication.jar”, que fornece as ferramentas necessárias para que o cliente realize corretamente a sua publicação e subscrição no tópico e onde se encontra todos os tópicos do middleware MobileHealthNet que as aplicações utilizam. Para que a aplicação Android possa utilizar das funcionalidades do MOBHA, é necessário que este exista na forma de biblioteca na pasta `libs` do projeto da aplicação.

O MOBHA possui várias interfaces, uma para cada serviço do middleware MobileHealthNet, que favorece o estabelecimento da comunicação entre o dispositivo móvel e o servidor. Dentre elas, destacam-se o `MOBHAAuthentication`, `MOBHAContent`, `MOBHAContext` e `MOBHAAUser`.

### 3.2.1 MOBHAAuthentication

O `MOBHAAuthentication` é uma interface para a realização do login e o logout do usuário. Como a comunicação entre o dispositivo móvel e o middleware MobileHealthNet acontece

através do paradigma publish/subscribe, utilizando tópicos, é necessário se inscrever no tópico, chamado Login, passando informações como o nome de usuário e a senha. O MOBHA, na realidade, faz essa função de subscrição ao tópico, cabendo a aplicação somente passar o mesmo como parâmetro para o método responsável por realizar esta subscrição que, neste caso, é o `publishLogin()`. A figura 3.10 mostra um exemplo de interface MOBHA com o `MOBHAAuthentication`.

```
public class MOBHAAuthentication extends MobileClient {  
  
    public MOBHAAuthentication() throws DomainParticipantNotCreatedException {  
        super();  
    }  
  
    public void publishLogIn(Login login) throws TopicNotRegisteredException,  
        DomainParticipantNotCreatedException {  
        man.publish(login);  
    }  
  
    public void subscribeToLoginRequest(PubSubTopicListener listener,  
        Appendable expression) throws TopicNotRegisteredException,  
        DomainParticipantNotCreatedException {  
        man.subscribe(listener, LoginRequest.class, expression.append());  
    }  
  
    public void publishLogInOut(LoginOut loginOut)  
        throws TopicNotRegisteredException,  
        DomainParticipantNotCreatedException {  
        man.publish(loginOut);  
    }  
  
    public void subscribeToGenericInformation(PubSubTopicListener listener,  
        Appendable expression) throws TopicNotRegisteredException,  
        DomainParticipantNotCreatedException {  
        man.subscribe(listener, GenericInformation.class, expression.append());  
    }  
}
```

Figura 3.10: Exemplo de interface MOBHA com o `MOBHAAuthentication`

### 3.2.2 MOBHAContent

O `MOBHAContent` é uma interface que provê várias operações com arquivos e diretórios, tais como *upload*, *download* do arquivo; buscar o mesmo por nome, metadados ou pelo seu proprietário; renomear, deletar ou mover o conteúdo de um diretório a outro do usuário; criar, listar, remover ou deletar um diretório, dentre outros. Para fazer o *upload* do arquivo, por exemplo, é inserido as principais informações do arquivo, inclusive seus metadados que podem variar de aplicação para aplicação, dentro do tópico `ContentUploadRequest`, passando o mesmo como parâmetro ao método `publishContentUpload()` que se encarrega de publicar

para que o servidor possa consumi-lo.

O *download* de um arquivo pode ser solicitado colocando as informações como o identificador ou o diretório do mesmo dentro do `ContentDownloadRequest` e passando este como parâmetro no método `publishDownloadRequest()`. A resposta do servidor, juntamente com o arquivo em uma array de bytes, caso ele exista, acontece por meio do tópico `ContentDownloadResponse`. De forma análoga, os outros métodos do `MOBHAContent` recebem os seus tópicos específicos para serem publicados. Este métodos são mostrados a seguir, com os respectivos tópicos de publicação:

- `searchContentByName()`: Procura um arquivo pelo seu nome. Tópico para publicação: `ContentSearchByMetadata`;
- `searchContentByMetadata()`: Oferece a possibilidade de buscar um arquivo por metadados. Tópico para publicação: `ContentSearchByMetadata`;
- `createDirectory()`: Cria um diretório para o usuário no servidor, permitindo uma melhor organização de seus arquivos. Tópico para publicação: `ContentAddDirectoryRequest`;
- `renameDirectory()`: Renomeia um diretório do usuário. Tópico para publicação: `ContentUpdateMetadata`;
- `listDirectory()`: Lista um diretório específico do usuário existente no servidor. Tópico para a publicação: `ContentListDirectory`.
- `delContent()`: Permite deletar um arquivo no servidor. Tópico para publicação: `ContentRemove`.

### 3.2.3 MOBHAContext

O `MOBHAContext` é a interface que permite com que informações de contexto, como nível da bateria do dispositivo ou a posição geográfica do usuário sejam compartilhadas na rede. O `MOBHAContext` implementa um listener que recebe tópicos do servidor que a aplicação esteja inscrita, para estes dados serem consumidos. Seus principais métodos são:

- `publishContextInformationSubscribe()`: Solicita informações de contexto de algum usuário. Tópico para publicação: `ContextInformationSubscribe`;
- `publishContextInformationUnsubscribe()`: Método utilizando para cancelar o envio de determinada informação de contexto do servidor ao cliente: Tópico para publicação: `ContextInformationUnsubscribe`.

### 3.2.4 MOBHAUser

O MOBHAUser está ligado aos serviços de usuário, permitindo a publicação de tópicos para que novos usuários sejam adicionados, removidos ou atualizados do MobileHealthNet. Os usuários que acessam a rede podem ser pacientes ou profissionais da saúde. Por conta disso, o MOBHAUser fornece dois métodos para a adição do usuário, o `publishProfessionalAdd()`, que publica o tópico `UserAdd` para adição de novos profissionais da saúde, e o `publishPatientAdd()`, que faz a operação de adicionar novos pacientes através também do tópico `UserAdd`. Este possui todos os dados relevantes para serem incluídos juntamente com o tipo de usuário a ser cadastrado. Observando a figura 3.11 é possível perceber que a estrutura do tópico `UserAdd` oferece a possibilidade de adicionar informações específicas para cada tipo de usuário. No atributo `userType`, por exemplo, define-se o tipo, se o usuário é paciente ou profissional da saúde. Caso o usuário seja um paciente, informações como a doença e a data do início do tratamento devem ser informadas. Se ele for um profissional da área da saúde, a especialidade em que ele atua deve ser adicionada com o seu perfil. Dados relevantes como nome, rg, cpf, nome de usuário (login), senha, data de nascimento, sexo e email, são pertinentes aos dois tipos de usuários, seja ele paciente ou profissional da área da saúde.

```
1 package br.ufma.lsd.mbhealthnet.communication.ddsttopics;
2 import java.io.Serializable;
3
4
5 // CoreDX DDL Generated code. Do not modify - modifications
6
7
8
9 public class User implements Serializable {
10
11     // instance variables
12     public String toUserName;
13     public String userName;
14     public String password;
15     public String name;
16     public String email;
17     public String rg;
18     public String cpf;
19     public String birthDate;
20     public String createDate;
21     public boolean secure;
22     public String userType;
23     public String especialidadeName;
24     public String careType;
25     public String disease;
26     public String projectId;
27     public String sexo;
28     public String treatmentStartDate;
29
```

Figura 3.11: Tópico UserAdd

Outros métodos importantes para o uso do serviço de usuário do MobileHealthNet, disponibilizados aos clientes por meio do MOBHAUser são:

- `publishUserRequest()`: Permite solicitar as informações de um usuário. Tópico para publicação: `UserRequest`;
- `publishPatientUpdate()`: Método utilizado para atualizar informações de um paciente. Tópico para publicação: `UserUpdate`;
- `publishProfessionalUpdate()`: Método utilizado para atualizar informações de um profissional da saúde. Tópico para publicação: `UserUpdate`;

Existem situações em que ao publicar alguma informação, espera-se receber outra informação. Isto pode ser exemplificado quando há intenção de adicionar algum usuário, no qual o tópico `UserAdd` é publicado com a intenção de adicionar algum usuário, e a resposta ao cliente é a confirmação se o usuário foi adicionado com sucesso ou não.

Para estas mensagens chegarem ao cliente, é necessário implementar na aplicação um listener, que herda características da classe `PubSubTopicListener`, presente na biblioteca `communication.jar`. Este listener fica esperando um tópico chegar até o mesmo, publicado pelo servidor. Para o tópico A chegar ao cliente, é necessário subscrever a aplicação neste tópico. O MOBHA já faz a subscrição em todos os tópicos de resposta, facilitando a implementação da aplicação, bastando que o desenvolvedor inclua no código o listener. Este listener possui um método chamado `processTopic()` que recebe como parâmetro um objeto, sendo ele um tópico. Dentro deste método podem ser feitas várias condicionais que, dependendo do tópico que chegar ao cliente, é realizada uma operação específica. A figura 3.12 mostra o `processTopic()`.

```
// Handle topic responses
PubSubTopicListener listener = new PubSubTopicListener() {

    @Override
    public void processTopic(Object ddsTopic) {

    }
};
```

Figura 3.12: Exemplo de listener para ouvir a chegada dos tópicos no lado do cliente

## ***4 Aplicação Colaborativa na Área da Educação Médica***

As duas unidades do HUUFMA, o PAPA e a Casa da Dor, possuem programas voltados para educação de pacientes e profissionais da saúde. Nestes programas são realizadas palestras educativas relacionadas tanto à asma, quanto à dor, com o objetivo de fornecer aos pacientes informações sobre prevenção ou tratamento de sua doença ou dor, o uso de medicamentos e como manusear aparelhos (como o pico de fluxo expiratório, no caso da asma). Os profissionais da saúde frequentam palestras com temas similares, porém, diferentemente dos pacientes, estas palestras têm como principal objetivo treinar os profissionais da saúde para posteriormente auxiliarem o paciente na prevenção e tratamento de sua doença ou dor, no uso de aparelhos e medicamentos.

Entretanto, muitos pacientes têm dificuldade em estarem presentes nestas palestras, seja devido a distância de onde residem até o local em que acontecem as palestras educacionais, ou por questões de saúde que os impedem de se locomoverem de um ponto a outro. Além disso, os problemas urbanos acabam se tornando barreiras, tais como a deficiência dos transportes públicos, que frequentemente carregam a quantidade de passageiros muito além do seu limite, ou não oferecem suporte para pacientes com deficiência física que utilizam cadeira de rodas, por exemplo.

Dificuldades também são encontradas pelos profissionais da saúde em participar destas palestras nos centros de tratamento, por conta da demanda de serviços presentes nos hospitais, inclusive a necessidade de alguns estarem em diferentes locais de atendimento à saúde durante todo o dia.

Educar pacientes sobre como manusear um aparelho ou usar corretamente um medicamento permite que a evolução positiva do tratamento ocorra mais rapidamente, pois é uma forma de manter o controle sobre o mesmo, como monitorar a pressão sanguínea ou a ingestão de um medicamento para dor, por exemplo, sem a presença de um profissional. Ensinar a prevenir doenças também é considerado um ponto importante, porque orienta o paciente a evitar situações que possam submetê-lo a doenças que, em conjunto com a enfermidade a qual ele está sendo tratado, possa vir a tornar o seu quadro clínico como crítico.

Informar os profissionais da saúde sobre como manipular aparelhos ou dosagem de medicamentos é uma forma de garantia ao paciente de que ele estará realizando os procedimentos corretos quando for ingerir algum medicamento ou fazer uso de aparelhos, considerando que normalmente os pacientes procuram com maior frequência os profissionais da saúde quando há dúvidas na realização destes procedimentos. Levar aos profissionais estas informações também é uma forma de renovar ou “reciclar” o seu conhecimento, já que atualmente as tecnologias na área médica evolui muito rapidamente, exigindo constante atualização das informações que eles adquiriram ao longo de anos de trabalho.

Portanto, é evidente a necessidade de levar conhecimentos aos pacientes sobre todos os aspectos que envolvam o seu tratamento, desde o uso de aparelhos até a forma como ingerir um medicamento, bem como aos profissionais da saúde que participam no acompanhamento do paciente. Considerando as dificuldades existentes tanto para os pacientes quanto aos profissionais da saúde de estarem presentes nos locais onde estas informações seriam transmitidas através das palestras educacionais, surgiu a necessidade destas informações irem ao encontro deles, tendo em vista a diminuição do esforço em se deslocarem de um ponto a outro no espaço e a melhor organização de seus horários disponíveis.

Para que estas informações possam estar disponíveis a qualquer momento e em qualquer lugar, se faz necessário a utilização de TIC's que permitam o tráfego de dados de um ponto a outro, tal como a Internet, e de dispositivos que deem suporte ao usuário de receberem ou enviarem estas informações de qualquer lugar do espaço, como dispositivos de computação portáteis, por exemplo.

Neste contexto, idealizou-se a aplicação *HealthEducation*, que tem por objetivo levar estas informações, encontradas anteriormente somente nas palestras dos centros médicos, aos pacientes e profissionais da saúde independente da sua localização geográfica através do uso de tecnologias de dispositivos de computação móveis. Estas informações são disponibilizadas ao usuário na forma de mídias, como vídeos, áudios ou apresentações. Além disso, cada mídia educacional pode receber comentários de todos que a acessam, inclusive votações que refletem na qualidade do conteúdo disponibilizado. Os comentários atribuídos aos vídeos, áudios ou apresentações, podem ser vistos por quaisquer usuários, sejam eles pacientes ou profissionais da saúde que acessam a aplicação, permitindo assim o compartilhamento de opiniões em torno de qualquer conteúdo educacional existente na base dados.

As seções a seguir mostrarão os requisitos elaborados para o desenvolvimento da aplicação; o resultado de sua implementação, mostrando na forma de imagens as funcionalidades existentes e o que é cada uma destas funcionalidades; e como a aplicação foi implementada, incluindo as tecnologias envolvidas.

## 4.1 Requisitos

Esta seção aborda os requisitos elaborados para o desenvolvimento da aplicação. O surgimento dos requisitos a seguir é resultado de interações que ocorreram com os profissionais da saúde do Hospital Universitário Dutra.

1. A identidade do usuário deve ser confirmada para que o mesmo possa acessar os recursos da aplicação através de algum mecanismo de autenticação, como o fornecimento de um usuário e senha.
2. A aplicação deverá permitir ao usuário profissional da saúde cadastrar novos elementos de mídia educativa.
3. A aplicação deve permitir aos usuários o acesso aos elementos de mídia educativa cadastrados, sejam elas áudio-aulas, vídeo-aulas ou apresentações.
4. A aplicação deverá permitir o *download* de mídias educacionais relativas a doença e ao tempo corrente do tratamento do paciente, caso este seja o solicitante.
5. A aplicação deve permitir aos usuários o registro de comentários a respeito das mídias educativas por eles acessadas.
6. A aplicação deve possuir um mecanismo de filtragem de conteúdo que recomende mídias aos usuários de acordo com os seguintes dados de contexto: doença do paciente e seu tempo de tratamento e a especialidade do profissional da saúde.
7. A aplicação deve possuir um mecanismo de busca que permita localizar conteúdos cadastrados de acordo com seus metadados: público alvo, comunidade alvo, descrição e tipo de conteúdo.
8. A aplicação deve organizar as mídias cadastradas em categorias. As seguintes categorias devem ser utilizadas: documentos, medicamentos e aparelhos.
9. A aplicação deve permitir que os usuários votem sobre a qualidade do conteúdo apresentada.

## 4.2 Funcionalidades e Layout de Telas

A aplicação *HealthEducation* possui muitas funcionalidades que tem por objetivo oferecer ao usuário uma melhor experiência com a rede social, através, por exemplo, do compartilhamento de mídias, inclusão de comentários sobre as mesmas e que podem ser vistos por outros usuários, entre outros.

### 4.2.1 Tela de abertura e login

Ao iniciar a aplicação, o usuário deve inserir seu nome de usuário e a senha cadastrada no servidor. Caso o usuário exista, suas informações são enviadas ao dispositivo, permitindo realizar as operações que dependam dos dados relativos a sua doença, por exemplo. A figura 4.1 mostra a tela de login.



Figura 4.1: Imagem da tela de login

### 4.2.2 Classificação das mídias

A aplicação classifica o conteúdo educacional por categorias, tais como uso de medicamento e manipulação de aparelhos, que podem ser definidas nos metadados da mídia no momento de realizar o upload da mesma. A classificação das mídias educacionais por categorias fornece a possibilidade do usuário visualizar somente o conteúdo da categoria selecionada, eliminando da lista de mídias aquelas que no momento não são de interesse.

As mídias também podem ser classificadas pela sua data de publicação ou pela quantidade de votações que recebeu dos usuários. Neste sentido, a qualidade do conteúdo educacional pode ser avaliada (Figura 4.2b), permitindo formar um conjunto de informações que melhor atendam tanto as necessidades dos pacientes quanto as dos profissionais da saúde. No caso do paciente, por exemplo, a avaliação é importante para que o profissional da saúde saiba quanto o conteúdo apresentado está satisfazendo a necessidade do paciente de obter as informações necessárias



Figura 4.2: Imagens das telas Categorias e Avaliar

durante o período de tratamento da doença, como a asma ou dor. E no caso dos profissionais da saúde que buscam se atualizarem quanto alguma informação relativa a sua especialidade, a avaliação da qualidade das mídias favorece uma melhor colaboração de informações entre eles mesmos.

### 4.2.3 Avaliação das mídias pelos usuários

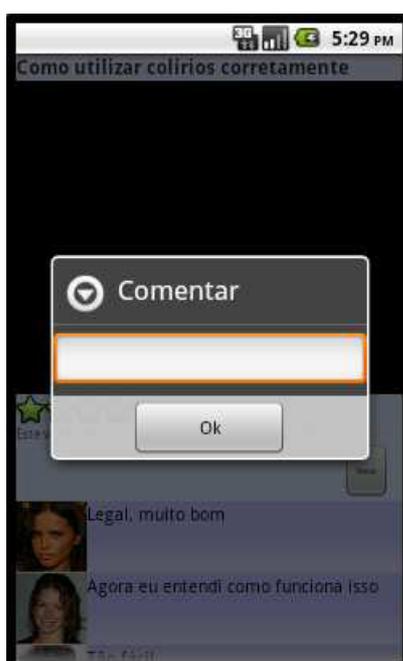
As mídias podem receber uma classificação na escala de 1 até 5, em que 5 significa a maior nota. As notas recebidas pelas mídias são representadas através de estrelas, ou seja, se o conteúdo recebeu uma média de nota 4, na lista a mídia será apresentada com 4 estrelas e assim sucessivamente. A aplicação *HealthEducation* possui uma aba nomeada de “Mais Votados” que mostra as mídias relativas ao perfil do usuário cuja média de votações que receberam seja maior ou igual a 3 estrelas. A figura 4.3 mostra a tela dos mais votados.

Os usuários podem dar a sua opinião em relação a mídia através de comentários, favorecendo a troca de opiniões entre os mesmos que procuram informações similares, como mostrado na figura 4.4a. Os comentários são exibidos em uma lista de comentários dentro do visualizador de mídias, onde as mídias educacionais do tipo áudio-aula ou vídeo-aula são executadas (Figura 4.4b). As apresentações educacionais são exibidas através de uma aplicação com suporte a visualização em pdf, como o *Adobe Reader*. A aplicação *HealthEducation* “invoca” a aplicação



Figura 4.3: Imagem da tela “Mais Votados”

*Adobe Reader* para que esta exiba a apresentação educacional ao usuário. Cada mídia possui a sua lista de comentários e cada comentário o a foto do usuário. Ao pressionar o botão “menu” no visualizador de mídias, é visto a opção de comentar sobre a mídia educacional.



(a) Imagem da tela Comentar



(b) Imagem da tela Visualizador de Mídias

Figura 4.4: Imagens das telas Buscar e Mídias Recentes

### 4.2.4 Buscando mídias

A aplicação também possui um sistema de buscas que permite ao usuário procurar os conteúdos educacionais de seu interesse filtrando a sua busca através das opções: buscar por conteúdo, público alvo, comunidade alvo ou por descrição. Na figura 4.5a é possível ver a tela onde o usuário busca a mídia educacional que deseja ver/ouvir de acordo com os metadados da mídia. Caso um paciente queira que seja exibidas mídias relacionadas a nebulização, basta que ele marque a opção “descrição” e escreva no campo “buscar” a palavra “nebulizador”, por exemplo, para que mídias relacionadas a nebulização sejam exibidas.



Figura 4.5: Imagens das telas Buscar e Mídias Recentes

### 4.2.5 Mídias Recentes

A aplicação também dispõe da listagem das mídias postadas mais recentemente, como mostrado na tela da figura 4.5b. Isto permite aos usuários verificarem quais os conteúdos relacionados ao seu perfil que foram recentemente colocados na rede, favorecendo aos pacientes para que tenham sempre conteúdos com informações mais atuais com o propósito de favorecer seu tratamento, e para os profissionais atualizarem seus conhecimentos na área.

## 4.3 Aspectos de Implementação

A aplicação *HealthEducation* faz uso dos conceitos de redes sociais, como compartilhamento de informações e opiniões acerca de algum assunto ou conteúdo, utilizando dispositivos móveis, porém com foco para a área da saúde.

### 4.3.1 Organização das telas e conteúdos

A aplicação *HealthEducation* possui uma *view* principal chamada `conteudo_tab_view.xml` que organiza quatro telas na forma de abas, são elas: categorias, mídias recentes e mais votados (Mais sobre elas na seção 4.4). Cada uma delas, exceto “categorias”, possui uma *view* em comum, a `listgridview.xml`, organizado de forma a exibir na tela do usuário a lista de mídias correspondentes àquela aba. Quando pressionada a aba “mais votados”, por exemplo, o usuário terá acesso a lista de mídias mais votadas e correspondentes ao seu perfil. Ao realizar o seu login, as informações do perfil do usuário são enviadas ao dispositivo móvel, permitindo saber, no caso do paciente, qual a sua doença e o período de tratamento que ele se encontra e, no caso do profissional da saúde, qual a sua especialidade. Com estes dados, a aplicação solicita ao servidor as informações necessárias para formar a lista de conteúdos relacionados ao perfil do usuário, tais como nome da mídia, número de votações, descrição e a data de `upload` da mídia. A data de `upload` auxilia na decisão de quais mídias existentes no servidor podem formar a lista presente na tela cuja aba corresponde às “mídias recentes”. Da mesma forma, o número de votações permite construir a lista de mídias na telas cuja aba está relacionada aos “mais votados”.

A tela “categorias” organiza as mídias por medicamentos, aparelhos e documentos. Quando o usuário clica em medicamentos, por exemplo, a aplicação procura os conteúdos que abordam informações sobre medicamentos e que sejam relacionados ao perfil do usuário, através de metadados. Com posse de quais conteúdos atendem a solicitação do usuário, a aplicação organiza uma lista de mídias e as exibe, seguindo a mesma forma de organizar a lista de mídias das telas “mais votados” e “mais recentes”, utilizando o `ListAdapter`. A aplicação *HealthEducation* permite também que o usuário possa buscar algum conteúdo por sua descrição, comunidade alvo, público alvo e tipo de conteúdo. Quando o usuário realiza uma busca, a classe `Sniffer` é instanciada chamando o método `get()` que recebe como parâmetro por qual tipo de metadado que o usuário quer realizar a busca, se por público alvo, comunidade alvo, descrição ou tipo de conteúdo, e o valor ou a chave da busca. Por exemplo, se o usuário desejar buscar conteúdos relacionados a dores de cabeça ele pode marcar a opção “descrição” e escrever “dores de cabeça”. A aplicação recolhe a opção marcada e o que o usuário escreveu (a chave) e publica no tópico `ContentSearchByMetaData` para que sejam buscados conteúdos que possuam este tipo de metadado. Uma vez existente estes conteúdos, suas principais informações, como nome, número

de votações, descrição, entre outros, são enviadas à aplicação para que ela possa organizar.

A informação de uma mídia é lançada em uma instância da classe `Media` (Figura 4.6), que possui todos os atributos necessários para receber tais informações. O conjunto de todos os objetos `Media` forma uma lista que, ao mudar a tela da aba “mídias recentes” para “mais votados”, por exemplo, os seus objetos são alterados para exibir somente as mídias mais recentes ou as mais votadas. Esta lista de mídias é na realidade uma array de `Media`, e para que possa ser exibida ao usuário, é necessário enviá-la a um adaptador de lista, chamado na aplicação de `ListAdapter`. Este adaptador se integra a *view* `medialist.xml`, fornecendo ao mesmo o título, descrição e o número de votações que o conteúdo recebeu.

```
public Media(String name, String path, String description,
             Metadata metadata, int idMedia, ArrayList<Comment> comments,
             Date uploadDate, int ratings) {

    this.name = name;
    this.path = path;
    this.description = description;
    this.metadata = metadata;
    this.idMedia = idMedia;
    this.comments = comments;
    this.ratings = ratings;
    this.uploadDate = uploadDate;
}
```

Figura 4.6: Construtor da classe `Media`

Os conteúdos recebidos pela aplicação colaborativa educacional através da comunicação entre o dispositivo móvel e o servidor devem estar disponíveis ao usuário com os seus principais metadados, tais como os comentários, nome da mídia, número de votações, entre outros. Para a reprodução das mídias, tais como áudio ou vídeo, é necessário instanciar a classe `MediaPlayer()` disponibilizada pelo `AndroidSDK`, passando como parâmetro ao seu construtor o diretório onde a mídia se encontra. No caso da aplicação *HealthEducation*, as mídias educacionais solicitadas pelo usuário são salvas, por padrão, diretamente no cartão de memória e posteriormente lançadas ao reproduzidor de mídias, caso o paciente ou o profissional queiram reproduzi-las.

Os conteúdos educacionais são transmitidos ao dispositivo móvel na forma de array de *bytes*, dentro do tópico `ContentDownloadResponse`. Esta array de *bytes* é colhida do tópico pela aplicação *HealthEducation* e convertida para os formatos `mp4` ou `3gp`, caso o tipo de conteúdo for um vídeo, e `mp3` caso o tipo do conteúdo for áudio. A conversão da array de *bytes* para os tipos de conteúdo, como áudio ou vídeo, é realizada chamando o método `writeByteToFile()` (figura 4.7), que recebe como parâmetro a array de *bytes* e a escreve no cartão de memória no formato adequado. A escrita no cartão de memória só é permitida adicionando no `AndroidManifest.xml` a linha mostrada na figura 4.8 no arquivo `AndroidManifest-`

.xml.

```
private void writeByteToFile (byte [] array) throws IOException{
    String strFilePath = "/mnt/sdcard/";
    String nameFile = "text.txt";
    File file = new File(strFilePath, nameFile);
    file.createNewFile();
    BufferedOutputStream out = new BufferedOutputStream(new FileOutputStream(file), 1);
    out.write(array);
}
```

Figura 4.7: Método writeByteToFile()

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Figura 4.8: Permissão para escrever no cartão de memória

Esta conversão para os tipos de mídia é necessário porque o reprodutor de mídias nativo do Android não aceita reproduzir os conteúdos passando como parâmetro um array de *bytes*. Cada conteúdo educacional pode ter vários comentários associados produzidos pelo usuário. Quando o profissional da saúde ou o paciente requisitam o conteúdo, os comentários relacionados ao mesmo também são solicitados e lançados na mesma activity que possui o reprodutor de mídias. A interface que representa a activity *MediaViewActivity* foi desenhado dividindo a tela em duas partes: a parte superior onde se encontra o reprodutor de mídias e a parte inferior em que são encontrados os comentários do conteúdo que está sendo reproduzido. Nesta mesma activity, é oferecida a opção de comentar e votar sobre o conteúdo. A opção de comentar é implementada através do método `showToComment()` que permite ao usuário inserir o seu comentário sobre o conteúdo educacional. A tela de comentário é elaborada através de um *Dialog*, em que possui um campo de inserção de texto onde a aplicação obtém o comentário do usuário e publica no tópico para que seja incrementada a lista de comentários, caso ela exista, ou criá-la caso seja o primeiro comentário.

O método que permite a realização de votações dos usuários às mídias é invocado através do botão “Votar” fixado na interface visual por meio de um “*Button*”. Na classe correspondente a activity, é implementado um listener que fica “ouvindo” o usuário pressionar o botão e definir qual a nota ele irá atribuir a mídia.

As mídias educacionais na forma de apresentação e que possuem o formato pdf, são abertas com o aplicativo nativo do Android para a execução destes tipos de mídia, como o *Polaris Viewer*. ou outros como o *Adobe Reader*.

### 4.3.2 Diagrama de classes

A relação entre as classes da aplicação *HealthEducation* ocorre de forma a prover uma melhor manipulação das mídias por parte dos usuários, consistindo desde a distribuição delas

por categorias, até a disposição das mesmas em uma aba relacionada a visualização de mídias recentemente publicadas no servidor pelos profissionais da saúde, por exemplo. Esta seção irá mostrar os diagramas das principais classes da aplicação *HealthEducation*, através de um diagrama geral para visualização destas classes principais e de outros diagramas que possuem os métodos e atributos destas classes. Algumas imagens serão rotacionadas para ficarem na posição vertical, em vista de uma melhor visualização dos diagramas.

O diagrama da figura 4.9 exibe de uma forma geral como as principais classes da aplicação *HealthEducation* se relacionam.

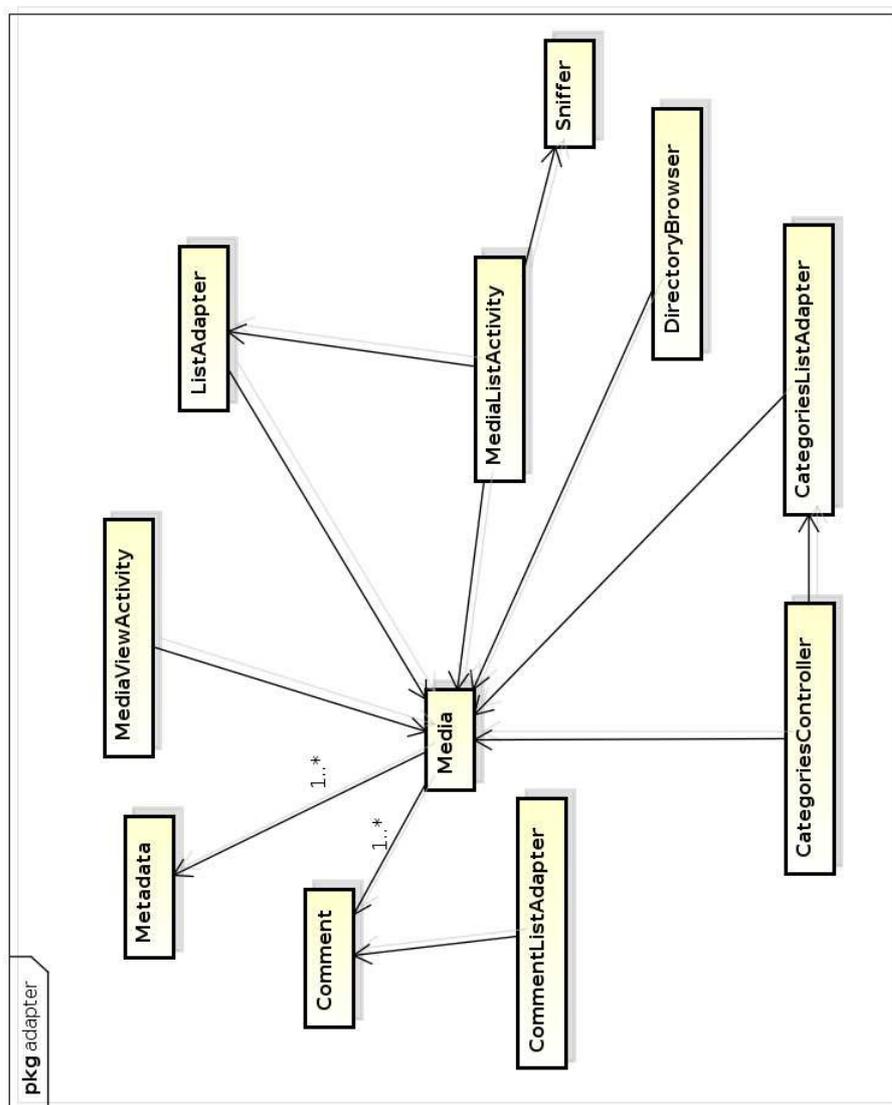


Figura 4.9: Diagrama de classes de visualização geral

A instância da classe *Media* possui todas as informações relacionadas a uma mídia educacional, tais como o nome do conteúdo, seu identificador na base de dados, diretório, entre outros. Na base de dados, as mídias educativas se relacionam com comentários e seus metadados, definidos no momento quando é realizado o upload das mesmas. Por conta disso, essa relação

deve ser visível pro usuário ao manipulá-las, obrigando a aplicação realizar instâncias de classes auxiliares para exibir as informações que estão agregadas a estas mídias, como os comentários e os metadados. Para cada comentário da mídia, é gerada uma instância da classe `Comment`. No objeto `Media` existe um atributo que guarda todos os objetos `Comment` em uma lista, podendo também esta lista ser incrementada quando o usuário fornece seu comentário sobre o conteúdo. No `Media` também existe um atributo que referencia um objeto da classe `Metadata`, no qual guarda as informações (metadados) que permitem as mídias serem facilmente organizadas e buscadas pelos usuários na aplicação.

A aplicação possibilita a visualização das mídias educacionais por categorias. Esta funcionalidade é realizada através da classe `CategoriesController`, que captura as seguintes intenções dos usuários: exibir uma lista de mídias relacionadas a medicamentos; ou relacionadas a aparelhos; ou relacionadas a apresentações. Após obtida a intenção, o objeto `CategoriesController` instancia a classe `CategoriesListAdapter` e a envia uma lista de mídias que se enquadram especificamente a categoria medicamentos, ou aparelhos, ou documentos que envolvam apresentações. O objeto `CategoriesListAdapter` obtém esta lista e a prepara para exibi-la na interface da aplicação.

A navegação por diretórios existentes no dispositivo móvel é permitida através da classe `DirectoryBrowser`, possibilitando também o compartilhamento das mídias selecionando-as na lista.

O objeto `MediaListActivity` cria uma lista de mídias de acordo com o perfil do usuário obtido durante o login. Dentro do perfil do usuário, existe a doença e o tempo de tratamento do paciente ou a especialidade do profissional da saúde. De posse desses dados, ela solicita ao servidor quais mídias existentes na base de dados e que correspondem a doença informada com o tempo de tratamento ou a especialidade fornecida. O servidor retorna as informações dos conteúdos existentes para a aplicação para então elaborar a lista de mídias. Esta classe também organiza as mídias de acordo com a aba que o usuário pressiona, tal como “mais votados” ou “mídias recentes”.

Para que os objetos `Media` sejam exibidos aos usuários, a instância da classe `MediaListActivity` envia uma lista de mídias ao `ListAdapter`. Este retira as informações existentes dentro dos objetos `Media` e as exibe na interface do usuário. Informações estas correspondentes ao nome do conteúdo, número de votações e a sua categoria.

O reprodutor de mídias nativo do Android, e invocado dentro do `MediaViewActivity`, que recebe o objeto `Media` com seus comentários. Exibir estes comentários na interface exige a presença de um adaptador, que neste caso é o `CommentListAdapter`, onde prepara uma lista com os comentários, nome de quem comentou e a foto do usuário.

A classe `Sniffer` é responsável por permitir que os usuários busquem conteúdos por metadados.

As subseções a seguir mostrarão os diagramas das principais classes da aplicação *HealthEducation* com os seus métodos.

### 4.3.3 Diagrama da classe Media

A classe *Media* é composta pelos atributos `name`, `path`, `description`, `numberVisualization`, `ratings` e `idMedia`. Estes atributos relacionam-se, respectivamente, ao nome do arquivo, seu diretório, descrição, número de visualizações (necessária para definir quais mídias são as mais vistas e, assim, coloca-las na tela correspondente a aba “mais vistos”), número de votações realizadas pelo usuário e o identificador do arquivo no banco de dados do servidor. Na figura 4.10 mostra a lista destes atributos inclusive os métodos presentes nesta mesma classe. Todos estes métodos são utilizados para inserir valor em uma variável e obter um valor na mesma variável. Os nomes dos métodos são definidos de acordo com a operação que exerce. Caso o método seja para inserir um valor em uma variável, seu nome é “set + nome da variável”. Caso seja para obter o valor em uma variável, seu nome é “get + nome da variável”. O método `setName()`, por exemplo, insere o valor na variável `name`. E o método `getName()` retorna um valor da variável `name`. Todos os outros métodos exercem as mesmas operações, de inserir e retornar um valor referente a uma variável.

### 4.3.4 Relação entre a classe Comment e CommentListAdapter

A classe *Comment* possui os atributos `userId`, `comment`, `userName` e `photo`. Estes são responsáveis por manter, respectivamente, os valores do identificador que o usuário possui na base de dados, o comentário realizado, o nome de usuário para a realização do login e a foto de quem comentou. Todos estes atributos serão exibidos na lista de comentários de uma mídia, através do *CommentListAdapter*. Antes do adaptador *CommentListAdapter* exibir os objetos *Comment* na lista de comentários, ele faz uma chamada ao método `getListComments()` para obter a lista de comentários. Na figura 4.11 é mostrada a relação entre as classes *Comment* e *CommentListAdapter*.

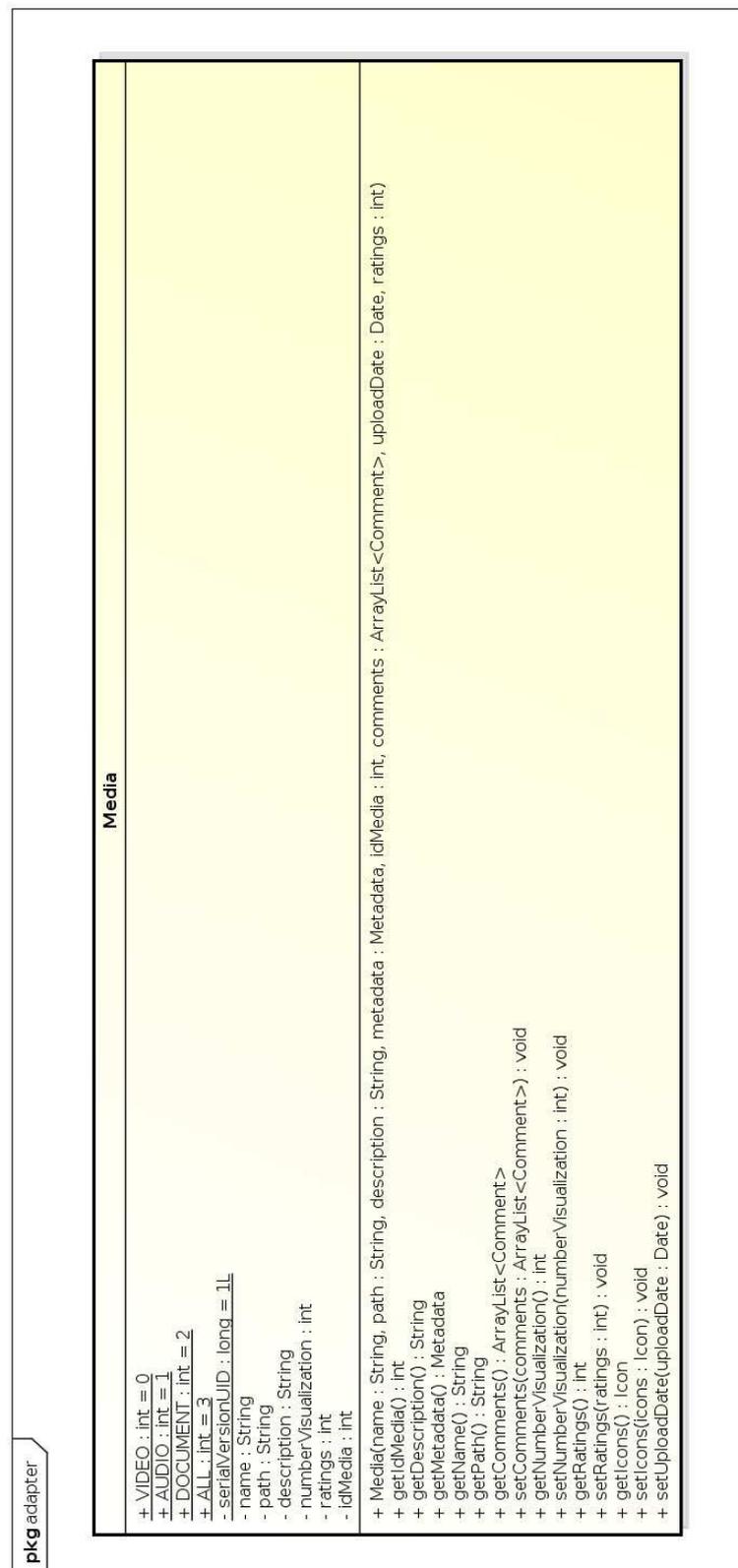


Figura 4.10: Diagrama da classe Media

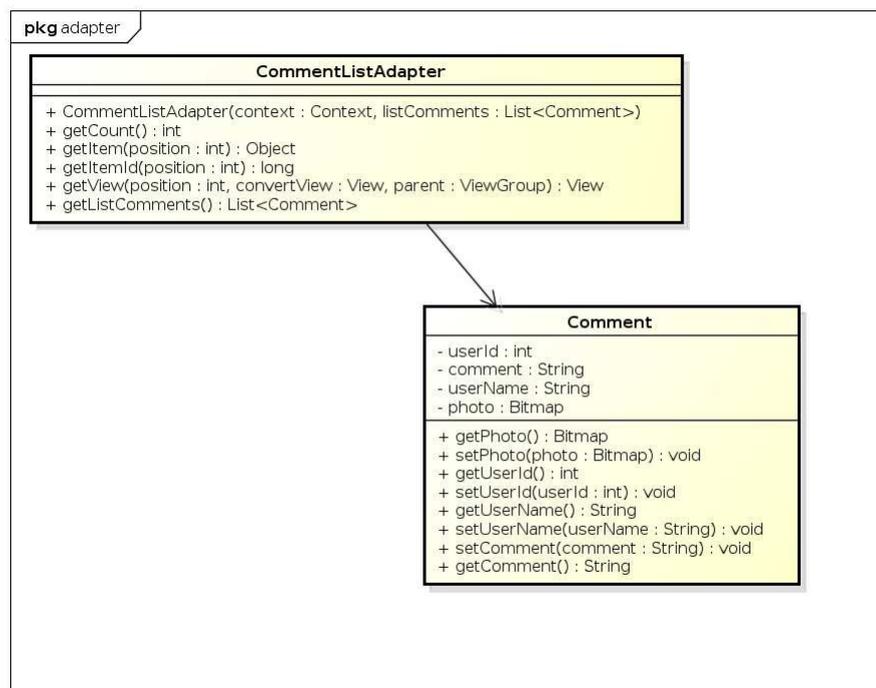


Figura 4.11: Diagramas das classe Comment e CommentListAdapter

### 4.3.5 Diagrama das classes MediaListActivity, ListAdapter e MediaViewActivity

O objeto `MediaListActivity` organiza todos os objetos `Media` em uma lista, adequando-a sempre que o usuário alterar as abas para “mídias recentes” ou “mais votados”. Uma vez tendo a lista de mídias de acordo com a solicitação do paciente ou profissional da área médica, o objeto `MediaListActivity` instancia a classe `ListAdapter`, enviando para a mesma a lista de mídias para que possa ser exibida na interface as mídias relacionadas ao perfil do usuário. Esta relação pode ser vista na figura 4.12

O método `setWhichList()` é utilizado para definir se a lista de mídias a ser exibida será composta somente por mais votadas ou recentemente postadas. A definição da lista só acontece quando uma das abas, “Mídias Recentes” ou “Mais Votados”, são pressionadas. O `getWhichList()` da classe `MediaListActivity` é invocado quando uma das abas, “Mais Votados” ou “Mídias Recentes”, são pressionadas e é utilizado quando se instancia a classe `ListAdapter`, passando como informação a este objeto se a lista será relacionada a aba “Mais Votados” ou “Mídias Recentes”.

A figura 4.13 mostra a classe `MediaViewActivity`, que invoca o reproduutor de mídias nativo do Android. Para exibir os comentários das mídias, o método `showComment()` é invocado.

O método `loadDocInReader()` chama o aplicativo nativo do Android para abrir apresen-

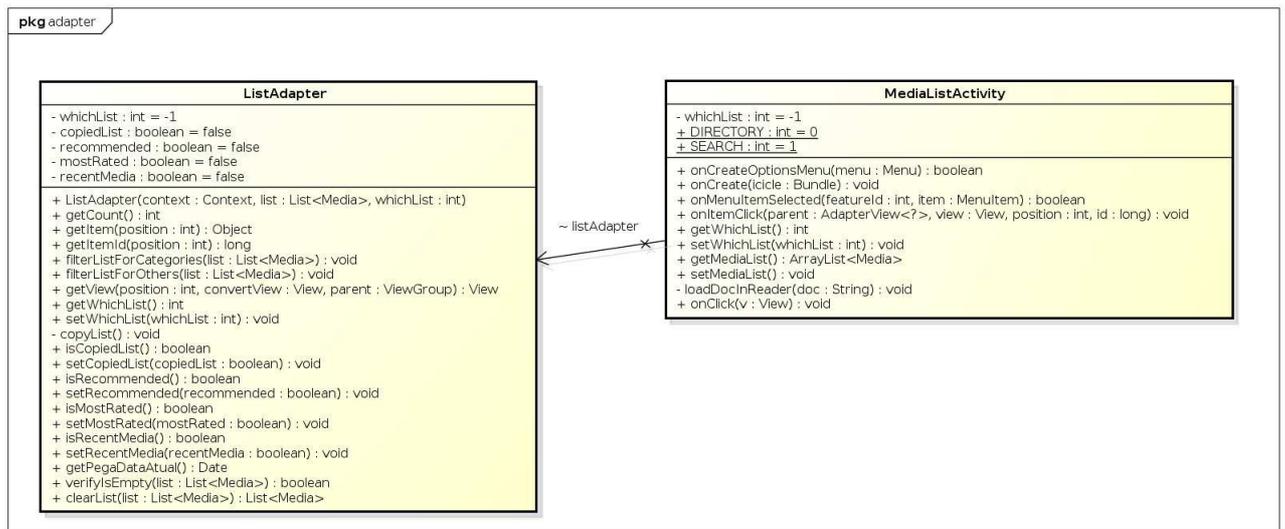


Figura 4.12: Diagramas das classe MediaListActivity e ListAdapter

tações no formato pdf, como o *Polaris Viewer*.

Já na classe `ListAdapter` existem dois métodos que informam ao adapter se a lista é voltada para a aba “Mídias recentes” e “Mais votados”: `isRecentMedia()` e `isMostRated()`. Se ao invocar o método `isRecentMedia()` for retornado verdadeiro, a lista é voltada para a aba “Mídias recentes”. Se ao invocar o método `isMostRated()`, a lista é voltada para a aba “Mais votados”. Vale ressaltar que somente um dos métodos pode retornar verdadeiro.

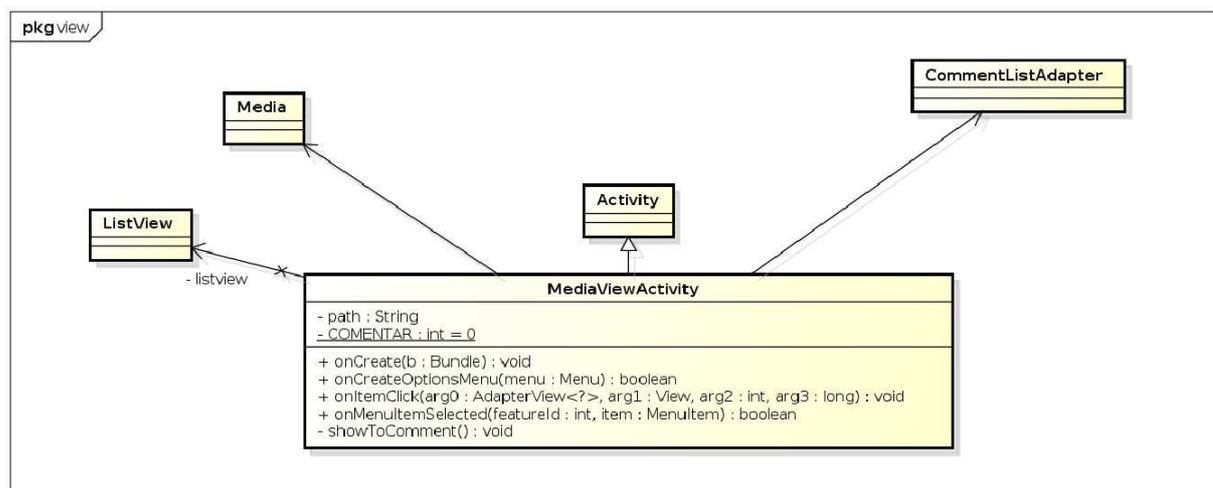


Figura 4.13: Diagrama da classe MediaViewActivity

### 4.3.6 Utilizando o MOBHAAuthentication

Usufruir de todos os serviços que o *middleware* MobileHealthNet tem a oferecer através de um dispositivo móvel utilizando o sistema *Android* só é possível com o componente MOBHA. O MOBHA é formado por interfaces que se comunicam com todos os serviços disponibilizados pelo *middleware* MobileHealthNet, como o serviço de usuário, conteúdo, chat, entre outros.

Quando a aplicação *HealthEducation* é inicializada o método `publishLogin()` da classe `MOBHAAuthentication`, presente no componente MOBHA, é invocado para que ocorra a autenticação do usuário no servidor, publicando no tópico `Login` o nome do usuário, senha e a opção de trafegar os dados do usuário de forma segura ou não. A figura 4.14 mostra a estrutura do tópico `Login`:

```
1 package br.ufma.lsd.mbhealthnet.communication.ddsttopics;
2 import java.io.Serializable;
3
4
5 // CoreDX DDL Generated code. Do not modify - modifications may
6
7
8
9 public class Login implements Serializable {
10
11     // instance variables
12     public String userName;
13     public String password;
14     public boolean secure;
15
16     // constructors
17     public Login() {}
18     public Login( String __f1, String __f2, boolean __f3 ) {
19         userName = __f1;
20         password = __f2;
21         secure = __f3;
22     }
23
24     public void clear() {
25         userName = null;
26         password = null;
27     }
28
29     public void copy( Login from ) {
30         this.userName = from.userName;
31         this.password = from.password;
32         this.secure = from.secure;
33     }
34
35 }; // Login
36
```

Figura 4.14: Estrutura do tópico `Login`

### 4.3.7 Utilizando o MOBHAUser para obter os dados dos pacientes ou dos profissionais da saúde

Cada paciente cadastrado no MobileHealthNet possui informações associadas ao seu perfil que são utilizadas para melhor distribuir conteúdos na aplicação *HealthEducation*, tais como o tempo de tratamento e a doença do paciente. Com estas informações é possível enviar conteúdos educacionais específicos de acordo com o tempo de tratamento do paciente ou a doença pelo qual está sendo tratado. Os conteúdos das mídias educacionais a serem enviados para os que estão com 6 meses de tratamento, por exemplo, são diferentes dos conteúdos a serem enviados para os pacientes que estão com 24 meses de tratamento. Ou seja, ao longo de todo o tratamento, o conteúdo educacional disponibilizado para o paciente vai variando, porque para cada período de tempo dentro do tratamento há a necessidade de passar informações que correspondem ao estado corrente do mesmo. O mesmo ocorre para as doenças, em que cada paciente só terá acesso ao conteúdo educacional correspondente à doença pela qual está sendo tratado, devido a necessidade de enviar para o paciente mídias educacionais que expliquem como manipular um aparelho para auxiliar no seu tratamento, dentre outros casos.

Para obter todas as informações de um usuário o método `publishUserRequest()`, da classe `MOBHAUser`, é chamado publicando no tópico `UserRequest` um dos quatro dados seguintes, como `userName`, `userRequestedName`, `userRequestedId` e o `userRequestedEmail`. Na aplicação *HealthEducation*, geralmente é publicado no tópico somente o `userName`, que corresponde ao nome do usuário, para obter a informação do paciente ou do profissional da saúde. As informações virão através do tópico `User`, sendo necessário a subscrição da aplicação neste tópico para o recebimento do perfil do usuário e a implementação de um listener na aplicação que espere até o momento do tópico que foi publicado no lado do servidor, chegar até o cliente. A figura 4.15 mostra as informações do usuário que podem ser enviadas através do tópico `User`.

O atributo `userType` é utilizado para diferenciar para qual tipo de usuário pertence as informações que estão sendo enviadas pelo tópico, se ele é um paciente ou profissional da saúde. Caso ele seja um paciente, o atributo `disease` informa qual a doença que o paciente possui, e o atributo `treatmentStartDate` relata a data do início do tratamento do paciente auxiliando a aplicação solicitar as mídias relacionadas ao período correto do tratamento. Se o usuário for um profissional da saúde, o atributo `especialidadeName` informa qual a sua especialidade, permitindo a aplicação solicitar conteúdos relacionados a área médica do profissional da saúde em questão.

### 4.3.8 Utilizando o MOBHAContent para obter conteúdos educacionais

Quando uma mídia educacional é solicitada pelo usuário, a aplicação *HealthEducation* invoca o método `downloadContent()` da classe `MOBHAContent`, publicando no tópico `ContentD`

```
1 package br.ufma.lsd.mbhealthnet.communication.ddsttopics;
2 import java.io.Serializable;
3
4
5 // CoreDX DDL Generated code. Do not modify - modifications
6
7
8
9 public class User implements Serializable {
10
11     // instance variables
12     public String toUserName;
13     public String userName;
14     public String password;
15     public String name;
16     public String email;
17     public String rg;
18     public String cpf;
19     public String birthDate;
20     public String createDate;
21     public boolean secure;
22     public String userType;
23     public String especialidadeName;
24     public String careType;
25     public String disease;
26     public String projectId;
27     public String sexo;
28     public String treatmentStartDate;
29
```

Figura 4.15: Informações trafegadas através do tópico User

ownloadRequest o “id” do conteúdo que se deseja realizar o download para o dispositivo móvel. Para receber o conteúdo, é necessário se inscrever ao tópico ContentDownloadResponse e implementar um *listener* que possa ouvir a chegada deste tópico no lado do cliente. O *listener* também pode ouvir tópicos que tragam informações genéricas como, por exemplo, um aviso de que o upload de um conteúdo foi realizado com sucesso. É necessário também que a aplicação se inscreva no tópico ContentInfoByIdResponse para receber as informações relativas ao conteúdo. Para ouvir estas informações genéricas, é necessário que a aplicação esteja inscrita ao tópico GenericInformation.

As duas figuras seguintes mostram, respectivamente, como o *listener* é preparado no código para receber estes tópicos para que assim a aplicação *HealthEducation* possa tratá-los e o tópico ContentDownloadResponse.

Ao chegar no cliente, o tópico ContentDownloadResponse traz dentro do atributo contentFile um array de *bytes* do conteúdo solicitado caso ele exista<sup>1</sup>. Cada conteúdo no MobileHealthNet pode ter vários tipos de metadados que são definidos dependendo da aplicação. Na

<sup>1</sup>Visualizando o tópico ContentDownloadResponse percebe-se que o atributo contentFile é na verdade um tópico chamado ContentFile. Este possui um atributo onde fica armazenado a array de *bytes*.

```

78 // Handle topic responses
79 PubSubTopicListener listener = new PubSubTopicListener() {
80
81     @Override
82     public void processTopic(Object ddsTopic) {
83         if (ddsTopic.equals(GenericInformation.class)) {
84             processGenericResponse((GenericInformation) ddsTopic);
85
86         } else if (ddsTopic.equals(ContentDownloadResponse.class)) {
87             processContentDownloadResponse((ContentDownloadResponse) ddsTopic);
88
89         }
90     }
91 };
92

```

Figura 4.16: Exemplo de listener para ouvir a chegada dos tópicos no lado do cliente

```

public class ContentDownloadResponse implements Serializable {

    // instance variables
    public String toUserName;
    public String topicId;
    public ContentFile contentFile;

    // constructors
    public ContentDownloadResponse() {
    }

    public ContentDownloadResponse(String __f1, String __f2, ContentFile __f3) {
        toUserName = __f1;
        topicId = __f2;
        contentFile = __f3;
    }

    public void clear() {
        toUserName = null;
        topicId = null;
        contentFile = null;
    }

    public void copy(ContentDownloadResponse from) {
        this.toUserName = from.toUserName;
        this.topicId = from.topicId;
        this.contentFile = from.contentFile;
    }

}; // ContentDownloadResponse

```

Figura 4.17: Tópico ContentDownloadResponse

aplicação *HealthEducation* os metadados definidos são tipo de conteúdo, categoria em que a mídia se enquadra, descrição, comunidade alvo e público alvo. Além destes, são também incluídos os comentários atribuídos pelo usuário às mídias e a classificação (se bom, regular ou ruim) que o conteúdo recebe por sua qualidade.

O tópico `ContentDownloadResponse` não traz junto com a array de *bytes* os metadados do conteúdo, sendo necessário fazer uma publicação a um outro tópico, chamado de `ContentInfoById`, informando o id do conteúdo que se deseja receber os metadados e que a aplicação se

subscriva no tópico `ContentInfoByIdResponse` para recebê-los. O tópico `ContentInfoByIdResponse` possui um atributo `contentMetadata` referente ao tópico `ContentMetadata`, onde os metadados do conteúdo solicitado se encontram. As figuras a seguir mostram a estrutura do tópico `ContentInfoByIdResponse` e do `ContentMetadata`, respectivamente.

```
1 package br.ufma.lsd.mbhealthnet.communication.ddsttopics;
2 import java.io.Serializable;
3
4
5 // CoreDX DDL Generated code. Do not modify - modifications may be overwritten.
6
7
8
9 public class ContentInfoByIdResponse implements Serializable {
10
11     // instance variables
12     public String toUserName;
13     public String topicId;
14     public ContentMetadata contentMetadata;
15
16     // constructors
17     public ContentInfoByIdResponse() {}
18     public ContentInfoByIdResponse( String __f1, String __f2, ContentMetadata __f3 ) {
19         toUserName = __f1;
20         topicId = __f2;
21         contentMetadata = __f3;
22     }
23
24     public void clear() {
25         toUserName = null;
26         topicId = null;
27         contentMetadata = null;
28     }
29
30     public void copy( ContentInfoByIdResponse from ) {
31         this.toUserName = from.toUserName;
32         this.topicId = from.topicId;
33         this.contentMetadata = from.contentMetadata;
34     }
35
36 }; // ContentInfoByIdResponse
37
```

Figura 4.18: Tópico `ContentInfoByIdResponse`

```
public class ContentMetaData implements Serializable {  
  
    // instance variables  
    public String contentId;  
    public String path;  
    public String contentName;  
    public String mimeType;  
    public String owner;  
    public String[] applicationMetadaDataName;  
    public String[] applicationMetadaDataValue;  
  
    // constructors  
    public ContentMetaData() {  
    }  
  
    public ContentMetaData(String __f1, String __f2, String __f3, String __f4,  
        String __f5, String[] __f6, String[] __f7) {  
        contentId = __f1;  
        path = __f2;  
        contentName = __f3;  
        mimeType = __f4;  
        owner = __f5;  
        applicationMetadaDataName = __f6;  
        applicationMetadaDataValue = __f7;  
    }  
}
```

Figura 4.19: Tópico ContentMetadadata

Este capítulo veio abordar as características da aplicação, mostrando suas funcionalidades. Foi visível também os aspectos que envolveram a implementação da aplicação, incluindo os requisitos que orientaram no desenvolvimento da mesma, e as relações existentes entre as classes principais.

## 5 *Considerações finais*

O MobileHealthNet é um middleware projetado para prover mecanismos de comunicação em tempo real e com suporte à qualidade de serviço, bem como rigorosos requisitos de privacidade e segurança necessários a softwares voltados à área da saúde. Ele possui também um amplo conjunto de serviços para a interação dos usuários em redes sociais, contemplando mecanismos de comunicação tanto síncrona, quanto assíncrona.

Esta monografia teve como foco o desenvolvimento de uma das aplicações propostas para a avaliação do middleware MobileHealthNet, cujo objetivo é educar, no caso de pacientes, sobre a doença e seu tratamento que formam o contexto de saúde no qual o mesmo está inserido e, no caso dos profissionais da saúde, incrementar ou atualizar informações a respeito da sua especialidade através de mídias educacionais como vídeo, áudio e apresentações, utilizando dispositivos móveis. A motivação para a realização deste trabalho partiu da necessidade de tornar disponíveis informações de saúde além dos encontros presenciais, como nas palestras realizadas nos centros de tratamento pelos profissionais da saúde.

A aplicação HealthEducation permite a interação entre pacientes explorando o conceito das redes sociais. Isto acontece por meio das próprias mídias educativas em que cada paciente pode deixar seu comentário sobre o conteúdo, favorecendo a troca de opiniões entre eles podendo, assim, fornecer acréscimo de informação sobre o conteúdo educacional.

Envolver mais profundamente o paciente em seu tratamento através de palestras ou pela transmissão de recursos multimídia por um dispositivo móvel, como na aplicação HealthEducation, fornecendo informações sobre a sua doença e a sua forma de tratamento auxiliando na evolução positiva da saúde do paciente, já que permite sua participação mais ativa durante o tratamento.

Para os profissionais da saúde, informações relacionadas a sua área não somente incrementa o seu conhecimento como auxilia a guiar melhor um paciente em seu processo de tratamento.

Esperamos que o uso desta aplicação possa contribuir para que pacientes se tornem agentes, ou seja, juntamente com os profissionais da saúde colaborem com o tratamento que estão recebendo através de informações relacionadas a cuidados da saúde e, além disso, favorecer outros pacientes que recebam o mesmo tratamento com estas informações compartilhadas.

---

Quanto aos profissionais da saúde, esperamos que possam compartilhar seus conhecimentos através de recursos disponibilizados pela aplicação, permitindo que outros profissionais incrementem seus conhecimentos em torno de suas especialidades.

## **5.1 Trabalhos futuros**

Uma das funcionalidades a serem acrescentadas neste trabalho nas próximas etapas de desenvolvimento é um sistema de recomendação, não baseado somente na doença e no tempo de tratamento do paciente, ou na especialidade do profissional da saúde, como ocorre hoje, mas a avaliação do gosto do usuário em relação às mídias apresentadas. Mesmo na forma como as mídias são buscadas atualmente pela aplicação, através de metadados, existem conteúdos que não agradam o usuário, recebendo baixa avaliação. A avaliação do conteúdo por estrelas, deverá ser aproveitado como um dos quesitos par a geração de recomendação de conteúdos.

Uma outra funcionalidade que pode ser acrescentada é a possibilidade de executar as mídias por stream. A aplicação atualmente só permite a execução das mídias se elas estiverem salvas em um cartão de memória. Ou seja, para exibir algum conteúdo, primeiro é feito o download do arquivo, depois ele é salvo no cartão de memória e, por fim, lançado no reprodutor de mídias.

## *Referências Bibliográficas*

- [Adelstein et al. 2005] Adelstein, F., Gupta, S. K. S., Schwiebert, L., and Richard, G. G. (2005). *Fundamentals of Mobile and Pervasive Computing*. McGraw-Hill Companies, 1st edition.
- [Batista 2012] Batista, R. (2012). Infraestrutura de comunicação para o middleware mobilehealthnet. Master's thesis, Universidade Federal do Maranhão.
- [da Saúde 2006] da Saúde, M. (2006). *Política Nacional de Atenção Básica*. Editora do Ministério da Saúde, 4a edição edition.
- [Demiris 2006] Demiris, G. (2006). The diffusion of virtual communities in health care: concepts and challenges. *Patient Education and Counseling*, 62(2):178–188.
- [Gao et al. 2011] Gao, H., Hu, J., Huang, T., Wang, J., and Chen, Y. (2011). Security issues in online social networks. *Internet Computing, IEEE*, 15(4):56–63.
- [Gomes 2011] Gomes, F.J.L.; Souza, M. A. R. (2011). *Desenvolvimento de aplicações para a plataforma Google Android*.
- [Google 2013] Google (2013). Google android. disponível em: <http://developer.android.com/guide/>. acessado em: 14/07/13.
- [Group 2001] Group, O. M. (2001). Data distribution service for real-time systems specification. disponível em: <http://www.omg.org/spec/dds/1.2/pdf/>. acessado em: 10/05/2013.
- [Ilioudis 2010] Ilioudis, C.; Fevgas, A. T. K. M. K. D. Z. (2010). An innovative e-health network for collaboration on emergency cases. In *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on*, pages 1–4.
- [Jain 2011] Jain, R.; Sonnen, D. (2011). Social life networks. In *IT Professional*, volume 13, pages 8 – 11.
- [Kaplan and Haenlein 2010] Kaplan, A. M. and Haenlein, M. (2010). Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53(1):59 – 68.
- [Li-Der Chou and Shin 2011] Li-Der Chou, Nien-Hwa Lai, Y.-J. C. Y.-J. C. J.-Y. Y. L.-F. H. W.-L. C. H.-Y. C. and Shin, H.-Y. (2011). Mobile social network services for families with children with developmental disabilities. In *IEEE Transactions on Information Technology in Biomedicine*, pages 585–593.
- [Lin and Kede 2011] Lin, X. and Kede, Q. (2011). Embeddedness, social network theory and social capital theory: Antecedents and consequence. In *Management and Service Science (MASS), 2011 International Conference on*, pages 1–5.

- [Matthews 2012] Matthews, Jerrid; Kulkarni, R. G. M. M.-T. (2012). Rapid dengue and outbreak detection with mobile systems and social networks. pages 178–191.
- [Mendes and da Silva e Silva 2013] Mendes, T. S. and da Silva e Silva, F. J. (2013). Desenvolvimento de aplicações colaborativas móveis voltadas ao domínio da saúde. In *Cadernos de Pesquisa (UFMA)*, volume 20.
- [MobiHealth 2007] MobiHealth (2007). Putting care in motion.
- [Moushumi Sharmin 2006] Moushumi Sharmin, Shameen Ahmed, S. I. A. M. M. H. A. J. K. (2006). Healthcare aide: Towards a virtual assistant for doctors using pervasive middleware. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 6 pp.–495.
- [Qualman 2011] Qualman, E. (2011). *Socialnomics: How social media transforms the way we live and do business*. John Wiley & Sons.
- [Rahman et al. 2010] Rahman, M., Gueaieb, W., and El Saddik, A. (2010). Ubiquitous social network stack for e-health applications. In *Medical Measurements and Applications Proceedings (MeMeA), 2010 IEEE International Workshop on*, pages 57 –62.
- [Rana et al. 2009] Rana, J., Kristiansson, J., Hallberg, J., and Synnes, K. (2009). An architecture for social mobile networking applications. In *2009 First International Conference on Computational Intelligence, Communication Systems and Networks*, pages 241 – 246.
- [Shibata and Sasaki 2004] Shibata, Y. and Sasaki, Y. (2004). Remote healthcare education based on synchronous and asynchronous high quality video application. In *Applications and the Internet Workshops, 2004. SAINT 2004 Workshops. 2004 International Symposium on*, pages 199 – 205.
- [Steele 2011] Steele, R. (2011). Social media, mobile devices and sensors: Categorizing new techniques for health communication. *Sensing Technology (ICST), 2011 Fifth International Conference on*, pages 187 – 192.
- [Teles et al. 2012] Teles, A. S., Pinheiro, D. N., Gonçalves, J. F., Batista, R., and da Silva e Silva, F. J. (2012). Mobilehealthnet: A middleware for mobile social networks in m-health. In *Proceedings of the 3rd International Conference on Wireless Mobile Communication and Healthcare (MobiHealth 2012)*.
- [Teles et al. 2013] Teles, A. S., Pinheiro, D. N., Gonçalves, J. F., Batista, R., da Silva e Silva, F. J., Pinheiro, V., and Endler, M. (2013). Redes sociais móveis: Conceitos, aplicações e aspectos de segurança e privacidade. In *31º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.
- [Varshney 2009] Varshney, U. (2009). *Pervasive Healthcare Computing: EMR/EHR, Wireless and Health Monitoring*. Springer Publishing Company, 1st edition.
- [Vega 2008] Vega, J. M. L. (2008). Plataforma de Trabajo Colaborativo sobre Middleware DDS. Master’s thesis, Universidad de Granada, Spain.
- [Wac 2009] Wac, K.; Bults, R. v. B. B. W. I. J. V. K. D. V. H. M. H. H. (2009). Mobile patient monitoring: The mobihealth system. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 1238–1241.

- [Xiong M. 2010] Xiong M., Parsons, J. E. J. N. H. S. D. C. (2010). Evaluating the performance of publish/subscribe platforms for information management in distributed real-time and embedded systems.
- [Yu and Siddiqui 2009] Yu, W. D. and Siddiqui, A. (2009). Towards a wireless mobile social network system design in healthcare. In *MUE '09 Proceedings of the 2009 Third International Conference on Multimedia and Ubiquitous Engineering*, pages 429–436.