



Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

16037 - Nelson Miguel Pereira Gomes



VITASENIOR-MT: VITABOX

Relatório de Projeto
Trabalho Final de Mestrado

Orientado por:

Professor: **Gabriel Pereira Pires**

Professor: **António Manuel Rodrigues Manso**

Dissertação apresentada ao Instituto Politécnico de Tomar
para cumprimento dos requisitos necessários
à obtenção do grau de Mestre em Engenharia Informática – Internet das Coisas

Resumo

A monitorização remota dos parâmetros de saúde é uma abordagem promissora para melhorar o estado de saúde e a qualidade de vida de determinados grupos de população, e que pode também aliviar os gastos dos sistemas de saúde. Os idosos, geralmente afetados por problemas crónicos, são um grupo específico da população que pode beneficiar fortemente das tecnologias de tele-saúde, permitindo-lhes alcançar uma vida mais independente, vivendo mais tempo nas suas próprias casas.

O trabalho desenvolvido neste projeto insere-se no projeto de I&D VITASENIOR-MT que tem por objetivo a implementação de um sistema de tele-saúde e teleassistência dirigido à população idosa. O projeto VITASENIOR-MT é constituído por várias partes, nomeadamente, 1) uma rede de sensores para monitorização ambiental da casa do utilizador, 2) um módulo (VITABOX) que gere a interação do utilizador com equipamentos médicos através da televisão usando o comando remoto, e que agrega todos os dados biométricos e ambientais e os reencaminha para o servidor da *cloud*; e 3) um sistema de informação que armazena todos os dados de todos os sistemas colocados nas várias habitações e os disponibiliza aos vários intervenientes do sistema (utilizador, cuidador e médico).

Este projeto de Mestrado focou-se no desenvolvimento do módulo VITABOX. Baseado na tecnologia *Nodejs*, implementou-se um sistema que faz a interação com a TV através do protocolo HDMI-CEC e recebe ordens do comando remoto da TV; comunica com equipamentos biométricos (medidor de pressão arterial, balança, pulsómetro, batimento cardíaco, medição de glucose) através de *Bluetooth*; adquire dados ambientais da rede de sensores; disponibiliza graficamente e com áudio-descrição os dados biométricos e ambientais recolhidos. O funcionamento e usabilidade do sistema foram testados com sucesso em ambiente laboratorial e por grupos de participantes, sendo os resultados muito satisfatórios. O sistema encontra-se ainda em fase de otimização de funcionalidades e será colocado em ambiente de teste real brevemente.

Palavras Chave (Tema): Téle-saúde, Téle-monitorização, Tv interativa

Palavras Chave (Tecnologias): *Vuejs, Nodejs, MongoDB, Bluetooth*

Abstract

Remote monitoring of health parameters is a promising approach to improve the health condition and quality of life of particular groups of the population, which can also reduce the current outlay and demands of healthcare systems. The elderly, usually affected by chronic comorbidities, are a specific group of the population that can strongly benefit from telehealth technologies, allowing them to reach a more independent life, by living longer in their own homes.

The work developed in this project inserts in the I&D VITASENIOR-MT project which aims at the implementation of a tele-health and teleassistance system directed at the elderly population. The VITASENIOR-MT project consists of various parts, namely: 1) a network of sensors for environmental monitoring of the user's home, 2) one module (VITABOX) that manages the user's interaction with medical equipment through the television using the remote control, and which includes all biometric and environmental data and forwards them to the cloud server; and 3) an information system that stores all the information of all the systems placed in the various dwellings and makes them available to the various stakeholders of the system (user, caregiver and doctor).

This Master's degree project focused on the development of the VITABOX module. Based on the Nodejs technology, a system was implemented that interacts with the TV through the HDMI-CEC protocol and receives orders from the remote control of the TV; communicates with biometric equipment (blood pressure meter, scale, heart rate monitor, heart rate meter, glucose measurement) through Bluetooth; acquires environmental data from the sensor network; provides graphically and with audio description the received biometric and environmental data. The operation and usability of the system were successfully tested in a laboratory environment and by groups of participants, and the results were very satisfactory. The system is still in the phase of optimization of functionalities and will be placed in real test environment briefly.

Keywords - Telehealth, Elderly, Healthcare, TV interaction, biometric data, environmental data

Agradecimentos

Pretendo em primeiro lugar, prestar o meu agradecimento a todos aqueles que de alguma forma contribuíram para que pudesse realizar este projeto.

Sem qualquer critério ou relevância agradeço aos bolseiros Pedro Dias e Diogo Mendes, que ajudaram na realização do projeto global e no desbloqueio de alguns problemas, e agradecer ao Dário Jorge que ajudou na realização desta parte do projeto no desenvolvimento de alguns módulos que serão descritos ao longo deste documento.

Não posso deixar de dar o meu agradecimento aos orientadores por toda a dedicação, empenho e acompanhamento na realização deste projeto e na escrita e correção deste relatório.

Este trabalho foi suportado financeiramente pelo projeto IC&DT VITASENIOR-MT CENTRO-01-0145-FEDER-023659 com fundos do FEDER através dos programas operacionais CENTRO2020 e FCT.

Agradeço ao lab. VITA.IPT e ao Instituto Politécnico de Tomar pelas condições proporcionadas para a concretização deste trabalho.

Índice

Resumo.....	v
Abstract	vii
Agradecimentos	ix
Índice.....	xi
Índice de Figuras.....	xvii
Índice de Tabelas	xxi
Notação e Glossário.....	xxiii
1 Introdução.....	1
1.1 Enquadramento	1
1.2 Trabalho desenvolvido	2
1.2.1 VITABOX	2
1.2.2 Interação VITABOX - Rede de sensores ambientais.....	2
1.2.3 Interação VITABOX - Equipamentos biométricos.....	3
1.2.4 Interação VITABOX-TV: HDMI	3
1.3 Tecnologias utilizadas	3
1.4 Contributos.....	5
1.5 Organização do relatório.....	6
2 Estado da arte	7
2.1 Sistemas de Tele-saúde	7
2.1.1 Nourish Sensors.....	7
2.1.1.1 Visão geral do sistema	7
2.1.1.2 Módulo de interface entre a rede de sensores e o sistema central	8
2.1.1.2.1 Hardware	9
2.1.1.2.2 Software	9
2.1.2 eVida.....	9

2.1.3	<i>MySignals</i>	10
2.1.4	10 Mil Vidas	11
2.1.5	<i>SMARTAL – SMART ASSISTED LIVING</i>	12
2.1.6	Monitorização remota da saúde para idosos através da televisão interativa	14
2.1.6.1	Arquitetura do sistema e fluxos de dados.....	14
2.1.6.2	Layout do aplicativo.....	16
2.1.6.3	Resultados	17
2.2	Comparação entre os vários sistemas	18
3	Tecnologias Utilizadas	19
3.1	HDMI-CEC	19
3.2	Unidade central de processamento	21
3.2.1	<i>Raspberry Pi 2</i>	21
3.2.2	<i>Raspberry Pi 3</i>	23
3.2.3	<i>Cubie Board 2</i>	24
3.2.4	<i>Comparação e Escolha</i>	25
3.3	Persistências de dados	25
3.3.1	Modelo Relacional.....	25
3.3.1.1	<i>MySQL</i>	26
3.3.1.2	<i>SQLite</i>	26
3.3.2	Modelo Não Relacional	27
3.3.2.1	<i>MongoDB</i>	27
3.3.2.2	<i>RethinkDB</i>	27
3.3.3	<i>Comparação e Escolha</i>	28
3.4	Servidor web	28
3.4.1	<i>Apache</i>	28
3.4.2	<i>NodeJs</i>	30
3.4.3	<i>Nginx</i>	30
3.4.4	<i>Comparação e Escolha</i>	31

3.4.4.1	<i>Noble</i>	32
3.4.4.2	<i>Child-process</i>	32
3.4.4.3	<i>Socket.io</i>	32
3.4.4.4	<i>Node-cec e cec-monitor</i>	33
3.4.4.5	<i>Mongoose</i>	33
3.4.4.6	<i>Audioconcat</i>	34
3.5	Interface Gráfica	34
3.5.1	<i>Backbone.js</i>	34
3.5.2	<i>React.js</i>	35
3.5.3	<i>Vue.js</i>	35
3.5.4	Comparação e Escolha	35
3.5.4.1	<i>Template Vue Paper Dashboard</i>	35
3.5.4.2	<i>JustGage</i>	36
3.5.4.3	<i>Canvas.js</i>	36
3.5.4.4	<i>EventBus</i>	37
3.5.4.5	<i>Internacionalização - I18n</i>	37
3.5.4.6	<i>Fontawesome</i>	37
4	VITABOX	39
4.1	Arquitetura global do Projeto VITASENIOR-MT	39
4.2	Arquitetura da VITABOX	40
4.3	HDMI: interação Televisão - VITABOX	42
4.4	Interação com os dispositivos <i>Bluetooth</i>	44
4.4.1	Sistema automático de captura de dados biométricos	47
4.5	Associação dos sensores ambientais à VITABOX	48
4.6	Armazenamento de dados	49
4.6.1	Sensores ambientais	49
4.6.1.1	Armazenamento - 1ª – Método utilizado	49
4.6.1.2	Armazenamento - 2ª – Método utilizado	50

4.6.2	Equipamentos Biométricos	52
4.6.3	Dados Pré-inseridos	53
4.7	VITABOX – Modos (<i>Prod, Dev</i>)	53
4.8	Internacionalização - I18n.....	55
4.9	Audio Descrição	56
4.10	Interface Gráfica.....	56
4.10.1	Painel de Configurações	56
4.10.2	Modo Avançado e Modo Básico.....	57
4.10.2.1	Modo Avançado	57
4.10.2.2	Modo Básico	58
4.11	Sistema de notificações e alertas.....	58
4.12	Comunicação com o Sistema Remoto	59
5	<i>Resultados experimentais</i>	63
5.1	Testes de carga	63
5.1.1	Teste - 1ª estrutura do Objeto de armazenamento de dados	64
5.1.2	Teste – 2º Objeto de armazenamento de dados.....	65
5.2	Testes na utilização de animações na interface gráfica	66
5.3	Testes de usabilidade	69
6	<i>Conclusões</i>	79
6.1	Objetivos alcançados	80
6.2	Limitações & trabalho futuro	81
	<i>Bibliografia</i>	83
Anexo 1	<i>Script para captura automática de dados da banda biométrica</i>.....	87
Anexo 2	<i>Script para execução no crontab</i>.....	89
Anexo 3	<i>Função para criar array de ficheiros de audio de um número</i>	91
Anexo 4	<i>Procedimento para a realização das tarefas</i>	93
Anexo 5	<i>Questionário de satisfação</i>	95

Índice de Figuras

<i>Figura 1 - Esquema geral do sistema - Nourish Sensors.</i>	8
<i>Figura 2 - Exemplo da interface gráfica do sistema.</i>	8
<i>Figura 3 - Esquema dos vários sensores do sistema MySignal.</i>	10
<i>Figura 4 - Interface do sistema MySignal para as diferentes plataformas.</i>	11
<i>Figura 5 - Exemplo da aplicação da SmartAL</i>	13
<i>Figura 6 - Esquema da rede SmartAL</i>	13
<i>Figura 7 - Rede de sensores biomédicos utilizados para monitorizar remotamente o Utente</i>	14
<i>Figura 8 - Arquitetura do sistema de monitorização remota</i>	15
<i>Figura 9 - Exemplo de autenticação na STB</i>	16
<i>Figura 10 - Exemplo da utilização da Teleassistência</i>	16
<i>Figura 11 - Descrição dos pins da ligação HDMI</i>	19
<i>Figura 12 - Endereçamento CEC (PA – Physical address; LA – Logical address).</i>	20
<i>Figura 13 - Estrutura do protocolo CEC</i>	21
<i>Figura 14 - Board Raspberry Pi 2 Model B</i>	22
<i>Figura 15 - Board Raspberry Pi 3</i>	23
<i>Figura 16 - Board Cubieboard2</i>	24
<i>Figura 17 – Exemplo do gráfico utilizando a biblioteca Justgage</i>	36
<i>Figura 18 - Exemplo do gráfico utilizando a nova versão do Justgage</i>	36
<i>Figura 19 - Visão global do projeto VITASENIOR-MT</i>	39
<i>Figura 20 – Arquitetura geral da aplicação VITABOX.</i>	41
<i>Figura 21 - Exemplo das teclas utilizadas na navegação na interface</i>	42
<i>Figura 22 - Exemplo da utilização das teclas para navegação na barra lateral</i>	43
<i>Figura 23 - Exemplo da utilização do comando dentro de uma opção da barra lateral</i>	43
<i>Figura 24 - Exemplo da listagem dos utilizadores e dos exames a ele atribuídos</i>	45
<i>Figura 25 - Exemplos do bloqueio do ecrã durante a execução dos exames</i>	45
<i>Figura 26 - Exemplos de mensagens informativas durante a execução dos exames</i>	46
<i>Figura 27 - Apresentação do resultado de um exame</i>	46

<i>Figura 28 - Mensagem de erro durante a execução do exame</i>	46
<i>Figura 29 - Mensagem de equipamento não encontrado</i>	46
<i>Figura 30 - Mensagem de exame cancelado por ordem do utilizador.</i>	47
<i>Figura 31 - Entrada adicionada ao crontab</i>	47
<i>Figura 32 – Nó da rede e border router. Exemplo de associação do um novo nó</i>	48
<i>Figura 33 –Estrutura do objeto Sensor 1ª abordagem</i>	50
<i>Figura 34 - Estrutura do objeto Sensor seguindo a 2ª abordagem</i>	51
<i>Figura 35 - Estrutura do objeto RawSensor</i>	51
<i>Figura 36 - Estruturado objeto do paciente</i>	52
<i>Figura 37 – Estrutura do objeto das configurações das teclas do comando remoto</i>	53
<i>Figura 38 – Scripts da definição do ambiente em execução</i>	54
<i>Figura 39 - Carregamento das configurações de acordo com o modo de execução do servidor</i>	55
<i>Figura 40 - Pannel de configurações</i>	57
<i>Figura 41 – Exemplo de alerta de um sensor</i>	58
<i>Figura 42 - Mensagem de notificação da existência de novas mensagens na VITABOX</i>	59
<i>Figura 43 - Exemplo da uma mensagem recebida pela VITABOX para o utilizador.</i>	59
<i>Figura 44 - Exemplo de utilizador com três exames e três variáveis de um exame (Pressão arterial)</i>	60
<i>Figura 45 - Exemplo de utilizador com cinco exames e duas variáveis de um exame (Pressão arterial)</i>	61
<i>Figura 46 - Histórico do sensor biométrico da medição da pressão arterial</i>	62
<i>Figura 47 - Histórico do sensor ambiental Temperatura</i>	62
<i>Figura 48 – Percentagem da utilização do CPU (%)</i>	64
<i>Figura 49 – Percentagem da utilização da memória RAM (%)</i>	64
<i>Figura 50 - Imagem retirada do equipamento ao fim de 25 minutos de funcionamento</i>	65
<i>Figura 51 - Imagem retirada do equipamento ao fim de 2 horas de funcionamento</i>	65
<i>Figura 52 - Percentagem da utilização do CPU (%)</i>	65
<i>Figura 53 - Percentagem da utilização da memoria RAM (%)</i>	66
<i>Figura 54 - Imagem retirada do equipamento ao fim de 9 horas de funcionamento</i>	66
<i>Figura 55 - Imagem retirada do equipamento ao fim de 20 horas de funcionamento</i>	66
<i>Figura 56 - Percentagem da utilização do CPU (%) com animações</i>	67

<i>Figura 57 - Percentagem da utilização do CPU (%) sem animações</i>	67
<i>Figura 58 - Percentagem da utilização da Memória RAM (%) com animações</i>	68
<i>Figura 59 - Percentagem da utilização do Memória RAM (%) sem animações</i>	68
<i>Figura 60 - Sequência da animação na apresentação dos resultados</i>	69
<i>Figura 61 – Apresentação de resultados sem animações</i>	69
<i>Figura 62 - Tempos de realização dos diferentes testes</i>	70
<i>Figura 63 - Relação entre o tempo médio total em relação comm a idade do utilizador</i>	71
<i>Figura 64 – Relação da idade dos participantes e o tempo médio da realização das tarefas</i>	72
<i>Figura 65 - Gráfico da média do grau de satisfação da realização dos exames</i>	73
<i>Figura 66 - Gráfico da média do grau de satisfação do histórico dos exames</i>	74
<i>Figura 67 - Gráfico da média do grau de satisfação do histórico dos sensores ambientais</i>	74
<i>Figura 68 - Gráfico da utilidade da informação do histórico dos exames realizados</i>	75
<i>Figura 69 - Gráfico da utilidade da informação do histórico dos sensores ambientais</i>	75
<i>Figura 70 - Gráfico da utilização da audio-descrição na aplicação</i>	76

Índice de Tabelas

<i>Tabela 1 - Hardware utilizado nas STB</i>	17
<i>Tabela 2 - Tempos gastos na autenticação</i>	17
<i>Tabela 3 - Tabela comparativa entre sistemas</i>	18
<i>Tabela 4- Designação da tecnologia CEC por fabricante</i>	20
<i>Tabela 5 - Características do Raspberry Pi 2 Model B</i>	22
<i>Tabela 6 - Características do Raspberry Pi 3 Model B</i>	23
<i>Tabela 7 - Características do CubieBoard 2</i>	24
<i>Tabela 8 - Tabela comparativa resumida dos microcomputadores considerados</i>	25
<i>Tabela 9 – Caraterização dos participantes que realizaram testes de usabilidade</i>	70
<i>Tabela 10 - Quantidade de erros cometidos por cada utilizador por tarefa</i>	71
<i>Tabela 11 - Tabela com a média das respostas da realização dos exames dadas pelos participantes</i>	72
<i>Tabela 12 - Tabela com a média das respostas do histórico dos exames dadas pelos participantes</i>	73
<i>Tabela 13 - Tabela com a média das respostas do histórico dos sensores ambientais dadas pelos participantes</i>	74

Notação e Glossário

API	<i>Application Programming Interface</i>
CEC	<i>Consumer Electronics Control</i>
CPU	<i>Central Process Unit</i>
CSI	<i>Camera Serial Interface</i>
DSI	<i>Display Serial Interface</i>
DSO	<i>Dynamic Shared Objects</i>
DVB-T	<i>Digital Video Broadcasting — Terrestrial</i>
GPIO	<i>General Purpose Interface Output</i>
HDMI	<i>High-Definition Multimedia Interface</i>
IMAP	<i>Internet Message Access Protocol</i>
IP	<i>Internet Protocol</i>
IPv6	<i>Internet Protocol Version 6</i>
JSON	<i>JavaScript Object Notation</i>
LAMP	<i>Linux, Apache, MySQL e Perl/PHP/Python</i>
MAC	<i>Media Access Control</i>
NoSQL	<i>Not Only Structured Query Language</i>
POP3	<i>Post Office Protocol 3</i>
RAM	<i>Random Access Memory</i>
SGBD	<i>Sistema de Gestão de Base de Dados</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SQL	<i>Structured Query Language</i>
SSL	<i>Secure Sockets Layer</i>
STB	<i>Set-Top-Box</i>
URL	<i>Uniform Resource Locator</i>
WAMP	<i>Windows, Apache, MySQL e Perl/PHP/Python</i>

1 Introdução

1.1 Enquadramento

Este trabalho foi realizado no âmbito da disciplina de projeto do mestrado em Eng. Informática-IdC, e enquadra-se no projeto financiado IC&DT VITASENIOR-MT: *Senior healthcare assistance in* Medio Tejo (Assistência aos cuidados de saúde de idosos no Médio Tejo), a decorrer no lab. VITA.IPT. O projeto VITASENIOR-MT tem como principal objetivo a monitorização de parâmetros de saúde (e.g., pressão arterial, peso, glicémia, oximetria) dos utentes, bem como de parâmetros ambientais das suas residências, nomeadamente, temperatura ambiente, níveis de monóxido e dióxido de carbono, humidade, etc. O responsável de saúde, consultando os históricos dos parâmetros clínicos poderá monitorizar o estado de saúde do utente e/ou antecipar diagnósticos clínicos, bem como definir planos de exames e dar recomendações, que chegarão ao utilizador através da televisão.

O utente inicia o processo de medição dos parâmetros clínicos através da televisão, interagindo com os dispositivos médicos através do comando da televisão. Poderá igualmente consultar o histórico das medições clínicas e dos dados ambientais recolhidos com uma rede de sensores distribuída na sua habitação.

Este projeto foi dividido em várias partes;

- Rede de sensores que recolhem dados ambientais com mecanismos e regras de segurança e de associação de novos equipamentos,
- VITABOX: Sistema de receção dos dados dos sensores ambientais, interligação com os sensores *Bluetooth* para a realização dos vários exames médicos, interligação por HDMI (*High-Definition Multimedia Interface*) para apresentação da interface gráfica na televisão, usando o comando remoto para interação com o utilizador; envio da informação recolhida dos sensores ambientais e biométricos para um sistema remoto (servidor), receção de configurações e alertas informativos enviados do sistema remoto.

Este módulo é a base da realização desta tese, que fisicamente é designado por VITABOX,

- Sistema remoto que dispõe de uma interface de visualização dos dados recebidos das várias VITABOXs e de uma interface de configuração dos equipamentos associados a cada VITABOX.

1.2 Trabalho desenvolvido

Neste projeto, foi desenvolvido trabalho principalmente nos módulos de interligação da VITABOX com a televisão recorrendo à exploração da tecnologia HDMI nomeadamente a propriedade CEC (*Consumer Electronic Control*), utilização de equipamentos *Bluetooth* para a realização de pequenos exames, desenvolvimento do módulo para a utilização de *Sockets* para a comunicação em tempo real. Houve várias partes que foram desenvolvidas em colaboração com outros membros do projeto VITASENIOR-MT.

A VITABOX dispõe de um servidor *Nodejs* que disponibiliza a interface gráfica na televisão. Neste servidor foram desenvolvidos outros módulos que dão a possibilidade de explorar a propriedade CEC para a navegação na interface utilizando o comando da televisão, módulo que com a adição de um outro módulo publico (*Noble*) deram a possibilidade de interagir com os equipamentos *Bluetooth*.

1.2.1 VITABOX

A VITABOX é o equipamento central em casa do utilizador final, o qual se encontra ligado à televisão. É responsável pela interligação dos diversos módulos constituintes do sistema: receção das informações recolhidas pelos sensores ambientais (humidade, temperatura ambiental, monóxido de carbono, etc.); apresentação visual/gráfica na televisão dos dados recolhidos na interação com os equipamentos *Bluetooth* (medidor de pressão arterial, temperatura corporal, glicémia, etc.) com visualização dos resultados obtidos; comunicação dos dados para o sistema remoto; disponibilização de uma interface gráfica na televisão para interação com o utilizador através do controlo remoto (comando) da mesma.

1.2.2 Interação VITABOX - Rede de sensores ambientais

Para monitorizar o ambiente utilizámos uma rede de *motes* (*transceiver wireless + sensors*) estrategicamente colocados na habitação do utilizador, onde cada um deles tem acoplados vários transdutores; temperatura, humidade, dióxido de carbono e monóxido de carbono. Este tópico será aprofundado no âmbito de outro trabalho relacionado com o projecto VITASENIOR-MT.

1.2.3 Interação VITABOX - Equipamentos biométricos

Para monitorizar os dados clínicos dos utentes utilizámos um conjunto de equipamentos de medição de parâmetros fisiológicos, nomeadamente, pressão arterial, temperatura corporal, balança eletrónica, glicémia, etc. Esses equipamentos usam tecnologia *Bluetooth* para estabelecer a comunicação com a VITABOX, e a manipulação e obtenção dos resultados da execução com o utente.

1.2.4 Interação VITABOX-TV: HDMI

A VITABOX encontra-se ligada à televisão através de um cabo HDMI. A VITABOX disponibilizará uma interface para visualização e utilização dos equipamentos indicados no ponto anterior, sendo a interação com o utilizador conseguida através do controlo remoto da televisão.

1.3 Tecnologias utilizadas

Devido à natureza do projecto e dos vários objetivos a que se propõem resolver, sentimos a necessidade de encontrar tecnologias de fácil integração e interligação que não exigissem muito recursos computacionais, pois em alguns casos teremos dispositivos com poucos recursos de processamento, memória e energia.

A VITABOX terá de disponibilizar uma interface gráfica que será exibida na televisão, gerir a interação HDMI com a televisão, gerir a receção de dados via Wi-Fi e enviar informação para um sistema remoto na *cloud*.

Foram analisados os servidores *Apache*, *Nodejs*, *Nginx*. Escolhemos o *Nodejs*¹ que é uma tecnologia que possui uma enorme diversidade de módulos que facilmente conseguem fazer a interligação entre os vários sistemas, e possibilita uma fácil integração da tecnologia de *WebSockets* [1] [2] para dar suporte a atualização de dados em tempo real, utilizados para a utilização do comando da televisão para interagir com a aplicação.

A utilização do servidor *Nodejs* é vantajosa devido à enorme comunidade na criação e suporte aos módulos que podem ser integrados no servidor *Nodejs* nomeadamente para

¹ Disponível em (<https://nodejs.org/en/>)

utilização do protocolo CEC presente no HDMI, tais como o *node-cec*² e *cec-monitor*³. Estes módulos conseguem monitorizar e interagir com a porta HDMI, o *Noble* para interagir com os equipamentos *Bluetooth* entre outros.

Como o sistema recebe uma grande quantidade de dados, tivemos a necessidade de os armazenar. Esses dados podem ser sujeitos a algum tipo de tratamento/processamento que são posteriormente enviados para o sistema remoto. Dentro das inúmeras possibilidades e diferentes abordagens, apenas foram considerados os mais conhecidos, para o modelo relacional considerámos o *MySQL*, *SQLite*, e no modelo não relacional considerámos o *MongoDB*, *RethinkDB* [3]. Dada a grande quantidade de dados que o sistema irá receber e gerar e da necessidade de rapidamente serem armazenados podendo os seus atributos variarem, a nossa escolha recaiu sobre o *MongoDB* [4], pois permite a flexibilidade da construção dos objetos a serem armazenados sem que se tenha concluído a definição dos vários atributos a armazenar, bem como o rápido armazenamento dos dados. É uma tecnologia de fácil integração no servidor *Nodejs* e para além disso a linguagem para interagir com os dados é de fácil utilização. Para efetuar a ligação entre a base de dados e o servidor foi utilizado o módulo *mongoose*.

Uma das nossas preocupações foi desenvolver um sistema de baixo custo. De acordo com esta premissa foram considerados diversos equipamentos com capacidade para alojar o servidor web, a base de dados local, que possuísse conectividade *Bluetooth* para interagir com os equipamentos biométricos, conectividade *Wi-Fi* e ou ligação *Ethernet* para comunicação com o sistema remoto, e com alguma capacidade computacional. Os equipamentos analisados foram o *Raspberry pi 2*, *Raspberry Pi 3*, *Cubie Board 2*. Após alguns testes a nossa escolha recaiu no *Raspberry Pi 3*, devido ao seu poder computacional e à comunidade que dá suporte ao equipamento.

Para o desenvolvimento de todas as páginas Web e componentes necessários para disponibilizar a interface gráfica foram analisadas diferentes plataformas, nomeadamente *Backbon.js*, *Reactjs* e *Vue.js*. Após análise entre as diferentes *frameworks* excluímos o *Backbone.js* pois é uma plataforma mais antiga face às restantes analisadas, tendo a nossa escolha recaído no *Vue.js* embora de acordo com o apresentado na análise efetuado em [5]

² Repositório disponível em (<https://github.com/patlux/node-cec>).

³ Repositório disponível em (<https://github.com/senzil/cec-monitor>).

aponta-se para uma maior utilização de *Reactjs*, mas o estudo conclui que todas as plataformas têm as suas vantagens e desvantagens.

1.4 Contributos

Desenvolvimento do servidor *Nodejs* onde foram adicionados novos módulos por nós desenvolvidos e interligados com módulos públicos de terceiros:

- Criação do módulo que com o auxílio de outros módulos públicos de terceiros é capaz de monitorizar e interagir com o canal HDMI, dotando o sistema da capacidade da utilização do comando remoto que acompanha a televisão para navegar na interface gráfica que é apresentada na televisão.
- Criação do módulo que consegue receber a indicação do equipamento *Bluetooth* em utilização e utilizando o módulo público de terceiros consegue interagir com o equipamento colocando-o em execução e recolhendo os dados por ele produzidos.
- Desenvolvimento do módulo para a comunicação em tempo real, que dota o sistema da utilização do comando remoto da televisão para navegar na interface disponibilizada pelo servidor.
- Desenvolvimento do módulo utilizado na interface gráfica para navegar entre as várias opções disponíveis, de acordo com as indicações provenientes do comando da televisão.
- Criação dos modos de *Prod* e *Dev* e do servidor e de *seeds* para auxiliar no desenvolvimento e *debugging* das várias partes da VITABOX.
- Desenvolvimento da interface gráfica, nomeadamente as seguintes secções:
 - utilização dos equipamentos biométricos,
 - histórico dos sensores ambientais e biométricos integrando bibliotecas de gráficos para auxílio na visualização dos dados,
 - criação do painel de configuração.
- Transformação do logotipo fornecido em animação.
- Co-autor na publicação de *paper* VITASENIOR-MT: *a telehealth solution for the elderly focused on the interaction with TV*, apresentado na conferência 20th IEEE International Conference on e-Health Networking, Application & Services (17th – 20th September 2018).

1.5 Organização do relatório

Para além deste capítulo, o relatório tem o capítulo 2 destinado ao estado da arte, onde serão descritos alguns projetos que se assemelham ou concorrentes ao apresentado, e que nos auxiliará na abordagem ao tema, analisando as dificuldades que tiveram bem como a orientação do caminho a seguir para o nosso desenvolvimento. No Capítulo 3 descrevem-se as tecnologias consideradas e as conclusões para a tomada de decisão de qual utilizar.

O Capítulo 4 descreve o trabalho desenvolvido e sua implementação e por fim o capítulo 5 apresenta os testes à solução apresentada, testes de usabilidade e de desempenho. O Capítulo 6 apresenta as conclusões retiradas da execução de todo este projeto.

2 Estado da arte

Os sistemas de tele-saúde são de grande pertinência dado o contexto demográfico de envelhecimento da população e do aumento da esperança de vida, que se regista na Europa. Neste capítulo iremos apresentar alguns sistemas similares, ilustrando os seus principais desenvolvimentos. Existem vários tipos de sistemas que pretendem ajudar os idosos a alcançar uma melhor qualidade de vida, alguns que disponibilizam os dados recolhidos através de dispositivos móveis, smartphones ou tablets, e outros através de computadores pessoais utilizando uma página de internet. Alguns precisam de maior interação do utente do que outros. Nesta secção, pretendemos dar a conhecer o que já se encontra desenvolvido e perceber como é que estes sistemas foram desenvolvidos, e em que áreas podemos melhorar ou fazer diferente.

2.1 Sistemas de Tele-saúde

2.1.1 *Nourish Sensors*

Nourish Sensors - Nourish Care Systems | Bournemouth, United Kingdom [6]. É um sistema de geração de alarmes inteligentes para melhoria da qualidade da vida através de uma rede de sensores para monitorização de idosos. Faz a telemonitorização do utilizador, armazenando dados da permanência numa divisão, o tempo em que o utente se encontra na cama ou sofá, tendo também a possibilidade da geração de alarmes.

2.1.1.1 Visão geral do sistema

Este projecto focou-se no desenvolvimento de uma rede de sensores responsável pela recolha de dados e posterior envio através de uma rede 3G para um servidor remoto central para armazenamento (ver Figura 1). Esta rede de sensores monitoriza dados vitais do utente, como por exemplo quedas, ausência da habitação, ausência de atividade, etc., de forma não intrusiva para o utilizador.

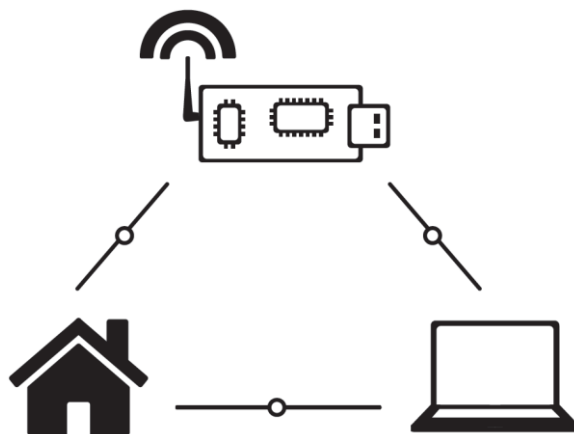


Figura 1 - Esquema geral do sistema - Nourish Sensors⁴.

O sistema central está dotado de uma interface gráfica (ver Figura 2) para visualização dos dados recolhidos na habitação do utente tendo a possibilidade de enviar alertas para a casa do utente. Esta interface gráfica é acessível de qualquer um dispositivo com um browser e acesso à internet.

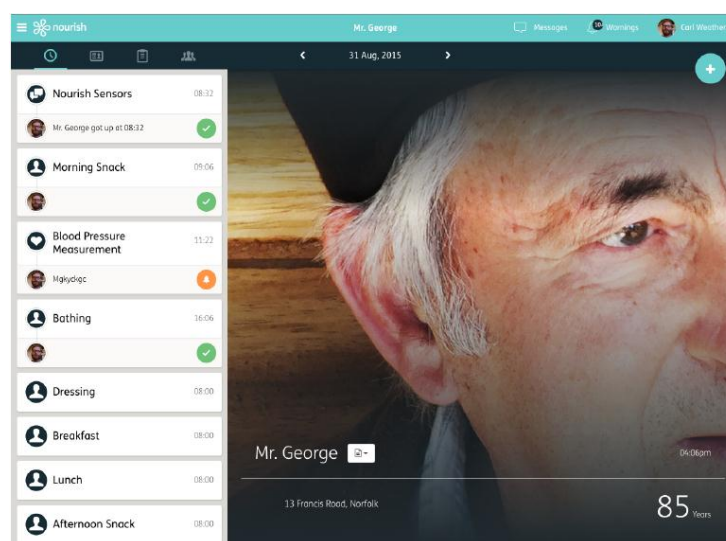


Figura 2 - Exemplo da interface gráfica do sistema⁵.

2.1.1.2 Módulo de interface entre a rede de sensores e o sistema central

Este módulo é constituído por um microcomputador capaz de correr um sistema operativo Linux, com acesso aos pinos digitais de leitura e escrita, e com uma interface USB utilizada para acoplar uma *pen* 3G utilizada para o envio dos dados para o sistema central.

⁴ Imagem retirada de <https://core.ac.uk/download/pdf/43584260.pdf>

⁵ Imagem retirada de <https://core.ac.uk/download/pdf/43584260.pdf>

Este equipamento tem também a responsabilidade do armazenamento dos dados provenientes dos sensores até que sejam enviados com sucesso para o sistema remoto.

2.1.1.2.1 Hardware

Este sistema tem como por base de *hardware* um microcontrolador responsável pela interligação entre os sensores e as redes *Ethernet* / *Wi-Fi* e ou 3G.

Os sensores serão ligados ao microcontrolador utilizando os pinos GPIO (*General Purpose Input Output*), com a utilização destes pinos é possível interagir diretamente com os sensores utilizando o protocolo de comunicação série.

Para a recolha dos dados recebidos pelos pinos GPIO foi considerado como *hardware* o *Raspberry Pi* e equipamentos semelhantes.

2.1.1.2.2 Software

O *software* utilizado neste sistema foi escolhido tendo em conta o *hardware* escolhido, visto que o mesmo é de baixos recursos e as linguagens de programação terão que estar de acordo com o *hardware*. O sistema operativo é uma distribuição *Linux*. O *firmware* que irá interagir com os pinos que comunicam com os sensores e com a comunicação dos dados foi desenvolvido em *C++* e *Python*.

2.1.2 eVida

A plataforma *eVida* foi testado em 11 utentes da Caritas em Coimbra conforme citado no Diário de Notícias [7]. Foi concebida no âmbito do projeto "*TICE.Healthy* - Saúde e Qualidade de Vida" [8]. Esta plataforma conta com um *kit* capaz de monitorizar dados vitais dos utentes:

- Peso,
- Pressão arterial,
- Glicemia,
- Ritmo cardíaco e
- Oximetria.

Esta plataforma é constituída por uma aplicação informática e um *kit* de sensores que de uma forma fácil e simples recolhe os diversos dados do utilizador, a qualquer momento. Os dados podem ser consultados por diversas entidades: o idoso, familiares e os profissionais.

2.1.3 MySignals

O sistema *MySignals* [9] é destinado a efetuar exames e pequenos diagnósticos no domicílio. Esta plataforma é composta por inúmeros equipamentos destinados à realização de alguns exames e diagnósticos (ver *Figura 3*).

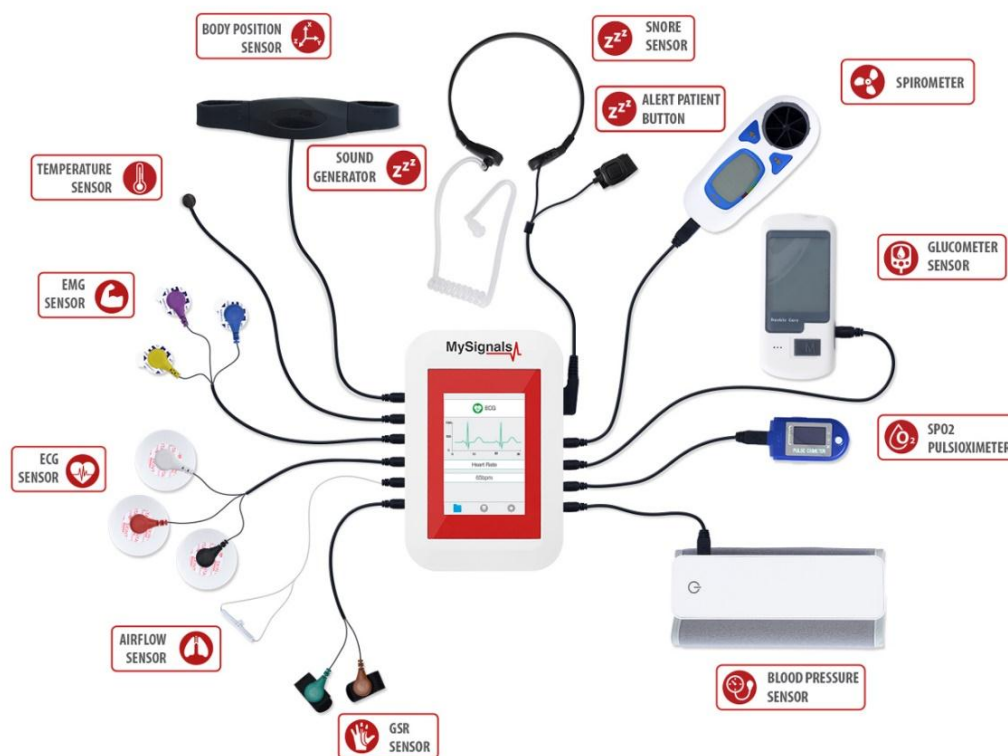


Figura 3 - Esquema dos vários sensores do sistema *MySignal*⁶.

Este equipamento também está dotado de uma interface de fácil usabilidade bem com uma interface *Web* que disponibiliza uma aplicação destinadas aos dispositivos moveis (*Android*, *IPhone*) (ver *Figura 4*).

Toda a informação obtidas pelos sensores é enviada para a *Cloud*, a informação transmitida é encriptada.

Este sistema requer algum conhecimento para o manuseamento do mesmo pois é composto por sistemas com algum rigor técnico e especificidade na sua utilização.

⁶ Imagem retirado do endereço (<http://www.my-signals.com/>)

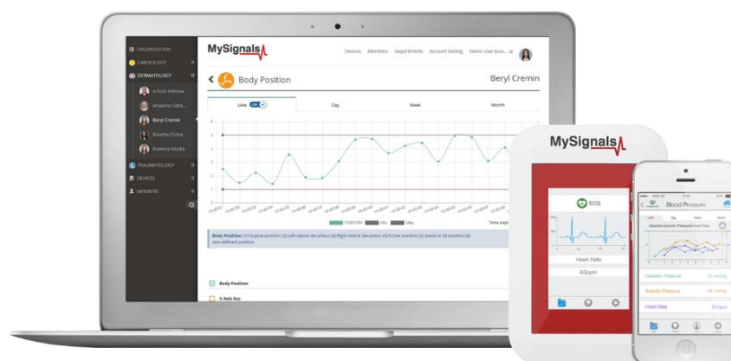


Figura 4 - Interface do sistema MySignal para as diferentes plataformas⁷.

2.1.4 10 Mil Vidas

O projeto 10 Mil Vidas [10] está focado na teleassistência e tele-saúde orientado para a população sénior, pretendendo fornecer os seguintes serviços:

- Segurança;
- Localização;
- Lembretes e gestão de medicamentos;
- Controlo de indicadores de Saúde;
- Aproximação da família.

Este sistema auxilia os idosos, estando dotado de um botão que o utente pressiona em caso de emergência, despoletando um alerta para as entidades previamente configuradas, familiares, instituições, vizinhos, que recebem a notificação de alerta. Um sistema de localização permite identificar a localização do idoso.

O sistema possui também um alerta que auxilia o utente na toma da medicação, e um sistema de gestão de *stocks*.

Neste projeto está contemplado um sistema de *call center* que funcionará 24h por 7 dias durante todo o ano.

Com o desenvolvimento desta plataforma que contem indicadores de análise onde se pode identificar o nível de satisfação e de utilização, este projeto prevê abranger 20 municípios.

⁷ Imagem retirado do endereço (<http://www.my-signals.com/>)

2.1.5 *SMARTAL – SMART ASSISTED LIVING*

O *SmartAL* [11] é um ecossistema tecnológico que pretende ajudar as pessoas do ponto de vista da saúde, social e de segurança (pessoal e doméstico), que se encontra ainda em fase de desenvolvimento.

O sistema tem a capacidade de integração de diversas tecnologias com gestão centralizada, conseguindo monitorizar o espaço envolvente bem como o(s) utente(s).

Estão disponíveis diversos serviços capazes de auxiliar o utilizador, o familiar ou cuidador, nomeadamente:

- Monitorização de sinais vitais,
- Alertas, notificações e gestão de lembretes,
- Gestão de atividades diárias,
- Pesquisas para avaliação de suporte social e de saúde,
- Vídeo vigilância,
- Automação residencial,
- Relatórios e indicadores estatísticos,
- Conteúdos educativos,
- Segurança doméstica.

A *SmartAL* utiliza diferentes tecnologias. Em casa é utilizada a televisão (ver *Figura 5*) como meio de interação e visualização dos dados recolhidos pelos sensores de saúde, permitindo a receção de alertas, avisos e também o controlo da automação residencial (domótica). Para além da televisão o sistema também possui uma aplicação *Android* e *IOS* (ver *Figura 5*) que podem ser utilizados para aquisição de dados que posteriormente serão enviados para o sistema central.

O *SmartAL* possui uma plataforma onde é possível configurar alertas, valores limite para despoletar eventos de aviso para as diferentes aplicações como para as pessoas indicadas. O cuidador tem ao seu dispor uma aplicação *web* (ver *Figura 5*) onde lhe é possível aceder e visualizar os dados em tempo real do local monitorizado.



Figura 5 - Exemplo da aplicação da SmartAL⁸

O sistema possui suporte para monitorização de sinais vitais, peso, pressão sanguínea, pulso, glicose que se encontra disponível na televisão da rede MEO (Altice) e através da aplicação no smartphone *Android* ou *IOS* (ver Figura 6).

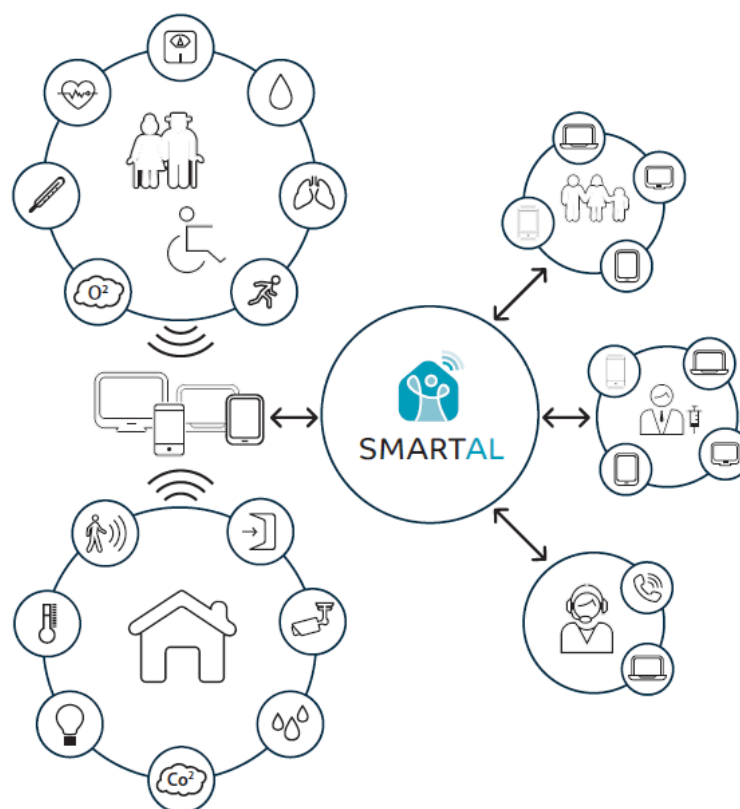


Figura 6 - Esquema da rede SmartAL⁹

⁸ Imagem retirada de http://www.alticelabs.com/content/products/BR_SMARTAL_ALB_EN.pdf

⁹ Imagem retirada de http://www.alticelabs.com/content/products/BR_SMARTAL_ALB_EN.pdf

2.1.6 Monitorização remota da saúde para idosos através da televisão interativa

Os criadores deste sistema de monitorização remota da saúde para idosos através da televisão interativa [12], tiveram a preocupação com o sistema eletrónico de saúde, no que diz respeito à manutenção dos dados e base de dados, segurança e acessos à informação, bem como sobre as implicações sociais de registar e comunicar as informações de saúde. Focaram-se também no design da interface, ajustando-o para ser utilizado por pessoas com baixos conhecimentos na área das tecnologias.

Deram especial atenção no utilizador final, o idoso, facilitando o acesso ao sistema bem como à informação que é disponibilizada, tendo a preocupação de desenvolver um sistema que pudesse ser administrado de forma autónoma pelos pacientes sem a necessidade de um técnico no local.

2.1.6.1 Arquitetura do sistema e fluxos de dados

A rede de dispositivos médicos (ver Figura 7) pode ser interligada por intermédio de uma rede *Wi-Fi*. Os dados recolhidos pela STB (*Set-Top-Box*) são comparados com os valores pré-configurados para verificar a necessidade da ativação dos alertas para o centro de assistência médica. Este alerta é enviado pela STB, uma vez que a quantidade de dados a enviar é limitada. Note-se que este serviço não está implementado em todos os países, como é o caso de Portugal.

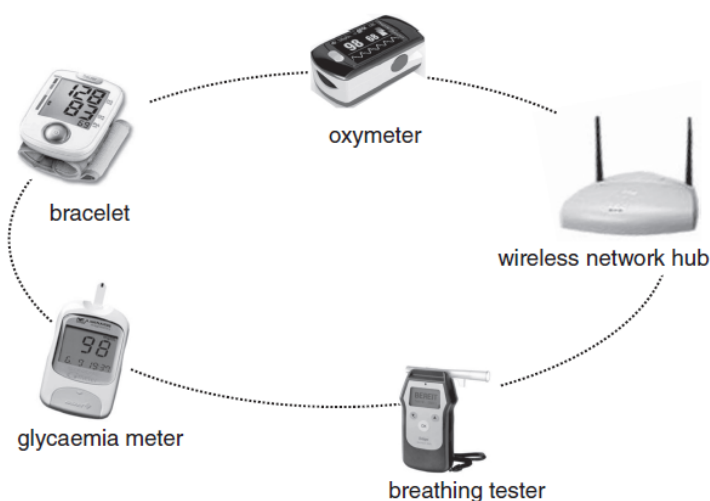


Figura 7 -Rede de sensores biomédicos utilizados para monitorizar remotamente o Utente¹⁰

¹⁰ Imagem retirada de <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-11-54>

Este sistema encontra-se preparado para vigiar o utente dentro da habitação, mas não está preparado para funcionar em situações de emergência, pois o envio de mensagens do centro de assistência ao utilizador não é possível.

A estrutura está centralizada e focada na privacidade do cliente. O sistema é constituído pela STB do cliente ligada por um canal DVB-T (*Digital Video Broadcasting - Terrestrial*), ao centro de saúde, onde se encontra um servidor de autenticação centralizado para proteção dos dados dos clientes e um servidor de armazenamento dos dados do utente. Existe um funcionário responsável que utiliza aplicações adequadas onde é possível realizar várias ações que ajudam a fornecer diferentes níveis de assistência médica ao utente (ver *Figura 8*).

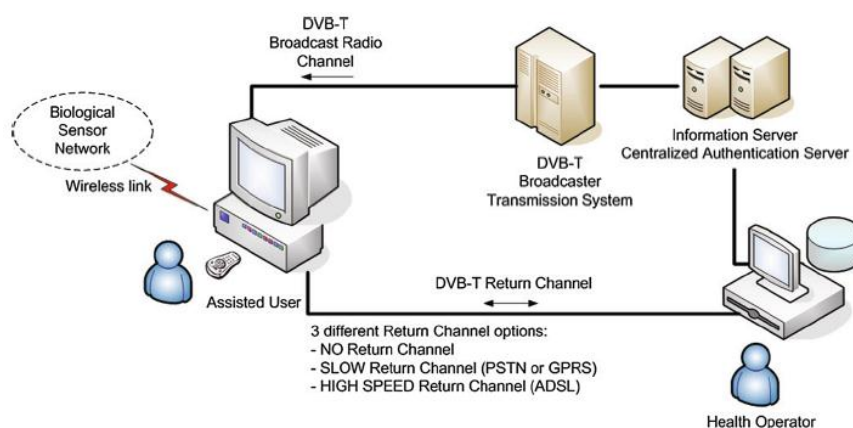


Figura 8 - Arquitetura do sistema de monitorização remota¹¹

Os intervenientes do sistema são o utente que pode interagir com o operador de saúde que se encontra noutro local, esta interação é efetuada com a utilização da STB que está em casa do utente. O utilizador possui um cartão pessoal que inere na STB, e a autentica no sistema central, enquanto o operador remoto pode monitorizar cada paciente e tem ao dispor a possibilidade de enviar mensagens de vários tipos, texto ou multimédia. O operador de radiodifusão é responsável pela entrega do áudio, vídeo e dados no canal de rádio, utilizando um fluxo de transporte MPEG-2 corretamente formatado. A infraestrutura do centro de saúde possui as bases de dados que armazenam os registos de saúde de cada paciente. A utilização do cartão pessoal que autentica o utilizador na STB facilita a autenticação do paciente,

¹¹ Imagem retirada de <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-11-54>

deixando de ser necessário a introdução de qualquer informação, reduzindo a probabilidade de erro.

2.1.6.2 Layout do aplicativo

O aplicativo foi concebido de forma a facilitar a interação com o paciente usando uma interface simples (ver Figura 9 e Figura 10).



Figura 9 - Exemplo de autenticação na STB¹²

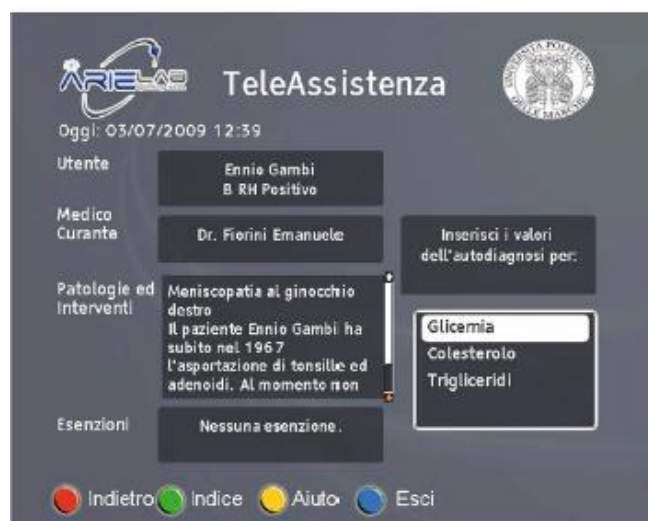


Figura 10 - Exemplo da utilização da Teleassistência¹³

¹² Imagem retirada de <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-11-54>

¹³ Imagem retirada de <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-11-54>

Os dados recolhidos pelos sensores na rede são formatados de acordo com a linguagem XML, que é utilizado por todas as entidades envolvidas no sistema, (o servidor de Informações, a aplicação e as instalações do centro de saúde), que podem ser trocadas através do canal de rádio DVB-T.

2.1.6.3 Resultados

A utilização de cartões para a identificação do utilizador foi testada, concluindo-se que ajudou na autenticação dos utilizadores que não se encontram tão familiarizados com o controlo remoto. Foram realizados testes com STB diferentes (ver *Tabela 1*) compatíveis com o sistema desenvolvido medindo os tempos de autenticação (ver *Tabela 2*). A diferença de tempos verificada entre as várias STB, segundo os autores, não causou qualquer transtorno para o utilizador.

Feature	STB1	STB2	STB3
Chip	STi5100	IBM PowerPC 405	STi7100
Amount of Memory	32 MB	64 MB	128 MB
Java Stack	Osmosys	Jet (open source)	Osmosys

Tabela 1 - Hardware utilizado nas STB¹⁴

	STB1	STB2	STB3
Get Data Holder	20 s	3 s	1 s
PIN Verification	3.5 s	1 s	1 s
Get Certificate	25 s	7 s	3 s

Tabela 2 - Tempos gastos na autenticação¹⁵

No decorrer dos testes foram efetuados testes de robustez e de possíveis falhas. Um dos testes passou por ativar todos os botões do comando remoto de forma a analisar se o sistema responde corretamente aos botões que foram definidos, o que se verificou positivos, foi testado também o envio de comandos em situações onde os mesmo não eram esperados, a aplicação não mostrou qualquer problema em lidar com essas situações. Em caso de falha no *link* de rede, o cabo de *Ethernet* foi desligado, surgindo uma mensagem que indica que

¹⁴ Imagem retirada de <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-11-54>

¹⁵ Imagem retirada de <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-11-54>

existe um problema na ligação onde o sistema não responde sem que o *link* seja restabelecido.

2.2 Comparação entre os vários sistemas

Devido à existência de diversos sistemas similares ao que nos propomos desenvolver a tabela (ver Tabela 3) ilustra as diversas diferenças entre as diferentes tecnologias.

Sistema	<i>Nourish Sensors</i>	<i>eVida</i>	<i>MySignals</i>	10 Mil Vidas	<i>SmartAL</i>	Televisão Interativa	VITASENIOR
Sensores Biométricos¹⁶	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Sensores Ambientais¹⁷	Não	Não	Não	Não	Sim	Não	Sim
Restrito ao Operador	Não	Não	Não	Não	Sim	Não	Não
Interação com a TV	Não	Não	Não	Não	Sim	Sim	Sim

Tabela 3 - Tabela comparativa entre sistemas

Como se verifica na tabela (ver Tabela 3) o sistema *SmartAL* é muito idêntico ao que nos propomos desenvolver mas de momento ainda não se encontra disponível nos equipamentos do operador. Apresentamos como principal vantagem do nosso sistema não estarmos limitados a nenhum operador, sendo o nosso sistema de fácil integração e independente do operador que o utente tenha contratado.

O produto *MySignals* tem como objetivo de negócio o armazenamento na *cloud* que se encontra disponibilizado e configurado no próprio equipamento.

O projeto da televisão interativa foi desenvolvido em Itália e encontra-se em funcionamento, os restantes projetos foram realizados no âmbito de prova de conceito na utilização de tecnologias capazes de monitorizar o utente e o espaço envolvente do mesmo, auxiliando a pessoa cuidadora ou responsável.

¹⁶ Exemplos; peso, pressão arterial, pulso, etc.

¹⁷ Exemplo: CO₂, monóxido de carbono, gás, temperatura ambiente, etc.

3 Tecnologias Utilizadas

3.1 HDMI-CEC

O HDMI-CEC é um recurso presente em todos os *transceivers* HDMI, em cabos a partir da versão 1.0. Esta tecnologia permite, por exemplo, com o simples comando da televisão comandar outro sistema ou dispositivo a ela ligado, Por exemplo, através do controlo remoto da televisão comandar a *playstation*, a *tvbox Android*, entre outros. Esta propriedade não obriga a que os sistemas interligados sejam do mesmo fabricante, mas apenas que ambos os aparelhos suportem a propriedade CEC.

De acordo com a especificação do HDMI [13] a característica CEC [14] encontra-se disponível no pin 13 (ver Figura 11).

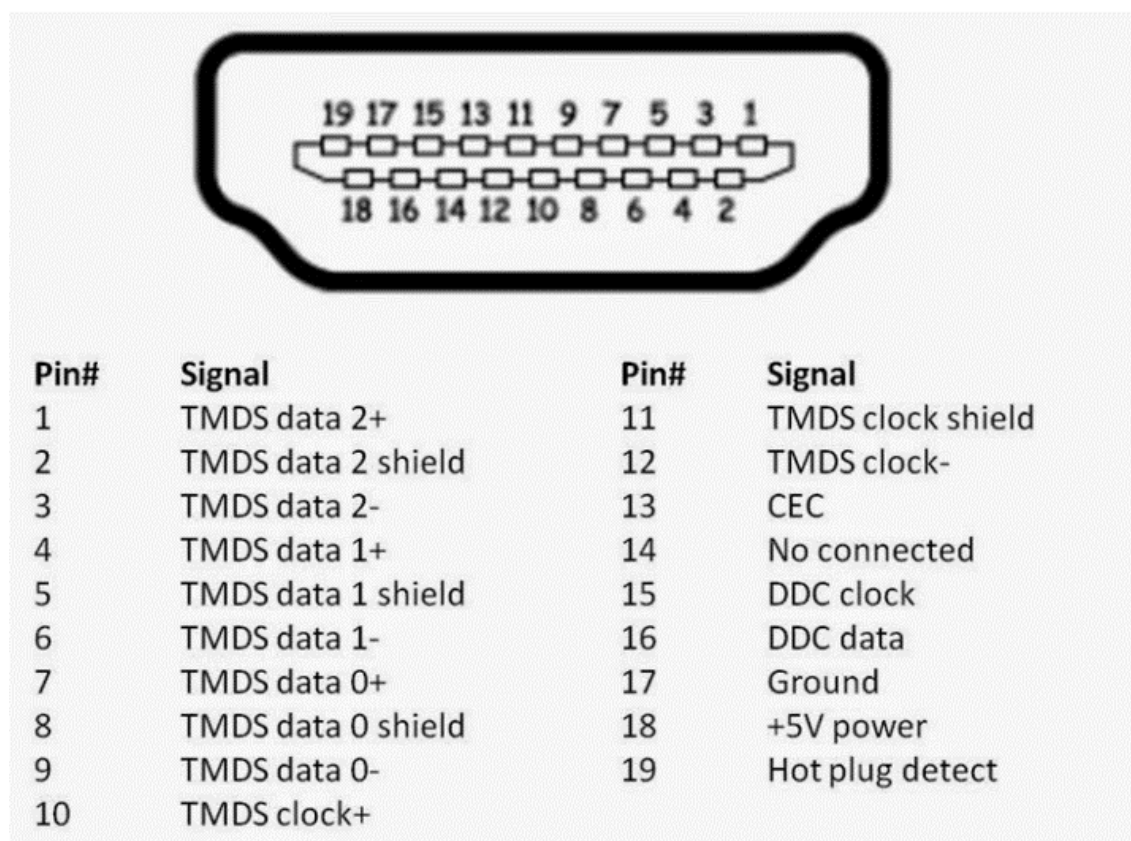


Figura 11 - Descrição dos pins da ligação HDMI¹⁸

¹⁸ Imagem retirado de <http://articles.triplewidemedia.com/choosing-right-video-cable-hdmi/>

De acordo com o fabricante, o CEC pode tomar outra designação conforme se verifica na Tabela 4.

Fabricante	Designação
<i>Samsung</i>	<i>Anynet</i>
<i>Sharp</i>	<i>Aquos Link</i>
<i>Sony</i>	<i>BRAVIA Link and BRAVIA Sync</i>
<i>Hitachi</i>	<i>HDMI-CEC</i>
<i>AOC</i>	<i>E-link</i>
<i>Pioneer</i>	<i>Kuro Link</i>
<i>Toshiba</i>	<i>CE-Link and Regza Link</i>
<i>Onkyo</i>	<i>RIHD (Remote Interactive over HDMI)</i>
<i>Runco International</i>	<i>RuncoLink</i>
<i>LG</i>	<i>SimpLink</i>
<i>ITT</i>	<i>T-Link</i>
<i>Panasonic</i>	<i>HDAVI Control, EZ-Sync, VIERA Link</i>
<i>Philips</i>	<i>EasyLink</i>
<i>Mitsubishi</i>	<i>NetCommand for HDMI</i>

Tabela 4- Designação da tecnologia CEC por fabricante

É possível controlar até 15 equipamentos interligados [14], sendo atribuído um endereço físico e lógico único a cada equipamento como mostrado na Figura 12.

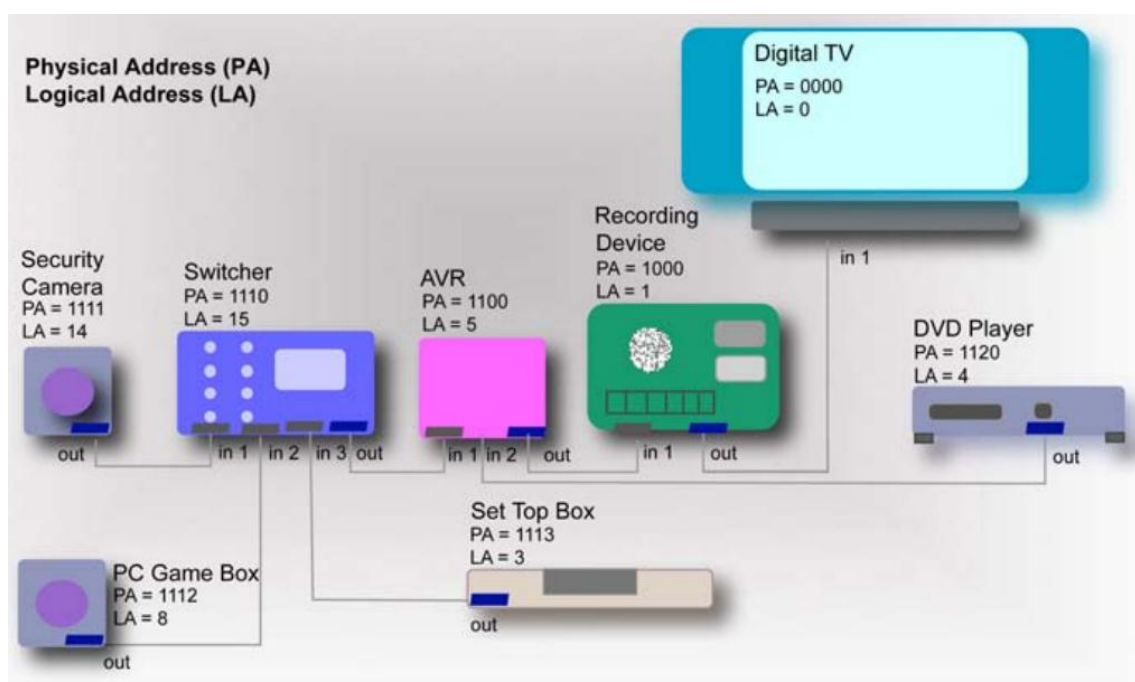


Figura 12 - Endereçamento CEC¹⁹ (PA – Physical address; LA – Logical address).

¹⁹ Imagem retirado de https://linuxtv.org/downloads/presentations/media_summit_2016_san_diego/cec-wide_1.pdf

O protocolo CEC [14] encontra-se definido pela estrutura representada na Figura 13. As mensagens estão em binário, sendo normalmente os códigos apresentados em hexadecimal. O *source* e o *destination address* possuem um domínio que varia entre “0” a “15”.

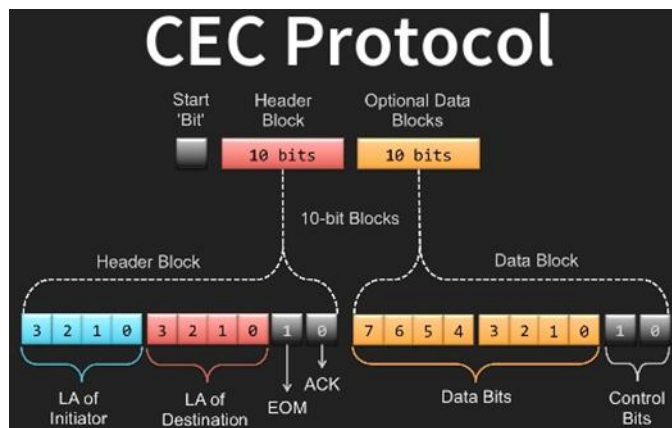


Figura 13 - Estrutura do protocolo CEC²⁰

Na página web *cec-o-matic*²¹ existe uma lista dos códigos disponíveis para utilização na comunicação CEC. Nos vários testes que foram efetuados verificámos que nem todos se encontram disponíveis, variando essa disponibilidade de fabricante para fabricante.

3.2 Unidade central de processamento

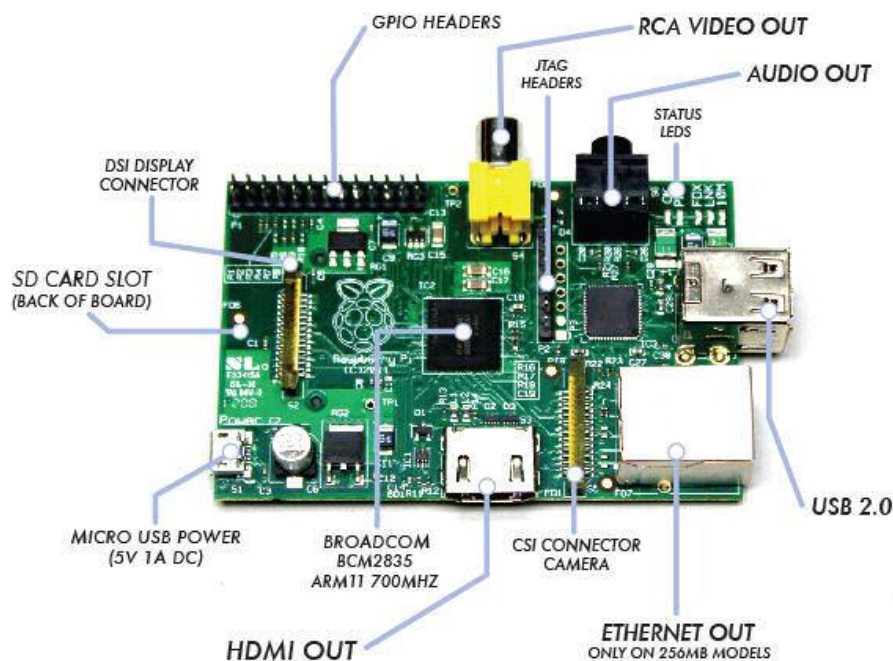
A VITABOX necessita de um equipamento que esteja dotado com uma ligação HDMI, algum poder computacional, ligação à Internet via Ethernet e ou *Wi-Fi*, e ligação *Bluetooth* de preferência todos incorporados; dimensões reduzidas e baixo custo. Foram considerados os microcomputadores apresentados nos pontos seguintes.

3.2.1 Raspberry Pi 2

Raspberry Pi 2 é um microcomputador com o tamanho aproximado de um cartão de crédito, que permite a ligação a um monitor ou TV, a utilização de periféricos como teclado ou rato. Todo o *hardware* é integrado numa única placa (ver Figura 14) com as características apresentadas na Tabela 5. O principal objetivo do seu aparecimento foi o de promover o ensino na informática e programação básica em escolas.

²⁰ Imagem retirado de <https://www.slideshare.net/AlesJohn/high-definition-fuzzing-exploring-hdmi-vulnerabilities>

²¹ Disponível em (<http://www.cec-o-matic.com/>).

Figura 14 - Board Raspberry Pi 2 Model B²²

Características	
Chip	Broadcom BCM2835 SoC full HD multimedia applications processor
CPU (Central Process Unit)	700 MHz Low Power ARM1176JZ-F Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor
Memory	512MB SDRAM
Ethernet	Onboard 10/100 Ethernet RJ45 jack
USB 2.0	Dual USB Connector
Video Output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audio Output	3.5mm jack, HDMI
Onboard Storage	SD, MMC, SDIO card slot
Dimensions	8.6cm x 5.4cm x 1.7cm

Tabela 5 - Características do Raspberry Pi 2 Model B²³

²² Imagem retirada de http://www.siongboon.com/projects/2013-07-08_raspberry_pi/

²³ Informação retirada de <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

3.2.2 Raspberry Pi 3

O *Raspberry Pi 3* (ver Figura 15) é uma evolução do *Raspberry Pi 2*, possuindo maior capacidade de processamento e muitas melhorias a nível de *hardware*, conforme indicado nas especificações do equipamento da Tabela 6.

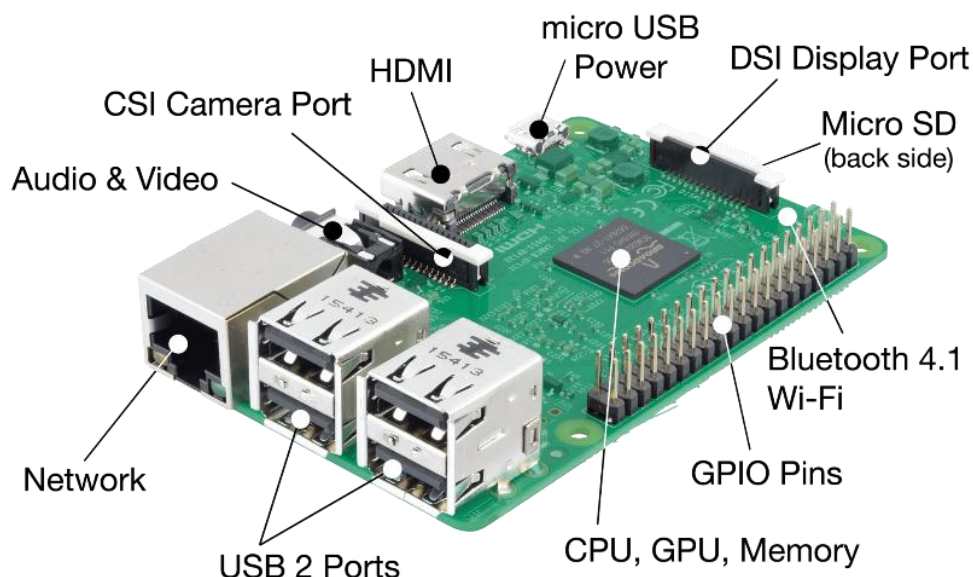


Figura 15 - Board Raspberry Pi 3²⁴

Características	
SoC	Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
GPU	Broadcom Videocore-IV
RAM	1GB LPDDR2 SDRAM
Networking	Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
Bluetooth	Bluetooth 4.2, Bluetooth Low Energy (BLE)
Storage	Micro-SD
GPIO	40-pin GPIO header, populated
Ports	HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, CSI (Camera Serial Interface), DSI (Display Serial Interface)
Dimensions	82mm x 56mm x 19.5mm, 50g

Tabela 6 - Características do Raspberry Pi 3 Model B²⁵

²⁴ Imagem retirado de <https://lectronichub.com/images/products/106-raspberry-pi-3-b-2.png>

²⁵ Informação retirada de <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

3.2.3 Cubie Board 2

O *Cubie board 2* possui dimensões e características semelhantes à *Raspberry Pi 2* e 3 referidos anteriormente (ver Figura 16 e Tabela 7). No entanto, para além de estar munido de um *slot* para um cartão SD também possui memória interna.

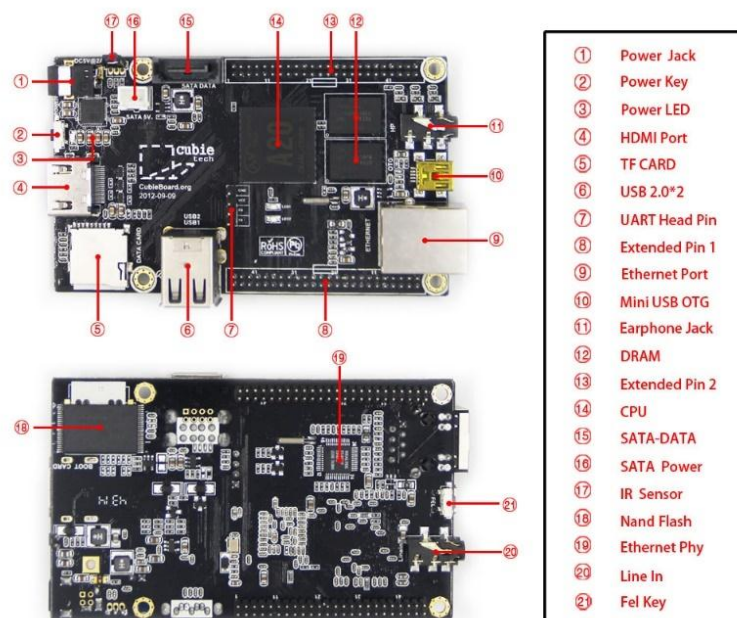


Figura 16 - Board Cubieboard2²⁶

Características	
CPU	ARM® Cortex™-A7 Dual-Core
GPU	ARM® Mali400MP2, Complies with OpenGL ES 2.0/1.1
Memory	1GB DDR3 @960M
Storage	4GB internal NAND flash, up to 64GB on uSD slot, up to 2T on 2.5 SATA disk
Power	5VDC input 2A or USB otg input
Networking	10/100 ethernet, optional Wi-Fi
USB	Two USB 2.0 HOST, one USB 2.0 OTG
Other	One IR

Tabela 7 - Características do CubieBoard 2²⁷

²⁶ Imagem retirada de <http://www.cubietech.com/product-detail/cubieboard2-dualcard/>

²⁷ Informação retirada de <http://cubieboard.org/2013/06/19/cubieboard2-is-here/>

3.2.4 Comparação e Escolha

De acordo com a tabela comparativa resumida dos microcomputadores (ver Tabela 8) considerados para alojamento da plataforma VITABOX, o Raspberry Pi 3 foi o escolhido por possuir *Bluetooth* integrado e possuir a capacidade de processamento necessária. Foi instalado o sistema operativo *Raspbian*, uma versão de 32 *bits* embora o equipamento seja de 64 *bits*, devido ao equipamento apenas possuir 1 Gb de memória RAM (*Random Access Memory*).

No início da realização do projecto foram efetuados alguns testes com o Raspberry Pi 2 o que se revelou com poucos recursos para suportar todo o processamento necessário para cumprir com os objetivos do projecto. Foi, no entanto, utilizado inicialmente na realização de testes isolados, como por exemplo a utilização do HDMI-CEC.

Microcomputador	CPU	RAM	Bluetooth (Integrado)
Raspberry Pi 2	700 MHz Low Power ARM1176JZ-F	512 Mb	Não
Raspberry Pi 3	quad-core (ARMv8) 64-bit @ 1.4GHz	1 Gb	Sim
Cubie board 2	ARM® Cortex™-A7 Dual-Core	1 Gb	Não

Tabela 8 - Tabela comparativa resumida dos microcomputadores considerados

3.3 Persistências de dados

Para o armazenamento dos dados recolhidos pela VITABOX, a armazenar localmente, foram consideradas várias abordagens dentro das estruturas relacionais SQL (*Structured Query Language*) ou não relacionais NoSQL (*Not Only Structured Query Language*).

3.3.1 Modelo Relacional

Nesta abordagem os dados são armazenados em tabelas, dentro da tabela os dados são dispostos em colunas e nas colunas o tipo de dados, ex. *double*, *integer*, *string*, etc.

A cada adição de dados é criada uma nova linha na tabela, sendo neste tipo de abordagem necessário que previamente seja criada a estrutura de armazenamento dos dados. Caso não exista a definição e a criação de toda a estrutura da base de dados, o armazenamento não é possível.

Neste tipo de abordagem é possível criar relações entre as várias tabelas que compõem a estrutura de armazenamento dos dados.

Eis alguns tipos de bases de dados que se incluem no modelo relacional:

- *MySql*
- *SQLite*
- *SQL Server*
- *Oracle*

As bases de dados seguintes foram as consideradas para utilização neste projecto.

3.3.1.1 *MySQL*

MySQL é um sistema de gestão de base de dados relacional multiutilizador de código aberto, sendo dos mais utilizados para dar suporte à manipulação de dados das aplicações.

Esta base de dados normalmente é instalada em conjunto com outras aplicações para sistema *Linux*, sendo normalmente designado por LAMP (*Linux, Apache, MySQL e Perl/PHP/Python*) e em windows WAMP (*Windows, Apache, MySQL e Perl/PHP/Python*). Atualmente este sistema de base de dados pertence à Oracle que tornou a sua utilização mais restrita. .

Características mais relevantes:

- Possui mecanismos de segurança,
- Garante a integridade dos dados nas transações,
- Permite a associação de eventos a determinadas ações na base de dados (*Trigger*),
- *Stored Procedures* - Criação de mecanismos para execução de várias instruções na base de dados, sendo o processamento efetuado pelo servidor de base de dados libertando a aplicação.

Para a administração desta base de dados existem diversos programas, nomeadamente o *phpmyadmin*, aplicação que é instalada no servidor de base de dados que disponibiliza uma interface *web*.

3.3.1.2 *SQLite*

O *SQLite* é uma base de dados relacional, idêntico ao descrito no ponto anterior, mas como o próprio nome indica, *light*, possui algumas limitações, como por exemplo a nível do acesso que apenas é efetuado por um único utilizador, e o armazenamento é efetuado num ficheiro local dentro do projeto. Como é para utilização de um único utilizador no caso de acesso

com vários utilizadores o servidor terá que realizar a gestão dos acessos aos dados sendo necessário atrasar os pedidos de forma a dar resposta a todas as solicitações.

A linguagem SQL é utilizada para a realização de *queries*, *insert*, *delete* e *update* dos registos. Para esta base de dados é recomendável a utilização de poucos dados por tabela pois encontra-se no mesmo servidor que a aplicação podendo comprometer o desempenho de todo o servidor.

3.3.2 Modelo Não Relacional

Esta abordagem normalmente designada por NoSQL (*Not only Structure Query Language*), foi desenvolvida com o intuito de criar aplicações e estruturas altamente escaláveis horizontalmente, facilitando a adição de novos atributos. Não é necessário que toda a estrutura esteja definida porque a qualquer momento podemos adicionar de uma forma fácil novos atributos, os quais se encontram ligados ao registo. Dentro deste modelo, encontramos o *MongoDB*, o *RethinkDB* descritas a seguir.

3.3.2.1 MongoDB

Segundo o artigo [15], o *MongoDb* é um SGBD (Sistema de Gestão de Base de Dados) do tipo NoSQL. É uma base de dados *Open Source* de alto desempenho, concebido para o armazenamento de grandes volumes de dados. Foi desenvolvida em C++ e trabalha com documentos do tipo JSON (*JavaScript Object Notation*).

Características principais:

- Alto desempenho
- Alta disponibilidade
- Altamente escalável
- Base de dados baseada em documentos JSON

3.3.2.2 RethinkDB

À semelhança da base de dados descrita anteriormente, a *Rethinkdb* é uma base de dados não relacional, NoSQL, orientada a documentos usando o formato JSON. Esta linguagem é muito intuitiva e voltada para aplicações *web*, onde são espoletados eventos após a alteração de registos na base de dados, dando a possibilidade da atualização de conteúdos em tempo real na aplicação.

Caraterísticas mais relevantes:

- Suporta *join* nas tabelas;
- *Dashboard web* integrado para administração das várias bases de dados;
- Suporte de *callback* em alterações ou atualizações das tabelas;
- Criação de índices secundários;
- Modelo de dados JSON e consistência imediata.

3.3.3 Comparação e Escolha

De acordo com alguns sites [16] [17] e artigos [18] [19] [20] onde são analisados e realizados alguns testes aos modelos reacionais e não relacionais sem que os autores tomem uma posição de defender este ou aquele modelo mas sim uma análise do contexto em que a base de dados será utilizada (necessidade do uso de transações), a utilização do modelo relacional é apontada como a melhor escolha. Para casos da necessidade de uma alta disponibilidade e de armazenamento de grande volume de dados, o modelo não relacional é apontado como a melhor escolha. Neste sentido foi decidido que a melhor abordagem para a aplicação seria a utilização de uma base de dados não relacional, pois o sistema gera uma grande quantidade de dados que deverão ser armazenados num curto espaço de tempo. Para além disso, existe uma aplicação para a apresentação dos dados recolhidos, que também gera dados. Tendo em conta todas estas situações, a escolha da base de dados recaiu na utilização do *MongoDB*, pois dentro das bases de dados não relacionais é uma das mais conhecidas tendo dado origem a uma enorme comunidade de suporte e desenvolvimento.

3.4 Servidor web

Para a disponibilização da interface gráfica, utilização do canal HDMI e interação com algum hardware presente no microcomputador, foram considerados alguns servidores dos mais conhecidos de código aberto. Os servidores considerados estão descritos nos pontos seguintes.

3.4.1 Apache

O servidor Apache é dos mais conhecidos e mais populares de código aberto, foi criado em 1995 por *Rob McCool*, funcionário da NCSA (*National Center for Supercomputing Applications*), normalmente utilizado em sistemas operativos *Linux*.

A maioria dos sites web do mundo utilizam este servidor para disponibilizar os seus conteúdos, efetuar compras online, etc.

Principais características:

- Suporte a scripts usando linguagens de programação *Perl*, *PHP*, *Shell Script*, *ASP*, etc.;
- Autenticação requerendo um nome de utilizador e palavra chave para acesso a uma determinada localização (suportando criptografia via *Crypto* e *MD5*);
- Exibição da página Web de acordo com o idioma requisitado pelo Cliente;
- Personalização de *logs*;
- Mensagens de erro;
- Suporte a *virtual hosting* (é possível servir duas ou mais páginas com endereços/portas diferentes através do mesmo processo ou usar mais de um processo para controlar mais de um endereço);
- Suporte a IP (*Internet Protocol*) *virtual hosting*;
- Suporte a *name virtual hosting*;
- Suporte a servidor *Proxy ftp* e *http*, com limite de acesso, *caching* (todas flexivelmente configuráveis);
- Suporte a *proxy* e redireccionamentos baseados em *URLs* (*Uniform Resource Locator*) para endereços Internos;
- Suporte a criptografia via *SSL* (*Secure Sockets Layer*), Certificados digitais;
- Módulos *DSO* (*Dynamic Shared Objects*) permitem adicionar/remover funcionalidades e recursos sem necessidade de recompilação do programa.

O Apache tentará fazer novos processos, novas *threads* para todas as solicitações em todas as ligações.

Utilizando o servidor *Apache* é necessário o desenvolvimento de toda a estrutura da aplicação:

- *Front-end* – Normalmente desenvolvido em *Javascript*, *CSS* e *HTML*,
- *Back-end* – Normalmente desenvolvido em *PHP* ou *Perl* ou *Python*,

No projeto VITABOX é necessário interagir com equipamentos *Bluetooth* durante a sua execução, para que esta interação seja possível é necessário o desenvolvimento de

bibliotecas noutra linguagem de programação que poderão não ser a mesma utilizada na aplicação, ex. PHP, *Perl*, *Python*, etc.

3.4.2 *NodeJs*

O Node.js²⁸ é uma aplicação que interpreta código *javascript* e que funciona do lado do servidor (*server-side*). O objetivo é ajudar os programadores na criação de aplicações de alta escalabilidade (como um servidor web), com códigos capazes de manipular dezenas de milhares de ligações simultâneas, numa única máquina física, sendo possível associar-lhe outros módulos que o complementam.

Características principais:

- Linguagem *Google Chrome Javascript V8*;
- Orientado a evento de I/O;
- *Threads* não bloqueantes;
- Lado do servidor Assíncrono.

O servidor *Nodejs* é utilizado o *Javascript* como linguagem de desenvolvimento das aplicações, é a mesma linguagem que é utilizada no desenvolvimento da interface da aplicação. Para a interação com o *hardware* que é necessário para o projeto VITABOX, apenas é necessário a inclusão de módulos capazes de interagir com o hardware utilizando a mesma linguagem de programação.

3.4.3 *Nginx*

O *Nginx* é um servidor HTTP de proxy reverso, gratuito com a capacidade de inúmeras funcionalidades.

Características:

- *Proxy* reverso;
- *Proxy* de servidor de email *SMTP* (*Simple Mail Transfer Protocol*), *POP3* (*Post Office Protocol 3*) e *IMAP* (*Internet Message Access Protocol*));
- *Proxy* de balanceamento de carga;
- Tolerância a falhas;

²⁸ Repositório disponível em (<https://nodejs.org/en/>).

- Suporte *SSL*;
- Suporte *STARTTLS*;
- Suporte *FastCGI*;
- Nome e IP baseados em servidores virtuais;
- *FLV streaming*;
- *MP4 streaming*, usando o módulo *MP4 streaming*;
- Controle de acesso e autenticação;
- Manipulação de arquivos estáticos;
- Manipulação de Cache do servidor e do *browser*.

À semelhança do servidor *Apache*, o servidor *Nginx* é responsável pela disponibilização da aplicação:

- *Front-end* – Normalmente desenvolvido em *Javascript*, *CSS* e *HTML*,
- *Back-end* – Normalmente desenvolvido em *PHP* ou *Perl* ou *Python*,

E à semelhança do *Apache* também é necessário o desenvolvimento de bibliotecas noutra linguagem de programação que poderá não ser a mesma utilizada no desenvolvimento da aplicação, ex. *PHP*, *Perl*, *Python*, etc., para a interação com *hardware* que o projeto VITABOX necessita (por exemplo, interação com os equipamentos *Bluetooth*).

3.4.4 Comparação e Escolha

Como se pode verificar na descrição de cada tecnologia considerada para a implementação de um servidor *Web*, verifica-se que todos possuem a capacidade de cumprir com os objetivos deste projeto, e tendo em conta os testes realizados e apresentados no site [21] e as considerações indicadas nos artigos [22] [23], a abordagem da utilização do *Nginx* em conjunto com o *Nodejs* seria uma boa aposta, mas devido ao *hardware* considerado a nossa escolha recaiu apenas na utilização do *Nodejs*, pois para além de utilizar a mesma linguagem de programação que é utilizada para desenvolver a interface gráfica, possui uma enorme comunidade de desenvolvimento de bibliotecas que facilmente são integradas no servidor para cumprir com os pressupostos do projeto.

Descrição de alguns módulos utilizados para expandir as funcionalidades do servidor:

3.4.4.1 *Noble*

O módulo *Noble*²⁹ é utilizado para interagir com os dispositivos *Bluetooth*. Este módulo consegue de uma forma fácil interagir com protocolo de comunicação dos equipamentos *Bluetooth* possibilitando desta forma a sua utilização e incorporação com todo o sistema disponibilizado pelo *Nodejs*.

Para conseguir interagir com o *hardware*, este módulo necessita da instalação de algumas bibliotecas e aplicações, nomeadamente:

- *bluetooth*,
- *blues*,
- *libbluetooth-dev*,
- *libudev-dev*

3.4.4.2 *Child-process*

O módulo chamado *Child-process*³⁰ fornece primitivas para a criação e destruição de processos, tem sempre três processos associados (*Child.stdin*, *child.stdout*, e *child.stderr*). Os fluxos de dados podem ser partilhados com o processo pai de forma a poderem comunicar entre si. Este módulo é utilizado para a criação de processos que irão de uma forma isolada interagir com os dispositivos *Bluetooth*, bem como com os equipamentos que recebem os dados recolhidos pelos sensores ambientais.

3.4.4.3 *Socket.io*

O *Socket.io*³¹ é um mecanismo capaz de abrir uma ligação entre o cliente e o servidor, possibilitando a troca de mensagens entre ambos sem a necessidade de executar um *refresh* na página, ou seja, sem a necessidade de efetuar um novo pedido ao servidor. Com isso é possível explorarmos o conceito de *real-time* na aplicação, onde tanto no servidor como no cliente se encontram mecanismos capazes de receber e enviar dados para atualização da informação a apresentar ao utilizador.

²⁹ Repositório disponível em <https://github.com/noble/noble>

³⁰ Documentação disponível em https://nodejs.org/api/child_process.html

³¹ Repositório disponível em <https://github.com/socketio/socket.io/>

Como não é garantida a compatibilidade com todos os *browsers*, a API (*Application Programming Interface*) de transporte tenta executar os diferentes pedidos pela seguinte ordem:

- *WebSockets*,
- *FlashSockets*,
- *AJAX long polling*,
- *AJAX multipart streaming*,
- *Forever Iframe*,
- *JSONP Polling*.

3.4.4.4 *Node-cec e cec-monitor*

Os módulos *Node-cec* e *Cec-Monitor* são destinados à interação com a interface HDMI, conseguindo monitorizar os eventos produzidos pelo canal HDMI, enviar comandos de ação ou consulta do estado atual do canal. Para o funcionamento destes módulos é necessário a instalação da biblioteca *cec-utils*.

O módulo *node-cec*³² possui uma maior transparência da informação apresentada nos eventos por cada tecla utilizada pelo comando da televisão, retornando de imediato a tecla pressionada, à qual e após consulta da base de dados é determinada a ação a ela atribuída.

O módulo *cec-monitor*³³ dispõe de uma constante monitorização de possíveis erros que ocorram bem como para o envio de eventos que provêm do lado do servidor com destino à televisão, como por exemplo a comutação de *source* (por exemplo, alternar entre o HDMI-1 para HDMI-2 ou vice-versa).

3.4.4.5 *Mongoose*

O módulo *mongoose*³⁴ faz a ligação entre o servidor e a base de dados. Este módulo dá a possibilidade de definir uma estrutura de armazenamento, tornando mais fácil a utilização da base de dados.

³² Repositório disponível em <https://github.com/patlux/node-cec>

³³ Repositório disponível em <https://github.com/senzil/cec-monitor>

³⁴ Repositório disponível em <https://github.com/Automattic/mongoose>

3.4.4.6 *Audioconcat*

O módulo *Audioconcat*³⁵ utiliza a aplicação *ffmpeg*³⁶ que foi instalada previamente no sistema operativo. Este módulo tem a capacidade de unir diversos ficheiros de áudio para a utilização na aplicação, conseguindo criar um único ficheiro possibilitando de uma forma fácil a sua reprodução.

3.5 Interface Gráfica

Para a apresentação dos dados recolhidos pelos sensores e para a utilização dos equipamentos biométricos foi necessário o desenvolvimento de uma interface gráfica, que pudesse ser usada de uma forma harmoniosa e intuitiva. Foram consideradas várias *frameworks* que nos auxiliassem no desenvolvimento. Nos pontos seguintes encontra-se uma descrição das plataformas consideradas.

3.5.1 *Backbone.js*

O *Backbone.js*³⁷ é uma *framework Javascript* que fornece componentes para melhorar a estrutura de aplicações *web*. Entre estes componentes encontram-se os *Models*, *Collections* e *Views*, com tratamento declarativo de eventos que liga tudo à sua API por meio de uma interface *RESTful JSON*.

Caraterísticas do *Backbone.js*

- Permite a programação atendendo ao paradigma *MVC* (*Model*, *View*, *Controller*);
- Ajuda na criação de estruturas bem definidas para os dados da aplicação;
- Simplifica a utilização das diferentes *views* que apresentam a informação ao utilizador;
- Permite a utilização de coleções de modelos;

Simplifica e ajuda a ordenar e a organizar o código da aplicação nas solicitações *Ajax* ao servidor.

³⁵ Repositório disponível em <https://github.com/h2non/audioconcat>

³⁶ Disponível em <https://www.ffmpeg.org/>

³⁷ Disponível em <http://backbonejs.org/>

3.5.2 *React.js*

Ferramenta desenvolvida para a criação de componentes, esta biblioteca surgiu em 2011 pelo *Facebook* que ajudou a impulsionar a *framework*. Em 2013 o código foi tornado publico originando uma maior divulgação e de adeptos.

O *React.js* é conhecido como uma biblioteca de *javascript* declarativa, para o desenvolvimento de interfaces do utilizador, onde tudo são componentes tornando-a flexível e eficiente [24].

3.5.3 *Vue.js*

O *Vuejs* foi apresentado em fevereiro de 2014 por *Evan*, que faz parte da equipa de desenvolvimento do projeto *vue.js* e que anteriormente trabalhava na *Google* no desenvolvimento de vários projetos utilizando *Angular.js*, tendo sido criado a partir do *Angular.js* dando origem a um sistema mais leve. Esta *framework* é utilizada para o desenvolvimento de páginas *web*, disponibilizadas pelo servidor. Esta tecnologia é idêntica à abordada no ponto anterior, ou seja, também é destinada à criação de interfaces por meio de componentes tendo nativamente a adição de *data bind*, *two way*, *events*, mas com uma linguagem de fácil interpretação. Com a utilização do *data bind* é possível, de uma forma fácil, atualizar uma variável que se encontra ligada a um elemento HTML, enquanto o *Vuejs* se encarrega de a atualizar sem recurso a um mecanismo externo.

3.5.4 Comparação e Escolha

A escolha da tecnologia a utilizar para o desenvolvimento da interface gráfica que será exibida na televisão do utilizador recaiu na utilização do *Vuejs*. Inicialmente foi considerada a utilização do *Backbone.js*, mas foi colocada de parte por se ter constatado que há indícios de abandono pelos desenvolvedores desta *framework*. Comparou-se também posteriormente o *Reactjs* e o *Vuejs* tendo-se verificado que eram muito idênticas, recaindo a nossa escolha no *Vuejs* devido ao seu código ser fácil de entender e leve.

3.5.4.1 *Template Vue Paper Dashboard*

O *template Vue Paper Dashboard* serviu de base para iniciar o desenvolvimento da interface gráfica. Este *template* encontra-se formatado e configurado, pronto a ser utilizado com diferentes componentes construídos pelos autores e prontos a serem utilizados ou modificados.

Este *template* encontra-se disponível em <https://github.com/creativetimofficial/vue-paper-dashboard>, e foi desenhado para que de uma forma automática, se auto ajustar às dimensões do equipamento em que está a ser mostrado, mantendo as proporções.

Conta com diversos exemplos de utilização dos componentes, e com toda a documentação necessária para a sua utilização.

3.5.4.2 JustGage

A biblioteca *justgage*³⁸ adicionada ao *template* serve para a criação de gráficos circulares, para mais facilmente apresentar valores recolhidos de um determinado sensor (ver Figura 17).



Figura 17 – Exemplo do gráfico utilizando a biblioteca Justgage

Posteriormente foi também adicionado a versão 2 da biblioteca *Justgage*³⁹ onde foi adicionada a possibilidade de configurar setores dando uma melhor interpretação dos valores a serem representados (ver Figura 18).

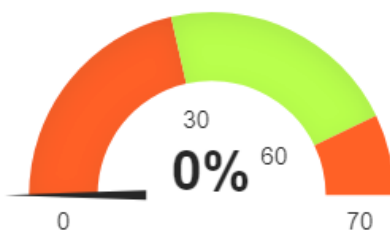


Figura 18 - Exemplo do gráfico utilizando a nova versão do Justgage

3.5.4.3 Canvas.js

A biblioteca *Canvas.js*⁴⁰ foi adicionada para apresentar os valores recolhidos pelos sensores ao longo do tempo de uma forma perceptível e atrativa.

³⁸ Repositório disponível em <https://github.com/toorshia/justgage>

³⁹ Repositório disponível em <https://github.com/letapjar/justgage/blob/master/jg2.js>

⁴⁰ Disponível em <https://www.chartjs.org/docs/latest/>

3.5.4.4 *EventBus*

O *EventBus* é o barramento de eventos disponibilizado pelo *Vue.js* que permite que elementos não relacionados na aplicação consigam comunicar, dando origem à exposição de eventos nas diferentes *views* para que seja possível manipulá-los, executando ações pré-estabelecidas.

A utilização deste módulo é simples porque basta importar a biblioteca e exportar a instância para as seções que pretendemos que comunique.

3.5.4.5 *Internacionalização - I18n*

Módulo *i18n*⁴¹ foi incorporado no *template*, o qual é responsável pela transição entre os vários idiomas disponíveis na interface gráfica.

3.5.4.6 *Fontawesome*

Bibliotecas adicionadas ao *template* escolhido para a disponibilização de uma vasta coleção de ícones para a representar determinados objetos, ações ou equipamentos.

⁴¹ Repositório disponível em <https://github.com/kazupon/vue-i18n>

- Recebe dados das VITABOXs (4) dos sensores ambientais⁴² e dos sensores biométricos⁴³.
2. Equipamentos que acedem ao *Backoffice*.
 3. Equipamentos que acedem ao *Frontend*.
 4. VITABOX sistema retratado no âmbito desta dissertação, este equipamento interage com diversos e diferentes equipamento:
 - Interage com a televisão (5),
 - Interage com os sensores ambientais (6),
 - Interage com os sensores biométricos (7),
 - Recebes dados de configuração do sistema remoto (1), envia os dados recebidos dos sensores ambientais e biométricos para o sistema remoto (1),
 5. Televisão do utilizador ligado por HDMI ao sistema VITABOX (4), disponibiliza uma interface para interação com o utilizador e recebe as ordens do utilizador para navegação na interface,
 6. Sensores ambientais⁴² estrategicamente colocados nas diferentes divisões da casa do utilizador,
 7. Sensores biométricos⁴³ e disponibilizados ao utilizador para a execução dos diferentes exames,
 8. Equipamento que separa a rede publica (internet) da rede privada (casa do utilizador).

4.2 Arquitetura da VITABOX

A arquitetura da VITABOX está retratada na figura (ver Figura 20), onde se encontra os diferentes módulos que foram desenvolvidos e a comunicação entre eles para que toda esta estrutura funcione.

⁴² Exemplo: CO₂, monóxido de carbono, gás, temperatura ambiente, etc.

⁴³ Exemplo: Medição da pressão arterial, temperatura corporal, glicémia, peso, etc.

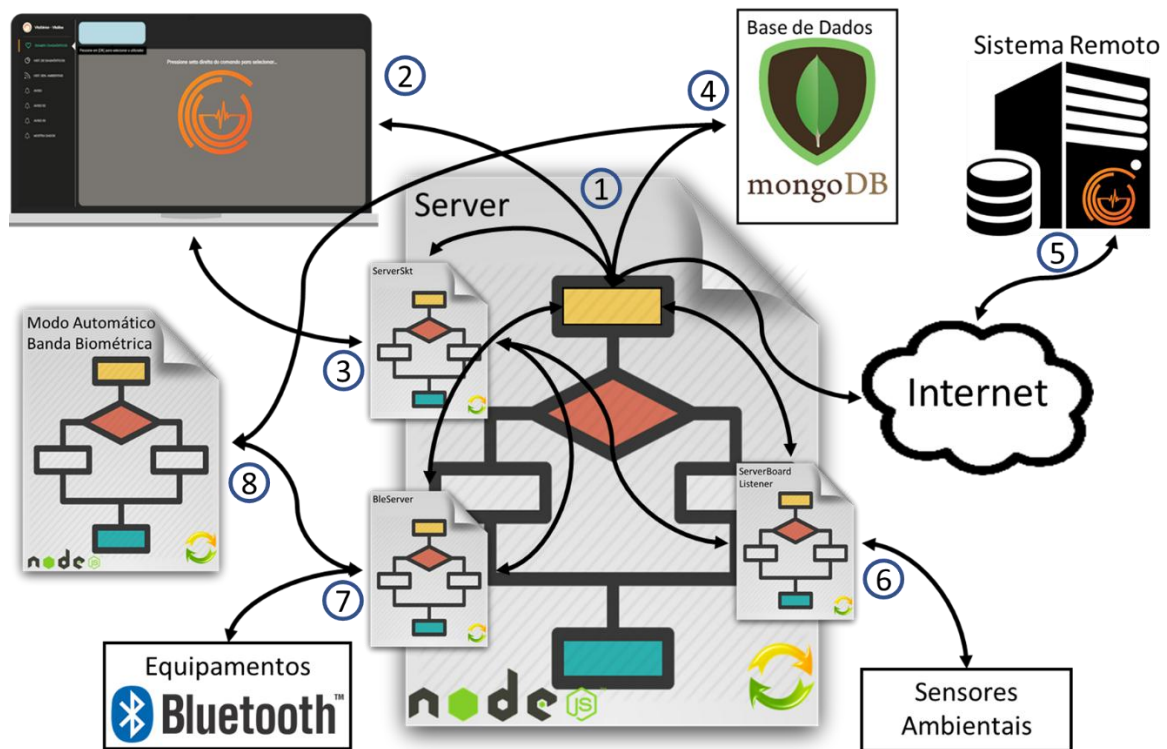


Figura 20 – Arquitetura geral da aplicação VITABOX.

Legenda da Figura 20:

1. Módulo principal que comunica com os restantes módulos,
 - Disponibiliza a interface gráfica (2),
 - Interage com o módulo de recolha dos dados dos sensores ambientais (6),
 - Interage com o módulo para a execução da utilização dos sensores biométricos (7),
 - Consulta e atualização de dados na base de dados (4).
2. Televisão do utilizador onde é disponibilizada a interface gráfica,
3. Módulo que monitoriza a interface HDMI recebendo os eventos provenientes do comando da televisão e enviando eventos para comutação de *source*.
4. Servidor de base de dados para armazenamento dos dados dos sensores biométricos e ambientais, informações do utilizador, ações configuradas referentes aos eventos das várias teclas do comando da televisão.
5. Sistema remoto para envio dos dados dos sensores ambientais e biométricos e receção das configurações dos utilizadores e dos vários sensores.

6. Módulo que dá origem a um novo processo para receção dos dados dos sensores ambientais,
7. Módulo que dá origem a um novo processo e que é criado para cada execução da utilização de um equipamento biométrico, quando a execução do equipamento biométrico o processo é destruído.
8. Módulo que dá origem a um novo processo a cada invocação para capturar dos dados da banda biométrica no modo automático, este módulo consulta a base de dados (4) da VITABOX para adquirir as informações do utilizador e da banda biométrica.

Esta aplicação dispõe de um servidor *nodejs* no qual se encontram diversos módulos que dão suporte à utilização de diferentes tecnologias, entre as quais o *Bluetooth*, a interface HDMI, etc.

4.3 HDMI: interação Televisão - VITABOX

Na interação com a porta HDMI, começou-se por testar os diferentes eventos / comandos presentes na página *web*⁴⁴. Verificou-se que para diferentes televisões alguns códigos não se encontravam implementados, daí terem sido utilizados os mais genéricos, bem como o utilização do mínimo de teclas do comando da televisão, como por exemplo as teclas direcionais, o *ok* e *return* (ver Figura 21).



Figura 21 - Exemplo das teclas utilizadas na navegação na interface

Associando os módulos *node-cec* e *cec-monitor* em conjunto com o *sockets.io* a correr no servidor *nodejs* torna-se possível encaminhar os eventos provenientes do comando do televisor para a interface gráfica.

⁴⁴ Disponível em <http://www.cec-o-matic.com/>

Na interface gráfica, utilizando o *eventbus* implementado no *Vue.js*, consegue-se interagir com a interface que é disponibilizada pelo servidor e apresentada na televisão. A cada evento do comando o mesmo é encaminhado para a interface gráfica. O *eventbus* sabe precisamente onde o utilizador se encontra permitindo apenas reagir às teclas configuradas, navegando de acordo com a utilização dessas teclas (ver Figura 22 e Figura 23).



Figura 22 - Exemplo da utilização das teclas para navegação na barra lateral

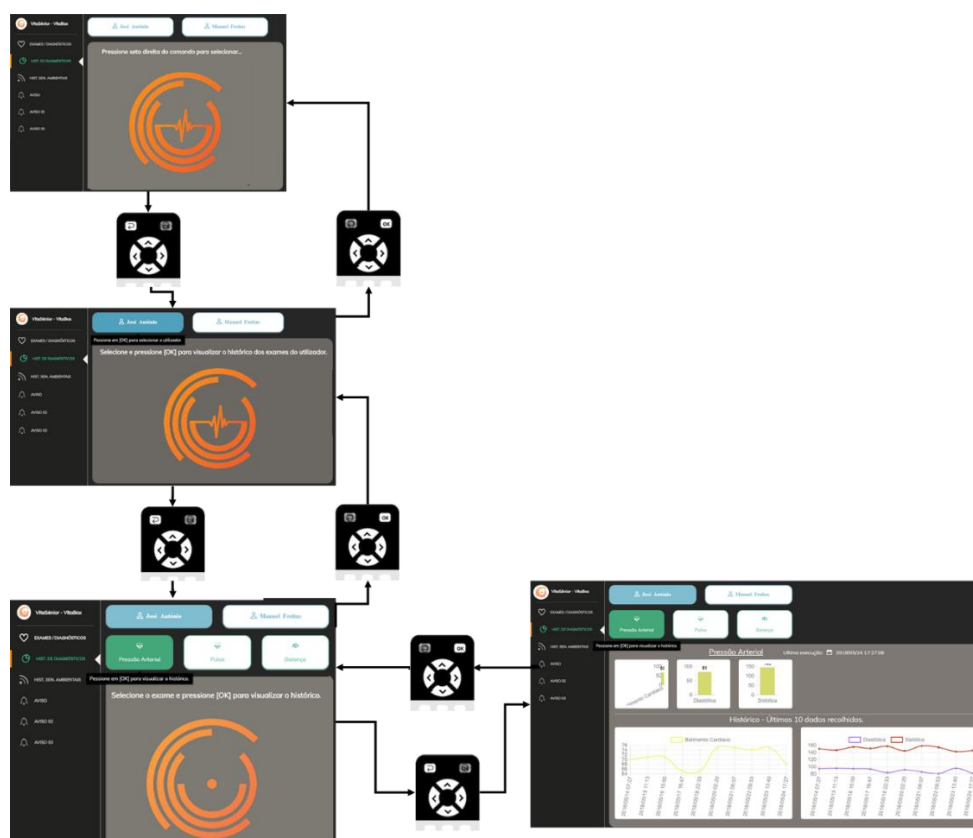


Figura 23 - Exemplo da utilização do comando dentro de uma opção da barra lateral

Todas as teclas são passíveis de configuração de acordo com as preferências de cada utilizador.

Todos os eventos produzidos pelo comando são encaminhados para o servidor e posteriormente por intermédio dos *sockets* são enviados para a interface de forma a serem tratados. Devido à estrutura da interface, existe um módulo “*main.js*” onde se encontram configuradas as características e os módulos a utilizar pela parte gráfica da aplicação. Neste módulo foi criado um método que está à escuta de todos os eventos provenientes do *socket* que os recebe do canal HDMI. Com esta abordagem, sabe-se qual a localização precisa a cada instante, sendo desta forma descartadas todas as teclas utilizadas pelo comando que não se encontram configuradas.

Na utilização do *eventbus* é possível definir uma identificação para encaminhar os eventos de acordo com a sua localização. Com esta característica foram criadas diferentes identificações que auxiliam no encaminhamento dos eventos de acordo com a sua posição a cada instante.

- Identificação para os eventos quando a posição atual é a barra lateral,
 - `EventBus.$emit('move-sidebar', -1)`
 - `EventBus.$emit('move-sidebar', 1)`
 - “-1”, “1” – Identificação se o evento corresponde à seta para cima ou para baixo
- Identificação para os eventos quando a posição atual é dentro dos componentes,
 - `EventBus.$emit('move-components, cmd)`
 - “cmd” – corresponde ao evento vindo do comando, posteriormente consultado a base de dados para adquirir a ação associada.
- Identificação para os eventos de alertas.
 - `EventBus.$emit('changeTab')`

Desta forma torna-se mais fácil e simples de encaminhar as decisões tomadas pelo utilizador para navegar na interface pelos diferentes menus conforme se verifica nas imagens (ver Figura 22 e Figura 23).

4.4 Interação com os dispositivos *Bluetooth*

Para a integração dos vários dispositivos *Bluetooth* houve o auxílio do bolseiro do Projeto VITASENIOR-MT, Pedro Dias, que forneceu os serviços e características necessárias para

interagir com os vários equipamentos. Os equipamentos médicos/monitorização *Bluetooth* efetuam diversas leituras ou executam exames fisiológicos. Estas leituras são realizadas em diferentes alturas através de diferentes equipamentos, daí ser necessário para cada execução criar um novo processo o qual é responsável por toda a execução, desde a deteção, recolha dos dados e conclusão da execução. De acordo com os diversos tipos de exames destinado a cada utente, os equipamentos foram previamente configurados e atribuídos no sistema remoto, e depois carregados ou atualizados na VITABOX. Quando a página destinada à execução dos exames é carregada é efetuada uma pesquisa à base de dados para carregar os diversos utentes afetos à VITABOX. Ao selecionar o utente serão listados os diversos exames que lhe são atribuídos (ver Figura 24).

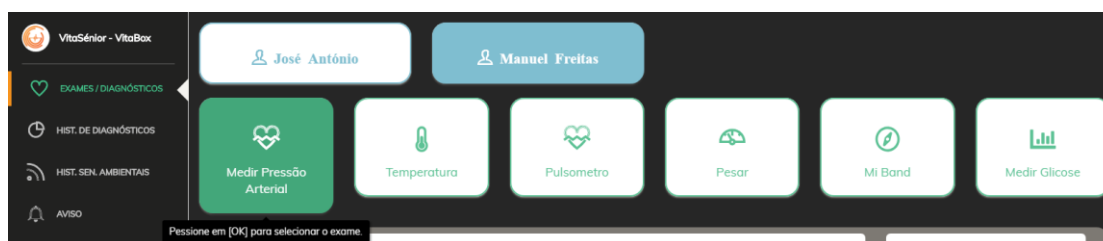


Figura 24 - Exemplo da listagem dos utilizadores e dos exames a ele atribuídos

Utilizado as teclas disponíveis para a navegação descritas no ponto (4.3) o utente tem a possibilidade de iniciar a execução do exame pretendido. Após o início da execução, a interface fica bloqueada (ver Figura 25) para que o utilizador não mude para outra secção da interface ficando sem saber a ação que se encontra em execução.

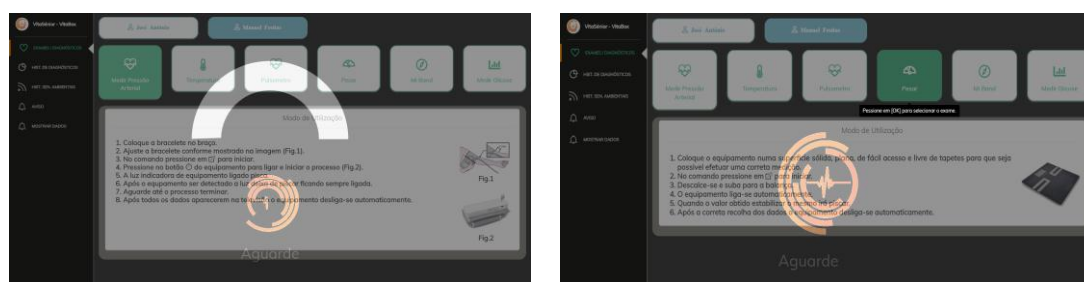


Figura 25 - Exemplos do bloqueio do ecrã durante a execução dos exames

Para cada exame existe um pequeno manual de instruções apresentado na página de como proceder para a sua utilização. Após o utilizador iniciar a processo de execução vão surgindo diversas mensagens informativas (ver Figura 26) do estado da execução. Na conclusão da execução, a interface é desbloqueada, apresentando os resultados e realçando-os como se verifica na imagem (ver Figura 27).

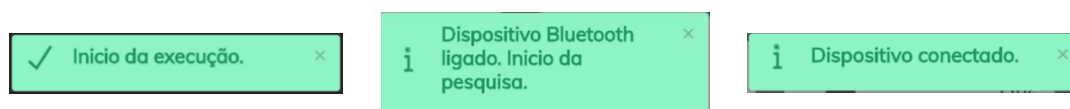


Figura 26 - Exemplos de mensagens informativas durante a execução dos exames

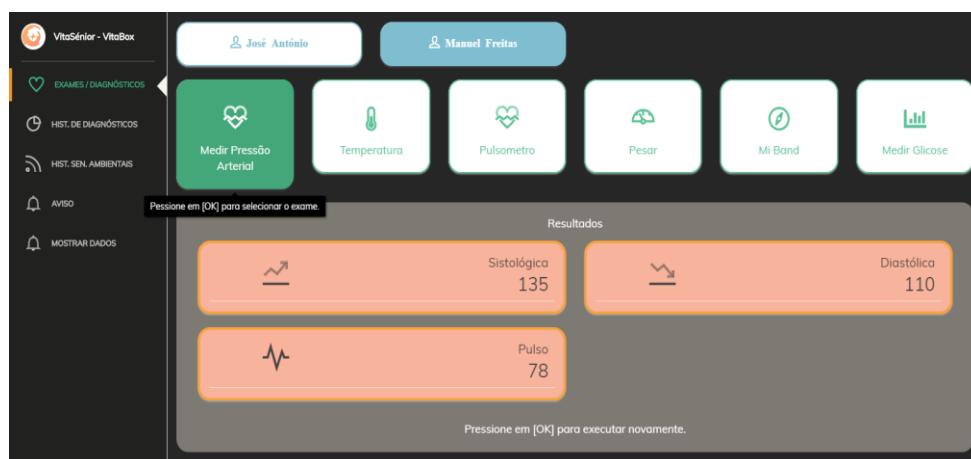


Figura 27 - Apresentação do resultado de um exame

Durante a execução dos exames podem surgir erros, quando esta situação ocorre o utilizador é informado com uma mensagem (ver Figura 28) e a interface é desbloqueada dando a oportunidade de uma nova tentativa de efetuar o mesmo exame ou outro como preferir.

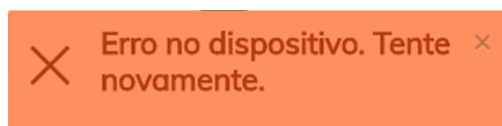


Figura 28 - Mensagem de erro durante a execução do exame

Quando o utilizador pretende realizar um exame, seleciona o exame desejado e pressiona a tecla *Ok* do comando da televisão. O servidor inicia a procura do dispositivo associado à execução. A pesquisa do equipamento dura no máximo um minuto. O utilizador dispõe deste tempo para colocar e ligar o equipamento caso isso ainda não tenha sido feito. Se o utilizador não iniciar o equipamento nesse minuto surge a informação de que o equipamento não foi encontrado (ver Figura 29) e a interface é desbloqueada sendo necessário voltar a dar início da realização do exame caso seja essa a intenção do utilizador, ou navegar pela aplicação.

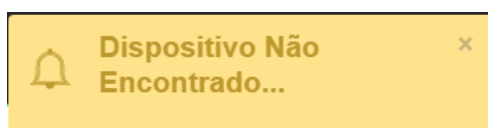


Figura 29 - Mensagem de equipamento não encontrado

Caso o utilizador inicie a realização de um exame e não seja essa a sua intenção a execução pode ser cancelada pressionando no comando da televisão na tecla *return* ficando a interface desbloqueada e surgindo a mensagem que o exame foi cancelado (ver Figura 30).

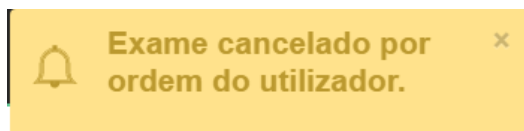


Figura 30 - Mensagem de exame cancelado por ordem do utilizador.

4.4.1 Sistema automático de captura de dados biométricos

Entre os vários equipamentos biométricos um deles consiste numa bracelete colocada no pulso do utente e que após a sua autenticação não necessita de mais nenhuma ação por parte do utilizador. Este equipamento tem a capacidade de medir o batimento cardíaco e o número de passos efetuados durante as 24 horas do dia, e como não é necessária nenhuma ação por parte do utilizador foi desenvolvido um *script* (Anexo 1) que tem a capaz de consultar a banda para aquisição desses dados.

Para que não fosse necessário a adição de um temporizador ao servidor para efetuar esta operação, foi adicionada uma nova entrada (ver Figura 31) no ficheiro *crontab* [25], este ficheiro faz parte do sistema operativo, que tem a capacidade de executar comandos em intervalos regulares com um mínimo de um minuto de intervalo, sem que seja necessário a instalação de outras aplicações ou bibliotecas, nessa entrada é invocado o *script* para invocar as várias chamadas à bracelete biométrica (Anexo 2) que será executado em intervalos regulares de trinta minutos.

```
*/30 * * * * root /home/pi/Desktop/Vitabox/Scripts/runAutoMiBand.sh
```

Figura 31 - Entrada adicionada ao *crontab*

O *script* (Anexo 1) que foi desenvolvido utilizando os mesmos bibliotecas que se encontram no servidor para a utilização dos diferentes equipamentos biométricos, devido a todos os equipamentos serem *Bluetooth* e apenas um é emparelhado com o *Raspberry Pi*, devido a esta situação podem ocorrer erros aquando da execução de um exame e o início do *script*, ou a execução do *script* e o utilizador dar início da realização de um exame, este problema está previsto sendo o utilizador informado da ocorrência de um erro (ver Figura 28), podendo o utilizador voltar a efetuar a ação.

4.5 Associação dos sensores ambientais à VITABOX

A primeira vez que é iniciada a VITABOX, da chave destinada a identificar o equipamento é introduzida, sendo necessário uma ligação à Internet, para que sejam descarregados os utilizadores e os vários sensores e transdutores associados à VITABOX.

Quando a VITABOX é iniciada, o sistema operativo é carregado, é feita uma consulta à base de dados local para recolher a lista dos sensores ambientais, e uma consulta ao sistema remoto para validação se a lista é a mais atual e caso necessário a mesma será atualizada. Com o resultado da consulta é criada uma lista em memória. Considerámos esta abordagem devido ao acesso à memória ser mais rápido do que à base de dados, não sendo necessário que a cada receção da informação dos sensores haja uma consulta à base de dados para encontrar a informação complementar dos sensores e transdutores.

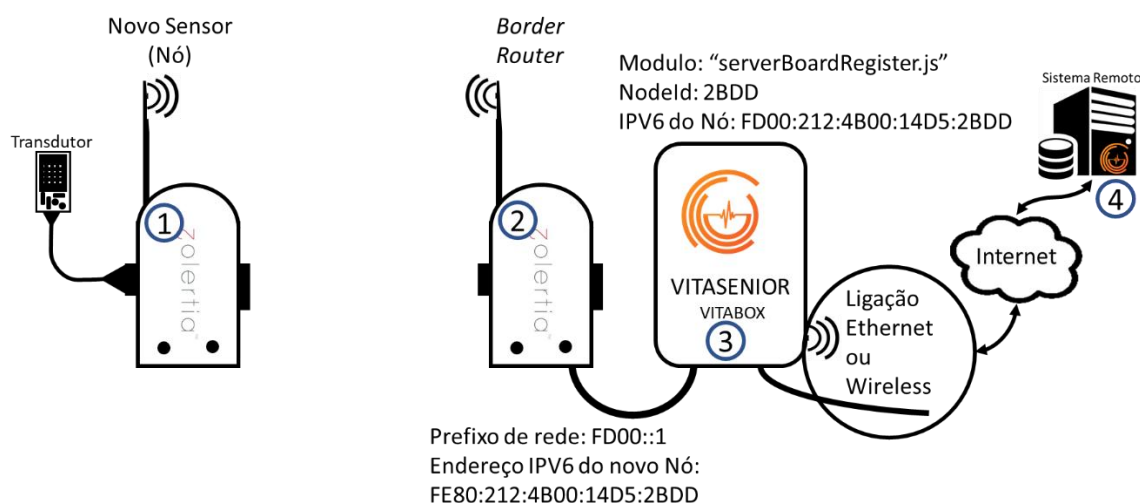


Figura 32 – Nó da rede e border router. Exemplo de associação do um novo nó

Quando um novo nó (1) se anuncia na rede e é detetado pelo *border router* (2), o *border router* (2) envia o endereço do novo nó (1) para a VITABOX validar (ver Figura 32).

Quando é iniciada a VITABOX (3) é carregada a lista de todos os sensores (nós) e transdutores que fazem parte do sistema e que se encontram configurados no sistema remoto (4). A lista dos equipamentos e nós conhecidos, que se encontram associados a cada VITABOX (guardada em memória e na base de dados), previamente configurados no sistema remoto, é atualizada no caso de um novo nó ser introduzido no sistema remoto, sem que seja necessário o reinício do sistema.

Quando a VITABOX (3) recebe no módulo “serverBoardRegister.js” o endereço do novo nó, é colocado o prefixo da rede caso não o possua. Através do último bloco do endereço IPV6 do nó é obtido o *nodeid* que irá identificar o nó.

O *nodeid* valida se o nó faz parte da lista de nós fornecida pelo sistema remoto (4):

- O nó pertence à lista de nós conhecidos,
 - A VITBOX (3) envia a mensagem, “*flag1[nodeid]*”, que indica ao *border router* (2) que o nó é conhecido e que pode receber dados do novo nó.
- O nó não pertence à lista de nós conhecidos,
 - A VITBOX (3) não conhece o *nodeid* e envia para o *border router* (2) a mensagem, “*flag2[nodeid]*”, que indica que o nó não é conhecido e toda a informação recebida do nó (1) é descartada.

Esta validação ajuda proteger o sistema da introdução de equipamentos não autorizados.

Este processo encontra-se mais detalhado noutro relatório em fase de realização no âmbito do projecto VITASENIOR-MT.

4.6 Armazenamento de dados

Tendo sido escolhida uma base de dados não relacional para o armazenamento dos dados recolhidos pelos sensores ambientais e biométricos, onde cada novo registo é um novo documento a ser armazenado, foi considerada uma estrutura que facilitasse as várias manipulações dos dados armazenados, conforme descrito nas secções seguintes.

4.6.1 Sensores ambientais

4.6.1.1 Armazenamento - 1ª – Método utilizado

Considerámos que a melhor forma de armazenar os novos registos seria a criação de um documento que representasse o sensor e o que a ele se encontra interligado (transdutores). Cada documento possui a informação de identificação do sensor e uma lista com todos os transdutores associados ao sensor. Cada posição da lista contém a informação necessária para a identificação do transdutor e uma lista com os diversos valores recolhidos por ele. Na lista dos valores para além do valor é adicionada a data e hora à qual o mesmo foi recolhido, como ilustra a estrutura do exemplo na Figura 33.

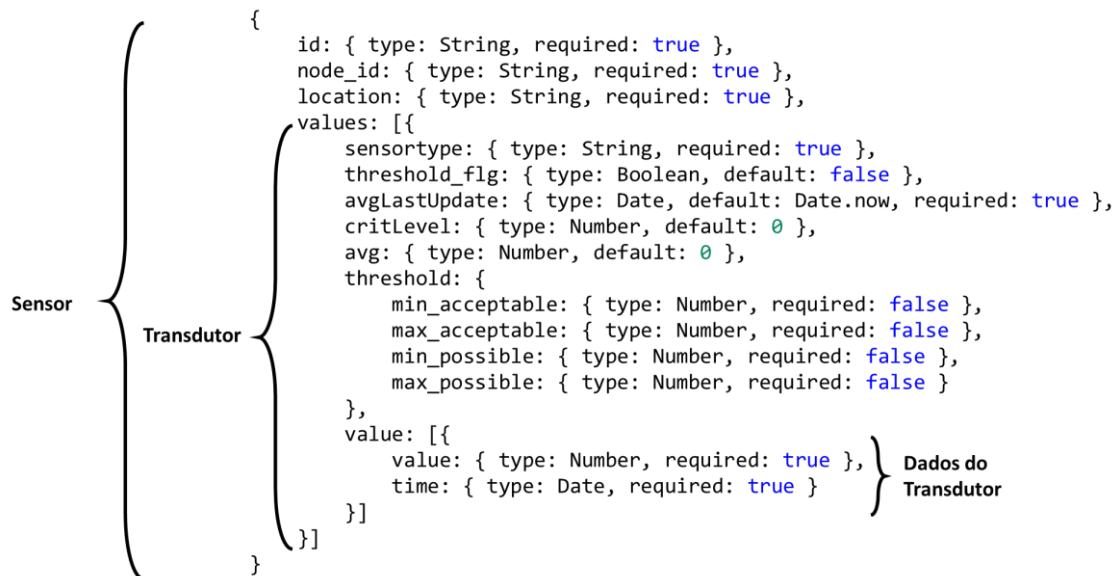


Figura 33 –Estrutura do objeto Sensor 1ª abordagem

Inicialmente esta estrutura pareceu ser a melhor forma de armazenar os dados na base de dados, mas após algum tempo de funcionamento do sistema, com registo de dados a uma cadência de um segundo por cada transdutor, verificou-se que a quantidade de dados a ser processada para cada inserção era cada vez maior. Esta constatação resultou dos testes descritos no capítulo 5.1 onde se mostra a utilização do processador e da memória ram do microcomputador.

O motivo do aumento de processamento devia-se ao facto de por cada adição de um novo valor do transdutor ser necessário seleccionar o transdutor dentro do sensor e colocar o novo valor. Depois da recolha dos últimos dez registos, calculava-se a média dos mesmos e armazenava-se o valor. Em concorrência era calculado o nível crítico do transdutor e atualizado em conformidade, e em caso da existência de um alarme era informado o sistema remoto sendo apresentado um alerta na TV.

4.6.1.2 Armazenamento - 2ª – Método utilizado

A estrutura do objeto do sensor foi otimizada e adotada uma abordagem diferente na forma de armazenamento dos diversos dados recolhidos pelos sensores.

Passou a ter-se um objeto sensor com a informação necessária para a sua identificação e um transdutor (ver Figura 34), ou seja, passámos a ter dados duplicados, mas foi a solução encontrada para seleccionar o enorme processamento apresentado no ponto anterior. Para

além do objeto sensor foi criado um novo objeto, o *rawSensor* (ver Figura 35), o qual contém a informação necessária para identificar corretamente o transdutor que adquiriu o valor.

Sensor {

```

  board_id: { type: String, required: true }, // Identificação da board
  sensor_id: { type: String, required: true }, // Identificação do transdutor
  node_id: { type: String, required: true }, // Identificação do mote na rede de sensores
  sensortype: { type: String, required: true },
  location: { type: String, required: true },
  measure: { type: String, required: true },
  threshold_min_acceptable: { type: Number, required: false },
  threshold_max_acceptable: { type: Number, required: false },
  threshold_min_possible: { type: Number, required: false },
  threshold_max_possible: { type: Number, required: false },
  avg: { type: Number, default: 0 },
  avgLastUpdate: { type: Date, default: Date.now, required: true },
  critState: { type: Boolean, default: false }
}
```

Figura 34 - Estrutura do objeto Sensor seguindo a 2ª abordagem

RawSensor {

```

  id: { type: String },
  patient_id: { type: String },
  board_id: { type: String, required: true },
  sensor_id: { type: String, required: true },
  location: { type: String },
  transducer: { type: String, required: true },
  measure: { type: String, required: true },
  datetime: { type: Date, required: true },
  value: { type: Number, required: true },
  bio: { type: Boolean, default: false },
  flg_available: { type: Boolean, default: true },
  flg_avg: { type: Boolean, default: false }
}
```

Figura 35 - Estrutura do objeto RawSensor

Com estas alterações houve uma melhoria significativa no processamento do microcomputador como se pode verificar no capítulo 5.1.2.

Com esta estrutura de objetos, a inserção de novos dados recolhidos pelos transdutores tornou-se direta pois não é necessário consultar os registos já armazenados. Em complemento com estas alterações passou-se a armazenar os dados apenas de dez em dez minutos, mas com pedidos aos sensores, de cerca de um em um minuto. A cada pedido ao sensor, ele devolve uma *string* com a estrutura de um objeto JSON, com os valores de todos os transdutores, durante este intervalo os dados são armazenados na memória *ram* e no fim do intervalo pré-definido é calculada a média dos mesmos e armazenado na base de dados, utilizando o a estrutura do objeto *rawSensor*.

Consideramos a abordagem do cálculo da média dos valores recolhidos no intervalo de tempo para ajudar a eliminar ou atenuar valores dos transdutores resultantes de anormalidades na leitura.

4.6.2 Equipamentos Biométricos

Foi contemplada a possibilidade da existência de várias pacientes associados à mesma VITABOX, com diferentes exames atribuídos a cada um. Neste sentido, para que o equipamento consiga distinguir quais os exames que foram atribuídos, foi criado um objeto *Patient* para acomodar as informações e os exames atribuídos pelo sistema remoto ao paciente (ver Figura 36). Como já foi indicado na secção 4.4 quando se pretende efetuar uma utilização de um equipamento biométrico é consultado o objeto do *Patient* (ver Figura 36) que contem a lista dos exames atribuídos ao utilizador selecionado.



Figura 36 - Estruturado objeto do paciente

Na utilização de um equipamento deste tipo, após a conclusão do processo de recolha dos dados, apenas são armazenadas e visualizadas as características que compõem o

equipamento biométrico definidas no sistema remoto. Para armazenar os dados recolhidos é utilizado o mesmo objeto que nos sensores ambientais, o objeto *rawSensor* (ver Figura 35), pois esta estrutura foi pensada para poder alojar os diferentes tipos de sensores, ambientais e biométricos. Como se pode verificar no objeto *rawSensor* existe uma *flag (bio)* que possibilita a diferenciação dos tipos de sensores armazenados, para que seja possível a sua seleção e visualização dos dados na interface gráfica.

4.6.3 Dados Pré-inseridos

Quando a VITABOX é iniciada, o sistema carrega automaticamente na base de dados as configurações de todas as teclas do comando remoto com a ação atribuída a cada evento (caso elas não existam).

Para o armazenamento destas configurações foi criado um objeto *Remote* (ver Figura 37), onde para cada tecla é atribuída a ação desejada.

```

Remote {
  // Nome da tecla
  name: { type: String, required: true },
  // Código da tecla recebida pelo HDMI-CEC
  code: { type: String, required: true },
  // Ação atribuída à tecla
  task: { type: String, required: true },
  timed_flg: { type: Boolean, default: false }
}

```

Figura 37 – Estrutura do objeto das configurações das teclas do comando remoto

A cada evento produzido pelas teclas do comando remoto e capturado no canal HDMI, os objetos do tipo *remote* são consultados a fim de indicar ao servidor e à interface gráfica a ação configurada.

4.7 VITABOX – Modos (*Prod*, *Dev*)

Aquando dos desenvolvimentos verificamos alguma dificuldade em relação à obtenção dos dados recebidos dos sensores ambientais pois os mesmos também se encontravam no início dos seus desenvolvimentos. Para contornar este problema foi implementada a possibilidade daa VITABOX ser iniciada em ambientes diferentes que neste caso foram atribuídas as designações de *Prod* e *Dev*.

- *prod* - Produção

- *dev* – Desenvolvimento

Todo o código em execução é comum a ambos os ambientes, à exceção do carregamento de alguns dados de testes, os quais apenas são carregados na ausência de objetos do tipo *Patient*, *Sensor* e *rawSensor*. Dentro da estrutura de pastas e ficheiros que constituem os elementos necessários para a execução da VITABOX existe um denominado por *populateDB.js* o qual contem algumas sementes que deram a possibilidade de avançar nos desenvolvimentos bem como alguns testes de comunicação com o sistema remoto, simulando alguns erros ou falhas que poderão ocorrer e causar alguns problemas. Por exemplo o tamanho máximo do conteúdo a ser enviado de uma única vez para o sistema remoto, tentativa de comunicação havendo uma falha no envio, etc.

O servidor *Nodejs* possui uma estrutura de ficheiros e pastas, nesta estrutura existe um ficheiro de configuração, o *package.json* onde são declarados todos os módulos utilizados. Este ficheiro auxilia na utilização de todo o projeto noutra localização de forma a utilizar os módulos externos nas versões consideradas nos desenvolvimentos. Na estrutura deste ficheiro existem vários blocos:

- Identificação do autor, versão,
- Identificação das dependências que auxiliam o projeto,
- Identificação de *scripts*.

Nos *scripts* (ver Figura 38) podemos indicar de acordo com o comando de início em que ambiente pretendemos estar a trabalhar, sendo as configurações necessárias carregadas de acordo com o ambiente em execução (ver Figura 39).

```
{
  . . .
  "scripts": {
    "start": "NODE_ENV=PROD node main.js",
    "dev": "NODE_ENV=DEV node main.js",
    "prod": "NODE_ENV=PROD node main.js",
    "band-dev": "NODE_ENV=DEV node startAutoMiBand.js",
    "band-prod": "NODE_ENV=PROD node startAutoMiBand.js"
  }
  . . .
}
```

Figura 38 – Scripts da definição do ambiente em execução

```

    (. . .)
    mode = process.env.NODE_ENV || "DEV";
    (. . .)
var Main = function () {
    (. . .)
    let pathCfg = './config-' + mode.toLowerCase() + '/config.js';
    config = require(pathCfg);
    (. . .)
};
new Main();
module.exports = Main;

```

Figura 39 - Carregamento das configurações de acordo com o modo de execução do servidor

Para que o servidor consiga saber que ambiente se encontra em execução, no arranque é definida uma variável de ambiente *NODE_ENV* que contem a identificação do ambiente utilizado, sendo por defeito o ambiente *dev*.

Para iniciar o servidor no modo desejado basta executar o comando *npm run dev* ou *npm run prod*. De acordo com o último parâmetro do comando é executado o *script* a ele associado e definida a variável de ambiente (ver Figura 38). Durante a execução do servidor esta variável encontra-se acessível para determinação das configurações a serem utilizadas.

Para além da definição do ambiente de execução do servidor também se encontra configurado o *script* a ser utilizado pelo modo automático de captura de dados biométricos explicado na secção 4.4.1. Para esta execução também se encontram configurados os dois modos de execução, *prod* e *dev* (ver Figura 38).

4.8 Internacionalização - I18n

Com o auxílio do módulo *i18n* que se encontra adicionado à interface gráfica, existe a possibilidade de colocar o texto na linguagem que melhor se adequa ao utilizador. De momento, encontra-se disponíveis duas linguagens, português (*pt*) e inglês (*en*), mas facilmente são adicionadas outras linguagens.

Esta alteração é conseguida, após a seleção da linguagem desejada no sistema, efetuando a alteração automaticamente, ou seja, o módulo a cada solicitação de alteração da linguagem altera a origem do ficheiro que contem o texto da linguagem selecionada. No servidor existem dois ficheiros JSON onde o seu nome corresponde à sigla da linguagem *pt.json* e

en.json, estes ficheiros correspondem à estrutura JSON, onde o texto a ser carregado é mapeado do ficheiro que corresponde à linguagem pretendida.

4.9 Audio Descrição

O sistema conta com audio-descrição, a qual tem a capacidade de reproduzir o texto apresentado no ecrã. Esta funcionalidade foi adicionada para auxiliar as pessoas com maior dificuldade na leitura.

Para esta implementação foram desenvolvidos dois algoritmos diferentes, um responsável pela reprodução de texto e outro para a reprodução de números.

No servidor existe uma pasta que contém os ficheiros de áudio, onde cada um corresponde a uma palavra ou algarismo, devidamente separados.

Para a reprodução de uma frase (*string*) enviada ao servidor, a mesma é processada separando-a em palavras, seleccionando cada palavra com o audio correspondente. Após a obtenção de todos os ficheiros é utilizado o módulo *audioconcat*, que se encontra incorporado no servido, que em conjunto com a aplicação *ffmpeg* faz a união de todos os ficheiros, resultando apenas um ficheiro de audio cujo nome é indicado à interface, e o mesmo é reproduzido.

Para a reprodução de números recorreu-se à utilização de um algoritmo que com base no número recebido consegue criar a sua representação em texto (*array* de ficheiros áudio, por exemplo “./100.mp3” “./e.mp3” “./20.mp3” “./e.mp3” “./3.mp3”; algoritmo em Anexo 3) sendo posteriormente unidos todos os ficheiros de áudio que o compõem o número.

4.10 Interface Gráfica

4.10.1 Painel de Configurações

Devido à existência de várias opções de configuração passíveis de serem alteradas de acordo com as preferências de cada utilizador, houve a necessidade de criar uma página de configurações, de fácil acesso e utilização. Qualquer utilizador pode aceder a esta página de configurações utilizando uma nova tecla do comando da televisão Figura 21.

A Figura 40 mostra o painel de configuração com as funcionalidades que podem ser alteradas na VITABOX.

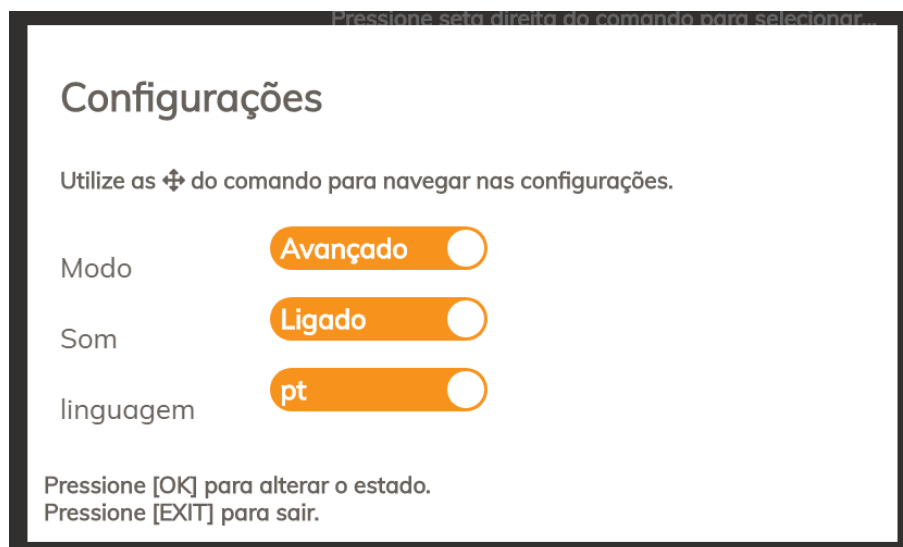


Figura 40 - Painel de configurações

Neste painel podemos alternar entre o modo Avançado onde o utilizador tem à sua disposição todos os itens de navegação na barra lateral e o modo Básico onde apenas estão disponíveis alguns itens, os considerados mais simples de utilização (ver secção 4.10.2). É possível ligar ou desligar o áudio da aplicação. Quando o áudio é desligado e o modo básico está ativado, o modo de visualização automático (visualização cíclica) que se encontra disponível no modo básico também fica desativado. Existe ainda a possibilidade de alternar entre as duas linguagens disponíveis na VITABOX.

4.10.2 Modo Avançado e Modo Básico

A aplicação da VITABOX possui várias opções disponíveis, o que pode originar alguma dificuldade de interação em utilizadores com menor apetência para o uso da tecnologia. Para salvaguardar essas situações foi implementada a possibilidade de comutar entre um modo mais simples (Básico) onde serão apresentadas poucas opções de fácil interpretação e utilização, e um modo avançado (Avançado) onde são apresentadas todas as opções disponíveis, para utilizadores mais experientes. A troca de modo é conseguida por intermédio do painel de configurações que se encontram disponíveis para o utilizador demonstrado na secção 4.10.

4.10.2.1 Modo Avançado

Neste modo o utilizador conta com todas as opções de navegação e interação disponíveis no sistema. Este modo é mais aconselhado para pessoas com maior destreza na utilização da tecnologia bem como na interpretação dos dados apresentados.

4.10.2.2 Modo Básico

No modo básico o utilizador conta com um número limitado de opções, onde a interação com o sistema é menor. Neste modo, o separador de execução de exames é idêntico ao do modo Avançado. Os outros separadores possuem a funcionalidade de áudio-descrição do conteúdo apresentado, e o sistema consegue automaticamente navegar por todos os itens presentes no separador descrevendo-os.

4.11 Sistema de notificações e alertas

O sistema monitoriza constantemente as variáveis ambientais. A receção dos dados dos sensores ambientais é validada com os perfis introduzidos no sistema remoto. No caso de os valores ambientais serem anómalos, indicando a existência de situação de risco ambiental, a VITABOX comuta automaticamente para a apresentação de um painel sinalizando a situação de risco (ver Figura 41). Esse painel de visualização fica bloqueado não respondendo a qualquer comando sem que o utilizador valide ter tomado conhecimento da situação pressionando em *Ok*, caso a TV se encontre desligada a mesma será ligada e apresentada a Figura 41. Para além do aviso local é enviado para o sistema remoto a indicação da ocorrência de uma situação de alarme, indicando a causa que lhe deu origem.



Figura 41 – Exemplo de alerta de um sensor

Para além dos alertas, a VITABOX também dispõe de um sistema de notificações em que o sistema mostra uma mensagem, gerada no servidor com base em informações introduzidas pelo médico relativamente à periodicidade dos exames ou prescrições. Apresentam-se de

seguida alguns exemplos de notificação (no estado atual do projeto, estas notificações foram apenas simuladas):

- Efetuar determinado exame,
- Tomar medicamentos,
- Consulta dia x,
- Etc.

As notificações aparecem na televisão, no canto inferior direito, independentemente do que o utilizador se encontra a visualizar no momento (ver Figura 42). Essa notificação aparece a piscar e está limitada a 13 caracteres. Para visualizar o conteúdo da notificação, o utilizador tem de pressionar a tecla OK, e o sistema comuta para o painel onde é apresentada a mensagem de notificação (ver Figura 43).

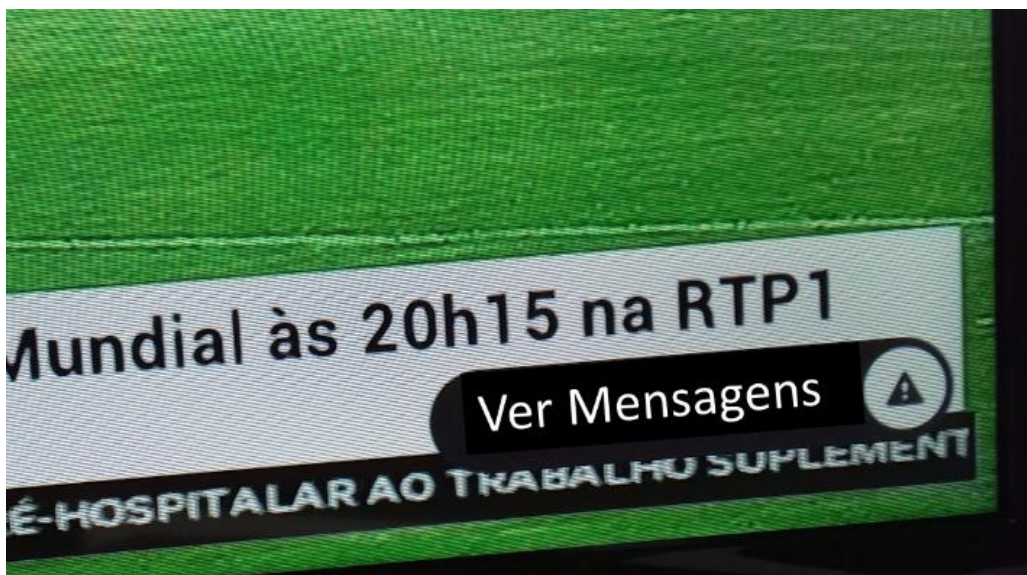


Figura 42 - Mensagem de notificação da existência de novas mensagens na VITABOX

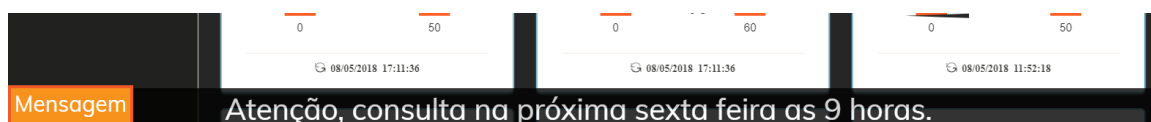


Figura 43 - Exemplo da uma mensagem recebida pela VITABOX para o utilizador.

4.12 Comunicação com o Sistema Remoto

Quando a VITABOX inicia, envia a chave que foi previamente atribuída e colocada no equipamento durante o aprovisionamento, a qual faz a sua identificação no sistema remoto, e permite adquirir um *token* para a realização de pedidos.

Após a aquisição do respetivo *token*, a VITABOX consulta o sistema remoto para adquirir todos os sensores e transdutores registados para o equipamento, bem como os utilizadores e os exames a eles associados, atualizando a base de dados local. Ou seja, existe sempre uma réplica do que se encontra no sistema remoto, o que permite o funcionamento do sistema mesmo no caso da inexistência de ligação à Internet.

A VITABOX necessita obrigatoriamente de ter uma ligação à internet na primeira vez que é iniciada pois precisa de receber as definições, utilizadores, sensores e transdutores para que o sistema possa começar a ser utilizado, no caso de posteriormente ao sistema se encontrar em funcionamento e falhar a ligação à internet, ou sistema reinicie e não exista a ligação à internet o sistema VITABOX irá funcionar normalmente com as configurações que possui e após o restabelecimento da ligação à internet o mesmo será atualizado automaticamente passando a utilizar as novas configurações, não sendo necessário que a VITABOX seja reiniciada.

O sistema remoto tem conhecimento de todas as variáveis que cada equipamento biométrico consegue adquirir, ou seja, o sistema remoto indica à VITABOX quais os equipamentos biométricos a atribuídos a cada utilizador e as variáveis de cada equipamento que devem ser recolhidos para apresentar na televisão e posteriormente enviar para o sistema remoto. Com esta flexibilidade no sistema remoto pode adicionar vários utilizadores a uma VITABOX, com exames biométricos diferentes entre eles e as variáveis a serem recolhidas de cada equipamento também podem ser diferentes de utilizador para utilizador (ver Figura 44 e Figura 45).

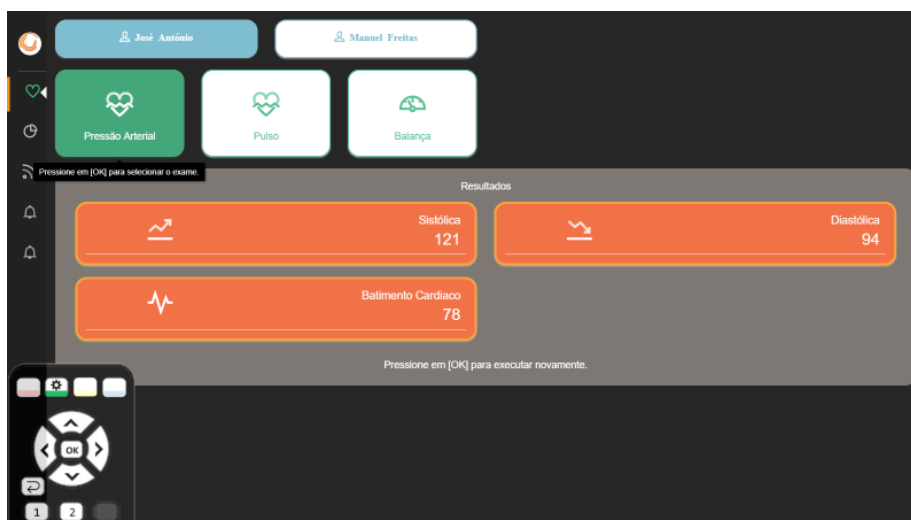


Figura 44 - Exemplo de utilizador com três exames e três variáveis de um exame (Pressão arterial)

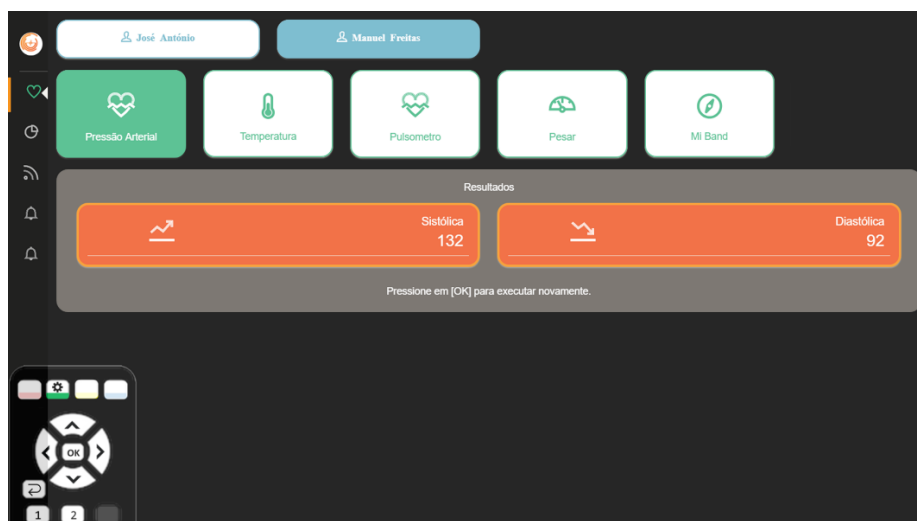


Figura 45 - Exemplo de utilizador com cinco exames e duas variáveis de um exame (Pressão arterial)

O sistema localmente vai armazenando os dados dos sensores e consultando a lista de registos presentes do tipo *rawSensor* que por intermédio de uma variável de controlo identifica se o registo já se encontra no sistema remoto. Aquando de um correto envio para o sistema remoto, a VITABOX localmente atualiza a variável de controlo, sendo guardado um histórico de vinte e quatro horas para os sensores ambientais e as últimas dez leituras para os dispositivos biométricos. Após o processo de atualização da variável de controlo de envio para o sistema remoto são verificados os registos do tipo *rawSensor* e todos que já tenham sido enviados e que estejam fora do limite do histórico estipulado serão eliminados da base de dados local.

O limite do histórico foi o considerado aceitável para apresentação dos dados na interface gráfica do sistema, a Figura 46 e Figura 47 foram gerados com dados aleatórios para demonstrar o resultado dos gráficos com a quantidade de dados considerada.

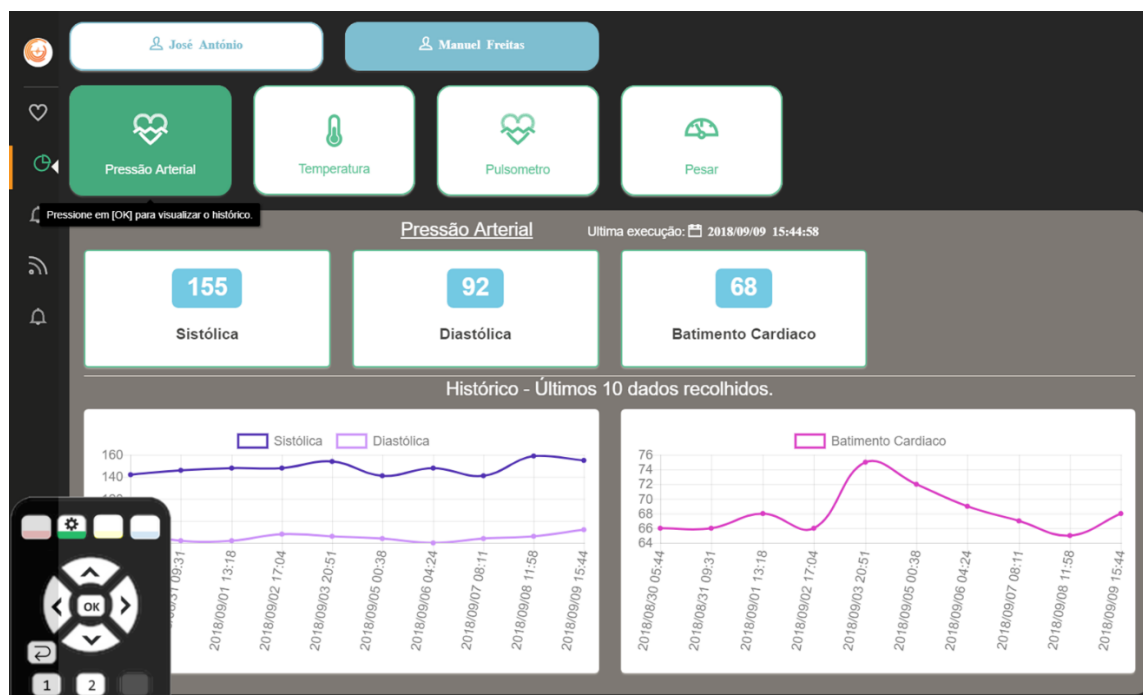


Figura 46 - Histórico do sensor biométrico da medição da pressão arterial



Figura 47 - Histórico do sensor ambiental Temperatura

5 Resultados experimentais

Este capítulo descreve os resultados experimentais realizados do ponto de vista de processamento computacional e de usabilidade, os quais serão apresentados nas secções seguintes.

5.1 Testes de carga

Durante o desenvolvimento da VITABOX deparámo-nos com diversos problemas relacionados com o desempenho do microcomputador escolhido bem como na conceção dos algoritmos. Como já foi indicado na secção 4.6.1, fomos forçados a repensar na forma de armazenamento dos dados dos sensores ambientais devido à exagerada utilização dos recursos de processamento e de memória.

Para a realização dos testes de carga foi criado um gerador de dados para simular os motes com vários transdutores associados a cada um. O gerador de dados simula seis motes com quantidade de transdutores variável a cada envio com uma cadência de envio de 300 milissegundos.

Para cada teste realizado no microcomputador encontrava-se em execução:

- O sistema operativo com todos os seus processos associados.
- O gerador e dados.
- Servidor *Nodejs* e os seus serviços:
 - *Sockets* para comunicação com a interface,
 - Receção de dados dos sensores,
 - Interação com a base de dados.
- Servidor de base de dados
- Interface VITABOX carregada.
- Processos acessórios para a criação dos *logs* de monitorização.

No processo de criação dos diferentes gráficos apresentados nos pontos seguintes, os dados foram agrupados por minuto realizando a média da utilização por processo de acordo com a aplicação.

5.1.1 Teste - 1ª estrutura do Objeto de armazenamento de dados

Na secção 4.6.1.1 foi pensado uma estrutura do objeto para os dados recebidos pelos sensores ambientais (*motes*) que se revelou não ser a melhor abordagem. A estrutura foi submetida à utilização de um gerador de dados para testar o desempenho do armazenamento, a qual se revelou não ser a melhor forma como se pode verificar pela visualização dos gráficos que representam a utilização do CPU (ver Figura 48) e da memória RAM (ver Figura 49).

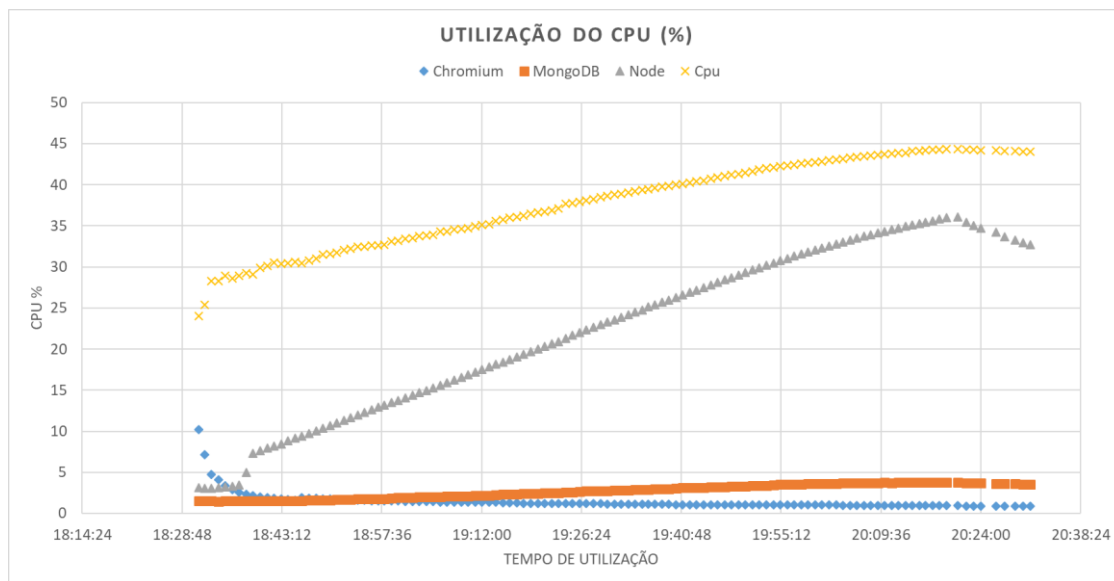


Figura 48 – Percentagem da utilização do CPU (%)

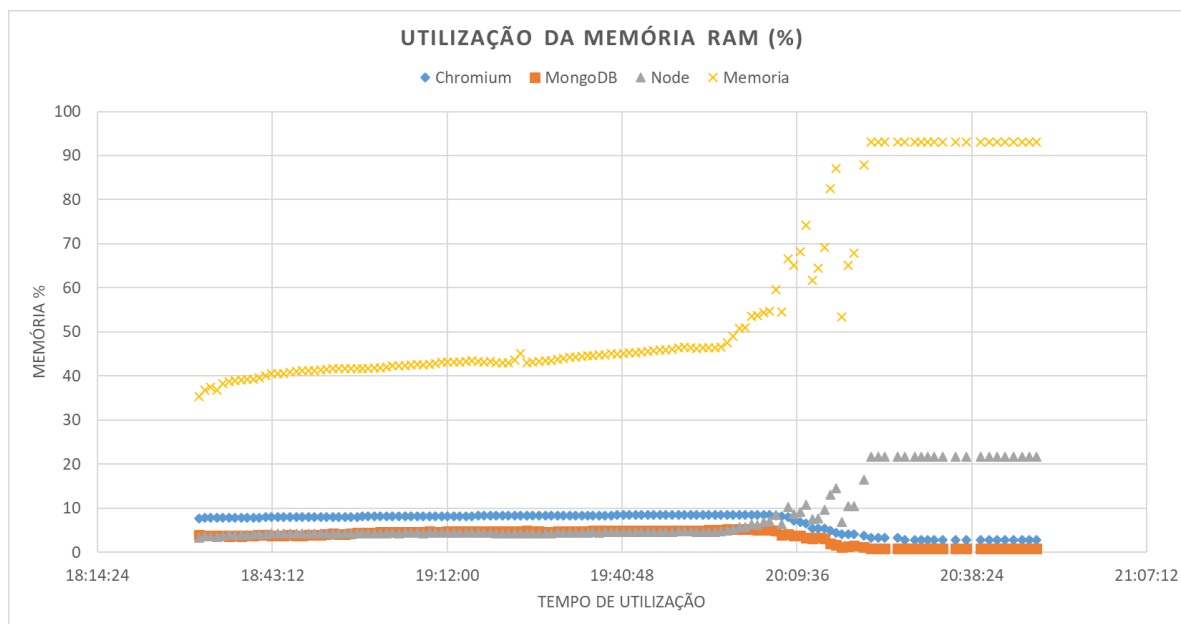


Figura 49 – Percentagem da utilização da memória RAM (%)

Verificamos que após algum tempo de funcionamento da VITABOX (aproximadamente 2 horas), a memória RAM deixa de ser suficiente para as várias solicitações e a memória Swap do sistema operativo fica esgotada (ver Figura 50 e Figura 51).

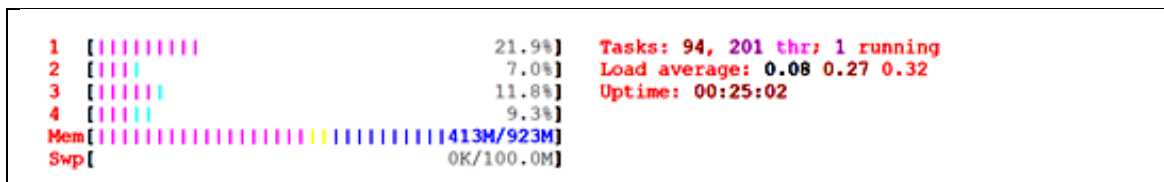


Figura 50 - Imagem retirada do equipamento ao fim de 25 minutos de funcionamento

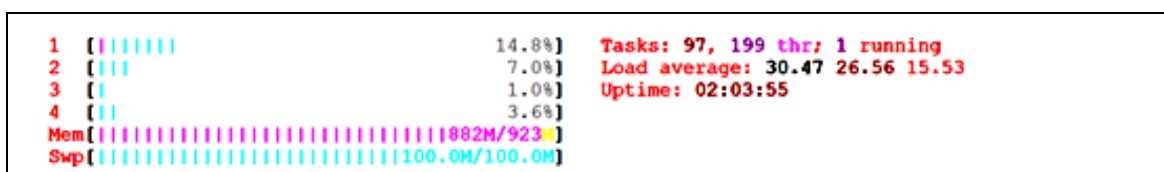


Figura 51 - Imagem retirada do equipamento ao fim de 2 horas de funcionamento

Como resultado, ao fim de algum tempo de funcionamento do microcomputador o mesmo deixava de responder aos pedidos solicitados pela interface gráfica.

5.1.2 Teste – 2º Objeto de armazenamento de dados

Na secção 4.6.1.2 foi repensada a forma de armazenamento dos dados e voltámos a submeter o sistema VITABOX ao teste de carga utilizando o mesmo cenário apresentado no ponto anterior.

Pela análise dos gráficos da utilização do CPU (ver Figura 52) e da utilização da memória RAM (ver Figura 53), verifica-se que todo sistema VITABOX se mostrou estável, apesar da utilização da memória RAM apresentar alguns “picos”.

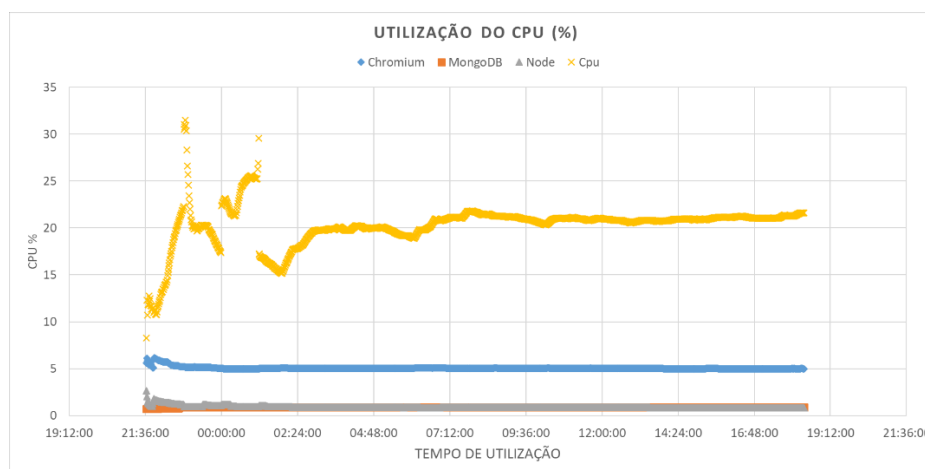


Figura 52 - Percentagem da utilização do CPU (%)

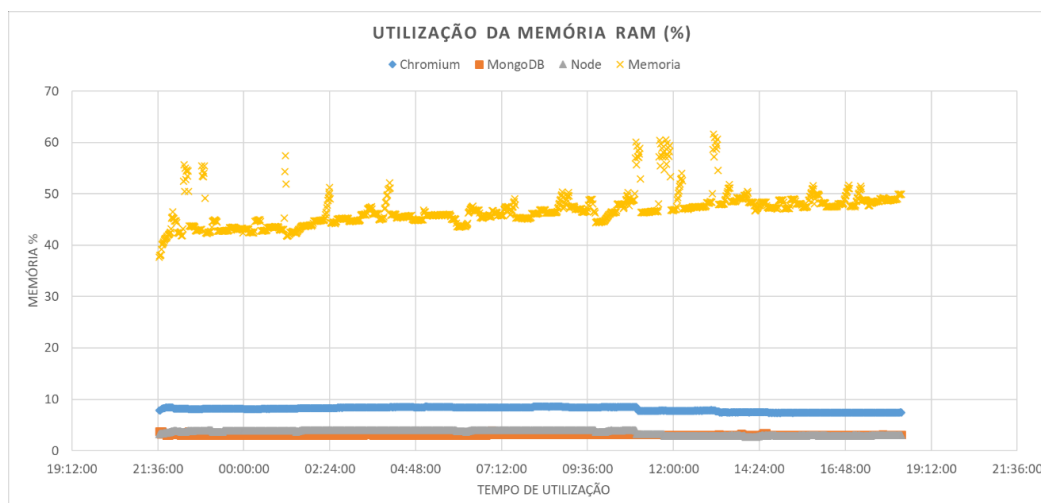


Figura 53 - Percentagem da utilização da memória RAM (%)

A memória RAM não mostrou muita variação, não sendo necessário a utilização da memória swap (ver Figura 54 e Figura 55). Conclui-se que a alteração na forma de armazenamento foi um bom progresso para a estabilização do funcionamento do sistema VITABOX.

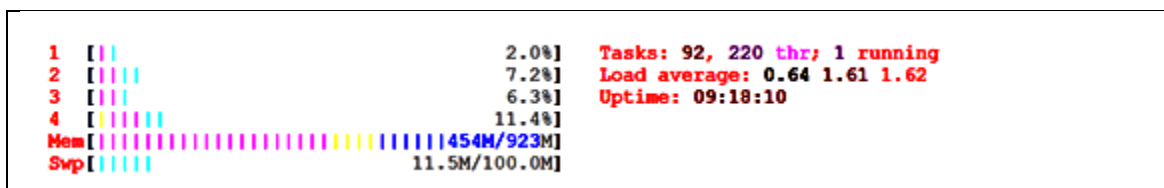


Figura 54 - Imagem retirada do equipamento ao fim de 9 horas de funcionamento

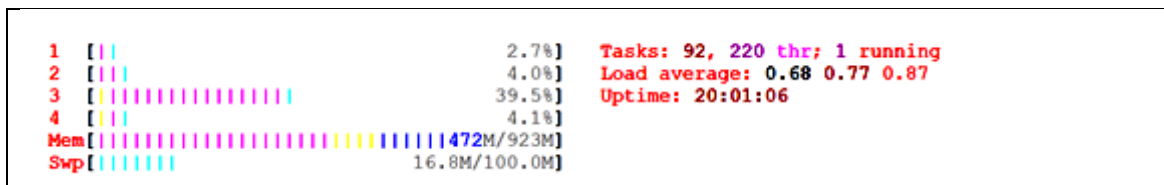


Figura 55 - Imagem retirada do equipamento ao fim de 20 horas de funcionamento

5.2 Testes na utilização de animações na interface gráfica

Para este teste foi utilizado em conjunto com o gerador de dados apresentado na seção 5.1.2 a utilização dos equipamentos biométricos.

Na interface gráfica, de forma a realçar os resultados obtidos dos sensores biométricos, entendeu-se que a utilização de animações seria uma boa forma de evidenciar a obtenção desses dados.

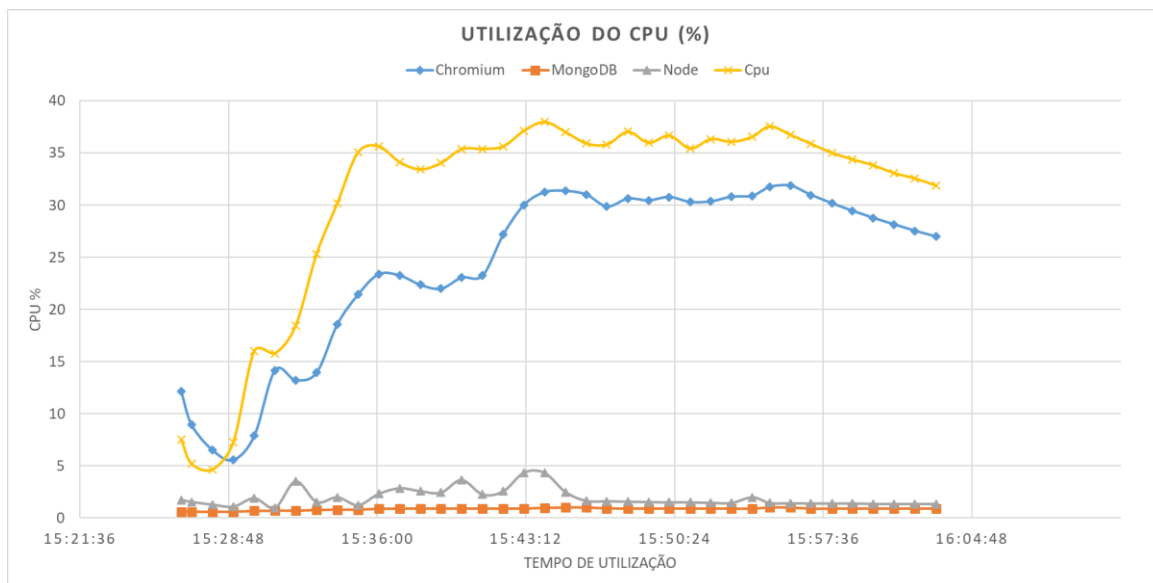


Figura 56 - Percentagem da utilização do CPU (%) com animações

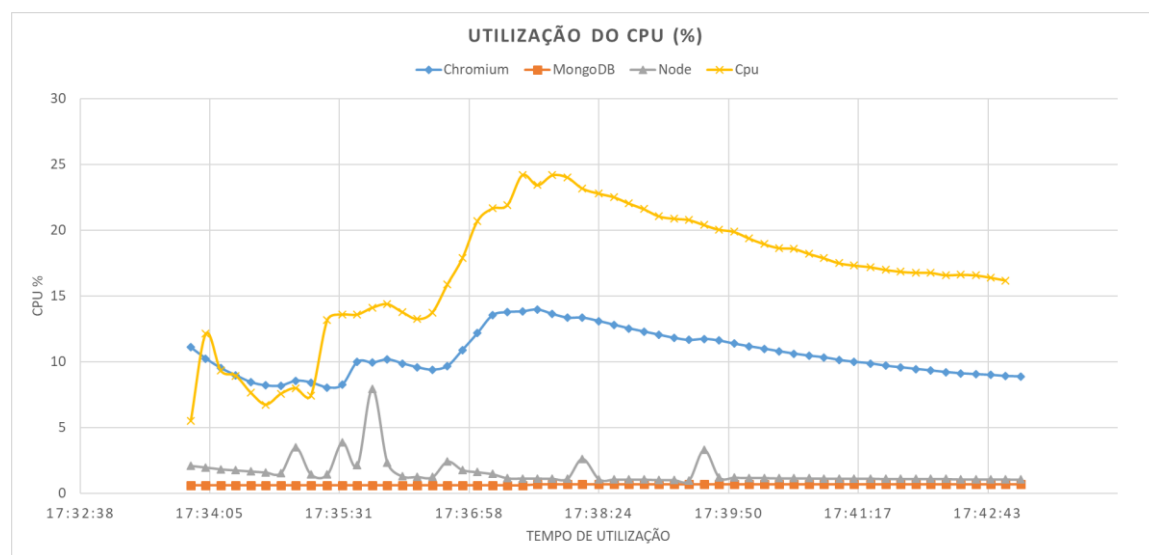


Figura 57 - Percentagem da utilização do CPU (%) sem animações

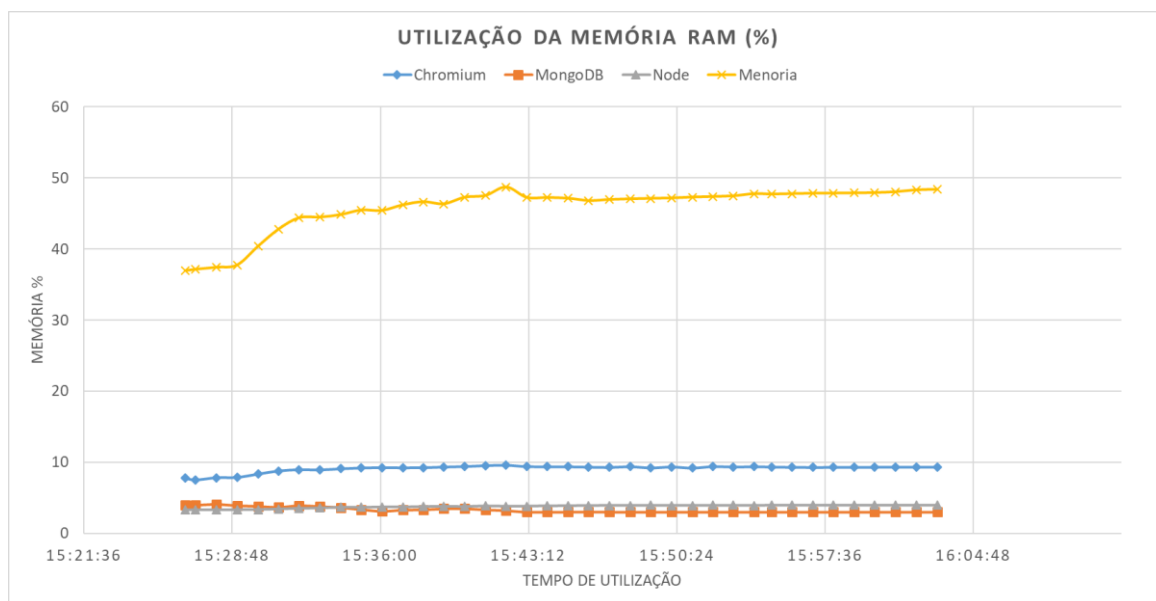


Figura 58 - Percentagem da utilização da Memória RAM (%) com animações

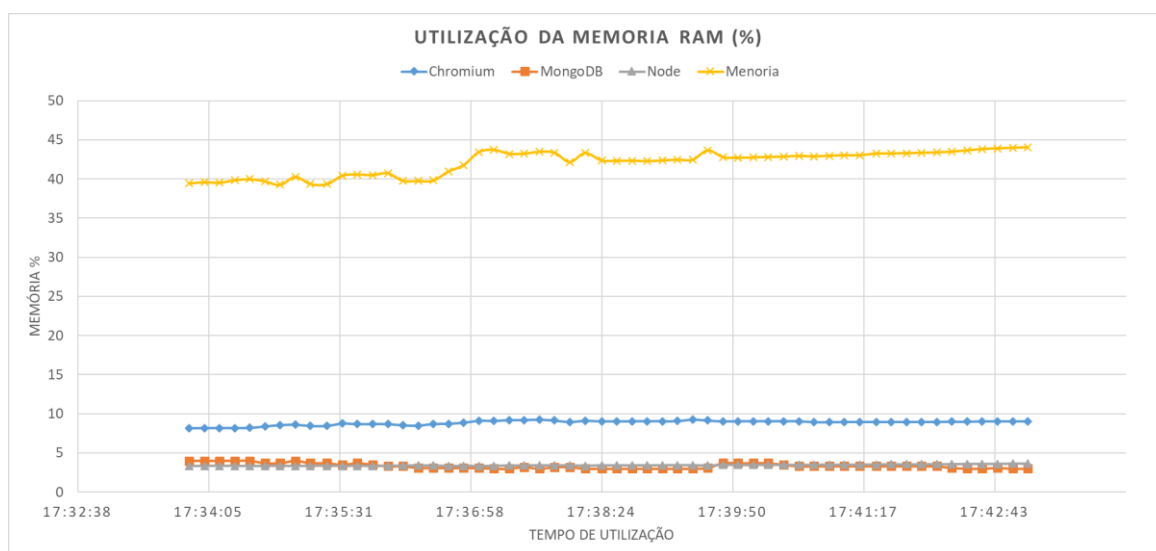


Figura 59 - Percentagem da utilização do Memória RAM (%) sem animações

Comparando os gráficos da utilização do CPU com animações (ver Figura 56) e sem animações (ver Figura 57), verifica-se que em ambos os gráficos existe um “pico” inicial que se deve ao carregamento da interface gráfica. Na Figura 56 verificam-se diversos “picos” que são originados aquando de cada apresentação de resultados (ver Figura 60) ao utilizador utilizando animações. Na utilização da memória RAM (ver Figura 58 e Figura 59) não se verifica alteração com a utilização de animações.

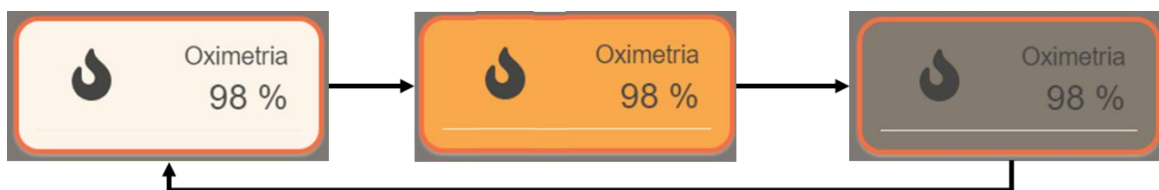


Figura 60 - Sequência da animação na apresentação dos resultados

De forma a ultrapassar estes “picos” no processamento foram realizadas alterações na forma de apresentação dos resultados, tornando perceptíveis para o utilizador eliminando o processamento desnecessário (ver Figura 61).

Considerámos que a alocação de recursos para realçar a apresentação dos resultados através de transições não era necessária pois conseguimos realçar os mesmos sem quaisquer transições, através de realce gráfico (ver Figura 61). Esses recursos computacionais podem assim ser utilizados em situações mais exigentes.

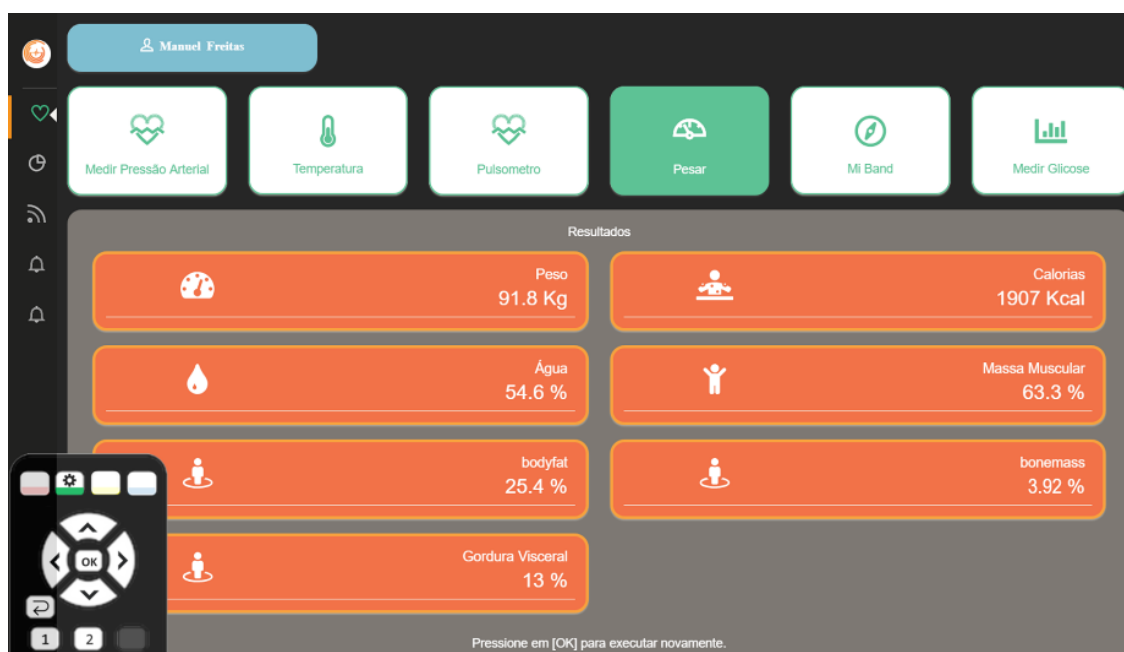


Figura 61 – Apresentação de resultados sem animações

5.3 Testes de usabilidade

Para a realização dos testes de usabilidade, oito participantes testaram a interação com o sistema (ver Tabela 9). A amostra é heterogénea, havendo participantes de idade compreendida entre 22 e 66 anos, de ambos os sexos e diferentes níveis de escolaridade. Todos os utilizadores tomaram contacto com o sistema VITABOX, interface, equipamentos biométricos, etc., pela primeira vez.

Foi criado um manual de procedimentos para a realização de várias tarefas (ver Anexo 4). Durante a realização das várias tarefas foram contabilizados manualmente os tempos de execução de cada tarefa e a quantidade de erros que ocorreram durante a sua realização. Após a conclusão de todo o procedimento foi solicitado aos participantes a realização de um pequeno questionário de satisfação (ver Anexo 5) relativamente à sua experiência de interação na realização das diferentes tarefas.

Participantes	Idade	Sexo	Utiliza Óculos	Escolaridade
P1	38	Masculino	Sim	Ensino Superior
P2	63	Feminino	Sim	Ensino Básico
P3	22	Feminino	Não	Ensino Superior
P4	52	Feminino	Não	Ensino Básico
P5	26	Masculino	Sim	Ensino Superior
P6	25	Feminino	Sim	Ensino Superior
P7	59	Feminino	Não	Ensino Básico
P8	66	Feminino	Não	Ensino Básico

Tabela 9 – Caracterização dos participantes que realizaram testes de usabilidade

Antes de iniciar as tarefas foi explicado o funcionamento do sistema e fornecido para leitura todo o procedimento para a realização das várias tarefas (ver Anexo 4). Comparando o tempo de realização das várias tarefas entre os diferentes utilizadores verifica-se que os tempos se encontram muito próximos (ver Figura 62).

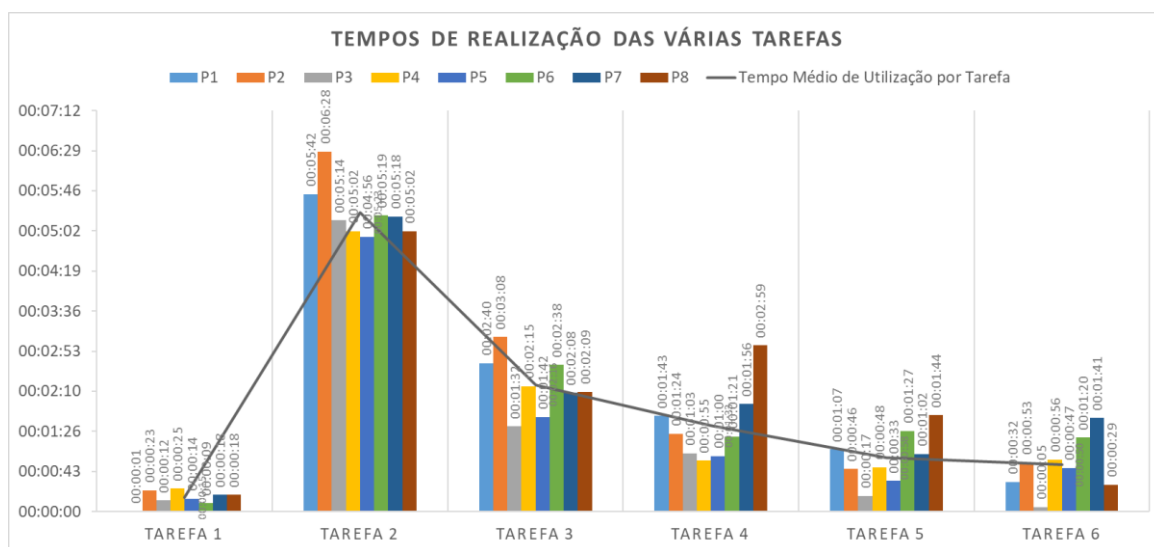


Figura 62 - Tempos de realização dos diferentes testes

O tempo médio máximo foi de 2 minutos e 10 segundos obtido pelo participante P2, um dos utilizadores com mais idade da amostra, e o tempo médio mínimo foi de 1 minuto 24 segundos obtido pelo participante P3, utilizador mais novo da amostra de participantes (ver

Figura 63). Os tempos foram homogêneos, tendo havido, no entanto uma diferença entre o valor médio máximo e o valor médio mínimo de 46 segundos.

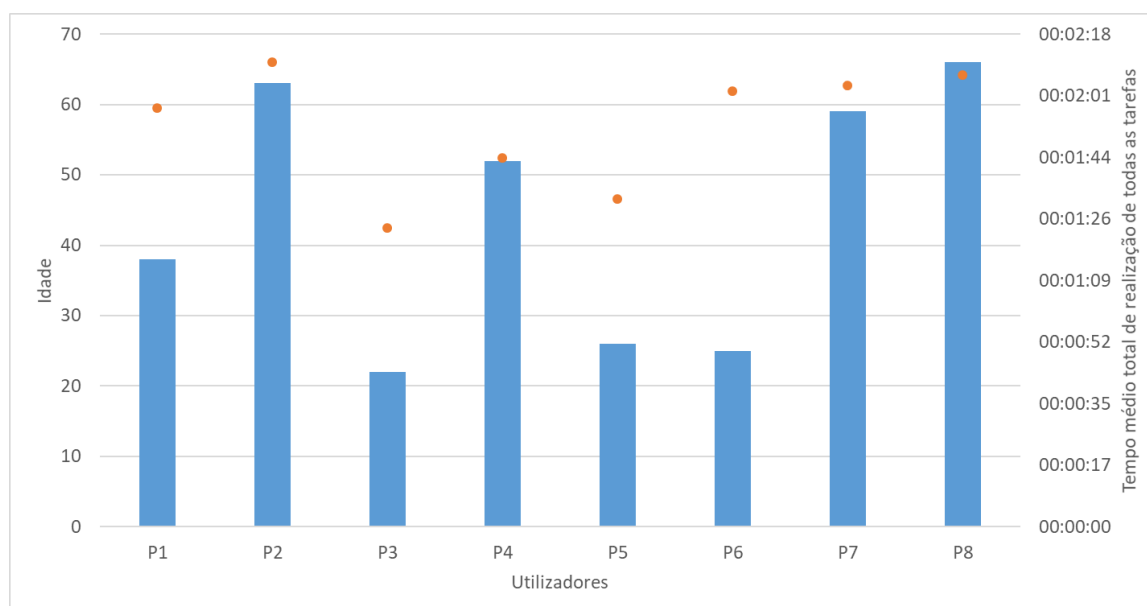


Figura 63 - Relação entre o tempo médio total em relação comm a idade do utilizador

O número de erros obtidos durante a realização das tarefas é mostrado na Tabela 10. Foi considerado como erro sempre que o utilizador pressionasse uma tecla diferente da ordem de execução da tarefa. Verificamos que a ocorrência de erros se encontra com uma maior incidência no decorrer da Tarefa 2.

Nome	Tarefa 1	Tarefa 2	Tarefa 3	Tarefa 4	Tarefa 5	Tarefa 6
P 1	0	2	0	0	0	0
P2	0	1	0	0	0	0
P3	0	0	0	0	0	0
P4	0	2	0	0	0	0
P5	0	2	1	0	0	0
P6	0	1	0	0	0	0
P7	0	0	0	0	0	0
p8	0	0	0	4	0	0

Tabela 10 - Quantidade de erros cometidos por cada utilizador por tarefa

Em conversa com os participantes, estes indicaram que a situação se deveu ao fato de ser o primeiro contacto com a aplicação, passando depois a ser intuitiva. A diminuição de erros ao longo das tarefas mostra que a curva de aprendizagem é rápida o que se deve à utilização sempre das mesmas teclas para as mesmas operações nas diferentes visualizações disponibilizadas pela aplicação.

Analisando o gráfico apresentado na Figura 64 podemos verificar que com a introdução da linha de tendência verifica-se a existência de uma correlação entre a idade dos participantes e o tempo médio da realização total das várias tarefas.

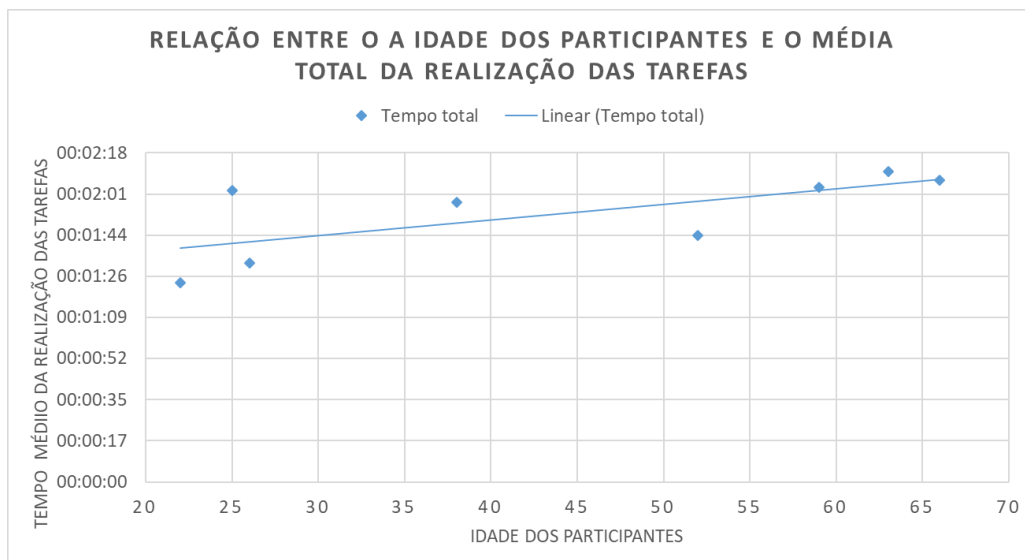


Figura 64 – Relação da idade dos participantes e o tempo médio da realização das tarefas

Analisando o gráfico da Figura 65 gerado a partir da Tabela 11 referente ao grau de satisfação dos utilizadores obtido a partir das respostas ao questionário relacionado à realização dos exames biométricos, verificamos que os participantes consideram a audio-descrição como pouco relevante para a aplicação, enquanto que a informação do resultado do exame é considerada importante. Os resultados das restantes perguntas obtiveram um resultado próximo do máximo.

Análise da execução dos exames	
Questão efetuado ao participante	Média das respostas
Acesso ao utilizador:	4,3
Acesso ao teste:	4,4
Compreensão da informação:	4,3
Tamanho da letra:	4,3
Cor:	4,5
Disposição da informação:	4,3
Utilidade da informação:	4,5
Realização do teste:	4,5
Tempo do teste:	4,6
Informação do resultado do teste:	4,6
Audio-descrição:	3,6

Tabela 11 - Tabela com a média das respostas da realização dos exames dadas pelos participantes



Figura 65 - Gráfico da média do grau de satisfação da realização dos exames

Criando o gráfico (ver Figura 66) a partir da Tabela 12 e o gráfico (ver Figura 67) a partir da Tabela 13 obtemos resultados idênticos aos mostrados anteriormente, onde se volta a verificar que o participantes não acham um mais valia na utilização da audio-descrição dos conteúdos apresentados em cada secção.

Podemos verificar que os participantes dão mais valor à informação apresentada no histórico dos exames do que o histórico dos sensores ambientais, onde em valor médio para a utilidade da informação para o histórico dos exames um valor de 4.4 contra 4.1 da utilidade do histórico dos sensores ambientais.

Análise do Histórico dos Exames	
Questão efetuado ao participante	Média das respostas
Acesso ao utilizador:	4,5
Acesso ao histórico:	4,2
Compreensão da informação:	4,3
Tamanho da letra:	4,3
Cor:	4,3
Disposição da informação:	4,3
Utilidade da informação:	4,4
Audio-descrição:	3,6

Tabela 12 - Tabela com a média das respostas do histórico dos exames dadas pelos participantes

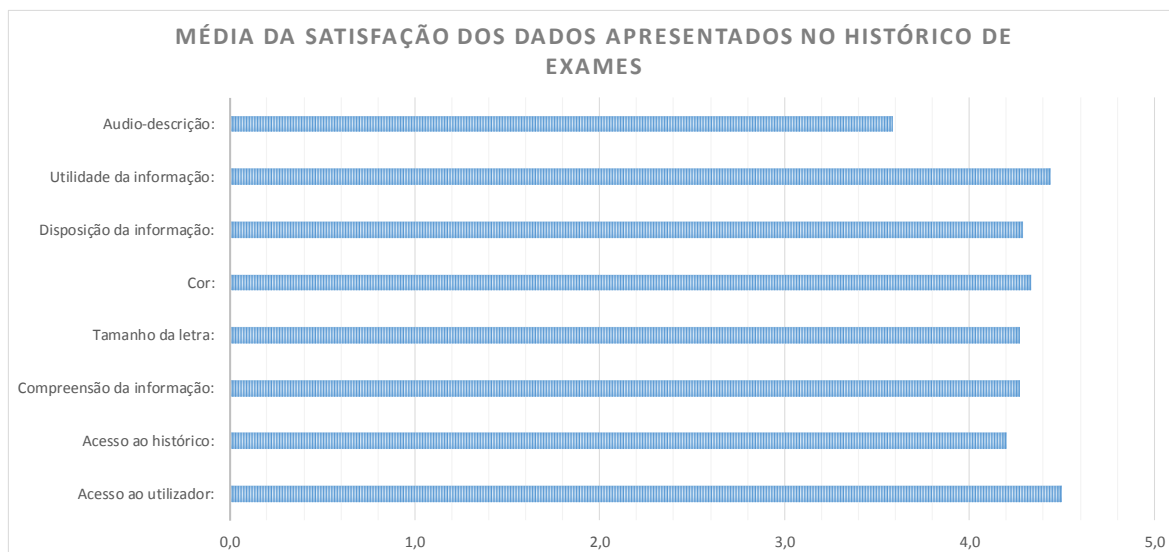


Figura 66 - Gráfico da média do grau de satisfação do histórico dos exames

Análise Histórico dos sensores ambientais	
Questão efetuado ao participante	Média das respostas
Acesso ao histórico:	4,4
Compreensão da informação:	4,3
Tamanho da letra:	4,4
Cor:	4,5
Disposição da informação:	4,2
Utilidade da informação:	4,1
Audio-descrição:	3,7

Tabela 13 - Tabela com a média das respostas do histórico dos sensores ambientais dadas pelos participantes

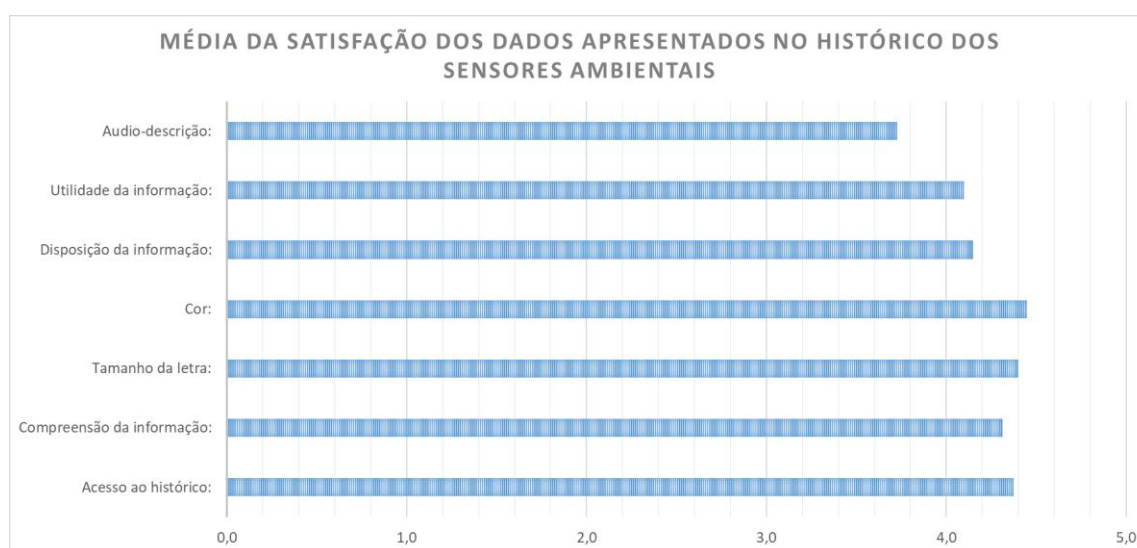


Figura 67 - Gráfico da média do grau de satisfação do histórico dos sensores ambientais

Durante o desenvolvimento da interface do projeto VITABOX, deparamo-nos com pontos que consideramos pertinentes, como por exemplo as cores a serem utilizadas, o tamanho da letra, a utilidade da informação apresentada, utilização do audio-descrição, aos quais precisamos de saber a opinião do utilizador, no questionário (ver Anexo 5) que foi solicitado aos utilizadores onde foram colocadas estas questões para serem analisadas.

No decorrer da execução das tarefas verificou-se um desinteresse na realização da tarefa da navegação do histórico dos sensores ambientais, indicando que a informação apresentada não lhes é relevante. Em relação ao histórico dos exames a informação apresentada é importante pois conseguiam verificar a variação de um determinado parâmetro ao longo das várias realizações do exame, como se pode verificar pela linha de tendência apresentada nos gráficos (ver Figura 68 e Figura 69), onde com o aumento da idade o desinteresse aumenta com uma maior acentuação no histórico dos sensores ambientais.

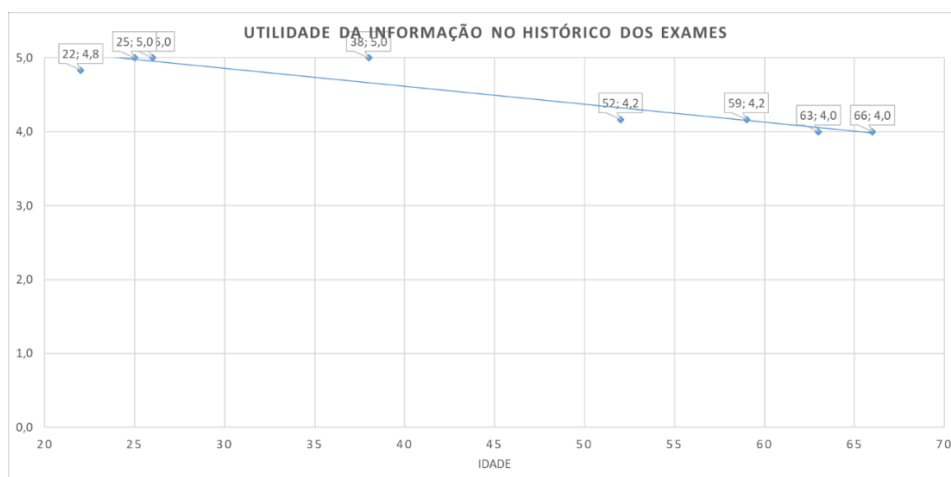


Figura 68 - Gráfico da utilidade da informação do histórico dos exames realizados

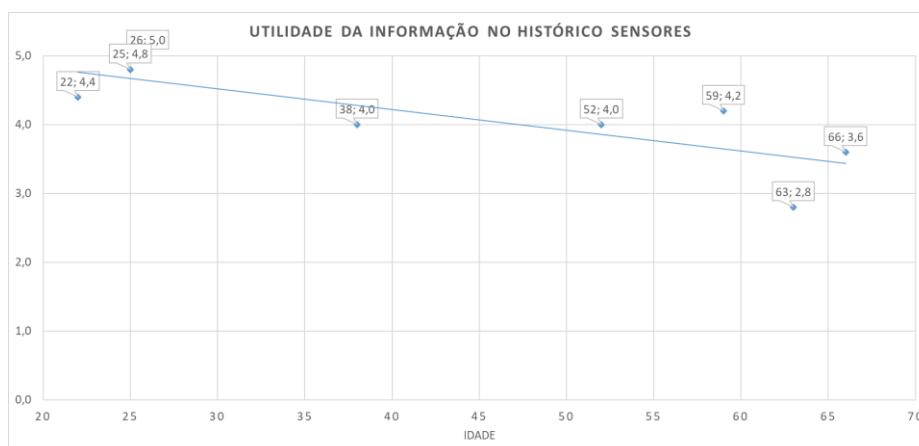


Figura 69 - Gráfico da utilidade da informação do histórico dos sensores ambientais

Como já foi indicado, a utilização da audio-descrição foi uma funcionalidade em alguns casos indicada como desnecessária na realização das diferentes tarefas. Para tentar perceber se existe alguma tendência na utilização da audio-descrição em relação com a idade, foi gerado o gráfico a partir das respostas do questionário referentes à utilização do audio-descrição (ver Figura 70). Verifica-se que não existe uma tendência e de uma forma geral os participantes não consideram uma funcionalidade de muita utilidade.

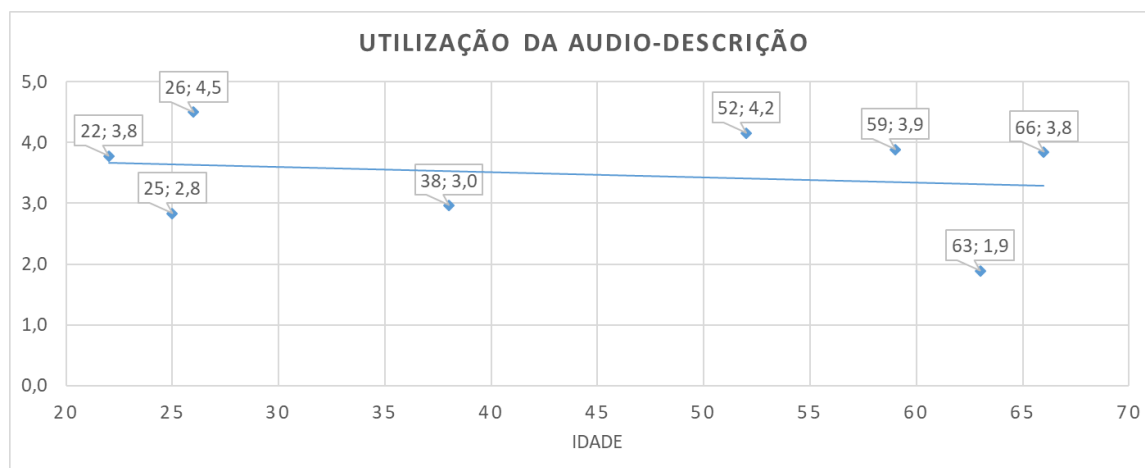


Figura 70 - Gráfico da utilização da audio-descrição na aplicação

Quando os participantes foram confrontados com o preenchimento do questionário (ver Anexo 5) comentaram que se estava a questionar várias vezes a mesmo tema. Neste sentido achamos que o questionário apresentado deveria ser refeito.

Devido ao desenvolvimento da interface gráfica ter sido realizado de uma forma homogênea, ou seja, utilizando sempre as mesmas cores, mesmo tamanho da letra, a disposição de conteúdos iguais na mesma localização, torna inúmeras questões repetidas do questionário desnecessárias. Sugere-se não realizar o questionário por tarefa, mas por tipo de informação, ou seja, realização dos exames, consulta dos históricos disponíveis, exames e sensores ambientais.

Proposta de novo questionário.

1. Analise global da aplicação.
 - Acesso à aplicação
 - Utilidade da aplicação
 - Tamanho da letra
 - Disposição da informação

- Compreensão dos menus
 - Cor dos elementos utilizada
 - Grau de dificuldade de navegação na aplicação
2. Realização dos vários exames
 - Geral
 - Acesso ao separador dos exames
 - Audio-descrição
 - Por cada exame
 - Utilidade do exame
 - Tempo da realização do exame
 - Dificuldade na realização do exame
 3. Consulta do histórico dos exames
 - Acesso ao separador dos exames
 - Audio-descrição
 - Utilidade do histórico dos exames
 4. Consulta do histórico dos sensores ambientais
 - Acesso ao separador dos exames
 - Audio-descrição
 - Utilidade do histórico dos exames
 5. Observações
 6. Caracterização do participante
 - Idade
 - Sexo
 - Escolaridade
 - etc.

6 Conclusões

A evolução da tecnologia tem permitido uma crescente melhoria na qualidade de vida das pessoas mais debilitadas como é o caso das pessoas idosas. Em particular, os sistemas de tele-saúde podem dar um contributo significativo na monitorização e prevenção de algumas doenças que são das maiores causas de morte da atualidade, nomeadamente, doenças cardiovasculares, diabetes, e doenças respiratórias crónicas. No entanto, para que os sistemas de tele-saúde sejam efetivamente utilizados é necessário que estes possam ser usados de forma eficaz e bem aceites pela população alvo. Neste projeto de mestrado desenvolveu-se uma componente de um sistema tele-saúde designada por VITABOX, a qual é responsável pela interação com o utilizador com o sistema através da TV, e pela interação com dispositivos médicos e sensores ambientais. A VITABOX faz parte da solução global VITASENIOR-MT, a qual foi concebida de raiz e se encontra ainda em fase de testes e melhoramento. A solução global tem sido desenvolvida por uma equipa multidisciplinar de investigadores do projeto VITASENIOR-MT.

A VITABOX faz a monitorização de variáveis ambientais através de uma rede de sensores que se encontram nos espaços de maior permanência na habitação, sendo de extrema relevância para a segurança e qualidade de vida dos residentes. Por exemplo, o monóxido de carbono é uma causa de morte em Portugal, principalmente no inverno devido à utilização de braseiras. A solução VITASENIOR-MT permite monitorizar os níveis de monóxido de carbono gerando alertas tanto localmente como para a *cloud*, caso os valores se encontrem em níveis acima do normal. A monitorização de outras variáveis ambientais, como a temperatura ambiente pode permitir a deteção de alterações bruscas causadas por possível foco de incêndio.

A VITABOX interliga-se com uma variedade de equipamentos médicos com comunicação *Bluetooth*. Estes equipamentos requerem processos de utilização simples sendo todo o processo de aquisição dos dados transparente de utilização. Foram testados o medidor da pressão arterial, o oxímetro, a medidor de glucose, a temperatura corporal e a balança. Para além da medição atual é possível visualizar o histórico de medições na VITABOX, possibilitando analisar a variação dos valores ao longo do tempo. A VITABOX permite também a monitorização constante da pulsação do utilizador, através de uma bracelete, desde que esta se encontre dentro do alcance *Bluetooth* do sistema. Esta monitorização permite

avaliar a variação dos valores da pulsação o que poderá ser usado para a detecção de determinadas patologias.

Todas estas operações, utilização dos equipamentos biométricos, visualização do histórico dos sensores biométricos e dos sensores ambientais, visualização dos alertas, estado atual dos sensores ambientais encontra-se disponível na televisão do utilizador à qual se encontra a VITABOX ligada pelo cabo HDMI e toda a navegação na aplicação é conseguida pela utilização do controlo remoto que acompanha a TV.

6.1 Objetivos alcançados

Neste projeto os seguintes objetivos propostos foram concretizados com sucesso:

- Utilização do comando da televisão do utilizador para interagir com a aplicação.
- Utilização de equipamentos biométricos⁴⁵ com interação pela aplicação.
- Armazenamento local e envio para a *cloud* dos resultados dos sensores biométricos⁴⁵.
- Modo automático da utilização de uma pulseira de medição da pulsação.
- Receção de dados dos sensores ambientais⁴⁶, armazenamento local e envio para a *cloud*.
- Detecção de possíveis alertas com base nos valores recebidos dos sensores ambientais⁴⁶, dando origem a avisos localmente na televisão e envio desses alertas para a *cloud*.

Os objetivos seguintes foram parcialmente atingidos:

- Receção de avisos provenientes da *cloud* para serem apresentados na televisão, com indicação da existência de avisos. Encontra-se preparado para a apresentar as mensagens, mas ainda não está concluída a receção das mensagens.
- Audio descrição durante a utilização da aplicação. Necessita de alguma otimização dos sons bem como dos conteúdos a serem reproduzidos.

⁴⁵ Medição da pressão arterial, temperatura corporal, oximetria, glicémia, etc.

⁴⁶ temperatura, humidade, monóxido de carbono, etc.

6.2 Limitações & trabalho futuro

Uma das maiores limitações no desenvolvimento deste projeto foi na definição e na forma de apresentação e interação da informação, cuja utilização será principalmente por pessoas idosas que na maioria dos casos possui baixa destreza na utilização da tecnologia.

O feedback de várias pessoas convidadas para testar o sistema VITASENIOR-MT, algumas delas com experiência no desenvolvimento de sistemas com características similares, permitiu-nos melhorar significativamente a usabilidade do sistema. Não foi, no entanto, possível aplicar todas as sugestões pois obrigariam a uma reestruturação completa da interface.

O microcomputador escolhido é uma arquitetura de 64 *bits*, mas apenas possui 1Gb de memória RAM, o que nos fez optar por um sistema operativo de 32 *bits* limitando-nos na utilização do *browser* (*chromium*) para a disponibilização da interface gráfica. Caso o microcomputador desse a possibilidade de instalar um sistema operativo de 64 *bits* teríamos a possibilidade de utilizar como *browser* o *chrome* que para além de disponibilizar a aplicação também disponibilizaria o motor capaz da reprodução do texto em audio, deixando de ser necessário a criação dos diversos ficheiros de áudio para a áudio-descrição.

Com base nas observações obtidas pelos questionários efetuados aos utilizadores que testaram o sistema, surgiram algumas opiniões que poderemos integrar no futuro. A disponibilização da áudio-descrição com a possibilidade de selecionar a voz, masculino ou feminino, dar a possibilidade ao utilizador de ajustar o tamanho da letra na interface gráfica, criação de uma página de início com a ajuda para a utilização do sistema, exemplo de uma pequena demonstração por intermédio de imagens, e a utilização das teclas com a ação que é espoletada na interface.

Bibliografia

- [1] G. L. Muller, “HTML5 WebSocket protocol and its application to distributed computing,” 11 Setembro 2014. [Online]. Available: <https://pdfs.semanticscholar.org/2556/ee58dce37d0e6db10f3cef4241d67ae8c090.pdf>. [Acedido em Maio 2018].
- [2] L. Srinivasan, J. Scharnagl e K. Schilling, “Analysis of WebSockets as the New Age Protocol for Remote Robot Tele-operation,” 11-13 Novembro 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667015343688>. [Acedido em Maio 2018].
- [3] T. d. S. Côrrea, D. E. C. d. Almeida e A. F. G. Neto, “Comparação entre banco de dados relacional e não relacional em arquitetura distribuída,” Setembro 2017. [Online]. Available: <https://www.inatel.br/biblioteca/pos-seminarios/seminario-de-desenvolvimento-em-soa-com-cloud-computing-e-conectividade/2017-4/9668-comparacao-entre-banco-de-dados-relacional-e-nao-relacional-em-arquitetura-distribuida/file>. [Acedido em Maio 2018].
- [4] Creative Commons - Attribution + Noncommercial © Copyright 2017. EFY Enterprise Pvt. , “Best open source databases for IoT applications,” opensourceforu.com, 26 Maio 2017. [Online]. Available: <https://opensourceforu.com/2017/05/best-open-source-databases-iot-applications/>. [Acedido em Maio 2018].
- [5] Medium, “ReactJS vs Angular5 vs Vue.js—What to choose in 2018?,” [Online]. Available: <https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d>. [Acedido em Junho 2018].
- [6] H. R. Fabião Pires da, “Desenvolvimento de uma rede de sensores para monitorização de idosos,” Setembro 2015. [Online]. Available: <https://core.ac.uk/download/pdf/43584260.pdf>. [Acedido em Janeiro 2018].

- [7] Diário de Notícias, “www.dn.pt,” 2 Março 2015. [Online]. Available: <https://www.dn.pt/portugal/interior/tecnologia-para-monitorizar-saude-de-idosos-testada-com-sucesso-em-coimbra-4429353.html>. [Acedido em Janeiro 2018].
- [8] P. S. D. O. C. A. N. C. Virginie Felizardo, “www.scielo.mec.pt,” 11 Novembro 2014. [Online]. Available: <http://www.scielo.mec.pt/pdf/rist/n14/n14a03.pdf>. [Acedido em Janeiro 2018].
- [9] MySignals, “www.my-signals.com,” 2017. [Online]. Available: <http://www.my-signals.com/>. [Acedido em Janeiro 2018].
- [10] PROJETO 10 MIL VIDAS, “www.ancs.pt,” Associação Nacional de Cuidado e Saúde, [Online]. Available: <http://www.ancs.pt/projeto-10-mil-vidas---ancs.html>. [Acedido em Janeiro 2018].
- [11] altice, “BR_SMARTAL_ALB_EN.pdf,” [Online]. Available: http://www.alticelabs.com/content/products/BR_SMARTAL_ALB_EN.pdf. [Acedido em Janeiro 2018].
- [12] S. Spinsante e E. Gambi, “Remote health monitoring for elderly through interactive television,” 21 Agosto 2012. [Online]. Available: <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-11-54>. [Acedido em Janeiro 2018].
- [13] L. P. C. P. C. E. I. B. S. I. I. S. C. T. I. T. C. Hitachi, “High-Definition Multimedia Interface - Specification Version 1.4,” Junho 2009. [Online]. Available: http://d1.amobbs.com/bbs_upload782111/files_51/ourdev_716302E34B9Q.pdf. [Acedido em Junho 2018].
- [14] HDMI Licensing, LLC., “High-Definition Multimedia Interface Specification - CEC Specification,” [Online]. Available: <https://pt.scribd.com/document/257897241/cec-specification>. [Acedido em Junho 2018].
- [15] C. Birgen, “ADVANCED PROCESS SIMULATION,” 8 Dezembro 2014. [Online]. Available:

- http://folk.ntnu.no/preisig/HAP_Specials/AdvancedSimulation_files/2014/AdvSim-2014__Birgen_Cansu_Databases.pdf. [Acedido em Junho 2018].
- [16] “SQL vs NoSQL, qual usar?,” TreinaWeb, 19 Junho 2017. [Online]. Available: <https://www.treinaweb.com.br/blog/sql-vs-nosql-qual-usar/>. [Acedido em Junho 2018].
- [17] C. Buckler, “SQL Vs NoSQL: The Differences,” 18 Setembro 2015. [Online]. Available: <https://www.sitepoint.com/sql-vs-nosql-differences/>. [Acedido em Junho 2018].
- [18] Y. Li e S. Manoharan, “A performance comparison of SQL and NoSQL databases,” 10 Outubro 2013. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6625441/>. [Acedido em Junho 2018].
- [19] B. F. Lóscio, H. R. d. Oliveira e J. é. d. S. Pontes, “NoSQL no desenvolvimento de aplicações Web,” [Online]. Available: http://www.addlabs.uff.br/sbsc_site/SBSC2011_NoSQL.pdf.
- [20] R. Cattell, “Scalable SQL and NoSQL data stores,” 6 Maio 2011. [Online]. Available: <https://dl.acm.org/citation.cfm?id=1978919>. [Acedido em Junho 2018].
- [21] wfl.com/, “apache-vs-nginx-vs-node-js-and-what-it-means-about-the-performance-of-wordpress-vs-ghost,” 29 Novembro 2017. [Online]. Available: <https://iwf1.com/apache-vs-nginx-vs-node-js-and-what-it-means-about-the-performance-of-wordpress-vs-ghost>. [Acedido em Junho 2018].
- [22] I. K. C.-I. D. K. D. Tselikas, “Is Node.js a viable option for building modern web applications? A performance evaluation study,” <http://www.scielo.br>, 28 Abril 2014. [Online]. Available: <http://www.scielo.br/pdf/ca/v23n1/a04v23n1.pdf>. [Acedido em Junho 2018].
- [23] R. Schroeder¹ e F. d. Santos, “ARQUITETURA E TESTES DE SERVIÇOS WEB DE ALTO DESEMPENHO COM NODE.JS E MONGODB,” [Online]. Available:

http://www.ceavi.udesc.br/arquivos/id_submenu/787/ricardo_schroeder_versao_final_.pdf. [Acedido em Junho 2018].

- [24] Udacity, “React: o que é e como funciona essa ferramenta?,” <https://br.udacity.com>, [Online]. Available: <https://br.udacity.com/blog/post/react-o-que-e-como-funciona>. [Acedido em Junho 2018].
- [25] K. Zonneveld, “crontab,” kvx.io, 29 Julho 2007. [Online]. Available: <https://kvz.io/blog/2007/07/29/schedule-tasks-on-linux-using-crontab/>. [Acedido em Maio 2018].

Anexo 1 *Script* para captura automática de dados da banda biométrica

```
'use strict'
require('colors');
var cp = require('child_process'),
    fs = require('fs'),
    connectDB = require('./lib/connectDB.js'),
    patientslib = require('./lib/patientslib.js'),
    rawsensorlib = require('./lib/rawsensorlib.js'),
    log = require('./lib/logger'),
    errorLog = log.errorlog,
    successlog = log.successlog,
    config = null,
    mode = process.env.NODE_ENV || "DEV";

var Main = function () {
    let pathCfg = './config-' + mode.toLowerCase() + '/config.js';
    successlog.info('Path Config -', pathCfg);
    config = require(pathCfg);
    this.mongodbConfig = config.mongodb;
    this.allPatientes = [];
};

Main.prototype.start = function () {
    var self = this;
    connectDB.connectDB(this.mongodbConfig, function () {
        successlog.info(`Data Base is up.`);
        patientslib.getAllPatientsIdMiBandDevice(function (patients) {
            self.allPatientes = patients;
            if (self.allPatientes.length > 0) {
                self.execBleMiBand(self.allPatientes.pop());
            } else {
                console.log("process End");
                process.exit(0);
            }
        })
    })
};

Main.prototype.execBleMiBand = function (pacientInfo) {
    var self = this;
    var args = {
        options: {
            bandfitness_auth: pacientInfo.bandfitness_auth
        },
        patientId: pacientInfo.userId,
        action: pacientInfo.BoardmodelName,
        macAddress: pacientInfo.mac_addr,
        boardId: pacientInfo.boardId,
        deviceId: pacientInfo.deviceId,
        BoardmodelId: pacientInfo.BoardmodelId,
        sensors: pacientInfo.Sensors
    };
    // inicia p script e envia as configuracoes do ficheiro inicial

```

```
var child = cp.fork('./lib/bleServer.js');
child.send({ "serverdata": args });
child.on('message', function (data) {
  if (data.proc === 'saveDataSensors') {
    rawsensorlib.insertManyData(data.dataSend);
  } else if (data.proc === 'saveAuthPatient') {
    patientslib.updateFlagAuthBandFit(data.dataSend);
  }
});
child.on('exit', function (data) {
  console.log("process End");
  process.exit(0);
})
}

var m = new Main();
m.start();

module.exports = Main;
```

Anexo 2 *Script para execução no crontab*

```
#!/bin/bash
COMMON="/home/pi/Desktop/VitaBox"
OUTLOGFILE="$COMMON/Scripts/mibandstdout.log"
ERRLOGFILE="$COMMON/Scripts/mibandstderr.log"
rm "$OUTLOGFILE"
rm "$ERRLOGFILE"
cd /home/pi/Desktop/VitaBox
sudo NODE_ENV=prod node startAutoMiBand.js >> "$OUTLOGFILE" 2>>
"$ERRLOGFILE"
```


Anexo 3 Função para criar *array* de ficheiros de audio de um número

```

/**
 * Constrói array
 * Ex. Resultado
 * array[0] - './caminho/100.mp3',
 * array[1] - './caminho/e.mp3',
 * array[2] - './caminho/20.mp3',
 * array[3] - './caminho/e.mp3',
 * array[4] - './caminho/3.mp3',
 * @param array - cada posição corresponde ao caminho e ficheiro
de audio
 * @param num - número a ser reproduzido
 */
function prossNums(array, num) {
  // verifica se existe o número se sim obtém o tamanho do
número ex. 123 = 3
  var expNum = num ? num.length : null;
  // Se existir o número
  if (num && expNum) {
    // se o número for constituído por uma casa
    // (0 - 9) ou inferior a 20 (0 - 19)
    if (expNum === 1 || parseInt(num) < 20) {
      // constrói o caminho e adiciona a extensão
      // ex. /caminho para o ficheiro/(0 - 19).mp3
      array.push(filePathName('numbers', num));
      // Se o número possuir mais de duas casas
    } else {
      // loop com limite da quantidade de casas
      for (let i = 0; i < expNum; i++) {
        // Se a quantidade de casas menos o valor do atual do
loop seja = 2
        // & a quantidade de casas superior a 2
        // & o valor atual do número não seja 0
        // ex. não pode ser 10, 20, 30 ...
        // Ex. número a tratar 120 - 3 casas e quando vai na
segunda casa (2) não é preciso colocar o (e.mp3)
        if (((expNum - i) === 2 && expNum > 2) &&
          parseInt(num[i]) !== 0) ||
          // OU se for na última posição do número também não é
para levar (e.mp3)
          (expNum - i) === 1 && parseInt(num[i]) !== 0) {
          array.push(filePathName('numbers', 'e'));
        }
        // Valor atual não é (0)
      }
    }
  }
}

```

```
        if (parseInt(num[i]) !== 0) {
            // caso em que o número é 100 - ficheiro de audio
(cem.mp3)
            if (parseInt(num) === 100) {
                // constrói o caminho e adiciona a extensão
                // ex. /caminho para o ficheiro/cem.mp3
                array.push(filePathName('numbers', 'cem'));
            } else {
                // constrói o caminho e adiciona a extensão
                // ex. /caminho para o ficheiro/(número atual).mp3
                // (número atual) - resultado da operação
                // num[i] * Math.pow(10, (expNum - i - 1))
                // 2 * 10^(3(total de casas) - i(posição atual do
loop) - 1)
                array.push(filePathName('numbers', num[i] *
Math.pow(10, (expNum - i - 1))));
            }
        }
    }
}
}
}
}

// função para criar o caminho e o nome do ficheiro com a
extensão
var filePathName = function (type, name) {
    return './sounds/' + type + '/' + name + '.mp3';
}
```

Anexo 4 Procedimento para a realização das tarefas

Procedimento para a realização das Tarefas VITASENIOR

Tarefas a realizar ao Sistema de Interface TV - VITABOX

O projeto VITASENIOR incorpora uma tarefa final de integração de sistema e validação experimental na residência dos participantes selecionados. Pretende-se realizar uma validação experimental com os utilizadores baseada em testes de usabilidade conduzidos em dois níveis diferentes de desenvolvimento do projeto, nomeadamente: durante o processo de design e na fase final do projecto. Os testes realizados ainda durante a fase de design têm como objetivo a realização de ajustes ao protótipo. Por outro lado, os testes realizados na fase final do projeto permitirão avaliar a facilidade e a frequência como que os participantes (idosos e cuidadores) usam o sistema tecnológico final proposto.

O procedimento de teste descrito neste documento refere-se ao teste realizado ao sistema de interface TV – VITABOX realizado durante a fase de design de projeto. O procedimento descreve-se de seguida.

Tarefa 1 Procedimento de teste ao sistema de interface TV – VITABOX

1. Ligar a televisão – A televisão deve estar a trabalhar a passar algum programa televisivo;
2. Para aceder à aplicação pressionar na Tecla 2;

Tarefa 2 - Realização de Exames de diagnóstico da saúde (Medições) do utilizador

3. Navegar no menu com as setas para cima ou para baixo até alcançar o menu das medições;
4. Pressionar a tecla da direita até seleccionar o utilizador pretendido e de seguida pressionar OK;
5. Carregar na seta direita até alcançar a opção de medições pretendida e de seguida carregar em OK;
6. Após seleccionar a medição seguir as instruções do exame;
7. Repetir o procedimento de 5 a 6 para realizar qualquer uma das medições possíveis.
8. Assim que as medições estejam todas concluída carregar na tecla Exit duas vezes para sair para o menu principal;

Tarefa 3 - Utilização do histórico de medições da saúde do utilizador

9. Navegar até à opção histórico de medições usando a seta para baixo;
10. Pressionar a tecla da direita até seleccionar o utilizador pretendido e de seguida pressionar OK;
11. Seleccionar o histórico de medições pretendido utilizando as setas esquerda ou direita;
12. Carregar no botão OK assim que o histórico de medições pretendido esteja seleccionado;
13. Repetir o procedimento de 11 a 12 para consultar qualquer um dos históricos de medições disponíveis;
14. Carregar na tecla Exit duas vezes para voltar ao menu principal;

Tarefa 4 - Utilização do histórico de sensores ambientais

15. Navegar até à opção histórico de sensores ambientais usando as setas para baixo ou para cima;
16. Seleccionar o histórico de sensores ambientais pretendido utilizando as setas esquerda ou direita;
17. Carregar no botão OK assim que o histórico de sensores ambientais pretendido esteja seleccionado;
18. Seleccionar a divisão pretendida com as teclas direita ou esquerda
19. Pressionar ok para visualizar ou esconder histórico do sensor seleccionado
20. Repetir o procedimento de 16 a 19 após pressionar a tecla Exit (uma vez) para poder consultar o histórico de outro sensor;
21. Carregar na tecla Exit duas vezes para voltar ao menu principal;

Tarefa 5 - Consultar avisos

22. Navegar até à opção avisos usando as setas para cima ou para baixo;
23. Navegar até à opção sensores ambientais usando as setas para cima e para baixo para verificar o estado atual dos sensores;

Tarefa 6 - Alterar as configurações

24. Para alterar as configurações disponíveis pressione o botão verde para abrir o menu configurações;

25. Utilize qualquer seta para navegar nas configurações disponíveis e pressione a tecla OK assim que a opção pretendida esteja selecionada;
26. Ao selecionar a opção modo pode optar entre a modo básico e modo avançado utilizando as teclas OK para alterar;
27. Ao selecionar a opção som pode ligar ou desligar o som da aplicação, utilizando a tecla OK para alterar;
28. Ao selecionar a opção linguagem pode selecionar entre português e inglês, utilizando a tecla OK para alterar;
29. Pressionar Exit para sair;
30. Pressionar o botão 1 para voltar para a emissão de TV.

Anexo 5 Questionário de satisfação

Questionário de Testes VITASENIOR

Questionário de satisfação do utilizador relativo à utilização do sistema Interface -TV - VITABOX

Avalie a sua satisfação com o sistema Interface TV - VITABOX, demonstrando a sua perceção em relação aos itens apresentados em baixo, para cada uma das tarefas realizadas durante o teste ao sistema.

- Marque o número que melhor reflete as suas impressões no uso do sistema.
- Para itens que não são aplicáveis, use N/A.

Tarefa 1

Como avalia globalmente aplicação do sistema Interface TV-VITABOX

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso à aplicação:		☐	☐	☐	☐	☐		☐
Compreensão dos menus:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐

Tarefa 2 Realização de testes de diagnóstico

Como avalia o menu de testes de diagnóstico

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao utilizador:		☐	☐	☐	☐	☐		☐
Compreensão do menu:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Navegação nos submenus:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Como avalia o teste de oximetria

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao teste:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Realização do teste:		☐	☐	☐	☐	☐		☐
Tempo do teste:		☐	☐	☐	☐	☐		☐
Informação do resultado do teste:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Como avalia o teste de temperatura

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao teste:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Realização do teste:		☐	☐	☐	☐	☐		☐
Tempo do teste:		☐	☐	☐	☐	☐		☐
Informação do resultado do teste:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Como avalia o teste de Bracelete

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao teste:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐

Como avalia o teste de peso

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao teste:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐

Cor:	☐ ☐ ☐ ☐ ☐ ☐	☐	Cor:	☐ ☐ ☐ ☐ ☐ ☐	☐
Disposição da informação:	☐ ☐ ☐ ☐ ☐ ☐	☐	Disposição da informação:	☐ ☐ ☐ ☐ ☐ ☐	☐
Utilidade da informação:	☐ ☐ ☐ ☐ ☐ ☐	☐	Utilidade da informação:	☐ ☐ ☐ ☐ ☐ ☐	☐
Realização do teste:	☐ ☐ ☐ ☐ ☐ ☐	☐	Realização do teste:	☐ ☐ ☐ ☐ ☐ ☐	☐
Tempo do teste:	☐ ☐ ☐ ☐ ☐ ☐	☐	Tempo do teste:	☐ ☐ ☐ ☐ ☐ ☐	☐
Informação do resultado do teste:	☐ ☐ ☐ ☐ ☐ ☐	☐	Informação do resultado do teste:	☐ ☐ ☐ ☐ ☐ ☐	☐
Audio-descrição:	☐ ☐ ☐ ☐ ☐ ☐	☐	Audio-descrição:	☐ ☐ ☐ ☐ ☐ ☐	☐

Como avalia o teste de pressão arterial

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao teste:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Realização do teste:		☐	☐	☐	☐	☐		☐
Tempo do teste:		☐	☐	☐	☐	☐		☐
Informação do resultado do teste:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Tarefa 3 Consulta do histórico dos testes de diagnóstico**Como avalia o menu de histórico de testes de diagnóstico**

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao utilizador:		☐	☐	☐	☐	☐		☐
Compreensão do menu:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Navegação nos submenus:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Como avalia o histórico de oximetria

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao histórico:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Como avalia o histórico de temperatura

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao histórico:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Como avalia o histórico de bracelete

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao histórico:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐

Como avalia o histórico de peso

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao histórico:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐

Cor:	☐ ☐ ☐ ☐ ☐	☐	Cor:	☐ ☐ ☐ ☐ ☐	☐
Disposição da informação:	☐ ☐ ☐ ☐ ☐	☐	Disposição da informação:	☐ ☐ ☐ ☐ ☐	☐
Utilidade da informação:	☐ ☐ ☐ ☐ ☐	☐	Utilidade da informação:	☐ ☐ ☐ ☐ ☐	☐
Audio-descrição:	☐ ☐ ☐ ☐ ☐	☐	Audio-descrição:	☐ ☐ ☐ ☐ ☐	☐

Como avalia o histórico de pressão arterial

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao histórico:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Tarefa 4 Consulta do histórico dos sensores ambientais**Como avalia o menu de histórico de sensores ambientais**

	Difícil	1	2	3	4	5	Fácil	N/A
Compreensão do menu:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Navegação nos submenus:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Como avalia o histórico de temperatura

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao histórico:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Como avalia o histórico de humidade

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao histórico:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Como avalia o histórico de CO2

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao histórico:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Como avalia o histórico de CO

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso ao histórico:		☐	☐	☐	☐	☐		☐
Compreensão da informação:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Utilidade da informação:		☐	☐	☐	☐	☐		☐
Audio-descrição:		☐	☐	☐	☐	☐		☐

Tarefa 5 -Avisos**Como avalia o menu avisos**

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso aos avisos:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐

Cor:	☐ ☐ ☐ ☐ ☐ ☐	☐
Disposição da informação:	☐ ☐ ☐ ☐ ☐ ☐	☐
Utilidade da informação:	☐ ☐ ☐ ☐ ☐ ☐	☐
Audio-descrição:	☐ ☐ ☐ ☐ ☐ ☐	☐

Tarefa 6 –Alteração das configurações

Como avalia o menu de configurações

	Difícil	1	2	3	4	5	Fácil	N/A
Acesso às configurações:		☐	☐	☐	☐	☐		☐
Tamanho da letra:		☐	☐	☐	☐	☐		☐
Cor:		☐	☐	☐	☐	☐		☐
Disposição da informação:		☐	☐	☐	☐	☐		☐
Configurações disponíveis		☐	☐	☐	☐	☐		☐
Alteração das configurações:		☐	☐	☐	☐	☐		☐
Navegação das configurações:		☐	☐	☐	☐	☐		☐

Avaliação qualitativa

Faça uma avaliação qualitativa da Interface TV-VITABOX e indique sugestões de melhoria:

Dados Gerais

	1	2	3	4	5	6	7	8	9	0		Masculino	Feminino
Idade:	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	Sexo:	☐	☐
	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐			
Utiliza olhos:			Sim				Não				Sabe ler:	Sim	Não
			☐				☐					☐	☐
Dificuldade em ouvir:			Sim				Não				Utiliza aparelho auditivo:	Sim	Não
			☐				☐					☐	☐