



Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

**Desenvolvimento de sistemas de interação
audiovisual em recetores de TV para
assistência a idosos**

Dário José Marques Jorge

Relatório de Projeto

Trabalho Final de Mestrado

Orientado por:

Professor Pedro Correia, Instituto Politécnico de Tomar
Professor Luís Agnelo de Almeida, Instituto Politécnico de Tomar

Relatório de Projeto apresentado ao Instituto Politécnico de Tomar
para cumprimento dos requisitos necessários à obtenção do grau de
Mestre em Produção de Conteúdos Digitais

RESUMO

Este relatório descreve o trabalho desenvolvido no âmbito do projeto VITASENIOR-MT, uma solução de teleassistência para idosos, com interação baseada em TV com a utilização do comando existente. A solução de interação proposta, pretende colmatar a resistência que a população idosa tem às novas tecnologias, utilizando um equipamento com o qual estão familiarizados.

No âmbito do projeto VITASENIOR-MT, foram desenvolvidos vários subprojectos, desde, a VITABOX, um sistema baseado em microcomputador que faz a gestão de todo o sistema em casa do utilizador, a interação com TV com comando, a ligação de uma rede de sensores ambientais, a ligação automática de sensores biométricos e a ligação com os serviços *cloud* que dão suporte a todo o sistema.

Sendo um projeto com várias componentes e em colaboração com várias pessoas e entidades, são apresentadas as contribuições individuais para o seu desenvolvimento, nomeadamente a implementação de interfaces de interação com o utilizador usando o comando da TV, nos seus modos básico e avançado, incluindo as interfaces de configurações, de informações de alertas, da página de boas-vindas com informações úteis e notificações ao utilizador, a ligação automática de *wifi*, assim como de um serviço de áudio-descrição em modo *offline*. Adicionalmente, foi implementado o envio e receção de dados e de notificações da VITABOX para a *cloud* e vice-versa, assim como a gestão e troca de dados entre a VITABOX e a rede de sensores ambientais.

O trabalho apresentado e os resultados obtidos contribuíram para uma versão estável do sistema, pronta a ser testada e validada em ambiente real.

Palavras-chave: Áudio-descrição, CEC, *Cloud*, HDMI, Teleassistência, Telesaúde, TV

ABSTRACT

This report describes the work developed in the VITASENIOR-MT project, a teleassistance solution for the elderly people, based on TV interaction using the existing command. The proposed interaction solution aims to overcome the resistance that the elderly population has to deal with the new technologies, using an equipment with which they are familiar.

In the VITASENIOR-MT project, several subprojects were developed, from to VITABOX, a microcomputer-based system that manages the entire system in the user's home, the interaction with TV with command, the connection of a network of environmental sensors, the automatic connection of biometric sensors and the connection with the cloud services that support the whole system.

Being a multi-component project with collaboration of several people and entities, individual contributions to its development are presented, namely the implementation of user interaction interfaces using the TV command with basic and advanced modes, including the settings interfaces, alert information, the welcome page with useful information and notifications to the user, automatic wifi connection, as well as an audio-description service in offline mode. In addition, the sending and receiving of data and notifications from VITABOX to the cloud and vice versa was implemented, as well as the management and exchange of data between VITABOX and the network of environmental sensors.

The work presented, and the obtained results contributed to a stable version of the system, ready to be tested and validated in real environment.

Keywords: Audio description, CEC, Cloud, HDMI, Teleassistance, Telehealth, TV

AGRADECIMENTOS

Este relatório de projeto de mestrado é o resultado de muitas horas de trabalho e é importante exprimir os meus sinceros agradecimentos a algumas pessoas que me ajudaram.

Aos orientadores Pedro Daniel Frazão Correia e Luís Agnelo de Almeida pela paciência, durante o processo de desenvolvimento da dissertação e por me terem sugerido este projeto.

Agradeço ao Nelson Gomes, pela ajuda durante a realização do projeto.

Agradeço aos bolsiros Pedro Nunes, Pedro Dias, Diogo Mendes e Ricardo António pela ajuda durante a realização do projeto.

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

Por fim, o meu profundo agradecimento a todas as pessoas que contribuíram para a concretização deste trabalho final de mestrado.

Este projeto 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.

ÍNDICE

Índice de Figuras	xv
Índice de Tabelas	xix
Lista de Abreviatura e Siglas.....	xxi
1 Introdução.....	1
1.1 Enquadramento/Problema.....	1
1.2 Arquitetura geral	3
1.3 Trabalho desenvolvido.....	4
1.4 Estrutura do documento	6
2 Estado da Arte	7
2.1 SmartAL.....	7
2.2 AAL	8
2.3 MySignals	8
2.4 10 Mil Vidas	9
2.5 Aingeru	10
2.6 Smart homes	10
2.7 Comparação	10
3 Tecnologias e ferramentas utilizadas.....	13
3.1 HDMI-CEC.....	13
3.2 NODE.JS.....	15

3.3	Base de Dados	15
3.4	HTTPS	16
3.5	SOCKET.IO	16
3.6	FFMPEG.....	16
3.7	VUE.JS	17
3.8	CoAP	18
3.9	CONTIKI-NG.....	18
3.10	Linux/Raspbian.....	18
3.11	Hardware	18
3.11.1	Microcomputador.....	18
3.11.2	Border Router e Sensores ambientais	20
3.11.3	Sensores Biométricos.....	21
4	Infraestrutura de comunicação da Vitabox.....	25
4.1	Infraestrutura da Vitabox.....	25
4.1	Aplicação Servidor	26
4.1.1	Base de dados.....	28
4.2	Comunicação VITABOX-TV.....	30
4.2.1	HDMI-CEC na VITABOX	30
4.2.2	Testes ao HDMI-CEC.....	31
4.2.3	Notificações	34
4.3	Comunicação com sensores biomédicos	34

4.4	Comunicação com rede de sensores ambientais	35
4.4.1	Registo dos sensores.....	37
4.4.2	Leitura dos sensores	38
4.4.3	Alertas gerados pela rede de sensores	39
4.5	Módulo de comunicação com a <i>cloud</i>	40
4.6	Conversão texto para fala (<i>text-to-speech</i>)	42
5	Interação baseada na TV.....	45
5.1	Introdução	45
5.2	Interação visual	45
5.1	Menus da Aplicação.....	47
5.1.1	Bem-vindo.....	47
5.1.2	Exames	48
5.1.3	Histórico dos exames	49
5.1.4	Sensores Ambientais	49
5.1.5	Histórico dos sensores ambientais.....	50
5.1.6	Alertas	51
5.2	Modais em uso	52
6	Teste de usabilidade ao sistema de interação	57
6.1	Testes utilizados.....	58
6.2	Metodologia	58
6.2.1	Participantes	59

6.2.2	Procedimento	60
6.2.3	Equipamento utilizado	60
6.3	Resultados.....	60
6.3.1	System Usability Scale (SUS)	65
6.4	Discussão de resultados	66
7	Conclusão	67
7.1	Trabalho futuro	68
8	Referências	69
	Anexos.....	73

Índice de Figuras

Figura 1 – Funcionalidades do Sistema VITASENIOR-MT.....	3
Figura 2 - Arquitetura Geral do sistema VITASENIOR-MT	4
Figura 3 – Interação com a VITABOX através da TV e respetivo comando.....	5
Figura 4 - Aplicação SmartAI	8
Figura 5 - Sensores MySignals	9
Figura 6 - Esquema do Cabo HDMI.....	14
Figura 7 - Adição dos endereços físicos e lógicos.....	15
Figura 8 - Raspberry Pi 3.....	20
Figura 9 – Zolertia	20
Figura 10 - Sensor de CO.....	21
Figura 11 – Sensor de Dióxido de carbono.....	21
Figura 12 - Sensor de temperatura e humidade	21
Figura 13 - Balança.....	22
Figura 14 - Sensor de Pressão Arterial	22
Figura 15 - Sensor de Glicemia	22
Figura 16 - Bracelete sensor de ritmo cardíaco.	23
Figura 17 – Diagrama de blocos da arquitetura da VITABOX.	26
Figura 18 – Arquitetura da aplicação servidora em NODE.JS.....	27

Figura 19 - Tabelas da base de dados	29
Figura 20 - Arquitetura HDMI-CEC na VITABOX.....	31
Figura 21-Evento a informar a disponibilidade do cec.....	32
Figura 22 - Função para forçar ligação do CEC.....	33
Figura 23 - OSD String no ecrã da TV	34
Figura 24 – Comunicação com os dispositivos biométricos	35
Figura 25 – Rede de sensores	36
Figura 26 - Registo dos sensores.	37
Figura 27 - Função para os pedidos COAP.	38
Figura 28 - Construção do objeto do sensor	38
Figura 29 - Criação do objeto em estado crítico.....	39
Figura 30 - Comunicação com a <i>cloud</i>	40
Figura 31 – Endpoint dos pacientes.....	41
Figura 32 – Endpoint das boards	41
Figura 33 – Endpoint dos tokens	41
Figura 34 - Função de envio dos registos para a <i>cloud</i>	42
Figura 35 - Algoritmo texto para fala.	43
Figura 36 - Concatenação dos ficheiros de áudio.....	44
Figura 37 - Comando da TV	45
Figura 38 - Comando da TV com as teclas selecionadas	46
Figura 39 - Menu Bem-Vindo.	47

Figura 40 - Menu dos Exames.	48
Figura 41 - Histórico dos sensores biométricos.	49
Figura 42 - Menu dos sensores ambientais	50
Figura 43 - Histórico dos sensores ambientais	51
Figura 44 - Alertas dos sensores ambientais.	52
Figura 45 - Modal configuração do <i>wifi</i>	53
Figura 46 - Modal Configuração.	54
Figura 47 - Menu avançado.	55
Figura 48 - Modo básico.	55
Figura 49 - Como avalia globalmente a aplicação do sistema interface TV-VITABOX?	61
Figura 50 - Como avalia globalmente os exames de saúde disponíveis?	62
Figura 51 - Como avalia o exame do peso.	63
Figura 52 - Como avalia o exame globalmente o acesso ao histórico dos exames?..	64
Figura 53 - Consulta de histórico de sensores ambientais.	65
Figura 54 - Questionário de Satisfação primeira parte.	73
Figura 55 - Questionário de Satisfação segunda parte.	74
Figura 56 - Questionário de Satisfação terceira parte.	75
Figura 57 - Escala de usabilidade do sistema primeira parte.	76
Figura 58 - Escala de usabilidade do sistema segunda parte.	77
Figura 59 - Procedimento dos testes primeira parte.	78

Figura 60 - Procedimentos dos testes segunda parte. 79

Índice de Tabelas

Tabela 1 - Comparação entre os sistemas telehealth.	11
Tabela 2 - Comparação entre os três microcomputadores	19

Lista de Abreviatura e Siglas

API	Application Programming Interface
HTML	Hypertext Markup Language
JS	JavaScript
stdin	Standard input
stdout	Standard output
stderr	Standard error
OSD	On-screen Display
UDP	User Datagram Protocol
ID	Identification
IP	Internet Protocol
CEC	Consumer Eletronics Control

1 Introdução

1.1 Enquadramento/Problema

A contínua baixa taxa de natalidade, assim como o aumento da longevidade, vão implicar uma mudança significativa na estrutura da população da União Europeia (EU-27). Esta alteração afetará tanto a faixa da população sénior, como a população ativa. É espetável que em 2060 a faixa etária acima dos 65 anos, ocupe 29.5% da população, sendo que 12%, terá 80 anos ou mais. Paralelamente, haverá também uma descida significativa na população ativa. Prevê-se que na faixa etária dos 15 aos 64 anos haja um decréscimo, passando dos 67% que existiam em 2010 para 56.2% em 2060 (Comission, 2014).

Consequentemente, com o aumento da idade média da população, prevê-se um aumento de doenças crónicas. Nos países da união europeia uma das principais causas de morte são as doenças cardiovasculares. Dentro deste grupo, 60% das causas são compostas por doenças cerebrovasculares e por doenças de isquémica cardíaca, não descurando os casos de hipertensão, diabetes e obesidade (OECD, 2010).

Esta mudança nas estruturas demográficas e de necessidades de cuidados médicos pela população, tem levado a uma transformação dos modelos económicos, dos sistemas de saúde e dos serviços relacionados com a proteção social da população, originando uma preocupação cada vez maior por parte dos países desenvolvidos. Neste contexto, a população mais idosa é um dos grupos mais vulneráveis devido ao aumento da comorbidade assim como pela falta de cuidadores que assegurem uma vigilância atempada e próxima. Consequentemente haverá uma sobrecarga adicional nos serviços de saúde.

Os sistemas de telesaúde (*telehealth*) são uma das soluções possíveis e viáveis para aliviar a pressão de procura sobre os serviços de saúde, mas também garantem o acesso aos serviços de saúde a faixas da população com problemas de mobilidade.

Esta solução oferece um vasto leque de serviços, tais como monitorização remota de sinais biométricos e ambientais, monitorização de atividades diárias, comunicação, apoio à medicação e reabilitação. Estes sistemas de telesaúde trazem grandes benefícios, contudo, ainda têm limitações que afetam a eficiência do seu uso, nomeadamente na sua forma de

interação com o utilizador final (Bujnowska-Fedak & Grata-Borkowska, 2015) (Majumder, et al., 2017) (Normie, 2011).

Devido a vários fatores, a população sénior tem mais dificuldades em interagir com as novas tecnologias, estes fatores são problemas de visão, audição, agilidade, de perceção, até a dificuldade natural de lidar com equipamentos informáticos, o que pode impossibilitar o uso de *smartphone*, *tablets* ou computadores. Uma das possíveis soluções para melhorar a interação com os sistemas de telesaúde é a utilização de televisores e do respetivo telecomando. Tratam-se de dois dispositivos já conhecidos da população idosa, fáceis de usar e disponíveis na maior parte das habitações. Esta solução de interação minimiza o esforço cognitivo necessário durante a interação com os sistemas telesaúde (G. Pires, 2018).

O trabalho apresentado neste relatório de projeto é parte integrante do sistema de telesaúde, VITASENIOR-MT, enquadrado no projeto IC&DT, N° 023659 - “VITASENIOR-MT - Assistência aos cuidados de saúde de idosos no Médio Tejo”. A Figura 1 mostra as suas funcionalidades principais.

O objetivo do projeto é resolver o problema que as tecnologias impõem a esta faixa de população, fazendo uso da televisão, como um meio de interação, oferecendo um ambiente familiar e fácil de usar. Tirando vantagem da Internet Of Things (IoT), o projeto pretende monitorizar vários parâmetros do local em que se encontra, nomeadamente os dados ambientais, como a humidade, a temperatura, o monóxido de carbono e o dióxido de carbono, recolhidos de forma automática. Por outro lado, o sistema permite a recolha de dados biométricos, como a pulsação, a oximetria, a glucose, a temperatura e o peso. Estes dados são recolhidos por iniciativa do utilizador (G. Pires, 2018).

O sistema proposto permite que o utilizador receba de forma automática através da TV notificações médicas, avisos e alertas resultantes da análise das medições dos sensores ambientais. A grande vantagem deste sistema é a utilização da TV e do seu comando remoto para interação da aplicação VITABOX. A simplicidade de utilização consegue-se através de alguns cliques no telecomando dando acesso a todas as funcionalidades da aplicação.

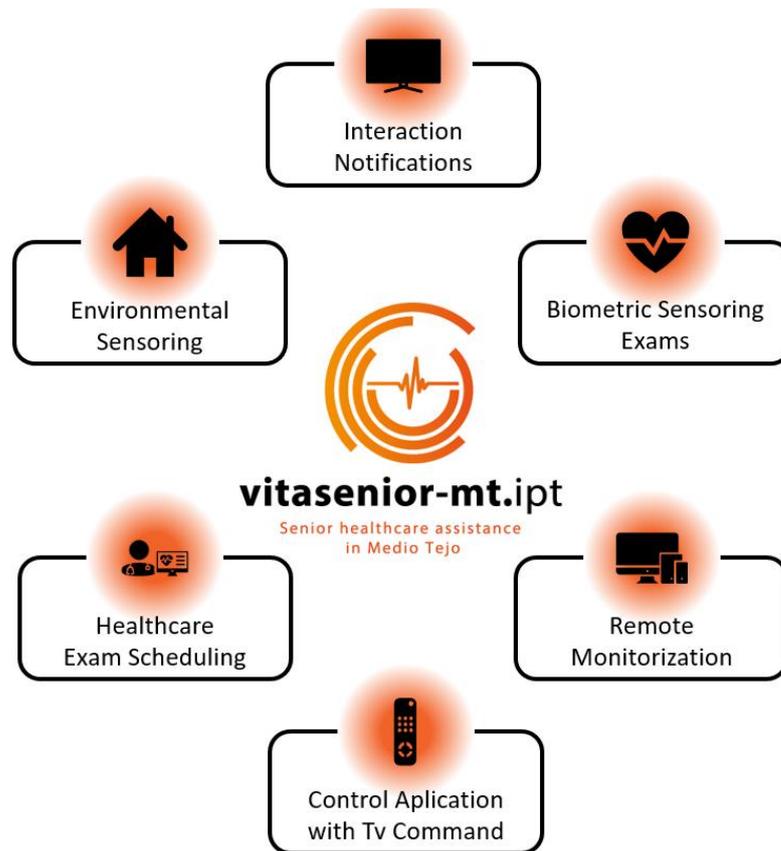


Figura 1 – Funcionalidades do Sistema VITASENIOR-MT.

1.2 Arquitetura geral

A Figura 2 representa a arquitetura da aplicação VITASENIOR-MT, uma solução de tele saúde. A solução proposta é composta de vários módulos: Rede de Sensores Wireless, VITABOX e a *cloud*. O módulo principal é a VITABOX e tem como umas das funções principais, a interação com o utilizador através de ligação com a TV e seu comando. O utilizador tem a capacidade de controlar os menus e funcionalidades usando o comando. Adicionalmente, a VITABOX tem funções de interação, aquisição, armazenamento e transmissão dos dados dos sensores biométricos e dos sensores ambientais. Devido a capacidade de memória reduzida, a informação é enviada para a *cloud*, mantendo na VITABOX os últimos dados.

A *cloud* tem a função de guardar, processar, analisar e distribuir os dados para as várias aplicações associadas. Adicionalmente o sistema de informação da *cloud* tem por função manter os perfis clínicos que servem de base aos alertas gerados adequados a cada utilizador.

O *backoffice* deste sistema permite a gestão dos dispositivos, dos utilizadores e do controlo de acessos e autenticação.

A *cloud* pode igualmente ser gerida através de um conjunto de aplicações web e mobile. A informação recolhida de cada utilizador poderá ser posteriormente analisada por médicos, prestadores de cuidados de saúde e familiares dos utentes.

Finalmente o sistema permite a geração de alertas para entidades de saúde e de segurança através do serviço IOC (Intelligent Operation Center), permitindo desencadear ações de emergência.

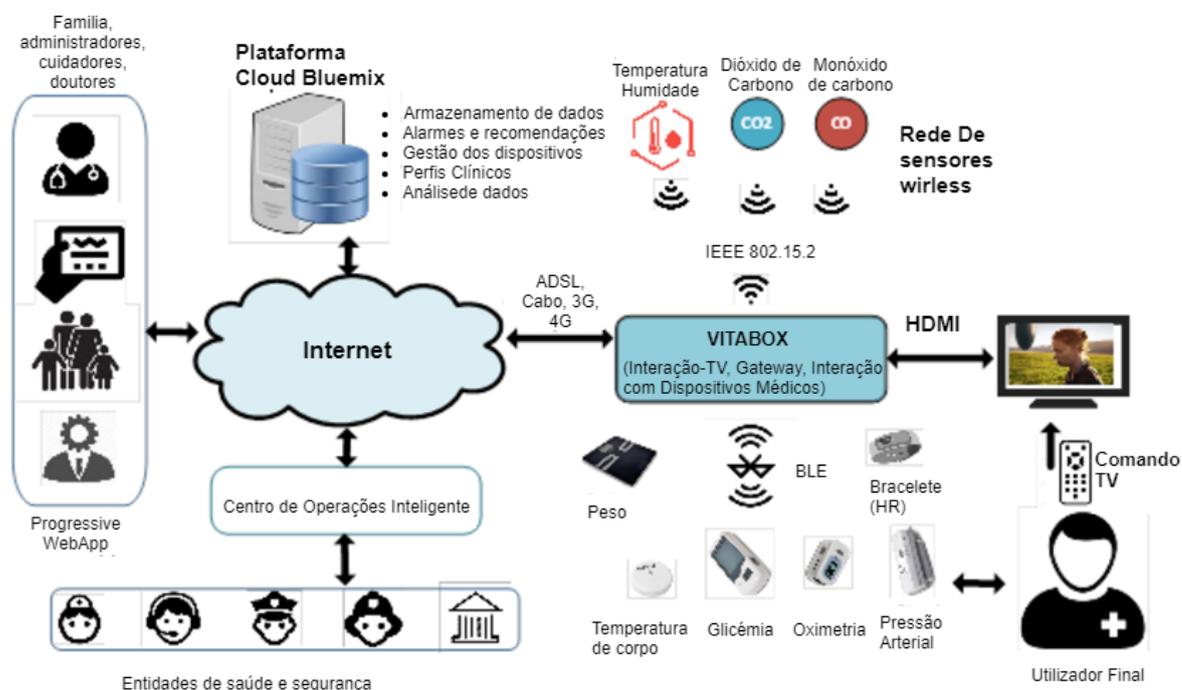


Figura 2 - Arquitetura Geral do sistema VITASENIOR-MT (G. Pires, 2018)

1.3 Trabalho desenvolvido

O projeto VITASENIOR-MT foi dividido em vários subprojectos: rede de sensores, desenvolvimento da VITABOX e serviços *Cloud*. O projeto da rede de sensores pretende desenvolver a ligação automática de cada sensor ao sistema, associando esquemas de segurança para a autenticação de cada nó de sensores. O projeto de serviços *cloud* pretende

desenvolver todo o *backoffice* da aplicação, as aplicações web e mobile, e toda API de acesso da VITABOX à *cloud*. O módulo da VITABOX, sob o qual incide este relatório, tem como função a comunicação entre a VITABOX e a TV, o desenvolvimento da interface de interação na TV, desde os alertas e notificações automáticas, assim como as aplicações por iniciativa do utilizador para medição de sinais biométricos e monitorização simples de dados ambientais. A interação com utilizador suporta áudio-descrição.

Como mostra na **Erro! A origem da referência não foi encontrada.**, todo o sistema de interação é baseado no comando da TV. Esta é uma funcionalidade central do projeto e da implementação uma vez que toda a interface foi pensada para que as funcionalidades do sistema fossem executadas da forma mais simples e com o menor número de passos possível.



Figura 3 – Interação com a VITABOX através da TV e respetivo comando.

O trabalho desenvolvido e apresentado neste relatório de projeto consistiu em:

- Implementação de interfaces de interação com o utilizador. Foram desenvolvidos o modo avançado e o modo básico da aplicação visual. Adicionalmente foram implementadas as interfaces de configurações, de informações de alertas, da ligação automática de *wifi*, da página de boas-vindas com informações úteis e notificações ao utilizador.

- Envio e receção dos dados e das notificações da VITABOX para a *cloud*, e vice-versa;
- Realização de toda a gestão e troca de dados entre a VITABOX e a rede de sensores desde o registo dos sensores ambientais, escuta de alertas e recolha de dados;
- Implementação de um algoritmo personalizado de áudio-descrição. Este algoritmo de conversão de texto para fala, permite o funcionamento em modo offline e a personalização dos ficheiros de áudio VITABOX.

1.4 Estrutura do documento

O presente relatório está organizado da seguinte forma. O capítulo dois apresenta o estado da arte das tecnologias de tele saúde e de teleassistência. O capítulo três descreve as tecnologias de *hardware* e software utilizadas para o desenvolvimento do protótipo. O capítulo quatro descreve a arquitetura da VITABOX, desde a comunicação entre a VITABOX e o recetor de TV, até à comunicação da rede de sensores. O capítulo cinco descreve aplicação visual baseada em TV. No capítulo 6 apresenta os testes realizados e os resultados obtidos e finalmente o capítulo 7 faz uma conclusão do trabalho.

2 Estado da Arte

A ideia da teleassistência não é nova. Esta tecnologia permite que a população mais idosa possa ter alguma autonomia, em que o objetivo é conseguir monitorizar de forma simples e menos invasiva possível, todas as componentes essenciais da saúde dos idosos.

Com a tecnologia ainda a evoluir para sistemas menos invasivos e de funcionamento *plug-and-play*, existem soluções mais imediatas de teleassistência, para minimizar o esforço pedido aos seniors. A fácil utilização da tecnologia é um fator crucial para a aceitação por parte dos idosos. A inclusão da televisão nos sistemas é um exemplo de um dispositivo de visualização familiar.

Em seguida descrevem-se alguns trabalhos de teleassistência relacionados.

2.1 SmartAL

A SmartAL (SmartAL, s.d.) é uma solução de teleassistência que tira partido das funcionalidades da televisão e da meo-box existente nas casas dos utilizadores. Através do comando da TV, o utilizador pode efetuar medições regulares dos dispositivos biométricos. Este sistema permite monitorizar o estado atual do utilizador, podendo gerar notificações e alertas a partir dos dados gerados.

A solução integra uma aplicação Android, que pode ser usado para gerir os dados dos sensores biométricos e transmitir os dados para o servidor. Existe também uma aplicação web, que os cuidadores podem aceder, para observar os dados em tempo real, configurar os limites dos dispositivos biométricos, tarefas diárias, alertas e outras informações.



Figura 4 - Aplicação SmartAl (SmartAL, s.d.).

2.2 AAL

O AAL (Ambient Assisted Living) é uma solução que pretende dar maior conforto e segurança aos utilizadores. Este conforto chega ao utilizador através da integração das novas tecnologias em casas inteligentes. No conceito do AAL tenta-se unificar todo o sistema, como o ambiente inteligente, a inteligência ambiental e os fundamentos do AAL (Angelo Costa, 2014) (Marcelino I, 2018) (Luís M. L. Oliveira, 2014).

O modelo proposto pelo AAL tem como público-alvo a população os seniores, disponibilizando vários serviços necessários para uma melhor monitorização dos sinais vitais. Assim como monitorização ambiental, de forma a conseguir reagir a problemas imediatos.

2.3 MySignals

MySignals é uma solução de *e-health*, para leitura de dispositivos biométricos e disponibiliza uma plataforma de desenvolvimento de aplicações (MySignals, s.d.).

Os dados recolhidos podem ser enviados para a *cloud*. Estando estes dados na *cloud* é possível ao utilizador aceder à informação através da aplicação web ou apps para os dispositivos moveis.

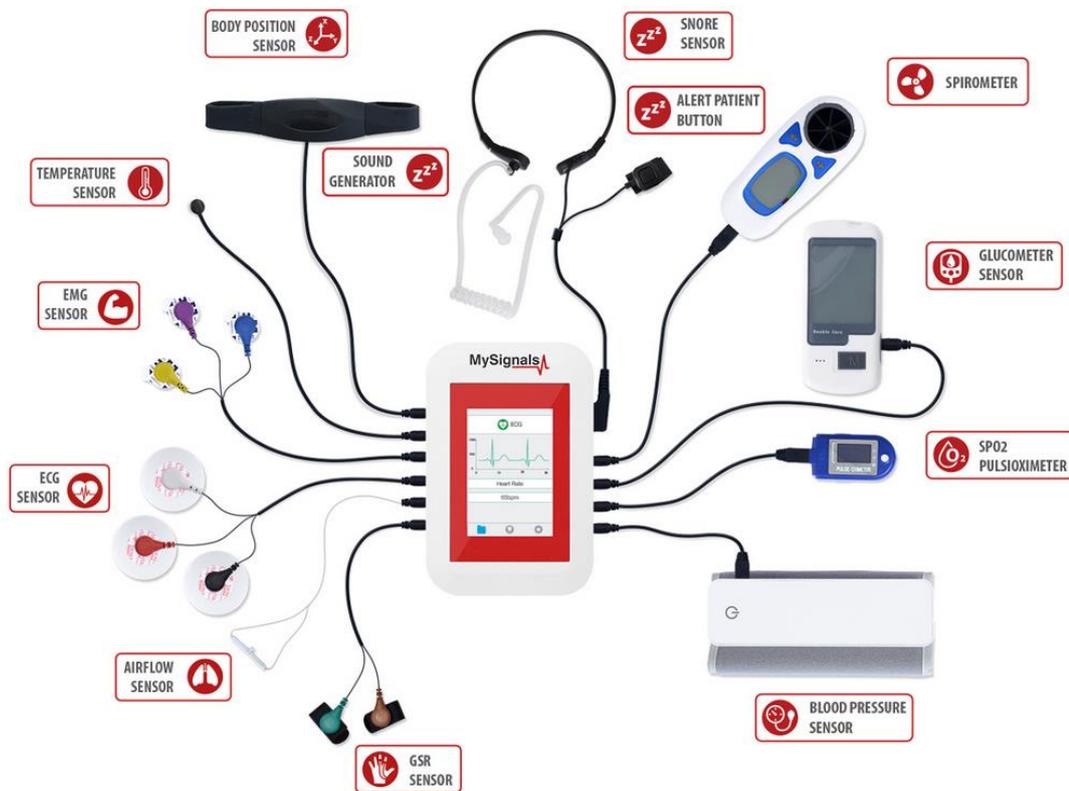


Figura 5 - Sensores MySignals (MySignals, s.d.).

2.4 10 Mil Vidas

O projeto 10 mil vidas (Projeto 10 Mil Vidas - ANCS, s.d.) pretende criar um modelo de apoio aos idosos, fazendo uso dos dispositivos atuais e aplicar com a teleassistência (10 Mil Vidas - PDF, s.d.).

O serviço de teleassistência também inclui alertas de monitorização da saúde, localização, possíveis marcações de consultas, uma linha de apoio vinte e quatro horas e uma plataforma de serviço.

A solução pretende garantir uma assistência 24h por dia a um grande número de idosos. O projeto está presentemente implantado em alguns municípios do país.

Esta solução inclui um botão de SOS caso haja uma emergência, localização através do uso do GPS e alertas caso o paciente saia de uma zona definida. Permite ainda o controlo de indicadores de saúde, recorrendo a medições com dispositivos biométricos, deteção de eventuais anomalias, lembretes, gestão de agenda e controlo da toma de medicação.

2.5 Aingeru

Aingeru (Tablado, 2004) é um sistema de teleassistência que utiliza um PDA (Personal Digital Assistance), como meio de interação com o utilizador. Como o dispositivo é portátil e tem acesso à tecnologia *wireless*, é fácil de o transportar.

O dispositivo é carregado com a aplicação Aingeru e tem acesso a várias funcionalidades, desde leituras biométricas, a processamento das medições e decisão sobre a forma de lançar alertas caso necessário.

2.6 Smart homes

O projeto Smart Home (Majumder, et al., 2017) é um sistema de teleassistência que tem como base a monitorização contínua de sinais biométricos. Utiliza dispositivos que comunicam através de *wireless* para um dispositivo que centraliza a informação e posteriormente envia esses dados para um dispositivo recetor, *smartphone* ou computador.

2.7 Comparação

Foi feita uma comparação entre as tecnologias mencionadas e a VITABOX. Na [Tabela 1] foram postas as várias funcionalidades que se determinou como essenciais no sistema de teleassistência, medição ambiental, medição dos sinais biométricos, texto para fala, serviços de *cloud*, o uso do comando da TV, alertas dos sensores ambientais, alertas dos sensores biométricos, se trabalha *offline*, se é independente de operadoras e se é portátil.

Tabela 1 - Comparação entre os sistemas telehealth.

	VITABOX (G. Pires, 2018)	SmartAL (SmartAL, s.d.)	Aingeru (Tablado, 2004)	Smart homes (Majumder, et al., 2017)	MySignals (MySignals, s.d.)
Medição ambiental	X	-	-	-	-
Medição Biométrica	X	X	X	X	X
Texto para fala	X	-	-	-	-
Serviço de Cloud	X	X	-	X	X
Uso do Comando da TV	X	X	-	-	-
Alertas dos Sensores Ambientais	X	-	-	-	-
Alertas dos Sensores Biométricos	X	X	X	X	X
Trabalha offline	X	-	-	-	X
Independente de Operadoras	X	-	-	-	X
Portátil	-	X	X	X	X

3 Tecnologias e ferramentas utilizadas

Este capítulo apresenta as tecnologias utilizadas em *hardware* e *software*. Descreve as tecnologias utilizadas para desenvolvimento do *software* da aplicação visual da VITABOX. Adicionalmente, realiza uma descrição do microcomputador escolhido para a implementação da VITABOX, e dos sensores utilizados.

3.1 HDMI-CEC

O protocolo HDMI (High-Definition Multimedia interface) é um protocolo para reprodução de áudio e vídeo, em dispositivos com entradas de HDMI. HDMI é um substituto digital aos *standards* do sinal analógico (HDMI, s.d.).

A vantagem do HDMI-CEC é o controlo dos dispositivos associados à TV, através do comando. O protocolo permite transmissão áudio e vídeo, assim como transmissão de dados de controlo dos dispositivos ligados à TV.

O Protocolo CEC (Consumer Electronics Control) é um protocolo associado do HDMI, desenhado para permitir aos utilizadores, através do comando da TV, controlar dispositivos ligados através do HDMI. O CEC também permite o controlo de dispositivos sem qualquer intervenção do utilizador, para além de ligar automaticamente todos os dispositivos com suporte ao CEC (Consumer Electronics Control, s.d.).

O controlo de dispositivos através do CEC implica enviar para a TV uma série de bytes do comando. Estes códigos de comando podem ter especificidades relacionadas com o fabricante dos dispositivos, pelo que neste projeto foram gerados a partir do *site* CEC-O-MATIC (CEC-O-MATIC, s.d.).

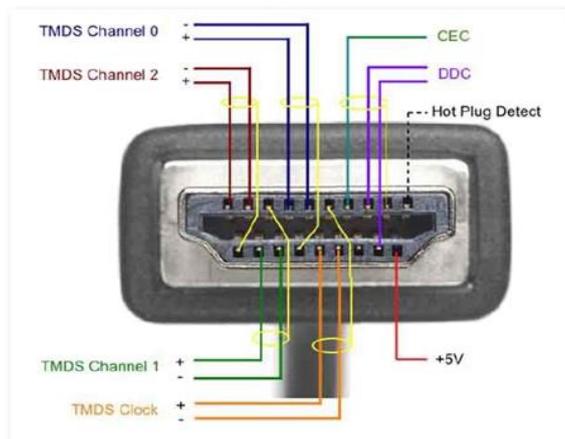


Figura 6 - Esquema do Cabo HDMI

O protocolo físico HDMI-CEC utiliza um dos 19 pinos do cabo HDMI, pino 13 (Figura 6). Trata-se de um canal bidirecional série que utiliza um só condutor e é baseado no *standard* do protocolo AV.link. Permite que o comando remoto da TV controle outros equipamentos interligados por HDMI. O CEC foi definido na especificação da versão 1.0 do protocolo HDMI e atualizado na versão 1.2. Na versão 1.3a adicionou-se um timer e comandos de áudio. Caso o dispositivo não tenha CEC é possível usar um adaptador USB para o PC (Consumer Electronics Control, s.d.).

Em qualquer configuração de HDMI, o display é considerado como *root*. São alocadas duas etiquetas especiais: o endereço físico de 0.0.0.0 e um endereço lógico de 0. Em qualquer sistema, os dispositivos CEC vão receber um endereço físico e um lógico. Os dispositivos que não têm CEC apenas vão receber um endereço físico. O endereço físico é gerado pela posição do dispositivo na árvore em relação ao *root*. Ao contrário, o endereço lógico é definido consoante a funcionalidade do dispositivo, como está descrito na especificação do CEC (Consumer Electronics Control, s.d.). Quando o CEC é ligado, envia um *broadcast* para verificar que dispositivos se encontram ligados, ficando à escuta das respostas criando uma lista de dispositivos.

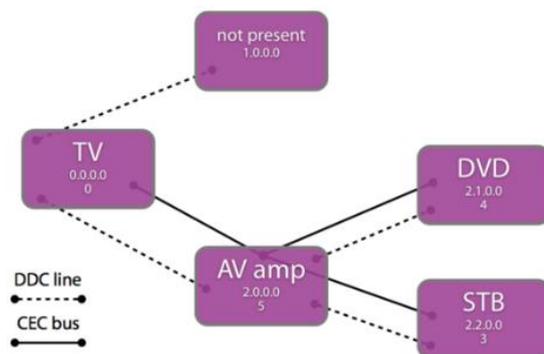


Figura 7 - Adição dos endereços físicos e

Para o uso do CEC é necessário conhecer os comandos disponíveis. Em (CEC-O-MATIC, s.d.) são apresentados os comandos existentes, cuja funcionalidades dependem da implementação de cada fabricante. Destacam-se os comandos utilizados na aplicação:

- **Active Source:** O dispositivo passa a ser a fonte ativa;
- **Standby:** Muda o dispositivo para *standby*;
- **Set OSD String:** Faz o display do texto na TV;
- **Set OSD Name:** Muda o nome do dispositivo da TV.

3.2 NODE.JS

NODE.JS (Docs | Node.js, s.d.) é um interpretador de código *Javascript* para o servidor capaz de receber e executar código do cliente. Trata-se um ambiente servidor em código aberto que interpreta *Javascript* em tempo real e utiliza o motor v8 do Google Chrome. Essencialmente converte a aplicação em código máquina, melhorando o desempenho. O NODE.JS usa uma arquitetura focada em eventos tendo por base o *javascript*.

3.3 Base de Dados

A VITABOX utiliza uma base de dados para armazenar todos os dados, criando registos para os valores biométricos e ambientais. A base de dados usada é a MONGODB (Open Source Document Database | MongoDB, s.d.). Optou-se por uma base de dados não relacional pelo facto de não ser necessário usar um ambiente transaccional.

A MONGODB não usa o ACID (Atomicidade, Consistência, Isolamento e Durabilidade), tornando as leituras e escritas mais rápidas e é baseada em documentos do tipo JSON (JSON, s.d.). O uso de documentos permite escalar e manter o rápido acesso. Para o servidor comunicar com a MONGODB instalou-se *mongoose*, um módulo que permite a comunicação com a base de dados e o NODE.JS. Este dá acesso a uma solução *schema-based* para modelar os dados, uteis para construção de *queries*.

3.4 HTTPS

HTTPS é uma implementação do protocolo HTTP sobre uma camada adicional de segurança que utiliza o protocolo SSL/TLS (HTTPS | Node.js v11.2.0 Documentation, s.d.). Os dados são transmitidos por meio de uma ligação cifrada e verifica a autenticidade do servidor e do cliente por meio de certificados digitais. A porta TCP usada é a 443.

3.5 SOCKET.IO

Socket.io (Socket.IO, s.d.), que é uma biblioteca *Javascript* para aplicações web em tempo real, permitindo comunicação bidirecional entre o cliente web e o servidor. O socket.io usa o *Polling* para estabelecer a ligação e após a ligação estabelecida muda para o *Websocket*. O uso de websockets minimizam os atrasos na comunicação e tornando a estrutura dos pacotes mais flexível.

Polling é um protocolo de transporte que envia de forma cíclica pedidos de HTTP ao servidor e após a resposta, fecha a ligação.

WebSocket é um protocolo de transporte para envio de pedidos de HTTP ao servidor. O servidor estabelece uma ligação bidirecional *statefull*, guarda a sessão, e a partir daí pode enviar e receber pedidos a qualquer momento.

3.6 FFMPEG

O *ffmpeg* (FFMPEG Documentation, s.d.) é uma ferramenta multiplataforma de suporte as funcionalidades multimédia. Permite uma grande interação com ficheiros de áudio e de vídeo, tanto para codificação como para descodificação, reprodução e *streaming*.

A ferramenta *ffmpeg* usa bibliotecas de código aberto, como o *libavcodec*, biblioteca de *codec* de áudio e vídeo, e *libavformat*, um multiplexador/demultiplexador de conteúdo de áudio e vídeo.

3.7 VUE.JS

Para criação de aplicações numa página única utilizou-se a tecnologia *VUE.JS* (Bemenderfer, s.d.) . Trata-se de uma *framework* de código aberto em *javascript*. No projeto utilizou-se o *template VUE.JS* baseado no Paper Dashboard (Vue Paper Dashboard, s.d.).

Foram usados, adaptados alguns módulos em *VUE.JS*, para construir a aplicação. Os principais módulos são o *vmarquee*, o *vmodal* e o *vue-socket.io*.

O *vmarquee* (wg5945, s.d.) é um módulo, que foi adaptado, para apresentar mensagens em rodapé, servindo para as notificações e para os alertas.

O *vue-socket.io* (MetinSeylan, s.d.) permite que que a *framework* do *VUE.JS* comunique com os sockets do servidor em tempo real, evitando o recurso a pedidos http.

O *vmodal* (euvl, s.d.) é um módulo adaptado de disponibilização de modais. Com o acesso a este módulo foi possível desenvolver *views* personalizadas para a aplicação, sendo usado em configurações de alertas, configuração do *wifi*, *etc.*.

As modais são elementos visuais de um *website* que se sobrepõem à aplicação através de scripts. Uma das vantagens do uso de modais, passa pelo facto de não ser necessário usar as janelas *pop-ups* ou o carregar da página. Estes são janelas de conteúdo que permitem mostrar informação rapidamente sem alterar o estado atual do *website*.

Para uma fácil navegação na aplicação e de disponibilização de dados críticos para o funcionamento em toda a aplicação, usou-se o *event-bus* do *VUE.JS* (Bemenderfer, s.d.). O *event-bus* é um componente que se encontra disponível de forma global para toda a aplicação e é um sistema baseado nos eventos, similar aos eventos do *javascript*.

O *118n* (kazupon, s.d.) é um processo de desenvolvimento ou de adaptação das aplicações, para dinamicamente alterar a linguagem da aplicação.

3.8 CoAP

O CoAP (Constrained Application Protocol) (CoAP — Constrained Application Protocol | Overview, s.d.) é um protocolo utilizado em dispositivos com restrições de recursos, como os sensores, que permite o funcionamento com uma baixa largura de banda e baixa disponibilidade. Este protocolo tem um *overhead* baixo e consome pouca energia.

3.9 CONTIKI-NG

CONTIKI é um sistema operativo de código aberto que corre em pequenos microcontroladores, ideal para dispositivos de pouca energia, sendo possível desenvolver aplicações que façam uso do hardware e da comunicação wireless de baixa alimentação.

O microcontrolador que integra os sensores da VITABOX utiliza o CONTIKI-NG (Home · contiki-ng/contiki-ng Wiki, s.d.), um sistema operativo de código aberto e múltipla-plataforma. O CONTIKI-NG é uma versão do CONTIKI e está vocacionado para comunicações com dispositivos de baixa energia utilizando protocolos padrão, como o ipv6/6LoWPAN, 6TiSCH, RPL e CoAP.

3.10 Linux/Raspbian

O Raspbian (Raspbian, s.d.) é um sistema operativo de código aberto baseado em Linux Debian, e otimizado para o *hardware* Raspberry Pi com processadores ARM. Este sistema operativo em versões mais recentes usa o PIXEL (Pi Improved X-Window Environment Ligthweight) como o ambiente de trabalho gráfico. A Vitabox utiliza o Raspbian dado o suporte que existe, no entanto existem outras opções para o sistema operativo e para o browser como o Chromium OS (Google Linux) e o Chromium (web browser).

3.11 Hardware

3.11.1 Microcomputador

As aplicações executadas na VITABOX têm requisitos de hardware específicos, O microcomputador precisa de ter um módulo *wifi*, para estabelecer uma ligação *wireless*, um

módulo Bluetooth, para a comunicação com os sensores biométricos, no mínimo uma entrada HDMI, duas portas *usb* e 1GB de RAM.

De início usou-se o Raspberry Pi 2 modelo B+ como *hardware* para a VITABOX, mas com o evoluir do projeto as necessidades cresceram, começou a ser necessário mais RAM no hardware para suportar a aplicação, começou a ser necessário o uso do Bluetooth e do wireless, pelo que o hardware atual não podia responder às necessidades acrescidas.

Considerando os requisitos de *hardware* foram analisados três tipos de minicomputadores: o Raspberry Pi 2, o Raspberry Pi 3 e o Rock64. Na Tabela 2 é possível consultar as características e a opção recaiu no Raspberry pi 3, dado que integra os módulos o *wifi* e o Bluetooth.

Tabela 2 - Comparação entre os três microcomputadores

	Raspberry Pi 2 Modelo B+	Raspberry Pi 3 Modelo B	Rock64
Preço	\$35	\$35	\$45
Socket	Broadcom BCM2837	Broadcom BCM2837	Rockchip RK3328
Arquitetura	ARMv8-A (64/32-bit)	ARMv8-A (64/32-bit)	ARMv8-A (64/32-bit)
Nº core	4	4	4
Clock CPU	900 MHz	1.2 GHz	1.5 GHz
GPU	VideoCore IV	VideoCore IV	Arm Mali-450
Ram	1 GB	1 GB	1/2/4 GB
Wireless	x	802.11n	X
Bluetooth	x	4.1	X
Consumo	800 mA	800 mA	500mA - 3A

No projeto foi necessário adquirir microcomputadores para a instalação e configuração da VITABOX. Na (Figura 8) mostra o microcomputador, Raspberry Pi3, escolhido.



Figura 8 - Raspberry Pi 3.

3.11.2 Border Router e Sensores ambientais

Os Zolertias (Figura 9), foi o *hardware* escolhido para integrar os sensores e o Border Router.



Figura 9 – Zolertia

Na (Figura 10) é representado o sensor de monóxido de carbono.



Figura 10 - Sensor de CO.

Na (Figura 11) é representado o sensor de Dióxido de carbono.



Figura 11 – Sensor de Dióxido de carbono

Na (Figura 12) tem representado os sensores de temperatura e humidade, sendo o mesmo transdutor a fazer as duas medições.



Figura 12 - Sensor de temperatura e humidade

3.11.3 Sensores Biométricos

Na (Figura 13) tem representado o dispositivo de medição de peso.

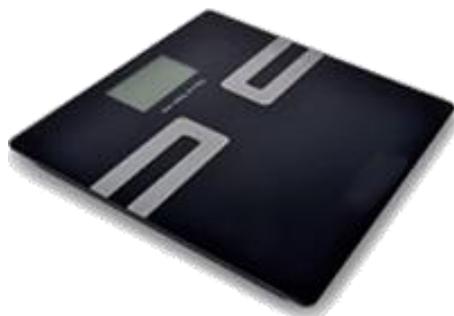


Figura 13 - Balança

Na (Figura 14) tem representado o dispositivo de medição pressão arterial.



Figura 14 - Sensor de Pressão Arterial

Na (Figura 15) tem representado o dispositivo de medição glicemia.



Figura 15 - Sensor de Glicemia

Na (Figura 16Figura 15) tem representado o dispositivo de medição de ritmo cardíaco.



Figura 16 - Bracelete sensor de ritmo cardíaco.

4 Infraestrutura de comunicação da Vitabox

Este capítulo apresenta a estrutura de comunicação da VITABOX nos seus diferentes módulos. Será dado maior destaque às componentes que estiveram mais diretamente relacionados com o trabalho desenvolvido.

4.1 Infraestrutura da Vitabox

A Figura 17 representa um diagrama de blocos da arquitetura da VITABOX, implementada num Raspberry Pi 3. Este é composto por vários módulos, correspondentes às diferentes funcionalidades:

- i) **Servidor**, é responsável pela gestão de todos os processos de interação com os diferentes módulos. Faz a comunicação com a *cloud* através de uma API, de onde recebe a identificação do utilizador, dos sensores e informações sobre notificações assim como, e para onde envia os dados recolhidos pela VITABOX. Paralelamente faz a gestão da comunicação com a televisão via HDMI e também com o cliente web de suporte à aplicação visual.
- ii) **Interação da VITABOX com a TV**, realizada através do protocolo *hdmi-cec*. A interpretação dos comandos é realizada utilizando a biblioteca *libcec*, que permite a comunicação entre o sistema operativo do microcomputador com a televisão. Adicionalmente implementa a aplicação visual de interação baseada numa aplicação WEB.
- iii) **Comunicação dos sensores biomédicos** através de Bluetooth Low Energy (BLE). Cada sensor, estando no alcance da VITABOX tem a capacidade de se ligar de forma automática, existindo uma ação de autenticação para o efeito.
- iv) **Ligação dos sensores ambientais**, com ligação pelo protocolo IEEE802.15.4 (IEEE 802.15.4, s.d.) ao *border router*. Este é responsável pela comunicação com o microcomputador através de um túnel entre a rede de sensores IEEE802.15.4 e a VITABOX.
- v) **Conversão text-to-speech**, é feita a conversão to texto para fala das ações, dos avisos e das notificações desencadeadas pela aplicação visual. Utiliza métodos de conversão *offline*, dando autonomia ao processo.

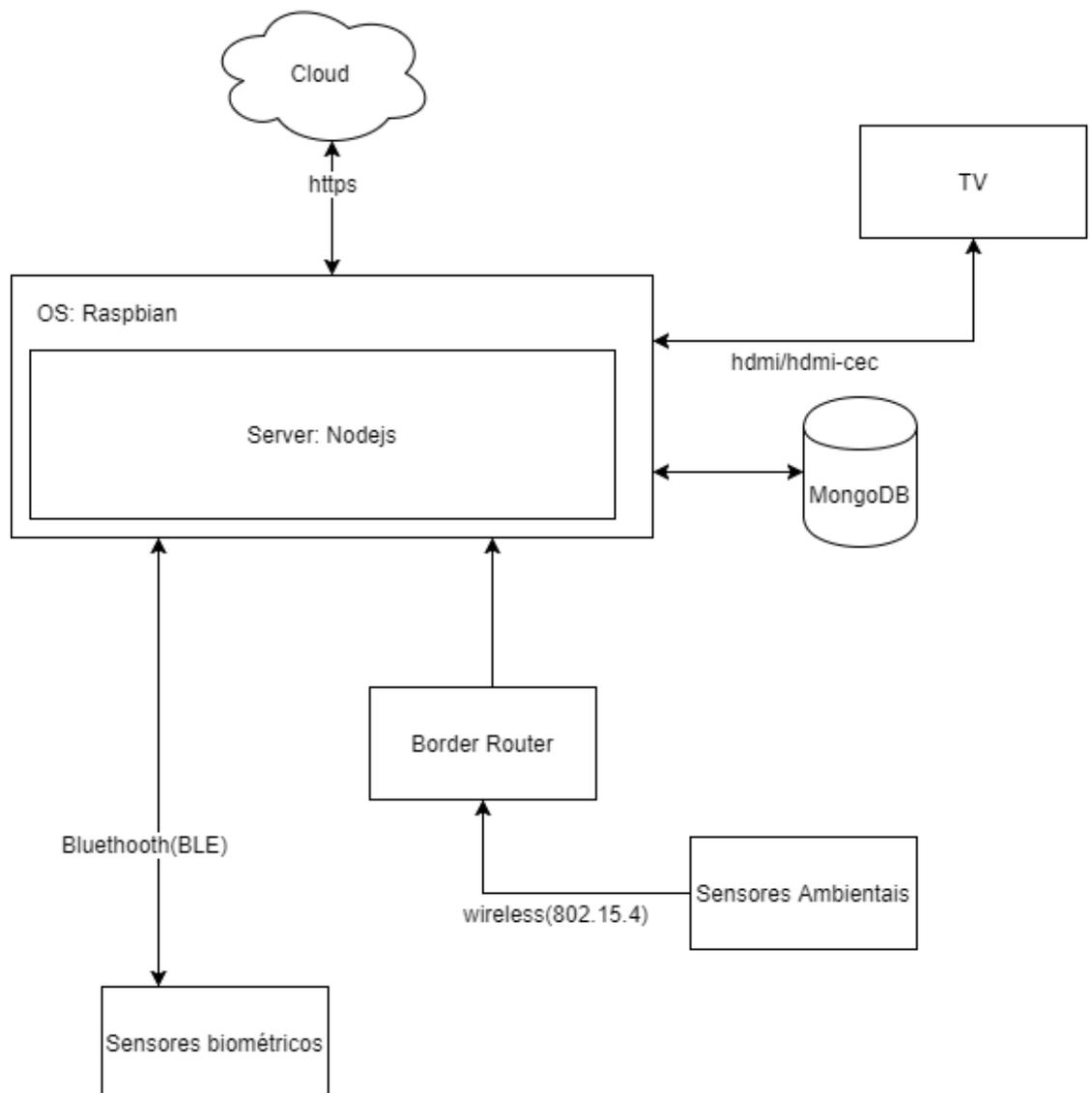


Figura 17 – Diagrama de blocos da arquitetura da VITABOX.

4.1 Aplicação Servidor

A aplicação Servidor da VITABOX foi implementada usando o NODE.JS. Esta faz a gestão de todo o processo implementado no microcontrolador

Tendo como base o NODE.JS, foi necessário utilizar vários módulos de modo a facilitar o desenvolvimento da aplicação. Estes comunicam com as bibliotecas do sistema operativo de forma a disponibilizar as suas funcionalidades em *javascript*.

A Figura 18 apresenta a arquitetura da aplicação Servidor em NODE.JS.

- Datagram: módulo para comunicação com a rede de sensores através do UDP.
- Blelib: biblioteca criada para facilitar a comunicação com os dispositivos Bluetooth.
- Audioconcat: biblioteca para comunicação com o *ffmpeg*, que disponibiliza a concatenação dos ficheiros de áudio.
- Socket.io: Módulo de comunicação por *websockets* para uma comunicação em tempo real entre o servidor e a aplicação web.
- Mongoose: Uma biblioteca para disponibilizar as ferramentas da base de dados do MONGODB.
- HTTPS: Comunicação com a *cloud* de forma segura.
- Nodecec: Módulo de comunicação com a biblioteca *cec-utils* para comunicar com o HDMI-CEC.

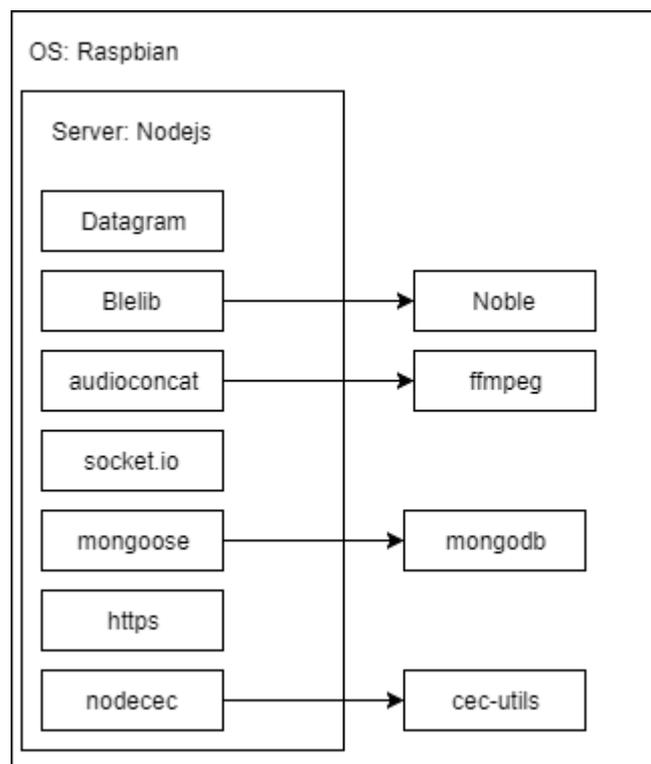


Figura 18 – Arquitetura da aplicação servidora em NODE.JS.

Durante o desenvolvimento da VITABOX, foi necessário adicionar ficheiros de configuração, de forma a poder facilmente configurar o servidor. Foram criados dois ficheiros. Um para modo de produção e um para o modo de desenvolvimento. Desta forma podemos obter comportamentos diferentes, testar a aplicação, fazer o *debug*.

Para além dos ficheiros de configuração, decidiu-se adicionar ferramentas de *debug*. No lado do servidor existem ficheiros de registo *logs* do *syslog* do sistema operativo RASPBIAN. No entanto, apesar de esse ficheiro ter toda a informação necessária para o *debug*, esta estaria misturada com outra informação, irrelevante para a aplicação, aumentando o tempo de procura da informação. Assim, decidiu-se desenvolver algumas funcionalidades, de forma a facilitar o desenvolvimento.

O módulo introduzido para gerar os *logs* em separado foi o *winston*. Com o *winston*, é possível gerar um ficheiro por dia, permitindo procurar num dia em específico e com o *timestamp*, gerado por registo. Desta forma consegue-se determinar a altura dos eventos ocorridos.

4.1.1 Base de dados

A Figura 19 apresenta as várias tabelas presentes na base de dados do sistema.

- A tabela da *Board*, guarda todo o objeto vindo da *cloud* que contem toda informação acerca dos *borders routers* que integram os sensores ambientais.
- A tabela do *Patient*, guarda toda a informação sobre os pacientes registados na VITABOX, assim como todos sensores biométricos a que lhe pertence.
- A tabela do *sensors*, guarda toda a informação acerca dos sensores ambientais registados.
- A tabela do *rawsensors* serve para guardar todos os valores dos sensores ambientais e biométricos, para depois serem enviados para a *cloud*.
- A tabela dos remotes, irá conter as configurações do comando da TV, sobre que funcionalidades cada tecla tem.
- A tabela das *settings*, guarda algumas das configurações da aplicação.
- A tabela do *iptables*, guarda todos os sensores que confirmam o registo.

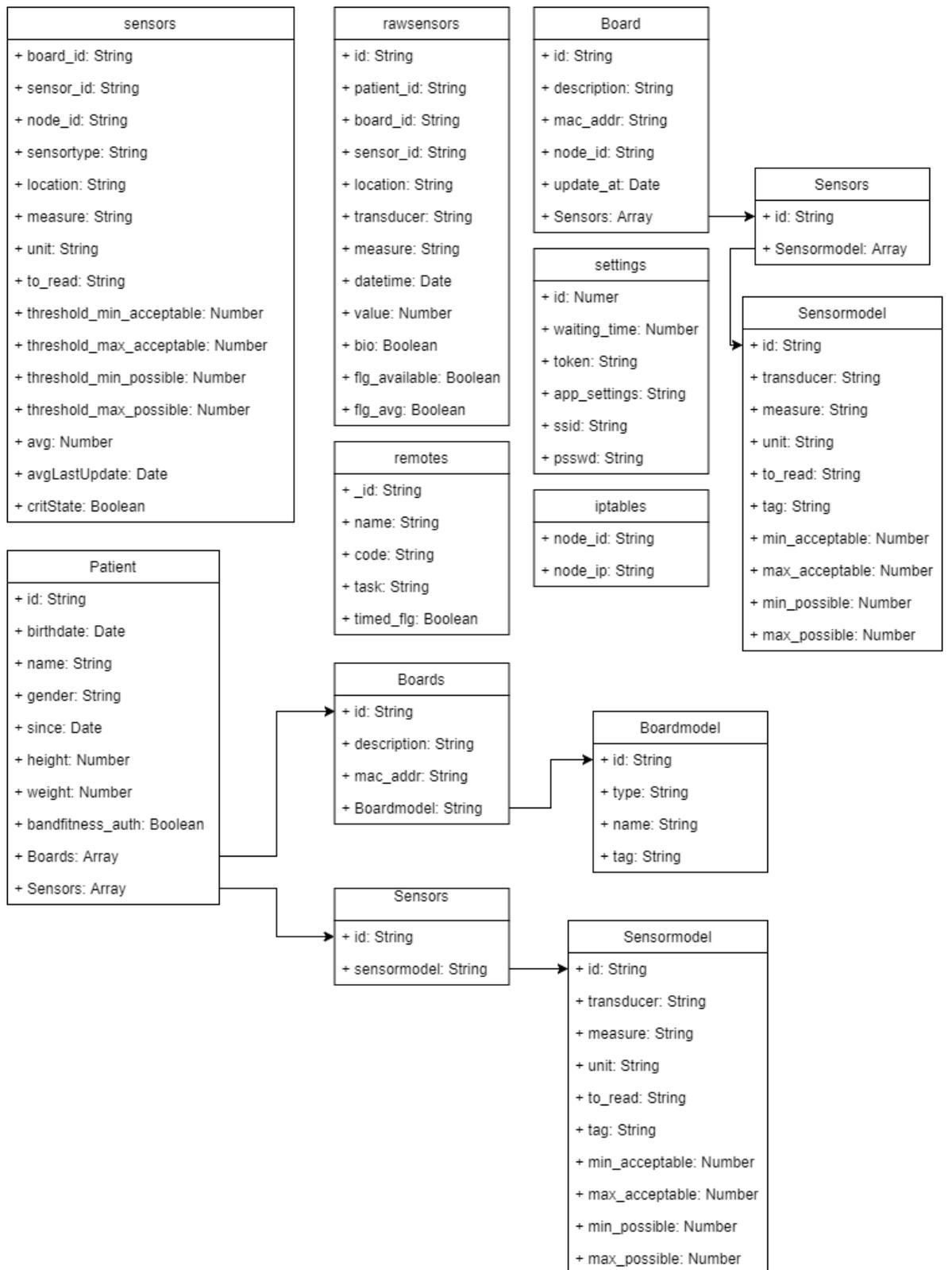


Figura 19 - Tabelas da base de dados

Durante o desenvolvimento, foi necessário popular a base de dados com dados temporários, para poder testar o comportamento da aplicação, enquanto a comunicação com a *cloud* e com os sensores não estava preparada, recorrendo às *seeds*.

As *seeds* em base de dados, é o processo de preencher os registos das tabelas antes de qualquer interação. Este é um processo automatizado usado na configuração inicial da base de dados.

Numa fase mais desenvolvida do projeto, tornou-se irrelevante o uso das *seeds*, vistos que já se tinha acesso a dados reais, para testar a aplicação.

Com a introdução dos modos de inicialização da aplicação, desenvolvimento e produção, percebeu-se a vantagem de usar as *seeds*, para popular as tabelas mais importantes da aplicação sem precisar de ligar com a *cloud* para o ser feito, como sendo o caso da tabela *remotes*, que define as funções de cada tecla do comando.

4.2 Comunicação VITABOX-TV

A comunicação entre a VITABOX e a TV é realizada com base no protocolo HDMI-CEC. Este permite interligação de vários dispositivos ao mesmo recetor de TV e a utilização do telecomando da TV para a interação da aplicação com o utilizador.

4.2.1 HDMI-CEC na VITABOX

O protocolo HDMI-CEC foi implementado na VITABOX através da biblioteca CEC-UTILS, que usa a livraria *libcec* como base para as suas funcionalidades, nomeadamente, o cliente CEC. A *libcec* é uma biblioteca para o CEC bus do HDMI. Permite aos utilizadores interagir com os dispositivos HDMI sem a necessidade de se preocupar com o *overhead*, *handshaking* e outras formas de mandar mensagem na comunicação.

A arquitetura de comunicação entre a VITABOX e a TV é apresentada na Figura 20. Através do NODE.JS, é adicionado um módulo (*nodecec*) que permite trabalhar com as funcionalidades do cec-utils a mais alto nível. O *nodecec* usa o comando cec-cliente para

interagir com o CEC, enviando e recebendo comandos. A ligação entre o *nodecec* e o *CEC-utils* é feita pelo *child-process* do NODE.JS.

O CEC-cliente também serve para fazer o *debugging* do sistema CEC, permitindo perceber o que se passa, dentro da informação que disponibiliza (Figura 20). O utilizador ao pressionar uma tecla do comando remoto, a TV recebe o comando associado a esse botão. A TV por sua vez envia o comando para o Raspberry Pi através do cabo HDMI, o *cec-utils* recebe o comando, o *nodecec* interceta-o e a aplicação interpreta o código.

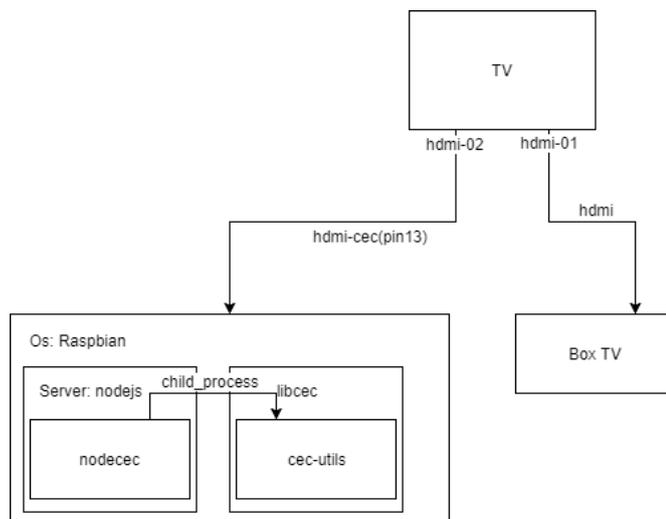


Figura 20 - Arquitetura HDMI-CEC na VITABOX

4.2.2 Testes ao HDMI-CEC

Durante o período de integração do HDMI-CEC, foram realizados testes de funcionamento com vários modelos de televisores. Foram utilizados modelos das marcas Samsung, Philips e LG.

A televisão da LG testada, não deu para interagir com a aplicação, devido a esta não ser compatível com o CEC.

Usando o televisor de marca Samsung, foi possível testar a aplicação uma vez que este já tinha suporte ao CEC. Durante os testes foram detetados vários problemas com o funcionamento da tecnologia. O primeiro problema foi o facto de por vezes não ser possível mudar de fonte de HDMI.

4.2.2.1 Teste à comutação

Uma possível solução foi formulada percebendo alguns comportamentos da ligação. A VITABOX necessita de estar ligada à porta HDMI 2 de forma a encontrar todas as fontes, nomeadamente o recetor de TDT. Os testes realizados para perceber o comportamento do CEC na TV, consistiram na avaliação das várias hipóteses de ligação entre a VITABOX e o recetor de TDT utilizando as fontes 1 e 2.

Nos testes realizados à TV da Samsung, percebeu-se que quando a fonte selecionada era a Fonte 1, o microcomputador que não iria encontrar a fonte 2, eliminado a possibilidade de comutação. No entanto, quando se liga o microcomputador à fonte 2, este já encontra a fonte 1, podendo assim realizar a comutação.

4.2.2.2 Teste à estabilidade da ligação

Os testes seguintes serviram assim para verificar a estabilidade do CEC. Esta foi avaliada através de três modos: i) reiniciando o servidor várias vezes, para deteção das anomalias; ii) colocando a TV em standby; iii) desligando a TV completamente.

Os testes mostraram que existia alguma instabilidade nas situações de standby da televisão e quando o servidor era reiniciando.

Experimentou-se verificar se dando algum tempo após o início do servidor, a ligação do CEC seria estabelecida. Após alguns testes, percebeu-se que a instabilidade, também não dependia do fator tempo, pelo que se teve de proceder a uma outra solução. Para isso, foi necessário verificar o estado atual durante o evento em que o CEC diz que está disponível, através do código 153. Este indica uma ligação desconhecida. Verificando o código e comparando com o estado atual da ligação, é possível forçar o estabelecimento da ligação com o CEC (Figura 21).

```
monitor.once(CECMonitor.EVENTS._READY, function () {
  setTimeout(function () { // corre o código após um certo tempo em ms
    if (monitor.GetPowerStatus(0) === 153) { // verifica o estado vendo o código 153 que é o unknown
      reconnect(2, 'vitaBox'); // função para forçar a ligação ao cec
    }
  }, waitingTimeTillCheckForCecIsAlive); // waitingTimeTillCheckForCecIsAlive variável de tempo em ms
});
```

Figura 21-Evento a informar a disponibilidade do cec.

Na solução proposta aproveitaram-se as funcionalidades do *child_process*. O *child_process* tem a capacidade de criar novos processos de um processo em execução. Este módulo ao ser usado, cria três canais de escuta, *stdin*, possível enviar dados, *stdout*, visualiza o estado do processo e o *stderr*, captura os erros ocorridos associados ao processo. Estes canais são estabelecidos de forma a conseguir interagir com o processo filho. Usando a função *exec* do *child_process* é possível executar um comando em *shell*, que irá servir para forçar a ligação do CEC (Figura 22).

```
/**
 * TODO: Força a ligação com cec
 * @param { Number } hdmi numero do canal de hdmi a ser usado
 * @param { String } osdName nome do canal cec a ser aberto
 */
function reconnect(hdmi, osdName) {
  let restartCec = exec('cec-client -p ' + hdmi + ' -o ' + osdName); //comando a ser executado em shell
  restartCec.stdout.on('data', (data) => { // captura toda a informação do do processo
    successlog.info(`Restarting Cec: ${data}`);
  });

  restartCec.stderr.on('data', (data) => { // captura a informação de problemas que acontecem no processo
    errorLog.error(`Could Not Restart Cec: ${data}`);
  });
}
```

Figura 22 - Função para forçar ligação do CEC.

Após a implementação da função para forçar a ligação do CEC, procedeu-se a novos testes de estabilidade voltando a testar a TV Samsung em *standby* e o reiniciar do servidor. Verificou-se que os resultados, nestes testes foram todos positivos, tendo o CEC agora estável.

Com a introdução de um novo recetor de TV da marca Philips, foi necessário realizar novos testes às funcionalidades CEC. Como é uma norma, cujas funcionalidades dependem do fabricante, os problemas existentes da Samsung, poderiam não ser os mesmos.

Dos novos testes realizados, percebeu-se que o problema que existia com o recetor Samsung, nomeadamente da ligação dos vários dispositivos às portas já não existia. Ou seja, neste caso é irrelevante em qual das fontes da TV é realizada a ligação.

Com a Philips, o *standard* do CEC mudou, sendo possível usar outras funcionalidades relativamente ao recetor Samsung. Nomeadamente, é possível lançar uma mensagem para a TV, independentemente da fonte a que estiver ligada, dando possibilidade envios de notificações da VITABOX para a TV.

4.2.3 Notificações

Com a funcionalidade OSD String da TV Philips, foi possível a introdução de notificações no projeto. O sistema de notificações servirá, por exemplo, para avisar o utilizador em situações onde não existem registos dentro do tempo limite, conforme indicação médica, ou qualquer outro tipo de lembretes que o sistema possua implementado.

A OSD String, nesta TV, imprime no ecrã, no canto inferior direito, uma pequena mensagem, com um limite máximo de treze caracteres e com um máximo de tempo de exposição de dois segundos (Figura 23).



Figura 23 - OSD String no ecrã da TV

Tendo em conta as limitações, nomeadamente o tempo de vida da mensagem, a solução encontrada consistiu em executar um ciclo que lança a mensagem novamente num período fixo de três mil e quinhentos milissegundos. Com este período, consegue-se que uma visualização natural a mensagem no ecrã.

4.3 Comunicação com sensores biomédicos

Na comunicação com os sensores biométricos, usou-se o módulo em NODE.JS e o NoBLE. Este módulo serve para comunicar com a biblioteca de Bluetooth, disponibilizando as ferramentas necessárias para esta comunicação.

Para facilitar a utilização do módulo NoBLE, foi implementado uma biblioteca em *NODE.JS* para disponibilizar as ferramentas necessárias para a aplicação, o *Blelib*. As

funções desta biblioteca são responsáveis por facilitar a ligação entre os dispositivos BLE com a VITABOX (Figura 24).

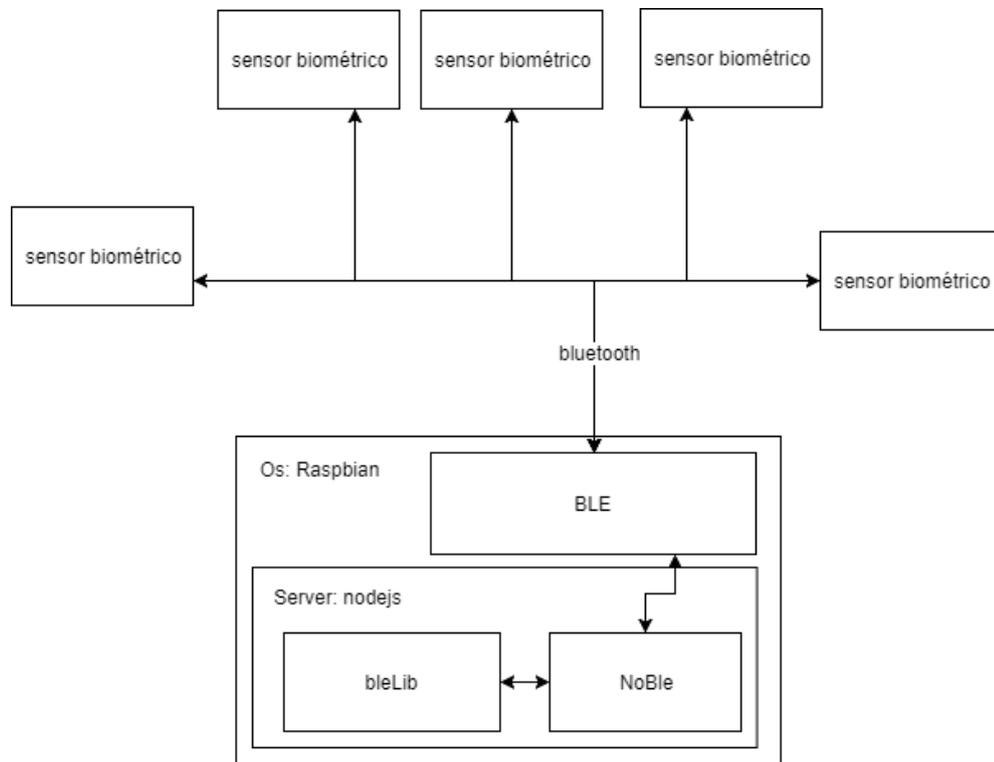


Figura 24 – Comunicação com os dispositivos biométricos

A VITABOX para poder fazer uso dos sensores biomédicos, precisa de ter acesso à informação acerca do utilizador existente na base de dados do servidor.

4.4 Comunicação com rede de sensores ambientais

Na comunicação com a rede de sensores usou-se um módulo nativo do NODE.JS, o *datagram*, para uma comunicação *UDP* com o *Border Router*. Este é implementado num ZOLERTIA ligado diretamente ao Raspberry Pi3. A introdução do *Border Router* foi necessária pelo facto de que a comunicação por *wireless* dos sensores através do ZOLERTIA e o Raspberry Pi3 ser incompatível, uma vez que a rede de sensores usa 802.15.4 e o Raspberry usa 802.11. Com o *Border Router* é criado um túnel para estabelecer a comunicação com a rede 802.15.4 (Figura 25).

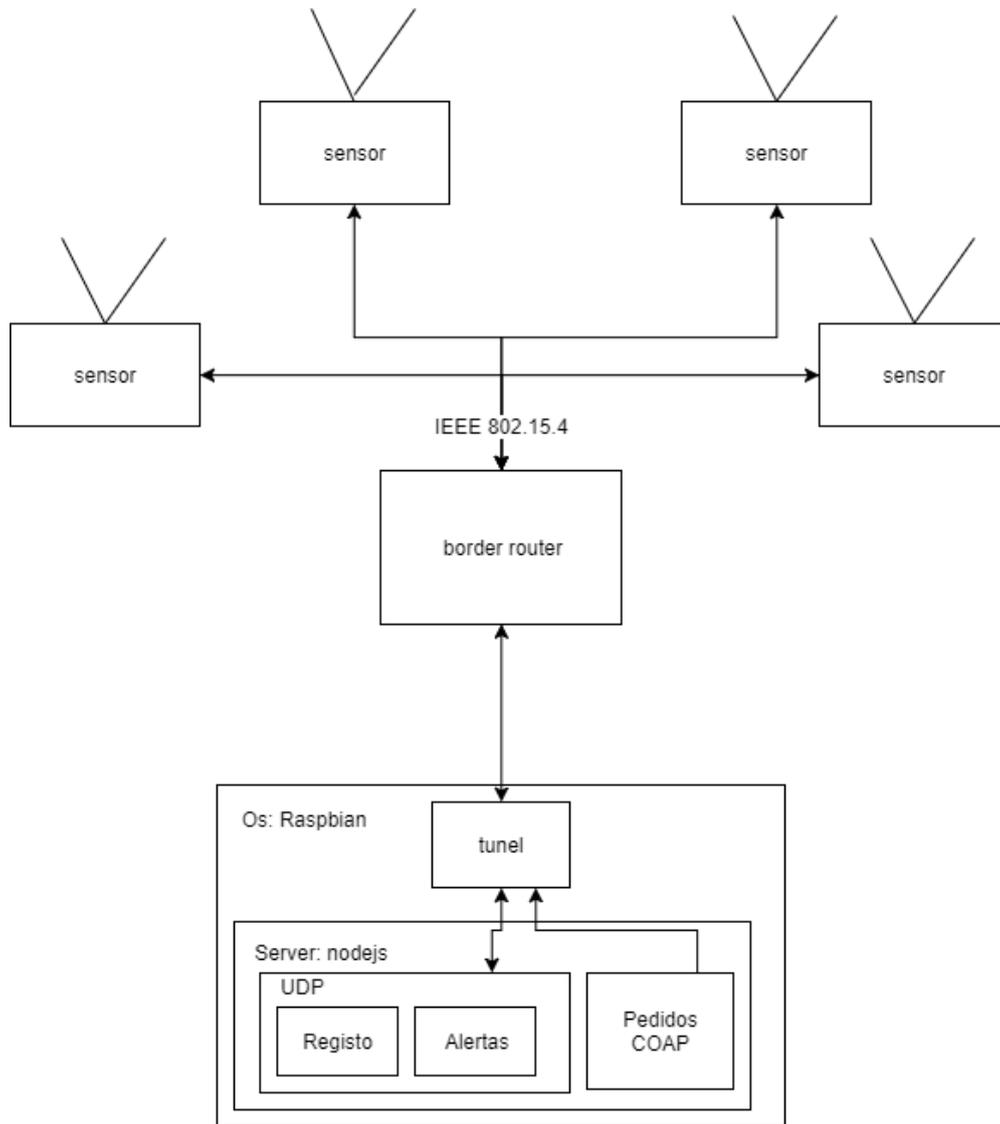


Figura 25 – Rede de sensores

Para a VITABOX realizar pedidos de dados aos sensores pela utilizou-se o protocolo *COAP* que tem funções similares ao HTTP, podendo ser facilmente usado em dispositivos com recursos restritos, tipicamente numa arquitetura IoT.

A VITABOX, através do NODE.JS, irá gerir toda a comunicação com a rede de sensores. Para isso, existem três processos de comunicação: i) registo dos sensores; ii) leitura dos valores medidos pelos sensores; iii) notificações de alerta feita pela rede de sensores em situações de medição de parâmetros fora do normal.

4.4.1 Registo dos sensores

Antes de o pedido de dados dos sensores, é necessário realizar o seu registo, para garantir que o utilizador apenas usa os que foram registados na *cloud*. A VITABOX recebe uma lista de todos os sensores atribuídos. Com essa lista, usando uma ligação UDP, a VITABOX fica à escuta de sensores que se tentem registar. Da lista disponibilizada, é realizada uma pesquisa dos sensores ativos.

Caso o sensor conste da lista, a VITABOX envia uma mensagem a autorizar o registo do sensor. Este é introduzido na tabela dos sensores registados *iptables* da base de dados. Caso não seja aceite, é enviada uma mensagem de rejeição (Figura 26).

```
// faz o substitui da string recebida
message = message.toString().replace('fe80', 'fd00').replace(/\0[\s\S]*$/g, '');
// divide a string pelos : guardando-a num array
var nodeidRaw = message.toString().split(":");
// agarra na ultima posição do array e guarda numa variável
var nodeid = nodeidRaw[nodeidRaw.length - 1].replace(/\0[\s\S]*$/g, '');
successlog.info(`NODE List ---- ${self.nodeIdList}`);
// procura na lista de sensores se existe o sensor a tentar registar-se
if (self.nodeIdList.find(nodeid)) {
    // cria uma variavel com uma mensagem
    var message1 = "flag1:" + nodeid;
    // envia a mensagem em como aceita o sensor
    self.server.send(message1, 0, message1.length, remote.port, remote.address, function (err, bytes) {
        if (err) throw err;
        successlog.info(`NODE ACCEPTED ---- ${nodeid}`);
        successlog.info(`FLAG SENT ---- ${message1}`);
        // envia o sensor registado para ser guardado para a tabela de sensores aceites
        process.send({
            ip: message,
            node_id: nodeid
        });
    });
} else {
    var message1 = "flag2:" + nodeid;
    // envia a mensagem em como rejeita o sensor
    self.server.send(message1, 0, message1.length, remote.port, remote.address, function (err, bytes) {
        if (err) throw err;
        successlog.info(`NODE DECLINED ---- ${nodeid}`);
        successlog.info(`FLAG SENT ---- ${message1}`);
    });
}
```

Figura 26 - Registo dos sensores.

4.4.2 Leitura dos sensores

O acesso aos sensores através de CoAP pressupõe a consulta da lista *iptables* com os endereços IP dos sensores registados na base de dados. Estes pedidos são realizados periodicamente. Se existem sensores na lista, é chamada uma função para realizar os pedidos COAP para a leitura dos sensores, onde se passa por parâmetro, o endereço IP e o ID do sensor (Figura 27).

```
/**
 * TODO: Começo do ciclo para os pedidos dos dados aos sensores
 * @param { }
 */
ServerBoardRegisterRevised.prototype.start = function () {
```

Figura 27 - Função para os pedidos COAP.

Quando se recebem valores dos transdutores do sensor por CoAP, é construído uma entidade designada por “*objeto do sensor*”, usando a função *buildObject*. Esta função de construção recebe por parâmetro, o tipo de *board* de sensores, o valor do transdutor, a informação desse transdutor existente na lista de sensores e um *array* que será preenchido com todos os transdutores da *board* (Figura 28).

```
/**
 * TODO: Criação dos objetos para a inserção dos dados no rawsensors e para o post na cloud
 * @param { String } sensortype tipo de transdutores
 * @param { Number } value valor dos transdutores
 * @param { Array } data lista de objetos dos sensores
 */
function buildObject(sensortype, value, listData, data) {
```

Figura 28 - Construção do objeto do sensor

Quando o objeto do sensor está construído, são executados dois procedimentos. Um será de inserção dos registos dos transdutores na base de dados na tabela *rawsensors* e a outra é a atualização dos valores na tabela *sensors*, para posteriormente serem apresentados na interface gráfica do utilizador. A tabela de sensores é atualizada com a média dos valores adquiridos, sendo o período de amostragem configurável. Adicionalmente é possível aceder e apresentar o valor instantâneo do transdutor.

4.4.3 Alertas gerados pela rede de sensores

No caso dos alertas gerados pela rede de sensores, o servidor fica à escuta dos valores a serem enviados. Os alertas ocorrem quando os valores ultrapassam os limites pré-definidos. Ao receber os dados dos sensores, é construído o objeto a ser enviado para ser registado na tabela dos *rawsensors*. Após o registo é verificado o sensor que está em estado crítico, enviando a informação para a aplicação (Figura 29).

```
/**
 * TODO: Criação dos objetos para a inserção dos dados no rawsensors
 * @param { String } node_id identificação do sensor
 * @param { String } sensortype tipo de transdutores
 * @param { Number } value valor dos transdutores
 * @param { Array } list lista de objetos dos sensores
 */
buildObject(node_id, sensortype, value, list) {
```

Figura 29 - Criação do objeto em estado crítico.

4.5 Módulo de comunicação com a *cloud*

A aplicação e a *cloud* comunicam utilizando o protocolo HTTPS para a troca de dados através de uma API desenvolvida no âmbito do projeto. No entanto, para envio das notificações, a comunicação é realizada usando *websockets* baseado no *socket.io* do NODE.JS (Figura 30).

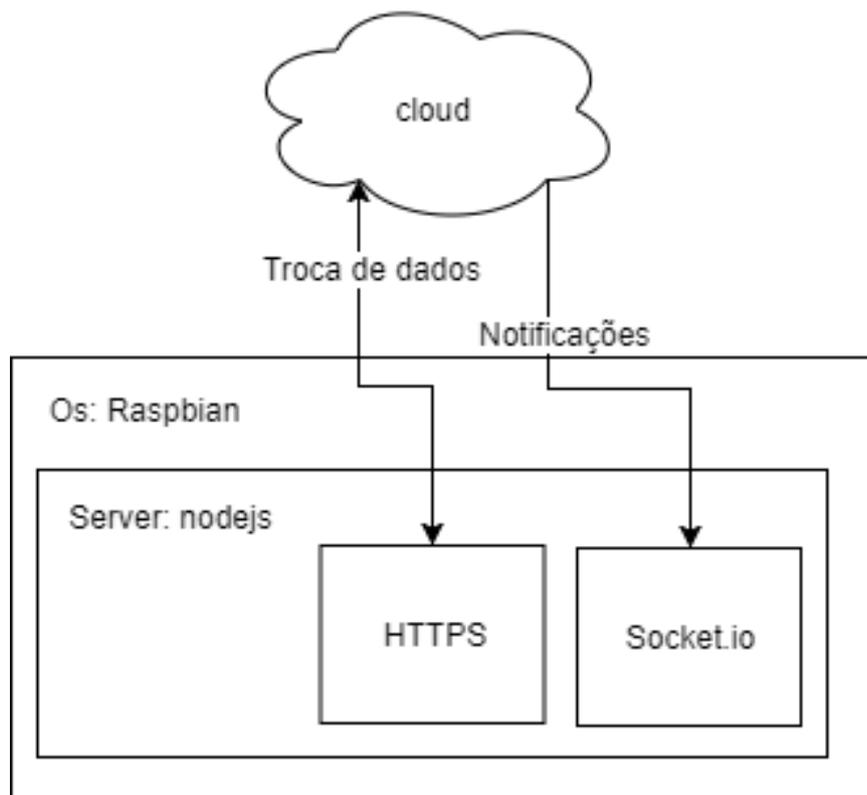


Figura 30 - Comunicação com a *cloud*.

Os dados transferidos pela VITABOX estão no sistema de informação e referem-se aos dados dos pacientes, sensores biométricos e ambientais registados e a respetivas identificações.

Os dados pertencentes aos pacientes registados na VITABOX, são enviados em formato JSON (Figura 31).



Figura 31 – Endpoint dos pacientes.

A Figura 32 apresenta a interface (*endpoint*) para o pedido dos dados pertencentes aos sensores biométricos e ambientais registados a uma VITABOX.

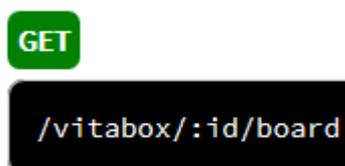


Figura 32 – Endpoint das boards

A Figura 33 apresenta a interface para a solicitação de um token para a VITABOX, que serve para validação do seu registo.



Figura 33 – Endpoint dos tokens

Depois de possuir a informação sobre os dispositivos, a tabela *rawsensors* é usada para guardar os valores adquiridos pelos sensores biométricos e ambientais.

Foi criada uma função para gerir o envio dos valores adquiridos. Periodicamente é verificado se existem novas leituras a serem enviados para a *cloud*. Independentemente do tipo de dados enviados, a VITABOX mantém os cem valores mais recentes (Figura 34). Caso não existam dados novos a serem enviados, o tempo de atualização aumenta.

```

/**
 * @method postSensorData
 * @description envia os dados para a cloud
 * @param { Function } callback
 */
RawSensor.prototype.postSensorData = function (callback) {

```

Figura 34 - Função de envio dos registos para a *cloud*.

4.6 Conversão texto para fala (*text-to-speech*)

As escolhas iniciais para serem usadas para a conversão *text-to-speech* foram a *responsivevoice.js* (ResponsiveVoice.JS, s.d.) a aplicação nativa do browser GOOGLE CHROME. No entanto, estas soluções dependem de ligação à rede, pelo que foram abandonadas.

Das soluções *offline*, experimentou-se a aplicação *espeak* (eSpeak, s.d.) e a aplicação *festival* (Festival Text-to-Speech Online Demo - Technical, s.d.). O problema com estas duas soluções foi o facto de a voz reproduzida pela *espeak* ser demasiado robótica, de difícil compreensão. A reprodução da aplicação *festival*, melhorava um pouco, no entanto não possuía suporte para a língua portuguesa. Após testar várias variantes do *text-to-speech* baseados em *espeak*, decidiu-se criar uma solução de raiz que funcionasse em modo *offline*.

A aplicação desenvolvida, é baseada no uso de bibliotecas em NODE.JS, de forma a acelerar o processo de desenvolvimento do algoritmo. Foi usada a biblioteca *audioconcat*, que realiza a interação com as ferramentas do *ffmpeg*, nomeadamente a concatenação dos ficheiros de áudio.

O algoritmo *text-to-speech* desenvolvido recebe um texto que deverá ser reproduzido através de áudio.

Numa primeira fase, foi necessário converter a *string* de texto num *array* separando todas as palavras e números pelos espaços.

O texto a converter foi separado em duas categorias: i) conversão de palavras; ii) conversão de números. O fluxograma do algoritmo de conversão *text-to-speech* está descrito na Figura 35.

Na conversão das palavras em áudio, é necessário em primeiro lugar que se proceda a uma limpeza do texto, retirando os caracteres especiais que possam existir. Deste modo obtém-se um texto mais fácil de converter. De seguida é-lhe atribuída um caminho ao ficheiro de áudio correspondente que é inserido no *array* (Figura 36).

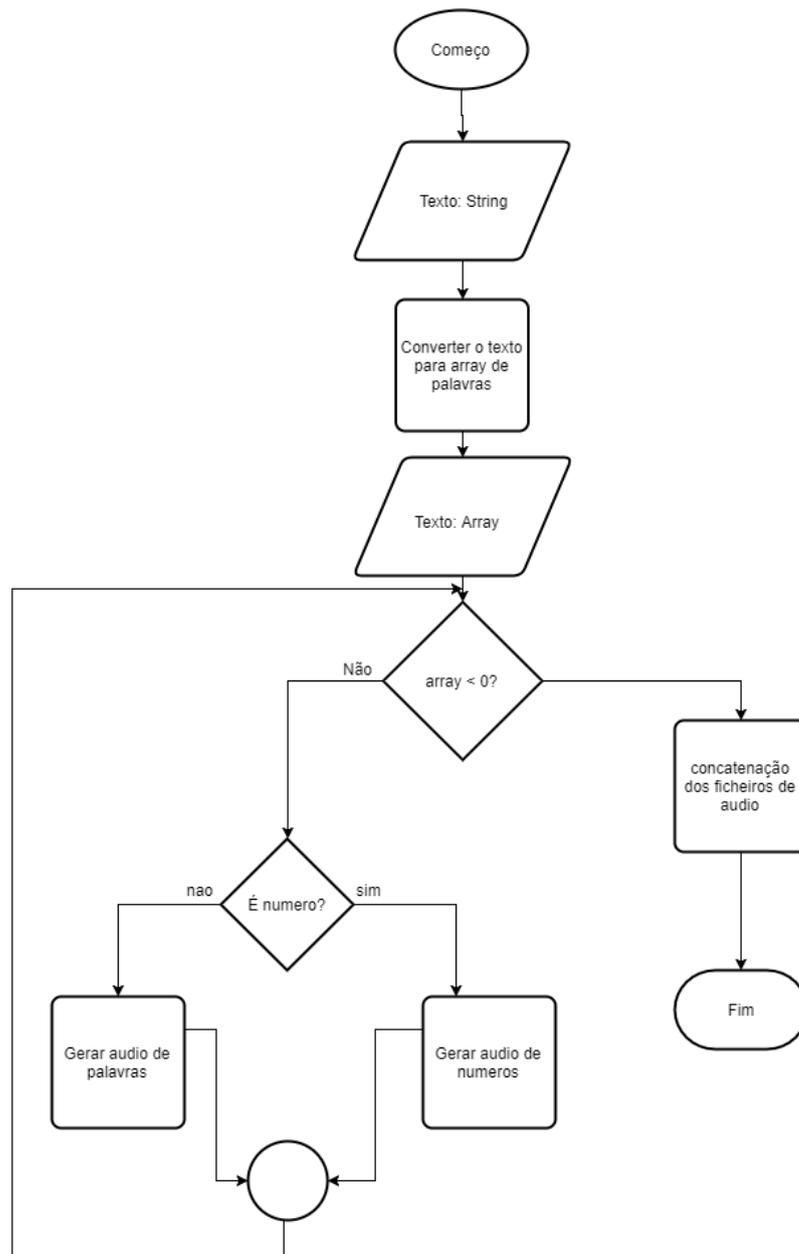


Figura 35 - Algoritmo texto para fala.

No processo de geração dos números para ficheiros de áudio, foi necessário dividir o número nos seus vários componentes, decimais, dezenas, centenas. Com esta separação, o número é convertido na *string* correspondente e é-lhe atribuída o caminho de um ficheiro de áudio pré-definido. O caminho é guardado num *array*, organizado pela mesma ordem do *array* inicial do texto.

A VITABOX tem uma biblioteca de ficheiros de áudio, esta biblioteca contém os números e as palavras individuais a ser usados. Com estes ficheiros de áudio e a lista de caminhos para esses subficheiros de áudio, a aplicação *audioconcat* sintetizar um único ficheiro áudio em mp3.

Na concatenação dos vários ficheiros de áudio num único ficheiro, usou-se uma das funcionalidades do *ffmpeg*, a concatenação.

Com ajuda do módulo *audioconcat*, é realizada a integração da funcionalidade de concatenação de uma forma simples de usar, sendo apenas necessário fornecer alguns parâmetros à função disponibilizada pelo módulo. Estes correspondem à lista de todos os caminhos para os ficheiros de áudio e o caminho para onde o ficheiro final vai ser gerado um novo ficheiro áudio .mp3 usando o *ffmpeg*, assim como o seu nome (Figura 36).

```
/**
 * TODO: Concatenação dos ficheiros de áudio
 * @param { Array } arr lista de caminhos para os ficheiros de áudio
 * @param { Function } callback
 */
function concatAll(arr, callback) {
```

Figura 36 - Concatenação dos ficheiros de áudio.

Ao juntar todas as palavras ficheiros num só, existe a vantagem de melhorar a qualidade e rapidez na reprodução do áudio gerado, quando comparado com a reprodução dos ficheiros de forma individual.

5 Interação baseada na TV

O capítulo 5 apresenta a aplicação de interação baseada na TV. São abordados os aspetos visuais e de interação implementados, assim como a descrição das opções tomadas para o seu desenvolvimento.

5.1 Introdução

Pretende-se que a interação baseada na TV seja simples, clara e concisa. Esta resulta de uma aplicação web, cuja seleção é realizada com recurso ao comando da TV. Este facto é determinante no modo como a aplicação é construída uma vez que reduz os passos possíveis para a realização de uma tarefa, quando comparada com outros tipos de interfaces como o rato ou ecrãs tácteis. Todo o layout da aplicação é pensado tendo em conta este princípio e também procurando menus e funcionalidades o mais simples possível. Adicionalmente o sistema suporta geração de mensagens áudio para ajuda na interação.

5.2 Interação visual

Na interação visual, o utilizador tem disponíveis vários periféricos, sendo o principal, o comando da TV (Figura 37). Através deste dispositivo o utilizador irá controlar a interface da aplicação disponibilizada.



Figura 37 - Comando da TV

A presente solução assume a existência de um recetor de televisão digital terrestre (TDT) ou uma box de TV por cabo conectados ao televisor. Isto significa que poderão existir dois comandos remotos: um para controlo da TV/VITABOX e um outro para a box TDT/TV_Cabo.

A descrição seguinte centra-se no comando remoto TV/VITABOX que controla a aplicação da VITABOX.

As teclas disponíveis ao utilizador no comando da TV (Figura 38), são as setas de direção, para navegar na aplicação, o botão de “ok” para seleccionar o menu, o botão de saída, para permitir sair do menu e a tecla 1 para mudança entre fontes HDMI.



Figura 38 - Comando da TV com as teclas seleccionadas

A fim de facilitar o uso do comando remoto, o utilizador tem hipótese de observador no ecrã da TV uma representação virtual do próprio telecomando onde se assinala o botão pressionado.

5.1 Menus da Aplicação

A aplicação tem vários menus de interação: o menu para a execução dos exames, o menu para o histórico das medições executadas, o menu para visualização do estado atual dos sensores, o menu de bem-vindo, que inclui as notificações e o menu para visualização dos alarmes.

5.1.1 Bem-vindo

O menu do bem-vindo (Figura 39), está dividido em duas seções, uma para a informação sobre as farmácias de serviço e o tempo, a outra para as notificações.

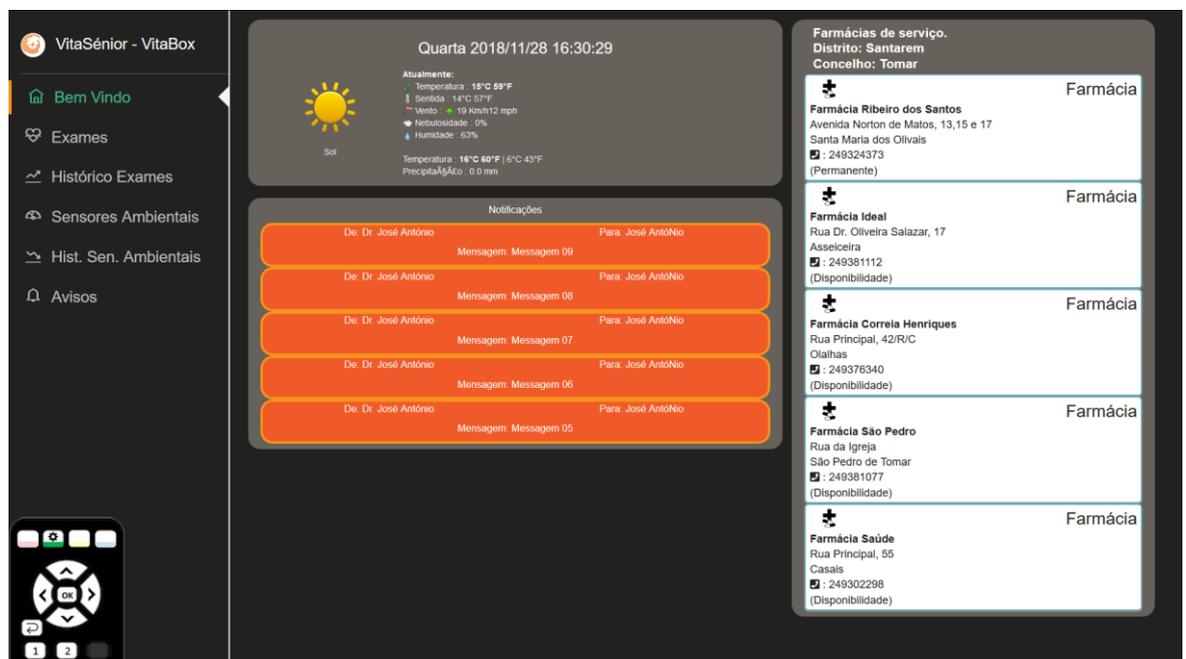


Figura 39 - Menu Bem-Vindo.

As farmácias de serviço são um recurso facultado por uma terceira parte e permite ter acesso às farmácias disponíveis dependendo da localização da VITABOX, considerando a localidade e o distrito. A informação meteorológica apresentada também está referenciada à localidade.

As notificações são apresentadas em caixas de texto, tendo o utilizador acesso às últimas cinco. Existem dois tipos de notificações, os avisos ao utilizador a relembrar que exames tem de realizar e as mensagens do médico.

Quando a mensagem de rodapé aparece significa que o utilizador terá de carregar no botão de ok para confirmar em como tomou conhecimento das notificações novas.

5.1.2 Exames

No menu dos exames (Figura 40), ao entrar, o utilizador dispõe de uma lista de utilizadores a ser seleccionado, devendo seleccionar o seu perfil.

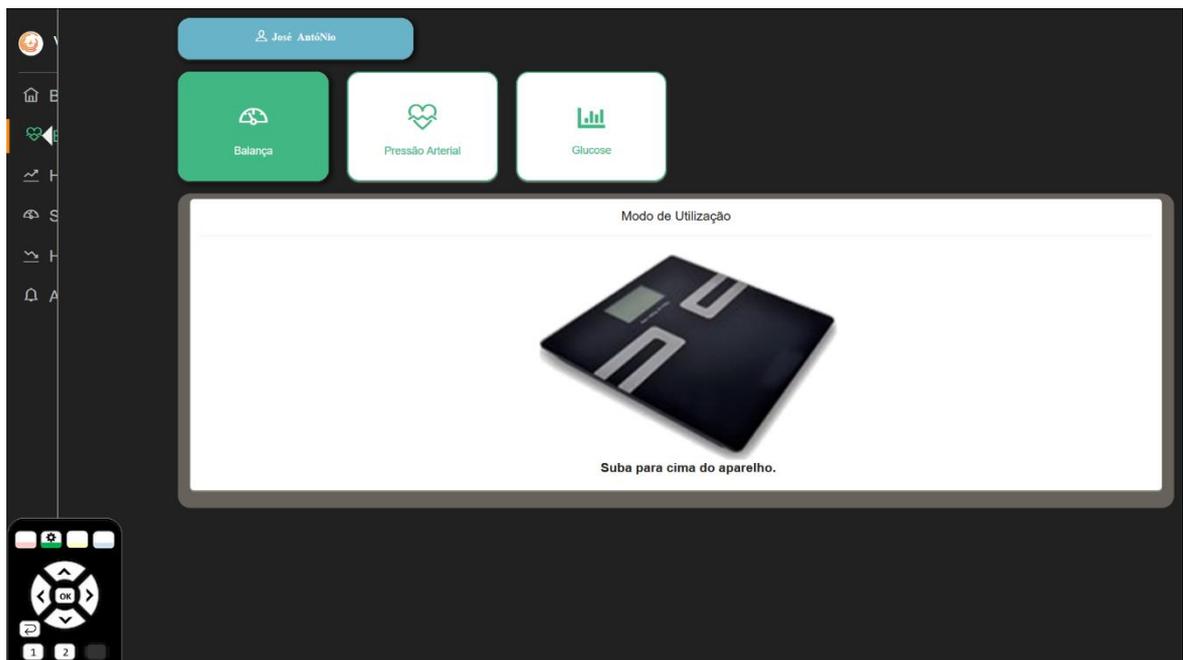


Figura 40 - Menu dos Exames.

Ao seleccionar o utilizador, é apresentado uma lista de todos os sensores biométricos associados. O próximo passo é a seleção do exame a ser executado, e para isso conta com um guia de procedimentos para a realização do exame.

Após a realização do exame, o valor é apresentado no ecrã e é enviado para a base de dados local. O utilizador pode de seguida escolher se pretende repetir esse exame ou se pretende fazer um exame diferente, ou até mesmo sair desse menu. Para sair do menu é só clicar no botão para sair.

5.1.3 Histórico dos exames

No menu do histórico dos exames (Figura 41), o utilizador pode consultar os resultados obtidos. Os procedimentos são semelhantes ao menu dos exames.

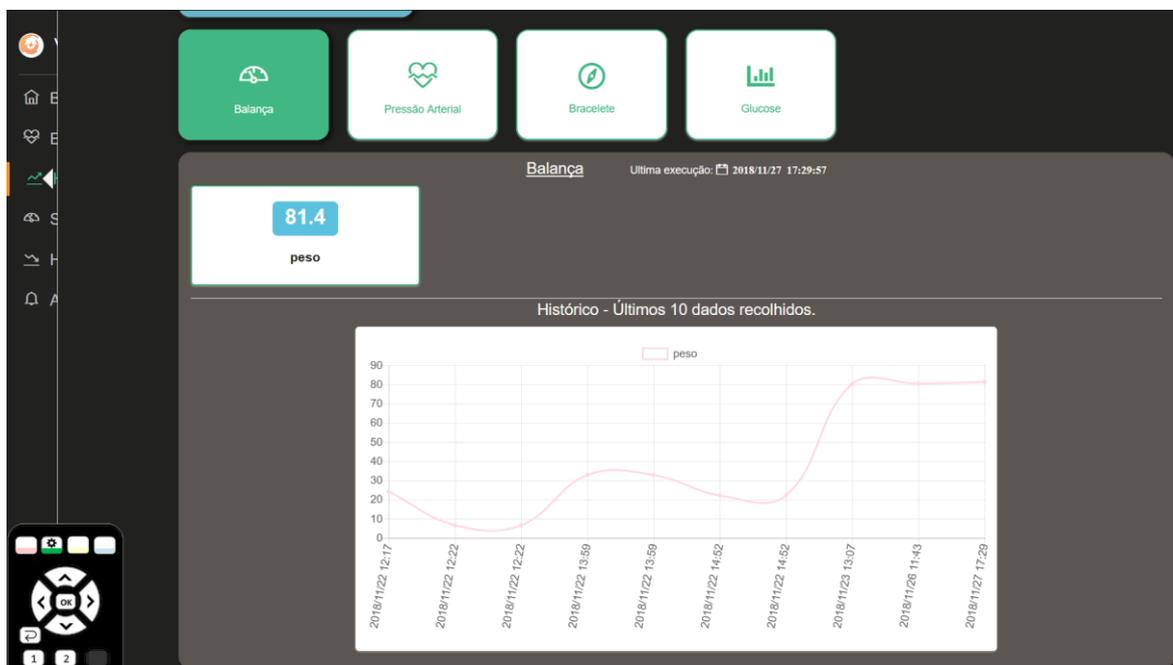


Figura 41 - Histórico dos sensores biométricos.

Ao entrar, tem-se uma lista de utilizadores e selecciona-se o que lhe pertence. Ao seleccionar o utilizador é apresentado uma segunda lista, com todos os sensores biométricos associados. Cada sensor da lista contém os valores dos exames realizados, sendo apresentado sob forma de gráfico para melhor comparação.

No histórico inclui-se dados do batimento cardíaco, oriundos da bracelete *miband*, que é o único dispositivo cuja recolha de informação é automática, bastando estar no raio de alcance do Bluetooth. A recolha das medições do batimento cardíaco (heart rate) armazenadas no bracelete ocorrem de meia em meia hora.

5.1.4 Sensores Ambientais

O menu dos sensores ambientais (Figura 42) mostra o estado atual de todos os sensores registados na VITABOX.

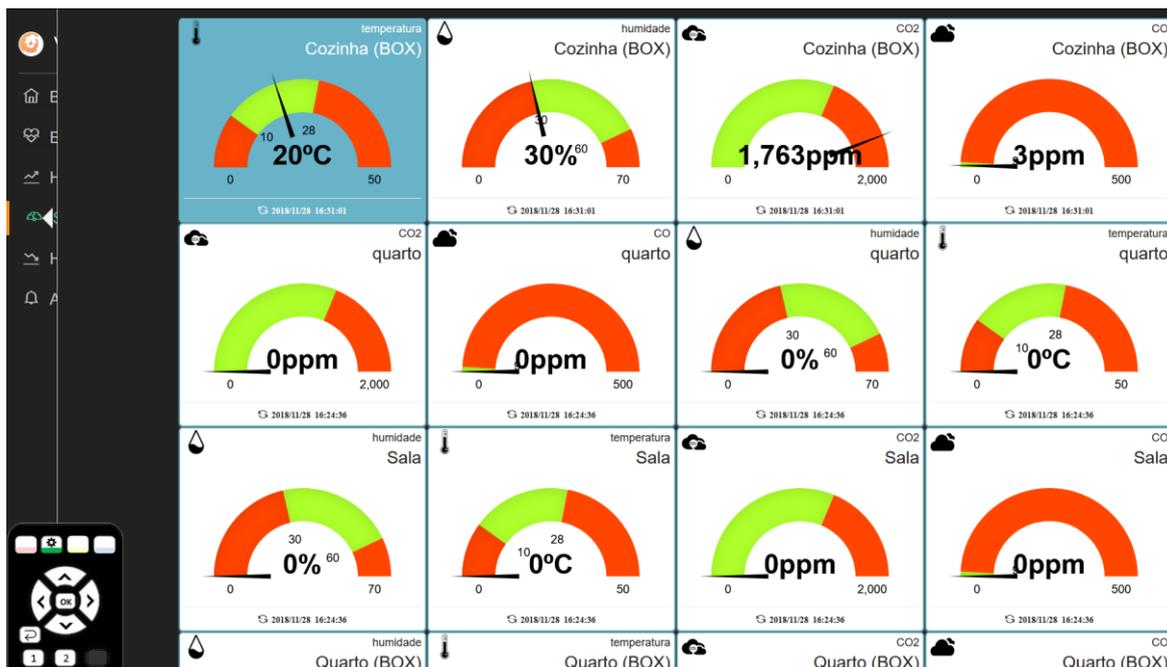


Figura 42 - Menu dos sensores ambientais

A informação é disposta em cartões, estando cada sensor identificado através de um símbolo elucidativo do parâmetro a medir, localização na casa, gráfico com a medição atual, limites e data da última atualização.

É possível interagir com estes cartões, através das setas de direção do comando, podendo navegar entre os vários sensores, tendo um áudio guia para assinalar a informação vital.

5.1.5 Histórico dos sensores ambientais

O utilizador pode verificar o histórico dos sensores ambientais (Figura 43), sendo que este menu é apresentado de forma semelhante ao do histórico dos sensores biométricos.



Figura 43 - Histórico dos sensores ambientais

É listada todos os tipos diferentes de sensores que pertencem à VITABOX e desta lista é seleccionada um deles.

A segunda lista apresenta todas as divisões onde está instalado o sensor do tipo que foi escolhido, mostrando um gráfico com todos os valores desse sensor nas últimas vinte e quatro horas.

5.1.6 Alertas

O último menu, é onde todos os sensores em estado crítico vão aparecer, formando uma lista semelhante ao do estado atual dos sensores ambientais.

A lista dos sensores em estado crítico, correspondem aos sensores que ultrapassam os máximos ou os mínimos definidos, sendo que quando isso acontece, independentemente do que o utilizador esteja a fazer, este é interrompido para que seja apresentado este menu importante.

Esta interação permite que o utilizador possa ter uma reação mais rápida em casos de emergência, e este aviso não desaparece enquanto o utilizador não seleccionar o botão “ok” provando que viu a mensagem e percebeu do estado em que se encontrava.

5.2 Modais em uso

Na aplicação, decidiu-se usar modais de forma a poder simplificar o uso de certas funcionalidades, sem adicionar mais menus de forma a não confundir o utilizador. Exemplo disso é o caso dos alertas que mostra ao utilizador que algo não se encontra bem (Figura 44).

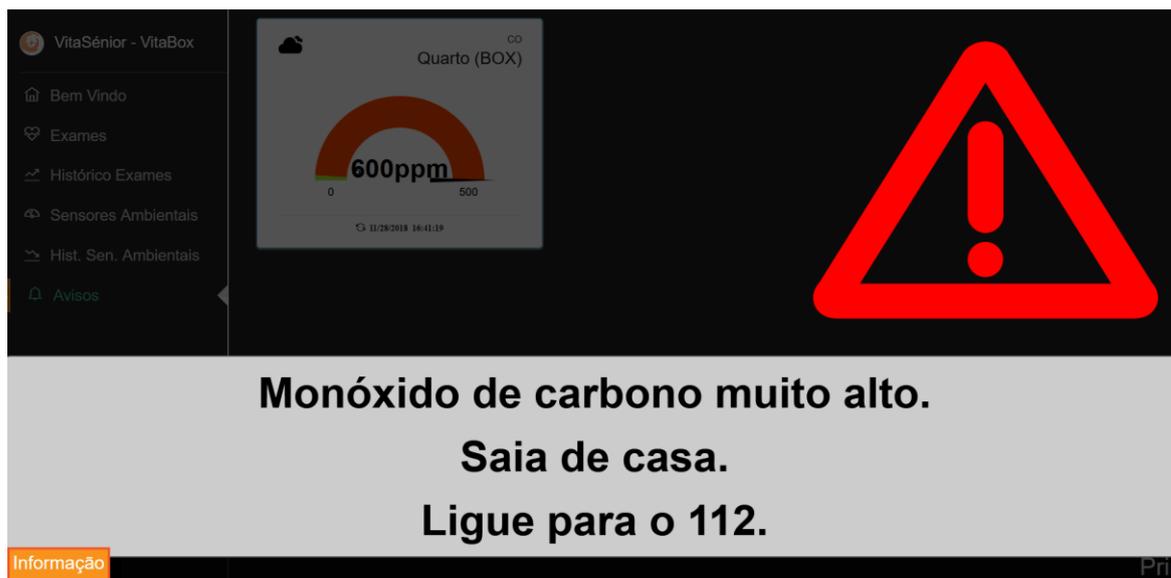


Figura 44 - Alertas dos sensores ambientais.

A modal dos alertas é composta de duas partes: i) o visual que vai piscando para chamar a atenção do utilizador; ii) e uma segunda que apresenta o procedimento que o utilizador terá que seguir caso a situação ocorra. Esta é seguida do áudio texto que dita todos os passos escritos no ecrã. Uma modal é um componente posicionado sobre alguma coisa no documento e é removido a ação do *scroll* sobre esse documento.

A modal do *wifi* (Figura 45) serve para configurar o *wireless* na aplicação sem precisar de sair da aplicação, facilitando a configuração da conectividade da VITABOX.

Nesta modal são apresentadas todas as redes disponíveis que o microcomputador consegue detetar. Depois de selecionado a rede é pedida a inserção da *password*.



Figura 45 - Modal configuração do *wifi*.

O utilizador terá acesso às configurações através do botão verde para poder escolher o comportamento da aplicação.

A modal das configurações (Figura 46) permite ao utilizador poder parametrizar a aplicação, por exemplo ligar ou desligar o áudio texto, seleccionar os modos de avançado e básico da aplicação, a linguagem ou aceder ao modal do wifi.

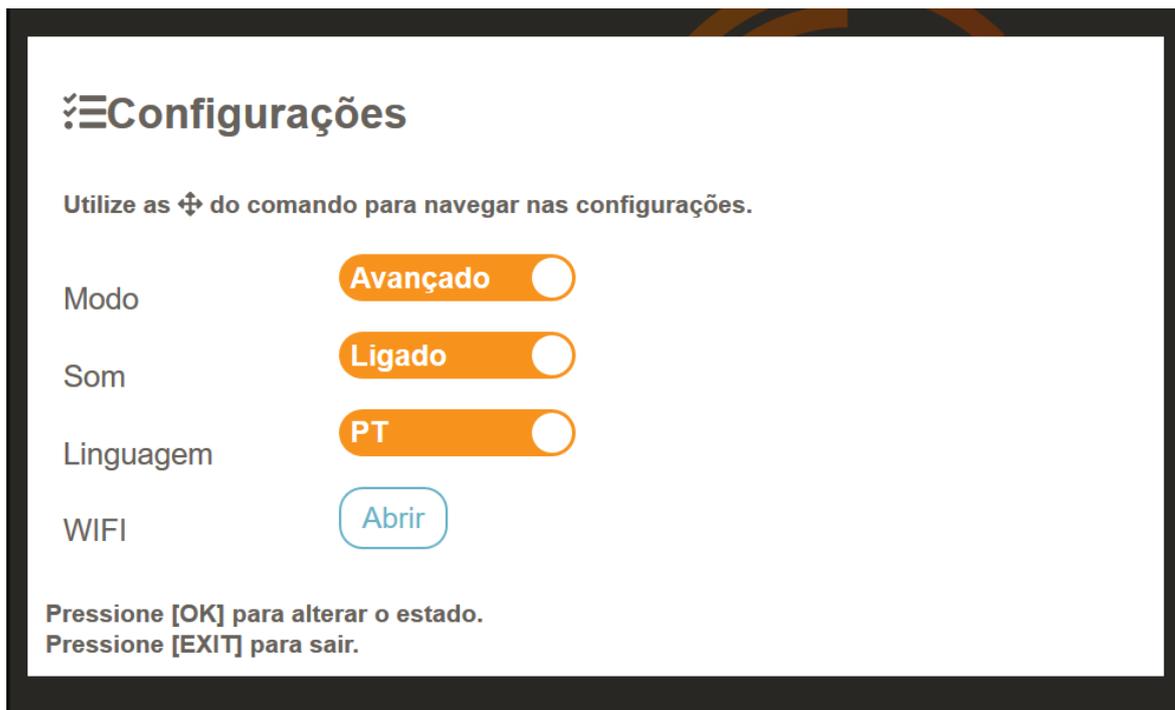


Figura 46 - Modal Configuração.

A opção para desligar o texto para fala, desativa todo o guia áudio texto, em toda a aplicação exceto nos avisos do estado crítico dos sensores.

A escolha entre a linguagem de PT ou EN pretende facilitar a introdução de textos dinâmicos para o *texto-to-speech*. A funcionalidade da tradução fica preparada para poder mudar de linguagem.

Os modos básico e avançado, foram criados por forma a poder incluir utentes com extrema dificuldades motoras ou de compreensão, simplificando ao máximo a interação manual.

No modo avançado (Figura 47) são disponibilizadas todas as funcionalidades da aplicação, como os históricos dos sensores ambientais e biométricos, e é retirado o modo automático no menu dos sensores ambientais.

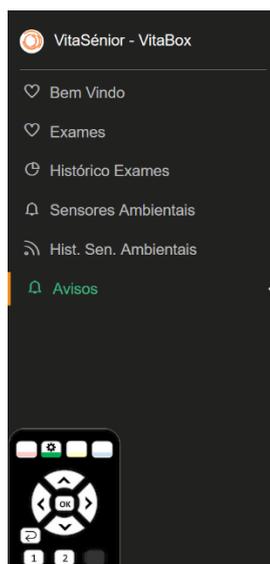


Figura 47 - Menu avançado.

No modo básico (Figura 48) é introduzido o sistema de leitura dos componentes automáticos no menu dos sensores ambientais. Esta leitura é feita percorrendo os cartões, lendo a informação essencial, para que o utilizador não precise de navegar e poder usufruir do menu.

A leitura automática, só se encontra disponível no menu dos sensores ambientais no modo básico.

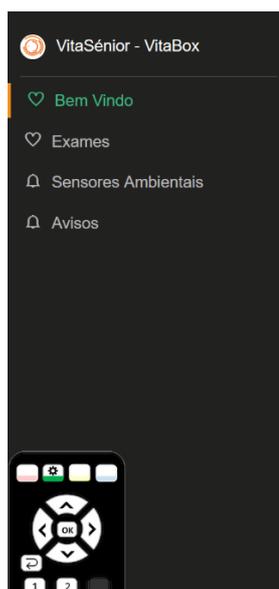


Figura 48 - Modo básico.

6 Teste de usabilidade ao sistema de interação

Após e durante o desenvolvimento de um produto é necessário aferir da sua usabilidade por forma a melhorar a sua aceitação. O presente capítulo discute que parâmetros importa avaliar e que ferramentas existem para realizar essa análise. Apresenta também um conjunto de testes realizados durante esta fase de desenvolvimentos, respetivos resultados e uma análise crítica dos resultados obtidos da interação com a VITABOX. Idealmente os testes envolvem experiências de interação em ambientes reais com o utilizador final. No entanto, nesta fase inicial, foram convidados indivíduos mais novos para despistar problemas mais transversais e dessa forma evitar sobrecarregar indivíduos mais idosos. A fase seguinte dos testes de usabilidade envolverá a população sénior. Na literatura científica existem vários instrumentos para aferir da usabilidade dos sistemas.

Descreve-se em seguida alguns desses mecanismos tendo em conta a sua aplicabilidade ao sistema VITASENIOR-MT e mais especificamente à VITABOX.

Concurrent Think Aloud (CTA) [40] é usado para compreender os pensamentos dos participantes enquanto interagem com o produto, fazendo-os pensar em voz alta. O objetivo é encorajar os participantes a manter uma linha de pensamento para entender os pensamentos dos participantes enquanto interagem com o produto.

Retrospective Think Aloud (RTA) [40] é pedido aos participantes para refazer os passos quando a sessão estiver completa. É frequente os participantes assistirem a um vídeo com as suas ações, que podem ou não conter os padrões da direção olhar.

Concurrent Probing (CP) [40] requer que os participantes trabalhem em tarefas e quando eles dizem algo interessante ou fazem algo único são feitas questões relacionadas.

Retrospective Probing (RP) [40] requer espera até que a sessão seja concluída. Depois são feitas perguntas sobre os pensamentos dos participantes. Quando o participante faz comentários ou ações, são tomadas notas e faz-se o acompanhamento.

Post-Study System Usability Questionnaire (PSSUQ) [42] é um instrumento desenvolvido pela IBM para avaliar a usabilidade. Este teste consiste em dezanove itens a abordar cinco características de usabilidade, como a conclusão rápida do trabalho, facilidade

de aprendizagem, documentação de alta qualidade e informação online, adequação e rápida aquisição de especialistas em usabilidade.

Self-Assessment Manikin (SAM) [43] é uma técnica de avaliação pictórica não verbal, que mede diretamente o prazer, a excitação e a dominância. Esta técnica tem sido usada com frequência em avaliações de anúncios e também de produtos.

System Usability Scale (SUS) [41] é uma ferramenta rápida e consistente para medir a usabilidade. Consiste num questionário de dez perguntas e cinco opções para resposta. As respostas possíveis vão desde “concorda fortemente” a “discordo fortemente” e são posteriormente convertidas em valores de 1 a 5. A apresentação dos resultados não é apresentada no formato de percentagens, pelo que a interpretação baseia-se no valor de score obtido. Apenas scores acima de 68 apontam para sistemas com usabilidade.

6.1 Testes utilizados

Neste trabalho de avaliação da VITABOX, optou-se pela abordagem de testes propostos pelo SUS [41] dado a sua fácil aplicabilidade, processamento dos dados recolhido e reconhecida fiabilidade pela comunidade científica. O teste SUS tem vantagens e é referido em mais de mil e trezentos artigos e publicações. As vantagens do SUS são:

- Fácil de escalar e administrar.
- Pode ser usado em amostras pequenas com bons resultados.
- Pode efetivamente diferenciar se o sistema é utilizável ou não.

Foi também desenvolvido um outro questionário a nível interno, com perguntas focadas em itens da interface da VITABOX.

6.2 Metodologia

Os testes para aferir da usabilidade do sistema consistiram na definição de um conjunto de tarefas de interação com o sistema de visualização gráfico da TV/VITABOX, tarefas de interação com os dispositivos de medição biométricos e tarefas de consulta de informação recolhida:

Tarefa 1 - Aceder a aplicação VITABOX utilizando o telecomando da TV.

Tarefa 2- Aceder ao menu de realização de um exame médico.

Tarefa 3 - Realizar o procedimento indicado para medição do peso.

Tarefa 4 - Consultar o histórico das medições biométricas realizadas.

Tarefa 5 - Consultar o histórico das medições dos parâmetros ambientais.

Tarefa 6- Verificar a existência de avisos.

Tarefa 7 - Aceder ao menu de configurações.

Tarefa 8 - Realizar o procedimento de medição de um dado biométrico utilizando a áudio descrição.

Vários indivíduos foram convidados realizar essas tarefas e no final preencheram dois questionários:

- Um dos questionários é composto por 10 questões e que pretende avaliar a usabilidade global do sistema. Baseia-se no instrumento de avaliação System Usability Scale (SUS) [41] onde as questões foram traduzidas para português. A compilação dessas 10 questões resultaram num único número indicativo da usabilidade do sistema percebida pelos participantes.
- Um segundo questionário contém questões específicas de cada tarefa

Em anexo estão disponíveis as imagens dos testes de usabilidade realizados no projeto. A Figura 54, Figura 55 e Figura 56 apresenta o questionário interno. A Figura 57 e Figura 58 apresentam o teste de usabilidade SUS.

6.2.1 Participantes

10 Indivíduos participaram de forma voluntária nas experiências de interação, três do sexo feminino e sete masculinos, com uma média de 24.2 anos e com um desvio padrão de $\sigma=2,149935$.

6.2.2 Procedimento

A cada participante foi explicado qual o objetivo dos testes, as funcionalidades da VITABOX e que tarefas deveriam realizar. Exemplificou-se como utilizar o telecomando. Os procedimentos a realizar encontram-se descritos na Figura 59 e Figura 60. Após a conclusão das tarefas foi pedido ao utilizador para preencher o questionário SUS e o questionário interno.

No questionário interno existem várias questões, sendo que por cada tarefa existem várias perguntas relacionadas com o item a analisar e onde a resposta é dada vai desde “mau” até “bom”, associado a números de 1 a 5.

6.2.3 Equipamento utilizado

VITABOX, sensores de ambientais, balança, recetor de TV e coando remoto

6.3 Resultados

O gráfico da Figura 49 refere-se a primeira questão Q1: “Como avalia globalmente a aplicação do sistema interface TV-VITABOX?”

P1 – Acesso á aplicação.

P2 – Utilidade da aplicação.

P3 – Tamanho da letra.

P4 – Disposição da informação.

P5 – Compreensão dos menus.

P6 – Cor dos elementos.

P7 – Navegação na aplicação.

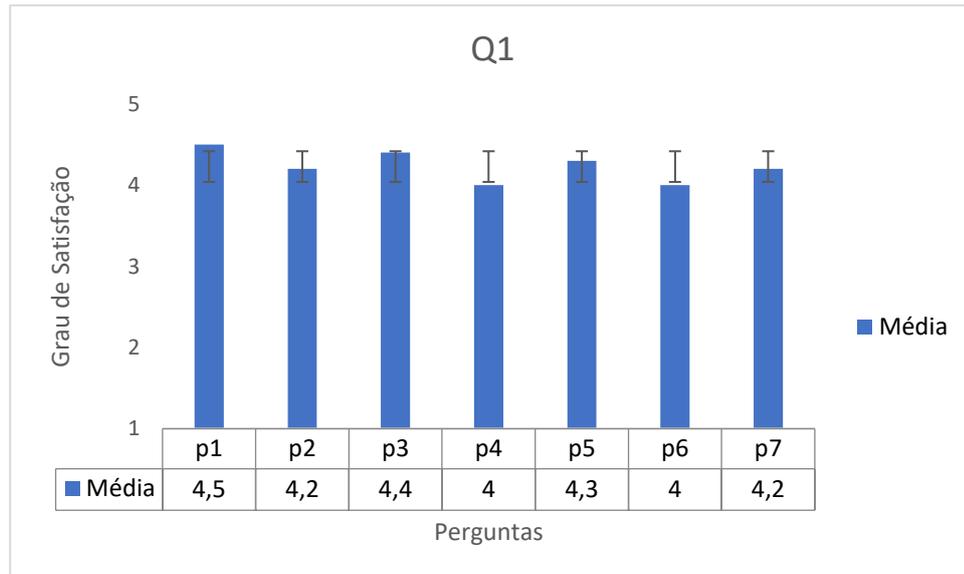


Figura 49 - Como avalia globalmente a aplicação do sistema interface TV-VITABOX?

Nesta questão, todas as perguntas apresentaram médias de quatro valores, percebendo assim que o nível de satisfação da aplicação a nível global é aceitável.

O gráfico representado na Figura 50 refere-se a segunda questão Q2: “Como avalia globalmente os exames de saúde disponíveis?”

P1 – Seleção do utilizador.

P2 – Acesso ao separador de exames.

P3 – Áudio-descrição.

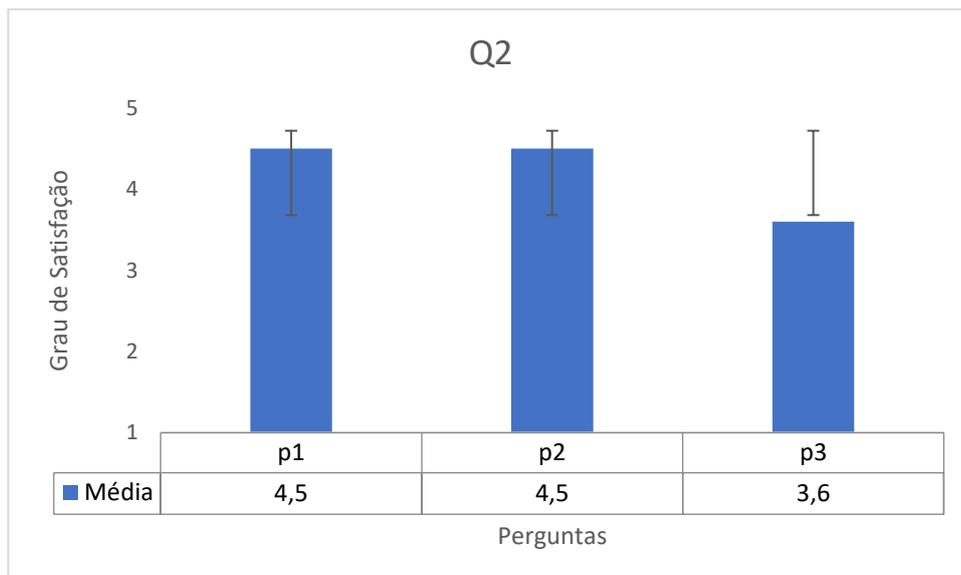


Figura 50 - Como avalia globalmente os exames de saúde disponíveis?

Nesta questão, percebe-se que a P3 da áudio-descrição não foi tao bem recebida estando numa média de três em relação á media de quatro de P1 e P2, mas que no global os exames são aceites pela sua importância.

O gráfico da Figura 51 refere-se a terceira a questão Q3: “Como avalia o exame do peso?”

P1 – Utilidade do exame.

P2 – Realização do exame.

P3 – Duração do exame.

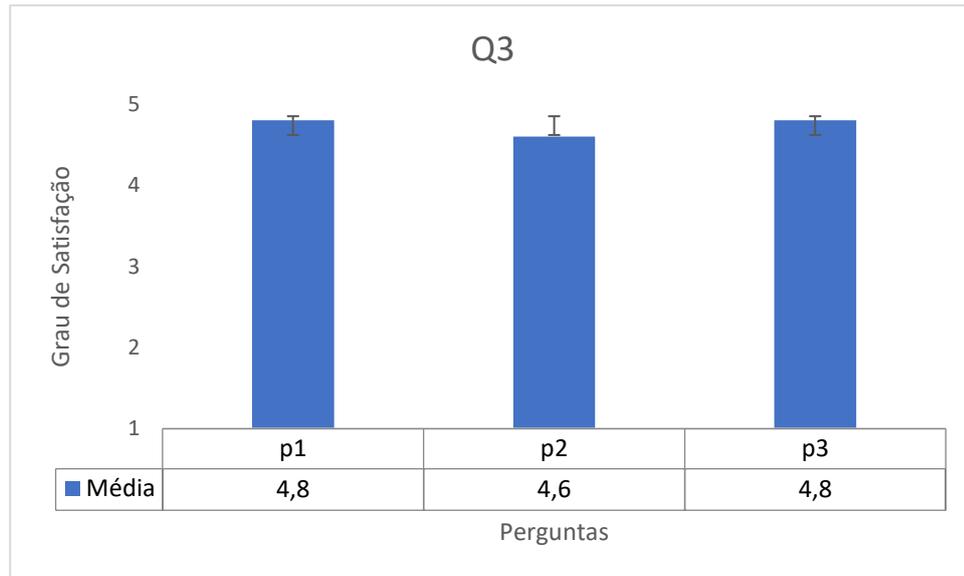


Figura 51 - Como avalia o exame do peso.

Nesta questão, a avaliação do peso foi muito boa, as três perguntas apresentam médias de quatro valores, percebendo que o exame do peso foi bem aceite e com um bom grau de satisfação.

O gráfico da Figura 52 refere-se a quarta a questão Q4: “Como avalia globalmente o acesso ao histórico dos exames?”

P1 – Seleção do utilizador.

P2 – Acesso ao separador de histórico de exames.

P3 – Áudio-descrição.

P4 – Utilidade do histórico de exames

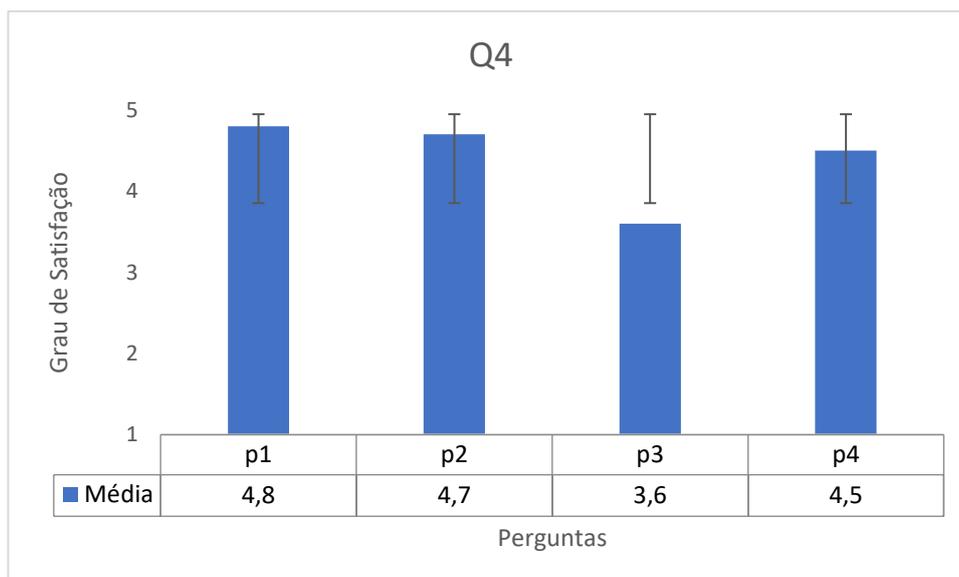


Figura 52 - Como avalia o exame globalmente o acesso ao histórico dos exames?

Nesta questão, mais uma vez nota-se que a áudio-descrição tem uma baixa aceitação, apresentando uma média dos três valores no P3 em relação ao P1, P2 e P4, que rondam os quatro de média, concluindo que no global do histórico do exame é importante, e é necessário melhorar a áudio-descrição.

O gráfico apresentado na Figura 53 refere-se à quinta questão Q5: “Consulta de histórico de sensores ambientais”.

P1 – Acesso ao separador de sensores ambientais.

P2 – Áudio-descrição.

P3 – Utilidade dos sensores ambientais.

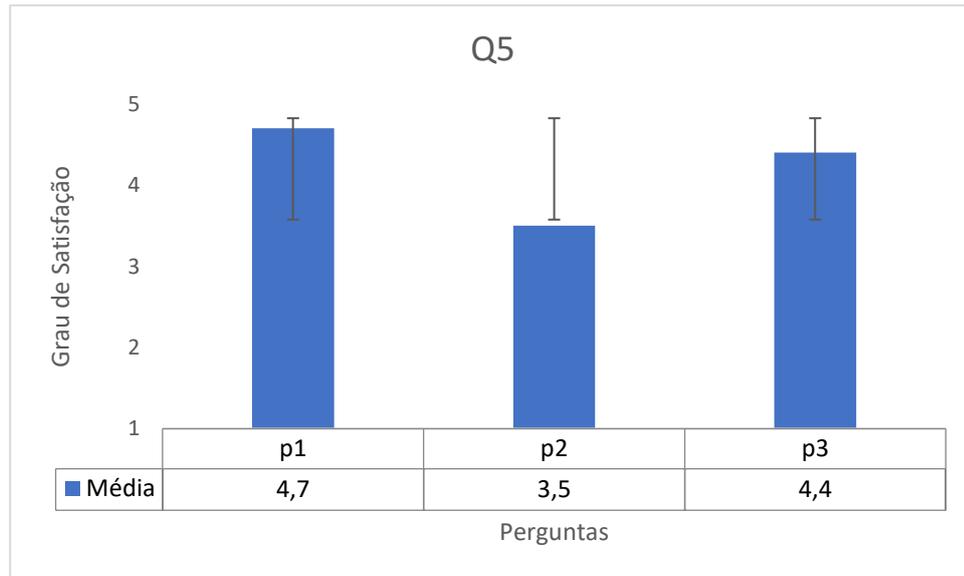


Figura 53 - Consulta de histórico de sensores ambientais.

Nesta questão, P1 e P3 apresenta uma média de quatro valores, notando que o P2 tem uma média de três valores média, mais uma vez é uma opção bem aceite, com apenas o problema que a áudio-descrição a precisa de melhorias.

6.3.1 System Usability Scale (SUS)

No teste SUS executado, foi pedido aos utilizadores para preencher dez perguntas acerca da aplicação. Estas têm cinco opções de escolha, desde “Discordo fortemente” a “concordo fortemente”, podendo escolher de 1 a 5.

Para aferição do teste, foi realizada a soma da pontuação de todas as questões de um utilizador e multiplicou-se por 2,5. Foi realizado a todos os utilizadores e fez-se a média de todos os resultados. O resultado obtido dos dez utilizadores foi uma média de 79 pontos, com a pontuação de 0 a 100. A interpretação desta escala é feita por percentil, com a pontuação de 79 é considerado um bom resultado.

6.4 Discussão de resultados

Os resultados dos dez utilizadores, indicam que de uma forma geral a aplicação foi bem aceite. Uma das funcionalidades pior classificadas no questionário interno, foi a áudio-descrição (ver gráficos Q5, Q4 e Q3). Apesar dos resultados não serem maus, conclui-se que é necessário proceder a alterações para melhorar a aceitação da VITABOX.

A avaliação global da aplicação realizada pelos dez utilizadores, através do teste SUS, indica um grau de satisfação é 79 pontos. Conclui-se que é uma boa pontuação, no entanto, não se descarta a necessidade de algumas melhorias para tentar subir o nível de satisfação, como a melhoria da qualidade dos ficheiros de áudio.

7 Conclusão

Com a contínua baixa taxa de natalidade assim como o aumento da longevidade, tem levado ao aumento significativo na população idosa. Este aumento causa uma sobrecarga aos serviços médicos, devido às crescentes necessidades dessa faixa da população. Este facto tem levado ao aparecimento de soluções de teleassistência que permitam aliviar a pressão sobre os sistemas de saúde assim como facilitar o auxílio de proximidade por parte dos cuidadores.

Existem vários problemas a ter em consideração ao desenhar uma solução de teleassistência, nomeadamente a resistência na integração de novas tecnologias por parte dos idosos, assim como as dificuldades audiovisuais e motoras de cada utilizador. O trabalho apresentado é parte de um projeto que é uma solução que inclui várias funcionalidades integradas numa única plataforma.

O trabalho realizado contribuiu para o desenvolvimento de uma aplicação visual baseada na utilização do comando da televisão com um sistema de áudio descrição em modo *offline*. Esta apresenta uma interface o mais simples possível tendo em conta o seu público-alvo, assim como as restrições do uso do comando. Esta aplicação necessita ainda de validação em tempo real, no entanto, os testes de usabilidade realizados em laboratório, permitem afirmar que a aplicação cumpre os objetivos iniciais. Os testes reais vão permitir validar e introduzir melhorias ao sistema. No entanto, dos testes efetuados, verificou-se que o sistema de interação está dependente do protocolo CEC implementado em cada recetor de TV, podendo limitar as capacidades de alguns dos comandos disponíveis e dessa forma condicionar o funcionamento da aplicação visual.

Foi implementado o sistema de áudio-descrição em modo *offline*. Esta opção permite o seu uso sem a necessidade de aplicações *online*, tornado o sistema mais flexível. No entanto, o seu uso está dependente da naturalidade das expressões apresentadas, pelo que poderá ser necessário um refinamento ao sistema para que este seja completamente aceite pelos utilizadores. Os testes reais de utilização irão permitir tirar conclusões com vista a introduzir melhorias a esta funcionalidade.

O sistema de alertas e notificações foi implementado com sucesso, considerando que este necessita de informação existente na *cloud* e da rede de sensores. A troca de informação entre estes módulos e a VITABOX está implementada e em fase de refinamento, pelo que dá garantias de um bom funcionamento em testes reais.

Em conclusão, o trabalho desenvolvido contribuiu para o bom funcionamento de todo o sistema de teleassistência VITASENIOR-MT, podendo afirmar que as contribuições apresentadas neste relatório permitiram para que este possa ser útil pelos seus utilizadores.

7.1 Trabalho futuro

Sendo um protótipo em fase de testes reais, o trabalho futuro passa por validar o sistema e realizar as alterações à interface visual resultante do retorno dos utilizadores com vista a ter uma aplicação o mais simples possível com o menor número de passos para obter uma funcionalidade.

Adicionalmente, o sistema de *text-to-speech* deve ser melhorado por forma a conseguir que a informação sonora seja o mais natural possível. Futuramente seria interessante implementar um sistema local de autenticação de utilizadores, por forma a poder garantir que o utilizador seleccionado para medição de dados biométricos é quem afirma ser.

8 Referências

10 Mil Vidas - PDF. (s.d.). (ANCS) Obtido de <https://www.ancs.pt/assets/10-mil-vidas.pdf>

Angelo Costa, P. N. (2014). A caregiver support platform within the scope of an ambient assisted living ecosystem. *Sensors (Basel, Switzerland)*, *14*(3), 5654-76.

Bemenderfer, J. (s.d.). *Creating a Global Event Bus with Vue.js*. Obtido de <https://alligator.io/vuejs/global-event-bus/>

Bujnowska-Fedak, M., & Grata-Borkowska, U. (2015). Use of telemedicine-based care for the aging and elderly: promises and pitfalls. *Smart Homecare Technology and TeleHealth*, *3*, 91–105.

CEC-O-MATIC. (s.d.). Obtido de <http://www.cec-o-matic.com/>

CoAP — Constrained Application Protocol | Overview. (s.d.). Obtido de <http://coap.technology/>

Comission, E. (2014). *Population ageing in Europe: facts, implications and policies Outcomes of EU-funded research*.

Consumer Electronics Control. (s.d.). Obtido de [https://elinux.org/CEC_\(Consumer_Electronics_Control\)_over_HDMI](https://elinux.org/CEC_(Consumer_Electronics_Control)_over_HDMI)

Docs | Node.js. (s.d.). Obtido de <https://nodejs.org/en/docs/>

eSpeak. (s.d.). Obtido de <http://espeak.sourceforge.net/>

euvi. (s.d.). *vue-js-modal*. Obtido de <https://github.com/euvi/vue-js-modal#readme>

Festival Text-to-Speech Online Demo - Technical. (s.d.). Obtido de <http://www.cstr.ed.ac.uk/projects/festival/morevoices.html>

FFMPEG Documentation. (s.d.). Obtido de <https://www.ffmpeg.org/documentation.html>

G. Pires, P. C. (2018). "VITASENIOR-MT: a telehealth solution for the elderly focused on the interaction with TV,". *IEEE 20th International Conference on e-Health*

Networking, Applications and Services (Healthcom), 17, p. 2496. Ostrava, Czech Republic.

HDMI. (s.d.). Obtido de https://www.hdmi.org/manufacturer/hdmi_2_0/index.aspx

Home · contiki-ng/contiki-ng Wiki. (s.d.). Obtido de <https://github.com/contiki-ng/contiki-ng/wiki>

HTTPS | Node.js v11.2.0 Documentation. (s.d.). Obtido de <https://nodejs.org/api/https.html>

IEEE 802.15.4. (s.d.). Obtido de <http://www.ieee802.org/15/pub/TG4.html>

JSON. (s.d.). Obtido de <https://www.json.org/>

kazupon. (s.d.). *Introduction | Vue I18n*. Obtido de <https://kazupon.github.io/vue-i18n/introduction.html>

Lang, M. M. (1994). Measuring emotion: the Self-Assessment Manikin and the Semantic Differential. *Journal of behavior therapy and experimental psychiatry*, 25(1), 49-59.

Luís M. L. Oliveira, J. J. (2014). Wireless Sensor Networks in IPv4/IPv6 Transition Scenarios. *Wireless Personal Communications*, 78(4), 1849--1862.

Majumder, S., Aghayi, E., Noferesti, M., Memarzadeh-Tehran, H., Mondal, H., Pang, Z., & Deen, M. (2017). Smart Homes for Elderly Healthcare—Recent Advances and Research Challenges. *Sensors, mdpi*, 17, 1-32.

Marcelino I, L. R.-M.-R. (2018). Active and Assisted Living Ecosystem for the Elderly. *Sensors (Basel)*, 18(4), 1246.

MetinSeylan. (s.d.). *Vue-Socket.io*. Obtido de <https://github.com/MetinSeylan/Vue-Socket.io>

Mobile Health to Support Ageing in Place: A Synoptic Overview. (s.d.).

MySignals. (s.d.). Obtido de http://www.libelium.com/downloads/documentation/mysignals_technical_guide.pdf

Normie, L. (2011). Technology for Ageing in Place. *IFA global ageing*, 7, 45-53.

OECD. (2010). Health at a Glance: Europe 2010. OECD Publishing -
http://dx.doi.org/10.1787/health_glance-2010-en.

Open Source Document Database / MongoDB. (s.d.). Obtido de
https://docs.mongodb.com/?_ga=2.12578676.1764170336.1542897993-1084996626.1538220748

PINE64. (s.d.). Obtido de https://www.pine64.org/?page_id=7147

Post-Study System Usability Questionnaire (PSSUQ). (s.d.).
(<https://www.conetrees.com/ux-glossary/post-study-system-usability-questionnaire-pssuq/>).

Projeto 10 Mil Vidas - ANCS. (s.d.). Obtido de <https://www.ancs.pt/projeto-10-mil-vidas---ancs.html>

Raspberry Pi 2 Model B. (s.d.). Obtido de <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

Raspberry Pi 3 Model B. (s.d.). Obtido de <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Raspbian. (s.d.). (Raspberry Pi Foundation) Obtido de <https://www.raspbian.org/>

ResponsiveVoice.JS. (s.d.). Obtido de <https://responsivevoice.org/>

Running a Usability Test. (s.d.). Obtido de <https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html>

SmartAL. (s.d.). Obtido de
http://www.alticelabs.com/content/products/BR_SMARTAL_ALB_EN.pdf

Socket.IO. (s.d.). Obtido de <https://socket.io/get-started/chat>

System Usability Scale (SUS). (s.d.). Obtido de <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

Tablado, A. &. (2004). Aingeru: An innovative system for tele-assistance of elderly people. *Journal on Information Technology in Healthcare*. Porto, Portugal.

Use of telemedicine-based care for the aging and elderly: promises and pitfalls. (s.d.).

Vue Paper Dashboard. (s.d.). Obtido de <https://www.creative-tim.com/product/vue-paper-dashboard>

wg5945. (s.d.). *vue-marquee-tips*. Obtido de <https://github.com/wg5945/vue-marquee-tips#readme>

Anexos

Questionário de Satisfação

Interface TV-VITABOX

Os dados são guardados de forma anónima

Data: _____

Utilização do sistema: _____ vez

Dados Gerais

Idade		
Sexo		
Utiliza óculos		
Dificuldade em ouvir		
Usa aparelho auditivo		
Sabe ler		

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.

Como avalia globalmente a aplicação do sistema Interface TV-VITABOX?

	Mau	1	2	3	4	5	Bom	N/A
Acesso à aplicação		○	○	○	○	○		
Utilidade da aplicação		○	○	○	○	○		
Tamanho da letra		○	○	○	○	○		
Disposição da informação		○	○	○	○	○		
Compreensão dos menus		○	○	○	○	○		
Cor dos elementos		○	○	○	○	○		
Navegação na aplicação		○	○	○	○	○		

Realização de exames de saúde

Figura 54 - Questionário de Satisfação primeira parte.

Como avalia globalmente os exames de saúde disponíveis?

	Mau	1	2	3	4	5	Bom	N/A
Seleção do utilizador		<input type="radio"/>						
Acesso ao separador de exames		<input type="radio"/>						
Áudio-descrição		<input type="radio"/>						

Como avalia o exame de peso

	Mau	1	2	3	4	5	Bom	N/A
Utilidade do exame		<input type="radio"/>						
Realização do exame		<input type="radio"/>						
Duração do exame		<input type="radio"/>						

Como avalia a bracelete (frequência cardíaca)

	Mau	1	2	3	4	5	Bom	N/A
Utilidade do exame		<input type="radio"/>						
Realização do exame		<input type="radio"/>						
Duração do exame		<input type="radio"/>						

Como avalia o exame de pressão arterial

	Mau	1	2	3	4	5	Bom	N/A
Utilidade do exame		<input type="radio"/>						
Realização do exame		<input type="radio"/>						
Duração do exame		<input type="radio"/>						

Consulta de histórico de exames

Como avalia globalmente o acesso ao histórico dos exames?

	Mau	1	2	3	4	5	Bom	N/A
Seleção do utilizador		<input type="radio"/>						
Acesso ao separador de exames		<input type="radio"/>						
Áudio-descrição		<input type="radio"/>						
Utilidade do histórico de exames		<input type="radio"/>						

Figura 55 - Questionário de Satisfação segunda parte.

Consulta de histórico de sensores ambientais

	Mau	1	2	3	4	5	Bom	N/A
Acesso ao separador de exames		<input type="radio"/>						
Áudio-descrição		<input type="radio"/>						
Utilidade do histórico de exames		<input type="radio"/>						

Avaliação qualitativa

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

Figura 56 - Questionário de Satisfação terceira parte.

System Usability Scale (Traduzido para português)

Questionário

Escala de Usabilidade do Sistema

Tecnologia VITABOX

1. Acho que gostaria de usar a tecnologia VITABOX com frequência.

Discordo Fortemente 1	2	3	4	Concordo Fortemente 5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. Acho a tecnologia VITABOX desnecessariamente complexa.

Discordo Fortemente 1	2	3	4	Concordo Fortemente 5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. Acho que tecnologia VITABOX é fácil de usar.

Discordo Fortemente 1	2	3	4	Concordo Fortemente 5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. Acho que necessito do suporte de um técnico especializado para conseguir utilizar a tecnologia VITABOX.

Discordo Fortemente 1	2	3	4	Concordo Fortemente 5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. Penso que as várias funcionalidades da tecnologia VITABOX estão bem integradas.

Discordo Fortemente 1	2	3	4	Concordo Fortemente 5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 57 - Escala de usabilidade do sistema primeira parte.

6. Penso que a tecnologia VITABOX apresenta muitas inconsistências.

Discordo Fortemente				Concordo Fortemente
1	2	3	4	5
<input type="radio"/>				

7. Imagino que a maioria das pessoas irá aprender a utilizar a tecnologia VITABOX com muita facilidade.

Discordo Fortemente				Concordo Fortemente
1	2	3	4	5
<input type="radio"/>				

8. Acho que a tecnologia VITABOX é muito difícil de usar.

Discordo Fortemente				Concordo Fortemente
1	2	3	4	5
<input type="radio"/>				

9. Senti-me muito confiante a utilizar a tecnologia VITABOX.

Discordo Fortemente				Concordo Fortemente
1	2	3	4	5
<input type="radio"/>				

10. Tive necessidade de aprender muitas coisas antes de conseguir utilizar a tecnologia VITABOX.

Discordo Fortemente				Concordo Fortemente
1	2	3	4	5
<input type="radio"/>				

Figura 58 - Escala de usabilidade do sistema segunda parte.

Procedimento de Testes VitaSenior

Teste 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 projeto. 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 1;

Tarefa 2 - Realização de Exames de diagnóstico da saúde do utilizador

1. Navegar no menu com as setas para cima ou para baixo até alcançar o menu dos exames;
2. Pressionar a tecla da direita até seleccionar o utilizador pretendido e de seguida pressionar OK;
3. Carregar na seta direita até alcançar o exame pretendido e de seguida carregar em OK;
4. Após seleccionar o exame pretendido seguir as instruções indicadas;
5. Repetir o procedimento de 3 a 4 para realizar qualquer um dos exames possíveis.
6. Assim que os exames estejam todas concluídos carregar na tecla Back duas vezes para sair para o menu principal;

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

1. Navegar até à opção histórico de exames usando a seta para baixo;
2. Pressionar a tecla da direita até seleccionar o utilizador pretendido e de seguida pressionar OK;

Figura 59 - Procedimento dos testes primeira parte.

3. Selecionar o histórico de exames pretendido utilizando as setas esquerda ou direita;
4. Carregar no botão OK assim que o histórico de exames pretendido esteja selecionado;
5. Repetir o procedimento de 3 a 4 para consultar qualquer um dos históricos de exames disponíveis;
6. Carregar na tecla Exit duas vezes para voltar ao menu principal;

Tarefa 4 - Utilização do histórico de sensores ambientais

1. Navegar até à opção histórico de sensores ambientais usando as setas para baixo ou para cima;
2. Selecionar o histórico de sensores ambientais pretendido utilizando as setas esquerda ou direita;
3. Carregar no botão OK assim que o histórico de sensores ambientais pretendido esteja selecionado;
4. Selecionar a divisão pretendida com as teclas direita ou esquerda
5. Pressionar ok para visualizar ou esconder histórico do sensor selecionado
6. Repetir o procedimento de 2 a 4 após pressionar a tecla Exit (uma vez) para poder consultar o histórico de outro sensor;
7. Carregar na tecla Exit duas vezes para voltar ao menu principal;

Tarefa 5 - Consultar avisos

1. Navegar até à opção avisos usando as setas para cima ou para baixo;
2. 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

1. Para alterar as configurações disponíveis pressione o botão verde para abrir o menu configurações;
2. Utilize qualquer seta para navegar as configurações disponíveis e pressione a tecla OK assim que a opção pretendida esteja selecionada;
3. Ao selecionar a opção modo pode optar entre a modo básico e modo avançado utilizando as teclas OK para alterar;
4. Ao selecionar a opção som pode ligar ou desligar o som da aplicação, utilizando a tecla OK para alterar;
5. Ao selecionar a opção linguagem pode selecionar entre português e inglês, utilizando a tecla OK para alterar;
6. Pressionar Exit para sair;
7. Pressionar o botão 1 para voltar para a emissão de TV.

Tarefa 7 – Realização de um exame de saúde com áudio-descrição

Tendo em conta os procedimentos descritos anteriormente, realize uma medição da pressão arterial usando áudio-descrição.

Figura 60 - Procedimentos dos testes segunda parte.