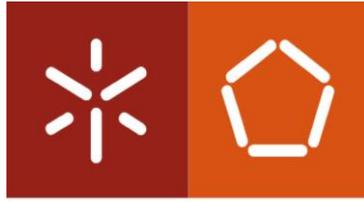


Universidade do Minho
Escola de Engenharia

José Mestre Inácio

**Realidade Aumentada em Aplicações Móveis para
Planeamento Cirúrgico Ortopédico**



Universidade do Minho
Escola de Engenharia

José Mestre Inácio

Realidade Aumentada em Aplicações Móveis para Planeamento Cirúrgico Ortopédico

Dissertação de Mestrado

Mestrado Integrado em Engenharia Biomédica

Ramo de Informática Médica

Trabalho efetuado sob a orientação do

Professor Doutor Victor Manuel Rodrigues Alves

DECLARAÇÃO

Nome: José Alberto Mestre Conceição Inácio

Título dissertação: Realidade Aumentada em Aplicações Móveis para Planeamento Cirúrgico Ortopédico

Orientador: Professor Doutor Victor Manuel Rodrigues Alves

Ano de conclusão: 2017

Designação do Mestrado: Mestrado Integrado em Engenharia Biomédica^[1]_[SEP]

Área de Especialização: Informática Médica

Escola: de Engenharia

Departamento: de Informática

DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA DISSERTAÇÃO

Universidade do Minho, ____/____/_____

Assinatura:

AGRADECIMENTOS

Obrigado a todas as pessoas que me acompanharam nesta jornada.

RESUMO

Cada vez mais as necessidades e os requisitos do paciente acompanham os desenvolvimentos tecnológicos na área da cirurgia médica. Tal acontece para que estes obtenham uma intervenção o mais eficiente e seguro possível por parte dos serviços de saúde e dos seus profissionais. No entanto, hoje em dia ainda é difícil implementar e operar diferentes tipos de tecnologias em ambientes médicos devido às desvantagens que estes podem trazer para os seus utilizadores e por todo o processo de aprendizagem que estas requerem. Numa primeira abordagem, este trabalho tem como objetivo esclarecer conceitos e reunir algumas soluções existentes para a resolução destes problemas, assim como as respetivas tecnologias utilizadas pelas mesmas. Posteriormente é elaborado e apresentado um conceito e protótipo de uma aplicação móvel de planeamento cirúrgico ortopédico que implementa tecnologias de Realidade Aumentada. A solução proposta pretende ajudar o cirurgião desde a fase de planeamento até a própria fase de intervenção cirúrgica. Para além de alguns exemplos e da apresentação do trabalho realizado para a solução, é também descrito o processo de implementação e a arquitetura do sistema. Tendo em conta o protótipo desenvolvido, são discutidas as vantagens da sua utilização num contexto cirúrgico e levantados alguns pontos de interesse futuro a serem estudados e implementados para a sua melhoria.

ABSTRACT

Patient needs and requirements are increasingly following technological developments in the area of medical surgery. This happens, so they can obtain the most efficient and safe intervention possible from the health services and their professionals. However, today it is still difficult to implement and operate different types of technologies in medical environments due to the disadvantages that they can bring to their users and the entire learning process they require. In a first approach, this work aims to clarify concepts and gather some existing solutions to solve these problems, as well as the respective technologies used by them. Subsequently, a concept and prototype of a mobile orthopaedic surgical planning application that implements Augmented Reality technologies is developed and presented. The proposed solution aims to help the surgeon from the planning stage to the surgical intervention phase itself. In addition to some examples and the presentation of the work developed for the solution, the implementation process and the system architecture are also described. Taking into account the developed prototype, the advantages of its use in a surgical context are discussed and are raised some future points of interest to be studied and implemented for its improvement.

ÍNDICE

Agradecimentos.....	v
Resumo.....	vii
Abstract.....	viii
Índice de Figuras.....	xi
Índice de Programas.....	xiii
Notação e Acrónimos.....	xiv
<i>Notação Geral</i>	xiv
<i>Acrónimos</i>	xiv
1. Introdução.....	1
1.1 <i>Contexto</i>	1
1.2 <i>Descrição do Problema</i>	3
1.3 <i>Objetivos</i>	4
2. Planeamento Cirúrgico Ortopédico e Aplicações Móveis de Realidade Aumentada.....	7
2.1 <i>A Importância do Planeamento e da Intervenção Cirúrgica na Ortopedia</i>	7
2.2 <i>Imagem Médica (IME)</i>	8
2.3 <i>Aplicações para Planeamento Cirúrgico Ortopédico</i>	11
2.4 <i>Realidade Aumentada</i>	13
2.5 <i>Aplicações de Realidade Aumentada e Sistemas de Navegação Computorizada</i>	16
3. ARPeek.....	21
3.1 <i>Introdução</i>	21
3.2 <i>Metodologia de Investigação</i>	21
3.3 <i>Tecnologias de Desenvolvimento</i>	23
3.3.1 <i>OpenCV</i>	23
3.3.2 <i>Framework Laravel</i>	24
3.3.3 <i>Unity & MonoDevelop</i>	25
3.3.4 <i>PeekMed – Software de Planeamento Cirúrgico</i>	27
3.4 <i>Levantamento e análise de Requisitos</i>	28
3.4.1 <i>Requisitos funcionais</i>	29
3.4.2 <i>Requisitos não funcionais</i>	30
3.5 <i>Implementação e Conceptualização da tecnologia</i>	31
3.5.1 <i>Aquisição da Imagem Médica e Renderização dos Modelos Anatómicos</i>	32
3.5.2 <i>O Planeamento Cirúrgico Ortopédico</i>	35

3.5.3	Utilização da aplicação de RA	36
3.5.4	Resultado Final.....	37
4.	Desenvolvimento do protótipo	38
4.1	<i>Estudos de Processamento de Imagem – Extração de Caraterísticas.....</i>	<i>38</i>
4.2	<i>A tecnologia de RA</i>	<i>40</i>
4.3	<i>Conversão ficheiro '.obj' num modelo 3D (Unity GameObject).....</i>	<i>47</i>
4.4	<i>Modelo de dados.....</i>	<i>50</i>
4.5	<i>Arquitetura do Sistema - Servidores e Web Pages</i>	<i>54</i>
4.6	<i>Aplicação móvel para iOS – Desenvolvimento e UI.....</i>	<i>61</i>
5.	Conclusões, Publicações e Trabalho Futuro	67
5.1	<i>Conclusões.....</i>	<i>67</i>
5.2	<i>Publicações</i>	<i>68</i>
5.3	<i>Trabalho Futuro</i>	<i>68</i>
	Referências.....	70

ÍNDICE DE FIGURAS

Figura 2.1- Modalidades de imagem médica. a) TC ao joelho onde é visível o contorno da cartilagem articular tibiofemoral e patelofemoral assim como dos limites dos meniscos [9]; b) RM ao joelho que permite a visualização de uma lesão bem demarcada na tibia proximal que afeta a superfície articular do mesmo [10]; c) endoscopia digestiva [11].	9
Figura 2.2 - Arquitetura de um sistema PACS [14].	11
Figura 2.3 - Tecnologias de RA. a) aplicação de RA aplicada à mobilidade em aeroportos [20]; b) Halolens, um protótipo de um dispositivo ótico e auditivo de RA desenvolvido pela MicroSoft [21]; c) aplicação móvel RA de utilizada na área do marketing para impulsionar e melhorar as vendas do setor mobiliário [22]; d) aplicação de RA aplicada ao setor industrial e de produção [23].	14
Figura 2.4 - O <i>software</i> WARM combina o modelo 3D virtual derivado durante o planeamento cirúrgico com imagens reais captadas pela câmara [34].	18
Figura 2.5- Co registo entre o modelo tridimensional e a fotografia sagital do paciente. Em A é possível visualizar a interface do utilizador da aplicação FUSED iOS [39]. A imagem de fundo no painel verde é a fotografia sagital do paciente. A imagem do primeiro plano no painel azul é o modelo tridimensional da cabeça. No painel vermelho, as 2 imagens são fundidas de acordo com os marcadores anatómicos da orelha e a localização do hematoma é projetada na imersão dos dois conteúdos – RA.....	20
Figura 3.1 - Arquitetura e fluxo de um sistema MVS. 1) o <i>Browser</i> envia um pedido ao Controlador que é rastreado para um determinado modelo; 2) o Controlador direciona esse pedido ao Modelo específico; 3, 4) o Modelo avalia e executa a ação pretendida sobre a base de dados e recebe os dados; 5) o Modelo retorna a informação indexada ao Controlador; 6) o Controlador envia um pedido de atualização à <i>View</i> com a informação solicitada; 7) A <i>View</i> processa a nova informação e incorpora-a em HTML; 8) O controlador retorna o HTML devolvido pela <i>View</i> ao <i>Browser</i>	25
Figura 3.2- Etapas à implementação da aplicação <i>ARPeek</i> no planeamento e intervenção cirúrgica ortopédica.....	32
Figura 3.3 - Pixel e Voxel. O valor de cada pixel numa imagem computacional representa o coeficiente de atenuação linear de um elemento com volume de um prisma quadrangular – o voxel [49]. ..	34
Figura 3.4 - Algoritmo <i>Marching Cubes</i> . a) célula utilizada no varrimento do algoritmo; b) tabela <i>look up</i> utilizada pelo algoritmo MC [51].	35

Figura 3.5 - Resultados da implementação do sistema ARPeek. a) utilização da aplicação de RA no processo pré cirúrgico onde é possível visualizar o modelo 3D do pé do paciente assim como fixação do <i>template</i> de uma prótese médica; b) resultado final da intervenção planeada em 'a'.	38
Figura 3.6 - Descrição dos passos para deteção do marcador fiducial.....	43
Figura 3.7 - Modelo de Câmara <i>Pinhole</i> para a calibração da câmara e reconstrução 3D [53].....	45
Figura 3.8 - <i>Bounding Box</i> . Uma representação 2D e 3D de uma caixa delimitadora computada para um determinado objeto [54].	49
Figura 3.9 – Possível modelo lógico de dados e dimensões do sistema apresentado. A 'SIH' pretende representar qualquer Sistema de Informação Hospitalar onde a solução seja implementada. A 'PACS_SYSTEM' representa o sistema de arquivo e comunicação de imagens também da unidade hospitalar em questão.	51
Figura 3.10 – Arquitetura do sistema para implementação da solução <i>ARPeek</i>	56
Figura 3.11 - Funcionamento de um Sistema de Roteamento – <i>Routing</i>	57
Figura 3.12 – Página inicial do protótipo do <i>Web Server</i> desenvolvido.	58
Figura 3.13 - <i>View</i> para registo de utilizador, controlador e <i>route</i> associados: a) com mensagens de erros de consistência e coerência de dados de dados b) corretamente preenchida.....	59
Figura 3.14 - <i>Views do web Server</i> . a) página para autenticação do utilizador; b) página pessoal onde são dispostos os modelos 3D respetivos aos ficheiros carregados, a informação que lhes é associada, informação pessoal do utilizador, botão de navegação para o menu de carregamento de outros modelos e a função de eliminar modelos.	60
Figura 3.15 - <i>View</i> e sistema de carregamento de um ficheiro para o servidor.	61
Figura 3.16 - Aplicação ARPeek – interfaces ao utilizador e navegação.	63
Figura 3.17 - Protótipo do ambiente de visualização do estado final de RA da aplicação.	66

ÍNDICE DE PROGRAMAS

Programa 1 - Passos de pré-processamento de imagem.....	40
Programa 2 - Especificações para a detecção do marcador fiducial no <i>frame</i> de vídeo.....	44
Programa 3 - Algoritmo para determinação do centroide de um modelo importado para Unity.	49
Programa 4 - Algoritmo para centralizar o modelo importado no ponto de origem virtual definido como o centro do marcador fiducial.....	50
Programa 5 - Processo de um pedido web entre a aplicação e servidor - Unity.....	62

NOTAÇÃO E ACRÓNIMOS

Notação Geral

A notação ao longo do documento segue a seguinte convenção:

- Texto em *itálico* – utilizado para equações e fórmulas matemáticas e palavras em língua estrangeira (e.g., Inglês). Adicionalmente utilizado para especificar comandos e funções de programação ou ações e funcionalidade de aplicações.
- Texto em **negrito** – utilizado para realçar uma letra, palavra, conceito ou expressão num parágrafo.

Acrónimos

2D - Bidimensional

3D - Tridimensional

CAOS – Cirurgia Ortopédica Computacionalmente Assistida

DB – *Data Base*

DICOM - *Digital Imaging and Communications in Medicine*

HIPPA - *Health Insurance Portability and Accountability Act*

HTTP - *Hypertext Transfer Protocol*

IA – Inteligência Artificial

IME – Imagem Médica ou Imagiologia

LED - *Light Emitting Diode*

MAR – *Mobile Augmented Reality*

MC – *Marching Cubes*

MVC – *Model-View Controller*

OpenCV – *Open Source Computer Vision*

PACS – *Picture Archive and Communication System*

PCE – Processamento Clínico Eletrónico

PHP - *Hypertext Processor*

RA – Realidade Aumentada
REST – *Representational State Transfer*
RGB – *Red Green & Blue*
RM – Ressonância Magnética
RV – Realidade Virtual
SAD – Sistemas de Apoio à Decisão
SDK – *System Development Kit*
SNC – Sistemas de Navegação Cirúrgica
SQL – *Structured Query language*
TC – Tomografia Computorizada
TI – Tecnologias da Informação
UE – *User Experience*
UI – *User Interface*
URI – *Uniform Resource Identifier*
URL - *Uniform Resource Locator*
UUID – *Universal Unique Identifier*
VC - Visão Computacional

1. INTRODUÇÃO

1.1 Contexto

A área médica é uma área em constante desenvolvimento e na qual surgem todos os dias problemas e necessidades distintas, dentro das suas subdisciplinas e na qual a resolução dos mesmos desperta um elevado interesse, não só social, como económico. Nos dias que correm, os pacientes assumem um papel de consumidores de serviços clínicos, requerendo serem tratados com recurso aos melhores e mais avançados meios de tecnologia e por equipas especializadas nas mesmas. Assim, para além das exigências dos pacientes, a evolução técnica dos profissionais de saúde resulta numa busca constante de soluções e melhorias aos sistemas e tecnologias presentes nesta área.

Sendo a área cirúrgica uma área crítica e de risco na maioria dos seus ramos clínicos, é sobre ela que incide o foco de aprimoramento dos recursos tecnológicos de modo a auxiliar os profissionais envolvidos na mesma e a poder oferecer mais garantias aos pacientes que a ela recorrerem.

Hoje em dia, o sucesso da maioria das cirurgias está relacionado com a qualidade e robustez de uma fase de pré planeamento. Esta consiste, sucintamente, na pré-avaliação do paciente através de informação clínica existente (análises, dados pessoais do paciente, dados do seu quotidiano e exames médicos), a qual irá permitir estabelecer procedimentos cirúrgicos adaptados às necessidades específicas do mesmo.

Para na fase de planeamento as equipas poderem trabalhar com informação pertinente, útil e detalhada, o campo da imagem médica foi alvo de grandes evoluções tecnológicas. A tomografia computadorizada (TC), a ressonância magnética (RM), radiografia e ultrassonografia permitiram que temas como o diagnóstico, prevenção e acompanhamento dos pacientes melhorasse assim como possibilitou o acesso a informações visuais de grande utilidade para a área cirúrgica.

Todo o tipo de informações envolvida neste processo de planeamento, e mesmo em qualquer outra área clínica, necessita de ser formalizada, padronizada e processada, não só para facilitar a sua interpretação e estudo, mas também para permitir a sua transmissão e comunicação. O formato *Digital Imaging and Communications in Medicine* (DICOM) reúne um conjunto de normas para o armazenamento, processamento e transmissão de informação médica, definindo um protocolo para a realização dos mesmos com grande importância para a área da imagem médica [1].

A imagem é algo que desperta aos seus observadores curiosidade a nível de interação, manipulação e sensibilidade. Tal facto levou ao surgimento e desenvolvimento de tecnologias de *Augmented Reality* (AR) não só em áreas de saúde, industriais e educacionais, como também no quotidiano [2]. De forma a fundir o conteúdo digital no mundo real, este tipo de tecnologias AR empregam outras como as de visão computacional, processamento de imagem e técnicas de computação gráfica [3].

Num contexto cirúrgico e de modo a auxiliar o cirurgião e a sua equipa na fase de planeamento, certas aplicações têm surgido de modo a gerar computacionalmente modelos tridimensionais (3D) da anatomia humana recorrendo a imagens obtidas e formalizadas por TC e RM [4]. Após geração destes modelos é possível que os profissionais analisem deficiências e inconsistências nos mesmos. Para além de ser possível visualizar dinamicamente o modelo gráfico principal nos ambientes de visualização destas aplicações, é ainda possível recorrer a outras funcionalidades para realizar estudos e medições sobre o mesmo ou até sobrepor outros objetos virtuais de forma a simular por exemplo a fixação de uma prótese ou de uma ferramenta cirúrgica. Finalizada essa análise e combinada com as restantes informações clínicas do paciente, é então possível realizar um estudo e planeamento das etapas e objetivos da intervenção. A nível de planeamento cirúrgico, as áreas nas quais este tipo de aplicações tem sofrido um maior desenvolvimento e despertado um maior interesse empresarial são o planeamento cirúrgico ortopédico, a oftalmologia e a medicina dentária.

A problemática da mobilidade, eficiência e conveniência deste tipo de soluções tem vindo a ser suprimida pelo grande avanço tecnológico e pela maior capacidade de, hoje em dia, a informação ser recolhida, partilhada e processada. Tudo isto é resultado de uma melhoria constante dos dispositivos móveis tanto a nível de hardware como também na sua capacidade de armazenamento e processamento de informação. A evolução das próprias redes de comunicação e partilha de dados entre dispositivos vêm auxiliar todo este processo, viabilizando a passagem deste tipo de aplicações de dispositivos fixos para móveis. Estas condições permitem que as aplicações que lidem com muitos conteúdos digitais e com um grande fluxo de dados possam ser implementadas em simples, portáteis e cómodos dispositivos, o que abre uma janela para o desenvolvimento de aplicações de realidade aumentada (RA) em diversas áreas, sendo uma das mais interessantes a área médico-cirúrgica.

1.2 Descrição do Problema

A área cirúrgica é uma área crítica e importante no que toca ao desenvolvimento da medicina e das tecnologias aplicadas a esta. Desde a etapa de planeamento até à intervenção em si, diversos sistemas, ferramentas e soluções têm sido desenvolvidas para facilitar o trabalho dos profissionais envolvidos e melhorar a qualidade e eficiência dos serviços prestados.

No entanto, atualmente um dos maiores problemas enfrentado pelos médicos e cirurgiões nos blocos operatórios é a falta de soluções e tecnologias que permitam, durante a intervenção, a visualização e reprodução dos conteúdos e estudos que foram preparados na fase de planeamento [4]. Assim, não é possível à equipa acompanhar a intervenção, avaliar os seus procedimentos e monitorizar as suas ações técnicas de uma forma prática, simples e intuitiva sem que existam limitações do espaço de visão/ação ou mais requisitos logísticos.

Uma primeira solução apresentada para esta problemática foram os Sistemas de Navegação Cirúrgicos (SNC). Inicialmente estes sistemas eram implementados somente no intuito de auxiliar e aumentar o campo de visão do cirurgião através de microcâmaras, microscópios e monitores de rastreamento. Estes sistemas permitiriam assim o acompanhamento em tempo real do estado de órgãos e outros tecidos anatómicos que não seriam visíveis ao cirurgião e à sua equipa, mais especificamente de tecidos internos do corpo do paciente.

Numa fase posterior, estes sistemas foram imergidos com tecnologias de AR onde, para além de o campo de visão do cirurgião ser aumentado, este poderia sobrepor sobre a imagem captada, em tempo real pelas microcâmaras, conteúdo gráfico previamente produzido relativo ao paciente e à área, sistema ou órgão sujeito a intervenção.

Numa sala de planeamento cirúrgico ou num bloco operatório é importante que os profissionais disponham de todos os recursos necessários ao acompanhamento e/ou intervenção e que estes lhes sejam facultados de maneira rápida e eficaz.

Apesar do sucesso inicial dos SNC, algumas limitações têm lhes sido apontadas ao longo do tempo, muitas delas associadas a questões de mobilidade e portabilidade. O extenso processo de configuração, a complexidade de manuseamento dos dispositivos envolvidos e o lento processo de aprendizagem e de implementação são algumas das desvantagens descritas. Sendo que todas as tecnologias de RA necessitam de um processo de calibração, este é também apontado como um desafio e ponto a melhorar nas soluções existentes, uma vez que muitas delas recorrem a processos demorados e difíceis que por vezes resultam, ainda assim, em imprecisões e falhas. Outra desvantagem apontada é a inexistência de uma uniformização dos processos e funcionalidades envolvidas nestes sistemas, ou

seja, a inexistência de soluções que englobem a fase de aquisição dos modelos tridimensionais, com a sua manipulação, visualização e associação a outras informações clínicas. Adicionalmente, o elevado investimento que estas tecnologias acarretam nos dias hoje é também um entrave à sua implementação nas unidades cirúrgicas, não só pela complexidade dos marcadores de rastreamento utilizados (abordado na secção de Planeamento Cirúrgico Ortopédico e Aplicações Móveis De RA), mas também devido ao custo dos componentes digitais de captação de imagem.

Apesar da já existência de aplicações móveis que implementam funções de RA, poucas são as desenvolvidas para uso clínico e/ou cirúrgico, sendo mais notória a sua utilização para fins lúdicos ou educativos. Os dispositivos portáteis e pessoais sofreram uma grande evolução tecnológica nos últimos tempos não só a nível de materiais e usabilidade como também a nível de processamento e funcionalidades. As tecnologias AR atuais de contexto clínico e cirúrgico falham no que toca a potencializar e explorar todas as vantagens e comodidades que estes dispositivos trazem aos seus utilizadores o que poderia resolver muitos dos problemas descritos anteriormente.

Assim sendo, a integração de aplicações móveis de RA nas práticas clínicas que recorrem às diversas modalidades de imagem médica, como a RM ou a TC, deve garantir a prestação das funcionalidades a que se comprometem, mas ter atenção outros fatores como a mobilidade, usabilidade, consistência dos dados, flexibilidade e acessibilidade [2]. Uma aplicação móvel de AR deverá procurar adaptar-se ao campo clínico em que é inserida de maneira a proporcionar uma experiência mais imersiva e envolvente entre o médico e o paciente, tanto no processo de diagnóstico como no de planeamento e acompanhamento cirúrgico, mas sem negligenciar a exigência e a minuciosidade do mesmo [3].

1.3 Objetivos

A importância deste trabalho centra-se na progressão e no aumento do uso dos recursos tecnológicos atuais na área da cirurgia médica de modo a melhorar a qualidade das intervenções clínicas tanto para os profissionais como para os pacientes assim como tentar reduzir os riscos e inconsistências das mesmas.

O objetivo principal desta dissertação prende-se pela projeção e desenvolvimento de uma solução que melhore certos procedimentos cirúrgicos, desde a fase de planeamento até à intervenção em bloco operatório, de uma forma simples e viável. Esta solução será inserida e projetada num contexto cirúrgico ortopédico, sendo esta uma área da cirurgia onde as funcionalidades e competências da solução poderão ser potencializadas através de tecnologias móveis de RA. Esta irá fornecer ao cirurgião e à sua equipa

uma ferramenta que integra a possibilidade de simular a intervenção ortopédica e definir algumas das suas diretrizes, com a capacidade de visualizar tudo isto durante a fase de bloco operatório de uma forma aumentada. Isto será possível graças à sobreposição dos conteúdos gráficos, desenvolvidos e preparados na fase de planeamento, sobre o ambiente operativo em tempo real. A visualização dos modelos 3D computadorizados irá ajudar e fornecer suporte ao cirurgião em certos detalhes da cirurgia sem que este perca a percepção do ambiente, do espaço e do corpo do paciente.

Para tal foi realizado, primeiramente, uma análise relativa às técnicas e tecnologias de AR implementadas atualmente a nível cirúrgico, assim como ao seu estado de evolução. Sobre estas foi realizada uma avaliação e levantamento dos requisitos dos seus utilizadores e avaliadores assim como das características dos meios em que são inseridas. Foi ainda analisada alguma literatura mais técnica para o esclarecimento de alguns conceitos relacionados com as mesmas.

Antes do desenvolvimento do protótipo foi realizado um levantamento das ferramentas e *software* necessários para o desenvolvimento do mesmo assim como a seleção dos que melhor se adequariam aos requisitos deste.

De forma a encontrar o melhor processo de calibração dos dispositivos de rastreamento envolvidos na solução AR (fase importante e essencial para qualquer aplicação de RA), foram realizados alguns estudos de processamento de imagem tendo em conta o tipo de dispositivos e o meio em que serão utilizados (salas de planeamento pré-cirúrgico, blocos operatórios e consultórios).

Após a conceptualização da arquitetura, da estrutura e de algumas funcionalidades da solução, foi desenvolvida uma aplicação protótipo e um servidor *web* de forma a testar e verificar a eficiência e usabilidade da solução proposta aquando inserida na aplicação de planeamento cirúrgico ortopédico já existente da *Peek Health S.A* (Ribeiro et al., 2015). Por fim, após algumas simulações e testes serem concretizados, foi realizada uma avaliação às competências da solução em conjunto com um levantamento das direções e melhorias a serem consideradas no futuro.

2. PLANEAMENTO CIRÚRGICO ORTOPÉDICO E APLICAÇÕES MÓVEIS DE REALIDADE AUMENTADA

2.1 A Importância do Planeamento e da Intervenção Cirúrgica na Ortopedia

O planeamento é uma disciplina importante na área da cirurgia médica. Nele é possível prever e antecipar problemas que surjam durante a intervenção, mas também desenvolver planos alternativos para casos de emergência ou surpresa que possam ocorrer durante a mesma. Esta fase consiste na definição e projeção de diretrizes apropriadas e específicas suportadas pela informação do paciente ou até mesmo por dados provenientes de outras especialidades médicas como exames, análises ou históricos clínicos. Nesta, deve ser realizada em equipa uma análise detalhada das opções e abordagens existentes assim como uma avaliação dos riscos e benefícios da cirurgia, que poderão ser posteriormente comunicados ao paciente. Neste cenário, a situação do paciente pode ser previamente e meticulosamente analisada com tempo e sem pressões de tomada de decisões. Desta forma, durante a intervenção, estes fatores podem ser atenuados e a equipa poder-se-á focar no seu rigor e eficácia.

O planeamento cirúrgico está associado a muitos casos de sucesso em diferentes especialidades cirúrgicas. A elaboração de um plano cirúrgico é também extremamente importante para a monitorização do tempo de recuperação do paciente. A confiança e requisitos do paciente são aspetos a ter em consideração. Assim, cada vez mais a relação entre o tempo de recuperação e a reabilitação está estreitamente relacionada com o sucesso do processo de planeamento [4]. Hoje em dia os cirurgiões e as suas equipas especializadas estão constantemente em busca de ferramentas que aumentem a sua competência e precisão de forma a garantirem aos seus pacientes uma melhor performance [5]. Atualmente, a maioria das cirurgias é precedida por uma fase de planeamento pré-operativo, nomeadamente em áreas como a ortopedia, odontologia, oftalmologia, cirurgia plástica e de reconstrução.

O objetivo principal do planeamento cirúrgico é a definição de vários parâmetros que assegurem uma intervenção a menos invasiva possível, simples e bem-sucedida, não só para o cirurgião e a sua equipa, como também para o próprio paciente.

Inicialmente este processo era realizado manualmente pelos profissionais e não requeria grandes dispositivos tecnológicos, também pela inexistência dos mesmos. Com a evolução das tecnologias, algumas soluções informáticas orientadas para uso médico foram desenvolvidas de forma a facilitar o

planeamento pré-operatório, nomeadamente na área da cirurgia ortopédica. Estas fornecem certas funções e módulos para a análise de diferentes regiões anatómicas e também disponibilizam modelos computacionais para simulação e visualização de implantes ortopédicos como próteses, placas e outros dispositivos de correção [6].

J. Ribeiro e V. Alves referem no seu trabalho que tem sido atribuída uma grande importância ao planeamento cirúrgico na área da ortopedia. Destacam ainda que outros autores referem o planeamento cirúrgico ortopédico como preponderante na redução da falha do cirurgião e que, ainda hoje, o facto de este não poder ser claramente perceptível e consultado durante a cirurgia em si, pode levar ao insucesso da mesma [7].

2.2 Imagem Médica (IME)

Com a evolução da medicina, das tecnologias inerentes à mesma e do próprio conhecimento humano, muitas áreas médicas sofreram grandes avanços. A especialização de profissionais e de tecnologias para áreas específicas, de forma a suprimir os requisitos dos seus profissionais e as necessidades dos seus pacientes, é cada vez mais notória nos hospitais. Hoje em dia, estas instituições acabam por funcionar e serem geridas como grandes empresas onde a inovação e melhoria constante são palavras de ordem.

As Tecnologias de Informação (TI) surgiram na área médica na segunda metade do século XX. Num curto espaço de tempo, os progressos e descobertas tecnológicas permitiram o desenvolvimento e diferenciação de diversas ferramentas de apoio. Inicialmente, ao diagnóstico e à decisão, com o Processo Clínico Eletrónico (PCE) e os Sistemas de Apoio à Decisão (SAD), respetivamente. Mais recentemente estes progressos abrangeram o próprio tratamento com a implementação de tecnologias de Inteligência Artificial (IA) e de Imagem Médica ou Imagiologia (IME) [7].

Os sistemas IME são uma área da medicina que evoluiu consideravelmente nos últimos anos devido aos interesses, não só sociais, como também económicos e empresariais.

A Imagem Médica é descrita como a técnica e processo não invasivo para obtenção de uma reprodução gráfica de um tecido ou órgão humano para análise clínica e intervenção médica assim como visualização da função dos mesmos em certas modalidades. As imagens são obtidas recorrendo a instrumentos mecânicos que, através da radiação ou de campos magnéticos, conseguem reproduzir tecidos humanos independentemente da sua densidade e opacidade aos mesmos. Assim, é possível a

caraterização de estruturas internas do corpo humano não visíveis a olho nu, devido à sobreposição de outras.

Os avanços nas tecnologias de aquisição de imagem como a RM e TC permitiram não só a obtenção de imagens com mais qualidade e de maior resolução assim como a obtenção das mesmas em maior número [8]. A Imagem Médica abrange ainda mais modalidades de imagem, tais como endoscopia ou microscopia.



Figura 2.1- Modalidades de imagem médica. a) TC ao joelho onde é visível o contorno da cartilagem articular tibiofemoral e patelofemoral assim como dos limites dos meniscos [9]; b) RM ao joelho que permite a visualização de uma lesão bem demarcada na tibia proximal que afeta a superfície articular do mesmo [10]; c) endoscopia digestiva [11].

Com o desenvolvimento e aumento do número de modalidades de IME e da sua importância e utilidade em contexto clínico e cirúrgico, a quantidade de informação e dados resultantes da mesma aumentou exponencialmente. Surgiu então a necessidade de padronizar e normalizar tanto os processos de aquisição, como os de armazenamento, processamento e transmissão destes. Apareceu assim o padrão *Digital Imaging and Communication in Medicine* (DICOM). Este define protocolos para diversas técnicas de imagem desde a radiografia até à tomografia computadorizada, ressonância magnética e ultrassonografia [1]. São também definidos padrões para a compressão das imagens, recriação, visualização 3D e apresentação da mesma com os metadados a ela associados. Assim, para além das informações relevantes que estão contidas nos pixels ou voxels, também estão padronizadas por este

protocolo outro tipo de informações, desde as relativas ao paciente aos quais pertencem, até às mais técnicas como o aparelho, posicionamento ou tipo de exame realizado [8].

Hoje em dia, a maioria dos fabricantes de sistemas de imagem médica disponibilizam *softwares* e soluções para arquivar, visualizar e analisar as mesmas em contextos personalizados às áreas clínicas como a oncologia, neurologia e cardiologia. Estes *Picture Archiving and Communication Systems* (PACS) (figura 2.2) são caracterizados por ambientes interativos de fluxo e visualização que permitem e facilitam a comunicação entre diversas unidades desde o processo de aquisição até ao de armazenamento e transmissão das respetivas imagens médicas [12]. Para tal, o formato DICOM é considerado o universal para este tipo de sistemas.

Recorrendo a diversas tecnologias *web* existentes, os PACS têm a capacidade de permitir acesso atempado e eficiente a imagens, interpretações e dados relacionados com as mesmas, reduzindo assim barreiras físicas e temporais associadas à recuperação, distribuição e exibição das mesmas.

A implementação de soluções externas a este tipo de sistemas é algo recorrente hoje em dia, sendo que estes são desenvolvidos também com o propósito de permitir a interoperabilidade de diversos componentes e implementação de aplicações exteriores. Como tal, é importante perceber qual a arquitetura primária destes e quais os seus componentes. O primeiro componente dos PACS são as modalidades de aquisição de imagem médica, desde a tomografia computadorizada até à ressonância magnética. Os dados obtidos nesta fase seguem para uma estação de garantia de qualidade onde é verificado se a disposição e posicionamento do paciente estão corretos assim como outros componentes e definições da modalidade de referência. O dispositivo de armazenamento central (arquivo) é responsável pelo armazenamento das imagens e, em alguns casos, de relatórios, medições e outras informações que a mesma contenha. Na estação de trabalho e leitura é onde geralmente um radiologista realiza a análise e estudo do paciente, formulando o seu diagnóstico, do qual geralmente resulta um relatório estruturado através de *softwares* específicos existentes nestas estações. Estes relatórios são posteriormente arquivados no sistema de informação hospitalar de modo a que os diferentes profissionais, mediante as suas permissões, possam aceder e consultar os mesmos [12][13].

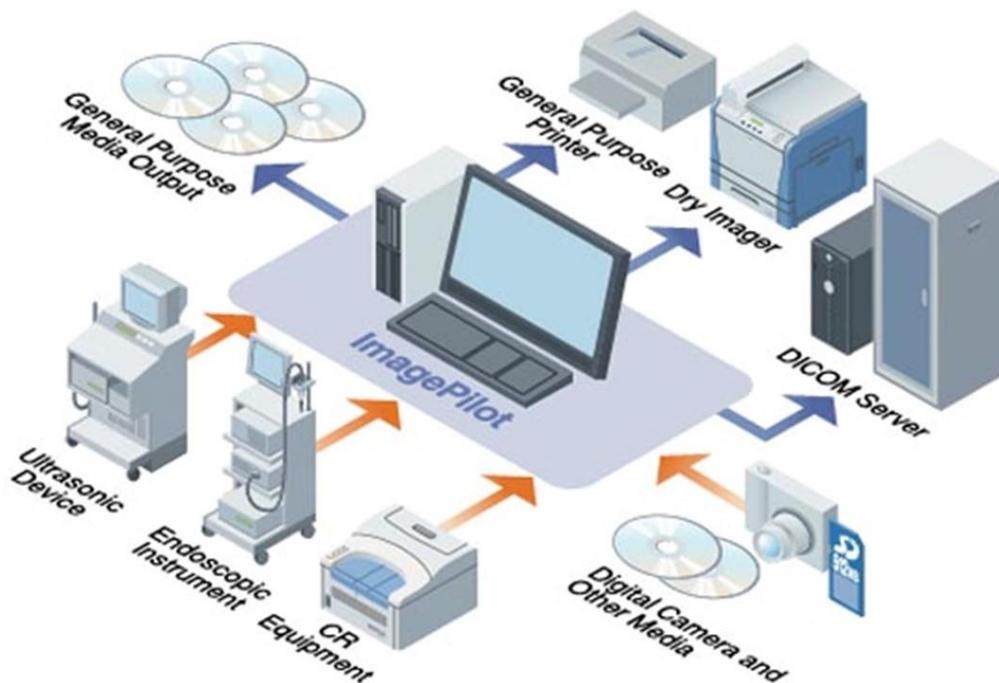


Figura 2.2 - Arquitetura de um sistema PACS [14].

2.3 Aplicações para Planejamento Cirúrgico Ortopédico

Muitos avanços e investimentos têm sido feitos nesta área da cirurgia de modo a promover o desenvolvimento de tecnologias que consolidem todas as necessidades que os profissionais têm na fase de planejamento, mas que permitam também a passagem e visualização dessa informação para o bloco operatório de uma maneira prática e competente.

Sendo que a área da ortopedia é das especialidades médicas que mais recorre às tecnologias de IME e na qual esta desempenha um papel preponderante no diagnóstico, tratamento e acompanhamento dos pacientes, a quantidade de informação e dados neste ramo cresceram abruptamente. Assim, com a necessidade de manipular, interagir e estudar essa informação, surgiram os Sistemas de Cirurgia Ortopédica Computacionalmente Assistida (CAOS). Estes sistemas vieram trazer uma nova abordagem ao processo cirúrgico por parte da equipa envolvida no mesmo, possibilitando o tratamento, manipulação e interação antes, durante e após a cirurgia, dos dados adquiridos. Um sistema CAOS assenta em quatro processos principais: a aquisição dos dados e modelação, planeamento pré cirúrgico, suporte durante a cirurgia e acompanhamento pós cirúrgico. A aquisição é realizada recorrendo a vários sistemas sendo que os mais relevantes são os de RM e TC por não serem invasivos e permitirem uma representação clara da patologia. A modelação é realizada por *softwares* de processamento de imagem que permitem

a segmentação das mesmas e a extração de características preponderantes para o sucesso da intervenção cirúrgica. Segue-se a fase de planeamento onde o cirurgião e a sua equipa podem ter a perceção da extensão da lesão ou doença, elaborar um plano de intervenção (por exemplo na escolha do bloco onde a intervenção irá ocorrer até à disposição do paciente e do tipo de anestesia necessária) e selecionar quais os materiais necessários, quer cirúrgicos quer a nível de implantes quando necessário [15].

Uma das primeiras aplicações destes sistemas foi no desenvolvimento de próteses e implantes em 1986 [7]. Os CAOS assumiram um papel interessante na área devido ao facto de os resultados finais nos pacientes apresentarem uma relação estreita entre a tecnologia e a experiência e precisão do cirurgião. Foi a partir deste tipo de soluções, onde foi notória a melhoria e sucesso dos procedimentos, que surgiu a necessidade das diversas áreas médicas associarem o conhecimento dos profissionais ferramentas tecnológicas a diversificadas e inovadores [7].

Atualmente existem algumas ferramentas tecnológicas que permitem realizar o planeamento cirúrgico ortopédico recorrendo a imagens bidimensionais (2D), como por exemplo o *OrthoView* [16]. Com a sua integração no sistema PACS da instituição hospitalar, é possível à equipa de cirurgia obter as imagens das modalidades de interesse e, após análise das mesmas, planear a intervenção assim como definir as dimensões dos implantes a serem utilizados. Esta ferramenta reproduz ainda um relatório final onde é exibida toda a informação resultante do planeamento. Apesar do seu conceito inovador, a aplicação apresenta algumas lacunas a partir do momento em que não é disponibilizado ao médico a possibilidade de este analisar a patologia de diversos ângulos e em detalhe devido à impossibilidade de visualizar tridimensionalmente o órgão em questão. Para o caso de fraturas ósseas, esta análise pode fazer grande diferença na abordagem a ser adotada durante a cirurgia, o que revela que a solução é insuficiente quando se procura precisão e rigor no processo de planeamento cirúrgico ortopédico.

Outra ferramenta semelhante à anterior é a *TraumaCad* [17]. Ambas as soluções fazem uso de *templates* 2D de próteses a serem colocadas na cirurgia. No entanto, como esta representação não vai além da representação em duas dimensões, é difícil para o cirurgião ter a verdadeira perceção espacial do trauma e realizar assim uma intervenção mais planeada e precisa.

Sendo que a visualização tridimensional dos objetos permite uma análise com mais cuidado e detalhe e sendo isso um fator de extrema importância quando os cirurgiões lidam com a saúde de um paciente, a *Materialise* desenvolveu uma aplicação móvel denominado por *Surgicase* [18]. Recorrendo a exames TC do paciente, o *software* permite a conversão das imagens 2D para uma representação 3D dos objetos analisados. Na aplicação o médico pode visualizar de diversas formas e ângulos os objetos tridimensionais reproduzidos assim como segmentá-los de acordo com o foco da intervenção.

Recorrendo a uma equipa de engenheiros de apoio da mesma empresa, o cirurgião pode solicitar que estes recriem e simulem a aplicação da prótese a ser aplicada, sendo esta também representada em 3D. Apesar de interessante, a abordagem proposta pela aplicação retira alguma autonomia ao cirurgião no contexto em que este não pode adicionar, manipular ou testar outro tipo de *templates* e disposições dos mesmos. Para além disso, o fluxo de planeamento poder-se-á revelar mais demorado sendo que é necessário recorrentemente e a cada alteração solicitar a intervenção de um engenheiro [7].

Em suma, têm surgido atualmente muitas aplicações que funcionam como módulos dos sistemas de PACS das unidades hospitalares. Hoje em dia, recorrendo ao estudo e desenvolvimento das tecnologias de imagem computacional e de processamento de imagem que recebem como input dados padronizados em formato DICOM, é possível reproduzir modelos gráficos tridimensionais de órgãos e/ou tecidos humanos assim como manipula-los e visualiza-los digitalmente. Apesar do sucesso inquestionável deste processo, é ainda difícil e limitado para o cirurgião ter a perceção de qual o impacto que a intervenção terá no paciente durante e após a mesma. Este fator ganha um maior peso em cirurgias que envolvam implantes como a cirurgia ortopédica [4]. Seria de todo o interesse e importância que o cirurgião e a sua equipa pudessem analisar conjuntamente o posicionamento e disposição dos implantes ou dispositivos no corpo do paciente, assim como registar a sua real posição e localização, o que só é atualmente possível em tempo e espaço real, ou seja, durante a própria intervenção.

2.4 Realidade Aumentada

Os avanços tecnológicos e a crescente curiosidade e necessidade de interação, manipulação e visualização da informação, foram os principais fatores que levaram ao desenvolvimento de tecnologias de RA [2].

A RA é definida pela integração e sobreposição de informações virtuais num ambiente de perceção e tempo real, permitindo ao utilizador manipular e visualizar informações de forma prática, intuitiva e emocionante [19]. A visualização da combinação entre conteúdo virtual e real pode ser feita usando dispositivos móveis, telas, dispositivos óticos transparentes, monitores, auscultadores e janelas fictícias que podem interagir com o utilizador ou individualmente.

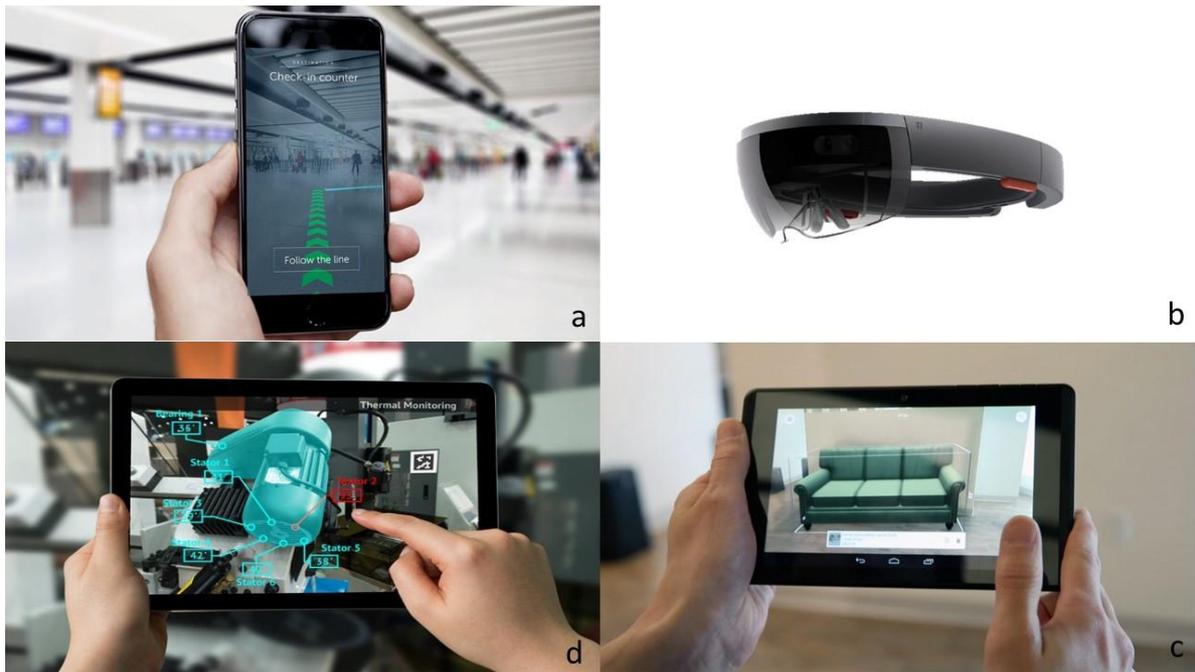


Figura 2.3 - Tecnologias de RA. a) aplicação de RA aplicada à mobilidade em aeroportos [20]; b) Halolens, um protótipo de um dispositivo ótico e auditivo de RA desenvolvido pela MicroSoft [21]; c) aplicação móvel RA de utilizada na área do marketing para impulsionar e melhorar as vendas do setor mobiliário [22]; d) aplicação de RA aplicada ao setor industrial e de produção [23].

A maioria das aplicações RA existentes sobrepõe imagens geradas/adquiridas computacionalmente sobre um ambiente real e são suportadas por princípios de reconhecimento de padrões e de estimativa de posicionamento no espaço tridimensional. Como tal, este tipo de soluções implica a realização cíclica de diferentes processos até ser possível ao utilizador ter a percepção e ideia de imersão do mundo virtual no real.

Primeiramente é necessário estimar qual a localização e a orientação relativa do dispositivo de visualização (geralmente uma câmara) perante o meio que está a ser captado. Este processo é realizado através do processo de calibração que permite estabelecer a relação entre as dimensões do ambiente real e o conteúdo gráfico [24]. Este é fundamental para o redimensionamento correto dos modelos virtuais de modo a tornar a sua visualização o mais aproximadamente possível com a realidade.

Para a calibração é necessária a existência de um dispositivo de aquisição de imagem que esteja associado às funcionalidades de visão computacional do sistema. Este deve ser capaz de detetar pontos estratégicos cujas dimensões reais entre eles sejam previamente conhecidas e fornecidas como entrada ao mesmo [3]. Estes algoritmos de visão computacional são basicamente divididos em duas etapas principais: a primeira é a extração das características da imagem capturada denominada por rastreamento; a segunda é a fase responsável pela calibração em si, ou seja, pela estimativa do posicionamento e orientação do dispositivo no ambiente real com os dados obtidos na primeira [25].

Diferentes métodos de rastreamento foram desenvolvidos para várias áreas para além da visão computacional como por exemplo para a robótica ou mesmo fotogrametria [3]. Esses métodos podem ser diferenciados com base nos equipamentos utilizados: métodos de rastreamento por sensores, métodos de rastreamento visual e métodos híbridos. Sendo que na maioria dos sistemas de RA, a câmara é o dispositivo mais implementado, os métodos de rastreamento visual assumem um papel importante na RA. Para além disso, a visão computacional baseada em marcas fiduciais é muito comum neste tipo de sistemas e aplicações devido à simplicidade, precisão e robustez que oferece [26]. Os marcadores fiduciais são objetos, imagens ou marcas posicionadas num ambiente real que apresentam características de fácil captura, identificação e extração, sendo estas características essenciais para a fase de rastreamento. Estes tipos de marcadores visuais não são invasivos nem dispendiosos e praticamente não requerem quaisquer cuidados de configuração e manutenção.

No entanto, atualmente existem outro tipo de marcadores, como os acústicos, os *Light Emitting Diode* (LED) ou até marcadores magnéticos. Estes tipos de marcadores são mais utilizados nos *softwares* de captura de movimento onde são extraídas características como ângulos, velocidades, acelerações e impulsos, que proporcionam uma representação digital mais precisa do movimento. Apesar de populares em diversos cenários, incluindo hospitalares, este tipo de marcadores apresentam um elevado custo e requerem ainda uma enorme organização, preparação e configuração do ambiente sobre o qual são utilizados. Para além da sua manutenção dispendiosa estes requerem ainda sensores de captação mais sofisticados e complexos. Estas características restringem adicionalmente a liberdade de movimento e sensação dos utilizadores e são um entrave à grande implementação das tecnologias de RA, que recorrem a este tipo de tecnologias, em contexto clínico e cirúrgico [27].

Hoje em dia muitas destas aplicações de RA recorrem a dispositivos móveis com hardwares de aquisição de imagem, as *mobile augmented reality* (MAR) *applications*. Devido à sua evolução e à sua capacidade de memorizarem e processarem dados, é possível correr *softwares* que realizem o próprio processamento da imagem e visualizar o conteúdo gráfico sobreposto à imagem real nas telas destes dispositivos móveis. Este tipo de abordagem permite ao utilizador uma maior independência e abstração aliada à capacidade de este poder aceder remotamente a conteúdos do seu interesse. Assim, a evolução dos sistemas de rede móvel permitiu também o desenvolvimento deste tipo de aplicações que trazem uma maior comodidade e oferta de funcionalidades aos seus utilizadores.

2.5 Aplicações de Realidade Aumentada e Sistemas de Navegação Computorizada

Atualmente não existe uma grande implementação de aplicações móveis de RA em contexto clínico ou cirúrgico. No entanto, bastantes foram desenvolvidas para diversas funcionalidades em muitas outras áreas tais como os jogos de vídeo, entretenimento social, editores de imagens, suporte a tarefas do quotidiano (como encontrar o carro ou outros objetos) ou até mesmo no controle da produção industrial [28]. Muitas destas aplicações inserem-se no conceito e na evolução dos chamados ambientes inteligentes, mas na área da medicina, a RA é ainda muito limitada à vertente educacional.

Aplicações de RA associadas a tecnologias de localização têm sido também implantadas em projetos para o desenvolvimento de cidades inteligentes e sustentáveis. Štorková, P. descreve o uso deste tipo de aplicações para na percepção e conhecimento da história das cidades, úteis para a evolução e modernização do turismo [29]. O aplicativo proposto deteta a localização e orientação do utilizador no espaço através do GPS do seu *smartphone* ou *tablet*. Conhecida essa informação, conteúdos multimédia sobre o local são sobrepostos à imagem captada em movimento pela câmara do dispositivo e o utilizador pode navegar e interagir com os mesmos. Estes podem ser sobre forma de texto (*links* para páginas *web* relacionadas, nomes de edifícios, informações históricas, horários de funcionamento), áudio (guias de áudio em diferentes idiomas, gravações antigas, comentários de voz de outros utilizadores da rede da aplicação), vídeo, fotografias e ainda animações com modelos gráficos de edifícios e personagens históricas por exemplo.

Da mesma forma, em *“Advanced Algorithms for Location-Based Smart Mobile Augmented Reality Applications”* é apresentada uma tecnologia similar, mas para a qual o utilizador define previamente quais os seus pontos de interesse e é gerada uma rede social à volta deste conceito com a partilha de fotografias, opiniões e experiências [30].

Num contexto semelhante, mas mais direcionado a questões de acessibilidade e integração de pessoas com dificuldades motoras, auditivas e de expressão, Rovadosky, D. et al. propõem o desenvolvimento de aplicações que poderiam ajudar este tipo de utilizadores em diversos ambientes como supermercados, lojas, bibliotecas ou museus [31]. Recorrendo aos seus dispositivos móveis e à existência de conteúdos visuais e auditivos associados à localização específica dos marcadores detetados no espaço, é possível melhorar o acesso, percepção e, conseqüentemente, a integração e conforto destes utilizadores. Com a evolução dos dispositivos móveis e de todas as funcionalidades integradas nestes, seria ainda possível gerar sinais sensoriais ao utilizador consoante a sua localização de modo a este se orientar corretamente dentro dos espaços. Note-se que neste tipo de aplicações o rastreamento e identificação destes marcadores poderia ocorrer de forma visual ou por GPS através da associação dos

conteúdos aos marcadores ou a localizações geográficas, respetivamente. Rashid, Z. et al. apresentam uma aplicação AR para pessoas com mobilidade reduzida e que enfrentam o problema de não alcançar ou visualizar os produtos das prateleiras mais altas numa livraria [32]. Além disso, um sistema de guia integrado orienta o utilizador para o produto desejado, através da tela dos dispositivos e da sobreposição de instruções e setas de orientação.

Na área da indústria, as aplicações móveis RA também provaram ser uma solução viável para muitos objetivos e barreiras existentes. Mura, M.D. et al. propõem no seu trabalho uma estação inovadora de montagem manual de peças que recorra a este tipo de tecnologias [33]. Esta estação combina o ambiente de produção real, com um ambiente virtual e sensores de força e torsão. Assim, recorrendo a dispositivos de visualização, o profissional pode estudar e planejar o processo de montagem numa experiência mais detalhada e envolvente, recebendo indicações de como posicionar e interagir com os vários componentes e partes do produto a serem montadas. O sistema proposto parece apresentar vantagens em relação às correntes, na medida em que as falhas e os erros humanos são minimizados assim como o investimento em tempo de formação é reduzido.

As aplicações de RA no campo da imagem e área médica têm surgido, sobretudo na integração e implementação dos Sistemas de Navegação Cirúrgicos com os sistemas PACS das unidades hospitalares. Com os avanços médicos, os Sistemas de Navegação Assistidos por computador que utilizam tecnologias de RA experimentaram um enorme desenvolvimento na área cirúrgica assim como em procedimentos pré-operatórios para diagnóstico de pacientes em especialidades como oncologia, urologia, ortopedia, odontologia e neurologia. Estas tecnologias fornecem aos profissionais uma maior capacidade de compreensão espacial através da perceção de profundidade e do *feedback* tátil.

Como Badiali et al. descrevem em [34] e Hallet et al. em [35], a maioria destes sistemas são compostos por um dispositivo ótico, o rastreador do sistema de navegação, marcadores físicos e um *software* para construção de modelos tridimensionais através do processamento de imagens médicas. O dispositivo ótico é responsável por capturar a imagem do ambiente real que envolve o paciente e os respetivos marcadores para calcular o posicionamento relativo do dispositivo rastreador no processo de calibração. Os sistemas descritos calculam a correspondência entre os pontos virtuais definidos sobre modelo 3D gerado e os pontos reais definidos sobre o paciente. Através do dispositivo ótico o cirurgião consegue, durante a intervenção, ter a perceção tridimensional entre a cena real e conteúdo gráfico resultante da fase de planeamento virtual. Desta forma é possível aos cirurgiões alcançarem tecidos alvo sem a probabilidade e a propensão de danificar os normais (figura 2.4). A solução descrita permite

aproximar e visualizar tumores escondidos por outros órgãos ou detetar conjuntos de tecidos e vasos camuflados sob tecido adiposo.

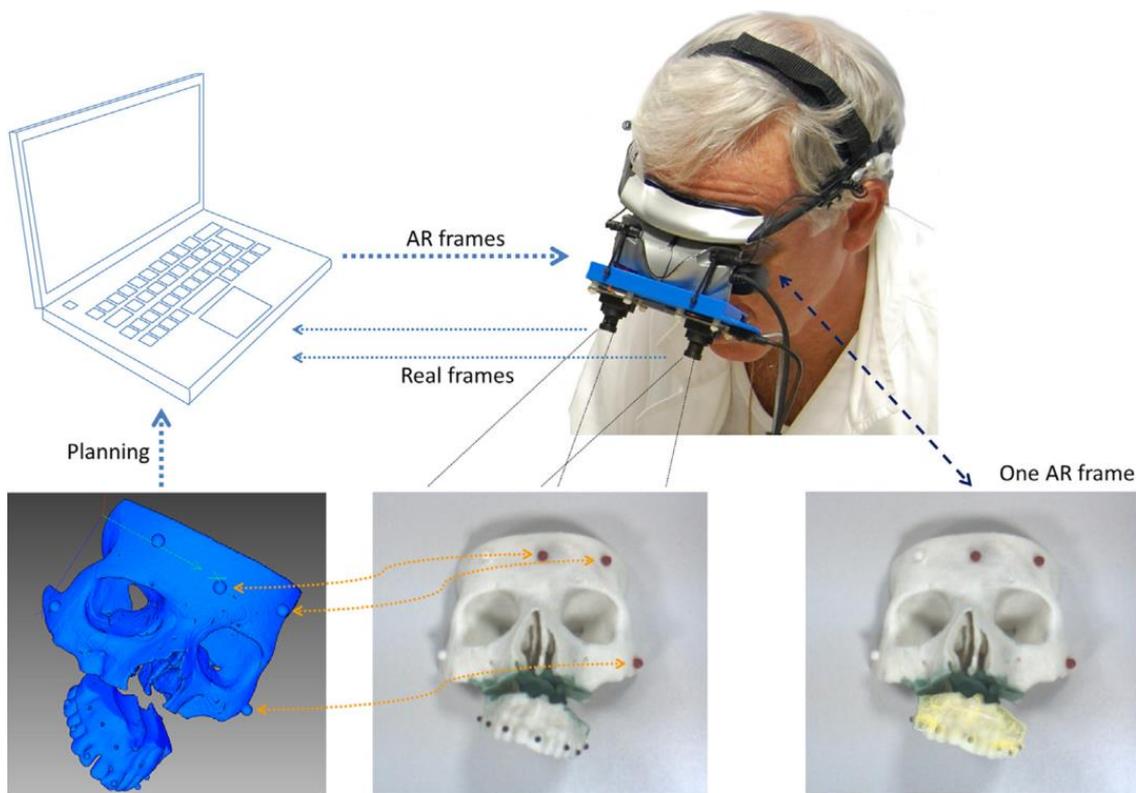


Figura 2.4 - O *software* WARM combina o modelo 3D virtual derivado durante o planeamento cirúrgico com imagens reais captadas pela câmara [34].

Segundo Conrad *et al.* e Sun *et al.*, a utilização da RA num contexto cirúrgico passa, não só, pela sobreposição das imagens reais e virtuais, mas também pelo desenvolvimento de variações nas características da imagem como a cor, transparência ou tipos de linha que permitam ao utilizador conhecer e perceber mais rapidamente o caso clínico [36, 37]. Como exemplo, é utilizada a visualização de um tecido inflamado num monitor de rastreamento, onde a área lesada aparece destacada com a cor vermelha.

Cabrilo et al em propuseram abordagem com tecnologias de RA auxiliar na recolha de vasos para doações doadores através de sua localização precisa e em tempo real. A configuração apresentada pretendia também auxiliar a realização de craniotomias adaptadas ao paciente e na identificação pré-operatória de vasos recetores selecionados para implementar na anastomose - comunicação entre os vasos. A caracterização das carótidas externas e internas dos pacientes é realizada através da segmentação manual de imagens tridimensionais obtidas por angiografia por tomografia computadorizada ou por ressonância magnética e posteriormente é executada a sobreposição das

mesmas sobre a superfície da cabeça do paciente. O *software* utilizado para a segmentação de imagens foi o *BrainLAB iPlan plataform* (ver referência [17]) e o dispositivo de imersão foi um microscópio digital. A calibração espacial, de foco e de *zoom* do microscópio é realizada através do sistema de neuro-navegação com a sobreposição de uma estrela de referência. Com a imobilização da cabeça do paciente é realizada a estereoscopia de imagens em pontos distintos que irá permitir obter informações tridimensionais do espaço e assim confirmar a precisão da sobreposição de imagens (virtual e real). Uma vez confirmada é injetada a imagem segmentada do doador sobre o recetor recorrendo ao ponto de anastomose definido para orientação. A abordagem apresentada é considerada uma solução apta à identificação correta dos vasos pretendidos e é capaz de reduzir a propensão à falha por parte do cirurgião [38].

Uma aplicação móvel de RA para localização de tumores em gânglios basais é descrita por Hou et al. em "*iPhone-Assisted Augmented Reality Localization of Basal Ganglia Hypertensive Hematoma*". Recorrendo a exames de ressonância magnética e através de um *software* de segmentação de imagem, é gerada uma representação tridimensional do tumor. De seguida, após o cálculo do seu centroide, é produzida também computacionalmente a superfície da pele do paciente através de métodos de renderização. Através da câmara do dispositivo móvel e do giroscópio integrado no mesmo, o cirurgião sobrepõe manualmente o modelo de tumor sobre a imagem captada do paciente de perfil. Este processo é realizado através da correspondência entre a imagem real e a linha de contorno do paciente, obtida pelo *software* de renderização. De seguida, a aplicação permite a elaboração de um esboço das características e linhas de incisão a serem realizadas sobre o paciente (figura 2.5). É neste momento que o cirurgião é capaz de executar e visualizar o processo de planeamento pré cirúrgico. Embora seja uma solução económica, o sistema apresentado requer um grande tempo para o processo de calibração e apresenta uma enorme propensão à imprecisão manual do profissional. [39].

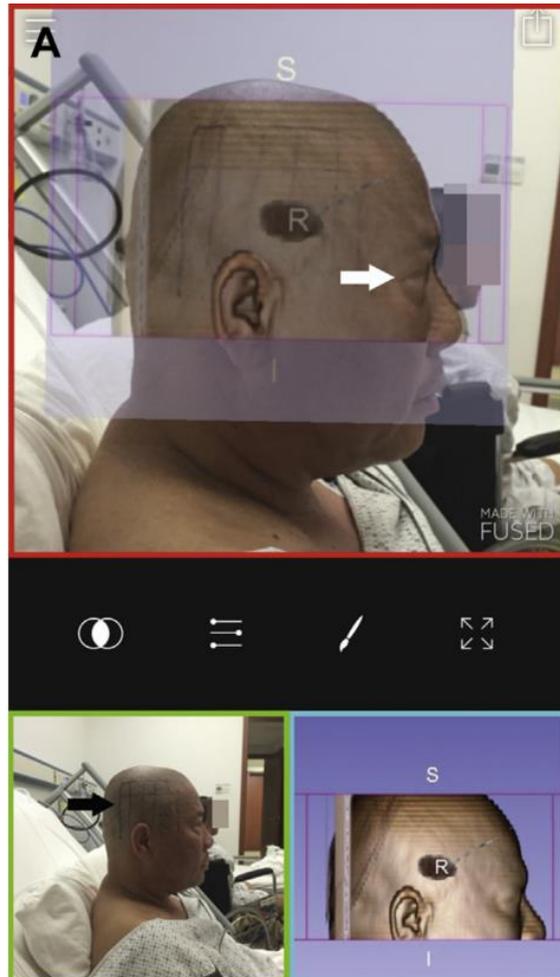


Figura 2.5– Co registo entre o modelo tridimensional e a fotografia sagital do paciente. Em A é possível visualizar a interface do utilizador da aplicação FUSED iOS [39]. A imagem de fundo no painel verde é a fotografia sagital do paciente. A imagem do primeiro plano no painel azul é o modelo tridimensional da cabeça. No painel vermelho, as 2 imagens são fundidas de acordo com os marcadores anatómicos da orelha e a localização do hematoma é projetada na imersão dos dois conteúdos – RA.

Alguns dos problemas dos sistemas referidos anteriormente são a qualidade dos gráficos, a sua capacidade de processamento, a durabilidade e preço do *software* e a impossibilidade de rastrear dinamicamente os objetos/estruturas pretendidas [3, 40]. Outras limitações apontadas são as pequenas, mas existentes imprecisões que existem na reconstrução dos modelos e no seu alinhamento com o cenário real desejado assim como pequenos atrasos temporais na sobreposição de imagens e calibração entre os dois espaços (virtual e real). A indeformabilidade e inadaptabilidade em tempo real dos modelos computacionais são também aspetos apontados como melhoria a este tipo de soluções [36, 41].

Concluindo, a utilização de aplicações móveis de AR consiste uma temática atual e de grande investimento seja na área científica, social ou económica. No entanto, o desenvolvimento e implementação deste tipo de aplicações em contexto médico de forma viável e consistente só se verifica ainda como meio educacional.

3. ARPEEK

3.1 Introdução

A solução ARPEEK foi projetada e desenvolvida de forma a melhorar a interação e experiência entre os profissionais de cirurgia ortopédica e os seus pacientes. Esta proporciona ao primeiro uma ferramenta que possibilita a realização de um diagnóstico médico, elaboração de um estudo pré-operatório completo de diretrizes para a cirurgia ortopédica e da simulação de implantes e de ferramentas de intervenção. Além disso, a ARPEEK combina o potencial das tecnologias RA, juntamente com as tecnologias móveis e web para o acesso remoto à informação. Desta forma, para além das funcionalidades acima descritas, este projeto permite a passagem, visualização e perceção de todas as informações e conteúdos gráficos resultantes da fase pré-operatória, para e durante a própria intervenção cirúrgica. Assim, a ARPEEK centra-se numa solução para a maioria dos problemas de usabilidade identificados até o momento nas tecnologias de planeamento e intervenção cirúrgica ortopédicas.

Esta solução tem como objetivo ser posteriormente implementada numa versão do *software* de planeamento cirúrgico ortopédico existente e desenvolvido pela *Peek Health S.A* [4] – *PeekMed* (descrita em 3.3.4 *PeekMed – Software de Planeamento Cirúrgico*).

3.2 Metodologia de Investigação

No desenvolvimento deste projeto, a metodologia adotada foi a da Ação / Investigação devido ao propósito do próprio trabalho, ou seja, a procura de uma solução viável para problemas concretos e reais que existem na área do planeamento e intervenção cirúrgica ortopédica.

A sistemática de Ação Investigação é utilizada em situações reais sendo que o seu principal objetivo é a resolução de problemas reais. Esta metodologia pode também ser empregue em pesquisas primárias de modo a esclarecer e simplificar contextos ou situações ambíguas. São então considerados diversos princípios como o envolvimento de diversas pessoas na pesquisa, a postura crítica e construtiva das mesmas, a procura pela novidade, o planeamento e a maior flexibilidade ou adaptabilidade perante novos contextos e subproblemas que surjam do inicial [42]. Como tal, esta foi a que, no entender dos autores desta dissertação, seria mais adequada para a elaboração deste trabalho.

Inicialmente foi feito um reconhecimento e identificação do contexto de desenvolvimento do projeto e traçou-se um plano inicial para o mesmo. Foram ainda identificados quais os requisitos, limitações e características dos utilizadores da solução, quer diretos ou indiretos, neste caso cirurgiões e pacientes, respetivamente. Após estudo das soluções relacionadas com as temáticas abordadas no trabalho, foi possível elaborar um estado de arte e fazer um levantamento das tecnologias utilizadas, quais os pontos fortes, quais as falhas e, para alguns casos, qual a sua aceitação *feedback* por parte dos utilizadores.

Desta forma, foi possível definir quais as tecnologias a serem utilizadas no desenvolvimento da solução mediante a integração de novas ideias ou do surgimento de certos obstáculos que necessitavam de uma nova abordagem e/ou resolução.

Na última fase e com base na arquitetura projetada para o sistema, foi desenvolvido um protótipo da aplicação móvel de RA, um servidor web de modo a simular a sua integração com o *software* da *Peek Health S.A* (Ribeiro et al., 2015) e foi elaborado um modelo de dados e armazenamento de informação que suportasse os requisitos funcionais e lógicos dos mesmos.

A metodologia definida seguiu então o padrão:

- Exploração e definição do problema assim como das suas características e objetivos funcionais a serem alcançados;
- Elaboração e constante atualização do estado de arte, levantamento das tecnologias de desenvolvimento existentes assim como a inserção de novos objetivos no projeto;
- Estudo das diferentes tecnologias e abordagens recolhidas de modo a selecionar as que cumprissem com os requisitos definidos para a solução;
- Conceptualização e estruturação da arquitetura do sistema a desenvolver assim como da estrutura do modelo de dados por trás da mesma;
- Elaboração dos protótipos da solução inseridos na arquitetura projetada;
- Realização de alguns testes funcionais à solução proposta e aos respetivos protótipos;
- Análise dos resultados, levantamento de novos requisitos, novas funcionalidades, das alterações a serem efetuadas ao modelo de dados e, conseqüentemente, renovação dos protótipos;
- Elaboração da dissertação para a partilha do conhecimento e descrição da abordagem inovadora proposta;

3.3 Tecnologias de Desenvolvimento

De forma a desenvolver uma solução que reunisse as ferramentas necessárias para a elaboração das funcionalidades pensadas para a ARPEEK, foi necessário recorrer a bibliotecas e *softwares* externos reconhecidos. Sendo que são agregadas diferentes tecnologias e conceitos, foi necessário fazer uma recolha e estudo de quais as melhores *frameworks* existentes de modo a criar uma solução que se destacasse das restantes e se comprometesse à sua usabilidade para utilização na área da cirurgia ortopédica. Para tal, foram então utilizadas as seguintes ferramentas: bibliotecas de *Open Source Computer Vision Library* (OpenCV), o *software* da *Peek Health S.A.*, a *framework PHP Laravel* em *SublimeText*, o motor multiplataforma *Unity* juntamente com a *framework MonoDevelop*, assim como linguagens de programação *C++*, *C#*, *PHP Hypertext Processor* e *Structured Query Language (SQL)* na conceção e manipulação do modelo de dados.

Os estudos de processamento de imagem foram desenvolvidos e realizados em *C++* utilizando bibliotecas OpenCV. Os modelos 3D dos tecidos do corpo humano foram construídos usando o *software Peek Health S.A* [4]. Estes são gerados importando as respetivas imagens DICOM através da extração da sua isosuperfície.

A aplicação móvel foi desenvolvida em *Unity* utilizando a linguagem *C#*. De maneira a poderem ser realizados testes à aplicação, foi desenvolvido um protótipo de um servidor web em *Laravel* utilizando *PHP*, assim como projetado e desenhado um modelo de dados povoado com alguns exemplos utilizando *SQL Querying* e o *MySQLWorkbench*. A nível de *front end* foi criada uma interface web que, ligada ao servidor, permitisse simular o procedimento de o médico se autenticar e fazer a importação dos modelos tridimensionais resultantes da renderização de imagens médicas, associados à informação dos respetivos pacientes.

3.3.1 OpenCV

O OpenCV é uma biblioteca de código aberto para *software* de *machine learning* e de visão computacional. Esta fornece uma infraestrutura comum e organizada para o desenvolvimento de aplicações em diferentes plataformas que também utilizem tecnologias de visão e perceção computacional.

Esta biblioteca disponibiliza inúmeros algoritmos agrupados em módulos, que permitem o desenvolvimento de diversas funcionalidades sendo que os de maior interesse para este trabalho foram

os de Processamento de Imagem e Vídeo I/O, os de Visão Computacional (VC), o módulo de estrutura de dados e álgebra linear (*cxcore*) e o de controlo de interface e dispositivos de entrada (*highgui*).

Apesar desta biblioteca ter sido desenvolvida em C++, o OpenCV também disponibiliza um suporte a outro tipo de linguagens de programação como o *Java*, *Python* e *Visual Basic* de modo a que os *developers* o possam integrar nas suas aplicações. Para além disso, é compatível com diversos Sistemas Operativos (SO) desde *Android*, *iOS*, *OS X*, *Windows* e *Linux*. Esta é uma das razões principais da escolha e utilização desta biblioteca para o desenvolvimento da solução, sendo que apresenta uma elevada escalabilidade [43].

O OpenCV é utilizado em diversas áreas e com diferentes propósitos desde a identificação de objetos, sistemas de reconhecimento facial, rastreamento de movimentos até à de maior interesse para este trabalho – RA.

3.3.2 *Framework* Laravel

O *Laravel* é uma *framework* de código aberto PHP para o desenvolvimento de sistemas web que utilizam o padrão de arquitetura de *software Model-View-Controller* (MVC) que permite a divisão das funcionalidades de um sistema web em camadas (figura 3.1). O objetivo principal deste tipo de arquitetura é a divisão de um problema maior em problemas mais pequenos tendo em conta a complexidade dos sistemas web desenvolvidos nos dias de hoje. Através da independência entre camadas implementada por este padrão, é possibilitado o reaproveitamento de código, facilitada a manutenção e adição de recursos e melhorada a gestão e distribuição de tarefas.

De uma forma sucinta e simples, o padrão MCV consiste nas camadas:

- Modelo (*Model*) consiste nos dados da aplicação, ou seja, nas regras de negócios, na lógica inerente à mesma, nos cálculos, consultas e restantes funcionalidades. É este que tem acesso a toda e qualquer informação proveniente de uma base de dados ou arquivo XML;
- Visão (*View*) é a forma como são representados os dados do modelo solicitado, ou seja, o tipo de *output* que estes irão ter para o utilizador (tabelas, diagramas, gráficos, etc.);
- Controlador (*Controller*) é responsável por gerir todo o fluxo de informação que passa pelo sistema. Este define quais as informações devem ser geradas, quais e quando devem ser acionadas as regras do sistema e para onde as informações devem ser encaminhada [44].

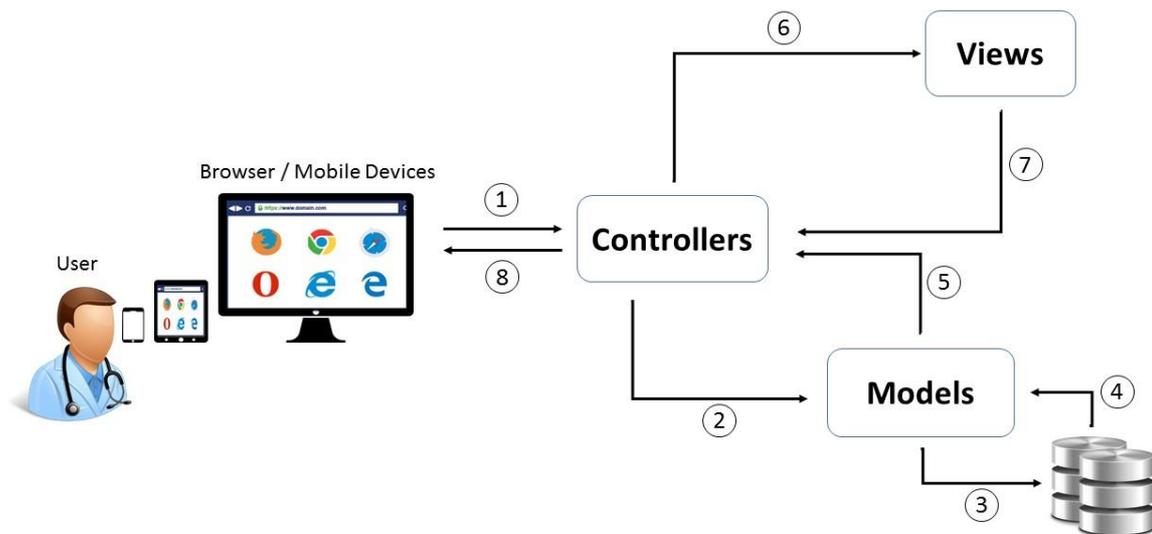


Figura 3.1 - Arquitetura e fluxo de um sistema MVS. 1) o *Browser* envia um pedido ao Controlador que é rastreado para um determinado modelo; 2) o Controlador direciona esse pedido ao Modelo específico; 3, 4) o Modelo avalia e executa a ação pretendida sobre a base de dados e recebe os dados; 5) o Modelo retorna a informação indexada ao Controlador; 6) o Controlador envia um pedido de atualização à *View* com a informação solicitada; 7) A *View* processa a nova informação e incorpora-a em HTML; 8) O controlador retorna o HTML devolvido pela *View* ao *Browser*.

Para além de ser uma ferramenta que utiliza uma sintaxe simples, o *Laravel* apresenta outras características aliciantes como a existência de um administrador de dependências que permite a gestão de pacotes externos à aplicação. Este sistema modular permite carregar todas as bibliotecas extras necessárias e geri-las num só pacote, o que facilita bastante a implementação e escalabilidade das soluções desenvolvidas nesta *framework*. Outra característica interessante é o conceito de 'rotas' que permite um mapeamento simples e intuitivo de URLs dos pedidos da aplicação. Esta apresenta ainda uma enorme abstração no acesso a base de dados e noutras utilidades que auxiliam o desenvolvimento e manutenção de uma aplicação web [45].

3.3.3 *Unity & MonoDevelop*

O *Unity* é um *Software Development Kit* (SDK) criado pela *Unity Technologies* utilizado para o desenvolvimento de videojogos, simuladores e aplicações em dispositivos móveis, consolas, desktops e web. Para além destas funcionalidades, o *Unity* destaca-se como o SDK pioneiro no desenvolvimento de aplicações de RA e Realidade Virtual (RV). Apesar de inicialmente, no ano de 2005, ter sido lançado apenas para o OSX, é agora uma plataforma que corre em mais de 20 sistemas operativos entre eles *Windows*, *Android* e *iOS* (*Unity Technologies* 2016).

As principais características que se destacam neste SDK é a facilidade, fluxo e rapidez de desenvolvimento que fornece aos seus utilizadores, para além do vasto suporte em multiplataforma. Esta agilidade operacional deve-se:

- À facilidade de importar, gerir e implementar extensões e bibliotecas externas à aplicação;
- Ao editor integrado que fornece uma enorme abstração a nível código;
- Ao excelente suporte à depuração, ou seja, na facilidade do processo de localização e redução dos erros da aplicação;
- À vasta e completa biblioteca que permite a integração e manipulação de imagem, som e outros dispositivos através de controladores, assim como funcionalidades de processamento de imagem eventos físicos e renderização;
- À utilização do *Integrated Development Environment IDE MonoDevelop* que, para além de permitir a desenvolvimento em linguagem C# no seu editor, traz ao utilizador diversos níveis de abstração desde o compilador de código integrado até à modelagem, distribuição e automatização de testes ao *software*.

O *Unity* é igualmente adequado para criar aplicações e/ou videojogos 2D e 3D. Para a solução proposta neste trabalho no que toca ao desenvolvimento de uma aplicação móvel de RA, o modo utilizado para o SDK foi o Full3D que permite, como que, uma visão pessoal e imersiva da cena pelo utilizador. Este modo disponibiliza funções específicas no editor e determina o modo de como as imagens são importadas, neste caso, enquanto 'texturas'. Adicionalmente são estabelecidos outros parâmetros como o posicionamento da câmara no espaço e a orientação, localização, iluminação, sombras e tempo dos objetos no espaço. O modo 3D recorre a geometria tridimensional com materiais e texturas renderizados na superfície dos objetos, de forma a que estes sejam representados de uma maneira sólida e projetados no ambiente desejado que, neste caso, será a imagem vídeo captada pela câmara do dispositivo móvel.

Na sua utilização para o desenvolvimento da solução proposta neste trabalho, o *Unity* foi escolhido devido ao facto de ser um SDK pioneiro e com uma documentação bastante completa no que toca à sua implementação em tecnologias AR. Para além disso, esta permitia a integração de uma biblioteca OpenCV no mesmo, apesar de não ter sido utilizada a versão profissional, tanto do *Unity* como do *OpencVForUnity*. Ao realizar esta integração foi possível reunir funcionalidades de processamento de imagem e visão computacional num *software* de desenvolvimento simplificado para aplicações móveis, utilizando as diversas funcionalidades e bibliotecas disponibilizadas por cada um. Esta

abordagem permitiu então o desenvolvimento do protótipo da aplicação móvel apresentado neste trabalho e descrito no respetivo capítulo.

3.3.4 *PeekMed* – Software de Planeamento Cirúrgico

PeekMed é um sistema de planeamento cirúrgico ortopédico que proporciona informação mais precisa, dinâmica e visual aos cirurgiões e às suas equipas.

Com recurso às suas mais variadas ferramentas digitais, este permite a simulação de vários procedimentos cirúrgicos ortopédicos como osteotomias conhecendo a sua localização exata e dimensão da cunha, redução de fraturas, segmentação dos ossos do paciente e respetivas medições detalhadas entre outras que podem ser consultadas em [46]. Para além do ambiente de trabalho e desenvolvimento poder ser manipulado em 2D e/ou 3D, é ainda possível recorrer e adicionar, ao mesmo, diversos modelos de próteses e/ou implantes de fabricantes.

Recorrendo a uma TC ou a um RM, praticamente todos os tipos de procedimentos cirúrgicos ortopédicos podem ser planeados e revistos virtualmente em 2D e 3D ou simultaneamente. Toda a aplicação assenta num *workflow* simples baseado em quatro etapas: importação de imagens, manipulação e interação do ambiente de trabalho 2D / 3D recorrendo à sua barra de ferramentas e edição de objetos, inserção de implantes e obtenção de um relatório final. Este relatório digital pode ser consultado em qualquer dispositivo móvel, a qualquer hora, em qualquer lugar e sobre o qual podem ainda ser efetuadas alterações/correções quando necessário, sendo que é também permitida a conexão e integração do ambiente de trabalho e da aplicação móvel com o sistema PACS local ou com a *cloud* do sistema.

Como é referido em tópicos anteriores, a implementação deste tipo de tecnologias traz muitas vantagens aos seus utilizadores. Para além de diminuir o tempo de planeamento de toda a cirurgia e de permitir uma melhor compreensão das suas etapas, quais os implantes a serem utilizados e da conseqüente redução dos custos de esterilização, o ambiente 3D *PeekMed* permite antecipar e prever algumas complicações intra e pós cirúrgicas. Com todo o detalhe, mas simplicidade dos processos, o sistema facilita a comunicação entre os elementos da equipa cirúrgica assim como a gestão de recursos (gestão de inventários e *stocks*). Outra característica interessante desta solução é a proximidade que cria entre o trabalho dos médicos e o paciente, uma vez que este pode também conhecer todo o processo ao qual será submetido.

3.4 Levantamento e análise de Requisitos

Para o desenvolvimento de soluções tecnológicas focadas no utilizador/cliente e para a sua viabilização é de enorme importância elaborar uma lista de requisitos. Este processo pode ser repetido e alterado ao longo de diversas etapas de elaboração do projeto. É nesta fase que devem ser compreendidos os domínios e reunidos os requisitos da aplicação junto dos *stackholders* do sistema. Assim, é possível resolver conflitos, definir prioridades e desenvolver uma solução coerente e coesa. Primeiramente, neste caso em específico, é importante definir e conhecer três elementos fundamentais da solução: o(s) utilizador(es) /cliente(s) alvo(s), o(s) ambiente(s) em que a solução será integrada e as informações que irão estar associadas à mesma.

Sendo a *ARPeek* uma solução de planeamento cirúrgico ortopédico esta terá, pelo menos dois ambientes onde poderá ser utilizada: a sala de intervenção cirúrgica e o consultório médico ou a sala de planeamento em equipa. Este tipo de ambientes deve ser o mais simples, espaçoso, organizado, de fácil manutenção e de máxima liberdade para os seus intervenientes. Simultaneamente, todos os materiais, ferramentas e informação devem estar ao mais rápido e intuitivo alcance. A *ARPeek* pretende ir mais longe, não limitando os seus utilizadores ao seu ambiente físico de trabalho, mas sim a qualquer lugar e a qualquer hora. Sendo uma aplicação móvel que corre em qualquer dispositivo móvel individual, qualquer ambiente menos formal pode tornar-se no local de estudo e revisão de um processo cirúrgico. Associando esta característica à evolução das tecnologias de comunicação, pretende-se que em qualquer altura poderão ser revistas e discutidas em equipa as *guidelines* para o calendário cirúrgico.

Os utilizadores alvo do *ARPeek* são os cirurgiões ortopédicos e as suas equipas de intervenção. Recorrendo e traduzindo a definição da *American Society of Anesthesiology* uma “*equipa cirúrgica é uma unidade que providencia toda a continuidade de cuidados que se inicia no período pré-operatório, se desenvolve no intraoperatório durante a realização dos procedimentos cirúrgicos e se estende à recuperação pós-operatória do doente. Cada especialista da equipa, seja o cirurgião, o anestesiológista ou o enfermeiro, detém preparação avançada para o seu desempenho nos três períodos descritos*” [47]. Como tal, a solução não pode nem deve limitar-se à utilização e perceção de quem é especializado na área da ortopedia, mas sim permitir uma comunicação acessível e de fácil assimilação aos restantes especialistas da equipa cirúrgica. Ao longo dos tempos a comunidade médica tem aceite a implementação de tecnologias que, de certa forma, facilitem o seu trabalho. Mais do que isso, estes procuram soluções que aumentem as suas capacidades, conhecimento, precisão e competência. Considerando o seu contexto de trabalho, estas ferramentas e aplicações terão de ser de fácil implementação e manutenção assim como de utilização simples, intuitiva e de fácil assimilação.

Sendo o *ARPeek* uma aplicação de RA para utilização médica, o tipo de informação que esta processa e dispõe vai desde conteúdos gráficos e imagens DICOM (dados resultantes das modalidades de Imagiologia – exames médicos) até dados numéricos, literais e/ou lógicos. Para além de diversas fontes e formatos, esta também tem um caráter confidencial elevado, uma vez que os dados se referem maioritariamente a pacientes ou têm como origem os profissionais e serviços de saúde.

Deste modo, foram levantados os requisitos funcionais e não funcionais para a solução *ARPeek*, descritos de seguida. Sendo suposta a implementação da *ARPeek* na ferramenta e aplicação de planeamento de cirurgia ortopédica - *PeekMed*, é importante salientar que diversos requisitos não estão descritos, uma vez que já foram avaliados e suprimidos no desenvolvimento da aplicação inicial [4][7][46].

3.4.1 Requisitos funcionais

No que toca ao desenvolvimento de *softwares*, os requisitos funcionais definem as funções e comportamentos desse mesmo *software* ou aplicação. Ou seja, estes descrevem o que a solução permite fazer em termos de tarefas, serviços e usabilidade. Desta forma, a aplicação *ARPeek* deve, em complemento ao *PeekMed*:

- Disponibilizar ao utilizador uma ferramenta onde este, através da câmara do seu dispositivo móvel, possa visualizar em ambiente e tempo real, toda a informação gráfica que resultou do processo de planeamento cirúrgico. Tal informação consiste em:
 - O modelo 3D anatómico da estrutura óssea a sofrer intervenção;
 - A representação da patologia sobre esse mesmo modelo;
 - Os *templates* das ferramentas cirúrgicas a utilizar na intervenção;
 - Os *templates* de próteses a aplicar durante a intervenção;
- Disponibilizar uma área onde o utilizador possa optar pelo marcador fiducial que preferir;
- Fornecer ao utilizador uma área pessoal onde este possa carregar as imagens médicas referentes à intervenção e complementar os seus conteúdos, tanto com anotações pessoais e *guidelines* como com outras informações clínicas do paciente;
- Disponibilizar uma área de pesquisa onde o utilizador possa encontrar mais rapidamente e filtrar a listagem dos conteúdos gráficos resultantes da fase de planeamento e respetiva informação através de dados do paciente ou até de datas de carregamento dos dados e ficheiros;

- Permitir que o utilizador se autentifique na aplicação de modo a utilizar as suas funcionalidades e aceda à sua informação pessoal e profissional;
- Facultar o acesso e visualização de toda a informação e conteúdos remotamente, assim como a sua importação desde que esteja assegurada uma ligação web;
- Possibilitar a partilha do conteúdo gráfico com outros utilizadores de modo a permitir a troca de conhecimentos e experiências entre os profissionais;
- Disponibilizar a ferramenta de *screenshots* à imagem resultante da tecnologia de RA, de modo a serem elaborados relatórios e documentação dos procedimentos cirúrgicos;
- Permitir a manipulação dos *templates* das ferramentas cirúrgicas e próteses virtuais por *touchscreen input*, de maneira a modificar a sua disposição sobre o modelo computacional representativo da zona óssea da patologia, assim como gravar o estado final do planeamento interventivo após esta manipulação. Possibilitando assim:
 - a simulação do uso dessas mesmas ferramentas na intervenção;
 - a previsão e simulação da disposição - orientação e localização – das próteses assim como dos seus eventuais ajustes;
 - a medição de distâncias, ângulos, entre outros parâmetros técnicos importantes à intervenção.

É importante reforçar que, devido à integração desta aplicação com o, já existente, *software PeekMed* é possibilitada a renderização dos modelos computacionais recorrendo às imagens médicas em formato DICOM.

3.4.2 Requisitos não funcionais

Os requisitos não funcionais, no desenvolvimento de aplicações e/ou *softwares* referem-se a questões de usabilidade, desempenho, eficiência, segurança e manutenção/atualização da(s) tecnologia(s) em questão. Estes refletem a qualidade e, de certa forma, o preciosismo com que é disponibilizada a solução ao cliente/utilizador.

Assim, os requisitos não funcionais levantados para este projeto foram:

- A aplicação deverá correr nos principais sistemas operativos associados a plataformas móveis;
- A aplicação deve apresentar uma interface simples e *user-friendly*;
- A aplicação deverá permitir a integração com o sistema PACS local e permitir o carregamento de informação a partir da mesma, com as devidas restrições de acesso e autenticações;

- A aplicação deve também poder aceder aos ficheiros pessoais guardados em memória no dispositivo móvel sobre a qual irá ser instalada;
- Deverá ser garantida a abstração máxima ao utilizador de todos os processos envolvidos na aplicação. Isto é, o processo que permite a relação entre o reconhecimento da zona de interesse/alvo e a consequente sobreposição de conteúdos gráficos sobre o ambiente real deve ser o mais descomplicado e intuitivo possível. Desta forma, no ambiente de estudo e/ou intervenção o utilizador só irá necessitar de:
 - Um dispositivo móvel que corra a aplicação de RA;
 - Um marcador fiducial de fácil impressão que permita a calibração e deteção do ambiente real ao *software*;
- A aplicação deverá fornecer ao utilizador o conteúdo gráfico com o detalhe e precisão máxima, fazendo por isso uso das tecnologias mais avançadas;
- Como tal, deve ser fomentada a sua constante evolução e atualização assim como o desenvolvimento de novas funcionalidades e levantamento de necessidades junto do utilizador;
- A aplicação deve garantir a privacidade e segurança de todos os dados;
- A aplicação e o seu sistema deverão ter uma elevada disponibilidade;
- Mediante a sua aquisição, a aplicação deverá apresentar uma eficiência máxima, processando vários pedidos num mínimo tempo possível tendo assim uma elevada capacidade de resposta;
- Por fim, a solução deverá ser utilizada de maneira consciente e dentro dos termos médico-legais permitidos.

3.5 Implementação e Conceptualização da tecnologia

Como já foi referido anteriormente, a ARPeek (figura 3.2) foi desenvolvida como ferramenta complementar à solução já existente de planeamento ortopédico – *PeekMed* – mas a mesma pode ser válida e implementada em outras situações uma vez que é independente da informação utilizada e da sua origem, sendo que existindo um *software* para renderização de modelos tridimensionais através de ficheiros DICOM, a aplicação e abordagem são viáveis. O sistema da solução proposta divide-se em 6 etapas principais (representadas pelas letras na figura 3.2) e abordadas nos seguintes subcapítulos.

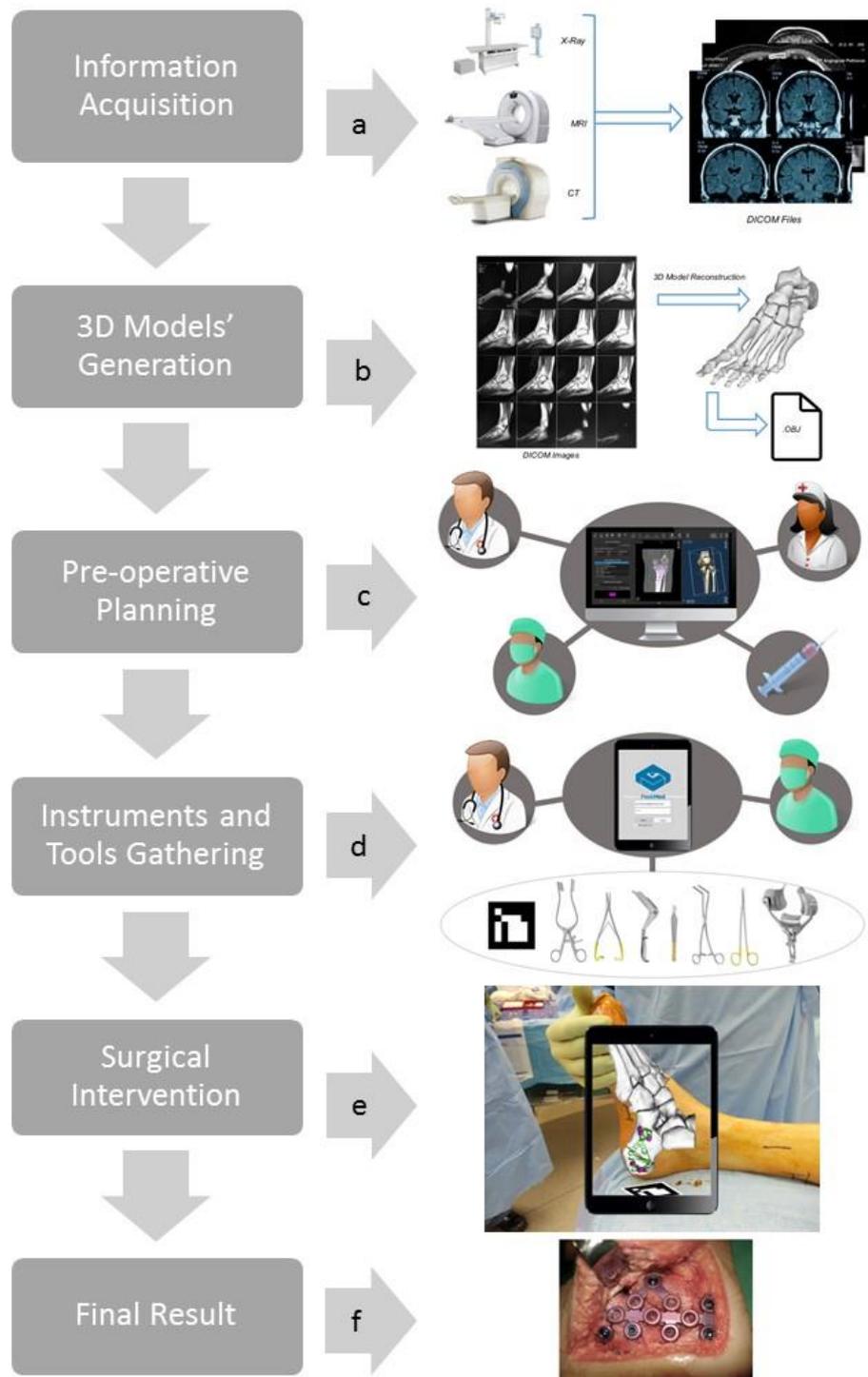


Figura 3.2- Etapas à implementação da aplicação *ARPeek* no planeamento e intervenção cirúrgica ortopédica.

3.5.1 Aquisição da Imagem Médica e Renderização dos Modelos Anatômicos

O fluxo da solução inicia-se quando o cirurgião requisita exames das modalidades de Imagem Médica de modo a poder analisar ou diagnosticar qual a patologia do paciente. Esta fase é operada por

profissionais especializados na tecnologia e nos aparelhos de tomografia computadorizada e/ou de ressonância magnética.

No ramo da ortopedia os exames TC são mais comuns uma vez que, para além de serem de mais rápida realização, permitem concretizar avaliações em planos axiais sem superposição de estruturas. Geralmente os exames de RM são considerados como complementares aos de TC uma vez que estes permitem caracterizar tipos específicos de tecidos baseados na intensidade de sinal (gordura, sangue e água), diferenciando outras patologias que, provavelmente, estarão fora da área da ortopedia.

Destacando então as imagens obtidas pela TC, estas são posteriormente guardadas em formato DICOM no Sistema de Armazenamento e Comunicação de Imagens local (PACS) – figura 3.2.a.

Um ficheiro DICOM é composto por duas partes: a parte da informação referente aos pixéis/voxéis das imagens e um cabeçalho onde estão descritos todos os metadados associados ao ficheiro. A última é organizada por etiquetas (*tags*) com todo o volume de informação para cada imagem (imagens planares resultantes da TC denominadas usualmente por fatias - *CT scan's slices*). Os metadados referem-se, não só a informação relativa ao paciente, mas também às respetivas séries de imagens que compõem o ficheiro.

É nesta fase que a aplicação do *software PeekMed* surge na solução. Após a aquisição, registo e *login* na aplicação é então possível importar os ficheiros DICOM de interesse de modo a iniciar o estudo e planeamento pré-operatório gerando o modelo 3D da estrutura em análise (figura 3.2.b). Por conveniência e segurança esses ficheiros podem ser carregados acedendo diretamente ao sistema PACS a que o médico está ligado, podendo também o mesmo os importar previamente para a sua máquina pessoal ou de trabalho. O *PeekMed* permite ainda nesta fase inicial guardar os modelos em diversos formatos sendo que o mais conveniente para os estudos do protótipo foi o *'obj'*.

A construção de modelos tridimensionais computadorizados através de um varrimento de TC recorre a algumas técnicas de renderização, sendo uma delas denominada por Renderização de Superfície (*Surface Rendering* – SR). Esta é definida como a visualização de um objeto 3D através de um conjunto de isosuperfícies. As isosuperfícies são geradas com pontos que apresentam a mesma intensidade de sinal. Neste caso, estes valores referem-se aos de atenuação da radiação e são definidos pela escala de *Hounsfield*.

Existem duas abordagens para a elaboração de isosuperfícies: uma utilizando contornos extraídos de cada *slice* de modo a criar uma superfície baseada em volume, ou então utilizando voxéis (neologismo com origem na junção das palavras "volume" e "pixel") com um valor predefinido da escala de Hounsfield. As imagens digitais são geralmente representadas por uma matriz bidimensional de

valores (*bit map*), onde cada elemento dessa matriz representa a intensidade de um pixel. Se considerarmos uma imagem como tendo volume ou profundidade, é então necessário adotar uma representação tridimensional, neste caso um elemento que represente volume recorrendo a outras propriedades – voxel [48] (figura 3.3). O valor de um voxel pode representar várias propriedades. Como referido anteriormente, para este caso os seus valores são representados em unidades *Hounsfield* que caracterizam a opacidade do material aos raios-X, ou seja, a impedância deste material à passagem deste tipo de radiação.

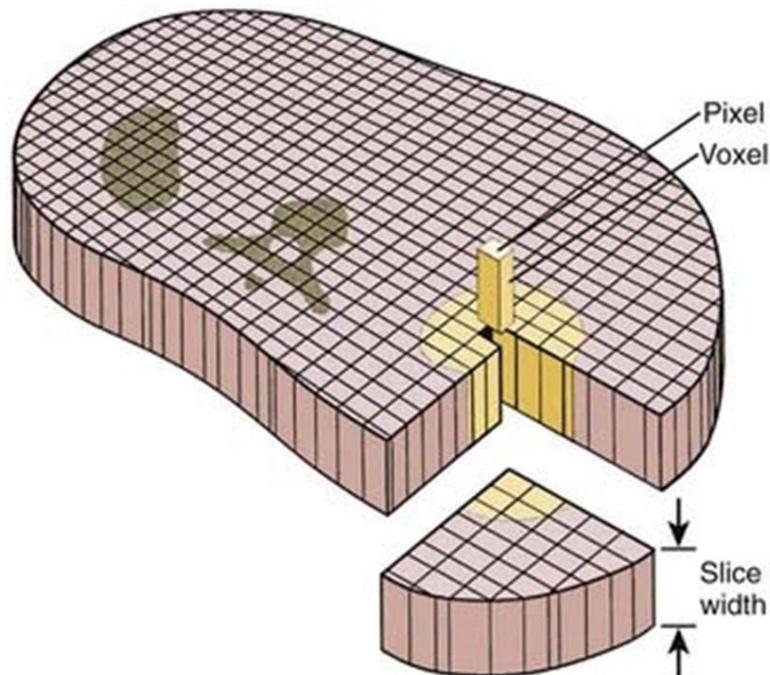


Figura 3.3 - Pixel e Voxel. O valor de cada pixel numa imagem computacional representa o coeficiente de atenuação linear de um elemento com volume de um prisma quadrangular – o voxel [49].

A reconstrução de modelos tridimensionais através de imagens DICOM pode ser realizada por múltiplos algoritmos sendo um deles o *Marching Cubes* (MC), algoritmo este implementado no *software PeekMed*. Como o próprio nome indica, este algoritmo funciona como um “cubo marchante” (figura 3.4.a) sobre o campo escalar representativo da imagem que vai determinar qual o polígono (8 vértices, logo $2^8 = 256$ possibilidades) que melhor se ajusta e representa a isosuperfície que o interceta recorrendo a uma tabela de consulta (figura 3.4.b). No final, os polígonos projetados são fundidos resultando a superfície final [50].

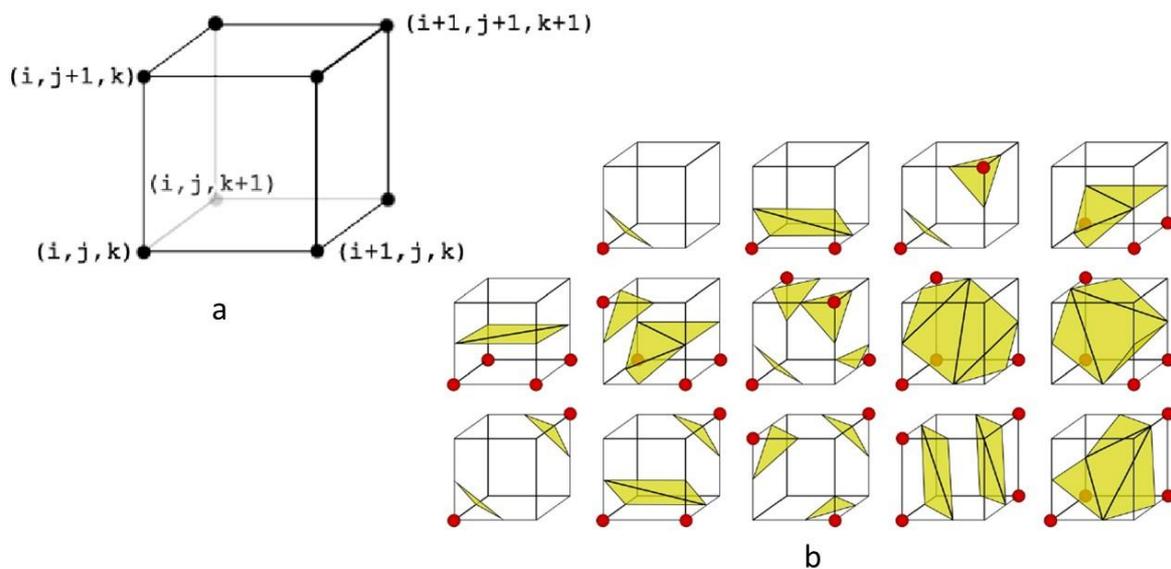


Figura 3.4 - Algoritmo *Marching Cubes*. a) célula utilizada no varrimento do algoritmo; b) tabela *look up* utilizada pelo algoritmo MC [51].

Toda a informação necessária para a implementação do algoritmo MC, como espessuras e posicionamentos, podem ser obtidas através da informação alocada às *tags* dos metadados dos ficheiros DICOM. Servem de exemplo a espessura da imagem planar ou do corte (0018,0050), a localização do corte (0020,1041), o número de cortes (0054,0081) e o espaço entre eles (0018,0088). Para mais informações, consultar Ribeiro, J., Alves, V., Silva, S., & Campos, J. (2015).

3.5.2 O Planeamento Cirúrgico Ortopédico

Após criação dos modelos computacionais surge a fase de planeamento propriamente dita. Juntamente com a sua equipa, o cirurgião pode visualizar o conteúdo gráfico e modificar a sua perspetiva de visualização permitindo-lhe(s) uma melhor perceção da lesão/patologia. É nesta fase que o médico pode também reunir e associar ao plano outro tipo de informações necessárias como dados pessoais do paciente, resultados de análises clínicas ou até outros problemas que este apresente que possam trazer complicações à intervenção. É aqui que tanto os anestesistas como os enfermeiros podem ser informados dos objetivos da cirurgia e das normas da segurança a seguir, além de poderem dar o seu contributo e orientação aos restantes elementos de modo a melhorar o plano interventivo.

Como referido anteriormente, o *software PeekMed*, possibilita também importar para o ambiente de desenvolvimento diversos “*templates* de fábrica” junto de alguns parceiros de ferramentas e próteses cirúrgicas como placas, parafusos ou até mesmo ortóteses. Estes modelos são dispostos sobre a

representação gráfica da estrutura em análise do paciente com as dimensões e proporções que teriam em ambiente real. Com esta função é possível manipular e simular a aplicação destes mesmos *templates* avaliando a sua disposição, dimensão, localização ou até mesmo utilidade para a cirurgia a ser realizada.

Este passo irá permitir ainda à equipa e ao cirurgião melhorar certos aspetos como o tempo da intervenção e preparação do bloco operatório, uma vez que são conhecidos *à priori* todos os materiais necessários e procedimentos, podendo gerir e destacar diferentes equipas para cada função e rentabilizar ao máximo todo o processo em paralelo com outras intervenções (figura 3.2.c).

Um aspeto importante a ser referido é que é neste passo que o acesso remoto a toda esta informação e aos modelos renderizados é garantido. Este tipo de abstração permite ao utilizador aceder aos dados e realizar todo o planeamento cirúrgico em qualquer lado, a qualquer momento, tanto no local de trabalho como em casa. Todas as alterações realizadas sobre o ambiente de desenvolvimento e planeamento podem ser guardadas e recomeçadas e o acesso à informação torna-se independente do local e do dispositivo em que a aplicação é executada.

3.5.3 Utilização da aplicação de RA

Para este passo da solução proposta, todo o material que é necessário, para além do previsto na fase de planeamento, é um dispositivo móvel com câmara a correr a aplicação e o marcador fiducial impresso. Este será fundamental para a implementação e funcionamento da função AR durante a fase pré-operatória ou de procedimento cirúrgico inclusive (figura 3.2.d).

É neste ponto que algumas das vantagens do *ARPeek* se começam a destacar e emergir das soluções de RA atuais. O facto de não existir a necessidade de realizar uma preparação prévia do bloco operatório, da configuração dos dispositivos e da gestão de espaço alivia toda a equipa das eventuais tarefas e fases de aprendizagem que a implementação desta tecnologia implicaria. Um médico auxiliar ou o próprio cirurgião podem segurar o dispositivo e, apontando-o para o marcador, é possível visualizar em tempo real o conteúdo gráfico resultante da fase de planeamento. A imersão do conteúdo computacional sobre a imagem real do paciente irá fornecer à equipa uma noção de profundidade muito mais ampla sem que estes percam noção do espaço que os rodeia. Este tipo de visualização imersiva poderá facilitar a monitorização da intervenção no que toca a cortes que sejam necessários realizar, disposição de próteses, medições ou simplesmente aumentando a precisão de todo o trabalho da equipa cirúrgica.

O marcador irá permitir ao *software* estimar a posição relativa do dispositivo e todo o conteúdo gráfico será redimensionado e projetado sobre o mesmo, como se este fosse a origem do espaço. De

salientar que este tipo de visualização sobre os modelos não é estático, ou seja, os conteúdos gráficos projetados no ecrã do dispositivo apesar de não alterarem a sua localização e orientação relativamente ao marcador, variam de dimensões de modo a dar a ideia de proximidade, rotatividade e distância aos mesmos. Por isto entenda-se que o médico pode aproximar-se, afastar-se, deslocar-se lateralmente ou até mudar a direção onde aponta a câmara do dispositivo que, enquanto o marcador for detetável, os modelos 3D serão visíveis e estarão como que presentes e visíveis no ambiente real. O objetivo principal e o interesse desta abordagem tecnológica é que a câmara do dispositivo irá funcionar como o olho do cirurgião, mas capaz de ver além dos tecidos reais (pele e outros órgãos) que escondem a estrutura óssea do paciente.

Nesta fase seria também possível ao cirurgião importar para o ambiente de RA, outros *templates* de próteses e utensílios médicos assim como interagir com estes. Através de um *input* por toque no ecrã, este poderia variar a sua orientação e posição de fixação no modelo da estrutura óssea do paciente, mas ao mesmo tempo visualizar as alterações sobre a imagem real do paciente. Adicionalmente seriam implementadas outras funcionalidades como a possibilidade de realizar *screenshots*, ativar ou desativar a visualização de referenciais auxiliares ou ainda exportar relatórios sobre o procedimento a ser realizado.

3.5.4 Resultado Final

A abordagem apresentada irá permitir ao profissional interagir com o conteúdo gráfico de maneira a explorar, extrair e analisar o mais pequeno detalhe numa experiência imersiva (figura 3.2.e e figura 3.5). Como é possível ver pela figura e figura 3.2.f e 3.5.b, o resultado final da cirurgia pode ser previamente avaliado. Isto irá permitir que a equipa possa verificar e corrigir ligeiras imperfeições e desvios, sugerir novas abordagens, aumentar a sua eficiência e precisão e monitorizar todos os passos da intervenção muito mais facilmente. Por fim, esta solução de planeamento e auxílio à cirúrgica ortopédica poderá contribuir para que as complicações durante a intervenção sejam mais previsíveis e contornáveis, reduzir o tempo da mesma, antecipar o seu resultado final, eventualmente melhorar os seus resultados e, conseqüentemente, a fase de recobro do paciente.

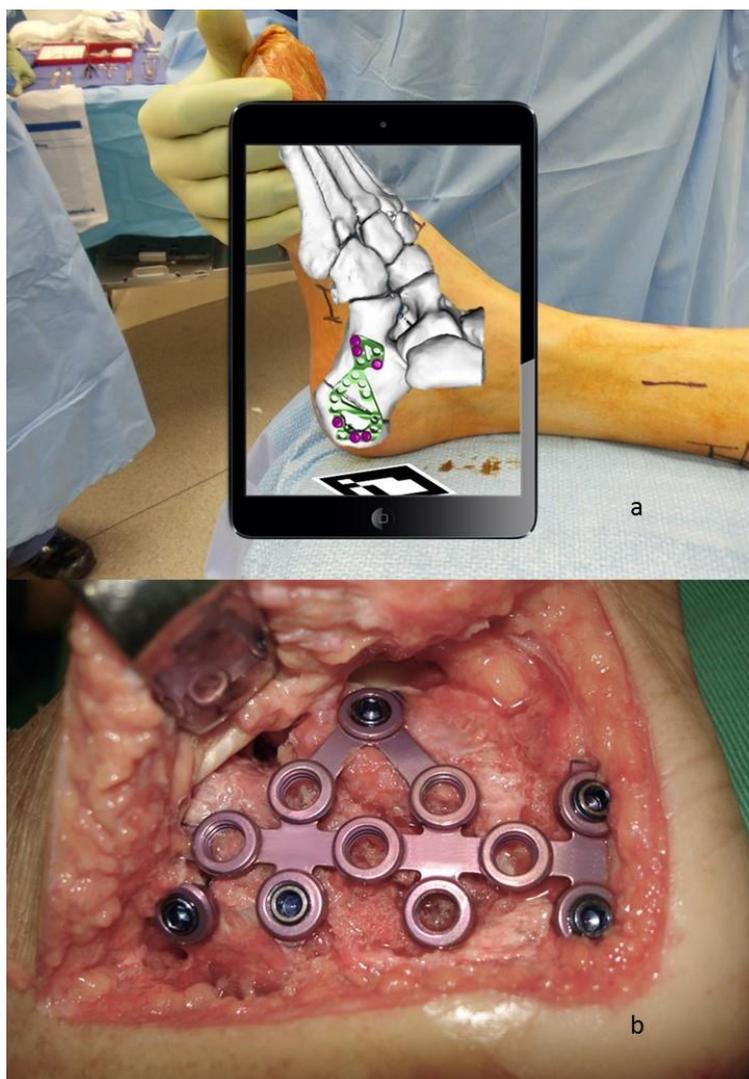


Figura 3.5 - Resultados da implementação do sistema ARPeek. a) utilização da aplicação de RA no processo pré cirúrgico onde é possível visualizar o modelo 3D do pé do paciente assim como fixação do *template* de uma prótese médica; b) resultado final da intervenção planejada em 'a'.

4. DESENVOLVIMENTO DO PROTÓTIPO

4.1 Estudos de Processamento de Imagem – Extração de Características

O processo de extração de características abrange diversas áreas, para além do processamento de imagem, como o reconhecimento de padrões e a aprendizagem por máquinas, apresentando um papel extremamente importante no desenvolvimento das tecnologias de RA. Este relaciona-se com a redução da dimensão e redundância da informação a ser analisada. Por outras palavras, partindo de um conjunto inicial de dados, a extração de características pretende reduzir esse mesmo conjunto a um que contenha apenas dados relevantes para o objetivo/estudo pretendido. No que diz respeito ao

processamento de imagem, os algoritmos desenvolvidos procuram isolar porções, áreas, cores ou contornos de vários tipos de imagem.

Previamente ao processo de extração das características pretendidas de uma imagem, é muitas vezes necessário um pré-processamento da mesma. Recorrendo também a diversos algoritmos é possível nesta fase reduzir os ruídos da imagem, converter escalas de cor e destacar contornos a partir da ênfase ou redução de sombras e de luminosidade [26].

Na área da imagem digital o menor elemento que permite caracterizar uma imagem é o pixel. Para efeitos computacionais, estes dispõem-se numa grelha endereçável, ou seja, com coordenadas físicas que permitem a sua identificação e às quais estão geralmente associados três canais de valores de intensidade – *Red*, *Green* e *Blue* (RGB) – representados por *bits*. Quanto maior for o número de *bits* que define o valor de cada canal, maior é o número de valores de intensidade (cores) possíveis de representar (por exemplo, 8 *bits* permitem exibir 256 tonalidades diferentes, $2^8 = 256$) [48].

Como referido anteriormente, no desenvolvimento do protótipo desta aplicação foram realizados diversos estudos de processamento de imagem na *framework Eclipse* em C++ recorrendo a bibliotecas OpenCV (funções de baixo nível de processamento de imagem). Com estes testes pretendia-se estudar quais os melhores métodos de processamento de imagem existiam de modo a ser possível definir quais as abordagens e técnicas a utilizar no processo de calibração e disposição (abordados no capítulo seguinte). Foram assim considerados para teste os seguintes fatores nos ambientes de bloco operatório e sala de planeamento cirúrgico:

- Luminosidade do ambiente: geralmente os ambientes de implementação da solução são espaços bem iluminados;
- Contrastes: os ambientes não costumam apresentar grandes contrastes;
- Cores do ambiente: na maioria dos casos os ambientes cirúrgicos e de planeamento apresentam cores neutras e simples;
- Contornos e formas usuais nestes ambientes: existem tanto formas geométricas simples com cantos bem definidos, como objetos com uma certa curvatura;

Tendo em conta estas características do ambiente e, neste caso, da imagem a ser captada pelo dispositivo, projetou-se que, qualquer que fosse o método de rastreamento utilizado, era necessário realizar algum tipo de pré-processamento de modo a reduzir a quantidade de dados a serem analisados e, conseqüentemente, reduzir o tempo de execução deste processo.

Assim, para cada *frame* de vídeo capturado optou-se pela abordagem descrita no programa 1.

```

if (Camera Updated) then
  for each frame do
    convert image to greyscale;
    smooth the image;
    threshold image;
  end

```

Programa 1 - Passos de pré-processamento de imagem.

A conversão de uma imagem RGB (espaço de cores *default* do OpenCV) para a escala de cinzentos permite analisar cada pixel como um único valor, ou seja, a informação desse pixel passa a estar presente num único canal de intensidade em vez de estar distribuída por três (vermelho, verde e azul) – função *cvtColor* [52]. Esta conversão é necessária para que posteriormente seja possível a segmentação da imagem (*Threshold*) e a sua transcrição para binário.

O OpenCV disponibiliza diversos algoritmos para “suavizar” (*smooth*) uma imagem. Estes permitem a redução de ruídos através da aplicação de filtros que vão modificar o sinal, eliminando pontos isolados e atenuando pontos adjacentes. Os filtros mais comuns são os lineares onde os valores dos pixels de saída são calculados como uma soma ponderada dos de entrada afetados pelos coeficientes do filtro – filtro *GaussianBlur* do OpenCV [52].

Por fim, a função *threshold* transforma uma imagem em binário uma vez que, para uma certa constante (entre 0 e 255), os valores de intensidade inferiores à mesma são interpolados a zero (preto) e os superiores a 255 a 1 (branco) – método *THRESH_BINARY_INV*. Esta separação de intensidades permite identificar e extrair pixels de objetos dos pixels de fundo da imagem através dos seus contornos e/ou sombras [52]. Neste caso o valor da constante utilizada foi 127 porque este permitia um isolamento equitativo de intensidades sem que muitos pormenores fossem perdidos.

Com os resultados destes testes experimentais foi possível passar à fase de desenvolvimento da tecnologia de RA, definindo assim qual a abordagem a definir e quais as tecnologias existentes a utilizar (abordado no subcapítulo seguinte).

4.2 A tecnologia de RA

Uma aplicação de RA consiste em três etapas fundamentais: primeiro é necessário inferir qual a localização relativa e orientação do dispositivo de rastreamento e/ou visualização (normalmente uma câmara); seguidamente é realizada a calibração do conteúdo virtual a ser visualizado (modelos 3D,

imagens, texto, entre outros); por fim dá-se a sobreposição desse mesmo conteúdo sobre o ambiente capturado em tempo real. Este é um processo iterativo e o seu *output* é contínuo uma vez que pretendido simular a ideia de imersão contínua de dois mundos: o real e o virtual.

A etapa de calibração, mencionada previamente, é responsável por estabelecer a relação entre o ambiente real e o virtual, isto é, (re)ajustar e (re)calcular as dimensões e orientação dos conteúdos gráficos, tendo em conta a sua localização pretendida na imagem capturada. Esta está intimamente ligada à primeira, uma vez que recebe e processa o seu *output* (parâmetros para o cálculo das novas dimensões e orientação).

As funções de visão computacional associadas ao dispositivo de captura de imagem são tecnologias necessárias para a realização da calibração. A captação e rastreamento de pontos estratégicos pela câmara do dispositivo permitem que, conhecendo previamente quais as distâncias reais entre eles e fornecendo-as ao sistema, este calcule os parâmetros utilizado para este processo [3]. Estes algoritmos de visão computacional dividem-se em dois passos principais: a extração das características da imagem capturada (tema abordado anteriormente); seguido pelo cálculo do posicionamento e orientação da câmara no ambiente real [25].

Foram desenvolvidos diferentes métodos de rastreamento para várias áreas desde a visão computacional até à robótica e à fotogrametria [3]. Estes métodos são diferenciados pelo tipo de equipamentos que utilizam: métodos de rastreamento por sensores (infravermelhos, LED's ou até biossensores), rastreamento visual e ainda os métodos híbridos. Sendo que, em grande parte dos sistemas de RA, a câmara é um *gadget* implementado na maioria dos seus dispositivos, os métodos de rastreamento visual assumem um papel importante neste contexto e foram os escolhidos para a solução apresentada neste projeto.

O objetivo principal dos métodos de rastreamento visual é estabelecer uma relação entre as coordenadas espaciais do vídeo em movimento e as do ambiente real capturado. Dentro destes métodos visuais existem diferentes abordagens: utilização de marcadores, utilização de um objeto com textura conhecida e, os mais complexos, recorrendo a algoritmos de aprendizagem máquina como redes neuronais, raciocínio baseado em casos e a aprendizagem por reforço.

Tendo em conta o cenário de implementação desta tecnologia e a análise realizada tanto às tecnologias *state of art*, como aos testes de processamento de imagem e aos requisitos da aplicação, a abordagem selecionada para este projeto foi a dos marcadores fiduciais. Esta escolhe deve-se ao facto de estes:

- Apresentarem características de fácil e rápida extração (forma, contornos e cantos facilmente identificáveis e cores preto e branco);
- Não apresentarem praticamente qualquer custo adicional;
- Serem de fácil obtenção e manutenção;
- Terem uma utilização configurável e de fácil apreensão para qualquer utilizador, mesmo quando não treinado;
- Apresentarem bons resultados práticos para ambientes controlados em outras aplicações.

Assim, para a deteção destes marcadores a abordagem utilizada pelos autores foi a apresentada na figura 3.6 (nota para a fase de pré-processamento que é explicada no subcapítulo anterior).

Após a primeira fase de aquisição de imagem e de pré processamento da mesma (figura 3.6.a, b) é então necessário extrair as características da imagem que permitam identificar o marcador e dessa forma estimar a sua posição relativa (figura 3.6.c). Recorrendo ao módulo *imgproc* do OpenCV, para além das funções aplicadas no passo de pré-processamento, foi ainda utilizada a função *findContours*. Esta irá receber a imagem segmentada (em binário, 8-bit) e irá tratar os valores diferentes de 0 como 1 e os iguais a 0 irão assim permanecer. Cada contorno encontrado será alocado numa matriz de pontos e no final todos contornos encontrados serão guardados numa lista de matrizes. Uma nova matriz é criada para guardar, respetivamente, qual o valor hierárquico do contorno encontrado pelo algoritmo. De uma forma sucinta, esta matriz vai conter informação sobre a relação que os contornos têm um com os outros, ou seja, se são contornos internos ou externos ou se estão dentro de outros contornos sendo que, desta maneira, cada contorno irá conter informação de qual o seu contorno pai e/ou filho. No entanto, para o caso em questão esta informação não seria necessária uma vez que o marcador em questão não possuía nem necessitava para a sua deteção, desse detalhe. O argumento *RETR_LIST* da função *findContours* define que todos os contornos serão armazenados, mas que a hierarquia entre eles não, o que diminuiu consideravelmente o tempo de processamento e reconhecimento do marcador. Por fim, o último argumento determina qual o método de aproximação de contorno definido para a função. O *CHAIN_APPROX_NONE* define que todos os pontos do contorno serão armazenados, ou seja, todos os pontos consecutivos ou subsequentes do contorno serão vizinhos horizontais, verticais ou diagonais, o que permite uma representação ao máximo detalhada dos contornos e que terá utilidade para os passos posteriores.

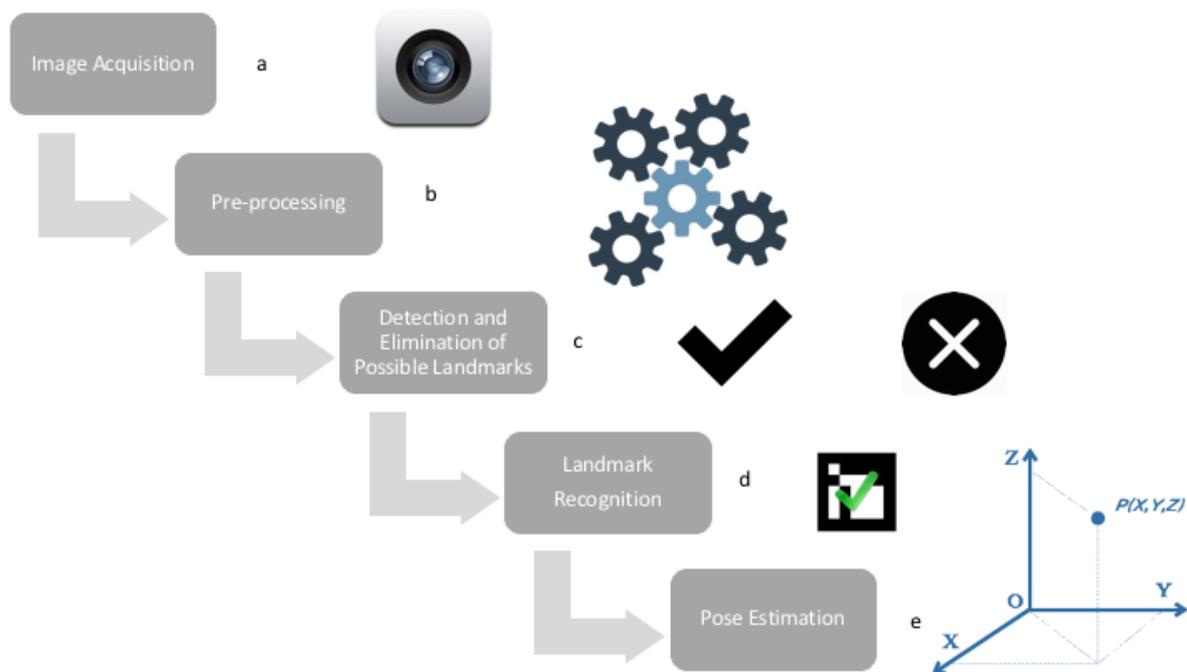


Figura 4.1 - Descrição dos passos para detecção do marcador fiducial.

Nesta fase, onde todos os dados iniciais foram reduzidos a um conjunto muito menor, é ainda necessário isolar qual dos contornos extraídos poderá de facto corresponder ao marcador. Sendo que este apresenta uma forma quadrada bem definida, dimensões conhecidas e elevado contraste (devido às suas cores branco e preto), é possível utilizar uma função de aproximação polinomial, seguida do programa 2 de maneira a reduzir ainda mais o conjunto de possíveis candidatos a marcador.

A função *approxPolyDP* do módulo de *Imgproc* do *OpenCV* aproxima uma curva, contorno ou polígono a outra curva, contorno ou polígono com menos vértices utilizando o algoritmo de *Ramer-Douglas-Peucker* [52]. Esta aproximação é feita de modo a que a distância entre os pontos do novo contorno seja menor ou igual à precisão especificada na função. Para o caso em questão, foi definido que esse valor dependia do tamanho do contorno a ser analisado multiplicado por um fator de 0,05 o que permitia tratar de forma similar contornos pequenos e/ou de maior dimensão. Esta abordagem possibilita, *a priori*, uma deteção do marcador independentemente da distância a que este se encontrar da câmara. Foi ainda definido como parâmetro da função um valor booleano (neste caso, *true*) que limita simplesmente que o contorno simplificado de saída seja fechado, ou seja, o ponto inicial que define o próprio coincide com o final.

```

if (number of points is 4) and (points describe a convex polygon) then
    check the distance between consecutive points;
    if (distance is similar) and (large enough) then
        calculate perimeter;
        if perimeter is acceptable then
            sort the contour's points in anti-clockwise order;
            save the reordered contour's point as possible marker;
        else
            save contour as rejected;
    else
        save contour as rejected;
else
    save contour as rejected;

```

Programa 2 - Especificações para a deteção do marcador fiducial no *frame* de vídeo.

Finalmente, recorrendo ao programa 2 é possível isolar com mais detalhe, quais os contornos que representam quadrados ou quadrados em perspetiva. Testando as características previamente conhecidas acerca desta forma geométrica como o número de pontos que a definem, a distância entre eles e a sua convexidade, é então possível ao *software* isolar quais os contornos da *frame* de vídeo representam quadrados. De salientar que a organização e reordenação dos pontos no sentido anti-horário tem especial importância para a identificação relativa destes (topo-esquerdo, baixo-esquerdo, baixo-direito e topo-direito) e para o procedimento descrito a seguir.

O próximo passo do desenvolvimento da tecnologia AR é o reconhecimento do marcador (figura 12.d). Como foi referido anteriormente, os contornos obtidos podem representar o marcador, mas também podem representar uma perspetiva do marcador, o que terá influência na sobreposição do conteúdo gráfico redimensionado e reorientado sobre o mesmo. Para realizar este reconhecimento, é então necessário obter a forma quadrada real do marcador. O método *getPerspectiveTransform* permite calcular a matriz de transformação em perspetiva a partir de quatro pontos e os seus correspondentes. Isto é, recebendo como argumento as coordenadas dos pontos na imagem fonte e as suas supostas coordenadas na imagem destino, este método calcula a matriz 3x3 da transformação em perspetiva [52].

Uma vez obtida a matriz é então possível aplica-la ao contorno recorrendo à função *warpPerspective*. Esta irá permitir a obtenção da imagem canónica do marcador, isto é, da sua representação bidimensional [52].

Nesta fase, o suposto marcador está dividido em 7×7 regiões iguais, das quais as 5×5 internas representam a informação real sobre o seu id. O método *getMarkerId* funciona como um algoritmo que irá percorrer todas essas regiões, analisar qual a sua cor (preto ou branco) e mapeando-as como 0 ou 1, respetivamente. O valor binário correspondente a este mapeamento irá corresponder ao id do marcador. Este algoritmo verifica uma vez mais se esta é a imagem do marcador desejado uma vez que, se as regiões de fronteira do suposto marcador não forem pretas, então este é excluído. Após este procedimento os pontos são novamente reordenados e os cantos do contorno são refinados com precisão subpixel de maneira a existir consistência e precisão máxima no processo de estimação do posicionamento real da câmara.

Resumidamente, neste ponto a posição bidimensional do contorno do marcador é conhecida. No entanto, é necessário estimar as suas coordenadas no ambiente real de maneira a que os modelos 3D possam ser exibidos de acordo com a localização desejada (figura 3.6.e).

O OpenCV dispõe de diversas funções, num modulo denominado *calib3d*, que permitem a calibração da câmara e reconstrução tridimensional. As funções deste módulo baseiam-se num modelo denominado por *Pinhole Camera Model* [53] – figura 3.7.

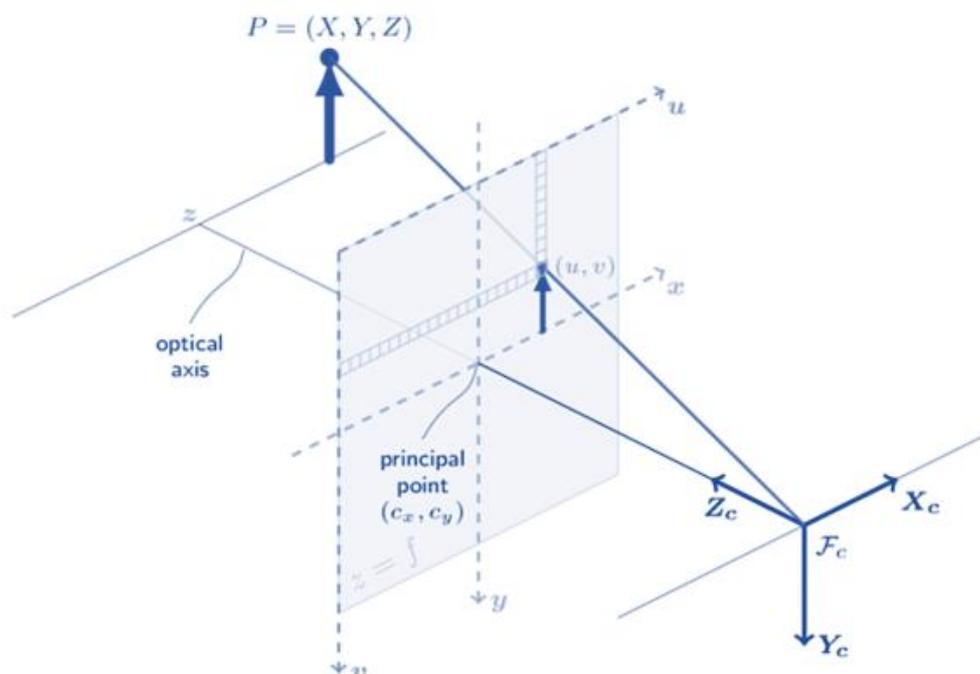


Figura 4.2 - Modelo de Câmara *Pinhole* para a calibração da câmara e reconstrução 3D [53].

No modelo *Pinhole* a cena exibida resulta da projeção de pontos 3D na imagem 2D planar utilizando parâmetros da matriz de transformação de perspetiva da seguinte equação:

$$s m' = A[R|t]M'$$

Ou

$$s \begin{bmatrix} u \\ v \\ l \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ l \end{bmatrix}$$

onde:

- (X, Y, Z) são as coordenadas 3D dos pontos no espaço cardinal do mundo real;
- (u, v) são as coordenadas do ponto de projeção em pixéis;
- (c_x, c_y) é o ponto principal que está normalmente no centro da imagem;
- f_x, f_y são os comprimentos de focagem expressos em unidade pixel.

Através da função *solvePnP* é possível calcular a disposição tridimensional de um objeto recorrendo à sua representação correspondente num ambiente bidimensional e a outros argumentos [53]. A matriz de entrada da câmara contém as coordenadas do ponto principal para a calibração e os comprimentos de focagem. O coeficiente de distorção é um vetor de 4, 5 ou 8 valores de distorção radial e/ou tangencial e permite quantificar o desvio que elementos retilíneos tenham na sua projeção. Estes valores não dependem da cena captada, mas sim aos parâmetros intrínsecos da câmara. O *raux*, que é posteriormente convertido em *Rvec*, é o vetor de rotação de saída da função. O *taux*, após convertido em *Tvec*, representa o vetor de translação que também é obtido com recurso à função *solvePnP*. Recorrendo posteriormente à função *Rodrigues* é possível converter o vetor de rotação numa matriz de rotação (*rotMat*) que será utilizada para obter as coordenadas espaciais dos pontos/contornos pretendidos na imagem captada pela câmara [53].

De maneira a que o *Unity* consiga interpretar estes parâmetros nas suas funções para manipulação de objetos, é necessário alocar os valores obtidos por estas funções do OpenCV na própria matriz de transformação associada ao objeto na GUI (define a posição, rotação e escala do objeto no cenário). É neste momento que o *software* consegue determinar quais as coordenadas 3D do marcador e desta forma projetar o conteúdo virtual sobre ele. É importante referir que alguns dos parâmetros destas equações são pré-definidos e dados como *input* para os posteriores cálculos, como por exemplo:

- As dimensões reais do marcador;
- A relação entre o marcador fiducial e os modelos que serão projetados pretende-se que seja de 1:1;
- O centro do marcador é considerado como a origem das coordenadas, ou seja, o ponto (0, 0, 0) no ambiente real.

Depois do modelo original ser submetido a um processo prévio de calibração são então calculados e aplicados os parâmetros para cada *frame* de vídeo captada. Enquanto a visualização do mesmo modelo for desejada, estes parâmetros irão alterar as coordenadas dos pontos que o definem e provocar uma variação ilusória das dimensões e orientação do conteúdo virtual.

É assim possível visualizar o modelo tridimensional totalmente adaptado ao ambiente captado e redimensionado tendo em conta a posição relativa e movimentação da câmara do dispositivo móvel face ao marcador fiducial, obtendo a ideia pretendida de imersão do mundo virtual sobre o real.

4.3 Conversão ficheiro '.obj' num modelo 3D (*Unity GameObject*)

Da fase de planeamento com o *software PeekMed* resulta um ficheiro no formato '.obj' que irá caracterizar o modelo computacional final em termos de pontos e texturas. Este ficheiro é o armazenado pelo servidor aquando *upload* do modelo por parte do médico na sua conta pessoal, ao qual estão associadas as restantes informações do paciente que permitam identifica-lo. Ao interpretar esse ficheiro, o *Unity* irá renderizar e dispor o modelo como um *GameObject* assim como transcreve-lo para o seu formato próprio de primitiva gráfica 3D – *Mesh Components* (componentes de malha). Para que esta conversão seja possível, o ficheiro que contenha a descrição do modelo terá de ser descarregado para o *Assets Directory* e o *Mesh Filter* ficará encarregue de passá-lo para um ou vários *Mesh Renders*. Um ficheiro '.obj' contém as coordenadas 3D, mapas de textura, faces poligonais e outras informações de objeto tridimensionais. O *Mesh Components* extrai esta geometria descritiva do modelo através do *Mesh Filter* e converte-a para o ambiente virtual de acordo com a matriz de transformação associada. É também nesta componente do *GameObject* e pela UI que o utilizador pode definir ou importar outras propriedades dos objetos como a luz, sombra, reflexão entre outros detalhes.

Apesar de o *Unity* intermediar a conversão deste tipo de ficheiros para a sua primitiva de construção de modelos gráficos, existia um problema na renderização dos mesmos. Este problema baseava-se no desalinhamento entre os eixos de coordenadas e escalas do ficheiro e do ambiente virtual da janela de desenvolvimento do *Unity*. Em termos práticos isto levava a que o objeto fosse colocado em coordenadas muito distantes da origem e não fosse possível visualiza-lo dentro do espaço e ângulo de visão da câmara do dispositivo. Foi então necessário criar uma classe que recalibrasse ponto a ponto as malhas que iriam compor o objeto – *ModelRenderizer*. Este processo só terá de ser realizado uma vez, aquando seleção na aplicação do modelo desejado de se visualizar de forma aumentada, sendo a tecnologia descrita no subcapítulo anterior responsável pela tecnologia de RA. Outro problema que surgiu

no processo de renderização dos modelos através do ficheiro '.obj' foi que a orientação das imagens DICOM era distinta da definida para o marcador. Como a orientação dos eixos eram opostas nos dois ambientes, quando o modelo era projetado este era visto na perspectiva posterior, ou seja, de costas. Estas adversidades são contornadas pelo *software PeekMed*, mas para efeitos práticos no desenvolvimento do protótipo não era possível implementar desde logo a aplicação de RA ao mesmo, sendo necessário desenvolver uma solução que lidasse com os problemas apresentados.

A classe *ModelRenderizer* é composta por quatro métodos fundamentais dependentes entre si e por quatro atributos: o modelo 3D que é um *GameObject*, a caixa limitadora do modelo do tipo *Bounds*, o novo quaternião do modelo do tipo *Quaternion* e um *array list* das *Bounds* originais do modelo. Os *Quaternions* são utilizados para representar rotações e têm uma estrutura extremamente complexa. O *Unity* dispõe de diversos métodos integrados que facilitam a utilização e variação desta estrutura.

O método *dicomReferencial2CurrentReferencial*, como o nome indica, será responsável por contornar o último problema descrito acima. Este irá processar os valores de rotação do modelo original (*Quaternion*) e aplicar-lhe uma rotação de 180° em torno do eixo vertical (*z*). Esta rotação irá fazer que todos os pontos sofram esta rotação e que a orientação dos dois referenciais esteja agora em concordância.

No entanto, os pontos das *Meshes* do modelo continuam a ter coordenadas muito distantes da origem do espaço virtual devido ao desalinhamento entre os seus eixos de coordenadas. A solução conceptualizada para resolver este problema passa por encontrar o centroide do modelo. Um centroide é conhecido como o centro geométrico de uma figura ou objeto. As *Bounds* (figura 3.8) são uma classe do *Unity* que permite, para este caso, delimitar o volume em torno das malhas, ou seja, são uma caixa alinhada aos eixos da malha que irá ter o menor volume possível, mas conter todos os pontos que definem a malha.

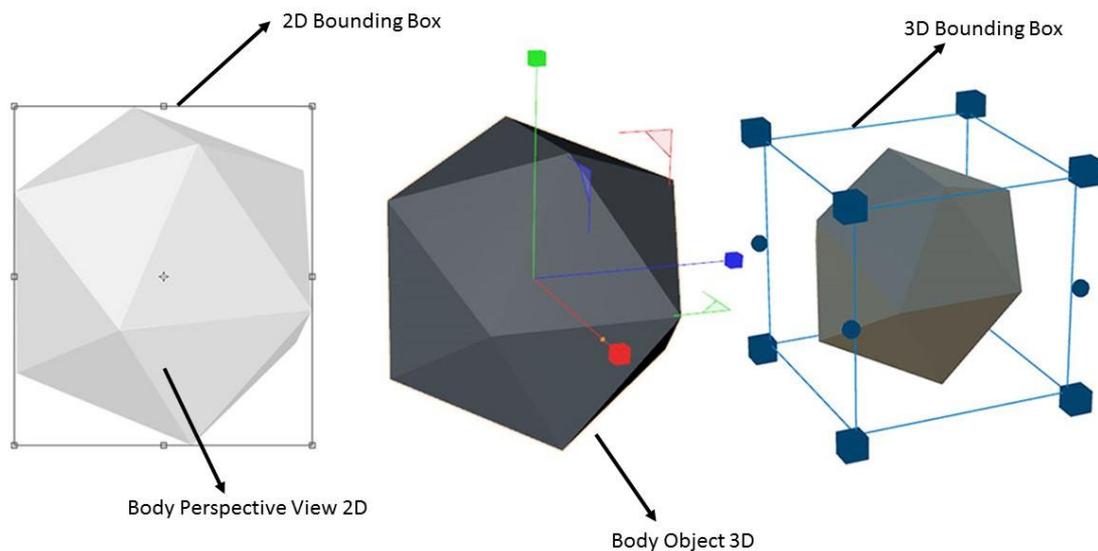


Figura 4.3 - *Bounding Box*. Uma representação 2D e 3D de uma caixa delimitadora computada para um determinado objeto [54].

Como um modelo pode ser representado por mais do que uma *Mesh*, cada uma terá a sua “caixa”. O programa 3 descreve o algoritmo utilizado para determinar o centroide do modelo, implementado pelo método *FindCentroid* e que se baseia no cálculo dos pontos medianos dos centros de cada uma dessas *boxes* (nota: o algoritmo descrito no programa 3 contém linguagem e tipos de dados específicos das classes *Unity* em C#).

```

get GameObject Meshes in ListOfMeshes;
foreach (Mesh in ListOfMeshes) do
    get Bound;
    add Bound to ListOfBounds;
end
foreach (Bound in ListOfBounds) do
    get Bound's center coordinates to a 3Vector(x, y, z);
    add 3Vector to ListOf3Vectors;
end
for ListOf3Vectors
    get the highest and lowest value of x;
    get the highest and lowest value of y;
    get the highest and lowest value of z;
end
Centroide.Center.x = (highest x + lowest x) / 2;
Centroide.Center.y = (highest y + lowest y) / 2;
Centroide.Center.z = (highest z + lowest z) / 2;
return 3Vector(Centroide.Center.x, Centroide.Center.y,
Centroide.Center.z) as Centroid Point;

```

Programa 3 - Algoritmo para determinação do centroide de um modelo importado para Unity.

Depois de calculado o centro geométrico do modelo, é necessário determinar a transformação que o desloque para o ponto de origem do ambiente virtual (neste caso o centro do marcador) e aplicar

esta mesma transformação a todos os pontos que compõem as malhas do modelo 3D. A soma de um ponto com um vetor representa a translação desse ponto em função desse mesmo vetor. Adicionalmente, um vetor pode ser calculado pela diferença entre dois pontos. O método *setMeshRendersOfModel* vai implementar o algoritmo descrito pelo programa 4 que se baseia nos princípios matemáticos das translações e operações com vetores e pontos (nota o algoritmo descrito no programa 4 contém linguagem e tipos de dados específicos das classes *Unity* em C#).

```

get GameObject CentroidPoint(x, y, z);
if (CentroidPoint(x, y, z) = OriginPoint(0, 0, 0))
    return;
else
dislocationVector3 = OriginPoint(0, 0, 0) - CentroidPoint(x, y,
z);
foreach(oldMesh in ListOfOldMeshes) do
    declare newMesh;
    get oldMeshVerticesArray in a ArrayOfVector3;
    declare newVerticesArray as ArrayOfVector3 with the same
length as oldMeshVerticesArray;
    for (i=0; i<length of oldMeshVerticesArray; i++)
        newVerticesArray [i] = oldMeshVerticesArray[i] +
dislocationVector3;
    end
    newMeshVertices = newVerticesArray;
    oldMesh = newMesh;
end

```

Programa 4 - Algoritmo para centralizar o modelo importado no ponto de origem virtual definido como o centro do marcador fiducial.

Após conclusão dos cálculos e das alterações resultantes da implementação destes métodos, o modelo 3D importado está agora pronto para ser processado pela tecnologia de RA (descrito no subcapítulo anterior) uma vez que se encontra orientado e centralizado na origem do mundo virtual.

4.4 Modelo de dados

A definição de um modelo de dados é uma etapa importante quando se projeta um sistema de uma aplicação. Pretende-se que a *ARPeek* utilize o sistema PACS para o carregamento dos ficheiros DICOM a partir dos quais os modelos 3D são gerados. Essa funcionalidade é possibilitada pela sua implementação no *software PeekMed* que já satisfaz este requisito assim como as questões de segurança associadas ao mesmo. No entanto, para a aplicação existem outras informações e dados importantes que expliquem e caracterizem o seu funcionamento e comportamento.

Apesar da aplicação desenvolvida ser para suposta integração com o *software PeekMed* e deste já ter o seu modelo de dados, para o desenvolvimento do protótipo do aplicativo e do respetivo sistema

foi necessário criar um modelo representativo e que permitisse testar a aplicação e o próprio servidor. A base de dados teste foi desenvolvida utilizando a ferramenta MySQL Workbench e o modelo lógico é o descrito pela figura 3.9. Importante referir que as *datasources* consideradas para o modelo foram apenas o sistema PACS e o conceito de um Sistema de Informação Hospitalar (SIH) de modo a elaborar o desenho do modelo. A ideia é representar qualquer *data warehouse* que esteja implementado num sistema de informação hospitalar aonde seja utilizada a aplicação *PeekMed* do qual possam ser extraídos, processados e guardados dados úteis à solução.

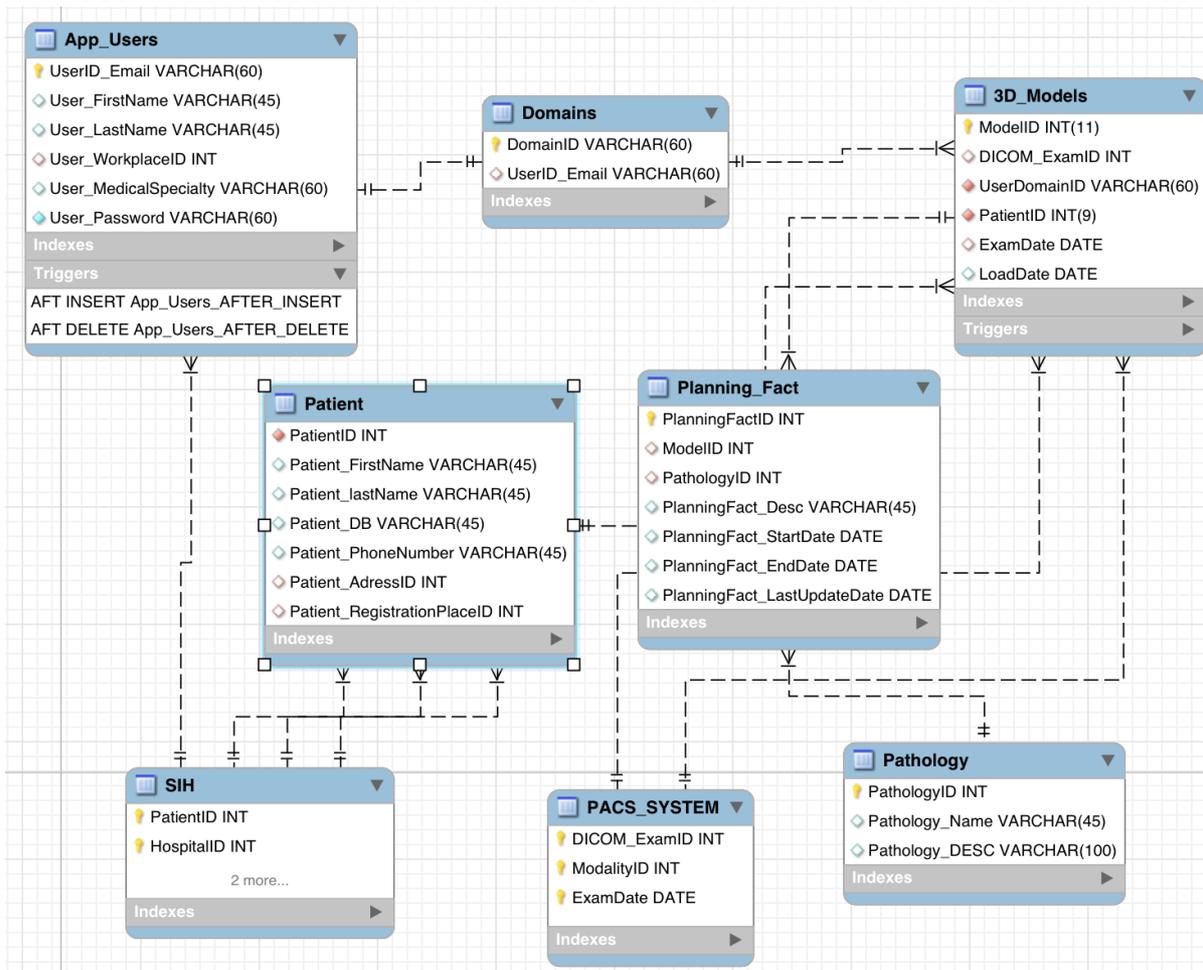


Figura 4.4 – Possível modelo lógico de dados e dimensões do sistema apresentado. A 'SIH' pretende representar qualquer Sistema de Informação Hospitalar onde a solução seja implementada. A 'PACS_SYSTEM' representa o sistema de arquivo e comunicação de imagens também da unidade hospitalar em questão.

Os objetos/dimensões identificadas e de interesse à aplicação foram os seguintes:

- **App_Users:** tabela que caracteriza o utilizador da aplicação pelo seu primeiro nome, último nome, local de trabalho e especialidade médica. A chave primária desta tabela é o atributo *UserID_Email* que não é nada mais nada menos que o email utilizado pelo mesmo aquando registo na aplicação *PeekMed*. Este atributo não poderá ser nulo e terá de ter caráter único.

Por fim o atributo *password* é também guardado aquando momento de registo. No entanto, este último, não é guardado na base de dados com a sua real descrição, mas sim recorrendo a uma função de encriptação do *Laravel* que permite garantir a segurança e confidencialidade da *password*. Por isto quer-se dizer que, mesmo que alguém consiga aceder à base de dados, não será possível identificar qual o real conjunto de caracteres forma uma determinada *password* uma vez que este valor está encriptado e é posteriormente processado pelo servidor de modo a ser realizada a autenticação do utilizador (esta função será explicada posteriormente no capítulo de ‘Arquitetura do Sistema - Servidores e Web Pages’);

- ***Domains***: tabela preenchida automaticamente, recorrendo a *triggers*, que identifica os domínios no servidor de cada utilizador, ou seja, o espaço físico que este terá no servidor e onde serão alocados os seus modelos computacionais importados para a aplicação e as suas respetivas informações da fase de planeamento;
- ***Patient***: dimensão que define o paciente associado ao evento do planeamento. Esta é caracterizada pelo seu primeiro e último nome, data de nascimento, contacto telefónico, morada (*id*) e local de registo (*id*). Apesar desta tabela poder estar integrada no *data warehouse* e simplesmente ser obtida através do mesmo, achou-se pertinente a sua individualização para o modelo de maneira a avaliar quantitativamente e qualitativamente os dados com que a aplicação iria trabalhar.
- ***Pathology***: tabela onde é caracterizada a patologia associada à fase de planeamento (facto) quanto ao seu nome e descrição. A chave primária desta tabela é o *id* da patologia (*PathologyID*) que poderá ser um código específico utilizado pelo próprio sistema de informação hospitalar (daí a sua associação ao SIH). Mais uma vez, pelos motivos descritos para a dimensão *Patient*, optou-se por representar esta tabela no modelo de dados.
- ***3D_Models***: tabela que caracteriza e identifica os ficheiros representativos dos modelos tridimensionais resultantes da fase de planeamento através de um *id* gerado automaticamente aquando carregamento dos mesmos para a aplicação principal. A cada ficheiro/modelo estão ainda associados o identificador do ficheiro DICOM que lhe deu origem, o *id* do paciente a que se referem, a data de realização do exame e a data de carregamento do modelo para a aplicação ARPeek. Por fim é ainda associado ao modelo qual o *UserDomainID*. A utilidade deste último atributo será também explicada no capítulo de ‘Arquitetura do Sistema - Servidores e Web Pages’.

- ***Planning_Fact***: tabela de planeamentos do sistema, onde o evento é o planeamento e os atributos que o caracterizam, para além do seu identificador (chave primária), são: o *id* do paciente, o *id* do modelo 3D resultante e utilizado para o evento, o identificador da patologia em causa e o *id* do utilizador da aplicação a quem pertence este evento (neste caso o email). No que toca aos restantes atributos, são ainda definidos a descrição do procedimento, a data de inicio do planeamento, a data de fim e a sua última data de atualização.

No modelo apresentado pela figura 3.9 é ainda possível verificar as seguintes relações entre dimensões:

- ***App_Users 1 : 1 Domains*** – a cada utilizador está associado um e só um único domínio e vice-versa;
- ***Domains 1 : ∞ 3DModels*** – cada domínio poderá conter mais do que um modelo 3D, mas cada modelo só poderá corresponder a um certo domínio. Consequentemente, um planeamento só poderá estar ligado a um único utilizador, mas um utilizador pode ser associado a mais do que um planeamento.
- ***Patient 1 : ∞ 3DModels*** – um paciente pode estar associado a mais do que um modelo no sistema, mas cada modelo 3D está relacionado a um único paciente. Assim, por consequência, um paciente poderá estar associado a vários planeamentos, enquanto cada planeamento só pode corresponder a um único paciente.
- ***Planning_Fact ∞ : 1 3D_Model*** – um planeamento só pode referir um único ficheiro/modelo 3D importado da fase de desenvolvimento, mas o mesmo modelo pode ser utilizado em vários planeamentos;
- ***Planning_Fact ∞ : 1 Pathology*** – um planeamento só pode estar associado a uma patologia, mas a mesma patologia pode estar associada a vários eventos.

Para além da conceptualização do modelo e das suas relações, foram ainda utilizadas outras funções para simular o funcionamento, integração e implementação de dados no mesmo recorrendo a *SQL functions* e *triggers*. Os *triggers* são procedimentos especiais armazenados na base de dados que são executados quando ocorrem determinadas ações. Este tipo de procedimentos está normalmente associado a ações de *Data Manipulation Language* (DML) como *INSERT*, *UPDATE* e *DELETE*. Este tipo de procedimentos permite também garantir de certa forma a coerência e consistência entre os dados e permite alguma abstração e delegação de tarefas ao programador e utilizador ou até mesmo ao servidor.

Desta forma, os triggers criados foram:

- O *trigger* **'App_Users_AFTER_INSERT'** vai acionar um evento que irá povoar automaticamente a tabela de domínios da base de dados. Sempre que for registado um utilizador pelo servidor e for realizada uma entrada na tabela *App_Users*, este *trigger* irá criar um *id* de domínio recorrendo à função *UUID()*. Esta função retorna um *Universal Unique Identifier* de acordo com a RFC 4122 e é um valor com 128-bits representado por uma *string* utf8. Os valores obtidos são únicos no tempo e no espaço o que garante praticamente a criação de valores não repetidos. Após gerar este valor, esse identificador será associado ao email do utilizador que acabou de se registar na respetiva tabela. No primeiro *upload* que o utilizador realizar na plataforma web, uma *path* para um espaço físico será criada com este *id*, sendo esse o caminho para o espaço em memória disponível para o utilizador carregar os seus modelos no servidor para a aplicação;
- O **'App_Users_AFTER_DELETE'** garante que quando um utilizador é apagado da tabela de *users*, o seu *DomainID* seja eliminado na tabela de domínios e, conseqüentemente, todos os modelos 3D que existam na sua respetiva tabela, associados ao *id* do domínio removido, sejam também eliminados. Por fim, todos os planeamentos associados a este utilizador são também eliminados da respetiva tabela;
- O **'3D_Models_AFTER_DELETE'** define que, quando um certo modelo é eliminado da tabela de modelos, o(s)evento(s) de planeamento associados a este modelo são também eliminados.

4.5 Arquitetura do Sistema - Servidores e *Web Pages*

De maneira a testar a possível implementação da solução de RA *ARPeek* com o *software PeekMed*, foi necessário desenvolver um servidor teste para a aplicação que permitisse simular o processo de comunicação entre a aplicação e o servidor, ou seja, processos como a validação de utilizadores, carregamento e descarregamento dos modelos 3D por parte dos médicos e ainda a associação dos mesmos a informação da situação e identidade do paciente. De salientar que o próprio *software PeekMed* é responsável por gerar os modelos 3D através das imagens DICOM carregadas pelo utilizador e por garantir a confidencialidade, gestão e segurança das informações. Ou seja, na aplicação e implementação final, o utilizador não fará o *upload* dos modelos em ficheiros *'.obj'*, mas sim dos ficheiros DICOM desejados para a sua conta.

Como foi referido anteriormente, este sistema *web* foi desenvolvendo recorrendo ao *Laravel*, uma *framework full stack* que suporta diversas tarefas desde o desenvolvimento e implementação do servidor *web* até à gestão de bases de dados e interface ao cliente.

A utilização do *Laravel* como ferramenta multitarefa permitiu uma abordagem mais intuitiva à arquitetura projetada para o sistema da aplicação que iria assentar numa arquitetura cliente-servidor. Recorrendo à comum separação de dois conceitos utilizados no desenvolvimento de *softwares* atuais, o sistema planeado seria constituído por dois servidores, um *back-end* e outro *front-end*. Esta ideia assenta na separação de dois tipos de serviços que seriam necessários implementar no sistema:

- Serviços de interface ao utilizador denominada pela camada de apresentação – responsável pela entrega e formatação da informação à camada da aplicação para posterior processamento e exibição;
- Serviços de acesso a dados – responsável por fornecer um acesso simplificado aos dados armazenados numa base de dados persistente.

Na arquitetura de *softwares* podem existir múltiplas camadas entre o *hardware* e o utilizador sendo que cada uma pode ser caracterizada como tendo um *front end* e um *back end*. O objetivo principal da separação destes dois conceitos e da sua utilização na projeção da arquitetura deste sistema é que a componente *front* represente uma abstração máxima e simplifique a interação do utilizador com o sistema fornecendo uma interface *user-friendly*. Por sua vez, a componente *back* ficaria encarregue de assegurar toda a lógica de negócio e armazenamento de informação na base de dados respetiva. O objetivo desta abordagem seria uma melhoria na distribuição de tarefas pelos componentes do sistema, facilitar a manutenção e evolução do mesmo e, conseqüentemente, esperar uma melhor performance do mesmo. Assim, a arquitetura pretendida para o sistema assemelha-se à apresentada na figura 3.10.

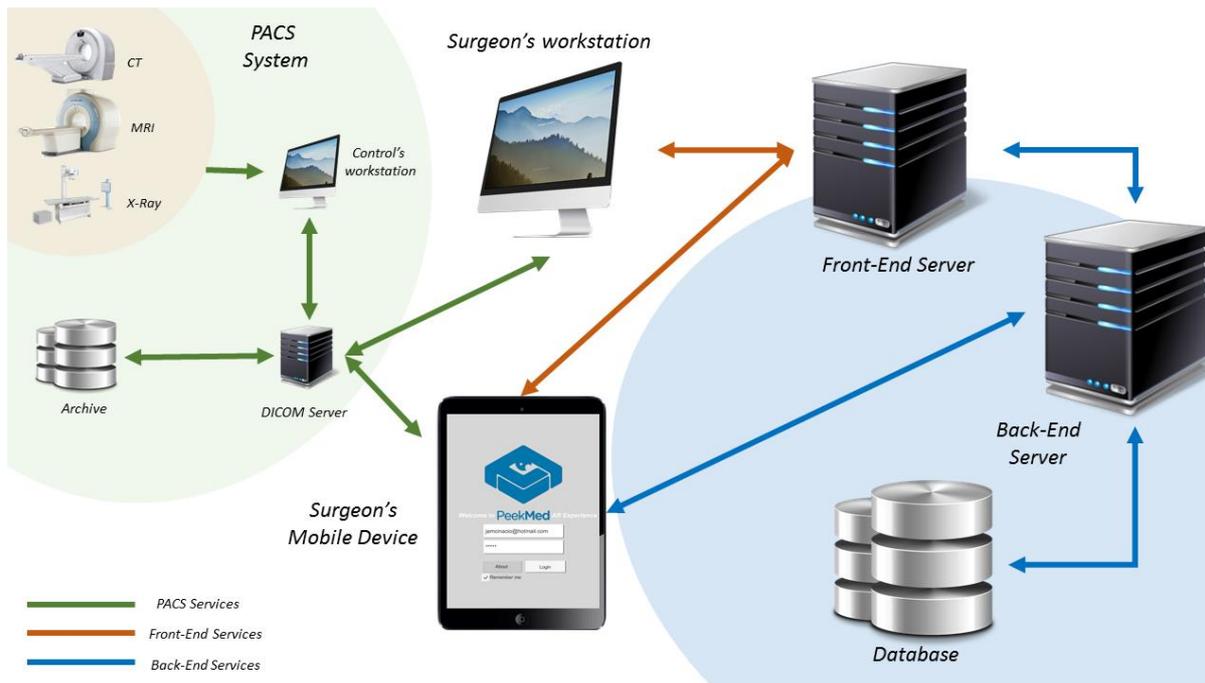


Figura 4.5 – Arquitetura do sistema para implementação da solução ARPeek.

Como foi referido anteriormente, a estrutura por trás do desenvolvimento de interfaces ao utilizador em *Laravel* é a MVC. Esta é assenta na arquitetura de comunicação *Representational State Transfer* (REST) que consiste:

- Num protocolo cliente-servidor onde cada mensagem HTTP contém toda a informação necessárias para o processamento do pedido;
- Nos recursos ou elementos de informação que são identificados pelos seus *Uniform Resource Identifier* (URI) e que podem representar objetos;
- Num conjunto de operações específicas que permitem manipular os recursos – GET, POST, DELETE, PUT, PATCH, etc.

Esta abordagem confere à arquitetura destes sistemas uma elevada performance, escalabilidade, portabilidade, confiabilidade, possibilidade de modificar e atualizar componentes dinamicamente e ainda a simplicidade e visibilidade da comunicação entre os seus elementos.

Outro conceito importante na implementação e desenvolvimento destes sistemas é processo de *Routing* (Roteamento, figura 3.11). Em redes computacionais, este sistema é o responsável por seleccionar a *path* (*Uniform Resource Locator* - URL) específica para o fluxo de dados na rede ou entre múltiplas redes. Neste caso, o roteamento é responsável por reencaminhar os pedidos ao controlador *Laravel* apropriado para que este possa processá-lo e iniciar o processo de resposta ou, de uma forma

simples, este irá mediar e definir as operações sobre os recursos do sistema do servidor *web*. O *Laravel* permite uma fácil gestão de todo este processo sendo que todos os *routes* estão definidos num diretório específico para ficheiros *routing*.

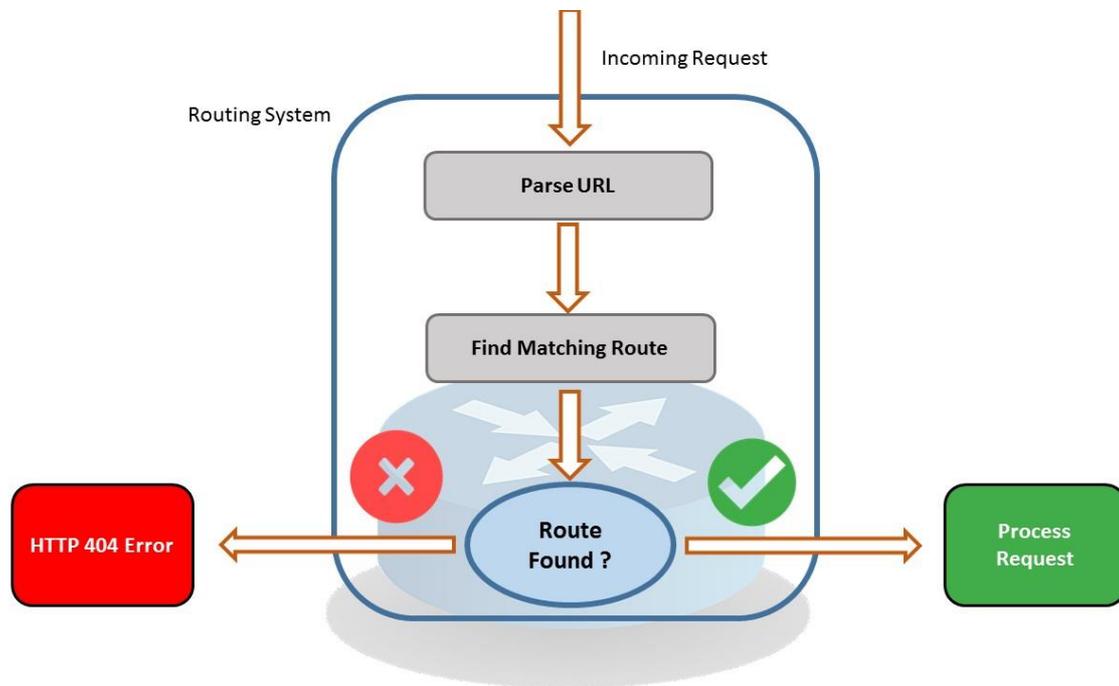


Figura 4.6 - Funcionamento de um Sistema de Roteamento – Routing.

Para o desenvolvimento do protótipo de servidor web neste trabalho foram então definidos e criados quatro controladores principais: o *loginController*, o *registerController*, o *modelsController*, e o *unityController*. Como alternativa a definir toda a lógica de gestão de pedidos no sistema de rotas, estes serão responsáveis por agrupar e gerir pedidos relacionados aos diversos contextos numa única classe definida para cada um. Para cada classe são definidos métodos que serão invocados através das *routes* e que irão processar os dados contidos nos pedidos invocados pela mesma.

Para a descrição e explicação do sistema de servidores web elaborado neste trabalho, a abordagem utilizada será *top to bottom*. Isto é, será feita uma exposição das *views*/interfaces ao utilizador e das suas finalidades e para cada uma delas procurar-se-á explicar de uma forma simples o sistema de roteamento por trás e o trabalho do controlador associado.

Quando o utilizador acede à *main path* do *web server* é-lhe exibida uma página introdutória (figura 3.12). Nela ele poderá consultar alguma informação sobre a aplicação e navegar por *links* relacionados com o projeto e temática. Nesta página é também onde o utilizador pode encaminhar-se para a página de registo ou de validação/*login*. Para tal são definidas duas *post routes* distintas

associadas aos respetivos botões, uma responsável por reencaminhar a utilizador para a página de login e outra para a de registo.

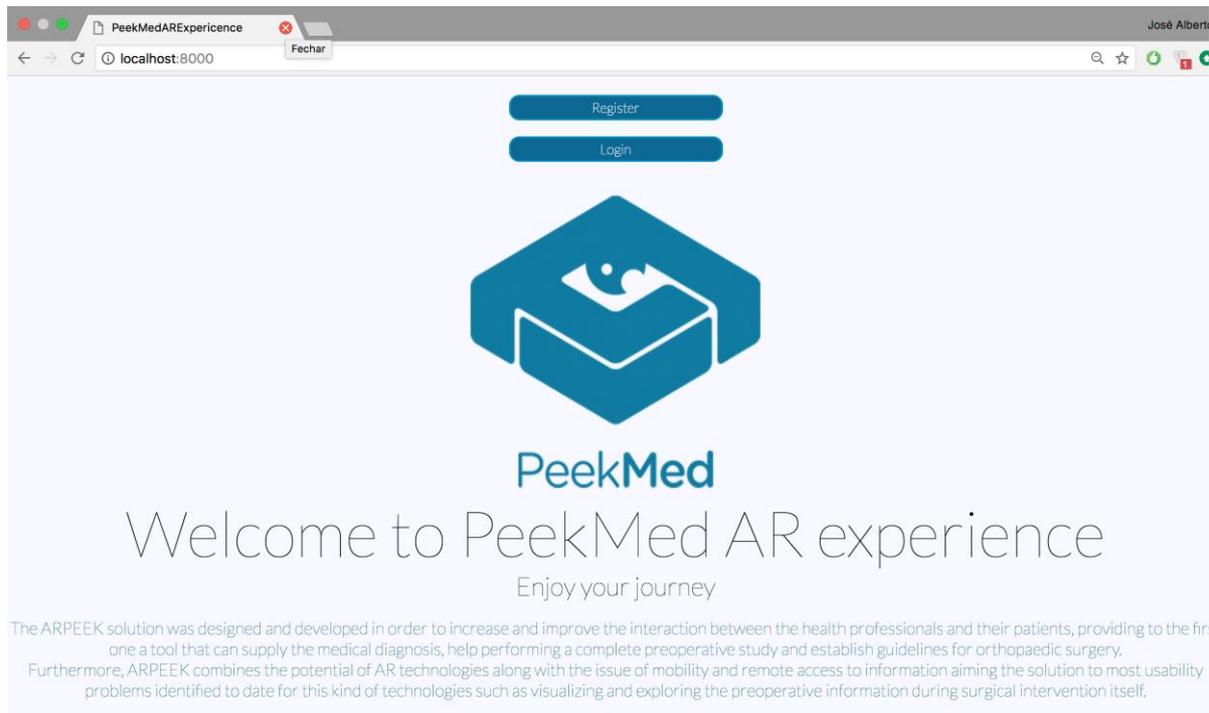


Figura 4.7 – Página inicial do protótipo do *Web Server* desenvolvido.

Um utilizador que ainda não se encontra registado no sistema, recorrendo à respetiva página, poder-se-á registar preenchendo os campos de primeiro e último nome, instituição de trabalho, especialidade, *email* e *password* (figura 3.13). De modo a conferir a credibilidade e viabilidade máxima do posterior processo de autenticação, foi definido que os atributos primeiro e último nome e *email* seriam de caráter obrigatório. Sendo o *email* definido como o nome do utilizador, para além de ser necessário verificar a sua formatação enquanto conjunto de caracteres que represente efetivamente um *email*, é ainda fundamental garantir a sua unicidade na tabela *App_Users* da base de dados. De maneira a garantir a consistência e coerência dos dados de *input* com os tipos definidos para a base de dados, são ainda realizadas algumas validações quanto ao tipo, número mínimo e máximo de caracteres permitidos. Por fim, a *password* escolhida pelo utilizador é requerida duas vezes de modo eliminar algum erro, lapso ou distração que possa ter ocorrido durante o preenchimento destes parâmetros. Nesta fase a *password* é transformada recorrendo à *função hash* do *Laravel*. Esta será responsável por codificar a *string password* numa linguagem de interpretação e leitura incomum o que confere segurança ao armazenamento destes dados na base de dados. De salientar que quando estes campos não são preenchidos devidamente ou que existe algum requisito ao registo não cumprido, o *controller* irá enviar para a sessão como resposta qual e onde se processa o problema.

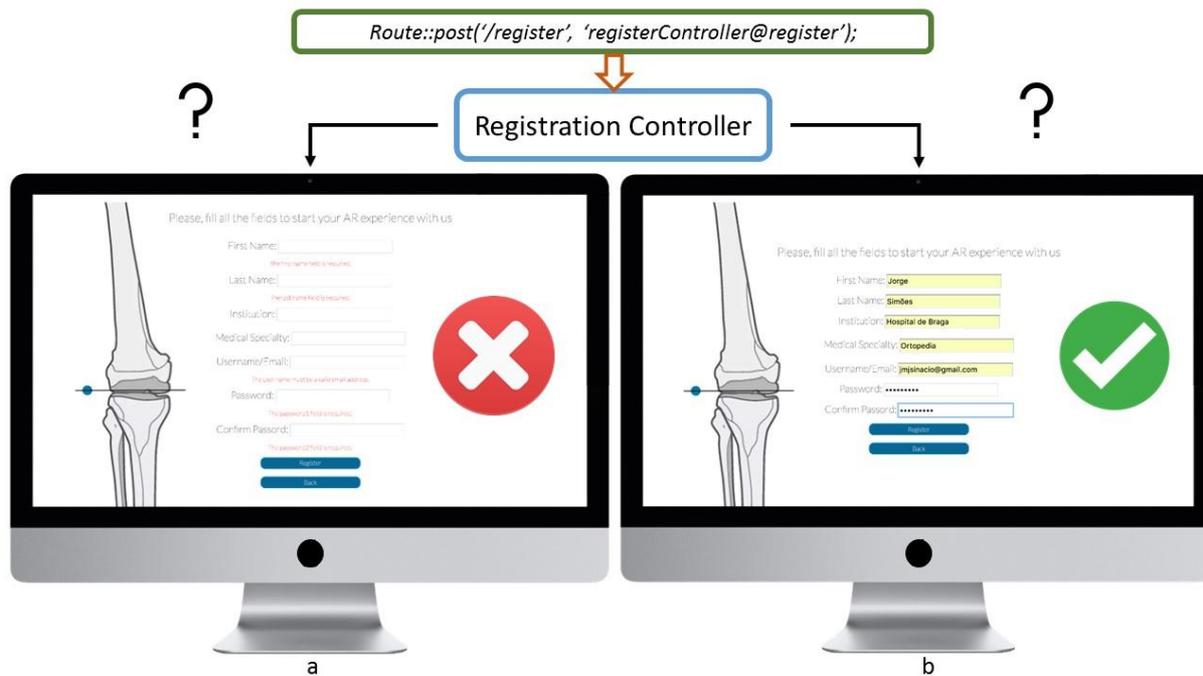


Figura 4.8 - *View* para registo de utilizador, controlador e *route* associados: a) com mensagens de erros de consistência e coerência de dados de dados b) corretamente preenchida.

Depois de validados os dados, o utilizador é então registado no sistema e poderá prosseguir para a fase de autenticação (figura 3.14.a), onde poderá ter acesso aos seus conteúdos e informações e manipular os seus objetos. Paralelamente a esta fase de registo, é gerado na base de dados o *id* do domínio do utilizador que será utilizado posteriormente.

O *loginController* irá recorrer ao método *login* para validar os dados de autenticação verificando se a *password* inserida corresponde ao *email* colocado. Após validação do utilizador, este método é também responsável por armazenar na *cache* da sessão o *id* do domínio do utilizador de maneira a evitar pedidos repetidos ao servidor. Quando o utilizador efetua o *logout* esta *cache* é limpa de modo a garantir a inacessibilidade aos domínios e consequentemente à informação contida nestes por *users* estranhos. Este controlador define também o método *showData*, implementado pelo *login*, que vai retornar uma lista de todos os modelos 3D que o utilizador carregou e da informação associada aos mesmos, sendo este conteúdo exibido na página pessoal do utilizador (figura 3.14.b).

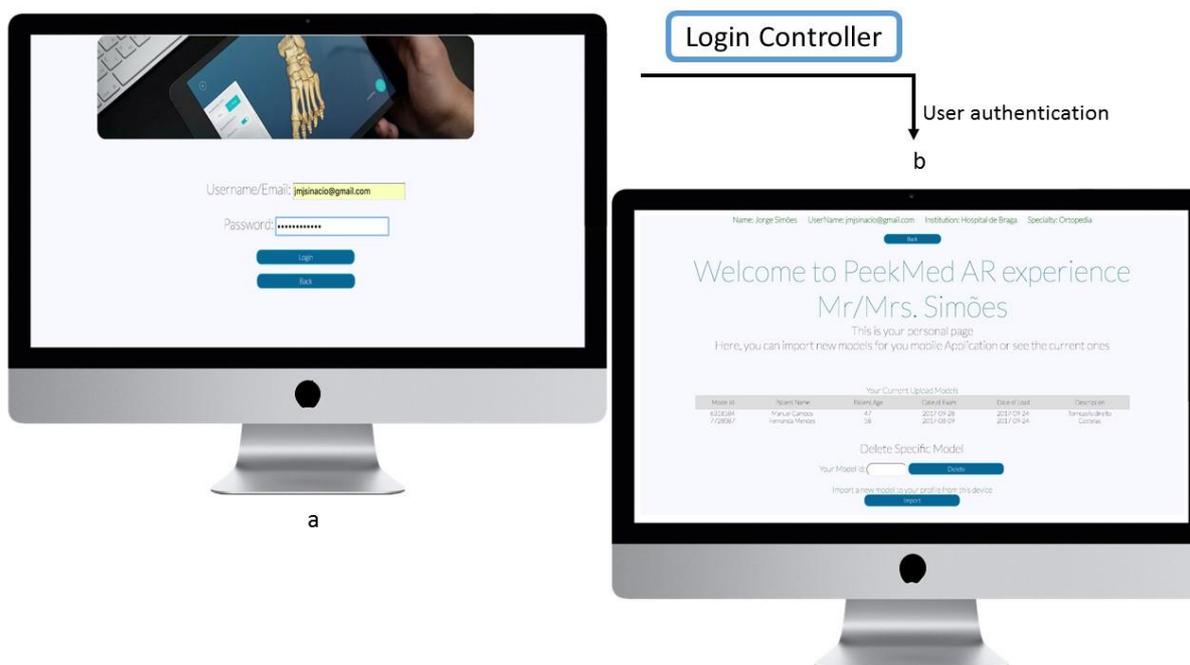


Figura 4.9 - Views do web Server. a) página para autenticação do utilizador; b) página pessoal onde são dispostos os modelos 3D respetivos aos ficheiros carregados, a informação que lhes é associada, informação pessoal do utilizador, botão de navegação para o menu de carregamento de outros modelos e a função de eliminar modelos.

É nesta página pessoal que o utilizador poderá gerir os seus modelos, apagando-os através do seu identificador e recorrendo ao método *deleteModel* do *modelsController*, ou importar novos. Por questões de segurança, sempre que um utilizador requer a eliminação de um modelo inserindo o *id* do mesmo, é verificado se este tem privilégios sobre o mesmo, ou seja, se de facto o *user* pode apaga-lo. Esta verificação é feita pelo identificador do domínio, guardado previamente na *cache* da sessão. Processualmente, se o número do modelo 3D não estiver associado na base de dados ao domínio do utilizador que realizou o pedido de eliminação, então o controlador impedirá que o processo ocorra.

Caso seja selecionada a opção de importar um novo modelo, o utilizador será reencaminhado para uma página pela *get route* associada ao *URL* onde poderá iniciar o processo. Quando o ficheiro de um modelo é carregado para o servidor, é ainda necessário identificar esse mesmo ficheiro pelo *id* do paciente e data de realização do exame sendo estes parâmetros de carácter obrigatório. Podem adicionalmente ser alocados ao objeto outras informações como o nome do paciente, a sua idade e uma breve descrição da patologia.

O ficheiro carregado é validado quanto ao seu tipo e a sua extensão é guardada. Após validação da formatação e tipo de caracteres inseridos para os restantes dados, o método *upload* do controlador irá criar um *id* único para o modelo de sete valores numéricos, recorrendo à função *generateID* elaborada que irá gerar o valor recorrendo à função *rand* e garantir que o *id* gerado não existe na base de dados.

Este *id*, juntamente com o de domínio do utilizador e a extensão do ficheiro, irá gerar a *path* de armazenamento e acesso ao ficheiro no servidor (figura 3.15 – *Path Calculation*). Todos estes dados são também armazenados na tabela de *3D_Models* da base de dados para posteriores processos e pedidos da aplicação, sendo ainda armazenada a data de realização do carregamento do modelo (figura 3.15 – *Create DB Entry*). Caso o procedimento se realize sem problemas o utilizador poderá ver uma mensagem de sucesso na interface de *upload* e optar entre carregar outro ficheiro ou regressar à sua página pessoal.

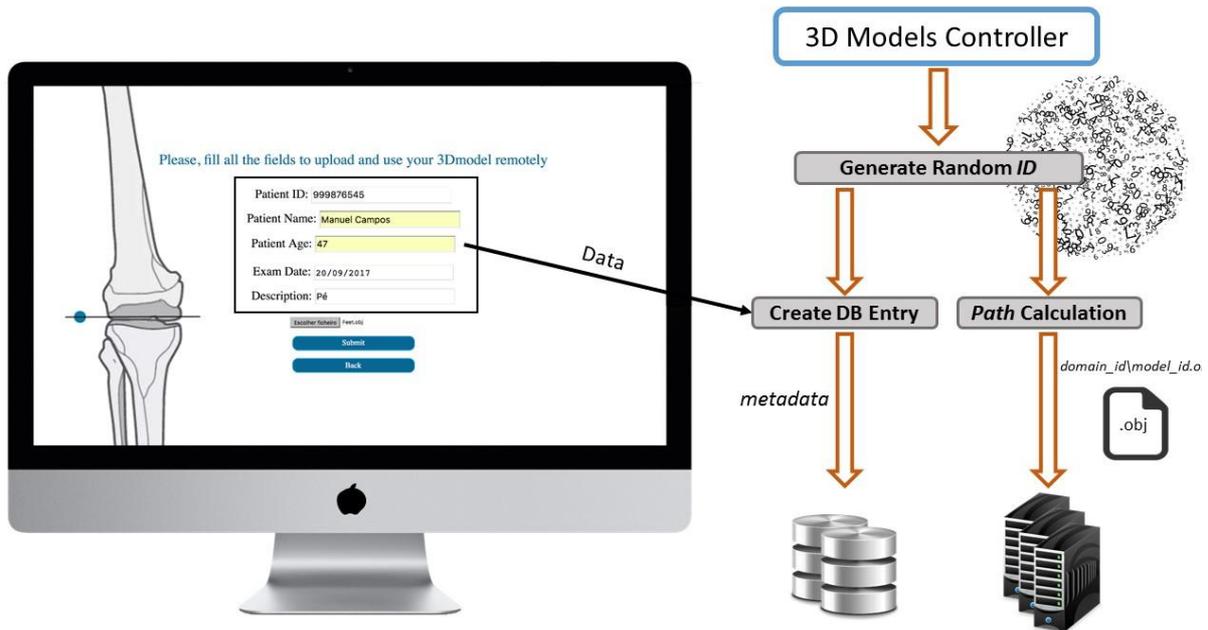


Figura 4.10 - View e sistema de carregamento de um ficheiro para o servidor.

O *modelsController* é também responsável por gerir os pedidos da aplicação ao servidor para os ficheiros que irão depois ser interpretados pelo *Unity* e através dos quais os modelos tridimensionais serão renderizados. O controlador, mediante os privilégios ou não do utilizador sobre o modelo, irá calcular a *path* do ficheiro através dos dados anexados ao pedido e retornará o ficheiro “.obj” à aplicação que ficará encarregue do resto do procedimento de renderização.

As funcionalidades do controlador *unityController* serão referenciadas no próximo subcapítulo.

4.6 Aplicação móvel para *iOS* – Desenvolvimento e UI

Consequentemente à fase de levantamento de requisitos e definição dos objetivos da aplicação, foram conceptualizados os seus estados e subestados assim como desenhadas as páginas que a UI da aplicação móvel viria a adotar.

Como foi referido anteriormente, a aplicação móvel foi desenvolvida no *Unity* e em C#. Para além de permitir a implementação das bibliotecas OpenCV para a tecnologia de RA, esta *Game Developer Tool* possibilita através do seu ambiente de trabalho um desenho intuitivo e simples de aplicações móveis híbridas, ou seja, compatíveis com diversos sistemas operativos. O *Unity* disponibiliza ainda a utilização de co-rotinas que são como funções com a capacidade de pararem a sua execução e retornar o controlo ao *Unity*, a outra função, ou a um processo externo e prosseguir no ponto e estado em que foram pausadas. Estas são declaradas recorrendo à classe *IEnumerator*. O comando '*yield*' é um tipo de '*return*' especial que garante que a função continuará a ser executada a partir da linha após a qual foi interrompida. Do lado do *Unity*, o processo para se realizar um pedido *web* pode ser descrito pelo programa 5. Nele é possível ver uma operação POST realizado a um controlador do servidor *Laravel* identificável pelo URL definido no pedido ao qual está ainda associado um *form* contendo as informações a validar/processar pelo servidor.

```
declare IEnumerator coroutine;
set URL for web request;
declare WWWForm form;
index data in the form;
declare UnityWebRequest request;
set request as UnityWebRequest.Post(URL, form);
yield return www.Send();
while (request.isOpen)
    if (www.isError) then
        return www.error;
        request.close;
    else
        declare string response;
        response = www.getResponse;
        request.close;
        return response;
end
response.process();
```

Programa 5 - Processo de um pedido web entre a aplicação e servidor - Unity.

Outra componente do *Unity* de elevado interesse para o desenvolvimento desta aplicação é a classe *PlayerPrefs*. Esta permite armazenar em sessão dados e preferências do utilizador, permitindo assim a comunicação entre as diferentes interfaces gráficas e estados da aplicação.

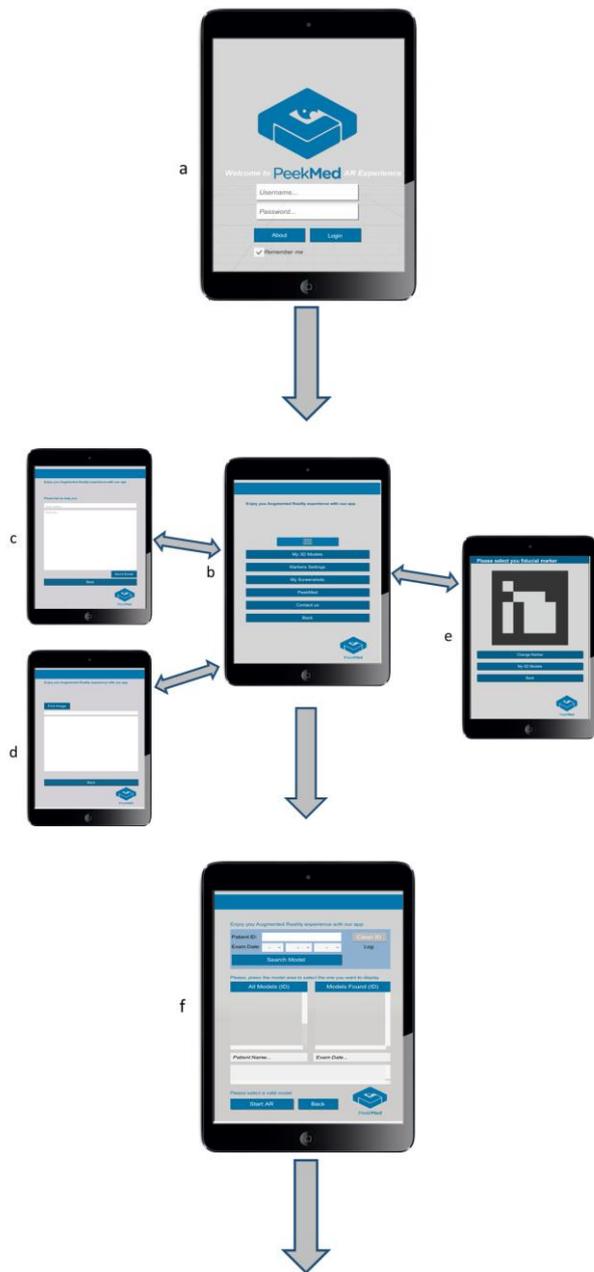


Figura 4.11 - Aplicação ARPeek – interfaces ao utilizador e navegação.

Após o utilizador instalar a aplicação no seu dispositivo móvel e estar registado pela página *web* (simulador da aplicação *PeekMed*) este terá acesso a uma página de autenticação de modo a aceder às funcionalidades da mesma e à sua informação onde utilizará como nome de utilizar o seu *email* e a password definida no momento de registo (figura 3.16.a). A aplicação irá enviar um pedido com um URL específico onde serão passados os parâmetros *email* e *password* através de uma co-rotina. O sistema de *routing* do *web server*, irá receber e analisar o URL, fazer a correspondência à *route* responsável e reencaminhar o pedido ao controlador específico (*unityController*) que irá consultar e validar os dados à base de dados executando o método *unityLogin* (consultar figura 3.11 referente ao processo de roteamento). Caso a associação entre os parâmetros se verifique, o *login* efetua-se com sucesso e os dados são armazenados no *PlayerPrefs* para futura utilização, caso contrário é enviada uma mensagem de erro à aplicação impedindo-a de prosseguir para o estado seguinte.

Depois do processo de validação de conta o utilizador passa o menu de navegação da aplicação (figura 3.16.b). Nesta interface é disposta alguma informação do utilizador obtida através da execução de uma co-rotina aquando o processo de *login*, que irá endereçar um pedido ao servidor para que este lhe envie os dados do utilizador autenticado. Neste menu, o *user* poderá optar por pelas seguintes funcionalidades:

- Contactar os administradores da aplicação para comunicar ou esclarecer alguma dúvida (figura 3.16.c - botão "*Contact Us*");
- Procurar e visualizar os seus *screenshots* obtidos anteriormente e guardados na respetiva pasta do dispositivo (figura 3.16.d - botão "*My Screenshots*");
- Configurar e selecionar o marcador fiducial que pretende utilizar para o processo de calibração e rastreamento da funcionalidade de RA (figura 3.16.e - botão "*Markers Settings*"). Numa primeira fase o utilizador poderá escolher, entre dois marcadores fiduciais, qual o que lhe mais convier caso já o tenha impresso ou qual o seu preferido de vir a imprimir;
- Aceder à página *web* principal da *Peek Health S.A* e assim obter mais informações e dicas sobre a aplicação e tecnologia ou sobre a empresa e os seus elementos (botão "*About Us*");
- Ou ainda retroceder para a página de *login*, saindo da aplicação (figura 3.16.a - botão "*Back*");

- Por fim, neste menu o utilizador poderá também, e principalmente, iniciar a sua experiência de RA optando por se direcionar à interface de seleção dos modelos (figura 3.16.f) acionando o botão “*My 3D Models*”.

Na interface de seleção dos modelos (figura 3.16.f), o utilizador poderá visualizar todos os modelos 3D que foram carregados na fase de planeamento ou então pesquisar e filtrar esse resultado pela data de realização do exame e/ou ainda pelo identificador do paciente. O(s) modelo(s) resultante(s) da pesquisa aparece(m) identificado(s) pelo seu *id* com uma cor destacada a verde. Caso não seja encontrado nenhum que preencha os parâmetros de pesquisa, é apresentada uma mensagem informando o *user*. Com uma simples seleção por *touch input* sobre o *id* do modelo (ou respetiva área do seu painel), é importado e preparada a fase de projeção. Todo este procedimento é mediado pela comunicação entre a aplicação e o servidor. Enviando pedidos URL com os dados dos elementos de pesquisa e acionando *co-rotinas* por eventos associados aos respetivos botões, a aplicação questiona o servidor sobre quais os elementos na base de dados fazem *matching* com a pesquisa realizada e este retorna uma lista de objetos. Para facilitar todo este processo de comunicação foi necessário definir a classe ‘*Modelo*’ do lado do *Unity* de maneira a ser possível estruturar as respostas do *web server* em listas de objetos e fazer a interpretação destes enquanto *JSON Objects (Javascript Object Notation)*. Esta abordagem foi adotada devido à simplicidade da notação *JSON* e da própria abstração que esta trazia quanto à formatação dos dados obtidos pelas *queries* realizadas à base de dados. Adicionalmente foi também necessário desenvolver uma classe que fizesse a interpretação e mapeamento entre os atributos do formato *JSON* e a classe ‘*Modelo*’ desenvolvida

No momento em que o utilizador carrega sobre o *id* do modelo que pretende visualizar é ainda projetada na *UI* alguma informação associada ao mesmo como dados pessoais do paciente, outros dados clínicos ou *guidelines* cirúrgicas. Paralelamente a este processo, é enviado para o *PlayerPrefs* qual o *id* do modelo selecionado pelo utilizador.

Aquando seleção do botão ‘*Start AR*’ é então possível visualizar o modelo 3D da fase planeamento sobreposto ao ambiente real (figura 3.16.e). A *co-rotina* associada a este botão irá enviar um pedido contendo o *id* do modelo pretendido e o do domínio do utilizador que requereu a sua visualização (obtidos pelo *PlayerPrefs*). Por sua vez, a *path* será analisada pelo sistema de *routing* e o pedido será direcionado para o *modelsController*. Este irá validar a segurança do pedido e determinar a *path* onde está armazenado o ficheiro referente ao modelo no servidor. Em caso de sucesso o conteúdo deste ficheiro é enviado à aplicação em formato *string* que será convertido num ficheiro recorrendo à classe *FileStream* de uma biblioteca *C#* que fornece um *Stream* para um arquivo, onde são suportadas

operações de leitura e escrita síncronas e assíncronas. O ficheiro resultante será armazenado numa pasta de recursos da aplicação e com um nome único. Caso já exista um ficheiro em *cache* com essa identificação este é apagado e o novo é carregado, caso contrário o procedimento ocorre normalmente. Finalmente, os restantes componentes de renderização ficarão encarregues de interpretar o ficheiro e dele obter o modelo 3D (processo explicado no subcapítulo ‘Conversão ficheiro ‘.obj’ num modelo 3D (*Unity GameObject*)’).

Finalizado estes procedimentos e recorrendo à tecnologia implementada de RA, o utilizador poderá visualizar o modelo 3D sobre o ambiente real, apontando a câmara para o marcador fiducial localizado estrategicamente (figura 3.16.g) e aceder ao último estado da aplicação.

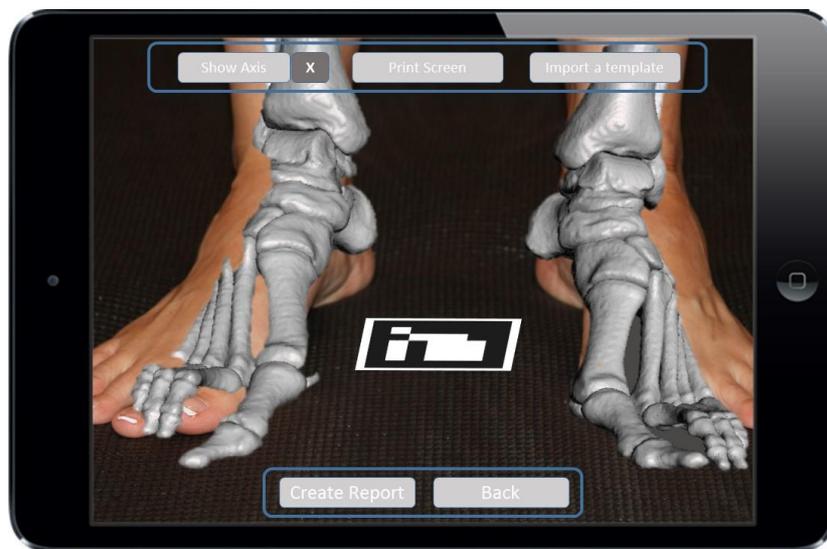


Figura 4.12 - Protótipo do ambiente de visualização do estado final de RA da aplicação.

Para esta última interface foram projetadas as funcionalidades de realizar dinamicamente outras ações como importar novos *templates* para os dispor e manipular por toque no ambiente de visualização, realizar capturas de ecrã, visualizar os eixos espaciais, criar um relatório ou retroceder para seleção de outro modelo 3D de planeamento, acionando os respetivos botões (figura 3.17).

5. CONCLUSÕES, PUBLICAÇÕES E TRABALHO FUTURO

5.1 Conclusões

O principal objetivo deste trabalho é apresentar um novo conceito e abordagem ao planeamento ortopédico através da utilização de tecnologias de RA em dispositivos móveis.

As soluções existentes até ao momento nesta área apresentam certas limitações como a difícil configuração e manutenção dos dispositivos, elevados custos associados, a fraca qualidade e detalhe na reconstrução dos modelos 3D, imperfeições a nível de disposição e visualização dos conteúdos, um extenso processo de aprendizagem para os utilizadores, elevados tempos de processamento (renderização e disposição dos modelos), a falta de portabilidade e ao nível do acesso remoto aos conteúdos gráficos e aos restantes dados do sistema.

Tendo em conta o papel preponderante que o planeamento tem no sucesso de intervenções cirúrgicas, nomeadamente na área da ortopedia, a solução proposta procura contribuir não só nesta fase, mas também na própria etapa de intervenção. Focando-se no seu utilizador, nas suas necessidades e no seu perfil, a solução ARPEEK associada ao *software PeekMed* pretende fornecer uma variedade de ferramentas e funcionalidades que permitam ao cirurgião, juntamente com a sua equipa, estudar e recriar praticamente todos os procedimentos que irão realizar, tais como a aplicação de próteses e/ou instrumentos médicos nas mais diversas estruturas ortopédicas dos pacientes (ossos, articulações entre outros) e ao mesmo tempo auxiliar a reprodução e visualização desse mesmo estudo durante a fase de intervenção. Todas as fases do planeamento podem ser discutidas e simuladas virtualmente tendo em conta toda a informação reunida acerca da condição do paciente, o que permite adaptar os procedimentos à situação e necessidades do próprio. Adicionalmente, a implementação desta solução poderá reduzir o stress e pressão a que os profissionais são frequentemente submetidos durante a cirurgia uma vez que os procedimentos foram previamente estudados, simulados, revistos e pretendem-se monitorizados e seguidos em tempo e ambiente real.

A tecnologia de RA proposta neste trabalho poderá, de uma forma prática e intuitiva, representar uma solução para a problemática da comunicação, aquisição e visualização de toda a informação (resultante da fase de planeamento) no decorrer da cirurgia e mediante as características do espaço em questão. Com a sobreposição do conteúdo gráfico sobre o ambiente real no ecrã de um dispositivo móvel, o cirurgião e a equipa podem monitorizar todos os passos da cirurgia como planeado e comparar a

situação real com a situação desejada/padrão representada virtualmente. Desta maneira é possível atenuar os momentos de embaraço e indecisão durante a intervenção e, eventualmente, melhorar os resultados e repercussões pós cirúrgicas da mesma.

A aplicação ARPeek introduz assim um conceito diferente para o planeamento cirúrgico onde o cirurgião juntamente com a sua equipa pode aprimorar e realizar todas as suas ações, desde a fase de diagnóstico e planeamento com o estudo do paciente até à intervenção propriamente dita e talvez passando pela fase pós-operatória. Para além do elevado nível de abstração disponibilizado aos utilizadores, esta solução apresenta uma grande adaptabilidade e interoperabilidade o que pode possibilitar uma melhoria de diversos procedimentos cirúrgicos noutras áreas que não a ortopédica.,

Comparando com as soluções existente, a implementação da aplicação ARPeek no *software PeekMed* adiciona tecnologias de RA ao planeamento cirúrgico numa abordagem simples e eficaz que beneficia tanto os seus utilizadores – os profissionais de saúde - como para os pacientes.

5.2 Publicações

O projeto descrito nesta dissertação foi inserido na publicação '*Next-Generation Mobile and Pervasive Healthcare Solutions*' da IGI Global. Esta contribuição teve o título de '*Augmented Reality in Surgery: A New Approach to Enhance the Surgeon's Experience*' e teve como apoio a COMPETE: POCI-01-0145-FEDER-007043 e FCT – Fundação para a Ciência e Tecnologia no âmbito do projeto UID/CEC/00319/2013.

5.3 Trabalho Futuro

O sistema *ARPeek* proposto apresenta uma solução viável para a realização e reprodução em ambiente de intervenção do planeamento cirúrgico ortopédico recorrendo às ferramentas do *software PeekMed* e das tecnologias AR que este acrescenta ao último. Este trabalho visava a conceptualização desta solução e o desenvolvimento de um protótipo inicial para posterior implementação e associação ao *software* da *Peek Health S.A.* Devido às limitações dos recursos físicos, tecnológicos e temporais, não foi possível aprofundar o desenvolvimento do protótipo da aplicação móvel. Assim, quanto ao projeto seria interessante manter um trabalho, pesquisa e atualização contínua das tecnologias e metodologias utilizadas no desenvolvimento do mesmo. Relativamente ao protótipo e à própria solução, teria interesse desenvolver funcionalidades e abordar/explorar temas como:

- A construção de modelos computacionais recorrendo a outras modalidades de imagem médica como a RM;
- Desenvolver e implementar ferramentas para a segmentação de imagens e assim a permitir a criação de modelos anatómicos específicos ainda mais pormenorizados;
- O estudo e desenvolvimento de marcadores fiduciais de baixo custo e com biocompatibilidade de modo a melhorar e facilitar o processo de calibração e rastreamento;
- A implementação da funcionalidade de importar *templates* de próteses e outras ferramentas de apoio cirúrgico para o ambiente de desenvolvimento em RA;
- A consequente manipulação desses *templates* recorrendo a *touch input*;
- O estudo e desenvolvimento de técnicas de rastreamento e calibração que recorressem a tecnologias de inteligência artificial e aprendizagem máquina, permitindo ainda uma maior abstração ao utilizador uma vez que deixaria de ser necessária a presença de um marcador físico no espaço real. Estas técnicas permitiriam o reconhecimento computacional automático e autónomo de características humanas assim como de estruturas anatómicas tais como mãos, pés, pernas, dedos, joelhos, cotovelos, ombros, etc.;
- A criação e utilização de modelos computacionais dinâmicos, isto é, em vez destes apresentarem uma estrutura superficial fixa no tempo e no espaço, poderiam variar a sua forma, orientação e até disposição de acordo com o desenrolar da intervenção cirúrgica, simulando realmente o impacto que esta teria sobre eles;
- A criação, utilização e monitorização de modelos computacionais que alterassem, mais uma vez, a sua localização, orientação e deformação, mas de acordo com os eventos naturais e fisiológicos do paciente através do *input* de outros dados de monitorização;
- A implementação de servidores compatíveis com o HIPPA garantindo a segurança de armazenamento dos dados do paciente;
- Melhorias a nível das interfaces ao utilizador (*User Interface – UI*) e da experiência (*User Experience – UX*).

REFERÊNCIAS

1. Mildenberger P, Eichelberg M, Martin E (2002) Introduction to the DICOM standard. *Eur Radiol* 12:920–927. doi: 10.1007/s003300101100
2. Pelargos PE, Nagasawa DT, Lagman C, et al (2016) Utilizing virtual and augmented reality for educational and clinical enhancements in neurosurgery. *J Clin Neurosci*. doi: 10.1016/j.jocn.2016.09.002
3. Siltanen S (2012) Theory and applications of marker-based augmented reality.
4. Ribeiro J, Alves V, Silva S, Campos J (2015) A 3D Computed Tomography Based Tool for Orthopedic Surgery Planning. 121–137. doi: 10.1007/978-3-319-13407-9_8
5. Herford AS, Miller M, Lauritano F, et al (2017) The use of virtual surgical planning and navigation in the treatment of orbital trauma. *Chinese J Traumatol* 20:9–13. doi: 10.1016/j.cjtee.2016.11.002
6. Shiha A, Krettek C, Hankemeier S, et al (2010) The use of a professional graphics editing program for the preoperative planning in deformity correction surgery : A technical note. *Injury* 41:660–664. doi: 10.1016/j.injury.2009.10.051
7. Ribeiro J, Alves V (2012) Planeamento Cirúrgico Ortopédico assistido por computador - Uma abordagem 3D.
8. Weese J, Lorenz C (2016) Four challenges in medical image analysis from an industrial perspective. 33:44–49. doi: 10.1016/j.media.2016.06.023
9. (2016) Standing CT vs. MRI for Advanced Visualization of Knee Cartilage and Meniscus. <http://www.curvebeam.com/blog/standing-ct-vs-mri-for-advanced-visualization-of-knee-cartilage-and-meniscus/>. Accessed 6 May 2017
10. Web Pathology (2017) Chondromyxoid Fibroma: MRI. <http://www.webpathology.com/image.asp?case=331&n=4>. Accessed 6 May 2017
11. Anderson Menezes Pimenta (2012) Endoscopia na Obesidade Mórbida. <https://dinhopimenta.blogspot.pt/2012/06/endoscopia-na-obesidade-morbida.html>. Accessed 6 May 2017
12. Alalawi ZM, Eid MM, Albarrak AI (2016) Assessment of picture archiving and communication system (PACS) at three of ministry of health hospitals in Riyadh region – Content analysis. *J Infect Public Health* 9:713–724. doi: 10.1016/j.jiph.2016.09.004
13. Bellon E, Feron M, Deprez T, et al (2011) Trends in PACS architecture. 78:199–204. doi:

- 10.1016/j.ejrad.2010.05.025
14. Konica Minolta Pacs System. <https://www.konicaminolta.eu/de/healthcare/products/veterinary-solutions/regius-sigma/workflow.html>. Accessed 5 Jun 2017
 15. Wade RH, Kevu J, Doyle J (1998) Pre-operative planning in orthopaedics : a study of surgeons ' opinions. 29:785–786.
 16. Materialise Surgical Planning Software Orthoview Materialise Medical. <http://www.materialise.com/en/medical/materialise-orthoview>. Accessed 28 Oct 2016
 17. BRAINLAB TraumaCad - Introduction. <http://www.traumacad.com>. Accessed 15 Oct 2016
 18. Materialise SurgiCase Connect now available for iPad Materialise. <http://www.materialise.com/en/press-releases/surgicase-connect-now-available-for-ipad>. Accessed 28 Oct 2016
 19. Helie S (2017) ScienceDirect Augmented reality : An ecological blend. *Cogn Syst Res* 42:58–72. doi: 10.1016/j.cogsys.2016.11.009
 20. Hendricks B (2017) The AR 9: Augmented Reality Today. <https://www.linkedin.com/pulse/ar-9-augmented-reality-today-benjamin-hendricks/>. Accessed 6 Sep 2017
 21. BIANCHINI R (2017) Microsoft HoloLens mixed reality device – our test and review. <https://www.inexhibit.com/case-studies/microsoft-hololens-mixed-reality-device-test-and-review/>. Accessed 13 Jul 2017
 22. A Voz Da Indústria (2017) Realidade Aumentada na Indústria: quais são as vantagens? <http://avozdaindustria.com.br/realidade-aumentada-na-industria-quais-sao-as-vantagens/>. Accessed 6 Oct 2017
 23. Elizabeth Woyke (2016) How Stores Will Use Augmented Reality to Make You Buy More Stuff.
 24. Hofmam M, Carvalho FG, Szenberg F, et al (1900) Um Estudo sobre Marcas Fiduciais em Realidade Aumentada : Combinando Detecção de Linhas com Calibração de Câmera. 337–348.
 25. Khandelwal P, Swarnalatha P, Bisht N, Prabu S (2015) Detection of Features to Track Objects and Segmentation Using GrabCut for Application in Marker-less Augmented Reality. *Procedia Comput Sci* 58:698–705. doi: 10.1016/j.procs.2015.08.090
 26. Hagbi N, Bergig O, El-Sana J, Billinghamurst M (2009) Shape recognition and pose estimation for mobile Augmented Reality. *IEEE Int Symp Mix Augment Real* 65–71. doi: 10.1109/TVCG.2010.241
 27. Kapoor P, Ghufuran U, Gupta M, Agarrwal A (2013) Marker-less Detection of Virtual Objects using Augmented Reality. *Proc Conf Adv Commun Control Syst 2013 (CAC2S 2013) Marker-less*

2013:694–699.

28. The 20 Best Augmented-Reality Apps & Digital Trends. <http://www.digitaltrends.com/mobile/best-augmented-reality-apps/>. Accessed 23 Oct 2016
29. Štorková P (2015) Using augmented reality as a medium for teaching history and tourism. 174:926–931. doi: 10.1016/j.sbspro.2015.01.713
30. Geiger P, Schickler M, Schobel J, Reichert M (2016) Advanced Algorithms for Location-Based Smart Mobile Augmented Reality Applications. 94:97–104. doi: 10.1016/j.procs.2016.08.017
31. Rovadosky D, Pavan W, Dalbosco J, Cervi C (2012) Uma Ferramenta de Realidade Aumentada Usando Dispositivo Móvel com Sistema Operacional Android. *UpfTcheBr* 25–37.
32. Rashid Z, Melià-seguí J, Pous R, Peig E (2016) Using Augmented Reality and Internet of Things to improve accessibility of people with motor disabilities in the context of Smart Cities. *Futur Gener Comput Syst*. doi: 10.1016/j.future.2016.11.030
33. Mura MD, Dini G, Failli F (2016) An Integrated Environment Based on Augmented Reality and Sensing Device for Manual Assembly Workstations. *Procedia CIRP* 41:340–345. doi: 10.1016/j.procir.2015.12.128
34. Badiali G, Ferrari V, Cutolo F, et al (2014) Augmented reality as an aid in maxillofacial surgery: Validation of a wearable system allowing maxillary repositioning. *J Cranio-Maxillofacial Surg* 42:1970–1976. doi: 10.1016/j.jcms.2014.09.001
35. Hallet J, Soler L, Diana M, et al (2015) Trans-thoracic minimally invasive liver resection guided by augmented reality. *J Am Coll Surg* 220:e55–e60. doi: 10.1016/j.jamcollsurg.2014.12.053
36. Conrad C, Fusaglia M, Peterhans M, et al (2016) Augmented Reality Navigation Surgery Facilitates Laparoscopic Rescue of Failed Portal Vein Embolization. *J Am Coll Surg* 223:e31–e34. doi: 10.1016/j.jamcollsurg.2016.06.392
37. Sun G chen, Wang F, Chen X lei, et al (2016) Impact of Virtual and Augmented Reality Based on Intraoperative Magnetic Resonance Imaging and Functional Neuronavigation in Glioma Surgery Involving Eloquent Areas. *World Neurosurg* 96:375–382. doi: 10.1016/j.wneu.2016.07.107
38. Cabrilo I, Schaller K, Bijlenga P (2014) Augmented Reality-Assisted Bypass Surgery: Embracing Minimal Invasiveness. *World Neurosurg* 83:596–602. doi: 10.1016/j.wneu.2014.12.020
39. Hou YZ, Ma LC, Zhu RY, Chen XL (2016) iPhone-Assisted Augmented Reality Localization of Basal Ganglia Hypertensive Hematoma. *World Neurosurg* 94:480–492. doi: 10.1016/j.wneu.2016.07.047
40. Thomas DJ (2016) Augmented reality in surgery: The Computer-Aided Medicine revolution. *Int J*

- Surg 36:25. doi: 10.1016/j.ijsu.2016.10.003
41. Hakky T, Martinez DR, Lipshultz LI, et al (2015) Mp23-11 Augmented Reality Assisted Urologic Surgery (Araus): a Surgical Training Tool. J Urol 193:e271. doi: 10.1016/j.juro.2015.02.1254
 42. Brien RO (2008) An Overview of the Methodological Approach of Action Research. 1–18.
 43. OpenCV About - OpenCV library. <http://opencv.org/about.html>. Accessed 25 Sep 2016
 44. Pop D, Altar A (2014) Designing an MVC Model for Rapid Web Application Development. Procedia Eng 69:1172–1179. doi: 10.1016/j.proeng.2014.03.106
 45. Laravel Laravel Documentation. <https://laravel.com/>. Accessed 30 Oct 2016
 46. Peek Health SA A game changer in the pre-operative planning process A game changer in the pre-operative planning process.
 47. Caseiro JM (2006) A Equipa Médico-Cirúrgica. 15:
 48. HUGHES JF, DAM A VAN, MCGUIRE M, et al (2014) Computer Graphics Principles and Practice, 3rd ed. Addison-Wesley
 49. ClinicalGate (2015) Imaging in Radiation Oncology. <https://clinicalgate.com/imaging-in-radiation-oncology/>. Accessed 8 Nov 2016
 50. Custodio L, Etiene T, Pesco S, Silva C (2013) Practical considerations on Marching Cubes 33 topological correctness. Comput Graph 37:840–850. doi: 10.1016/j.cag.2013.04.004
 51. J W Laprairie, Mark and Ca L, and Hamilton H (2017) Isovox: A Brick-Octree Approach to Indirect Visualization.
 52. OpenCV Image Processing (imgproc.). <http://docs.opencv.org/2.4/modules/imgproc/doc/imgproc.html>. Accessed 20 Nov 2016
 53. Camera Calibration and 3D Reconstruction — OpenCV 2. http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html.
 54. Windows Dev Center App bar and bounding box. https://developer.microsoft.com/en-us/windows/mixed-reality/app_bar_and_bounding_box. Accessed 10 Aug 2017