

DRAFT

PSOCAT: Sistema Móvel de Apoio/Suporte a Doentes com Psoríase



Gonçalo Simão Teixeira Alves do Vale

Faculdade de Ciências e Tecnologia

Universidade Fernando Pessoa

Tese submetida ao grau de

Master of Informatics Engineering (Major in Mobile Computing)

2021

DRAFT

Resumo

A Tele dermatologia é, hoje em dia, uma ferramenta muito importante para auxiliar a monitorização do estado de saúde dos pacientes e a gestão da sua condição da pele, permitindo o diagnóstico precoce de possíveis lesões e doenças dermatológicas. Esta área encontra-se em constante evolução e desenvolvimento, sendo bastante utilizada na troca de informação médica entre especialistas e pacientes, proporcionando um apoio à distância através de dispositivos móveis.

Em particular as aplicações móveis desempenham um papel muito importante na recolha periódica de dados dermatológicos dos pacientes, de modo a que possam ser avaliados pelos profissionais que os seguem bem como auxiliar o auto-cuidado. Com base nesta informação, as equipas podem avaliar a evolução dos pacientes e adequar os meios de tratamento de forma mais precisa. Em particular a Psoríase, uma doença inflamatória de evolução crónica, caracterizada por pápulas na pele, geralmente vermelhas e placas descamativas, enquadra-se nas doenças dermatológicas que podem beneficiar de um acompanhamento remoto mais próximo com recurso a aplicações móveis.

Neste trabalho, propõe-se uma aplicação móvel desenvolvida para dispositivos Android, que permite aos pacientes realizar uma auto-monitorização e avaliação dos aspetos relacionados com a doença. Nomeadamente, permite o registo e visualização do histórico de fotos das zonas de pele afetadas, bem como da realização de avaliações periódicas recorrendo ao instrumento PASI que mede o índice de severidade da psoríase, segmentada por área do corpo.

A aplicação móvel desenvolvida, foi submetida a um estudo de usabilidade de modo a avaliar a facilidade de utilização da mesma por vários utilizadores. Desse estudo resultou um feedback que permitiu afinar alguns aspetos da interatividade da aplicação móvel, que achamos irão contribuir para uma maior e melhor aceitação desta app, por parte da comunidade de pessoas que sofre de psoríase.

Abstract

Tele dermatology is, nowadays, a very important tool to assist in monitoring the health status of patients and in the management of their skin condition, allowing an early diagnosis to analyze possible lesions and / or dermatological diseases. This area is in constant evolution and development, being widely used in the exchange of medical information between specialists and patients, providing remote support through mobile devices.

Mobile applications, in particular, play a very important role in the periodic collection of dermatological data from patients, so that they can be evaluated by the professionals who follow them as well as helping with self-care. Based on this information, teams can assess the evolution of patients and adapt the means of treatment more precisely. In particular Psoriasis, an inflammatory disease of chronic evolution, characterized by papules on the skin, usually red and scaly plaques, is part of dermatological diseases that can benefit from closer remote monitoring using mobile applications.

In this work, we propose a mobile application developed for Android devices, which allows patients to self-monitor and assess aspects related to the disease. In particular, it allows recording and viewing the photo history of affected skin areas, as well as conducting periodic assessments using the PASI instrument which measures the psoriasis severity index, segmented by body area.

The mobile application developed, was submitted to a usability study in order to assess its ease of use by various users. This study resulted in feedback that allowed us to fine-tune some aspects of the interactivity of the mobile application, which we believe will contribute to greater and better acceptance of this app, by the community of people suffering from psoriasis.

DRAFT

Aos Meus Pais

Agradecimentos

O trabalho intenso e exaustivo ao longo do desenvolvimento deste projeto não teria sido possível sem o apoio de várias pessoas. Em primeiro lugar, é importante mencionar a co-orientadora Professora Doutora Isabel F. Almeida que sugeriu o tema e acompanhou o desenrolar do projecto inicialmente. Foi igualmente essencial o contributo dos orientadores Professores Rui Moreira e José Torres, que foram importantes desde o início e que sem eles seria muito difícil encontrar o caminho para finalizar este trabalho, muito graças aos seus conselhos e recomendações ao longo do seu desenvolvimento, tendo constantemente desafiado o mesmo a superar todas as expectativas. Também o apoio da família, amigos e colegas de faculdade foi crucial durante todo este percurso, de maneira a nunca desistir, ultrapassar os obstáculos que foram aparecendo e finalmente alcançar a meta definida. Por isso, resumidamente gostaria de agradecer a todas as pessoas que contribuíram para a realização e conclusão da presente dissertação.

Índice

Índice	vi
Índice de Figuras	viii
Índice de Códigos Fonte	xi
Índice de Tabelas	xii
Lista de Acrónimos	xv
1 Introdução	1
1.1 Contexto e Motivação	1
1.2 Objectivos	2
1.3 Estrutura da Tese	3
2 Sistemas e Aplicações Móveis na Saúde	5
2.1 Sistemas e Aplicações Móveis em Dermatologia	5
2.2 Análise comparativa dos trabalhos/projetos recolhidos	15
2.3 Testes de Usabilidade em Sistemas e Aplicações Móveis	16
3 Sistema PSOCAT	20
3.1 Requisitos do Sistema PSOCAT	20
3.1.1 Requisitos Funcionais	20
3.1.2 Requisitos Não Funcionais	31
3.2 Arquitetura do Sistema PSOCAT	35
3.2.1 Arquitetura do Backend	36
3.2.2 Arquitetura da Aplicação Móvel	38
3.3 Implementação do Sistema PSOCAT	42
3.3.1 Implementação do Backend	42
3.3.2 Implementação da Aplicação Móvel	47
4 Avaliação da Usabilidade da Aplicação PSOCAT	64
4.1 Metodologia de Avaliação	64

4.2	Resultados da Avaliação	70
4.3	Alterações à Aplicação PSOCAT	74
5	Conclusão	75
5.1	Trabalho Futuro	75
	Referências Bibliográficas	77

Índice de Figuras

2.1	Modelos de usabilidade principais	18
3.1	Caso de Uso 1: Permitir ao utilizador efetuar o registo através da aplicação	26
3.2	Caso de Uso 2: Permitir ao utilizador mudar a língua/idioma da aplicação	27
3.3	Caso de Uso 3: Permitir ao utilizador efetuar o login na aplicação	27
3.4	Caso de Uso 4: Permitir ao utilizador autenticar-se com dados de um login feito previamente	27
3.5	Caso de Uso 5: Permitir ao utilizador efetuar o logout seguro da aplicação	28
3.6	Caso de Uso 6: Permitir ao utilizador visualizar a sua página inicial	28
3.7	Caso de Uso 7: Permitir ao utilizador editar o seu perfil	28
3.8	Caso de Uso 8: Permitir ao utilizador capturar e fazer <i>upload</i> de fotos das suas lesões	29
3.9	Caso de Uso 9: Permitir ao utilizador visualizar a sua lista de fotos	29
3.10	Caso de Uso 10: Permitir ao utilizador eliminar uma foto previamente guardada	29
3.11	Caso de Uso 11: Permitir ao utilizador realizar o questionário PASI para detalhar e avaliar as suas lesões	30
3.12	Caso de Uso 12: Permitir ao utilizador visualizar o histórico das avaliações PASI efetuadas	30
3.13	Caso de Uso 13: Permitir ao utilizador eliminar uma avaliação previamente guardada	30
3.14	Caso de Uso 14: Permitir ao utilizador visualizar mais detalhes sobre cada avaliação	31
3.15	Caso de Uso 15: Permitir ao utilizador visualizar a sua lista de eventos relacionados com a sua medicação	31
3.16	Caso de Uso 16: Permitir ao utilizador ver/editar determinado evento selecionado	31
3.17	Arquitetura do Sistema PSOCAT	35
3.18	Modelo da Arquitetura em camadas do Sistema PSOCAT	36
3.19	Arquitetura do Backend do Sistema PSOCAT	37
3.20	Diagrama E-R da base de dados MySQL do Sistema PSOCAT	38

3.21	Arquitetura do Frontend do Sistema PSOCAT	39
3.22	Arquitetura MVC em android do Sistema PSOCAT	39
3.23	Navegação da parte pública da aplicação android do Sistema PSOCAT . .	41
3.24	Navegação da parte privada da aplicação android do Sistema PSOCAT . .	42
3.25	Tabela do utilizador da base de dados MySQL	43
3.26	Tabela do perfil do utilizador da base de dados MySQL	44
3.27	Ecrã inicial da aplicação PSOCAT	48
3.28	Ecrã do menu inicial da aplicação PSOCAT	50
3.29	Ecrã de registo da aplicação PSOCAT	54
3.30	Exemplo de erro no ecrã de registo da aplicação PSOCAT, neste caso por faltar indicar a data de nascimento	55
3.31	Exemplo de aviso de não estar conectado à internet no ecrã de registo da aplicação PSOCAT	56
3.32	Exemplo de aviso de erro ao tentar efetuar o registo no ecrã de registo da aplicação PSOCAT	57
3.33	Exemplo de aviso de ter efetuado o registo com sucesso no ecrã de registo da aplicação PSOCAT	57
3.34	Ecrã da página inicial da aplicação PSOCAT	59
3.35	Menu lateral de navegação na atividade privada da aplicação PSOCAT . .	60
3.36	Ecrã da lista de fotos do utilizador da aplicação PSOCAT, quando não existem fotos a mostrar	61
3.37	Ecrã da lista de fotos do utilizador da aplicação PSOCAT	63
4.1	Exemplo de uma questão do CSUQ	65
4.2	Exemplo da avaliação de uma tarefa do formulário de avaliação da usabi- lidade	69

Índice de Códigos Fonte

3.1	Instrução para efetuar a ligação ao servidor MySQL na base de dados especificada no ficheiro <code>conn.php</code>	44
3.2	Instrução responsável por integrar o código do ficheiro <code>conn.php</code>	45
3.3	Instrução responsável por receber a variável de identificador 'email', enviado pelo método POST	45
3.4	Instrução responsável por obter a informação da tabela do utilizador com o email recebido	45
3.5	Instrução responsável por inserir a informação do utilizador na tabela respetiva	46
3.6	Instrução responsável por terminar a ligação ao servidor MySQL	46
3.7	Instrução responsável por guardar o resultado da <i>query</i> num <i>array</i> e retornar em formato JSON	47
3.8	Ficheiro XML do ecrã inicial	48
3.9	Método <i>onCreate</i> na atividade inicial da aplicação	49
3.10	Instrução em que o tempo de início da aplicação é especificado numa variável em milisegundos na classe <code>Configs</code>	49
3.11	Instrução responsável pelo início da atividade do menu inicial e de terminar de seguida a atividade inaugural	50
3.12	Configurações específicas na construção android e dependências da aplicação no ficheiro <code>build.gradle</code> da aplicação PSOCAT	51
3.13	Código inicial do ficheiro <code>AndroidManifest.xml</code>	51
3.14	Código relativo às permissões do ficheiro <code>AndroidManifest.xml</code>	52
3.15	Código relativo à secção da aplicação e de uma atividade do ficheiro <code>AndroidManifest.xml</code>	52
3.16	Funções relativas à base de dados local na classe <code>AppUtils</code>	53
3.17	Métodos para obter e guardar a linguagem da base de dados SQLite	53
3.18	Código JAVA responsável por adicionar um fragmento específico a um <i>FrameLayout</i> com o id especificado	54
3.19	Exemplo de pedido Volley ao servidor, relativo ao serviço de registo	56
3.20	Código utilizado para criar a cifra da password no modo de encriptação	58

3.21	Código utilizado para codificar a cifra com o email do utilizador(data) e retornar a <i>hash</i> em formato Base64	58
3.22	Código utilizado para criar a cifra da password no modo de decriptação .	58
3.23	Código utilizado para descodificar a cifra com o valor descodificado da <i>hash</i> (data) em formato Base64 e retornar o resultado em formato <i>String</i> .	58
3.24	Código responsável por iniciar o contador de inatividade que irá chamar o método <i>run</i> após o período de tempo especificado pela variável <code>Configs.TIMEOUT_TIME</code>	60
3.25	Código relativo às instruções para limpar toda a <i>cache</i> , terminar a sessão atual e voltar ao menu inicial da aplicação PSOCAT	60
3.26	Exemplo de obtenção de uma resposta/resultado em <i>cache</i> de um pedido efetuado anteriormente	62
3.27	Exemplo de guardar uma resposta/resultado em <i>cache</i> de um pedido efetuado	62

Índice de Tabelas

2.1	Trabalhos/Projetos na área da dermatologia e aspetos analisados	15
3.1	Requisito Funcional 1: Permitir ao utilizador efetuar o registo através da aplicação	21
3.2	Requisito Funcional 2 :Permitir ao utilizador selecionar a língua/idioma da aplicação	21
3.3	Requisito Funcional 3: Permitir ao utilizador efetuar o login na aplicação	21
3.4	Requisito Funcional 4: Permitir ao utilizador autenticar-se com dados de um login feito previamente	22
3.5	Requisito Funcional 5: Permitir ao utilizador efetuar o logout seguro da aplicação	22
3.6	Requisito Funcional 6: Permitir ao utilizador visualizar a sua página inicial	22
3.7	Requisito Funcional 7: Permitir ao utilizador editar o seu perfil	22
3.8	Requisito Funcional 8: Permitir ao utilizador capturar e fazer <i>upload</i> de fotos das suas lesões	23
3.9	Requisito Funcional 9: Permitir ao utilizador visualizar a sua lista de fotos	23
3.10	Requisito Funcional 10: Permitir ao utilizador eliminar uma foto previamente guardada	23
3.11	Requisito Funcional 11: Permitir ao utilizador efetuar um questionário PASI para detalhar e avaliar as suas lesões	23
3.12	Requisito Funcional 12: Permitir ao utilizador visualizar o histórico das avaliações PASI efetuadas	24
3.13	Requisito Funcional 13: Permitir ao utilizador eliminar uma avaliação previamente guardada	24
3.14	Requisito Funcional 14: Permitir ao utilizador visualizar mais detalhes sobre cada avaliação	24
3.15	Requisito Funcional 15: Permitir ao utilizador visualizar a sua lista de eventos relacionados com a sua medicação	25
3.16	Requisito Funcional 16: Permitir ao utilizador ver/editar determinado evento selecionado	25

3.17	Requisito Não Funcional 1: O sistema deverá permitir uma interação simples e intuitiva	32
3.18	Requisito Não Funcional 2: O sistema deverá ser disponibilizado em Português e Inglês	32
3.19	Requisito Não Funcional 3: O sistema deverá restringir as credenciais do utilizador	32
3.20	Requisito Não Funcional 4: O sistema deverá permitir a gestão de permissões da app	33
3.21	Requisito Não Funcional 5: O sistema deverá ser compatível com dispositivos android desde a API 24 até à mais recente	33
3.22	Requisito Não Funcional 6: O sistema deverá possuir um tempo de resposta rápido	33
3.23	Requisito Não Funcional 7: O sistema deverá permitir encriptação de dados privados	33
3.24	Requisito Não Funcional 8: O sistema deverá ter alta disponibilidade para os seus utilizadores	34
3.25	Requisito Não Funcional 9: O sistema deverá possuir acesso à internet	34
3.26	Requisito Não Funcional 10: O sistema deverá manter o utilizador informado sobre erros e/ou falhas que ocorram	34
3.27	Requisito Não Funcional 11: O sistema deverá permitir ao utilizador atualizar a versão da aplicação	34
3.28	Requisito Não Funcional 12: O sistema não deverá permitir ao utilizador efetuar a captura de ecrã na aplicação	35
4.1	Medição de atributos de usabilidade em aplicações móveis	66
4.2	GQM para o Modelo de Usabilidade PACMAD extendido	67
4.3	Resultados da Avaliação de Usabilidade	71
4.4	Resultados da tarefa 1: Efetuar a mudança de idioma na aplicação.	71
4.5	Resultados da tarefa 2: Efetuar um registo com sucesso na aplicação.	71
4.6	Resultados da tarefa 3: Efetuar um login com sucesso na aplicação.	72
4.7	Resultados da tarefa 4: Alterar os dados de perfil inseridos no registo.	72
4.8	Resultados da tarefa 5: Efetuar o logout e voltar a autenticar-se com outro user.	72
4.9	Resultados da tarefa 6: Registrar na aplicação uma foto da mão/braço ou outra zona do corpo.	72
4.10	Resultados da tarefa 7: Visualizar a lista de fotos já capturadas.	72
4.11	Resultados da tarefa 8: Eliminar uma foto existente.	73
4.12	Resultados da tarefa 9: Efetuar o cálculo do índice PASI.	73
4.13	Resultados da tarefa 10: Visualizar um registo de avaliações PASI.	73

4.14 Resultados da tarefa 11: Obter mais detalhes sobre cada avaliação visualizada anteriormente.	73
4.15 Resultados da tarefa 12: Visualizar a lista de eventos da aplicação durante determinado período/intervalo de datas	73
4.16 Resultados da tarefa 13: Visualizar mais informação sobre cada evento selecionado anteriormente	74
4.17 Sumarização das alterações efetuadas após a avaliação da usabilidade da aplicação	74

Lista de Acrónimos

ABCD *Asymmetry Border Colour Diameter*

AES *Advanced Encryption Standard*

API *Application Programming Interface*

APK *Android Package*

BSA *Body Surface Area*

CSUQ *Computer System Usability Questionnaire*

DLQI *Dermatology Life Quality Index*

GQM *Goal Question Metric*

HSV *Hue, Saturation, Value*

HTTP *Hypertext Transfer Protocol*

IDE *Integrated Development Environment*

IPM *Image Processing Module*

ISO *International Organization for Standardization*

JSON *JavaScript Object Notation*

ML *Machine Learning*

MVC *Model View Controller*

MySQL *'My' Structured Query Language*

NAPSI *Nail Psoriasis Severity Index*

PACMAD *People At the Centre of Mobile Application Development*

PASI *Psoriasis Area Severity Index*

PPPASI *Palmoplantar Psoriasis Area Severity Index*

PSOCAT *Psoriasis Capture Analyze Treat*

RGB *Red, Green, Blue*

SAF *Store And Forward*

SA-PASI *Self-Administered Psoriasis Area Severity Index*

UI *User Interface*

WAMP *Windows, Apache, MySQL, and PHP*

XML *Extensible Markup Language*

Capítulo 1

Introdução

Neste capítulo será efetuada uma descrição geral do trabalho desenvolvido, contextualizando a motivação e descrevendo os objetivos principais do projeto. Termina-se com uma breve descrição da estrutura da dissertação.

1.1 Contexto e Motivação

As doenças da pele têm um importante impacto na qualidade de vida dos doentes, em particular se se tratarem de doenças crónicas. A Psoríase é uma das dermatoses crónicas com maior prevalência e maior impacto psicossocial, que se caracteriza por alternância de períodos de remissão e de exacerbação da doença. Torna-se portanto importante que as pessoas possuam formas de ajuda na gestão deste tipo de doenças.

Por outro lado, como é possível verificar no artigo (de Saúde, 2011), em Portugal de forma particular os dermatologistas não estão distribuídos equitativamente pelo território nacional. Existe maior concentração de especialistas nas principais áreas urbanas do litoral. Adicionalmente o número de recursos humanos necessários não são suficientes para cobrir as necessidades de apoio.

Por estas razões, a Tele dermatologia aparece como uma possível ferramenta, essencial para monitorizar e diagnosticar problemas dermatológicos da pele, e ajudar na gestão do seu tratamento. Por exemplo a recolha de imagens da pele, obtidas via telemóveis, podem ser suficientes para o diagnóstico de doenças dermatológicas e diagnóstico precoce do cancro da pele, bem como para a monitorização da eficácia terapêutica, embora dependentes da sua qualidade e da avaliação concomitante por profissionais de saúde. Deste modo, a identificação e diagnóstico de algumas doenças de pele poderia ser auxiliado pela existência de aplicações móveis adequadas e adotadas pelos profissionais.

De todas as doenças dermatológicas existentes, consideramos neste trabalho a Psoríase, que se trata de uma doença inflamatória crónica da pele, muito comum, afetando cerca de 2% da população em todo o mundo (Schreier et al., 2008). A prevalência geral da psoríase é semelhante em homens e mulheres. As lesões de psoríase podem aparecer em todo o corpo mas as regiões mais comuns são cotovelos, joelhos e couro cabeludo (Schreier et al., 2008).

Apesar da Psoríase poder ser tratada de várias formas, não existe atualmente uma cura específica e a terapia de longo prazo é necessária. Portanto, a monitorização clínica contínua, com avaliação periódica do estado da doença é crucial. É por esse motivo que se torna importante disponibilizar meios para tornar a vida destas pessoas mais fácil, auxiliando-as no acompanhamento, monitorização e tratamento da doença, sendo fundamental salientar, das técnicas utilizadas para avaliar e classificar esta doença, o uso da escala Psoriasis Area Severity Index (PASI), que se trata de um dos questionários utilizados para calcular a severidade das lesões observadas (Koller et al., 2011).

Torna-se portanto essencial investigar trabalhos e projetos científicos aplicados na dermatologia em geral e na psoríase em particular. Pretende-se analisar de que maneira o uso da tele dermatologia contribuiu e continua a contribuir para o desenvolvimento de potenciais aplicações para ajudar os pacientes que sofrem deste tipo de doenças.

1.2 Objectivos

O objetivo deste trabalho passa por criar um sistema baseado em dispositivos móveis para apoiar a auto-monitorização dos pacientes com psoríase. Este sistema pretende auxiliar os doentes na recolha, processamento e armazenamento de informações sobre as lesões de pele, bem como no registo de avaliações periódicas.

A recolha de artigos sobre aplicações semelhantes/relacionadas tem como intuito auxiliar no desenvolvimento deste trabalho. É importante perceber aquilo que já existe no mundo atual de maneira a não só utilizar essa informação para a construção da aplicação pretendida, mas também entender as vantagens e desvantagens dos projetos pesquisados para os pacientes no seu geral.

Outro ponto crucial diz respeito à necessidade do sistema possuir uma arquitetura adequada dos elementos que o constituem assim como permitir a sua interação. Basicamente, para o sistema funcionar devidamente é importante a comunicação entre a aplicação móvel e o servidor específico de forma a satisfazer os pedidos efetuados pelos vários utilizadores que usem a aplicação.

A aplicação desenvolvida deverá cumprir determinados requisitos que serão necessários para que vá de encontro com as expectativas dos utilizadores e permitam tornar a aplicação fácil de utilizar, isto é, que as suas funcionalidades sejam simples de executar pela diversidade de utilizadores existente. Assim, no desenvolvimento da aplicação é essencial seguir um modelo de usabilidade específico, ou seja, um modelo que permita avaliar a aplicação no que diz respeito a determinados aspetos da funcionalidades da mesma. O objetivo passa por obter feedback por parte de utilizadores, tendo em conta diferentes métricas de avaliação relativas ao comportamento e desempenho da aplicação. Este feedback será posteriormente utilizado para poder efetuar alterações e correções na aplicação de modo a melhorar a usabilidade da mesma.

1.3 Estrutura da Tese

A dissertação encontra-se organizada em cinco capítulos: primeiro a introdução, seguida do estado da arte com uma revisão sobre sistemas e aplicações móveis na área da saúde e uma comparação entre eles, bem como uma introdução ao tema da usabilidade, uma vez que será avaliada na aplicação do sistema desenvolvido. O capítulo seguinte descreve o sistema desenvolvido, nomeadamente os requisitos, arquitetura e implementação do *backend* e da aplicação móvel. Esta aplicação será posteriormente analisada e avaliada no quarto capítulo, no que diz respeito à sua usabilidade. Descreve-se a metodologia de avaliação, os resultados obtidos e as alterações decorrentes na aplicação. Finaliza-se a dissertação com uma conclusão, onde se resumem as principais contribuições do trabalho e se alinham possíveis trabalhos futuros relativamente ao sistema desenvolvido.

No primeiro capítulo referem-se os conceitos base desta dissertação, introduzindo o assunto que se pretende tratar, as motivações e objetivos para a sua resolução. Por fim, é explicada a estrutura da presente tese.

Segue-se o estado da arte com uma revisão sobre sistemas e aplicações móveis na área da promoção da saúde, mais propriamente na dermatologia e na psoríase em particular, seguida de uma análise comparativa entre os trabalhos/projetos recolhidos. Este capítulo aborda ainda a importância da usabilidade na atualidade e a sua evolução.

No terceiro capítulo apresentam-se os requisitos do sistema desenvolvido, assim como a sua arquitetura e implementação. Neste último ponto, é então detalhada a arquitetura dos constituintes do sistema, nomeadamente a aplicação móvel e o *backend*, assim como a implementação de ambos.

No quarto capítulo é apresentada uma avaliação da usabilidade da aplicação móvel desenvolvida. É descrita a metodologia utilizada de modo a poder analisar o desempenho e eficácia da app, apresentando de seguida as alterações efetuadas na mesma em função dos resultados obtidos.

Por fim, no quinto e último capítulo é apresentada a conclusão do trabalho desenvolvido. O capítulo é posteriormente finalizado com o conseqüente trabalho futuro.

Capítulo 2

Sistemas e Aplicações Móveis na Saúde

Este capítulo procura fazer uma revisão do estado da arte sobre os sistemas e aplicações móveis usadas no acompanhamento e promoção da saúde de forma genérica, focando posteriormente e mais especificamente as aplicações direcionadas para a área dermatológica e a psoríase em particular. Efectua-se ainda uma comparação e análise entre os diferentes sistemas e aplicações identificados. O capítulo encerra com uma secção dirigida para os testes de usabilidade, realçando a necessidade e importância destes no sucesso da adoção das aplicações móveis.

2.1 Sistemas e Aplicações Móveis em Dermatologia

Segue-se uma revisão acerca dos vários sistemas e aplicações móveis relacionadas com a área da saúde, focando-se em particular na dermatologia e em aspetos diretamente relacionados com doenças específicas. As aplicações preconizam a utilização de dispositivos móveis com o intuito de auxiliar na gestão e monitorização dos problemas da pele (e.g. com recolha de imagens, testes, etc.), demonstrando que a tele dermatologia se encontra bem presente e em constante evolução atualmente.

Muitas das aplicações possuem algoritmos próprios, quer para o cálculo da severidade das doenças quer para a classificação das imagens e fotos das lesões dos pacientes. Naturalmente são de maior interesse as técnicas utilizadas que mais vão de encontro com a aplicação que se pretende desenvolver, assim como outros detalhes relevantes que sejam apropriados à mesma.

Foi realizado um estudo de diferentes artigos que descrevem sistemas e aplicações móveis desenvolvidas no âmbito da dermatologia, tendo sido essencial conhecer e analisar estes projetos, particularmente os que se focam concretamente nas aplicações que auxiliam pessoas com psoríase, onde a tele dermatologia possui um papel fundamental na vida das

peessoas, sendo um dos tópicos principais a ter em conta nos sistemas estudados.

A teledermatologia diz respeito a uma das aplicações mais comuns da Telemedicina, que faz referência a tecnologias de telecomunicação usadas para trocar informação à distância por determinado meio de comunicação. De acordo com alguns artigos recolhidos sobre o tema como (Kaliyadan and Venkitakrishnan, 2009), (Yarak et al., 2017) e (Miot et al., 2005), a teledermatologia basicamente possui dois métodos principais, *Store And Forward* (SAF) e *Live Interactive*, existindo também outros modelos híbridos.

SAF é uma abordagem mais reconhecida e comum, que se trata de comunicação à distância e captura de fotos dos pacientes, que são armazenadas e mais tarde consultadas pelos respetivos médicos, sendo que a interação presencial entre paciente e dermatologista se torna desnecessária. Com custos baixos, sendo fácil de utilizar e não necessitando de agendar consultas, revela-se uma técnica bastante viável e económica.

Live interactive basicamente permite a comunicação à distância em tempo real entre utentes e médicos, sendo um dos benefícios desta técnica, facilitando aos cuidadores de saúde o acesso às informações dos seus pacientes. Por outro lado, os custos das tecnologias utilizadas assim como a necessidade de múltiplas câmaras para as videoconferências são alguns dos problemas que esta abordagem possui, tal como a necessidade de conexão à internet e uma elevada largura de banda disponível, sendo situações que podem complicar a interação entre os utilizadores, dificultando a marcação de consultas e aumentando os tempos de espera.

Das várias técnicas/métodos existentes, serão cruciais aqueles que necessitem de dispositivos móveis para o seu correto funcionamento. Deste modo, a teledermatologia com o suporte de sistemas/aplicações móveis é o foco principal no que diz respeito aos artigos recolhidos, sendo este tema aquele que possui maior relevância para o projeto desenvolvido.

No primeiro artigo recolhido, (Clark et al., 2018) , a teledermatologia aparece em destaque numa revisão às aplicações móveis relacionadas. Nesta revisão, é também efetuada uma análise comparativa entre a teledermatologia associada a dispositivos móveis e a dermatologia presencial, sendo avaliada a precisão e concordância do diagnóstico das condições de pele.

Após alguns estudos/análises de dados de pessoas em diferentes zonas do mundo, concluiu-se que o diagnóstico presencial continua a ser superior ao diagnóstico da teledermatologia com dispositivos móveis. Os resultados devem-se ao facto de a precisão e concordância

da tele dermatologia com dispositivos móveis continuar a não atingir os mesmos resultados que a dermatologia presencial. Apesar disso, o hardware e software utilizado nas aplicações móveis evoluiu imenso nos últimos anos e os resultados são muito mais apelativos, sendo que com a evolução da tecnologia a tele dermatologia baseada em dispositivos móveis poderá se revelar uma solução muito mais viável futuramente.

A tele dermatologia continua a ser o tópico em destaque no artigo (Moreno-Ramírez and Argenziano, 2017), onde se refere uma rede de tele dermatologia em Espanha cujas práticas se revelaram positivas para os pacientes que sofrem de cancro da pele. Desses pontos positivos destacam-se a redução dos tempos de espera por parte dos pacientes, o que se traduz num tratamento mais rápido e eficaz, para além de aumentar o número de consultas possíveis ao mesmo tempo. Os custos das tecnologias utilizadas também se revelaram relativamente baixos.

Relativamente às aplicações móveis utilizadas, umas são destinadas para os cuidadores de saúde e/ou médicos, como a *Skindiag* que é uma aplicação grátis que possibilita o diagnóstico e a assistência terapêutica, permitindo a troca de informações entre especialistas, discussões de diversos casos e consulta de casos antigos. Das aplicações destinadas aos seus pacientes, destaca-se a aplicação *SkinVision* que é responsável por verificar as lesões de pele, procurando manchas e/ou outros sinais de lesão. Os pacientes podem fornecer fotos das suas lesões que a aplicação irá posteriormente avaliar e fornecer uma avaliação prévia. Apesar desta aplicação ser bastante popular, ainda existem dúvidas sobre a veracidade dos seus resultados.

Outra aplicação móvel é retratada no artigo (Rizvi et al., 2020), sendo que facilita o trabalho dos médicos e dermatologistas na Noruega permitindo, além de possibilitar a troca de informação entre os especialistas, diminuir o tempo de espera das consultas e a sua duração. Com a limitação do número de dermatologistas na Noruega, assim como a necessidade de consultas presenciais, é importante a utilização de um aplicativo que recolha a informação e permita a sua troca à distância entre pacientes e especialistas, facilitando a monitorização dos pacientes e reduzindo a sobrecarga de consultas presenciais.

A aplicação móvel desenvolvida utiliza uma lente de aumento dermatoscópio conetada à câmara do dispositivo móvel, para obter as imagens relativas às lesões de pele. Em termos de custos de transportes e outros custos relacionados, esta abordagem revelou-se positiva para grande parte dos pacientes, para além de possuir resultados positivos.

Nos artigos (Sáenz et al., 2018) e (Sáenz et al., 2016), é retratado um caso de estudo numa área rural na Colombia, onde o uso de diagnóstico remoto com o auxílio de uma

aplicação móvel surge como uma possível solução para as pessoas dessa região. Devido à ausência de dermatologistas nesta área rural, à quase inexistência de meios tecnológicos e ao afastamento das pessoas dos estabelecimentos hospitalares, esta aplicação tem como objetivo resolver estes problemas permitindo a comunicação à distância entre os cuidadores de saúde e os dermatologistas de maneira a possibilitar a monitorização do paciente e a troca de informação relativa às suas lesões.

Esta aplicação, de nome *Skinhealth*, consiste num sistema que suporta o diagnóstico das lesões de pele dos pacientes, usando ontologia e uma base de conhecimento direcionada para a dermatologia integrada com a própria aplicação. Resumidamente um cuidador de saúde, com ou não conhecimento de dermatologia, primeiro captura/obtem a informação da parte dos pacientes relativamente às lesões de pele que estes possuem. De seguida, a aplicação liga-se a um servidor e transfere os dados obtidos. O servidor possui um serviço web que recebe os pedidos e comunica com a base de conhecimento existente num servidor externo para analisar a informação recebida. Por fim, são retornados os resultados para o servidor que por sua vez retorna essa informação à aplicação móvel.

Os resultados obtidos por parte da aplicação foram comparados por especialistas/dermatologistas, chegando à conclusão que apesar de se revelarem favoráveis é preciso efetuar mais testes e obter mais dados para comprovar a sua eficiência, existindo ainda discordância nos valores obtidos. Geograficamente e economicamente, é uma possível solução para o futuro dos pacientes que sofrem de doenças dermatológicas.

No artigo seguinte, (Vasconcelos and Rosado, 2014), encontra-se retratada uma aplicação móvel com o apoio de técnicas de imagem de modo a determinar certas doenças no ramo da dermatologia existentes. O principal objetivo passa por utilizar um dispositivo móvel para obtenção de imagens de lesões das doenças, e depois com o auxílio de técnicas de imagem baseadas na deteção de reflexões tendo em conta determinados espaços de cor, como *Hue*, *Saturation*, *Value* (HSV), no que diz respeito à variância da luminosidade presente. De seguida, graças à grande variedade de cores existentes, torna-se mais simples a obtenção de resultados mais concretos da análise das imagens capturadas. São utilizados neste procedimento alguns algoritmos de filtragem como o filtro de Gauss e o de retina Parvo, de modo a suavizar as imagens, atenuar a luminosidade e realçar os contornos das fotos para serem de seguida segmentadas utilizando um *threshold* de frequência máxima dos histogramas da diferença da máscara. Os resultados da presente abordagem revelaram-se mais tarde positivos.

Com o auxílio de dispositivos móveis, algumas técnicas utilizadas em combinação e tendo como alvos as imagens capturadas, são cruciais para extrair a informação ne-

cessária acerca do estado de severidade dos pacientes, sendo classificadas de acordo com as informações recolhidas. Também é igualmente importante a eficiente análise das fotos capturadas pelos dispositivos móveis, sendo importante que a análise seja eficaz, e que para isso as fotos sejam capturadas em alta qualidade para melhorar os resultados da sua extração e processamento.

No artigo (Okuboyejo, 2013), o foco é direcionado para a telemedicina, cuja aplicação referida e retratada baseia-se num sistema de inteligência artificial com o objetivo de analisar, processar e classificar a informação de uma livraria que é consultada por vários especialistas na área da dermatologia. Para a obtenção das imagens, os dispositivos móveis são os meios utilizados, sendo que tal como noutros artigos pesquisados, possibilitam a diminuição de especialistas necessários para classificar as mesmas.

Na análise das imagens, um determinado algoritmo possui a função de extrair informações e classificar as diferentes lesões de pele existentes, o que facilita o diagnóstico e torna o tratamento eficaz. O sistema utiliza então técnicas de análise de cor, *threshold* e também *region growing*, com o intuito de processar e segmentar as imagens recolhidas da aplicação móvel. Por fim, cada imagem é classificada utilizando a regra *Asymmetry Border Colour Diameter* (ABCD) e um classificador denominado *Multilayer Perceptron*, de modo a ser possível considerar se as lesões são de facto malignas. O sistema acabou por nunca ser testado.

No artigo (Rosado and Ferreira, 2013), é apresentada outra aplicação móvel desenvolvida para analisar e classificar lesões na pele e também supervisionar os pacientes, neste caso específico o público caucasiano. O sistema da aplicação, após extração e recolha dos dados necessários, utiliza a regra ABCD de maneira a detetar anomalias existentes na pele, sendo constituído, em termos de arquitetura, por um dispositivo *frontend* (dispositivo móvel), que obtém os dados das imagens, depois de capturadas, e os envia para análise por parte do servidor. Posteriormente o servidor, tratando-se de um serviço web RESTful em JAVA, recebe os dados e devolve a informação resultante da análise dos mesmos. Por fim, um dispositivo *backend* vai receber a resposta e executar uma análise *Image Processing Module* (IPM), ou seja, filtra inicialmente e realça de seguida o contraste das cores das imagens que são depois segmentadas com o apoio do método de Otsu's, extraíndo os dados necessários a partir do espaço de cores *Red, Green, Blue* (RGB) sendo depois utilizado um algoritmo de *clustering* para detetar as cores. O sistema ainda não foi testado.

Também no artigo (Becevic et al., 2013) se insere na mesma tecnologia e enaltece a importância da usabilidade e eficácia como fatores importantes na aplicação móvel do qual faz parte, tendo como função monitorizar e diagnosticar os pacientes. O artigo realça a

necessidade de seguir padrões específicos relativos ao sistema em questão, assim como ter a responsabilidade de manter a integridade dos dados de cada utilizador apelando a que os pacientes sejam devidamente responsáveis quanto à sua saúde e no que diz respeito a seguirem as indicações dos médicos.

Visa a criação de uma aplicação eficaz e útil, para que os utentes sejam responsáveis e cuidadosos com a sua saúde, de maneira a contribuir para a gestão mais eficiente por parte dos seus cuidadores de saúde, para os poderem monitorizar quanto aos seus dados, à sua evolução e imagens recolhidas e analisadas. Tratando-se de uma aplicação de recolha de fotos, análise, interação e assistência entre médicos e pacientes possui como objetivos também permitir que as suas funções sejam desempenhadas da melhor forma possível, para que seja eficiente, e possui também um sistema de notificações para alertar os cuidadores acerca de mudanças e informações relacionadas com os seus utentes.

Alguns dos aspetos/termos referidos anteriormente, nomeadamente a usabilidade, são elementos de grande importância para uma aplicação móvel que se pretende desenvolver, de maneira a corresponder às expectativas e necessidades dos seus utilizadores. Os artigos recolhidos normalmente direcionam as suas aplicações a uma parte da população mundial, ou então um tipo de raça específico, mas preferencialmente a ideia passa por abordar na presente tese uma solução viável para a população no seu global que sofre de psoríase, a doença dermatológica em questão a tratar.

O artigo (Paudel and Chudal, 2020) enquadra-se na pandemia atual, existindo uma zona rural no Nepal em que uma rapariga de 12 anos foi diagnosticada com necrólise epidérmica tóxica induzida por carbamazepina. Trata-se de uma emergência dermatológica com risco de vida e alta mortalidade se não tratada a tempo. Com a inexistência de assistência médica na área, incluindo dermatologistas, a tele dermatologia com o apoio de dispositivos móveis revelou-se fundamental para salvar a rapariga que foi assistida com um diagnóstico e tratamento adequado ao caso específico, sendo portanto muitas vezes usada como solução a muitos casos relativos à pandemia corrente.

O artigo seguinte, (Schreier et al., 2008), direciona-se especificamente para a psoríase, tendo como objetivo a criação de uma aplicação móvel que tem como função permitir aos pacientes partilharem os seus dados hospitalares que sejam necessários, assim como capturar fotos das lesões que possuem e enviarem para os respetivos cuidadores de saúde. O que difere perante as aplicações demonstradas de artigos anteriores, é que permite a geração de alertas e notificações, que são os principais responsáveis pela monitorização dos pacientes por parte dos médicos, que possuem acesso constante aos dados dos seus utentes, podendo mais tarde dar feedback aos mesmos.

A aplicação tem como intuito tornar mais fácil a comunicação entre pacientes e médicos, tal como outras aplicações já referidas, mas apesar dos resultados se terem revelado positivos, no diagnóstico de cada utente continua a ser essencial o papel do dermatologista, algo que deveria ser função da aplicação em questão. Ainda assim, a aplicação possui várias funcionalidades que se enquadram na aplicação que se pretende desenvolver na presente tese, podendo também utilizar *Self-Administered Psoriasis Area Severity Index* (SA-PASI) na avaliação das lesões.

O artigo (Frühauf et al., 2010) diz respeito a um estudo piloto sobre o uso da teledermatologia para monitorizar pacientes de elevado risco com psoríase. Este estudo visa principalmente demonstrar que a teledermatologia é uma opção viável apesar de os pacientes continuarem a confiar mais na monitorização presencial.

Basicamente, pessoas com psoríase e que utilizassem dispositivos móveis foram selecionadas a fim de capturar fotos das suas lesões e registar informações históricas acerca das mesmas. Os pacientes fotografaram as suas lesões tendo em conta a área e o questionário PASI. O *Palmoplantar Psoriasis Area Severity Index* (PPPASI) é outro dos índices também utilizados. Os dados recolhidos são enviados para um servidor web, sendo as instruções de tratamento comunicadas por email e/ou mensagem rápida.

Os resultados obtidos, em comparação com aqueles obtidos por diagnóstico presencial, revelaram-se semelhantes o que indica que a teledermatologia móvel é uma possível solução para monitorizar a gravidade da doença. Maiores estudos serão necessários para confirmar e avaliar o impacto do acompanhamento remoto nos pacientes e a sua influência nos resultados obtidos.

Nos próximos artigos, (Trettin et al., 2021a) e (Trettin et al., 2021b), é retratada uma aplicação móvel desenvolvida para pacientes de psoríase que se encontram a receber tratamento biológico na Dinamarca. O objetivo da aplicação passa por projetar/criar uma solução centrada no paciente com base nas suas necessidades e dos cuidadores de saúde. Pretende-se explorar as experiências e perceções dos pacientes e dos seus cuidadores de saúde sobre o uso de uma aplicação móvel, tornando os pacientes mais ativos, informados e envolvidos nos seus cuidados e fomentando a sua relação com os respetivos cuidadores de saúde. Para além da aplicação se revelar uma mais valia para pessoas afastadas de estabelecimentos hospitalares, permite reduzir os custos de transportes e tornar as consultas mais simples e flexíveis, oferecendo uma maior liberdade para os pacientes que se podem preparar para cada consulta.

A aplicação desenvolvida foi sendo corrigida e melhorada tendo em conta vários testes efetuados com pacientes selecionados, a fim de tornar a aplicação centrada nos mesmos e permitir a comunicação e manutenção dos relacionamentos com os especialistas/médicos que os acompanham. A aplicação móvel possui várias funcionalidades, entre as quais o preenchimento por parte dos pacientes de questionários prévios à consulta, como o questionário *Dermatology Life Quality Index (DLQI)*, permite também a consulta por video entre pacientes e especialistas, a auto-monitorização dos pacientes e o registo de dados dos resultados relatados pelos mesmos.

Todo o estudo efetuado para o desenvolvimento da aplicação diz respeito a uma abordagem de design participativo, que se trata de uma revisão contínua da aplicação até obter uma solução final aceitável, em termos qualitativos. Os resultados revelaram-se bastante positivos, pois a aplicação desenvolvida revelou-se uma solução viável para substituir a necessidade de monitorização e diagnóstico presencial, sendo necessária no entanto a adaptação dos pacientes e cuidadores de saúde a estas novas práticas.

No último artigo, (Koller et al., 2011), mais uma vez a tele dermatologia móvel aparece como possível solução para os pacientes de psoríase a receber tratamento com produtos biológicos. Este tipo de tratamento requer monitorização intensiva, sendo importante o constante contacto entre pacientes e cuidadores de saúde, o que implica várias consultas presenciais e deslocações por parte dos pacientes.

A aplicação móvel desenvolvida possibilita a monitorização remota dos pacientes, permitindo aliviar o fardo da doença no que diz respeito aos elevados custos relativos às consultas cara a cara, e também aumentar a rapidez da comunicação entre pacientes e cuidadores de saúde. Dos pontos cruciais a avaliar, destacam-se a qualidade das fotos da câmara do dispositivo móvel utilizado, o uso do PASI para avaliar as fotos, a gestão de eventos e o feedback dos questionários preenchidos dos pacientes. Tal como no artigo anterior, o intuito da aplicação móvel desenvolvida também passa por fomentar a relação entre pacientes e cuidadores de saúde e a gestão da doença por ambas as partes.

O sistema da aplicação móvel permite a comunicação entre pacientes e cuidadores de saúde, enviando aos últimos os dados preenchidos pelo paciente e as fotos das lesões capturadas. Nas fotos, o questionário PASI avalia a presença de eritema, infiltração e escamação. Os dados são recebidos pelos cuidadores de saúde por métodos *Hypertext Transfer Protocol (HTTP)* que depois analisam a informação recebida e respondem aos respetivos pacientes. Os resultados obtidos revelaram que, caso todas as condições necessárias estejam reunidas e as funcionalidades do aplicativo estejam devidamente operacionais, a aplicação móvel é uma mais valia e uma ferramenta com potencial de vir a

substituir a monitorização presencial graças aos vários benefícios que oferece, sendo o nível de satisfação dos utilizadores bastante elevado.

Para além dos artigos estudados, uma pesquisa das aplicações móveis mais conhecidas e utilizadas atualmente foi efetuada de forma a perceber o que já existe no que se refere à dermatologia, mais especificamente no que diz respeito à monitorização de pacientes com psoríase. Quer as aplicações quer os artigos recolhidos baseiam-se numa lista de funcionalidades e potencialidades que vão de encontro com a aplicação que se pretende desenvolver, dizendo respeito aos seguintes aspetos:

- **Aspeto 1:** Referente às doenças de pele que o respetivo trabalho/projeto monitoriza.
- **Aspeto 2:** Capacidade em identificar as zonas afetadas no corpo por parte do trabalho/projeto respetivo.
- **Aspeto 3:** Relativo às métricas de avaliação que são utilizadas em cada trabalho/projeto.
- **Aspeto 4:** Indica se o respetivo trabalho/projeto permite o cálculo da medicação a ser aplicada pelo paciente.
- **Aspeto 5:** Representa os modelos de usabilidade utilizados para avaliar o trabalho/projeto relativo.

Na tabela seguinte encontra-se representada uma comparação entre os artigos e aplicativos que foram pesquisados e recolhidos de acordo com os aspetos referidos previamente, de forma a validar e comprovar se os projetos analisados possuem os requisitos necessários para a presente aplicação móvel a desenvolver.

Trabalhos/Projetos	Aspeto 1	Aspeto 2	Aspeto 3	Aspeto 4	Aspeto 5
(Clark et al., 2018)	Cancerígenas e malignas	Não	Precisão e Concordância	Não	Não referido
(Moreno-Ramírez and Argenziano, 2017)	Cancerígenas e malignas	Não	Não referido	Não	Não referido
(Rizvi et al., 2020)	Erupções cutâneas, úlceras e lesões de pele	Não	Não referido	Não	Não referido

(Sáenz et al., 2018)	Cancerígenas e malignas	Não	Ontologia para dermatologia	Não	Não referido
(Sáenz et al., 2016)	Cancerígenas e malignas	Não	Ontologia para dermatologia	Não	Não referido
(Vasconcelos and Rosado, 2014)	Cancerígenas e malignas	Sim	Reflexão das imagens	Não	Não referido
(Okuboyejo, 2013)	Cancerígenas e malignas	Sim	Regra ABCD e Multilayer Perceptron	Não	Não referido
(Rosado and Ferreira, 2013)	Cancerígenas e malignas	Sim	Regra ABCD	Não	Não referido
(Becevic et al., 2013)	Cancerígenas e malignas	Não	Não referido	Não	Não referido
(Paudel and Chudal, 2020)	Necrólise epidérmica tóxica	Não	Não referido	Não	Não referido
(Schreier et al., 2008)	Psoríase	Não	SA-PASI	Não	Não referido
(Frühauf et al., 2010)	Psoríase	Não	PASI e PPPASI	Não	Não referido
(Trettin et al., 2021a)	Psoríase	Não	Questionário DLQI	Não	Não referido
(Trettin et al., 2021b)	Psoríase	Não	Questionário DLQI	Não	Não referido
(Koller et al., 2011)	Psoríase	Não	PASI	Não	Não referido
Psoriasis Calc	Psoríase	Não	BSA, PASI, NAPSÍ e Algoritmo de comorbidades	Não	Não referido
Psoriasis App Sorea	Psoríase	Não	Humor e Condição da pele	Não	Não referido

SymTrac™ Psoriasis	Psoríase	Não	Questionário DLQI e Diário de sintomas de psoríase	Não	Não referido
Psoriasis Monitor	Psoríase	Não	Humor, Stress e Condição da pele	Não	Não referido
All Psoriasis App: Manage and Care	Psoríase	Não	Severidade da pele e SA-PASI	Não	Não referido

Tabela 2.1: Trabalhos/Projetos na área da dermatologia e aspetos analisados

2.2 Análise comparativa dos trabalhos/projetos recolhidos

É possível concluir com a análise da tabela anterior que os artigos, apesar da sua maioria se abranger às doenças de pele no seu geral, possuem funcionalidades importantes a ter em conta para a aplicação móvel pretendida, entre as quais o uso de técnicas para identificar e analisar fotos/imagens capturadas pelos dispositivos móveis e o uso de diferentes métricas de avaliação, sendo o questionário PASI o que mais se destaca. Em contraste, as aplicações investigadas apenas se resumem à psoríase, sendo que na sua generalidade se baseiam na captura de fotos das lesões dos pacientes que são avaliadas por diferentes métricas de avaliação e posteriormente comunicadas para os respetivos cuidadores de saúde. As aplicações investigadas e analisadas continuam a possuir limitações no que diz respeito à identificação das áreas afetadas e ao cálculo da medicação a ser aplicada, em que o papel dos médicos/dermatologistas continua a ser preponderante. Nos trabalhos/projetos recolhidos, apesar dos modelos de usabilidade utilizados não serem devidamente referidos/especificados, as aplicações implementadas eram testadas e em muitos casos comparadas com as abordagens cara a cara a fim de observar e avaliar o comportamento das aplicações desenvolvidas.

A dependência que as aplicações móveis continuam a possuir dos cuidadores de saúde é um fator a ter em atenção, sendo que a aplicação que se pretende desenvolver deverá possuir as funcionalidades necessárias para a monitorização dos pacientes, com o acompanhamento mínimo necessário dos respetivos médicos. Para além disso, é importante

que a aplicação siga determinados padrões e regras no que diz respeito ao seu design e arquitetura, sendo que estes detalhes e pormenores se podem revelar imprescindíveis no seu desenvolvimento, servindo de base para fomentar a usabilidade e eficiência da mesma, tornando-a o mais viável possível para os utilizadores.

2.3 Testes de Usabilidade em Sistemas e Aplicações Móveis

Para que a aplicação móvel desenvolvida corresponda às expectativas dos seus utilizadores, é importante que sejam efetuados testes de usabilidade para corrigir os problemas identificados, perceber os aspetos que podem ser melhorados e possibilitar que a aplicação evolua de acordo com as necessidades existentes.

O conceito de usabilidade é o tópico principal do artigo (Kurosu, 2015), em que este termo possui várias definições de diferentes autores e padrões *International Organization for Standardization* (ISO). Vários modelos de usabilidade foram criados e desenvolvidos de maneira a caracterizar em que consiste o termo usabilidade e como esta pode ser corretamente avaliada.

No artigo (Zahra et al., 2017), é apresentada uma abordagem geral da avaliação da usabilidade nas aplicações móveis na atualidade, sendo desenvolvidos diferentes modelos para analisar o sistema. Muitos modelos são muito genéricos, adaptando-se a diferentes tipos de aplicações, sendo que como a tecnologia se encontra em constante evolução, os modelos acabam por se revelar insuficientes para avaliar de maneira complexa e eficaz os diversos aplicativos existentes e desenvolvidos. Os modelos customizados surgem portanto como uma possível solução para este problema.

Um dos modelos principais referidos neste artigo, que também é especificamente retratado no artigo (Harrison et al., 2013) também recolhido, trata-se do modelo *People At the Centre of Mobile Application Development* (PACMAD), que foi desenvolvido a partir do modelo de usabilidade de Nielsen, com o propósito de solucionar as limitações dos modelos de usabilidade existentes para aplicações móveis, levando à necessidade da elaboração de um modelo mais complexo. Jakob Nielsen é considerado um dos maiores gurus da usabilidade, sendo que escreveu vários livros como (Nielsen, 1997), e o seu modelo de usabilidade foi aquele que provavelmente teve o maior impacto no pensamento de usabilidade em geral. O termo usabilidade possui várias definições de acordo com os diferentes modelos e normas existentes, como se pode verificar no artigo (Abran et al., 2003). Segundo Nielsen, a usabilidade é um atributo de qualidade para avaliar a facilidade de uso de uma interface, sendo que a própria palavra também se refere a métodos para melhorar a facilidade de uso durante o processo de design. Segundo o seu modelo,

Nielsen dividiu a usabilidade em cinco componentes:

- **Eficiência:** Referente ao comportamento do utilizador para atingir um objetivo. É possível medir esse atributo analisando quanto tempo um usuário leva para completar uma tarefa.
- **Satisfação:** Possui relação com o sentimento e nível de satisfação do utilizador. É o atributo mais qualitativo a avaliar.
- **Aprendizagem:** Significa o quão é fácil aprender, ou o quão é fácil atingir um determinado objetivo. Este atributo pode ser medido em relação à eficácia e eficiência na primeira tentativa bem sucedida.
- **Memorização:** Representa a capacidade do utilizador adquirida ao longo de um determinado período de tempo. É possível quantificar este atributo analisando a eficácia e eficiência em momentos distintos, com um intervalo de tempo entre eles.
- **Erros:** Indica a tendência do utilizador para cometer erros durante o uso da aplicação, e como esses erros são tratados. Podemos verificar este atributo analisando o número de erros que ocorrem.

Como referido anteriormente, o modelo PACMAD herdou características do modelo de usabilidade criado por Nielsen, mais propriamente os padrões ISO e outras propriedades do modelo, com o objetivo de criar um modelo de usabilidade mais complexo para solucionar os problemas correntes que as aplicações móveis possuem atualmente. Segundo o artigo (Zahra et al., 2017), o modelo PACMAD foi criado em 2013, tendo sido melhorado posteriormente em 2015, com o intuito de criar um modelo de usabilidade unicamente exclusivo para os dispositivos móveis, trazendo especificações e atributos que não existiam em modelos existentes. Este modelo contempla três fatores muito importantes a ter em conta, que são o utilizador, a tarefa e o contexto. O utilizador refere-se ao público alvo para quem a aplicação se direciona, sendo importante o conhecer assim como especificações e limitações da própria aplicação. A tarefa representa os recursos necessários para atingir o objetivo assim como o seu nível de complexidade. Por fim, o contexto está relacionado com o uso da aplicação respetiva, ou seja, como o utilizador vai usar a aplicação e em que condições. Estes fatores derivam do modelo ISO e Nielsen, que possuem atributos fundamentais para o modelo PACMAD.



Figura 2.1: Modelos de usabilidade principais

Na figura anterior, podemos visualizar os principais modelos de usabilidade referidos anteriormente, assim como os atributos que cada um deles possui, o que em suma representa a evolução dos modelos de usabilidade. O modelo de usabilidade PACMAD encontra-se presente na avaliação de inúmeras aplicações móveis desenvolvidas de diferentes áreas de negócio, como por exemplo na área comercial/marketing como retrata o artigo (Az-zahra et al., 2019), na monitorização de jovens autistas no artigo (Ghabban et al., 2021), na indústria de construção no artigo (Jailani et al., 2015) e também no ramo hospitalar e/ou área da saúde no artigo (Kasali et al., 2020). Este modelo também se destaca por contribuir para validar/confirmar que as diferentes aplicações desenvolvidas se adaptam aos diversos tipos de dispositivos e plataformas móveis existentes, como se pode verificar nos artigos (do Nascimento Mendes and Dias-Neto, 2016) e (Cho and Huh, 2017), permitindo ultrapassar limitações existentes nas aplicações móveis das diferentes áreas/ramos.

Apesar da utilidade do modelo PACMAD, ainda continuam a existir limitações e problemas nas aplicações móveis, sendo por isso necessário customizar e construir um modelo mais robusto e completo, originando o modelo extendido PACMAD para avaliação de métricas de usabilidade usando uma abordagem *Goal Question Metric* (GQM), isto é, incluindo o modelo GQM, retratado e explicado nos artigos (Basili et al., 1994) e (Dalton, 2019), de maneira a adicionar outras métricas essenciais e seguir orientações específicas e bem delineadas. Em suma, GQM diz respeito a uma abordagem orientada a objetivos estabelecida para métricas de software para melhorar e medir a qualidade do software.

Este novo modelo PACMAD estendido encontra-se retratado no artigo (Fabil, 2015), sendo que este modelo melhorado e incorporado acrescenta orientações para o uso de métricas de maneira a avaliar cada atributo correspondente, assim como a utilização de um questionário de satisfação do utilizador para uma avaliação completa da usabilidade. Para além disso, este modelo conta com o auxílio de questões que correspondem aos objetivos que a aplicação pretende cumprir, sendo que as métricas de usabilidade do PACMAD se inserem nos objetivos referentes a cada questão que a aplicação irá responder.

A usabilidade da aplicação que se pretende desenvolver será baseada neste modelo mais complexo, sendo que inicialmente após revisão da informação proveniente dos artigos estudados, é essencial definir os requisitos da aplicação móvel a desenvolver, de maneira a abordar e permitir que o utilizador possa cumprir os principais objetivos e funções do tema da presente dissertação.

Capítulo 3

Sistema PSOCAT

Neste capítulo será abordado o sistema desenvolvido e em particular a aplicação móvel. Serão apresentados inicialmente os requisitos do sistema, seguida da sua arquitetura e por fim alguns aspetos da implementação da aplicação e dos seus serviços.

O sistema *Psoriasis Capture Analyze Treat* (PSOCAT) integra vários componentes cuja interação permite oferecer um conjunto de funcionalidades para o suporte e monitorização de pessoas com psoríase. Deste modo o sistema deve responder às interações do utilizador com a aplicação móvel, de forma eficiente e garantindo uma troca de informação segura e persistente.

O processo de engenharia de software utilizado na criação deste sistema (Sommerville, 2011) seguiu uma metodologia tradicional com um conjunto de atividades relacionadas para levar à produção do produto de software final. Começou-se pelo levantamento e especificação dos requisitos e restrições iniciais do funcionamento do sistema e posteriormente desenhou-se a sua arquitetura e procedeu-se à implementação de acordo com as especificações que foram sendo refinadas num processo iterativo conjunto.

3.1 Requisitos do Sistema PSOCAT

Foram inicialmente identificados os vários requisitos que o sistema deveria possuir de forma a satisfazer as necessidades dos seus utilizadores, nomeadamente os requisitos funcionais presentes nos casos de uso, bem como os requisitos não funcionais. Os requisitos especificados serão apresentados de seguida, começando pelos requisitos funcionais.

3.1.1 Requisitos Funcionais

Para garantir uma execução correta das funcionalidades da aplicação, os requisitos funcionais desempenham um papel fundamental. Um requisito funcional diz respeito a uma

necessidade por parte do utilizador a que o sistema deverá corresponder, definindo determinada funcionalidade associada, independentemente de como esta é desenvolvida.

Identificador	RF01
Nome	Permitir ao utilizador efetuar o registo através da aplicação
Descrição	Cada utilizador poderá registar-se através da aplicação. Após executar a aplicação, o utilizador poderá selecionar a secção de registo apenas necessitando de introduzir a informação necessária, como o nome de utilizador, a password pretendida e o seu endereço de email. Deverá também indicar o tipo de utilizador que pretende ser, isto é, paciente ou médico. Se todos os dados inseridos estiverem corretos, o registo será completado.
Caso de Uso associado	[UC01]

Tabela 3.1: Requisito Funcional 1: Permitir ao utilizador efetuar o registo através da aplicação

Identificador	RF02
Nome	Permitir ao utilizador selecionar a língua/idioma da aplicação
Descrição	Após entrar na aplicação, o utilizador poderá selecionar a secção das definições e optar pela língua/idioma (cf. Inglês ou Português).
Caso de Uso associado	[UC02]

Tabela 3.2: Requisito Funcional 2 :Permitir ao utilizador selecionar a língua/idioma da aplicação

Identificador	RF03
Nome	Permitir ao utilizador efetuar o login na aplicação
Descrição	Para se autenticar na aplicação, inicialmente o utilizador deverá selecionar a secção de login e inserir as credenciais necessárias, ou seja, o nome de utilizador e a sua password. Se os dados forem validados, o login é completado com sucesso.
Caso de Uso associado	[UC03]

Tabela 3.3: Requisito Funcional 3: Permitir ao utilizador efetuar o login na aplicação

Identificador	RF04
Nome	Permitir ao utilizador autenticar-se com os dados de um login feito previamente
Descrição	Após autenticação bem sucedida por parte do utilizador, se este mais tarde tentar autenticar-se novamente, os dados do login anterior são guardados, necessitando apenas de inserir a sua password e após validação, entrar de novo na sua área privada.
Caso de Uso associado	[UC04]

Tabela 3.4: Requisito Funcional 4: Permitir ao utilizador autenticar-se com dados de um login feito previamente

Identificador	RF05
Nome	Permitir ao utilizador efetuar o logout seguro da aplicação
Descrição	Na aplicação cada utilizador, seja paciente ou médico, poderá efetuar com segurança o logout da aplicação, dirigindo-se à opção para efetuar o logout, independentemente de algum erro acontecer, saindo da sua área privada sem qualquer tipo de risco de perda de dados.
Caso de Uso associado	[UC05]

Tabela 3.5: Requisito Funcional 5: Permitir ao utilizador efetuar o logout seguro da aplicação

Identificador	RF06
Nome	Permitir ao utilizador visualizar a sua página inicial
Descrição	Após se autenticar, o utilizador irá visualizar a página inicial com dados acerca do mesmo na aplicação, podendo também aceder a este ecrã através do menu lateral.
Caso de Uso associado	[UC06]

Tabela 3.6: Requisito Funcional 6: Permitir ao utilizador visualizar a sua página inicial

Identificador	RF07
Nome	Permitir ao utilizador editar o seu perfil
Descrição	Já autenticado, o utilizador poderá aceder à sua secção de perfil com a sua informação pessoal, podendo a editar e salvar a informação alterada.
Caso de Uso associado	[UC07]

Tabela 3.7: Requisito Funcional 7: Permitir ao utilizador editar o seu perfil

Identificador	RF08
Nome	Permitir ao utilizador capturar e fazer <i>upload</i> de fotos das suas lesões
Descrição	Na aplicação, deverá ser possível que o utilizador capture fotografias das suas lesões, através da câmara do seu dispositivo móvel, sendo possível após captura descartar essas fotografias ou adicionar à sua lista de fotografias persistentes.
Caso de Uso associado	[UC08]

Tabela 3.8: Requisito Funcional 8: Permitir ao utilizador capturar e fazer *upload* de fotos das suas lesões

Identificador	RF09
Nome	Permitir ao utilizador visualizar a sua lista de fotos
Descrição	Cada utilizador pode dirigir-se à secção da lista de fotos guardadas das suas lesões e visualizar a informação de cada uma.
Caso de Uso associado	[UC09]

Tabela 3.9: Requisito Funcional 9: Permitir ao utilizador visualizar a sua lista de fotos

Identificador	RF10
Nome	Permitir ao utilizador eliminar uma foto previamente guardada
Descrição	Na lista de fotos, o utilizador pode apagar uma determinada foto previamente guardada na aplicação, sendo a lista atualizada posteriormente.
Caso de Uso associado	[UC10]

Tabela 3.10: Requisito Funcional 10: Permitir ao utilizador eliminar uma foto previamente guardada

Identificador	RF11
Nome	Permitir ao utilizador realizar o questionário PASI para detalhar e avaliar as suas lesões
Descrição	Na secção da calculadora PASI, o utilizador poderá preencher um questionário para avaliar as suas lesões e depois guardar no histórico da aplicação.
Caso de Uso associado	[UC11]

Tabela 3.11: Requisito Funcional 11: Permitir ao utilizador efetuar um questionário PASI para detalhar e avaliar as suas lesões

Identificador	RF12
Nome	Permitir ao utilizador visualizar o histórico das avaliações PASI efetuadas
Descrição	Cada utilizador pode dirigir-se à secção do histórico das avaliações PASI guardadas e visualizar a informação de cada uma.
Caso de Uso associado	[UC12]

Tabela 3.12: Requisito Funcional 12: Permitir ao utilizador visualizar o histórico das avaliações PASI efetuadas

Identificador	RF13
Nome	Permitir ao utilizador eliminar uma avaliação previamente guardada
Descrição	No histórico, o utilizador pode apagar uma determinada avaliação previamente guardada na aplicação, sendo a lista atualizada posteriormente.
Caso de Uso associado	[UC13]

Tabela 3.13: Requisito Funcional 13: Permitir ao utilizador eliminar uma avaliação previamente guardada

Identificador	RF14
Nome	Permitir ao utilizador visualizar mais detalhes sobre cada avaliação
Descrição	No histórico, o utilizador possui uma opção para poder visualizar mais detalhes sobre cada avaliação previamente guardada na aplicação.
Caso de Uso associado	[UC14]

Tabela 3.14: Requisito Funcional 14: Permitir ao utilizador visualizar mais detalhes sobre cada avaliação

Identificador	RF15
Nome	Permitir ao utilizador visualizar a sua lista de eventos relacionados com a sua medicação
Descrição	Cada utilizador pode dirigir-se à secção da lista de eventos do calendário relacionados com a medicação a tomar/aplicar e visualizar cada um.
Caso de Uso associado	[UC15]

Tabela 3.15: Requisito Funcional 15: Permitir ao utilizador visualizar a sua lista de eventos relacionados com a sua medicação

Identificador	RF16
Nome	Permitir ao utilizador ver/editar determinado evento selecionado
Descrição	Na lista de eventos, o utilizador possui uma opção que o irá redirecionar para cada evento no calendário, onde para além de o poder visualizar mais detalhadamente, também o poderá editar.
Caso de Uso associado	[UC16]

Tabela 3.16: Requisito Funcional 16: Permitir ao utilizador ver/editar determinado evento selecionado

Serão utilizados diferentes diagramas de casos de uso para melhor detalhar e documentar a funcionalidade específica de cada requisito. Um diagrama de casos de uso representa basicamente um cenário que descreve determinada funcionalidade, através das interações entre o utilizador e o sistema, assim como os passos e dependências existentes.

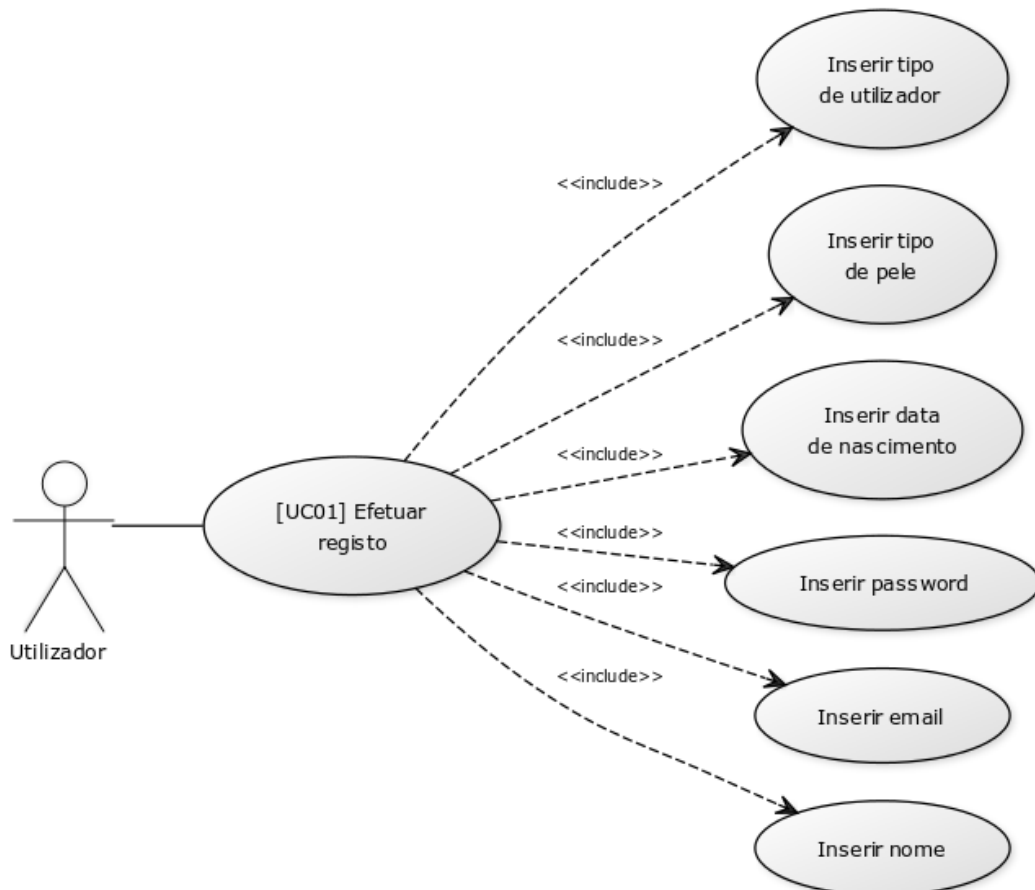


Figura 3.1: Caso de Uso 1: Permitir ao utilizador efetuar o registo através da aplicação

Como é possível observar no primeiro caso de uso acima, a funcionalidade de registo (Efetuar registo) inclui ações/etapas necessárias para ser completada (Inserir nome, Inserir email, etc..). Os restantes casos de uso funcionam de maneira similar, incluindo passos necessários para a sua completude.

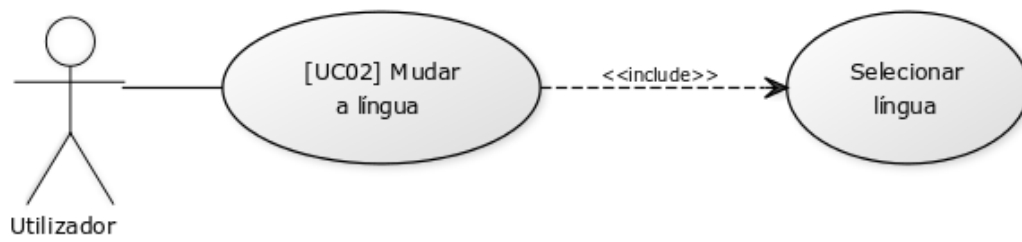


Figura 3.2: Caso de Uso 2: Permitir ao utilizador mudar a língua/idioma da aplicação

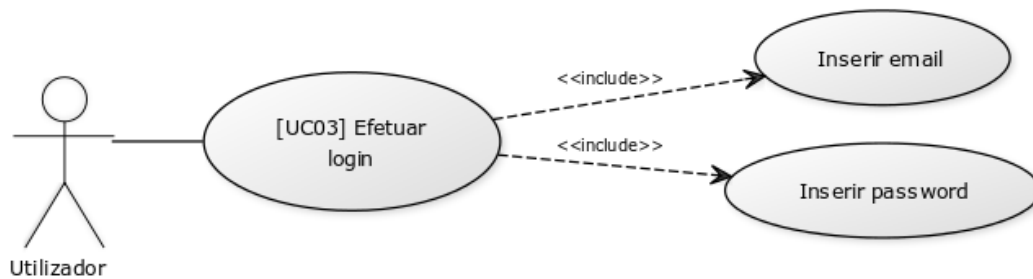


Figura 3.3: Caso de Uso 3: Permitir ao utilizador efetuar o login na aplicação

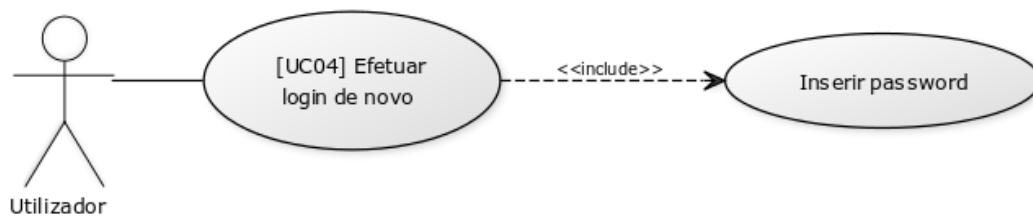


Figura 3.4: Caso de Uso 4: Permitir ao utilizador autenticar-se com dados de um login feito previamente

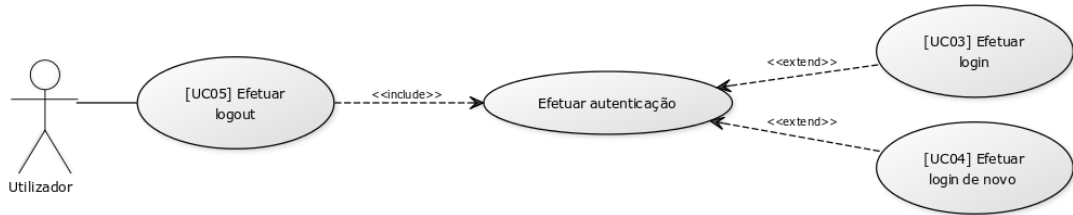


Figura 3.5: Caso de Uso 5: Permitir ao utilizador efetuar o logout seguro da aplicação

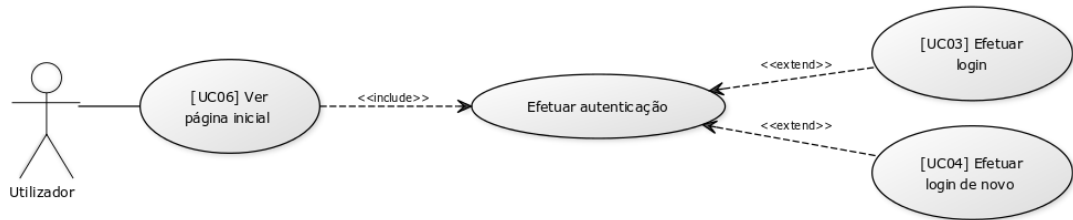


Figura 3.6: Caso de Uso 6: Permitir ao utilizador visualizar a sua página inicial

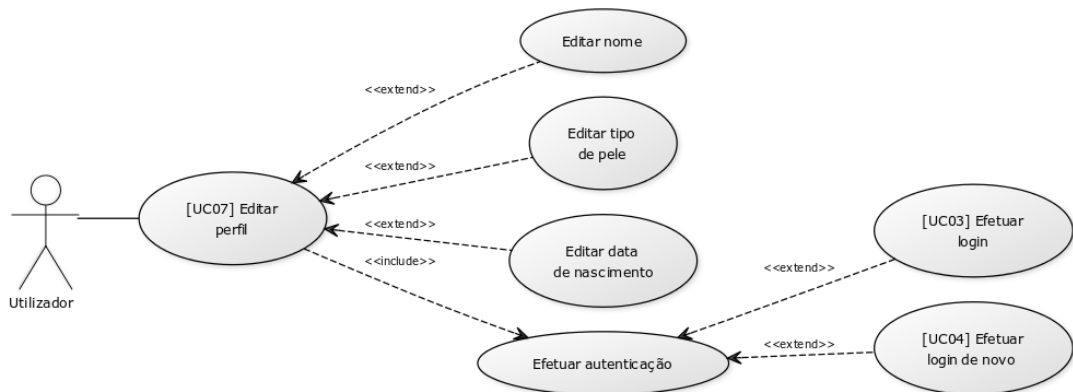


Figura 3.7: Caso de Uso 7: Permitir ao utilizador editar o seu perfil

Alguns casos de uso, tal como os anteriores, possuem também passos opcionais, ou seja, que não são necessários para a completude da tarefa em questão. Por exemplo, na funcionalidade anterior (Editar perfil) o utilizador pode ou não alterar alguns dos seus dados pessoais (Editar nome, Editar tipo de pele, etc..).

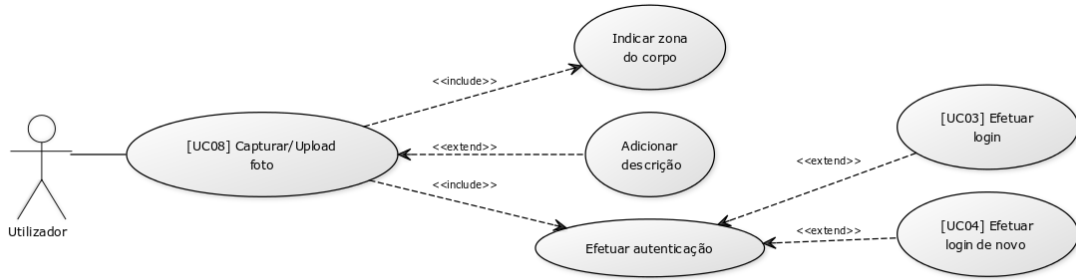


Figura 3.8: Caso de Uso 8: Permitir ao utilizador capturar e fazer *upload* de fotos das suas lesões

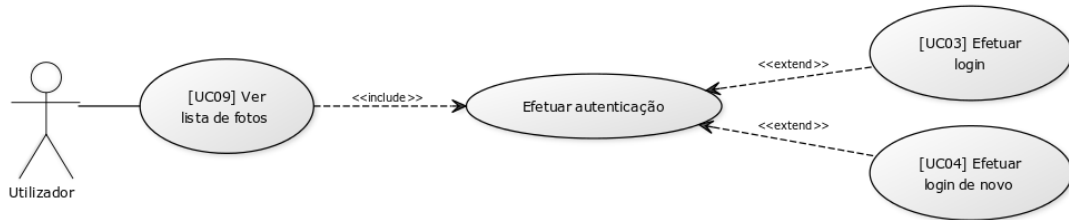


Figura 3.9: Caso de Uso 9: Permitir ao utilizador visualizar a sua lista de fotos

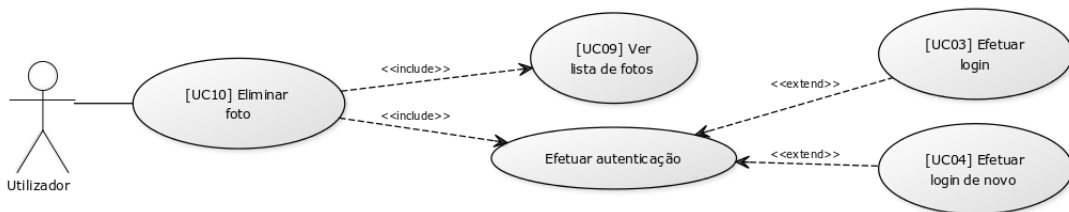


Figura 3.10: Caso de Uso 10: Permitir ao utilizador eliminar uma foto previamente guardada

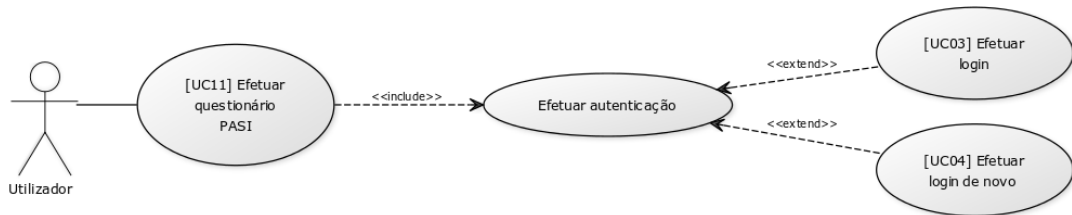


Figura 3.11: Caso de Uso 11: Permitir ao utilizador realizar o questionário PASI para detalhar e avaliar as suas lesões

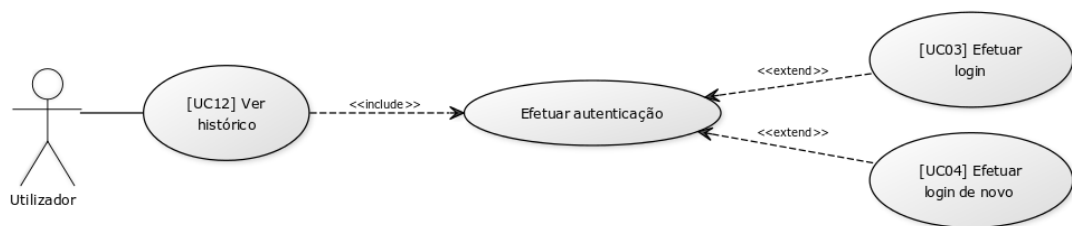


Figura 3.12: Caso de Uso 12: Permitir ao utilizador visualizar o histórico das avaliações PASI efetuadas

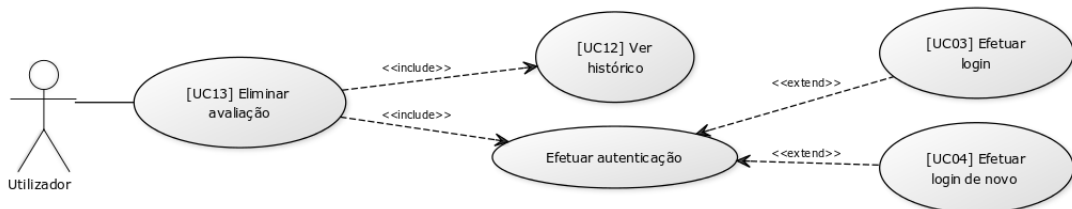


Figura 3.13: Caso de Uso 13: Permitir ao utilizador eliminar uma avaliação previamente guardada

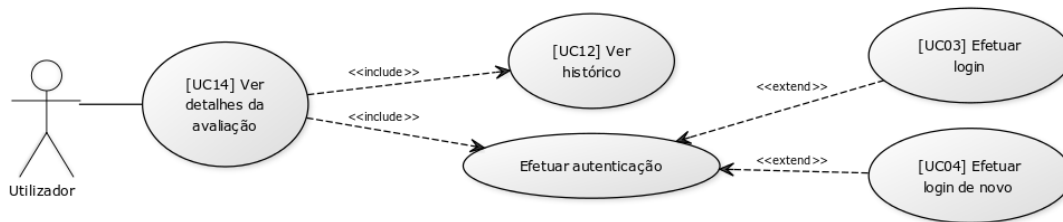


Figura 3.14: Caso de Uso 14: Permitir ao utilizador visualizar mais detalhes sobre cada avaliação

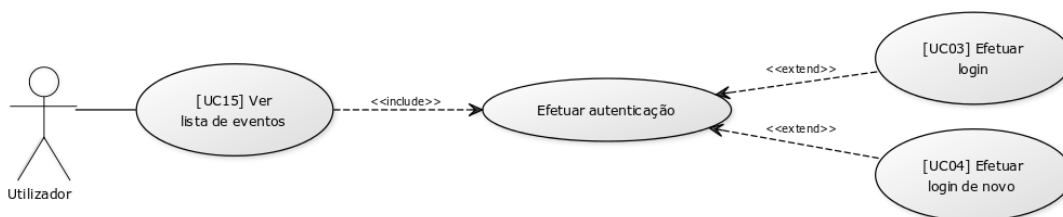


Figura 3.15: Caso de Uso 15: Permitir ao utilizador visualizar a sua lista de eventos relacionados com a sua medicação

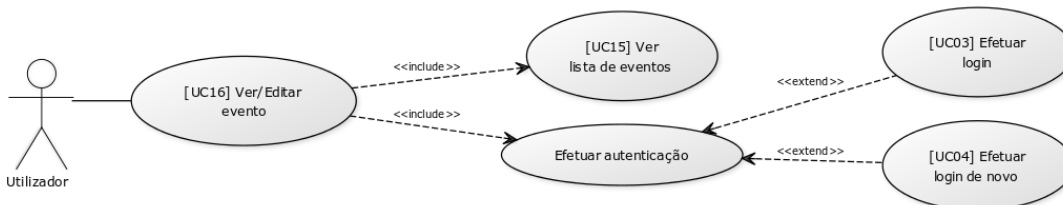


Figura 3.16: Caso de Uso 16: Permitir ao utilizador ver/editar determinado evento selecionado

3.1.2 Requisitos Não Funcionais

Outro tipo de requisitos igualmente importantes para o sistema desenvolvido são os requisitos não funcionais. Um requisito não funcional define determinado atributo de qualidade ou restrição relativa a características ou comportamentos do sistema, de modo a que este possa cumprir a execução das funcionalidades referentes aos requisitos funcionais. Neste caso, dizem respeito a requisitos por parte do sistema móvel, por exemplo, de forma a

tornar a utilização da aplicação móvel satisfatória para os utilizadores, em termos de usabilidade, performance, segurança, entre outras características.

Identificador	RNF01
Nome	O sistema deverá permitir uma interação simples e intuitiva
Categoria	Usabilidade
Descrição	A aplicação deverá ser fácil de usar pelo utilizador, permitindo navegar e aceder a todas as funcionalidades sem dificuldades de orientação.

Tabela 3.17: Requisito Não Funcional 1: O sistema deverá permitir uma interação simples e intuitiva

Identificador	RNF02
Nome	O sistema deverá ser disponibilizado em Português e Inglês
Categoria	Multilinguismo
Descrição	A aplicação deverá oferecer duas línguas/idiomas (cf. Português e Inglês), suportando a comunidade de língua oficial Portuguesa e adaptar-se a uma potencial utilização mais universal.

Tabela 3.18: Requisito Não Funcional 2: O sistema deverá ser disponibilizado em Português e Inglês

Identificador	RNF03
Nome	O sistema deverá restringir as credenciais do utilizador
Categoria	Segurança
Descrição	A aplicação deverá, no que diz respeito ao registo de novos utilizadores, restringir o tipo de caracteres usados no nome do utilizador e password, assim como o limite de caracteres máximo.

Tabela 3.19: Requisito Não Funcional 3: O sistema deverá restringir as credenciais do utilizador

Identificador	RNF04
Nome	O sistema deverá permitir a gestão de permissões da app
Categoria	Segurança
Descrição	A aplicação deverá permitir ao utilizador gerir as permissões no sistema android, para que este possa aceitar ou rejeitar o acesso a dados do dispositivo por parte da aplicação.

Tabela 3.20: Requisito Não Funcional 4: O sistema deverá permitir a gestão de permissões da app

Identificador	RNF05
Nome	O sistema deverá ser compatível com dispositivos android desde a <i>Application Programming Interface</i> (API) 24 até à mais recente
Categoria	Compatibilidade
Descrição	A aplicação deverá ser compatível com qualquer tipo de dispositivo android desde a API mínima especificada até às plataformas mais recentes.

Tabela 3.21: Requisito Não Funcional 5: O sistema deverá ser compatível com dispositivos android desde a API 24 até à mais recente

Identificador	RNF06
Nome	O sistema deverá possuir um tempo de resposta rápido
Categoria	Desempenho
Descrição	A aplicação deverá ser responsiva face às interações do utilizador com a aplicação, ou seja, deve responder com rapidez e sem tempos mortos que indiquem bloqueamentos.

Tabela 3.22: Requisito Não Funcional 6: O sistema deverá possuir um tempo de resposta rápido

Identificador	RNF07
Nome	O sistema deverá permitir encriptação de dados privados
Categoria	Segurança
Descrição	A aplicação deverá permitir a encriptação de dados do perfil, utilizando criptografia com os dados de registo e autenticação.

Tabela 3.23: Requisito Não Funcional 7: O sistema deverá permitir encriptação de dados privados

Identificador	RNF08
Nome	O sistema deverá ter alta disponibilidade para os seus utilizadores
Categoria	Disponibilidade
Descrição	O sistema deverá estar constantemente disponível para os seus utilizadores, estando o seu servidor operacional 24 horas por dia, para que o acesso aos seus dados seja possível.

Tabela 3.24: Requisito Não Funcional 8: O sistema deverá ter alta disponibilidade para os seus utilizadores

Identificador	RNF09
Nome	O sistema deverá possuir acesso à internet
Categoria	Disponibilidade
Descrição	A aplicação deverá estar conectada à internet para que esta possa comunicar com o servidor, devendo avisar o utilizador quando não se encontra ligada, podendo manter os dados localmente.

Tabela 3.25: Requisito Não Funcional 9: O sistema deverá possuir acesso à internet

Identificador	RNF10
Nome	O sistema deverá manter o utilizador informado sobre erros e/ou falhas que ocorram
Categoria	Confiabilidade
Descrição	A aplicação deverá informar o utilizador caso ocorra algum erro durante determinada ação/tarefa e/ou ao chamar determinado serviço.

Tabela 3.26: Requisito Não Funcional 10: O sistema deverá manter o utilizador informado sobre erros e/ou falhas que ocorram

Identificador	RNF11
Nome	O sistema deverá permitir ao utilizador atualizar a versão da aplicação
Categoria	Manutenibilidade
Descrição	A aplicação deverá automaticamente verificar a sua versão atual e permitir ao utilizador atualizar a aplicação corrente a partir da loja.

Tabela 3.27: Requisito Não Funcional 11: O sistema deverá permitir ao utilizador atualizar a versão da aplicação

Identificador	RNF12
Nome	O sistema não deverá permitir ao utilizador efetuar a captura de ecrã na aplicação
Categoria	Segurança
Descrição	A aplicação não deverá permitir que o utilizador efetue a captura de ecrã em qualquer parte da aplicação, para impedir guardar essa informação no dispositivo android.

Tabela 3.28: Requisito Não Funcional 12: O sistema não deverá permitir ao utilizador efetuar a captura de ecrã na aplicação

Após a análise e definição de requisitos, é possível iniciar a construção do modelo conceitual relativo à arquitetura do sistema a desenvolver. O próximo passo trata-se portanto de edificar os componentes principais do sistema e permitir a sua interação.

3.2 Arquitetura do Sistema PSOCAT

A arquitetura do sistema é baseada na integração da aplicação android com um servidor *backend*, que pode estar alocado remotamente ou numa máquina virtual. Este servidor, por sua vez, comunica com uma base de dados para obtenção da informação necessária, previamente solicitada pela aplicação, que é posteriormente devolvida pelo servidor.

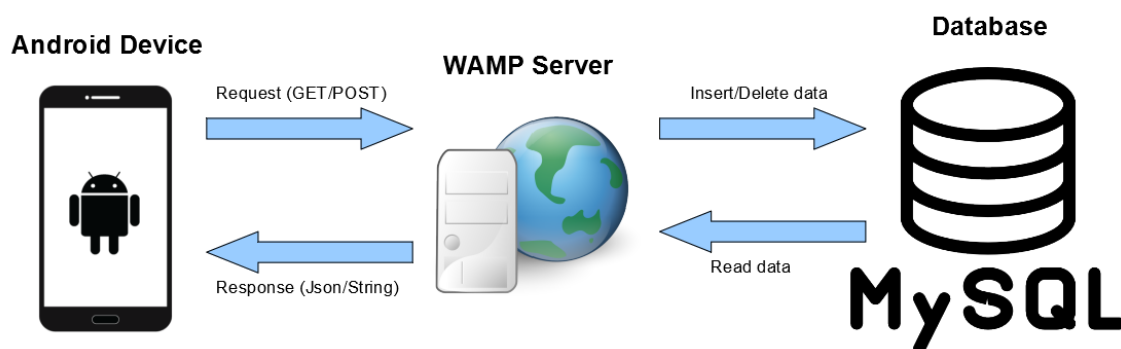


Figura 3.17: Arquitetura do Sistema PSOCAT

A figura acima representa um esboço de como se desenrola a comunicação entre os componentes principais do sistema desenvolvido. A arquitetura do sistema PSOCAT pode ser representada por um modelo composto por três níveis/camadas. A primeira camada diz respeito à interface com a qual o utilizador poderá interagir, ou seja a aplicação móvel

desenvolvida, sendo esta responsável pelo processamento das entradas e eventos de interfaces, como botões, introdução de texto, etc. A segunda camada está relacionada com a lógica de negócio do sistema que se encontra implementada no código da aplicação e no servidor que é constituído por Windows, Apache, MySQL e PHP (WAMP). Esta camada é a grande responsável pelas respostas ao utilizador por parte da camada relativa à interface. A terceira e última camada trata-se do banco de dados do sistema utilizada, 'My' *Structured Query Language* (MySQL), que armazena os dados manipulados pelo sistema, que são acedidos pela camada anterior. Na figura seguinte podemos observar o modelo da arquitetura em camadas do sistema PSOCAT, sendo necessário primeiramente tratar da arquitetura de cada componente do sistema.

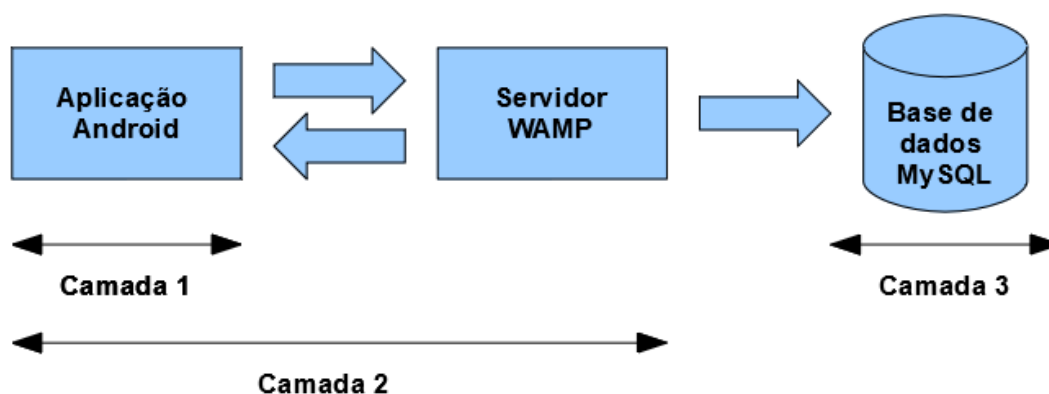


Figura 3.18: Modelo da Arquitetura em camadas do Sistema PSOCAT

3.2.1 Arquitetura do Backend

O *backend* do sistema diz respeito à parte da aplicação com a qual os utilizadores não vão poder interagir nem visualizar. A arquitetura do *backend* é constituída por um servidor WAMP, que se trata de um software que cria um ambiente de desenvolvimento PHP, MySQL e Apache, sendo que este último se refere ao tipo de servidor web usado, simples e gratuito. Foram pesquisados alguns artigos sobre aplicações móveis que utilizam uma abordagem semelhante na parte do *backend* ((Popovici et al., 2010) e (Watkari and Karmore, 2015)), auxiliando na elaboração do *backend* da aplicação desenvolvida.

O servidor WAMP pode ser alocado num servidor remoto ou numa máquina local. Recebe os serviços/pedidos HTTP efetuados a partir da aplicação, que se baseiam em métodos GET/POST para obter a informação enviada posteriormente pela base de dados associada. Um pedido GET é utilizado normalmente para solicitar determinada informação sobre certo recurso, enquanto que um pedido POST envia dados no seu pedido para várias

ações, como criar, obter, atualizar e/ou eliminar informação sobre determinado recurso.

Posteriormente comunica com uma base de dados MySQL, através de pedidos feitos em PHP, para inserir/ler e/ou eliminar dados da mesma relativamente a cada utilizador, sendo estes verificados e validados pela base de dados que depois retorna a informação pretendida. O PHP lê a informação devolvida e retorna uma resposta, em formato *JavaScript Object Notation* (JSON) ou *String*, desses dados para a aplicação móvel, que depois trata da resposta de acordo com o serviço específico. É possível visualizar na figura seguinte o papel do *backend* para o funcionamento do sistema, desde que recebe os pedidos do utilizador através da interface da aplicação móvel até que a resposta por parte do servidor é retornada.

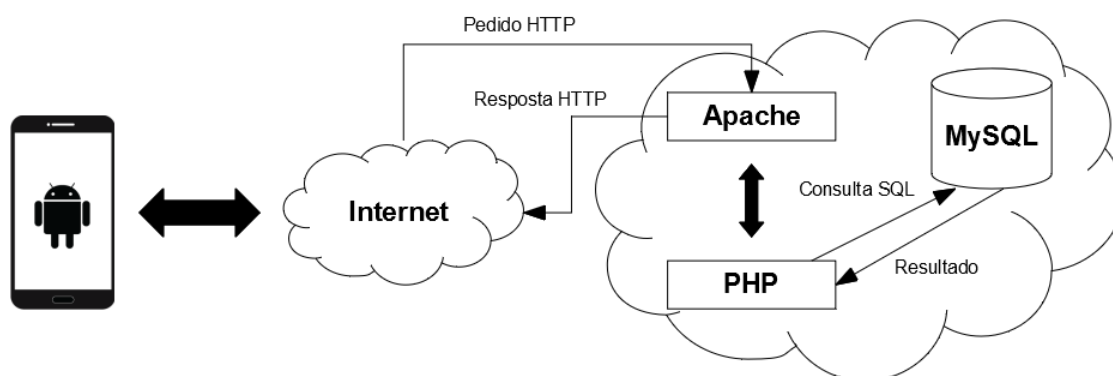


Figura 3.19: Arquitetura do Backend do Sistema PSOCAT

No que diz respeito à base de dados, a sua criação e estrutura final pode ser explicada por um modelo de Entidade-Relação (E-R), que basicamente se trata de um diagrama que ilustra como as várias entidades, que podem ser por exemplo pessoas, objetos ou conceitos, se relacionam entre si dentro de um sistema. Estes diagramas utilizam diferentes formas para representar a interação entre as entidades, assim como os seus relacionamentos e atributos.

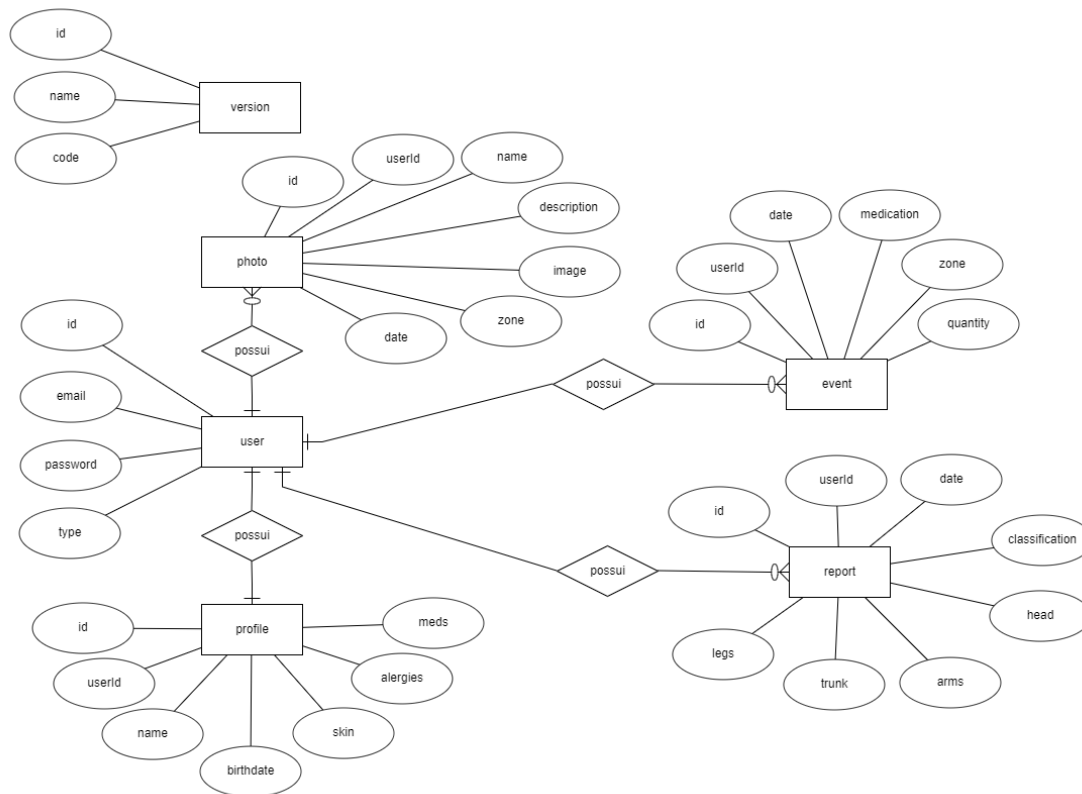


Figura 3.20: Diagrama E-R da base de dados MySQL do Sistema PSOCAT

No diagrama anterior os rectângulos correspondem às entidades (*user*, *profile*, *version*, *etc*), os atributos estão representados por círculos (*id*, *name*, *email*, *password*, *etc*) e as relações entre entidades dizem respeito aos losangos. Para exemplificar, como podemos observar a entidade *user*, representada por um rectângulo, possui diferentes atributos como o *type*, a *password*, o *email* e o *id*, todos eles representados por círculos e possui uma entidade *profile* sendo esta relação representada por um losango.

3.2.2 Arquitetura da Aplicação Móvel

No que diz respeito à arquitetura da aplicação móvel, é crucial que a sua construção seja bem estruturada e organizada para que, não só esteja em termos de programação bem construída, mas também que se revele eficiente e simples de usar por parte dos seus utilizadores. Esta parte da arquitetura corresponde ao *frontend* do sistema, ou seja, a parte da aplicação e/ou interface visível por parte do utilizador com a qual este poderá interagir, comunicando com o *backend* para saber que respostas devolver aos seus usuários.

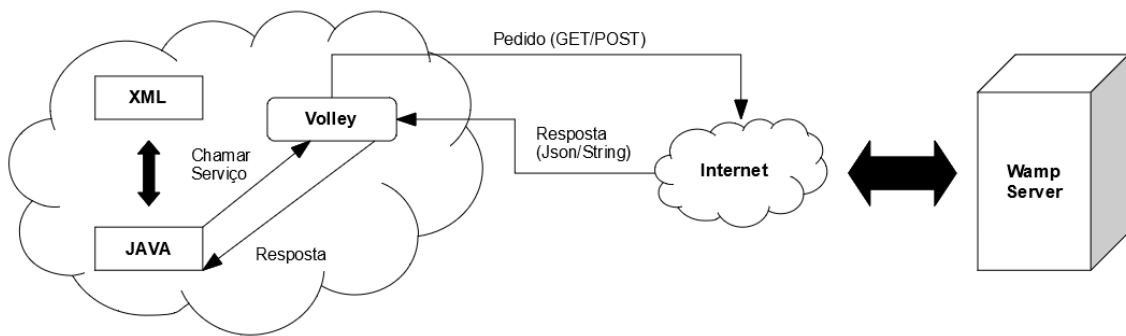


Figura 3.21: Arquitetura do Frontend do Sistema PSOCAT

Na figura acima podemos visualizar o papel do *frontend* no sistema desenvolvido, sendo que o utilizador interage com a interface gráfica da aplicação móvel desenvolvida, normalmente associada ao *Extensible Markup Language* (XML) que comunica com o código JAVA da aplicação. O JAVA obtém a informação do *backend* através da livreria HTTP Volley cuja função principal é executar pedidos de métodos GET/POST, sendo que posteriormente o servidor devolve as respostas à aplicação em formato JSON ou *String*.

Relativamente à estrutura da aplicação android, esta segue um padrão de arquitetura muito comum e tradicional, *Model View Controller* (MVC), sendo uma das escolhas mais utilizadas no que diz respeito ao desenvolvimento de aplicações móveis, como podemos verificar no artigo (Thakur et al., 2020). Segundo o MVC, a aplicação possui 3 componentes, a vista que representa a interface da aplicação apresentando as informações de forma visual ao utilizador, o modelo que diz respeito à lógica de negócio sendo responsável por gerir a forma como os dados se comportam por meio das funções, lógica e regras estabelecidas e por fim o controlador que serve de ligação entre o modelo e a vista, sendo o intermediário entre os pedidos recebidos pela vista e as respostas fornecidas pelo modelo.

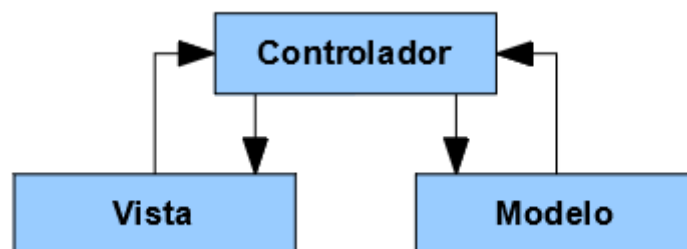


Figura 3.22: Arquitetura MVC em android do Sistema PSOCAT

Relativamente à aplicação android, a vista é representada pelos ecrãs XML da aplicação, designados de *layouts*, que fazem parte das atividades e/ou fragmentos e possuem recursos visuais que permitem capturar as informações do utilizador. As atividades e os fragmentos da aplicação fazem parte do controlador, recebendo os dados enviados pelo utilizador e transmitindo-os para o modelo. O modelo é constituído pela Volley API com a qual o controlador comunica e que por sua vez é responsável pelos pedidos HTTP ao servidor que armazena e/ou obtém a informação necessária da base de dados MySQL, que também pertence ao modelo da aplicação. Por fim, o modelo é responsável por devolver as respostas adequadas de acordo com a lógica de negócio da aplicação. O artigo (Lachgar et al., 2018) foi importante para perceber o que já existe em termos de API, de modo a decidir a que utilizar na aplicação desenvolvida.

A estrutura de navegação da aplicação pode ser organizada em duas partes, pública e privada. A parte pública da aplicação android consiste na sequência de ecrãs desde que o utilizador abre a aplicação até autenticar-se/efetuar login, sendo que nesse momento o utilizador entra na sua parte privada.

Basicamente, quando o utilizador abre a aplicação aparece um ecrã de início que serve como um ecrã introdutório para a aplicação desenvolvida, demorando alguns segundos até desaparecer e mostrar o menu de início. Neste menu existem três opções possíveis, uma que mostra ao utilizador o ecrã de definições da aplicação, outra que mostra o ecrã de registo onde o utilizador poderá se registar e outra que o redireciona para o ecrã de login onde se poderá posteriormente autenticar. Toda esta sequência de ecrãs poderá ser visualizada na figura seguinte.

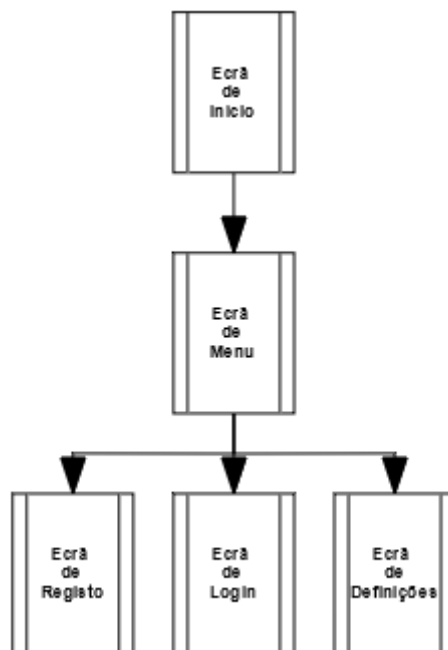


Figura 3.23: Navegação da parte pública da aplicação android do Sistema PSOCAT

Quando o utilizador se autentica, este entra na parte privada da aplicação, apenas destinada a ele próprio. O primeiro ecrã da parte privada diz respeito à página inicial, que possui um menu lateral com diferentes opções para o utilizador. Para além de permitir ao utilizador terminar a sessão voltando para o ecrã de login da parte pública, o menu dispõe de várias opções entre as quais:

- **Perfil:** Permite visualizar o ecrã de perfil do utilizador.
- **Lista de fotos:** Mostra ao utilizador o ecrã com a sua lista de fotos capturadas.
- **Adicionar/Importar foto:** Redireciona o utilizador para o ecrã onde este poderá capturar e/ou importar uma nova foto das suas lesões.
- **Calculadora PASI:** Possibilita o utilizador efetuar um cálculo das suas lesões no ecrã relativo à calculadora PASI.
- **Histórico:** Opção que permite ao utilizador entrar no ecrã do histórico das suas avaliações de lesões efetuadas anteriormente.
- **Eventos:** Onde o utilizador poderá consultar a sua lista de eventos calendarizados e relacionados com a tomada de medicação relativa à aplicação, no ecrã respetivo.

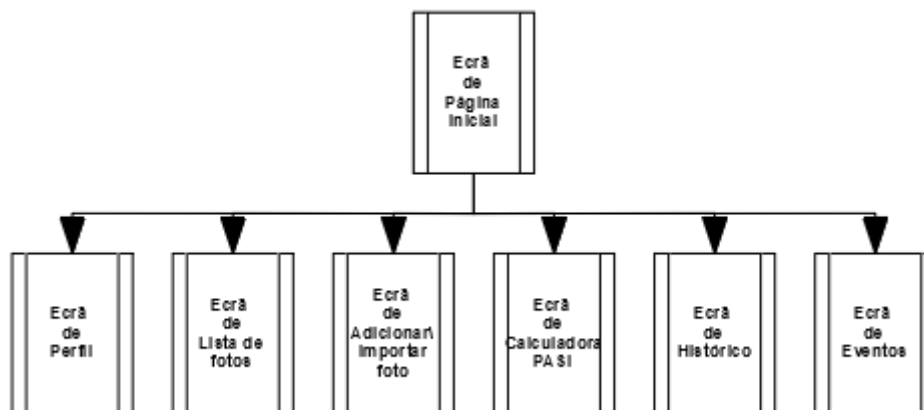


Figura 3.24: Navegação da parte privada da aplicação android do Sistema PSOCAT

A lógica de ecrãs da parte privada pode ser exemplificada/demonstrada na figura anterior. Após o levantamento/especificação dos requisitos/restrições necessários para o funcionamento do sistema desenvolvido, juntamente com a definição/estruturação da sua arquitetura e lógica de ecrãs da aplicação android, deve-se prosseguir para a implementação do sistema PSOCAT de acordo com as regras/normas estabelecidas pelos requisitos especificados e consoante o papel dos vários intervenientes mencionados na arquitetura do sistema, de forma a que cada um cumpra a sua devida função com o intuito de possibilitar o fluxo de dados/informação entre eles.

3.3 Implementação do Sistema PSOCAT

O sistema, de modo a integrar a aplicação móvel com o servidor *backend* e permitir a sua interação, deverá ser implementado de maneira a que permita a comunicação entre ambos, para que o fluxo entre os principais componentes do sistema seja possível. Os componentes do *frontend* encontram-se implementados de forma a receber as interações do utilizador, interpretá-las corretamente e comunicar com os componentes do *backend*, que também se encontram implementados de maneira a receber os pedidos, obter a informação necessária e retornar essa mesma informação. Com o intuito de explicar todo este mecanismo, é retratado de seguida em que consiste a implementação relativa ao *backend* e à aplicação android desenvolvida, respetivamente.

3.3.1 Implementação do Backend

No que diz respeito ao *backend*, como já referido, possui uma base de dados MySQL que possui a informação relacionada com todos os utilizadores registados na aplicação, ou

seja, é onde todos os dados se encontram devidamente armazenados. MySQL diz respeito a um sistema de gerenciamento de banco de dados relacional baseado numa linguagem SQL, que significa Structured Query Language, ou seja, permite a consulta estruturada de dados.

De acordo com o modelo de Entidade-Relação (E-R) referido anteriormente, um dos seus principais componentes diz respeito às entidades, que possuem determinados atributos e podem se relacionar umas com as outras. As entidades na base de dados MySQL podem ser representadas por tabelas que são basicamente conjunto de dados relativos a determinado objeto, possuindo campos/atributos relativos ao mesmo.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(11)			No	None	AUTO_INCREMENT
2	email	text	latin1_swedish_ci		No	None	
3	password	text	latin1_swedish_ci		No	None	
4	type	int(11)			No	None	

Figura 3.25: Tabela do utilizador da base de dados MySQL

Na figura acima encontra-se representada uma das tabelas da base de dados, a tabela do utilizador, que contém a informação principal relativa ao mesmo. Esta tabela possui quatro atributos, um dos quais diz respeito ao identificador do próprio utilizador(id) que é único e cujo valor é auto incrementado na base de dados. Para além do id, a tabela também tem atributos do tipo texto referente ao email e à password encriptada do utilizador, e um atributo do tipo inteiro que identifica o tipo de utilizador, paciente ou médico. O id do utilizador vai ser importante para permitir as relações com as outras tabelas existentes, isto é, para além de servir como chave primária na própria tabela será também uma referência nas outras tabelas, onde aí se denomina chave estrangeira.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	id	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/> 2	userId	int(11)			Yes	NULL	
<input type="checkbox"/> 3	name	text	latin1_swedish_ci		No	None	
<input type="checkbox"/> 4	birthdate	text	latin1_swedish_ci		No	None	
<input type="checkbox"/> 5	skin	text	latin1_swedish_ci		No	None	
<input type="checkbox"/> 6	alergies	text	latin1_swedish_ci		Yes	NULL	
<input type="checkbox"/> 7	meds	text	latin1_swedish_ci		Yes	NULL	

Figura 3.26: Tabela do perfil do utilizador da base de dados MySQL

Na figura anterior é possível visualizar uma das tabelas com as quais a tabela do utilizador se relaciona, a tabela do perfil. Nesta tabela, existe um atributo chamado `userId` que funciona como chave estrangeira na própria tabela e é uma referência à chave primária na tabela do utilizador, correspondendo ao atributo `id`.

Para permitir a vários utilizadores o acesso aos seus dados ao mesmo tempo, torna-se crucial o papel do servidor Apache, que diz respeito a um servidor web simples e gratuito que consiste num sistema concorrente e *multithreaded*, permitindo não só que cada pedido seja tratado paralelamente, como possibilita o tratamento de vários pedidos ao mesmo tempo. O servidor Apache comunica com a base de dados MySQL através de vários serviços em linguagem PHP, sendo o primeiro passo conectar-se à base de dados existente.

```

1 <?php
2 $db_name = "Psocat";
3 $mysql_username = "root";
4 $mysql_password = "";
5 $server_name = "localhost";
6 $connection = mysql_connect($server_name, $mysql_username, $mysql_password);
7 mysql_select_db($db_name, $connection);
8 ?>

```

Código Fonte 3.1: Instrução para efetuar a ligação ao servidor MySQL na base de dados especificada no ficheiro `conn.php`

O código apresentado acima é responsável por efetuar a conexão do servidor à base de dados MySQL utilizada, sendo o método `mysql_connect` responsável por abrir uma nova ligação ao servidor MySQL de acordo com os parâmetros indicados previamente. De

seguida, o método *mysql_select_db* seleciona a base de dados pretendida com o nome da variável do primeiro parâmetro.

Este passo está presente em todos os serviços, sendo necessária a ligação à base de dados antes de efetuar os pedidos à mesma necessários. Este código é integrado em cada serviço através da instrução em baixo, com o auxílio da expressão *require* utilizada.

```
1 <?php
2 require "conn.php";
3 ?>
```

Código Fonte 3.2: Instrução responsável por integrar o código do ficheiro conn.php

O PHP recebe os pedidos HTTP, que basicamente se baseiam em métodos GET/POST. Recebe por método POST informações que são guardadas em variáveis específicas, como por exemplo o email do utilizador. Estas variáveis são essenciais para posteriormente ler/insérer/alterar/eliminar a informação da base de dados MySQL do utilizador respetivo. Podemos visualizar a obtenção destes dados através de uma instrução como a representada a seguir.

```
1 <?php
2 $email = $_POST['email'];
3 ?>
```

Código Fonte 3.3: Instrução responsável por receber a variável de identificador 'email', enviado pelo método POST

```
1 <?php
2 $checkUserExistsQuery = "SELECT * FROM User WHERE email = '$email' ";
3 $checkUserExistsResult = mysql_query($checkUserExistsQuery)
4     or die('error: ' .mysql_error());
5 ?>
```

Código Fonte 3.4: Instrução responsável por obter a informação da tabela do utilizador com o email recebido

No código anterior, é executada uma *query*, ou seja, um pedido à base de dados MySQL que neste caso se trata de obter informação do utilizador que possui o email recebido por POST anteriormente. O método *mysql_query* utiliza o parâmetro relativo à *query* especificada para obter da base de dados a informação requisitada e retorna o resultado. Caso ocorra um erro na ligação ao servidor/base de dados, o serviço retorna essa informação com a instrução *mysql_error*.

As instruções de código PHP apresentadas são importantes na estrutura dos serviços relativos às funcionalidades da aplicação móvel desenvolvida, tal como o registo de um utilizador. Esta funcionalidade, assim como noutras funcionalidades, é iniciada com a ligação ao servidor e base de dados MySQL, sendo depois obtidas as variáveis POST e posteriormente efetuado o pedido para obter a informação do utilizador respetivo. Estes são os passos básicos necessários de uma funcionalidade. Se o utilizador já existir retorna essa informação à aplicação em formato *String*. Se não existir, é inserido na base de dados o novo utilizador através de uma *query*.

```
1 <?php
2 $registQuery = "INSERT INTO User (email, password, type)
3     VALUES ('$email', '$password', '$type)";
4 $registQueryResult = mysql_query($registQuery)
5     or die('error: ' .mysql_error());
6 ?>
```

Código Fonte 3.5: Instrução responsável por inserir a informação do utilizador na tabela respetiva

Nas instruções acima, podemos observar o pedido responsável por inserir na base de dados o novo utilizador, recebendo o email, a *hash* da password que é encriptada na implementação da aplicação móvel e o tipo de utilizador. Após o pedido, a restante informação do utilizador é adicionada na tabela de perfil respetiva através de uma *query* semelhante. No final é retornada informação do sucesso ou insucesso do serviço, terminando o mesmo com o método *mysql_close* que fecha/termina a ligação ao servidor MySQL.

```
1 <?php
2 mysql_close($connection);
3 ?>
```

Código Fonte 3.6: Instrução responsável por terminar a ligação ao servidor MySQL

Na funcionalidade de obtenção da lista de fotos do utilizador, para além dos passos iniciais necessários, o resultado da *query* que devolve a lista de fotos solicitada é armazenado num *array* através do método *mysql_fetch_assoc* e retornado para a aplicação móvel em formato JSON através do método *json_encode*. Por fim, é terminada a ligação ao servidor MySQL. As fotos de cada utilizador encontram-se em formato Base64, isto é, foram codificadas em ficheiros de dados binários na aplicação móvel, tendo sido posteriormente

enviadas para o servidor e por fim armazenadas na base de dados MySQL.

```
1 <?php
2 $rows = array();
3 while ($row = mysql_fetch_assoc($getPhotoListResult)) {
4     $rows[] = $row;
5 }
6 echo json_encode($rows);
7 ?>
```

Código Fonte 3.7: Instrução responsável por guardar o resultado da *query* num *array* e retornar em formato JSON

Para garantir o funcionamento das várias funcionalidades do sistema implementado, para além do *backend* é necessária a implementação correta da aplicação móvel desenvolvida, ou seja, a parte relativa ao *frontend* do sistema. A seguir, é então descrita a implementação relativa à componente android do sistema desenvolvido.

3.3.2 Implementação da Aplicação Móvel

Para o desenvolvimento da aplicação móvel foi utilizado o Android Studio, que diz respeito ao ambiente de desenvolvimento integrado (IDE) oficial para o sistema operativo android. Este IDE fornece as ferramentas mais rápidas e eficazes para a criação de aplicativos para todos os tipos de dispositivos android.

Através do Android Studio, é possível começar o desenvolvimento de uma nova aplicação android, sendo o JAVA a principal linguagem utilizada. O utilizador quando abre a aplicação, visualiza a atividade inicial que diz respeito ao ecrã de entrada para o menu inicial.

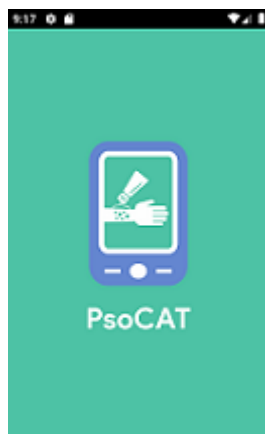


Figura 3.27: Ecrã inicial da aplicação PSOCAT

Os ecrãs da aplicação dizem respeito à interface gráfica com a qual os utilizadores interagem, sendo que na própria aplicação encontram-se implementados em XML. Estes ficheiros XML são designados por *layouts*, e resumidamente possuem vários elementos visuais que podem ser objetos *View* e *ViewGroup*. Os objetos *View* dizem respeito a componentes visuais do *layout* com os quais o utilizador poderá interagir, enquanto que os objetos *ViewGroup* são blocos invisíveis que podem conter um ou mais objetos *View* estruturados de determinada forma.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:background="@color/MainGreen">
6
7     <ImageView
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:layout_centerInParent="true"
11        android:scaleType="centerCrop"
12        android:src="@drawable/ic_splash_screen" />
13
14 </RelativeLayout>
```

Código Fonte 3.8: Ficheiro XML do ecrã inicial

No ficheiro XML acima, relativo ao ecrã inicial da aplicação, podemos visualizar um *RelativeLayout* que se trata de um objeto *ViewGroup* que organiza os seus objetos *Views* em posições relativas. O *RelativeLayout* possui largura e altura relativas ao

dispositivo usado, definidas nos atributos *android:layout_width* e *android:layout_height*. Este objeto *ViewGroup* possui um objeto *View*, do tipo *ImageView*, que será visível ao utilizador mostrando uma imagem e possui as dimensões necessárias tendo em conta o seu conteúdo (*wrap_content*) encontrando-se centrado no ecrã da aplicação pelo atributo *android:layout_centerInParent*.

Os ecrãs XML encontram-se normalmente associados a atividades ou fragmentos, sendo neste caso associado à atividade inicial da aplicação. Podemos visualizar a seguir o exemplo da atividade inicial da aplicação, em que o ecrã XML é definido no primeiro método chamado numa atividade, designado de *onCreate*. Neste método, o *layout* do primeiro ecrã é passado como argumento na função *setContentview*, indicando a interface com a qual o utilizador irá interagir.

```
1 @Override
2     protected void onCreate(Bundle savedInstanceState) {
3         super.onCreate(savedInstanceState);
4         setContentView(R.layout.activity_splash_screen);
```

Código Fonte 3.9: Método *onCreate* na atividade inicial da aplicação

Após 3 segundos, relativos ao tempo de início, a aplicação redireciona o utilizador para o ecrã relativo ao menu inicial. O tempo de início encontra-se definido numa classe de variáveis estáticas chamada *Configs*.

```
1 public class Configs {
2
3     public static final int SPLASH_SCREEN_TIME = 3000;
```

Código Fonte 3.10: Instrução em que o tempo de início da aplicação é especificado numa variável em milisegundos na classe *Configs*

Resumidamente, é iniciada a atividade relativa ao menu inicial da aplicação terminando de seguida a atividade inaugural. Em primeiro lugar é criado um objeto do tipo *Intent* que indica/especifica uma determinada operação a ser realizada pela atividade atual, sendo o primeiro parâmetro relativo à atividade atual e o segundo relativo à atividade a ser iniciada pela operação. O método *startActivity* inicia a nova atividade de acordo com o objeto *Intent* recebido e posteriormente o método *finish* termina a atividade inicial. Podemos visualizar esse processo na instrução seguinte.

```
1 Intent intent = new Intent(this, MenuActivity.class);
2 startActivity(intent);
3 finish();
```

Código Fonte 3.11: Instrução responsável pelo início da atividade do menu inicial e de terminar de seguida a atividade inaugural

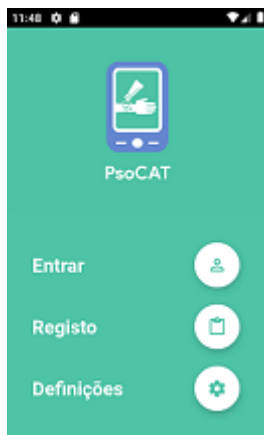


Figura 3.28: Ecrã do menu inicial da aplicação PSOCAT

Na aplicação desenvolvida no ambiente de desenvolvimento Android Studio, existem componentes impriscindíveis para o funcionamento e comportamento correto da aplicação tal como o Gradle que se trata de um sistema de construção utilizado para automatizar a construção, teste, implementação, etc. Todos os projetos android necessitam deste componente para gerar um ficheiro *Android Package* (APK), em que o Gradle pega em todos os ficheiros XML e JAVA e compacta-os num arquivo único com esse nome. O ficheiro `build.gradle` faz parte dos *scripts* Gradle no módulo do projeto onde todas as dependências são definidas e as versões sdk do android são declaradas.

```
1 android {
2     compileSdkVersion 30
3     defaultConfig {
4         applicationId "com.psocat"
5         minSdkVersion 24
6         targetSdkVersion 30
7         versionCode 202001211
8         versionName "1.0"
9     }
10 }
11 dependencies {
12     implementation 'com.android.volley:volley:1.2.0'
13 }
```

Código Fonte 3.12: Configurações específicas na construção android e dependências da aplicação no ficheiro build.gradle da aplicação PSOCAT

Podemos observar no ficheiro build.gradle algumas configurações importantes, como por exemplo:

1. android: Diz respeito a configurações específicas de construção android.

- **compileSdkVersion:** Usado para definir o nível da API do aplicativo, podendo este utilizar os seus recursos e de um nível inferior.
- **applicationId:** Utilizado para identificar o id único da publicação da aplicação.
- **minSdkVersion:** Define o nível mínimo da API necessário para executar a aplicação.
- **targetSdkVersion:** Define o nível da API usado para testar a aplicação.
- **versionCode:** Define o código da versão da aplicação. Sempre que é atualizada, a aplicação incrementa o seu valor em 1 ou mais.
- **versionName:** Define o nome da versão da aplicação. Também pode aumentar com uma nova atualização.

2. dependencies: Especifica as dependências necessárias para construir o projeto, como o caso da livraria Volley declarada pelo termo *implementation*.

Estas configurações são importantes para definir o tipo de dispositivos android para a qual a aplicação desenvolvida irá funcionar, assim como possibilitar melhorias e determinados comportamentos por parte do aplicativo implementado. Também é importante para o funcionamento da aplicação o manifesto android, que se trata de um ficheiro XML que descreve informações necessárias sobre o aplicativo para as ferramentas de compilação do android, para o sistema operacional android e para o Google Play.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     package="com.psocat">
```

Código Fonte 3.13: Código inicial do ficheiro AndroidManifest.xml

No início do ficheiro é definido o nome do pacote e id do aplicativo (`package="com.psocat"`) que é essencial para aceder a todas as classes e recursos da aplicação

desenvolvida. É no manifesto que as permissões necessárias para correr a aplicação se encontram designadas, sendo que se o dispositivo android não permitir alguma delas poderá impedir alguma funcionalidade da aplicação de funcionar. Alguns exemplos de permissões dizem respeito à necessidade do uso da câmara do dispositivo, do uso da internet, etc, como se pode observar no código seguinte.

```
1 <uses-permission android:name="android.permission.INTERNET" />
2 <uses-feature
3     android:name="android.hardware.camera"
4     android:required="true" />
```

Código Fonte 3.14: Código relativo às permissões do ficheiro AndroidManifest.xml

No manifesto existe uma secção relativa à aplicação, especificando inicialmente informação relativa à mesma como o nome (`android:label`), logótipo utilizado (`android:icon`) e outros dados. Dentro dessa secção encontram-se especificadas cada atividade da aplicação desenvolvida, como por exemplo a atividade relativa ao menu, após o código referente à secção da aplicação.

```
1 <application
2     android:label="@string/app_name"
3     android:icon="@mipmap/ic_launcher">
4     <activity
5         android:name=".Public.Menu.MenuActivity">
6     </activity>
7 </application>
```

Código Fonte 3.15: Código relativo à secção da aplicação e de uma atividade do ficheiro AndroidManifest.xml

Todas as atividades da aplicação, quer da parte pública quer privada, herdam uma atividade base com instruções específicas que devem ser tratadas em todas as atividades existentes. Estas instruções encontram-se implementadas no método *onCreate*, dizendo respeito à necessidade de evitar/desabilitar o utilizador de efetuar capturas de ecrã em toda a aplicação, o que está de acordo com os requisitos do sistema, e também à inicialização da base de dados local SQLite, que diz respeito a um banco de dados relacional de código aberto que permite realizar operações como manipular, armazenar e recuperar dados persistentes no banco de dados. A classe AppUtils é responsável por iniciar, caso ainda não exista, a base de dados local.

```
1 public class AppUtils {
2     public static boolean checkIfDatabaseIsCreated() {
3         return MobileConfigData.getSingleton().getDatabaseHelper() != null;
4     }
5     public static void initDatabase(BaseActivity activity) {
6         MobileConfigData.getSingleton().setDatabaseName(Configs.DATABASE_NAME);
7         MobileConfigData.getSingleton().setDatabaseVersion(Configs.DATABASE_VERSION);
8         MobileConfigData.getSingleton().initDatabaseHelper(activity);
9     }
10 }
```

Código Fonte 3.16: Funções relativas à base de dados local na classe AppUtils

Basicamente o método de iniciar/criar esta base de dados, *initDatabase*, irá ser chamado em qualquer atividade da aplicação caso a base de dados ainda não tenha sido criada, o que é verificado pelo método *checkIfDatabaseIsCreated*. Relativamente à base de dados local SQLite criada, esta permite guardar e obter dados sobre configurações da aplicação PSOCAT e do último utilizador autenticado na mesma. As funções *getValueFromKey* e *setValue*, permitem respetivamente obter informação da base de dados local e guardar informação na mesma. Estes métodos funcionam de forma semelhante às *queries* utilizadas na base de dados MySQL, sendo que para obter os dados relativos ao parâmetro *key*, que identifica a informação requerida, a aplicação efetua um pedido à base de dados local e devolve a informação pretendida. Para guardar dados, é recebido o parâmetro *value* que diz respeito ao valor a inserir na base de dados local da *key* respetiva. A informação obtida e guardada na base de dados SQLite encontra-se em formato *String*.

Para exemplificar, a aplicação android possui funções para obter a linguagem corrente assim como para guardar/atualizar a linguagem por interação do utilizador. Os métodos para guardar e obter a linguagem encontram-se apresentados a seguir, usando os métodos base para guardar e obter dados passando os parâmetros respetivos.

```
1 public static String getLanguage() {
2     return getValueFromKey("Language");
3 }
4 public static void setLanguage(String language) {
5     setValue("Language", language);
6 }
```

Código Fonte 3.17: Métodos para obter e guardar a linguagem da base de dados SQLite

Voltando ao menu inicial da aplicação, este herda a atividade base da aplicação assim como as outras atividades da parte pública. O ecrã XML possui basicamente apenas um

componente *Framelayout* que se refere a um único elemento, geralmente um fragmento.

Qualquer atividade pode integrar/adicionar um fragmento no componente de visualização indicado. No caso da atividade do menu inicial, esta adiciona o fragmento do menu no componente de visualização do seu ecrã XML, criando uma nova instância desse fragmento.

```
1 FragmentManager fragmentManager = getSupportFragmentManager();
2     MenuFragment menuFragment = MenuFragment.newInstance();
3     FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
4     fragmentTransaction.add(R.id.content_frame, menuFragment);
5     fragmentTransaction.commit();
```

Código Fonte 3.18: Código JAVA responsável por adicionar um fragmento específico a um *Framelayout* com o id especificado

Um fragmento representa um componente ou uma parte da interface do utilizador numa atividade. Uma atividade pode ter vários fragmentos, podendo ser adicionados, combinados e reutilizados para criar uma *User Interface* (UI) diversificada. A UI de um fragmento é definida no método inicial da sua criação, assim como numa atividade, dizendo respeito também a um ficheiro XML.

A atividade do menu inicial é composta por um fragmento que vai mostrar ao utilizador a sua interface gráfica, correspondente às 3 opções do menu. Cada uma das opções remete para um ecrã diferente, sendo iniciada uma nova atividade dependendo da opção escolhida pelo utilizador. Com o intuito de exemplificar como se procede a ligação da aplicação android com o *backend*, de seguida será demonstrada a funcionalidade relativa ao registo de um utilizador.



Figura 3.29: Ecrã de registo da aplicação PSOCAT

O ecrã XML da nova atividade iniciada possui, para além de um componente *Framelayout* referente ao fragmento respetivo, um componente visual relativo a uma barra de ferramentas superior, que indica a funcionalidade do ecrã atual e contém uma opção para voltar ao ecrã anterior. A mesma estrutura verifica-se no ecrã de autenticação e de definições. Todos os fragmentos da aplicação herdam um fragmento base que possui configurações que estão presentes em todos os fragmentos que estendem dele, como por exemplo a configuração da barra de ferramentas caso exista. A UI relativa ao fragmento possui uma lista de elementos dispostos verticalmente, que dizem respeito aos dados necessários para o registo do utilizador, estando representados por diferentes componentes como caixas de texto editáveis, *spinners* e botões de opções. O utilizador, após preencher os dados necessários, deverá pressionar o botão do ecrã para confirmar o registo. Se algum campo não se encontrar devidamente preenchido, a aplicação irá alertar o utilizador como se pode verificar no exemplo seguinte.



Figura 3.30: Exemplo de erro no ecrã de registo da aplicação PSOCAT, neste caso por faltar indicar a data de nascimento

Caso o dispositivo do utilizador não possua acesso à internet, a aplicação não poderá comunicar com o *backend* e irá mostrar um alerta. Este comportamento é verificado em toda a aplicação.



Figura 3.31: Exemplo de aviso de não estar conectado à internet no ecrã de registo da aplicação PSOCAT

Se todos os dados estiverem corretamente preenchidos, é chamado o método de registo da classe `NetworkHelper` que é responsável por efetuar os vários pedidos HTTP ao servidor a partir de vários métodos que possui. No caso do registo, é chamado um método da classe `NetworkHelper` que recebe os dados preenchidos pelo utilizador e efetua um pedido HTTP de método POST pela livraria Volley ao servidor.

```
1 StringRequest stringRequest =
2     new StringRequest (Request.Method.POST,
3         url,
4         response -> ...,
5         error -> ...
6     ) {
7         @Override
8         protected Map<String, String> getParams() throws AuthFailureError {
9             Map<String, String> params = new HashMap<>();
10            ...
11            return params;
12        }
13    }
```

Código Fonte 3.19: Exemplo de pedido Volley ao servidor, relativo ao serviço de registo

Nos pedidos HTTP, existe uma variável `url` que indica o endereço correspondente ao ficheiro PHP do *backend* responsável pelo serviço respetivo. Os dados do utilizador são indicados num método do próprio pedido designado `getParams` que possui uma variável `params` que mapeia os campos relativos aos parâmetros recebidos. O pedido fica à escuta da resposta do *backend*, recebendo a mesma em formato `String`.

Se algum erro ocorrer ou o utilizador já se encontrar registado, a aplicação mostra uma mensagem de erro. Se o registo for efetuado com sucesso, a aplicação informa o utilizador com uma mensagem de sucesso.

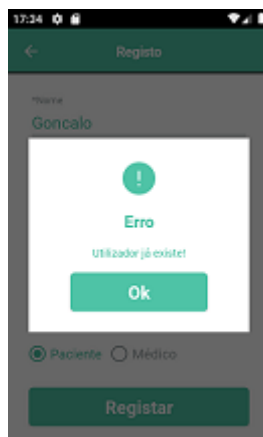


Figura 3.32: Exemplo de aviso de erro ao tentar efetuar o registo no ecrã de registo da aplicação PSOCAT



Figura 3.33: Exemplo de aviso de ter efetuado o registo com sucesso no ecrã de registo da aplicação PSOCAT

A password do utilizador é encriptada antes de ser enviada para o *backend* no serviço de registo. A classe `CryptationUtils` possui os métodos necessários quer para codificar, quer para descodificar a password respetiva.

O método de encriptação, *encryption*, recebe os dados relativos ao email do novo utilizador e à password preenchida pelo mesmo, criando uma chave ou *hash* resultante

da informação do utilizador. É utilizada a técnica *Advanced Encryption Standard* (AES) que é dos padrões de criptografia avançada mais comuns e utilizados atualmente, sendo que utiliza a password fornecida para criar uma cifra simétrica relativa a uma instância AES em modo de encriptação para gerar uma chave que é posteriormente codificada em Base64 através do email, o que oferece segurança e robustez aos dados do utilizador.

```
1 SecretKeySpec key = generateKey(password);
2 Cipher c = Cipher.getInstance("AES");
3 c.init(Cipher.ENCRYPT_MODE, key);
```

Código Fonte 3.20: Código utilizado para criar a cifra da password no modo de encriptação

```
1 byte[] encVal = c.doFinal(data.getBytes());
2 return Base64.encodeToString(encVal, Base64.DEFAULT);
```

Código Fonte 3.21: Código utilizado para codificar a cifra com o email do utilizador(data) e retornar a *hash* em formato Base64

O método de decriptação, *decryption*, recebe a chave ou *hash* do utilizador respetivo e a password inserida pelo utilizador que se tenta autenticar. A password é usada para criar uma cifra simétrica relativa a uma instância AES em modo de decriptação e a cifra é descodificada com o valor descodificado da *hash* em formato Base64. Se o valor retornado coincidir com o email do utilizador inserido, significa que a autenticação foi bem sucedida.

```
1 SecretKeySpec key = generateKey(password);
2 Cipher c = Cipher.getInstance("AES");
3 c.init(Cipher.DECRYPT_MODE, key);
```

Código Fonte 3.22: Código utilizado para criar a cifra da password no modo de decriptação

```
1 byte[] decodedValue = Base64.decode(data, Base64.DEFAULT);
2 byte[] decValue = c.doFinal(decodedValue);
3 return new String(decValue);
```

Código Fonte 3.23: Código utilizado para descodificar a cifra com o valor descodificado da *hash*(data) em formato Base64 e retornar o resultado em formato *String*

Assim que o utilizador conseguir efetuar o seu registo com sucesso, este agora poderá se autenticar na aplicação. Para isso, primeiro deverá voltar ao menu inicial e selecionar a

opção de login/autenticação.

No pedido de autenticação são enviados o email e a password inseridos pelo utilizador, com o intuito de receber a informação do utilizador da base de dados que possui esse mesmo email. Caso exista, retorna essa informação que é validada pela aplicação, efetuando a decriptação da chave recebida pela password que o utilizador inseriu para se autenticar. Caso o resultado bata certo com o email do utilizador, o acesso aos dados do utilizador é concedido efetuando novo pedido HTTP para obter o nome do utilizador e o seu tipo. Se tudo correr bem, a aplicação entra na parte privada do utilizador autenticado, iniciando a atividade referente à página inicial que é apresentada na figura seguinte.



Figura 3.34: Ecrã da página inicial da aplicação PSOCAT

A parte privada da aplicação é constituída por uma única atividade. Esta atividade herda/estende a atividade de sessão que é uma atividade específica para a parte privada da aplicação.

Por sua vez, a atividade da sessão herda a atividade base da aplicação, possuindo as mesmas características que as outras atividades mas com algumas configurações adicionais. As configurações adicionais dizem respeito ao tratamento da inatividade da aplicação assim como limpar as informações da cache retidas no Volley.

A inatividade da aplicação é verificada por uma classe do tipo Timer que executa uma operação se o utilizador não interagir com a aplicação durante determinado período de tempo, especificado pela variável `Configs.TIMEOUT_TIME`. Essa operação irá limpar toda a *cache*, terminar a sessão do utilizador e voltar para a parte pública da aplicação, terminando a atividade corrente e chamando a atividade do menu inicial. Já no menu inicial, a aplicação mostra uma mensagem ao utilizador a informar que a sua sessão

expirou, tendo recebido uma instrução extra no *intent*.

```
1 timer = new Timer();
2 timer.schedule(new TimerTask() {
3     @Override
4     public void run() {
5         ...
6     }
7 }, Configs.TIMEOUT_TIME);
```

Código Fonte 3.24: Código responsável por iniciar o contador de inatividade que irá chamar o método *run* após o período de tempo especificado pela variável *Configs.TIMEOUT_TIME*

```
1 CacheUtils.clearAllCache(this);
2 finish();
3 Intent intent = new Intent(this, MenuActivity.class);
4 intent.putExtra(MenuActivity.SESSION_EXPIRED, true);
5 ...
6 startActivity(intent);
```

Código Fonte 3.25: Código relativo às instruções para limpar toda a *cache*, terminar a sessão atual e voltar ao menu inicial da aplicação PSOCAT

Sempre que o utilizador interage com a aplicação, quando esta se encontra na parte privada, a atividade da sessão despoleta um método que força o recomeço do tempo de inatividade, ou seja, sempre que o utilizador permanecer ativo a aplicação não irá voltar para a parte pública por inatividade. Para além de possuir no seu *layout* uma barra de ferramentas e um fragmento associado, a atividade privada da aplicação possui também um componente de navegação lateral que permite ao utilizador visualizar as opções existentes no menu lateral da parte privada da aplicação.

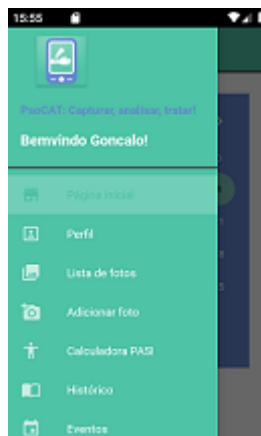


Figura 3.35: Menu lateral de navegação na atividade privada da aplicação PSOCAT

Sempre que o utilizador selecionar uma das opções do menu lateral de navegação, à exceção da última que permite ao utilizador terminar a sessão e voltar para a parte pública, a aplicação verifica qual foi a opção escolhida e chama um método que permite a troca do fragmento atual que se encontra visível no componente de *Framelayout* da atividade privada. O novo fragmento visível diz respeito à funcionalidade selecionada e possui uma UI associada.

Por exemplo, se o utilizador escolher a opção da Lista de fotos, a aplicação troca o fragmento da atividade privada, para mostrar a UI da lista de fotos. O ecrã XML deste fragmento é composto por filtros relativos às fotos das lesões do utilizador que irão ser visualizadas. O utilizador poderá escolher a opção do *spinner* relativa às zonas do corpo das fotos visualizadas e escolher o intervalo de datas em que as fotos foram capturadas/armazenadas na aplicação. Por defeito, ao entrar neste ecrã, a aplicação mostra automaticamente as fotos das lesões em qualquer zona do corpo entre determinado intervalo de tempo já definido por padrão.

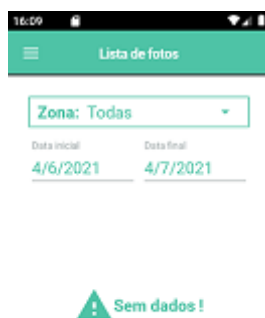


Figura 3.36: Ecrã da lista de fotos do utilizador da aplicação PSOCAT, quando não existem fotos a mostrar

Na figura anterior, podemos visualizar o que acontece quando não existem fotos a visualizar por parte da aplicação, mostrando uma mensagem/alerta. Caso exista informação retornada, a aplicação mostra a lista de fotos ao utilizador, sendo representada por um componente de visualização relativo a uma lista vertical.

Como já referido anteriormente, as fotos das lesões do utilizador são guardadas na base de dados em formato Base64. Na aplicação móvel, a classe *ImageUtils* possui os métodos não só para converter as imagens em formato Base64 como efetuar o inverso

também.

Quando o utilizador captura e/ou importa uma foto das suas lesões para a aplicação, esta imagem, que se encontra em formato *bitmap*, é convertida para uma *String* Base64 antes de ser enviada para o servidor. Quando o utilizador pede ao servidor a sua lista de fotos, estas são devolvidas no mesmo formato Base64. A classe *ImageUtils* converte esses ficheiros binários para formato *bitmap*, para que as fotos possam ser visualizadas nos componentes visuais da UI do fragmento respetivo.

Na funcionalidade de obter a lista de fotos, assim como noutras funcionalidades da parte privada, os pedidos HTTP efetuados pelo Volley possuem uma componente de *cache* que verifica se o pedido já foi efetuado anteriormente. Quando um pedido, já feito anteriormente, é novamente efetuado existe uma classe chamada *CacheUtils* que verifica se a resposta a esse serviço já se encontra guardada em *cache* anteriormente. Caso exista, é utilizada essa resposta em vez de efetuar o pedido, senão efetua o pedido e guarda o mesmo em *cache* posteriormente.

```
1 String responseInCache = CacheUtils.getResponseInCache(request, url);
```

Código Fonte 3.26: Exemplo de obtenção de uma resposta/resultado em *cache* de um pedido efetuado anteriormente

```
1 stringRequest.setShouldCache(shouldCache);  
2 request.add(stringRequest);
```

Código Fonte 3.27: Exemplo de guardar uma resposta/resultado em *cache* de um pedido efetuado

O método *getResponseInCache* da classe *CacheUtils* recebe o endereço do serviço respetivo na variável *url* para verificar se possui dados em *cache* relativos a esse serviço e retorna a resposta em formato *String*. Se não possuir dados guardados anteriormente, efetua o pedido respetivo e com o método *setShouldCache* indica se a resposta a esse pedido deve ser ou não guardada em *cache*. A classe *CacheUtils* também possui métodos para limpar a *cache* de um serviço, quando a informação deste é atualizada, e outro para limpar a *cache* de todos os serviços, para quando o utilizador sair da sua parte privada.

Caso o serviço retorne resultados relativos às fotos do utilizador, a aplicação organiza a informação JSON recebida no formato *String* em vários objetos diferentes, cada um relativo a uma única foto. A lista de elementos recebida é enviada para a componente visual da UI responsável por dispor a lista de fotos verticalmente, permitindo assim a sua

visualização por parte do utilizador.

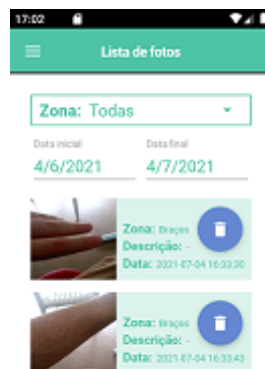


Figura 3.37: Ecrã da lista de fotos do utilizador da aplicação PSOCAT

Em suma, os principais mecanismos e aspetos de implementação referentes ao *backend* e *frontend* da aplicação são importantes para o funcionamento e comportamento correto de todo o sistema da aplicação móvel desenvolvida, sendo essencial de acordo com o processo de engenharia de software da criação do sistema que todas as especificações iniciais sejam cumpridas de forma a comprovar o funcionamento do mesmo, em termos da sua arquitetura e implementação.

Após cumprir as atividades iniciais respeitantes à especificação de requisitos e restrições da aplicação e à sua arquitetura e implementação, para finalizar o processo de engenharia de software da criação do sistema é necessária a validação/avaliação do software desenvolvido, de maneira a verificar que as normas e restrições estão a ser tratadas devidamente e analisar e avaliar a usabilidade da aplicação com o intuito da mesma poder evoluir futuramente.

Capítulo 4

Avaliação da Usabilidade da Aplicação PSOCAT

Este capítulo tem como objetivo estudar a aplicação desenvolvida, pretendendo recolher informação sobre a sua utilização por parte dos utilizadores e avaliar a usabilidade da mesma em situações reais de utilização. É crucial aferir como os potenciais utilizadores se sentem com a aplicação, de modo a entender os aspetos que podem ser melhorados ou corrigidos. O estudo baseia-se no modelo de usabilidade estendido PACMAD para avaliação de métricas de usabilidade usando uma abordagem GQM para avaliar a facilidade com que os utilizadores conseguem interagir com a app e identificar potenciais problemas a corrigir ou melhorar posteriormente.

4.1 Metodologia de Avaliação

Para a aplicação se revelar útil para os seus utilizadores, assim como aumentar o seu número, é essencial que esta siga padrões/normas de usabilidade apropriadas. E neste âmbito, são essenciais elementos como o contexto móvel, como já referido, a conectividade existente nos dispositivos móveis, assim como os vários tamanhos de tela e resoluções existentes nestes, as suas capacidades de processamento e métodos de entrada de dados nos aparelhos. Todos estes elementos são imprescindíveis para testar e avaliar a usabilidade da aplicação móvel desenvolvida, avaliação esta que irá contar com o auxílio de determinados atributos, específicos do modelo de usabilidade escolhido.

O modelo de usabilidade estendido PACMAD, utilizado na avaliação da aplicação desenvolvida, acrescenta orientações para o uso de métricas de maneira a avaliar cada atributo correspondente, assim como a utilização de um questionário de satisfação do usuário para uma avaliação completa da usabilidade. Segundo o artigo (Lewis, 1995), o questionário de usabilidade utilizado tem como objetivo medir a satisfação do utilizador com a aplicação

testada, sendo denominado *Computer System Usability Questionnaire* (CSUQ). É constituído por 20 perguntas relacionadas com a satisfação dos participantes com os vários aspetos da aplicação móvel, com escalas de 5 pontos.

1. No geral, estou satisfeito com quão fácil é utilizar esta aplicação móvel.

1	2	3	4	5
Discordo Totalmente	Discordo	Não concordo nem discordo	Concordo	Concordo Totalmente

Figura 4.1: Exemplo de uma questão do CSUQ

Podemos visualizar acima um exemplo de uma questão do respetivo questionário, com uma resposta numa escala que vai desde "1- Discordo Totalmente" até "5- Concordo Totalmente", em termos de satisfação do participante. O CSUQ é essencial para a medição das métricas relativas à satisfação do utilizador, sendo preenchido após a testagem efetuada pelo mesmo para avaliar estas métricas do modelo de usabilidade utilizado.

O modelo de usabilidade PACMAD extendido possui como atributos:

- **Eficácia:** Diz respeito a se um objetivo é ou não atingido, sendo medido tendo em conta a completude ou não de uma determinada tarefa;
- **Eficiência:** Indica o quão bem um determinado objetivo é atingido e é medido tendo em conta o tempo de execução da tarefa;
- **Satisfação:** Indica o sentimento e/ou nível de satisfação do utilizador, que é o elemento mais qualitativo e difícil de avaliar;
- **Erros:** Indica o quanto o utilizador está propenso a cometer erros durante o uso da aplicação, e como estes erros são tratados. Podemos verificar este elemento tendo em conta a quantidade de erros que ocorrem em cada tarefa efetuada;
- **Aprendizagem:** Indica a facilidade de aprendizagem, isto é, em atingir um determinado nível de proficiência. Medida tendo em conta a eficácia e a eficiência;
- **Memorização:** Medida através dos resultados distintos obtidos ao longo do tempo, de maneira a avaliar a melhoria de desempenho entre eles;
- **Carga cognitiva:** Indica a influência do ambiente na utilização da aplicação, podendo ser medida através da testagem em diferentes situações.

Atributo de usabilidade	Métricas de usabilidade
Eficácia	<ul style="list-style-type: none"> - Conclusão da tarefa - Número de passos efetuados corretamente - Conclusão da tarefa dentro dos tempos estabelecidos - Número de erros ocorridos
Eficiência	<ul style="list-style-type: none"> - Tempo para completar a tarefa - Duração de cada passo
Satisfação	<ul style="list-style-type: none"> - Satisfação do utilizador e aceitação por parte do mesmo no que diz respeito à aplicação em si
Aprendizagem	<ul style="list-style-type: none"> - Tempo para completar a tarefa a primeira vez - Número de tentativas até completar a tarefa - Tempo até completar a tarefa em tempo satisfatório
Memorização	<ul style="list-style-type: none"> - Número de cliques até terminar a tarefa - Tempo para finalizar a tarefa - Número de passos para terminar a tarefa
Erros	<ul style="list-style-type: none"> - Número de erros ocorridos
Carga cognitiva	<ul style="list-style-type: none"> - Tempo para a conclusão da tarefa - Tempo de resposta e reação do utilizador face à tarefa a concluir - Número de erros ocorridos - Complexidade da tarefa - Número de cliques necessários - Pedidos de ajuda necessários - Duração da ajuda prestada

Tabela 4.1: Medição de atributos de usabilidade em aplicações móveis

Cada atributo possui determinadas métricas de usabilidade que resumidamente servem

para avaliar cada um deles. Basicamente em cada tarefa, em cada cenário, estas métricas são utilizadas para avaliar a aplicação relativamente ao seu desempenho e comportamento no que diz respeito a cada um dos atributos de usabilidade do modelo utilizado. Na figura anterior, é possível visualizar as métricas de usabilidade correspondentes a cada um dos atributos de usabilidade para avaliar a aplicação. Em conjunto com o modelo GQM, é possível obter a tabela seguinte:

Objetivo	Questões	Q code	Métricas	M code
Eficácia	- A tarefa foi concluída com sucesso? - Todos os passos foram feitos corretamente? - Quantos cliques foram necessários até finalizar a tarefa? - Ocorreram muitos erros durante a realização da tarefa? - A tarefa foi concluída em tempo aceitável?	Q1 Q2 Q3 Q4 Q5	- Completude da tarefa - Número de passos para concluir a tarefa - Número de cliques necessários para completar a tarefa - Número de erros ocorridos durante a tarefa - Completude da tarefa dentro do tempo limite	EFCAM1 EFCAM2 EFCAM3 EFCAM4 EFCAM5
Eficiência	- Em quanto tempo foi concluída a tarefa? - Quanto tempo demorou cada passo?	Q6 Q7	- Tempo de conclusão da tarefa - Duração de cada passo efetuado	EFCEM1 EFCEM2
Satisfação	- Como se sentiu o utilizador ao efetuar a tarefa? - O utilizador ficou satisfeito ao completar a tarefa?	Q8 Q9	- Satisfação do utilizador e aceitação face à tarefa efetuada	SM1 - 2M9
Aprendizagem	- A aplicação tornou-se mais fácil de usar? - Quantas tentativas precisou o utilizador para completar a tarefa? - O utilizador demonstrou maior facilidade em interagir com a aplicação com o tempo?	Q10 Q11 Q12	- Tempo necessário para terminar a tarefa pela primeira vez - Número de tentativas até completar a tarefa - Tempo necessário para completar a tarefa dentro dos tempos limites	AM1 AM2 AM3
Memorização	- O utilizador revelou facilidade em memorizar detalhes importantes em cada tarefa após o primeiro uso? - O utilizador conseguiu completar a tarefa no primeiro uso? - A aplicação revelou-se simples de compreender?	Q13 Q14 Q15	- Número de cliques necessários - Tempo para a conclusão de cenário - Número de passos até terminar a tarefa	MM1 MM2 MM3
Erros	- Em cada tarefa, a aplicação ajudou a ultrapassar os erros ocorridos? - O utilizador conseguiu responder mais rapidamente face aos erros que apareciam? - Quantos erros os utilizadores cometeram?	Q16 Q17 Q18	- Número de erros cometidos durante a tarefa	ERM1
Carga cognitiva	- O utilizador conseguiu completar a tarefa dentro do tempo limite? - A tarefa revelou-se demasiado complexa para o utilizador? - Durante a realização da tarefa ocorreram erros que prejudicaram o desempenho do utilizador? - O utilizador necessitou de auxílio para a completude da tarefa? - O desempenho do utilizador foi aquele que era esperado inicialmente?	Q19 Q20 Q21 Q22 Q23	- Tempo para a conclusão da tarefa - Número de cliques - Complexidade da tarefa - Número de erros ocorridos - Pedidos de ajuda - Duração da(s) ajuda(s) prestada(s) - Tempo de resposta e reação do utilizador face à tarefa a concluir	CCM1 CCM2 CCM3 CCM4 CCM5 CCM6 CCM7

Tabela 4.2: GQM para o Modelo de Usabilidade PACMAD estendido

Resumidamente, cada atributo é visto como um objetivo a alcançar por parte da aplicação, sendo alcançado consoante as questões relativas a esse mesmo objetivo forem corretamente respondidas, de acordo com as métricas relacionadas. Deste modo, pretende-se avaliar a aplicação no que diz respeito à qualidade e facilidade de interação dos diferentes participantes e se esta se demonstra eficiente e eficaz no papel que desempenha. Pretende-se utilizar a versão atual da aplicação PSOCAT que oferece já todas as funcionalidades base identificadas na análise de requisitos.

Relativamente aos participantes, para além da necessidade de possuírem um dispositivo android com as capacidades necessárias, é importante que sejam selecionados tendo em conta a facilidade que demonstram na sua utilização. Devem também possuir conhecimento acerca do que é a psoríase e/ou se possível possuírem o diagnóstico da doença e/ou outras relacionadas. Cada participante será esclarecido sobre os objetivos do estudo de usabilidade, assim como o facto dos dados utilizados serem fictícios. Os dados relacionados com a usabilidade do estudo não serão associados à identificação das pessoas, pelo que será mantido o anonimato de cada participante.

Para a avaliação da aplicação, os participantes inicialmente recebem um formulário de avaliação da usabilidade, com um pequeno contexto acerca do assunto, uma secção onde devem introduzir algumas informações pessoais, as tarefas a efetuar e as métricas que serão avaliadas. As funcionalidades que irão ser avaliadas podem resumir-se na seguinte lista de tarefas:

- T1** : Efetuar a mudança de idioma na aplicação.
- T2** : Efetuar um registo com sucesso na aplicação.
- T3** : Efetuar um login com sucesso na aplicação.
- T4** : Alterar os dados de perfil inseridos no registo.
- T5** : Efetuar o logout e voltar a autenticar-se com outro user.
- T6** : Registrar na aplicação uma foto da mão/braço ou outra zona do corpo.
- T7** : Visualizar a lista de fotos já capturadas.
- T8** : Eliminar uma foto existente.
- T9** : Efetuar o cálculo do índice PASI.
- T10** : Visualizar um registo de avaliações PASI.
- T11** : Obter mais detalhes sobre cada avaliação visualizada anteriormente.
- T12** : Visualizar a lista de eventos da aplicação durante determinado período/intervalo de datas.
- T13** : Visualizar mais informação sobre cada evento selecionado anteriormente.

T1: Efetuar a mudança de idioma na aplicação.

Atributo	Q code	M code	Descrição	Valor		
Eficiência	Q1 Q2 Q3 Q4 Q5	EFCAM1	Completeness da tarefa (Sim/Não)			
		EFCAM2	Número de passos para concluir a tarefa (n°)			
		EFCAM3	Número de cliques necessários para completar a tarefa (n°)			
		EFCAM4	Número de erros ocorridos durante a tarefa (n°)			
		EFCAM5	Completeness da tarefa dentro do tempo limite (Sim/Não)			
Atributo	Q code	M code	Descrição	Valor		
Eficiência	Q6 Q7	EFCEM1	Tempo para a conclusão da tarefa (minutos/segundos)			
		EFCEM2	Duração de cada passo (minutos/segundos)			
Atributo	Q code	M code	Descrição	Valor		
Satisfação	Q8 Q9	SMI-SMI 9	Satisfação do utilizador e aceitação face à tarefa efetuada (Questionário de usabilidade)			
		Atributo	Q code	M code	Descrição	Valor
Aprendizagem	Q10 Q11 Q12	AM1	Tempo necessário para terminar a tarefa a primeira vez (minutos/segundos)			
		AM2	Número de tentativas até completar a tarefa (n°)			
		AM3	Tempo necessário para completar a tarefa dentro dos tempos limites (minutos/segundos)			
Atributo	Q code	M code	Descrição	Valor		
Memorização	Q13 Q14 Q15	MM1	Número de cliques necessários (n°)			
		MM2	Tempo para a conclusão de cenário (minutos/segundos)			
		MM3	Número de passos até terminar a tarefa (n°)			
Atributo	Q code	M code	Descrição	Valor		
Erros	Q16 Q17 Q18	ERM1	Número de erros cometidos durante a tarefa (n°)			
		Atributo	Q code	M code	Descrição	Valor
		Carga cognitiva	Q19 Q20 Q21 Q22 Q23	CCM1	Tempo para a conclusão da tarefa (minutos/segundos)	
CCM2	Número de cliques (n°)					
CCM3	Complexidade da tarefa (Pouco/Normal/Muito)					
CCM4	Número de erros ocorridos (n°)					
CCM5	Pedidos de ajuda (n°)					
CCM6	Duração da ajuda(s) prestada(s) (minutos/segundos)					
CCM7	Tempo de resposta e reacção do utilizador face à tarefa a concluir (minutos/segundos)					

Figura 4.2: Exemplo da avaliação de uma tarefa do formulário de avaliação da usabilidade

Para avaliar cada tarefa, é utilizada uma abordagem baseada no modelo de usabilidade

PACMAD estendido, como é possível visualizar na figura acima. Em cada tarefa, os atributos de usabilidade serão analisados e avaliados consoante as métricas de avaliação que vão sendo registadas durante a testagem de cada funcionalidade, à exceção da satisfação que será avaliada pelo CSUQ. No final do formulário existe uma secção de dúvidas e questões dos participantes acerca da aplicação testada, assim como possíveis sugestões.

Com o feedback recebido pretende-se corrigir eventuais dificuldades ou problemas que venham a ser detetados pelos utilizadores, como a identificação de erros de serviços, problemas de interface ou outras melhorias técnicas relacionadas com o design e implementação da aplicação móvel. Por esta razão, será muito importante conseguir reter da parte dos participantes as suas dificuldades e problemas sentidos durante os testes, de modo a saber os aspetos a melhorar ou corrigir posteriormente. É necessário validar os resultados obtidos de modo a perceber as limitações da aplicação e os aspetos por onde pode melhorar para futuramente se encontrar finalizada da melhor maneira e proporcionar a melhor experiência possível para os seus usuários.

4.2 Resultados da Avaliação

Os resultados obtidos da avaliação da usabilidade da aplicação revelaram-se bastante satisfatórios, não só no que diz respeito à realização das tarefas do formulário, como também os participantes demonstraram facilidade em interagir com a aplicação desenvolvida. Do mesmo modo, no questionário de satisfação, o feedback por parte dos participantes foi bastante positivo, sendo que a aplicação se revelou simples, eficiente e fácil de utilizar. Para além disso, os resultados permitiram identificar os principais problemas existentes na aplicação testada, as suas limitações e aspetos por onde é possível melhorar, sendo que as respostas dos participantes na secção de questões/dúvidas e sugestões contribuíram para perceber possíveis melhorias a efetuar nas funcionalidades testadas, assim como correções e/ou desenvolvimentos posteriores de utilidade para o sistema da aplicação. Os participantes/voluntários selecionados, para além de possuírem os critérios/requisitos necessários, tinham maioritariamente sexo masculino, idades variadas e possuíam doenças dermatológicas como a psoríase e/ou outras.

Na tabela seguinte é possível visualizar um sumário dos resultados da avaliação da usabilidade da aplicação testada, com os principais problemas detetados e possíveis soluções a aplicar. Posteriormente, também são apresentadas as tabelas relativas a cada uma das tarefas efetuadas por cada participante, em que cada uma foi avaliada de forma a verificar não só como se comporta a aplicação em cada uma das funcionalidades testadas, mas também se a tarefa é completada com sucesso.

Número de participantes	5
Públicos-alvo	<ul style="list-style-type: none"> - Pessoas com psoríase ou doenças relacionadas - Pessoas que possuam um dispositivo android e que o saibam utilizar
Principais problemas	<ul style="list-style-type: none"> - Lógica de mostrar/esconder o teclado virtual - Ecrã de início demora demasiado tempo - Timeout da aplicação demasiado curto - Design da aplicação necessita melhorias no seu geral - Zona de registo precisa ser melhorada
Soluções a aplicar	<ul style="list-style-type: none"> - Tornar mais simples a utilização do teclado virtual - Tempo em iniciar a aplicação mais curto - Aumentar o tempo relativo ao timeout da aplicação - Correções na funcionalidade de registo - Melhorias genéricas na aplicação

Tabela 4.3: Resultados da Avaliação de Usabilidade

Percentagem de sucesso	87%
Média de tentativas	1,4
Problemas identificados	- O início da aplicação demora demasiado tempo
Sugestões	- Diminuir o tempo de início da aplicação

Tabela 4.4: Resultados da tarefa 1: Efetuar a mudança de idioma na aplicação.

Percentagem de sucesso	67%
Média de tentativas	1,8
Problemas identificados	<ul style="list-style-type: none"> - O teclado virtual dificulta o registo do utilizador - Dificuldade em escolher a data de nascimento - Spinners não possuem um design simples de interagir - Falta de informação sobre a funcionalidade
Sugestões	<ul style="list-style-type: none"> - Melhorar a lógica do teclado virtual - Melhorar o design dos spinners - Adicionar ajudas/dicas - Voltar ao ecrã anterior após registo

Tabela 4.5: Resultados da tarefa 2: Efetuar um registo com sucesso na aplicação.

Percentagem de sucesso	90%
Média de tentativas	1,2
Problemas identificados	- O teclado virtual dificulta o login do utilizador
Sugestões	- Melhorar a lógica do teclado virtual

Tabela 4.6: Resultados da tarefa 3: Efetuar um login com sucesso na aplicação.

Percentagem de sucesso	90%
Média de tentativas	1,2
Problemas identificados	- O teclado virtual dificulta a alteração - Spinners não possuem um design simples de interagir
Sugestões	- Melhorar a lógica do teclado virtual - Melhorar o design dos spinners

Tabela 4.7: Resultados da tarefa 4: Alterar os dados de perfil inseridos no registo.

Percentagem de sucesso	87%
Média de tentativas	1,4
Problemas identificados	- O teclado virtual dificulta a tarefa
Sugestões	- Melhorar a lógica do teclado virtual

Tabela 4.8: Resultados da tarefa 5: Efetuar o logout e voltar a autenticar-se com outro user.

Percentagem de sucesso	90%
Média de tentativas	1,2
Problemas identificados	- O teclado virtual dificulta a tarefa
Sugestões	- Melhorar a lógica do teclado virtual

Tabela 4.9: Resultados da tarefa 6: Registrar na aplicação uma foto da mão/braço ou outra zona do corpo.

Percentagem de sucesso	90%
Média de tentativas	1,2
Problemas identificados	- Nada a apontar
Sugestões	- Nada a apontar

Tabela 4.10: Resultados da tarefa 7: Visualizar a lista de fotos já capturadas.

Percentagem de sucesso	100%
Média de tentativas	1
Problemas identificados	- Nada a apontar
Sugestões	- Nada a apontar

Tabela 4.11: Resultados da tarefa 8: Eliminar uma foto existente.

Percentagem de sucesso	90%
Média de tentativas	1,2
Problemas identificados	- Spinners não possuem um design simples de interagir
Sugestões	- Melhorar o design dos spinners

Tabela 4.12: Resultados da tarefa 9: Efetuar o cálculo do índice PASI.

Percentagem de sucesso	90%
Média de tentativas	1,2
Problemas identificados	- Nada a apontar
Sugestões	- Nada a apontar

Tabela 4.13: Resultados da tarefa 10: Visualizar um registo de avaliações PASI.

Percentagem de sucesso	90%
Média de tentativas	1,2
Problemas identificados	- Nada a apontar
Sugestões	- Nada a apontar

Tabela 4.14: Resultados da tarefa 11: Obter mais detalhes sobre cada avaliação visualizada anteriormente.

Percentagem de sucesso	100%
Média de tentativas	1
Problemas identificados	- Nada a apontar
Sugestões	- Nada a apontar

Tabela 4.15: Resultados da tarefa 12: Visualizar a lista de eventos da aplicação durante determinado período/intervalo de datas

Percentagem de sucesso	100%
Média de tentativas	1
Problemas identificados	- Nada a apontar
Sugestões	- Nada a apontar

Tabela 4.16: Resultados da tarefa 13: Visualizar mais informação sobre cada evento selecionado anteriormente

Os resultados obtidos por parte de cada participante foram bastante satisfatórios, assim como a aprendizagem e compreensão da aplicação que cada um demonstrou durante os testes. Para além disso, foi possível verificar a existência de problemas que a aplicação possui e também possíveis melhorias e alterações a efetuar posteriormente, com o objetivo de os resolver e fomentar a usabilidade da aplicação desenvolvida.

4.3 Alterações à Aplicação PSOCAT

As alterações efetuadas na aplicação incluíram correções e melhorias nas diferentes funcionalidades da aplicação de acordo com o feedback obtido por parte dos participantes, e outras observações efetuadas no comportamento do sistema. O objetivo trata-se de melhorar a aplicação em vários aspetos, nomeadamente em termos de lógica, quer da aplicação android quer das funcionalidades implementadas, e tornar o design na sua generalidade mais apelativo e simples, de maneira a facilitar a interação entre o utilizador e a interface do sistema e fomentar a eficiência da aplicação, de forma a aumentar a satisfação dos seus utilizadores. É apresentada a seguir uma sumarização das alterações efetuadas posteriormente à interpretação dos resultados da avaliação da usabilidade da aplicação.

Alterações após avaliação da usabilidade da aplicação
- Melhorias na lógica de mostrar/esconder o teclado virtual para facilitar a interação dos utilizadores com a interface da aplicação
- Diminuição do tempo relativo ao ecrã de início da aplicação
- Aumento do tempo de inatividade do utilizador na aplicação
- Correções na funcionalidade de registo, adicionando ajudas/dicas na interface, melhorando o design e permitindo após registo voltar para o ecrã do menu inicial
- Modificar o design dos spinners na aplicação, de modo a tornar mais simples para os utilizadores interagir com a mesma
- Melhorias no design e na lógica da aplicação genéricas

Tabela 4.17: Sumarização das alterações efetuadas após a avaliação da usabilidade da aplicação

Capítulo 5

Conclusão

A tele dermatologia é uma área em constante evolução, em particular na utilização de tecnologias móveis para auxiliar a monitorização e gestão de vários aspetos ligados à saúde de cada paciente. Em particular surgem as aplicações que apoiam o diagnóstico e análise de lesões na pele associadas às doenças dermatológicas. Os trabalhos relacionados analisados possibilitaram compreender o papel da tele dermatologia no mundo atual, em particular os projetos redirecionados para as aplicações móveis na área da dermatologia.

Neste projeto, o sistema desenvolvido para suportar pessoas que sofrem de psoríase pretende contribuir para melhorar o acompanhamento e a comunicação à distância entre especialistas e pacientes através de dispositivos móveis. Entre outros aspetos, a aplicação móvel disponibiliza o instrumento PASI para registar periodicamente a severidade da doença. Esta informação é recolhida e enviada para o servidor juntamente com as fotos das lesões que também podem ser registadas. Esta informação alojada no *backend* é opcional, mas permitirá disponibilizar informação histórica da evolução das áreas afetadas com a equipa de saúde que acompanha o paciente.

Dada a importância da usabilidade no sucesso da adoção das aplicações móveis, procedeu-se a uma avaliação exaustiva da usabilidade da aplicação desenvolvida, para analisar a sua eficiência e detetar eventuais constrangimentos na sua utilização. Com esta avaliação, foi possível identificar e introduzir algumas melhorias na versão final disponibilizada.

5.1 Trabalho Futuro

Este projeto pretende contribuir para o suporte de pacientes e especialistas na área da dermatologia, mais especificamente no auxílio e promoção da saúde e qualidade de vida de pessoas que sofrem de psoríase. Temos consciência que a aplicação e sistema desenvolvidos se encontram ainda numa fase embrionária. Contudo, consideramos que se trata de

um projeto que oferece um suporte e mais valias importantes para o acompanhamento de pessoas com psoríase. Este protótipo poderá ser portanto o ponto de partida para o desenvolvimento de algo mais complexo, com o intuito de se tornar numa ferramenta cada vez mais importante no apoio à monitorização e gestão de doentes com psoríase.

Futuramente a solução proposta poderá ser complementada com várias funcionalidades que permitam auxiliar os profissionais de saúde na avaliação constante e mais próxima dos seus pacientes. Nesta fase permite já registar e anotar as fotos das lesões, bem como fazer uma autoavaliação da severidade da doença através do PASI. A aplicação deverá evoluir futuramente para suportar a classificação automática da área das lesões, para se poder calcular, também de forma automatizada, a quantidade exata de medicação a aplicar. Este aspeto é fundamental para auxiliar a adesão ao tratamento dos pacientes e poderá tirar partido da utilização de classificadores de Machine Learning (ML). Em suma, este projeto permitiu melhor compreender as necessidades mais básicas de suporte a pessoas com psoríase. Desta forma permitiu-nos estruturar, desenvolver e avaliar um sistema e uma aplicação móvel que poderá futuramente incorporar novas funcionalidades que apoiem cuidados de saúde personalizados.

Referências Bibliográficas

- A. Abran, W. Suryn, A. Khelifi, J. Rilling, A. Seffah, and F. Robert. Consolidating the iso usability models. 2003.
- Hanifah Muslimah Az-zahra, Nafilah Fauzi, and Agi Putra Kharisma. Evaluating e-marketplace mobile application based on people at the center of mobile application development (pacmad) usability model. *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, pages 72–77, 2019.
- V. Basili, G. Caldiera, and H. D. Rombach. The goal question metric approach. 1994.
- Mirna Becevic, Blake Anderson, J. Han, E. R. Mutrux, L. Hicks, K. Edison, and C. Shyu. Telemdid: Mobile technology applications for interactive diagnoses in teledermatology clinics. *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)*, pages 429–433, 2013.
- J. Cho and Won-Whoi Huh. Usability evaluation frameworks of mobile apps on smart phone. *Asia-pacific Journal of Multimedia services convergent with Art, Humanities, and Sociology*, 7:841–848, 2017.
- A. K. Clark, S. Bosanac, Baran Ho, and R. Sivamani. Systematic review of mobile phone-based teledermatology. *Archives of Dermatological Research*, 310:675–689, 2018.
- J. Dalton. Goal, question, metric (gqm). 2019.
- Nacional de Saúde. Aplicação de um sistema de teledermatologia no hospital do divino espírito santo nos açores: análise custo-efectividade. 2011.
- Ingrid do Nascimento Mendes and A. C. Dias-Neto. A process-based approach to test usability of multi-platform mobile applications. In *HCI*, 2016.
- Norasikin Fabil. Extension of pacmad model for usability evaluation metrics using goal question metrics (gqm) approach. 2015.
- J. Frühauf, G. Schwantzer, C. Ambros-Rudolph, W. Weger, V. Ahlgrimm-Siess, W. Salmhofer, and R. Hofmann-Wellenhof. Pilot study using teledermatology to manage high-need patients with psoriasis. *Archives of dermatology*, 146 2:200–1, 2010.

- F. Ghabban, Mohammed Hajjar, and Saad T. Alharbi. Usability evaluation and user acceptance of mobile applications for saudi autistic children. *Int. J. Interact. Mob. Technol.*, 15:30–46, 2021.
- R. Harrison, D. Flood, and D. Duce. Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science*, 1:1–16, 2013.
- Norleyza Jailani, Z. Abdullah, Marini Abu Bakar, and Harni Rohaida Haron. Usability guidelines for developing mobile application in the construction industry. *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, pages 411–416, 2015.
- Feroze Kaliyadan and S Venkitakrishnan. Teledermatology: clinical case profiles and practical issues. *Indian journal of dermatology, venereology and leprology*, 75 1:32–5, 2009.
- Funmilayo Kasali, Oluwabunmi Olaniyan, Ibidapo Akinyemi, O Alaba, O Awodele, and S Kuyoro. An enhanced usability model for mobile health application. 08 2020.
- S. Koller, R. Hofmann-Wellenhof, D. Hayn, W. Weger, P. Kastner, G. Schreier, and W. Salmhofer. Teledermatological monitoring of psoriasis patients on biologic therapy. *Acta dermato-venereologica*, 91 6:680–5, 2011.
- M. Kurosu. Usability, quality in use and the model of quality characteristics. In *HCI*, 2015.
- Mohamed Lachgar, Hanane Benouda, and Selwa Elfirdoussi. Android rest apis: Volley vs retrofit. *2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, pages 1–6, 2018.
- J. R. Lewis. Ibm computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *Int. J. Hum. Comput. Interact.*, 7:57–78, 1995.
- Hélio Amante Miot, Maurício Pedreira Paixão, and Chao Lung Wen. Teledermatologia - passado, presente e futuro. *Anais Brasileiros De Dermatologia*, 80:523–532, 2005.
- D. Moreno-Ramírez and G. Argenziano. Teledermatology and mobile applications in the management of patients with skin lesions. *Acta dermato-venereologica*, Suppl 218: 31–35, 2017.
- J. Nielsen. Usability engineering. In *The Computer Science and Engineering Handbook*, 1997.
- D. Okuboyejo. Automating skin disease diagnosis using image classification. *OMICS journal of radiology*, 2013.

- Vikash Paudel and Deepa Chudal. Carbamazepine-induced toxic epidermal necrolysis managed by mobile teledermatology in covid-19 pandemic in rural nepal. *Case Reports in Dermatological Medicine*, 2020, 2020.
- Eduard C. Popovici, Mihai Stangaciu, and Mihnea A. Magheti. Mobile application for news access and virtual community interactive services. In *2010 8th International Conference on Communications*, pages 401–404, 2010. doi: 10.1109/ICCOMM.2010.5509007.
- S. M. Rizvi, T. Schopf, A. Sangha, K. Ulvin, and P. Gjersvik. Teledermatology in norway using a mobile phone app. *PLoS ONE*, 15, 2020.
- Luas Rosado and Márcia Ferreira. A prototype for a mobile-based system of skin lesion analysis using supervised classification. *2013 2nd Experiment@ International Conference (exp.at'13)*, pages 156–157, 2013.
- J. P. Sáenz, M. Novoa, D. Correal, and B. Eapen. Skinhealth, a mobile application for supporting teledermatology: A case study in a rural area in colombia. In *MobiHealth*, 2016.
- J. P. Sáenz, M. Novoa, D. Correal, and B. Eapen. On using a mobile application to support teledermatology: A case study in an underprivileged area in colombia. *International Journal of Telemedicine and Applications*, 2018, 2018.
- G. Schreier, D. Hayn, P. Kastner, S. Koller, W. Salmhofer, and R. Hofmann-Wellenhof. A mobile-phone based teledermatology system to support self-management of patients suffering from psoriasis. *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5338–5341, 2008.
- I. Sommerville. *Software engineering / ian sommerville*. 2011.
- R. Thakur, U. Pandey, and J. Chatterjee. Study of mobile application development using mvc framework. 2020.
- B. Trettin, D. B. Danbjørg, F. Andersen, S. Feldman, and H. Agerskov. An mhealth app to support patients with psoriasis in relation to follow-up consultations: Qualitative study. 2021a.
- Bettina Trettin, Dorthe Boe Danbjørg, Flemming Andersen, Steven Feldman, and Hanne Agerskov. Development of an mhealth app for patients with psoriasis undergoing biological treatment: Participatory design study. *JMIR Dermatol*, 4(1):e26673, May 2021b. ISSN 2562-0959. doi: 10.2196/26673. URL <https://derma.jmir.org/2021/1/e26673>.

M. Vasconcelos and Luís Rosado. Automatic reflection detection on dermatological images acquired via mobile devices. In *MIUA*, 2014.

Pranjali N. Watkar and S. P. Karmore. Efficient integrity protection for android smartphone. 2015.

Samira Yarak, Evandro Eduardo Seron Ruiz, and Ivan Torres Pisa. A teledermatologia na prática médica. 2017.

F. Zahra, A. Hussain, and H. Mohd. Usability evaluation of mobile applications; where do we stand? 2017.