

# Universidade da Beira Interior

## Departamento de Informática



Departamento de  
Informática

### ***Nº 20 - 2019: An Internet of Things Application for Smart Homes to Monitor and React to Emergency Scenarios***

Elaborado por:

**Tomás Francisco Nogueira das Neves Marques Jerónimo**

Orientador:

**Professor Doutor Bruno M. C. Silva**

15 de Julho de 2019



# Agradecimentos

Este projeto representa o culminar da primeira fase do meu percurso académico, durante o qual tive o prazer de trabalhar com diversas pessoas às quais gostaria de agradecer aqui.

Em primeiro lugar, quero agradecer aqueles que me ajudaram a ultrapassar os diversos desafios encontrados durante o desenvolvimento deste projeto, e sem os quais a sua conclusão não seria possível. Assim, deixo aqui um obrigado ao professor Bruno Silva pela sua orientação, pela ajuda proporcionada na aprendizagem de todos os novos conceitos e tecnologias usados na realização deste trabalho e pela dedicação em garantir a realização dos objetivos definidos. Um muito obrigado também aos meus colegas do *Next Generation Networks and Applications Group* (NetGNA), aos quais destaco o companheirismo e interajuda que sempre que sempre me mostraram ao longo deste semestre.

Tal como já referi, este trabalho representou o fim de uma fase de aprendizagem, durante a qual tive a oportunidade de conhecer diversos indivíduos, que hoje tenho o prazer de chamar amigos, a quem não podia deixar de agradecer a amizade e todos os momentos partilhados nestes três anos. Sinto que posso olhar com orgulho para aquilo que alcancei e muito disso devo-o a vocês.

Por fim, mas não menos importante, quero deixar um agradecimento muito especial à minha família e à minha namorada pelo apoio e incentivo que sempre me proporcionaram, motivando-me sempre a ir mais além na busca dos meus objetivos.

A todos vocês, um muito obrigado,  
*Tomás Jerónimo*



# Conteúdo

<b>Conteúdo</b>	<b>iii</b>
<b>Lista de Figuras</b>	<b>vii</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento . . . . .	1
1.2 Motivação . . . . .	2
1.3 Objetivos . . . . .	3
1.4 Organização do Documento . . . . .	4
<b>2 Estado da Arte</b>	<b>5</b>
2.1 Introdução . . . . .	5
2.2 <i>Internet of Things</i> (IoT) na Área da Saúde . . . . .	5
2.3 Mobile Health . . . . .	7
2.4 Conclusões . . . . .	9
<b>3 Tecnologias e Ferramentas Utilizadas</b>	<b>11</b>
3.1 Introdução . . . . .	11
3.2 Aplicação Móvel . . . . .	11
3.2.1 Android Studio . . . . .	11
3.2.2 Android . . . . .	11
3.2.3 Java . . . . .	12
3.3 Aplicação <i>Web</i> . . . . .	12
3.3.1 Javascript . . . . .	12
3.3.2 NodeJS . . . . .	12
3.3.3 HTML5 . . . . .	12
3.3.4 CSS3 . . . . .	12
3.4 Relatório . . . . .	13
3.4.1 LaTeX . . . . .	13

3.5	Conclusões . . . . .	13
<b>4</b>	<b>Engenharia de Software</b>	<b>15</b>
4.1	Introdução . . . . .	15
4.2	Arquitetura de Sistema do Home Intelligent Assitant Service (HIAS)	15
4.2.1	HIAS Cloud Architecture . . . . .	17
4.2.2	<i>Medical Data Blockchain Access Record System</i> (MD.BARS)	17
4.3	Análise de Requisitos . . . . .	19
4.3.1	Requisitos para Aplicação <i>Web</i> . . . . .	19
4.3.2	Requisitos para Aplicação Móvel . . . . .	21
4.4	Casos de Uso . . . . .	23
4.4.1	Casos de Uso para Aplicação <i>Web</i> . . . . .	23
4.4.2	Casos de Uso para Aplicação Móvel . . . . .	25
4.5	Diagramas de Atividades . . . . .	28
4.5.1	Diagramas de Atividades para Aplicação <i>Web</i> . . . . .	28
4.5.2	Diagrama de Atividades para Aplicação Móvel . . . . .	32
4.6	Conclusões . . . . .	34
<b>5</b>	<b>Construção da Aplicação Móvel</b>	<b>35</b>
5.1	Introdução . . . . .	35
5.2	Detalhes da Construção . . . . .	35
5.2.1	Autenticação de utilizadores . . . . .	35
5.2.2	Armazenamento de Dados . . . . .	36
5.2.3	Troca de Dados entre a Aplicação e o Servidor HIAS . . . . .	37
5.2.4	Chamada de Emergência . . . . .	40
5.3	Manual de Utilizador . . . . .	41
5.4	Conclusões . . . . .	43
<b>6</b>	<b>Construção e Testes da Aplicação Web</b>	<b>45</b>
6.1	Introdução . . . . .	45
6.2	Detalhes da Construção . . . . .	45
6.2.1	Troca de Dados entre Aplicação e Servidor HIAS . . . . .	45
6.2.2	Início da Chamada . . . . .	46
6.2.3	Processamento de Voz . . . . .	47
6.2.4	Fim da chamada . . . . .	48
6.2.5	Troca de Dados entre Aplicação e Servidor MD.BARS . . . . .	48
6.3	Manual de Utilizador . . . . .	49
6.4	Testes e Validação do Reconhecimento de Voz da Aplicação <i>Web</i> . . . . .	51
6.5	Conclusões . . . . .	53

## CONTEÚDO

---

v

<b>7</b>	<b>Conclusões e Trabalho Futuro</b>	<b>55</b>
7.1	Conclusões Principais . . . . .	55
7.2	Trabalho Futuro . . . . .	56
	<b>Bibliografia</b>	<b>59</b>





# Lista de Figuras

2.1	Serviço CareAngel . . . . .	6
2.2	Serviço Apple Watch Healthcare . . . . .	6
2.3	Aplicação - Lose It! . . . . .	8
2.4	Aplicação - 112 Where ARE U . . . . .	9
4.1	Desenho Conceptual da Arquitetura de Sistema do HIAS . . . . .	16
4.2	Diagrama de Casos de Uso da Aplicação <i>Web</i> . . . . .	23
4.3	Diagrama de Casos de Uso da Aplicação Móvel . . . . .	25
4.4	Diagrama de Atividades da Aplicação <i>Web</i> - Início da Chamada . . . . .	28
4.5	Diagrama de Atividades da Aplicação <i>Web</i> - Processamento de Voz . . . . .	29
4.6	Diagrama de Atividades da Aplicação <i>Web</i> - Atualizar Sensores . . . . .	30
4.7	Diagrama de Atividades da Aplicação <i>Web</i> - Fim da Chamada . . . . .	31
4.8	Diagrama de Atividades da Aplicação Móvel . . . . .	33
5.1	Processo de Encriptação da Palavra-Passe . . . . .	36
5.2	Início da Chamada de Emergência . . . . .	40
5.3	Atividade <i>Login</i> da Aplicação Móvel . . . . .	41
5.4	Atividade Principal da Aplicação Móvel . . . . .	42
5.5	Atividade Editar Dados da Aplicação Móvel . . . . .	43
6.1	Processamento de Voz . . . . .	47
6.2	Comunicação entre Aplicação <i>Web</i> e Servidor MD.BARS . . . . .	49
6.3	Aplicação <i>Web</i> após Início da Chamada . . . . .	49
6.4	Apresentação de Dados do Utilizador a Pedido do Operador . . . . .	50
6.5	Formulário Gerado no Final da Chamada na Aplicação <i>Web</i> . . . . .	50
6.6	Resultados de uma das Leituras Realizadas ao Primeiro Guião . . . . .	52
6.7	Resultados de uma das Leituras Realizadas ao Segundo Guião . . . . .	53



## Lista de Tabelas

4.1	Caso 1 - Terminar Chamada . . . . .	24
4.2	Caso 2 - Preencher Relatório . . . . .	24
4.3	Caso 1 - Iniciar Sessão . . . . .	26
4.4	Caso 2 - Editar Dados . . . . .	26
4.5	Caso 3 - Ver Dados dos Sensores . . . . .	27
4.6	Caso 4 - Iniciar Chamada de Emergência . . . . .	27
4.7	Caso 5 - Terminar sessão . . . . .	27



## Lista de Excertos de Código

5.1	Armazenameto de Dados após o <i>Login</i> . . . . .	36
5.2	Acesso à Informação Armazenada em Memória. . . . .	37
5.3	Implementação do Método GET na Aplicação Móvel. . . . .	38
5.4	Implementação do Método POST na Aplicação Móvel. . . . .	39
6.1	Implementação do Método GET na Aplicação <i>Web</i> . . . . .	46
6.2	Implementação do Método POST na Aplicação <i>Web</i> . . . . .	46



# Acrónimos

<b>API</b>	<i>Application Programming Interface</i>
<b>CSS</b>	<i>Cascading Style Sheets</i>
<b>CODU</b>	Centro de Orientação de Doentes Urgentes
<b>eHealth</b>	<i>electronic Health</i>
<b>GPS</b>	Sistema de Posicionamento Geográfico
<b>HIAS</b>	Home Intelligent Assitant Service
<b>HTML</b>	<i>HyperText Markup Language</i>
<b>HTTP</b>	<i>HyperText Transfer Protocol</i>
<b>IA</b>	Inteligência Artificial
<b>IDE</b>	Ambiente de Desenvolvimento Integrado
<b>INEM</b>	Instituto Nacional de Emergência Médica
<b>iOS</b>	<i>iPhone Operating System</i>
<b>IoT</b>	<i>Internet of Things</i>
<b>JSON</b>	<i>JavaScript Object Notation</i>
<b>MD.BARS</b>	<i>Medical Data Blockchain Access Record System</i>
<b>mHealth</b>	<i>Mobile Health</i>
<b>NetGNA</b>	<i>Next Generation Networks and Applications Group</i>
<b>SHA-256</b>	<i>Secure Hashing Algorithm, 256-Bits</i>
<b>SIEM</b>	Sistema Integrado de Emergência Médica

<b>URL</b>	<i>Uniform Resource Locator</i>
<b>W3C</b>	<i>World Wide Web Consortium</i>



# Capítulo 1

## Introdução

### 1.1 Enquadramento

O Instituto Nacional de Emergência Médica (INEM) é o organismo responsável por coordenar o funcionamento, no território nacional, do Sistema Integrado de Emergência Médica (SIEM), de forma a garantir aos sinistrados ou vítimas de doença súbita a pronta e correta prestação de cuidados de saúde.[10]

São transferidos para os Centro de Orientação de Doentes Urgentes (CODU) do INEM os pedidos de socorro efetuados através do Número Europeu de Emergência - 112, referentes a situações de urgência ou emergência na área da saúde. [11] O funcionamento do CODU é assegurado 24 horas por dia sendo que, em 2018, foi registada uma média de 3.8 18 chamadas diárias [13] e uma média de 13 segundos para atendimento das chamadas recebidas [12], sendo depois adicionado o tempo necessário para o envio das viaturas de socorro. Este tempo é dependente de vários fatores como a identificação da vítima, a sua localização e o tipo de ocorrência. Só depois de ter toda esta informação disponível é que o operador poderá fazer uma avaliação da situação e ativar os meios mais adequados. O tempo de resposta é, por este motivo, bastante difícil de medir e pode ser prejudicado devido a diferentes problemas, de destacar, a dificuldade em identificar a localização da vítima e dificuldade em transmitir informações ao operador sobre o estado atual desta [33].

Uma forma de solucionar estes problemas passará por usufruir das vantagens proporcionadas por soluções móveis, sensores, *wearables* e sistemas baseados em *Internet of Things* (IoT) [24]. Este é um termo utilizado para descrever um sistema de computadores, máquinas, sensores, outros objetos e pessoas, interligados, cada um com um identificador único e a capacidade de transmitir informação para uma rede sem a necessidade de interação humana. O número de dispositivos com capacidade de fazer parte de um destes sistemas é cada vez maior, e como resul-

tado deste aumento têm sido criados cada vez mais projetos focados em serviços ou aplicações com o objetivo de monitorizar, com recurso a sensores, dados de saúde que podem depois ser armazenados e transmitidos para outros dispositivos. Para além disto, estes dispositivos vêm muitas vezes equipados com um poderoso Sistema de Posicionamento Geográfico (GPS) cujos dados obtidos podem ser enviados, em caso de emergência, para dar ao operador uma perceção muito mais realista da localização da vítima [33, 8, 17, 3].

Podemos ainda complementar estes dispositivos se recorrermos à Inteligência Artificial (IA) para tratar todos os dados recolhidos de forma a otimizar as informações partilhadas com os profissionais de emergência, fornecendo apenas os dados relevantes para cada situação. A IA é uma das melhores formas de gerir vastas quantidades de informação que, com o crescimento de popularidade das redes de Internet de alta velocidade e o avanço na tecnologia de sensores, são cada vez mais comuns em serviços de IoT [2, 26].

Em síntese, neste projeto, pretende-se criar um sistema baseado em IoT para emergências médicas. Este sistema almeja aumentar a eficácia do serviço ao diminuir o tempo desde o atendimento, por parte dos operadores, até à ativação dos meios de saúde. Para tal, este sistema será capaz de monitorizar dados de saúde do sinistrado, em tempo real, e detetar a sua localização geográfica assim que a chamada é iniciada, enviado estes e todos os outros dados previamente armazenados ao operador, de forma a que este possa tomar decisões mais rápidas e informadas sobre as medidas a tomar e os meios a disponibilizar para o salvamento.

## 1.2 Motivação

A principal motivação para a escolha deste trabalho foi a premissa que este abordaria problemas verificados num serviço já vastamente utilizado e, cujas soluções desenvolvidas teriam um impacto no quotidiano dos seus utilizadores e ajudariam a melhorar o desempenho do mesmo. Para além disto, as tecnologias e áreas abordadas encontram-se entre as com melhores perspetivas de evolução nos próximos anos, e das quais se espera um maior impacto no futuro da nossa sociedade [28] [34] [21].

Destas tecnologias e áreas, importa especialmente destacar a *electronic Health* (eHealth). Esta é uma área que se foca no uso de dispositivos móveis para monitorizar remotamente os dados vitais nos seus utilizadores. Estes dispositivos fazem uso de sensores para recolher diferentes tipos de dados, que podem depois ser armazenados, transmitidos e avaliados, com o objetivo de melhorar o processo de decisão dos profissionais de saúde durante o tratamento dos pacientes [20].

Outra característica que levou à escolha deste projeto foi o facto deste se encontrar enquadrado com outros trabalhos em desenvolvimento. É pretendido

que este sistema integre um serviço denominado Home Intelligent Assitant Service (HIAS) cujo objetivo é a coleção e uso compartilhado de dados entre diferentes aplicações relacionadas com saúde e bem estar. Esta é uma característica que diferenciou este projeto de todos os outros e uma das principais razões para a sua preferência.

Em síntese, todas as características apresentadas fazem com que este trabalho seja um desafio, cuja conclusão seja representativa da evolução do autor ao longo do seu percurso acadêmico.

## 1.3 Objetivos

O principal objetivo deste projeto é a construção de uma solução IoT integrada num ecossistema denominado HIAS, que facilite a comunicação entre os operadores dos serviços de emergência e os utilizadores do serviço.

Para tal, deverão ser desenvolvidas duas aplicações: uma aplicação móvel, para o uso da população em geral, e uma aplicação *web* que será usada pelos profissionais de emergência médica.

Os principais objetivos da aplicação móvel são a recolha de diferentes tipos de dados sobre o utilizador, tais como, dados pessoais, dados de saúde e dados recolhidos em tempo real. Estes dados deverão ser, posteriormente, enviados para um servidor, onde serão armazenados até que uma emergência ocorra e seja necessária a sua consulta. Para além disto, será através desta aplicação que o utilizador realizará a conexão com a aplicação *web* dos serviços de emergência.

A aplicação *web* deverá estar constantemente em contacto com o servidor e, caso seja iniciada uma chamada de emergência, esta deve requisitar todos os dados, relevantes, sobre o utilizador que iniciou chamada. Ao longo da conversa, o operador irá tentar obter mais informações sobre a ocorrência, devendo a aplicação ser capaz de detetar, através de reconhecimento de voz, palavras sensíveis aos dados disponíveis sobre o utente sinistrado, e apresentar essas informações ao operador ao longo da chamada.

Para além disto, deve ser garantido que as aplicações, e todos os dados por elas obtidos, são compatíveis com todos os projetos com os quais este trabalho se integra.

De forma a assegurar a conclusão destes objetivos foram definidas as seguintes tarefas, com o intuito de estruturar o trabalho que deve ser realizado:

- Revisão da literatura sobre computação móvel, computação móvel em saúde e sistemas de apoio à decisão;
- Estudo da arquitetura do sistema: análise de requisitos e linguagens de programação a utilizar;

- Construção da solução;
- Testes e validação do sistema;
- Documentação e relatório de projeto.

## 1.4 Organização do Documento

De forma a melhorar a estrutura deste documento, este encontra-se dividido em sete secções principais:

1. O primeiro capítulo – **Introdução** – é feita a apresentação do projeto, identificando os objetivos e a área em que este se enquadra;
2. O segundo capítulo – **Estado da Arte** – são apresentadas alguns projetos com objetivos semelhantes e é feita uma comparação com as funcionalidades desenvolvidas neste trabalho;
3. O terceiro capítulo – **Tecnologias e Ferramentas Utilizadas** – são apresentadas as tecnologias utilizadas no desenvolvimento das diversas partes deste projeto;
4. O quarto capítulo – **Engenharia de Software** – é feita uma descrição do serviço em que este projeto se enquadra, dos requisitos e das decisões tomadas em relação à arquitetura de cada uma das aplicações desenvolvidas;
5. O quinto capítulo – **Construção da Aplicação Móvel** – é feita uma descrição da aplicação desenvolvida para o uso da população geral e são descritas as funcionalidades implementadas em maior detalhe;
6. O sexto capítulo – **Construção e Testes da Aplicação Web** – é apresentada uma descrição da aplicação desenvolvida para o uso dos operadores dos serviços de emergência, são descritas as funcionalidades implementadas em maior detalhe e são apresentados os testes realizados para a avaliação do trabalho desenvolvido;
7. O sétimo capítulo – **Conclusões e Trabalho Futuro** – é feita uma revisão do documento e análise do trabalho realizado. São comparados os objetivos iniciais com os atingidos e discutidas ideias para a continuação do projeto.

# Capítulo 2

## Estado da Arte

### 2.1 Introdução

Neste capítulo é feita uma introdução aos temas da *Mobile Health* (mHealth) e IoT na área da saúde, onde são referidos os seus objetivos, a sua evolução e o seu enquadramento com os atuais serviços de emergência. São depois apresentadas algumas aplicações, que foram alvo de pesquisa durante a fase de planeamento, e é feita uma comparação com as funcionalidades planeadas para o projeto descrito neste documento.

### 2.2 IoT na Área da Saúde

A população mundial e a esperança de vida são estatísticas que têm registado um aumento significativo nos últimos anos. Este aumento vem como resultado do melhoramento das condições de vida das populações, incluindo um melhoramento nas instituições de saúde. No entanto, para manter este nível é necessário que a área da saúde continue a evoluir. [5]

Este processo evolutivo pode passar por utilizar as vantagens da IoT para complementar os serviços prestados. Por exemplo, a utilização de sensores para monitorizar os dados vitais dos paciente, ou dispositivos inteligentes para transmitir essa informação aos agentes necessários. [32] [29]

A junção destas duas áreas, para além de facilitar diversas tarefas, vem também aliviar os prestadores de saúde de alguns trabalhos monótonos, libertando-os para realizar outras tarefas de maior importância. [31]

De seguida são apresentadas as aplicações consideradas durante o estudo da área que esta secção reflete:

- CareAngel [7] - o objetivo desta plataforma é fornecer um serviço de assistência remota a idosos, recorrendo às tecnologias de IA e reconhecimento

de voz. O serviço permite que os utilizadores idosos façam chamadas para um assistente virtual, recolhendo assim diversas informações, tal como a pressão arterial. A plataforma trata depois de partilhar esta informação com os agentes autorizados (familiares ou profissionais de saúde);



Figura 2.1: Serviço CareAngel

- Apple Watch Healthcare [4] - este serviço utiliza os diversos sensores integrados no dispositivo inteligente no qual se enquadra, para recolher diferentes tipos de informação sobre os dados vitais do seu utilizador. Esse dados são depois tratados pela aplicação, com o objetivo de prever potenciais problemas de saúde que possam afetar o utilizador, avisando-o antecipadamente para que este possa procurar ajuda médica.

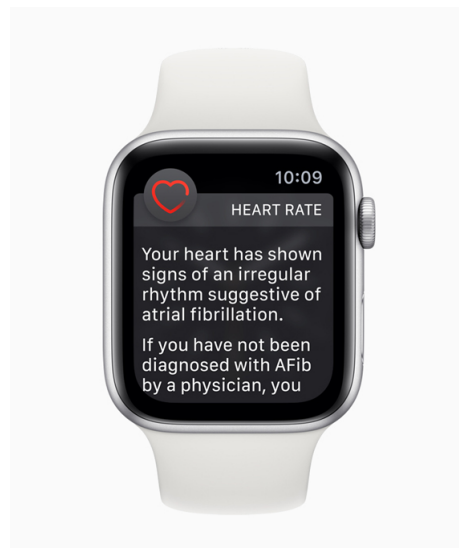


Figura 2.2: Serviço Apple Watch Healthcare

A informação recolhida dos serviços estudados foi tida em conta no planeamento deste projeto, tendo sido feito um levantamento das vantagens de cada um. Foram depois consideradas as características que faria sentido aplicar e como é que estas poderiam ser aplicadas no contexto do trabalho descrito neste documento.

## 2.3 Mobile Health

Os dispositivos móveis revolucionaram a forma como os seus utilizadores acessam e transmitem informação e o número de aplicações que tiram proveito destas características continuam a aumentar. Entre estas encontram-se as aplicações de mHealth, cujo objetivo é a melhoria das condições de saúde dos seus utilizadores, através da monitorização de dados vitais e registo do histórico de saúde. Estas aplicações têm também a capacidade de providenciar acesso imediato a estes dados, em qualquer altura e em qualquer lugar, e transmiti-los, com a mesma eficiência, para os serviços de emergência, em caso de urgência [22].

Da mesma forma, hospitais, centros de saúde e outros centros de assistência médica têm também evoluído no que toca à tecnologia de registo dos dados dos seus pacientes, o que se traduz numa melhoria na qualidade, segurança e rapidez do atendimento [30].

Se a estes fatores adicionarmos o melhoramento dos serviços de comunicação móvel, para o transporte de dados, conseguimos identificar uma oportunidade para o aparecimento de serviços que, ao incluírem todas estas inovações, criam uma ponte entre os utilizadores e os serviços de emergência. Para além de acelerarem a ativação dos meios de emergência, para o salvamento dos utilizadores, estes serviços poderão tornar-se essenciais no processo de decisão do tratamento de doentes, e permitir que estes sejam transferidos para diferentes unidades de saúde sem que os seus dados sejam perdidos no processo [16].

No entanto, apesar do mercado das aplicações mHealth se encontrar em evidente crescimento, e de avanços significativos se terem verificado, por exemplo, nos campos da Cardiologia [23], Obesidade [18] e requisição de serviços de emergência [6], este ainda se encontra muito dirigido apenas para utilizadores específicos, focando-se apenas num dos lados deste espectro.

Durante a fase de planeamento do trabalho apresentado neste documento, foram estudadas algumas aplicações móveis, que se integram na área estudada, de forma a compreender o mercado em que se integram. De seguida, destacam-se estas aplicações.

- Cardiógrafo [23] - esta aplicação tem como objetivo medir o ritmo cardíaco dos seus utilizadores e usar esses resultados para criar perfis individuais. A solução apresentada neste documento tem o objetivo de monitorizar diferentes dados sensoriais, para além daqueles recolhidos por esta aplicação,

fazendo depois uma demonstração destes dados, em tempo real, numa das suas atividades;

- Lose It! [18] - o objetivo desta aplicação (figura 2.3) é ajudar os seus utilizadores a perder peso, ao monitorizar a sua alimentação e fornecer informações detalhadas, como calorias e nutrientes. É depois criado um perfil individual, para cada utilizador, com objetivos e metas diárias. Tal como nesta aplicação, a solução apresentada pelo autor também deverá incluir um perfil individual para os seus utilizadores. Os dados recolhidos serão depois utilizados para a sua assistência, caso seja necessário.

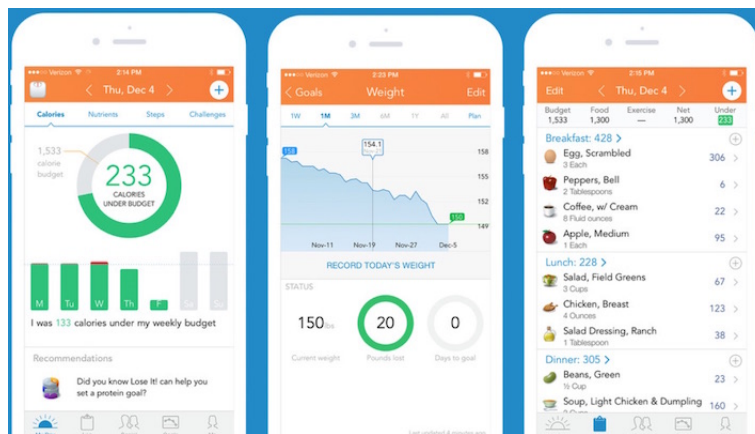


Figura 2.3: Aplicação - Lose It!

- 112 Where ARE U [6] - a aplicação (figura 2.4) permite realizar chamadas para o número de emergência europeu, enviando, automaticamente, para os serviços de emergência os dados de localização, e outros dados que o utilizador forneça à aplicação. A aplicação móvel, planeada no trabalho descrito neste documento, tem algumas características em comum com esta aplicação, como a recolha de dados GPS e a criação de um perfil para cada utilizador, no entanto irá fornecer mais funcionalidades que a diferenciarão, tal como a recolha de dados sensoriais, e a integração num serviço pré-estabelecido.



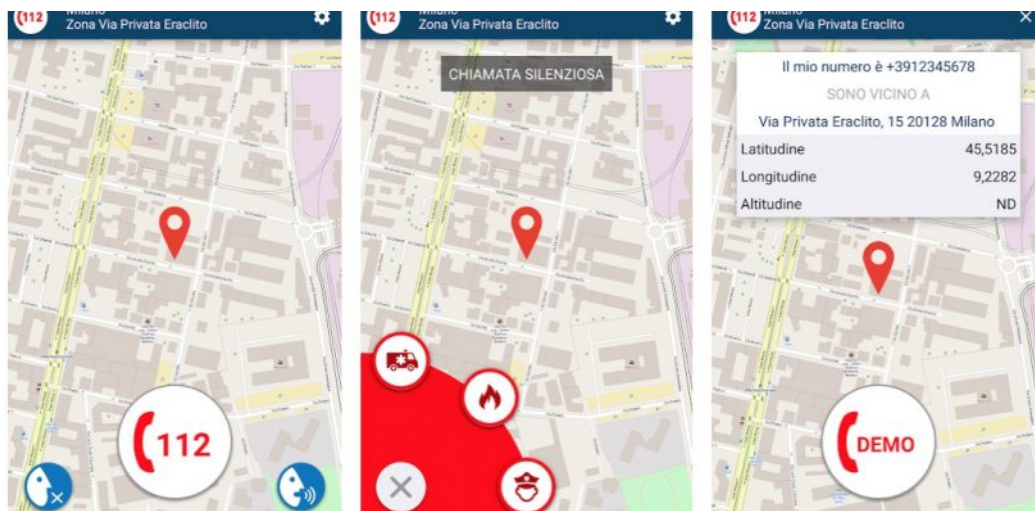


Figura 2.4: Aplicação - 112 Where ARE U

Todas as aplicações mencionadas têm características em comum com o projeto, descrito neste documento, no entanto este é mais abrangente pois tem como objetivo recolher um conjunto maior de dados e trabalhar diretamente com um sistema já desenvolvido, para transmitir estes dados, em tempo real, aos serviços de emergência. Outra característica diferenciadora é a incorporação de uma funcionalidade de processamento de voz, do lado dos operadores dos serviços de emergência, com o objetivo de diminuir o tempo necessário para avaliar a situação, e disponibilizar os meios de socorro mais adequados.

## 2.4 Conclusões

Com este capítulo é feito um enquadramento das aplicações estudadas, durante a fase de planeamento deste projeto, com a área em que estas se inserem. É feita uma descrição das funcionalidades que serviram de inspiração para a definição de alguns dos requisitos da solução apresentada neste documento e são também mencionados alguns pontos que se pretende aprimorar. Nos capítulos seguintes estes requisitos serão apresentados, bem como as tecnologias usadas para os desenvolver.



## Capítulo 3

# Tecnologias e Ferramentas Utilizadas

### 3.1 Introdução

Este capítulo descreve as tecnologias utilizadas para desenvolver as diferentes partes que compõem este trabalho. Primeiro são referidas as tecnologias utilizadas no desenvolvimento da aplicação móvel e, de seguida, as tecnologias usadas na aplicação *web*. Por fim, referem-se as ferramentas usadas na elaboração deste documento.

### 3.2 Aplicação Móvel

#### 3.2.1 Android Studio

O Android Studio é o Ambiente de Desenvolvimento Integrado (IDE) oficial no desenvolvimento de aplicações com o sistema operativo Android. O Android Studio oferece diversos recursos para aumentar a produtividade na criação de aplicações, tais como um editor de código e avançadas ferramentas de desenvolvedor. [14]

#### 3.2.2 Android

É um sistema operativo, desenvolvido pela empresa Google, para dispositivos móveis tais como *smartphones* ou *tablets*. É atualmente a plataforma de desenvolvimento móvel mais utilizada, e isto deve-se principalmente às seguintes razões: é gratuita, existem várias ferramentas, também estas gratuitas, disponíveis para

o desenvolvimento de aplicações, e é a plataforma dominante no mercado dos dispositivos móveis.

### 3.2.3 Java

É uma linguagem de programação, orientada a objetos, desenvolvida na década de 90 e adquirida, em 2008, pela empresa Oracle Corporation. O Java é o padrão global para o desenvolvimento de aplicações móveis e conteúdos baseados na *web*. Esta linguagem conta com mais de 9 milhões de desenvolvedores em todo o mundo [9].

## 3.3 Aplicação Web

### 3.3.1 Javascript

É uma linguagem de programação, de alto nível, primariamente usada no desenvolvimento *web*. O seu objetivo é complementar os *websites* com elementos dinâmicos e interativos.

### 3.3.2 NodeJS

Node.js é um ambiente de tempo de execução que permite executar código, em Javascript, fora do *browser* como uma aplicação independente. Isto permite criar ferramentas para linhas de comando e *scripts* de servidores para produzir conteúdo dinâmico para as páginas *web*. [25]

### 3.3.3 HTML5

*HyperText Markup Language* (HTML) é uma linguagem de marcação utilizada na construção de páginas *web*. Foi desenvolvida por Tim Berners-Lee e a sua primeira versão foi publicada em 1991. Graças a sua popularidade é considerado o padrão oficial da *web* e as suas especificações são mantidas e desenvolvidas pelo *World Wide Web Consortium* (W3C).

### 3.3.4 CSS3

*Cascading Style Sheets* (CSS) é a linguagem utilizada para descrever estilos reutilizáveis para a apresentação de documentos escritos numa linguagem de marcação. Em 1996, foi especificada pelo W3C e hoje é utilizada para editar a aparência das páginas *web*.

## **3.4 Relatório**

### **3.4.1 LaTeX**

É uma linguagem de marcação usada na preparação de documentos, que possui recursos para a produção de documentação técnica e científica. Este sistema permite que o utilizador se foque no conteúdo do documento, sem ter de se preocupar com a formatação. [27]

## **3.5 Conclusões**

Em suma, este capítulo descreve as tecnologias estudadas e aplicadas na elaboração do projeto. Nas próximas secções será demonstrado como é que estas tecnologias foram aplicadas e qual o impacto de cada uma no resultado final deste trabalho.



## Capítulo 4

# Engenharia de Software

### 4.1 Introdução

Este capítulo tem como principal objetivo detalhar as decisões tomadas em relação à arquitetura e *design* das aplicações, durante o seu planeamento.

A primeira secção apresenta a arquitetura de sistema do HIAS, onde são apresentados os projetos que este envolve e como os mesmos interagem entre si.

Depois é feito um levantamento de requisitos das duas aplicações e, por fim, são apresentados vários diagramas que representam as ações que poderão ser tomadas pelo utilizador durante a execução das mesmas.

### 4.2 Arquitetura de Sistema do HIAS

O objetivo desta secção será apresentar o HIAS e alguns dos projetos, desenvolvidos por diferentes alunos (de licenciatura e mestrado), que este inclui e que interagem diretamente com o trabalho descrito neste relatório. Para tal será feita uma breve descrição de dois projetos onde é explicada a contribuição de cada um para o sistema e a forma como estes interagem com o trabalho.

A figura 4.1 é uma representação desta arquitetura, onde podem ser visualizados todos os módulos que esta inclui e as diferentes interações, entre eles. De forma a simplificar a sua compreensão, foi utilizado um esquema de cores para legendar os diferentes módulos. Assim, a azul encontram-se representados os módulos constituintes do trabalho do autor, e a laranja os módulos ao encargo de outros alunos.

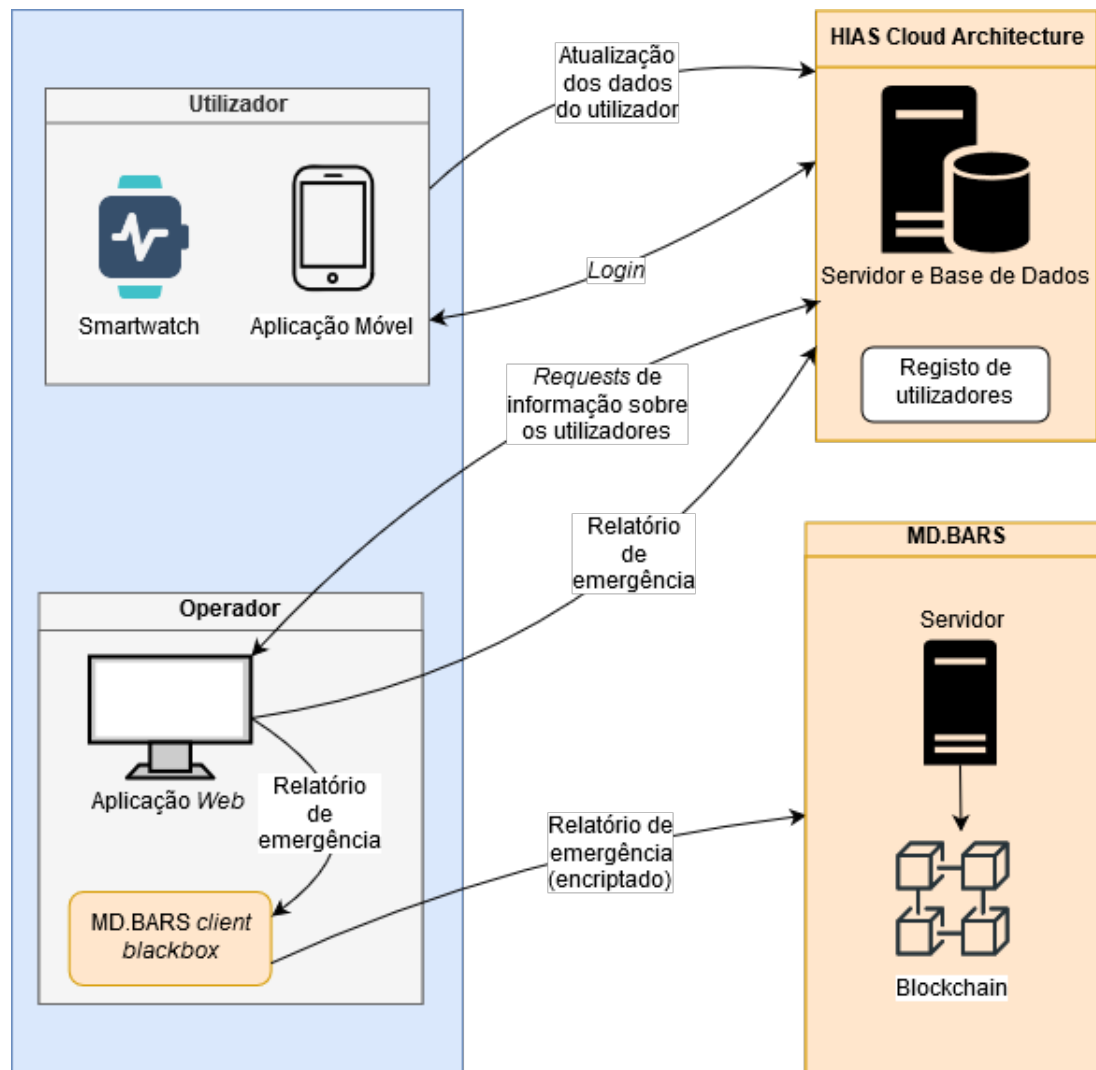


Figura 4.1: Desenho Conceptual da Arquitetura de Sistema do HIAS



### 4.2.1 HIAS Cloud Architecture

Este projeto, criado no âmbito de um projeto de licenciatura tem como objetivo a criação de um servidor capaz de tratar dados provenientes das diferentes aplicações constituintes do HIAS. O servidor encontra-se ligado a uma base de dados onde os dados são armazenados e onde poderão ser acessadas a pedido de qualquer uma das aplicações ligadas ao sistema. Isto facilita a partilha de informações e diminui a redundância de dados entre as diferentes aplicações. Para além disto, dentro do âmbito deste projeto, foi também criado uma página de registo para a qual são redirecionados os utilizadores das diferentes aplicações, permitindo assim que estes tenham acesso à totalidade do sistema, e a todas as funcionalidades que este inclui, não apenas a um dos serviços.

No âmbito das aplicações desenvolvidas, no presente projeto, não existe qualquer troca de dados entre as mesmas, devendo os dados primeiro ser guardados na base de dados. Portanto, podemos dividir as interações com o servidor em duas categorias: interações Aplicação Móvel - Servidor e interações Aplicação *Web* - Servidor.

Na categoria, Aplicação Móvel - Servidor, consideramos os seguintes eventos:

- envio de dados para atualizar as informações do utilizador;
- envio de dados para armazenar as informações recolhidas pelos sensores;
- enviar a informação que o utilizador iniciou uma chamada de emergência, informação essa que será depois transmitida para a aplicação *web*.

Na categoria, Aplicação *Web* - Servidor, consideramos os seguintes eventos:

- receber informação sobre o utilizador que iniciou a chamada de emergência;
- pedir informações adicionais sobre o mesmo utilizador;
- enviar um relatório, depois do fim da chamada, com alguns dados sobre os intervenientes e algumas informações sobre o desfecho da situação reportada, para ser armazenado na base de dados.

### 4.2.2 *Medical Data Blockchain Access Record System (MD.BARS)*

O segundo módulo da arquitetura do sistema, com que este projeto comunica, foi criado no âmbito de uma dissertação de mestrado e tem como objetivo a criação de uma forma segura de consultar e tratar informações sobre o histórico de saúde dos utilizadores do HIAS sem que informações pessoais sobre os proprietários desses mesmos dados sejam reveladas. Para tal, este serviço foi desenvolvido

com base na plataforma Ethereum [1], uma plataforma descentralizada e baseada no protocolo de *blockchain* [19].

Este projeto integra-se com o HIAS, como referido em cima, ao permitir o tratamento anónimo das informações dos utilizadores do sistema, obtidas graças a relatórios gerados após a conclusão de uma chamada de emergência. No fim de uma chamada, é pedido ao operador que faça uma pequena descrição da ocorrência que acabou de atender e, no final, essa informação é enviada, juntamente com o email do sinistrado e a data da ocorrência, para o MD.BARS *client blackbox*. Aqui a informação é encriptada e, por fim, enviada para o servidor do MD.BARS. Uma descrição mais detalhada deste processo encontra-se em 6.2.5.

## 4.3 Análise de Requisitos

Nesta secção é feito um levantamento dos requisitos funcionais e não funcionais das diferentes aplicações desenvolvidas para este projeto. Graças a estes requisitos é possível fazer um estudo das necessidades do utilizador e definir objetivos para o desenvolvimento das aplicações.

### 4.3.1 Requisitos para Aplicação Web

#### Requisitos Funcionais

Foram definidos os seguintes requisitos tendo em conta as necessidades dos operadores e do sistema:

**[RF 01]** - Ao iniciar a aplicação, esta deve comunicar com o servidor do HIAS de forma a receber a informação sobre o utilizador que iniciou a chamada de emergência. Depois de recebida, essa informação deve ser armazenada em memória.

**[RF 02]** - O ecrã deve estar dividido, verticalmente, em duas secções. Na direita devem ser apresentadas, constantemente, as seguintes informações: nome do utilizador, número de telemóvel, idade, género, batimento cardíaco, temperatura corporal e localização do utilizador. No lado esquerdo serão apresentadas as restantes informações sobre o utilizador, à medida que o operador as requisitar.

**[RF 03]** - Durante a chamada, a aplicação deve utilizar a tecnologia *speech-to-text* para processar a voz do operador e identificar certas palavras-chave. Cada uma destas palavras está associada a alguma informação que deve ser mostrada no ecrã caso a palavra chave seja identificada.

**[RF 04]** - A aplicação deve estar em comunicação constante com o servidor de forma a atualizar os dados dos sensores em tempo real.

**[RF 05]** - A aplicação deve ser capaz de identificar se os valores recolhidos pelos sensores atingem certos limites. Isto pode significar que o utilizador se encontra numa situação de risco, devendo ser exibido um aviso no ecrã de forma a informar o operador.

**[RF 06]** - No topo superior direito do ecrã deve existir um botão que o operador deverá pressionar no fim da chamada. Este botão vai desencadear o aparecimento de um formulário onde o operador deverá inserir as seguintes informações: nome e resumo da ocorrência. Para além destas informações,

a aplicação deve preencher automaticamente a data da ocorrência e o nome do utilizador.

[RF 07] - As informações recolhidas do formulário, no fim da chamada, deverão ser enviadas para o servidor HIAS e para o servidor MD.BARS.

### Requisitos Não Funcionais

Nesta secção, para uma melhor compreensão, os requisitos foram separados nas seguintes categorias:

- Usabilidade

[RNF 01] - A aplicação deve funcionar em qualquer um dos seguintes *browsers*: Google Chrome, Mozilla Firefox e Microsoft Edge.

[RNF 02] - A aplicação deve permitir que utilizadores com conhecimento básico em tecnologias *web* sejam capazes de usar a aplicação em toda a sua extensão.

- Disponibilidade

[RNF 03] - A aplicação deve ser capaz de comunicar com o servidor do serviço HIAS em qualquer altura do dia.

- Segurança

[RNF 04] - Antes de serem enviados para o servidor do projeto MD.BARS, os dados devem passar por um módulo, instalado localmente, que os irá encriptar segundo os critérios definidos pelo desenvolvedor desse projeto.

- Eficiência

[RNF 05] - Os *requests* entre a aplicação e o servidor do serviço HIAS não devem atrasar o serviço.

### 4.3.2 Requisitos para Aplicação Móvel

#### Requisitos Funcionais

Foram definidos os seguintes requisitos tendo em conta as necessidades dos utilizadores e do sistema:

[RF 01] - Ao abrir a aplicação, o sistema deve verificar se o utilizador já se encontra com sessão iniciada. Caso isto se confirme, o utilizador é levado para a página principal, caso contrário é redirecionado para a página de autenticação.

[RF 02] - O sistema deve permitir que o utilizador faça *login* com um *email* e uma *password*. Estas informações devem depois ser enviadas para o servidor, onde os dados inseridos devem ser confirmados. Caso a autenticação seja confirmada, o utilizador deve ser direcionado para a página principal, caso contrário aparecerá uma mensagem de erro no ecrã e o utilizador deverá repetir o processo.

[RF 03] - Na atividade de *login* deve existir um botão que direcione o utilizador para a página de registo do HIAS.

[RF 04] - A página principal da aplicação deve dar acesso às seguintes funcionalidades: "Editar dados", "Informações em tempo real", "Terminar sessão" e "Iniciar chamada de emergência".

[RF 05] - Na janela "Editar dados" o sistema deverá permitir alterar as seguintes informações: número de telefone, morada, peso, altura, tipo sanguíneo, se tem ou não epilepsia, se tem ou não asma, se tem ou não alergias, e observações.

[RF 06] - Na janela "Informações em tempo real", devem ser apresentados dois gráficos que mostrem os dados obtidos pelos sensores da pulseira: o primeiro deve mostrar os dados de batimento cardíaco e o segundo os dados da temperatura corporal. Para além disto, a aplicação deve ser capaz de obter informações exatas sobre as coordenadas GPS e apresentar essas informações na forma de um mapa.

[RF 07] - Ao clicar em "Iniciar chamada de emergência", a aplicação deverá executar duas tarefas: iniciar uma chamada para o número de emergência pré-definido; enviar, para o servidor, informações sobre o utilizador que iniciou a chamada. O servidor tratará depois de reenviar estas informações para a aplicação *web*.

[RF 08] - No menu principal, ao clicar no botão "Terminar sessão", a aplicação deve fechar, terminado a sessão do utilizador. Desta forma, da próxima vez que o utilizador abrir a aplicação este deverá ser redirecionado para a página de *login*.

### Requisitos Não Funcionais

Nesta secção, para uma melhor compreensão, os requisitos foram separados nas seguintes categorias:

- Usabilidade:

[RNF 01] - A aplicação deve ser suportada por qualquer *smartphone*, com sistema operativo Android, com versão 7.0 ou superior.

[RNF 02] - A aplicação deve ser desenhada de forma a que qualquer utilizador com conhecimentos básicos em dispositivos Android seja capaz de utilizar a aplicação em toda a sua extensão.

- Disponibilidade:

[RNF 03] - O sistema deve estar disponível para ser acedido 24 horas por dia.

[RNF 04] - O sistema deve estar disponível para trocar informações com o servidor a qualquer altura.

[RNF 05] - O sistema deve ser capaz de funcionar mesmo sem conexão à Internet ou sem serviço GPS usando, nestes casos, as informações mais recentemente armazenadas sobre estes serviços.

- Segurança:

[RNF 06] - A aplicação deve pedir credenciais de autenticação sempre que o utilizador não se encontre já autenticado.

[RNF 07] - As credenciais devem ser compostas por um *email* válido e uma palavra-passe.

[RNF 08] - A palavra-passe deve ser encriptada usando o algoritmo *Secure Hashing Algorithm*, 256-Bits (SHA-256).

- Eficiência:

[RNF 09] - Os *requests* entre a aplicação e o servidor não devem atrasar o serviço.

## 4.4 Casos de Uso

Nas próximas duas secções são apresentados os diagramas de casos de uso para cada uma das aplicações desenvolvidas. A cada diagrama está associado um conjunto de tabelas onde cada caso de uso é descrito em maior detalhe.

### 4.4.1 Casos de Uso para Aplicação Web

Na figura 4.2 pode ser verificado que o autor para esta aplicação será o Operador (dos serviços de emergência), e também que existem dois casos em que este irá interagir com a aplicação. É depois apresentada uma tabela, para cada um destes casos, onde é descrito o ator, as pré-condições, os dados e a descrição de cada um.

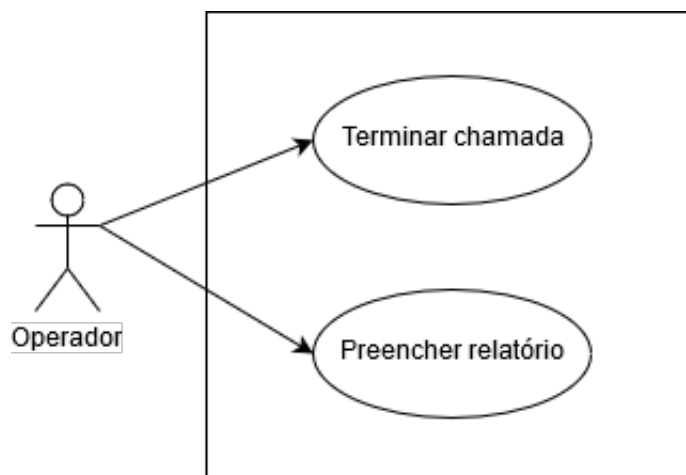


Figura 4.2: Diagrama de Casos de Uso da Aplicação Web

<b>Nome</b>	Terminar chamada
Ator	Operador
Pré-condições	N/A
Dados	N/A
Descrição	Um botão no canto superior direito do ecrã que permite ao operador dar a indicação ao programa que a chamada terminou, o que vai desencadear o aparecimento de um formulário com informações sobre a chamada.

Tabela 4.1: Caso 1 - Terminar Chamada

<b>Nome</b>	Preencher relatório
Ator	Operador
Pré-condições	N/A
Dados	Nome do operador, nome do utilizador, data, resumo da chamada
Descrição	É apresentado um formulário com os seguintes dados: nome do operador, nome do utilizador, data e resumo da chamada. O nome do utilizador e a data devem ser preenchidos automaticamente pela aplicação, enquanto que o nome do operador e o resumo da chamada devem ambos conter campos editáveis que permitam o preenchimento por parte do operador.

Tabela 4.2: Caso 2 - Preencher Relatório



#### 4.4.2 Casos de Uso para Aplicação Móvel

Na figura 4.3 pode ser verificado que o autor para esta aplicação será o Utilizador, e também que existem cinco casos em que este irá interagir com a aplicação. É depois apresentada uma tabela, para cada um destes casos, onde é descrito o ator, as pré-condições, os dados e a descrição de cada um.

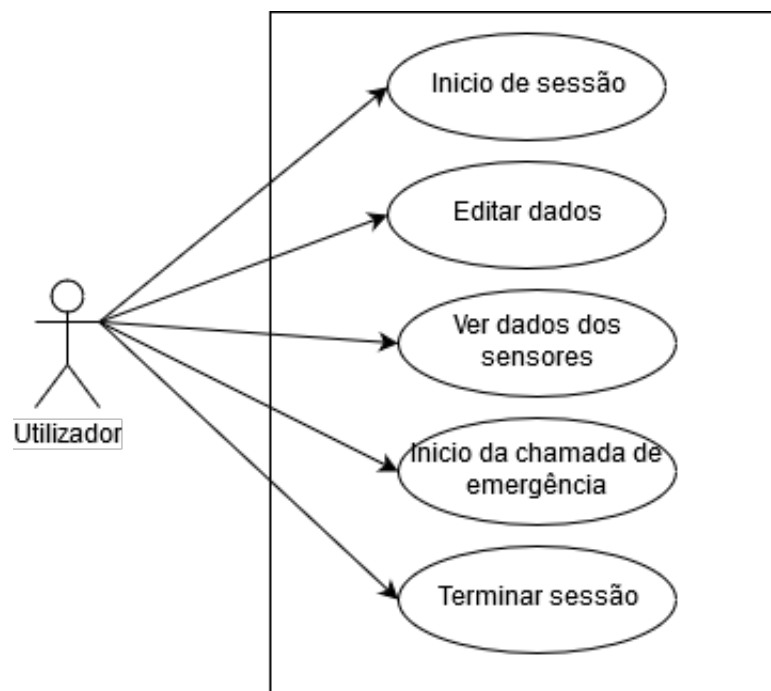


Figura 4.3: Diagrama de Casos de Uso da Aplicação Móvel

<b>Nome</b>	Início de Sessão
<b>Ator</b>	Utilizador
<b>Pré-condições</b>	Utilizador já se encontrar registado no HIAS
<b>Dados</b>	<i>Email</i> e palavra passe
<b>Descrição</b>	São apresentadas duas caixas de texto onde o utilizador colocará o <i>email</i> e a palavra passe, respetivamente. Por baixo são apresentados dois botões com as seguintes denominações: <u>Login</u> e <u>Registar</u> . Ao clicar no botão <u>Login</u> os dados são enviados para o servidor, com a palavra passe encriptada, onde serão verificados. Por outro lado, ao clicar em <u>Registar</u> , o utilizador é enviado para o <i>website</i> de registo do HIAS.

Tabela 4.3: Caso 1 - Iniciar Sessão

<b>Nome</b>	Editar dados
<b>Ator</b>	Utilizador
<b>Pré-condições</b>	N/A
<b>Dados</b>	Dados Pessoais (Número de telefone, Morada) e Dados de Saúde (Peso, Altura, Tipo Sanguíneo, Diabetes, Epilepsia, Asma, Alergias, Observações)
<b>Descrição</b>	É apresentado um formulário com todos os dados, editáveis, que constituem o perfil do utilizador. No final da atividade existirá um botão, denominado <u>Guardar</u> . Ao terminar as alterações que achar necessárias, o utilizador clicará no botão <u>Guardar</u> e os dados alterados serão enviados para o servidor, e ao mesmo tempo, armazenados na memória do dispositivo.

Tabela 4.4: Caso 2 - Editar Dados

<b>Nome</b>	Ver dados dos sensores
<b>Ator</b>	Utilizador
<b>Pré-condições</b>	N/A
<b>Dados</b>	Coordenadas GPS, Batimento cardíaco, Temperatura corporal
<b>Descrição</b>	<p>Nesta janela serão apresentados dois gráficos que mostrarão os dados referentes ao batimento cardíaco e à temperatura corporal do utilizador, que estarão a ser recolhidos através de sensores.</p> <p>De seguida serão apresentadas as coordenadas GPS do utilizador, seguidas por um mapa que mostrará a sua localização.</p>

Tabela 4.5: Caso 3 - Ver Dados dos Sensores

<b>Nome</b>	Iniciar chamada de emergência
<b>Ator</b>	Utilizador
<b>Pré-condições</b>	N/A
<b>Dados</b>	N/A
<b>Descrição</b>	<p>Na janela principal, ao clicar no botão <u>Chamada de Emergência</u> as informações do utilizador são enviadas para o servidor do HIAS, onde serão depois reencaminhadas para o operador que ficará encarregado pela chamada, e uma chamada para o número 112 é iniciada.</p> <p>Enquanto a chamada estiver a decorrer, os dados recolhidos pelos sensores serão enviadas, em tempo real, para o servidor, que tratará de as reencaminhar para o operador.</p>

Tabela 4.6: Caso 4 - Iniciar Chamada de Emergência

<b>Nome</b>	Terminar sessão
<b>Ator</b>	Utilizador
<b>Pré-condições</b>	N/A
<b>Dados</b>	N/A
<b>Descrição</b>	<p>Esta ação fará com que os dados armazenados no dispositivo sejam eliminados e o utilizador será enviado para a página de <i>login</i> onde terá que realizar uma nova autenticação.</p>

Tabela 4.7: Caso 5 - Terminar sessão

## 4.5 Diagramas de Atividades

Os diagramas de atividade são usados para descrever a coordenação entre as diferentes atividades de um sistema. No contexto deste documento serão mostrados dois destes diagramas, um para cada aplicação desenvolvida, que representam a coordenação entre as diferentes atividades de cada uma.

### 4.5.1 Diagramas de Atividades para Aplicação Web

Para demonstrar as diferentes fases que devem ocorrer ao utilizar a aplicação, e como esta se deve comportar em cada uma, desenvolveu-se um conjunto de diagramas, cada um representando uma destas fases. Quando uma chamada é iniciada (figura 4.4), é enviado um *request* para o servidor HIAS onde são requisitados os dados do utilizador que iniciou a chamada. Os dados recebidos são então guardados em memória, e os mais relevantes para o início da conversa são imediatamente apresentados no ecrã. Nesta altura, o operador encontra-se em condições de iniciar a conversa com o utilizador que acionou a chamada de emergência.

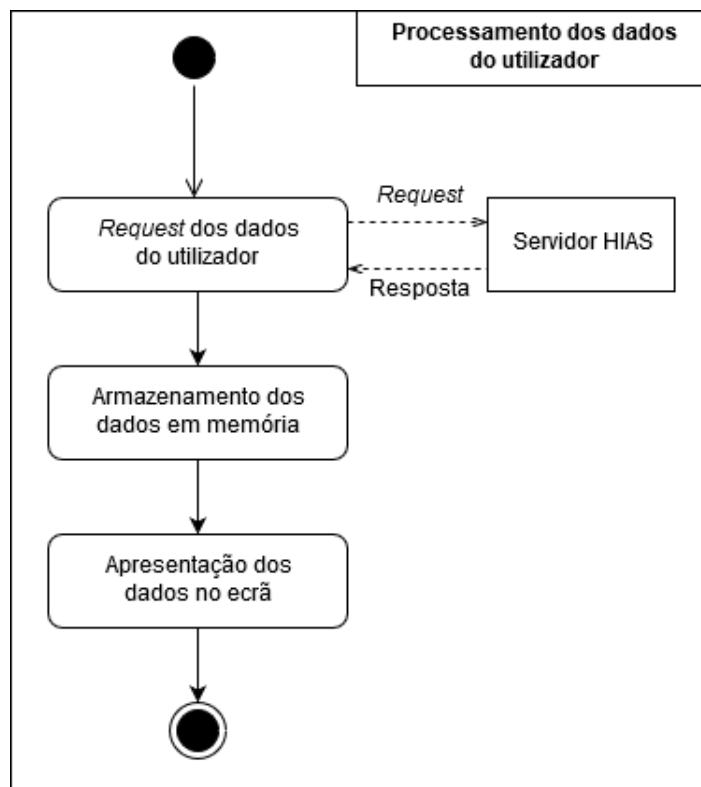


Figura 4.4: Diagrama de Atividades da Aplicação Web - Início da Chamada

Durante esta conversa, a aplicação deverá processar as frases do operador (figura 4.5) e procurar por certas palavras chave, pré-definidas. Cada uma destas palavras estará associada a certas informações, que serão mostradas ao operador caso a palavra correspondente seja encontrada.

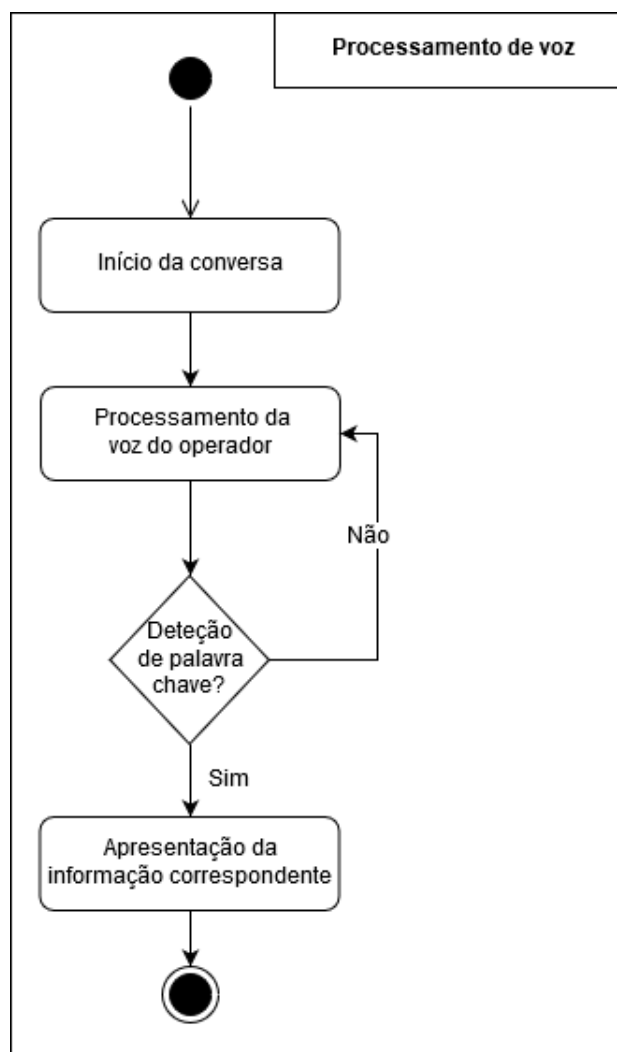


Figura 4.5: Diagrama de Atividades da Aplicação Web - Processamento de Voz

Ao mesmo tempo, a aplicação deve realizar *requests* constantes para o servidor HIAS para receber novas informações atualizadas dos dados recolhidos pelos sensores (figura 4.6). Com a resposta do servidor, a aplicação deverá verificar se os valores recebidos se encontram dentro de um intervalo considerado normal. Caso os valores não se encontrem dentro deste intervalo, um alerta deverá ser apresentado ao operador, informando-o desta situação. Depois disto, ou caso os valores

se encontrem dentro do normal, a informação sobre os sensores apresentada no ecrã é atualizada.

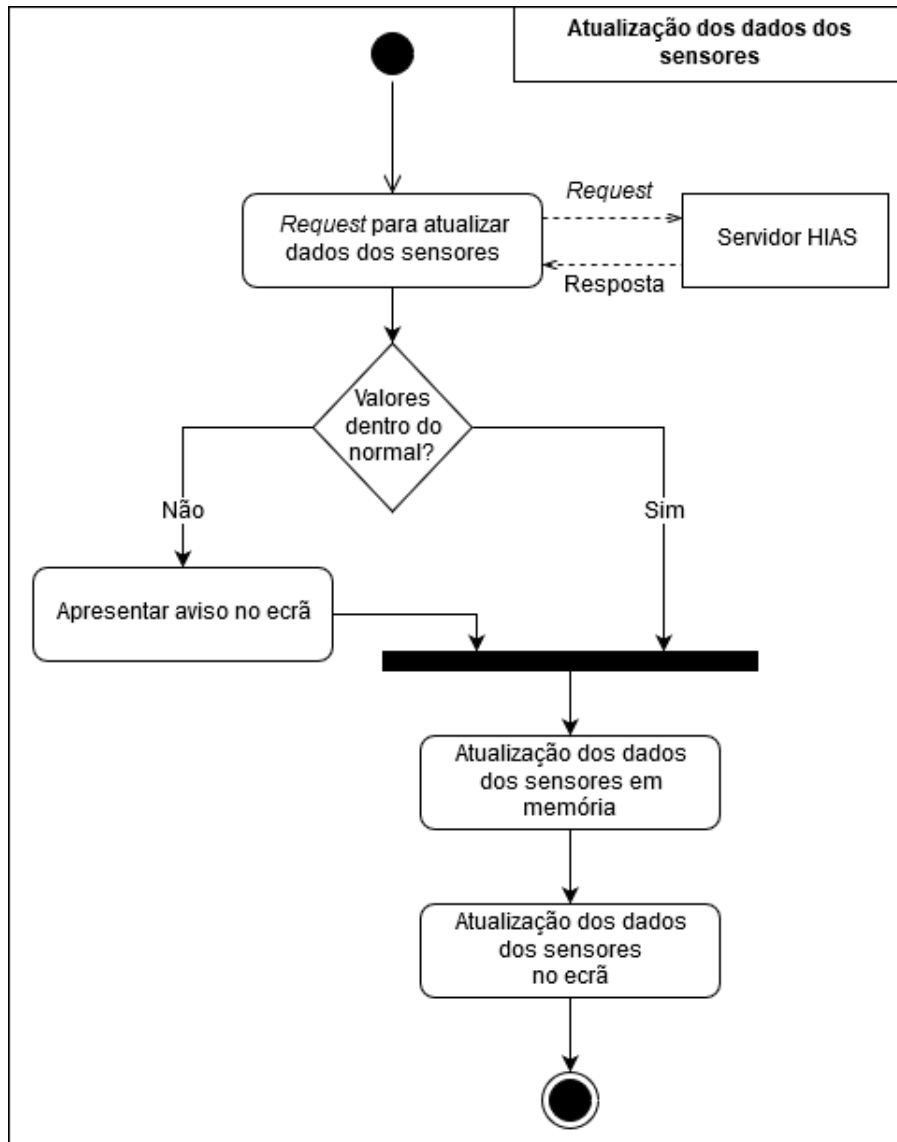


Figura 4.6: Diagrama de Atividades da Aplicação Web - Atualizar Sensores

Ao terminar a conversa (figura 4.7), será pedido ao operador que complete um formulário com algumas informações sobre a ocorrência que acabou de atender. As informações descritas neste formulário serão depois enviadas para dois servidores diferentes: o primeiro será o servidor HIAS onde as informações serão adicionadas ao histórico de chamadas, e o segundo será o servidor MD.BARS.

No entanto, antes de chegarem a este servidor, as informações terão de ser enviadas, localmente, para um módulo onde serão encriptadas, e só então poderão ser enviadas para o servidor MD.BARS onde passarão a fazer parte de uma coleção anónima de dados de saúde de diferentes pacientes.

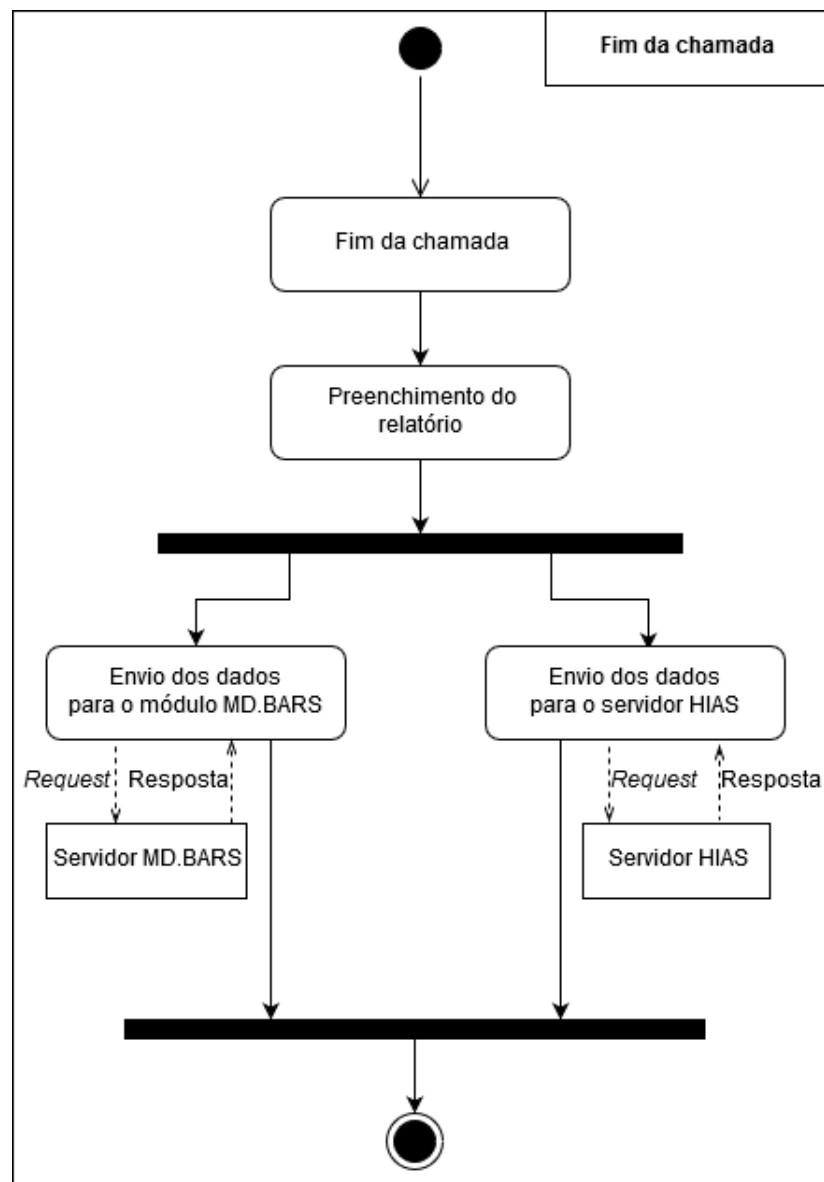


Figura 4.7: Diagrama de Atividades da Aplicação Web - Fim da Chamada

### 4.5.2 Diagrama de Atividades para Aplicação Móvel

O diagrama representado em 4.8 mostra o fluxo entre as atividades da aplicação móvel. Ao iniciar a aplicação o utilizador pode ser levado para uma de duas atividades: no caso de já se encontrar autenticado será levado para o Menu Principal, caso contrário será a página de Login que irá ser aberta.

Neste último caso, o utilizador poderá inserir os seus dados do HIAS, que serão depois enviados para o servidor onde serão então verificados. No caso de ainda não se encontrar registado, o utilizador poderá fazê-lo no *website* para onde será redirecionado ao escolher essa opção. Após realizar uma autenticação correta, o utilizador é então direcionado para o Menu Principal.

Na atividade Menu Principal o utilizador terá acesso a todas as funcionalidades desta aplicação, e a partir daqui poderá realizar uma das seguintes tarefas:

- Editar dados - nesta atividade o utilizador poderá verificar as informações guardadas no seu perfil, atualizá-las e adicionar novos dados que não se encontrem preenchidos. Estes dados são depois enviados para o servidor do HIAS onde são depois armazenados;
- Ver informação em tempo real - nesta atividade o utilizador pode verificar os dados que estão a ser recebidos pelos sensores associados ao sistema;
- Iniciar chamada de emergência - esta é a funcionalidade principal da aplicação, e ao ser selecionada é enviado um *request* para o servidor de forma registar a chamada no serviço. Ao mesmo tempo, é iniciada uma chamada para o número de emergência 112;
- Terminar sessão - nesta opção a aplicação irá eliminar da memória toda a informação armazenada sobre o utilizador, e este será redirecionado para a página de Login da próxima vez que iniciar a aplicação.



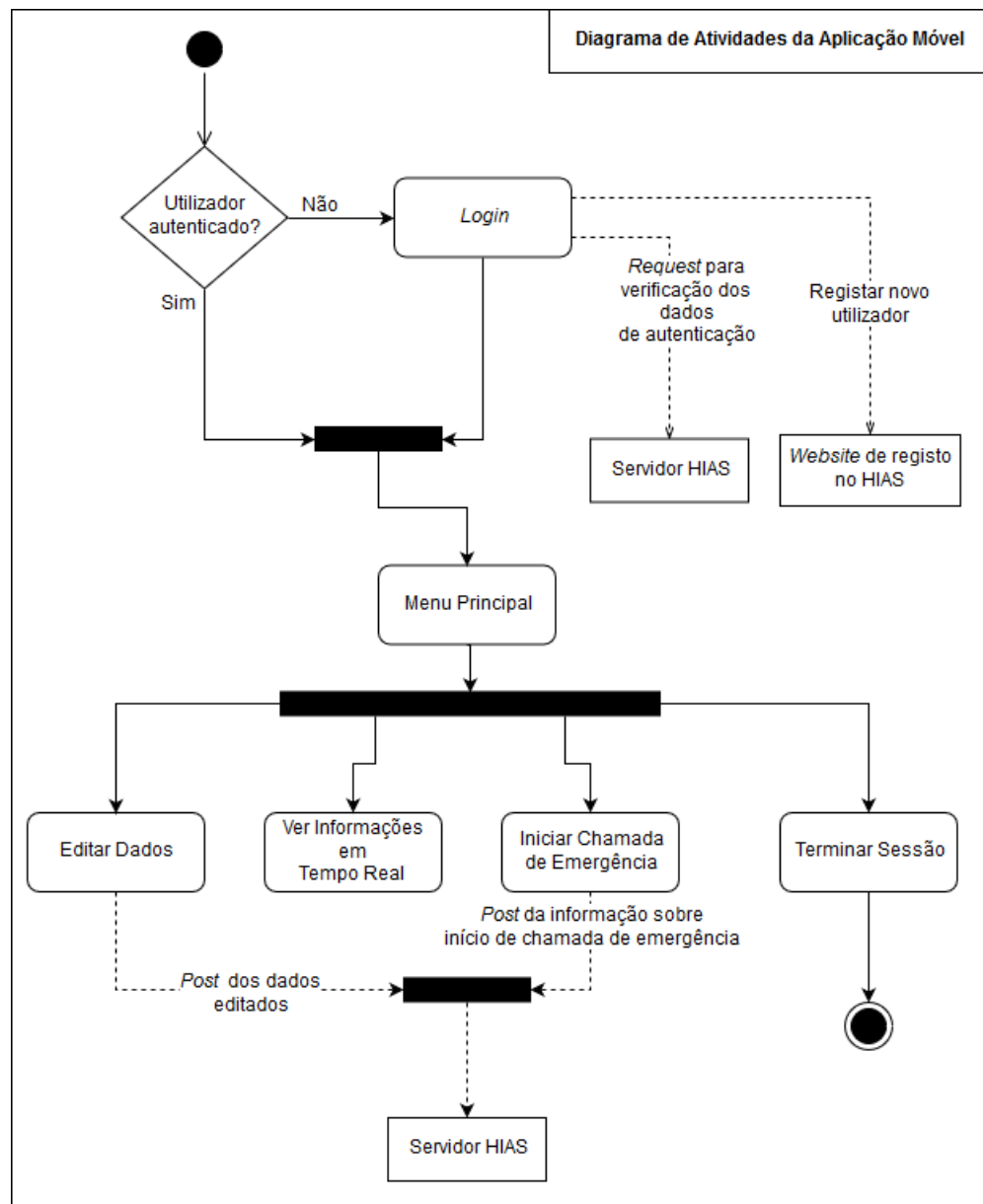


Figura 4.8: Diagrama de Atividades da Aplicação Móvel

## 4.6 Conclusões

A partir da planificação apresentada neste capítulo foi feito o desenvolvimento do trabalho, que se encontra apresentado nas próximas secções. No contexto do próprio documento, neste capítulo são apresentadas todas as informações necessárias para visualizar as funcionalidades disponíveis em ambas as aplicações.

## Capítulo 5

# Construção da Aplicação Móvel

### 5.1 Introdução

Neste capítulo é feita a descrição da aplicação móvel, desenvolvida para o uso da população geral, e é demonstrado como foram implementados os diversos requisitos descritos nos capítulos anteriores. Para tal, são explicados diversos detalhes da solução desenvolvida e é apresentado um manual de utilizador onde é mostrado como aceder e utilizar as funcionalidades implementadas.

### 5.2 Detalhes da Construção

#### 5.2.1 Autenticação de utilizadores

Neste secção será descrito o processo de autenticação de utilizadores na aplicação móvel desenvolvida. É importante referir que para tal o utilizador já deve estar registado no HIAS e que a implementação deste registo é uma funcionalidade externa ao trabalho do autor. No entanto, é possível aceder à página de registo a partir da aplicação usando o botão Registar que se encontra na atividade de *login*.

Para se autenticar, o utilizador deverá inserir o *email* e palavra-passe utilizados no registo do HIAS. Estes dados são depois enviados para o servidor, onde serão verificados. Mais informações sobre o processo de transmissão de dados para o servidor podem ser encontradas no capítulo 5.2.3.

Antes disso, no entanto, os dados devem ser preparados para serem enviados, mais especificamente, a palavra-passe deve ser protegida segundo os requisitos definidos no planeamento do projeto.

Como pode ser visualizado na figura 5.1, foram criadas duas funções para este processo. A primeira irá encriptar o texto-limpo recebido usando a função de *hash* SHA-256 e devolver a mensagem em bytes. A segunda função recebe esta

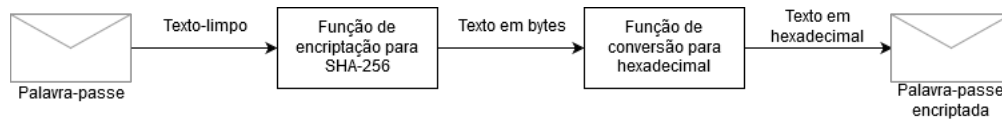


Figura 5.1: Processo de Encriptação da Palavra-Passe

mensagem e converte-a para hexadecimal. Nesta altura a palavra-passe encontra-se pronta para ser enviada para o servidor.

Caso a verificação da autenticação seja positiva, os dados do utilizador são recebidos do servidor e procede-se para o seu armazenamento, que pode ser consultado no capítulo 5.2.2. Caso contrário, é apresentada uma mensagem flutuante a informar o utilizador que os dados inseridos se encontram incorretos e *login* deve ser repetido.

## 5.2.2 Armazenamento de Dados

De forma a manter um registo local dos dados do utilizador, recebidos do servidor, foram utilizadas *SharedPreferences*. Este método de armazenamento é utilizado para guardar tipos primitivos de dados, a que se pretenda aceder mais tarde, e garante uma forte consistência das informações guardadas [15]. Apesar de existirem outros métodos que permitem o armazenamento de grandes quantidades de informação, devido à baixa complexidade dos dados tratados, este método foi preferido graças a sua simplicidade.

Após o utilizador se autenticar na aplicação, com sucesso, as suas informações pessoais e de saúde são enviadas pelo servidor e devem ser armazenadas para poderem ser apresentadas noutras atividades. Para tal, foi criada uma classe, nomeada "*user.data*", dentro da qual foi criado um tipo primitivo de dados para cada informação que seria necessário guardar.

Para editar estes dados, é então necessário instanciar a classe referida em cima, e passar todas as alterações pretendidas por um objeto *Editor* que irá garantir a sua consistência. Por fim, é feita a submissão dos dados e estes são armazenados em memória. No trecho de código seguinte demonstra-se como foi implementado o armazenamento de dados após estes terem sido recebidos do servidor:

```
SharedPreferences mySharedPreferences ;
...
mySharedPreferences = getSharedPreferences("user.data", Activity
.MODE_PRIVATE);
SharedPreferences.Editor editor = mySharedPreferences.edit();
editor.putString("Email", object.getString("email"));
editor.putString("FirstName", object.getString("firstName"));
...
```

```
editor.commit();
```

Excerto de Código 5.1: Armazenamento de Dados após o *Login*.

Quando, mais tarde, se pretender aceder a estas informações basta instanciar novamente a classe "*user.data*" e invocar o método *get* correspondente para cada dado. Isto foi aplicado, por exemplo, na atividade Editar Dados, e o excerto de código seguinte mostra como foi implementado:

```
SharedPreferences mySharedPreferences;
...
public void loadUserInfo() {
    mySharedPreferences = getSharedPreferences("user.data",
        Activity.MODE_PRIVATE);
    ...
    tv_nome = (TextView) findViewById(R.id.tv_nome);
    tv_nome.setText(mySharedPreferences.getString("FirstName", null) + " " + mySharedPreferences.getString("LastName", null));
    ...
}
```

Excerto de Código 5.2: Acesso à Informação Armazenada em Memória.

Por fim, é importante referir que todos os dados armazenados apenas são mantidos enquanto o utilizador se encontrar autenticado na aplicação. Quando este decidir terminar a sua sessão todos os dados são eliminados e o utilizador terá de se autenticar novamente, repetindo o processo de armazenamento de dados.

### 5.2.3 Troca de Dados entre a Aplicação e o Servidor HIAS

Um dos principais objetivos para esta aplicação é a implementação de um protocolo de comunicação com o servidor HIAS que permita a troca de informações entre os dois sistemas. Foi então necessário implementar *HyperText Transfer Protocol (HTTP) requests* para realizar pedidos de informação e enviar dados para serem armazenados na base de dados do serviço.

No contexto desta aplicação, foram implementados dois métodos diferentes:

- GET - este método é usado para resgatar dados do servidor e, nesta aplicação, foi utilizado na autenticação do utilizador;
- POST - este método é usado para inserir ou atualizar dados no servidor: Foi usado na edição de dados do utilizador e após o início da chamada de emergência.

De seguida são apresentados dois excertos de código, cada um com a implementação de um destes métodos.

O primeiro excerto mostra a implementação do método GET:

```
String url = "https://hiawebservice1.azurewebsites.net/
    ConfirmLogin/" + email.getText().toString() + "/" + passHex;
new getConfirmLogin(login.this).execute(url);
...
class getConfirmLogin extends AsyncTask<String, Void, String> {
    ...
    protected String doInBackground(String... connUrl){
        HttpURLConnection conn=null;
        ...
        try{
            final URL url = new URL(connUrl[0]);
            conn=(HttpURLConnection) url.openConnection();
            conn.setRequestProperty("Content-Type", "application
                /json; charset=utf-8");
            conn.setRequestMethod("GET");
            int result=conn.getResponseCode();
            if(result==200){
                ...
                while((line=reader.readLine())!=null){
                    status=line;
                }
            }
        }
        ...
    }
    protected void onPostExecute(String result){
        super.onPostExecute(result);
        if(result!=null){
            try{
                JSONArray jsonArray = new JSONArray(result);
                JSONObject object = jsonArray.getJSONObject(0);
                ...
            }
            ...
        }
    }
}
```

Excerto de Código 5.3: Implementação do Método GET na Aplicação Móvel.

Para implementar este método foi criada uma classe, dentro da qual se realizou a conexão e comunicação com o servidor. Em primeiro lugar instanciou-se o objeto *HTTPConnection* com o *Uniform Resource Locator* (URL) pré-definido, e o método GET. O passo seguinte foi declarar os cabeçalhos do pedido, onde foi definido que o formato a usar na transmissão de dados seria *JavaScript Object Notation* (JSON). De seguida, é lida a resposta devolvida pelo servidor, e por fim define-se a forma como serão tratados os dados recebidos. Para este caso, o conteúdo do objeto JSON é lido e guardado em memória, tal como descrito na

secção 5.2.2.

O segundo trecho de código mostra a implementação do método POST:

```
new postUpdateData().execute("https://hiawebservice1.
azurewebsites.net/UpdateData?Email=" + Email + "&PhoneNumber="
+ PhoneNumber + ... + "&Observations=" + Observations);
...
class postUpdateData extends AsyncTask<String, Void, String> {
    ...
    protected String doInBackground(String... connUrl){
        HttpURLConnection conn=null;
        BufferedReader reader;
        try{
            final URL url = new URL(connUrl[0]);
            conn=(HttpURLConnection) url.openConnection();
            conn.setDoOutput(true);
            conn.setDoInput(true);
            conn.setChunkedStreamingMode(0);
            conn.setRequestProperty("Content-Type", "
application/json; charset=utf-8");
            conn.setRequestMethod("POST");
            ...
        }
    }
    ...
    protected void onPostExecute(String result){
        super.onPostExecute(result);
        if(result!=null){
            Toast.makeText(editar_info.this, "Informacao
Atualizada", Toast.LENGTH_LONG).show();
        }
        ...
    }
    ...
}
```

Excerto de Código 5.4: Implementação do Método POST na Aplicação Móvel.

Neste método, os dados que se pretende transmitir são anexados ao URL usado na comunicação com o servidor. Assim sendo, para este método foi apenas necessário realizar a conexão com o servidor, instanciando o objeto *HTTPConnection* com o URL, onde estão contidos os dados, e o método POST. O formato JSON foi novamente definido para a troca de dados, no entanto o servidor, neste caso, apenas irá devolver uma *flag* com a confirmação do recebimento dos dados.

### 5.2.4 Chamada de Emergência

A funcionalidade que dá uma vantagem a esta aplicação em relação ao método tradicional é o facto de esta permitir que os serviços de emergência sejam contactados rapidamente enquanto que, ao mesmo tempo, os dados do utilizador são enviados para o operador, tal como é mostrado na figura 5.2. Isto dá ao operador um conhecimento básico sobre a situação e sobre o utilizador, à qual, numa situação tradicional, este apenas teria acesso após interrogar o utilizador.

Esta funcionalidade pode, então, ser ativada ao clicar no botão Chamada de Emergência, no menu principal da aplicação. Isto irá desencadear, em primeiro lugar, que a aplicação inicie uma chamada telefónica para o número de emergência pré-definido e, em segundo lugar, que esta faça um *request* para o servidor HIAS com as informações do utilizador. Estas informações são depois redirecionadas para o operador dos serviços de emergência, através da aplicação *web*. Para uma descrição mais detalhada desta transferência de dados deve ser consultada a secção 5.2.3.



Figura 5.2: Início da Chamada de Emergência



## 5.3 Manual de Utilizador

Ao abrir a aplicação pela primeira vez, o o utilizador deverá realizar a autenticação dos seus dados do HIAS. Esta autenticação será realizada na atividade mostrada na figura 5.3 e é pedido o *email* e a palavra-passe do utilizador.

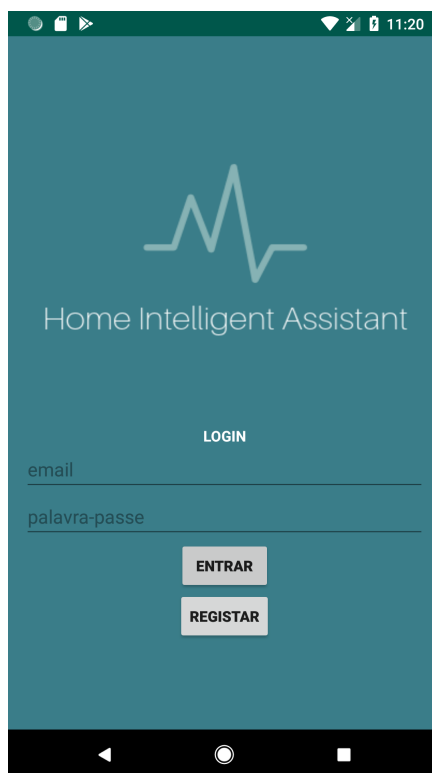


Figura 5.3: Atividade *Login* da Aplicação Móvel

Ao concluir o *login*, o utilizador é levado para a página principal da aplicação, representada na figura 5.4, a partir da qual poderá aceder a todas as funcionalidades disponíveis.

As atividades que podem ser acedidas a partir da página principal são as seguintes:

1. Editar dados

Ao seleccionar esta opção, o utilizador é levado para a página representada pela figura 5.5, onde lhe são apresentados todos os seus dados pessoais e de saúde a que este poderá realizar alterações. No final o utilizador poderá clicar em Guardar e as atualizações serão armazenadas.

## 2. Chamada de emergência

Com este botão é automaticamente iniciada uma chamada para o número de emergência definido e todas as outras operações são realizadas sem que o utilizador necessite de interagir com a aplicação.

## 3. Logout

Ao clicar neste botão, o utilizador é levado novamente para a página de *login* onde terá de realizar a autenticação novamente.

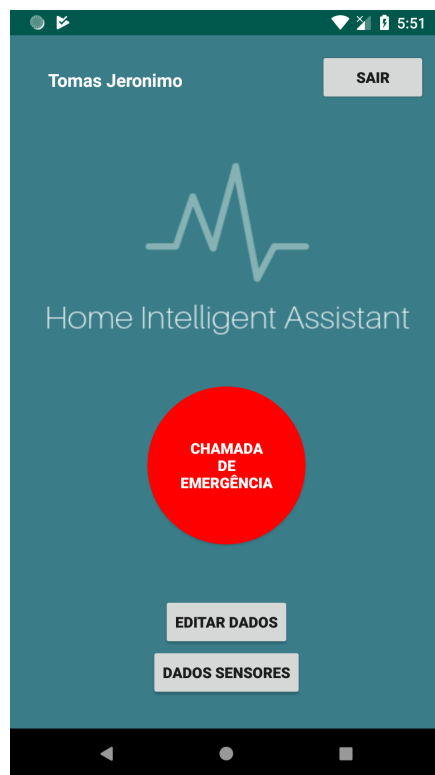


Figura 5.4: Atividade Principal da Aplicação Móvel

Tomas Jeronimo

Dados Pessoais

Número de telefone: 924221275

Morada : Travessa da Fonte

Dados de Saúde

Peso (cm) : 60

Altura (cm) : 180

Tipo Sanguíneo : A

Diabetes : No

Epilepsia : No

Asma : No

Figura 5.5: Atividade Editar Dados da Aplicação Móvel

## 5.4 Conclusões

A aplicação descrita neste capítulo representa o ponto de entrada de toda a informação utilizada neste projeto e é apresentada a forma como a comunicação com o servidor HIAS foi implementada, de forma permitir a transmissão desta informação para o resto do sistema. No capítulo seguinte, irá ser descrita a forma como esta informação é transmitida e tratada pela segunda aplicação desenvolvida no contexto deste trabalho.



## Capítulo 6

# Construção e Testes da Aplicação Web

### 6.1 Introdução

Neste capítulo é feita a descrição da aplicação *web*, desenvolvida para o uso dos operadores dos serviços de emergência, e é demonstrado como foram implementados os diversos requisitos descritos nos capítulos anteriores. Para tal, são explicados diversos detalhes da solução desenvolvida e é apresentado um manual de utilizador onde é mostrado como aceder e utilizar as funcionalidades implementadas. No fim do capítulo são apresentados os testes realizados ao algoritmo de processamento de voz desenvolvido, e são tiradas algumas conclusões a partir dos resultados obtidos.

### 6.2 Detalhes da Construção

#### 6.2.1 Troca de Dados entre Aplicação e Servidor HIAS

Após ter sido iniciada uma chamada de emergência, e os dados do utilizador terem sido recebidos no servidor HIAS, é necessário que exista uma forma de comunicação entre o servidor e a aplicação *web*, para que esses dados possam ser apresentados ao operador.

Para tal, tal como na aplicação móvel, foram estabelecidos *requests* HTTP aos quais a aplicação *web* recorrerá para requisitar estas informações.

Estes *requests* foram desenvolvidos com recurso a jQuery, uma biblioteca de funções Javascript com o objetivo de simplificar diversos *scripts*, entre os quais se incluem funções de comunicação entre cliente e servidor.

Foram então implementados os seguintes métodos:

- GET - método usado para resgatar dados do servidor, e no contexto da aplicação usado para receber os dados do utilizador sinistrado;
- POST - método usado para inserir ou atualizar dados, que nesta aplicação foi utilizado para enviar o formulário preenchido pelo operador no fim da chamada.

De seguida é apresentado um exemplo de como cada um destes métodos foi implementado, começando pelo método GET:

```
$.getJSON("/requestData/" + value, function answ(data) {  
    ...  
});
```

Excerto de Código 6.1: Implementação do Método GET na Aplicação Web.

Para implementar este método, usando a biblioteca jQuery, basta apenas fazer a chamada do método, com o prefixo "\$", onde é também declarado o formato a usar na transferência de dados. É depois necessário usar como parâmetros um URL pré-definido e uma função onde será depois tratada a resposta do servidor.

O método POST foi implementado como mostra o excerto seguinte:

```
$.post("/insereData/", {nomeUser: nomeUser}, ack, 'json');
```

Excerto de Código 6.2: Implementação do Método POST na Aplicação Web.

A chamada deste método é novamente feita usando o prefixo "\$", no entanto esta função carece de mais parâmetros como o URL, o objeto com as informações que se pretende transmitir, uma *flag* que verifica o recebimento, ou não, desta informação, e por fim o formato usado para a construção do objeto de dados.

## 6.2.2 Início da Chamada

Sendo o objetivo desta aplicação fornecer ao operador dos serviços de emergência o máximo de informação necessária para que este possa acionar os meios de socorro necessários, no menor tempo possível é necessário que, em primeiro lugar, a aplicação receba os dados do utilizador.

Para isso é feito um pedido ao servidor HIAS, o qual devolve todas as informações que tiver disponíveis. Este processo encontra-se descrito, em mais detalhe, na secção 6.2.5.

Com estas informações, a aplicação irá executar as seguintes funções:

- Apresentação dos dados pessoais (nome, número de telefone, idade e género) no canto superior esquerdo do ecrã;

- Apresentação dos dados sensoriais, imediatamente após os dados pessoais;
- Apresentação das coordenadas GPS do utilizador (e um mapa centrado nessas coordenadas) no canto inferior esquerdo do ecrã;
- Armazenamento de todas as informações em memória;
- Ativação do microfone e do algoritmo de processamento de voz (secção 6.2.3).

Os dados apresentados dão ao operador as informações básicas sobre o utilizador, existindo assim uma poupança de tempo que de outra forma seria gasto se o operador o tivesse que interrogar para as obter.

### 6.2.3 Processamento de Voz

Durante a chamada é necessário que a voz do operador seja processada de forma a detetar pedidos para novas informações. Para tal, foi utilizado uma *Application Programming Interface* (API) de *speech-to-text* para converter a voz do operador em texto que possa ser processado pela aplicação.

Na figura 6.1 é mostrado o processo de processamento da voz do operador, desde que esta é detetada até ser associada a uma das informações armazenadas.

Cada uma das frases de texto detetadas é dividida em palavras, e cada uma dessas palavras é comparada com um dicionário de palavras-chave previamente definidas. Cada palavra-chave encontra-se associada a um tipo de informação sobre o utilizador, armazenada em memória, podendo existir mais do que uma palavra-chave para cada informação.

Ao encontrar uma correspondência, a informação associada é apresentada ao operador, no lado direito do ecrã, e o processo é repetido até que todas as palavras tenham sido processadas ou até que o operador termine a chamada.

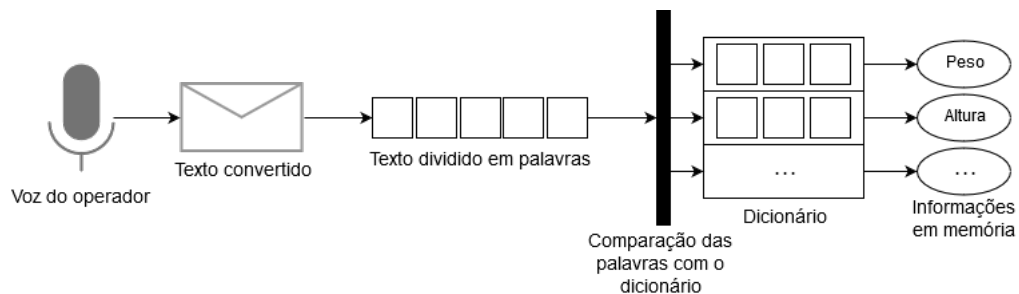


Figura 6.1: Processamento de Voz

### 6.2.4 Fim da chamada

Terminada a chamada, o operador terá que premir o botão no canto superior direito do ecrã, sendo-lhe apresentado um formulário, no qual são pedidas as seguintes informações:

- Nome do utilizador;
- Data;
- Nome do operador;
- Resumo da chamada.

Os primeiros dois dados são preenchidos automaticamente pela aplicação, devendo o operador preencher os últimos dois manualmente.

Por fim, o operador clica para enviar o relatório e a aplicação enviará os dados não só para o servidor HIAS mas também para o servidor MD.BARS. Mais detalhes sobre a comunicação entre a aplicação e estes servidores podem ser encontradas nas secções 6.2.1 e 6.2.5, respetivamente.

### 6.2.5 Troca de Dados entre Aplicação e Servidor MD.BARS

A comunicação entre estes dois sistemas acontece apenas uma vez, após o fim da chamada, e o objetivo é a transferência dos dados do formulário para o servidor MD.BARS. Tal como referido no capítulo 4.2 estes dados irão tornar-se anónimos, sendo por isso necessário que esse processo ocorra antes do envio dos dados para o servidor.

A figura 6.2 mostra o percurso que as informações irão percorrer antes de chegarem ao servidor, e tal como se pode ver, estas são enviadas, localmente, para um módulo do MD.BARS onde irá ocorrer o processo de anonimização. É de notar que o desenvolvimento deste módulo não fez parte dos objetivos do trabalho a que este documento se refere. No contexto deste trabalho, apenas foi necessário estabelecer um protocolo de comunicação entre a aplicação e este módulo. Isto foi conseguido realizando HTTP *requests*, da mesma forma que foi implementado o protocolo de comunicação com o servidor HIAS, referido no capítulo 6.2.1.

Basta apenas referir que, para além de anonimizados, os dados são também encriptados, no módulo MD.BARS, podendo finalmente ser enviados, de forma segura, para o servidor MD.BARS.



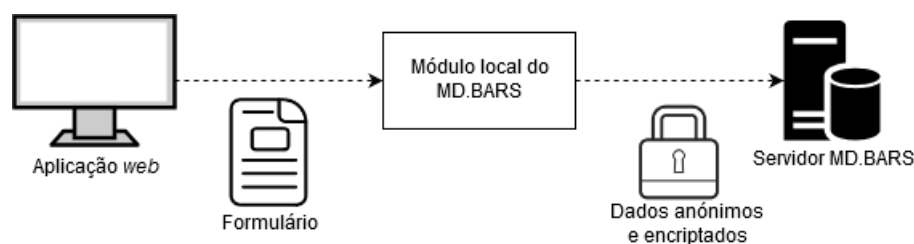


Figura 6.2: Comunicação entre Aplicação *Web* e Servidor MD.BARS

## 6.3 Manual de Utilizador

Ao ser iniciada, aplicação irá preencher o ecrã com os dados do utilizador que iniciou a chamada de emergência, dando ao operador uma ideia geral sobre o utilizador e a situação em que este se encontra. Na figura 6.3 é apresentado esse mesmo estado da aplicação.

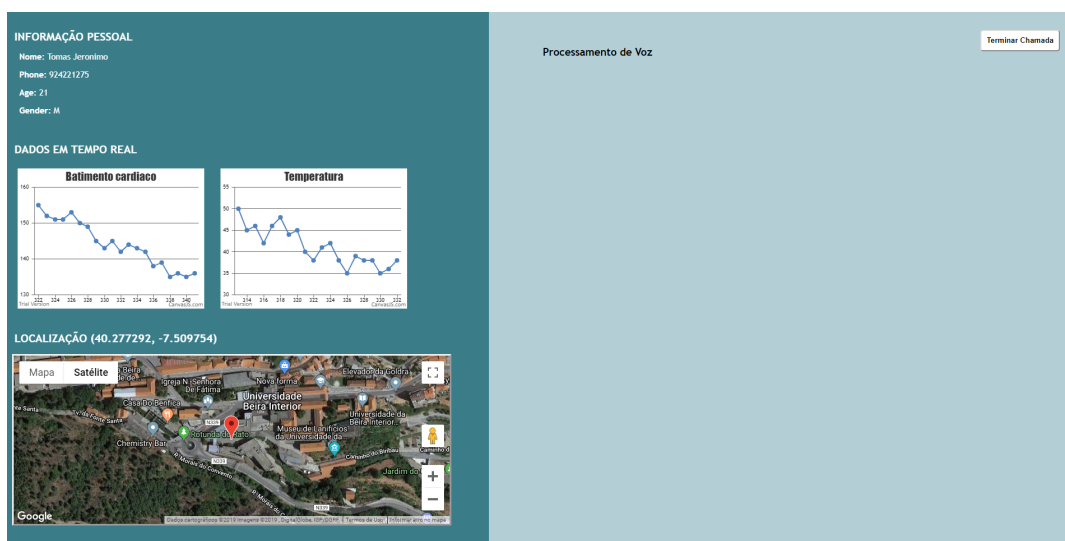


Figura 6.3: Aplicação *Web* após Início da Chamada

Durante a chamada, o processamento da voz do operador estará ativo, e a aplicação irá apresentar, no lado direito do ecrã, diversos dados sobre o utilizador, de forma a auxiliar o operador e responder aos seus pedidos. Exemplos de alguns tipos de informações que podem ser apresentadas durante o decorrer da chamada podem ser visualizados na figura 6.4.

No fim da chamada, o operador deverá clicar no botão, no topo direito do ecrã, para terminar a chamada, o que irá resultar na geração de um formulário, no qual o operador terá de preencher os dados em falta (nome do operador e

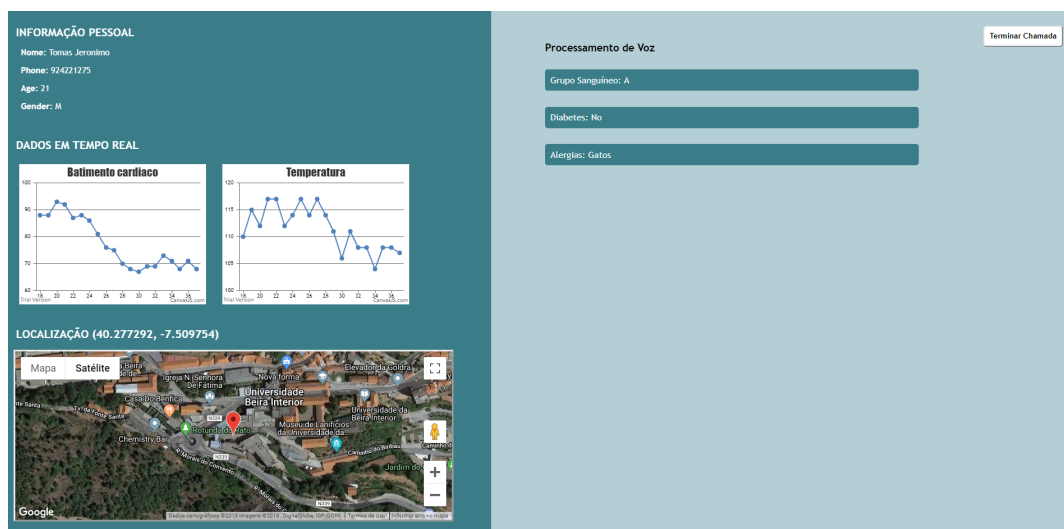


Figura 6.4: Apresentação de Dados do Utilizador a Pedido do Operador

sumário da chamada), tal como é mostrado na figura 6.5. Após terminar e clicar no botão Enviar, a aplicação volta ao seu estado inicial, pronta para receber uma nova chamada.

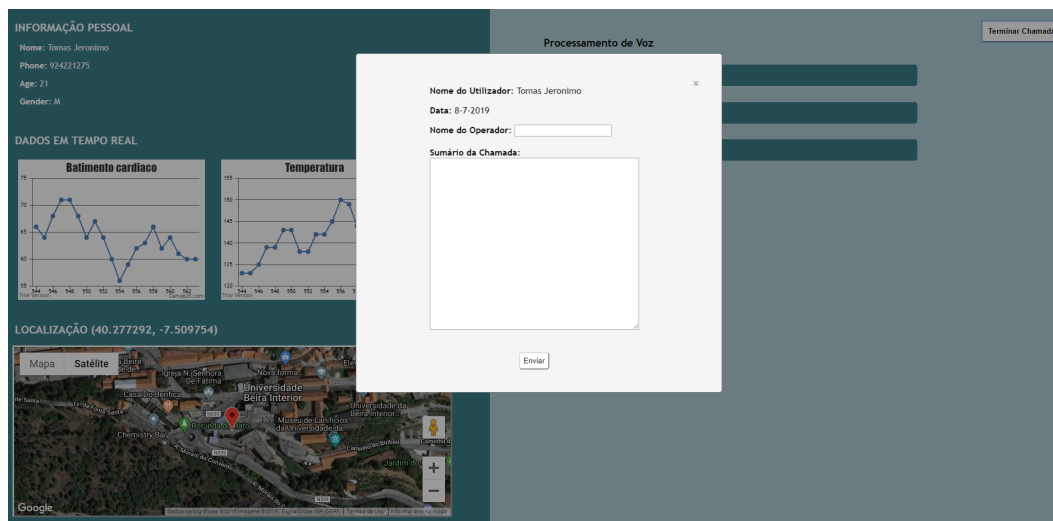


Figura 6.5: Formulário Gerado no Final da Chamada na Aplicação Web

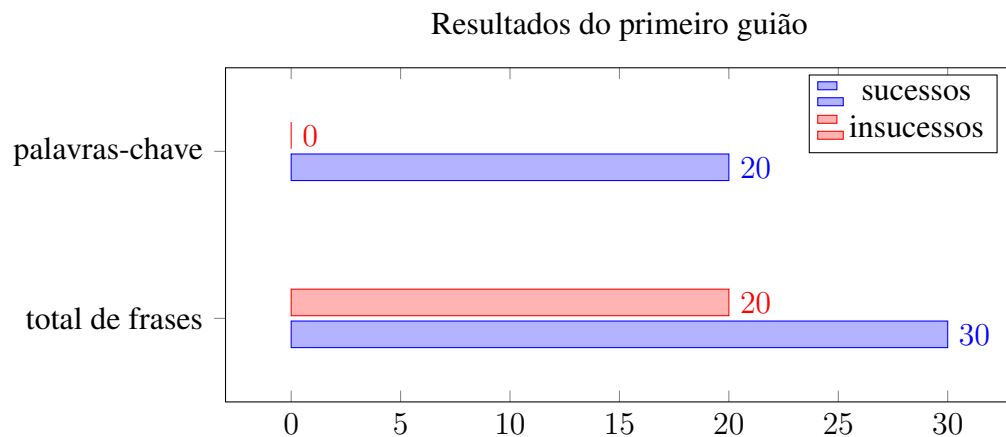
## 6.4 Testes e Validação do Reconhecimento de Voz da Aplicação Web

Nesta secção serão demonstrados os testes realizados para avaliar e validar o desempenho do reconhecimento de voz da aplicação *web*. Para tal foram escritos dois guiões, com o objetivo de simular o comportamento do operador em duas situações diferentes. Para cada guião procurou-se observar duas variáveis: o total de palavras-chave detetadas e o total de frases traduzidas sem qualquer erro. Foi feito um total de dez testes, para cada guião, e os resultados obtidos foram organizados em dois gráficos. De seguida é apresentado cada um dos guiões desenvolvidos, tendo sido sublinhadas as palavras-chave presentes em cada um.

Na primeira situação, pretende-se simular uma chamada em que o utilizador se encontra com dificuldades em respirar. O guião utilizado foi o seguinte:

*"Serviço de emergência, em que posso ajudar? Sente dificuldade em respirar? Tente colocar-se numa posição confortável. Tem algum tipo de alergias? Os serviços de emergência foram acionados para a sua localização."*

Os resultados observados para este guião encontram-se descritos no gráfico seguinte:



Os números apresentados no gráfico são o resultado de dez leituras do primeiro guião. Tal como é mostrado, o sistema foi capaz de identificar todas as palavras-chave lidas, em todos os testes, tendo depois apresentado a informação correspondente no ecrã e foi capaz de identificar, totalmente, 60% das frases lidas. Para além destes dados foi também possível identificar que a maioria dos erros ocorreu no início das frases, tal como demonstra a figura 6.6 que mostra os resultados

de uma das leituras realizadas. Isto leva a concluir que existirá um atraso entre o reconhecimento de voz e o início do seu processamento. Por outro lado, foi também verificado que a existência de ruído pode causar o sistema a ter dificuldades a reconhecer o fim do *input* de voz, levando a um atraso na apresentação de resultados.

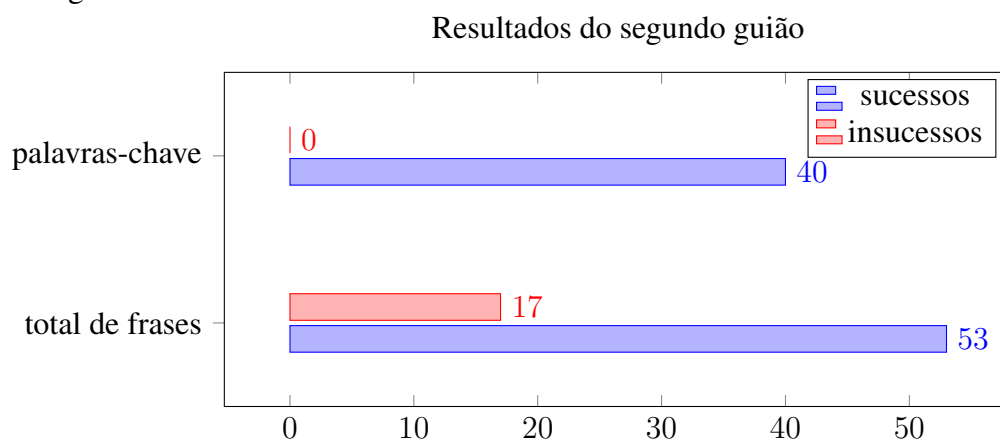
```
Speech Recognition Supported
TRANSCRIPT: serviço de emergência Em que posso ajudar
TRANSCRIPT: Centro dificuldade em respirar
TRANSCRIPT: tem que colocar numa posição confortável
TRANSCRIPT: tem algum tipo de alergias
TRANSCRIPT: os serviços de emergência foram acionados para a sua localização
```

Figura 6.6: Resultados de uma das Leituras Realizadas ao Primeiro Guião

Com o segundo guião, o objetivo foi simular uma situação onde um utilizador sente uma dor muscular, podendo esta estar ligada ao seu historial de diabetes. Este guião encontra-se a seguir:

*"Serviço de emergência, em que posso ajudar? A dor que sente é dor muscular? Sente fadiga ou cansaço? Tem historial de diabetes? Qual o seu tipo sanguíneo? Sabe indicar o seu peso e altura? Os serviços de emergência foram acionados para a sua localização."*

Os resultados observados nas leituras deste guião podem ser visualizados no seguinte gráfico:



Este gráfico representa os dados obtidos após dez leituras do segundo guião e, tal como observado no primeiro gráfico, também aqui o sistema foi capaz de

identificar todas as palavras-chave. Em relação ao total de frases, é mostrado que o sistema foi capaz de reconhecer, aproximadamente, 76% das frases lidas, sem qualquer erro. Mais uma vez foi observado que, tal como no caso anterior, a maioria dos erros de processamento ocorreu no início de cada frase. Na figura 6.7 é apresentado uma das leituras onde este caso se verifica.

Speech Recognition Supported
TRANSCRIPT: serviço emergência Em que posso ajudar
TRANSCRIPT: a dor que sentem muscular
TRANSCRIPT: sem fadiga ou cansaço
TRANSCRIPT: tem historial de diabetes
TRANSCRIPT: qual o seu tipo sanguíneo
TRANSCRIPT: Sabe indicar o seu peso e altura
TRANSCRIPT: o serviço emergência foram ativado espaço localização

Figura 6.7: Resultados de uma das Leituras Realizadas ao Segundo Guião

Podemos concluir que os resultados desta amostra são positivos, e demonstram a fiabilidade do processamento de voz implementado, no entanto, são também prova que existem pontos a ser melhorados, tais como, a diminuição do tempo necessário na transição entre o reconhecimento e o processamento de voz, e estudar soluções para diminuir a entrada de ruído ou separa-lo do *input* do operador.

## 6.5 Conclusões

Este capítulo representa o ponto do sistema onde os dados, recolhidos na aplicação móvel, são tratados e usados para consulta dos serviços de emergência. É mostrada a implementação da comunicação entre os diversos sistemas associados a este serviço e como os dados recolhidos no fim da chamada serão utilizados por esses mesmos sistemas. Por fim são mostrados os resultados dos testes realizados para avaliar o processamento de voz da aplicação e é feito um levantamento dos principais problemas encontrados. No próximo capítulo serão abordadas algumas soluções para estes problemas, e também algumas funcionalidades que estão planeadas para complementar o trabalho apresentado nesta secção.



# Capítulo 7

## Conclusões e Trabalho Futuro

Neste capítulo pretende-se apresentar uma síntese do projeto realizado, fazendo uma comparação entre os objetivos propostos e os objetivos alcançados, de forma a avaliar o trabalho realizado pelo autor.

Assim, a primeira secção irá focar-se nos objetivos do projeto. São apresentados os objetivos atingidos e a sua implementação, e são explicadas as razões pelas quais alguns dos objetivos não foram alcançados. Por fim, é feita uma revisão das tarefas estruturais deste projeto e do estudo realizado para cada uma.

Na última secção, são referidas as implementações que o autor estaria interessado em desenvolver no futuro, como novas funcionalidades ou a integração de novos sistemas.

### 7.1 Conclusões Principais

Este projeto tinha como objetivo principal a construção de um sistema IoT para cenários de emergências médicas, que inclui duas aplicações, em diferentes plataformas, móvel e *web*. A aplicação móvel, construída para o sistema operativo Android, deveria ser capaz de comunicar com o servidor HIAS e realizar troca de informações, receber e tratar dados de saúde em tempo real, inclusive provenientes de sensores incorporados em dispositivos móveis, por exemplo, um *smartwatch*. Deveria igualmente permitir a autenticação dos utilizadores previamente registados no HIAS. Para a aplicação *web*, o objetivo era que esta fosse capaz de realizar processamento de voz e fornecer dados do utilizador, ao operador, à medida que este os fosse requisitando. Para além disto, a aplicação teria de conseguir comunicar e trocar dados com o servidor HIAS e com o servidor MD.BARS.

Podemos concluir que o objetivo principal deste projeto, bem como todos os objetivos parciais, foram concluídos com sucesso. Foi realizada uma revisão so-

bre a computação móvel, mais especificamente sobre o desenvolvimento de aplicações Android, e foi também feito um estudo sobre a área da IoT, a área da eHealth, e sobre o desenvolvimento *web*, mais especificamente um estudo das tecnologias: JavaScript, NodeJS e RESTfull Web Services.

Em relação ao estudo da arquitetura do sistema, este foi feito visando a resolução dos problemas encontrados nos serviços de emergência estudados e tendo em conta a necessidade de integrar as aplicações desenvolvidas num serviço de casa inteligente previamente existente.

Foram então desenvolvidas as duas aplicações mencionadas. Para a aplicação móvel foi criado um formulário para recolher dados sobre o utilizador e foi desenvolvido um protocolo de comunicação com o servidor central do HIAS. Porém, devido a uma falha na requisição dos dispositivos que seriam usados para recolher os dados sensoriais, a recolha de dados vitais (batimento cardíaco e temperatura corporal) não foi realizada. Contudo, o sistema está perfeitamente preparado de forma modular para integrar os dispositivos e assim realizar a recolha e tratamento dos seus dados.

Em relação à aplicação *web*, foi desenvolvido um algoritmo para a comparação da voz do operador com um dicionário de palavras sensíveis, e foram estabelecidos protocolos de comunicação para a troca de dados com ambos os servidores, referidos em cima. Foram depois realizados testes de voz, de forma a avaliar a fiabilidade da solução construída, onde se verificaram resultados positivos e foram identificadas informações que poderão ser usadas para melhorar este processo, no futuro.

Por fim, foi realizado este documento de forma a especificar o trabalho desenvolvido na elaboração deste projeto, no qual foi incluído um manual de utilização para cada uma das aplicações desenvolvidas.

## 7.2 Trabalho Futuro

Em relação ao trabalho futuro, existem vários objetivos que o autor gostaria de ver implementados neste projeto.

Em baixo é apresentada uma lista, onde estes objetivos se encontram detalhados:

- criação de uma aplicação para *smartwatches*, capaz de recolher diferentes dados sensoriais do seu utilizador, e transferir esses dados para a aplicação móvel, onde estes poderiam ser visualizados e enviados para o servidor HIAS;
- expandir a aplicação Android a outras plataformas, nomeadamente *iPhone Operating System* (iOS) e *Windows Phone*;



- estabelecer uma comunicação com outros sistemas integrados no HIAS, e usar esses sistemas como novas fontes de informação;
- aprofundar os conhecimentos nas áreas de IA e *machine learning* e usar este estudo para melhorar o processamento de voz da aplicação *web*. O objetivo será criar um algoritmo capaz de se adaptar à voz do operador, e assimilar os seus pedidos de informação mais comuns, de forma a otimizar a apresentação de informação, para cada operador;
- realizar um teste piloto, simulando uma situação de emergência, real, e observar e documentar o desempenho do sistema.



# Bibliografia

- [1] Ethereum, 2019. [Online] <https://www.ethereum.org/>. Último acesso a 11 de junho de 2019.
- [2] F. Al-Turjman. *Artificial Intelligence in IoT*. ISBN: 9783030041090. 9783030041106. Springer Nature, 2019.
- [3] F. Andriopoulou, T. Orphanoudakis, and T. Dagiuklas. Iota: Iot automated sip-based emergency call triggering system for general ehealth purposes. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) Wireless and Mobile Computing, Networking and Communications (WiMob)*, 978-1-5386-3839-2, pages 362–369. IEEE, 2017.
- [4] Apple. Healthcare- Apple Watch - Apple, 2019. [Online] <https://www.apple.com/healthcare/apple-watch/>. Último acesso a 22 de junho de 2019.
- [5] A. Archip, N. Botezatu, E. Serban, P. C. Hergelegiu, and A. Zala. An iot based system for remote patient monitoring. In *17th International Carpathian Control Conference (ICCC)*, pages 1–6, 2016.
- [6] AREU. 112 Where ARE U, 2014. [Online] [https://play.google.com/store/apps/details?id=it.Beta80Group.whereareu&hl=pt\\_PT](https://play.google.com/store/apps/details?id=it.Beta80Group.whereareu&hl=pt_PT). Último acesso a 22 de junho de 2019.
- [7] CareAngel. Virtual Nurse Assitant, 2019. [Online] <https://www.careangel.com/>. Último acesso a 22 de junho de 2019.
- [8] R. Case, S. Cartledge, J. Siedenburger, K. Smith, L. Straney, B. Barger, J. Finn, and J.E. Bray. Identifying barriers to the provision of bystander cardiopulmonary resuscitation (cpr) in high-risk regions: a qualitative review of emergency calls. *Resuscitation*, 129(18):43–47, 2018.

- [9] Oracle Corporation. Obtenha Informações sobre a Tecnologia Java, 2019. [Online] [https://www.java.com/pt\\_BR/about/](https://www.java.com/pt_BR/about/). Último acesso a 05 de junho de 2019.
- [10] Portal de dados aberto da Administração Pública. Instituto Nacional de Emergência Médica (INEM), I.P, 2019. [Online] <https://dados.gov.pt/pt/organizations/inem/>. Último acesso a 02 de junho de 2019.
- [11] Serviço Nacional de Saúde. O que é o CODU?, 2017. [Online] <https://www.inem.pt/2017/05/26/o-que-e-o-codu/>. Último acesso a 02 de junho de 2019.
- [12] Serviço Nacional de Saúde. INEM reduz o tempo médio para atendimento nos CODU, 2018. [Online] <https://www.inem.pt/2018/05/08/inem-reduz-o-tempo-medio-para-atendimento-nos-codu/>. Último acesso a 02 de junho de 2019.
- [13] Serviço Nacional de Saúde. Evolução Diária das Chamadas de Emergência Atendidas no Centro de Orientação de Doentes Urgentes (CODU) , 2019. [Online] <https://bit.ly/2HSNgbv>. Último acesso a 02 de junho de 2019.
- [14] Android Developers. Conheça o Android Studio, 2019. [Online] <https://developer.android.com/studio/intro/>. Último acesso a 05 de junho de 2019.
- [15] Android Developers. SharedPreferences, 2019. [Online] <https://developer.android.com/reference/android/content/SharedPreferences>. Último acesso a 20 de junho de 2019.
- [16] J. W. Dexheimer and E. M. Borycki. Use of mobile devices in the emergency department: A scoping review. *Health Informatics Journal*, 21(4):306–315, 2015.
- [17] Sdongo E., Bolvinou A., Tsogas M., Amditis A., Guerra B., and Manso M. Next generation automated emergency calls. In *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2331-9860, pages 1–4. IEEE, 2017.
- [18] FitNow. Lose It!, 2016. [Online] [https://play.google.com/store/apps/details?id=com.fitnow.loseit&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.fitnow.loseit&hl=pt_PT). Último acesso a 22 de junho de 2019.
- [19] V. Gupta. A brief history of blockchain. *Harvard Business Review Digital Articles*, pages 2–4, 2017.

- [20] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. Kwak. The internet of things for health care: A comprehensive survey. *IEEE Access*, 3:678–708, 2015.
- [21] G Jayavardhana, B. Rajkumar, Slaven M., and Marimuthu P. Internet of things (iot): A vision, architectural elements, and future directions. *Elsevier*, 29(7):1645–1660, 2013.
- [22] C. K. Kao and D. M. Liebovitz. Consumer mobile health apps: Current state, barriers, and future directions. *Elsevier*, 9(5):106–115, 2017.
- [23] MacroPinch. Cardiógrafo-Cardiograph, 2011. [Online] [https://play.google.com/store/apps/details?id=com.macropinch.hydra.android&hl=pt\\_PT](https://play.google.com/store/apps/details?id=com.macropinch.hydra.android&hl=pt_PT). Último acesso a 22 de junho de 2019.
- [24] E. Mohn. Internet of Things. Salem Press Encyclopedia of Science, 2018. [Online] <https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,cookie,shib,uid&db=ers&AN=100558386&lang=pt-pt&site=eds-live&scope=site>. Último acesso a 04 de junho de 2019.
- [25] NodeJS. About Node.js, 2018. [Online] <https://nodejs.org/en/about/>. Último acesso a 06 de junho de 2019.
- [26] A. A. Osuwa, E. B. Ekhonoragbon, and L. T. Fat. Application of artificial intelligence in internet of things. In *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN) Computational Intelligence and Communication Networks (CICN), 2017 9th International Conference*, ISBN: 978-1-5090-5000-0, pages 169–173. IEEE, 2017.
- [27] The LaTeX Project. LaTeX - A document preparation system, 2019. [Online] <https://www.latex-project.org/>. Último acesso a 06 de junho de 2019.
- [28] P.P. Ray, D. Dash, and D. De. Edge computing for internet of things: A survey, e-healthcare case study and future direction. *Journal of Network and Computer Applications*, (140):1–22, 2019.
- [29] U. Satija, B. Ramkumar, and M. M. Sabarimalai. Real-time signal quality-aware ecg telemetry system for iot-based health care monitoring. *IEEE Internet of Things Journal*, 4(3):815–823, 2017.
- [30] B. M. C. Silva, J. J. P. C. Rodrigues, I. de la Torre Díez, M. López-Coronado, and K. Saleem. Mobile-health: A review of current state in 2015. *Elsevier*, 1(56):265–272, 2015.

- 
- [31] M. S. Uddin, J. B. Alam, and S. and Banu. Real time patient monitoring system based on internet of things. In *4th International Conference on Advances in Electrical Engineering (ICAEE)*, page 516. IEEE, 2017.
  - [32] J. Wan, M. A. A. H. Al-awlaqi, M. Li, M. O’Grady, X. Gu, J. Wang, and N. Cao. Wearable iot enabled real-time health monitoring system. *EURASIP JOURNAL ON WIRELESS COMMUNICATIONS AND NETWORKING*, 2018.
  - [33] M. Weinlich, P. Kurz, M. B. Blau, F. Walcher, and S. Piatek. Significant acceleration of emergency response using smart phone geolocation data and a worldwide emergency call support system. *PLoS ONE*, 13(5):1–10, 2018.
  - [34] A. Čolaković and M. Hadžialić. Internet of things (iot): A review of enabling technologies, challenges, and open research issues. *Computer Networks*, (144):1739, 2018.