



**Plataforma para *m-Health* baseada no Padrão OpenEHR,
em Comunicações M2M e em Computação em Nuvem.**

JESÚS NOEL SUÁREZ RUBÍ

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**Plataforma para *m-Health* baseada no Padrão OpenEHR,
em Comunicações M2M e em Computação em Nuvem.**

JESÚS NOEL SUÁREZ RUBÍ

ORIENTADOR: PAULO ROBERTO DE LIRA GONDIM

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA
ELÉTRICA**

PUBLICAÇÃO: PPGEE.DM-637/2016

BRASÍLIA/DF: julho de 2016

UNIVERSIDADE DE BRASÍLIA

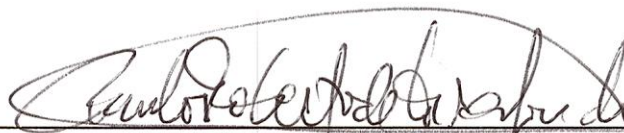
**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**PLATAFORMA PARA M-HEALTH BASEADA NO PADRÃO
OPENHR, EM COMUNICAÇÕES M2M E EM COMPUTAÇÃO EM
NUVEM**

JESUS NOEL SUÁREZ RUBÍ

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.


APROVADA POR:



PAULO ROBERTO DE LIRA GONDIM, Dr., ENE/UNB
(ORIENTADOR)



PAULO HENRIQUE PORTELA DE CARVALHO, Dr., ENE/UNB
(EXAMINADOR INTERNO)



CARLOS HUMBERTO LLANOS QUINTERO, Dr., ENM/UNB
(EXAMINADOR EXTERNO)

Brasília, 22 de julho de 2016.

FICHA CATALOGRÁFICA

RUBÍ, JESÚS NOEL SUÁREZ

Plataforma para m-Health baseada no Padrão OpenEHR, em Comunicações M2M e em Computação em Nuvem [Distrito Federal] 2016.

xviii, 123p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2016).

Dissertação de Mestrado – Universidade de Brasília.

Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. *m-Health*

2. *OpenEHR*

3. Computação em Nuvem

4. Arduino

5. Comunicação M2M

6. Apnéia

REFERÊNCIA BIBLIOGRÁFICA

RUBÍ, J.N.S. (2016). Plataforma para m-Health baseada no Padrão OpenEHR, em Comunicações M2M e em Computação em Nuvem. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGCEEDM – 637/2016, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 123p.

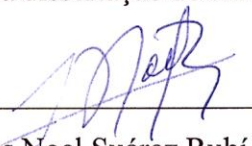
CESSÃO DE DIREITOS

AUTOR: Jesús Noel Suárez Rubí.

TÍTULO: “Plataforma para *m-Health* baseada no Padrão OpenEHR, em Comunicações M2M e em Computação em Nuvem”.

GRAU/ANO: Mestre/2016

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.



Jesús Noel Suárez Rubí

UNIVERSIDADE DE BRASÍLIA

Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

70910-900 – Brasília – DF – Brasil.

Dedicado a mi esposa Yarisley, a mis padres y familia, por el amor y el apoyo que me han brindado siempre, por la confianza que tuvieron en mi, por saber soportar mis momentos de estrés y alentarme a seguir hasta el final. Con todo mi amor para ellos.

AGRADECIMENTOS

Agradeço a Deus por ter me dado a oportunidade de cumprir o sonho de ser mestre.

A minha esposa por estar ao meu lado, pela sua compreensão, pela sua ajuda, pela sua confiança em mim e pelo seu amor.

A meus pais, minha sogra e resto da família em Cuba por ter me apoiado neste tempo que tenho estado longe deles, mas mesmo estando longe me fizeram sentir que estavam por perto, me cuidando e me acompanhando.

Ao meu orientador, pela sua paciência, pelo apoio, por ter me recebido como seu orientando e ter me guiado no desenvolvimento do meu projeto de mestrado, Prof. Paulo muito obrigado.

Às pessoas que conheci em Brasília e às que conviveram comigo no dia a dia fazendo me sentir como em família com elas. Obrigada Ruben, Oisys, Elier, Matilde pelo seu apoio, carinho e bons conselhos.

Ao pessoal do Laboratório de Redes da UnB e ao Programa de Pós-graduação em Engenharia Elétrica da UnB.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro.

A todos os que de algum jeito se interessaram pelo sucesso deste trabalho.

Muito obrigado, Noel.

ABSTRACT:

Platform for m-Health, based on OpenEHR Standard, M2M Communications and Cloud Computing.

Recent developments of mobile communications and Internet as well as its subsequent dissemination in different sectors of the society has favoured the beginning of new services, accessible from mobile terminals, for the furnishment of medical information, monitoring, vigilance and health treatment.

Monitoring of physiological parameters by means of electronic equipment has allowed health professionals conducting diagnostics of different types of diseases, either in person in hospitals, laboratories and specialized clinics, as well as through telemonitoring.

On the other hand, in terms of public policy, it turns out that there are important initiatives, such as the e-SUS, the electronic patient record and the possible adoption of the OpenEHR (Open Electronic Health Record) in Brazil. These initiatives go towards a progressive implementation of e-Health in our SUS (Unified Health System) and should be considered for effective computerization of health services in our country.

This paper proposes the development of a computational platform that serves as the basis for the creation of e-Health/m-Health software applications, having as premise the adoption of standards like OpenEHR, as well as the support of M2M Communications and Cloud computing. Additionally, some applications will be built, as examples of the use of the platform, including an application dedicated to the monitoring of parameters related to sleep apnea.

As motivation for the work focused on sleep apnea, it turns out that this disease is associated with increased risk of several others, such as hypertension, myocardial infarction, stroke and diabetes, in addition to producing reflexes that include behavioral (e.g. irritability) and cognitive (e.g. difficulty to learn) changes. There is also the risk of accidents (e.g. driving vehicles), as well as economic impacts, due to the reduction of productivity. For such a disease, the possibility of remote monitoring (e.g. at home patient), followed by the transmission of measured parameters and of screening and monitoring of patients by doctors and healthcare entities, allows to leverage public policies capable of reducing the impacts already mentioned, in addition to reducing costs of patient displacement.

Some impacts of the proposed work can be highlighted: economic (e.g. cost reduction), social (e.g. better service to the population, access to health care), academic (production of articles and new technologies), as well as better support for clinical information and decision support.

Finally, it is proposed an integration of concepts in the areas of e-Health/m-Health, Internet of things, M2M Communications, Cloud Computing and its integration with concepts such as Electronic Health Records, as well as service-oriented architectures based on REST, HTTP Polling, HTTP Long Polling and Websocket.

RESUMO

Plataforma para *m-Health* baseada no Padrão OpenEHR, em Comunicações M2M e em Computação em Nuvem.

Os recentes desenvolvimentos das comunicações móveis e da Internet e sua subsequente disseminação nos mais diversos setores da sociedade têm favorecido o surgimento de novos serviços, acessíveis a partir de terminais móveis, para o provimento de informações médicas, monitoramento, vigilância e tratamento de saúde.

O monitoramento de parâmetros fisiológicos por meio de equipamentos eletrônicos tem permitido aos profissionais da saúde a realização de diagnósticos de diferentes tipos de doenças, tanto de forma presencial em hospitais, laboratórios e clínicas especializadas, como também por meio de telemonitoramento.

Por outro lado, em termos de políticas públicas, verifica-se que há iniciativas importantes, como o e-SUS, o Prontuário Eletrônico do Paciente e a possível adoção do OpenEHR (*Open Electronic Health Record*) no Brasil. Tais iniciativas caminham na direção de uma progressiva implantação de *e-Health* em nosso SUS (Sistema Único de Saúde) e devem ser consideradas para a efetiva informatização dos serviços de saúde em nosso país.

Este trabalho propõe o desenvolvimento de uma plataforma computacional que sirva como base para a criação de aplicativos de *e-Health/m-Health*, tendo como premissas a adoção de padrões como o OpenEHR, bem como o suporte de Comunicações M2M e de Computação em Nuvem. Adicionalmente, algumas aplicações serão construídas, como exemplos do emprego da plataforma, incluindo uma voltada para o monitoramento de parâmetros relativos à apnéia do sono.

Como motivação para o trabalho voltado para a apnéia do sono, verifica-se que essa enfermidade é associada ao aumento do risco de diversas outras, tais como hipertensão, infarto do miocárdio, derrame e diabetes, além de produzir reflexos que incluem alterações comportamentais (p.ex. irritabilidade) e cognitivas (p.ex. dificuldade de aprender). Existe também o risco de acidentes (p.ex. na direção de veículos), bem como impactos econômicos, em razão da redução da produtividade. Para tal enfermidade, a possibilidade de monitoramento remoto (p.ex. na casa do paciente), seguida da transmissão de parâmetros medidos e de triagem e acompanhamento de pacientes por médicos e entidades da área de saúde permite alavancar políticas públicas capazes de reduzir os impactos já mencionados, além de reduzir custos de deslocamento do paciente.

Alguns impactos do trabalho proposto podem ser destacados: de natureza econômica (p.ex. redução de custos), social (p.ex. melhor atendimento à população, acesso aos cuidados de saúde), acadêmico (produção de artigos e de novas tecnologias), além de melhor suporte à informação clínica e apoio à decisão.

Finalmente, é proposta uma integração de conceitos das áreas de *e-Health/m-Health*, Internet das coisas, Comunicação M2M, Computação em Nuvem e sua integração com conceitos como Registros Eletrônicos de Saúde, bem como com arquiteturas orientadas a serviços baseados em REST, HTTP *Polling*, HTTP *Long Polling* e *Websocket*.

SUMÁRIO

SIGLAS.....	xiii
FIGURAS.....	xv
LISTA DE TABELAS.....	xviii
1 - INTRODUÇÃO.....	1
1.1 - MOTIVAÇÃO.....	1
1.2 - OBJETIVOS.....	4
1.2.1 - Objetivo Geral.....	4
1.2.2 - Objetivos Específicos.....	4
1.3 - METODOLOGIA.....	5
1.4 - CONTRIBUIÇÕES.....	6
1.5 - ORGANIZAÇÃO.....	6
2 - TECNOLOGIAS RELEVANTES E CONCEITOS BÁSICOS.....	8
2.1 - CUIDADO DA SAÚDE COM MEIOS ELETRÔNICOS.....	8
2.1.1 - e-Health.....	8
2.1.2 - Telemedicina.....	9
2.1.3 - m-Health.....	10
2.2 - PRINCIPAIS CONCEITOS DE TECNOLOGIAS DA INFORMAÇÃO.....	11
2.2.1 - Redes de Sensores Sem Fio.....	12
2.2.2 - Comunicação Máquina a Máquina.....	12
2.2.3 - Computação em Nuvem.....	14
2.2.4 - Arquitetura Orientada a Serviços (SOA).....	19
2.2.5 - REST-Based Web Services.....	21
2.2.6 - HTTP Polling.....	22
2.2.7 - HTTP Long Polling.....	22

2.2.8 - Websockets.....	23
2.3 - ASPECTOS LIGADOS À PADRONIZAÇÃO/NORMATIZAÇÃO EM <i>E-HEALTH</i>	23
2.3.1 - Prontuário Eletrônico do Paciente (PEP)	24
2.3.2 - OpenEHR - Conceitos e Definições	25
2.3.3 - Utilização do padrão OpenEHR no Brasil.....	27
2.3.4 - HL7.....	28
2.3.5 - TR 102 732 V1.1.1 M2M ETSI.....	28
2.4 - CONSIDERAÇÕES FINAIS.	30
3 - TRABALHOS RELACIONADOS	31
3.1 - TRABALHOS REFERENTES A PLATAFORMA DE TELEMONTORAMENTO.....	31
3.1.1 - e-Health Sensor Platform da Cooking Hacks.[22]	42
3.2 - TRABALHOS FOCADOS EM PADRONIZAÇÃO.....	44
3.3 - TRABALHOS FOCADOS EM APNÉIA DO SONO	46
3.4 - CONSIDERAÇÕES FINAIS	52
4 - DEFINIÇÃO DE REQUISITOS E DESENVOLVIMENTO DA PLATAFORMA	54
4.1 - REQUISITOS DA PLATAFORMA.....	54
4.1.1 - Requisitos de Alto Nível	54
4.1.2 - Requisitos Funcionais.....	55
4.1.3 - Requisitos não Funcionais.....	56
4.1.4 - Aspectos Legais.....	56
4.2 - ARQUITETURA ALVO E DIAGRAMA DE IMPLANTAÇÃO.....	57
4.2.1 - Arquitetura Alvo.....	57
4.2.2 - Diagrama de Implantação	60

4.3 - SISTEMA DE CAPTURA E GATEWAY M2M	61
4.3.1 - Problemas na escalabilidade do sistema de captura	62
4.3.2 - Dispositivo M2M.....	65
4.3.3 - GATEWAY M2M.....	69
4.4 - NETWORK SERVICE CAPABILITY PROVIDER.....	72
4.4.1 - Comunicação entre o gateway M2M e o NSCL.....	73
4.4.2 - HTTP Polling NSCL	74
4.4.3 - HTTP Long Polling NSCL.....	75
4.4.4 - WebSocket NSCL	76
4.5 - APLICATIVOS M2M.....	77
4.5.1 - Aplicativo gateway M2M.....	77
4.5.2 - Aplicativo M2M do lado médico	78
4.6 - CONSIDERAÇÕES FINAIS	81
5 - DEFINIÇÃO DE REQUISITOS E DESENVOLVIMENTO DO PROTÓTIPO PARA APNÉIA	83
5.1 - REQUISITOS DA APLICAÇÃO	83
5.1.1 - Requisitos Funcionais.....	84
5.1.2 - Requisitos não Funcionais.....	84
5.2 - ARQUITETURA ALVO E DIAGRAMA DE IMPLANTAÇÃO.....	84
5.3 - SISTEMA DE CAPTURA E GATEWAY M2M	85
5.3.1 - Dispositivo M2M.....	85
5.3.2 - Gateway M2M.....	89
5.4 - NETWORK SERVICE CAPABILITY PROVIDER.....	89
5.5 - CONSIDERAÇÕES FINAIS	89
6 - AVALIAÇÃO DE DESEMPENHO DA PLATAFORMA.....	91
6.1 - DESCRIÇÃO DOS TESTES	91

6.2 - TESTETESTES COM UM SÓ FLUXO.....	94
6.2.1 - Resultados para cenário rede DOCSIS.....	95
6.2.2 - Resultados para cenário rede 3G/4G.....	98
6.3 - TESTE COM OS TRÊS FLUXOS.....	100
6.3.1 - Resultados para cenário de rede DOCSIS.....	101
6.3.2 - Resultados para cenário rede 3G/4G.....	104
6.4 - ANÁLISE CONJUNTA DOS ATRASOS.....	106
6.5 - CONSIDERAÇÕES FINAIS.....	107
7 - CONCLUSÕES E TRABALHOS FUTUROS.....	109
REFERÊNCIAS BIBLIOGRÁFICAS.....	111
APÊNDICE I - Carta de Aceitação de artigo no evento 7th International Conference on Information and Multimedia Technology (ICIMT 2015).....	116
APÊNDICE II – Artigo “OpenEHR Semantics as a Tool for Reusability in M2M E-Health Devices.”, submetido no evento (ICIMT 2015).....	117

SIGLAS

AES	<i>Advanced Encryption Standard</i>
ANS	Agência Nacional de Saúde Suplementar
ANSI	<i>American National Standards Institute</i>
API	<i>Application Programming Interface</i>
ADL	<i>Archetype Definition Language</i>
CKM	<i>Clinical Knowledge Manager</i>
CFM	Conselho Federal de Medicina
e-SUS	<i>Electronic-Sistema Único de Saúde</i>
e-Health	<i>Electronic Health</i>
EHR	<i>Electronic Health Record</i>
ECG	Eletrcardiograma
EEG	Eletrencefalograma
ETSI	<i>European Telecommunications Standard Institute</i>
XML	<i>eXtensible Markup Language</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile Communications</i> (GSM: originalmente, <i>Groupe Special Mobile</i>)
HL7	<i>Health Level 7</i>
HRV	<i>Heart Rate Variability</i>
HIS	<i>Hospital Information System</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
ISO	<i>International Organization for Standardization</i>
IoT	<i>Internet of Things</i>
JSON	<i>JavaScript Object Notation</i>
M2MDS	<i>M2M Device Sensors</i>
M2MSG	<i>M2M Smartphone Gateway</i>
M2M	<i>Machine to Machine</i>
m-Health	<i>Mobile Health</i>
NSCL	<i>Network Service Capabilities Layer</i>

NO-REM	<i>No Rapid Eye Movement</i>
ORU	<i>Observation Request Unsolicited</i>
OpenEHR	<i>Open Electronic Health Record</i>
OSI	<i>Open System Interconnection</i>
POP	<i>Post Office Protocol</i>
PEP	Prontuário Eletrônico do Paciente
QoE	<i>Quality of Experience</i>
QoS	<i>Quality of Service</i>
REM	<i>Rapid Eye Movement</i>
RES	Registro Eletrônico de Saúde
RMD	<i>Remote Monitoring Device</i>
REST	<i>Representational State Transfer</i>
RTT	<i>Round Trip Time</i>
SpO2	Saturação Percutânea de Oxigênio
SOA	<i>Service Oriented Architecture</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SUS	Sistema Único de Saúde
SQL	<i>Structured Query Language</i>
TI	Tecnologia da Informação
TIC	Tecnologia de Informação e Comunicação
TCP	<i>Transmission Control Protocol</i>
TISS	Troca de Informações em Saúde Suplementar
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
WPA2	<i>Wi-Fi Protected Access 2</i>
WiFi	<i>Wireless Fidelity</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>

FIGURAS

Figura 1-1: Plataforma Proposta.....	4
Figura 2-1: Diagrama de representação conceitual.	11
Figura 2-2: Camadas da arquitetura M2M [7].....	13
Figura 2-3: Definição de funcionamento de computação em nuvem.....	15
Figura 2-4: Distribuição de camadas das especificações do OpenEHR.....	27
Figura 3-1: Plataforma para o monitoramento dos níveis de glicose.	32
Figura 3-2: Plataforma para o monitoramento de recém-nascidos.....	33
Figura 3-3: Plataforma para apoio à medição de pressão arterial em mulheres grávidas.....	34
Figura 3-4: Plataforma para assistência de pacientes em fases de reabilitação motora.	35
Figura 3-5: Plataforma para monitoramento de parâmetros fisiológicos e do contexto do paciente.....	36
Figura 3-6: Plataforma para monitoramento de parâmetros fisiológicos em doenças crônicas.....	37
Figura 3-7: Plataforma para monitoramento da diabetes baseada em smartphone.	38
Figura 3-8: Grupos de sensores disponíveis no <i>e-Health Sensor Platform</i>	43
Figura 3-9: Integração do <i>e-Health Shield</i> na placa Arduino UNO.	43
Figura 3-10: Imagem do grupo de sensores disponíveis na plataforma.	44
Figura 4-1: Arquitetura predominante em trabalhos voltados para <i>e-Health</i>	58
Figura 4-2: Arquitetura para um ambiente M2M.	59
Figura 4-3: Diagrama de Implantação/Distribuição.....	61
Figura 4-4: Comunicação entre o <i>AgentDevice</i> e o <i>AgentGateway</i>	65
Figura 4-5: Diagrama de classes no Dispositivo M2M.....	66
Figura 4-6: Diagrama de fluxo de desenvolvimento do Dispositivo M2M.	69

Figura 4-7: Diagrama de classes no Gateway M2M.	70
Figura 4-8: Diagrama de fluxo de desenvolvimento do Gateway M2M.....	72
Figura 4-9: Diagrama de fluxo de desenvolvimento do Gateway M2M.....	73
Figura 4-10: Resumo da Arquitetura empregada.	74
Figura 4-11: Comportamento da técnica de HTTP <i>Polling</i>	75
Figura 4-12: Técnica de HTTP Long <i>Polling</i>	76
Figura 4-13: Exemplo do comportamento da proposta baseada em <i>Websockets</i> ...	77
Figura 4-14: Tela de cadastro	78
Figura 4-15: Tela de autenticação	79
Figura 4-16: Tela de cadastro de novo paciente	79
Figura 4-17: Tela de gerenciamento de e-mails	80
Figura 4-18: Lista de Pacientes	80
Figura 4-19: Tela de seguimento dos dados em tempo real.	81
Figura 5-1: Diagrama de Implantação (apnéia do sono).	85
Figura 5-2: Sensores empregados e localização no paciente.	86
Figura 5-3: Arquétipos do OpenEHR empregados.	88
Figura 6-1: Atraso no cenário de rede DOCSIS, 1 fluxo.	95
Figura 6-2: Uso de CPU na nuvem.	96
Figura 6-3: Uso de memória RAM na nuvem.	96
Figura 6-4: Intervalos de confiança da média dos atrasos na transmissão de 1 fluxo na Rede DOCSIS.	98
Figura 6-5: Atraso no cenário operadora celular, 1 fluxo.	98
Figura 6-6: Uso de CPU na nuvem.	99
Figura 6-7: Intervalos de confiança da média dos atrasos na transmissão de 1 fluxo na Rede 3G/4G.	100
Figura 6-8: Atraso no cenário de rede DOCSIS, 3 fluxos.....	101

Figura 6-9: Intervalos de confiança da média dos atrasos na transmissão de 3 fluxos na Rede DOCSIS.....	102
Figura 6-10: Uso de CPU na nuvem.	103
Figura 6-11: Uso de memória RAM na nuvem.....	104
Figura 6-12: Atraso no cenário operadora celular, 3 fluxos.....	105
Figura 6-13: Intervalos de confiança da média dos atrasos na transmissão de 3 fluxos na Rede 3G/4G.	106

LISTA DE TABELAS

Tabela 2-1: Definições de telemedicina ao longo do tempo.	10
Tabela 3-1: Comparação das plataformas tratadas nos trabalhos relacionados.	41
Tabela 3-2: Comparação dos trabalhos referentes a apnéia do sono.....	52
Tabela 4-1: Funcionalidades do <i>AgentDevice</i>	66
Tabela 4-2: Funcionalidades do <i>CommunicationDevice</i>	67
Tabela 4-3: Funcionalidades do <i>BaseObservation</i>	67
Tabela 4-4: Funcionalidades do <i>SensorDriver</i>	68
Tabela 4-5: Funcionalidades do <i>AgentGateway</i>	70
Tabela 4-6: Funcionalidades do <i>CommunicationDevice</i>	71
Tabela 5-1: Apnéia do sono e o OpenEHR.	87
Tabela 6-1: Configuração dos experimentos.....	92
Tabela 6-2: Resultados das Métricas na Rede DOCSIS com 1 Fluxo	97
Tabela 6-3: Resultado das Métricas na Rede 3G/4G com 1 Fluxo	100
Tabela 6-4: Resultados das Métricas na Rede DOCSIS com 3 Fluxos.....	102
Tabela 6-5: Resultado das Métricas na Rede Celular com 3 Fluxos.....	105
Tabela 6-6: Percentil de superioridade e razão de atraso médio	107

1 - INTRODUÇÃO

Este capítulo pretende apresentar uma visão geral sobre a dissertação, assim como os motivos que promoveram o interesse no tema tratado. São apresentados também os objetivos gerais e específicos, além de aspectos de natureza metodológica, juntamente com as contribuições e as conclusões.

1.1 - MOTIVAÇÃO

A evolução tecnológica trazida pelas redes de comunicação e, em particular, pelo advento da Internet é, sem dúvidas, considerável. Assim, é possível fazer vídeo-chamadas, consultar a Web para obter informação, armazenar informação pessoal na nuvem (fotos, dados etc.), fazer compras *online*, agendamentos e transações bancárias a partir de terminais fixos e móveis como *tablets*, *smartphones* e PDAs.

A oferta desses serviços se dá por meio de provedores e companhias de telecomunicações que trabalham continuamente na melhoria da qualidade dos serviços oferecidos (QoS do inglês, *Quality of Services*) com o objetivo de melhorar a qualidade percebida pelos usuários (QoE do inglês, *Quality of Experience*). Do lado dos usuários, tem-se o crescimento constante de aplicações e do número de aparelhos ligados à Internet. Assim, se têm geladeiras, meios de transportes, roupas, relógios e diversos equipamentos “inteligentes” que, usando a Internet, permitem novos serviços inimagináveis alguns anos atrás.

Na área de saúde também se tem aproveitado essa evolução, por meio da qual o monitoramento constante com base em equipamentos eletrônicos permite aos profissionais da saúde realizar diagnósticos de diferentes tipos de doenças.

Observa-se, também, que a permanência do doente em um hospital ou clínica especializada impõe muitas vezes custos elevados, os quais podem ser minimizados por serviços de *home-care*, que podem ser complementados por meio de sistemas de telemonitoramento.

Neste sentido, muitas empresas desenvolveram arquiteturas de *hardware* e *software* para fornecer atendimento remoto a partir da residência do paciente ou com base em sistemas de telemonitoramento. Esses sistemas exigem uma grande variedade de sensores (para a captura dos dados) e diferentes mecanismos e arquiteturas de rede para enviá-los

aos EHRs [2] (*Electronics Health Records*), surgindo como primeiro grande desafio em *m-Health* o tratamento e o fornecimento remoto das informações através das tecnologias de comunicação e de redes.

Assim, quando dados e informações importantes são armazenados em bancos de dados e disponibilizados no momento certo e na granularidade adequada, os médicos podem obter um melhor diagnóstico e serviços de saúde de maior qualidade podem ser fornecidos.

Um EHR pode armazenar o histórico clínico de um paciente através de um identificador único, por exemplo, o ID nacional de saúde (Identificador único comumente associado a cada paciente numa instituição de saúde pública). Pode ser implementado localmente, dentro de uma única instituição, ou desenvolvido em uma estrutura mais complexa (um só país, ou a nível internacional). As principais vantagens são que os médicos tenham acesso facilitado à informação e que uma ampla gama de novas aplicações e arquiteturas possam ser criadas.

Em termos de histórico, cada instituição médica tinha inicialmente o seu próprio EHR baseado em diferentes arquiteturas de *software* e modelos de representação de informações. Porém, quando um usuário começava a ser atendido por uma nova instituição tinha que levar todo o seu histórico clínico, surgindo então os seguintes problemas:

- 1- Como representar os dados de modo que sejam compatíveis entre diferentes instituições;
- 2- Como transmitir esses dados entre as diferentes instituições.

Para resolver o primeiro problema, padrões como OpenEHR e EN ISO 13606 foram criados, padronizando a representação de dados de saúde.

Para resolver o segundo problema, outro conjunto de padrões conhecidos como HL7 (do inglês, *Health Level 7*) foi criado para o intercâmbio de mensagens entre os sistemas de saúde. No entanto, esses padrões não se concentram na interoperabilidade entre os sensores e equipamentos eletrônicos voltados para *e-Health/m-Health*.

O processo de envio das informações coletadas por sensores não é padronizado, portanto, cada fabricante usa seus protocolos proprietários, levando a que as plataformas

sejam desenvolvidas especificamente para um determinado conjunto de sensores. Quando um sensor muda, todo o sistema deve ser atualizado, o que leva a problemas de escalabilidade e reutilização. Por exemplo, sensores simples, tais como acelerômetros, glicômetros e oxímetros, entre outros, podem variar de um fabricante para outro, o que leva a pequenas alterações no sistema, enquanto sistemas de monitoramento mais complexos, como eletrocardiógrafos, eletroencefalógrafos, monitores de pressão arterial, que processam sinais de maior complexidade, levam a grandes mudanças e envolvem custos elevados podendo causar uma não-atualização do sistema. Isto provoca que sensores mais econômicos, etc., não possam ser integrados. Portanto, um mecanismo de integração padronizado, seguro e livre de integração de tais dispositivos com as plataformas existentes é necessário.

Novos paradigmas de rede e de computação ubíqua para melhorias na assistência médica remota também podem ser afetados por essa necessidade de interoperabilidade, que envolve a captura de dados em tempo real e o processo de armazenamento de dados (por exemplo, sinais vitais) relacionados aos pacientes.

Dentre esses novos paradigmas de rede e de computação voltados para a captura e transmissão de dados, destacam-se Internet das Coisas (IoT, do inglês, *Internet of Things*), Comunicação Máquina a Máquina e Computação em Nuvem aplicados a *e-Health*, podem tomar vantagens na padronização dos sistemas de captura. Além disso, uma integração dos conceitos anteriores pode resultar numa arquitetura de menores custos de implantação.

Este novo paradigma vem com a desvantagem de que não padronizam uma plataforma de desenvolvimento que permita a criação simples de novos aplicativos, que envolvam o tratamento de outras doenças e capazes de serem integrados a sistemas já existentes.

A Figura 1-1 resume a plataforma tratada neste trabalho, para a qual os conceitos de M2M e Computação em Nuvem são combinados com os padrões OpenEHR e HL7 (voltados para m-Health) visando cumprir com cada um dos objetivos apresentados a continuação.

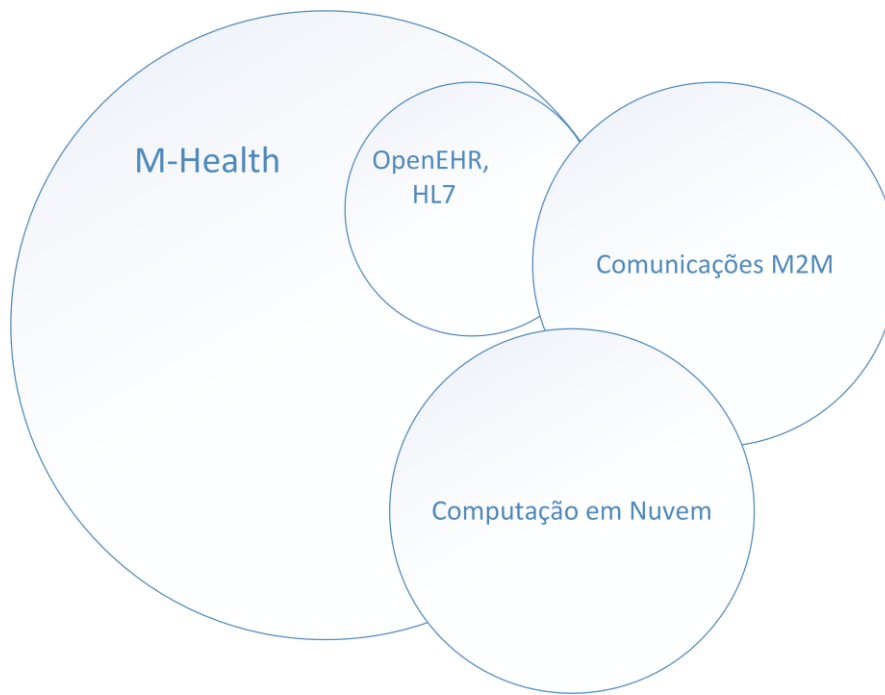


Figura 1-1: Plataforma Proposta.

1.2 - OBJETIVOS

1.2.1 - Objetivo Geral

Propor, desenvolver e avaliar uma plataforma de telemonitoramento aberta para favorecer a criação e emprego de aplicativos voltados para *e-Health/m-Health*.

1.2.2 - Objetivos Específicos

- Explorar as características e conceitos de TICs aplicados ao contexto considerado (p.ex., *e-Health*, *m-Health* e *Cloud-based e-Health*);
- Estudar e avaliar plataformas de *e-Health*;
- Estudar e selecionar padrões de integração e interoperabilidade entre sistemas voltados para *e-Health*;
- Propor e desenvolver uma plataforma que favoreça a criação de novos aplicativos de monitoramento baseados em *m-Health*;

- Prover telemonitoramento e acesso da equipe de saúde aos dados e parâmetros fisiológicos dos pacientes;
- Especificar sensores e demais componentes adequados à coleta dos parâmetros de interesse que exemplifiquem a integração e compatibilização com a plataforma.
- Implementar protótipo para avaliação da efetividade da plataforma proposta para a realização de triagem relativa à apnéia do sono;
- Avaliar o comportamento e o desempenho da plataforma em termos de métricas de rede.

1.3 - METODOLOGIA

A partir da definição do tema e de um problema de pesquisa, foi realizado o estudo de conceitos julgados importantes para o entendimento do tema, seguido de levantamento e revisão bibliográfica, que permitiu identificar propostas recentes, especialmente em termos de plataformas voltadas para *e-health*, com foco precípuo em monitoramento de parâmetros fisiológicos. Foram também verificados padrões de normatização de organizações como o ETSI e o HL7, visando assegurar interoperabilidade com sistemas tratados por aquelas entidades.

Trata-se aqui de um trabalho de pesquisa aplicada, para o qual foram utilizados dados obtidos de forma experimental, colhidos na transmissão em redes e ambientes reais e de laboratório.

Foram empregados conceitos e técnicas da engenharia de *software* para definição de proposta e implementação de uma nova plataforma para monitoramento remoto de parâmetros fisiológicos que integra padrões como OpenEHR e tecnologias recentes como Comunicação Máquina a Máquina e Computação em Nuvem.

A avaliação de desempenho da plataforma proposta foi tratada por meio de testes reais em duas redes de longa distância (Sem fio, baseada em padrões 3GPP e cabeada, baseada no padrão DOCSIS). Em ambos os casos foram analisados o atraso fim a fim e a perda de mensagens (com base em tendência central e dispersão), empregando três abordagens de sistemas “publish/subscribe” diferentes.

1.4 - CONTRIBUIÇÕES

As seguintes contribuições podem ser destacadas:

- Levantamento bibliográfico de características e conceitos de TIC aplicados ao contexto de *m-Health*;
- Levantamento bibliográfico e discussão sobre arquiteturas recentes voltadas para *e-Health*;
- Levantamento de requisitos, aplicações e conceitos relacionados a Engenharia de *Software* voltados para *e-Health*.
- Proposta de infraestrutura de *software* em termos de componentes e respectivas linguagens de programação.
- Proposta de plataforma de desenvolvimento aberta para a criação de aplicativos de monitoramento remoto de parâmetros fisiológicos;
- Avaliação da proposta em termos de métricas de rede;
- Disponibilização de uma implementação para o monitoramento remoto de parâmetros fisiológicos voltados à detecção da apnéia do sono;
- Artigo aceito em conferência internacional, e artigo submetido para periódico.

1.5 - ORGANIZAÇÃO

O Capítulo 2 apresenta conceitos importantes como *e-Health*, telemedicina, *m-Health*, entre outros voltados para o cuidado da saúde mediante equipamentos eletrônicos e sistemas de telecomunicações. Aborda também tecnologias como M2M, IoT, Computação em Nuvem, HTTP *Polling* e *Websocket*, cuja conceituação se faz necessária para o entendimento da proposta. Por fim é apresentado o padrão OpenEHR e alguns aspectos sobre sua adoção no Brasil.

No Capítulo 3 são apresentadas arquiteturas recentes voltadas para o telemonitoramento em sistemas de *e-Health*. Estas propostas são comparadas em termos de abrangência e de desempenho. Também inclui um levantamento bibliográfico referente à apnéia do sono, analisando diferentes propostas e ferramentas (algoritmos de detecção, sensores) que ajudam no diagnóstico desta doença.

O Capítulo 4 apresenta os requisitos da plataforma (Alto Nível, Funcionais, Não Funcionais) além de tratar algumas considerações legais acerca do emprego da plataforma. Também propõe uma arquitetura para padronização da plataforma, considerando diferentes padrões e a necessidade de atendimento aos requisitos elicitados anteriormente. São detalhados os aspectos referentes a engenharia de *software*, (representação dos dados, diagramas de fluxo), bem como equipamentos empregados, além de propor uma implementação de plataforma para criação de aplicativos *e-Health*.

No Capítulo 5 é desenvolvida uma plataforma voltada para a coleta de parâmetros fisiológicos referentes à apnéia do sono, permitindo a triagem de pacientes

No Capítulo 6 é feita uma avaliação da plataforma. Para atingir esse objetivo se definem métricas próprias como base para o desenvolvimento e execução de testes, por exemplo: tempo de retardo na entrega dos dados (“Round Trip Time RTT”) avaliando finalmente a utilidade das ferramentas desenvolvidas. É feita uma comparação entre alternativas de técnicas do tipo “publish/subscribe” com base em duas redes reais.

Finalmente, o Capítulo 7 resume os resultados obtidos levando a diferentes conclusões do trabalho desenvolvido e se propõem trabalhos futuros que brindam continuidade a esta dissertação.

2 - TECNOLOGIAS RELEVANTES E CONCEITOS BÁSICOS.

Este capítulo pretende apresentar tecnologias, conceitos básicos e as áreas de pesquisa englobadas nesta dissertação. Neste sentido, é feito um levantamento conceitual que serve como base para o cumprimento dos objetivos traçados neste trabalho.

2.1 - CUIDADO DA SAÚDE COM MEIOS ELETRÔNICOS.

Na atualidade praticamente toda a sociedade demanda comunicação ágil dado que, em todo momento, os indivíduos se encontram na troca constante de informações. Ante esta realidade, têm surgido novas tecnologias, ferramentas e aparelhos que facilitam, mediante sua implementação e uso, o fortalecimento da sociedade da informação. O setor da saúde não é alheio a esta realidade e tem identificado, nas tecnologias da informação e das telecomunicações, um aliado para aumentar a eficiência e melhorar a qualidade na prestação de serviços de saúde redundando em um maior bem-estar da população. Neste âmbito a seguir são introduzidos alguns dos principais conceitos relativos ao cuidado da saúde empregando meios eletrônicos.

2.1.1 - e-Health

Ao dia de hoje, é comum observar pesquisadores, diferentes companhias e profissionais da saúde, tratarem acerca do termo “e-Health” em cenários que envolvem o cuidado da saúde e equipamentos eletrônicos ou a Internet. Mas definir este conceito não é uma tarefa simples. Olhando alguns anos atrás se tem publicações [1] que avaliam o surgimento e o uso do termo ao redor do ano 1999, sendo usado pela primeira vez por líderes da indústria e pessoal de marketing. O termo parece ser uma derivação de outras palavras como “e-Commerce” que se refere ao comércio eletrônico, ou “e-Business”, que se refere aos negócios eletrônicos.

Em uma revisão sistemática da literatura [2], os autores conduzem a pesquisa fazendo uma revisão de mais de 400 trabalhos que incluem os termos “e-Health”, “eHealth” e “electronic health”. Como resultado, se encontraram as definições mais comuns, a saber:

- 1- “Atividades do cuidado da saúde baseadas na Internet”;
- 2- “Modelo de negócio usando tecnologias para assistir a provedores de serviços de saúde no cuidado dos pacientes e no oferecimento de serviços”;

- 3- “Campo genérico das tecnologias de comunicação e informação empregado em medicina e cuidado da saúde”.

Uma definição mais recente é apresentada em [1]:

“*e-Health* é um campo emergente na interseção das informáticas médicas, saúde pública e negócios, referindo-se aos serviços de saúde e entrega de informação através da Internet ou tecnologias semelhantes. Em termos de desenvolvimento, não é só tecnológico, mas também um novo estado da mente, um jeito de pensar, uma atitude, um compromisso para a rede, um pensamento global, para melhorar o cuidado da saúde, local, regional e global com o uso da informação e as tecnologias de comunicação”.

Como se pode apreciar, *e-Health* não é só um compromisso entre informáticos, eletrônicos, pesquisadores em geral e os médicos, porém, é um conceito que abarca diferentes entidades, centrada na melhoria da qualidade dos serviços de saúde oferecidos à sociedade.

O autor [1] também sugere alguns requisitos a serem atendidos por um sistema voltado para *e-Health* tais como: eficiência; melhoria da qualidade do cuidado da saúde; ser um sistema baseado em evidências para sua validação; interação entre os consumidores dos serviços de maneira confiável; de fácil uso.

2.1.2 - Telemedicina

Em determinados cenários o termo “Telemedicina” é intercambiado arbitrariamente e de maneira errada, com o termo “e-Health”, conforme a análises apresentada em [3]. Os autores fazem um estudo do termo telemedicina, baseado no emprego do mesmo em artigos publicados até a data da pesquisa. Como resultado, foram encontradas 104 definições diferentes para o termo e, também para o termo “e-health” com cada definição associada a um determinado contexto. A Tabela 2-1 mostra algumas destas definições:

Tabela 2-1: Definições de telemedicina ao longo do tempo.

Ano	Definição
1970	Prática da medicina, sem a confrontação usual entre o paciente e o médico, lograda através de sistemas de comunicação interativos baseados em áudio e vídeo
1993	Sistema de entrega de cuidados da saúde no qual os profissionais examinam pacientes distantes empregando tecnologias de telecomunicações.
2001	Telemedicina é a comunicação audiovisual interativa entre provedores de cuidado da saúde e seus pacientes, ignorando a distância geográfica.
2004	Usualmente se refere ao emprego de tecnologias baseadas em informação, como computadores e sistemas de comunicação, para prover cuidado da saúde através de distancias geográficas.

Com o uso da telemedicina, a sociedade experimenta diferentes vantagens nos serviços médicos, especialmente a diminuição dos custos de deslocamento e a possibilidade de realização de consultas a especialistas de forma remota.

Finalmente observa-se que telemedicina está mais enfocada na realização de consultas e diagnósticos remotos, enquanto *e-Health* encontra-se mais relacionado com o desenvolvimento de soluções e equipamentos que cuidem da saúde do usuário. Surge então a necessidade de contar com uma plataforma que o acesso dos médicos aos dados e parâmetros obtidos, em qualquer momento e em tempo real, estando os médicos ou profissionais da saúde em geral, afastados geograficamente.

2.1.3 - m-Health

m-Health ou “mobile-health” é um conceito que relaciona serviços de *e-Health* com configurações sem fio e móveis [4].

“O emprego da computação móvel e as tecnologias da informação nos cuidados da saúde e a saúde pública.” [5].

Um aplicativo que pertence à categoria das “m-health-applications” pertence também à área de *e-health* assim como à telemedicina. Tais aplicações comumente fazem parte de um sistema para cuidado da saúde, que permita o monitoramento de diferentes parâmetros desde qualquer lugar e de movimento, usando equipamentos voltados para *e-*

health e transmitindo, esses dados através de dispositivos de comunicação móvel. Esta última característica também vincula *m-health* com telemedicina pois ambos conceitos perseguem superar as barreiras de localização entre os pacientes e os médicos. Pode também ser definida como:

O diagrama exposto na Figura 2-1 representa a interação entre os conceitos tratados até este momento. A região hachurada corresponde ao conjunto de aplicações que provêm serviços de cuidado da saúde com aparelhos eletrônicos e que por sua vez permitem o acesso remoto dos profissionais a determinados dados obtidos no contexto do paciente. Pode-se observar que *m-Health* não cobre toda a região sublinhada. Isto se deve ao fato de que só os aplicativos de *e-Health* e Telemedicina que empreguem sistemas de comunicação móveis são atendidos pelo este conceito.

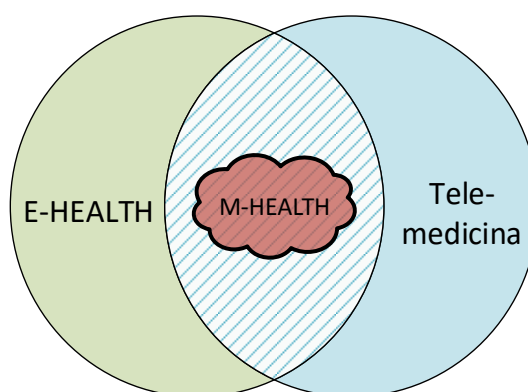


Figura 2-1: Diagrama de representação conceitual.

Conceitos tratados posteriormente neste trabalho serão incluídos no diagrama anterior em termos de áreas de pesquisas. O objetivo final é situar a implementação da plataforma no escopo acadêmico adequado.

2.2 - PRINCIPAIS CONCEITOS DE TECNOLOGIAS DA INFORMAÇÃO.

O conceito de Tecnologia da Informação refere-se ao uso de equipamentos de telecomunicações e computadores (ou *smartphones*, *tablets*) para a transmissão, o processamento e armazenamento de dados. O conceito abrange questões próprias da informática, eletrônica e telecomunicações.

Com uma abordagem não exaustiva, mas não incompleta, uma infraestrutura de TIC voltada para o domínio da saúde pode ser abordada em três grandes grupos. Tem-se então

os grupos que atendem às capacidades de: (i) processar - transformar dados, atuar em mecanismos que operam mudanças de estados, sensores e atuadores; (ii) armazenar - persistir, manter e recuperar dados; e (iii) comunicar - transportar de um ponto ao outro o processado e o armazenado. A seguir discutimos os principais conceitos utilizados nesta dissertação no âmbito de TIC que permitem a implementação/implantação dos grupos mencionados.

2.2.1 - Redes de Sensores Sem Fio

Uma rede de sensores sem fio é, tipicamente, uma rede de sensores interligados por meio de enlaces não cabeados. Sua aplicação no cuidado da saúde cresce continuamente em termos de aplicativos desenvolvidos, devendo atender a requerimentos tais como pouco peso, com capacidades de computo, monitoramento, de comunicação e de fácil colocação no paciente, sem interromper a rotina do mesmo.

2.2.2 - Comunicação Máquina a Máquina

Machine to Machine refere-se à comunicação entre dois dispositivos através de uma rede com/sem fio [6]. Os principais desafios deste tipo de rede consideram o gerenciamento remoto de dispositivos (i.e. um sistema de ar condicionado controlado remotamente por um smartphone) e prestação de serviços a um acesso remoto aos dados do dispositivo (i.e. acesso remoto aos dados de temperatura de uma habitação para controlar o funcionamento do sistema de ar condicionado). A principal característica é a possível interação de dois dispositivos compatíveis sem a presença humana. Um aplicativo dentro de uma arquitetura M2M geralmente consiste de um módulo de *hardware* instalado em um dispositivo cliente (*frontend*), que se comunica com o sistema de *back-end* que finalmente fornece serviços a clientes finais (Outros dispositivos ou humanos).

Numa arquitetura M2M, Figura 2-2, pode ser observado um comportamento vertical iniciado na camada inferior com o dispositivo M2M e terminando num aplicativo no topo da arquitetura.

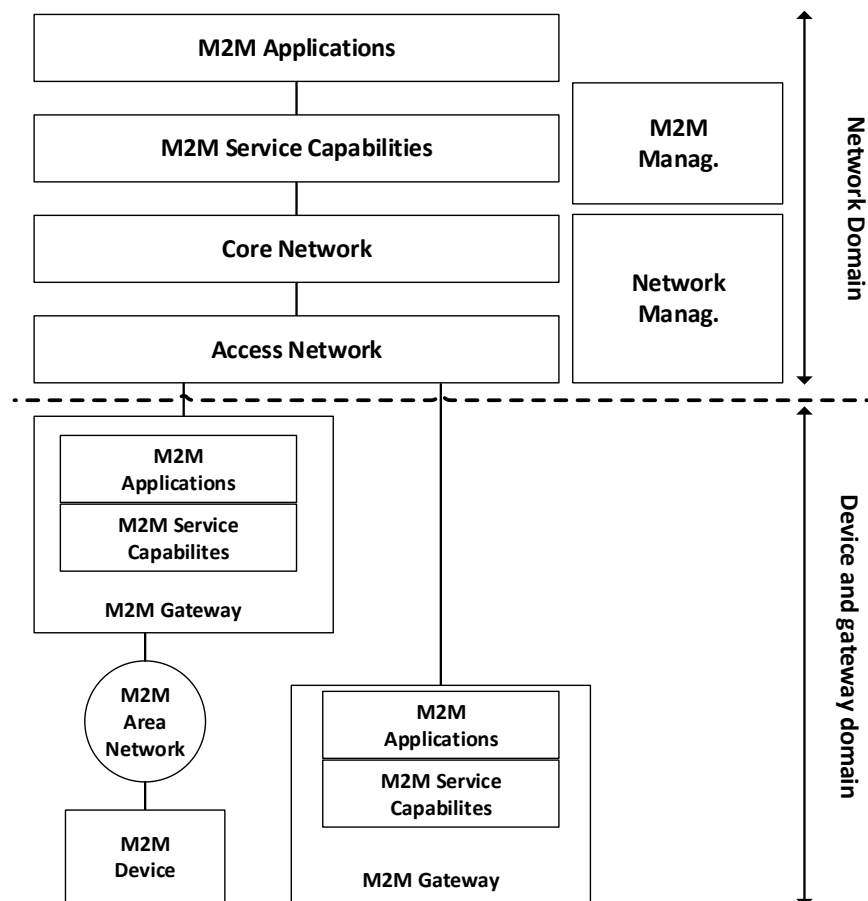


Figura 2-2: Camadas da arquitetura M2M [7].

Nesta arquitetura, os dispositivos M2M estão interligados ao Gateway através de uma rede conhecida como Rede de Área M2M. Esses dispositivos representam o extremo da arquitetura onde se realiza o processo de coleta de dados (p.ex. temperatura de habitações, dados referentes a processos em linhas de produção, ou qualquer tipo de dado alvo no sistema M2M considerado).

O Gateway M2M garante a comunicação com camadas superiores provendo, ao mesmo tempo, serviços para os dispositivos da rede M2M. Outra entidade a destacar é o Provedor de Capacidades de Serviços onde são distribuídos os diferentes serviços para os dispositivos M2M e que gerencia a interação entre os mesmos e os usuários finais do sistema (Seres Humanos ou Aplicativos M2M).

Detalhando cada um dos componentes:

Dispositivo M2M: Dispositivo capaz de responder a um pedido de dados contidos no mesmo dispositivo ou capaz de transmitir dados de forma autônoma. Sensores e

dispositivos de comunicação são os pontos finais de aplicações M2M. Geralmente, os dispositivos podem se conectar diretamente à rede de um operador, ou eles vão provavelmente interligar usando tecnologias WPAN como ZigBee ou Bluetooth.

Rede de Área M2M: Fornece conectividade entre Dispositivos M2M e Gateways M2M (p.ex. rede de área pessoal).

Gateway M2M: Equipamento que usa os recursos M2M para garantir interfuncionamento e interligação à rede de comunicação. Gateways e roteadores são os pontos finais da rede da operadora em cenários onde os sensores e dispositivos M2M não se conectam diretamente à rede. Assim, gateways e roteadores têm de garantir que os dispositivos da rede possam ser alcançados a partir do exterior e vice-versa. Conseqüentemente, haverá um fluxo de controle entre o gateway e da plataforma do operador que tem de ser distinguido do canal de dados que é a transferência de dados de aplicação M2M.

Rede de acesso: Rede através da qual o gateway M2M se conecta à Internet (p.ex. 3G, 4G, DOCSIS, WiFi).

Núcleo da Rede: Núcleo da Internet ou outra rede privada a qual interliga o Serviço de Capacidades M2M com o Gateway M2M.

Serviços de Capacidades M2M: Disponibiliza um conjunto de recursos a serem consumidos pelos Dispositivos ou Aplicativos M2M. Através desta entidade é que se obtém a troca de informação entre sensores e aplicativos (extremos na arquitetura), permitindo um processamento próprio da lógica de negócio estabelecida na arquitetura.

Aplicativo M2M: Contém a camada de *middleware* pela qual os dados atravessam diferentes serviços de aplicativos e é usado pelo motor de processamento próprio do negócio. Os aplicativos podem ser do usuário final (apresentando os resultados no *backend*) ou podem rodar nas diferentes camadas da arquitetura como processadores de dados, seguindo a lógica de negócio para a qual foi criada a arquitetura.

2.2.3 - Computação em Nuvem

Computação em Nuvem (do inglês, *Cloud Computing*) é definido pelo *National Institute of Standards and Technology* (NIST) [8] como um modelo que permite, de forma conveniente, o acesso à rede sob demanda a um conjunto compartilhado de recursos de

computação configuráveis (por exemplo, redes, servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e lançados com o mínimo de esforço de gestão ou a interação de um prestador de serviços.

Ainda segundo o NIST [8] o modelo de nuvem está composto por cinco características essenciais, três modelos de serviço e quatro modelos de desenvolvimento, como mostra a Figura 2-3.

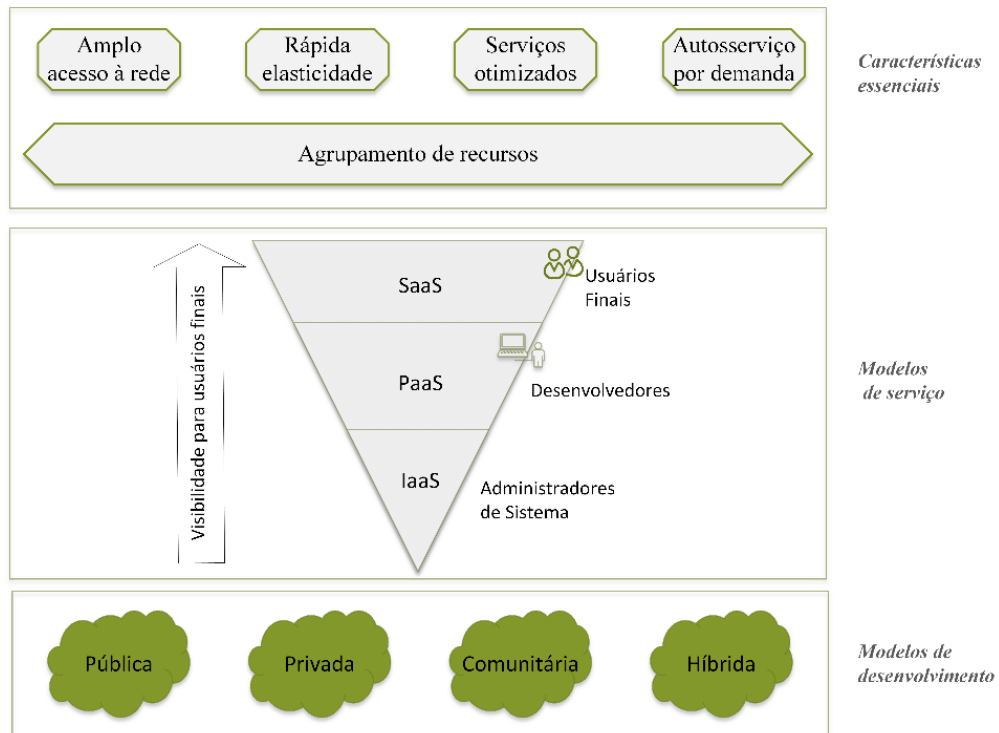


Figura 2-3: Definição de funcionamento de computação em nuvem.

Características essenciais:

- Autosserviço por demanda: o usuário pode adquirir unilateralmente recursos computacionais, como tempo de processamento no servidor ou armazenamento, através da rede, na medida em que necessite e sem precisar de interação humana com os provedores de cada serviço. Estes serviços podem ser encontrados junto de fornecedores como a *Google*, *Amazon* ou *Salesforce* entre outros.
- Amplio acesso à rede ou ubiquidade: Os recursos computacionais encontram-se disponíveis através da Internet e podem ser acedidos através de mecanismos padronizados, que possibilitem o uso por plataformas heterogêneas, tais como por

exemplo, telefones celulares, *tablets*, laptops, computadores pessoais ou outras tecnologias.

- Agrupamento de recursos: os recursos do provedor de computação são agrupados para atender vários consumidores através de um modelo *multi-tenancy*, com diferentes recursos físicos e virtuais atribuídos dinamicamente e de acordo com a demanda do consumidor. Há um senso de independência local em que o cliente geralmente não tem nenhum controle ou conhecimento sobre a localização exata dos recursos disponibilizados, mas pode ser capaz de especificar o local em um nível maior de abstração (por exemplo, país, estado ou data center). Exemplos de recursos incluem o armazenamento, processamento, memória, largura de banda de rede e máquinas virtuais.
- Rápida elasticidade: recursos podem ser adquiridos de forma rápida e elástica, em alguns casos automaticamente, caso haja a necessidade de escalar com o aumento da demanda e devem também ser liberados, na diminuição dessa demanda. Para os usuários, os recursos disponíveis para uso parecem ser ilimitados e podem ser adquiridos em qualquer quantidade e a qualquer momento. Deste modo, o usuário só paga aquilo que consome em cada momento, evitando assim custos desnecessários
- Serviços otimizados: sistemas em nuvem automaticamente controlam, monitoram e otimizam a utilização dos recursos, em algum nível de abstração adequado para o tipo de serviço (por exemplo, armazenamento, processamento, largura de banda e contas de usuários ativos). Este monitoramento pode ser controlado e reportado, mas de forma transparente tanto para o fornecedor como para o consumidor do serviço utilizado.

Modelos de serviço:

- Software como um Serviço (SaaS, do inglês, *Software as a Service*): é a última camada do modelo, onde o usuário tem acesso a aplicação disponibilizada na nuvem. Assim o modelo de serviço permite ter aplicações de interesse comum para um grande número de usuários, todas elas instaladas na infraestrutura da nuvem. As aplicações são acessíveis de vários dispositivos cliente através de uma interface *thin client*, como por exemplo um *browser Web*. Todo o controle e gestão da infraestrutura da rede, sistemas operativos, servidores e armazenamento são

efetuados pelo prestador do serviço, libertando o consumidor dessas tarefas. Mesmo assim ainda é possível definir algumas configurações específicas para o usuário na aplicação. Um bom exemplo de SaaS é a App do *Google*, onde aplicações como o *Gmail*, *Google Docs*, *Google Sites*, entre outras, são utilizadas por milhares ou milhões de usuário.

- Plataforma como um Serviço (PaaS, do inglês, *Platform as a Service*): a capacidade fornecida ao consumidor é a de realizar *deploy* de uma aplicação em uma infraestrutura pré-definida ou adquirir aplicações criadas usando linguagens de programação e as ferramentas suportadas pelo provedor de PaaS. O consumidor não administra ou controla a infraestrutura básica como rede, servidores, sistemas operacionais, ou armazenamento, mas tem controle sobre os aplicativos utilizados e eventualmente hospedagem de aplicativos e configurações de ambiente. Como exemplo temos a *AppEngine* do *Google* e a plataforma *Azure* da *Microsoft*.
- Infraestrutura como um Serviço (IaaS, do inglês, *Infrastructure as a Service*): a capacidade prevista para o consumidor é a de processamento, armazenamento, redes e outros recursos computacionais fundamentais para que o consumidor seja capaz de implantar e executar programas arbitrários, que podem incluir sistemas operacionais e aplicativos. O consumidor não administra ou controla a infraestrutura de nuvem subjacente, mas tem controle sobre os sistemas operacionais, armazenamento, aplicativos implantados, e, eventualmente, o controle limitado de componentes de rede (por exemplo, firewall).

Modelos de desenvolvimento:

- Nuvem Pública: a infraestrutura disponível e fornecida através de recursos compartilhados e com acesso aberto ao público em geral. Pode ser administrada por um setor empresarial, acadêmico ou organização governamental, ou alguma destas combinações e está localizada nas instalações do fornecedor.
- Nuvem Privada: Utiliza as características de nuvem pública, mas em um ambiente de rede privada, normalmente no *DataCenter* da empresa, fornecendo o acesso a apenas aos servidores da empresa, sendo assim segurança e privacidade o grande diferencial de nuvens privadas.

- Nuvem Comunitária: este modelo fornece uma infraestrutura compartilhada por uma comunidade de organizações com interesses em comum.
- Nuvem Híbrida: a infraestrutura de nuvem é uma composição de duas ou mais infraestruturas distintas (comunitária, privada ou pública), que permanecem em entidades únicas, mas são conectadas por tecnologia padronizada ou proprietária, que disponibilizam dados e portabilidade de aplicações. Com este modelo, a organização pode definir o que pretende colocar na nuvem pública e o que pretende manter mais reservadamente na nuvem privada, ou seja, aqueles dados que são de maior importância para o negócio, permitindo balancear entre custo e benefício.

Modelos baseados em nuvem oferecem várias vantagens sobre os modelos tradicionais de *software* local, sendo as principais: a redução nos ciclos de implementação e menor custo total de implantação, sem contar que os serviços de computação em nuvem geralmente são baseados em contratos de serviço, que eliminam a necessidade de altos investimentos iniciais em *hardware* e *software*. Além disso, a solução é “escalável”, ou seja, à medida que a organização cresce (ou decresce) o provimento de serviços pode ser aumentado ou reduzido sem impactos operacionais para a empresa.

Assim a computação em nuvem oferece muitas vantagens que podem ser aproveitadas para o setor de *e-Health* [9]. Dentre tais vantagens, sistemas de saúde com recursos limitados podem dispor de aplicativos de grande escala instalados na nuvem. Também a computação em nuvem deve facilitar, por exemplo, o crescimento das plataformas interoperáveis em Saúde (HIE, do inglês, *Health Information Exchange*), possibilitando uma sensível redução de custos e um aumento significativo de facilidades para a Cadeia Assistencial. A expansão da computação em nuvem deve incrementar consideravelmente a informatização no setor de Saúde. Da mesma forma, a informação centralizada em nuvem ajuda a detectar erros, avaliar diagnósticos equivocados, acompanhar o progresso do paciente e, acima de tudo, tomar decisões com mais rapidez e assertividade. Também médicos podem acessar os dados clínicos do paciente (através de um Registro Eletrônico de Saúde em nuvem) de qualquer lugar e compartilhá-los com seus pares a qualquer momento. Ganha o hospital (redução de custos), ganha o médico (produtividade) e certamente ganha o paciente, que recebe um atendimento mais ágil e seguro.

2.2.4 - *Arquitetura Orientada a Serviços (SOA)*

SOA (do inglês, *Service Oriented Architecture*) [10-11] faz referência a uma arquitetura baseada em serviços, na qual as funcionalidades implementadas em cada aplicação existente são disponibilizadas para as outras aplicações em forma de serviços. Estes serviços são conectados por um “barramento de serviços” que os disponibiliza por Web Services no caso de utilização da Internet, ou outra forma de comunicação.

Um projeto orientado a serviços transforma seus aplicativos e os de seus associados em peças tecnológicas de construção capazes de se encaixarem, virtualmente, em qualquer configuração. A única limitação real do que pode ser feito com essas peças reside na imaginação e perspectiva do seu construtor e não, como até agora, na tecnologia (sem a sua rigidez e incompatibilidade). SOA transforma a tecnologia em um instrumento ágil da estratégia do negócio.

Sumarizando SOA estabelece uma estrutura para a integração de aplicações independentes de modo que a partir da rede, suas funcionalidades podem ser acessadas sendo oferecidas como serviços. O modo mais comum para implementá-la é através de serviços Web, tecnologia baseada em padrões e independente da plataforma, com os quais SOA pode descompor aplicações monolíticas em um conjunto de serviços e implementar essa funcionalidade de uma forma modular.

Desta maneira, uma arquitetura SOA possibilita uma infraestrutura para computação distribuída, por meio de serviços que podem ser fornecidos e consumidos dentro de uma organização e entre organizações, por meio de redes de comunicação. Uma arquitetura SOA básica é caracterizada pelas interações entre três tipos de agentes de *software*: os provedores de serviço, os consumidores de serviço e os registro de serviço [12].

- **Características de uma arquitetura orientada a serviços [10],[12]:**

Baixo acoplamento: cada atividade é implementada como um serviço, ou seja, um componente independente que poderá ser utilizado quantas vezes for necessário em partes diversas do sistema. Assim, vemos que SOA é, essencialmente, uma arquitetura baseada em componentes, onde cada componente preserva a sua independência e interage, apenas, através de interfaces bem definidas.

Neutralidade de implementação: não há limitações em relação às tecnologias,

linguagens ou plataformas a serem utilizadas. O desenvolvedor deve, apenas, especificar adequadamente o que o seu serviço faz e informações como tipos de dados de entrada e saída. Assim, outros desenvolvedores poderão fazer o uso adequado deste serviço baseado nesta interface que foi especificada e que será a única parte realmente visível do serviço. Detalhes de implementação específicos de um componente (serviço) não são importantes para o restante do sistema e, por isso, não devem estar disponíveis.

Interoperabilidade: possibilidade de sistemas coexistir e comunicar-se independente de fabricantes ou tecnologias.

- **Benefícios de SOA [12-13]:**

Aplicações mais produtiva e flexíveis: a estratégia orientada a serviços permite alcançar uma maior produtividade dos recursos de TI existentes (aplicações e sistemas instalados ou mais) e obter maior valor dos mesmos para a organização.

Desenvolvimento de aplicações mais rápido e econômico: produto da integração flexível dos dados e a criação de um repositório de serviços utilizáveis que podem ser combinados em serviços de maior nível, se alcança uma aceleração no desenvolvimento de projetos de TI e uma diminuição nos custos do desenvolvimento de soluções e ciclos de prova, pois são eliminadas redundâncias e a sua execução é atingida em menos tempo.

Aplicativos mais seguros e gerenciáveis: SOA fornece uma infraestrutura e documentação comum para desenvolver serviços que são seguros, previsíveis e gerenciáveis, além de facilitar a possibilidade de adicionar novos serviços e recursos para o gerenciamento de processos críticos do negócio. Uma vez que o acesso é aos serviços e não aos aplicativos, SOA otimiza os investimentos feitos em TI, aumentando a capacidade de introduzir novos recursos e melhorias.

Minimização do risco de tempo de inatividade ou perda de dados: isto é devido a que proporciona desempenho, escalabilidade, segurança e alta disponibilidade sem precedentes.

Melhora a capacidade de inovar e diferenciar: através de dados abrangentes, precisos e oportunos para desenvolver novas funcionalidades rapidamente.

2.2.5 - *REST-Based Web Services*

REST (do inglês, *Representational State Transfer*) [14] é um estilo de arquitetura de *software* para a implementação de Web Services. O criador do REST define-o assim [15]:

“Conjunto coordenado de restrições arquitetônicas que tenta minimizar a latência e comunicação de rede e, ao mesmo tempo, maximizando a independência e a escalabilidade das implementações de componentes. REST permite o armazenamento em cache e reutilização de interações, substituíbilidade dinâmica de componentes e processamento de ações por intermediários, respondendo assim às necessidades de uma Internet escalável. ”

Algumas das razões pelas quais a arquitetura REST tem um alto grau de popularidade são sua simplicidade e facilidade de uso, bem como a compatibilidade devido ao emprego de protocolos nativos da Web, como HTTP.

Os princípios do REST são baseados em:

- conceito de recursos (um documento é um recurso);
- identificação única de cada recurso (a URL do documento);
- relações entre os recursos;
- uso de padrões (HTML, HTTP, XML);
- possíveis várias formas de um recurso;
- comunicação simples através do protocolo HTTP.

Existe uma tendência de não considerar REST como um serviço Web. No entanto, é uma arquitetura orientada a serviços (SOA) que usa o protocolo HTTP e primitivas de HTTP para acessar recursos. Quando um pedido é feito através da primitiva GET do protocolo HTTP, alguns dados específicos são solicitados diretamente. Assim, em REST, para consumir um serviço ou recurso só precisamos saber sua URL, uma vez que permite o acesso imediato a um serviço ou serviços disponíveis. A abordagem REST fornece acesso simples a serviços criados sob esse paradigma e objetos podem ser publicados ou recebidos em diferentes formatos, como:

- Texto simples
- XML
- JSON (*JavaScript Object Notation*)

No desenvolvimento de um serviço de Web REST, métodos HTTP comuns (GET, POST, PUT e DELETE) podem ser utilizados para a leitura, envio, atualização e exclusão de informações sobre o servidor.

2.2.6 - HTTP Polling

No início, os aplicativos Web foram criados num ambiente cliente servidor onde o navegador era que sempre iniciava a comunicação. O cliente fazia um *request* e o servidor dava uma resposta. Neste sentido, o servidor dependia de uma requisição por parte de um cliente para transmitir uma informação. Este fato afeta o processo de comunicação entre dois usuários que empregam o mesmo servidor para troca de informação. Por exemplo, um aplicativo de chat tem dois ou mais usuários em uma sala, cada vez que um deles encaminha uma mensagem para o grupo, o servidor tem que distribuir a mensagem para cada um dos usuários. Mas como fazer isso, se o servidor só responde a requisições de clientes? Neste sentido é que nasce o HTTP *Polling* [16]. Os clientes se mantêm fazendo requisições ao servidor ao longo da seção de comunicação. Quando o servidor tem alguma mensagem nova, a encaminha na resposta, em outros casos retorna vazio o corpo da resposta.

É notável que esta técnica não é eficiente em termos de quantidade de *requests*; todavia, em casos de erros frequentes de conexão, constitui uma técnica que pode contribuir para um bom desempenho.

2.2.7 - HTTP Long Polling

Visando otimizar a técnica de HTTP *Polling*, o *Long Polling* [16] trabalha seguindo uma filosofia de que só fazer *request* para ter uma resposta. Por exemplo, o cliente faz uma requisição para o servidor procurando uma mensagem nova; caso exista: o servidor retorna a resposta com a mensagem no corpo similar ao comportamento na técnica de *Polling* (o cliente faz outro *request*); caso não exista: o servidor, segura a requisição até a chegada de alguma mensagem para o cliente. Logo da chegada da mensagem, responde com ela. Esse processo é repetido ao longo da comunicação.

Pode-se observar que para cada mensagem só vai ser feito um *request* caso de não haver erros de rede. Mas que acontece se se perdeu a conexão com o servidor, etc.? Para isto o desenvolvedor define um intervalo de tempo no qual se não se tem uma resposta o

processo reiniciasse de novo.

2.2.8 - Websockets

Websocket [17] é uma tecnologia que permite a comunicação bidirecional por canais full-duplex sobre um único soquete TCP (*Transmission Control Protocol*). Ele é projetado para ser executado em browsers e servidores web que suportem o protocolo HTML5, mas pode ser usado por qualquer cliente ou servidor de aplicativos. A API *Websocket* está sendo padronizada pelo W3C e o protocolo *Websocket* está sendo padronizado pelo IETF.

Websocket foi desenvolvido para ser implementado em navegadores Web e servidores Web, mas pode ser usado por qualquer cliente ou aplicação servidor. O protocolo *Websocket* é um protocolo independente baseado em TCP. Sua única relação com o HTTP é que seu *handshake* é interpretado por servidores HTTP como uma requisição de *Upgrade*.

A especificação do protocolo *Websocket* [17] define dois tipos de esquemas de URL, **ws:** e **wss:**, para conexões não criptografadas e criptografadas respectivamente. Além do esquema de nomes, os demais componentes da URL são definidos para usar a sintaxe genérica de URI.

Atualmente, os principais navegadores Web brindam compatibilidade com a tecnologia *Websocket* só no caso do navegador padrão do sistema operacional Android é que se encontram problemas de compatibilidade. Neste último caso, se pode suplantar o navegador pelo Chrome ou Firefox.

2.3 - ASPECTOS LIGADOS À PADRONIZAÇÃO/NORMATIZAÇÃO EM E-HEALTH.

A padronização e a normatização atinentes às tecnologias de informação e das comunicações constituem aspectos essenciais à implantação de sistemas de *e-Health* e de *m-Health*. Neste sentido, esta seção abordará, de forma resumida, alguns dos aspectos a serem considerados para o desenvolvimento de aplicativos em *software* e equipamentos a serem integrados a tais sistemas.

No que se segue, abordam-se três aspectos essenciais ao escopo do projeto, que envolvem, para a camada de aplicação:

- Prontuário Eletrônico do Paciente (PEP);
- OpenEHR.
- HL7
- Padrão V1.1.1 TR 102 732

2.3.1 - Prontuário Eletrônico do Paciente (PEP)

O Conselho Federal de Medicina (CFM), por meio da resolução 1638/2002, define o prontuário médico como:

“Documento único constituído de um conjunto de informações, sinais e imagens registradas, geradas a partir de fatos, acontecimento e situações sobre a saúde do paciente e a assistência a ele prestada, de caráter legal, sigiloso e científico, que possibilita a comunicação entre os membros da equipe multiprofissional e a continuidade da assistência prestada ao indivíduo. ”

O prontuário eletrônico do paciente (PEP) pode ser caracterizado como um prontuário médico armazenado eletronicamente, por meio da utilização de ferramentas de tecnologia da informação (TI). O acesso às informações do PEP deve ocorrer de maneira muito mais confiável e segura. Por meio do registro eletrônico das informações, os dados podem ser acessados com mais agilidade, viabilizando o compartilhamento e o acesso simultâneo às informações dos pacientes [41].

Por meio da evolução tecnológica das redes de comunicação e das ferramentas de tecnologia da informação (TIC), o PEP, utilizado inicialmente para o uso interno nas instituições de saúde, evoluiu para um sistema compartilhado de informações médicas, definido como Registro Eletrônico de Saúde (RES) ou, em inglês, EHR (*Electronic Health Record*), que permite o compartilhamento das informações de um ou mais indivíduos, inter e multi-instituição, em uma região (município, estado ou país) ou até mesmo entre um grupo de hospitais. A ISO (do inglês, *International Organization for Standardization*) define o EHR como um repositório de informações a respeito da saúde de indivíduos, em uma forma processável eletronicamente [42].

2.3.2 - *OpenEHR* - Conceitos e Definições

A *OpenEHR* [18] é uma organização sem fins lucrativos, que trabalha na interoperabilidade e computabilidade em saúde eletrônica (*e-Health*). Seu principal objetivo é permitir a viabilidade econômica para a construção de sistemas para área de saúde que sejam adaptáveis e facilmente mantidos[43]. Ela apoia a pesquisa, o desenvolvimento e a implementação de um padrão aberto, conhecido também como *OpenEHR*. Esse padrão é direcionado para sistemas de prontuário eletrônico e de registro eletrônico de saúde (EHR).

As especificações do *OpenEHR* são baseadas na combinação de 15 anos de pesquisa em EHR das comunidades científicas europeia e australiana e considera metodologia de construção de arquétipos, para o desenvolvimento de especificações. Tendo em vista que a arquitetura do *OpenEHR* é altamente genérica e baseada em arquétipos, ela satisfaz vários requisitos que vão além dos conceitos originais do “EHR clínico” [44].

Arquétipos, conforme [47], "são pedaços de conhecimento que indicam como representar conceitos ou informação. Um arquétipo que representa o conceito “peso corporal”, por exemplo, pode definir que este conceito é formado por uma quantidade (um número e uma unidade) e por uma data que indica o momento em que o peso foi medido. Desta forma, a utilização de arquétipos permite compartilhar conceitos e conhecimento clínico, formando sistemas a partir da união de diversos deles".

Os arquétipos são definidos em uma linguagem de computação chamada de ADL (do inglês, *Archetype Definition Language*) [3]. Diferentes arquétipos podem ser agrupados em um "*template*". Um *template* geralmente é usado para definir uma ficha clínica, ou um formulário que será preenchido. Em um *template*, é possível determinar os arquétipos que serão utilizados, as ocorrências que devem existir de cada arquétipo, elementos que, dentro de um arquétipo, serão utilizados, entre outras definições. A partir de um *template*, é possível inferir uma interface que permita criar uma aplicação de entrada de dados que colete as informações a partir dos conceitos e definições contidas nos diversos arquétipos. Quando as informações descritas pelos "*templates*" e arquétipos são instanciadas, ou seja, quando os dados clínicos reais são informados, através do preenchimento de um formulário, por exemplo, dá-se origem a um documento chamado de "*composition*". São

as "*compositions*" que efetivamente contém os dados clínicos que formam as diversas fichas clínicas que compõem o prontuário do paciente.

Os *softwares* que seguem o padrão *OpenEHR* são mais abertos, adaptáveis e colaborativos, pois se baseiam em modelos de informação padronizados e em código aberto [45]. No caso de organizações governamentais, o padrão *OpenEHR* oferece três funcionalidades essenciais:

- Uma abordagem baseada em uma plataforma computacional para a Saúde, que provê o compartilhamento de dados de saúde para o uso de múltiplas aplicações. Isso leva a uma economia na utilização de sistemas de Tecnologia de Informação e Comunicação (TIC) direcionados para a área de saúde. Nesse caso, vários fornecedores podem fornecer sistemas de *backend* ou *frontend* que podem ser integrados por meio de uma interface previamente definida;
- Os dados clínicos e o modelamento dos processos são gerenciados pelos médicos de forma separada do *software*, por meio da utilização de arquétipos do *OpenEHR*;

Para a integração com sistemas e padrões existentes, por meio de um gerenciamento semântico do *OpenEHR*, um marco regulatório importante para a utilização efetiva do PEP/RES foi a resolução CFM nº 1.821, de 11 de julho de 2007, que “aprova as normas técnicas concernentes à digitalização e uso dos sistemas informatizados para a guarda e manuseio dos documentos dos prontuários dos pacientes, autorizando a eliminação do papel e a troca de informação identificada em saúde”.

A estrutura do *OpenEHR* pode ser definida usando uma abordagem de três camadas. Como mostra a Figura 2-4 o modelo de referência pertence à camada base e contém as definições das estruturas de dados, tipos, os aspectos de segurança e um atributo identificador que suporta a indexação de cada modelo de referência. A próxima camada contém os modelos relacionados à semântica médica (os arquétipos). A camada superior oferece diferentes serviços para o contexto externo de acordo com uma arquitetura orientada a serviços.

Virtual EHR			
terminology service	demographic service	EHR service	archetype service
EHR Extract			
EHR	Demographic	Integration	Template OM
Composition		OpenEHR Archetype Profile	
Security	Common	Archetype OM	ADL
Data Structures			
Data Types			
Support (identifiers, terminology, access)			

Figura 2-4: Distribuição de camadas das especificações do OpenEHR.

Diferentes arquétipos podem ser agrupados em um *template*. Um *template* pode ser visto como uma composição de arquétipos para definições mais complexas, isto é, um arquétipo pode se referir apenas a uma medida de um parâmetro fisiológico, enquanto que o *template* pode se referir a todos os parâmetros fisiológicos referentes a uma doença cardiovascular, que inclui arquétipos para pressão do sangue, frequência cardíaca, sinais eletrocardiográficos, dentre outros. Um *template* é frequentemente utilizado para a definição de um modelo de relatório de caso ou um formulário a ser preenchido.

Outra característica interessante é a existência de repositórios globais que fornecem arquétipos padronizados em diferentes idiomas. Esta é uma vantagem em termos de interoperabilidade e reutilização de código-fonte e aplicações. Além disso, este mecanismo de distribuição centralizado padroniza processos de semântica médica. Um repositório que inclui a definição de *templates*, arquétipos, modelos, etc. pode ser encontrado em [19].

2.3.3 - Utilização do padrão OpenEHR no Brasil

O Ministério da Saúde, por meio da portaria nº 2.073, de 31 de agosto de 2011, regulamentou o uso de padrões de interoperabilidade e informação em saúde para sistemas de Saúde no âmbito do Sistema Único de Saúde (SUS). A portaria define no capítulo II que será utilizado o modelo de referência *OpenEHR* para definição do Registro Eletrônico de Saúde (RES). Além disso, no mesmo capítulo a portaria define que para a

interoperabilidade entre sistemas de saúde suplementar serão utilizados os padrões TISS (Troca de Informações em Saúde Suplementar).

O padrão TISS foi estabelecido pela Agência Nacional de Saúde Suplementar (ANS) e continua em evolução, envolvendo, dentre diversos aspectos, a construção de arquétipos que possam ser utilizados para transportar qualquer informação representada no TISS. Ele também visa desenvolver um *software* protótipo que utilize o modelo de referência do OpenEHR para a troca de informação no TISS.

Ademais, algumas empresas desenvolvedoras de sistemas na área de Saúde Eletrônica já estão construindo sistemas baseados no padrão OpenEHR. Isso é consequência direta do reconhecimento do valor do OpenEHR e da utilização de arquétipos para a representação do conhecimento na área de saúde.

2.3.4 - HL7

HL7 (Saúde Nível 7) [20] refere-se a um conjunto de normas internacionais para a transferência de dados clínicos ou administrativas entre aplicativos de *software* utilizados por prestadores de cuidados de saúde. HL7 v2.x é uma dessas normas baseadas em mensagens desenvolvidas pela *Health Level Seven Internacional* e um padrão ANSI acreditado, para o desenvolvimento de mecanismos coerentes e extensíveis para o intercâmbio, gestão e integração de informação eletrônica no domínio clínico.

O nome refere-se à camada de aplicativos do modelo OSI de sete camadas, que descreve como os dados são trocados, o momento do intercâmbio e o tratamento de erros entre os diferentes prestadores de cuidados. Com as normas HL7 v2.x, definições para diferentes tipos de mensagem, que podem ser úteis para padronizar a comunicação entre os “e-Health Devices”, estão disponíveis. Por exemplo a mensagem ORU (*Observation Request Unsolicited*) é utilizada para tratar os dados medidos pelo sensor como uma observação de um parâmetro específico.

2.3.5 - TR 102 732 V1.1.1 M2M ETSI

O padrão TR 102 732 V1.1.1, é definido pelo ETSI (do inglês, *European Telecommunications Standards Institute*) [7] como um dos padrões M2M, voltado para aplicativos de *e-Health*.

Neste padrão, as seguintes entidades são definidas:

- Paciente: Um "paciente" pode ser qualquer indivíduo ou substituto que poderia usar um dispositivo de monitoramento remoto para recolher medições, dados ou eventos. Medições do paciente são independentes de onde elas foram tiradas.
- Dispositivos de Monitorização Remota (RMD): Um RMD é um dispositivo M2M eletrônico com um sensor, uma interface de usuário e / ou atuador e uma interface na rede M2M. O dispositivo recolhe informação do paciente e comunica com o provedor de serviços M2M e / ou aplicação de M2M por uma rede M2M. O RMD também pode se comunicar com essas entidades através de um gateway M2M. A Figura 1 mostra o RMD na camada dispositivo M2M.
- Provedor de capacidades de Serviço M2M: É uma entidade de rede que fornece serviços de comunicação para entidades de aplicação M2M. Brinda suporte funcional específico para o intercâmbio de informação em saúde.
- Entidade aplicação M2M: É uma aplicação de alto nível, tais como um centro de análise de dados, uma rede de cuidados integrados de saúde, uma organização de serviços de saúde, um banco de registro de saúde, ou de uma rede pública de saúde e / ou rede especialidade.
- Coordenador de Cuidados: Inclui pessoas físicas ou aplicações de supervisão clínica sobre as informações recebidas a partir do dispositivo (s) do paciente. O coordenador de cuidados pode intervir se as medições ou alertas indicam que houve uma mudança no estado de saúde do paciente, ou se as medições caíram fora de um intervalo pré-determinado. Ele também pode informar ao médico sobre o estado do paciente, se as medições indicam um potencial problema de saúde.
- Registro Eletrônico de Saúde: também é um registro médico mantidos pelo sistema de saúde em formato digital para um indivíduo, ou um registro médico mantido pelo indivíduo para si.
- Os clínicos: Ele inclui médicos, enfermeiros, assistentes médicos, psicólogos e outros profissionais médicos que avaliam medições remotas para determinar intervenções clínicas adequadas, se necessário.

Outros aspectos, tais como casos de uso e de mensagens estão detalhadas no padrão V1.1.1 TR 102 732 e serão abordadas em nossa proposta descrita na próxima seção.

2.4 - CONSIDERAÇÕES FINAIS.

Neste capítulo foram apresentados os conceitos básicos e tecnologias que apoiam o desenvolvimento deste trabalho.

Conceitos como *e-Health*, *m-Health* e telemedicina foram tratados para permitir contextualizar o foco de nosso trabalho, bem como a caracterização de tecnologias como Comunicação M2M, Computação em Nuvem, SOA e REST, serviu para criar bases conceituais na definição de uma plataforma escalável, com suporte à mobilidade e de implantação relativamente simples.

Em termos de padronização da informação em *e-Health*, os padrões OpenEHR e HL7 foram considerados para a representação e a transmissão dos dados, respectivamente. É importante destacar que o OpenEHR, apesar de não estar focado na representação de variáveis do contexto do paciente, não impõe restrições para que variáveis do contexto também possam ser descritas por arquétipos referentes a observações. Dessa forma, uma abordagem inicial de gerenciamento de contexto pôde ser criada, utilizando os recursos desse padrão.

Acredita-se, assim, que as principais bases conceituais necessárias ao entendimento do trabalho e ao posterior aprofundamento de forma integrada tenham sido, então, providas no capítulo ora encerrado.

3 - TRABALHOS RELACIONADOS

Este capítulo tem por objetivo descrever sucintamente um conjunto de trabalhos relacionados, subdivididos em três partes:

- 1- Trabalhos sobre plataformas de *e-Health/m-Health* e arquitetura de rede subjacente;
- 2- Trabalhos focados em aspectos de padronização;
- 3- Trabalhos focados na doença apnéia do sono.

Nas seções seguintes, faz-se um estudo do estado da arte relativo a propostas de sistemas de *e-Health*, com base nos três subconjuntos acima. Este resumo de trabalhos foi o resultado de uma busca conduzida pela seguinte metodologia: foram inseridas as palavras-chave, "e-Health", "electronic health", "Internet of things", "IoT", "architecture", "sensor", "platform", "EHR", "m-Health" e "Telemedicina", em diferentes motores de busca como "google", "bing", "yahoo" e sítios próprios de diferentes entidades de pesquisa e desenvolvimento. Dos resultados obtidos foram selecionados os pertencentes ao período anual 2009 a 2015.

3.1 - TRABALHOS REFERENTES A PLATAFORMA DE TELEMONTORAMENTO

Existem diferentes propostas de plataformas que pretendem realizar o monitoramento de parâmetros fisiológicos e do contexto do paciente e o acesso remoto dos profissionais aos dados capturados. Dependendo do desenho, se tem diferentes entidades envolvidas, que vão desde a captura de dados até o armazenamento em servidores remotos e o acesso por usuários. Algumas das propostas são baseadas na nuvem enquanto outras em servidores de armazenamento locais e aplicativos em smartphones.

Para caracterizar um sistema de *e-Health* em termos de arquitetura e de uma plataforma que implemente tal arquitetura é preciso tratar os seguintes aspectos:

- Sistema de captura de dados: Sensores empregados, agregador e transmissor de dados;
- Comunicação entre as entidades, considerando tecnologias, padrões de representação dos dados, protocolos;

- Sistema de armazenamento: Nuvem, banco de dados locais, etc.;
- Sistema de disposição dos dados: aplicativos Web, aplicativos em nuvem, aplicativos para plataformas móveis;
- Casos de uso da arquitetura: classes de doenças, abrangência geográfica, caracterização da população-alvo (pacientes).

Segue-se então o primeiro subconjunto de trabalhos relacionados.

Y. E. Rivera [21] trata o monitoramento remoto dos níveis de glicose em pessoas diabéticas, considerando que as variações dos níveis de glicose podem produzir danos irreversíveis ao longo do tempo. Isso motiva os autores a desenvolver uma plataforma que permite comunicar para especialistas a condição do paciente (quantidade de glicose no sangue, localização) e, assim, proporcionar mecanismos de atenção urgente, em caso de condições desfavoráveis para a saúde. Como central de processamento, no lado do usuário, foi utilizado um micro controlador Arduino Mega 250 e o e-Health Sensor Shield [22] para a leitura dos dados capturados pelo glicômetro. Para a interligação com a Web e obtenção de coordenadas geográficas foi usado um modulo sim908, que opera em bandas 850/900/1800/1900 MHz. Dessa forma, o dispositivo se conecta à Web usando uma conexão GPRS. Para o armazenamento um banco de dados relacional MySQL foi criado. Na interface gráfica se usou a linguagem PHP. Como interface de rádio foi utilizado GSM. Na Figura 3-1 é resumida esta proposta.

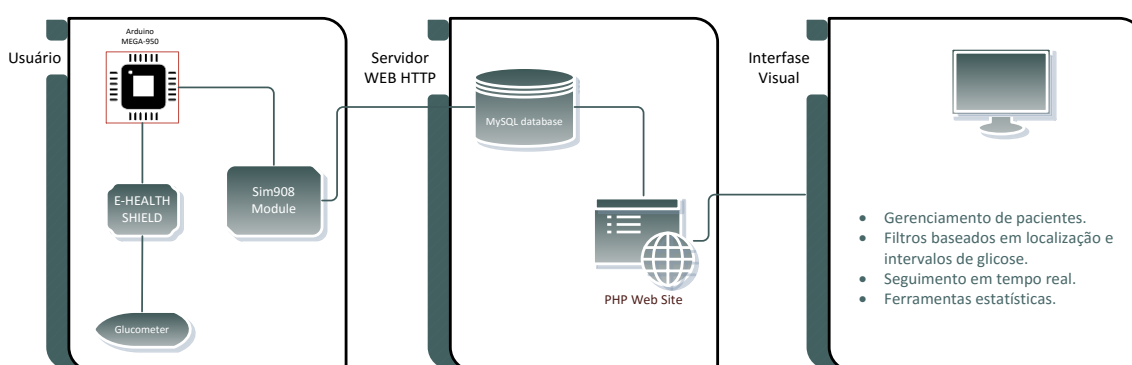


Figura 3-1: Plataforma para o monitoramento dos níveis de glicose.

W. Chen et al. [23] consideram o acompanhamento de recém-nascidos com sensores com fio complica o desenvolvimento natural da criança, a manipulação de médicos e a interação com os pais. Os autores propõem a criação de um sistema baseado em sensores colocados na roupa da criança controlados por um processador central (mostrado na

Figura 3-2). Este controle central faz a conversão analógico-digital, contando com uma interface sem fio, encaminha os dados para uma unidade de processamento central, acessível por médicos (computador central unidade de terapia intensiva, onde atua o médico de plantão). Como coletor de dados e conversor foi projetada uma placa com o protótipo Arduino Pro Mini Controller. Para fornecer comunicação sem fio é usado o módulo BlueSMirf. Um estudo foi realizado para garantir um baixo consumo de energia e baixa interferência de rádio sobre os pacientes; dentre os diferentes protocolos de comunicação, para este cenário, Zigbee e Bluetooth, foram considerados os melhores.

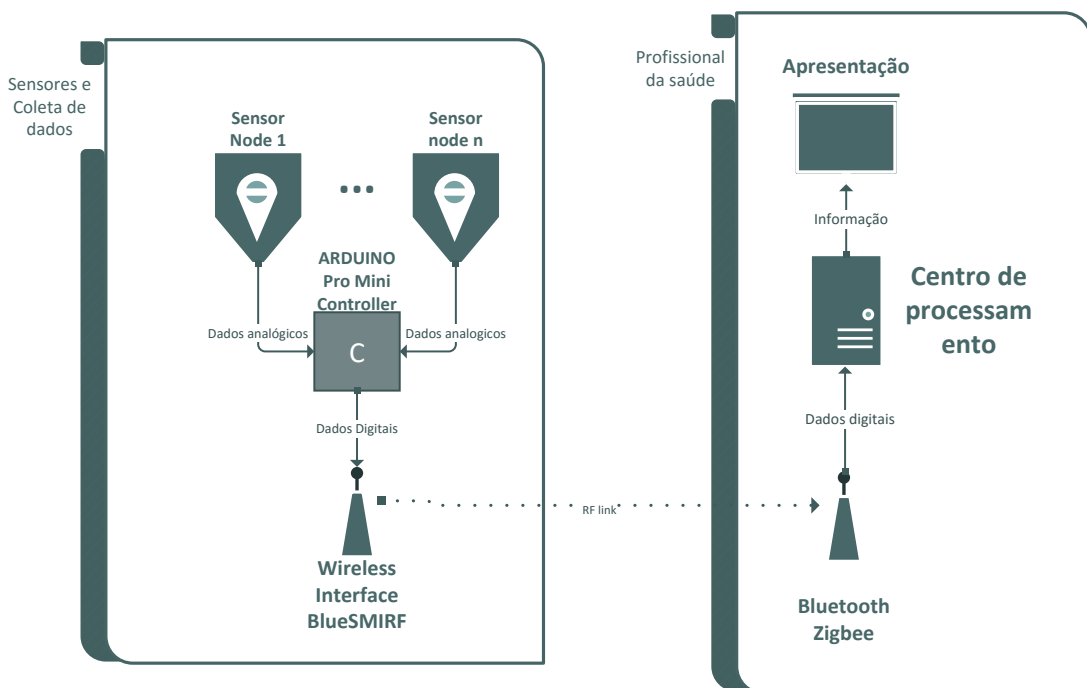


Figura 3-2: Plataforma para o monitoramento de recém-nascidos.

E. Smithayer [24] trata a importância do monitoramento sistemático da pressão arterial em mulheres grávidas em regiões rurais, para envio a médicos situados remotamente. Ele propôs um sistema, mostrado na Figura 3-3, para orientação visando fazer corretamente as medições da pressão arterial. Para isso utiliza dois sensores de força para detectar se o braço está adequadamente apoiado em qualquer superfície, um acelerômetro para detectar a não movimentação do braço, Arduino Uno para obter as informações coletadas pelos sensores, que serão encaminhadas para um aplicativo em um iPhone.

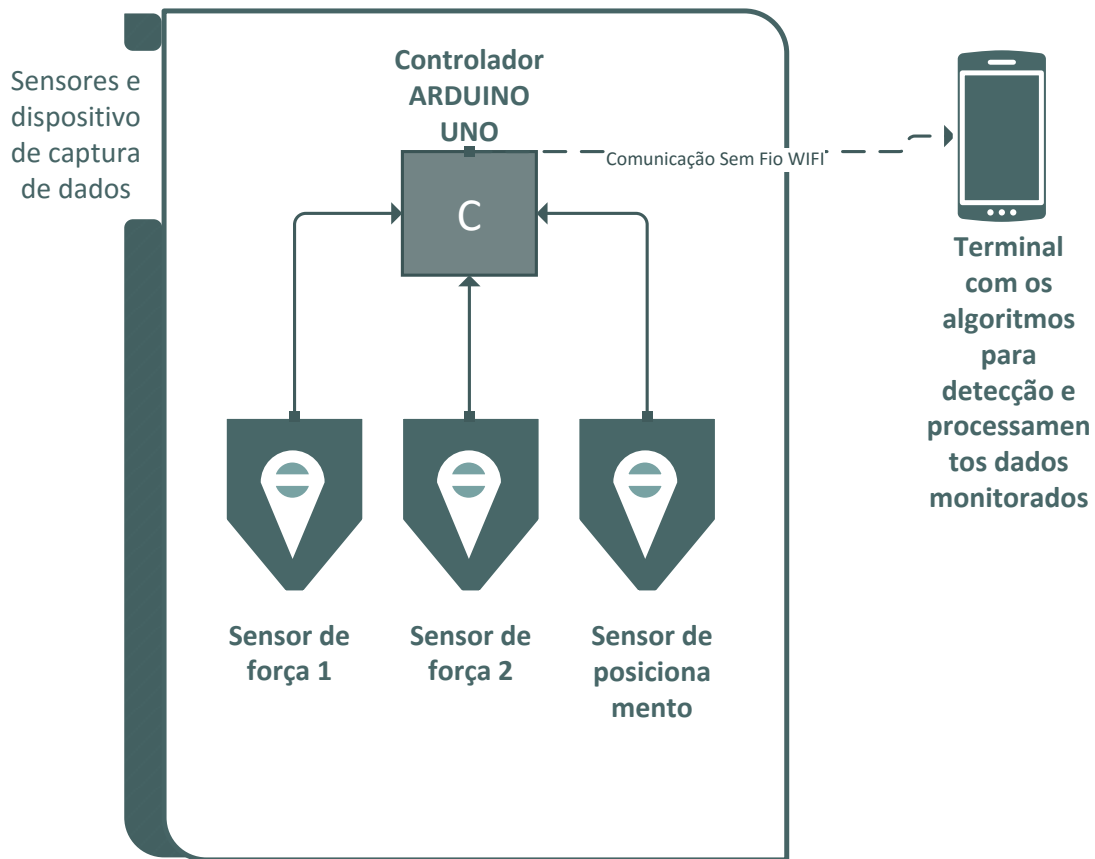


Figura 3-3: Plataforma para apoio à medição de pressão arterial em mulheres grávidas.

O. Ogunduyile et al. [25] propõem um protótipo que oferece serviços de diagnóstico a partir de dados fisiológicos de um paciente distante, recolhidos em tempo real. O trabalho é referido ao monitoramento de pacientes que estão fazendo reabilitação motora. Eles empregam uma rede sem fio denominada SOWBAN, do inglês, “*Service Oriented Wireless Body Area Network*” com base em sensores axiais para detectar os diferentes movimentos do paciente, ou seja, correndo, descansando, caminhando e realizando atividades perigosas, como possíveis quedas. Além dos acelerômetros, um sensor de SpO2 (referente à saturação de oxigênio) é usada para medições da taxa de oxigênio no sangue.

A arquitetura, mostrada na Figura 3-4, é dividida em três camadas:

- Camada Arduino: o Arduino é usado para coletar os dados dos sensores no SOWBAN. Neste protótipo tem uma ligação com fios como meio de transmissão entre os sensores e o Arduino.

- Camada Central Inteligente (*Smartphone*): aqui, os parâmetros monitorados são processados sendo provida uma interface gráfica para o paciente que mostra os dados e faz o *upload* para o servidor remoto.
- Servidor de saúde: é um servidor usado para armazenar registros para cada paciente. Os registros são disponibilizados para o médico logo após o *upload*.

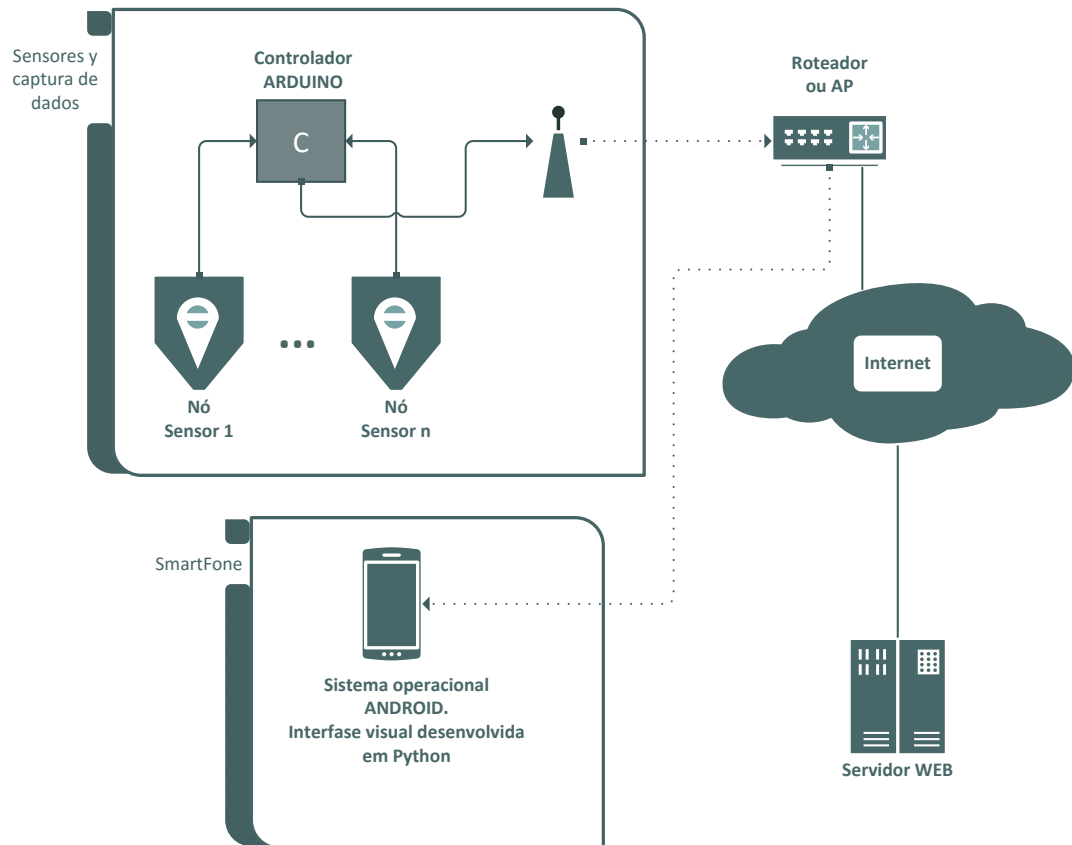


Figura 3-4: Plataforma para assistência de pacientes em fases de reabilitação motora.

M Bhatti e M. Masud [26] descrevem a plataforma mostrada na Figura 3-5. Esta plataforma tem a vantagem de, além de controlar diferentes parâmetros fisiológicos do paciente, obter diferentes características do contexto como o posicionamento e a atividade motora no caso do paciente e de luminosidade, temperatura, pressão e eletromagnetismo para o local onde ele se encontra. O paciente pode, por exemplo, tomar decisões sobre quais são os parâmetros que serão enviados para o servidor na “nuvem” e quais são os valores para os quais alarmes têm de desencadear. Na proposta são utilizados: (i)

acelerômetros para detectar a postura do paciente e estado de movimento, (ii) sensores para captura dos parâmetros fisiológicos comuns, (iii) sensores de luz, temperatura, pressão atmosférica e campos eletromagnéticos, (iv) uma placa Arduino para recepção sem fio e encaminhamento dos dados recolhidos pelos sensores, (v) Smartphone com Android OS, (vi) operadora de rede celular e (vii) armazenamento em nuvem através de servidores Web e bancos de dados.

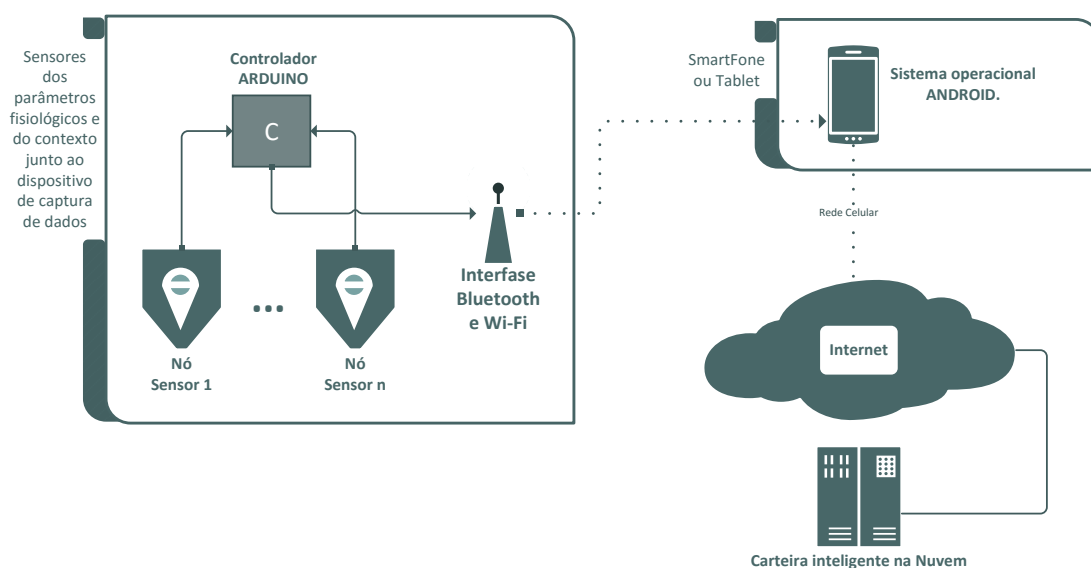


Figura 3-5: Plataforma para monitoramento de parâmetros fisiológicos e do contexto do paciente.

S. Fuicu et al. [27] se concentram em doenças crônicas, para as quais um acompanhamento constante dos parâmetros fisiológicos pode ajudar a evitar o aparecimento de crises. Para isto os autores a propõem a plataforma, mostrada na Figura 3-6, que permite: (i) monitoramento constante de parâmetros fisiológicos e (ii) categorizar os tipos de doenças e parâmetros relativos a cada doença. Eles empregam: (i) uma rede com fios de sensores, (ii) uma plataforma Arduino para o controle dos sensores e (iii) telefone móvel com sistema operacional Windows Phone 8, que pela interface Bluetooth recebe os dados obtidos por Arduino. A proposta também tem uma aplicação que notifica pacientes de diferentes variações sobre o parâmetro monitorado, este módulo pode ser operado para transmitir as informações para servidores na Internet. Para o armazenamento remoto, disponibilidade e manutenção de dados, empregam os serviços de nuvem da Microsoft Azure.

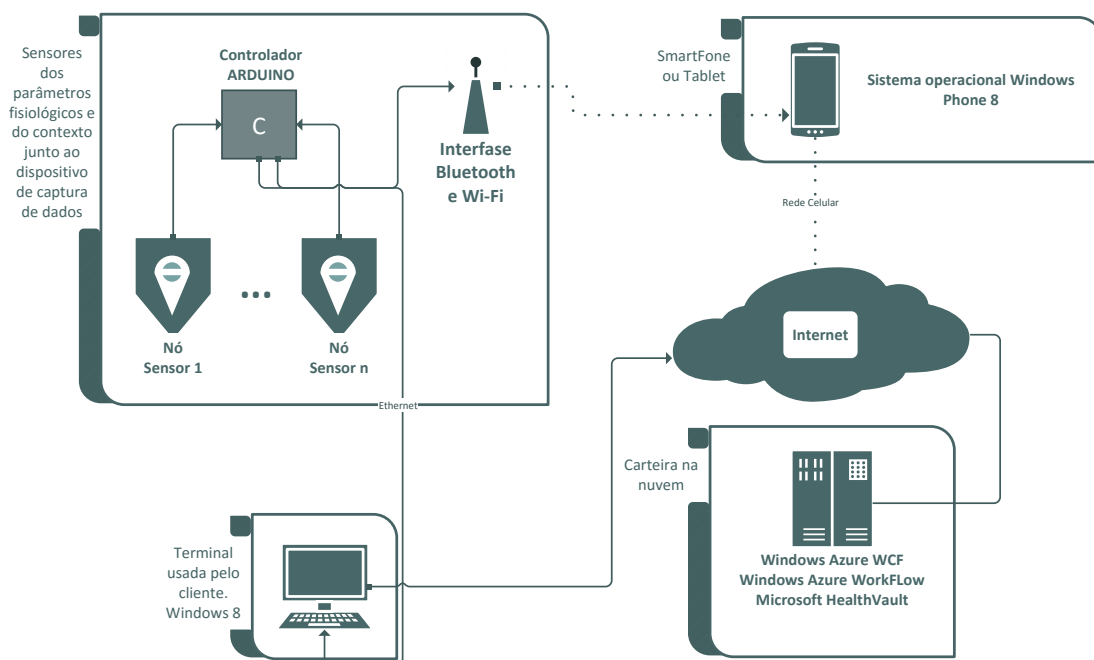


Figura 3-6: Plataforma para monitoramento de parâmetros fisiológicos em doenças crônicas.

N. A. Nayan e F. A. Azami [28] consideram também o cenário de pessoas diabéticas. A plataforma, mostrado na Figura 3-7, é dedicada à monitorização contínua de parâmetros, tais como a glicose, a resposta muscular e os níveis de oxigênio no sangue. Os autores dizem que os periféricos, como glicômetro e um sensor de resposta galvânica da pele pode criar um custo muito baixo e sistema de utilidade alta. A arquitetura é muito semelhante à tratada em[24] mas substituindo o computador por um smartphone .

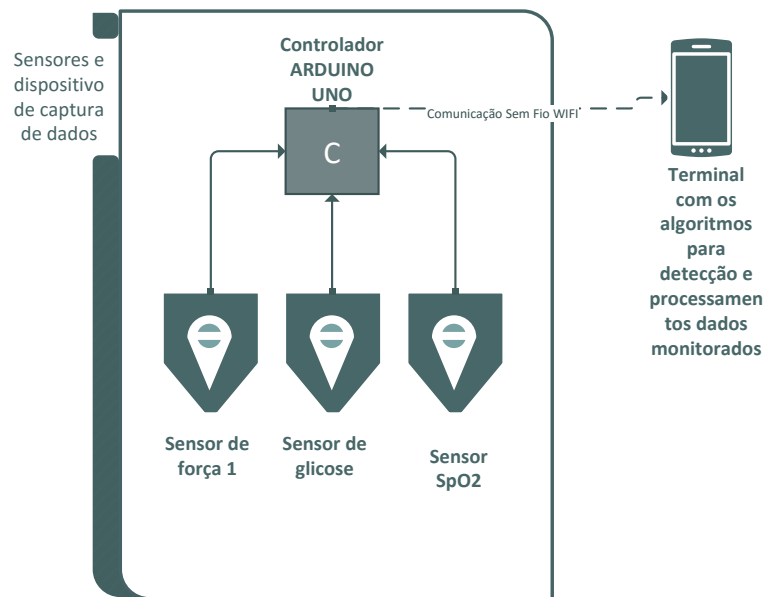


Figura 3-7: Plataforma para monitoramento da diabetes baseada em smartphone.

P. Maia et al. [29] propõem uma plataforma Web chamada *EcoHealth* para conectar a doutores com pacientes que são monitorados num ambiente de redes de sensores heterogêneos. A plataforma é baseada em padrões da Internet, tendo a capacidade de integrar a informação de diferentes tipos de sensores para prover mecanismos de monitoramento, processamento, visualização, armazenamento e a capacidade de avisar mediante mensagens tanto aos médicos como aos pacientes. No artigo são apresentados os esquemas de funcionamento da plataforma.

Para lograr a comunicação heterogênea dos diferentes sistemas de captura (sensores ou redes de sensores com variados e complexos *hardware* e *software*) planejam uma camada intermediária que sirva como abstração para a integração dos diferentes tipos de *hardware*.

O projeto divide-se em três camadas principais. A camada base tem como objetivo desenvolver um esquema de drivers que possibilite integrar a cada um dos sensores cumprindo todos eles com uma mesma interfase assim garantindo a interoperabilidade com as camadas superiores. A segunda camada brinda a abstração mediante serviços que se comunicam com a camada de aplicação Web e a nuvem: Armazenamento, Conexão, Manipulação de dados, Visualização e serviços de segurança. E a terceira camada é o sistema Web *EcoHealth* na nuvem que brinda serviços de manutenção de usuários, médicos, autoatendimento, assistência remota, dentre outros.

J. Ko et al. [30] fazem uma revisão geral do uso de redes de sensores no cuidado da saúde. Diferentes cenários práticos são analisados:

- 1- O primeiro cenário difere bastante do que estamos acostumados no que se refere ao emprego de redes de sensores para a saúde. Aqui são monitorados os médicos e não os doentes. O cenário que exemplifica a utilidade é uma área onde se tem muitas vítimas ao mesmo tempo, por exemplo uma zona de desastre. Neste cenário é preciso conhecer a localização dos médicos e socorristas assim como os reportes de estado de cada um dos pacientes tratados.
- 2- O segundo cenário é o clássico de monitoramento de parâmetros vitais em hospitais.
- 3- Monitoramento de parâmetros na vida diária de um idoso ou algum doente com algum tipo de deficiência que precise tratamento contínuo é o terceiro cenário.
- 4- Outro uso das redes de sensores sem fio é no uso de doentes com incapacidades motoras onde os sensores capturam informações do estado do corpo e encaminham para uma central de processamento que acionara os motores em dependência da função que se deseje projetar.

Além dos diferentes cenários, são estudadas algumas das principais regras que tem que cumprir um sistema médico baseado em redes de sensores:

- 1- Confiança: Os dados têm que ser recolhidos no processador de dados integralmente, o sistema tem que se confiável fim a fim.
- 2- Privacidade e segurança: o sistema tem que apresentar mecanismos que evitem intrusos protegendo a privacidade
- 3- Baixo consumo: O consumo de energia é um fator chave no uso destas tecnologias pois um descarregamento rápido das baterias pode levar a condições de perigo ao paciente.

I. Orha e S. Oniga [31] apresentam um sistema automatizado para o monitoramento dos parâmetros fisiológicos humanos. O sistema permite a recollecção de dados de 9 sensores. Os autores, como em [32] utilizam a plataforma de desenvolvimento *e-Health Sensor Shield V2.0* da *CookingHack*. Até o momento só são obtidos os dados e transmitidos para um computador via comunicação serie ou pela interfase sem fio utilizando o modulo ZigBee para Arduino. Foi desenvolvido um aplicativo na linguagem

Python que permite variar diferentes parâmetros dos sensores em tempo real. Como trabalhos futuros propõem incluir diferentes módulos de conexão e ampliar a plataforma para a Web disponibilizando serviços em tempo real acessíveis remotamente pelos médicos e criando sistemas de detecção automática de deterioro da saúde dos usuários. Como se pode apreciar esta proposta não tem um caráter inovador mas serve para descrever, mas detalhadamente, o que é a *e-Health Sensor Platform* da *CookingHack*.

As propostas tratadas mostram que todos os sistemas contam com uma camada de *hardware* intermediária entre a rede de sensores e a Internet ou serviços da nuvem. Comumente é utilizado um smartphones como mediador entre a captura de dados e o sistema de armazenagem. Tem sentido pois atualmente é comum o uso das novas tecnologias da telefonia na vida diária e deste jeito se garante a mobilidade do usuário aumentando a quantidade de cenários nos quais pode ser aplicado o sistema.

Com base na Tabela 3-1 verifica-se que a plataforma proposta neste trabalho pretende integrar as vantagens de cada uma das plataformas anteriormente tratadas com padrões recentes como comunicação M2M. Pode-se observar que nossa proposta tem um comportamento similar às propostas [26] e [27] mas com a grande diferença de que nós propomos um mecanismo de padronização do sistema de captura, baseado nas regras de semântica providas a partir do OpenEHR e com foco na interoperabilidade. Adicionalmente, observa-se que tentativas de integração com o padrão OpenEHR no nível do sistema de captura não foram encontradas na literatura consultada. Outro aspecto a considerar é que a nossa proposta faz uma integração com uma arquitetura M2M focada na área de *m-Health*.

Tabela 3-1: Comparação das plataformas tratadas nos trabalhos relacionados.

Proposta	Exames ou Doenças alvo	Entidades		Arduino	Suporte à mobilidade	Acesso remoto	Computação em Nuvem	Emprego de smartphon e
		Rede	Sensores					
Y. E. Rivera [21]	Medição de Níveis de glicose.	GPRS	E-HEALTH SENSOR SHIELD Glicômetro digital	ARDUINO MEGA 250	X	X	---	---
W. Chen, et al. [23]	Monitoramento de recém-nascidos	Bluetooth com uma PC	Sensores localizados na roupa da criança	ARDUINO Pro Mini	---	---	---	---
E. Smithayer [6]	Monitoramento de pressão sanguínea em mulheres grávidas	Enlace cabeado PC (Serial Com.)	Sensores de força e de localização	Arduino UNO	---	---	---	---
O. Ogunduyile, et al. [25]	Reabilitação Motora	SOWBAN (Sistema de captura), Internet (Servidor WEB)	Acelerômetros, SpO2	Arduino	X	X	---	X
M Bhatti and M. Masud [26]	Múltiplas	Bluetooth entre o sistema de captura e o smartphone, Wifi, 3G, 4G para os serviços na nuvem	Acelerômetros, parâmetros fisiológicos, luminosidade, temperatura, pressão atmosférica, campo eletromagnético	Arduino UNO	X	X	X	X
S. Fuicu, et al. [27]	Doenças Crônicas	Bluetooth, Wifi entre o sistema de captura e o smartphone, Wifi, 3G, 4G para os serviços na nuvem	Múltiplos sensores	Arduino	X	X	X	X
N. A. Nayan and F. A. Azami [28]	Pessoas com diabetes	Wifi entre o sistema de captura e o smartphone	Glicômetro, resposta galvânica da pele	Arduino UNO	X	---	---	X
P. Maia, et al. [29]	Múltiplas	Internet Standars	Múltiplos	---	---	X	---	---

J. Ko, et al. [30]	Múltiplas	Rede Ad-HOC	Múltiplos	----	----	----	----	----
I. Orha and S. Oniga [31]	Múltiplas	ZigBee para conectar com o PC	E-HEALTH SENSOR SHIELD.	Arduino	----	X	----	----
Plataforma proposta nesta dissertação	Múltiplas	Múltiplos (p.ex. Bluetooth, ZigBee, Wifi, GSM, 3G, 4G)	Múltiplos (p.ex. e-Health Sensor Shield da CookingHack)	Arduino/ Outros com base em C++	X	X	X	X

Um outro diferencial: a nossa proposta emprega três técnicas de “*publish/subscribe*” diferentes, permitindo uma maior compatibilidade com ambientes baseados na nuvem.

Como um último diferencial, cabe destacar que o foco da proposta não está limitado a uma doença específica nem a um conjunto de sensores predefinidos, dado que se propõe uma plataforma para a “criação” de novos aplicativos, sem restrições de sensores e doenças alvo específicos.

3.1.1 - *e-Health Sensor Platform da Cooking Hacks*. [22]

Em questões de *software* as propostas anteriores diferem em vários aspectos, mas as arquiteturas sob o ponto de vista de *hardware* apresentam maior similaridade. Algumas estão desenvolvidas para o acesso aos dados localmente enquanto outras levam os dados até servidores Web, mas, todas fazem uso de alguma rede de sensores um controlador intermédio e finalmente um dispositivo de processamento de dados.

Em termos de pesquisa, caso que se deseje fazer uma nova proposta de plataforma de monitoramento ou se tenha como objetivo aprofundar nesta área do monitoramento de parâmetros fisiológicos, contar com os aparelhos eletrônicos que permitam a obtenção dos dados é fundamental.

Em [21] [31] os autores empregam a *e-Health Sensor Platform da Cooking Hacks* mostrando que é uma excelente opção para o desenvolvimento de protótipos dado que simplifica o processo de desenvolvimento de *hardware*. Este fato promoveu o emprego da mesma nesta dissertação. Na seguinte figura pode-se observar a arquitetura da plataforma e quais são os parâmetros fisiológicos que tem suporte até o momento:

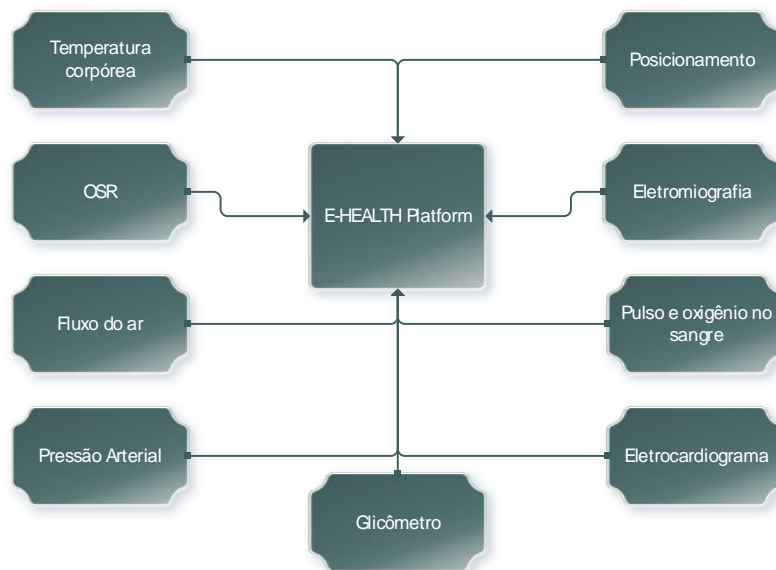


Figura 3-8: Grupos de sensores disponíveis no *e-Health Sensor Platform*.

Na seguinte figura ilustra-se como se integra a plataforma com uma placa Arduino que permita fazer a leitura dos dados analógicos e a conversão digital:

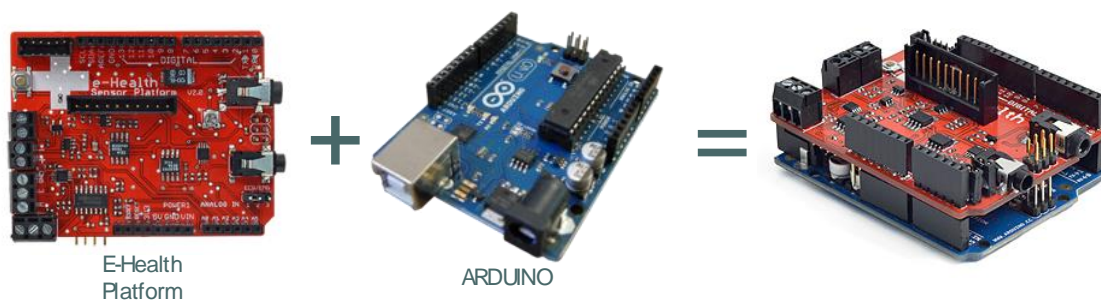


Figura 3-9: Integração do *e-Health Shield* na placa Arduino UNO.

Na seguinte imagem se apresentam as componentes reais as quais são conectadas num enlace com fio à placa *e-Health*.



Figura 3-10: Imagem do grupo de sensores disponíveis na plataforma.

Uma vantagem relacionada à segurança é que na camada de enlace implementa AES 128 para o 802.14.5/ZigBee e WPA2 para o WIFI enquanto na camada de aplicação usa o protocolo HTTPS provendo segurança ponto a ponto nas conexões de um nó sensor e o servidor Web. Conta com inúmeros exemplos de uso, códigos fonte e tutoriais que cobrem quase todo o espectro de aplicação. As vantagens anteriores assim como o preço (plataforma com todos os sensores) perto dos 450 euros fazem ao *e-Health Sensor Platform* a variante mais completa e acessível do mercado para trabalhos centrados em fazer provas de conceitos.

3.2 - TRABALHOS FOCADOS EM PADRONIZAÇÃO

Um segundo subconjunto de trabalhos relacionados foi tratado, relativo a alguns dos principais padrões em uso ou propostos para as áreas de e-Health/m-Health. Apesar do grande número de padrões para tais áreas, o que será aqui apresentado tem foco, principalmente, nos padrões OpenEHR e HL7.

P. R. Silva-Ferreira et al. [33] propõem uma arquitetura para a compilação de dados sobre pacientes de diferentes instituições de saúde. A proposta abrange sistemas de informação baseados em HL7 e um registro eletrônico de saúde com base em OpenEHR. P. Vieira Marques et al. [34] concebem um mecanismo de integração de dados através do qual um EHR virtual é criado para se comunicar com diferentes instituições EHR por meio de agentes, fornecendo como entidade de “tradução” das informações médicas,

independentemente da localização. Eles propuseram um mecanismo de busca na rede hospitalar nacional através da comunicação entre os agentes instalados em cada EHR.

C. Martinez-Costa et al. [35] desenvolveram um conjunto de recursos para atingir a interoperabilidade semântica entre padrão OpenEHR e ISO 13606. A proposta é baseada em agentes para a pesquisa e recolha de informações. Cada HIS contém um agente que atua como um par na comunicação entre HISs diferentes, como na proposta anterior.

E. Osorio, L. Ferreira, R. Abreu, and F. Sousa [36] abordam a interoperabilidade entre sistemas baseados em HL7 e em OpenEHR. Propõem um sistema que assume que as mensagens podem chegar em três formatos diferentes, (i) HL7 v2, (ii) V2.xml e (iii) OpenEHR.

A entidade *Message Broker*, que funciona como um agente, recebe dados de aplicações externas, os converte para um formato V2.XML e os encaminha para a entidade Cliente EHR, responsável por interações com o sistema de gerenciamento de banco de dados.

J. D. Trigo et al. [37] se focam em padrões de interoperabilidade e propostas técnicas para a aplicação harmoniosa dos padrões OpenEHR e ISO / IEE 11073. Eles usaram o conceito de dispositivos de concentração como uma entidade entre o HIS e o sistema de captura (i.e. Laptops, PCs, Smartphones). O dispositivo recolhe e processa todos os dados detectados antes de encaminhá-los para as camadas superiores.

S. Frade et al. [38] realizaram uma revisão de implementações de sistemas de armazenamento para HIS baseados em OpenEHR. Na revisão foram tidas em conta as diferentes representações de dados, sistemas de gerenciamento de banco de dados utilizados e linguagens de consulta. Eles concluíram que a maioria das implementações são baseadas em sistemas de gerenciamento de banco de dados relacionais através de SQL como a linguagem de consulta. Além disso, os autores analisaram alguns trabalhos que utilizam *Arquetype Query Language* (Linguagem de consulta de arquétipos) para consultar os dados médicos.

Um exemplo de arquitetura que integra os padrões OpenEHR e HL7 é provida em [39]. Os autores propõem uma arquitetura M2M (Machine to Machine) baseada em dispositivo M2M para captura de dados e um smartphone como um gateway M2M. O smartphone emprega o formato das mensagens HL7 como protocolo de comunicação

com o EHR baseado em OpenEHR. A proposta não aborda problemas de interoperabilidade entre os sensores de parâmetros fisiológicos empregados.

3.3 - TRABALHOS FOCADOS EM APNÉIA DO SONO

Um terceiro subconjunto de trabalhos relacionados foi tratado, relativo a propostas voltadas para o diagnóstico ou triagem de pacientes sujeitos a apnéia do sono.

A apnéia do sono [40-51] é uma doença que aparece quando as pessoas durante o sono param e iniciam a respiração repetidamente. Isso faz com que o período de sono seja interrompido e que o organismo não descanse o suficiente provocando ao dia seguinte uma sobrecarga no corpo. Essa sobrecarga pode acabar em diferentes tipos de acidentes ou propiciar o desenvolvimento de outras doenças como hipertensão, estresse, dor de cabeça, déficit de atenção entre outras.

A apnéia do sono pode ser dividida em duas categorias:

- apnéia obstrutiva do sono: Ocorre quando os músculos da garganta relaxam durante o sono provocando a obstrução e diminuindo os níveis de oxigênio no sangue. Principais causas são: obesidade, anatomia das amígdalas e circunferência do pescoço.
- apnéia central do sono: Muito menos comum e ocorre quando o cérebro não consegue transmitir sinais para os músculos da respiração. As principais causas são os acidentes vasculares cerebrais e a insuficiência cardíaca.

Várias propostas têm sido desenvolvidas para tratar a apnéia mediante o uso de meios eletrônicos. O método de diagnóstico mais empregado [52], é a polissonografia. Este estudo tem como base o monitoramento com os seguintes equipamentos:

1. Eletroencefalógrafo: Para medir a atividade cerebral;
2. Eletroculógrafo: Para medir a atividade ocular;
3. Eletromiógrafo: Para medir o comportamento muscular em áreas como o tórax;
4. Sensores de fluxo de ar: Para calcular o volume de ar inalado e exalado.

O principal inconveniente dos tratamentos correntes é que o doente tem que permanecer às noites em hospital ou clínica especializada, conhecidas como laboratórios do sono, para fazer o monitoramento do sono, fazendo que os custos sejam elevados e que atualmente mais do 80 % dos doentes não conheçam a sua situação.

Diferentes estudos acadêmicos propuseram arquiteturas de *software* e *hardware* para tratar a apnéia do sono, que incorrem em problemas relacionados com as velhas tecnologias utilizadas. O rastreamento de alguns parâmetros (i.e., fluxo de ar e oximetria) pode ser útil para a detecção da doença e um eventual tratamento específico. Não obstante, o estudo tem de ser baseado nos dados capturados por múltiplos sensores. J. A. Mithun, V. Rajkumar e V. Shanmugapriya [53] demonstraram que a detecção de apnéia do sono com um único sensor é uma tarefa quase impossível.

A seguir, analisamos algumas das propostas fundamentais para a detecção de apnéia do sono.

K.Mm Al-Ashmouny et al.[54] propõem um Sistema baseado em lógica difusa para detectar anomalias no período de sono. O sistema difuso tem como entrada as medições de um sensor do fluxo de ar nasal. Depois de um post-processamento dos sinais obtidos, classificam os eventos em normais ou anormais. Para o caso dos eventos anormais se acrescenta informação coletada por sensores localizados no abdômen e no tórax para classificar se realmente o doente tem apnéia do sono.

Para chegar a conclusões sobre o método de detecção os autores propõem fazer estudos sobre os resultados obtidos em testes feitos com pacientes reais. Finalmente comparam os resultados obtidos com bases de dados tratadas previamente por especialistas da rama. Tem como conclusão que a proposta é factível e que tem grandes vantagens econômicas respeito ao estudo polisonográfico.

L. S. Correa et al. [55] analisam três sinais derivados do ECG com o objetivo de analisar o comportamento e a relação com a respiração. Os parâmetros são: Amplitude da curva R, Variabilidade do Pulso, Área da curva R num determinado intervalo. Para avaliar o trabalho foi empregado um banco de dados [56] com a informação referente ao monitoramento do ECG de 70 pacientes com uma taxa de 100 Hz num período meio de 8 horas. Tem como conclusão que duas das propostas apresentadas servem como mecanismos de diagnóstico de apnéia do sono.

G. Matsuoka et al. [57] conduzem um estudo do comportamento do eletroencefalograma (EEG) em oito pacientes monitorados e com apnéia detectada previamente. Além do EEG é empregado um sensor de oxigênio no sangue (sensor SpO₂). Ambas medições servem para detectar a apnéia do sono uma vez detectadas caídas no

nível de oxigênio por um período de 10 segundos ao mesmo tempo que se detecta uma excitação no comportamento cerebral. Concluem na proposição de um método baseado em umbrais para a detecção de quando as anomalias nos sinais são causadas pela apnéia. Finalmente avaliam o sistema como uma ferramenta visual útil para profissionais da saúde na etapa de diagnóstico. Tem como principal inconveniente o custo de fazer o EEG que na maioria dos casos envolva a permanência do paciente no hospital por períodos que vão de dias até semanas.

Wenlong et al. [58] propõem mecanismos para a detecção de apnéia do sono mediante a análise de frequência do HRV (*Heart Rate Variability*). A partir do sinal do ECG pode-se extrair um outro sinal conhecido como a variabilidade da frequência cardíaca. Um estudo com 80 pacientes foi conduzido para capturar amostras relevantes. Desses 80 pacientes, 64 apresentam diferentes níveis de apnéia do sono e 16 não. A faixa etária varia de 21 até 74 anos. Mediante a análise de frequência do HRV obtiveram que os pacientes com apnéia do sono tinham um decréscimo nas componentes de altas frequências e um incremento nas baixas frequências. Finalmente propõem um mecanismo de processamento de sinais para diagnosticar a apnéia do sono.

O Sistema desenhado por H. Chien-Chang & S. Ping-Ta [59] emprega um filtro passa baixas para limpar o sinal de eletroencefalografia. Logo é aplicada a transformada de Hilbert-Huang. Finalmente é detectada a duração dos episódios de apnéia analisando a variação do espectro de frequência. Como conclusão se tem que a transformada de Hilbert-Huang permite uma dilatação temporal do sinal, permitindo escalas maiores de tempo e maior precisão no processamento.

D.Falie & EM. Ichim [52] apresentam outra proposta para atender a apnéia do Sono que não emprega dados fisiológicos no Sistema de captura. Uma câmera de vídeo 3D é utilizada para gravar o período do sono no paciente. A imagem é dividida em 12 áreas regulares. O algoritmo se tem como parâmetro a média da distância para cada um dos quadrantes. Neste sentido o algoritmo propõe determinar o período do sono REM (*Rapid eye movement*) e o NO-REM (*No Rapid eye movement*). Logo se demonstra uma relação entre o trabalho torácico e a média da distância, além de detectar movimentação ao longo do período do sono. Concluem o trabalho com a avaliação de especialistas da área, tendo como resultado que pode ser uma ferramenta a mais para a detecção da apnéia do sono. Tem como inconveniente que a câmera tem que ser situada muito perto do paciente além

dos custos de infraestrutura.

Townsend et al. [60] propõe um novo cenário para a detecção de apnéia do sono. Como ferramentas, empregam duas bandas de indutância para plestimografia respiratória, colocadas ao redor do peito e o abdômen, para medir o esforço realizado no processo de respiração. Uma rede de sensores de pressão para detecção da movimentação na cama é empregada como substituta da câmera 3D do cenário anterior. Como dados de prova, foi feita uma simulação com usuários reais que simularam o comportamento da apnéia mediante comportamentos similares no processo de respiração. Esta classe de propostas tem essa vantagem pois não tem em conta o estado de sono nem sinais de encefalografia.

M. Habul et al. [40] apresentam uma arquitetura e protótipo de *hardware* como ferramenta de ajuda ao diagnóstico pela parte dos médicos. A arquitetura é dividida em cinco entidades: armazenamento, controle de consumo, gerenciador do sinal, micro controlador e comunicação externa. As sinais tratadas são obtidas a partir de sensores de fluxo de ar.

J. R. Williamson et al. [43] apresentam um trabalho baseado em parâmetros tratados anteriormente com a consideração da aplicação em crianças nascidas antes das 29 semanas de gestação. Se ressalta a importância de detectar e tratar a apnéia devido a sua consequência e efeitos negativos nesta etapa de crescimento. São empregados sensores transcutâneo para medir a saturação de oxigênio no sangue usando oxímetros de pulso colocados nos pés, eletrocardiógrafos e sensores plestimográficos.

O escopo próprio da apnéia brinda uma margem temporal no processo de diagnóstico [45]. Um paciente com riscos de ter a doença pode continuar sua vida cotidiana tal como aconteceu até o momento, mas pensar em telemedicina ou tele monitoramento não carece de sentido. Muitas vezes acontece que nos primeiros dias de estudo ainda não se tem informação relevante, sobre todo para os casos onde o paciente ainda não está numa etapa avançada da doença. Isto provoca que os estudos se prolonguem, obrigando ao paciente a assistir a consultas cada certo tempo para entregar os dados coletados.

Consideremos o seguinte cenário:

Suponha-se que o paciente vai à primeira consulta. Leva os equipamentos para fazer as medições e recebe as explicações de como utilizar os aparelhos. Depois de uma semana de coleta de dados, volta à consulta e o médico nota que, por exemplo, os eletrodos do

ECG, foram mal colocados. Isto leva a ter que repetir todo o procedimento por mais uma semana. Outro exemplo é que: os dois primeiros dias bastaram ao médico para diagnosticar apnéia central, neste caso os equipamentos se mantiveram ocupados por cinco dias quando poderiam ser empregados no seguinte paciente na fila.

É por isso que uma plataforma para tele monitoramento pode melhorar ainda mais tanto o diagnóstico como o sistema de saúde.

Alqassim S. et al. [45] propõem uma plataforma de telemedicina focada na apnéia do sono que permite o acesso remoto dos médicos. A arquitetura emprega um telefone celular na parte do paciente para receber e reencaminhar os dados do censo para o servidor de processamento e armazenagem. Quando um novo conjunto de dados com informações relevantes chega ao servidor, o médico é notificado via SMS solicitando sua posterior análise dos dados. Como meio de comunicação se empregam interfaces Wifi, 3G e GSM.

A

Tabela **3-2** apresenta um resumo destes trabalhos voltados para apnéia do sono. Câmeras de vídeo, máquinas de EEG e sensores plestimográficos, são tipicamente mais caros e menos acessíveis. Sinais de ECG, fluxo de ar e de níveis de oxigênio no sangue, que podem ser capturadas por equipamentos mais simples e em maior escala de acesso para a população geral, também são importantes para a detecção de apnéia do sono. Portanto, o presente estudo é focado em tais tipos de sinais onde além dos parâmetros referentes a ECG, SpO2 e fluxo de ar incorpora um acelerômetro para detectar se o paciente está em pé, deitado ou sentado. Outro aspecto a considerar é que, como diferencial de nossa proposta, nenhum dos trabalhos anteriormente tratados fazem uso das vantagens dos sistemas em nuvem.

Tabela 3-2: Comparação dos trabalhos referentes a apnéia do sono.

Proposta	Parâmetros	Acesso Remoto	Cuidado Caseiro	Computação em Nuvem	Hardware components
K.Mm Al-Ashmouny et al. [54]	Fluxo de ar Atividade Abdominal Atividade Torácica	Não	Sim	No	Sensor Plestimográficos, Fluxo de ar
L. S. Correa et al.[55]	ECG	Não	Sim	Não	ECG
G. Matsuoka et al. [57]	EEG SpO2	Não	Não	Não	EEG, SpO2
Wenlong et al. [58]	Batimentos Cardíacos, ECG	Não	Sim	Não	ECG
H. Chien-Chang & S. Ping-Ta [59]	EEG	Não	Não	Não	EEG
D.Falie & MS. Ichim[52]	Câmera 3D para monitorar o comportamento da caixa torácica	Não	Sim	Não	Câmera 3D
Townsend et al.[60]	Sensor Plestimográfico, Detecção de movimento	Não	Sim	Não	Sensor Plestimográfico , Sensor de pressão
Habul M. et al. [40]	Fluxo de ar	Não	Sim	Não	Sensor de fluxo de ar
J. R. Williamson et al. [43]	Oxigênio no sangue	Não	Não	Não	Sensor Plestimográficos, Fluxo de ar
Alqassim S. et al. [45]	Parâmetros fisiológicos configuráveis	Sim	Sim	Não	----
Nossa proposta	SpO2, fluxo de ar, ECG, Postura do Paciente	Sim	Sim	Sim	Sensor SpO2, Sensor de Fluxo de Ar, ECG, Acelerômetro

3.4 - CONSIDERAÇÕES FINAIS

As propostas tratadas mostram que todos os sistemas contam com uma camada de *hardware* ou um equipamento intermediário entre a rede de sensores e a Internet ou serviços da nuvem. Comumente é utilizado um smartphone como mediador entre a captura de dados e o sistema de armazenamento, o que permite que, com o suporte à mobilidade do usuário, seja aumentada a quantidade de cenários nos quais pode ser aplicado o sistema.

Em termos de mecanismos e processos de padronização, as referências consideradas têm maior foco em problemas de representação semântica. O padrão OpenEHR pode ser visto como um referencial no tocante à representação da semântica das informações ligadas à saúde e a interoperabilidade. HL7 v2 é uma alternativa para a comunicação entre

os dispositivos; a escolha é avaliada pelo seu uso nos estudos acima mencionados. Outro aspecto relevante é que os estudos só têm tratado os problemas de interoperabilidade na camada de aplicação. Essa interoperabilidade pode ser conseguida através da criação de agentes intermediários que entendem as informações nos diferentes formatos fornecidos pelos diferentes normas, incluindo protocolos da camada inferior.

A criação de uma plataforma que abrange desde a captura de dados até a disponibilidade para acesso remoto pelos clínicos envolve várias entidades, como os equipamentos para armazenamento, os sistemas de captação, as rede(s) de comunicação, e as aplicações visuais.

4 - DEFINIÇÃO DE REQUISITOS E DESENVOLVIMENTO DA PLATAFORMA

Neste capítulo, realiza-se inicialmente a apresentação dos requisitos estabelecidos para a construção da plataforma, procurando-se identificar e sistematizar as necessidades para o desenvolvimento deste projeto. Em seguida, é focalizada a arquitetura de rede subjacente integrando os conceitos citados anteriormente, permitindo definir as diferentes entidades da rede e como eles devem interagir. A plataforma, inspirada nos trabalhos [25] e [27], pretende introduzir um mecanismo de padronização relativo ao processo de captura o que leva a uma análise dos aspectos referentes a padrões empregados na comunicação e como eles permitem obter uma proposta de padronização. Finalmente, neste capítulo são apresentadas as diferentes funcionalidades da plataforma, detalhando cada uma delas nas diferentes camadas da arquitetura.

4.1 - REQUISITOS DA PLATAFORMA

4.1.1 - *Requisitos de Alto Nível*

Apresenta-se aqui, inicialmente, uma identificação dos atores envolvidos no processo de uso da plataforma e, em seguida trata-se de requisitos de alto nível funcionais e não-funcionais.

Identificação dos Atores

Neste trabalho não é tratada só uma plataforma para atender apnéia do sono, mas uma plataforma para o desenvolvimento de novos aplicativos para variadas doenças. Não obstante, se propõe como exemplo de uso a criação de um aplicativo ou plataforma para tratar o monitoramento de parâmetros fisiológicos referentes a apnéia do sono. Porém, foi decidido separar os atores em três categorias apresentadas a seguir:

a. Atores no processo de desenvolvimento

Desenvolvedor de software: Encarregado do processo de criação da nova plataforma originada a partir das ferramentas disponibilizadas em este trabalho.

Técnicos da saúde: Profissionais da saúde envolvidos no desenvolvimento das novas plataformas. São os encarregados de definir os parâmetros a serem monitorados, assim como arquétipos relevantes disponibilizados no OpenEHR.

b. Atores no cenário de uso da nova plataforma.

Profissional da saúde: Especialista de determinada área médica que indica o monitoramento, processa os dados, controla os parâmetros de medição com o fim de diagnosticar determinada doença, ou de monitorar a pacientes com doenças crônicas.

Paciente: Indivíduo com suspeitas de determinadas doenças para o qual o diagnóstico com a plataforma é indicado pelo Profissional da saúde.

Desenvolvedor/Mantenedor: Forma parte da equipe técnica dando suporte para a equipe médica. Mediante análises das experiências passadas pelos médicos, brinda suporte técnico e se mantém na melhora constante da plataforma.

c. Atores no cenário de uso de monitoramento para apnéia do sono.

Profissional da saúde pneumologista: Especialista focado na detecção e tratamento de apnéia do sono, que indica o monitoramento, processa os dados e controla os parâmetros de medição com o fim de diagnosticar a presença da doença do paciente.

Paciente de apnéia do sono: Indivíduo com suspeitas de padecer apnéia do sono para o qual o diagnóstico com a plataforma de monitoramento voltada para apnéia do sono é monitorado.

4.1.2 - Requisitos Funcionais

Foram considerados dois tipos de requisitos: funcionais e não funcionais. A seguir, apresenta-se lista de requisitos funcionais:

- 1- Permitir a coleta de dados relativos a pacientes em ambientes caseiros, hospitalar e clínicos;
- 2- Favorecer a transmissão dos dados coletados para centro de triagem e para diversos usuários (profissionais da saúde, familiares e o próprio paciente);
- 3- Permitir um registro histórico acerca de cada paciente;
- 4- Permitir o acesso a bases de dados coletados ao respeito dos pacientes e as enfermidades que se destina;
- 5- Realizar controle de acesso a dados armazenados e em transito;
- 6- Permitir ajustar parâmetros das medições em tempo real;
- 7- Permitir o desenvolvimento de novos aplicativos focados em doenças de natureza

- variada onde o monitoramento remoto é uma característica principal;
- 8- Permitir interoperabilidade de equipamentos desenvolvidos seguindo a plataforma descrita;
 - 9- Garantir compatibilidade com qualquer dos arquétipos de observação definidos no OpenEHR;
 - 10- Permitir a visualização em tempo real dos dados sempre que se tenha acesso á plataforma;
 - 11- Prover mecanismos de comunicação entre as partes envolvidas (e-mail, chat, notificações).

4.1.3 - Requisitos não Funcionais

Como requisitos não funcionais da plataforma, foram considerados os seguintes:

- 1- A plataforma deve ser compatível com:
 - a. Web Sockets;
 - b. HTTP *Polling*;
 - c. HTTP *Long Polling*;
 - d. Microsoft SQL;
 - e. Serviços Web REST.
- 2- Deve ser compatível com o Sistema Operacional Android no dispositivo móvel que atua como Device Gateway;
- 3- Deve ser compatível com qualquer navegador Web que suporte HTML 5 e *Websocket*;
- 4- Permitir acesso remoto Web desde os sistemas operacionais Microsoft Windows, Apple OS-X, ou Linux UBUNTU;
- 5- Deve ser compatível com a última especificação do OpenEHR;
- 6- Qualquer troca de mensagens entre as entidades envolvidas deve seguir o padrão HL7.

4.1.4 - Aspectos Legais

Esta plataforma está destinada ao pessoal pertencente à área científica que pesquisa no aperfeiçoamento dos serviços de cuidados da saúde que trabalham na criação de novos aplicativos orientados às áreas de *e-Health*, *m-Health*, Telemedicina e Ambiente de vida

assistido. O objetivo deve estar vinculado com a criação de sistemas nos quais a captura, acesso e manutenção de parâmetros fisiológicos pode ser relevante para tratar doenças nos escopos anteriormente citados.

É importante destacar que a plataforma apresentada a seguir é apenas uma prova de conceito com fins de pesquisa e que não tem sido validada por nenhuma instituição acreditada e que nem os criadores deste trabalho, nem a instituição na qual foi realizada se fazem responsáveis por nenhum dano pessoal, material ou moral pelo uso da mesma.

A plataforma ainda não tem sido acreditada por nenhuma instituição competente para o uso em ambientes reais.

4.2 - ARQUITETURA ALVO E DIAGRAMA DE IMPLANTAÇÃO

4.2.1 - Arquitetura Alvo

A Figura 4-1 [26],[61] mostra uma arquitetura amplamente utilizada em aplicações de *e-Health* para monitoramento de parâmetros. Este tipo de arquitetura tem uma grande quantidade de vantagens:

1. O emprego do smartphone garante a mobilidade do usuário.
2. O acesso à Internet é garantido pela própria operadora de telefonia celular do usuário, liberando as instituições médicas de ter que contratar serviços de terceiros.
3. O servidor remoto pode trabalhar como *backend* para diferentes categorias de aplicativos.
4. O sistema de captura fica independente do sistema de armazenamento uma vez que sua comunicação tem que passar obrigatoriamente por uma entidade intermediária. Isto permite que técnicas para padronização na captura dos dados possam ser desenvolvidas mais claramente.
5. O emprego do smartphone e as capacidades atuais dos mesmos permitem que uma maior quantidade de aplicativos possa ser desenvolvida.

Em termos de funcionamento, o Sistema de Captura estabelece a conexão com o smartphone o qual reencaminha os dados obtidos para o servidor remoto, fazendo uso do plano de dados do paciente ou de algum acesso diferente a Internet como Wifi ou WiMax. Uma vez que os dados chegam no servidor, eles são processados e inseridos no EHR,

permitindo o acesso aos médicos via os diferentes aplicativos, Web, mobile, etc. que proveem os diferentes serviços finais.

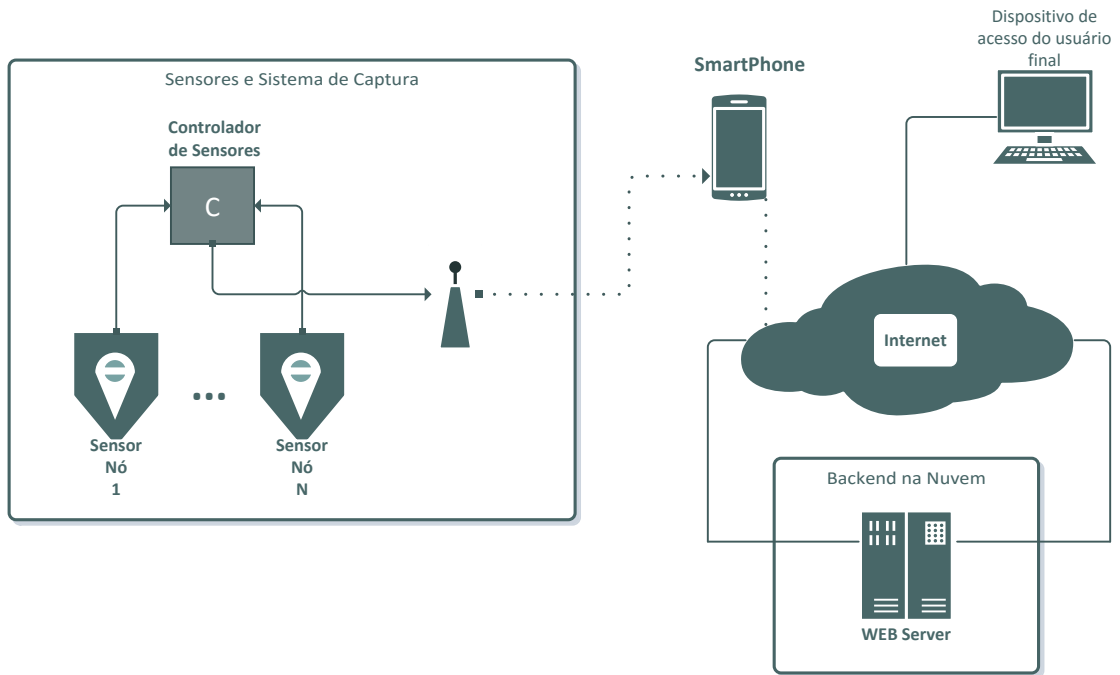


Figura 4-1: Arquitetura predominante em trabalhos voltados para *e-Health*.

Esta arquitetura pode ser transformada facilmente para se ajustar a um ambiente baseado em comunicações M2M [39]. A Figura 4-2 mostra a interação entre o gateway M2M e o dispositivo M2M. O núcleo da rede M2M é representado pela Internet e os serviços da rede do NSCL podem ser desenvolvidos como um conjunto de serviços Web hospedados no servidor. Os diferentes aplicativos de usuário, seja para médicos ou pacientes, podem ser tratados como aplicativos M2M que empregam os recursos providos pelo NSCL com o fim de fazer operações sobre os diferentes dados armazenados no EHR. Finalmente o EHR armazena toda a informação clínica, as ontologias e os dados próprios de cada uma das observações.

No entanto, outras entidades estão envolvidas neste processo de atenção remota de pacientes. Formalmente e de acordo ao padrão ETSI TR 102 732 V1.1.1 [7] e as definições providas em [39] as entidades principais envolvidas no processo são:

- *M2M Device Sensors (M2MDS)*: Sensores que medem os parâmetros objetivos. Eles exigem uma interface de rede para fornecer comunicação com o M2M gateway;

- *M2M Smartphone Gateway (M2MSG)*: ele age como um gateway M2M e fornece comunicação entre o dispositivo M2M e o NSCL. É o encarregado de controlar a inicialização e o comportamento dos sensores. Também pode fornecer interfaces visuais para a configuração da rede, autenticação, as notificações médico-paciente e controle do comportamento dos sensores envolvidos (p. ex. configuração dos parâmetros iniciais, controle de comandos para iniciar e parar os sensores, etc.);
- *Network Service Capabilities Layer*: Prestador de serviços para inserção, consulta, atualização e eliminação dos dados no EHR. Também pode habilitar serviços para a comunicação entre médicos e pacientes, para a manutenção de credenciais, serviços de autenticação, etc.;
- EHR: representado por um conjunto de recursos para o registro de pacientes, médicos e observações dos parâmetros monitorados, este sistema de armazenamento pode ser implementado através da especificação OpenEHR;
- *M2M Application*: disponibiliza interfaces visuais para o acesso aos dados obtidos. Os aplicativos podem ser orientados para pacientes ou médicos e para diferentes tipos de plataforma.

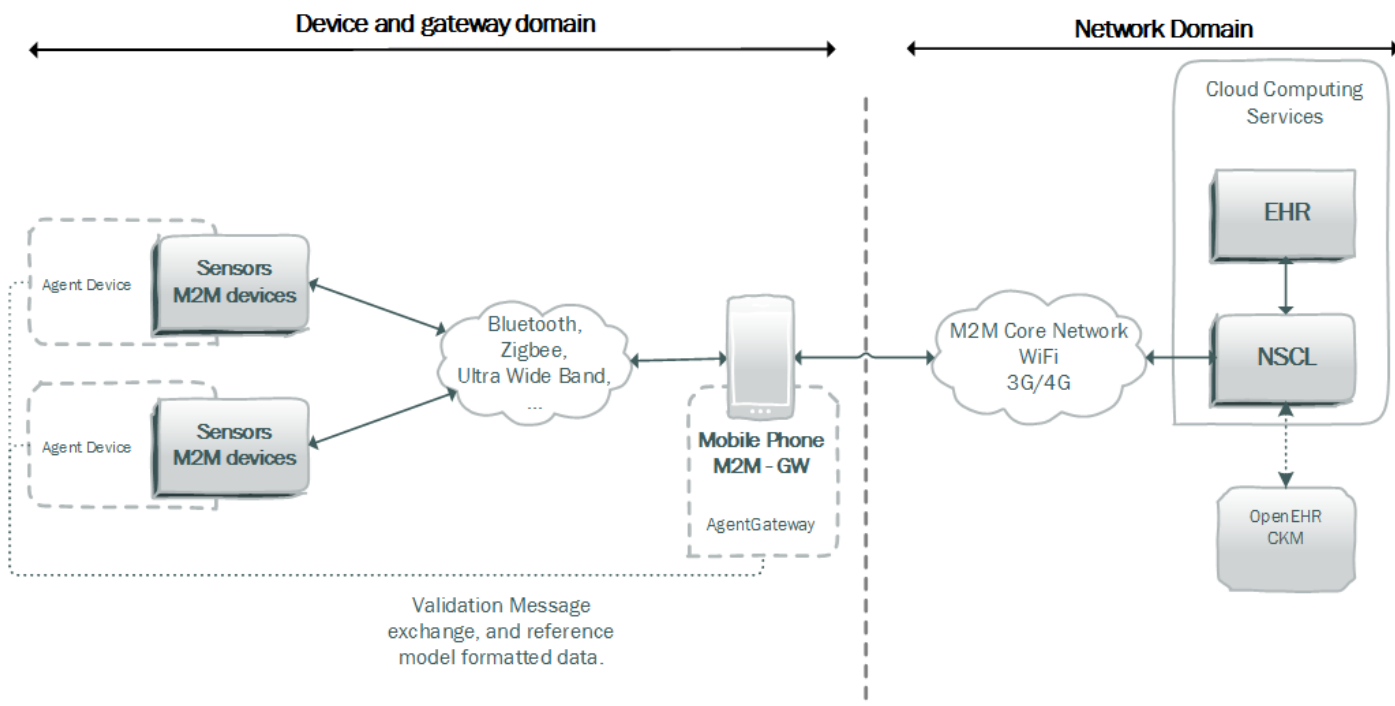


Figura 4-2: Arquitetura para um ambiente M2M.

Finalmente para prover uma maior escalabilidade, garantir resiliência dos dados, prover mecanismos de segurança e diminuir os custos de implantação, foi decidido acrescentar os serviços em nuvem nesta arquitetura.

A computação em nuvem tem muitas vantagens no setor de *e-Health* [9]. O recurso de infraestrutura como um serviço habilita em grande escala o desenvolvimento de sistemas de saúde com recursos limitados. A infraestrutura de *hardware* na nuvem aumenta, dependendo das necessidades do sistema estabelecido. Então, decidimos desenvolver as funcionalidades do NSCL empregando três técnicas de comunicação as quais vão ser comparadas em capítulos posteriores. As técnicas são: HTTP *Polling*, HTTP *Long Polling* e *Websockets*. Nos casos de HTTP *Polling* e *Long Polling* os recursos no servidor são providos como um conjunto de APIs REST que se comportam como tratado nas seções 2.3.1 e 2.3.2 deste trabalho. Já no caso de *Websockets* se emprega a biblioteca SignalR.

4.2.2 - Diagrama de Implantação

A Figura 4-3 apresenta um diagrama de distribuição/implantação das componentes de *software* e *hardware* da plataforma, tratadas neste capítulo.

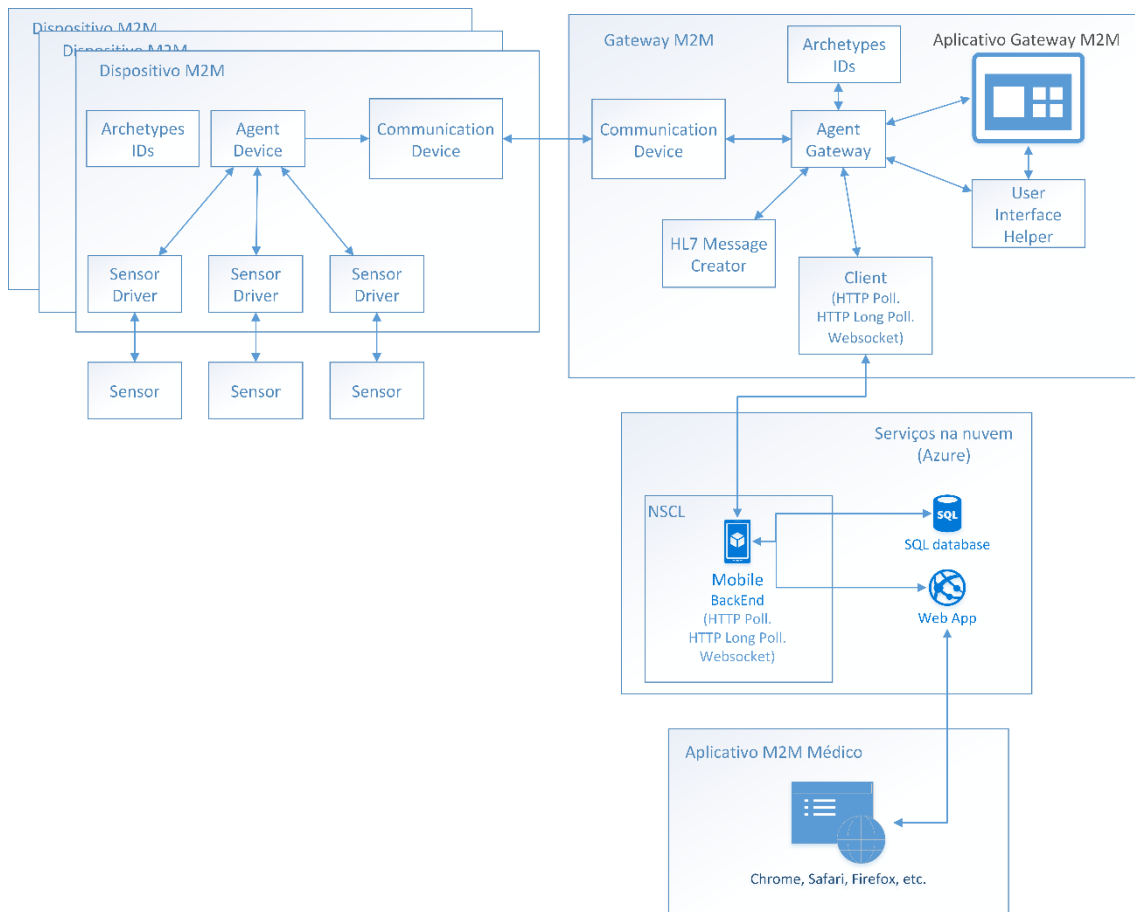


Figura 4-3: Diagrama de Implantação/Distribuição.

Como sistema de captura se consideram o dispositivo M2M (seção 4.3.2) e o gateway M2M (seção 4.3.3). Neles é que ocorre todo o processo de coleta de dados e a interação com o paciente. O NSCL (seção 4.4) localizado na nuvem atua como *backend* para os diversos aplicativos desenvolvidos no gateway além de prover serviços para aplicativo Web disponibilizado para os profissionais da saúde e para o próprio usuário (paciente). O aplicativo M2M Médico (seção 4.5) é uma interface visual desenvolvida para o acesso aos dados recolhidos e para a criação de mensagens, notificações e e-mails que permitem a comunicação ativa entre ambos participantes. A seguir, cada uma dessas entidades é detalhada, abordando problemas apresentados durante o processo de desenvolvimento da plataforma.

4.3 - SISTEMA DE CAPTURA E GATEWAY M2M

Tal foi tratado anteriormente, os diferentes fabricantes de equipamentos empregam seus próprios protocolos nos dispositivos de captura de dados. Por exemplo, medidores

de pressão sanguínea podem disponibilizar os dados de diferentes maneiras ou intervalos dependendo da localização geográfica, etc. A pressão diastólica (pressão a qual o coração consegue vencer) e a sistólica (pressão constante exercida em estado natural pelo coração) podem variar no intervalo na qual elas são disponibilizadas. Por exemplo: no Brasil uma pressão de 12 diastólica e 8 sistólica pode ser representada em outros países como 120 diastólica e 80 sistólica. Outro exemplo é o do termômetro e a temperatura corpórea, a qual pode ser tratada em diferentes medidas de grandeza como grau Celsius ou graus Fahrenheit. Por este motivo foi decidido tratar o problema da compatibilidade entre os diferentes sensores orientados a um mesmo alvo de medições.

4.3.1 - Problemas na escalabilidade do sistema de captura

Definir um esquema que permita estabelecer internacionalmente uma padronização neste tipo de grandezas não é impossível. As definições do OpenEHR permitem restringir este tipo de observações mediante a definição da semântica médica referente a dita observação.

Neste sentido, propomos a extensão da semântica OpenEHR para o domínio M2M. Na definição OpenEHR, um modelo de referência descreve o modo como os dados devem ser tratados. Por exemplo, considere o arquétipo "temperature.v1 OpenEHR-EHR-OBSERVATION.body" fornecido no OpenEHR *Clinical Knowledge Manager* (CKM) [18] para a temperatura corpórea. Ele aponta para um modelo de referência que tem sete atributos obrigatórios (temperatura corporal, data e hora, etc.) e nove opcionais. A semântica na definição do arquétipo restringe o comportamento do Modelo de Referência para graus Celsius em uma precisão de três casas decimais.

Se um desenvolvedor de termômetro eletrônico cria algum equipamento que cumpre todas essas restrições semânticas, pode-se dizer que o equipamento atendeu a todas as restrições arquétipo ou é "temperature.v1 OpenEHR-EHR-OBSERVATION.body" compatível.

Neste sentido criamos a seguinte definição:

X compatível: Seja *X* um arquétipo referente a uma observação definido no OpenEHR. Um equipamento *E* é “*X* compatível” se as observações capturadas por *E* cumprem com todas as restrições definidas em *X*.

Então um arquétipo OpenEHR pode ser utilizado como uma identificação do tipo de dados fornecido pelo equipamento e as restrições de dados que o sistema tiver que cumprir. Em nossa arquitetura, criamos os seguintes agentes no lado da rede M2M para tentar representar essas restrições:

- *AgentDevice*: localizado no dispositivo M2M, que fornece comunicação com o *AgentGateway* e armazena um registo com a identificação dos arquétipos *Xs* para os quais ele é *X* compatível;
- *AgentGateway*: localizado no M2M gateway. Armazena registros dos Arquétipos relevantes para a aplicação de triagem e fornece um mecanismo de comunicação bidirecional com o *AgentDevice*.

Quando um *AgentGateway* detecta a presença de um *AgentDevice* na rede, uma troca de mensagens pode ser iniciada como se segue:

- *AgentGateway* solicita os arquétipos *Xs* para os quais o *AgentDevice* é *Xs* compatível;
- *AgentDevice* responde com uma lista com pelo menos um ID de arquétipo;
- O *AgentGateway* detecta os parâmetros compatíveis através de uma comparação entre os ID de arquétipos recebidos e os ID alvo da triagem;
- O *AgentGateway* envia uma mensagem com as coincidências como parâmetro;
- O *AgentDevice* inicializa os sensores correspondentes e inicia a transmissão dos dados.

Até este ponto podemos contar com uma definição que permite associar observações de parâmetros fisiológicos com restrições descritas na semântica médica do OpenEHR. Cabe esclarecer que neste trabalho se assume que a garantia de se um equipamento é *X* compatível se resolve através de avaliações subjetivas e que não vai ser provido um mecanismo automático para esta determinação.

O segundo desafio em relação ao problema da escalabilidade dos sensores envolve a definição de uma estrutura de dados computacional padronizada para o armazenamento e envio de observações relativas aos parâmetros. Na nossa proposta, as estruturas de dados, suportadas no processo de detecção, são geradas automaticamente. Para isto foi desenvolvido um *parser* que a partir dos arquétipos e modelos de referência definidos no OpenEHR gera estruturas nas linguagens C++ e Java. C++ foi usado como primeira

linguagem devido ao grande número de propostas baseadas em Arduino. No entanto, num futuro se pretendem criar novos geradores para linguagens como Java, Python, etc.

No processo de desenvolvimento de *software* para o dispositivo M2M, tais tipos gerados permitem a normalização do formato dos dados garantindo a interoperabilidade. Cada estrutura de objeto gerada expõe as seguintes funcionalidades:

- 1- Construtor: recebe como parâmetros os diferentes dados capturados e retorna uma instância da observação.
- 2- Serializer: Serializa a estrutura em formato XML.
- 3- Deserializer: Recebe como parâmetro uma estrutura serializada e retorna uma instância do objeto.

Quando os dados brutos capturados pelo sensor são recebidos no *DeviceAgent*, este os armazena em uma estrutura orientada a objetos C++ que permitem a serialização em um formato compatível com uma comunicação serial através da conexão por exemplo Bluetooth.

Então uma abstração dos dispositivos médicos pode ser obtida e as aplicações de saúde podem ser compatíveis com um maior número de sensores. A compatibilidade com os dispositivos M2M é garantida através de um formato global, orientada pelo Modelo de Referência definido no OpenEHR, enquanto a precisão dos dados pode ser restringida pela semântica dos arquétipos nas fases de desenvolvimento. Logicamente, um processo de avaliação do dispositivo será necessário para garantir que o sensor tenha cumprido todas as restrições dos arquétipos.

Cabe destacar que os dados semânticos não precisam ser explicitamente armazenados no M2MDevice, sendo a semântica do OpenEHR implicitamente estendida durante a criação do dispositivo M2M por um agente que conhece o ID dos arquétipos para os quais todas as restrições estão em conformidade.

A Figura 4-4 resume a ideia de estender a semântica do OpenEHR ao contexto dos dispositivos M2M.

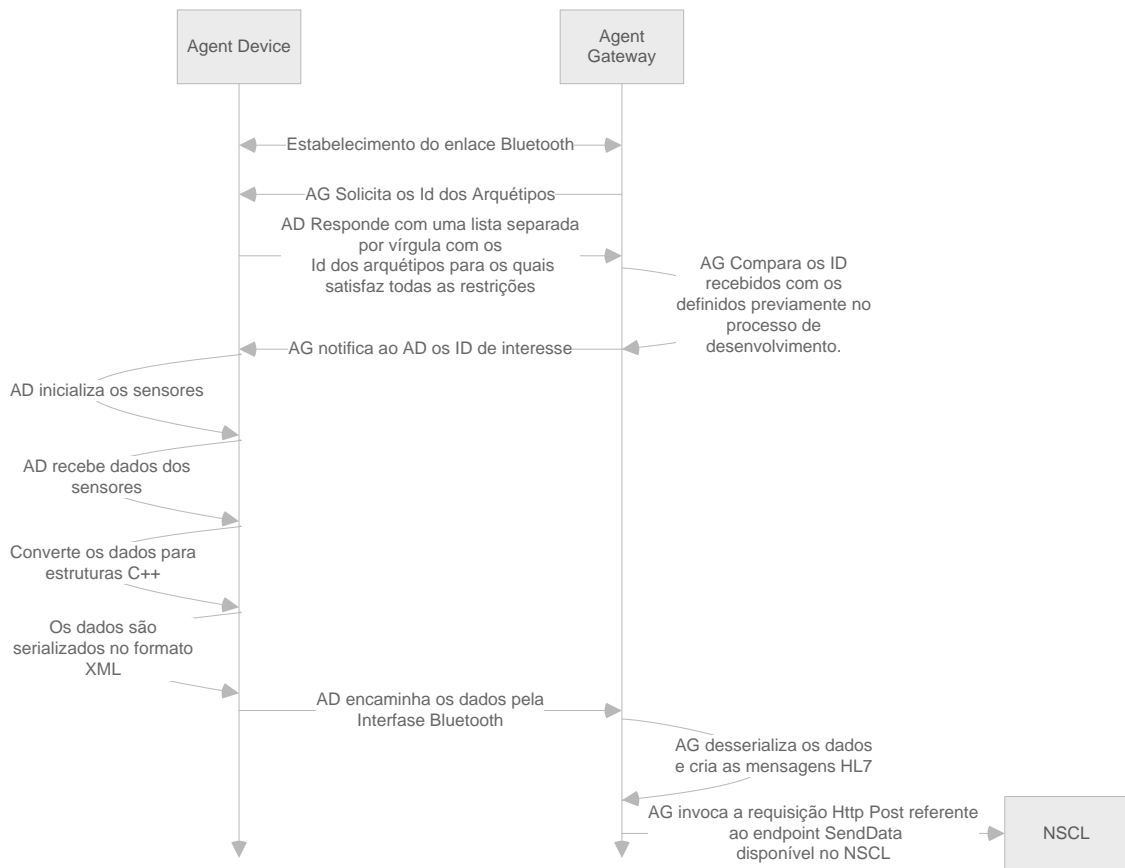


Figura 4-4: Comunicação entre o *AgentDevice* e o *AgentGateway*.

Uma vez exposta a nossa alternativa para padronização nas camadas inferiores da arquitetura prosseguimos a definir formalmente a plataforma.

No sistema de captura temos duas entidades principais: (i) Dispositivo M2M e (ii) Gateway M2M.

4.3.2 - Dispositivo M2M

Para o dispositivo M2M foi desenvolvido um conjunto de bibliotecas que provem suporte ao tratamento dos sensores, comunicação com o Gateway M2M desenvolvimento do *AgentDevice* etc. A Figura 4-5 apresenta o diagrama de classes desenvolvido.

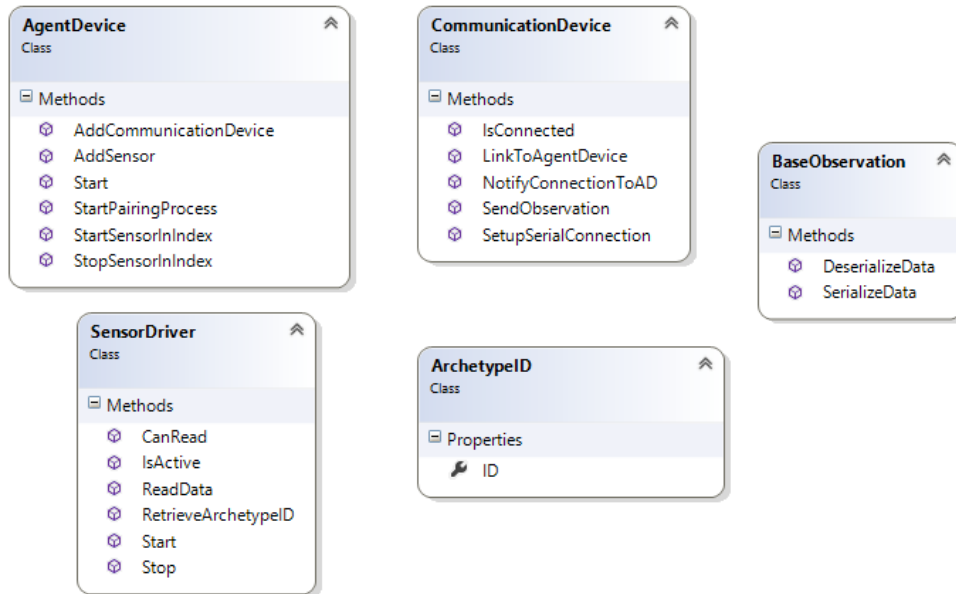


Figura 4-5: Diagrama de classes no Dispositivo M2M.

AgentDevice: O *AgentDevice* é a classe principal no processo de captura. Suas funcionalidades são descritas em seguida.

Tabela 4-1: Funcionalidades do *AgentDevice*.

Função	Parâmetros	Descrição
AddCommunicationDevice	CommunicationDevice cd	Inserir um dispositivo de comunicação cd como uma das interfaces de rede para a comunicação com Gateway M2M.
AddSensor	SensorDriver sd	Inserir o sensor sd na lista de sensores disponíveis para o monitoramento.
Start	Sem parâmetros	Inicializa o <i>AgentDevice</i> . Na chamada prepara a todos os sensores e ao dispositivo de comunicação.
StartPairingProcess	Sem Parâmetros	Inicializa a comunicação com o GatewayDevice através do dispositivo de comunicação. Nesta fase os ID dos arquétipos compatíveis são intercambiados e se inicializam cada um dos sensores.
startSensorInIndex	Int index	Ordena a começar as medições do sensor index
stopSensorIndex	Int index	Ordena a detenção do sensor index.

CommunicationDevice: Classe base da qual tem que herdar todos os dispositivos de comunicação.

Tabela 4-2: Funcionalidades do *CommunicationDevice*.

Função	Parâmetros	Descrição
IsConnected	Sem parâmetros	Retorna se o dispositivo encontrasse no médio de uma conexão estabelecida.
LinkToAgentDevice	<i>AgentDevice</i> ad	Cria uma referência ao <i>AgentDevice</i> ad para fazer o disparo de eventos como conexão sucedida, etc..
NotifyConnectionToAD	Sem parâmetros	Notifica ao <i>AgentDevice</i> de uma conexão bem sucedida
SendObservation	BaseObservation bo	Encaminha para o <i>AgentGateway</i> a observação bo pela conexão serie.
SetupSerialConnection	Int baudRate	Inicializa a conexão serie com o <i>AgentDevice</i> seguindo a taxa de símbolos baudRate

BaseObservation: Classe base da qual tem que herdar todas as estruturas geradas com o uso da ferramenta anteriormente descrita.

Tabela 4-3: Funcionalidades do *BaseObservation*.

Função	Parâmetros	Descrição
DeserializeData	Sem parâmetros	Serializa o objeto para ser encaminhado via a comunicação serie.
SerializeData	Sem parâmetros	Des serializa o objeto, para ser processado no <i>AgentGateway</i>

SensorDriver: Classe base da qual tem que herdar todos os sensores empregados no processo de captura.

Tabela 4-4: Funcionalidades do *SensorDriver*.

Função	Parâmetros	Descrição
CanRead	Sem parâmetros	Retorna se o sensor conta com dados para serem lidos.
IsActive	Sem parâmetros	Retorna se o sensor está sendo empregado na captura.
ReadData	Sem parâmetros	Retorna um arreglo de BaseObservation que contém as observações capturadas até o momento e que ainda não tem sido lidas.
RetrieveArchetypeID	Sem parâmetros	Retorna uma coleção dos arquétipos para os quais o sensor é compatível.
Start	Sem parâmetros	Inicializa o sensor
Stop	Sem parâmetros	Ordena a detenção do sensor index.

ArchetypeID: Classe empregada para representar os arquétipos para os quais o sensor é compatível.

Finalmente o processo de desenvolvimento de um novo Dispositivo M2M empregando esta plataforma segue o fluxo descrito na Figura 4-6 Cabe destacar que neste diagrama se simplificaram algumas etapas de desenvolvimento como a implementação do *SensorDriver* e do *CommunicationDevices*.

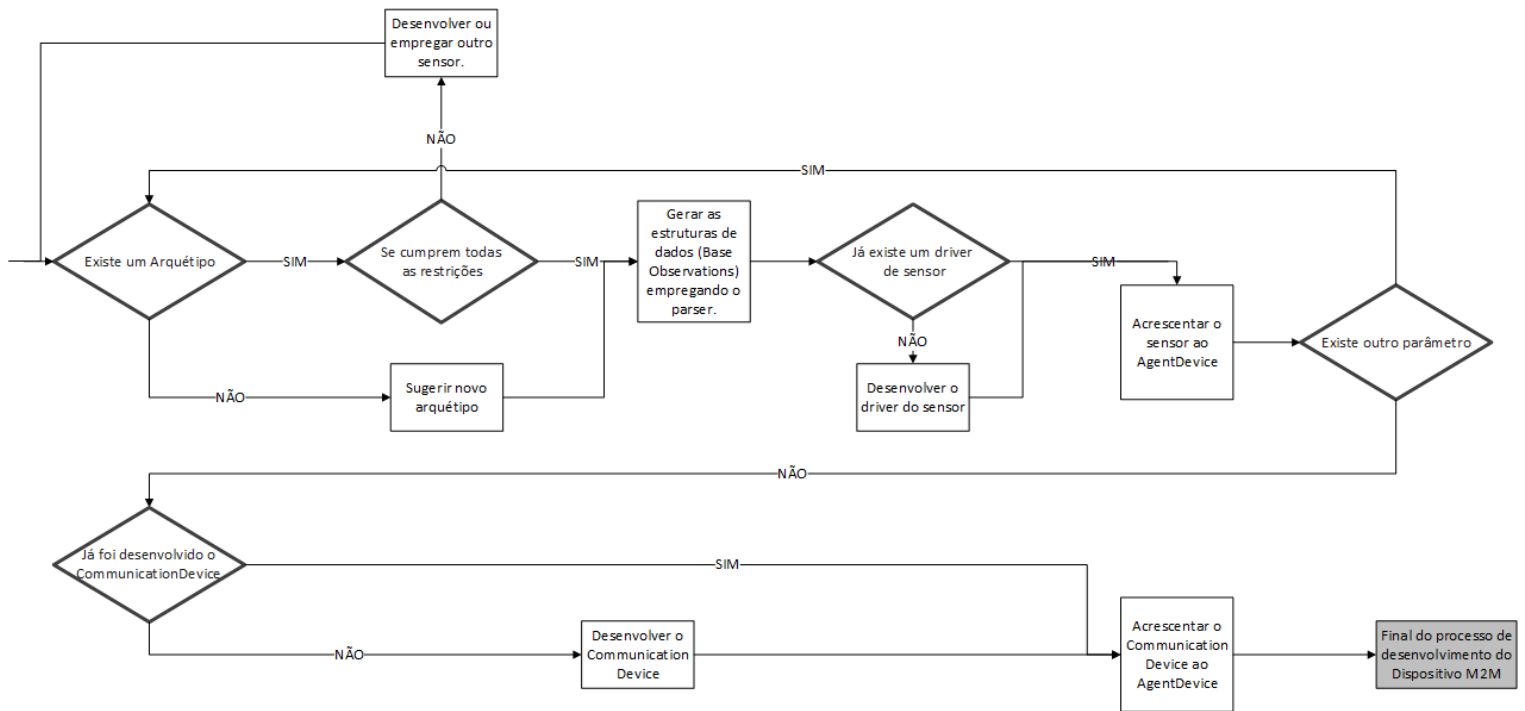


Figura 4-6: Diagrama de fluxo de desenvolvimento do Dispositivo M2M.

4.3.3 - GATEWAY M2M

Para o gateway M2M foi desenvolvido um conjunto de bibliotecas que brindam suporte à comunicação com o Dispositivo M2M e para consumir os diferentes serviços do NSCL. A Figura 4-7 apresenta o diagrama de classes desenvolvido.

Tal foi tratado anteriormente o padrão HL7 v2.7 define um conjunto de mensagens para prover comunicação entre o NSCL e o gateway M2M. Neste sentido a nossa proposta trabalha de um jeito similar à alternativa apresentada por C. Pereira, et al. [39], mas com as diferenças de que agora essas mensagens vão ser tratadas como recursos representadas na notação de objetos do *JavaScript* JSON. Finalmente as mensagens ORU e mensagens para tratamento demográfico do paciente são criadas com o fim de prover a integração com o *backend* REST.

O diagrama de classes da Figura 4-7 expõe as diferentes funcionalidades providas na plataforma.

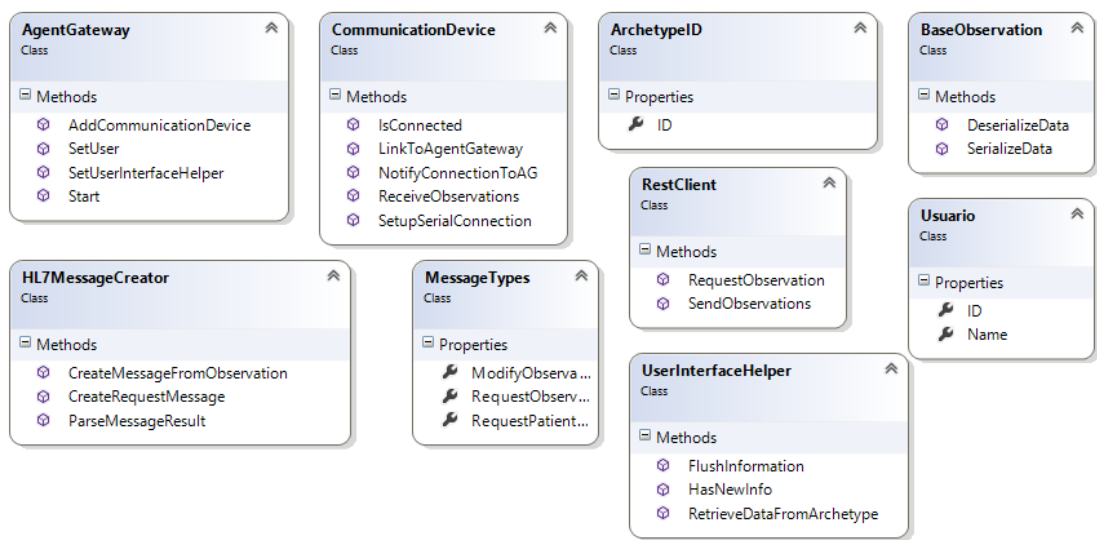


Figura 4-7: Diagrama de classes no Gateway M2M.

AgentGateway: Classe principal do Gateway M2M. Tem como objetivo prover mecanismos de comunicação entre o dispositivo M2M e o NSCL.

Tabela 4-5: Funcionalidades do *AgentGateway*.

Função	Parâmetros	Descrição
AddCommunicationDevice	CommunicationDevice cd	Insere um dispositivo de comunicação cd como uma das interfaces de rede para a comunicação com Gateway M2M.
SetUser	User u	Armazena a informação referente ao usuário que será empregada na criação de todas as mensagens.
Start	Sem parâmetros	Inicializa ao <i>AgentGateway</i> . Na chamada prepara a todos ao dispositivo de comunicação e inicia a processo de troca de mensagens para validação dos arquétipos.
SetUserInterfaceHelper	UserInterfaceHelper uih	Armazena o helper uih.

CommunicationDevice: Classe base da qual tem que herdar todos os dispositivos de comunicação.

Tabela 4-6: Funcionalidades do *CommunicationDevice*.

Função	Parâmetros	Descrição
IsConnected	Sem parâmetros	Retorna se o dispositivo encontrasse no médio de uma conexão estabelecida.
LinkToAgentGateway	AgentGateway ad	Cria uma referência ao <i>AgentGateway</i> ad para fazer o disparo de eventos como conexão sucedida, etc..
NotifyConnectionToAG	Sem parâmetros	Notifica ao <i>AgentGateway</i> de uma conexão bem sucedida
ReceiveObservation	BaseObservation bo	Recebe uma observação do <i>AgentDevices</i> pela conexão serie. Passa a notificação para o <i>AgentGateway</i>
SetupSerialConnection	Int baudRate	Inicializa a conexão serie com o <i>AgentDevice</i> seguindo a taxa de símbolos baudRate

MessageType: Descreve os diferentes tipos de mensagens que o HL7MessageCreator é capaz de criar.

Finalmente podemos resumir que no desenvolvimento do Gateway M2M se tem que passar por três etapas:

1. Desenvolvimento do *AgentGateway*;
2. Configuração do cliente REST;
3. Desenvolvimento da interfase visual.

No primeiro passo, o diagrama da Figura 4-8 expõe o fluxo de desenvolvimento, já os outros dois passos dependem de: (i) questões próprias do *backend* desenvolvido, pois se precisam das URL, os cabeçalhos HTTP requeridos pelo servidor para tratar segurança e (ii) das funcionalidades que se querem disponibilizar para o paciente através da interfase visual.

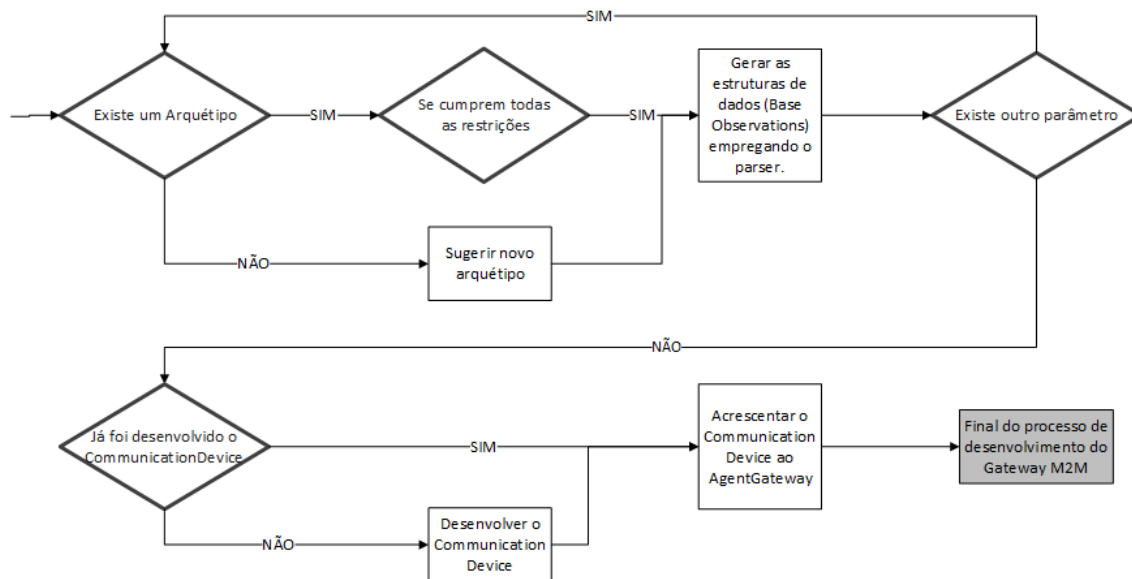


Figura 4-8: Diagrama de fluxo de desenvolvimento do Gateway M2M.

Finalmente passamos à descrição das funcionalidades do NSCL.

4.4 - NETWORK SERVICE CAPABILITY PROVIDER.

As seguintes funções API foram criadas como serviços REST para hospedar na nuvem:

- *Login*: Usa o método HTTP Post e recebe o ID de usuário e a senha.
- *SendData*: emprega o método HTTP POST e recebe uma mensagem HL7 v2.7 ORU com os dados de sensoriamento codificados. Esta função processa as mensagens HL7 e inserir os dados no EHR.

As seguintes restantes APIs foram desenvolvidas para outras aplicações M2M que solicitam informações do EHR:

- *RequestData*: ele recebe uma solicitação de mensagem HL7 v2.7, recupera as informações do EHR e retorna uma mensagem HL7 solicitada com os dados codificados.
- *NotifyPatient*: ele recebe uma ID do paciente e uma mensagem e notifica o M2MSG Através do Sistema de mensagens Google Cloud.

Finalmente, para criar o EHR baseado no OpenEHR, são providos dois *scripts* de TRANSACT-SQL para a criação e população inicial do banco de dados. Os usuários da plataforma tem que ter em conta como é que os *script* vão ter acesso ao banco de dados.

Uma possível solução é o emprego do mesmo esquema num serviço móvel que tenha como tabelas referenciadas as mesmas que as criadas com o *script*.

Foram considerados provedores de serviços em nuvem capazes de habilitar pontos extremos *JavaScript*. Num futuro se pretende desenvolver as funcionalidades do *backend* para diferentes plataformas como PHP, .NET, etc. Logo o fluxo de desenvolvimento do *backend*, incluindo NSCL, Banco de dados, etc. se resume na Figura 4-9.

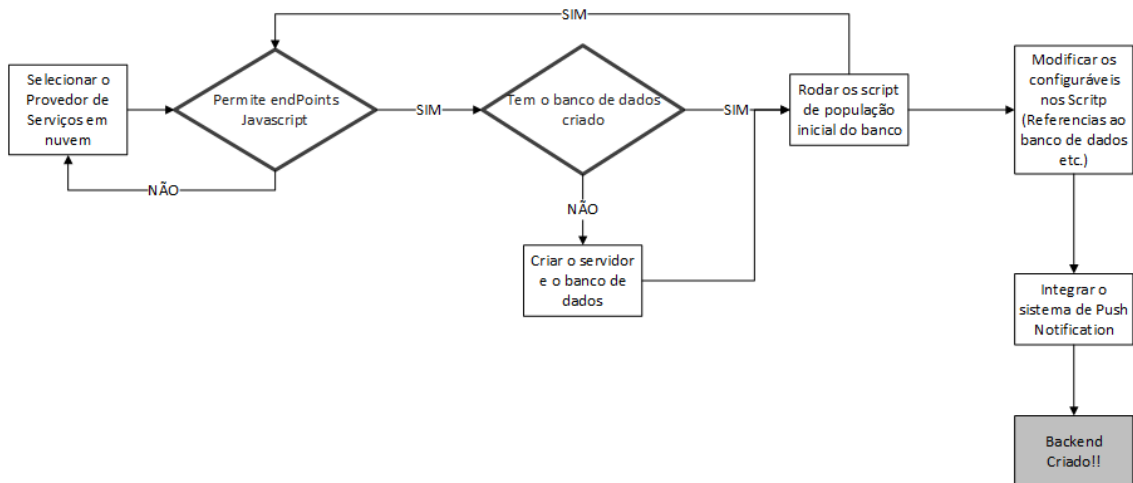


Figura 4-9: Diagrama de fluxo de desenvolvimento do Gateway M2M.

4.4.1 - Comunicação entre o gateway M2M e o NSCL

Na primeira abordagem os serviços do NSCL são providos como um conjunto de APIs REST hospedadas na nuvem ao mesmo tempo que os aplicativos M2M Web e o sistema de armazenamento para o EHR.

A Figura 4-10 mostra cada uma das entidades da rede considerada nesta dissertação e resume como elas vão interagir entre si.

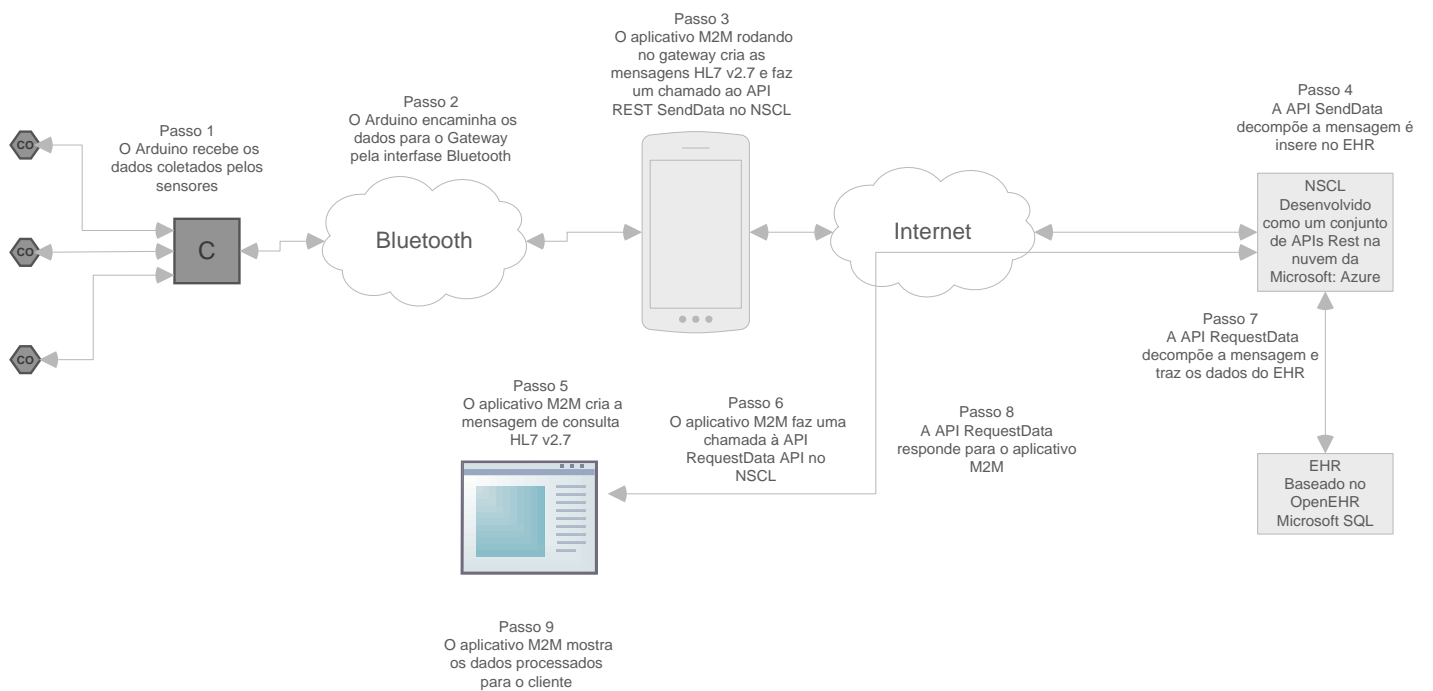


Figura 4-10: Resumo da Arquitetura empregada.

4.4.2 - HTTP Polling NSCL

Para esta primeira proposta, tanto o Gateway M2M assim como o aplicativo M2M do lado médico vão se comportar como clientes *Long Polling*. A cada momento esses clientes encaminham requisições às APIs correspondentes tentando obter uma resposta.

No caso das APIs REST no servidor elas se comportam como um *endpoint* REST que vai dar resposta automaticamente logo depois de receber a requisição.

A Figura 4-11 serve para uma maior compreensão do comportamento desta técnica.

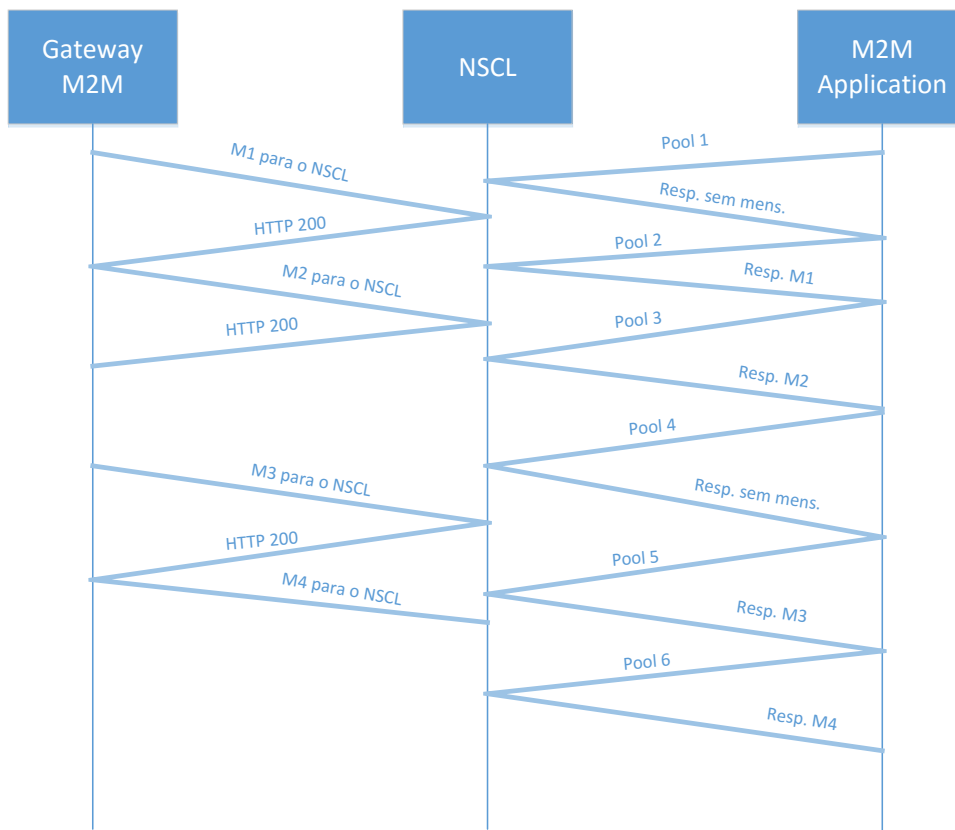


Figura 4-11: Comportamento da técnica de HTTP *Polling*

Pode-se apreciar que no caso do HTTP *Polling* se tem requisições para as quais o NSCL ainda não tem resposta, o que leva a consumo de banda maior e a maior atraso introduzido pelo fato de que o cliente assume que vai ter uma resposta e não faz uma requisição até a chegada de uma resposta.

4.4.3 - HTTP Long Polling NSCL

Já na versão de *Long Polling* o servidor tem um comportamento diferente. Quando um cliente solicita por exemplo uma observação, a API correspondente consulta se já essa observação encontra-se disponível. Caso positivo, retorna automaticamente para o cliente com a mensagem ORU que contém os dados referentes à observação. Caso negativo, a API continua aguardando até a chegada dos dados ou de estourar o tempo definido para a requisição.

A Figura 4-12 mostra como se comporta a técnica de *Long Polling* no processo de triagem na nossa plataforma.

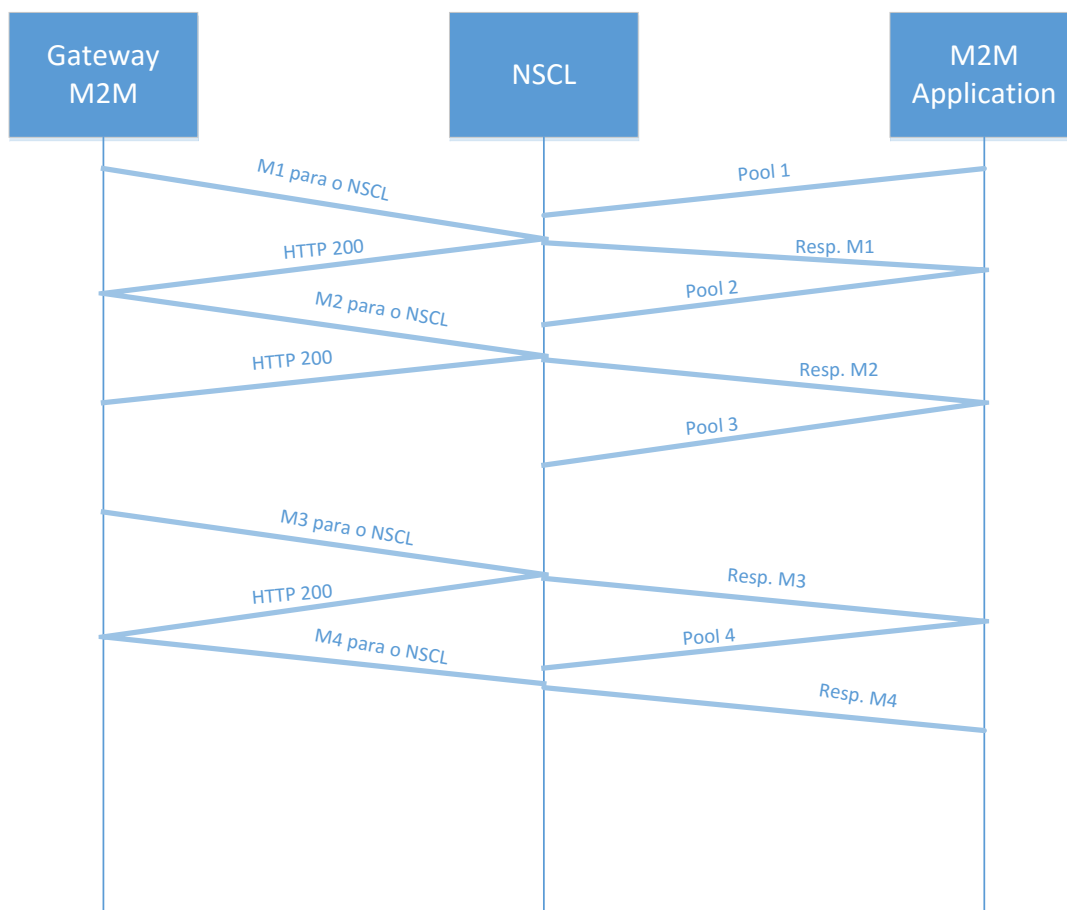


Figura 4-12: Técnica de HTTP Long *Polling*

Já nesta proposta, se diminui a quantidade de requisições feitas pelo cliente. O NSCL tenta aguardar para dar resposta ao cliente até a chegada da mensagem. O delay também diminui pelo fato de que cada vez que uma mensagem chega do Gateway M2M se encaminha para o cliente, caso de estar numa situação de “Long” *Polling*.

4.4.4 - *Websocket NSCL*

No caso da versão que implementa *Websocket*, um cliente é capaz de solicitar um determinado dado encaminhando a mensagem ORU correspondente como um fluxo de bytes pela comunicação bidirecional. Uma vez obtida e codificada dentro da mensagem ORU, a resposta é empurrada para o cliente. Consideramos nunca fechar a conexão devido a que outras mensagens relativas a e-mails, chat, notificações, etc. vão ser mandadas pelo mesmo socket. Isto fez que diminuía a latência acrescentada pelos *Handshakes* HTTP da comunicação através do socket.

A Figura 4-13 mostra um exemplo do comportamento desta proposta no nosso

sistema.

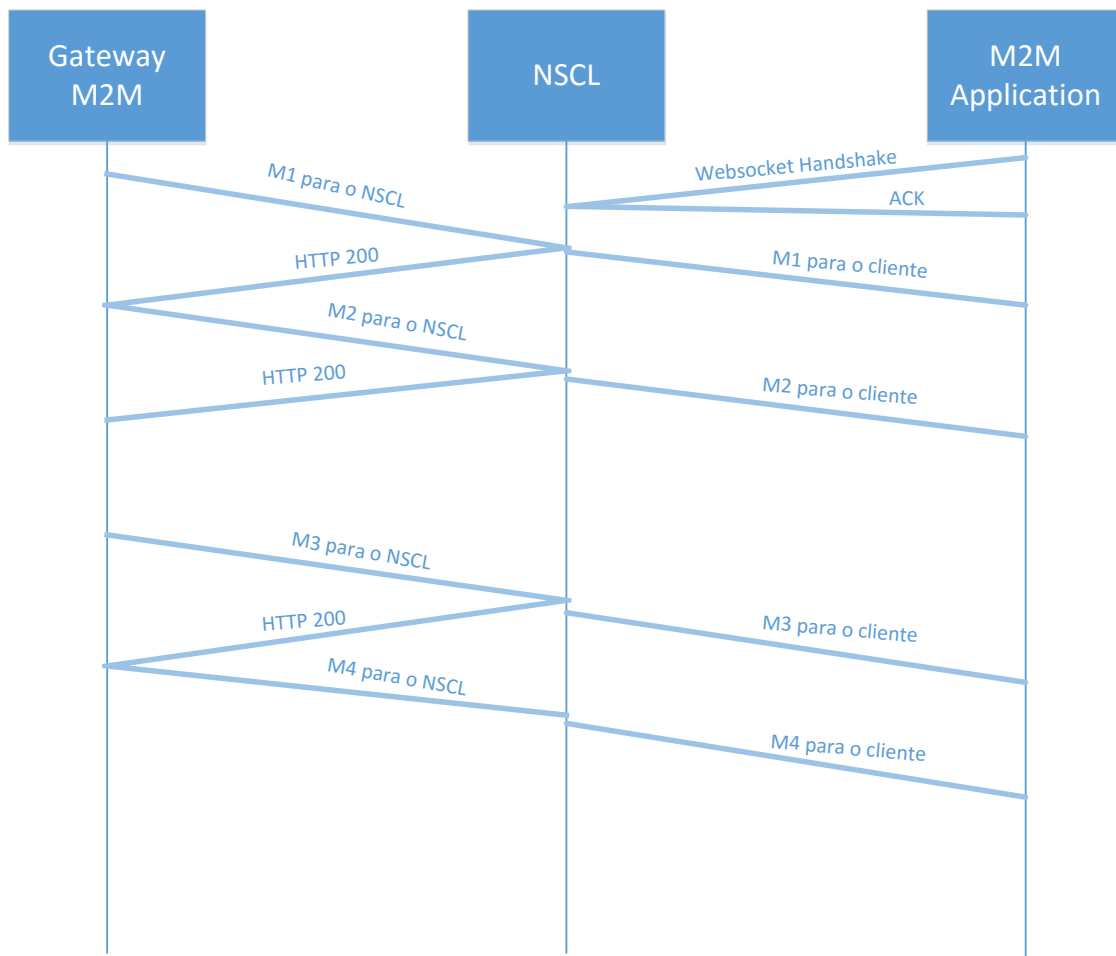


Figura 4-13: Exemplo do comportamento da proposta baseada em *Websockets*.

Em termos de protocolo, esta abordagem é a que menos dados de controle emprega na camada de aplicação, mas será que verdadeiramente é a de melhor desempenho? Para dar resposta a esta interrogante, em seções futuras se faz uma comparação com em cenários reais que permitirá estabelecer uma jerarquia em termos de desempenho.

4.5 - APLICATIVOS M2M

Considerando ambos atores (paciente, médico) foi decidido criar duas interfaces visuais como entrada ao sistema.

4.5.1 - Aplicativo gateway M2M

É focado no paciente e permite controle de acesso mediante usuário e senha. Dependendo do tipo do NSCL, consome serviços do servidor para validar a autenticidade

do paciente.

Quando paciente inicia o aplicativo se pede para ele inserir as credenciais, logo de autenticado, se passa à fase de *pairing* com o DeviceM2M. Nesta etapa, dependendo do protocolo e tecnologia de comunicação, se mostram os dispositivos disponíveis e se aguarda à escolha de um deles. Logo depois, o paciente pode iniciar o monitoramento quando quiser.

As figuras a seguir mostram as telas para cada um desses passos.

4.5.2 - Aplicativo M2M do lado médico

Foi desenvolvido como um aplicativo Web usando a tecnologia ASP.Net e hospedado no serviço de aplicativo da nuvem da Microsoft Azure (*apneaplatform.azurewebsites.net*). Está composto pelas seguintes funcionalidades:

Cadastro Figura 4-14: Permite o cadastro de um médico que vai usar a plataforma.

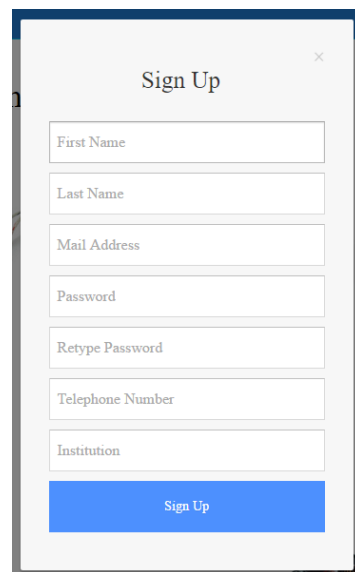
A screenshot of a web-based 'Sign Up' form. The form is titled 'Sign Up' and includes a close button (X) in the top right corner. It contains several input fields: 'First Name', 'Last Name', 'Mail Address', 'Password', 'Retype Password', 'Telephone Number', and 'Institution'. At the bottom of the form is a blue button labeled 'Sign Up'.

Figura 4-14: Tela de cadastro

Login Figura 4-15: Permite autenticação pela parte do profissional da saúde, serve como controle de acesso dado que os pacientes também podem acessar via portal.

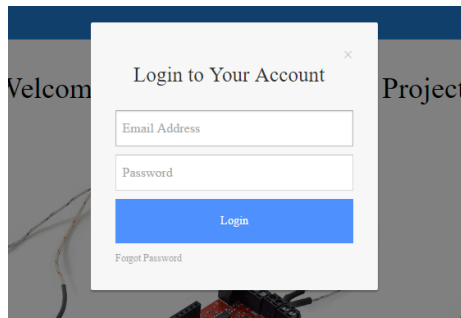


Figura 4-15: Tela de autenticação

Register Patient Figura 4-16: Permite o cadastro de um paciente. Só um médico pode cadastrar um paciente.

Figura 4-16: Tela de cadastro de novo paciente

E-mail Figura 4-17: Com esta funcionalidade ambos atores podem trocar mensagens seguindo o padrão de troca de informação via e-mail. Cabe destacar que como não é o foco deste trabalho, não foi empregado nenhum padrão para manutenção de pastas e nenhum protocolo específico para a criação envio e manutenção de servidores e-mail tipo POP, SMPT, etc.

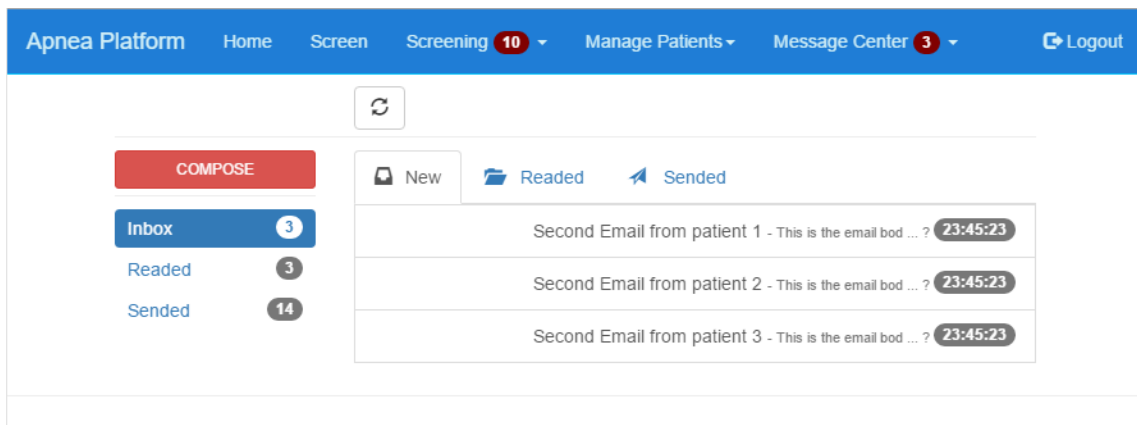


Figura 4-17: Tela de gerenciamento de e-mails

Listar Pacientes, Figura 4-18: Permite verificar o estado de cada paciente, filtrar pelos diferentes atributos, etc.

Show entries Search:

First name	Last name	E-mail	National Health ID	Telephone
Patient	1	patient1@email.com	0000001	1111111111
Patient	2	patient2@email.com	0000002	1111111112
Patient	3	patient3@email.com	0000003	1111111113

Showing 1 to 3 of 3 entries Previous **1** Next

Figura 4-18: Lista de Pacientes

Screening, Figura 4-19: Nesta tela se podem analisar os gráficos gerados a partir dos dados coletados do paciente. Por exemplo, a Figura 4-19 mostra o gráfico correspondente a medições de ECG com problemas. Neste sentido o médico pode notificar ao paciente da incorreta utilização ou colocação dos sensores, via notificações empurradas ou via chat.

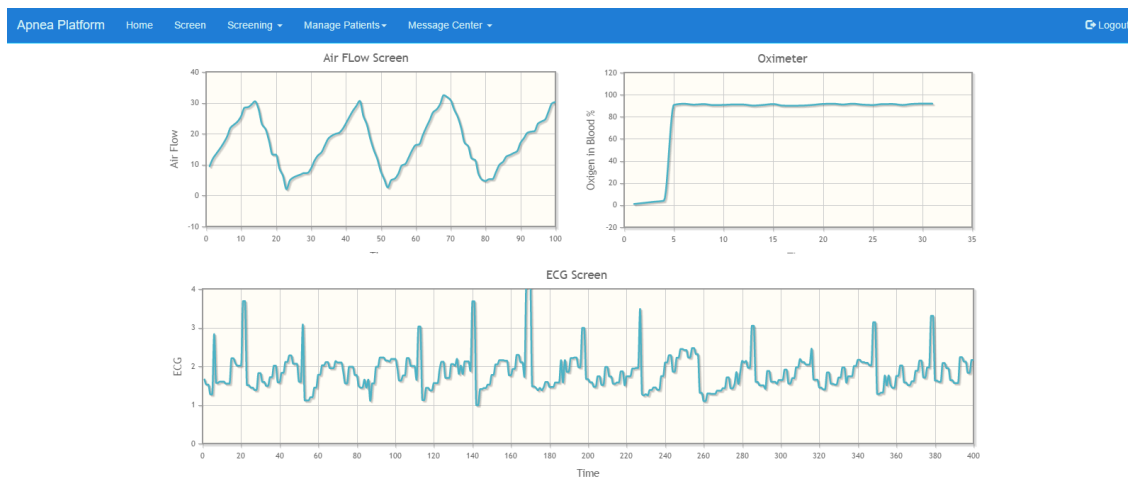


Figura 4-19: Tela de seguimento dos dados em tempo real.

Chat: Funcionalidades comuns de sala de chat privativa entre o paciente e o médico. Foi desenvolvido empregando *Websocket* e mensagens marcadas com o identificador de chat.

Push Notifications: Empregando os servidores da Google GCM se identifica cada instalação de um Device e determinado aplicativo para logo poder notificar.

4.6 - CONSIDERAÇÕES FINAIS

Neste capítulo foram tratados os requisitos da plataforma a ser desenvolvida, a partir da identificação dos atores participantes, bem como foram tratados alguns dos principais aspectos legais referentes ao emprego da mesma.

Em termos de requisitos funcionais, o mais importante é o de fornecer acesso aos dados colhidos em tempo real, enquanto dos requisitos não funcionais cabe destacar a compatibilidade da plataforma com três técnicas de comunicação sobre o protocolo HTTP consideradas: *Websockets*, *Polling* e *Long Polling*.

A plataforma desenvolvida tem como objetivo, facilitar o processo de desenvolvimento de novos aplicativos voltados para *e-Health*, além de garantir interoperabilidade dos equipamentos, sistemas de comunicação, etc. Os casos de uso, apesar de não apresentados, podem ser construídos e discutidos a partir dos atores considerados e das funcionalidades disponibilizadas pela plataforma.

É importante ressaltar que o desenvolvimento da plataforma foi facilitada devido à

escolha de serviços em nuvem assim como de uma arquitetura M2M. A escolha de serviços em nuvem foi importante dado que a característica de infraestrutura como serviço propiciou simplicidade no momento de implantação inicial, além de aumentar a escalabilidade do sistema no que se refere ao número de usuários. Também, a característica de software como serviço, permite que os usuários finais acessem os dados coletados através do aplicativo desenvolvido e disponibilizado na nuvem.

Por outro lado, a adoção de uma arquitetura M2M para comunicação entre “máquinas” simplificou o processo de desenvolvimento. Seguindo o proposto pelo ETSI [7], o desenvolvimento dos serviços providos no NSCL como recursos REST favorece a simplicidade em termos de integração, já que se empregam as requisições HTTP como padrão de comunicação entre entidades no sistema, sendo o HTTP um protocolo padrão da Internet. Outra vantagem observada no emprego de uma arquitetura M2M foi a de poder contar com um smartphone como Gateway M2M, permitindo tratar assim a mobilidade dos usuários. Esta última característica resulta interessante dado o fato que a “Rede de Área Local” da arquitetura M2M se desloca completamente com o paciente, bem como é utilizada de forma integrada a redes 3G/4G como redes de acesso ao Núcleo da Rede M2M.

Também é importante destacar que o modelo de arquitetura M2M se mostra adequado ao ambiente de coleta de dados, com base em sensores. Os dispositivos de captura, mesmo sendo diferentes, apresentaram boa capacidade de conexão com o Arduino e por via de consequência com o Gateway, facilitando assim a integração de novos sensores à plataforma.

5 - DEFINIÇÃO DE REQUISITOS E DESENVOLVIMENTO DO PROTÓTIPO PARA APNÉIA

São detalhados, neste capítulo, os aspectos tidos em conta no processo de desenvolvimento de uma aplicação para o monitoramento de parâmetros referentes à apnéia do sono, empregando a plataforma desenvolvida.

5.1 - REQUISITOS DA APLICAÇÃO

A criação de um aplicativo ou plataforma para tratar o monitoramento de parâmetros fisiológicos referentes a apnéia do sono exige uma análise dos atores envolvidos, os parâmetros a considerar assim como os requisitos funcionais e não funcionais. Neste sentido, recapitulemos quais são os atores envolvidos tratados na seção 4.1.1 desta dissertação:

Profissional da saúde pneumologista: Especialista focado na detecção e tratamento de apnéia do sono, que indica o monitoramento, processa os dados e controla os parâmetros de medição com o fim de diagnosticar a presença da doença do paciente.

Paciente de apnéia do sono: Indivíduo com suspeitas de padecer apnéia do sono para o qual o diagnóstico com a plataforma de monitoramento voltada para apnéia do sono é monitorado.

Uma vez definidos os atores, procedemos à apresentação dos parâmetros tidos em conta. Da seção 3.3 temos que os parâmetros mais empregados, no processo de diagnóstico de apnéia do sono, são ECG, fluxo de ar e SpO₂, os quais vão ser utilizados neste protótipo. Além desses parâmetros, foi decidido incluir informação do contexto do paciente no aplicativo. Para isso foi determinado encaminhar a localização do paciente e a postura do mesmo:

Localização: Representada em coordenadas (latitude e longitude).

Postura: Refere-se à postura do paciente no momento das observações dos parâmetros fisiológicos (deitado, sentado ou em pé).

Como requisitos funcionais e não funcionais temos:

5.1.1 - Requisitos Funcionais

- 1- Permitir a coleta de dados de ECG, fluxo de ar, SpO2, localização e postura, relativos a pacientes com suspeita de condição de apnéia do sono;
- 2- Favorecer a transmissão dos dados coletados para centro de triagem e para diversos usuários (profissionais da saúde, familiares e o próprio paciente);
- 3- Permitir um registro histórico acerca de cada paciente;
- 4- Permitir o acesso a bases de dados;
- 5- Realizar controle de acesso a dados armazenados e em trânsito;
- 6- Permitir interoperabilidade de equipamentos desenvolvidos segundo a plataforma descrita;
- 7- Garantir compatibilidade com qualquer dos arquétipos de observação definidos no OpenEHR;
- 8- Permitir a visualização em tempo real dos dados sempre que se tenha acesso à plataforma;
- 9- Prover mecanismos de comunicação entre as partes envolvidas (e-mail, chat, notificações).

Como requisitos não funcionais do aplicativo, foram tratados os seguintes:

5.1.2 - Requisitos não Funcionais

O aplicativo deve atender a todos os requisitos não-funcionais definidos anteriormente na seção 4.1.3.

5.2 - ARQUITETURA ALVO E DIAGRAMA DE IMPLANTAÇÃO

A arquitetura seguida foi a mesma que a definida na seção 4.2 deste trabalho. A figura seguinte apresenta o diagrama de implantação próprio para o cenário de apneia do sono:

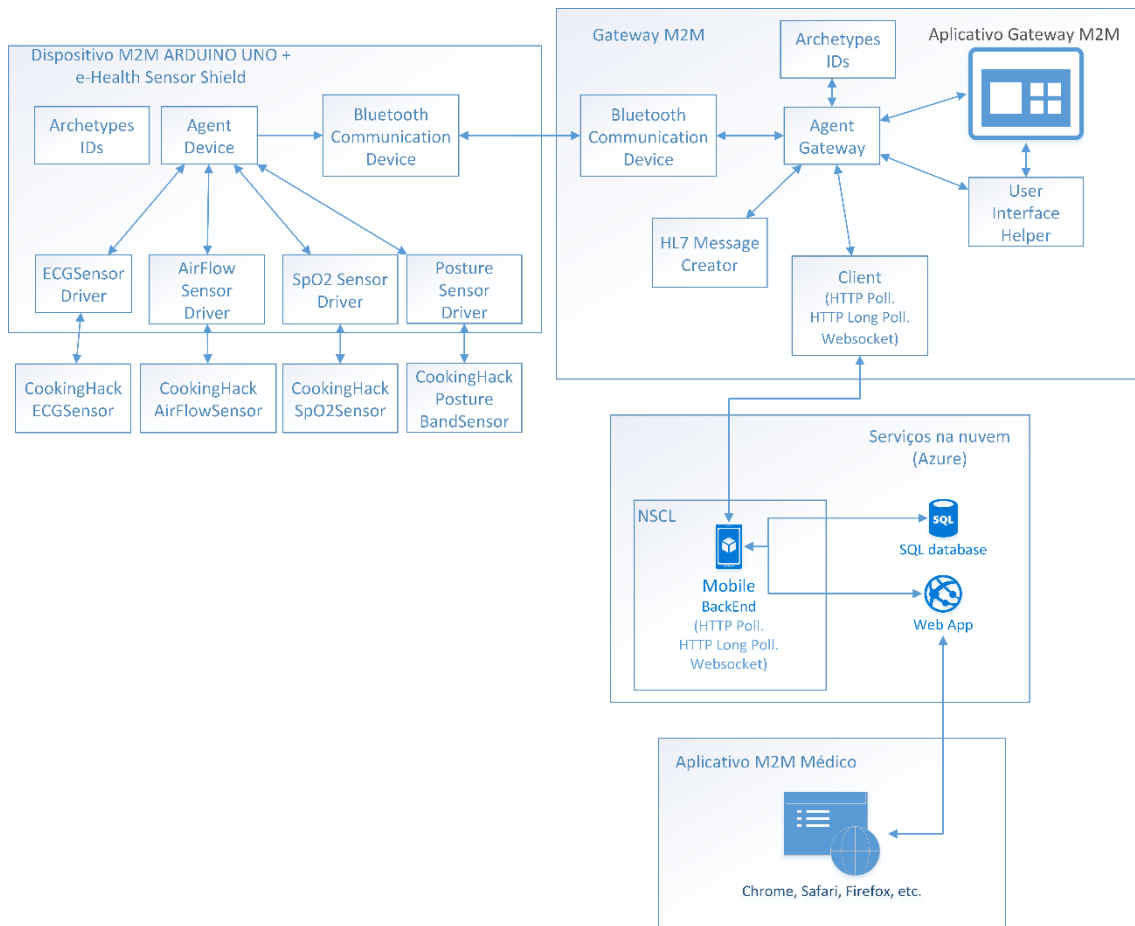


Figura 5-1: Diagrama de Implantação (apnéia do sono).

Pode-se observar que agora a plataforma é instanciada empregando uma placa de desenvolvimento Arduino UNO e o Sensor Shield da CookingHack. Tanto para o gateway quanto para o dispositivo M2M, os IDs dos arquétipos seguidos são especificados posteriormente na seção 5.4 deste trabalho.

5.3 - SISTEMA DE CAPTURA E GATEWAY M2M

5.3.1 - Dispositivo M2M

O Kit de sensores da *CookingHack* [22] e uma plataforma de desenvolvimento Arduino UNO [62] foram empregados na captura dos parâmetros (ECG, fluxo de ar, Oxigênio no sangue). A Figura 5-2 mostra os sensores e a localização de cada um deles no paciente.

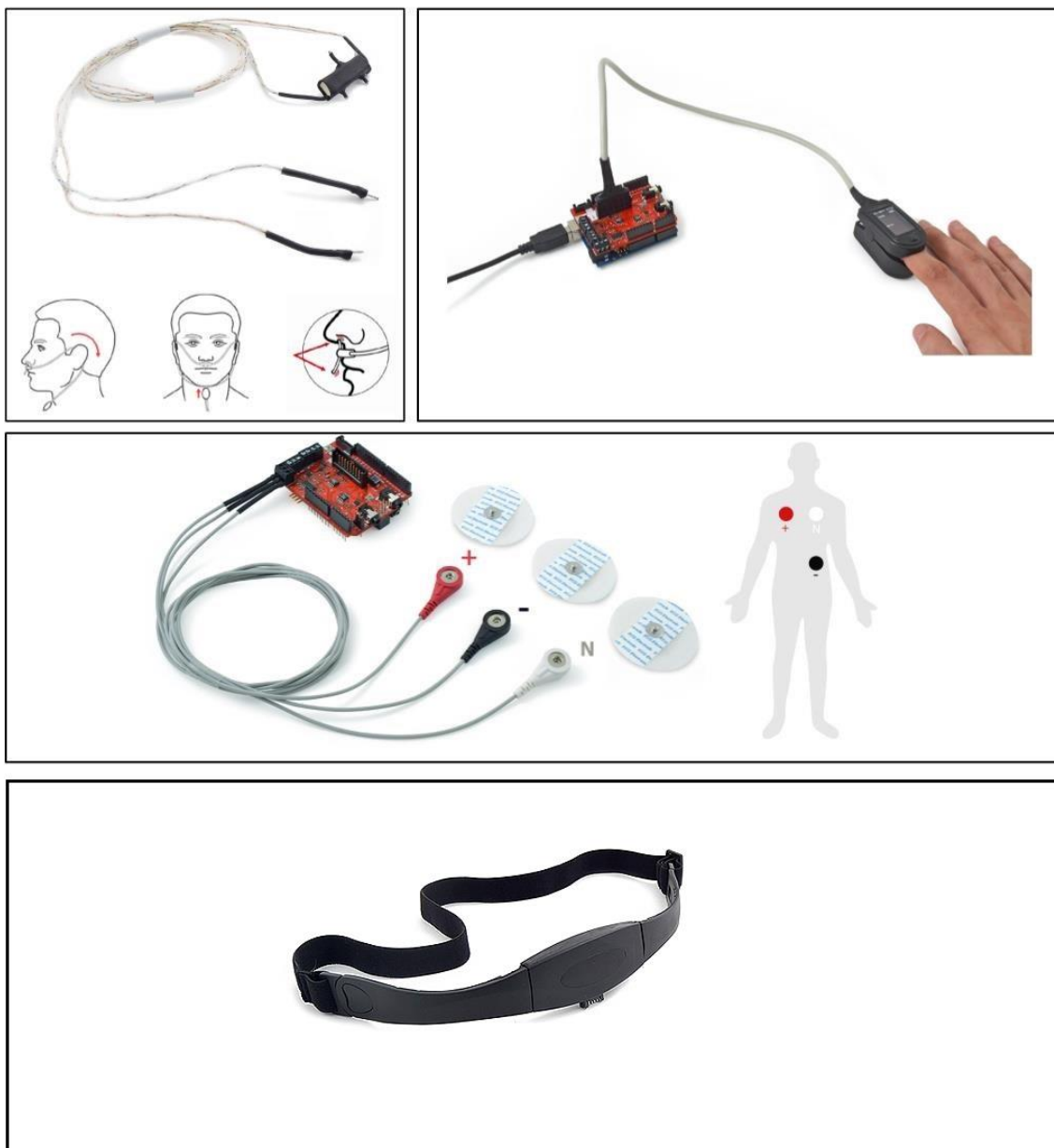


Figura 5-2: Sensores empregados e localização no paciente.

Uma placa HC06 Bluetooth interface do fabricante Guangzhou HC Information Technology Co. serve para a conexão com o M2M gateway.

Verifica-se que já existem arquétipos relativos ao fluxo de ar, ECG e oximetria no OpenEHR. A Tabela 5-1 mostra os arquétipos definidos naquele padrão que podem ser reutilizados para cobrir um possível diagnóstico. Estes arquétipos são o resultado de uma pesquisa feita no CKM do OpenEHR.

Tabela 5-1: Apnéia do sono e o OpenEHR.

ID do Arquétipo	Descrição	Dados	Dados de Estado (Opcional)	Modelo de Referência
openEHR-EHR-OBSERVATION.indirect_oximetry.v1	Método de monitorar as medições de gases sanguíneos, tais como SpO2 e EtCO2 por meio de corrente indiretos e não-invasivos.	- Percentagem de SpO2 (%) -Forma de Onda	- Ambiente de oxigênio - Método de entrega de oxigênio	Observação
openEHR-EHR-OBSERVATION.respiration.v1	Como deveriam ser registradas as características observadas de respiração espontânea como parte de um exame 'sinais vitais'.	- Taxa de Quantidade (Frequência / min) - Rhythm Regular / Irregular	- Ambiente de oxigênio - Fatores de Confusão	Observação
openEHR-EHR-OBSERVATION.ecg.v1	Um eletrocardiograma (ECG) é uma interpretação da atividade elétrica do coração ao longo do tempo, de forma não invasiva gravados utilizando eletrodos de pele externos.	- Parâmetros de derivações (1-12 derivações opcional)	- Nível de esforço - Fatores de Confusão - Tilt	Observação

A Figura 5-3 mostra a localização destes arquétipos no OpenEHR CKM e como eles são definidos.

Pode-se observar que não se tem um arquétipo definido para o posicionamento (postura) do paciente. Isto é devido ao fato de que OpenEHR ainda não tem um arquétipo para definir observações referentes à postura. Para isso nós definimos um arquétipo simples que contém um valor de tipo texto com as opções de: deitado, sentado e em pé, que vai ser empregado só para mostrar a capacidade da plataforma de tratar informações do contexto do paciente.

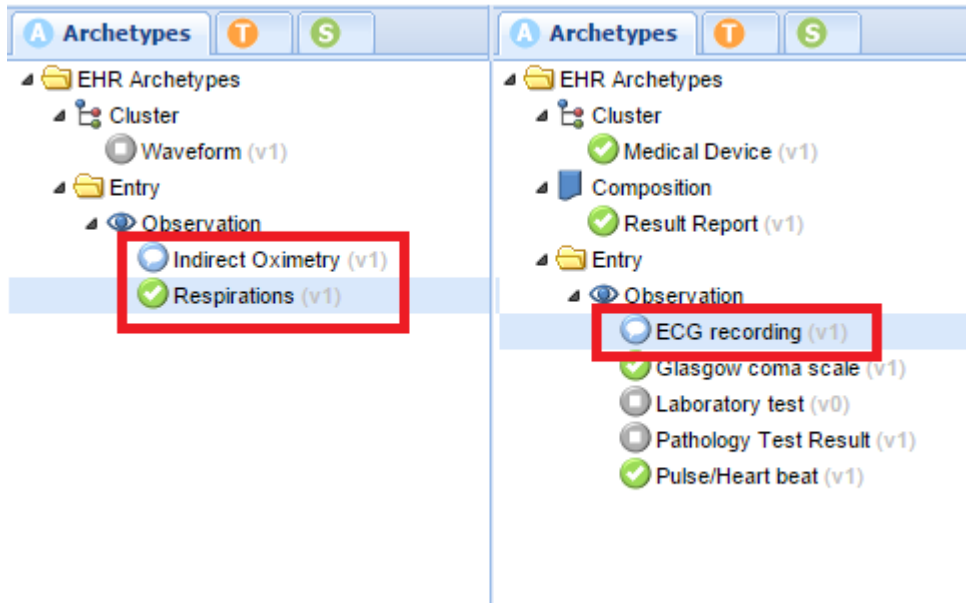


Figura 5-3: Arquétipos do OpenEHR empregados.

O passo seguinte é gerar as estruturas que herdam do *BaseObservation*. Para isto empregamos o *Parser* desenvolvido nesta dissertação e geramos os arquivos:

1. IndirectOximetryObservation.cc
2. RespirationsObservation.cc
3. ECGrecordingObservation.cc
4. PositionObservation.cc

Estes arquivos implementam cada uma das funcionalidades da classe abstrata *BaseObservation* e vão representar uma observação de cada parâmetro a nível de dado computacional.

Uma vez construídos os tipos de dados que vão conter as observações se dá passo a desenvolver os *SensorDriver* para cada um dos sensores empregados. Se tem como resultado 4 classes que neste caso representam o controlador do sensor de fluxo de ar, o controlador do sensor de ECG e o controlador do oxímetro. Cada um destes tipos implementam as funcionalidades da classe abstrata *SensorDriver*.

Logo é preciso desenvolver o *CommunicationDevice*, dado que neste cenário é empregado um modulo Bluetooth, foi criado o arquivo *BluetoothDevice.cc* que cometem a classe do mesmo nome e implementa cada uma das funcionalidades do tipo abstrato *CommunicationDevice*.

Finalmente no *script* principal do Arduino é instanciado o *agenteDevice* e são ligados os controladores dos sensores e o dispositivo de comunicação.

5.3.2 - Gateway M2M.

Como Gateway M2M foi empregado um Smartphone Marca ASUS modelo Zenfone 5 com as seguintes características: Processador Intel Atom 1.4 GHz, 2GB memória Ram, rede de acesso 3G, Wifi, Bluetooth. A aplicação M2M que é executada no gateway M2M foi desenvolvida para Android OS em Java.

Da mesma forma que para o dispositivo M2M se geram os arquivos para a representação dos dados estandardizada. Neste caso são empregados os mesmos arquétipos que na seção anterior, mas são gerados arquivos (classes) na linguagem Java para compatibilizar com o desenvolvimento dos aplicativos no smartphone Android.

Também é desenvolvido o *CommunicationDevice* para controlar o dispositivo Bluetooth do *smartphone*, este dispositivo é ligado ao *AgentGateway* e se configuram os parâmetros próprios à configuração com o *backend*.

Logo o aplicativo pode ser executado e as telas para *Login e display* do monitoramento são mostradas.

5.4 - NETWORK SERVICE CAPABILITY PROVIDER.

Para a representação do NSCL e o EHR foram empregados os Serviços da nuvem da Microsoft conhecida como AZURE. Para o *backend Javascript* se empregou o serviço Mobile Service, o qual possibilita a criação de APIs REST associadas a tabelas específicas do banco de dados. No caso de *Websocket*, o servidor foi desenvolvido junto com o aplicativo Web que atua como aplicativo M2M. Para a representação do EHR foram rodados os *scripts* num servidor SQL SERVER da Microsoft provido também nos serviços em nuvem.

5.5 - CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado um exemplo de uso da plataforma para o monitoramento de parâmetros relativos à apnéia do sono. O protocolo Bluetooth atuou como base na comunicação entre a rede de sensores e o gateway M2M. Arquétipos referentes a ECG, SpO2e fluxo de ar, bem como parâmetros próprios de medições

cardíacas, de saturação de O₂ e respiratórias, foram empregados, permitindo avaliar o correto desenvolvimento da plataforma, além de servir como prova de conceito dos mecanismos de integração desenvolvidos.

6 - AVALIAÇÃO DE DESEMPENHO DA PLATAFORMA

Uma vez desenvolvido o protótipo para monitorar parâmetros referentes à apnéia do sono, procedemos a fazer um estudo do desempenho de cada uma das três técnicas baseadas em "publish/subscribe". Para lograr uma avaliação comparativa, preparamos os seguintes cenários de teste:

Cenário em rede DOCSIS: Para este cenário, temos como Gateways M2M três telefones inteligentes Android: 2 Asus Zenfone 5 com 2GB de memória RAM e processador Quad Core Snapdragon a 1.2GHz e outro LG g2 com processador Octa Core a 1.4ghz ARM e 2GB de memória RAM. O acesso à Internet foi através de conexão banda larga DOCSIS de 30 Mbps e um ponto de acesso Wifi.

Cenário em rede 3G/4G estático: Os mesmos aparelhos foram empregados para o teste, mas com acesso via rede de dados móveis de uma operadora celular e redes de acesso 3G e 4G. Neste cenário, sempre a coleta de métricas se fez em posição de repouso.

6.1 - DESCRIÇÃO DOS TESTES

Considera-se que, a partir da coleta de um ou mais parâmetros fisiológicos (feita por meio de medições por sensores junto a um paciente), os dados experimentais relativos a tais parâmetros serão transmitidos em horários pré-definidos, tendo como origem o dispositivo que atua como Gateway M2M (onde são armazenados temporariamente) e como destino o servidor NSCL, para posterior apresentação a um ou mais usuários da plataforma.

O armazenamento temporário ocorre apenas para fins experimentais, para permitir o envio simultâneo posterior, em paralelo, por 3 (três) terminais e sob mesmas condições de rede, de dados coletados a partir de sensores individuais, com base em diferentes técnicas de "publish-subscribe".

Para o processo de transmissão entre Gateway M2M e aplicativo M2M (passando pelo servidor NSCL), foram consideradas somente as mensagens ORU relativas às observações dos parâmetros fisiológicos (ECG, Fluxo de Ar e SpO2), os quais serão tratados pelas aplicações que interagem com os usuários da plataforma.

Assim, considerou-se uma amostra ao conjunto de tempos de envio de dados referentes a transmissões realizadas entre Gateway M2M e o aplicativo M2M em horários

predeterminados.

Os terminais móveis celulares foram preparados para teste e comparação das técnicas de "publish-subscribe, conforme Tabela 6-1:

Tabela 6-1: Configuração dos experimentos.

Gateway M2M	Técnica	Tipo de servidor
Asus Zenfone 5	HTTP <i>Polling</i>	Servidor REST HTTTPoling
LG g2	HTTP Long <i>Polling</i>	Servidor REST <i>Long</i> <i>Polling</i>
Asus Zenfone 5	<i>Websocket</i>	Servidor habilitado para <i>Websocket</i>

Cada amostra é composta de 48.000 observações obtidas em 20 min de monitoramento a uma taxa de 40 amostras/seg, ($40 \times 20 \times 60 = 48.000$). As mensagens foram disparadas a uma taxa de 24 mensagens/seg. tentando cumprir com uma taxa de atualização estável para o sistema perceptivo humano.

Para fins de caracterização de métricas, consideremos T_m como o atraso de cada mensagem referente a uma observação entregue ao *Gateway M2M* para posterior envio. T_m refere-se ao tempo que demora uma mensagem em chegar ao aplicativo M2M, após haver sido enviada pelo *Gateway M2M*. Pode ser calculado como:

$$T_m = T_a - T_g \quad (6.1)$$

onde T_a é o instante no qual a mensagem chegou ao aplicativo M2M e T_g corresponde ao momento no qual a mensagem saiu do *Gateway M2M*.

As métricas tidas em conta foram:

- 1- Atraso médio fim a fim (Média amostral), expresso por:

$$\bar{\mu} = \frac{\sum_{i=1}^{24} \bar{t}_i}{24} \quad (6.2)$$

onde:

$$\bar{t}_i = \frac{\sum_{m=1}^{\text{numero de observações}} T_m}{\text{numero de observações}} = \frac{\sum_{m=1}^{48000} T_m}{48000} \quad (6.3)$$

refere-se à média do atraso para cada amostra i

2- Atraso médio mínimo, expresso por:

$$\overline{T_{min}} = \frac{\sum_{i=1}^{24} \min\{T_i\}}{24} \quad (6.4)$$

3- Atraso médio máximo, expresso por:

$$\overline{T_{max}} = \frac{\sum_{i=1}^{24} \max\{T_i\}}{24} \quad (6.5)$$

4- Desvio padrão de uma dada amostra m , que pode ser obtido como:

$$S_m = \sqrt{\frac{\sum_{i=0}^{\text{numero de observações}} (T_i - \bar{t}_i)^2}{\text{numero de observações} - 1}} \quad (6.6)$$

5- Perda de mensagens média:

$$\overline{P_m} = \frac{\sum_{i=1}^{24} P_m}{24} \quad (6.7)$$

onde P_m se refere a um contador que reflete o número de mensagens que tiveram que ser retransmitidas produto de falhas na comunicação entre o NSCL e o *Gateway* M2M ou o Aplicativo M2M.

Para cada cenário foram feitas 24 transmissões de 20 minutos em horários diferentes do dia, permitindo estimar a média das métricas tidas em conta. Dessas 24 transmissões, 12 foram feitas ao longo do dia e 12 durante a noite. Cada uma dessas transmissões foi considerada como uma amostra, permitindo obter a média amostral.

Além das métricas descritas anteriormente, foi feita uma estimativa dos intervalos de confiança da média populacional do atraso, μ , e do erro aleatório em cada um dos testes seguindo a abordagem proposta em [63].

A partir das transmissões realizadas em diferentes horários (horas cheias) tomadas como amostras, define-se o valor do parâmetro *graus de liberdade* como:

$$\gamma = (24 - 1) = 23 \quad (6.8)$$

O fato de ter-se um número de amostras menor que 30 leva ao emprego da técnica demonstrada pelo estatístico inglês W. S. Gosset conhecido como Teste “*t-Student*”. Como o desvio padrão da população S também é desconhecido, foi calculado o desvio amostral para cada transmissão S_m (Equação 6.6), permitindo estimar o desvio padrão médio como se apresenta a continuação:

$$S = \frac{\sum_{a=0}^{24} S_m}{24} \quad (6.9)$$

Foi considerado um intervalo de confiança de 95% e probabilidade de erro (nível de significância) $\alpha = 0.05$.

Finalmente na estimação do intervalo de confiança para a média se empregou a tabela t de “Student” considerando o valor:

$$n - 1 = 23 \quad (6.10)$$

tendo como resultado

$$t\left(23, \frac{\alpha}{2}\right) = 2,069 \quad (6.11)$$

O intervalo da média do atraso pode então ser calculado como:

$$\bar{\mu} - e \leq \mu \leq \bar{\mu} + e \quad (6.12)$$

sendo,

$$e = t\left(23, \frac{\alpha}{2}\right) * S/\sqrt{24} \quad (6.13)$$

para uma probabilidade de 0.95 do valor real da média pertencer a dito intervalo.

Os resultados obtidos são mostrados e discutidos nas seções a seguir.

6.2 - TESTETESTES COM UM SÓ FLUXO.

No primeiro teste foi considerado somente o fluxo de ECG encaminhado desde o gateway M2M para o servidor NSCL. Tanto o servidor como o gateway foram sincronizados temporalmente.

6.2.1 - Resultados para cenário rede DOCSIS

Na Figura 6-1, pode-se apreciar que a variante baseada em *Websocket* é a melhor, mas mesmo assim a proposta baseada em *Long Polling* fica bem perto desta solução. Isto permite determinar que em ambientes onde não se tem suporte para *Websocket*, o sistema ainda continua estável seguindo a variante de *Long Polling*. Mesmo assim, em ambientes na nuvem se prefere o sistema que poupa em termos de largura de banda empregada.

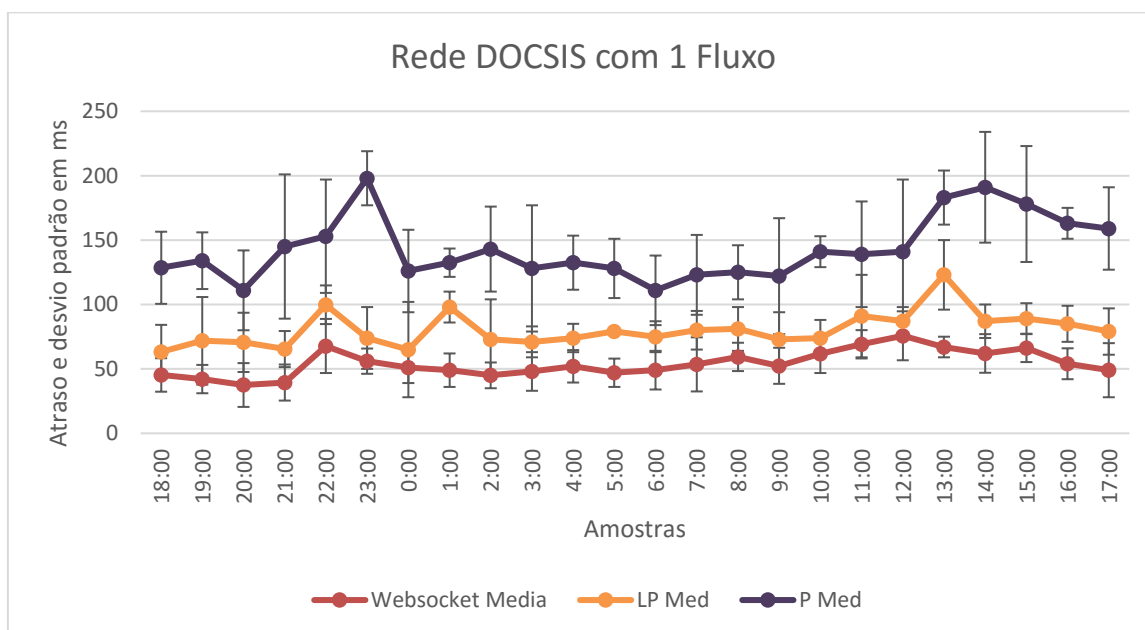


Figura 6-1: Atraso no cenário de rede DOCSIS, 1 fluxo.

Quanto ao desempenho da técnica de *Polling*, foi verificado que o pior desempenho esteve na transmissão feita às 23:00 horas. Tentando obter uma explicação para este acontecimento, procuramos os dados referentes a consumo de recursos na nuvem onde, como se pode observar na Figura 6-2, o consumo de CPU da infraestrutura na nuvem estava bastante elevado. Isto provavelmente acontece devido a que o plano de serviços contratado é de tipo “básico” onde parte dos recursos de infraestrutura são compartilhados com outros usuários dos serviços em nuvem.

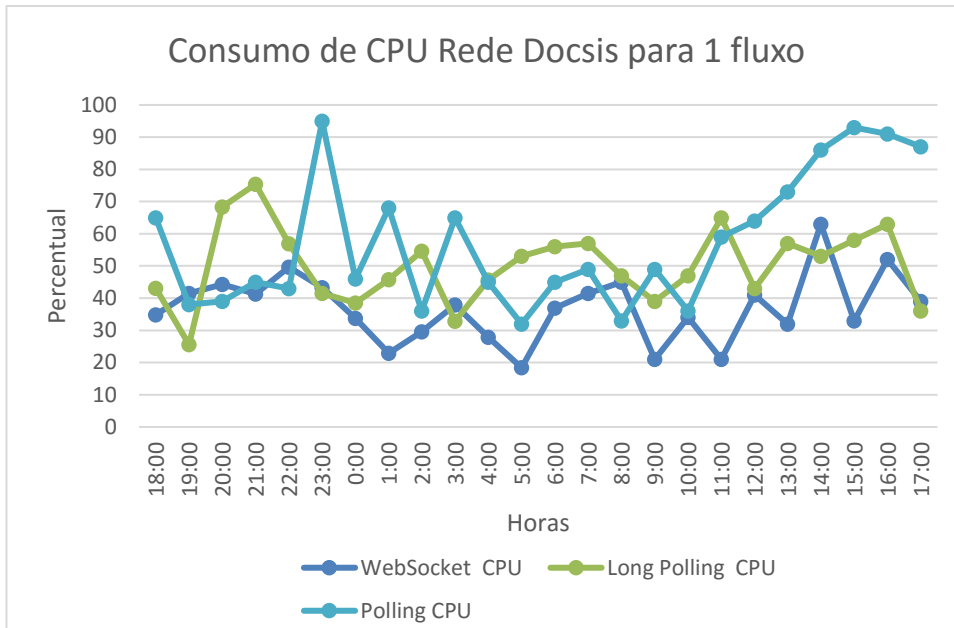


Figura 6-2: Uso de CPU na nuvem.

Quanto ao consumo de memória RAM (Figura 6-3) não foi possível estabelecer uma relação direta com o desempenho de cada proposta, dado que não foi encontrada uma situação em que tenha havido piora do desempenho por falta de recursos deste tipo.

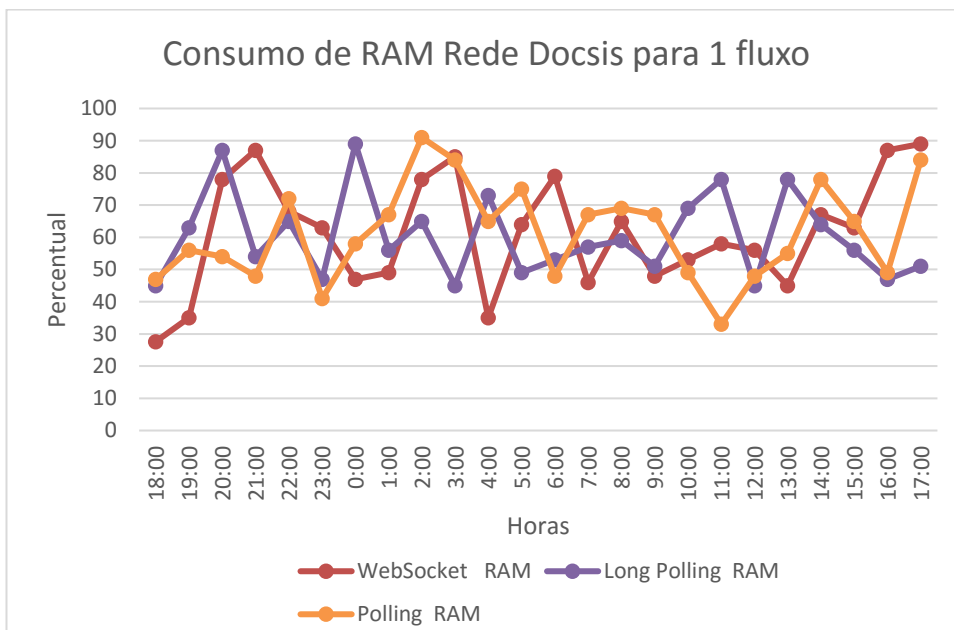


Figura 6-3: Uso de memória RAM na nuvem.

Tabela 6-2: Resultados das Métricas na Rede DOCSIS com 1 Fluxo

Técnica	$\overline{T}_{m\acute{a}x}$	$\overline{T}_{m\acute{i}n}$	\bar{t}_i	S	Intervalo de confiança de 95%	\overline{P}_m
<i>Polling</i>	278.16 ms	91.48 ms	143,14 ms	31.54 ms	[129.81,156.46]	1.67%
<i>Long Polling</i>	153.48 ms	49.18 ms	80,85 ms	18,77 ms	[72.44,88.27]	0.82%
<i>Websocket</i>	101.36 ms	34.17 ms	49.61 ms	12,75 ms	[42.49,54.19]	0.31%

Na Tabela 6.2 são apresentados os resultados das métricas consideradas no cenário utilizando a rede DOCSIS na transmissão de um fluxo ECG. Pode-se apreciar que métricas como a perda média de mensagens \overline{P}_m ou a média do atraso mínimo, alcançam valores menores quando se trata da proposta baseada em *Websockets*. Comparando as técnicas HTTP *Polling* e *Long Polling* é notável que a última apresenta melhor desempenho. Assim é possível estabelecer uma hierarquia no que respeita as técnicas, sendo *Websockets* a de melhor desempenho e HTTP *Polling* a de pior.

Outros resultados importantes são os referentes a intervalos de confiança para a média do atraso. A Figura 6-4 mostra os intervalos para os quais a média do atraso deve pertencer para cada uma das técnicas no cenário em análises. Lembrando: esses intervalos foram criados considerando a probabilidade de cada uma dessas medias estimadas pertencer ao intervalo é de 0.95.

Pode-se apreciar na Figura 6-4 que os intervalos não apresentam sobreposição, pelo que se pode afirmar com um 95% de certeza que a técnica baseada em *Websockets* teve melhor desempenho médio e que se mantém a hierarquia definida anteriormente.

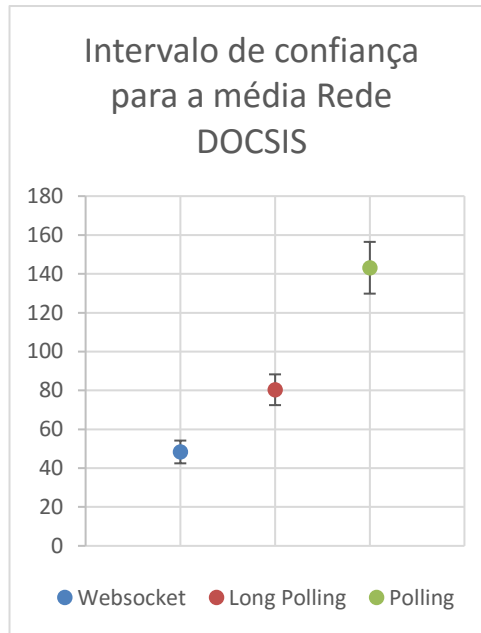


Figura 6-4: Intervalos de confiança da média dos atrasos na transmissão de 1 fluxo na Rede DOCSIS.

6.2.2 - Resultados para cenário rede 3G/4G

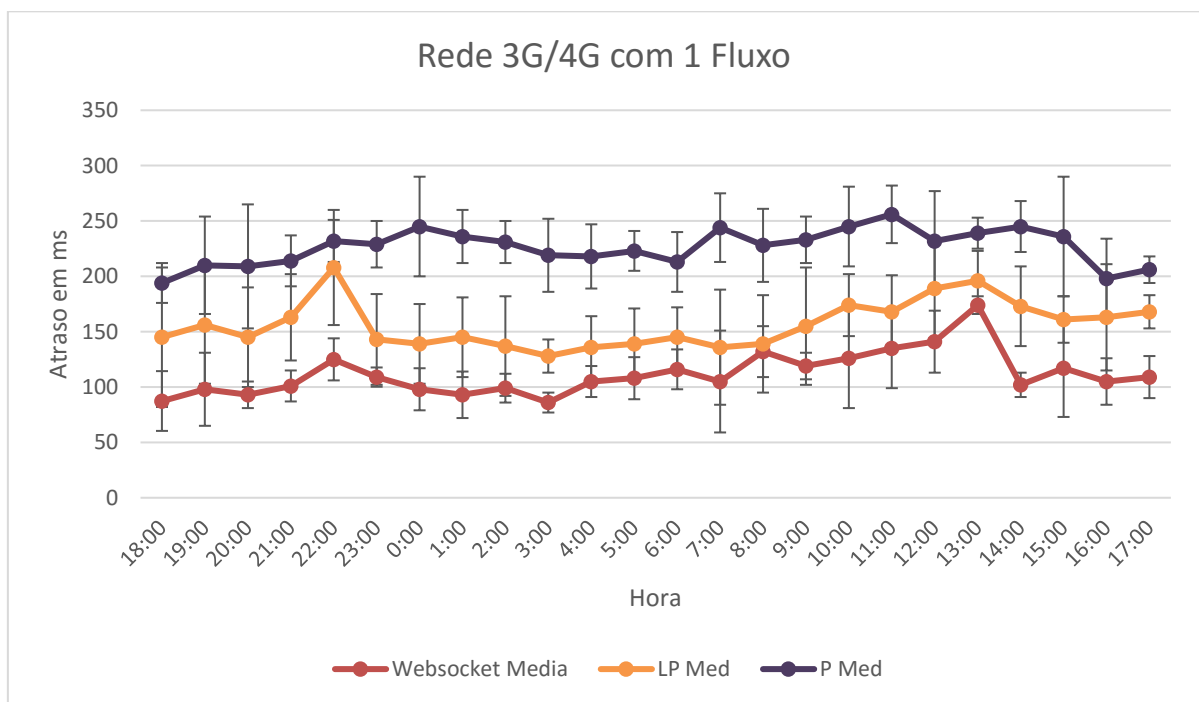


Figura 6-5: Atraso no cenário operadora celular, 1 fluxo.

Já neste cenário, se pode apreciar como as técnicas *Websocket* e *Long Polling* se aproximam em termos de atrasos, isto se deve a que a taxa de erro na rede é mais alta e

no caso do *Websocket*, mais conexões se tem que abrir produto de quedas. Mesmo assim, ainda se comporta como a melhor solução. Não obstante, é apreciável que em ambientes com elevado número de falhas pode ocorrer que seja mais conveniente empregar uma técnica de *Polling* em lugar de *Websocket*.

Semelhante ao cenário anterior, nas transmissões feitas nas 13:00 horas e nas 22:00 pode-se observar um desempenho irregular nas propostas baseadas em *Websockets* e em *Long Polling*. A Figura 6-6 mostra que nessas transmissões o consumo de recursos, na infraestrutura na nuvem foi alto. No caso do consumo de memória RAM, o desempenho do sistema se mostrou estável sem impacto negativo no desempenho da plataforma.

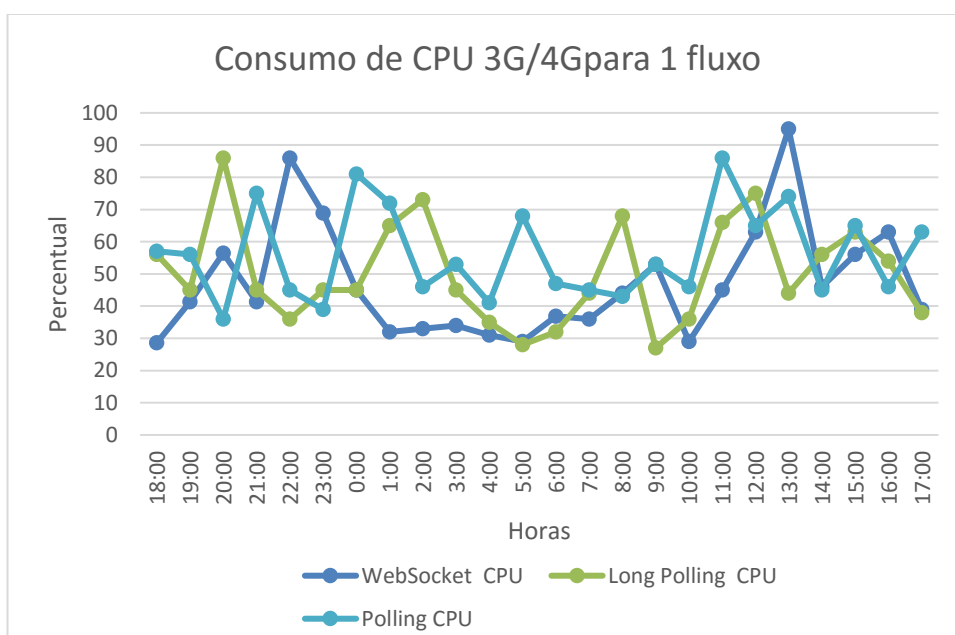


Figura 6-6: Uso de CPU na nuvem.

Considerando a transmissão do fluxo ECG através da rede 3G/4G, a Tabela 6-3 e a Figura 6-7 apresentam os resultados das métricas e os intervalos de confiança da média dos atrasos respectivamente. Os resultados corroboram a aproximação dos atrasos nas técnicas, onde o valor máximo no *Websocket* intersecta a média de desempenho do ambiente baseado em *Long Polling*. Mesmo sendo pequena, existe sobreposição nos intervalos referentes as técnicas *Websocket* e *Long Polling* e não se pode aceitar a hipóteses de que se cumpre a hierarquia. Não obstante, os centros dos intervalos brindam uma ideia de escolha a qual finalmente foi seguida neste trabalho, decidindo que a plataforma se comportara respeitando a hierarquia *Websocket*, *Long Polling* e *Polling*, tentando que a técnica de maior eficácia seja a empregada a cada momento.

Tabela 6-3: Resultado das Métricas na Rede 3G/4G com 1 Fluxo

Técnica	$\overline{T}_{máx}$	$\overline{T}_{mín}$	\bar{t}_i	S	Intervalo de confiança de 95%	\overline{P}_m
Polling	328.14 ms	124.41 ms	226.45 ms	48.91 ms	[213.43,238.25]	3.52%
Long Polling	312.73 ms	98.03 ms	156.43 ms	24.41 ms	[139.95,172.04]	3.03%
Websocket	228.4 ms	60.3 ms	111.8 ms	21.65 ms	[102.65, 120.94]	0.94%

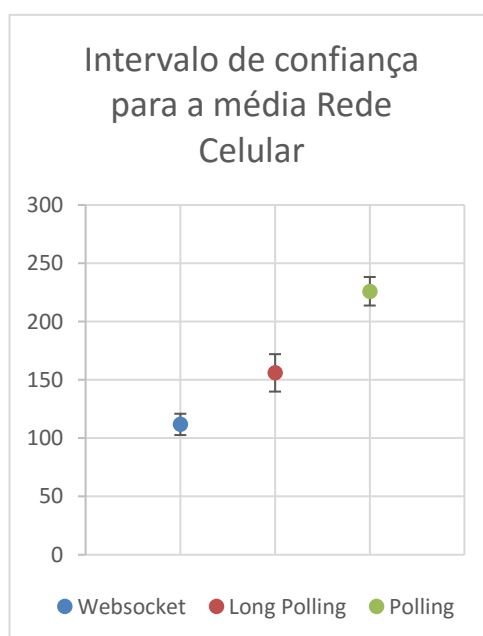


Figura 6-7: Intervalos de confiança da média dos atrasos na transmissão de 1 fluxo na Rede 3G/4G.

6.3 - TESTE COM OS TRÊS FLUXOS.

Cenário semelhante ao anterior, só que atendendo mais dois fluxos diferentes (Fluxo de Ar e Oxigênio no sangue). Neste teste, cada *request* é feito três vezes uma para cada fluxo; os *request* no caso de *Polling* são invocados assíncronos enquanto as mensagens enviadas pelo socket são tratadas sequencialmente devido ao fato de que só uma conexão TCP está sendo aberta na sessão de comunicação. Este é o pior caso para a técnica *WebSocket* e dado que se trata de uma prova comparativa, decidimos empregar o cenário

de pior condição de cada técnica.

6.3.1 - Resultados para cenário de rede DOCSIS

No cenário de rede DOCSIS, ainda se mantém que a proposta de *Websocket* é a de melhor desempenho. Mesmo assim um \bar{t}_i com valores da ordem de 520ms para a técnica de *Polling* ainda pode ser considerado como um resultado bom pois impacta pouco a qualidade de experiência por parte de quem recebe a informação como cliente. Já no caso do *Websocket*, pode ser considerado quase instantâneo (o atraso fica perto dos 172 ms) destacando a não sobreposição com a proposta baseada em *Long Polling*.

No cenário em análise, seguindo a mesma metodologia de avaliação das seções anteriores, são apresentados: (i) na Figura 6-8 o atraso médio e o desvio padrão, (ii) na Tabela 6-4 os valores referentes as métricas e (iii) na Figura 6-9 os intervalos de confiança.

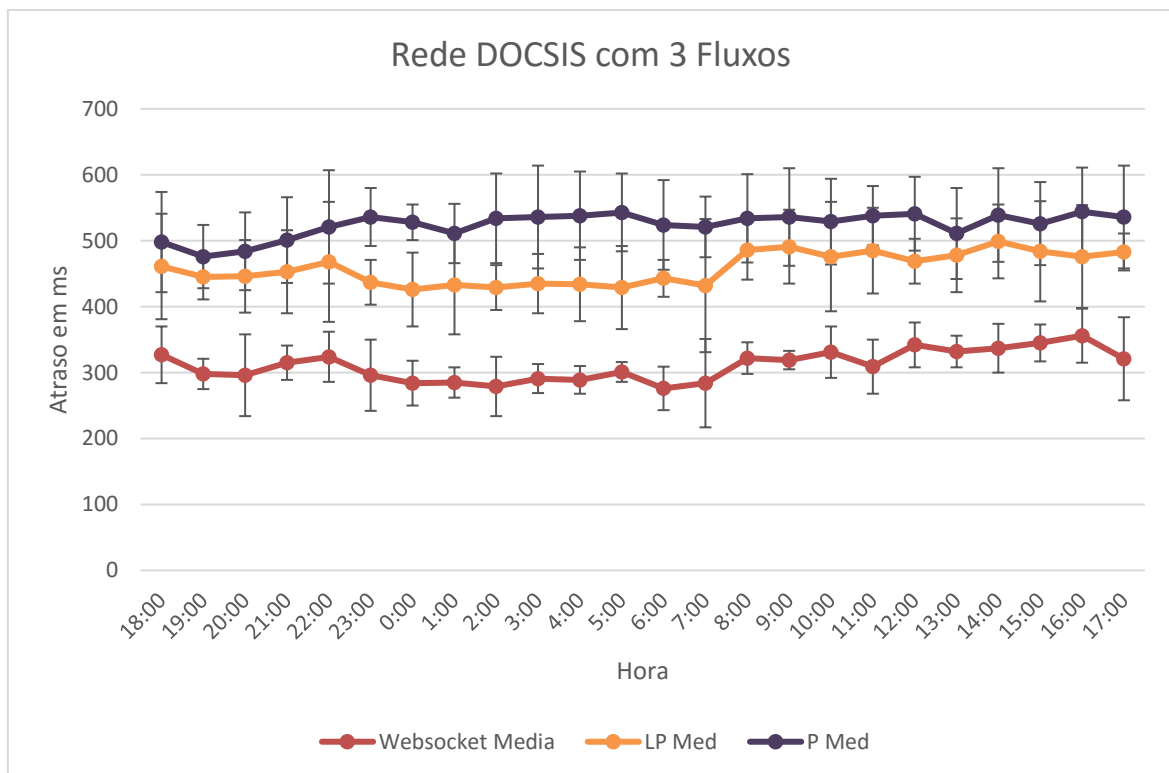


Figura 6-8: Atraso no cenário de rede DOCSIS, 3 fluxos.

Tabela 6-4: Resultados das Métricas na Rede DOCSIS com 3 Fluxos

Técnica	$\overline{T}_{máx}$	$\overline{T}_{mín}$	\bar{t}_i	S	Intervalo de confiança de 95%	\overline{P}_m
Polling	647.93 ms	427.31 ms	524.37 ms	62.115	[493.44,554.76]	4.68%
Long Polling	631.37 ms	345.05 ms	458,25 ms	58.00	[433.72,482.49]	3.26%
Websocket	421.45 ms	198.37 ms	310.79 ms	35.45	[295.43,324.67]	2.21%

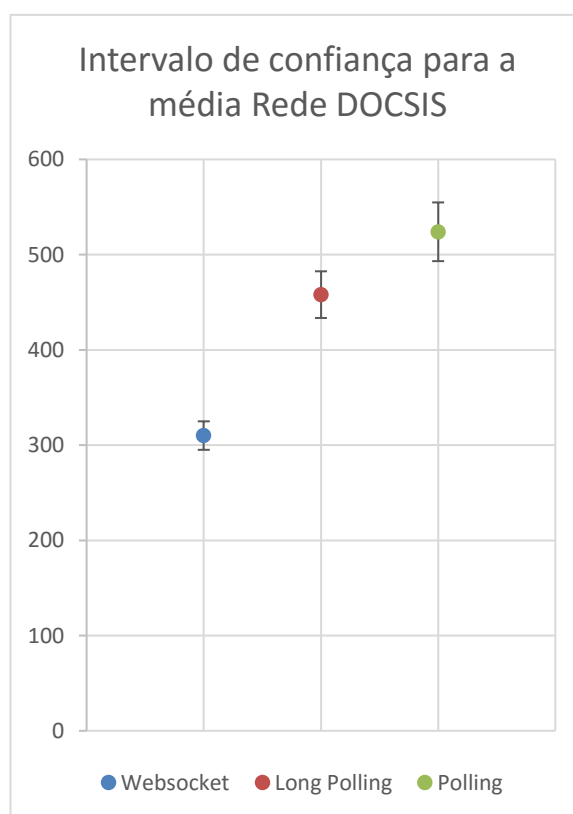


Figura 6-9: Intervalos de confiança da média dos atrasos na transmissão de 3 fluxos na Rede DOCSIS.

A proposta baseada em *Websockets* manteve o melhor desempenho onde o intervalo de confiança da média dos atrasos ficou bem distante das outras duas alternativas. No caso das técnicas baseadas em *Polling*, não se pode afirmar que uma seja sempre superior à outra, dada a sobreposição dos intervalos de confiança. Dita sobreposição ocorre

produto de que para uma maior quantidade de fluxos, tem-se uma maior quantidade de mensagens, provocando que o tempo em que o servidor *Long Polling* aguarda para dar resposta seja menor. É dizer: o servidor *Long Polling* sempre terá uma mensagem para enviar levando a que ambas propostas sejam similares com o aumento da quantidade de fluxos.

O consumo de recursos na nuvem se manteve aceitável para as três técnicas. Não foi detectada deterioração do desempenho com o aumento do número de fluxos. As Figuras 6-10 e 6-11 confirmam as afirmações anteriores.

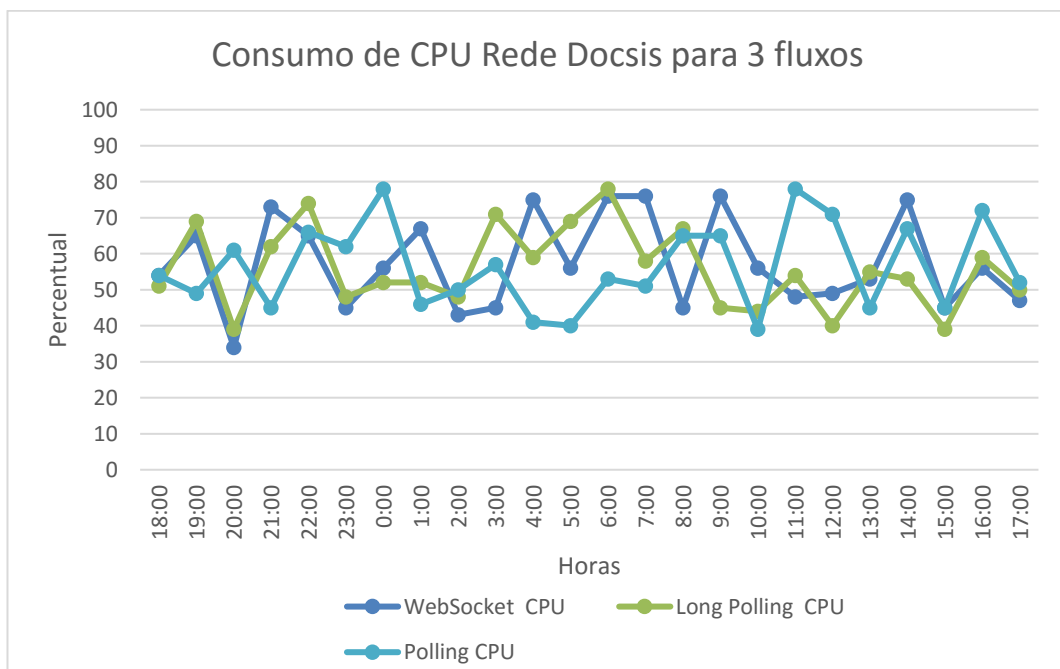


Figura 6-10: Uso de CPU na nuvem.

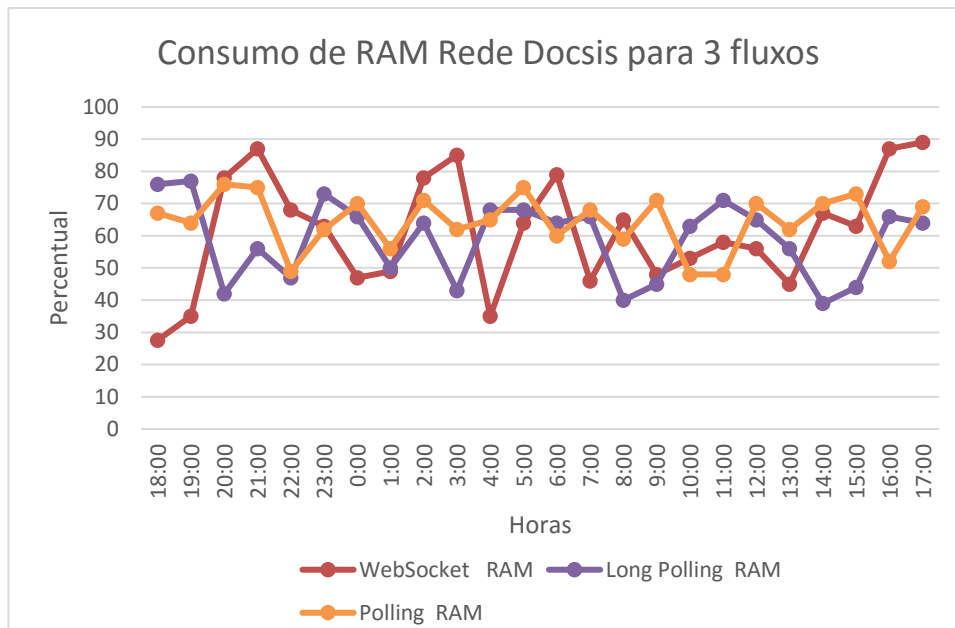


Figura 6-11: Uso de memória RAM na nuvem.

6.3.2 - Resultados para cenário rede 3G/4G

Já nesta versão se precisa de um *buffer* maior no lado do cliente para permitir maior tempo de *buffering*. O fato de que, em uma rede celular, a largura de banda é menor e a instabilidade do meio sem fio provocam que a taxa de perda aumente e que o atraso seja maior, comparado com redes locais. Este resultado era esperado e porem a plataforma se vê afetada em termos de desempenho nesta categoria de cenários.

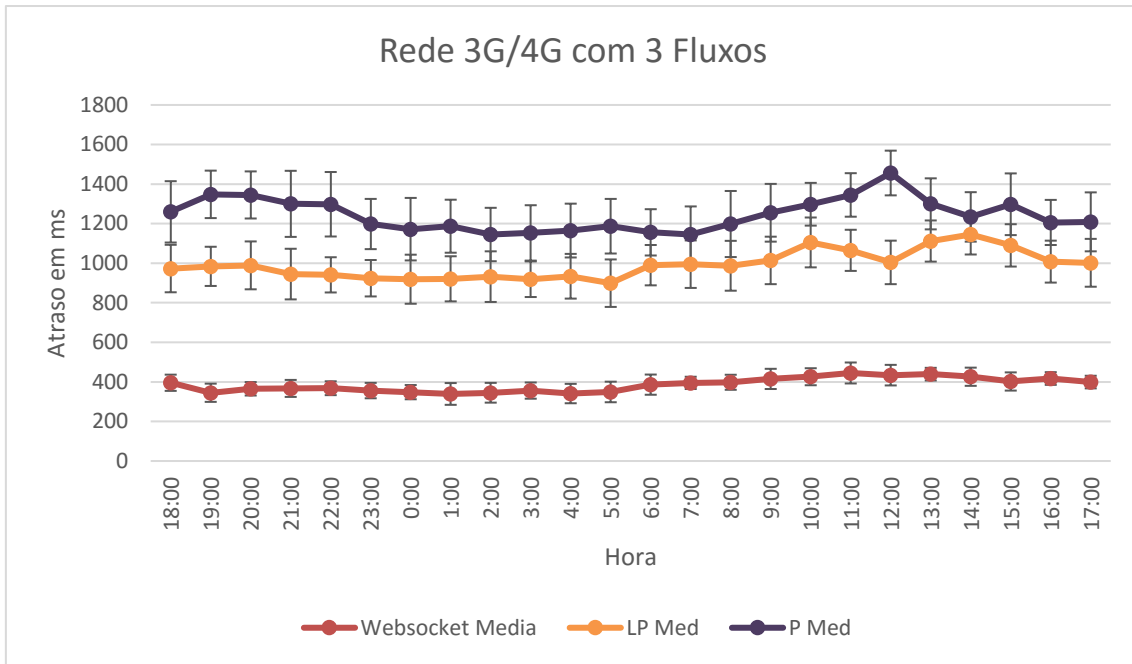


Figura 6-12: Atraso no cenário operadora celular, 3 fluxos.

Tabela 6-5: Resultado das Métricas na Rede Celular com 3 Fluxos

Técnica	$\overline{T}_{m\acute{a}x}$	$\overline{T}_{m\grave{i}n}$	\bar{t}_i	S	Intervalo de confiança de 95%	\overline{P}_m
Polling	1928.43 ms	677.47 ms	1244.06 ms	1142.66	[1181.91,1306.08]	12.54%
Long Polling	1201.39 ms	596.07 ms	991.54 ms	104.07	[944.54,1037.45]	6.88%
Websocket	578.14 ms	314.34 ms	385.68 ms	43.66	[366.41,403.58]	4.82%

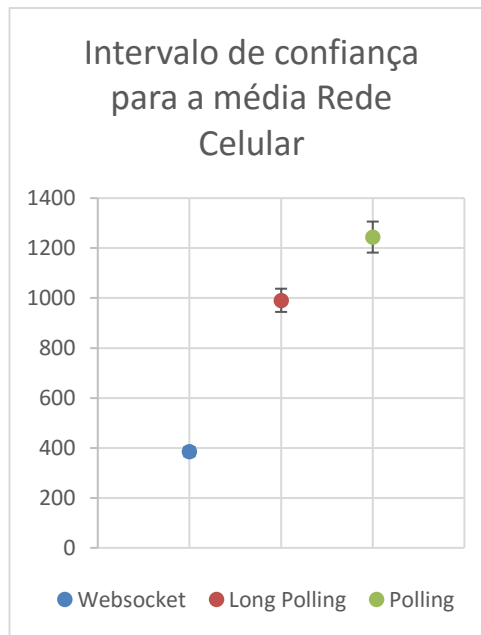


Figura 6-13: Intervalos de confiança da média dos atrasos na transmissão de 3 fluxos na Rede 3G/4G.

Os resultados apresentados na Tabela 6-5, na Figura 6-12 e Figura 6-13 corroboram a afirmação anterior. Pode-se observar que o aumento do atraso nas propostas baseadas em *Polling* é significativo comparado com a proposta com base em *Websocket*. Isto é influenciado pelo aumento de tráfego de “controle” introduzido pelo protocolo HTTP. Porém, ainda é possível definir a mesma hierarquia seguida até o momento, considerando os intervalos de confiança da média do atraso.

6.4 - ANÁLISE CONJUNTA DOS ATRASOS

Finalmente, com fim de extrair relações quantitativas a partir dos resultados obtidos, foi decidido fazer uma análise estatística acima dos dados capturados.

Primeiramente foi definida uma métrica para quantificar o volume de mensagens para as quais um sistema foi superior ao outro em termos de atrasos.

Para isto as seguintes definições vão servir como base.

- (i) Seja N a quantidade de mensagens transmitidas nos testes, se define como Percentil de superioridade $Ps(x, y)$ como o percentil de N de mensagens da técnica x que teve um atraso menor que na técnica y .

- (ii) Seja x, y duas técnicas diferentes, se define como razão de atraso médio à divisão $r(x, y) = \bar{\mu}_x / \bar{\mu}_y$. Com está grandeza é possível fazer a análise de convergência ao valor 1 brindando uma similaridade de desempenho.

A Tabela 6-6 mostra esses valores para cada um dos experimentos realizados.

Tabela 6-6: Percentil de superioridade e razão de atraso médio

Experimento	$Ps(ws, lp)$	$Ps(lp, p)$	$r(ws, lp)$	$r(lp, p)$
Teste com um fluxo, cenário rede DOCSIS	95%	98%	0.61	0,59
Teste com um fluxo, cenário operadora celular	90%	96%	0.79	0.7
Teste com três fluxos, cenário de rede DOCSIS	100%	86%	0,63	0.84
Teste com três fluxos, cenário operadora celular	100%	90%	0,45	0,79

Analisando esses valores se observa o melhor desempenho da proposta baseada em *Websockets* e a superioridade da técnica de *Long Polling* sobre *Polling*. Não obstante precisamos uma estimativa da média obtida a partir de estudos estatísticos que avaliem a superioridade do desempenho em cada proposta.

6.5 - CONSIDERAÇÕES FINAIS

A avaliação de desempenho da plataforma permitiu estabelecer uma hierarquia no comportamento de cada uma das propostas em dois cenários diferentes e com fluxos concorrentes diferentes. Esta hierarquia é útil considerando que o protocolo *Websockets* não é aplicável a todos os ambientes. O desempenho empregando *Long Polling* e até *HTTP Polling* ainda pode ser considerado aceitável, mesmo sendo com maiores atrasos e perdas.

Uma vantagem a destacar e que não foi tratada previamente, é a habilidade dos serviços em nuvem de escalar rapidamente quanto ao número de requisições recebidas

no servidor. No caso do Azure, ele permite uma alta elasticidade pois podem ser configuradas regras para a escala do número de instâncias para o aplicativo Web, ou serviço móvel. Estas regras podem ser baseadas no consumo de CPU, de memória RAM e no número de requisições na fila HTTP para os casos de Polling. O tempo que demora em criar uma nova instância está perto de 30 segundos (para a primeira instancia escalada) e de 10 minutos para as outras novas em sequência.

7 - CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma proposta de plataforma para o desenvolvimento de aplicativos voltados para telemonitoramento em *m-Health*, integrando o padrão OpenEHR e tecnologias como Comunicação Máquina a Máquina e Computação em nuvem. No processo foram explorados os conceitos de TICs aplicados ao contexto do cuidado da saúde empregando meios eletrônicos. Também foram estudadas diferentes plataformas de *e-Health* disponíveis na literatura.

Foi proposto um método para expandir as semânticas do padrão OpenEHR ao contexto do dispositivo de captura. Com essa extensão, foi demonstrado que é possível uma maior compatibilidade e, por enquanto, uma maior escalabilidade do sistema em termos de sensores empregados.

O emprego de recursos em nuvem e a implementação do NSCL da arquitetura M2M nos mesmos permitiram uma facilidade em termos de desenvolvimento e de escalabilidade do sistema. A característica do gateway M2M desenvolvido em um smartphone permitiu garantir mobilidade por parte dos usuários finais, além de ser um fator importante em termos de simplicidade da plataforma e de custos monetários.

O *kit* de desenvolvimento da *Cooking Hacks (e-Health Sensor Shield)* permitiu a realização de provas de conceitos da plataforma desenvolvida, além de simplificar a definição do grupo de sensores empregados para teste de integração.

Além da plataforma, foi desenvolvido um protótipo para o monitoramento de parâmetros relativos à apnéia do sono. Com este protótipo, foi possível obter uma relação quantitativa do desempenho de técnicas de comunicação como: HTTP *Polling*, HTTP *Long Polling* e *Websocket*, onde os dois primeiros foram desenvolvidos integralmente ao padrão REST. O estudo dos atrasos, perdas e a análise de desempenho, permitiram definir uma hierarquia entre os diferentes mecanismos de comunicação, onde *Websocket* apresentou um desempenho superior em termos de latência e perdas no que respeita às outras duas propostas.

Concluimos que foi possível a criação de uma plataforma para acesso de equipes médicas e profissionais da saúde em geral a dados capturados em tempo real, capaz de atender aos requisitos anteriormente estabelecidos.

Não obstante, outros aspectos devem ser estudados para caracterizar a plataforma desenvolvida em maior profundidade; neste sentido, propõem-se os seguintes trabalhos futuros:

- Avaliar a plataforma em termos de consumo energético;
- Expandir as funcionalidades para diferentes plataformas (p. ex. iOS, Windows Phone) e linguagens de desenvolvimento (p. ex. Python, C#);
- Propor mecanismos de automação para diferentes tipos de provedores de serviços em nuvem;
- Analisar a viabilidade de expandir conceitos de parâmetros do ambiente ou contexto do paciente (p.ex. umidade relativa do ar, pressão atmosférica, nível de poluição, temperatura, entre outros) no OpenEHR;
- Estudar e incluir na plataforma conceitos relativos ao contexto do paciente;
- Tratar de forma mais ampla os aspectos de gerenciamento de contexto não somente com relação ao paciente, mas também com relação a profissionais de saúde e à infraestrutura de comunicação.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] G. Eysenbach, "What is e-health?," *Journal of Medical Internet Research*, vol. 3, p. 20, 2001.
- [2] H. Oh, C. Rizo, M. Enkin, and A. Jadad, "What is eHealth (3): a systematic review of published definitions," *Journal of Medical Internet Research*, vol. 7, 2005.
- [3] S. Sood, V. Mbarika, S. Jugoo, R. Dookhy, C. R. Doarn, *et al.*, "What is telemedicine? A collection of 104 peer-reviewed perspectives and theoretical underpinnings," *Telemedicine and e-Health*, vol. 13, pp. 573-590, 2007.
- [4] C. C. Poon, Y.-T. Zhang, and S.-D. Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health," *Communications Magazine, IEEE*, vol. 44, pp. 73-81, 2006.
- [5] C. Free, G. Phillips, L. Watson, L. Galli, L. Felix, *et al.*, "The effectiveness of mobile-health technologies to improve health care service delivery processes: a systematic review and meta-analysis," *PLoS Med*, vol. 10, p. e1001363, 2013.
- [6] D. Lucic, A. Caric, and I. Lovrek, "Standardisation and regulatory context of machine-to-machine communication," in *2015 13th International Conference on Telecommunications (ConTEL)*, pp. 1-7, 2015.
- [7] "Machine to Machine Communications," 2014, <http://www.etsi.org/technologies-clusters/technologies/m2m>, Acessado em: 27/6/2015.
- [8] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," *National Institute of Standards and Technology*, vol. 53, p. 50, 2012.
- [9] N. Aissaoui, M. Aissaoui, and Y. Jabri, "For a Cloud Computing based Open Source E-Health Solution for Emerging Countries," *International Journal of Computer Applications*, vol. 84, pp. 1-6, 2013.
- [10] E. Newcomer and G. Lomow, *Understanding SOA with Web services* Addison-Wesley, 2005.
- [11] K. M. Dhara, M. Dharmala, and C. K. Sharma, "A Survey Paper on Service Oriented Architecture Approach and Modern Web Services," All Capstone Projects, Paper 157, <http://opus.govst.edu/capstones/157>, 2015.
- [12] H. S. Fugita, "MAPOS: método de análise e projeto orientado a serviços," Dissertação de Mestrado em Engenharia de Computação, Universidade de São Paulo, 2009.
- [13] P. M. Nadkarni and R. A. Miller, "Service-oriented architecture in medical software: promises and perils," *Journal of the American Medical Informatics Association*, vol. 14, pp. 244-246, 2007.
- [14] L. Richardson and S. Ruby, "REST full Web Services. Web services for the real world.", O'Reilly Media, Inc., 2008.
- [15] M. Jakl, "Rest representational state transfer," University of Technology of Vienna, 2008.

- [16] R. Lomotey, R. Kazi, and R. Deters, "Near real-time medical data dissemination in m-Health," in *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, pp. 67-74, 2012.
- [17] I. Fette, Alexey Melnikov, "RFC 6455: The websocket protocol," December, 2011.
- [18] The OpenEHR Foundation, "OpenEHR architecture overview," 2014, <http://www.openehr.org/>, Acessado em: 12/12/2015.
- [19] OpenEHR, "OpenEHR Clinical Knowledge Manager," 2015, <http://www.openehr.org/ckm/>, Acessado em: 12/2015.
- [20] HL7, "Standard: Refinement, Constraint and Localization, Release 2," 2015, <http://www.hl7.org/implement/standards/ansiapproved.cfm>, Acessado em: 7/8/2015.
- [21] Y. E. Rivera, "Desarrollo de un prototipo ARDUINO-MOVIL en el área de la telemedicina para el monitoreo remoto de personas diabéticas," Engineering Systems, Corporación Universitaria Americana, 2014.
- [22] Cooking Hacks, "e-Health Sensor Platform Complete Kit V2.0 for Arduino, Raspberry Pi and Intel Galileo [Biometric / Medical Applications]," 2015, <http://www.cooking-hacks.com/ehealth-sensors-complete-kit-biometric-medical-arduino-raspberry-pi>, Acessado em: 18/03/2015.
- [23] W. Chen, S. T. Nguyen, R. Coops, S. B. Oetomo, and L. Feijs, "Wireless transmission design for health monitoring at neonatal intensive care units," in *Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009. 2nd International Symposium on*, pp. 1-6, 2009.
- [24] E. Smithayer, "Sensor-based system for verifying blood-pressure measurement position," *Dartmouth College, Computer Science, Hanover, NH*, Tech. Rep. TR2012-720, 2012.
- [25] O. Ogunduyile, K. Zuva, O. Randle, and T. Zuva, "Ubiquitous healthcare monitoring system using integrated triaxial accelerometer, spo2 and location sensors," *arXiv preprint arXiv:1309.1542*, 2013.
- [26] A. M Bhatti and M. Masud, "Context Aware Intelligent Wallet for Healthcare," *International Journal of Computer Applications*, vol. 96, pp. 31-36, 2014.
- [27] S. Fuicu, A. Avramescu, D. Lascu, M. Marcu, and R. Padurariu, "Real time e-health system for continuous care," in *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, pp. 436-439, 2014.
- [28] N. A. Nayan and F. A. Azami, "Continuous Vital Signs Monitoring using Smartphones," presented at the i-IDeA, International Innovation, Design and Articulation, India, 2014.
- [29] P. Maia, T. Batista, E. Cavalcante, A. Baffa, F. C. Delicato, *et al.*, "A Web Platform for Interconnecting Body Sensors and Improving Health Care," *Procedia Computer Science*, vol. 40, pp. 135-142, 2014.

- [30] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, *et al.*, "Wireless sensor networks for healthcare," *Proceedings of the IEEE*, vol. 98, pp. 1947-1960, 2010.
- [31] I. Orha and S. Oniga, "Wearable sensors network for health monitoring using e-Health platform," *Carpathian Journal of Electronic & Computer Engineering*, vol. 7, 2014.
- [32] D. C. Nazario, J. L. Todesco, M. A. R. Dantas, I. Tromel, and A. Neto, "A quality of context evaluating approach in an ambient assisted living e-Health system," in *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*, pp. 158-163, 2014.
- [33] P. R. Silva-Ferreira, J. H. Patriarca-Almeida, P. M. Vieira-Marques, and R. J. Cruz-Correia, "Improving expressiveness of agents using openEHR to retrieve multi-institutional health data: Feeding local repositories through HL7 based providers," in *Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on*, pp. 1-5, 2012.
- [34] P. Vieira-Marques, J. Patriarca-Almeida, S. Frade, G. Bacelar-Silva, S. Robles, *et al.*, "OpenEHR aware multi agent system for inter-institutional health data integration," in *Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on*, pp. 1-6, 2014.
- [35] C. Martínez-Costa, M. Menárguez-Tortosa, and J. T. Fernández-Breis, "An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes," *Journal of biomedical informatics*, vol. 43, pp. 736-746, 2010.
- [36] E. Osorio, L. Ferreira, R. Abreu, and F. Sousa, "Interoperability in Ambient Assisted Living using OpenEHR," in *e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on*, pp. 394-398, 2013.
- [37] J. D. Trigo, C. D. Kohl, A. Eguzkiza, M. Martinez-Espronedada, A. Alesanco, *et al.*, "On the seamless, harmonized use of ISO/IEEE11073 and openEHR," *Biomedical and Health Informatics, IEEE Journal of*, vol. 18, pp. 872-884, 2014.
- [38] S. Frade, S. Miranda Freire, E. Sundvall, J. H. Patriarca-Almeida, and R. Cruz-Correia, "Survey of openEHR storage implementations," in *2013 IEEE 26th International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 303-307, 2013.
- [39] C. Pereira, S. Frade, P. Brandao, R. Correia, and A. Aguiar, "Integrating data and network standards into an interoperable e-Health solution," in *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*, pp. 99-104, 2014.
- [40] M. Habul, R. Hallmark, A. Kotlyanskiy, P. Parekh, and R. Kamoua, "Sleep apnea diagnostic device," in *Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island*, pp. 1-5, 2011.
- [41] R. Jane, J. A. Fiz, J. Sola-Soler, J. Mesquita, and J. Morera, "Snoring analysis for the screening of sleep apnea hypopnea syndrome with a single-channel device developed using polysomnographic and snoring databases," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pp. 8331-8333, 2011.

- [42] D. Townsend, R. Goubran, and F. Knoefel, "Amplitude-based central apnea screening," in *Medical Measurements and Applications Proceedings (MeMeA), 2011 IEEE International Workshop on*, pp. 395-398, 2011.
- [43] J. R. Williamson, D. W. Bliss, D. W. Browne, P. Indic, E. Bloch-Salisbury, *et al.*, "Using physiological signals to predict apnea in preterm infants," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pp. 1098-1102, 2011.
- [44] L. Almazaydeh, K. Elleithy, and M. Faezipour, "Detection of obstructive sleep apnea through ECG signal features," in *Electro/Information Technology (EIT), 2012 IEEE International Conference on*, pp. 1-6, 2012.
- [45] S. Alqassim, M. Ganesh, S. Khoja, M. Zaidi, F. Aloul, *et al.*, "Sleep Apnea Monitoring using mobile phones," in *e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on*, pp. 443-446, 2012.
- [46] B. Koley and D. Dey, "Automated detection of apnea and hypopnea events," in *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*, pp. 85-88, 2012.
- [47] G. Gaurav, S. Mohanasankar, and V. J. Kumar, "Apnea sensing using photoplethysmography," in *2013 Seventh International Conference on Sensing Technology (ICST)*, pp. 285-288, 2013.
- [48] R. Thanuja and R. Balakrishnan, "Real time sleep apnea monitor using ECG," in *Information & Communication Technologies (ICT), 2013 IEEE Conference on*, pp. 973-976, 2013.
- [49] P. Pierleoni, L. Pernini, A. Belli, and L. Palma, "Real-time apnea detection using pressure sensor and tri-axial accelerometer," in *Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on*, pp. 513-516, 2014.
- [50] J. Jiayi and E. Sanchez-Sinencio, "A Home Sleep Apnea Screening Device With Time-Domain Signal Processing and Autonomous Scoring Capability," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 9, pp. 96-104, 2015.
- [51] C. Lili, Z. Xi, and S. Changyue, "An Automatic Screening Approach for Obstructive Sleep Apnea Diagnosis Based on Single-Lead Electrocardiogram," *IEEE Transactions on Automation Science and Engineering*, vol. 12, pp. 106-115, 2015.
- [52] D. Falie and M. Ichim, "Sleep monitoring and sleep apnea event detection using a 3D camera," in *8th International Conference on Communications (COMM)*, pp. 177-180, 2010.
- [53] J. A. Mithun, V. V. Rajkumar, and Shanmugapriya, "Apnea Detecting Sensors," in *Bioengineering Conference, 2006. Proceedings of the IEEE 32nd Annual Northeast*, pp. 201-202, 2006.
- [54] K. M. Al-Ashmouny, A. A. Morsy, and S. F. Loza, "Sleep Apnea Detection and Classification Using Fuzzy Logic: Clinical Evaluation," in *Engineering in*

- Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pp. 6132-6135, 2005.
- [55] L. S. Correa, E. Laciari, V. Mut, A. Torres, and R. Jane, "Sleep apnea detection based on spectral analysis of three ECG - derived respiratory signals," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 4723-4726, 2009.
- [56] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, *et al.*, "Physiobank, physiotookit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. e215-e220, 2000.
- [57] G. Matsuoka, T. Sugi, F. Kawana, and M. Nakamura, "Automatic detection of apnea and EEG arousals for sleep apnea syndrome," in *ICCAS-SICE, 2009*, pp. 4651-4654, 2009.
- [58] X. Wenlong and L. XiaoFang, "Sleep Apnea Assessment by ECG Pattern," in *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, pp. 1-4, 2009.
- [59] H. Chien-Chang and S. Ping-Ta, "An intelligent sleep apnea detection system," in *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, vol. 6, pp. 3230-3233, 2010.
- [60] D. Townsend, M. Holtzman, R. Goubran, M. Frize, and F. Knoefel, "Effect of windowing on central apnea detection," in *Medical Measurements and Applications Proceedings (MeMeA), 2010 IEEE International Workshop on*, pp. 117-120, 2010.
- [61] R. Gao, L. Yang, X. Wu, T. Wang, S. Lu, *et al.*, "A phone-based e-health system for osas and its energy issue," in *Information Technology in Medicine and Education (ITME), 2012 International Symposium on*, vol. 2, pp. 682-686, 2012.
- [62] "Arduino," <http://arduino.cc/>, Acessado em: 28/02/2015.
- [63] "Estatística aplicada, Instituto de Engenharia de Produção & Gestão," https://sinop.unemat.br/site_antigo/prof/foto_p_downloads/fot_4666apostila_de_estatistica_completa_pdf.pdf, Acessado em: 10/07/2016.

APÊNDICE I - Carta de Aceitação de artigo no evento 7th International Conference on Information and Multimedia Technology (ICIMT 2015)



Acceptance Notification of ICIMT 2015

Paper ID: NB3002

Paper Title: OpenEHR semantics as a tool to achieve reusability in M2M E-Health devices

Authors: Jesus Noel Suárez Rubí, Paulo Roberto da Lira Gondim

To whom it may concern,

Congratulations! The review processes for 2015 7th International Conference on Information and Multimedia Technology (ICIMT 2015) has been completed. The conference received submissions from 34 different countries and regions, which were reviewed by international experts, and about 50 papers have been selected for presentation and publication. Based on the recommendations of the reviewers and the Technical Program Committees, we are pleased to inform you that your paper identified above has been accepted for publication and oral presentation. You are cordially invited to present the paper orally at ICIMT to be held in Barcelona, Spain, during December 21-22, 2015.

If the registration procedure (available on the next page) is completed before/on the set deadline, the paper will be published in *Journal of Advances in Information Technology* (ISSN: 1798-2340, DOI: 10.12720/jait), Abstracting/Indexing: INSPEC; EBSCO; ULRICH's Periodicals Directory; WorldCat; CrossRef; Genamics JournalSeek; Google Scholar; Ovid LinkSolver, etc.

Please strictly follow the instructions of the format specified in the conference template while preparing your final paper. If you have any problem in preparing the final paper, please feel free to contact us via icimt@iacsit.org. For the most updated information on the conference, please check the conference website at <http://www.icimt.org/>. The Conference Program will be available at the website in early December, 2015.

We are looking forward to meeting you in Barcelona, Spain.

Yours sincerely,
ICIMT 2015 Organizing Committees
icimt@iacsit.org
Barcelona, Spain



APÊNDICE II – Artigo “OpenEHR Semantics as a Tool for Reusability in M2M E-Health Devices.”, submetido no evento (ICIMT 2015)

OpenEHR Semantics as a Tool for Reusability in M2M E-Health Devices.

Jesus Noel Suárez Rubí, Paulo Roberto de Lira Gondim

Department of Electrical Engineering, Faculty of Technology – University of Brasilia

nsuarezrubi@aluno.unb.br, pgondim@unb.br

Abstract

Recent development of communication networking technologies, including Internet, sensor networks and M2M (Machine-to-Machine) communication has given rise to new architectures, applications and standards for the meeting of almost all current challenges in e-Health. Among such standards, OpenEHR has been recognized as a very important one, allowing to separate medical semantics from data representation. However, it does not meet the requirements related to the interoperability of e-Health devices in M2M networks. The present manuscript proposes an extension of the semantics described in OpenEHR for reaching sensor devices and achieving interoperability, quality of the sensing process and scalability in the M2M domain.

Keywords: e-Health, Machine to Machine, OpenEHR, HL7, Interoperability, Data Representation.

1. Introduction

e-Health [1] has been accepted in many developed and emerging countries, and many companies have developed hardware and software architectures to provide remote care at home level. These applications demand a large variety of sensors that capture data and different mechanisms that send them to the EHRs [2] (Electronic Health Records).

An important challenge in E-Health concerns the delivery of information through communication technologies. If a greater amount of significant data and information is stored and available at the right point in time for physicians, better diagnoses can be obtained. Such data can be stored and made available through the use of Electronic Health Records (EHR) in Hospitals Information Systems (HIS), currently used in many institutions worldwide. An EHR can store the clinical history of a patient through a unique identifier, e.g. the national health ID. EHRs can be implemented locally, inside a single institution, or developed in a more complex structure at country or international levels. The main advantage is physicians have clean and clear access to information and a wide range of new applications and architectures can be created.

At first, each medical institution had its own EHR implemented with different software architectures and

information representation models. However, such features have quickly become a disadvantage, regarding users' mobility among institutions, cities and countries. A demand for information exchange has been raised, so as to provide access to data on patients in a larger number of scenarios and different standards (as OpenEHR and EN ISO 13606 for data, information and medical semantics representation) have been created. Another set of standards known as HL7, for message interchange between e-Health systems, can provide interchange mechanisms between the different systems involved in the healthcare process. However, they do not focus on interoperability between sensors and electronic equipment for e-Health.

The way sensors deliver information is not standardized, therefore, each manufacturer uses its proprietary protocols and platforms are developed specifically for a particular set of sensors. When a sensor changes, the entire system must be updated, which incurs in scalability and reusability issues. For example, simple sensors, such as glucometers, accelerometers, oximeters, among others, may vary from one manufacturer to another, which leads to minor changes in the system, while other more complex monitor's systems, as electrocardiographs, electroencephalographs, blood pressure monitors, with more complex signals, lead to major changes and involve high costs and may cause a non-update of the system.

In many current scenarios, the user (a patient, in this case) has advanced technologies, such as smart watches, environment sensors, etc., which can complement the information in the medical diagnosis process. Therefore, a standardized, safe and clear integration mechanism of such devices with the existing platforms is mandatory.

New network paradigms and ubiquitous computing for improvements in the remote healthcare can be affected by this need of interoperability that involves real-time data capture and the data storage process (for example, vital signals) related to patients.

1.1 Machine to Machine Communications

Machine to Machine (M2M) refers to the direct communication between devices through a mobile or fixed network [3]. In an M2M environment, the main goals are concerned with remote management, remote data access and maintenance of the data available in the devices. Two compliant devices can interact to solve objective problems

without the human presence. M2M applications usually consist of a hardware module installed in a frontend device, which communicates with the appropriate software support backend to provide different services to the final clients. The architectural elements follow a vertical behavior from the M2M Device at the bottom layer and the M2M application at the top layer, as shown in Figure 1.

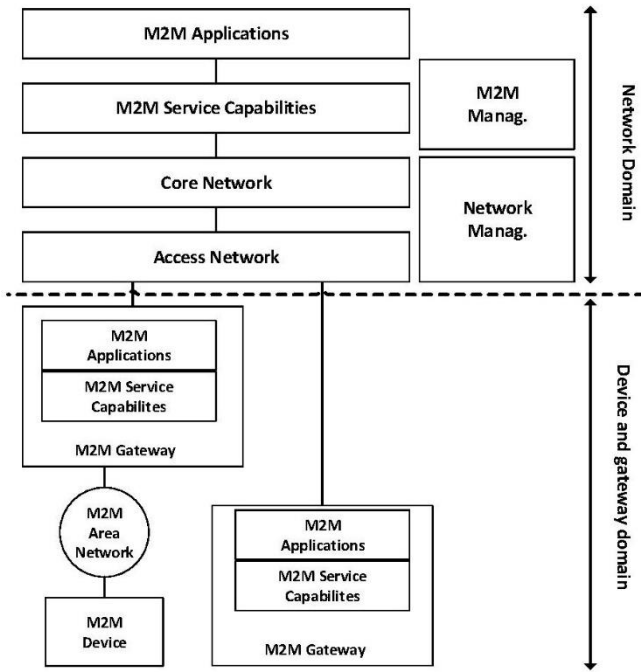


Figure 1 M2M architecture [4].

For an e-Health application, the M2M devices become e-Health devices that sense the different parameters of the patient or patient’s environment and forward the data to the final applications, at the top layer of the M2M architecture (in this case, applications for the management of the EHR and visual interfaces, algorithms, etc. that display and process the data). The following standard can be helpful for data and semantics representation in the EHR.

1.2 OpenEHR

OpenEHR [5] is an open standard intended for electronic health record systems. Based on a multi-level modeling approach, it separates medical semantics from data representation, so that domain experts can be directly involved in the semantics and define models known as archetypes and processes of clinical information systems using a much easier terminology.

The OpenEHR structure can be defined using a three level/layer approach. As shown in Figure 2, the reference model is the base layer and contains definitions of Data

Structures, Types, security aspects, and an identifier attribute that supports the indexation of each reference model. The next layer contains the models related to medical semantics, as archetypes and templates. Archetypes are defined in a computable language, called ADL (Archetype Definition Language). The top layer provides different services to the external context according to a service-oriented architecture.

For the sake of better understanding of OpenEHR structure, a brief discussion involving archetypes and templates is useful.

Different archetypes can be grouped into a template. A template can be seen as a composition of archetypes for more complex definitions, i.e. an archetype may refer only to a measure of a single physiological parameter, whereas a template can refer to a complete physiological parameters screening of a cardiovascular disease that includes archetypes for blood pressure, heart rate, electrocardiographic signals, etc. A template is often used for the definition of a case report form or a form to be filled. It enables the determination of the constructs to be used, events that must exist for each construct, elements to be used within an archetype, among other settings.

Another interesting characteristic is the existence of global repositories that provide standardized archetypes in different languages. This is an advantage in terms of interoperability and re-use of source code and applications. Moreover, this centralized distribution mechanism standardizes processes of medical semantics. A repository that includes the definition of templates, archetypes, models, etc. can be found in[6].

Virtual EHR			
terminology service	demographic service	EHR service	archetype service
EHR Extract			
EHR	Demographic	Integration	Template OM
Composition		OpenEHR Archetype Profile	
Security	Common	Archetype OM	ADL
Data Structures			
Data Types			
Support (identifiers, terminology, access)			

Figure 2 OpenEHR layers[5]

1.3 HL7

HL7 (Health Level 7) [4] refers to a set of international standards for the transfer of clinical or administrative data between software applications used by healthcare providers. HL7 v2.x is one of such message-based standards developed

by the Health Level Seven International and an ANSI accredited standard for the development of coherent and extensible mechanisms for the exchange, management and integration of electronic information in the clinical domain. The name refers to the applications layer of the OSI seven layers model, which describes how data are exchanged, the timing of the interchange and the handling of errors between different care providers. With the HL7 v2.x standards, definitions for different kinds of message, which can be helpful to standardize the communication in the e-Health Devices, are available. The ORU Message (Observation Result Unsolicited) is one of the most important messages in our work, because we treat the data measured by a sensor as observations related to a specific parameter.

This manuscript also proposes an architecture for the standardization of data collection and interoperability between M2M devices and applications focused on e-Health and that uses semantic information and the reference model of the OpenEHR standard. The main challenge is to provide transparency, ease of use and compatibility with different types of monitoring devices and e-Health applications.

2. Related works

P. R. Silva-Ferreira et al. [7] proposed an architecture for the compilation of data on patients from different health care institutions. It covers HL7-based information systems and an OpenEHR-based electronic health record. P. Vieira-Marques et al. [8] designed a data integration mechanism through which a virtual EHR is created and communicates with different institutions EHR by means of agents to provide doctors with information regardless of the location. They proposed a search mechanism within the national hospital network through communication among agents installed in each EHR.

C. Martinez-Costa et al. [9] developed a set of features to achieve semantic interoperability between standard OpenEHR and EN ISO 13606. The proposal is based on agents for the search and collection of information. Each HIS contains an agent who acts as a pair in the communication between different HISs, as in the previous proposal.

E. Osorio, L. Ferreira, R. Abreu, and F. Sousa [10] addressed interoperability between HL7 and OpenEHR-based HIS and proposed a system that assumes messages can arrive in three different formats, namely HL7 v2, V2.xml and OpenEHR. The MessageBroker entity, which works as an agent, receives data from external applications, converts them to a V2.XML format and forwards them to the EHRClient entity, responsible for interactions with the database management system.

J. D. Trigo et al. [11] focused on interoperability standards and proposed techniques for a harmonious application of OpenEHR and ISO / IEE 11073. They used the concept of concentration devices as an entity between HIS and the capture system (i.e. Laptops, PCs, Smartphones). The device recollects and processes all sensed data before forwarding them to the higher layers.

S. Frade et al. [12] conducted a review of storage implementations of OpenEHR systems and different representations of data, database management systems employed and query languages. They concluded most implementations are based on relational database management systems through SQL as the query language. Also, the authors analyzed some works that use Archetype Query Language to query the archetype medical data.

An example of OpenEHR and HL7-based system is provided in [13]. The authors propose an M2M (Machine to Machine) architecture based on a device M2M network for data capture and a smartphone as an M2M gateway. The smartphone employs HL7 as the communication protocol with an OpenEHR-based HIS. The proposal does not address problems of interoperability between the sensors of physiological parameters employed.

All such studies have addressed problems of semantic representation. The OpenEHR standard can be seen as a benchmark that regards semantics and interoperability. -HL7 v2 is an alternative for communication between devices; the choice is assessed by its use in the aforementioned studies. Another relevant aspect is the studies have treated problems of interoperability in the application layer. Such interoperability can be achieved through the creation of intermediate agents that understand the information in the different formats provided by the different standards, including lower layer protocols.

Our aim is to create a similar behavior in lower layers of an e-health architecture, so that the process of data capture can be standardized. As a first stage, we must define the network architecture to be used and the different entities that involve the remote monitoring system. Section 3 addresses all details of the architecture and the software development process and Section 4 introduces the proposal for interoperability that uses the OpenEHR semantics in the capture system's layer.

3. Target Architecture

Figure 3 [14, 15] shows a widely used architecture where the Internet access is guaranteed through a smartphone and a wireless communication network. The capturing system establishes a connection with the smartphone that forwards the sensed data to the web server (or EHR implementation)

periodically, through the mobile cellular network. Once the data have reached the Web server, the medical team can access them remotely via Websites or specific applications developed for the different types of services provided by the HIS.

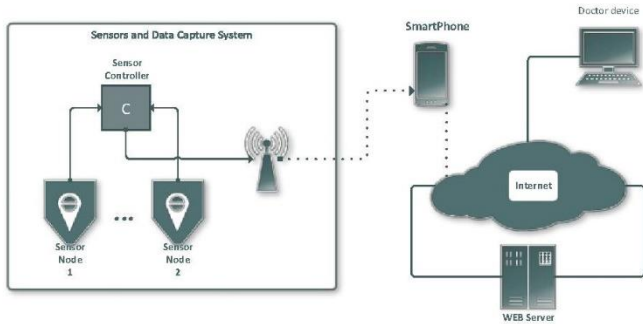


Figure 3 Mixed Architecture.

The main advantage of such an architecture is the Internet connection is guaranteed by the data plane hired by the patient through his/her cellular network operator. This feature frees the healthcare systems from hiring third-party services and has become an important aspect in countries of low economic performance or institutions that have started the automation process of healthcare. The user uses his/her phone with its data plane agreement to forward the information to the HIS. The use of smartphones increases the systems' functionalities regarding mobility and scalability of applications and such an architecture depends on the quality of service provided by the carriers.

As in [13], this architecture can be easily transformed into an M2M environment. Figure 4 shows the interaction between M2M gateway and the M2M device (respectively, the smartphone and the Sensor Controller in Figure 3). The M2M Core network can be represented as the Internet and the NSCL can be implemented as a set of services in the web server of Figure 3. The application of the doctor's devices

can be represented as an M2M application that uses the NSCL implementation as backend services. Finally, the EHR that stores all business data can be implemented and hosted in the remote server.

More formally and according to the ETSI TR 102 732 V1.1.1 [16], the main entities of the architecture displayed in Figure 4 are:

M2M Device Sensors (M2MDS): sensors assembled that measure the target parameters. They require a network interface to provide communication with the M2M Gateway;

M2M Smartphone Gateway (M2MSG): it acts as an M2M Gateway and provides communication between the M2M Device and the Service Capabilities Provider and concomitantly controls the startup and the behavior of the sensors. It can also provide visual interfaces for the network setup, authentication, doctor-patient notifications, and control of the sensors behavior (i.e. configuration of initial parameters, control commands to start and stop the sensors, etc.);

Network Service Capabilities Layer: it provides a set of services to all M2M applications for insertion, retrieval, update and deletion of data in the EHR and also enables communication between doctors and patients;

EHR: represented by a set of features for the registration of patients, doctors, and observations of the monitored parameters, this storage system can be implemented through the OpenEHR specification;

M2M Application: it enables health professionals to access captured data and can be developed as a smartphone or web application.

The protocols and hardware components used are described in the next section.

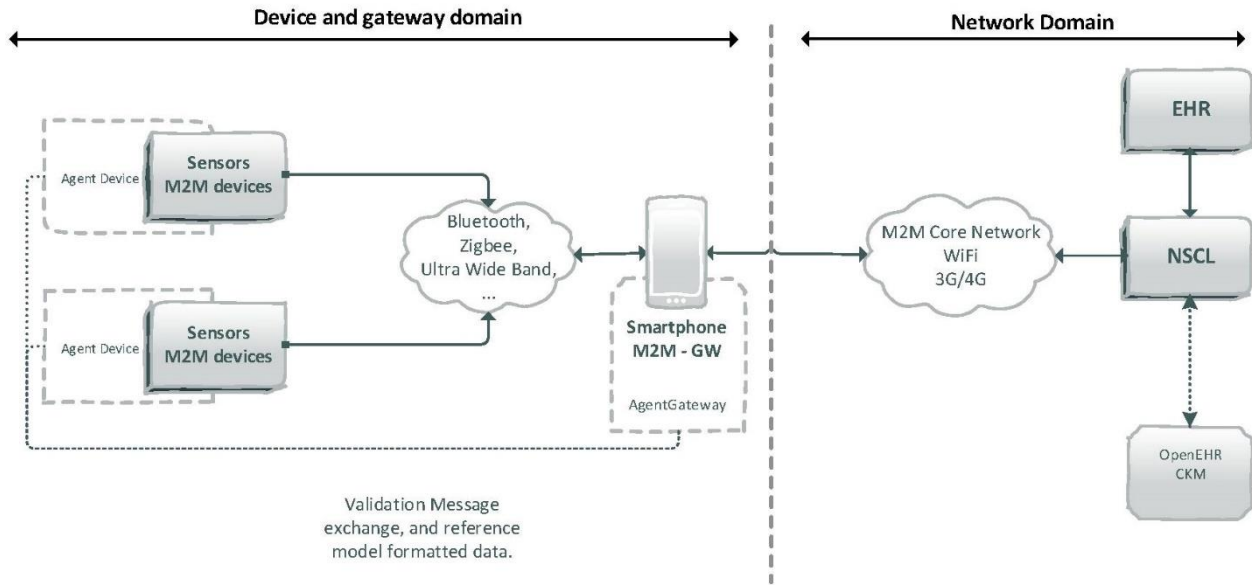


Figure 4 Target Architecture.

3.1 Hardware and Protocols Used

Cookinghack E-HEALTH Sensor Kit [17] and an Arduino Uno [18] development platform were used as sensors for the capture of physiological parameters (ECG, Airflow, Oxygen in the Blood, Temperature,...) and an HC06 Bluetooth interface from Guangzhou HC Information Technology Co. provided a communication interface in the M2M network. An Asus Zenfone5 smartphone with Android Lollipop operational system was used as an M2MSG. The Bluetooth interface of the smartphone and the HC06 modules enabled communication in the devices and gateway domain. An internet access, via Wi-Fi, connected the M2MSG with the NSCL.

As addressed in the previous sections, OpenEHR is a powerful tool for data and semantic representations, therefore, we have developed an OpenEHR-based EHR. A Microsoft SQL database, hosted in the Microsoft Azure Cloud System, was designed for the data storage and the XML representation was set as an attribute in the relational model for more advanced queries in future developments. Only archetypes relative to respiratory observations and diseases were used.

The HL7 v2.7 standard provided communication messages between the NSCL and the M2M gateway and between the M2M gateway and the M2M devices. A set of Rest APIs was developed in the Microsoft Azure cloud service provider to represent the NSCL and the following API functions were created for integration with the M2MSG:

- 1- Login: Using Http Post Method and receive the user Id and the password. **Returns true or false.**
- 2- SendData: Using Http Post Methods and receive an HL7 v2.7 ORU Message with the sensed data encoded. Such functions parse the HL7 messages and insert the data into the EHR.

The following Rest APIs were developed for other M2M applications that request information from the EHR:

- 1- RequestData: it receives an HL7 v2.7 request message, retrieves the information from the EHR and returns an HL7 message with the encoded requested data.
- 2- Notify Patient: it receives a Patient ID and a message and notifies the M2MSG through the Google Cloud Message System.

The M2M application that runs in the M2M gateway was developed for Android OS in Java; it connects Arduino via Bluetooth and receives the data captured by the three sensors. Once data has arrived, the application generates the corresponding HL7 message and consumes the Rest API in the NSCL.

The overall communication process among M2MSG, NSCL, EHR and the M2M client's applications follows the behavior shown in the diagram of Figure 5.

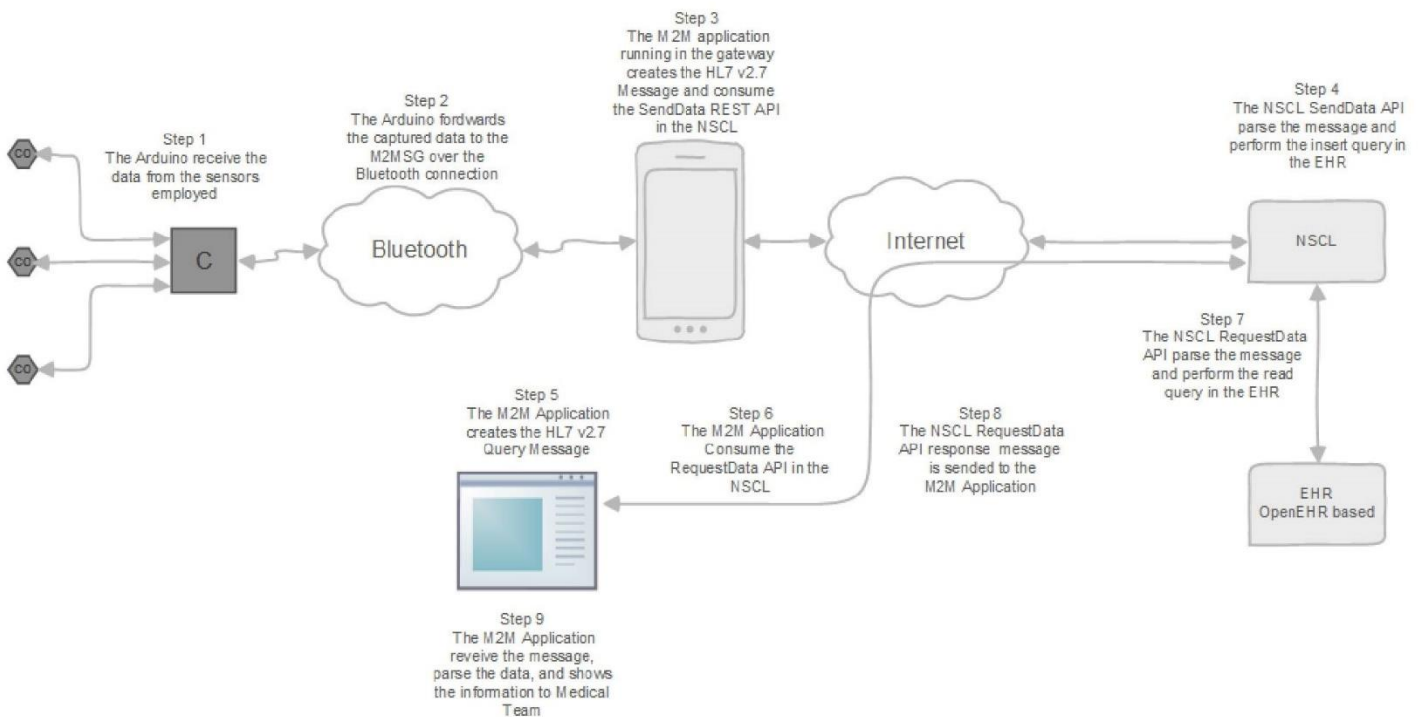


Figure 5 Architecture Behavior.

4. Extending OpenEHR semantics to a Sensor Context

The proposed architecture involves standards related to e-Health, M2M communication and cloud computing. The standardization of communication between the M2M gateway and NSCL is achieved through the HL7 v2.7 standard and the information stored follows the OpenEHR description, enabling the query of semantic data.

The first issue for the handling of the sensor scalability problem concerns the way a device can expose the data recollected. We propose the extension of the OpenEHR semantics into the M2M domain. In the OpenEHR definition, a reference model describes the way the data should be treated. For example, consider the “openEHR-EHR-OBSERVATION.body_temperature.v1” archetype provided in the OpenEHR Clinical Knowledge Manager (CKM) [5] for body temperature. It points to a reference model that has seven mandatory attributes (Body temperature, DateTime, etc.) and nine optional ones. The semantics in the archetype definition restrict the Reference Model behavior to Celsius or Fahrenheit degrees at a three-decimal precision. If an Electronic Thermometer developer

creates some equipment that fulfills all such semantic constraints, one can say the equipment has met all the temperatures archetype restrictions or it is “openEHR-EHR-OBSERVATION.body_temperature.v1” compatible.

An OpenEHR archetype can be used as an identification of the data type provided by the equipment and the data constraints the system has met. In our architecture, we have created the following agents on the M2M network side:

AgentDevice: located in the M2M Device, it provides communication with *AgentGateway* and stores a register with the Id of the archetypes or templates fulfilled by the device;.

AgentGateway: located in the M2M Gateway Application, it stores registers of Archetypes relevant to the screening application and provides a bi-directional communication mechanism with *AgentDevice*.

When an *AgentGateway* detects the presence of an *AgentDevice* in the network, a message exchange is initiated as follows:

- a) *AgentGateway* requests the archetypes met by *AgentDevice*;
- b) *AgentDevice* responds with a comma-separated list with at least one archetype ID;
- c) *AgentGateway* detects the compatible parameters through a comparison between the received archetype ID and the target;
- d) *AgentGateway* sends a message with the matches as a parameter;
- e) *AgentDevice* initializes the corresponding sensors and starts the transmission of the data.

The second challenge regarding the sensor scalability problem involves the definition of a computational data structure that stores and sends an observation of the parameters.

For an automatic generation of the data types that support the sensing process, we have developed a parser that provides C++ language structures from the archetypes and reference models of OpenEHR. C++ was used because our M2M device is Arduino-Based and has an analogic thermometer. In the software development process for the M2M device, such generated types can enable data format standardization and ensure interoperability. The raw data captured by the sensor can be received by *AgentDevice*, stored in an Object-Oriented structure with fully typed attributes and forwarded to the M2M gateway.

Then the C++ objects should be sent in a serialized XML format through a serial communication over the Bluetooth connection. Figure 6 shows a summary of the interaction of the proposed agents.

An abstraction of medical devices can be obtained and the e-Health applications can be compatible with a higher number of sensors. The compatibility with the M2M devices is guaranteed through a global format, guided by the OpenEHR Reference Model, while the data accuracy can be restricted by the archetypes semantics in the device's development phases. Logically, a device evaluation process will be necessary to ensure the sensor has fulfilled all archetypes constraints. The semantic data do not need to be explicitly present in the M2M device. The OpenEHR semantics are implicitly extended during the creation of the M2M device by an agent that knows the archetypes ID with which all constraints comply.

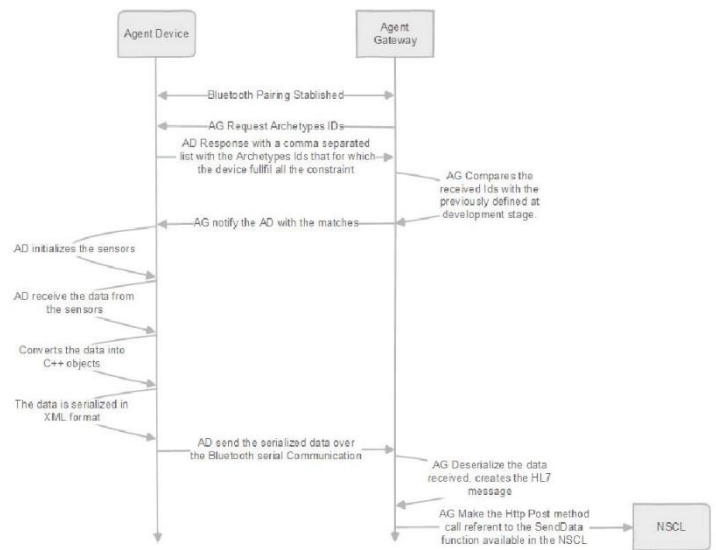


Figure 6 Behavior of the Capture System.

5. Conclusions

This manuscript has presented an M2M architecture for e-Health in which the semantics of the OpenEHR dual model approach have been implicitly extended to the M2MDevice to provide scalability to actual e-Health Systems. The global impact of OpenEHR can be exploited, so as to restrict the quality of the sensed data and standardize the data format at low levels in the e-Health architecture.

The architecture enables the use of M2M and Cloud Computing concepts for reductions in both implementation time and costs of the entire system, and leverages the potential of OpenEHR specifications.

New scenarios can be included and new applications can take advantage of the non-depending sensor characteristic of the proposal, allowing an extended view of the application of OpenEHR semantics for the sake of achieving reusability and interoperability in M2M e-Health devices. Finally, as future work, we plan to update the architecture for the new version of the HL7 v3 standard, and make experiments in a real environment to validate the whole system proposed .

Acknowledgment

The authors acknowledge the financial support from the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) to Jesús Noel Suárez Rubí through the PPGEE Program of the University of Brasilia.

References

- [1] G. Eysenbach, "What is e-health?," *Journal of medical Internet research*, vol. 3, 2001,
- [2] K. Häyrynen, K. Saranto, and P. Nykänen, "Definition, structure, content, use and impacts of electronic health records: a review of the research literature," *International journal of medical informatics*, vol. 77, 2008, pp. 291-304.
- [3] D. Lucic, A. Caric, and I. Lovrek, "Standardisation and regulatory context of machine-to-machine communication," in *Telecommunications (ConTEL), 2015 13th International Conference on*, 2015, pp. 1-7.
- [4] HL7, *Standard: Refinement, Constraint and Localization, Release 2*, Available: <http://www.hl7.org/implement/standards/ansiapproved.cfm>, Accessed: 7/8/2015.
- [5] T. Beale, S. Heard, D. Kalra, and D. Lloyd, *OpenEHR architecture overview*, Available: <http://www.openehr.org/>, Accessed: 12/12/2015.
- [6] OpenEHR, *OpenEHR Clinical Knowledge Manager*, Available: <http://www.openehr.org/ckm/>, Accessed: 12/2015/2015.
- [7] P. R. Silva-Ferreira, J. H. Patriarca-Almeida, P. M. Vieira-Marques, and R. J. Cruz-Correia, "Improving expressiveness of agents using openEHR to retrieve multi-institutional health data: Feeding local repositories through HL7 based providers," in *Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on*, 2012, pp. 1-5.
- [8] P. Vieira-Marques, J. Patriarca-Almeida, S. Frade, G. Bacelar-Silva, S. Robles, *et al.*, "OpenEHR aware multi agent system for inter-institutional health data integration," in *Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on*, 2014, pp. 1-6.
- [9] C. Martínez-Costa, M. Menárguez-Tortosa, and J. T. Fernández-Breis, "An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes," *Journal of biomedical informatics*, vol. 43, 2010, pp. 736-746.
- [10] E. Osorio, L. Ferreira, R. Abreu, and F. Sousa, "Interoperability in Ambient Assisted Living using OpenEHR," in *e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on*, 2013, pp. 394-398.
- [11] J. D. Trigo, C. D. Kohl, A. Eguzkiza, M. Martinez-Espronedá, A. Alesanco, *et al.*, "On the seamless, harmonized use of ISO/IEEE11073 and openEHR," *Biomedical and Health Informatics, IEEE Journal of*, vol. 18, 2014, pp. 872-884.
- [12] S. Frade, S. Miranda Freire, E. Sundvall, J. H. Patriarca-Almeida, and R. Cruz-Correia, "Survey of openEHR storage implementations," in *Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on*, 2013, pp. 303-307.
- [13] C. Pereira, S. Frade, P. Brandao, R. Correia, and A. Aguiar, "Integrating data and network standards into an interoperable e-Health solution," in *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*, 2014, pp. 99-104.
- [14] R. Gao, L. Yang, X. Wu, T. Wang, S. Lu, *et al.*, "A phone-based e-health system for osas and its energy issue," in *Information Technology in Medicine and Education (ITME), 2012 International Symposium on*, 2012, pp. 682-686.
- [15] A. M. Bhatti and M. Masud, "Context Aware Intelligent Wallet for Healthcare," *International Journal of Computer Applications*, vol. 96, 2014, pp. 31-36.
- [16] E. T. S. I. (ETSI), *Machine to Machine Communications*, Available: <http://www.etsi.org/technologies-clusters/technologies/m2m>, Accessed: 27/6/2015.
- [17] Libelium, *E-HEALTH SENSOR PLATFORM COMPLETE KIT V2.0 FOR ARDUINO, RASPBERRY PI AND INTEL GALILEO [BIOMETRIC / MEDICAL APPLICATIONS]*, Available: <http://www.cooking-hacks.com/ehealth-sensors-complete-kit-biometric-medical-arduino-raspberry-pi>, Accessed: 18/03/2015/2015.
- [18] *Arduino*, Available: <http://arduino.cc/>, Accessed: 28/02/2015.