



ANDRÉ FILIPE
BARRELA ALVES

**DESENVOLVIMENTO DE
APLICAÇÕES MÓVEIS PARA A
EMERGÊNCIA PRÉ-HOSPITALAR**

Tese de Projeto submetido como requisito parcial para obtenção do grau de **Mestre em Engenharia de Software**

ORIENTADORES

André Sabino

Professor Doutor, Universidade Europeia

João Ventura

Professor Doutor, Instituto Politécnico de Setúbal

JÚRI

Claúdio Sapateiro

Presidente: Professor Doutor, IPS

Armanda Rodrigues

Arguente: Professora Doutora, FHCT UNL

André Sabino

Orientador: Professor Doutor, Universidade Europeia

novembro, 2022

"Technology is best when it brings people together."

- Matt Mullenweg

Agradecimentos

Tendo em conta que é com os erros que mais aprendemos, quero começar por agradecer a todas as peripécias que me trouxeram a esta etapa. Foi sem dúvida uma fase bastante desafiante, e se o consegui fazer foi graças a todas as pessoas que no meu caminho se cruzaram.

Quero agradecer aos meus colegas de mestrado, em especial ao João Pereira, pela longa dedicação aplicada em todos os projetos. Pela exigência demonstrada e por alinhar sempre nas ideias às vezes mais diferenciadas e mais trabalhosas. Pela confiança aplicada nas situações mais stressantes e pela metodologia de trabalho que conseguimos atingir ao longo deste mestrado. Foi sem dúvida um excelente colega de trabalho com o qual espero ter o privilégio de vir a privar novamente. Não tenho qualquer dúvida que um dia vais mesmo construir um castelo com todas pedras que apareçam no teu caminho.

Um agradecimento especial ao professor André Sabino que foi uma peça fundamental para a conclusão deste projeto e que me manteve sempre focado nos objetivos. Obrigado pela metodologia de trabalho recorrente aplicada e pelo conhecimento partilhado. Tornou este processo mais prático e objetivo, algo que na minha ótica fez toda a diferença para colmatar algumas das minhas lacunas.

Obrigado aos meus colegas de trabalho que me aturaram com um pouco menos de energia resultante das longas noites de trabalho. Agradeço também aos meus amigos que não me incentivaram a sair durante a altura em que mais necessitava de ficar dedicado a este projeto.

À minha família por nunca duvidarem das minhas capacidades mesmo sem percecionarem as vantagens de continuar a amplificar os meus conhecimentos académicos, e em especial ao meu irmão de quem uso como força de vontade para continuar a concluir todas as etapas na minha vida. É para que tu vejas que nada nesta vida é inalcançável quando existe vontade de o fazer. Podemos fazer tudo e tudo podemos fazer, só precisamos de umas *Monster* e umas longas horas extra de trabalho.

A todos, um **MUITO OBRIGADO**

Resumo

Durante um processo de pedido de ajuda para uma emergência pré-hospitalar é necessário a realização de uma chamada para o número europeu de emergência médica (112). Em situações de stress esta chamada pode ser bastante ineficiente devido às dificuldades de transmissão de informação verbal entre quem realiza a chamada e o operador.

Uma dificuldade de comunicação durante uma chamada de emergência resulta muitas vezes numa espera prolongada pelos meios de socorro e pode, em alguns casos, ser crucial para o estado vital das vítimas. Mesmo que toda a informação seja transmitida com sucesso, existe por vezes ainda uma dificuldade por parte dos técnicos de emergência pré-hospitalar em identificar o local exato onde as vítimas se encontram com base em apenas uma morada.

Aplicou-se um processo ágil de engenharia de *software* com vista à instanciação de uma aplicação móvel que tem como principal objetivo encaminhar toda a informação pré-preenchida das vítimas, e outras informações acerca da ocorrência, como a localização do próprio dispositivo móvel. A utilização da aplicação deverá permitir, não só facilitar a comunicação durante o pedido de auxílio, como também otimizar o processo de emergências médicas possibilitando um sistema de acompanhamento de informações de cada vítima e ocorrência.

Foi realizado um estudo de validação com uma amostra representativa da sociedade foi possível verificar uma elevada taxa de aceitação para a integração desta aplicação móvel num contexto real. Os utilizadores inquiridos revelaram ter interesse em incluir este tipo de tecnologia no seu quotidiano, destacando a importância do impacto que determinada ferramenta poderia ter na salvaguarda de vidas.

Palavras-chave: Emergência Pré-Hospitalar, Arquitetura Orientada a Serviços, Aplicação Móvel, Desenvolvimento Ágil

Abstract

During a request for help process for a pre-hospital emergency, a call to the European emergency medical number (112) is necessary. In stressful situations this call can be quite inefficient due to the difficulties of verbal information transmission between the caller and the operator.

Communication difficulties during an emergency call often result in a prolonged wait for rescue services and can, in some cases, be crucial to the vital status of victims. Even if all information is successfully transmitted, there is sometimes still a difficulty for pre-hospital emergency responders to identify the exact location of victims based on a single address.

An agile software engineering process was applied for the instantiation of a mobile application whose main goal is to forward all pre-completed information of the victims, and other information about the occurrence, such as the location of the mobile device itself. The use of the application should allow not only to facilitate communication during the request for help, but also to optimize the process of medical emergencies by enabling a system for monitoring the information of each victim and occurrence.

A validation study was conducted with a representative sample of society and it was possible to verify a high rate of acceptance for the integration of this mobile application in a real context. The surveyed users showed interest in including this type of technology in their daily lives, highlighting the importance of the impact that this tool could have on saving lives.

Keywords: Pre-Hospital Emergency, Service Oriented Architecture, Mobile Application, Agile Development

Conteúdo

Agradecimentos	v
Resumo	vii
Abstract	ix
Lista de Tabelas	xv
Lista de Figuras	xvii
Lista de Siglas e Acrónimos	xix
1 Introdução	1
1.1 Motivação	1
1.2 Problema	3
1.3 Objetivos Gerais e Resultados Esperados	5
1.4 Metodologia utilizada	5
1.5 Estrutura do Documento	6
2 Estado da Arte	9
2.1 Trabalhos relacionados	9
2.2 Discussão	13
3 Planeamento e Estudo de Soluções Tecnológicas	15
3.1 Planeamento Geral	15
3.2 Estudo de Soluções Tecnológicas	16
3.2.1 <i>REST API</i>	16
3.2.2 <i>Django Rest Framework</i>	18
3.2.3 <i>PostgreSQL</i>	19
3.2.4 <i>MySQL</i>	19
3.2.5 <i>PWA</i>	19
3.2.6 <i>Outsystems Platform</i>	20
3.3 Discussões	22
3.3.1 Planeamento	22
3.3.2 Soluções Tecnológicas	23

4	Análise, Desenho e Implementação	25
4.1	Análise e Definição de Requisitos	25
4.1.1	Descrição Geral do Sistema	26
4.1.1.1	Descrição Geral do <i>Software</i>	27
4.1.1.2	Funcionalidades do <i>Software</i>	27
4.1.1.3	Caraterísticas dos Utilizadores	28
4.1.1.4	Dependências e Pressupostos	28
4.1.2	Especificação de Requisitos	29
4.1.2.1	Requisitos Funcionais	29
4.1.2.2	Requisitos Não Funcionais	31
4.1.3	Discussão	33
4.2	Desenho de <i>Software</i>	33
4.2.1	Associação de Requisitos a <i>Wireframes</i>	34
4.2.1.1	RF1 - Restrição de Uso	34
4.2.1.2	RF2 - Preenchimento da informação clínica do utilizador	35
4.2.1.3	RF3 - Registo de outros perfis	35
4.2.1.4	RF4 - Registo de uma ocorrência	35
4.2.1.5	RF5 - Seleção da tipologia da emergência	36
4.2.1.6	RF6 - Seleção da(s) vítima(s) registadas	36
4.2.1.7	RF8 - Detecção da localização da ocorrência	37
4.2.1.8	RF9 - Realização de chamadas	37
4.2.1.9	RF10 - Listagem de Ocorrências	37
4.2.2	Discussão	37
4.3	Arquitetura do Sistema	38
4.3.1	Arquitetura SIREPH	38
4.3.1.1	Configuração do servidor	39
4.3.1.2	Modelo de Dados SIREPH	40
4.3.1.3	Aplicação <i>Django</i>	43
4.3.2	Arquitetura <i>SmartCall</i>	44
4.3.2.1	SmartCall Integration Service	44
4.3.2.2	SmartCall Mobile Core Service	45
4.3.2.3	SmartCall Mobile Business Logic	49
4.3.2.4	SmartCall User Interface	49
4.4	Desenvolvimento Aplicaional	50
4.4.1	Ecrã Inicial	50
4.4.2	Tutorial	52
4.4.3	<i>Login</i>	52
4.4.4	Registo do utilizador	52
4.4.5	Notícias	52

4.4.6	Perfis	52
4.4.7	Listagem de Ocorrências	53
4.4.8	Criação de Ocorrência	53
4.5	Discussão	55
5	Validação	57
5.1	Descrição de Protocolos	57
5.1.1	Guião de Validação Aplicacional	58
5.1.2	Questionário de Validação Aplicacional	58
5.2	Análise de Resultados	59
5.2.1	Caraterização demográfica	60
5.2.2	Resultados obtidos	62
5.2.3	Discussão	65
6	Conclusões	67
6.1	Cumprimento de Requisitos	67
6.2	Processo de Engenharia de Software	69
6.3	Trabalho Futuro	70
	Bibliografia	73
	Anexos	79
I	Verbetes Nacional de Socorro	79
	Apêndices	81
A	Wireframes da aplicação	81
A.1	Wireframes do tutorial da aplicação	81
A.2	Wireframe de <i>log in</i> da aplicação	82
A.3	Wireframes do processo de registo do Utilizador	82
A.4	Wireframe do ecrã inicial	84
A.5	Wireframes do processo de criação de perfis	84
A.6	Wireframes da listagem de Ocorrências	85
A.7	Wireframes do processo de criação de uma Ocorrência	85
B	Modelo de Dados do sistema SIREPH	87
C	Modelo de Dados do Armazenamento Local	89
D	Ecrãs da aplicação	91
D.1	Ecrã do tutorial da aplicação	91
D.2	Ecrã de <i>login</i> da aplicação	92

D.3	Ecrãs do processo de registo do Utilizador	93
D.4	Ecrã de notícias	95
D.5	Ecrãs do processo de criação de perfis	96
D.6	Ecrãs da listagem de Ocorrências	97
D.7	Ecrãs do processo de criação de uma Ocorrência	98
E	Guião Aplicacional de Testes	101
F	Minuta de Declaração de Consentimento	103
G	Questionário de Validação Aplicacional	105

Lista de Tabelas

1.1	Indicadores de desempenho do INEM em 2021	4
3.1	Tabela de convenção de nomenclatura de módulos <i>Outsystems</i>	22
4.1	Requisito Funcional 1 - Restrição de Uso.	29
4.2	Requisito Funcional 2 - Preenchimento da informação clínica do utilizador padrão.	29
4.3	Requisito Funcional 3 - Registo de outros perfis.	30
4.4	Requisito Funcional 4 - Registo de uma ocorrência.	30
4.5	Requisito Funcional 5 - Seleção da tipologia da emergência.	30
4.6	Requisito Funcional 6 - Seleção da(s) vítima(s) registadas.	30
4.7	Requisito Funcional 7 - Seleção da(s) vítima(s) não registadas.	31
4.8	Requisito Funcional 8 - Detecção da localização da ocorrência.	31
4.9	Requisito Funcional 9 - Realização de chamadas.	31
4.10	Requisito Funcional 10 - Listagem de Ocorrências.	31
4.11	Requisito Não Funcional 1 - Responsividade.	31
4.12	Requisito Não Funcional 2 - Usabilidade.	32
4.13	Requisito Não Funcional 3 - Disponibilidade.	32
4.14	Requisito Não Funcional 4 - Segurança.	32
4.15	Requisito Não Funcional 5 - Integrabilidade.	32
4.16	Requisito Não Funcional 6 - Integridade.	32
4.17	Requisito Não Funcional 7 - Reatividade.	32
4.18	Correlação de requisitos com <i>wireframes</i>	34
5.1	Resultados das confirmações de sucesso de ações de teste.	62
5.2	Mediana das respostas de usabilidade associada a cada funcionalidade.	64
6.1	Grau de satisfação de requisitos funcionais.	67
6.2	Grau de satisfação de requisitos não funcionais.	69

Lista de Figuras

1.1	Processo do SIEM.	2
2.1	Esquema arquitetural <i>Sahana</i>	12
3.1	Cronograma do Projeto Proposto.	16
3.2	Estrutura de um pedido <i>HTTP</i>	17
3.3	Esquema Arquitetural <i>Outsystems</i>	21
3.4	Cronograma do Projeto Real.	23
4.1	Esquema Geral de Integração de aplicações com o sistema SIREPH.	27
4.2	Processo de criação de uma ocorrência na aplicação.	36
4.3	Diagrama Arquitetural do sistema.	39
4.4	Esquema de funcionamento do <i>Nginx</i> e <i>Gunicorn</i> com <i>Django</i>	40
4.5	Modelo de Dados SIREPH simplificado.	41
4.6	Modelo Arquitetural <i>Outsystems</i> da aplicação <i>SmartCall</i>	44
4.7	Modelo de dados do utilizador na aplicação.	46
4.8	Modelo de dados da tipologia de ocorrências na aplicação.	48
4.9	Comparação entre <i>wireframe</i> e materialização do ecrã inicial da aplicação.	51
4.10	Processo de criação de uma ocorrência atualizado.	54
5.1	Género dos utilizadores de teste.	60
5.2	Grupo Etário dos utilizadores de teste.	60
5.3	Grau académico dos utilizadores de teste.	61
5.4	Setor Profissional dos utilizadores de teste.	61
5.5	Usabilidade associada às funcionalidades da aplicação <i>SmartCall</i>	63
5.6	Integrabilidade da aplicação <i>SmartCall</i> em contexto real.	64

Lista de Siglas e Acrónimos

ANTEPH Associação Nacional de Técnicos de Emergência Pré-Hospitalar.

API Application Programming Interface.

AVC Acidente Vascular Cerebral.

BD Base de Dados.

BI Bilhete de Identidade.

CC Cartão de Cidadão.

CI/CD Continuous Integration/Continuous Delivery.

CIAV Centro de Informação Antivenenos.

CODU Centros de Orientação de Doentes Urgentes.

CPS Child Protection System.

DRF Django Rest Framework.

EIS Enterprise Information System.

GPS Global Positioning System.

GUID Globally Unique Identifier.

HTML HyperText Markup Language.

HTTP Hypertext Transfer Protocol.

HTTPS Hyper Text Transfer Protocol Secure.

IGAS Inspeção Geral das Atividades em Saúde.

INEM Instituto Nacional de Emergência Médica.

ISO International Organization for Standardization.

JSON JavaScript Object Notation.

MPR Missing Persons Registry.

PLS Posição Lateral de Segurança.

PSP Polícia de Segurança Pública.

PWA Progressive Web App.

REST Representational State Transfer.

RPC Remote Procedure Call.

SAV Suporte Avançado de Vida.

SBV Suporte Básico de Vida.

SGBD Sistema de Gestão de Base de Dados.

SI Sistemas de Informação.

SIEM Sistema Integrado de Emergência Médica.

SII Sistema de Informação Integrado.

SIREPH Sistema Integrado de Resposta à Emergência Pré-Hospitalar.

SIRESP Sistema Integrado de Redes de Emergência e Segurança de Portugal.

SNS Serviço Nacional de Saúde.

SNS24 Centro de Contacto do Serviço Nacional de Saúde.

SOA Service-oriented architecture.

SOAP Simple Object Access Protocol.

SQL Structured Query Language.

SR Shelter Registry.

TEPH Técnicos de Emergência Pré-Hospitalar.

TETRICOSY Telephonic Triage and Counseling System.

TI Tecnologias da Informação.

UAT Testes de Aceitação de Utilizadores.

UI User Interface.

UML Unified Modeling Language.

VCS Volunteer Coordination System.

WSGI Web Server Gateway Interface.

XML Extensible Markup Language.

Capítulo 1

Introdução

Com a atual pandemia a atingir um estado de estagnação, as consequências originadas pela mesma começam a ser alvo de avaliações com o objetivo de melhorar os sistemas e processos afetos. Segundo o Ministério da Saúde [1], o impacto económico identificado no Serviço Nacional de Saúde (SNS) é facilmente justificável pela subida de gastos com pessoal, aquisição de equipamentos e fornecimentos externos. Além das medidas de prevenção extraordinárias tomadas por parte do SNS para fazer face à pandemia, emergências pré-hospitalares não relacionadas com a mesma não deixaram de existir, o que originou um sobrecarregamento nos sistemas envolvidos na gestão do sistema de resposta a emergências pré-hospitalares.

Com vista a caraterizar a rede de sistemas e processos envolvidos no SNS, a Inspeção Geral das Atividades em Saúde (IGAS)¹ realizou um conjunto de auditorias [2] onde, para além de outras conclusões, foi possível verificar que os mecanismos de atuação em rede e todos os processos associados aos mesmos necessitavam de ser atualizados de forma a tornar os mesmos mais eficientes.

Através de uma análise rápida de alguns dos processos do SNS relacionados com as emergências pré-hospitalares, é possível verificar que para que estes se tornem mais eficientes, uma simples atualização não seria suficiente, dado que os diferentes Sistemas de Informação (SI) envolvidos não efetuam troca de informação entre eles.

1.1 Motivação

Atualmente, numa situação de emergência, é despoletado o Sistema Integrado de Emergência Médica (SIEM), apresentado na Figura 1.1, que é responsável por ativar e coordenar todas as ações necessárias para o socorro a vítimas. Este processo inicia-se com uma chamada telefónica para o número europeu de emergência (112) que será recebida por um operador da Central de Emergência da Polícia de Segurança Pública (PSP) e, se a situação em questão se justificar, transferida para um dos Centros de Orientação de Doentes Urgentes (CODU) do Instituto Nacional de Emergência Médica (INEM). Estas centrais são responsáveis pela triagem, pela ativação dos

¹ Entidade responsável pelo controlo de todas as instituições, públicas ou privadas, relacionadas com a área da Saúde.

respetivos meios de pré-socorro e até pela ativação de auxílio indireto em situações mais críticas. Se os meios de pré-socorro a serem ativados pertencerem ao INEM é utilizado um sistema interno, conhecido como Sistema Integrado de Redes de Emergência e Segurança de Portugal (SIRESP)² para comunicar à corporação de bombeiros a informação necessária acerca da ocorrência. Por outro lado, se os meios de pré-socorro não pertencerem ao INEM, o corpo de bombeiros é informado da ocorrência através de uma chamada telefónica. Quando a informação da ocorrência chega aos Técnicos de Emergência Pré-Hospitalar (TEPH), o socorro é realizado assim que estes se encontrem perto da vítima, sendo esta posteriormente transportada assim que possível para uma unidade de saúde onde receberá os tratamentos de saúde adequados à sua situação.

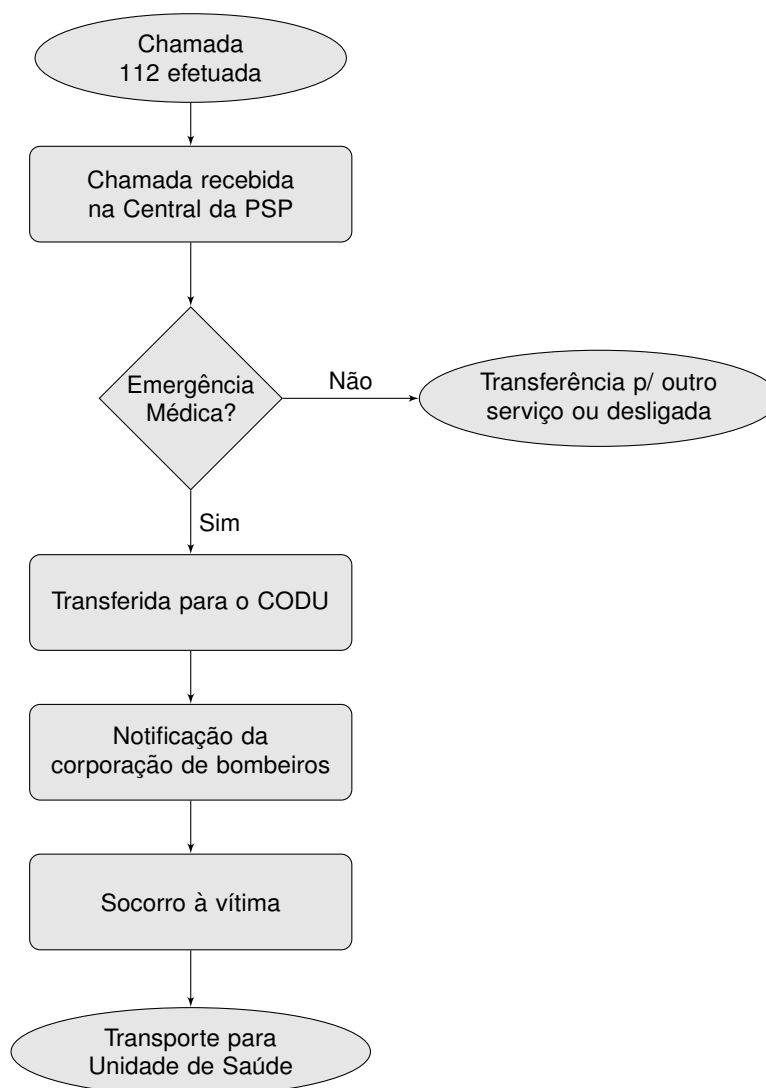


Figura 1.1: Processo do SIEM.

Em situações de emergência, é fundamental minimizar o tempo de resposta dos meios de socorro. Segundo Massano [3], em Portugal, mais de 20.6 mil habitantes por ano sofrem de Acidente Vascular Cerebral (AVC). Esta é a segunda maior causa de morte no país e obriga a que o tempo de atuação dos meios de socorro seja o mínimo possível de forma a diminuir as sequelas

²Rede de comunicações exclusiva do Estado Português para o comando, controlo e coordenação de comunicações em todas as situações de emergência e segurança.

para a vítima. Para que se consiga reduzir este tempo de resposta, segundo o SNS [4], é então importante que seja fornecido, durante a chamada com o CODU, de forma simples e clara as seguintes informações com a seguinte ordem:

1. A localização exata da emergência, com pontos de referência;
2. O número do contacto que está a efetuar a chamada;
3. O número de vítimas;
4. O sexo e idade aproximada das vítimas;
5. Uma descrição da situação destacando os detalhes pela sua criticidade.

Apesar de existirem muitos processos identificados que poderão ser melhorados no que envolve a emergência pré-hospitalar e os tempos de resposta associados à mesma, o investimento dedicado a estes não reflete essa preocupação. Prova disto é o facto do orçamento de estado previsto alocado aos bombeiros, para o ano de 2022, ser de 29 milhões [5] quando, segundo Jaime Marta Soares³, “Só dois corpos de bombeiros profissionais custam 65 milhões de euros a duas câmaras municipais” [6].

De forma a colmatar esta falta de investimento, a Associação Nacional de Técnicos de Emergência Pré-Hospitalar (ANTEPH) existe com o objetivo de regulamentar e defender os direitos dos TEPH que, pela natureza do seu trabalho, desempenham atividades de desgaste rápido com salários baixos e sem subsídios de risco. Para além deste objetivo, a ANTEPH recomenda ainda a existência de Centrais Unificadas de Emergência, com o objetivo de facilitar toda a comunicação e transmissão de informação entre os diferentes intervenientes (Forças de Segurança, INEM, Bombeiros, Cruz Vermelha Portuguesa e Proteção Civil) no processo de uma ocorrência ([7]).

1.2 Problema

Para além da falta de comunicação da informação entre sistemas identificada pela ANTEPH, durante o processo de uma ocorrência os TEPH são obrigados ao preenchimento de um verbete com a informação acerca da mesma. A utilização deste verbete deveria possibilitar a comunicação e validação de todos os procedimentos que foram realizados ou administrados à(s) vítima(s), de forma a que os restantes intervenientes conseguissem perceber rapidamente todo o histórico da ocorrência. Com este sistema analógico, durante uma ocorrência, nem sempre existe a disponibilidade dos TEPH para preencher toda a informação necessária dado que a prioridade dos mesmos está focada no socorro à(s) vítima(s). Nestas situações, a transmissão de informação sobre a situação acontece na maior parte das vezes de forma verbal com os restantes meios de emergência médica hospitalar.

³Ex-presidente da Câmara de Vila Nova de Poiares.

A falta de informação acerca das ocorrências de emergências pré-hospitalares constitui ainda outro desafio. Considerando a visão de Lawton [8], num contexto organizacional, a possibilidade de uma empresa conseguir recolher, armazenar, aceder e analisar a informação gerada pelo seu negócio pode permitir que sejam tomadas melhores decisões. O facto de não existir a possibilidade das entidades envolvidas nos processos de emergência pré-hospitalar conseguirem realizar qualquer uma destas atividades sobre a informação que poderá ser gerada, faz com que as decisões relativas à melhoria de processos não tenham uma análise profunda associada aos mesmos. De acordo com Mehrotra et al. [9], por omissão, esta situação ocorre porque a informação gerada nestes processos tem um elevado carácter de diversidade e origens diferentes, que provoca que a mesma esteja dispersa por hierarquias de armazenamento, comunicação e unidades de processamento diferentes.

Apesar de ainda não ser possível gerir toda a informação envolvida nos processos de emergência pré-hospitalar, o INEM conseguiu armazenar alguma informação referente a chamadas efetuadas. A Tabela 1.1, construída com base em indicadores de desempenho partilhados pelo próprio Instituto, permite concluir que ainda existe espaço para melhorar a gestão destas chamadas de emergência. Em 2021, mais de 90 mil chamadas efetuadas foram reencaminhadas para o Centro de Contacto do Serviço Nacional de Saúde (SNS24), dado que não foram consideradas como uma situação de emergência. Este mesmo valor corresponde a cerca de 250 chamadas diárias atendidas, e que estão a impedir que o sistema cumpra o seu propósito de socorrer todos os indivíduos em situações de emergência médica. Esta situação repete-se ainda para outros contactos nacionais de emergência que não são tão conhecidos pela população, como é o caso do Centro de Informação Antivenenos (CIAV).

Tabela 1.1: Indicadores de desempenho do INEM em 2021. Fonte: INEM [10].

Mês	Chamadas Atendidas	Acionamentos	Chamadas transferidas SNS24
Janeiro	126.883	114.448	9.756
Fevereiro	85.011	79.056	5.761
Março	94.614	89.609	6.091
Abril	99.143	92.640	6.742
Maio	114.454	106.412	7.377
Junho	115.242	106.334	7.843
Julho	124.610	113.452	8.971
Agosto	126.515	115.273	9.109
Setembro	116.490	106.885	7.680
Outubro	122.751	112.881	7.130
Novembro	122.122	112.950	7.539
Dezembro	123.840	113.429	8.949
Total	1.371.675	1.263.369	92.948

1.3 Objetivos Gerais e Resultados Esperados

Considerando os problemas apresentados na secção anterior e a proposta da ANTEPH para a criação de Centrais Unificadas de Emergência, este projeto surge com o objetivo de participar num Sistema Integrado de Resposta à Emergência Pré-Hospitalar (SIREPH) que possibilitará a todos os envolvidos nestes processos de emergência, recolher, armazenar, aceder e analisar a informação de forma a que seja possível:

1. Possibilitar a transmissão de informação pelos diferentes intervenientes;
2. Melhorar o processo de tomada de decisões;
3. Melhorar tempos de resposta e a operacionalidade do próprio sistema.

O SIREPH é constituído por diversas aplicações que, de forma individual, pretendem satisfazer os objetivos referidos anteriormente. Desta forma, o foco deste documento está na demonstração da potencialidade de implementação de uma ferramenta *mobile* capaz de resolver os défices referidos na secção 1.2, mais concretamente, no que diz respeito às chamadas que são realizadas para o número europeu de emergência (112) e na diminuição do tempo de transmissão de toda a informação necessária apresentada em 1.1.

Com a implementação desta ferramenta é esperado que o número de chamadas consideradas como falsas emergências (i.e., situações cuja gravidade e sensibilidade ao tempo não configuram uma emergência médica) reduza através de uma triagem realizada na própria aplicação, consoante a respetiva situação. Deverá ainda ser possível através desta solução, transmitir para o SIREPH toda a informação relevante para a abertura de uma ocorrência no sistema, sendo ela:

1. A localização exata da ocorrência com base nos serviços de localização do *smartphone*;
2. O número de contacto do dispositivo que está a efetuar o processo;
3. A informação do perfil das vítimas (quando possível);
4. O tipo de emergência a ocorrer.

A informação acima referida deverá ser facultada ao utilizador de forma simples e rápida aquando da instalação da aplicação. Após o envio da mesma, a aplicação deverá efetuar a chamada para o serviço (canal) a que mais se adequa a emergência para que o número de chamadas transferidas diminua e o tempo de acionamento dos meios de socorro necessários sejam o mais curto possível.

1.4 Metodologia utilizada

Segundo Patil et al. [11], as práticas tradicionais de gestão e desenvolvimento de projetos não se adequam no desenvolvimento de aplicações móveis. Esta incompatibilidade é destacada não só

pelo facto de existir uma elevada diversidade de dispositivos, mas também pela necessidade da conectividade de rede e outras especificações dos próprios *smartphones*. A metodologia *Waterfall*, conhecida como uma das mais tradicionais, obriga a que os requisitos do sistema sejam definidos logo no começo. Esta condição torna o modelo muito pouco flexível sempre que seja necessário efetuar uma alteração ao mesmo.

Dado que é comum no desenvolvimento de aplicações móveis existirem alterações recorrentes aos requisitos do próprio sistema, ou até ao plano do próprio, considerou-se que a metodologia mais adequada para estes é uma metodologia *Agile*. Esta metodologia promove uma iteração contínua entre partes interessadas, no desenvolvimento e testes durante todo o ciclo de vida do projeto [12], oferecendo robustez face a eventuais alterações de requisitos.

Como um dos casos de maior sucesso de desenvolvimento de aplicações móveis, o *Spotify*, aplicação de *streaming* de música, recorreu ao uso de uma metodologia *Agile* que lhes permitiu conceber uma arquitetura para manter diversos sistemas independentes e desacoplados entre eles, permitindo assim que as diferentes equipas se concentrassem no seu sistema ao mesmo tempo que estimulou a utilização de um modelo interno de *open source* de forma a facilitar o esclarecimento de dúvidas entre equipas.

Para esta realidade foi utilizado a metodologia *Kanban*. Este método, utilizado para definir, gerir e otimizar o trabalho a executar, permite maximizar a eficiência através da visualização de todas as tarefas a executar [13]. Consiste, fundamentalmente, num quadro organizado em três colunas onde se distribuem as tarefas relacionadas com o projeto. A primeira coluna é responsável pela alocação de todas as tarefas que ainda estão por desenvolver. Quando uma tarefa está em execução, essa é transferida para a segunda tabela que permite visualizar todas as tarefas que estão em progresso. Ao terminar uma tarefa, esta deverá ser colocada na terceira coluna onde são persistidas todas as tarefas já concluídas.

Em relação ao presente projeto, esta metodologia faz sentido não só pela fácil prosperidade à mudança das aplicações móveis, mas também devido à composição do SIREPH por diversos sistemas que requer uma comunicação entre os mesmos. Desta forma, foi construído o cronograma apresentado no Subcapítulo 3.1 que representa o planeamento do projeto descrito neste relatório.

1.5 Estrutura do Documento

Este documento está ainda estruturado com os seguintes capítulos:

- Capítulo 2 (Estado da Arte) - aprofunda alguns trabalhos relacionados com a temática explorada no presente documento;
- Capítulo 3 (Planeamento e Estudo de Soluções Tecnológicas) - inclui o processo de planeamento e análise de tecnologias que possibilitaram a construção da solução que visa cumprir com os objetivos propostos;
- Capítulo 4 (Implementação) - consiste nos detalhes técnicos acerca do desenvolvimento da

solução anteriormente desenhada e planeada, desde o processo de desenho da mesma até à sua materialização;

- Capítulo 5 (Validação) - apresenta e analisa os resultados obtidos através dos testes de aceitação da ferramenta construída e os protocolos utilizados para a execução dos mesmos;
- Capítulo 6 (Conclusões) - resumo de conclusões obtidas referenciando os objetivos cumpridos e o trabalho a realizar no futuro.

Capítulo 2

Estado da Arte

Neste capítulo serão abordados alguns trabalhos relacionados com a temática onde se insere o projeto do presente documento. Será ainda discutido a relevância que esses trabalhos relacionados produziram para o desenvolvimento do projeto desenvolvido.

2.1 Trabalhos relacionados

A intervenção de pessoas externas às situações de emergência é tradicionalmente encarada como um problema ético na gestão da própria ocorrência, devido ao facto da maioria dos indivíduos não possuir conhecimento relacionado com técnicas de Suporte Básico de Vida (SBV) [14]. Por outro lado, um estudo elaborado por Al-Akkad e Zimmermann reporta que a maior parte dos inquiridos, com e sem formação de SBV, estariam dispostos a contribuir para o processo de gestão de crises desde artefactos tecnológicos apropriados lhes fossem fornecidos [15]. Considerando que a tecnologia disponível nos *smartphones* apresenta um enorme potencial e que, segundo o Grupo Marktest, a mesma está à disponibilização de 89.9% da população nacional [16], o desenvolvimento de aplicações móveis para apoiar a gestão de pré-emergência hospital tem potencial.

O número de utilizadores atuais de *smartphones* em Portugal está projetado para atingir quase 7.5 milhões no ano de 2022 [17]. Dado que a população residente com mais de 60 anos corresponde a cerca de 3 milhões de habitantes [18] e que a população jovem é cada vez mais propícia a possuir conhecimentos tecnológicos básicos, é possível concluir que o número de utilizadores de *smartphones* continuará a crescer durante algum tempo [17].

Al-Akkad e Zimmermann afirmam que com a quantidade de sensores que são hoje introduzidos nos *smartphones* (Global Positioning System (GPS), bússola, câmara fotográfica, entre outros) existe uma enorme quantidade de dados que podem ser recolhidos em diversas situações e que podem ser bastante benéficos dependendo da forma como sejam utilizados [15]. Como se trata de uma infraestrutura já implementada com base em redes móveis, o esforço para criar qualquer tipo de aplicação é minimizado e possibilita diversas formas de exploração. A localização do dispositivo,

mapeada para um determinado instante de tempo, constitui informação tão relevante como a identificação do próprio utilizador [19]. Ao recorrer à utilização de sensores de redes móveis em situações de emergência é possível fornecer cobertura em áreas de difícil implementação e manutenção mesmo em situações adversas.

Em 2009, Wirz et al. realizaram um estudo de aceitação de sistema móvel que se propunha a apoiar eventos públicos de larga escala em situações de emergência. Com base nas respostas de 92 participantes num evento público, foi verificado que os dispositivos móveis são o meio preferido enquanto assistentes móveis de emergência [14]. Os autores deste estudo defendem ainda que para além dos engenheiros de telecomunicações e dos TEPH, a utilização de *smartphones* em emergências obriga a intervenção ainda de psicólogos para estudar a forma como os indivíduos se comportam com estes dispositivos nestas situações. Apesar da maioria dos inquiridos concordarem com a intervenção ativa de civis em situações de emergência através do uso de *smartphones*, existe ainda quem desaconselha fortemente a integração de indivíduos sem antecedentes de primeiros socorros, em casos de emergência. Ao confrontar estes indivíduos com a aplicação móvel, os mesmos referiram que a utilizariam, salientado que a utilização da aplicação necessitaria de ser simples, de forma a que fossem capazes de executar tarefas o mais rapidamente possível. Foi ainda mencionado que haveriam certas condições que deveriam ser garantidas, como a privacidade dos dados e o custo associado à utilização da própria aplicação.

Jayaraman et al. realizaram outro estudo com base num sistema com uma arquitetura orientada a serviços (*Service-oriented architecture (SOA)*), com o objetivo de partilhar e visualizar eficientemente dados de triagem, e possibilitar uma melhor tomada de decisões para os gestores das entidades envolvidas em processos de emergência [20]. Utilizando a classificação da criticidade de uma determinada situação de emergência, um bom sistema para lidar com a gestão destas situações requer bom desempenho da equipa técnica e dos seus dispositivos móveis, principalmente se o tratamento for fornecido longe de unidades hospitalares. Aplicar medidas eficazes e rápidas para chegar a tempo ao doente e/ou à unidade de saúde pode ajudar a reduzir problemas graves, e consequentemente, melhorar as hipóteses de recuperação da vítima.

Jayaraman et al. referem ainda que os processos de apoio à decisão neste tipo de sistemas são desafiantes, devido à complexidade das tarefas abordadas, tempo de resposta imprevisível, georreferenciação e necessidade de processamento de grandes quantidades de informação clínica num curto espaço de tempo. Atualmente, os serviços baseados em funcionalidades como a localização estão a convergir para uma melhoria da eficácia dos sistemas omnipresentes, que fornecem serviços instantâneos e fiáveis para o fornecimento de dados [20]. Este tipo de sistemas, conhecido como computação ubíqua consiste na integração de computação em objetos utilizados no quotidiano sem que sejam percecionados pelo utilizador [21]. No estudo relatado Jayaraman et al. indicam que o objetivo destes sistemas é estimular a transmissão entre as diferentes entidades médicas, melhorar as práticas utilizadas no setor da saúde e compreender e interpretar os mecanismos de raciocínio médico.

A utilização de ontologias¹ na gestão de emergências pode resultar num benefício significativo devido ao facto de as situações de emergência poderem envolver diferentes entidades e implicar vários processos que podem complicar o processo de tomada de decisão. Jayaraman et al. relatam a elaboração de uma ontologia que se destina a descrever as ações a considerar consoante o tipo de emergência. Esta ontologia foi utilizada como base para construir uma aplicação móvel denominada *iTriage* cuja fase de validações permitiu validar a sua capacidade de calcular com precisão as prioridades dos pacientes com base na triagem elaborada na ontologia anteriormente referida [20]. Com as validações realizadas na aplicação, Jayaraman et al. concluíram que a informação requisitada pela aplicação ao utilizador chega ao centro de gestão de emergências com uma maior precisão, resultando assim numa gestão mais eficiente destas situações.

De forma geral, os sistemas de resposta a emergências médicas podem ser divididos em dois tipos [22]:

- Sistema de alerta de emergência para alvos ativos - Os indivíduos em situações de emergência partilham a informação associada à sua localização para os TEPH. Um exemplo desta tipologia acontece quando uma pessoa utiliza o *GPS* para transmitir a sua localização por um determinado sistema;
- Sistema de salvamento passivo - Os TEPH aguardam passivamente por um sinal para iniciar um salvamento como acontece atualmente com os serviços de emergência públicos.

Os sistemas de salvamento mencionados implicam a utilização de informações de localização pessoal, que podem ser divulgadas a terceiros, comprometendo assim a privacidade do utilizador. Liutkauskas et al. sugeriram uma encriptação utilizando identificadores de tamanho constante [23], como o *Globally Unique Identifier (GUID)* que é definido como um número de referência único que identifica uma informação concreta num sistema. Errapotu et al. referem um modelo que pretende preservar a privacidade de um utilizador durante um processo de emergência iniciado com a transmissão da localização do próprio [24]. O modelo utiliza um conjunto de esquemas de autenticação com vista a proteger a privacidade da informação garantido que a chave utilizada apenas seja interpretada pelos sistemas que estão a efetuar a comunicação. Outra das opções mencionadas neste estudo é o armazenamento temporário da localização do utilizador, porém, como o tempo de resposta poderá ser imprevisível, não é possível concluir um período de tempo válido para o tempo de vida desta informação.

Em 2016 a cidade de Davao, nas Filipinas, teve pela primeira vez implementada uma linha centralizada dedicada à gestão de pedidos de emergências. De acordo com Edillo et al., esta linha servia o propósito de gerir apenas as emergências médicas urgentes, à semelhança do que acontece em Portugal com o 112. Um dos principais problemas detetados com esta implementação foi a dificuldade verificada em obter a localização exata das vítimas [25]. Este estudo propõe então o desenvolvimento de uma aplicação que funcionaria em conjunto com o sistema implementado,

¹Descrição que permite estudar conceções da realidade e relações numa área temática, bem como regras de combinação de conceitos e relações para definir extensões ao vocabulário.

de forma a conseguir captar a localização exata das vítimas sem que existisse a necessidade de solicitar a mesma.

Edillo et al. descrevem o desenvolvimento de uma aplicação denominada *i911* para dispositivos *Android* que, para além de partilhar a localização do dispositivo, permitia ainda efetuar uma triagem face à tipologia da ocorrência observada. Esta triagem servia maioritariamente para selecionar o serviço para o qual o utilizador deveria realizar a chamada de forma a reduzir o tempo de reposta dos meios de socorro. Ao efetuar um teste sobre a aplicação desenvolvida, 52% dos utilizadores identificaram a sua satisfação com a utilização da aplicação móvel para a obtenção de ajuda médica. Adicionalmente, mais de 60% dos inquiridos afirmaram que existia interesse da parte dos próprios em integrar mais funcionalidades, como a utilização de imagens ou voz na aplicação. Edillo et al. concluíram que a integração da aplicação com o sistema anteriormente implementado foi um sucesso, porém seria necessário a realização de testes mais exigentes de forma a determinar a existência de alguma fraqueza no sistema desenvolvido.

Um dos casos que ficou conhecido pela forma como foi desenvolvido foi o projeto *Sahana*. Este projeto consistiu no desenvolvimento de uma plataforma de gestão de informação gerada durante eventos de emergências de larga escala, com o objetivo a curto prazo de colmatar a difícil agregação de informação gerada durante as ações de salvamento devido à incidência de um tsunami no Sri Lanka em 2004 [26]. O que tornou este projeto um caso de estudo tão bem sucedido foi o facto do mesmo ter sido desenvolvido no curto espaço de tempo de 2/3 semanas e que sem este *software* teria sido impossível uma coordenação de esforços de forma a minimizar o impacto nos indivíduos afetados [27]. Careem et al. defendem que a facilidade de construção de todo o sistema envolvido neste projeto é justificada pela utilização de uma arquitetura modular. Na Figura 2.1 é possível verificar os módulos envolvidos no sistema construído.

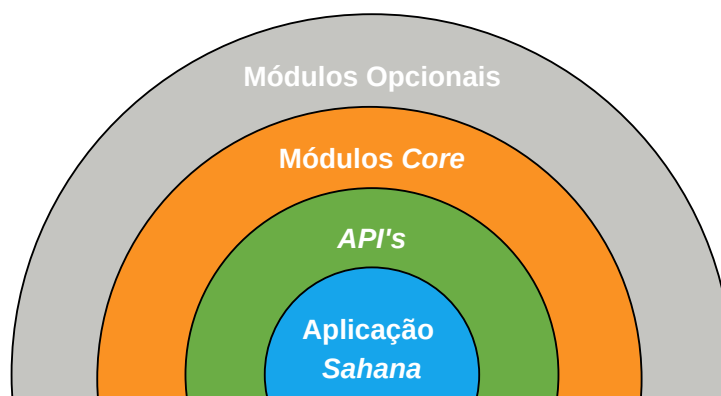


Figura 2.1: Esquema arquitetural *Sahana*.

Como é possível verificar no esquema acima a aplicação *Sahana* é consumida por bibliotecas e serviços externos que permitem de certa forma a extensão das capacidades da própria, como é o caso de uma *API* de localização que permitia a seleção e armazenamento de localizações de forma hierárquica. Os módulos *Core* são responsáveis pela principal gestão de serviços como o registo de pessoas desaparecidas (*Missing Persons Registry (MPR)*) e de acampamentos

montados (*Shelter Registry (SR)*). Já os módulos opcionais foram responsáveis por funcionalidades extras que também beneficiaram todo o sistema, como é o caso do sistema de coordenação de voluntários (*Volunteer Coordination System (VCS)*) ou do sistema de proteção de crianças (*Child Protection System (CPS)*).

Careem et al. concluem então que um sistema de partilha de informação para diferentes organizações durante esta tipologia de desastres é bastante vantajosa dado a necessidade de coordenação de inúmeras entidades.

2.2 Discussão

Com base na revisão literária elaborada na secção anterior, foi possível concluir que vários aspetos dos procedimentos relacionados com a pré-emergência hospitalar podem ser melhorados com a integração de tecnologia móvel, atualmente com elevada adoção pela sociedade.

Existe então espaço para aproveitar os recursos dos dispositivos móveis de cada indivíduo, de forma a considerar o tipo de envolvimento que o mesmo teria durante o processo de gestão de crises. Torna-se assim possível envolver qualquer indivíduo no processo de emergência pré-hospitalar, mesmo em casos onde não existe formação em suporte básico de vida, ficando a utilização circunscrita ao pedido de auxílio.

A utilização de dispositivos permite ainda a captação de diversos dados que poderão ser relevantes em situações de emergências sem que o próprio utilizador intervenha na mesma. A informação sobre a localização é um exemplo prático do tipo de dados que poderá ser obtido através de um sensor de um dispositivo móvel e que poderá ajudar a otimização dos processos envolvidos.

Através da informação obtida pelos inquiridos dos diversos estudos, foi possível verificar que as aplicações afetas à área de emergências médicas requerem uma utilização simples e prática mesmo em situações de stress.

Dos estudos observados verificou-se que a utilização de um SOA pode possibilitar uma integração mais facilitada com outros serviços, garantindo ainda alguma segurança a nível de autenticação de forma a proteger a informação comunicada. Adicionalmente foi possível verificar que as aplicações que estavam integradas num sistema conseguiram obter uma taxa de aceitação bastante positiva.

Capítulo 3

Planeamento e Estudo de Soluções Tecnológicas

Este capítulo irá abordar todo o planeamento envolvido no presente projeto, detalhando as fases mais relevantes do mesmo, e as tecnologias escolhidas para o seu desenvolvimento durante a fase de Análise do mesmo.

3.1 Planeamento Geral

Uma prática bastante comum na gestão de projetos de Tecnologias da Informação (TI) é a criação de um cronograma que representa todas as fases de um projeto. A utilização desta ferramenta permite não só uma maior visibilidade das datas-limite de cada tarefa assim como a duração das mesmas [28].

A primeira fase do projeto correspondeu ao desenvolvimento do cronograma de planeamento apresentado na Figura 3.1. Com este cronograma foi possível organizar as diferentes fases do projeto e planejar antecipadamente a data associada aos entregáveis do mesmo. A duração do projeto foi planificada com uma duração de 11 meses, considerando este planeamento o início do mesmo.

A fase de Levantamento de Requisitos foi planeada para decorrer durante o período de 2 meses. Desta fase resultaria um documento, inserido no subcapítulo 4.1, com a descrição das principais funcionalidades da aplicação a desenvolver e a relevância da implementação das mesmas.

Para a fase de desenvolvimento do âmbito do projeto estava planificado a execução do enquadramento introdutório à temática, tal como a definição do problema, e o desenvolvimento do Estado da Arte. Nesta fase foi ainda agendada uma entrega intermédia que tinha como objetivo a apresentação da temática analisada no projeto presente.

Na fase de Desenvolvimento foram planeadas duas tarefas. O Desenho e Análise do Sistema e o Desenvolvimento da Aplicação que só poderia ser iniciado com a conclusão do Desenho e

Análise do Sistema. Esta fase foi planeada com uma duração de 4 meses.

Para a fase de Validação foram reservados 2 meses que coincidiram com o término do projeto. Nesta fase o objetivo seria disponibilizar a um grupo restrito de utilizadores e avaliar a experiência dos mesmos de forma a produzir resultados qualitativos para uma futura análise e discussão de resultados.

Após todas as fases anteriores estarem concluídas, o projeto é dado como concluído aquando da conclusão do presente relatório, estando a entrega planificada para o final do mês de outubro.

Fases	Meses										
	2021	2022									
	Dezembro	Janeiro	Fevereiro	Março	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro
1 - Planeamento											
2- Levantamento de Requisitos											
3 - Desenvolvimento do Âmbito											
Enquadramento Introdutório											
Entrega Intermédia			◆								
Estado da Arte											
4- Desenvolvimento											
Desenho do Sistema											
Desenvolvimento Aplicacional											
5- Fase de Testes											
Testes de Aceitação											
Análise de Resultados											
6 - Entrega Final											◆

Figura 3.1: Cronograma do Projeto Proposto.

3.2 Estudo de Soluções Tecnológicas

Durante o processo de planeamento de um projeto aplicacional outras etapas importantes são a análise de diferentes soluções tecnológicas e a avaliação de alternativas [29], dado que permitem perceber qual a melhor opção que se adequa ao problema apresentado. Assim sendo, neste sub-capítulo será abordado um contexto introdutório as tecnologias escolhidas para os diferentes subsistemas envolvidos no desenvolvimento do presente projeto.

3.2.1 REST API

Como referido no subcapítulo 1.2, parte do problema identificado só seria possível resolver com um Sistema de Informação Integrado (SII) que possibilitaria a partilha de informação pelos diferentes intervenientes. Uma das formas mais seguras para garantir a comunicação de uma aplicação com outros softwares distintos é a utilização de uma *Representational State Transfer (REST) Application Programming Interface (API)* [30]. Também conhecida como *RESTful API* esta tipologia de arquitetura de software utiliza *requests*¹ *Hypertext Transfer Protocol (HTTP)* para

¹Mensagem enviada por um determinado cliente a um determinado servidor.

aceder e manipular a informação de outros serviços externos ao software em questão. Utilizando estes *requests* é possível recorrer aos seguintes métodos para efetuar as respetivas ações [31]:

- Método *GET* - Obter informação;
- Método *POST* - Criar ou atualizar recursos;
- Método *PUT* - Tem o mesmo efeito que o *POST*, porém este é idempotente²;
- Método *DELETE* - Eliminar informação.

Relativamente a questões de segurança, as *REST API* possuem meios que permitem controlar que softwares têm acesso às mesmas. Este controlo é efetuado através de *Basic HTTP Authentication* o qual é enviado no *header* do *request* de forma codificada em Base64³. O texto enviado nestas condições normalmente corresponde às credenciais de um utilizador, porém poderá ser necessário encaminhar um *token* caso seja necessário efetuar múltiplas chamadas ao serviço em questão [31]. Na figura 3.2 é possível observar a estrutura de um *request HTTP* com os diferentes elementos acima referidos.

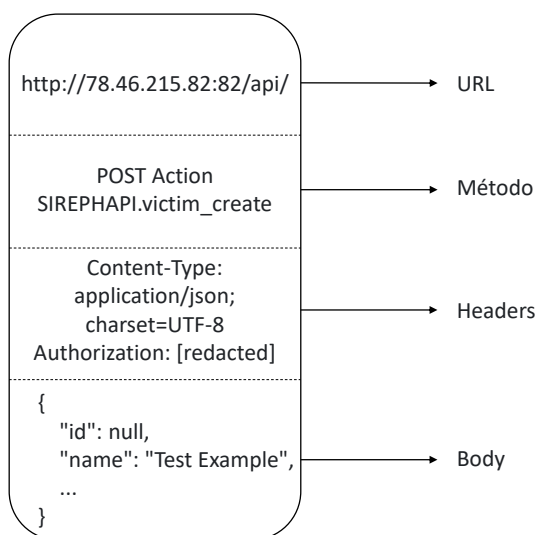


Figura 3.2: Estrutura de um pedido *HTTP*.

Uma das maiores dificuldades encontradas na utilização deste tipo arquitetural diz respeito à validação do mesmo. Para a realização de testes de uma *REST API* é necessário uma configuração inicial e verificações de *inputs* e *outputs* de cada *request* enviado [30]. De forma a colmatar estas necessidades a utilização de uma ferramenta como o *Swagger* pode trazer benefícios como [32]:

- Geração automática de documentação da *API*;
- Validação de funcionalidades durante o desenvolvimento.

²Que tem a propriedade de poder ser aplicado mais do que uma vez, sem que o resultado se altere.

³Codificação frequentemente utilizada para converter texto em um código com 64 caracteres.

Para além do *REST* existe ainda outro método de acesso a serviços que consiste num protocolo de comunicação de troca de informação com base em *Extensible Markup Language (XML)*. Este método é denominado de *Simple Object Access Protocol (SOAP)* e requer um documento *XML* com um *Remote Procedure Call (RPC)*, que inclui um conjunto padrão de regras para definir as mensagens, ligações, operações e a localização do próprio serviço [33]. Em contrapartida, para serviços *REST* não existe um conjunto padrão de regras para descrever a sua interface, pelo que é necessário que os utilizadores compreendam o contexto e o conteúdo a ser transmitido pelo próprio. Os serviços *REST* são ainda recomendados pelas seguintes características [30]:

- Facilidade de integração com outras aplicações;
- Utilidade para sistemas que incluem um controlo de acesso restrito, como é o caso de dispositivos móveis.

3.2.2 *Django Rest Framework*

De forma a otimizar o desenvolvimento de *REST API's* surgem bibliotecas como *Django Rest Framework (DRF)* que têm como objetivo disponibilizar um conjunto de ferramentas simples e personalizáveis baseadas em tecnologias como *Python* e *Django* [34].

Relativamente à sua estrutura geral, a *DRF* é composta por:

- *Models* que representam de forma genérica a tabela associada a um determinado objeto que é, normalmente, comunicado em *JavaScript Object Notation (JSON)* ou *XML*;
- *Serializers* que são responsáveis por efetuar o mapeamento entre as *views* e os *models*;
- *Views* que garantem o tratamento dos *requests* enviados para os diferentes *endpoints* da *API* de forma a que os mesmos obtenham uma resposta.

Para além da facilidade associada à implementação da *DRF*, outras vantagens encontradas que dificilmente se encontra noutras *frameworks*, são a possibilidade de customização de *url's* para os *endpoints* da *API* e um *API browser* que permite testar a própria *API* sem que seja necessário qualquer configuração extra [34].

No que diz respeito a políticas de segurança, uma das funcionalidades mais utilizadas em *API's* é a associação de credenciais de um determinado utilizador ou sistema, tal como referido no subcapítulo anterior. Por essa razão, a *framework* em análise possibilita a utilização de *packages* como o *OAuth* [35] que é responsável pelas validações de controlo de acesso à *API* ou o *authtoken* que possibilita a associação de um valor único associado a cada utilizador criado que será validado aquando da receção dos *requests*.

Outra das particularidades da utilização de *DRF* é que as aplicações desenvolvidas em *Django* utilizam um protocolo que as permite comunicar diretamente com *web servers* a partir de uma linguagem padrão. A essa comunicação padrão é dado o nome de *Web Server Gateway Interface (WSGI)* e é persistida num ficheiro gerado automaticamente denominado de *wsgi.py*.

3.2.3 *PostgreSQL*

Sempre que um software produza ou tenha a necessidade de manipular informação, torna-se necessário a utilização de um Sistema de Gestão de Base de Dados (SGBD). Estes sistemas têm como principal objetivo retirar a responsabilidade da gestão de Base de Dados (BD) dos softwares a que a elas acedam.

PostgreSQL é um SGBD *open source* que disponibiliza suporte tanto para BD relacionais como não relacionais, como é o caso de informação transacionada em *json* [36]. Parte do suporte dado por este tipo de SGBD foca-se em garantir a segurança da informação através de um controlo de acesso de permissões, possibilitar um conjunto de controlos para efetuar alterações da própria informação e gerir a performance da leitura e escrita dos dados. Por outras palavras, estes sistemas garantem o armazenamento, a segurança e a manutenção de toda a informação alojada [36].

Esta tecnologia utiliza como linguagem de programação o Structured Query Language (SQL), conhecida pela sua sintaxe simples e compreensiva mesmo para quem não esteja familiarizado com a mesma.

Quando é necessário a persistência de um modelo de dados bem estruturado, *PostgreSQL* pode ser uma boa opção dado que nestas condições costuma apresentar uma melhor qualidade de performance que outros SGBD [37]. É também utilizado em situações em que é necessário uma BD mais complexa e personalizada.

3.2.4 *MySQL*

Outra das tecnologias estudadas como forma de verificação para o desenvolvimento do projeto descrito neste documento foi o *MySQL*. Este é um sistema *open source* que se foca em gerir base de dados relacionais.

Os casos de estudo mais conhecidos em que o *MySQL* é utilizado são aplicações *web* que necessitam de um modelo de dados simples mas onde a *performance* também é um critério relevante. É um SGBD bastante prático e permite que qualquer utilizador com pouca experiência consiga manobrar o *software* sem grandes dificuldades.

3.2.5 *PWA*

Uma *Progressive Web App (PWA)* é uma aplicação *web* que por determinadas características são executadas via *browser* e apresentam um comportamento muito idêntico às aplicações nativas em *Android* e *iOS* [38]. Estas aplicações possuem vantagens como:

- Podem ser construídas com tecnologias *web*;
- Têm menos custos de distribuição dado que não é necessário desenvolver várias aplicações para os diferentes sistemas;
- São responsivas e adaptam-se a qualquer ecrã;

- Apresentam uma boa performance;
- Funcionam *offline*, se necessário.

Porém, se o objetivo da aplicação a desenvolver requerer um alto desempenho, uma *pwa* pode não ser a melhor solução. No caso de aplicações de jogos, estas, se for algo mais complexo que um simples jogo de palavras, irá apresentar uma lentidão associada que não é desejada.

Outra das grandes vantagens apresentada neste tipo de tecnologias, e se não das mais importantes, é que uma *PWA* apenas funciona com conexões de *Hyper Text Transfer Protocol Secure (HTTPS)* [38]. Esta característica acaba por valorizar estas aplicações dado que as mesmas garantem uma conexão segura.

3.2.6 *Outsystems Platform*

Outsystems é uma plataforma de desenvolvimento *low code* que possibilita a construção de aplicações *web* e *mobile* sem a necessidade de escrever muitas linhas de código. De forma geral, esta plataforma compila os recursos mais utilizados em aplicações em *widgets*⁴, os quais permitem uma maior rapidez na construção das diferentes funcionalidades de uma aplicação [39].

Muitas vezes a plataforma *Outsystems* é referida como sendo composta pelos seguintes pilares [40]:

1. Natureza *Fullstack*;
2. Integração e Extensão;
3. Suporte a todo o ciclo de desenvolvimento;
4. Acesso a múltiplas experiências digitais.

O primeiro pilar indica que a plataforma de *Outsystems* possui todos os recursos necessários para iniciar um desenvolvimento de aplicações *web* ou *mobile* através de um ambiente dividido em quatro camadas. Estas camadas são responsáveis por macro tarefas como a criação do próprio modelo de dados, a implementação de processos de negócio síncronos e assíncronos, desenvolvimento de lógica com regras de negócio complexas e desenvolvimento *User Interface (UI)* através de ferramentas visuais.

O próximo pilar de *Outsystems* diz respeito à capacidade de integração e extensão da própria plataforma com outros serviços. Esta habilidade permite uma integração com as infraestruturas já desenvolvidas, tais como, base de dados, *Enterprise Information System (EIS)*⁵, serviços *cloud*, *REST* ou *SOAP*. Para além das integrações referidas anteriormente, *Outsystems* disponibiliza ainda um repositório denominado de *forge* onde é possível encontrar diversas extensões ou componentes desenvolvidos por outros programadores.

No que toca ao suporte do ciclo de desenvolvimento, esta plataforma utiliza um método de entrega de alta frequência conhecido como Continuous Integration/Continuous Delivery (CI/CD)

⁴Elemento de uma interface com o qual é possível iterar.

⁵Tipologia de um SI que tem como objetivo otimizar processos de negócio através da integração com o mesmo.

que possui técnicas de automação em todas as fases envolvidas na entrega de uma aplicação. A plataforma dispõe ainda de uma ferramenta de *debug* em browsers ou em dispositivos móveis e de um gestor de *deploys*, denominado por *Lifetime*, que permite uma passagem segura da aplicação entre ambientes retirando a preocupação ao utilizador de verificar todas as dependências e a criação de *scripts* de base de dados. Desta forma, a arquitetura da aplicação em questão nunca ficará comprometida durante a transição de um ambiente para outro.

Dado as largas possibilidades disponíveis na plataforma de *Outsystems*, esta permite acesso a um vasto conjunto de experiências digitais, no sentido em que um programador conseguirá desenvolver vários tipos de aplicações (*web*, *mobile* ou *PWA*). Todos os elementos estéticos são acessíveis e podem ser parametrizados, pelo que é possível construir qualquer aplicação que seria construída com uma linguagem de programação normal. A integração com serviços ou BD externas também é uma possibilidade que permite estender uma aplicação para algo muito mais amplo.

A *Outsystems*, como método de boas práticas, aconselha a utilização de um esquema arquitetural como referência para a construção das suas aplicações dado que estas são construídas em módulos. Cada módulo é responsável por uma determinada funcionalidade geral aplicacional e o mesmo deverá funcionar corretamente de forma isolada, pelo que, é necessário ter algumas considerações durante o desenvolvimento de cada um. O esquema arquitetural apresentado na Figura 3.3 representa os diferentes grupos de módulos defendidos pela própria *Outsystems*.

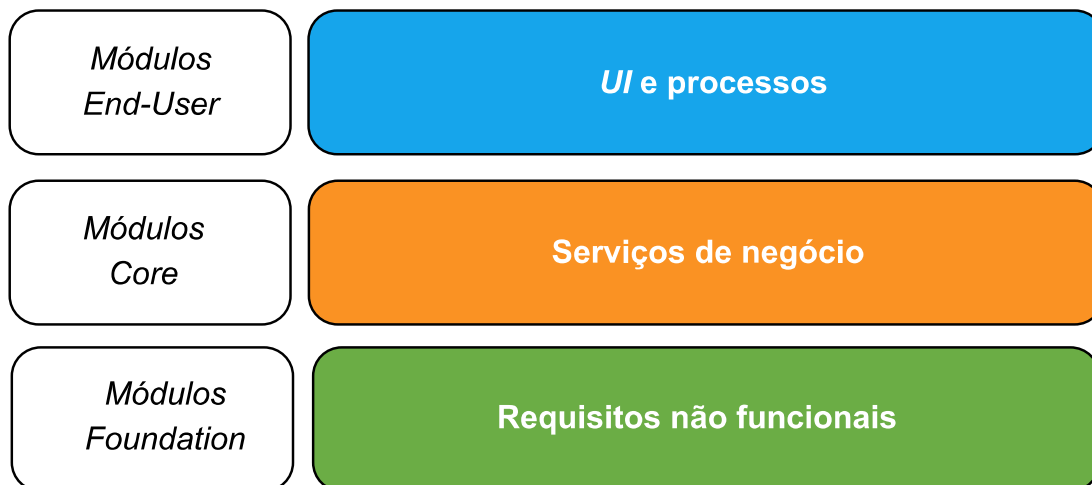


Figura 3.3: Esquema Arquitetural *Outsystems*.

Os módulos de *End-user* são responsáveis por incluir todo o *front-end* e interfaces das aplicações. Estes módulos poderão consumir quaisquer outros módulos abaixo, mas nunca poderão ser consumidos pelos mesmos. Esta última situação causaria uma *upward reference* que iria originar um problema de referências ao atualizar e efetuar passagens entre ambientes.

Os módulos *Core* são compostos desde toda a lógica aplicacional à persistência dos próprios dados. Sempre que for necessário desenvolver *software* que deverá ser reutilizado, esse deverá estar presente neste tipo de módulo. Relativamente às referências face a estes módulos, estes

podem ser consumidos por módulos acima destes e, ocasionalmente, por módulos também eles *core*.

Por último, os *Foudation Modules* contemplam a integração de aplicações ou serviços externos, o que possibilita uma rápida e simples integração com a aplicação em desenvolvimento. Podem ainda estar aqui introduzidas extensões desenvolvidas com recurso a C# de forma a expandir a potencialidade das aplicações.

Cada um dos módulos *Core* e *Foundation* podem ser compostos por vários tipos de módulos e deverão ser configurados com a própria nomenclatura associada pela *Outsystems*, seguindo a convenção apresentada na Tabela 3.1.

Tabela 3.1: Tabela de convenção de nomenclatura de módulos *Outsystems*.

Módulo	Sufixo	Descrição
Core	<i>_CS</i>	Serviços Core com entidades públicas, ações e blocos.
	<i>_BL</i>	Lógica de Negócio isolada
	<i>_CW</i>	<i>Core Widgets</i>
	<i>_ENG</i>	<i>Motores de cálculo complexos.</i>
	<i>_SYNC</i>	Lógica para sincronizar dados com um sistema externo quando o objetivo é tornar o sistema agnóstico.
	<i>_API</i>	Expõem API's a consumidores externos.
Foundation	<i>_LIB</i>	Módulo de bibliotecas.
	<i>_IS</i>	Serviços de Integração de API's.
	<i>_DRV</i>	Múltiplos integrações com serviços externos do mesmo contexto.
	<i>_TH</i>	Temas
	<i>_PAT</i>	Padrões UI reutilizáveis

Tal como referido na secção 3.2.5, *Outsystems* também possibilita o desenvolvimento de aplicações no formato *PWA*. A utilização desta funcionalidade permite que uma aplicação desenvolvida em *mobile* e gerada apenas para *Android* consiga executar corretamente em *iOS* utilizando como recurso um *web browser*.

3.3 Discussões

3.3.1 Planeamento

Face ao planeamento realizado no arranque inicial deste projeto apresentado na Figura 3.1 foi necessário efetuar algumas alterações a nível de alocação de recursos. Estas alterações consistiram maioritariamente num ajuste de tempo, associado às tarefas que foram detalhadas inicialmente, devido à incapacidade de realizar as mesmas em simultâneo com outras obrigações laborais e académicas.

Na Figura 3.4 é apresentado o cronograma que coincide com a realidade do desenvolvimento deste projeto.

Fases	Meses										
	2021	2022									
	Dezembro	Janeiro	Fevereiro	Março	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro
1 - Planeamento											
2- Levantamento de Requisitos											
3 - Desenvolvimento do Âmbito											
Enquadramento Introdutório											
Entrega Intermédia			◆								
Estado da Arte											
4- Desenvolvimento											
Desenho do Sistema											
Desenvolvimento Aplicacional											
5- Fase de Testes											
Testes de Aceitação											
Análise de Resultados											
6 - Entrega Final											◆

Figura 3.4: Cronograma do Projeto Real.

Como é possível validar ao observar o cronograma atualizado, o plano do projeto manteve-se inalterado até à fase de Entrega Intermédia. A interrupção verificada entre março e maio foi justificada pela sobrecarga de tarefas simultâneas que impediram a dedicação à continuidade do desenvolvimento do projeto.

Uma das grandes alterações entre o que foi planeado e a realidade está relacionado com o desenvolvimento do Estado da Arte. Esta tarefa sofreu uma despriorização em relação às restantes devido ao facto de ser necessário sempre alguns ajustes até à conclusão da mesma.

A fase de Desenvolvimento do projeto que inclui o Desenho e Análise do Sistema e todo o Desenvolvimento Aplicacional foi reduzido para um período de três meses. A alocação de apenas um mês para o desenvolvimento da aplicação móvel foi possível dado a escolha de uma plataforma *low code* como linguagem de programação.

Em relação à Fase de Testes existiu a necessidade de alocar a mesma a um período de dois meses. Por essa razão os Testes de Aceitação foram executados durante o mês de setembro e a Análise dos Resultados provenientes destes ocorreram durante o mês de outubro.

A data de entrega final do projeto manteve-se para o final do mês de outubro, tal como planeado anteriormente.

3.3.2 Soluções Tecnológicas

Como um dos problemas identificados anteriormente em 1.2 seria a partilha de informação por várias entidades, existiu a necessidade de optar pela utilização de uma *REST API*. Dada a complexidade e extensão do sistema, tornou-se oportuno a utilização de uma framework que permitisse a integração e exposição da *API* desenvolvida. Foi por essa razão que se optou pela utilização de *DRF* para o seu desenvolvimento.

O *PostgreSQL* acabou por ser selecionado enquanto SGBD por apresentar vantagens a nível de performance, referidas em 3.2.3, e pelo facto de possuir várias ferramentas adicionais, que

poderão vir a ser úteis com a escalabilidade do sistema, que não existiam no sistema mais simples *MySQL*.

Relativamente à tecnologia que se iria utilizar para desenvolver a aplicação móvel, a escolhida foi *Outsystems*. Como esta permite um rápido desenvolvimento de aplicações e ainda possibilita a conversão de aplicações móveis para *PWA* esta tornou-se facilmente a única alternativa válida face a outras possíveis linguagens de programação. Adicionalmente, como que já existia conhecimento com a própria tecnologia, não foi necessário realizar uma aprendizagem profunda acerca da mesma.

Capítulo 4

Análise, Desenho e Implementação

Neste capítulo será detalhado todo o processo de implementação do projeto estudado no presente documento. Após a construção do planeamento referente ao projeto e a seleção das tecnologias que fariam parte de todo o sistema, a próxima fase a desenvolver seria o desenvolvimento do desenho de toda a arquitetura e da aplicação, de forma a possibilitar uma abordagem de comunicação mais assertiva com o cliente ao longo de todo o processo de desenvolvimento aplicacional. No final deste capítulo é esperado uma conclusão em relação às decisões tomadas que afetaram todo o processo anteriormente referido.

4.1 Análise e Definição de Requisitos

Como Brooks referiu em [41], a etapa que mais esforço requer durante o desenvolvimento de um determinado software é a escolha de quais funcionalidades se irão materializar. O autor defende que esta etapa é de extrema importância dado que a passagem de algo conceitual para requisitos técnicos profundamente detalhados pode fazer a diferença entre o sucesso do desenvolvimento e o fracasso do mesmo. Macaulay e Mylopoulos em [42] também defendem que a definição de requisitos é a etapa onde a aceitação de todas as funcionalidades, todos os ecrãs e todas as restrições devem ficar refletidas e focam-se em três princípios que mais tarde surgem detalhados na norma *International Organization for Standardization (ISO) 29148:2018* [43]:

- 1. Capacidade de Implementação** - É necessário que os requisitos detalhados neste processo sejam capazes de ser implementados e validados com base em necessidades reais.
- 2. Aceitação de requisitos por parte de *stakeholders*** - Todos os requisitos devem ser revistos pelas partes envolvidas. Estes só aceitarão os requisitos se de uma forma global os mesmos responderem às necessidades solicitadas.
- 3. Referenciação de um ponto de partida para a interface aplicacional** - O detalhe presente nos requisitos deverá permitir visualizar a estrutura do *software* sem que seja necessário um elevado conhecimento do âmbito do mesmo. Poderá ainda ser utilizado como ponto de referência para as validações do próprio *software*.

Para definir um conjunto de requisitos bem estruturados não basta apenas considerar as necessidades do cliente. Esta etapa requer que sejam efetuados alguns desenvolvimentos sobre as mesmas dado que, geralmente, estas carecem de alguma objetividade e viabilidade para a sua implementação [43]. É também relevante perceber onde o projeto está inserido e quais as suas condições de forma a que todos os envolvidos estejam conscientes e de acordo com tudo o que é proposto nesta fase.

Pela razão mencionada no parágrafo anterior, a etapa de Análise e Definição de Requisitos foi estruturada da seguinte forma:

1. Descrição Geral do Sistema;
2. Requisitos Funcionais e Não Funcionais.

4.1.1 Descrição Geral do Sistema

O *software* a ser desenvolvido no presente projeto está inserido num sistema integrado denominado SIREPH. O SIREPH é um SI, construído com o objetivo de colmatar os problemas identificados em 1.2, que consiste em uma *API* que trata de toda a comunicação entre a BD e aplicações externas que a consumam. Desta forma, este sistema tem como objetivo integrar um conjunto de suporte às respostas de emergências pré-hospitalares, tais como:

1. Otimizar os procedimentos utilizados pelos técnicos durante as ocorrências;
2. Otimizar a transmissão de informação referente às ocorrências para diferentes instituições (Central de Emergência/Associação e Hospitais);
3. Possibilitar à população em geral um meio de contacto direto com as centrais de emergência de forma a acelerar todo o processo relacionado com a ocorrência.

Ao materializar cada ponto acima descrito obter-se-á três aplicações, uma para cada ponto referenciado, que poderão partilhar, analisar e atualizar a informação que por elas é gerada. Esta característica apenas é possível dado que todas elas estão integradas neste sistema.

Para responder ao primeiro ponto acima referido, existe uma aplicação dedicada aos TEPH em que o objetivo desta aplicação será a diminuição da necessidade do preenchimento manual do verbete (Anexo I), simplificar a comunicação entre as entidades envolvidas, dinamizar a visualização dos protocolos existentes de forma a auxiliar o registo de informação relevante para a ocorrência em questão e facilitar o processo de acionamento da Via Verde (Emergências Hospitalares Urgentes). Esta aplicação deverá ainda possibilitar o registo de todos os técnicos, a adição da equipa ativa no momento e a associação de diversas equipas à mesma ocorrência.

O objetivo do sistema apresentado no segundo ponto é respondido com uma aplicação composta por duas áreas que se subdividem pelas instituições que a ela acedem (Centrais de Emergência e Hospitais). A área designada para a Central é considerada como um *BackOffice* onde é possível gerir as equipas de técnicos que respondem às ocorrências, de forma

a possibilitar a atualização, análise e a criação de relatórios com métricas que poderão ajudar a melhorar a resposta do mesmo. O acesso para os Hospitais deverá ser condicionado à leitura das ocorrências, recebidas pelos mesmos, garantindo uma transmissão da informação mais assertiva e rápida, para que o conhecimento dos procedimentos efetuados às vítimas seja persistido pelo sistema.

O último ponto apresentado será colmatado com a criação da aplicação em estudo no presente documento, pelo que a mesma será descrita e detalhada mais à frente.

De forma a facilitar a compreensão da integração destas aplicações com o sistema SIREPH, é apresentada a Figura 4.1 de forma a perceber que não existe comunicação direta entre as mesmas. O esquema apresentado está organizado segundo a estrutura utilizada para os esquemas de arquitetura de *Outsystems* em que os componentes externos como *API's* são representados em laranja e as interfaces de utilização para utilizadores finais a azul.



Figura 4.1: Esquema Geral de Integração de aplicações com o sistema SIREPH.

4.1.1.1 Descrição Geral do *Software*

A aplicação a ser desenvolvida neste estudo tem como objetivo principal possibilitar aos indivíduos que se encontrem perante uma situação de emergência, utilizar um conjunto de funcionalidades que possam facilitar a gestão de uma ocorrência, como a transmissão da localização exata do local da ocorrência e outras informações que pretendem ajudar a agilizar o processo de gestão a mesma.

Toda a informação gerada pela aplicação deverá ser persistida no sistema SIREPH com intuito que a mesma possa posteriormente ser analisada e/ou utilizada por diferentes entidades de forma a possibilitar uma melhor tomada de decisão na melhoria dos processos envolvidos na gestão de emergências pré-hospitalares e uma otimização de todo o processo envolvido.

4.1.1.2 Funcionalidades do *Software*

Em relação às funcionalidades da aplicação, estas resultaram da agregação das necessidades do cliente alvo de um processo de *brainstorming*. Dessa forma, a aplicação deve desempenhar as seguintes funções:

- Registo da informação clínica do utilizador;
- Registo da informação clínica de familiares do utilizador;

- Seleção da tipologia de emergência;
- Obtenção da localização exata do dispositivo móvel;
- Envio da informação de uma ocorrência para o SIREPH;
- Efetuar a chamada para o contacto mais adequado de acordo com a emergência.

4.1.1.3 Caraterísticas dos Utilizadores

Todos os indivíduos de uma sociedade estão aptos a utilizar a aplicação em estudo desde que sejam portadores do conhecimento base de funcionamento de um *smartphone* e de aplicações móveis. Considerando esta ligeira limitação, é de esperar que a maioria dos utilizadores estejam concentrados em idades inferiores a 50 anos, dado que a utilização destes dispositivos móveis é mais frequente nestas faixas etárias.

Para uma utilização correta da aplicação não será necessário fornecer formação a utilizadores visto que a mesma possui um tutorial com três breves passos que identifica as principais funções e capacidades da própria. Adicionalmente, um dos requisitos a validar é a facilidade de utilização da própria aplicação, sendo que esta necessita de ser intuitiva o suficiente para que possa ser utilizada em situações de stress.

Para utilizar a aplicação os utilizadores deverão estar de acordo com os termos e condições da mesma que engloba a partilha da informação introduzida na aplicação com as diferentes entidades envolvidas, de forma a que seja possível enviar toda a informação considerada relevante para a abertura e análise de uma ocorrência.

A informação partilhada poderá ainda ser alvo de análises com o objetivo de melhorar o processo de tomada de decisões, pelo que, o utilizador também necessita de concordar com esta.

4.1.1.4 Dependências e Pressupostos

Como primeiro pressuposto para uma correta execução da aplicação é necessário que a mesma seja executada num *smartphone android* ou *iOS*. Para que a aplicação funcione como esperado é necessário que a comunicação com a *API* do sistema SIREPH esteja sempre operacional. De forma a garantir esta operacionalidade é essencial uma conexão à Internet, seja por redes móveis ou *Wi-Fi*. As permissões de acesso à câmara do dispositivo e à localização do mesmo devem também ser concedidas de forma a que seja possível obter informação através desses mesmos sensores.

Devido a questões de permissões de *deployment* a aplicação apenas funcionará corretamente enquanto *PWA* no browser da Google (Google Chrome) em dispositivos *iOS*. Relativamente aos dispositivos *Android*, a aplicação funcionará corretamente tanto em browsers como em *PWA* instalada no próprio dispositivo.

4.1.2 Especificação de Requisitos

Segundo a norma *ISO 29148:2018*, um requisito pode ser descrito utilizando uma linguagem natural onde deverá ser explícito qual o assunto e os elementos necessários para expressar a exigência associada [43]. Na mesma norma é possível verificar a importância que a priorização de requisitos, através de escalas de 1 a 5 ou com três termos (Alto, Médio e Baixo), pode ter em etapas futuras. Em situações de mudança pode ser necessário realizar trocas de requisitos ou até excluir alguns do próprio âmbito do projeto. Nesses casos é utilizado, como termo de comparação, a prioridade que cada requisito pode ter para o projeto corrente.

Como forma de priorizar os requisitos definidos para o presente projeto foi utilizado uma escala de três termos, sendo definido como 'Alto' os requisitos mais prioritários e 'Baixo' os requisitos com menor prioridade de implementação. A escolha desta escala justifica-se pelo facto do âmbito do projeto ser bastante objetivo e o esforço alocado ao desenvolvimento do mesmo ser pouco extenso.

4.1.2.1 Requisitos Funcionais

Um requisito funcional é definido como sendo um requisito que descreve uma ação que o próprio *software* deverá ser capaz de realizar [44]. De seguida será apresentado uma listagem tabular com todos os requisitos funcionais definidos para o presente projeto e a prioridade associada aos mesmos.

Tabela 4.1: Requisito Funcional 1 - Restrição de Uso.

RF1	Restrição de Uso	Prioridade	Alto
	A aplicação deve obrigar a que o utilizador esteja registado na mesma de forma a que toda a informação do seu perfil seja persistido no sistema. Para a utilização da aplicação, não deverá ser necessário o utilizador recorrer a nenhum tipo de formação, pelo que esta deverá fornecer toda a informação necessária para a sua utilização.		

Tabela 4.2: Requisito Funcional 2 - Preenchimento da informação clínica do utilizador padrão.

RF2	Preenchimento da informação clínica do utilizador padrão	Prioridade	Alto
	A aplicação deve obrigar o registo da informação clínica do utilizador habitual do <i>smartphone</i> . Este será considerado como o utilizador padrão da aplicação e todas as ocorrências abertas por este ficarão associadas ao seu perfil mesmo que este não faça parte das vítimas da emergência em questão. A informação a preencher deverá respeitar a informação solicitada no novo verbete nacional de socorro (Anexo I), nomeadamente, a identificação e o historial clínico da vítima.		

Tabela 4.3: Requisito Funcional 3 - Registo de outros perfis.

RF3	Registo de outros perfis	Prioridade	Médio
A aplicação deverá possibilitar a adição de diversos perfis de outros indivíduos de forma a que seja possível ao utilizador padrão efetuar o registo de uma ocorrência para outros indivíduos que façam parte do seu ciclo familiar. A informação a preencher para estes casos deverá ser a mesma que a solicitada no registo do utilizador padrão.			

Tabela 4.4: Requisito Funcional 4 - Registo de uma ocorrência.

RF4	Registo de uma ocorrência	Prioridade	Alto
A aplicação providenciará a criação de uma ocorrência através do clique de um botão de emergência. Após este clique será despoletado um processo de solicitação de informação ao utilizador, onde o mesmo deverá indicar a tipologia da emergência para que esta seja encaminhada para o canal/contacto correto. No caso de uma emergência pré-hospitalar a informação clínica da(s) vítima(s) será anexada à ocorrência e enviada para o SIREPH.			

Tabela 4.5: Requisito Funcional 5 - Seleção da tipologia da emergência.

RF5	Seleção da tipologia da emergência	Prioridade	Alto
De forma dinâmica a aplicação deverá demonstrar um processo simples, ordenado pelas situações mais recorrentes e graves, de forma a possibilitar que o utilizador selecione, em poucos cliques, o tipo de emergência associada à situação atual. Cada tipologia de emergência estará associado a um canal (SNS24, 112, etc...) de forma a que cada situação seja encaminhada para os respetivos canais. Este processo de organização da tipologia das emergências deverá ser customizável.			

Tabela 4.6: Requisito Funcional 6 - Seleção da(s) vítima(s) registadas.

RF6	Seleção da(s) vítima(s) registadas	Prioridade	Médio
Durante o processo de registo de uma ocorrência, o utilizador deverá conseguir selecionar as vítimas que estão associadas à situação emergente atual. Esta seleção deverá possibilitar que o utilizador padrão possa ou não estar incluído como vítima. A informação do utilizador padrão será utilizada para associar a criação da ocorrência ao mesmo, de forma, a possibilitar um <i>tracking</i> de ocorrência por utilizador.			

Tabela 4.7: Requisito Funcional 7 - Seleção da(s) vítima(s) não registradas.

RF7	Seleção da(s) vítima(s) não registradas	Prioridade	Baixo
A aplicação deverá possibilitar a leitura de um <i>QR Code</i> durante a seleção da(s) vítima(s). Este <i>QR Code</i> será mostrado no ecrã de bloqueio dos indivíduos que possuam a aplicação instalada. Desta forma será possível abrir ocorrências para indivíduos fora dos perfis registados pelo utilizador padrão.			

Tabela 4.8: Requisito Funcional 8 - Detecção da localização da ocorrência.

RF8	Detecção da localização da ocorrência	Prioridade	Alto
Durante o processo de envio da informação da ocorrência para o SIREPH, a aplicação deverá aceder à localização do dispositivo e anexar a mesma à informação da ocorrência.			

Tabela 4.9: Requisito Funcional 9 - Realização de chamadas.

RF9	Realização de chamadas	Prioridade	Médio
Depois do envio da informação de uma ocorrência para o SIREPH a aplicação deverá realizar uma chamada para o canal relacionado com a tipologia de emergência selecionada.			

Tabela 4.10: Requisito Funcional 10 - Listagem de Ocorrências.

RF10	Listagem de Ocorrências	Prioridade	Baixo
A aplicação deve ainda permitir a visualização do histórico das ocorrências abertas pelo utilizador padrão, informando o estado das mesmas.			

4.1.2.2 Requisitos Não Funcionais

Os requisitos não funcionais dizem respeito a todos os requisitos que, de certa forma, definem características que o próprio *software* deve alcançar mas que não estão diretamente relacionadas com questões funcionais [44]. Segue abaixo uma listagem tabular, estruturada de forma idêntica à anterior, dos requisitos não funcionais definidos para o projeto em estudo.

Tabela 4.11: Requisito Não Funcional 1 - Responsividade.

RNF1	Responsividade	Prioridade	Alto
A aplicação deve ser desenvolvida como uma <i>PWA</i> de forma a estar otimizada para todos os <i>smartphones</i> mais recentes.			

Tabela 4.12: Requisito Não Funcional 2 - Usabilidade.

RNF2	Usabilidade	Prioridade	Alto
A interface gráfica deverá estar preparada para ser utilizada em situações de emergência, apresentando uma forma de ser utilizada de forma simples e fácil.			

Tabela 4.13: Requisito Não Funcional 3 - Disponibilidade.

RNF3	Disponibilidade	Prioridade	Médio
A aplicação só funcionará em contexto online, pelo que será sempre necessário uma conexão à Internet para que a mesma funcione corretamente.			

Tabela 4.14: Requisito Não Funcional 4 - Segurança.

RNF4	Segurança	Prioridade	Médio
A aplicação deverá assegurar que todos os dados presentes na mesma estejam seguros, de modo a impossibilitar a partilha de informação com terceiros estranhos ao sistema.			

Tabela 4.15: Requisito Não Funcional 5 - Integrabilidade.

RNF5	Integrabilidade	Prioridade	Alto
A aplicação deverá estar integrada com o sistema externo SIREPH, de forma a possibilitar a transmissão de informação entre estes.			

Tabela 4.16: Requisito Não Funcional 6 - Integridade.

RNF6	Integridade	Prioridade	Médio
A aplicação deverá garantir a consistência e a precisão dos dados durante todo o ciclo de vida da informação, de modo a minimizar ao máximo a ocorrência de falhas antes, durante e após os processos de emergência.			

Tabela 4.17: Requisito Não Funcional 7 - Reatividade.

RNF7	Reatividade	Prioridade	Médio
A aplicação deve disponibilizar a informação assim que esta esteja disponível sem impactar as restantes funcionalidades independentes, de forma a que não seja detetada nenhuma latência associada.			

4.1.3 Discussão

Após uma análise dos requisitos definidos e detalhados foi realizada uma seleção de quais os requisitos que deveriam ser considerados para a materialização da aplicação e quais não seriam incluídos na mesma.

Pela complexidade anexa ao desenvolvimento do requisito definido na tabela 4.7 foi decidido que o mesmo não iria ser incluído no *backlog* aplicacional devido ao facto de este não ser necessário para a realização da validação da própria aplicação.

Dado que a aplicação desenvolvida para o presente projeto iria consistir num protótipo, também não faria sentido desenvolver, na sua totalidade, o requisito apresentado na tabela 4.9. Desta forma, procedeu-se à implementação parcial deste requisito visto que a aplicação seria utilizada apenas em contexto de testes de aceitação, não faria sentido incluir a funcionalidade de marcação e chamada para os números de emergência.

4.2 Desenho de *Software*

Desde bastante cedo que a fase de desenho de um *software* é conhecida por ser um processo que utiliza os requisitos especificados como *input* para criar uma representação visual do que deverá ser a própria aplicação [45]. O processo de transformação destes conceitos para algo tangível permite evitar erros futuros, mais concretamente na fase de desenvolvimento ou de validação da aplicação [46]. Qualquer erro relacionado com a definição de requisitos detetado numa fase posterior à mesma tem um custo demasiado elevado para o projeto, dado que poderá ser necessário refazer algum do trabalho já realizado. Por essa razão, é possível afirmar que a utilização de técnicas de *design* podem ajudar a economizar tempo e custos durante o ciclo de desenvolvimento de um projeto [47].

Como em diferentes etapas do desenvolvimento de um projeto, também para um processo de desenho da aplicação existe várias técnicas e ferramentas que podem ser utilizadas. De uma forma geral, as técnicas utilizadas mais aceites são [48]:

1. *Wireframes*;
2. *Mockups*;
3. Protótipos.

De forma a clarificar que as técnicas acima referidas têm objetivos diferentes, Uzayr utiliza uma analogia para com o corpo humano. Nessa referência, os *wireframes* são comparados ao esqueleto do corpo humano visto que estes apresentam a estrutura que serve de base para um *software*. Os *mockups* acrescentam à comparação anterior todas as características físicas que permitem reconhecer uma pessoa como a pele, o cabelo, entre outros, dado que já apresentam alguma sensibilidade para a aparência do próprio software. Por último, os protótipos são vistos como um corpo humano em completo funcionamento, que ao possuir todas as características atrás

referidas e um cérebro está em plena capacidade apta de ser utilizado para estudos ou até fases de validação do *software*.

Como o objetivo deste projeto passa pela construção rápida de uma aplicação, apenas interessa desenhar algo que sirva de base de comunicação e concordância para o cliente. Por essa razão, a técnica adotada foi a elaboração de *wireframes* que são definidos como desenhos técnicos que demonstram as funcionalidades ao longo da aplicação por meio de formas geométricas que identificam os objetos a colocar nos ecrãs da mesma [47].

4.2.1 Associação de Requisitos a *Wireframes*

Como referido anteriormente, os *wireframes* foram construídos com base nos requisitos definidos, principalmente em requisitos funcionais. De forma a clarificar a associação de cada um dos requisitos aos *wireframes* desenvolvidos, será apresentada a tabela 4.18 seguida de uma descrição detalhada de como esses *wireframes* respondem aos requisitos associados.

Tabela 4.18: Correlação de requisitos com *wireframes*.

	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9	RF10
Apêndice A.1	x									
Apêndice A.2	x									
Apêndice A.3	x	x								
Apêndice A.4				x						
Apêndice A.5			x							
Apêndice A.6										x
Apêndice A.7				x	x	x		x	x	

4.2.1.1 RF1 - Restrição de Uso

Como forma de receção do utilizador na aplicação, foi criado um tutorial (Apêndice A.1) com o objetivo de transmitir ao utilizador a informação necessário para uma utilização correta da própria aplicação. Este tutorial é composto por três passos onde é possível, de uma forma simples e rápida, apresentar aos utilizadores as principais funcionalidades e objetivos da aplicação, tornando todo o processo de habituação à mesma mais prático. Os ecrãs a serem desenvolvidos com base nestes *wireframes* apresentam a seguinte informação:

- Primeiro ecrã - Apresenta uma referência à introdução facilitada da informação clínica do utilizador;
- Segundo ecrã - Refere a possibilidade de adição de vários perfis de outros indivíduos;
- Terceiro ecrã - Comunica a forma com a qual a aplicação consegue otimizar o processo de pedido de ajuda, realçando o envio da informação previamente preenchida e a localização do dispositivo.

Para restringir o acesso a toda a aplicação foi desenhado um ecrã (Apêndice A.2) em que é possível ao utilizador efetuar o *login* na aplicação. Caso o utilizador não se encontre registado na aplicação, é apresentado a possibilidade de efetuar um novo registo através de um botão.

Para concluir a resposta a este requisito, foram desenhados ecrãs presentes no Apêndice A.3 que representam todo o processo de registo de um novo utilizador. A informação necessária para o sucesso desse registo encontra-se refletida nos primeiros 4 ecrãs apresentados no apêndice.

4.2.1.2 RF2 - Preenchimento da informação clínica do utilizador

Para além do registo de um novo utilizador, o Apêndice A.3 apresenta ainda como será inserida toda a informação clínica do próprio. No quinto e sexto ecrã do apêndice é possível verificar que a informação considerada clinicamente relevante como as doenças crónicas, alergias conhecidas e medicação habitual possuem um atributo binário que despoletará a inserção de vários registos numa lista associada a cada característica referida. Deverá ainda ser possível adicionar o grupo sanguíneo do utilizador dado que o mesmo poderá mostrar-se útil em algumas ocasiões.

Como forma de garantir transparência para o utilizador em relação ao destino da informação introduzida na aplicação, a mesma apresenta um conjunto de consentimentos relacionados com a finalidade de informar que os dados partilhados no sistema têm como objetivo melhorar os processos envolvidos, podendo essa mesma informação ser alvo de análises por parte de entidades envolvidas.

4.2.1.3 RF3 - Registo de outros perfis

Para o processo de registo de outros perfis foram criados os *wireframes* apresentados no apêndice A.5. Estes *wireframes* possuem a mesma informação presente no registo do utilizador padrão com a exceção da palavra-passe e *email*. É também possível preencher toda a informação clínica do indivíduo para o qual se está a criar o perfil e observar uma listagem de todos os perfis associados à conta do utilizador atual.

4.2.1.4 RF4 - Registo de uma ocorrência

Para dar início ao processo de abertura de uma ocorrência na aplicação foi criado o botão "Pedir Ajuda", visível no Apêndice A.4, que despoleta o processo representado nos *wireframes* do Apêndice A.7. Para clarificar este processo de criação de uma ocorrência foi elaborado o esquema presente na Figura 4.2 que traduz a estrutura e funcionalidade do próprio.

Este processo inicia-se pela Seleção de vítimas que será abordado na secção 4.2.1.6.

De seguida o utilizador é encaminhado para outro ecrã que tem como objetivo percecionar o Estado de Consciência da vítima. O segundo ecrã do Apêndice A.7 é responsável por esta decisão de onde resulta a primeira ramificação do processo e permite a divisão do mesmo em duas grandes vertentes, ocorrências com vítima(s) conscientes ou inconscientes.

Relativamente ao processo de uma ocorrência com uma vítima inconsciente o próximo passo é a verificação de respiração. Se a vítima não respirar deve ser efetuada uma chamada para o canal

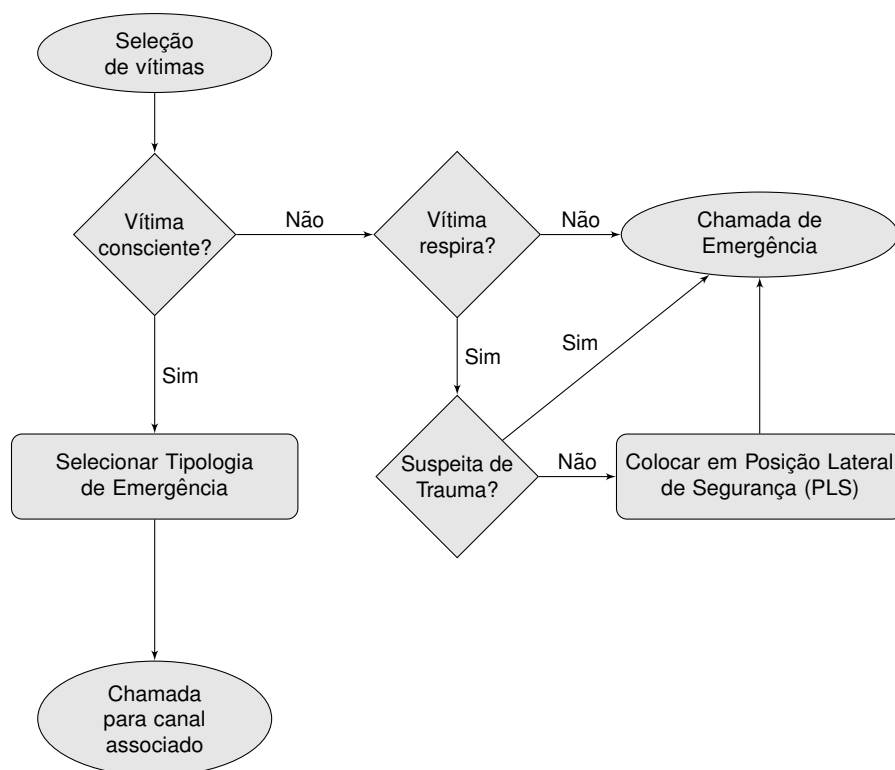


Figura 4.2: Processo de criação de uma ocorrência na aplicação.

de emergência o quanto antes para que a informação da mesma seja encaminhada e o socorro à vítima seja o mais rápido possível. Nestes casos, a razão pela qual a vítima se encontra neste estado não é muito relevante, pelo que a informação de que a vítima se encontra inconsciente e sem respirar é motivo suficiente para justificar o pedido de ajuda. Quando a vítima se encontra inconsciente e a respirar surge uma verificação de forma a tentar perceber se a mesma poderá ter sido vítima de algum tipo de trauma. Esta verificação permite transmitir a informação de que a vítima poderá, ou não, ser deslocada da sua posição inicial sem que exista o perigo de causar mais sequelas. Para casos em que não existe qualquer suspeita de trauma, será aconselhado a colocação da vítima em PLS.

4.2.1.5 RF5 - Seleção da tipologia da emergência

Em processos de ocorrências em que a vítima se encontre consciente, será necessário selecionar a tipologia de emergência (sexto ecrã do Apêndice A.7) para posteriormente selecionar o canal para o qual se deverá efetuar a chamada. Cada tipologia estará associada a um canal diferente consoante a sua criticidade de forma a colmatar um dos problemas apresentados em 1.2. As tipologias que serão apresentadas na aplicação foram selecionadas com base em Informação partilhada pelo INEM [10].

4.2.1.6 RF6 - Seleção da(s) vítima(s) registadas

A seleção de vítimas registadas consiste numa funcionalidade na qual é permitida a seleção de um ou vários perfis criados na aplicação incluindo o do utilizador padrão. O primeiro ecrã do

Apêndice A.7 é responsável por esta etapa do processo. Esta é uma etapa crucial para a criação de uma ocorrência dado que uma ocorrência só poderá ser criada com sucesso se houver vítimas a associar à mesma.

4.2.1.7 RF8 - Detecção da localização da ocorrência

A detecção da localização do dispositivo que está a executar a aplicação é uma funcionalidade que é executada em *background* e que por essa razão não é visível nos *wireframes*. Este é o último passo a ser executado antes do envio da informação da ocorrência ao sistema SIREPH sendo por isso efetuado na transição para o ecrã de Chamada de Emergência.

4.2.1.8 RF9 - Realização de chamadas

Para a seleção e associação de chamadas a um determinado canal (112 e SNS24) foi considerado o sistema já implementado pelo CODU, denominado *Telephonic Triage and Counseling System (TETRICOSY)*. Este sistema tem como objetivo reduzir o erro em respostas a ocorrências e priorizar as mesmas consoante a sua criticidade, baseando-se nos seguintes princípios [49]:

Prioridade 1 Ocorrência que requer meios de Suporte Avançado de Vida (SAV) devido ao estado da vítima ser considerado crítico. Situações em que a vítima necessita de intervenções imediatas.

Prioridade 3 Vítima que apresenta sintomatologia que requer de SBV num espaço temporal não tão alarmante quanto as situações anteriores.

Prioridade 5 e Outras Prioridades Ocorrências que pela sua criticidade não necessitam de qualquer meio de socorro.

Em relação à aplicação refletida no presente documento, a Prioridade 1 e 3 foram consideradas como tipologias associadas a chamadas de emergência (112) enquanto que as restantes serão encaminhadas para o canal da SNS24. A associação de tipologias e diferentes canais foi desenhada de forma dinâmica para permitir uma adição de novos registos no futuro, sem que seja necessário efetuar alterações a nível do código da aplicação.

4.2.1.9 RF10 - Listagem de Ocorrências

Para observar a lista de ocorrências associadas ao utilizador foi criado os *wireframes* presentes no Apêndice A.6 onde foi ainda adicionado a funcionalidade de observar alguns detalhes acerca das mesmas, como as vítimas associadas, a data da ocorrência e a tipologia da mesma.

4.2.2 Discussão

Em algumas circunstâncias, durante a fase de desenvolvimento dos *wireframes*, os requisitos definidos foram adaptados e explorados numa ótica mais prática e aproximada à realidade, o que fez com que existissem algumas alterações que necessitassem de ser incorporadas. Alterações

como a adição da funcionalidade de partilha de notícias referentes à área de negócio abordada neste projeto, foram replicadas para o *wireframe* presente no Apêndice A.4 mesmo não possuindo um requisito funcional que a justifica-se. Esta adição foi discutida entre os diferentes *stakeholders* do projeto que rapidamente observaram o valor que a mesma poderia demonstrar com a sua presença na aplicação final.

No subcapítulo 4.2.1 foi possível observar a mais valia que a transformação de requisitos em *wireframes* possibilitou ao observar a proximidade de desenho entre ambos.

4.3 Arquitetura do Sistema

Desde há muito que o recurso a modelos de arquitetura de sistemas são bastante utilizados como forma facilitadora da compreensão, análise e validação de um determinado sistema de software [50]. Independentemente da técnica ou ferramenta utilizada, estes modelos são compostos por representações dos elementos de um sistema e das interações entre os mesmos [51].

Com o avanço da tecnologia surge a *cloud computing* com um paradigma em relação ao padrão arquitetural utilizado, dado que estes possuem recursos de software dinâmico [52]. Dada esta característica, pode torna-se complexo a construção de modelos arquiteturais dado que os sistemas que recorrem a esta dinamicidade, como os sistemas orientados a dispositivos móveis, requerem uma elevada escalabilidade e usabilidade por forma a possibilitar o crescimento exponencial que poderá ter este tipo de *software*.

Uma das formas utilizadas mais aceites para representar os modelos arquiteturais de um determinado sistema são as linguagens de modelação como o *Unified Modeling Language (UML)*. O grande problema da representação deste tipo de modelos não está baseado na técnica utilizada, mas sim no nível de detalhe que deverá ser utilizado para a construção dos mesmos [53], pois o aumento da complexidade do modelo apresentado provoca uma diminuição da capacidade de comunicação e transmissão da informação pelo mesmo.

De forma a facilitar essa compreensão foi então desenvolvido um esquema, com recurso a *UML*, presente na Figura 4.3 onde é possível observar a iteração que os serviços externos têm com a aplicação estudada no presente documento.

Pelas razões acima apresentadas e de forma a facilitar a organização do modelo arquitetural construído, este subcapítulo encontra-se organizado da seguinte forma:

- Modelo Arquitetural do sistema SIREPH;
- Modelo Arquitetural da aplicação *SmartCall*¹.

4.3.1 Arquitetura SIREPH

Com base no esquema arquitetural do sistema SIREPH apresentado na Figura 4.3 é possível verificar a configuração efetuada no servidor remoto da *Hetzner*. Esta empresa é um dos maiores

¹ Nome escolhido para a aplicação desenvolvida para o estudo do presente documento.

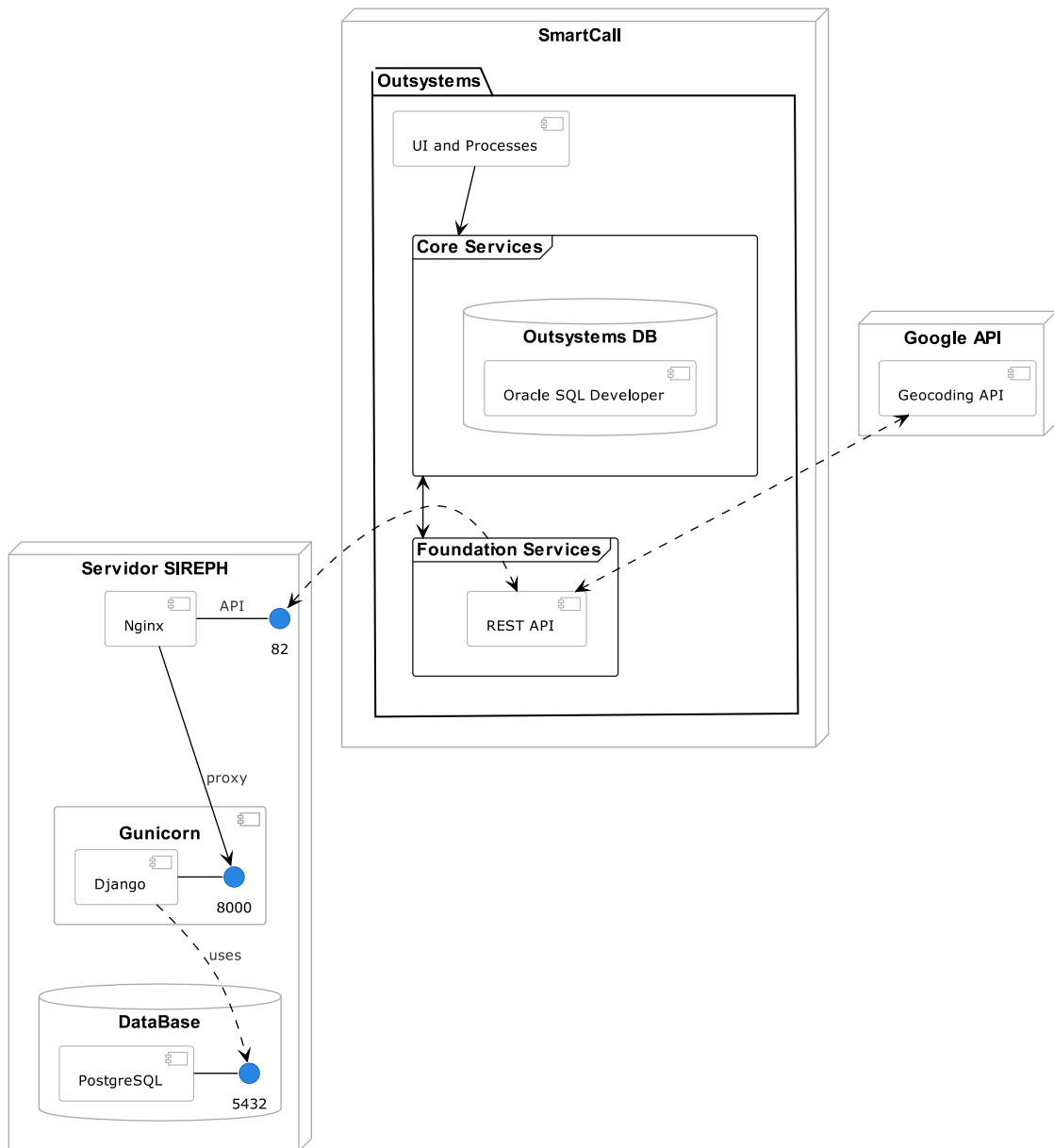


Figura 4.3: Diagrama Arquitetural do sistema.

data centers da Europa e tem como estratégia de negócio o aluguer de infraestruturas tecnológicas avançadas a um preço acessível [54]. Dessa forma, foi reservado um servidor remoto, alojado na Alemanha, com suporte a sistema *linux* para alojar a *API* que seria acedida pelas diferentes aplicações.

4.3.1.1 Configuração do servidor

Para realizar o *deploy* na máquina fornecida foi necessário efetuar algumas configurações iniciais recorrendo a outras ferramentas como o *Gunicorn* que é uma biblioteca bastante utilizada como provedor de aplicações que suportam *WSGI* e é conhecida pela sua fiabilidade e escalabilidade [55]. Porém, a integração entre *Django* e *Gunicorn* não seria suficiente para permitir a exposição da *API*. Foi ainda necessário adicionar à infraestrutura o *Nginx*, um *web server* para tratar e

processar todos os pedidos efetuados pelo *browser* [56]. Para perceber como é que estas ferramentas funcionam em conjunto foi elaborado o esquema presente na Figura 4.4.

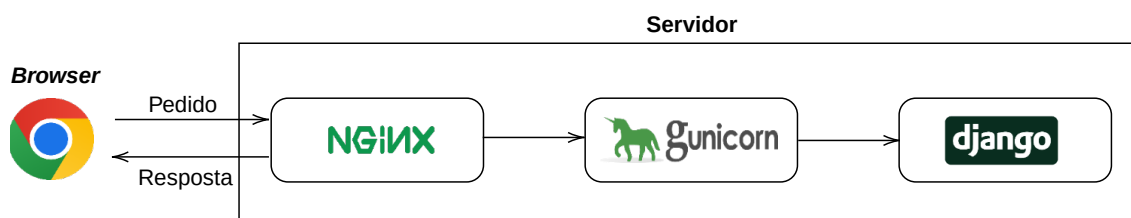


Figura 4.4: Esquema de funcionamento do *Nginx* e *Gunicorn* com *Django*.

Quando o *browser* efetua um pedido, este é diretamente recebido pelo *Nginx* que verificará qual a tipologia do conteúdo a enviar. Se o conteúdo de resposta for da responsabilidade do *Django*, como é o caso de uma página *HyperText Markup Language (HTML)* dinamicamente gerada, o pedido é passado ao *Gunicorn* e posteriormente ao *Django*. No exemplo fornecido, quando a página estiver gerada o *Gunicorn* será responsável pelo envio da página para o *Nginx* que retornará a resposta ao *browser* de forma a que seja possível visualizar a informação recebida.

Com a configuração da infraestrutura realizada, foi possível colocar a aplicação em *Django* no *Gunicorn* exposta através da porta 8000 para que usando um servidor *proxy* com *Nginx* fosse possível aceder ao mesmo. Foi assim possível usar a porta 82 para expor a *API* desenvolvida em *DRF* para que fosse possível consumir a mesma através da aplicação em estudo. Com base no esquema da Figura 4.3, é ainda possível verificar que a aplicação desenvolvida em *Django* utiliza uma base de dados *PostgreSQL* que está exposta na porta 5432. O modelo de dados desta base de dados e os métodos que o utilizam irão ser detalhados nas seguintes secções.

4.3.1.2 Modelo de Dados SIREPH

Relativamente ao modelo de dados do sistema SIREPH, este está representado na sua totalidade no Apêndice B. Este modelo foi construído com base na informação presente no verbete nacional de socorro em Anexo I, tendo sido estendido consoante seria necessário a adição de mais informação nas aplicações consumidoras. A extensão deste modelo é justificada pela escalabilidade e integração do mesmo com as diferentes aplicações referidas em 4.1.1.1. Por essa razão, foi elaborado o modelo de dados simplificado apresentado na Figura 4.5 com o objetivo de identificar e detalhar apenas as tabelas incluídas no âmbito da aplicação *SmartCall*.

A tabela *User* é utilizada para guardar o registo dos utilizadores que deverão ter acesso à *API* desenvolvida. Esta tabela persiste informação como a *password* encriptada, a data do último *login*, o *username*, o nome, o apelido, o email, a data de criação do utilizador e se o registo está ou não ativo. Existem ainda os campos *is_superuser* e *is_staff* que, respetivamente, permite atribuir um conjunto de acessos de administrador aos utilizadores e verificar se os utilizadores criados são utilizadores técnicos do sistema (TEPH).

Cada utilizador que é registado no sistema tem um *token* associado de forma a que seja possível validar os pedidos que estes efetuam. Para que tal associação seja possível, foi

adicionado a tabela *Token* que regista o *Id* do Utilizador associado a cada *token*.

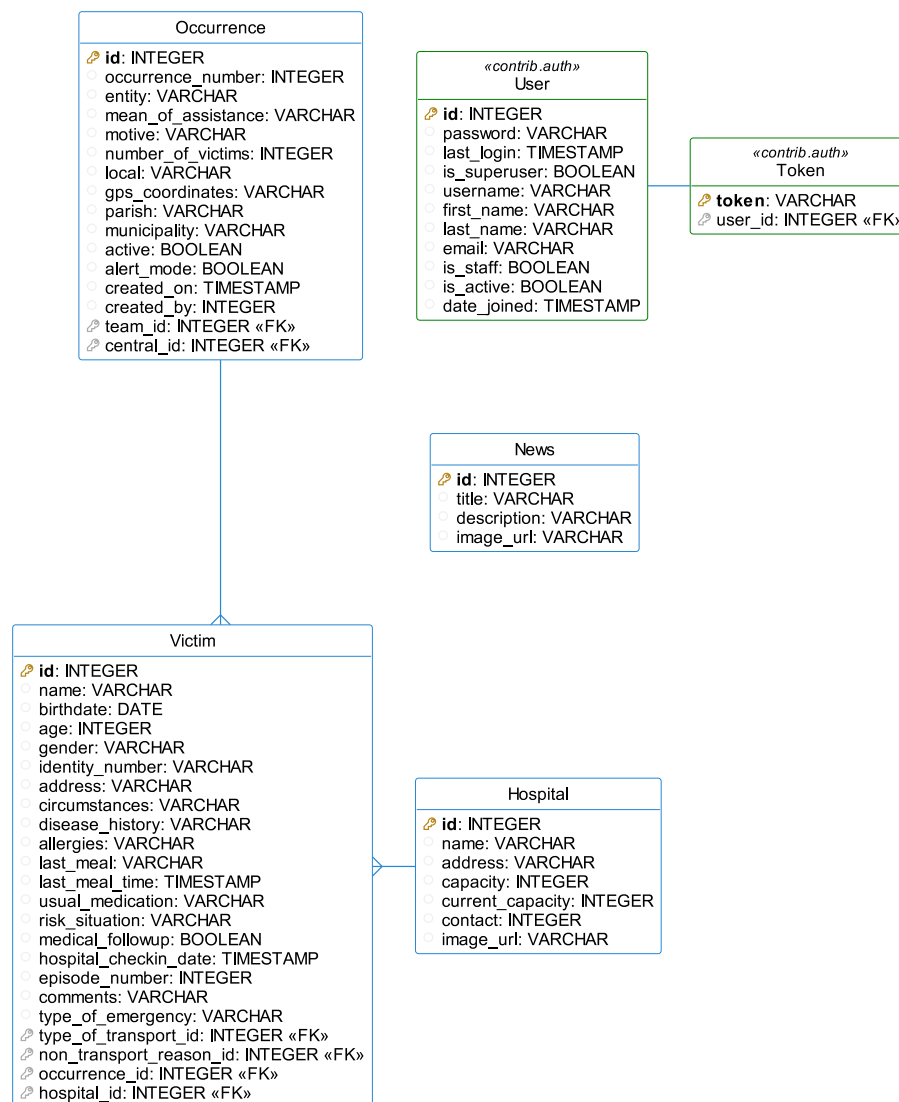


Figura 4.5: Modelo de Dados SIREPH simplificado.

Relativamente à tabela *News*, esta tem como propósito persistir a informação das notícias que deverão ser partilhadas com os utilizadores da aplicação. A utilização desta tabela no sistema permite que todas as aplicações consumidoras tenham acesso às notícias atualizadas sem que seja necessário consumir outro serviço externo. Cada notícia deve ser persistida com informação referente ao título, à descrição da mesma e uma imagem alusiva caso necessário.

A tabela *Occurrence* é responsável por persistir a informação associada às ocorrências. Nesta tabela, é persistido a seguinte informação:

- *occurrence_number* - Número Identificador da ocorrência no sistema;
- *entity* - Nome da entidade que presta o socorro;
- *mean_of_assistance* - Identificação do meio de socorro;
- *motive* - Motivo de acionamento do meio de socorro;

- `number_of_victims` - Número de vítimas associadas à ocorrência;
- `local` - Identificação do local, morada e/ou pontos de referência;
- `gps_coordinates` - Coordenadas geográficas do local da ocorrência;
- `parish` - Freguesia do local da ocorrência;
- `municipality` - Município do local da ocorrência;
- `active` - Estado que indica se a ocorrência ainda se encontra ativa;
- `alert_mode` - Estado que indica se a ocorrência foi iniciada por um civil e carece de atenção por parte das centrais de emergência;
- `created_on` - Data de criação da ocorrência;
- `created_by` - *Id* do utilizador que criou a ocorrência;
- `team_id` - *Id* da equipa de emergência associada à ocorrência;
- `central_id` - *Id* da central assignada à ocorrência.

Uma ocorrência pode ter várias vítimas associadas. Pode essa razão, a tabela *Victim* persiste a informação do *Id* da Ocorrência no campo `occurrence_id`. Para além deste campo a tabela persiste ainda informação característica das vítimas como o nome, a data de nascimento, a idade, o género, o número de identificação (Cartão de Cidadão (CC) ou Bilhete de Identidade (BI)) e a morada. Adicionalmente é ainda guardado a informação clínica da vítima no momento da abertura da ocorrência como:

- `circumstances` - Descrição cinemática, fator(es) desencadeante(s), testemunhos, entre outros;
- `disease_history` - Doenças conhecidas mais relevantes;
- `allergies` - Alergias ou episódios recorrentes conhecidos;
- `last_meal` - Última refeição consumida;
- `last_meal_time` - Hora da última refeição consumida;
- `usual_medication` - Terapêutica habitual da vítima;
- `risk_situation` - A preencher em caso de maus tratos ou condições sociais adversas;
- `medical_followup` - Campo que indica se a vítima foi acompanhada por um médico;
- `hospital_checkin_date` - Data de chegada da vítima ao hospital;
- `episode_number` - Número de episódio referente à Unidade de Saúde;

- `comments` - Indicar tudo o que seja relevante caso não tenha sido descrito nos campos anteriores;
- `type_of_emergency` - Condição clínica ou hipótese de diagnóstico;
- `type_of_transport_id` - Tipo de transporte: Primário ou Secundário;
- `non_transport_reason_id` - Razão a preencher caso não tenha sido efetuado transporte para uma Unidade de Saúde;
- `hospital_id` - Identificador do Hospital para o qual a vítima foi transportada.

Para situações em que a vítima já possua um hospital associado é possível obter a informação do mesmo através da relação criada entre esta e a tabela *Hospital*. Esta tabela regista informação referente ao nome, morada, capacidade total, capacidade atual, contacto e um *url* da imagem do hospital.

4.3.1.3 Aplicação Django

A aplicação desenvolvida com *DRF* consiste numa *REST API* que tem como objetivo efetuar a gestão dos registos persistidos no modelo de dados anteriormente apresentado construída como descrito em 3.2.2. Esta encontra-se acessível através do *url* `http://78.46.215.82:82/api/`.

De forma a que todas as aplicações que pretendam consumir o sistema SIREPH consigam ter conhecimento de todos os métodos desenvolvidos, foi utilizado o *Swagger* pelas razões já referidas em 3.2.1.

De seguida, serão apresentados todos os métodos presentes na *API* que foram utilizados pelas diferentes funcionalidades da aplicação *SmartCall*. Os métodos não mencionados no presente documento encontram-se documentados em [57].

- `user_create (/user/)` - Método *POST* responsável pela criação ou atualização de utilizadores. Este método recebe um objeto do tipo *User* e caso não exista nenhum erro é retornado sucesso com o objeto enviado e o identificador do utilizador associado;
- `login_create (/login/)` - Método *POST* que realiza o *login* de um determinado utilizador. Este método recebe como *input* um *username* e uma *password*. Após validação das credenciais recebidas, este retorna sucesso com um objeto *User* e o *token* associado ao mesmo;
- `user_occurrence_list (/user/{user_id}/occurrences)` - Método *GET* que obtém todas as ocorrências, abertas ou fechadas, por um determinado utilizador. Este método recebe o *token* como *Authorization Key* e um identificador do utilizador para o qual se pretende receber a informação. Quando devolvido com sucesso a resposta deste consiste numa lista de objetos *Occurrence* com a informação de todas as *Victim's* associadas;

- `hospitals_list (/hospitals/)` - Método *GET* que devolve todos os hospitais registados na tabela *Hospital*. Este tem como input o *token* associado ao utilizador como forma de validar a permissão de acesso à informação. Caso sucesso este método devolve uma listagem de registos de hospitais;
- `news_list (/news/)` - Método *GET* que devolve todos os registos da tabela *News* quando o pedido é bem sucedido. À semelhança do anterior, este método apenas tem o *token* como input;
- `occurrence_create (/occurrence/)` - Método *POST* responsável pela criação de uma ocorrência no sistema. Este método recebe a informação de um objeto *Occurrence* e o *token* como forma de validação de autenticação. Quando registado com sucesso, devolve o objeto criado com o identificador respetivo;
- `victims_create (/victims/)` - Método *POST* que recebe uma lista de vítimas a associar a uma ocorrência. Este método tem como *input* uma lista de objetos *Victim* e o *token* de autenticação.

Todos os *inputs* e *outputs* serão tratados no módulo da aplicação *SmartCall* responsável pela integração de sistemas externos que será descrito na próxima secção.

4.3.2 Arquitetura *SmartCall*

Relativamente à arquitetura desenhada para a aplicação *mobile* desenvolvida foi considerado o esquema arquitetural recomendado pela *Outsystems* detalhado em 3.2.6, incluindo a nomenclatura para os mesmos.

De forma a simplificar a arquitetura demonstrada na Figura 4.3 foi elaborado o esquema apresentado na Figura 4.6 com os módulos que foram desenvolvidos durante o desenvolvimento da aplicação.

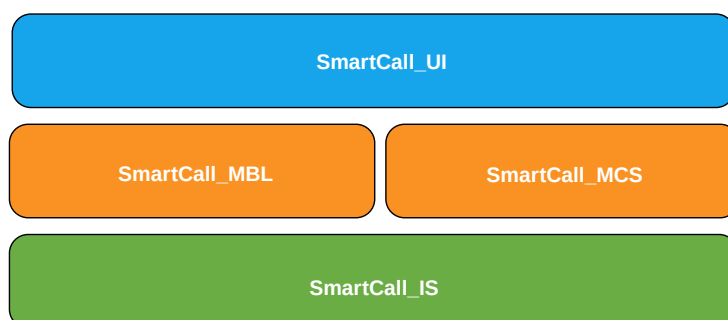


Figura 4.6: Modelo Arquitetural *Outsystems* da aplicação *SmartCall*.

4.3.2.1 *SmartCall* Integration Service

O módulo `SmartCall_IS` representa um *Foundation Service* responsável pela integração da aplicação com todos os serviços externos necessários para a mesma. É neste módulo que

está criado o serviço que irá consumir tanto a *API* do SIREPH como outra *API* que se mostrou necessária para a obtenção da informação da localização com base nas coordenadas obtidas pelo dispositivo. Esta segunda *API*, *Google Geocoding API*, é integrada neste sistema com o objetivo de efetuar a conversão de coordenadas geográficas em detalhes da morada, como a rua, o número da porta, o código-postal, entre outros [58]. Através desta conversão é então possível comunicar ao utilizador a localização num formato que este irá compreender. De forma a que fosse possível utilizar esta *API* com sucesso, foi apenas necessário requerer uma chave de autenticação e associar a mesma a uma conta *Google* para efeitos de validação de acessos para cada pedido efetuado.

Relativamente à integração da *API* do SIREPH, de forma a otimizar o módulo em questão, apenas foi realizado o consumo dos métodos que eram necessários para o funcionamento da própria aplicação. Por essa razão, os métodos que estão a ser consumidos pela aplicação são:

- *user_create* - Funcionalidade responsável pela criação de um utilizador no sistema SIREPH;
- *login_create* - Método responsável pela autenticação de um utilizador. Este método recebe como *input* o *username* e a *password* e se validados retorna o *token* que será necessário para as restantes comunicações;
- *hospitals_list* - Este método envia uma listagem de todos os hospitais presentes no sistema;
- *news_list* - Responsável pela obtenção das notícias a partilhar na aplicação;
- *occurrence_create* - Método responsável pela criação de uma ocorrência no sistema;
- *victim_create* - Ação responsável por criar ou atualizar qualquer registo de uma vítima;
- *user_occurrences_list* - Método que devolve a listagem de ocorrências e todo o seu detalhe associado ao utilizador em sessão.

Com a exceção dos primeiros dois métodos acima referidos, para a obtenção de uma resposta com sucesso é necessário a comunicação do *token* de autenticação recebido durante o processo de *login*.

4.3.2.2 SmartCall Mobile Core Service

O módulo *SmartCall_MCS* é responsável pela gestão de uma base de dados necessária para garantir a persistência da informação dos utilizadores da aplicação móvel.

Durante o desenvolvimento da aplicação tornou-se necessário persistir alguns dados num servidor da *Outsystems* de forma a efetuar a gestão de acessos através das plataformas fornecidas pela *Outsystems*. Como alguma da informação persistida apenas seria utilizada no contexto desta aplicação, a partilha e persistência na íntegra da mesma com o sistema não se tornou relevante. Na Figura 4.7 é possível observar o modelo de dados criado para a persistência da informação anteriormente referida.

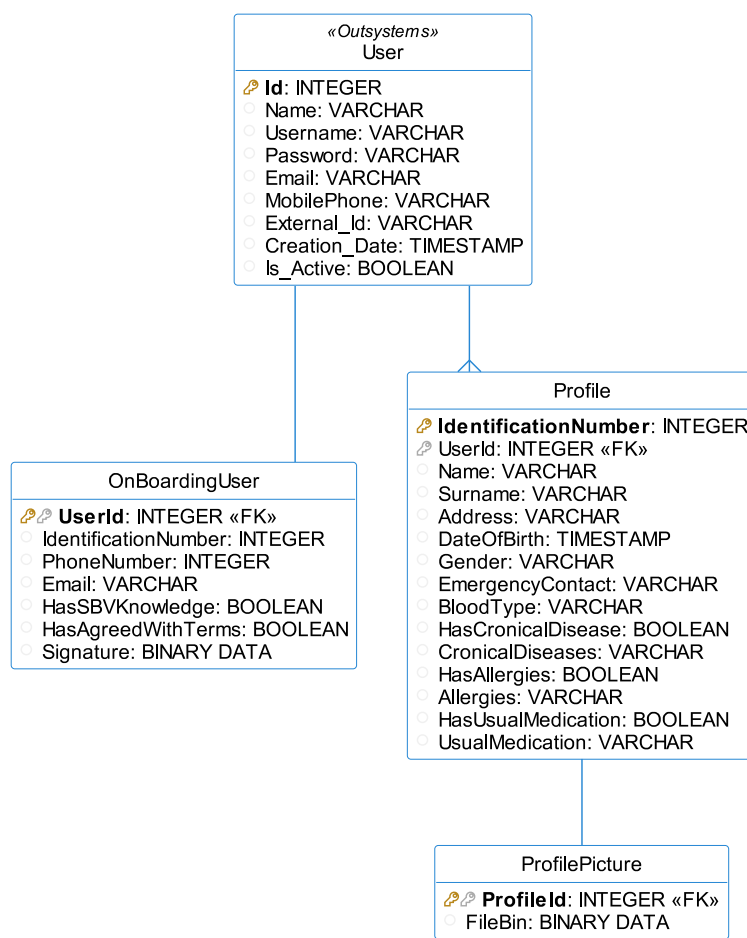


Figura 4.7: Modelo de dados do utilizador na aplicação.

De forma a acelerar muitos processos o *Outsystems* já possui algumas tabelas de sistema como é o caso da tabela *User*. Esta persiste a informação dos utilizadores que possam ter acesso ao ambiente aplicacional e permite a gestão dos mesmos através do *Service Studio*. A utilização desta tabela foi necessária para que o potencial de informação local fosse aproveitado na sua totalidade, como é o caso da persistência de dados em sessão. Em relação à adição de um registo nesta tabela, esse apenas acontece após a criação do utilizador no sistema SIREPH dado que se tornou necessário a persistência do *Id* externo do utilizador no campo *External_Id*.

A tabela *OnBoardingUser* tem como objetivo registar a informação alocada ao utilizador aquando do registo do mesmo. Esta tabela apenas mantém dados associadas ao utilizador aplicacional e alguns meios de contacto que possam ser úteis no âmbito da aplicação como o número de telemóvel e o email, é ainda solicitado o preenchimento do cartão de cidadão (campo *IdentificationNumber*) dado que será esse o valor persistido como *username* do utilizador. O campo *HasSBVKnowledge* guarda a informação se o utilizador padrão da aplicação tem algum conhecimento de técnicas de SBV. Se a resposta for afirmativa, então a aplicação poderá ser configurada de forma a apresentar indicações mais concretas do que fazer em certas tipologias de ocorrências. A persistência da informação sobre a autorização de termos e condições da utilização da aplicação está refletida nos campos *HasAgreedWithTerms* e *Signature* que, para efeitos de

auditorias, poderão vir a mostrar-se úteis. O resto da informação do utilizador é persistida na tabela *Profile* e será detalhada de seguida.

No que toca à informação que é necessária persistir para os perfis que futuramente serão utilizados para abrir uma ocorrência, esta encontra-se persistida na tabela *Profile*. Esta tabela persiste a informação clínica que poderá ser alterada sempre que necessário de forma a que seja enviada a informação mais recente e correta para o sistema SIREPH. Para além da informação básica característica (número de identificação, nome, apelido, morada, data de nascimento e género) é ainda solicitado:

- Contacto de emergência (*EmergencyContact*) - Relevante para situações em que as vítimas não possuam nenhum contacto de emergência no SNS ou que o mesmo esteja desatualizado;
- Grupo sanguíneo (*BloodType*) - Para casos em que o indivíduo tenha conhecimento de qual o seu tipo de sangue pode ajudar a antever a necessidade do mesmo;
- Doenças Crónicas (*CronicalDiseases*) - Listagem concatenada de doenças crónicas que o indivíduo possui;
- Alergias (*Allergies*) - Listagem concatenada de alergias que o indivíduo possui. Esta informação pode prevenir a administração de algum tipo de medicação que possa causar uma reação alérgica à vítima;
- Medicação recorrente (*UsualMedication*) - Listagem concatenada de medicação que o indivíduo usa regularmente. Ao ter conhecimento de qual a medicação tomada pela vítima, os intervenientes têm uma maior facilidade em tratar a mesma;

De forma a tornar a aplicação fácil de utilizar, existe a possibilidade de adicionar uma fotografia a cada perfil de um indivíduo. Esta fotografia poderá facilitar o processo de seleção de vítimas durante a abertura de uma ocorrência, visto que, a facilidade de interpretação de uma imagem é superior à de um texto. Segundo as boas práticas recomendadas pela *Outsystems* [59], a informação de ficheiros persistida numa base de dados deverá ser, sempre que possível, isolada da tabela a que se refere por forma a evitar que a informação seja sempre requerida mesmo quando não é necessária. Por essa razão, foi criada a tabela *ProfilePicture* que tem o objetivo de persistir a fotografia associada a cada perfil registado sem que haja a necessidade de carregar a mesma sempre que se aceda à informação clínica de um indivíduo. Esta tabela apenas possui a informação do perfil a que se refere e a própria imagem.

De forma a suportar a funcionalidade de seleção da tipologia da ocorrência, e garantir a escalabilidade da aplicação, foi ainda criado o modelo apresentado na Figura 4.8, utilizando entidades estáticas visto que estes registos não seriam alvo de atualizações recorrentes.

A tabela *Motive* observada na Figura 4.8 é responsável por persistir os registos selecionados como tipologias de ocorrências [10]. Cada tipologia terá uma descrição técnica associada sendo que o texto apresentado na aplicação será algo mais genérico (campo *label*) e compreensivo

para todos os indivíduos. Como exemplo prático a tipologia de Obstrução da Via Aérea terá essa mesma descrição porém, na aplicação aparecerá como Engasgo para que evitar qualquer hesitação sobre a sua definição. O campo *Order* é utilizado para atribuir a ordem com que as tipologias aparecerão na aplicação considerando uma ordem decrescente face ao número de ocorrências verificadas para cada tipologia em 2021 [10].

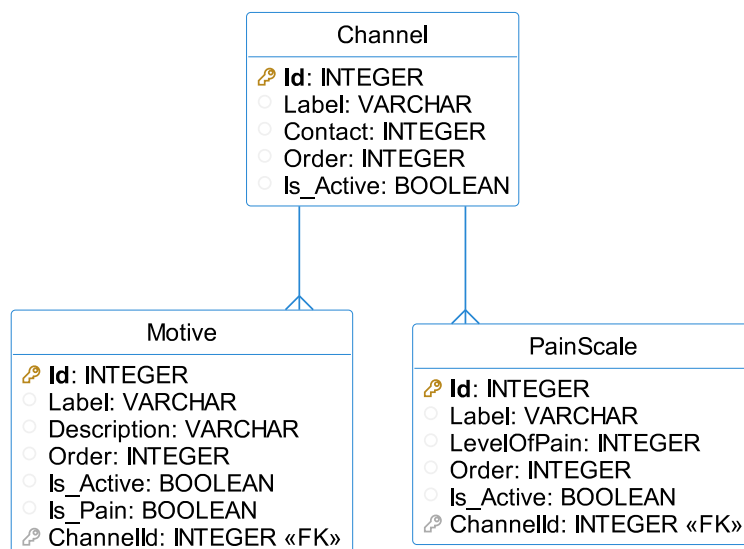


Figura 4.8: Modelo de dados da tipologia de ocorrências na aplicação.

Cada tipologia inserida na aplicação deverá ter um canal associado. Na tabela *channel* estão persistidos os canais para associação às próprias tipologias. Cada canal de emergência tem um contacto associado e uma *label* que o permite identificar quando demonstrado ao utilizador através da aplicação (ex: 112 - Linha de Emergência Médica ou 808242424 - Linha Saúde 24).

Durante o processo de desenvolvimento aplicacional surgiu ainda a necessidade de distinguir algumas tipologias de ocorrência dado que, consoante a sua criticidade, estas poderiam resultar na seleção de diferentes canais. As tipologias que se encontravam nesta situação eram, na sua totalidade, tipologias associadas à dor. Por essa razão, e com base no algoritmo descrito em 4.2.1.8 foi criada uma tabela denominada de *PainScale* com o objetivo de persistir informação associada a uma escala de dor. Para perceber se determinada tipologia (*motive*) deveria ou não estar associada a essa escala foi adicionado o campo *Is_Pain* de forma a que fosse possível efetuar a associação do canal através de um dos seguintes registos ordenados de forma crescente consoante a sua criticidade:

1. Dor Suave - Nível de Dor 2
2. Dor Moderada - Nível de Dor 4
3. Dor Forte - Nível de Dor 6
4. Dor Muito Forte - Nível de Dor 8
5. Dor Máxima - Nível de Dor 10

Para o âmbito de validação da própria aplicação foi considerado que um nível de dor superior a 6 estaria associado ao canal da Linha de Emergência Médica e os restantes à Linha Saúde 24.

4.3.2.3 SmartCall Mobile Business Logic

No módulo *SmartCall_MBL* podemos observar toda a lógica de negócio necessária a utilizar na interface da aplicação criada. As ações que contemplam esta lógica de negócio permitem a preparação da informação que a aplicação recebe ou envia para os serviços externos através da utilização de estruturas criadas com esse mesmo propósito.

4.3.2.4 SmartCall User Interface

Dado que a aplicação desenvolvida se trata de uma aplicação para dispositivos móveis, o módulo de interfaces gráficas requer a utilização de um armazenamento local para a persistência de dados em sessão orientados a objetos. Este armazenamento local será alimentado sempre que um utilizador efetue o *login* com sucesso através da ação de sincronização. Desta forma, toda a informação da aplicação se encontrará guardada localmente, evitando assim a necessidade de obter informação a todo o momento. No Apêndice C pode verificar-se todo o modelo de dados associado a este armazenamento.

Como o intuito do armazenamento local passa por manter os dados necessários para a execução da aplicação disponíveis a qualquer momento, foi relevante a existência de tabelas quase idênticas às referidas em 4.3.2.2 como:

- *LocalUser*;
- *LocalProfile*;
- *LocalMotive*;
- *LocalChannel*;
- *LocalPaiScale*.

Em relação às restantes tabelas como é o caso da tabela *LocalVictim* esta é utilizada durante o processo de criação de uma ocorrência para persistir a informação dos perfis que foram selecionados como vítimas. A informação clínica dos perfis selecionados é transportada para esta tabela e posteriormente o estado da vítima (estado de consciência e respiração) é atualizado no campo *Circumstances*. Uma vítima estará sempre associada a uma ocorrência sendo que essa relação é persistida no campo *LocalOccurrenceId*. É ainda permitido observar qual o hospital a que cada vítima está associada de forma a poder demonstrar essa informação nos detalhes de cada ocorrência através do campo *LocalHospitalId*.

A tabela *LocalHospital* irá ser responsável por persistir a informação de todos os hospitais obtidos através do método *hospitals_list*. Esta tabela terá informação referente ao nome, morada, contacto e um *url* de uma imagem da instituição de forma a que quando for necessário

mostrar a informação de um hospital seja possível mostrar uma imagem do mesmo e tornar a visualização mais apelativa.

Em relação à tabela responsável por registar a informação de uma ocorrência esta denomina-se de *LocalOccurrence*. Nesta tabela é persistido o número de ocorrência que é automaticamente gerado através da geração de um *GUID*². A informação relativa à tipologia da ocorrência é guardada no campo *LocalMotiveld* e o canal a que esta ficará associado é persistido em *LocalChannelId*. Tornou-se necessário persistir esta última informação pois a relação entre o motivo e o canal pode depender de uma escala de dor tal como mencionado em 4.3.2.2. A informação da localização do dispositivo é persistida nos campos *GPSCoordinates* e *Local*, sendo que apenas é possível preencher este último com a utilização da *Geocoding API* da Google para efetuar a conversão de coordenadas em morada. O campo *AlertMode* é utilizado para informar as restantes aplicações que esta ocorrência merece atenção no sentido em que foi criada e ainda não possui nenhuma equipa técnica e/ou central associada.

Existe ainda a tabela *LocalNews* que, há semelhança da tabela de hospitais, é responsável por persistir a informação de todas as notícias a demonstrar ao utilizador. Estes registos possuem um título, uma descrição que contempla o corpo da notícia e um *url* de uma imagem para casos em que faça sentido adicionar uma.

4.4 Desenvolvimento Aplicacional

Um dos grandes desafios que existe no desenvolvimento de aplicações móveis é o facto de existir a necessidade de criar várias aplicações para as diferentes plataformas (*Android* e *iOS*) [60]. Como referido em 3.3, a tecnologia escolhida para desenvolver a aplicação *SmartCall* foi *Outsystems*, o que permitiu uma despreocupação com a distribuição da aplicação final tal como explicado em 3.2.6. A utilização desta tecnologia permitiu a construção de toda a aplicação, incluindo as integrações necessárias com serviços externos, em apenas algumas semanas.

Com o desenvolvimento dos módulos detalhados em 4.3.2 para satisfazer as necessidades das funcionalidades introduzidas em cada ecrã, foi possível materializar os wireframes desenhados (4.2.1). De seguida será detalhado a materialização de cada ecrã da aplicação de forma comparativa com os *wireframes*, demonstrando a solução final.

4.4.1 Ecrã Inicial

De forma a facilitar a visualização do primeiro exemplo, foi adicionado a Figura 4.9 que permite observar o *wireframe* desenvolvido (4.9a) comparativamente com o ecrã inicial materializado (4.9b e 4.9c).

Observando a imagem acima apresentada é possível verificar que todas as funcionalidades gerais introduzidas no *wireframe* foram desenvolvidas com a exceção de um menu lateral. Durante o processo de desenvolvimento concluiu-se que pela simplicidade da estrutura da aplicação

²Número de referência único que identifica uma informação concreta num sistema.

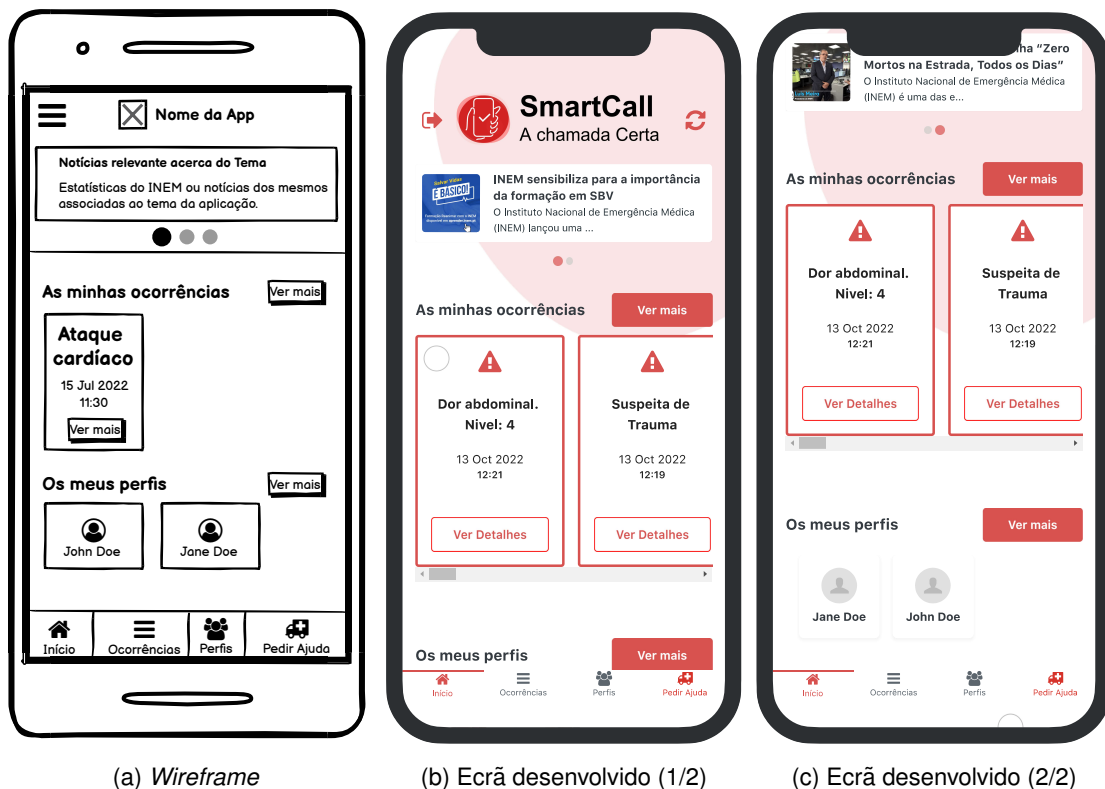


Figura 4.9: Comparação entre *wireframe* e materialização do ecrã inicial da aplicação.

não seria necessário um menu lateral dado que todos os atalhos necessários conseguiram ser introduzidos no ecrã. No topo do ecrã 4.9b, junto do nome da aplicação, é possível verificar dois botões, um para efetuar o término da sessão do utilizador atual e outro para realizar o processo de sincronização manual. É possível verificar que o bloco das notícias ficou tal como planificado no *wireframe* adicionando ainda uma imagem para situações em que fosse necessário.

Na secção "As minhas ocorrências" é possível observar uma lista ordenada por ordem de abertura das ocorrências abertas pelo utilizador em que é possível visualizar o detalhe de cada uma através do botão "Ver Detalhes". Ao clicar no botão "Ver mais" desta secção, o utilizador será reencaminhado para o ecrã apresentado no Apêndice D.6a.

A secção "Os meus perfis", apenas visível em 4.9c devido ao *scroll* da aplicação, demonstra uma lista de todos os perfis associados ao utilizador e a sua fotografia caso esta exista. Ao clicar no botão "Ver mais" desta secção, o utilizador será endereçado para o ecrã presente no Apêndice D.5a.

Ainda em relação ao ecrãs demonstrado em 4.9b e 4.9c, é possível observar o menu apresentado no fundo do ecrã. Este menu composto por quatro *links* que remetem o utilizador para os seguintes ecrãs da aplicação:

- Início - Página Inicial da aplicação demonstrada na Figura 4.9b;
- Ocorrências - Listagem de ocorrências do utilizador (Apêndice D.6a);
- Perfis - Lista de perfis associados ao utilizador (Apêndice A.5a);

- Pedir Ajuda - Início do processo de abertura de uma ocorrência (Apêndice D.7a).

4.4.2 Tutorial

Relativamente aos ecrãs do tutorial inicial da aplicação (Apêndice D.1), o desenvolvimento destes foi realizado exatamente como os *wireframes* desenvolvidos (Apêndice A.1). Não existe nenhuma diferença relevante pelo que o funcionamento desejado foi o adquirido.

4.4.3 Login

O ecrã de *login* desenvolvido pode ser observado no Apêndice D.2a e sofreu uma alteração com base no *wireframe* (Apêndice A.2). A este ecrã foi adicionado a possibilidade de voltar para o último ecrã do tutorial (Apêndice D.1c) que já possuía um botão para iniciar o processo de registo de um novo utilizador, pelo que, a persistência do botão "Registar", tal como desenhado, tornou-se redundante. Por essa razão foi retirado esse botão.

Quando um utilizador efetuar o *login* com sucesso, este será reencaminhado para o ecrã apresentado no Apêndice 4.9b.

4.4.4 Registo do utilizador

Relativamente aos ecrãs responsáveis pelo processo de registo de novos utilizadores, estes, baseados nos *wireframes* apresentados no Apêndice A.3 obtiveram todas as suas funcionalidades implementadas como é possível verificar no Apêndice D.3. As funcionalidades e informações solicitadas nestes ecrãs foi detalhada enquanto desenho aplicacional na secção 4.2, pelo que não fará sentido a repetição da mesma nesta secção.

4.4.5 Notícias

O ecrã de notícias foi o único ecrã cuja funcionalidade não foi originada por um requisito funcional, pelo que foi considerado como não prioritário tanto para o desenvolvimento dos *wireframes* como para o desenvolvimento aplicacional. Visto que o esforço associado à implementação deste não era exigente, o mesmo foi desenvolvido (Apêndice D.4a) mesmo não tendo um *wireframe* como ponto de partida.

Este ecrã serve o propósito de partilhar notícias relevantes e relacionadas com a área de inserção da aplicação e possibilita ao utilizador a visualização de uma imagem alusiva à notícia, o título e o corpo da própria notícia. Desta forma os utilizadores poderão ter acesso a uma amplitude maior de informação que teriam sem esta funcionalidade.

4.4.6 Perfis

Relativamente ao ecrã de listagem de perfis desenvolvido apresentado no Apêndice D.5a este sofreu umas alterações face à distribuição dos seus elementos. Para um utilizador adicionar um novo perfil é necessário utilizar o botão no canto inferior direito com o sinal "+" para visualizar as

opções de ações que poderá realizar neste ecrã (D.5b). Ao adicionar um novo perfil, o utilizador será encaminhado para os ecrãs já referidos anteriormente presentes nos Apêndices D.3c a D.3h.

Para proceder à eliminação de um perfil será apresentado uma *popup* na aplicação que solicita quais dos perfis, com a exceção do perfil do utilizador, deverá ser eliminado tal como é demonstrado no Apêndice D.5c.

4.4.7 Listagem de Ocorrências

Os ecrãs desenvolvidos para a listagem de ocorrências dos utilizadores sofreram algumas adições face ao construído nos *wireframes* (Apêndice A.6). No Apêndice D.6a e D.6b é possível observar que foi adicionado a este ecrã a potencialidade de observar, respetivamente, quais as ocorrências que o utilizador tem abertas e quais estão fechadas. Este filtro pode ser útil como forma de prevenir que sejam abertas novas ocorrências para a mesma situação emergente.

No Apêndice D.6c é possível verificar os detalhes de uma ocorrência. Neste ecrã é possível observar o número, a tipologia, a data e hora de criação, a(s) vítima(s) associada(s) e o local da ocorrência. É ainda possível observar o local num mapa, escondido por defeito, de forma a obter uma informação mais exata em relação à localização da ocorrência.

Para casos em que a vítima já esteja associada a um hospital, esta informação também será demonstrada no ecrã de detalhe como pode ser observado no Apêndice D.6d.

O facto do ecrã de detalhe da ocorrência estar muito mais completo que o seu *wireframe* justifica-se pela falta de dados inicial. Apenas quando a aplicação começou a gerar dados foi possível analisar os mesmos de forma a perceber qual a informação que poderia ser trabalhado e mostrada ao utilizador enquanto detalhe de uma ocorrência.

4.4.8 Criação de Ocorrência

Em relação aos ecrãs associados aos *wireframes* do Apêndice A.7 foram efetuadas alterações relativamente ao processo em si. Essas alterações resultaram na criação de mais um ecrã e na adição de uma funcionalidade, de forma a obter mais informação sobre a situação de emergência, que serão descritos mais à frente nesta secção.

O processo de criação de uma ocorrência continua a iniciar-se na seleção da(s) vítima(s) a associar à mesma e está apresentada no Apêndice D.7a. De seguida o utilizador é questionado acerca do estado de consciência da(s) vítima(s) como apresentado no Apêndice D.7b. Aqui existe a primeira ramificação do processo dependendo da resposta do utilizador. Caso a(s) vítima(s) se encontre(m) inconsciente(s), é apresentado o ecrã do Apêndice D.7c que questiona o utilizador acerca do estado da respiração da(s) mesma(s). Se esta(s) não respirar(em), então é imediatamente criada uma ocorrência e o canal selecionado será o número nacional de socorro (112) como pode ser visto no ecrã do Apêndice D.7i. Para os casos em que a(s) vítima(s) respire(m), é questionado ao utilizador se existe alguma possibilidade de trauma (Apêndice D.7d) e se sim, e caso o utilizador tenha conhecimentos de SBV, então é demonstrado a indicação

de que a vítima deverá ser colocada em PLS (Apêndice D.7e). Qualquer um destes desfechos indicados anteriormente resultarão também eles numa chamada de emergência para o 112.

De forma a clarificar as alterações no processo implementado referido anteriormente, foi elaborado o esquema observado na Figura 4.10. Como as alterações apenas afetaram o processo desde a Seleção da tipologia de Emergência, o restante esquema não demonstrado nesta figura manter-se-á idêntico ao esquema observado anteriormente na Figura 4.2.

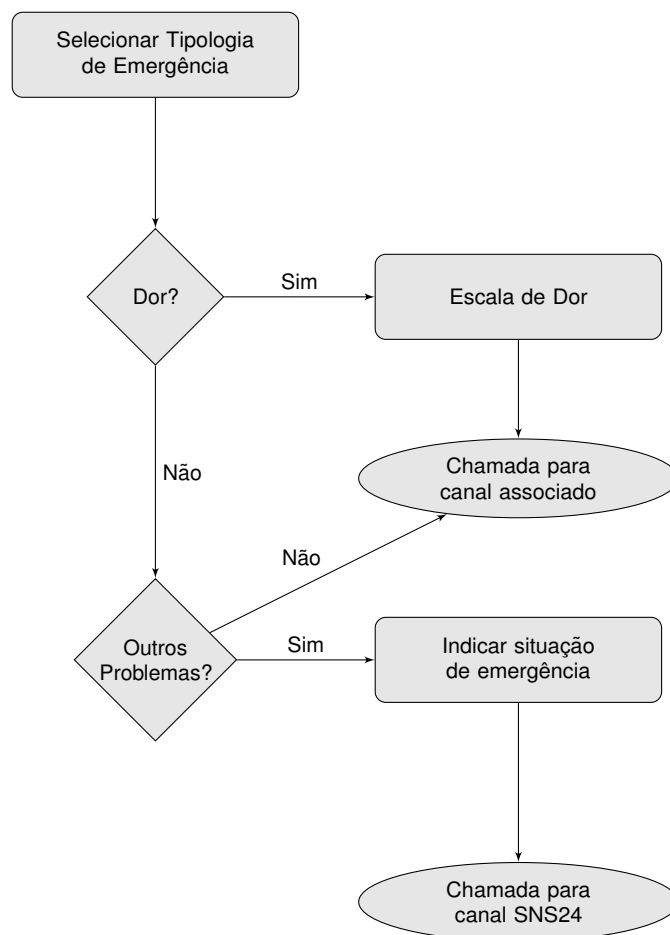


Figura 4.10: Processo de criação de uma ocorrência atualizado.

Ao ecrã de Seleção de Tipologia de ocorrências foi adicionado uma barra de pesquisa no topo, visível no Apêndice D.7f, com o objetivo de facilitar a pesquisa. Foram utilizadas formas circulares sensivelmente maiores que a maioria dos polegares por forma a que o utilizador tivesse uma boa visibilidade de qual o valor que está a escolher. Nesse mesmo apêndice é possível verificar que o valor selecionado foi "Problemas Respiratórios".

Após a seleção de um tipo de ocorrência, é efetuada a validação se esta é do tipo dor. Para os casos em que a tipologia é do tipo dor, é demonstrada uma escala de dor como a apresentada no Apêndice D.7g. Caso a tipologia de ocorrência seja do tipo dor ou outra tipologia que não "Outros problemas" então o utilizador é encaminhado para o canal correspondente à mesma.

Caso o tipo de ocorrência selecionado seja "Outros Problemas", o utilizador é solicitado a introduzir mais detalhe acerca da mesma, como apresentado no Apêndice D.7h, e depois será

encaminhado para o SNS24 (Apêndice D.7j). Esta tipologia é encaminhada para este canal com a justificação de que se for necessário o próprio encaminha a chamada para o 112. Desta forma, consegue-se prevenir o encaminhamento de chamadas desnecessárias.

4.5 Discussão

Considerado todas as práticas utilizadas para a implementação e desenvolvimento da aplicação detalhadas neste capítulo conclui-se que as soluções tecnológicas selecionadas (secção 3.3.2) foram uma opção correta.

Com a elaboração dos wireframes foi possível obter um ponto de partida para praticamente toda a estrutura aplicacional facilitando de imediato o desenvolvimento da mesma. A opção de desenvolvimento do sistema em *DRF* também permitiu evitar um esforço excessivo para a exposição e desenvolvimento de uma *API* prestes a ser consumida.

Onde o ganho de tempo foi bastante visível foi através da utilização da plataforma de *outsystems* que permitiu um desenvolvimento de uma aplicação móvel em cerca de 3/4 semanas poupando o esforço alocado á infraestrutura para a suster, dado que a própria plataforma já disponibiliza todo o ambiente referente a um ambiente de desenvolvimento.

Com toda a aplicação desenvolvida foi ainda possível disponibilizar a mesma como uma pwa de forma a que fosse acedida através de um link ou pela visualização de um *Qr Code*, em qualquer dispositivo móvel sem que fosse necessário a compilação nativa para cada sistema.

Capítulo 5

Validação

Neste capítulo será abordado todo o processo de testes de aceitação da aplicação *SmartCall* realizados com uma amostra de indivíduos selecionada. Dessa forma, serão descritos os protocolos utilizados para a elaboração destes testes de aceitação seguida de uma análise de resultados e discussão dos mesmos.

5.1 Descrição de Protocolos

Após a conclusão do desenvolvimento da aplicação *SmartCall* torna-se pertinente efetuar a validação da mesma, de forma a perceber se os objetivos para os quais esta foi desenvolvida foram atingidos. Uma das práticas comuns para a validação de um determinado *software* é a utilização de intervenientes de forma a perceber se a solução fornecida é adequada à sua utilização [61]. Este método conhecido como Testes de Aceitação de Utilizadores (UAT) é realizado quando todos os testes unitários¹ estão finalizados para que a deteção de erros seja minimizada, dado que esta é muitas vezes considerada como a última fase final de qualquer processo de desenvolvimento.

Em relação aos testes UAT da aplicação desenvolvida, estes foram preparados de forma a que os utilizadores de teste utilizassem a aplicação com a menor informação externa possível. Assim, foi elaborado um documento que pretendia indicar ao utilizador quais os objetivos a atingir durante o período de validação. Isto significa que durante o teste da aplicação o utilizador não receberia mais indicações do que aquelas que a aplicação fornecia e das que estavam presentes no guião aplicacional.

O guião aplicacional, apresentado no Apêndice E, foi construído com o objetivo de ser simples e direto de forma a que os utilizadores obtivessem uma experiência o mais aproximada da realidade possível sem que existisse qualquer influência durante a mesma. Como é possível verificar, este documento é iniciado por uma secção onde é descrito o objetivo do mesmo e uma breve descrição do objetivo da aplicação a validar. A terceira parte está dedicada às instruções de instalação da aplicação, dado que como a aplicação se encontra alojada num ambiente de desenvolvimento

¹ Testes executados pelos programadores durante o processo de desenvolvimento.

foi necessário efetuar uma compilação através do *Outsystems* por forma a que fosse possível partilhar a *PWA* através de um *Qr Code*.

5.1.1 Guião de Validação Aplicacional

Relativamente às ações de teste sugeridas aos utilizadores, estas não possuíam muito detalhe com o objetivo de perceber se a aplicação teria um comportamento intuitivo ou se a mesma não se adequava ao pensamento e funcionamento da maioria das aplicações móveis. Ao indicar apenas os cenários das situações observadas seria possível verificar se existiria alguma dificuldade durante o processo de criação de uma ocorrência para cada um dos cenários. Estes cenários foram criados com o objetivo de:

Cenário 1 A criação desta ocorrência deveria ser simples e direta. O utilizador deveria ser capaz de perceber que a vítima (o próprio) se encontrava consciente e que por essa razão teria de escolher uma tipologia de ocorrência. A tipologia a escolher deveria ser "Dor no Peito" com uma "Dor Moderada" na escala de dor.

Cenário 2 Neste cenário o utilizador ao estar envolvido num acidente de viação deveria conseguir perceber que as vítimas estariam conscientes. A tipologia a selecionar seria "Acidente de carro" e esta ocorrência deveria ter duas vítimas associadas, pelo que, seria necessário a criação de um novo perfil para um segundo indivíduo.

Cenário 3 No último cenário é transmitida a informação de que a vítima está inconsciente, pelo que, nestes casos o importante é que a resposta à vítima seja o mais rápido possível de forma a evitar o máximo de sequelas possíveis. Por essa razão, assim que é respondido que a vítima está inconsciente, estando ou não a respirar, é aberta a ocorrência e despoletada o restante processo de gestão com o 112.

Por último, foi solicitado aos utilizadores de teste que realizassem o preenchimento de um questionário (Apêndice G) que seria utilizado para avaliar a experiência de validação com a própria aplicação de modo a ser possível retirar conclusões acerca das mesmas.

Para questões legais, de confidencialidade e para garantir o consentimento de análise dos dados inseridos pelos utilizadores foi ainda requisitado a estes o preenchimento de uma declaração em que a sua minuta pode ser observada no Apêndice F.

5.1.2 Questionário de Validação Aplicacional

Como referido anteriormente foi elaborado um questionário com o objetivo de apurar a eficiência da aplicação na realidade social atual. Através deste questionário tentou apurar-se se o modo como a aplicação se encontrava desenvolvida refletia as necessidades de um indivíduo numa situação de emergência. A composição deste questionário consiste em:

Informação demográfica A captação deste tipo de informação é relevante dado que pode possibilitar a perceção de padrões relacionados com o comportamento dos utilizadores

durante a experiência de validação da aplicação. As características da amostra de indivíduos que participaram neste questionário consiste na seguinte informação:

- Idade;
- Género;
- Formação académica;
- Profissão;
- Anos de atividade profissional.

A solicitação desta informação pode ser observada no Apêndice G.1a e G.1b.

Confirmação de sucesso de ações Como forma inicial para distinção imediata da experiência de cada um dos utilizadores foi necessário perceber se os mesmos conseguiram efetuar com sucesso cada uma das ações propostas. Assim foi adicionado as seguintes questões acerca de cada uma das ações solicitadas;

- Conseguiu registar o perfil de utilizador?
- Conseguiu registar outro(s) perfis na aplicação?
- Conseguiu registar a ocorrência associada ao Cenário 1?
- Conseguiu registar a ocorrência associada ao Cenário 2?
- Conseguiu registar a ocorrência associada ao Cenário 3?

Estas questões, incorporadas ao longo de todo o questionário, apenas possibilitariam a introdução uma resposta de "Sim" ou "Não".

Escala de Likert De modo a que fosse possível averiguar a usabilidade da aplicação, foi introduzido no questionário duas questões que permitiam, com recurso a uma escala *Likert* que é uma das técnicas mais utilizadas para concluir o nível de compromisso do utilizador [62], perceber o nível de concordância dos inquiridos em relação a:

- Facilidade de realização de cada ação na aplicação;
- Integrabilidade da aplicação em contexto real.

Comentários Como forma de possibilitar alguma justificação foi acrescentado vários campos onde os utilizadores tinham a oportunidade de exprimir qualquer informação que considerariam relevantes consoante o que estava a ser analisado no momento.

5.2 Análise de Resultados

Os protocolos anteriormente apresentados foram colocados em execução numa amostra de indivíduos de idades e contextos sociais distintos. Neste subcapítulo será efetuado uma caracterização demográfica dos utilizadores que participaram nesta validação seguida de uma análise dos resultados obtidos baseados na experiência dos mesmos.

5.2.1 Caraterização demográfica

A amostra de utilizadores que participaram na fase de validação da aplicação foram compostos por 53% de indivíduos do género feminino e 47% do género masculino como pode ser observado na Figura 5.1. Segundo as estatísticas provisórias elaboradas em 2021 por PORDATA, em Portugal existem cerca de 5.423.632 indivíduos do género feminino e 4.921.170 do género masculino, o que equivale respetivamente a 52% e a 48%. Com base nesta estatística, a distribuição da amostra de utilizadores pelo género pode ser considerada como adequada à realidade vivida em Portugal.

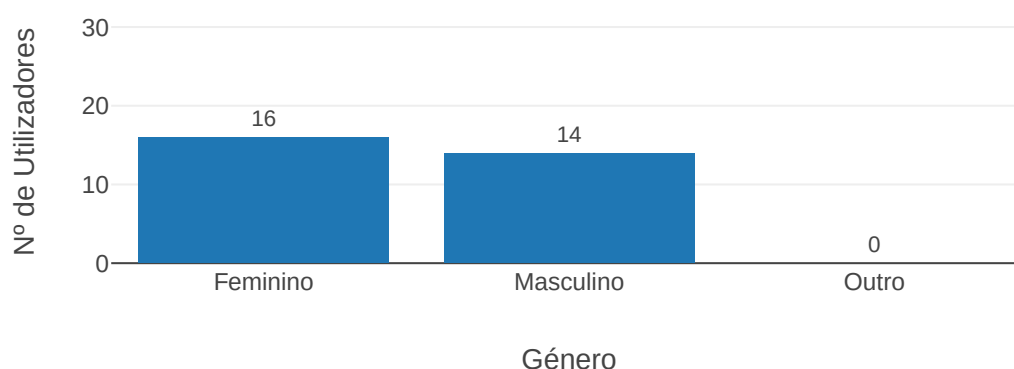


Figura 5.1: Género dos utilizadores de teste.

Relativamente à distribuição da amostra de utilizadores pelos grupos etários foi elaborado o histograma presente na Figura 5.2 onde é possível verificar que 10% da amostra total de utilizadores (3 indivíduos) são menores de idade. Os grupos etários de idades compreendidas dos 18 a 20, 31 a 40 e 41 a 50 anos são também eles compostos por uma amostra de 10% cada. Os restantes 18 utilizadores estão concentrados no grupo etário entre os 21 e 30 anos de idade e correspondem a 60% da amostra total.

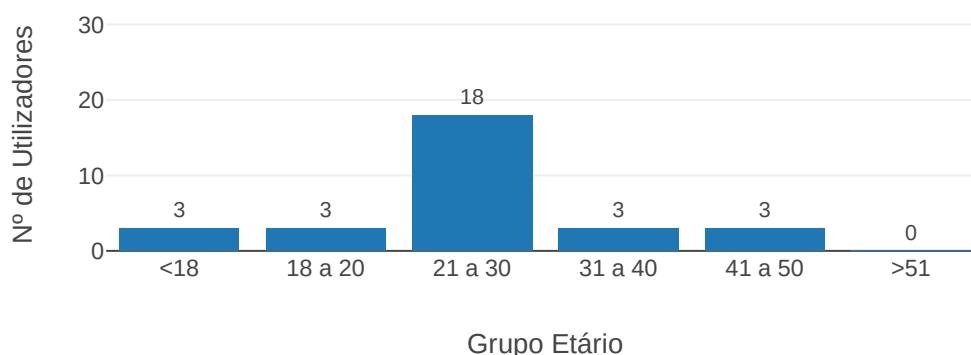


Figura 5.2: Grupo Etário dos utilizadores de teste.

No que toca ao grau académico da amostra de utilizadores de teste, esta encontra-se demonstrada na Figura 5.3. É possível verificar que apenas 1 utilizador de teste se encontra no Ensino Básico, o que corresponde a uma percentagem de 3,33%, enquanto 30% dos utilizadores

reportam possuir o Ensino Secundário. O grau académico que possui maior representação é a Licenciatura, estando esta composta por 12 indivíduos que representam 40% na amostra total. Relativamente a Mestrado, 7 utilizadores relatam possuir este grau resultando assim em 23,3% da amostra, e apenas 1 utilizador comunicou possuir doutoramento.

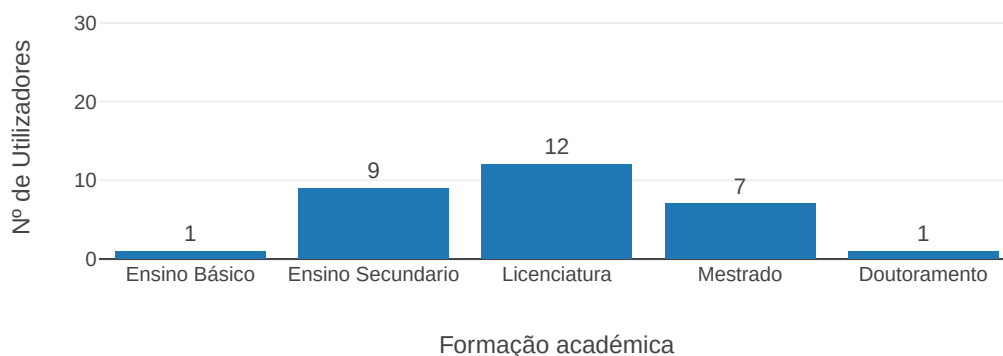


Figura 5.3: Grau académico dos utilizadores de teste.

Em relação ao setor profissional em que os utilizadores se encontram, estes foram analisados e categorizados com base em respostas abertas do questionário. Dessa forma foi elaborado o gráfico presente na Figura 5.4 que se encontra dividido em 5 setores: Educação, Estudantes, Saúde, Serviços e Tecnologia. No setor de Educação está incluído 2 Educadoras de Infância, 1 Assistente Operacional Camarária e um professor que perfazem um total de 13,3% da amostra. Incluído ainda na amostra, existe 7 estudantes que correspondem a 23,3% da mesma, não existindo informação extra acerca da área de estudos dos mesmos. Relativamente ao setor da Saúde foram contempladas 2 utilizadoras, uma Enfermeira e a uma Bombeira, correspondendo a 6,7% do total. A representar o setor de Serviços existem 4 utilizadores de diversas áreas profissionais que compõem 13,3% da amostra total. No setor Tecnológico observa-se a maior frequência com 43,3% resultante da associação de 13 utilizadores a este setor.

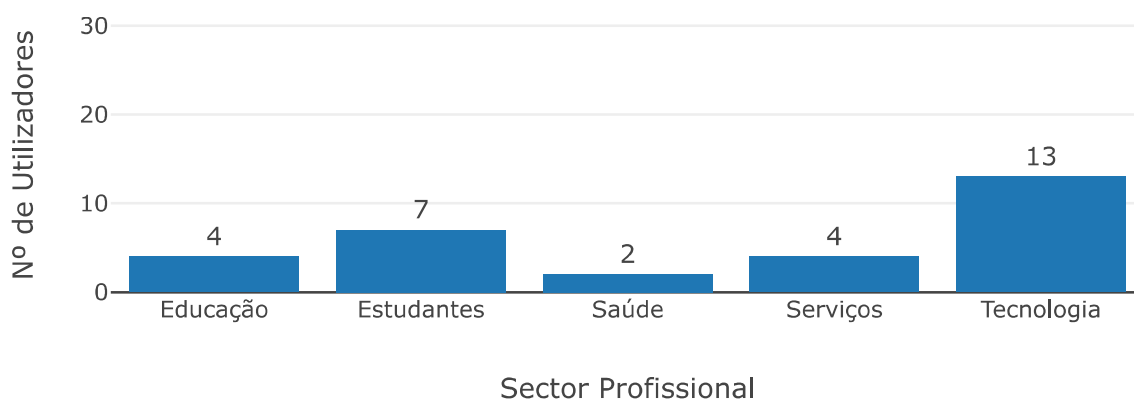


Figura 5.4: Setor Profissional dos utilizadores de teste.

5.2.2 Resultados obtidos

Ao analisar os dados inseridos no questionário em relação ao sucesso da realização das ações de testes solicitadas foi possível elaborar a Tabela 5.1. Esta tabela está construída de forma a que seja possível verificar com clareza o número de utilizadores que conseguiram concluir determinada com sucesso e os que não conseguiram.

Tabela 5.1: Resultados das confirmações de sucesso de ações de teste.

Ação	Sim	Não
Conseguiu registar o perfil de utilizador?	30	0
Conseguiu registar outro(s) perfis na aplicação?	27	3
Conseguiu registar a ocorrência associada ao Cenário 1?	30	0
Conseguiu registar a ocorrência associada ao Cenário 2?	26	4
Conseguiu registar a ocorrência associada ao Cenário 3?	24	6

Relativamente à primeira ação de teste, o processo de Registo do Utilizador, todos os inquiridos conseguiram executar a ação com sucesso, mostrando ainda algum apreço acerca da intuitivo do mesmo nos comentários associados a esta questão.

No que diz respeito ao processo de registo de outros perfis, 3 utilizadores, ou seja, 10% da amostra total relata que obteve dificuldades a efetuar o registo de perfis associados a familiares. Das informações extra indicadas, é possível associar esta dificuldade à falta de compreensão do *workflow* da aplicação, pois 2 destes utilizadores reportaram que a criação de um novo perfil deveria ser realizada aquando do registo do utilizador padrão e não antes de realizar um pedido de ajuda.

Na realização da abertura da ocorrência associada ao Cenário 1, todos os utilizadores reportaram sucesso pelo que não existiu nenhum comentário associado a esta ação de teste.

Em relação à realização de uma ocorrência associada ao Cenário 2, 13,3%, ou seja, 4 utilizadores reportaram não ter conseguido efetuar a mesma com sucesso. A razão apresentada por alguns destes utilizadores como justificação para a dificuldade verificada está relacionada com a criação de outros perfis, sendo que, 1 do 4 utilizadores não apresentou qualquer motivo para o insucesso deste cenário.

No Cenário 3 verifica-se a maior taxa de insucesso das ações de teste. 20% da amostra de utilizadores indica que não conseguiu realizar este cenário com sucesso, o que equivale a 6 utilizadores. Pelos comentários introduzidos, a razão pela qual não foi possível efetuar a criação desta ocorrência está relacionada com o facto de ao indicar que a vítima se encontra inconsciente a aplicação não permite a introdução da tipologia de emergência, não tendo por isso ter sido possível dar a indicação de que se tratava de um acidente de viação.

Relativamente à facilidade de utilização com base nas funcionalidades da aplicação, foi elaborado o gráfico apresentado na Figura 5.5. Esta representação gráfica permite observar a percentagem de utilizadores consoante a sua concordância pelas diferentes ações realizadas na

fase de validação da própria aplicação.

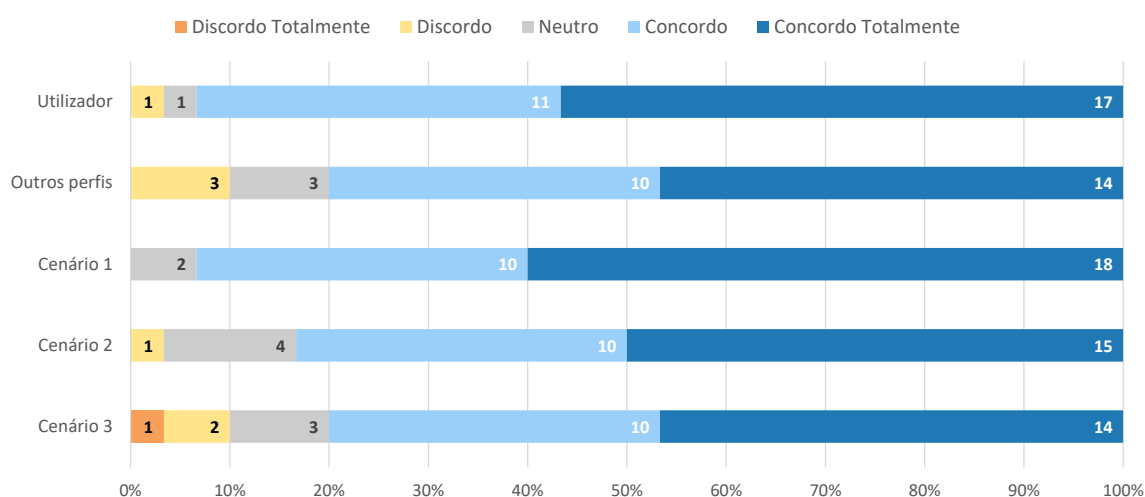


Figura 5.5: Usabilidade associada às funcionalidades da aplicação *SmartCall*.

Como é possível observar, mais de 90% dos utilizadores concordaram que o registo do utilizador é um processo fácil de realizar. Apenas 1 utilizador (3,3%) discordou da usabilidade deste processo, sendo que não indicou nenhuma razão que fundamenta a sua discordância para com esta funcionalidade. Não revelando também a razão pela qual não concorda com esta característica, 1 utilizador reporta que não concorda mas que também não discorda que o registo do utilizador é de fácil utilização, assumindo assim uma posição neutra no gráfico acima apresentado.

Analizando os resultados em relação à funcionalidade de criação de outros perfis é possível verificar que 80% dos utilizadores concordam com a facilidade de utilização da mesma, sendo que 43,3% dizem concordar totalmente. 10% dos utilizadores demonstraram que não concordavam nem discordavam e os restantes 10% de utilizadores discordam da facilidade deste processo. Foi possível ainda confirmar que os 3 utilizadores que discordaram da facilidade associada ao processo de registo de perfis foram os mesmos utilizadores que não conseguiram completar a ação de teste com sucesso.

Na criação da ocorrência associada ao Cenário 1 verifica-se a maior taxa de concordância. 93,3% dos utilizadores revelam concordar com a facilidade de efetuar o pedido de ajuda solicitado, sendo que 60% afirmam concordar totalmente. Nesta funcionalidade nenhum utilizador apresentou discordância, pelo que os restantes 6,7% optaram por se manifestar com uma opinião neutra.

Ao solicitar um pedido de ajuda associada ao Cenário 2, 83,3% dos utilizadores manifestaram concordar com a usabilidade sendo que 50% afirmaram que concordavam totalmente. 13,3% dos utilizadores não se comprometeram com a sua concordância, pelo que estão incluídos na representação neutra. Apenas um utilizador apresentou discordar da facilidade de submissão da situação solicitada, correspondendo assim a uma taxa de 3.3%.

No última barra do gráfico observa-se que 3 dos utilizadores afirmaram não concordar com a facilidade de introdução do Cenário 3, onde 3,33% dos utilizadores totais discordam totalmente. 10% dos utilizadores relatam não concordar nem discordar. 80% dos utilizadores concordam que

a introdução do Cenário 3 foi de fácil realização, tendo 46,7% concordado totalmente.

Com os resultados obtidos foi ainda elaborado a Tabela 5.2 onde é possível observar a mediana calculada para cada um dos valores aglomerados na Figura 5.5.

Tabela 5.2: Mediana das respostas de usabilidade associada a cada funcionalidade.

Ação	Mediana	Descrição
Registo do perfil do utilizador	5	Concordo Totalmente
Registo de outros perfis	4	Concordo
Registo do Cenário 1	5	Concordo Totalmente
Registo do Cenário 2	4	Concordo
Registo do Cenário 3	4	Concordo

A utilização do cálculo da mediana torna-se relevante dado que esta representa a tendência central da amostra de dados. Um valor alto associado a esta métrica reflete que os dados recolhidos se concentram maioritariamente em valores também eles altos, enquanto que um valor baixo da mesma significa que a maioria dos utilizadores demonstraram uma discórdia com a usabilidade das funcionalidades.

Como se pode verificar, com estas medianas é possível afirmar que as ações de registo do Utilizador e do Cenário 1 foram ações relativamente práticas e simples de executar dado que a centralização da amostra de dados se focou no Concordo Totalmente. No que diz respeito às ações de registo de Outros Perfis, do Cenário 2 e do Cenário 3, a amostra central de dados concentrou-se um pouco mais abaixo dado que se verificou um menor número de utilizadores totalmente em concordância com a usabilidade da aplicação nestas ações de teste.

Relativamente à outra questão qualitativa apresentada no questionário de validação, foi elaborado o gráfico apresentado na Figura 5.6.

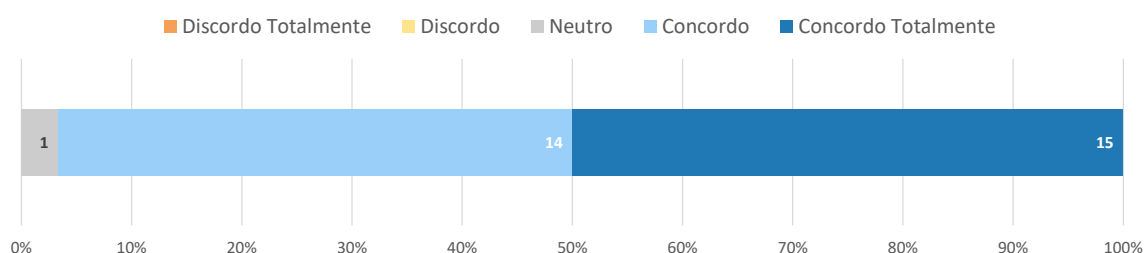


Figura 5.6: Integrabilidade da aplicação *SmartCall* em contexto real.

No gráfico acima é possível observar que 96,7%, ou seja, 29 dos utilizadores concordam que a aplicação desenvolvida neste projeto poderá ser útil num contexto real, sendo que 50% afirmam concordar totalmente com a mesma. É possível observar ainda que apenas 1 utilizador demonstrou não concordar, porém este afirma também que não discorda da integrabilidade da aplicação na corrente sociedade.

5.2.3 Discussão

Após análise dos resultados foi possível concluir que a maioria das ações que os utilizadores reportaram não conseguir realizar com sucesso deveu-se a uma perceção errada do *workflow* da aplicação. Num contexto real, o registo de perfis pode ser efetuado em qualquer altura desde que o utilizador padrão tenha sessão iniciada na aplicação pelo que a dificuldade demonstrada nesta funcionalidade não tem um impacto muito significativo. O facto da aplicação não solicitar a tipologia de emergência nas situações em que as vítimas estão inconscientes deve-se ao facto deste estado já ser considerado um estado de emergência e, por essa razão, não existe a necessidade de validar a criticidade da situação observada. Em relação à discordância apresentada à facilidade do registo do utilizador, considerando que todos os utilizadores conseguiram realizar o próprio registo, é possível verificar que existem oportunidades para tornar este processo mais simples e intuitivo.

De forma a colmatar as discordâncias apresentadas, poderá recorrer-se ao tutorial inicial da aplicação para clarificar de um modo mais objetivo as questões acima identificadas pelos utilizadores durante a fase de validação.

Relativamente à taxa de aceitação da aplicação desenvolvida, foi possível verificar que a mesma foi bastante positiva. Esta taxa de aceitação por parte dos utilizadores revela o interesse por aplicações móveis que poderão, de certa forma, ajudar a otimizar processos que atualmente utilizamos no nosso quotidiano. Para justificar o interesse demonstrado pelos utilizadores, foi possível analisar alguns comentários dos mesmos, onde foi mencionado que a utilização da aplicação poderia ser bastante benéfica em relação ao tempo de chegada dos TEPH ou que a mesma, em conjunto com o próprio sistema desenvolvido (SIREPH), poderia vir a colmatar a dificuldade de persistência de informação relativa ao histórico de pacientes entre as diversas entidades relacionadas.

Capítulo 6

Conclusões

Este capítulo relata quais os objetivos cumpridos com o projeto desenvolvido e quais os próximos passos a completar enquanto trabalho futuro proposto.

6.1 Cumprimento de Requisitos

De forma a perceber quais os objetivos cumpridos com o desenvolvimento do projeto descrito no presente documento, foram elaboradas as Tabelas 6.1 e 6.2. Na Tabela 6.1 é possível observar o grau de satisfação de cada requisito funcional.

Tabela 6.1: Grau de satisfação de requisitos funcionais.

	Descrição	Grau de Satisfação
RF1	Restrição de Uso	Satisfeito
RF2	Informação clínica do utilizador	Satisfeito
RF3	Registo de outros perfis	Satisfeito
RF4	Registo de uma ocorrência	Satisfeito
RF5	Seleção da tipologia da emergência	Satisfeito
RF6	Seleção de vítimas registadas	Satisfeito
RF7	Seleção de vítimas não registadas	Não Satisfeito
RF8	Deteção de localização	Satisfeito
RF9	Realização de chamadas	Parcialmente Satisfeito
RF10	Listagem de ocorrências	Satisfeito

Em relação ao requisito RF1, este foi satisfeito com a limitação do acesso à informação da aplicação através do registo do utilizador. Ao restringir a utilização da aplicação por sessão tornou-se sempre possível perceber qual o responsável pela criação de determinada informação.

A persistência da informação clínica do utilizador foi garantida de forma a que a mesma pudesse sofrer alterações sempre que necessário, possibilitando que a mesma estivesse sempre atualizada, satisfazendo assim totalmente o requisito RF2.

Ao permitir que fosse possível adicionar novos perfis de possíveis vítimas, satisfaz-se o requisito RF3. A transmissão da informação relevante de uma ocorrência para o sistema SIREPH, que outrora era solicitada ao utilizador via chamada telefónica, com a utilização da aplicação esta informação passa a ser enviada através de comunicações de serviços. Isto permite um elevado ganho de tempo pois ao enviar toda a informação associada aos perfis das vítimas não existe a necessidade de solicitar esta mesma informação durante a chamada com o CODU. É assim possível afirmar que a integração da aplicação no sistema SIREPH permite uma otimização da operacionalidade do próprio sistema.

Realizando o processo de pedido de ajuda da aplicação com sucesso, é possível a criação de uma ocorrência como resultado do mesmo. Este desfecho permite então satisfazer o requisito RF4. Ainda durante este processo, e graças ao modelo de triagem construído, é selecionada a tipologia da ocorrência com base num conjunto de características observadas em cada situação e as vítimas associadas à mesma. Estas duas características permitem satisfazer os requisitos RF5 e RF6.

O requisito RF7, tal como decidido e indicado durante a fase de Análise 4.1.3, não foi incluído na fase de desenvolvimento dado que a sua implementação não possuía uma prioridade alta. Assim, este requisito não foi satisfeito com o desenvolvimento do corrente projeto, tendo no entanto sido considerado durante o processo de desenho, de forma a facilitar a sua implementação no futuro.

Na informação partilhada é enviada também a localização da ocorrência obtida através do sensor de *GPS* que é o objetivo do requisito RF8. Isto permite, não só satisfazer este requisito, mas ainda diminuir as dificuldades que os meios de socorro poderiam encontrar com a informação de uma morada não exata, favorecendo assim os tempos de resposta.

Com base no requisito RF9, para realizar uma chamada de emergência é necessário a seleção de um canal associado a cada tipologia de ocorrência. A aplicação desenvolvida permite esta associação, retornando o respetivo canal assim que a tipologia da emergência seja selecionada. A razão pela qual este requisito foi apenas parcialmente satisfeito, está relacionada com o facto da aplicação não realizar a chamada de emergência real, dado que não foi possível contemplar esta funcionalidade nos cenários de teste.

Através da aplicação e, de forma a satisfazer o requisito RF10, é possível o utilizador observar uma listagem de todas as ocorrências a ele associadas e o detalhe das mesmas.

Na Tabela 6.2 é possível observar a associação dos requisitos não funcionais definidos para a aplicação com o seu grau de satisfação após a conclusão da mesma.

O requisito de responsividade foi satisfeito através da utilização da tecnologia *Outsystems*. Com recurso a uma *PWA* foi então possível garantir uma interface geral e adaptada a qualquer ecrã.

Em relação ao requisito RNF2 foi possível garantir uma interface simples e limpa de forma a que se utilizada numa situação de stress não resultasse na introdução de informação errada. Foram assim utilizados botões grandes com texto simples e claro para permitir uma imediata

Tabela 6.2: Grau de satisfação de requisitos não funcionais.

	Descrição	Grau de Satisfação
RNF1	Responsividade	Satisfeito
RNF2	Usabilidade	Satisfeito
RNF3	Disponibilidade	Parcialmente Satisfeito
RNF4	Segurança	Parcialmente Satisfeito
RNF5	Integrabilidade	Satisfeito
RNF6	Integridade	Parcialmente Satisfeito
RNF7	Reatividade	Satisfeito

correta interpretação do mesmo. Os resultados da validação com utilizadores (ver Capítulo 5) suportam a satisfação deste requisito através da elevada taxa de concordância com a usabilidade da aplicação.

Relativamente à disponibilidade da aplicação definida no requisito RNF3, esta encontra-se parcialmente satisfeita dado que a aplicação se encontra alojada num ambiente de desenvolvimento e validação. De forma a garantir a total satisfação deste requisito seria necessário investir num servidor de produção para alojar a aplicação *Outsystems*.

O sistema construído suporta algumas políticas de segurança, porém seria necessário avaliar de forma mais profunda todo o sistema de forma a que fosse possível garantir a totalidade de satisfação do requisito RNF4.

Uma das principais funcionalidades disponibilizada pela aplicação desenvolvida é a sua integração com um sistema que permite a persistência e acompanhamento da informação de cada paciente. A transmissão desta informação permite que todas as entidades envolvidas num processo de emergências consigam aceder e atualizar a mesma, satisfazendo assim o propósito do requisito RNF5. De forma a garantir a integridade de toda a informação envolvida neste sistema seria necessário uma maior dedicação à arquitetura de todos os módulos envolvidos no mesmo. Só com essa atenção é que seria possível garantir uma satisfação total do requisito RNF6.

Através da utilização da plataforma *Outsystems*, foi ainda utilizado o recurso a computação reativa disponibilizado pela mesma, que permitiu garantir a satisfação do requisito RNF7 sem que lhe tenha sido oferecido uma prioridade elevada.

6.2 Processo de Engenharia de Software

Todo o processo de engenharia envolvido na realização deste projeto culminou na validação da aplicação por parte de uma amostra de utilizadores. Com base nos resultados obtidos através do questionário de validação da aplicação foi possível verificar uma elevada taxa de aceitação por parte dos mesmos, apesar de o planeamento inicial ter sofrido algumas alterações devido à acumulação de tarefas. Esta fase de validação permitiu ainda observar o notório interesse revelado pelos utilizadores em aplicações que possam integrar sistemas de saúde e, em última

instância, ajudar a qualidade e o acesso dos mesmos.

Com a conclusão deste projeto, a utilização da tecnologia *Outsystems* revelou-se uma grande vantagem, dado que permitiu o desenvolvimento rápido da aplicação. Utilizada em comum com uma metodologia *agile*, que permite que sejam efetuadas alterações em qualquer fase do projeto, resultando assim numa enorme escalabilidade associada à aplicação desenvolvida. Os resultados verificados na fase de validação são resultado do potencial disponibilizado por esta tecnologia, pois considerando o tempo de desenvolvimento de 3/4 semanas, com uma tecnologia não *low code*, seria mais difícil satisfazer os mesmos requisitos, com a mesma taxa de aceitação por parte dos utilizadores.

Relativamente aos objetivos para os quais a aplicação foi desenvolvida, como a diminuição do número de chamadas consideradas como falsas emergências, não foi possível obter resultados quantitativos que pudessem ser alvo de análise, dado que a amostra disponível para testes dos mesmos era limitada. Num contexto real, sugere-se que a associação de diferentes canais às tipologias de ocorrências pré-definidas na aplicação, considerando que a informação introduzida pelo utilizador é exata, pode resultar numa diminuição de ocorrências reportadas para o canal errado. Eventualmente, esta associação permitirá, por si só, uma redução do tempo de resposta dos meios ou entidades associadas às emergências, dado que o indivíduo que solicita o pedido de ajuda não perde tempo a ser reencaminhado entre chamadas.

Este processo de engenharia de *software* serve de exemplo de como projetos na área da tecnologia podem afetar o quotidiano de uma sociedade de forma positiva. Em última instância, se bem aplicada e, em comunhão com o sistema desenvolvido, esta aplicação pode ajudar a salvar vidas dado que o tempo de resposta em determinadas situações é a diferença entre a vida e a morte das vítimas.

6.3 Trabalho Futuro

Em alguns dos comentários elaborados pelos utilizadores de teste foram identificadas algumas sugestões de alteração de interface que poderiam ajudar a melhorar a perceção e funcionamento da aplicação. Desta forma, como trabalho futuro, a análise e implementação dessas mesmas sugestões serão consideradas como relevantes.

Os requisitos funcionais identificados como não satisfeitos ou satisfeitos parcialmente no subcapítulo 6.2, que não foram considerados para o desenvolvimento inicial da aplicação *SmartCall* devem também ser considerados como trabalho a realizar no futuro. Idealmente esta aplicação deveria possibilitar a abertura de ocorrências para todos os civis, mesmo aqueles que não estejam associados com o utilizador padrão. Desta forma, seria possível concentrar o uso da aplicação para todas as situações que poderiam acontecer no quotidiano.

Seria importante realizar testes de utilizador no contexto de um simulacro da proteção civil. Com a realização de testes nestas circunstâncias, seria possível perceber outro tipo de métricas que não foram possíveis avaliar na fase de validação realizada. Se realizado um cenário

de simulacro, seria possível obter dados acerca do tempo útil para efetuar um pedido de ajuda, que poderiam ser cruzados com a informação da duração dos processos atuais. Ao realizar esta comparação seria então possível afirmar qual o impacto direto no tempo de solicitação de ajuda através da aplicação, podendo posteriormente usar o mesmo como argumento diferenciador para a recomendação da utilização da aplicação numa sociedade.

Bibliografia

- [1] M. da Saúde. Relatório de contas do ministério da saúde e do sns 2020. *Relatório de Contas*, 2020. URL https://www.acss.min-saude.pt/wp-content/uploads/2016/10/Relatorio_Contas_MS_SNS_2020.pdf.
- [2] I. G. das Atividades em Saúde (IGAS). Auditoria à capacidade de resposta em situações de emergência. *IGAS Newsletter*, 2, 2 2020. URL https://www.igas.min-saude.pt/wp-content/uploads/2022/02/Newsletter_-IGAS_fevereiro_2022.pdf.
- [3] J. Massano. Dia mundial do cérebro 2017 dedicado ao AVC. *Promoting global neurological education and training*, 7 2017. URL <https://wfneurology.org/2017-07-17-wfn-spn-wbd>.
- [4] S. N. de Saúde (SNS). O que fazer em caso de emergência?, 10 2017. URL <https://www.inem.pt/2017/10/23/o-que-fazer-em-caso-de-emergencia-2/>.
- [5] R. Portuguesa. Orçamento prevê aumento do financiamento permanente para bombeiros, 10 2021. URL <https://www.portugal.gov.pt/pt/gc22/comunicacao/comunicado?i=orcamento-preve-aumento-do-financiamento-permanente-para-bombeiros>.
- [6] R. Cipriano. Como é que é ser bombeiro em Portugal? *Observador*, 8 2016. URL <https://observador.pt/2016/08/11/como-e-que-e-ser-bombeiro-em-portugal/>.
- [7] Lusa. Associação pede criação de centrais unificadas de emergência de âmbito regional, 2 2021. URL <https://saudemais.tv/noticia/28916>.
- [8] G. Lawton. Making business intelligence more useful. *Computer*, 39:14–16, 9 2006. ISSN 0018-9162. doi: 10.1109/MC.2006.318.
- [9] S. Mehrotra, X. Qiu, Z. Cao, e A. Tate. Technological challenges in emergency response [guest editors' introduction]. *IEEE Intelligent Systems*, 28:5–8, 7 2013. ISSN 1541-1672. doi: 10.1109/MIS.2013.118.
- [10] INEM. Dashboard de indicadores de desempenho do inem, 2021. URL <https://extranet.inem.pt/stats/dashboard/>.
- [11] V. Patil, S. Panicker, e M. Kv. Use of agile methodology for mobile applications. *International Journal of Latest Technology in Engineering, Management and Applied Science*, V, 11 2016.

- [12] L. Adkins. *Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition: Addison-Wesley Signature Series - Cohn*. Pearson Education, Inc., 12 2021. ISBN 0321637704.
- [13] S. Nakazawa, K. Komatsu, T. Tanaka, e K. Matsumoto. Development and evaluation of large-screen digital kanban with smartphone operation. In *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 295–300, 2017. doi: 10.1109/IIAI-AAI.2017.151.
- [14] M. Wirz, D. Roggen, e G. Troster. Decentralized detection of group formations from wearable acceleration sensors. In *2009 International Conference on Computational Science and Engineering*, volume 4, pages 952–959, 2009. doi: 10.1109/CSE.2009.199.
- [15] A. Al-Akkad e A. Zimmermann. User study: Involving civilians by smart phones during emergency situations. *ISCRAM 2011*, 2011.
- [16] G. Marktest. Utilização de smartphone em crescimento, 2020. URL <https://www.marktest.com/wap/a/n/id~2767.aspx>.
- [17] Statista. Forecast of smartphone user numbers in portugal from 2015 to 2022, 2022. URL <https://www.statista.com/statistics/566177/predicted-number-of-smartphone-users-in-portugal/>.
- [18] PORDATA. População residente de portugal, 2022. URL <https://www.pordata.pt/Portugal/>.
- [19] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, e Srivastava. Participatory sensing. In *Worldwide Sensor Web Workshop*, 2006.
- [20] P. P. Jayaraman, K. Gunasekera, F. Burstein, P. D. Haghighi, H. S. Soetikno, e A. Zaslavsky. An ontology-based framework for real-time collection and visualization of mobile field triage data in mass gatherings. In *2013 46th Hawaii International Conference on System Sciences*, pages 146–155, 2013. doi: 10.1109/HICSS.2013.92.
- [21] K. Hu, A. Rahman, V. Sivaraman, e P. Ray. Improving air pollution forecast with ubiquitous mobile sensor network. In *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 405–406, 2014. doi: 10.1109/ICUFN.2014.6876823.
- [22] S.-Y. Chiou e Z.-Y. Liao. A real-time, automated and privacy-preserving mobile emergency-medical-service network for informing the closest rescuer to rapidly support mobile-emergency-call victims. *IEEE Access*, 6:35787–35800, 2018. doi: 10.1109/ACCESS.2018.2847030.
- [23] V. Liutkauskas, D. Matulis, e R. Pleštys. Location based services. In *Elektronika Elektrotechnika*, volume 52, 2004.

- [24] S. M. Errapotu, J. Loveless, R. Yu, S. Ren, M. Pan, e Z. Han. Privacy preserving clock auction for emergency demand response in colocation data centers. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, 2017. doi: 10.1109/ICC.2017.7997114.
- [25] S. B. Edillo, P. J. E. Garrote, L. C. C. Domingo, A. G. Malapit, e B. S. Fabito. A mobile based emergency reporting application for the philippine national police emergency hotline 911: A case for the development of i911. In *2017 Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, pages 1–4, 2017. doi: 10.23919/ICMU.2017.8330110.
- [26] S. Foundation. Sahana foundation - open source disaster management solutions, 2022. URL <https://sahanafoundation.org/>.
- [27] M. Careem, C. De Silva, R. De Silva, L. Raschid, e S. Weerawarana. Sahana: Overview of a disaster management system. In *2006 International Conference on Information and Automation*, pages 361–366, 2006. doi: 10.1109/ICINFA.2006.374152.
- [28] N. H. Taba e A. S. Khatavakhotan. It system development via project management: A phenomenological and statistical study. In *2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, pages 1–3, 2017. doi: 10.1109/ICSIMA.2017.8311995.
- [29] N. Bushuyeva, D. Bushuiev, V. Busuieva, e I. Achkasov. It projects management driving by competence. In *2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, volume 2, pages 226–229, 2018. doi: 10.1109/STC-CSIT.2018.8526680.
- [30] H. Garg e M. Dave. Securing iot devices and securelyconnecting the dots using rest api and middleware. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–6, 2019. doi: 10.1109/IoT-SIU.2019.8777334.
- [31] D. Gourley, B. Totty, M. Sayer, A. Aggarwal, e S. Reddy. *HTTP: The Definitive Guide*. Definitive Guides. O'Reilly Media, Incorporated, 2002. ISBN 9781565925090. URL <https://books.google.pt/books?id=XrhQAAAAAAAJ>.
- [32] Swagger. Swagger documentation, 2022. URL <https://swagger.io/docs/specification/about/>.
- [33] S. Malik e D.-H. Kim. A comparison of restful vs. soap web services in actuator networks. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 753–755, 2017. doi: 10.1109/ICUFN.2017.7993893.
- [34] D. R. Framework. Django rest framework documentation, 2022. URL <https://www.django-rest-framework.org/>.
- [35] D. R. Framework. Django rest framework authentication, 2022. URL <https://www.django-rest-framework.org/api-guide/authentication/>.

- [36] PostgreSQL. Postgresql documentation, 2022. URL <https://www.postgresql.org/docs/>.
- [37] M.-G. Jung, S.-A. Youn, J. Bae, e Y.-L. Choi. A study on data input and output performance comparison of mongodb and postgresql in the big data environment. In *2015 8th International Conference on Database Theory and Application (DTA)*, pages 14–17, 2015. doi: 10.1109/DTA.2015.14.
- [38] D. Fortunato e J. Bernardino. Progressive web apps: An alternative to the native mobile apps. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6, 2018. doi: 10.23919/CISTI.2018.8399228.
- [39] R. Martins, F. Caldeira, F. Sá, M. Abbasi, e P. Martins. An overview on how to develop a low-code application using outsystems. In *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, pages 395–401, 2020. doi: 10.1109/ICSTCEE49637.2020.9277404.
- [40] Outsystems. Outsystems documentation, 2022. URL <https://success.outsystems.com/Documentation>.
- [41] Brooks. No silver bullet essence and accidents of software engineering. *Computer*, 20:10–19, 4 1987. ISSN 0018-9162. doi: 10.1109/MC.1987.1663532.
- [42] L. Macaulay e J. Mylopoulos. Requirements engineering: An educational dilemma. *Automated Software Engineering*, 2:343–351, 12 1995. ISSN 0928-8910. doi: 10.1007/BF00871804.
- [43] . Iso/iec/ieee international standard - systems and software engineering – life cycle processes – requirements engineering. *ISO/IEC/IEEE 29148:2018(E)*, pages 1–104, 2018. doi: 10.1109/IEEESTD.2018.8559686.
- [44] B. John. Software requirements. *TECHNICAL REPORT SEI CURRICULUM MODULE, SEI-CM-19-1.2 SOFTWARE ENGINEERING INSTITUTE, CARNEGIE MELLON UNIVERSITY USA*, 1990.
- [45] S. S. Yau e J. J.-P. Tsai. A survey of software design techniques. *IEEE Transactions on Software Engineering*, SE-12(6):713–721, 1986. doi: 10.1109/TSE.1986.6312969.
- [46] C. V. Ramamoorthy, A. Prakash, W. T. Tsai, e Y. Usuda. Software engineering: Problems and perspectives. *Computer*, 17:191–209, 1984.
- [47] M. J. Hamm. *Wireframing Essentials*. Packt Publishing, 2014.
- [48] S. Uzayr. *Mastering UI Mockups and Frameworks*. CRC Press, 3 2022. ISBN 9781003214748. doi: 10.1201/b22860.
- [49] INEM. Relatório de atividade do codu. *Despacho n.º 14041/2012, de 29 de outubro*, 2021.

- [50] Y. Ni, S. Ying, L. Zhang, J. Wen, P. Ye, e J. Zhang. Modeling aspect-oriented software architecture. In *2009 International Conference on Industrial and Information Systems*, pages 108–113, 2009. doi: 10.1109/IIS.2009.55.
- [51] J. Ding. An approach for modeling and analyzing dynamic software architectures. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 2086–2092, 2016. doi: 10.1109/FSKD.2016.7603503.
- [52] E. Astesiano, editor. *Fundamental Approaches to Software Engineering, 1st International Conference, FASE'98, Held as Part of the European Joint Conferences on the Theory and Practice of Software, ETAPS'98, Lisbon, Portugal, March 28 - April 4, 1998, Proceedings*, volume 1382 of *Lecture Notes in Computer Science*, 1998. Springer. ISBN 3-540-64303-6. doi: 10.1007/BFb0053578.
- [53] E. Yuan. Towards ontology-based software architecture representations. In *2017 IEEE/ACM 1st International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering (ECASE)*, pages 21–27, 2017. doi: 10.1109/ECASE.2017.5.
- [54] Hetzner. Hetzner about us, 2022. URL <https://www.hetzner.com/unternehmen/ueber-uns>.
- [55] Gunicorn. Gunicorn documentation, 2022. URL <https://docs.gunicorn.org/en/stable/>.
- [56] Nginx. Nginx documentation, 2022. URL <https://docs.nginx.com/>.
- [57] Swagger. Sireph api documentation, 2022. URL <http://78.46.215.82:82/api/swagger/>.
- [58] Google. Web services google — geocoding api, 10 2022. URL <https://developers.google.com/maps/documentation/geocoding/overview>.
- [59] Outsystems. Database optimization, 2022. URL https://success.outsystems.com/Documentation/Best_Practices/Development/Database_optimization.
- [60] A. Ahmad, C. Feng, M. Tao, A. Yousif, e S. Ge. Challenges of mobile applications development: Initial results. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 464–469, 2017. doi: 10.1109/ICSESS.2017.8342956.
- [61] K. Ganesh, S. Mohapatra, A. S.P., e P. Sivakumar. User acceptance test, 06 2014.
- [62] B. M. McSkimming, S. Mackay, e A. Decker. Investigating the usage of likert-style items within computer science education research instruments. In *2021 IEEE Frontiers in Education Conference (FIE)*, pages 1–8, 2021. doi: 10.1109/FIE49875.2021.9637198.

Anexo I

Verbete Nacional de Socorro

INEM		REPÚBLICA PORTUGUESA		SNS SERVIÇO NACIONAL DE SAÚDE		VERBETE NACIONAL DE SOCORRO										
OCORRÊNCIA	Entidade	Meio					Nº Evento									
	Motivo	Nº Vítimas					/ /									
	Local						Caminho do local :									
	Freguesia	Concelho					Chegada à vítima :									
IDENTIFICAÇÃO	Nome						Caminho U. Saúde :									
	Nascimento / /	Idade	Sexo	M	F	Nº SNS	Chegada U. Saúde :									
	Residência						Disponível :									
AVALIAÇÃO	Hora Min	AVDS GCS	Vent. cmH ₂ O	SpO ₂ %	O ₂ Sup. L/min	EIC O ₂ mmHg	Pulso bpm	ECG ver verso	P. Arterial Sistólica	P. Arterial Diastólica	Pete	Temp. °C	Pupilas	Dor 0 a 10	Glicemia mg/dL	NEWS/TAP 0 a 18 / 0 a 3
	:															
	:															
	:															
HISTÓRIA CLÍNICA	Circunstâncias															
	Histórico de doenças															
	Alergias															
	Medicação habitual															
Última refeição							Situação de risco									
EXAME DA VÍTIMA, PROCEDIMENTOS E TERAPÊUTICA	SINAIS E SINTOMAS		RCP	VA / Ventilação	Circulação	Protocolos	Escalas									
	Presenciada		Desobstrução	Controlo Temp.	Imobilização	Cincinnati										
	SBV/DAE :		T. Orofaringeo	Controlo Hemo.	VVAVC	PROACS										
	SIV/SAV :		T. Laríngeo	Penso	VV Coronária	RTS										
	1º Ritmo ⚡		Másc. laríngeo	Torniquete	VV Sepsis	MGAP										
	Nº Choque(s)		T. Endotraqueal	Cinto Pélvico	VV Trauma	RACE										
	Recup. :		Vent. Mecânica	Acesso venoso	VV PCR											
	Susp. :		CPAP													
	C. Mecânicas				TEPH											
	Não realizado				SIV											
		FÁRMACO		Dose	Via	Ef. Adv.										
OBSERVAÇÕES	NÃO TRANSPORTE		TRANSPORTE		Primário	Secundário										
	Abandonou o local		Acompanhamento médico													
	Decisão médica		Unidade de Saúde de origem													
	Morte		Unidade de Saúde de destino													
Recusou e assinou S N		Episódio nº.														
Desativação																
Tipo de emergência		Assinatura resp. meio / Nº														

Mod. INEM. 4/15/01 Informação sob sigilo profissional. Política de privacidade e segurança de dados pessoais disponível para consulta em www.inem.pt. | Original para o INEM. Duplicado para a Unidade de Saúde. Triplicado para o meio.

(a) Frente do Verbete nacional de Socorro

ANEXO I. VERBETE NACIONAL DE SOCORRO

RACE (eventual necessidade de trombetomia se ≥5)				ESCALA DE COMA DE GLASGOW ATUALIZADA			
	ESQUERDA	DIREITA	VALOR	OLHOS	VERBAL	MOTOR	
Paresia facial	Ausente	Ausente	0	Esontânea	4	Orientada	5
	Ligeira	Ligeira	1	Ao som	3	Confusa	4
	Moderada/severa	Moderada/severa	2	À Pressão	2	Palavras	3
Paresia MS	Ausente/Ligeiro (>10seg)	Ausente/Lig (>10seg)	0	Ausente	1	Sons	2
	Moderada (<10seg)	Moderada (<10seg)	1	Não testável	NT	Ausente	1
	Severa (não levanta)	Severa (não levanta)	2			Não testável	NT
Paresia MI	Ausente/Ligeiro (>5seg)	Ausente/Ligeiro (>5seg)	0				
	Moderada (<5seg)	Moderada (<5seg)	1				
	Severa (não levanta)	Severa (não levanta)	2				
Desvio oculocéfálico	Direito ausente	Esquerdo ausente	0				
	Direito presente	Esquerdo presente	1				
Agnosia Afasia	Reconhece o braço E o déficit	Afasia obedece a 2 ordens	0				
	Não reconhece o braço OU o déficit	Afasia obedece a 1 ordens	1				
	Não reconhece NEM braço NEM o déficit	Não executa ordens	2				

NEWS (aplicável a não grávidas e idade ≥16 anos, risco elevado de mortalidade se ≥3)							
PARÂMETRO FISIOLÓGICO	3 vermelho	2 amarelo	1 verde	0	1 verde	2 amarelo	3 vermelho
FREQUÊNCIA RESPIRATÓRIA	≤8	9-11	12-20			21-24	≥25
SPO2	≤91	92-93	94-95	≥96			
O2 SUPLEMENTAR		Sim	Não				
TEMPERATURA	≤35	35.1-36.0	36.1-38.0	38.1-39.0	≥39.1		
PRESSÃO ARTERIAL SISTÓLICA	≤90	91-100	101-110	111-219		≥220	
FREQUÊNCIA CARDÍACA	≤40	41-50	51-90	91-110	111-130	≥131	
NÍVEL DE CONSCIÊNCIA			A			V.D ou S	

TRIÂNGULO DE AVALIAÇÃO PEDIÁTRICA (aplicável em idade <16 anos em escala de 0 a 3)								
GRAVIDADE	0	Nula	1, 2 ou 4 verde	Baixa	≥5 ou 3 amarelo	Moderada	≥7 vermelho	Elevada

I IDENTIFICAÇÃO				SITUAÇÃO ATUAL				B ACKGROUND				A VALIAÇÃO				R ECOMENDAÇÕES			
Profissional de saúde Nome e idade da vítima Situação que motivou a ocorrência				Descrição da condição clínica Principais alterações Sinais e sintomas				Histórico de doenças Medicação habitual Alergias				Tipo de emergência identificada Procedimentos realizados Protocolos instituídos				Recomendações Estudos ou avaliações indicadas Proposta de tratamento			

APARÊNCIA		TONUS MUSCULAR ANORMAL		ESFORÇO RESPIRATÓRIO		Sons anormais		CIRCULAÇÃO		Pele pálida	
(+1 se observado um ou mais campos)	Não interage			(+1 se observado um ou mais campos)	Posição tripé/anormal			(+1 se observado um ou mais campos)	Pele marmoreada		
	Difícil de consolar				Tiragem				Cianose		
	Olhar anormal				Adejo nasal						
	Choro / Discurso anormal				Apneia/Gasping						

VIA VERDE SEPSIS (suspeita de VVS se ≥1 em cada um dos quatro campos)				RTS (referenciação a centro de trauma se ≤10)			
TEMPERATURA	SUSPEITA DE INFECÇÃO	INFLAMAÇÃO		GCS	FR (cpm)	PAS (mmHg)	
< 35 °C	1 Cefaleia	1 AVDS	1	13 a 15	4	10 a 29	4
≥ 38 °C	1 Alteração de consciência	1 FC > 90 bpm	1	9 a 12	3	> 29	3
SIGLA SECS	1 Dispneia / Tosse	1 FR > 22 cpm	1	6 a 8	2	6 a 9	2
	1 Dor abdominal			4 a 5	1	1 a 5	1
ASS	1 Icterícia			3	0	0	0
AV1	1 Disúria / Polaquiúria	1 Lactato > 2 mmol/L	1				
AV2	1 Dor lombar	1 TAS < 90 mmHg	1				
AV3	1 Sinais inflamatórios cutâneos	1 PaO2 < 60 mmHg	1				
BRD	1 Critério do clínico responsável	1 SatO2 < 90%	1				
BRE	MGAP (referenciação a centro de trauma se <18)						
ESV	MECANISMO LESÃO	IDADE	GCS	PAS (mmHg)	TOTAL		
EV	Penetrante	0 <60	5	>120	5 23-29	baixo	
FA	Fechado	4 >60	0	60-123	3 18-22	médio	
				<60	0 <18	alto	

ESCALA PROACS (risco elevado de mortalidade se ≥3)		
IDADE	< 72	0
	≥ 72	1
TAS (mmHg)	< 116	1
	≥ 116	0
CLASSE DE KILLIP	1	0
	2	1
	3	1
	4	3
ELEVAÇÃO DE ST	sim	1
	não	0

DECLARAÇÃO DE RECUSA | DECLARATION OF REFUSAL

(Quando sublinhado, risque o que não interessa)

Para os devidos efeitos, declaro ter sido informado(a) dos riscos inerentes à minha decisão de recusar o(s) procedimento(s) recomendados/transporte de ambulância e assumir toda a responsabilidade pelas eventuais consequências.

For the purposes stated, I hereby declare that I was informed of the inherent risks of my decision to refuse the recommended medical procedure(s)/ambulance transportation and take full responsibility for any consequences.

Nome do doente/representante legal | Name of the patient/legal representative

Documento de identificação/n.º | ID Document/n.º

Assinatura | Signature

Assinalar se identificação não confirmada por documento idóneo ☐

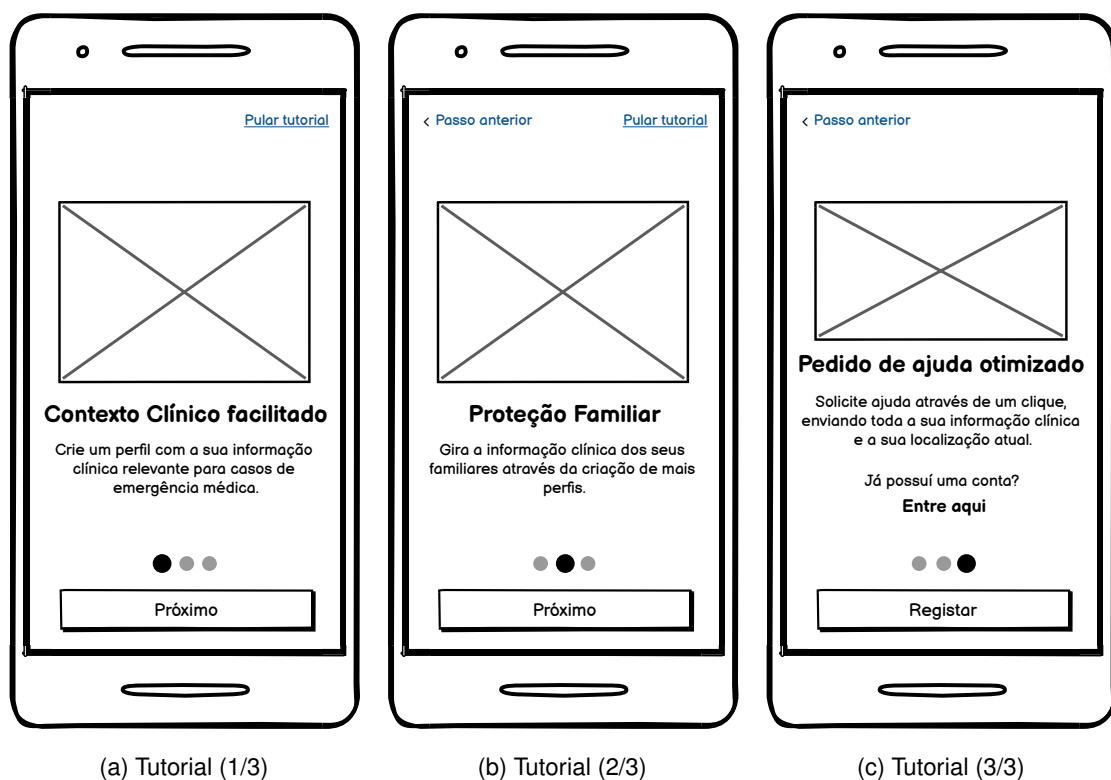
Informação sob sigilo profissional. Política de privacidade e segurança de dados pessoais disponíveis para consulta em www.inem.pt. | DECLARAÇÃO DE RECUSA assinada em duplicado (original para entidade e duplicado para signatário).

(b) Verso do Verbetes nacional de Socorro

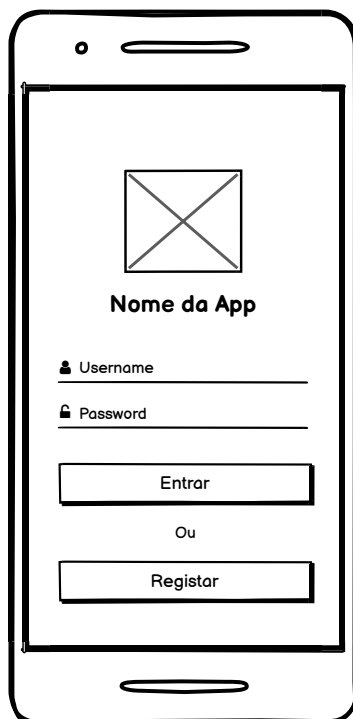
Apêndice A

Wireframes da aplicação

A.1 Wireframes do tutorial da aplicação

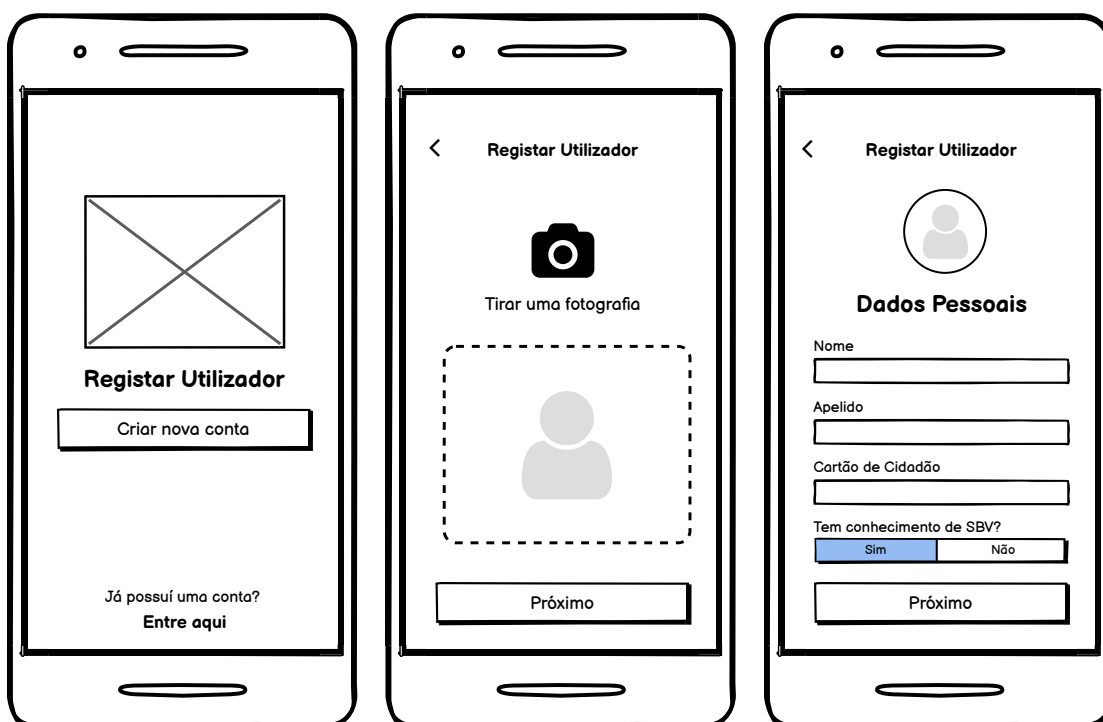


A.2 Wireframe de log in da aplicação



(a) Login da aplicação

A.3 Wireframes do processo de registo do Utilizador



(a) Início

(b) Fotografia do utilizador

(c) Dados pessoais (1/2)

(d) Dados pessoais (2/2)

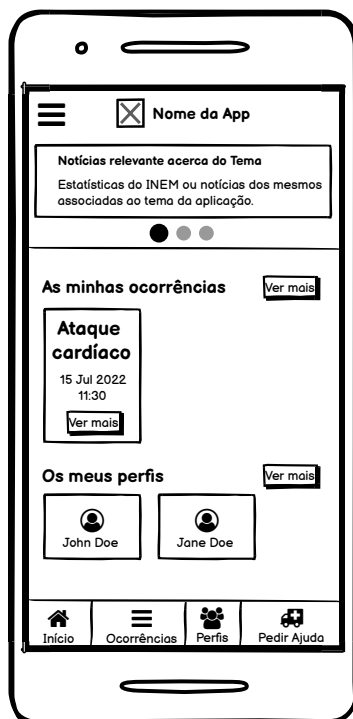
(e) Informação clínica (1/2)

(f) Informação clínica (2/2)

(g) Assinatura do utilizador

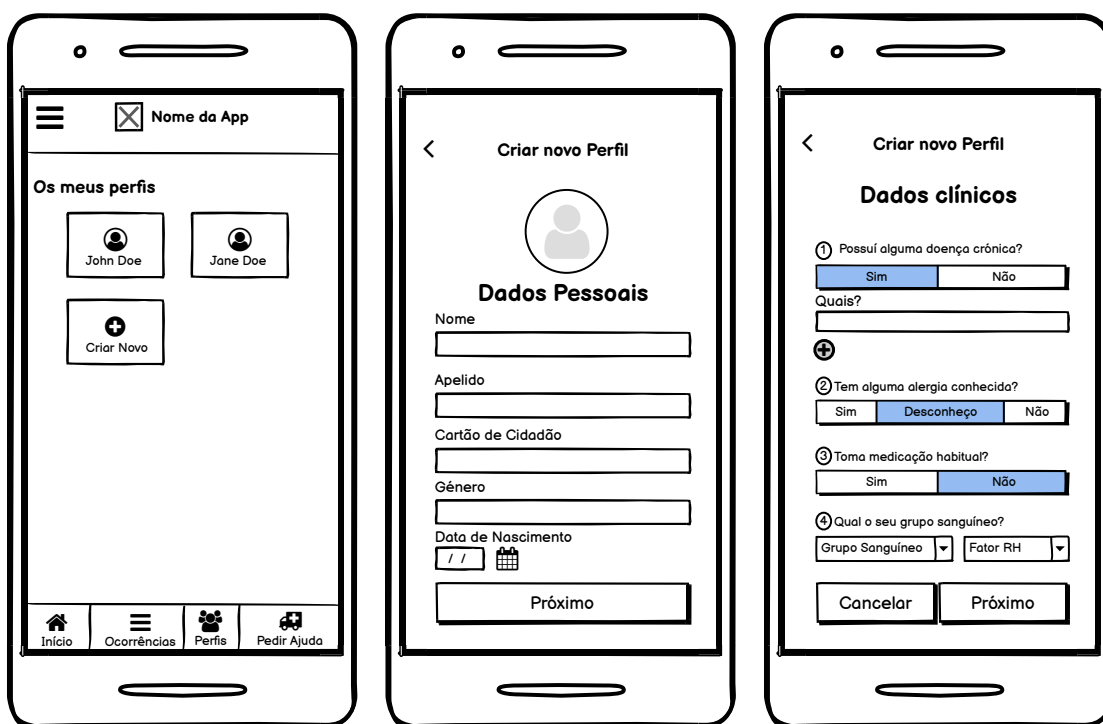
(h) Sucesso

A.4 Wireframe do ecrã inicial



(a) Ecrã Inicial

A.5 Wireframes do processo de criação de perfis

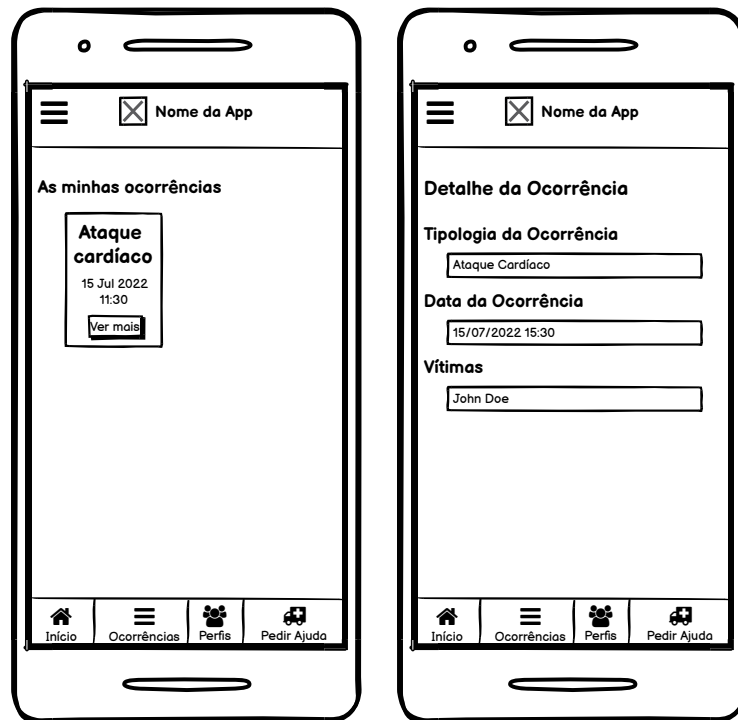


(a) Lista de perfis

(b) Dados Pessoais

(c) Informação clínica

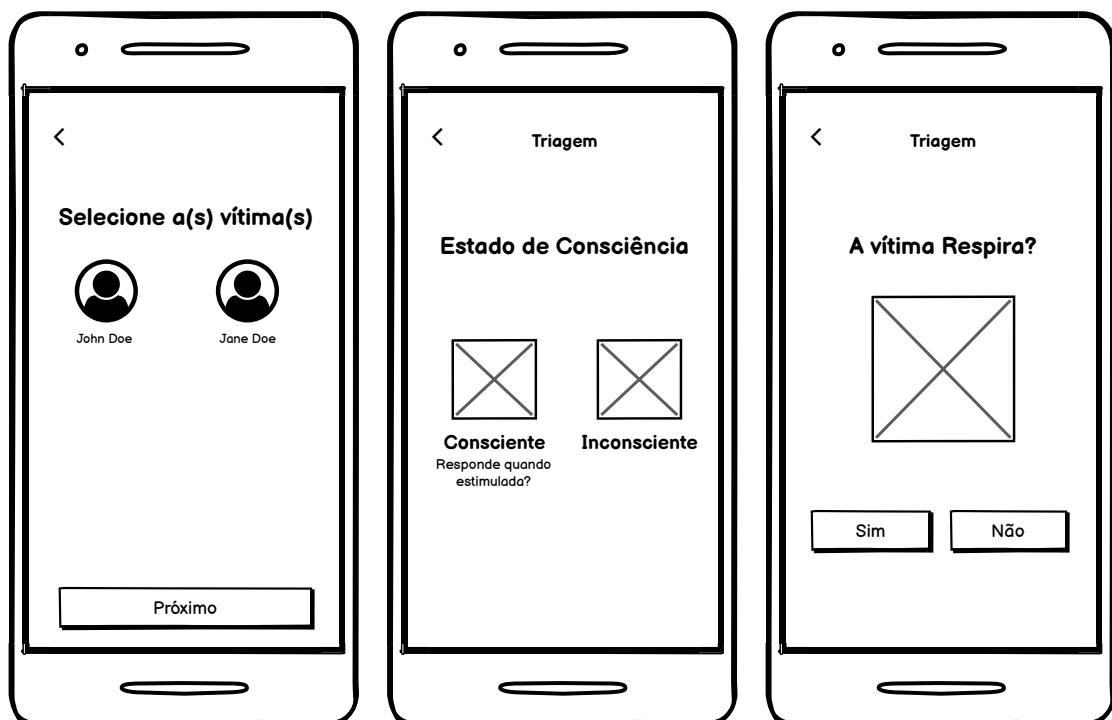
A.6 Wireframes da listagem de Ocorrências



(a) Lista de ocorrências

(b) Detalhe de ocorrência

A.7 Wireframes do processo de criação de uma Ocorrência



(a) Seleção de vítima(s)

(b) Estado de consciência

(c) Estado de respiração



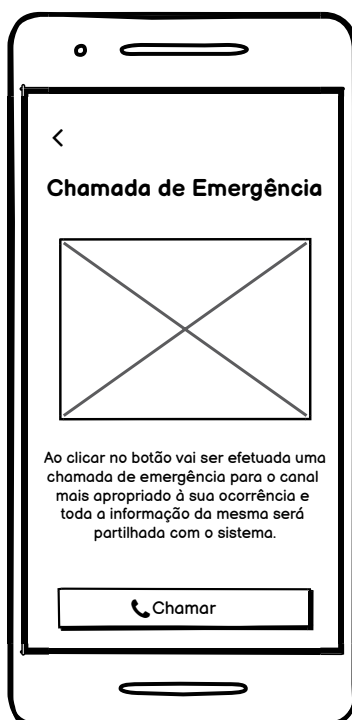
(d) Situação traumática



(e) PLS



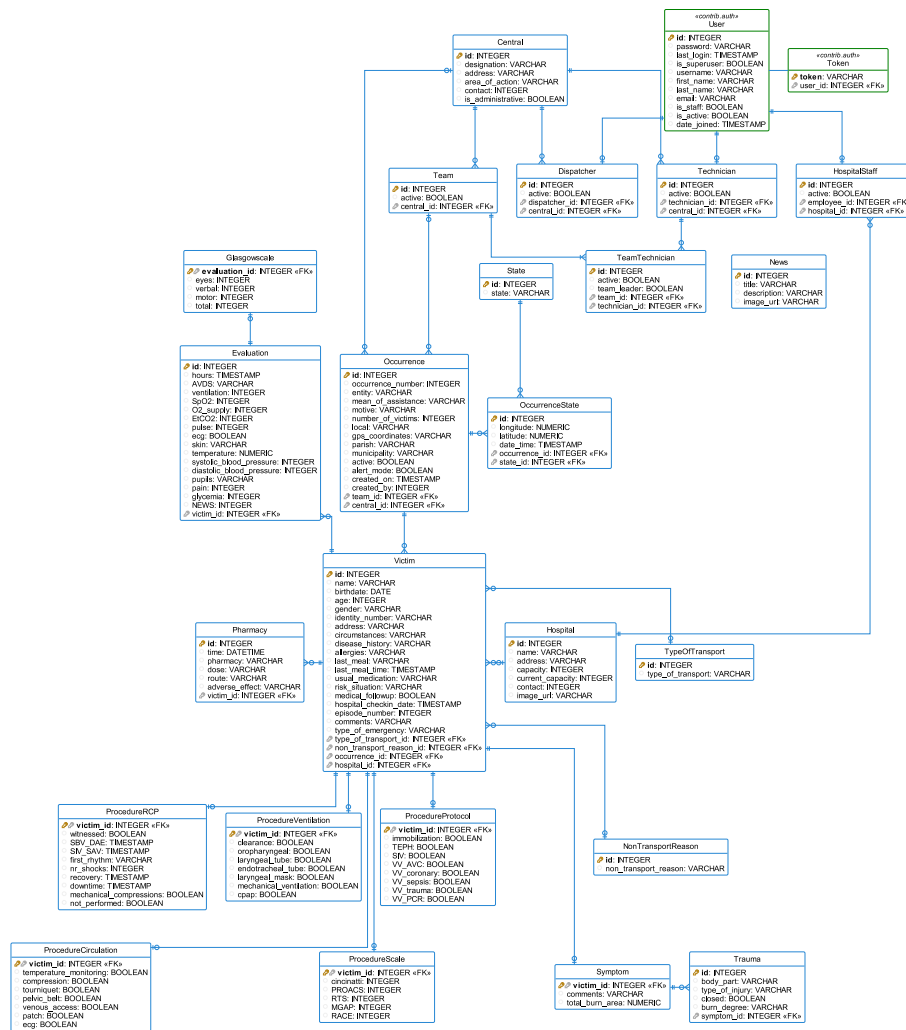
(f) Tipologia de ocorrências



(g) Chamada de emergência

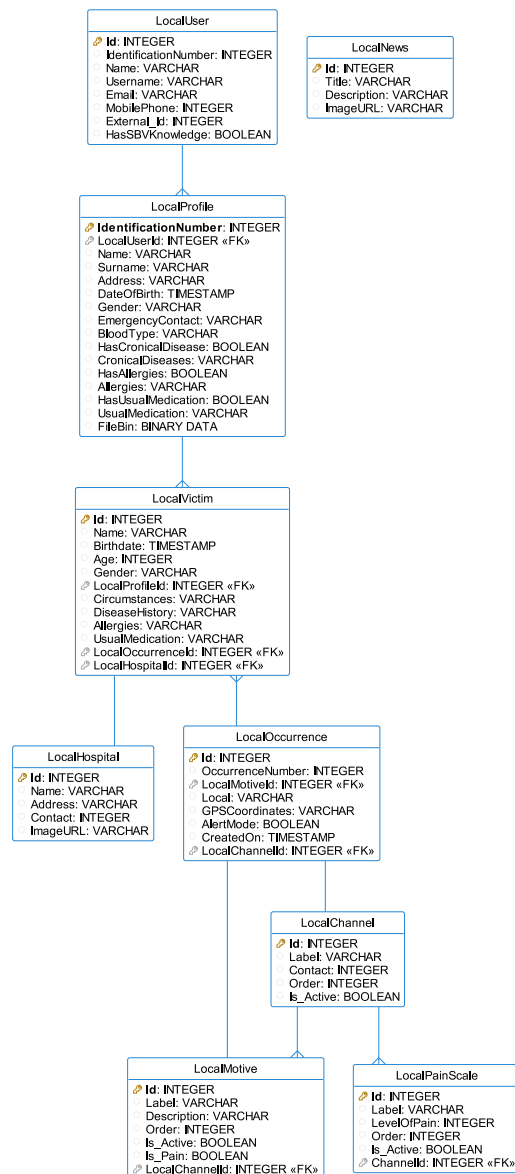
Apêndice B

Modelo de Dados do sistema SIREPH



Apêndice C

Modelo de Dados do Armazenamento Local



Apêndice D

Ecrãs da aplicação

D.1 Ecrã do tutorial da aplicação

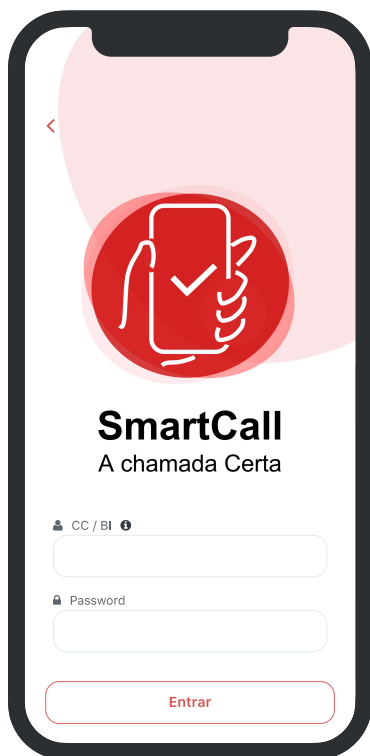


(a) Tutorial (1/3)

(b) Tutorial (2/3)

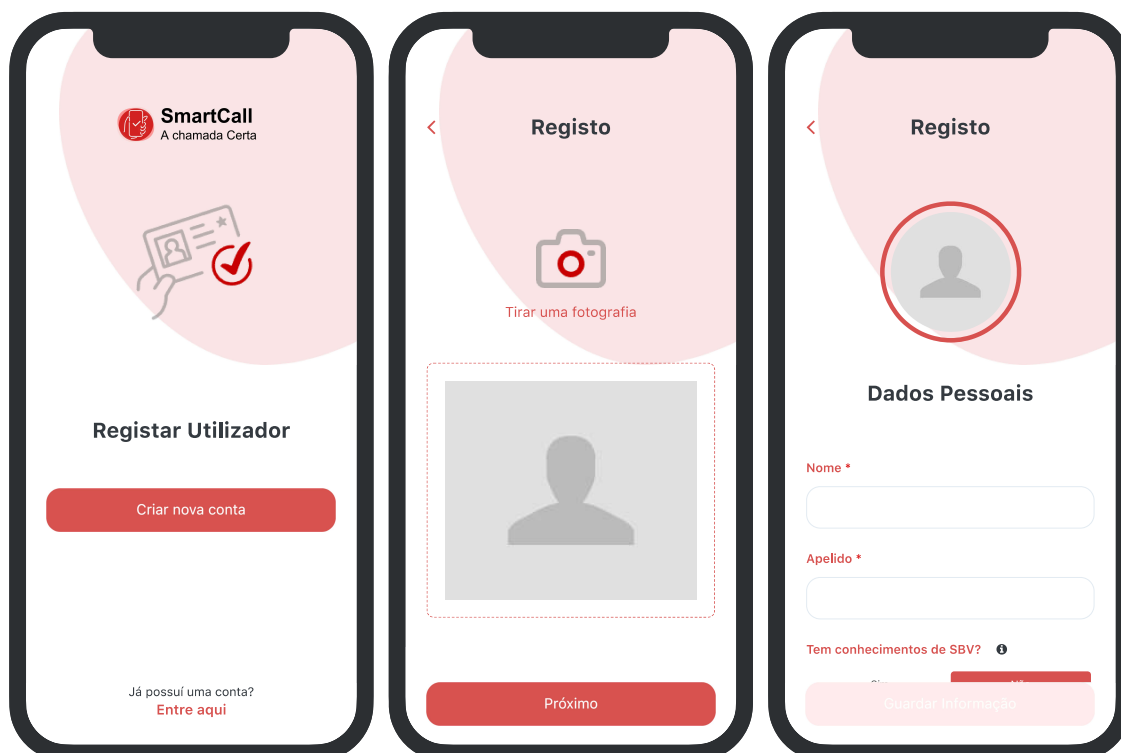
(c) Tutorial (3/3)

D.2 Ecrã de *login* da aplicação



(a) Ecrã de *login* da aplicação

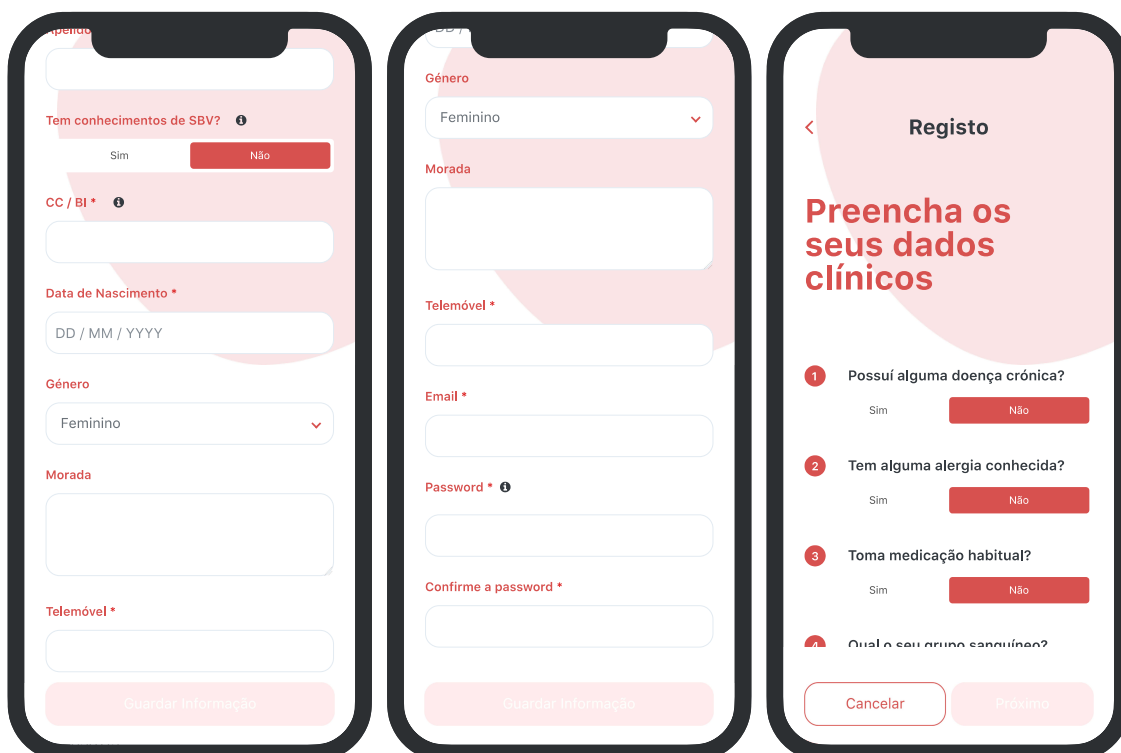
D.3 Ecrãs do processo de registo do Utilizador



(a) Início

(b) Fotografia do utilizador

(c) Dados pessoais (1/3)



(d) Dados pessoais (2/3)

(e) Dados pessoais (3/3)

(f) Informação clínica (1/3)

Sim Não

4 Qual o seu grupo sanguíneo?

RH Positivo RH Negativo Não Sei

A- AB- B- O-

5 Qual o seu contacto de emergência?

6 Aceita os **termos e condições** desta aplicação?

Sim Não

7 Autoriza a utilização dos seus dados pessoais para efeitos de partilha com entidades envolvidas na resposta de emergências?

Sim Não

8 Consente o uso da sua informação

Cancelar Próximo

(g) Informação clínica (2/3)

A- AB- B- O-

5 Qual o seu contacto de emergência?

6 Aceita os **termos e condições** desta aplicação?

Sim Não

7 Autoriza a utilização dos seus dados pessoais para efeitos de partilha com entidades envolvidas na resposta de emergências?

Sim Não

8 Consente o uso da sua informação para efeitos estatísticos para efeitos de otimização dos processos de emergências pré-hospitalares?

Sim Não

Cancelar Próximo

(h) Informação clínica (3/3)

< Registo

Assine abaixo de forma a confirmar o seu registo

Limpar

Criar Conta

(i) Assinatura do utilizador

✓

Bem-vindo, John Doe

O seu registo foi efetuado com sucesso.

Log in

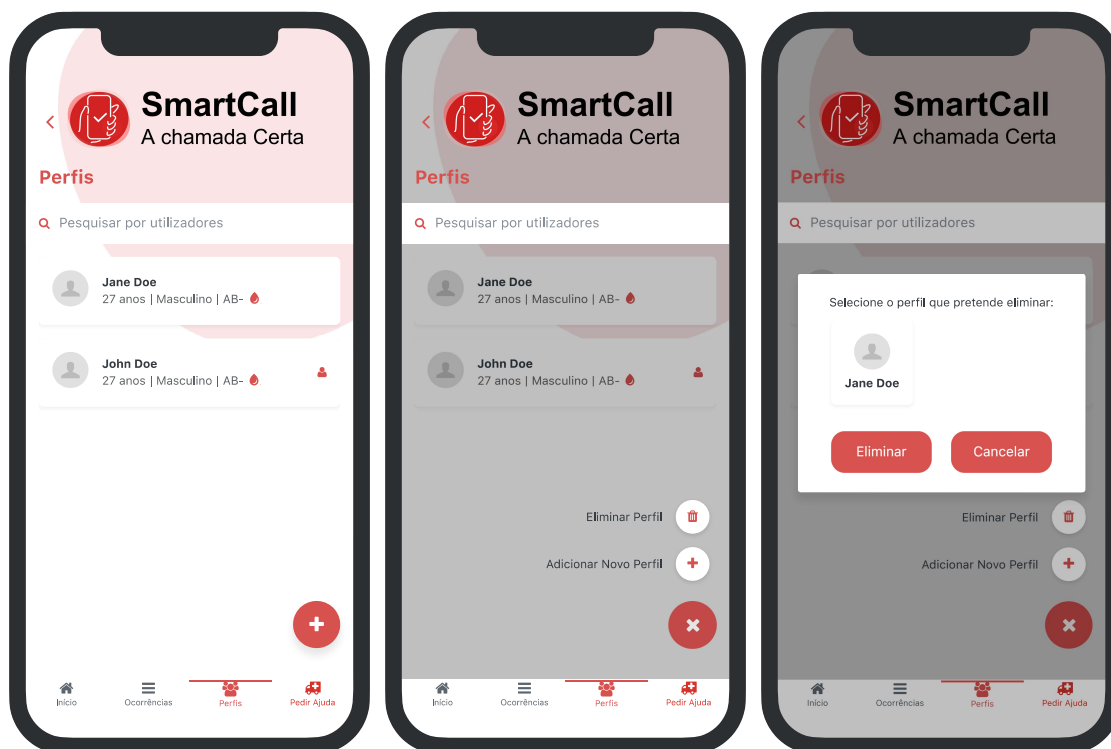
(j) Sucesso

D.4 Ecrã de notícias



(a) Detalhe da notícia

D.5 Ecrãs do processo de criação de perfis

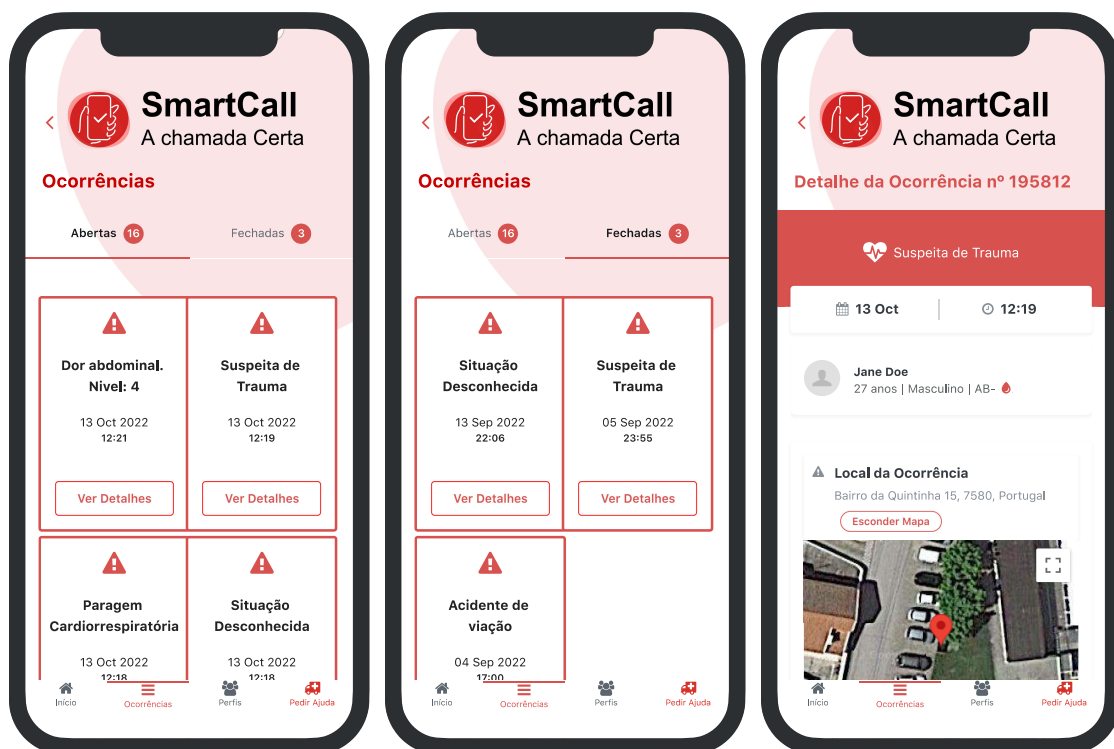


(a) Lista de perfis

(b) Ações sobre perfis

(c) Eliminar perfil

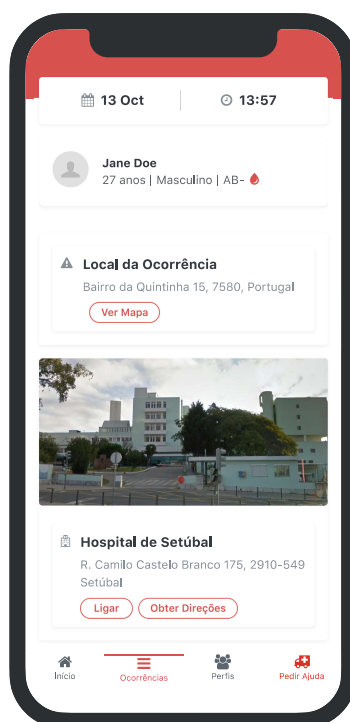
D.6 Ecrãs da listagem de Ocorrências



(a) Ocorrências abertas

(b) Ocorrências fechadas

(c) Detalhe da ocorrência (1/2)



(d) Detalhe da ocorrência (2/2)

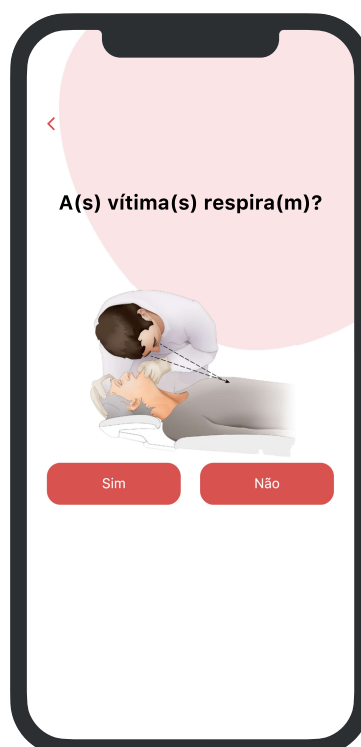
D.7 Ecrãs do processo de criação de uma Ocorrência



(a) Seleção de vítimas



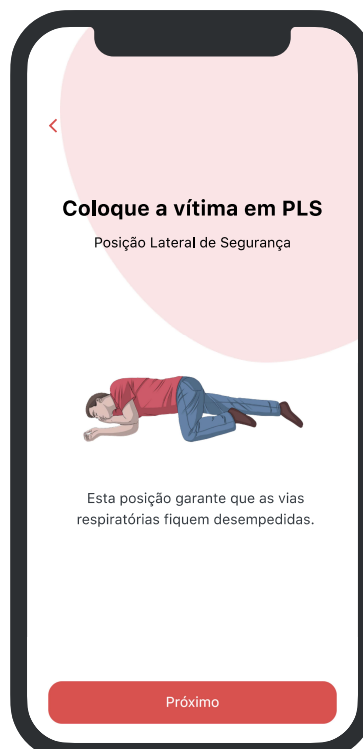
(b) Estado de consciência



(c) Estado da respiração



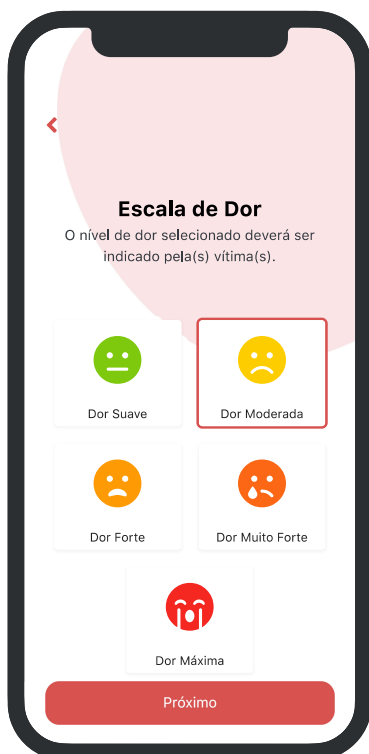
(d) Situação traumática



(e) PLS



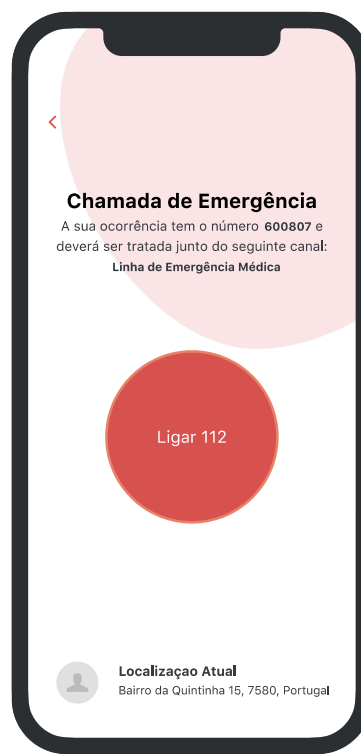
(f) Tipologias de Emergência



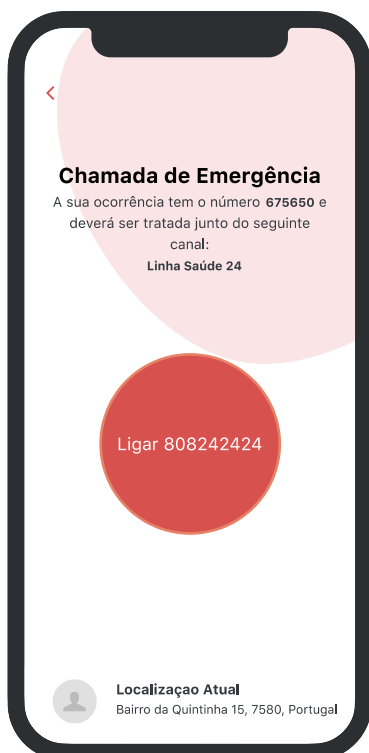
(g) Escala de dor



(h) Outros problemas



(i) Chamada para 112



(j) Chamada para SNS24

Apêndice E

Guião Aplicacional de Testes

Guião Aplicacional

Objetivo

Este guião tem como objetivo descrever a fase de validação de testes da aplicação desenvolvida no âmbito do projeto de final de Mestrado de Engenharia de Software.

Descrição da Aplicação

A aplicação SmartCall – A Chamada Certa tem como objetivo a partilha da informação necessária aquando de uma emergência pré-hospitalar.

Instalação da aplicação



Para proceder à instalação da aplicação, basta seguir os seguintes passos:

1. Digitalizar o código QR aqui disponível com o *smartphone*.
2. Quando o *website* abrir, poderá guardar a aplicação no ecrã para facilitar o acesso à mesma ou usar a aplicação através do *browser*.

Ações de Teste

Pretende-se que efetue ações na aplicação. Note que para cenários em que o número de vítimas seja superior a 1, será necessário a criação de mais perfis na aplicação.

1. Criação de uma conta de utilizador.
2. Criar ocorrências com base nos seguintes cenários abaixo descritos através do botão 
Cenário 1. Encontra-se com uma dor moderada no peito.
Cenário 2. Experimentou um acidente de viação com mais 1 pessoa.
Cenário 3. Outra pessoa foi atropelada por um automóvel, mas encontra-se inconsciente.

Fase de Validação

Com base na experiência que obteve durante a realização das ações descritas anteriormente, é necessário:

- Preenchimento do questionário de validação aplicacional disponível em <https://forms.gle/u4E3Je1Lxc5hRfWS9>;
- Preenchimento e envio da declaração de consentimento disponível em <https://bit.ly/3UkJgnm>.

Apêndice F

Minuta de Declaração de Consentimento

DECLARAÇÃO DE CONSENTIMENTO DO TRATAMENTO DE DADOS PESSOAIS

Eu, _____ (nome completo), portador(a) do Cartão de Cidadão n.º _____, com validade até ____ / ____ / _____, para os efeitos previstos no artigo 13.º do Regulamento Geral de Proteção de Dados (RGPD), (EU)2016/679 do Parlamento Europeu e do Conselho de 27 de abril de 2016, **declaro que fui informado(a) e que** pelo presente meio **expresso o meu consentimento** sobre o tratamento dos meus dados pessoais gerados, no âmbito do estudo do projeto/tese de Mestrado em Engenharia de Software, e que os mesmos serão tratados de forma anónima e apenas para efeitos de investigação científica no estudo acerca da utilização de tecnologias móveis em Emergências Pré-hospitalares.

_____(Local), __ (DD), de ____ (MM) de 2022

(Assinatura)

(Nome)

Apêndice G

Questionário de Validação Aplicacional

Questionário de Validação Aplicacional

O objetivo deste questionário tem como foco a percepção e análise qualitativa da experiência do utilizador na aplicação desenvolvida no âmbito do projeto/tese de final de Mestrado.

**Required*

1. Idade *

Mark only one oval.

☐ até 18 anos

☐ 18 a 20 anos

☐ 21 a 30 anos

☐ 31 a 40 anos

☐ 41 a 50 anos

☐ mais de 51 anos

2. Género *

Mark only one oval.

☐ Feminino

☐ Masculino

☐ Outro

3. Formação académica *

Mark only one oval.

☐ Ensino básico

☐ Ensino Secundário

☐ Licenciatura

☐ Mestrado

☐ Doutoramento

(a) Questionário (1/5)

4. Profissão *

5. Anos de atividade da Profissão anteriormente introduzida *

Mark only one oval.

- ☐ até 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 10 anos
- ☐ mais de 10 anos

**Validação
Aplicacional**

Com base na experiência que obteve durante a realização das ações descritas no guião aplicacional, responda às seguintes perguntas.

6. Conseguiu registar o perfil de utilizador? *

Mark only one oval.

- ☐ Sim
- ☐ Não

7. Tem algum comentário a acrescentar acerca do registo do perfil do utilizador?

(b) Questionário (2/5)

8. Conseguiu registar outro(s) perfis na aplicação? *

Mark only one oval.

☐ Sim

☐ Não

9. Tem algum comentário a acrescentar acerca do registo de outros perfis na aplicação?

10. Conseguiu registar a ocorrência associada ao Cenário 1? *

Mark only one oval.

☐ Sim

☐ Não

11. Tem algum comentário a acrescentar acerca do registo da ocorrência associada ao Cenário 1?

12. Conseguiu registar a ocorrência associada ao Cenário 2? *

Mark only one oval.

☐ Sim

☐ Não

13. Tem algum comentário a acrescentar acerca do registo da ocorrência associada ao Cenário 2?

14. Conseguiu registar a ocorrência associada ao Cenário 3? *

Mark only one oval.

☐ Sim

☐ Não

15. Tem algum comentário a acrescentar acerca do registo da ocorrência associada ao Cenário 3?

(d) Questionário (4/5)

16. Considerou fácil realizar as seguintes ações na aplicação? *

Mark only one oval per row.

	Discordo totalmente	Discordo	Não concordo Nem discordo	Concordo	Concordo totalmente
Registo do perfil do utilizador	☐	☐	☐	☐	☐
Registo de outros perfis	☐	☐	☐	☐	☐
Registo do Cenário 1	☐	☐	☐	☐	☐
Registo do Cenário 2	☐	☐	☐	☐	☐
Registo do Cenário 3	☐	☐	☐	☐	☐

17. Considera que a aplicação poderá ser útil num contexto real? *

Mark only one oval.

- ☐ Discordo totalmente
☐ Discordo
☐ Não concordo Nem discordo
☐ Concordo
☐ Concordo totalmente

18. Tem algum comentário a acrescentar acerca da aplicação de forma geral?

(e) Questionário (4/5)