

Mestrado em Informática e Sistemas

**Desenvolvimento de uma plataforma para aquisição de
dados em amputados do membro inferior**

Relatório apresentado para a obtenção
do grau de Mestre em Informática e Sistemas
Especialização em Desenvolvimento de *Software*

Autor

André Filipe Coelho Dias

Orientador

Prof. Doutora Anabela de Jesus Gomes

Departamento de Engenharia Informática e de Sistemas

Instituto Superior de Engenharia de Coimbra

Supervisor

Eng. Luís André Gonçalves Ferreira

Sensing Future Technologies

Coimbra, Dezembro, 2016

Agradecimentos

Acima de tudo queria agradecer à minha orientadora Professora Doutora Anabela de Jesus Gomes, por toda a ajuda, motivação e empenho demonstrado.

Queria também agradecer ao meu orientador na empresa Engenheiro Luís André Gonçalves Ferreira, por toda a ajuda e orientação dada no decorrer do projeto.

Quero também agradecer à *Sensing Future Technologies* e a todos os seus colaboradores, entre eles, quero agradecer em especial ao Professor Doutor Carlos Alcobia, ao Engenheiro Pedro Mendes e à Engenheira Cláudia Tonelo, por estarem sempre presentes e prontos a ajudar em algumas dificuldades.

A toda a minha família, pela ajuda incondicional e todo o companheirismo.

Por fim, mas não menos importante quero agradecer a todos os meus amigos que me ajudaram de alguma forma durante este último ano. Quero agradecer à Carina por todas as palavras de força e coragem, pela paciência e por me encorajar nos momentos mais difíceis.

Resumo

O desenvolvimento de uma prótese e as afinações posteriores para proporcionar um melhor conforto ao paciente é, na maioria das vezes um processo moroso e dispendioso, quer para o paciente quer para a entidade que a desenvolve. Para otimizar o processo de desenvolvimento de próteses bem como obter próteses mais confortáveis para os seus utilizadores, foi elaborada uma pesquisa e analisados distintos projetos e produtos já existentes no mercado. Assim, neste documento, serão apresentadas várias técnicas utilizadas atualmente, permitindo identificar algumas lacunas nas soluções existentes.

O objetivo do presente estágio é o de proporcionar condições para conceber próteses mais confortáveis e mais bem adaptadas aos pacientes, através de um processo mais ágil e rápido. Assim o estágio passa pelo desenvolvimento de uma plataforma informática para aquisição e apresentação de parâmetros físicos, para personalização do *socket* do membro inferior. Esta plataforma irá recolher dados do *Socket Master*, dispositivo este com a forma de um *socket* munido de uma rede de sensores de pressão e temperatura, permitindo analisar quais os pontos onde o paciente exerce maior pressão e assim ser calculada a forma do *socket* para determinado paciente.

Palavras-Chave

Amputação, Coto, Conforto, Dados Biomecânicos, Malha de Sensores, Próteses, *Socket*, *Socket Master*, Visualização 3D

Abstract

Prosthesis development and the subsequent adjustments to provide better comfort for the patient, it is in most cases a lengthy and expensive process, for both the patient and to the entity that develops it. To better define the techniques used in the development of prosthesis, a research was done. Different projects and products, which are already on the market for this purpose, were also analyzed. Thus, several techniques currently in use will be presented in this document, allowing the identification of some gaps in the existing solutions.

The aim of this internship involves the development of an IT platform for physical parameters acquisition and representation for lower limb socket customization in order to draw and design a more comfortable and best suited prosthesis to the patient through a more agile and faster process. This platform will collect data from Socket Master, which is a device with the shape of a socket, provided with a pressure and temperature sensors network, so as to analyze which points the patient is exerting more pressure.

Keywords

Amputation, Biomechanical Data, Comfort, Prosthesis, Residual Limb, Socket, Socket Master, Sensors Network, 3D visualization

Índice

Agradecimentos	ii
Resumo.....	iv
Abstract	vi
Índice	viii
Índice de Figuras	xii
Índice de Tabelas.....	xiv
Conceitos.....	xvi
Siglas e Acrónimos.....	xviii
1 Introdução	1
1.1 Enquadramento.....	1
1.2 Objetivos	1
1.3 Empresa de Acolhimento	2
1.3.1 Caracterização	2
1.3.2 Localização do Estágio.....	2
1.3.3 Metodologia Adotada pela Empresa.....	2
1.4 Plano de Trabalho	3
1.4.1 Cronograma do Projeto.....	4
1.5 Estrutura da Dissertação	5
2 Estado da Arte	7
2.1 Técnicas de construção de próteses	7
2.1.1 Método tradicional.....	7
2.1.2 Método Digital.....	11
2.2 Meias Prostéticas	14
2.3 Projetos e Produtos no mercado	15
2.3.1 F-Socket System	15
2.3.2 MIT lab's FitSocket	16
2.3.3 Prosthetic Socket Design	17
2.3.4 <i>Variable impedance prosthetic socket</i>	17
2.3.5 Síntese dos projetos e produtos analisados.....	19
2.4 Software de instrumentação para análise de dados.....	19
2.4.1 <i>Body Track</i>	20
2.4.2 Footwork PRO	20

2.4.3	F-Socket System	20
2.4.4	<i>Fore Site SS System</i>	21
2.4.5	Tune.....	22
2.4.6	Síntese dos <i>softwares</i> analisados.....	22
3	Apresentação do Projeto	25
3.1	Consórcio do <i>Socket Master</i>	25
3.1.1	<i>Socket Master</i>	26
3.1.2	Malha de Sensores	28
3.1.3	<i>Indenter</i>	29
3.2	Requisitos.....	30
3.2.1	Requisitos funcionais.....	31
3.2.2	Requisitos não funcionais.....	32
3.3	Casos de Uso	32
4	Arquitetura do Sistema	35
4.1	Visão geral	35
4.1.1	Módulo de <i>Hardware</i>	36
4.1.2	Módulo de <i>Software</i>	36
4.2	Conceção da base de dados	36
4.2.1	Modelo entidade relacionamento	36
4.2.2	Modelo conceptual de dados.....	39
4.2.3	Modelo físico de dados	39
4.3	Linguagens e Tecnologias.....	43
4.3.1	.NET	43
4.3.2	J2EE.....	44
4.3.3	PHP	44
4.3.4	Análise comparativa entre as tecnologias apresentadas.....	44
4.4	Bibliotecas e <i>Frameworks</i>	46
4.4.1	Bootstrap.....	46
4.4.2	jQuery.....	48
4.4.3	DataTables.....	50
4.4.4	AJAX.....	50
4.4.5	Three.js.....	51
4.4.6	Laravel	52
5	Desenvolvimento e Implementação	55
5.1	<i>Mockups</i> do projeto	55
5.2	Interfaces do Sistema.....	56

5.2.1	Interface para inserção de pacientes	57
5.2.2	Interface para listagem de pacientes	59
5.2.3	Interface para mostrar dados de um paciente	60
5.2.4	Interface para gestão das consultas de um paciente.....	61
5.2.5	Interface para visualizar a imagem tridimensional do coto do paciente	62
5.3	Visualização tridimensional do coto do paciente	62
5.3.1	Definição matemática das divisões do coto.....	63
5.3.2	Modelo 3D do coto em <i>Matlab</i>	65
5.3.3	Integração do <i>software</i> com o módulo de <i>hardware</i>	66
6	Testes e Validação	67
6.1	Testes Unitários.....	68
6.2	Testes de Integração	69
6.3	Testes de Usabilidade.....	70
6.3.1	Métodos Analíticos.....	71
6.3.2	Métodos Empíricos.....	73
7	Outras atividades desenvolvidas.....	75
7.1	<i>Smartherapy</i>	75
7.2	<i>FallSensing</i>	76
7.3	Website – Sensing Future Technologies	78
8	Conclusão e Trabalho Futuro	79
	Referências Bibliográficas	81
	Apêndice A Proposta de Estágio	85
	Apêndice B <i>SCRUM</i>	91
	Apêndice C Definição dos Casos de Uso	93
	Apêndice D Descrição das tabelas da Base de Dados	99
	Apêndice E <i>Mockups</i> do projeto	105
	Apêndice F Testes unitários	115
	Apêndice G Testes de Integração.....	121
	Apêndice H Testes de Usabilidade com Utilizadores	125
	Apêndice I Questionário.....	127

Índice de Figuras

Figura 2-1 - Componentes de uma prótese do membro inferior (Fonte: Meyer, 2012)	7
Figura 2-2 - Recolha de medidas com fita métrica (Fonte: Queiroz, 2008)	8
Figura 2-3 - Recolha de medidas com o paquímetro (Fonte: Queiroz, 2008)	9
Figura 2-4 - Ajuste do molde da prótese (Fonte: Queiroz, 2008)	9
Figura 2-5 - Processo tradicional de desenvolvimento de uma prótese (Fonte: Elaborada pelo autor)	10
Figura 2-6 - Digitalização 3D do coto do paciente (Fonte: Fourroux Prosthetics, 2014)	12
Figura 2-7 - Digitalização do resíduo da perna do paciente (Fonte: Fourroux Prosthetics, 2014)	12
Figura 2-8 - Processo de recolha de imagens com laser para desenvolvimento de um socket (Fonte: Elaborada pelo autor)	13
Figura 2-9 - F-Socket System da Tekscan (Fonte: Tekscan, 2012)	15
Figura 2-10 – FitSocket (Fonte: Kira, 2015)	16
Figura 2-11 - Estrutura do coto com mapa de cores evidenciando a resistência da pele (Fonte: Kira, 2015)	16
Figura 2-12- Digitalização do coto do paciente (Fonte: Fourroux Prosthetics, 2014)	17
Figura 2-13 - Socket impresso em impressora 3D (Fonte: Sengen, 2013)	18
Figura 2-14 - Representação da pressão exercida na planta dos pés (Fonte: Flávio, 2009)	20
Figura 2-15 - Interfaces do Software do F-Socket System (Fonte: Tekscan, 2012)	21
Figura 2-16 - ForeSite SS Mattress System (Fonte: XSENSOR, 2012)	21
Figura 2-17 - ForeSite SS System (Fonte: XSENSOR, 2012)	21
Figura 3-1 - Esquema do Socket Master (Fonte: Elaborada pelo consórcio)	26
Figura 3-2 - Esquema de uma estrutura lateral do Socket Master (Fonte: Elaborada pelo consórcio)	27
Figura 3-3 - Estrutura do Socket Master (Fonte: Elaborada pelo consórcio)	27
Figura 3-4 - Circuito eletrónico (Fonte: Elaborada pelo consórcio)	28
Figura 3-5 - Esquema a evidenciar a integração dos chips no Socket Master (Fonte: Elaborada pelo consórcio)	29
Figura 3-6 - Representação do indenter (Fonte: Elaborada pelo consórcio)	30
Figura 3-7 - Diagrama de Casos de Uso (Fonte: Elaborada pelo autor)	33
Figura 4-1 - Arquitetura Geral do Projeto (Fonte: Elaborada pelo autor)	35
Figura 4-2 – Modelo entidade-relacionamento (Fonte: Elaborada pelo autor)	37
Figura 4-3 - Modelo Conceptual de Dados (Fonte: Elaborada pelo autor)	41
Figura 4-4 - Modelo Físico de Dados (Fonte: Elaborada pelo autor)	42
Figura 4-5- Exemplo de responsividade – Bootstrap (Fonte: Hayaran, 2014)	47
Figura 4-6 - Funcionamento da tecnologia Ajax (Fonte: adaptada de Magento, 2014)	51
Figura 5-1 - Mockups desenhados à mão (Fonte: Elaborada pelo autor)	55
Figura 5-2 - Mockup da página principal da aplicação (Fonte: Elaborada pelo autor)	56
Figura 5-3 - Interface da página principal do sistema (Fonte: Elaborada pelo autor)	57
Figura 5-4 - Interface para inserção de paciente (Informações gerais) (Fonte: Elaborada pelo autor)	58
Figura 5-5 – Interface para inserção de paciente (Medidas básicas) (Fonte: Elaborada pelo autor)	58

Figura 5-6 – Interface para inserção de paciente (Propriedades mecânicas) (Fonte: Elaborada pelo autor).....	59
Figura 5-7 - Interface para listagem de pacientes (Fonte: Elaborada pelo autor)	60
Figura 5-8 - Interface para mostrar informações dos pacientes (Fonte: Elaborada pelo autor) ..	61
Figura 5-9 - Interface para gestão de consultas de um paciente (Fonte: Elaborada pelo autor) ..	62
Figura 5-10 - Particionamento do socket (Fonte: Elaborada pelo consórcio).....	63
Figura 5-11 - Visualização do modelo 3D do Socket com um ponto de pressão simulado (Fonte: Elaborada pelo autor)	65
Figura 6-1 - Modelo "V" (Fase de Desenvolvimento Vs. Fase de Testes) (Fonte: Aparecido, 2015)	67
Figura 7-1 - Versão trial implementada no website (Fonte: Elaborada pelo autor)	76
Figura 7-2 - Projeto FallSensing (Fonte: www.fallsensing.com).....	76
Figura 7-3 - Esquema a identificar os vários tipos de utilizadores e interações possíveis no sistema (Fonte: Elaborada pelo autor).....	77

Índice de Tabelas

Tabela 1-1 - Cronograma do Projeto	4
Tabela 2-2 - Síntese de projetos e produtos analisados	19
Tabela 2-3 - Síntese dos softwares analisados.....	22
Tabela 4-4 - Comparação das tecnologias para desenvolvimento web.....	45
Tabela 4-5 - Prefixos das colunas	48
Tabela 6-6 - Casos de Testes Unitários.....	69
Tabela 6-7 - Casos de Testes de Integração	70

Conceitos

Amputação transfemoral – Perda parcial do segmento corporal situado entre a articulação da anca e do joelho.

Amputação transtibial – Perda parcial do segmento corporal situado entre a articulação do joelho e do tornozelo.

Coto – É a parte do membro que permanece após uma amputação, ou seja, a parte residual do membro amputado.

Mockups – É um modelo de um projeto, usado para demonstração, avaliação de *design*.

Prótese – Componente artificial que tem por finalidade suprir necessidades e funções de indivíduos amputados.

Protésico – Técnico responsável pela confecção da prótese.

Socket – Componente artificial que é ajustado ao membro amputado, tratando-se da parte da prótese que encaixa no coto, também designado por encaixe da prótese.

Siglas e Acrónimos

AJAX – *Asynchronous Javascript and XML*

API – Application programming interface

BD – Base de dados

CAD/CAM – *Computer-Aided Design / Computer-Aided Manufacturing*

CIF – Classificação Internacional de Funcionalidade, Incapacidade e Saúde

CLR – *Common Language Runtime*

CSS – *Cascading Style Sheets*

DOM – *Document Object Model*

HTML – *Hyper Text Markup Language*

IU / UI – *User Interface*

JEE – Java Enterprise Edition

JSON – *JavaScript Object Notation*

MVC – *Model View Controller*

OMS – Organização Mundial de Saúde

ORM – *Object-relational mapping*

PHP – *Hypertext Preprocessor*

STL – *StereoLithography*

UC – *Use Case* (Caso de Uso)

URL – *Uniform Resource Locator*

WebGL – Web Graphics Library

XML – *eXtensible Markup Language*

2D – Duas dimensões

3D – Três dimensões

1 Introdução

1.1 Enquadramento

Existem diversas causas para a amputação dos membros inferiores, as razões mais comuns são de ordem vascular, como por exemplo, diabetes e tabagismo, para além destas, existem outras razões, como tumores, acidentes de viação ou laborais ou até mesmo por deficiência congénita. Em alguns países, a guerra continua a causar muitas amputações (Limites, 2011).

As amputações causadas por doenças vasculares atingem principalmente pessoas com uma faixa etária mais avançada, pois estas estão mais suscetíveis a doenças degenerativas, pertencendo uma grande parte destas amputações a idosos com diabetes.

As amputações traumáticas incidem principalmente em pessoas com idades mais jovens, principalmente do género masculino, pois encontram-se mais expostos a acidentes de viação e de trabalho (Nesi, 2005).

Na atualidade, desenvolver uma prótese de membros inferiores para um paciente amputado é um processo moroso e dispendioso. Fazer uma prótese e garantir que a mesma apresente um adequado nível de conforto para o paciente que a vai utilizar é um dos grandes problemas que existem. Muitas das vezes as próteses dos pacientes são desconfortáveis impossibilitando-os de conseguir andar com elas, o que por vezes limita a sua utilização a algumas horas por dia, pois se a usarem muitas horas seguidas, correm o risco de causar ferimentos no coto, podendo até mesmo criarem inflamações e outras complicações de saúde.

A presente dissertação tem por objetivo a obtenção do grau de Mestre, e encontra-se elaborada no âmbito da Unidade Curricular de Estágio, pertencente ao Mestrado de Informática e Sistemas, lecionado no Departamento de Engenharia Informática e de Sistemas, do Instituto Superior de Engenharia de Coimbra.

1.2 Objetivos

O projeto *Socket Master*, que constitui a principal atividade do estágio realizado, tem como principal objetivo o desenvolvimento de uma plataforma informática para a aquisição e apresentação de parâmetros físicos para a personalização de *sockets* dos membros inferiores. Este projeto está inserido no programa da União Europeia designado de Horizonte2020 (H2020), e conta com a participação de nove entidades, sendo uma delas a empresa *Sensing Future Technologies*.

Por forma a alcançar este objetivo principal, foram estipulados vários objetivos específicos de que se destacam, a implementação de um algoritmo para processamento dos dados recolhidos pelos sensores, a implementação de um *software* que mostre os dados recolhidos em tempo real, o desenvolvimento de uma interface para apresentação do formato 3D, permitindo a construção do *socket* otimizado para o paciente, e a integração do *software* e *hardware*.

1.3 Empresa de Acolhimento

1.3.1 Caracterização

A *Sensing Future Technologies* foi constituída em Dezembro de 2011, estando sediada na incubadora de empresas do Instituto Pedro Nunes desde 2012. É uma *startup* que está vocacionada para conceber, desenvolver e implementar dispositivos médicos tecnológicos para terapia e reabilitação física. Para tal, dispõe de uma equipa multidisciplinar, com experiência e conhecimentos nas áreas da engenharia mecânica, com uma sólida experiência em instrumentação, controlo de sistemas e robótica. Estas competências, aliadas a importantes sinergias desenvolvidas na área da saúde, permitem à *Sensing Future Technologies* definir especificações técnicas e desenvolver novos dispositivos médicos e aplicações direcionadas à área da Saúde (Sensing Future Technologies, 2016).

A empresa detém atualmente dois produtos comerciais que mantém em constante desenvolvimento: o *Smartertherapy* e o *PhysioSensing*. O *Smartertherapy* é uma aplicação informática, para registo clínico e que tem como base a Classificação Internacional de Funcionalidade, Incapacidade e Saúde da Organização Mundial de Saúde (OMS), também conhecida por CIF. Vocacionado também para a área de terapia e reabilitação física, o *PhysioSensing* é direcionado para a utilização por parte de profissionais de saúde, e consiste numa plataforma de pressões para treino e avaliação de equilíbrio e transferência de carga durante as atividades de reabilitação física.

1.3.2 Localização do Estágio

Este estágio foi efetuado nas instalações da *Sensing Future Technologies* que tem a sua sede em Coimbra, no Instituto Pedro Nunes – Bloco C, Rua Pedro Nunes, 3030-199 Coimbra, Portugal. O orientador na empresa foi o Engenheiro Luís Ferreira.

1.3.3 Metodologia Adotada pela Empresa

A metodologia de trabalho foi organizada de forma temporal através do programa de trabalho definido para este projeto. Tendo em conta a complexidade do *software*, optou-se por fazer o seu desenvolvimento assente numa metodologia ágil, dessa forma foi possível desenvolver o projeto de uma forma incremental e receber *feedback* por parte dos clientes ao longo do seu desenvolvimento. Assim a metodologia escolhida para o desenvolvimento foi genericamente o SCRUM (Apêndice B).

Tendo em conta a dimensão da equipa de desenvolvimento do projeto, foram feitas algumas alterações à metodologia SCRUM, assim, as *daily scrum meeting* foram efetuadas não diariamente, como a metodologia define, mas sim em três dias por semana, à segunda-feira, quarta-feira e sexta-feira. E estas pequenas reuniões informais cuja duração não excediam os cinco minutos tinham como principal objetivo responder às seguintes perguntas: o que tem sido feito, quais as próximas tarefas, e as dificuldades sentidas nas tarefas desenvolvidas.

Foram definidas dezoito *sprints* para o desenvolvimento do projeto. As *sprints* tinham a duração de duas semanas, dando início na segunda-feira e término na sexta-feira da segunda semana. As *sprint review meeting* estavam marcadas para as 16 horas da segunda sexta-feira da *sprint*, ou seja, coincidiam com o término da *sprint*. Nessa reunião eram apresentados em termos gerais os objetivos alcançados durante a *sprint* terminada, também nesta reunião eram debatidos os pontos que deviam ser melhorados no desenvolvimento do projeto, por forma a aumentar a produtividade nas *sprints* seguintes.

1.4 Plano de Trabalho

Para melhorar a organização do projeto, desenvolveu-se um plano de trabalho onde se descrevem todas as atividades (A) e as respetivas tarefas (T), apresentadas seguidamente:

A1 – Análise ao estado da arte

T1 – Pesquisa do processo de construção e colocação de próteses

T2 – Pesquisa sobre as meias de próteses atuais

T3 – Pesquisa de sensores e instrumentos de medida usados na colocação de próteses

T4 – Pesquisa de projetos e produtos que utilizem instrumentação na colocação ou uso de próteses

T5 – Pesquisa e estudo de *software* de instrumentação em próteses de membro inferior

A2 – Problema e levantamento de requisitos

T1 – Levantamento dos requisitos para a solução a desenvolver

A3 – Arquitetura do sistema: estudo e abordagem à tecnologia a utilizar

T1 – Pesquisa e estudo de ferramentas a utilizar e identificação de linguagem de programação de acordo com os requisitos identificados

T2 – Apresentação da arquitetura global do sistema

A4 – Desenvolvimento e implementação da solução

T1 – Desenvolvimento do algoritmo

T2 – Desenvolvimento de software e interfaces

T3 – Integração do software com o módulo de hardware

A5 – Teste do sistema

T1 – Definição de testes a realizar

T2 - Teste e verificação do sistema

1.4.1 Cronograma do Projeto

As tarefas descritas anteriormente, foram executadas de acordo com a calendarização que é apresentada na Tabela 1-1.

Tabela 1-1 - Cronograma do Projeto

		2015			2016						
		10	11	12	1	2	3	4	5	6	7
A1	Análise ao estado da arte										
	T1 Pesquisa do processo de construção e colocação de prótese										
	T2 Pesquisa sobre as meias de prótese atuais										
	T3 Pesquisa de sensores e instrumentos de medida usados na colocação de prótese										
	T4 Pesquisa de projetos e produtos que utilizem instrumentação na colocação ou uso de próteses										
	T5 Pesquisa e estudo de <i>software</i> de instrumentação em próteses de membro inferior.										
A2	Problema e levantamento de requisitos										
	T1 Levantamento dos requisitos para a solução a desenvolver										
A3	Arquitetura do sistema: estudo e abordagem à tecnologia a utilizar										
	T1 Pesquisa e estudo de ferramentas a utilizar e identificação de linguagem de programação de acordo com os requisitos identificados										
	T2 Apresentação da arquitetura global do sistema										
A4	Desenvolvimento e implementação da solução										
	T1 Desenvolvimento de algoritmo										
	T2 Desenvolvimento de <i>software</i> e interfaces										
	T3 Integração do <i>software</i> com o módulo de hardware										
A5	Teste do sistema										
	T1 Definição de testes a realizar										
	T2 Teste e verificação do sistema										

1.5 Estrutura da Dissertação

A dissertação apresenta uma estrutura de oito capítulos, começando na introdução e terminando na conclusão e trabalho futuro. Dos capítulos principais são excluídos as referências e os apêndices.

O presente capítulo é constituído pela introdução ao projeto, neste são definidos o enquadramento, os objetivos, a empresa de acolhimento, o plano de trabalho e na presente secção a estrutura da dissertação.

No segundo capítulo, é feita uma análise ao estado da arte, onde são apresentadas e comparadas as várias técnicas utilizadas na atualidade para o fabrico de próteses. Partindo do método tradicional, pouco rigoroso, e passando para métodos que requerem mais qualidade e mais segurança. É também feita uma descrição do *software* para análise de dados, enumerando alguns dos já existentes no mercado. Por fim, é feita uma abordagem dos produtos existentes no mercado para apoiar o desenvolvimento de próteses.

Segue-se o terceiro capítulo, constituído pela apresentação do projeto, bem como, o consórcio do qual faz parte. São também descritas as estruturas fulcrais ao projeto, sendo estas: *Socket Master*, sensores e o *Indenter*.

No quarto capítulo é apresentada a arquitetura do sistema, onde estão apresentados os requisitos do sistema, a conceção da base de dados, o modelo entidade relacionamento, os diagramas de casos de uso, onde se podem analisar as várias funcionalidades que o utilizador tem à disposição no sistema, e o modelo conceptual e físico de dados. Ainda é realizada uma análise às diferentes tecnologias possíveis para o desenvolvimento do projeto, enumerando as vantagens e desvantagens de cada uma delas e justificando a escolha da tecnologia eleita. Também são abordadas as diferentes *frameworks* e bibliotecas utilizadas.

Segue-se o quinto capítulo, relativo ao desenvolvimento e implementação do projeto. Inicia-se com uma explicação sobre os *mockups*, seguindo-se a apresentação e justificação das interfaces implementadas no projeto. Ainda é apresentada a estrutura tridimensional do coto do paciente, onde são expostas as equações matemáticas necessárias para obter um modelo tridimensional do coto, são também exibidas as potencialidades do *Matlab* para interpretar as equações matemáticas e representar as mesmas sobre uma vista a três dimensões.

O sexto capítulo refere-se aos testes e validação do *software*, onde se apresentam os testes e o planeamento de testes a que a aplicação foi sujeita, para que a implementação dos requisitos funcionais e não funcionais fosse verificada e validada. Detalhando os testes unitários, teste de integração e testes de usabilidade.

No sétimo capítulo, são descritas outras atividades que foram desenvolvidas durante o estágio, e que não se encontram diretamente relacionadas com o projeto *Socket Master*.

Termina-se com o oitavo capítulo, onde se apresentam as principais conclusões do trabalho desenvolvido durante o estágio, bem como algumas sugestões de trabalho a desenvolver no futuro.

2 Estado da Arte

Cada vez mais, o desenvolvimento de próteses é feito recorrendo a tecnologia e ao avanço científico. Porém, muitos destes produtos são maioritariamente fabricados de forma artesanal em pequenas empresas, empresas estas com poucos recursos, em termos de tecnologia, deixando desta forma o controlo de toda a tecnologia inerente nas mãos de grandes multinacionais.

2.1 Técnicas de construção de próteses

Uma prótese é uma componente artificial que tem por finalidade suprir necessidades e funções de indivíduos amputados constituída por vários componentes entre os quais: o encaixe, articulações, tubos de ligação, adaptadores e pé, conforme se pode visualizar na Figura 2-1. Todos estes componentes possuem características distintas de modo a formar um aparelho que cumpra as exigências do paciente (Meyer, 2012).

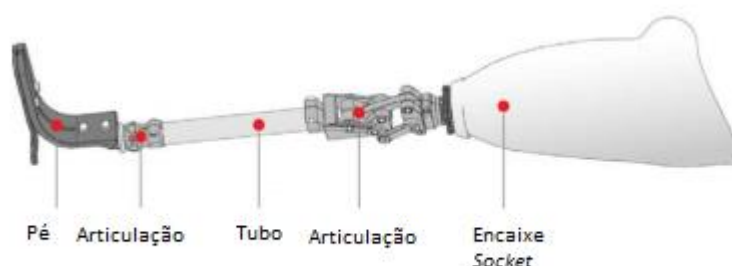


Figura 2-1 - Componentes de uma prótese do membro inferior (Fonte: Meyer, 2012)

O encaixe (*socket*) é considerado o componente mais importante da prótese, uma vez que é a parte que se encontra em contacto direto com o coto do paciente. Tem como função fixar a prótese ao coto, permitindo a transmissão de forças por forma a controlar os movimentos da perna.

Ao longo deste capítulo são abordadas várias técnicas utilizadas na atualidade para a construção de próteses.

2.1.1 Método tradicional

Nos dias de hoje, a grande maioria das próteses são feitas através de um método manual (Leal, 2011), que será descrito nesta secção, sendo também apresentadas as suas vantagens e desvantagens.

Neste método, o processo de desenho e construção de uma prótese tem início entre duas a três semanas após o paciente começar a usar uma meia elástica na perna amputada. Desde o primeiro molde até à obtenção da prótese final, existem várias fases, a saber: avaliação, medição e fundição, montagem da prótese de testes, avaliação de diagnósticos e alinhamentos.

A avaliação é a primeira fase no processo de fabrico de uma prótese, consistindo na recolha de informações do paciente. Durante esta fase, o protésico inquirir o paciente de forma a obter algumas informações, tais como: o seu peso, altura, atividades físicas que costumava fazer antes da amputação, os seus potenciais objetivos, e as suas limitações físicas ou psicológicas que possam interferir no desenvolvimento da prótese.

Após esta primeira fase de avaliação, começa a fase de medição. É aqui que o protésico recolherá algumas medidas do coto da perna do paciente, tais como, comprimento, diâmetro, entre outras, todas estas medições são alcançadas de forma manual recorrendo ao uso da fita métrica e do paquímetro artesanal.

O protésico faz as medições dos comprimentos da circunferência do coto em vários pontos, medindo também a altura do coto em relação ao joelho ou anca mediante seja uma amputação transtibial ou transfemoral respetivamente. Estas medições são feitas através do uso de uma fita métrica como é visível na Figura 2-2.



Figura 2-2 - Recolha de medidas com fita métrica (Fonte: Queiroz, 2008)

O paquímetro é utilizado para as medições do diâmetro de vários segmentos do coto como se fossem circunferências (Figura 2-3), no entanto note-se, que nenhuma curva do coto corresponde a uma circunferência.



Figura 2-3 - Recolha de medidas com o paquímetro (Fonte: Queiroz, 2008)

De seguida envolve-se o coto com película aderente para proteção das ligaduras de gesso, e procede-se à tiragem do molde da perna, fazendo-se em seguida o contra-molde, vulgarmente designado de molde negativo (Figura 2-4) (Marques, 2013).



Figura 2-4 - Ajuste do molde da prótese (Fonte: Queiroz, 2008)

Na fase de fundição, o protésico coloca uma meia de verificação feita de plástico transparente para determinar a exatidão do ajuste da prótese, sendo desta forma possível analisar onde o *socket* necessita ser modificado, de modo a proporcionar um melhor conforto ao paciente.

Por último vem a fase do alinhamento, sendo nesta fase que são feitas algumas alterações ao *socket* do paciente por forma a proporcionar-lhe um melhor conforto. Existem dois tipos de alinhamento: o estático e o dinâmico. O alinhamento estático consiste em verificar como o *socket* está a interferir com o paciente, analisando por exemplo os pontos onde o paciente faz mais pressão. O alinhamento dinâmico consiste em observar como se comporta o paciente enquanto caminha com a prótese, fazendo uma análise da marcha do paciente. A fase de alinhamento é uma fase morosa e requer que os pacientes e os protésicos estejam juntos em

diversas consultas. É nesta fase que irão ser feitos ajustes ao *socket* até a obtenção da prótese final (estabilizada) (Amputee Coalition, 2014).

Como se pode verificar na Figura 2-5, o processo tradicional de desenvolvimento de uma prótese inicia-se na fase da avaliação, onde são recolhidas algumas informações (informações genéricas) do paciente, após esta fase, inicia-se a medição, precedida da fundição, e no fim é feito o alinhamento da prótese. Após este processo é verificado se determinada prótese é confortável para o paciente, em caso afirmativo o processo termina, e obtém-se deste modo a prótese do paciente, em caso negativo, volta-se novamente à fase de medição iniciando-se um novo ciclo no processo.

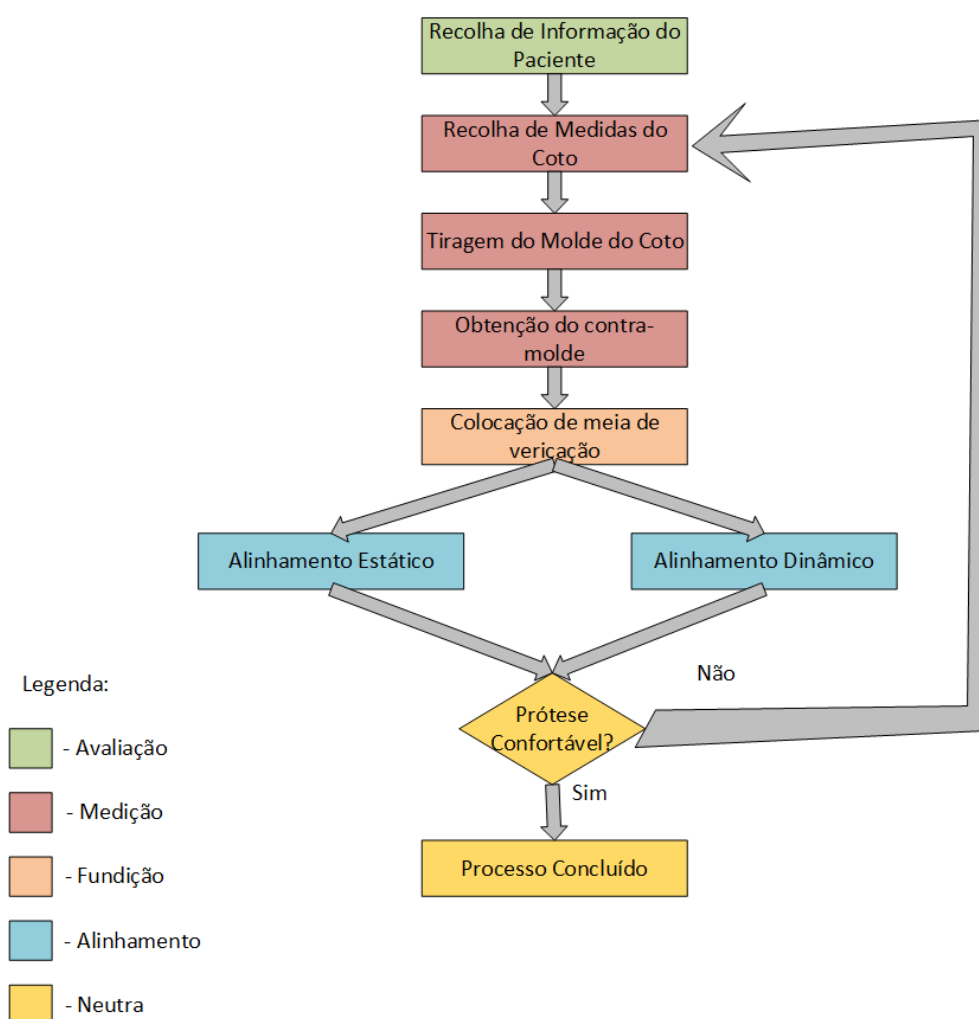


Figura 2-5 - Processo tradicional de desenvolvimento de uma prótese (Fonte: Elaborada pelo autor)

Este processo pode ser iterado N vezes até à obtenção da prótese ideal. É um processo que exige muito tempo quer do protésico quer do paciente, conduzindo muitas vezes à desmotivação deste último.

A motivação do paciente é um fator fulcral, pois a construção da prótese envolve uma intensa participação e tempo despendido por parte do paciente, sendo necessário que este se encontre motivado para investir esse tempo (Leal, 2011). Também é importante reter que a habilidade e

experiência do protésico é um fator determinante no processo tradicional, certamente um protésico com maior experiência não necessitará de repetir o processo tantas vezes como um protésico inexperiente.

Saliente-se que o método tradicional do processo de produção de próteses revela algumas questões problemáticas que, por consequência, se traduzem em desvantagem neste tipo de processo. Uma vez que as medições do coto, como a altura e comprimento, não são suficientes para representar a forma tridimensional do coto, uma boa prótese depende quase exclusivamente da habilidade do protésico, tendo em conta as várias etapas exigidas para o processo de produção, torna-se um processo demorado. Devido ao grau elevado de incerteza associado às medições, a possibilidade de efetuar ajustes à prótese existe mas, poderá causar incómodo ao paciente.

Em suma, no processo tradicional, os dados são recolhidos de uma forma manual, recorrendo a instrumentação como por exemplo a fita métrica e o paquímetro. Estes instrumentos de medição, permitem obter dados relativos ao coto do paciente, entre os quais, o comprimento do coto, distância desde a extremidade do coto até ao joelho, distância desde o joelho até ao chão e diâmetros da barriga da perna. O processo de desenvolvimento do *socket* é alcançado recorrendo a moldes de gesso, tira-se o molde do coto do paciente, e de seguida tira-se o contra molde, por forma a ficar com a forma do coto do paciente para se trabalhar a prótese a desenvolver. No método tradicional, quer a aquisição de dados quer o fabrico da prótese manualmente estão suscetíveis a erros, sendo este um dos principais pontos negativos deste método.

2.1.2 Método Digital

Atualmente as pessoas exigem mais qualidade e segurança nos tratamentos a que são submetidas, não se querendo submeter a processos passíveis de biocompatibilidade, com custos e tempos de produção elevados até conseguir obter uma prótese final, como aqueles inerentes aos métodos tradicionais. A digitalização 3D poderá dar uma resposta mais conveniente e expedita a esta necessidade. Os sistemas CAD/CAM utilizam esta tecnologia para construir o modelo tridimensional do coto do paciente.

A digitalização tridimensional é uma tecnologia que consiste em recolher dados dos objetos físicos e através de *software*, obter curvas, texturas e detalhes bastantes precisos, permitindo obter dados da superfície do coto de um paciente, sem a necessidade de contato direto, conseguindo-se assim gerar um modelo virtual. Neste modelo virtual é possível trabalhar o *socket* do paciente de acordo com as suas necessidades específicas, eliminando assim o processo de moldagem com gesso (Brendler, Muller, Silva, & Teixeira, 2014).

Através da digitalização 3D é possível determinar as medidas e a forma do coto do paciente, existindo também um modelo para trabalhar virtualmente na construção do *socket*. Assim é possível de uma forma rápida e económica começar a trabalhar no *socket* para o paciente.

A utilização de imagens 3D captadas por laser é assim uma forte alternativa ao método tradicional. Através desta tecnologia, é possível melhorar o processo de produção do *socket*

para próteses, uma vez que as imagens captadas são baseadas nas características morfológicas de cada paciente.

Com esta tecnologia é possível obter um modelo tridimensional do coto do paciente (Figura 2-6) e calcular todas as dimensões do mesmo de uma forma rápida e exata. Através deste modelo é possível criar um *socket* de uma prótese virtual (Curralo, Queijo, & Rocha, 2012).



Figura 2-6 - Digitalização 3D do coto do paciente (Fonte: Fourroux Prosthetics, 2014)

Alguns dos benefícios desta técnica são: rapidez (aquisição rápida da forma do coto); precisão (os dados recolhidos são precisos); consistência (todas as medidas recolhidas são consistentes, não estão dependentes por exemplo do nível de maturidade do protésico); limpeza (é um processo limpo, através deste método não é necessário a utilização de gesso) (Figura 2-7) (Fourroux Prosthetics, 2014).



Figura 2-7 - Digitalização do resíduo da perna do paciente (Fonte: Fourroux Prosthetics, 2014)

O processo de desenvolvimento de *socket* através deste método acaba por ser quase sequencial, como é visível na Figura 2-8.

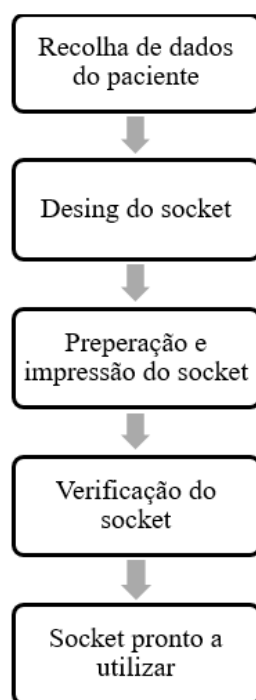


Figura 2-8 - Processo de recolha de imagens com laser para desenvolvimento de um socket (Fonte: Elaborada pelo autor)

Na recolha de dados, são adicionalmente feitas algumas questões ao paciente, nomeadamente: idade, motivação, exercício físico, atividade profissional, entre outras. Também nesta fase são feitas marcas no coto para iniciar o processo de digitalização. No processo de *design* do *socket*, é possibilitada a sua visualização e são feitos alguns ajustes, sendo nesta fase que o técnico analisa a veracidade das imagens captadas, e elimina ruído das amostras, caso este exista. Neste ponto o técnico consegue ajustar os pontos de pressão, assim como melhorar a forma e as curvas do *socket*. A preparação e impressão é a fase em que o técnico inspeciona todo o modelo, corrigindo algumas imperfeições, e por fim, gera o ficheiro com o molde tridimensional para enviar para a impressora 3D. O processo de verificação consiste em analisar o modelo impresso, removendo algumas imperfeições. Por fim, o *socket* está pronto a ser utilizado, no entanto ainda podem ser feitos alguns ajustes para uma correta ligação entre o *socket* e o paciente.

Este processo tem, para além das vantagens já mencionadas, ainda a possibilidade de guardar, em formato digital, o modelo tridimensional do coto do paciente para futuras utilizações. Enquanto no processo tradicional se for necessário trabalhar novamente no molde será obrigatório refazer o molde manualmente. Também a geometria do coto e as dimensões são mais precisas recorrendo a imagens 3D. Adicionalmente, todos os passos necessários ao desenvolvimento do *socket* podem ser feitos virtualmente, não sendo necessário a presença do paciente (Materialise, 2014).

Outra vantagem deste processo, é o facto do fabrico do *socket* poder ser remetido para uma fábrica própria. A clínica molda o *socket* através do *software* e envia um ficheiro para a fábrica responsável pela sua fabricação, assim a clínica reduz custos pois não necessita de ter

todo o equipamento requerido para a produção do *socket*, nem necessita de técnicos especializados na fabricação do mesmo (Caldwell & Sanders, 2013).

Em síntese, esta técnica baseia-se na digitalização recorrendo a laser para obtenção da forma e das dimensões do coto do paciente, o *socket* do paciente é trabalhado posteriormente com a ajuda de um *software* (CAD), o processo de fabrico da prótese é realizado através de impressões 3D (CAM). Esta técnica comparativamente à técnica tradicional é melhor, pois existe uma maior precisão dos dados recolhidos, o processo de construir o *socket* é conseguido através de um *software* para manipulação de modelos 3D, o que permite ao protésico trabalhar o modelo sem ser necessária a presença do paciente. Ainda assim esta técnica não é a ideal, uma vez que a forma e as dimensões do coto do paciente são obtidas partindo de uma posição estática, não tendo em conta pressões exercidas entre o coto e o *socket*.

2.2 Meias Prostéticas

A utilização de uma meia prostética para o membro inferior amputado é fundamental para a colocação de uma prótese, esta visa principalmente, proteger o coto da fricção existente no contacto.

Com vista a aumentar o conforto de utilização da prótese, existem alguns tipos de meias prostéticas, entre os quais se destacam, meias de compressão do coto e meia prostéticas de gel. A meia prostética de compressão do coto é constituída por tecido altamente elástico, permitindo um ajuste adequado ao membro amputado, eliminando desta forma, o atrito e a pressão exercida pela prótese, evitando o surgimento de feridas. A meia prostética de gel, também designada de meia de silicone, é um produto ortopédico, constituída por tecido de alta elasticidade, revestida no seu interior por gel polímero. Esta auxilia no alívio de dores e na eliminação da pressão e atrito, possibilitando um bom ajuste ao coto, devido ao óleo mineral medicinal que se encontra na camada de gel, tendo como principal função hidratar e proteger a pele (Araújo, 2010).

É muito importante referir que na parte inicial do projeto, pensou-se que o desenvolvimento do projeto se baseava na criação de um *software* para visualização de meias prostéticas em 3D. No entanto com o progresso do projeto verificou-se que o termo “meia prostética” não seria adequado, tendo em conta a definição mencionada no parágrafo anterior. Assim, restabeleceu-se que o objetivo do projeto se basearia na criação de um *software* para visualização do *socket* em 3D. Pois, *socket* é o componente artificial que é ajustado ao membro amputado, tratando-se da parte da prótese que encaixa no coto, também designado por encaixe da prótese.

Inicialmente foram definidas várias tarefas, entre elas: "Pesquisa de sensores e instrumentos de medida usados na colocação de próteses" (T3) e "Pesquisa de projetos e produtos que utilizem instrumentação na colocação ou uso de próteses" (T4), no entanto, com o decorrer do projeto, verificou-se que o seu planeamento não foi o mais adequado, uma vez que, o que realmente interessa é efetuar uma pesquisa de projetos e produtos existentes no mercado que utilizem sensores e instrumentos de medida para a conceção de *sockets*. A colocação das

próteses é feita manualmente pelo paciente, podendo o protésico auxiliá-lo nas primeiras vezes.

2.3 Projetos e Produtos no mercado

Atualmente no mercado já existem bastantes produtos que recorrem à tecnologia CAD/CAM para trabalhar o *socket* através de um processo virtual recorrendo à utilização de computador. Os processos base para recolha de dados e produção da prótese são os já mencionados, porém as combinações entre eles podem ser variadas. Uns obtêm dados biomecânicos como a espessura e elasticidade da pele, outros a forma e o tamanho do membro amputado, outros a rigidez ou pressão. A forma de obtenção dos dados também pode ser variada. Apesar de recorrer normalmente a laser, existem produtos que complementam esta recolha com ultrassons, ressonância magnética ou utilizando sensores. Serão, então abordados de seguida alguns produtos existentes no mercado que auxiliam o processo de desenvolvimento de próteses recorrendo a novas tecnologias.

2.3.1 F-Socket System

O F-Socket System (Figura 2-9) é um produto já existente no mercado, sendo pertença da empresa Tekscan, e consiste num dispositivo capaz de adquirir a pressão sentida entre o *socket* e o coto do paciente, permitindo assim construir uma imagem onde é possível analisar o mapa de cores relativo às pressões exercidas (Tekscan, 2012).



Figura 2-9 - F-Socket System da Tekscan (Fonte: Tekscan, 2012)

A aquisição dos valores de pressão é efetuada com recurso a um sensor portátil, o protésico necessita colocar este sensor em determinado ponto do *socket* para recolher os valores de pressão.

Este produto apresenta como vantagens o facto de possibilitar a aquisição de valores de pressão durante a marcha do paciente, posteriormente estes dados podem ser analisados e feitas correções no *socket* final. A grande desvantagem deste produto é os valores de pressão serem recolhidos através de um dispositivo portátil, sendo necessário colocá-lo manualmente entre o coto e o *socket*. A cada aquisição os valores de pressão não são recolhidos em torno da

totalidade do coto, mas sim numa determinada zona, sendo necessária a recolocação do dispositivo para aquisição numa outra zona.

2.3.2 MIT lab's FitSocket

O FitSocket é um projeto desenvolvido pelo grupo de biomecatrónica dos laboratórios do *Massachusetts Institute of Technology* (MIT) (Figura 2-10). Este projeto consiste numa ferramenta não invasiva capaz de adquirir propriedades mecânicas do membro residual do paciente. Através desta técnica é possível saber quais as partes do coto mais sensíveis, as mais resistentes, determinar a percentagem de massa gorda, massa muscular, analisar a distância da superfície até ao osso, proporcionando um desenvolvimento de próteses com maior conforto a cada paciente. Esta técnica visa melhorar o conforto e diminuir o medo da utilização de próteses feitas pelas técnicas tradicionais.

Este aparelho permite adquirir as propriedades biomecânicas e de rigidez do coto do paciente, permitindo através dos dados recolhidos construir um protótipo virtual do coto do paciente de uma forma bastante rápida. Neste modelo poder-se-ão analisar quais as partes do resíduo do membro amputado mais resistentes, e as mais sensíveis, como ilustra Figura 2-11, sendo assim possível construir um *socket* tendo em conta os pontos onde o paciente exerce maior pressão (Kira, 2015).

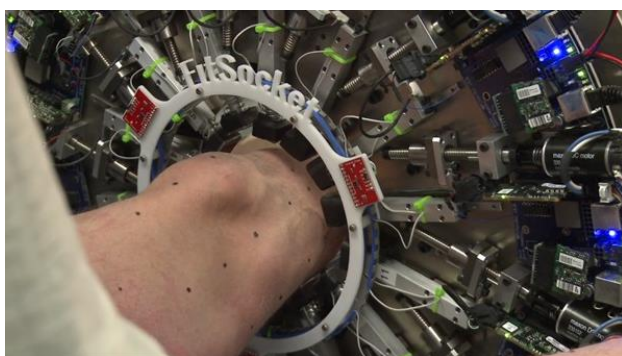


Figura 2-10 – FitSocket (Fonte: Kira, 2015)



Figura 2-11 - Estrutura do coto com mapa de cores evidenciando a resistência da pele (Fonte: Kira, 2015)

Foi realizado um estudo com seis pacientes por forma a testar este aparelho e se realmente seria um método de sucesso. Estes seis pacientes realizaram algumas sessões nos laboratórios do MIT e correu conforme o previsto, contudo o laboratório quer expandir o seu número de casos de estudo, com mais pacientes e condições de teste mais rigorosas.

Este projeto apresenta algumas vantagens, entre as quais o facto de adquirir a pressão e as propriedades mecânicas do coto do paciente, como desvantagens salienta-se o facto de as medições serem feitas numa posição que não a da utilização da prótese no dia-a-dia, este sistema é bastante complexo e tem custo elevados.

2.3.3 Prosthetic Socket Design

Este produto permite criar um *socket* em poucas horas, digitalizando o coto do paciente, e em seguida através da tecnologia CAD obter uma imagem tridimensional representando a forma do coto do paciente. Partindo deste objeto 3D é fabricado um *socket* personalizado para o paciente. O paciente deverá então experimentar este *socket* fornecendo *feedback* sobre o mesmo para que possam ser elaboradas melhorias.

O sistema é baseado na digitalização do coto através de um laser (Figura 2-12), e evidência alguns benefícios, tais como: a rapidez, a exatidão dos dados recolhidos, a consistência e limpeza do processo. Como desvantagens salienta-se o facto de os valores de pressão não serem tidos em conta para a conceção do *socket*, desta forma o *socket* não tem em conta as deformações ocorridas no coto aquando da utilização da prótese.



Figura 2-12- Digitalização do coto do paciente (Fonte: Fourroux Prosthetics, 2014)

2.3.4 Variable impedance prosthetic socket

Variable impedance prosthetic socket é um projeto desenvolvido nos laboratórios dos MIT, e os seus principais objetivos são o desenvolvimento de um *socket* utilizando processos de CAD/CAM, recolha de pontos de pressão e recorrendo a ressonâncias magnéticas (Sengeh, 2013). O *socket* é construído fisicamente recorrendo a uma impressora 3D, visível na Figura 2-13.

É baseado na junção de três técnicas, em primeiro lugar é feita a digitalização através de um laser, de seguida através do produto “F-Socket System”, são adquiridos vários pontos de pressão ao redor do coto do paciente, por fim são feitas ressonâncias magnéticas ao coto, para ter em consideração a distância da pele ao osso nos diferentes pontos do *socket*.



Figura 2-13 - Socket impresso em impressora 3D (Fonte: Sengeh, 2013)

Este projeto tem em consideração vários fatores relacionados com o coto do paciente, entre eles, a forma do coto adquirida pela digitalização do mesmo, os pontos de pressão recolhidos pelo produto “F-Socket System” e ainda tem em consideração a posição do osso no coto recorrendo a ressonâncias magnéticas.

Como vantagens apresenta o facto de recolher dados de três formas diferentes, e todas elas importantíssimas para conceber um *socket* adaptado ao coto do paciente, no entanto apresenta algumas desvantagens entre as quais encontra-se novamente o facto dos valores de pressão serem adquiridos recorrendo a um dispositivo portátil, impossibilitando a aquisição dos valores de pressão na totalidade do coto do paciente.

2.3.5 Síntese dos projetos e produtos analisados

Na Tabela 2-2 estão sintetizados alguns projetos e produtos, bem como as suas descrições.

Tabela 2-2 - Síntese de projetos e produtos analisados

Nome	Descrição
F-Socket System	Consiste na aquisição de vários pontos de pressão entre o coto e a prótese do paciente com recurso a um sensor portátil, o protésico necessita colocar este sensor em determinado ponto do <i>socket</i> para recolher os valores de pressão.
MIT lab's FitSocket	Consiste na aquisição das propriedades mecânicas do coto do paciente recorrendo à técnica da indentação. Após a aquisição é construído um modelo tridimensional do coto do paciente, para manipulação do <i>socket</i> virtualmente.
Prosthetic Socket Design	Consiste na digitalização do coto do paciente, com recurso a um laser. Após a aquisição é construído um modelo tridimensional do coto do paciente, para manipulação do <i>socket</i> virtualmente.
Variable Impedance Prosthetic Socket	É baseado na junção de três técnicas, em primeiro lugar é feita a digitalização através de laser, de seguida através do produto “F-Socket System”, são adquiridos vários pontos de pressão ao redor do coto do paciente, por fim são feitas ressonâncias magnéticas ao coto, para ter em consideração a distância da pele ao osso nos diferentes pontos do <i>socket</i> .

2.4 Software de instrumentação para análise de dados

Após uma pesquisa exaustiva de *softwares* que permitam analisar os dados e fornecer um mecanismo para a sua visualização de uma forma mais cómoda para o utilizador, encontraram-se alguns, sendo que apenas um está diretamente relacionado com o fabrico de *sockets*, o F-Socket System, descrito de seguida.

2.4.1 Body Track

Body Track é um produto pertença da Vista Medical, e consiste num tecido inteligente, com sensores de pressão, o qual pode ser instalado em camas, para obtenção dos pontos de pressão, aquando do paciente deitado, pode ser utilizado num tapete de golfe, por forma a analisar os picos de pressão aquando de uma tacada por parte de um jogador.

Este produto dispõe de um *software* para análise e visualização dos dados adquiridos.

2.4.2 Footwork PRO

Esta ferramenta consiste em analisar a pisada de um indivíduo, conseguindo mapear a pressão exercida na planta do pé através de cores (Figura 2-14), mostrando uma imagem bidimensional da sola do pé, evidenciando os pontos onde a pressão exercida é superior, bem como os pontos onde ocorre menor pressão (Shopping Ortopédico, 2012).

Esta é uma ferramenta utilizada principalmente por ortopedistas, desta forma conseguem-se, obter os dados de um paciente através de uma plataforma de pressão, e guardar os seus dados sob a forma digital, ficando estas informações sempre associadas ao paciente.

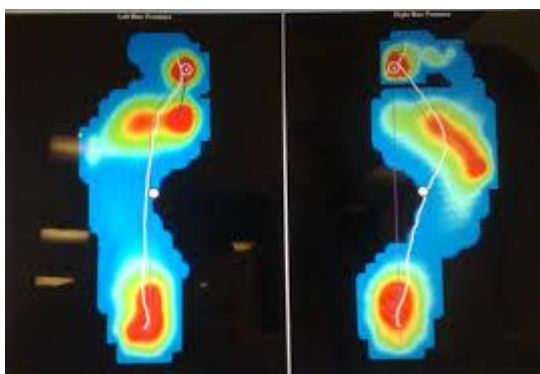


Figura 2-14 - Representação da pressão exercida na planta dos pés (Fonte: Flávio, 2009)

2.4.3 F-Socket System

De acordo com o fabricante as funcionalidades do *software* incluem: visualização 2D e 3D em tempo real das pressões exercidas; visualização da área de contacto, média, pico e centro de pressão; visualização *frame a frame*; comparações do antes e após consultas; medição de distâncias entre dois pontos; análise de regiões específicas, importação e exportação de ficheiros de vídeo.

Na Figura 2-15 pode observar-se algumas interfaces gráficas deste produto.

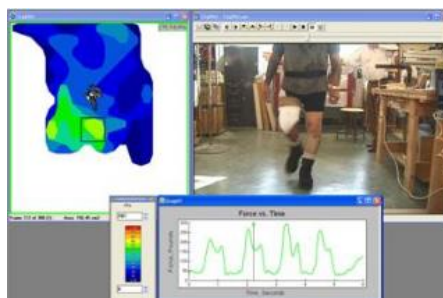


Figura 2-15 - Interfaces do Software do F-Socket System (Fonte: Tekscan, 2012)

2.4.4 Fore Site SS System

Fore Site SS System é um produto desenvolvido pela XSENSOR, esta é uma empresa que desenvolve produtos para analisar pressões. Analisando o seu portfólio encontraram-se dois produtos, o ForeSite SS System, que analisa a pressão exercida por um paciente com mobilidade reduzida na sua cadeira de rodas (Figura 2-17), e o ForeSite SS Mattress System (Figura 2-16) que consiste em monitorizar a distribuição de cargas enquanto o paciente está deitado. Ambos os sistemas recorrem a imagens 2D (XSENSOR, 2012).



Figura 2-17 - ForeSite SS System (Fonte: XSENSOR, 2012)



Figura 2-16 - ForeSite SS Mattress System (Fonte: XSENSOR, 2012)

Os sensores destes produtos encontram-se integrados em tecidos inteligentes, são bastantes finos, flexíveis e leves, o que permite o seu ajuste fácil à cadeira de rodas, podendo medir-se quer a pressão exercida quando se está sentado. Através do *software*, é possível visualizar em tempo real o mapa de pressão com esquema de cores a evidenciar os pontos com maior e menor pressão.

2.4.5 Tune

Tune é um produto desenvolvido pela Kinematix, uma empresa sediada na cidade do Porto. O produto é baseado em sensores de pressão integrados numa palmilha que depois deve ser colocada dentro dos sapatos dos utilizadores. Os valores de pressão são recolhidos ao longo da caminhada ou corrida, e é possível visualizar através de uma aplicação *mobile* os dados recolhidos.

2.4.6 Síntese dos *softwares* analisados

Na Tabela 2-3 encontra-se uma síntese de vários *softwares* analisados.

Tabela 2-3 - Síntese dos *softwares* analisados

Nome	Descrição
Body Track	Este produto não está diretamente relacionado com as próteses, no entanto é um produto que consiste num tecido inteligente, com sensores de pressão, podendo ser instalado em camas.
Footwork PRO	Este produto não está diretamente relacionado com as próteses, no entanto o seu <i>software</i> permite a visualização da pressão exercida na planta dos pés.
F-Socket System	A nível de <i>software</i> , permite a visualização das pressões exercidas recorrendo a um mapa de cores.
Fore Site SS System	Este produto não está diretamente relacionado com as próteses, no entanto permite a visualização dos pontos de pressão, neste caso os sensores encontram-se integrados em tecidos inteligentes.
Tune	Este produto não está diretamente relacionado com as próteses. Este consiste numa palmilha com sensores, esta palmilha é colocada nos sapatos dos utilizadores, e os dados da pressão exercida são recolhidos e visualizados num <i>software</i> .

Após uma análise às técnicas e metodologias utilizadas atualmente para o fabrico de *sockets*, conclui-se que a pressão exercida entre o coto do paciente e o *socket* na metodologia

tradicional é analisada apenas no processo de afinação da prótese, e é feito de uma forma manual, ou seja, através do tato do protésico e da sua experiência.

Alguns métodos recorrem à tecnologia, adquirem valores de pressão, mas não nas posições mais realísticas, ou não na totalidade do coto. Desta forma o projeto *Socket Master* tem como objetivo a aquisição de valores de pressão entre o coto e o *socket* em posições normais do dia-a-dia de um paciente.

3 Apresentação do Projeto

Neste capítulo é apresentado o projeto no qual o estágio foi desenvolvido. Como já foi mencionado, o estágio visa o desenvolvimento de uma plataforma informática para aquisição e apresentação de parâmetros físicos para personalização de *sockets* do membro inferior, bem como, de um mecanismo para fazer a gestão dos vários pacientes registados no sistema.

Como já foi referido, a pressão exercida entre o coto do paciente e o *socket* nas metodologias tradicionais, é analisada no processo de afinação de próteses, recorrendo ao tato do protésico e à sua experiência. Alguns métodos recorrem à tecnologia, baseados em valores de pressão, no entanto estes não são adquiridos nas posições mais normais do dia-a-dia de um paciente, ou não na totalidade do coto. Com o intuito de colmatar esta lacuna existente nas técnicas analisadas, este projeto visa desenvolver uma rede de sensores para integrar na prótese, garantindo-se que os valores de pressão são tidos em conta, logo no início do processo de desenvolvimento da prótese.

3.1 Consórcio do *Socket Master*

Este projeto é parte integrante do Consórcio *Socket Master*, um projeto europeu enquadrado no tema Horizonte 2020 – (Tópico: ICT-02-2014 - *Smart System Integration*), cujo consórcio é constituído por oito entidades, sendo uma delas a empresa de acolhimento deste estágio. Como é prática comum neste tipo de projetos, foram definidas várias tarefas para cada uma das entidades, sendo que a *Sensing Future Technologies* ficou responsável pelo desenvolvimento de uma plataforma informática para aquisição e apresentação de dados físicos de *sockets* para amputados do membro inferior, bem como, integrar o sistema de aquisição desenvolvido por outra empresa nesta mesma plataforma.

O projeto é constituído por sete atividades que se descrevem de seguida:

- Gestão do projeto, atividade que se encontra diretamente interligada à organização reuniões, avaliação e acompanhamento da correta evolução do projeto;
- Desenvolvimento do *Socket Master*, esta será uma estrutura onde serão posteriormente integrados os sensores;
- Conceção dos sensores para posterior integração no *Socket Master*;
- Desenvolvimento de um módulo para aquisição de dados diretamente a partir dos sensores;
- Análise das propriedades mecânicas e desenvolvimento de um *indenter* para adquirir automaticamente estas propriedades do coto dos pacientes;
- Desenvolvimento de uma plataforma informática para fazer a gestão dos vários pacientes, aquisição e apresentação dos dados recolhidos pelo *Socket Master* e pelo *Indenter*;

- Testes ao produto desenvolvido, com no mínimo cinquenta pacientes piloto.

De forma a melhor compreender as tarefas do estagiário neste projeto, é fundamental perceber a estrutura do projeto, pelo que de seguida são apresentados o *Socket Master*, a malha de sensores e o *Indenter*.

3.1.1 *Socket Master*

A estrutura principal deste projeto consiste num dispositivo de medição, designado de *Socket Master*. Esta estrutura deve ser capaz de se ajustar às diferentes formas dos cotos dos pacientes, devendo possibilitar a integração de uma malha de sensores e conter mecanismos automáticos para facilitar os ajustes.

Na Figura 3-1 encontra-se um esquema geral da estrutura deste equipamento que é composto por quatro estruturas laterais, e dois anéis (um com raio menor para a parte inferior, e outro com raio maior para a parte superior). Na figura à esquerda encontra-se a representação transversal da estrutura, e à direita encontra-se a representação numa vista de topo.

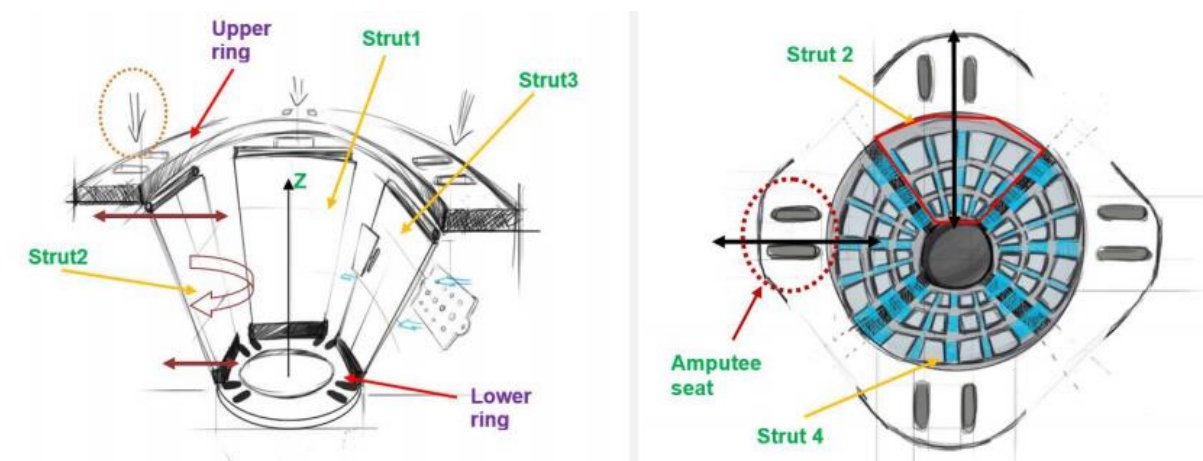


Figura 3-1 - Esquema do Socket Master (Fonte: Elaborada pelo consórcio)

Cada uma das estruturas laterais do *Socket Master* será composta por três *Pad array's* (Figura 3-2), e cada um destes será composto por três circuitos eletrónicos com vários sensores. Por cada lado do *Socket Master*, existirão 9 circuitos (3×3).

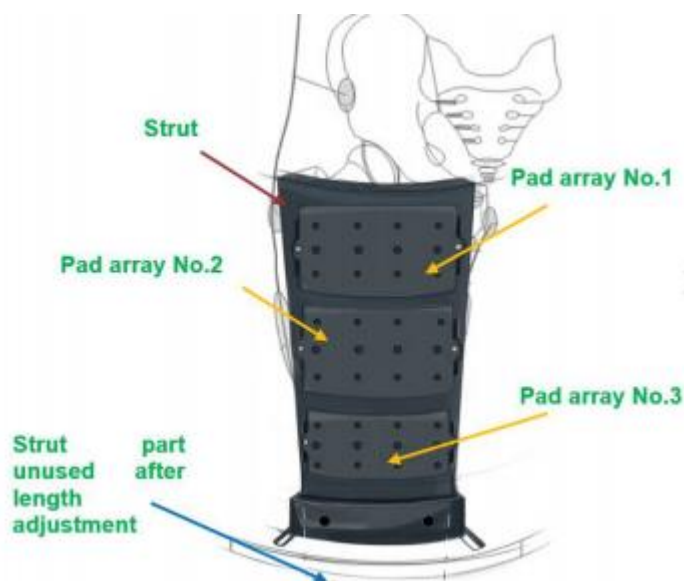


Figura 3-2 - Esquema de uma estrutura lateral do Socket Master (Fonte: Elaborada pelo consórcio)

A possibilidade de ajustar a distâncias destas quatro estruturas em relação ao coto do paciente, torna possível ajustar o *Socket Master* em termos de espessura. Esta característica permite tonar o *Socket Master* num instrumento de medição universal, sendo assim é possível, ajustá-lo aos diferentes cotos de pacientes. Também os Pad array's podem ser ajustados, permitindo colocá-los em contacto direto com o coto do paciente.

Na Figura 3-3, pode analisar-se a estrutura e a forma estipulada para o desenvolvimento do *Socket Master*.



Figura 3-3 - Estrutura do Socket Master (Fonte: Elaborada pelo consórcio)

3.1.2 Malha de Sensores

Outro ponto fulcral para o sucesso deste projeto é a construção dos sensores, a sua correta instalação e o módulo para fazer a recolha de dados. Na Figura 3-4 pode observar-se através de um diagrama como será composto cada um dos chips eletrónicos. Note-se que para o total do *socket* são necessários trinta e seis chips eletrónicos ($9 \times 4 = 36$), uma vez que cada um dos quatro lados do *Socket Master* inclui nove chips.

Cada um destes circuitos eletrónicos será composto por cinco sensores de pressão e um sensor de temperatura. Os sensores de pressão têm uma gama de medição entre os 0 e os 400 kPa, e o sensor de temperatura entre os 0 e os 50 °C. O circuito será revestido com um material polimérico para que a distribuição de tensões seja uniforme nos vários sensores de pressão.

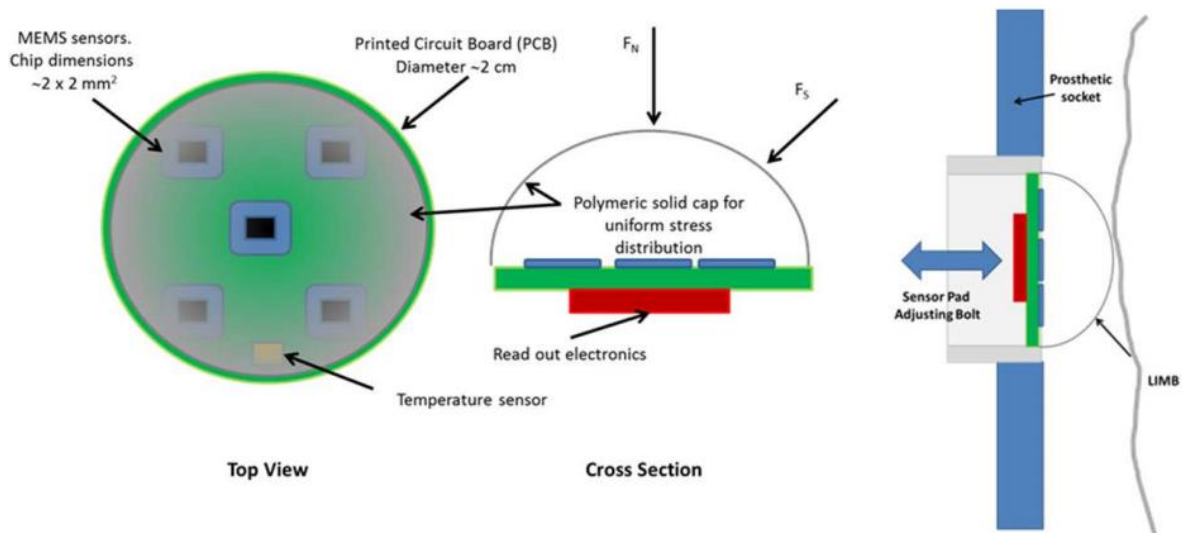


Figura 3-4 - Circuito eletrónico (Fonte: Elaborada pelo consórcio)

Na Figura 3-5 apresenta-se o processo de integração dos vários sensores no *Socket Master*. Como se pode verificar, a integração dos vários sensores no *Socket Master* terá uma espessura ideal entre os cinco e os dez milímetros.

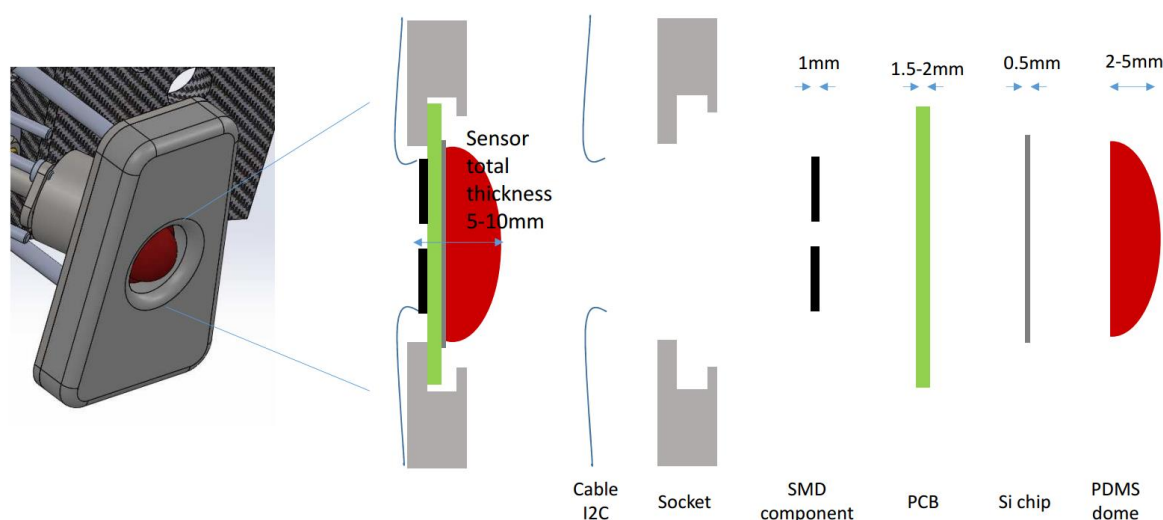


Figura 3-5 - Esquema a evidenciar a integração dos chips no Socket Master (Fonte: Elaborada pelo consórcio)

Para fazer a aquisição de dados será utilizado um controlador da marca Raspberry¹ pelo facto de ter um conjunto de características de interesse, nomeadamente ter um sistema operativo instalado, possuir placa de rede para comunicar com os sensores e o servidor, e ter a possibilidade de instalação de um cartão SD possibilitando a criação um sistema para registo de erros. O *firmware* desenvolvido para este dispositivo permitirá assim: comunicar com todos os sensores integrados no *Socket Master*; iniciar o processo de aquisição de dados; enviar os dados recolhidos para o servidor; executar a calibração e o diagnóstico de cada um dos sensores; possibilitar a existência de um sistema de *backups*; possibilitar um sistema para registo de erros em ficheiros de texto.

3.1.3 Indenter

Por forma a determinar uma distribuição de pressão mais precisa e otimizada, é necessário conhecer as propriedades biomecânicas do coto do paciente, mais propriamente conhecer os valores correspondentes à elasticidade da pele, elasticidade da gordura e elasticidade do músculo. Para determinar estes valores vai ser utilizado o método de indentação, também conhecido por teste de indentação instrumentada ou nano indentação, cuja finalidade é investigar as propriedades mecânicas de pequenos volumes de material, pressionando para tal a pele do coto do paciente e medindo as suas propriedades mecânicas, tratando-se de um teste não invasivo.

Esta técnica apresenta várias vantagens, entre elas, o facto de não utilizar radiação, ser uma técnica de baixo custo e ter um alto grau de fiabilidade fornecendo medições muito precisas. Como desvantagens apresenta o facto de ser necessário a presença de um paciente para efetuar as recolhas, sendo também imprescindível a existência de técnicos com formação

¹ A versão do *Raspberry* ainda se encontra em definição.

especializada para trabalhar com o *indenter* de forma a evitar erros o processo de aquisição de dados.

Na Figura 3-6 é visível uma representação de como será o aspeto do *indenter*.



Figura 3-6 - Representação do indenter (Fonte: Elaborada pelo consórcio)

O equipamento necessário para a indentação é formado por três componentes básico, nomeadamente: um indentador de geometria específica geralmente montado numa coluna rígida, um acionador para aplicar a força, e um sensor para medir os deslocamentos do indentador. À medida que o indentador se aproxima do material, ocorre ao mesmo tempo uma deformação elástica e plástica na pele do paciente. Através do deslocamento do indenter na deformação elástica é possível determinar as propriedades elásticas do material.

Apresentados os componentes fundamentais do *Socket Master* interessa especificar os requisitos pretendidos relativamente a todo o processo de gestão e aquisição de dados de utilizadores que necessitam de um *socket* e como tal de toda a interação envolvida.

3.2 Requisitos

Os requisitos de *software* são considerados um fator determinante para o sucesso ou fracasso do projeto. Ao processo de levantar, analisar, documentar, fazer a gestão e controlar qualidade do *software*, dá-se o nome de Engenharia de Requisitos (Ricardo, 2012).

Esta secção aborda de forma sucinta os requisitos funcionais e não funcionais do projeto. Em relação aos requisitos não funcionais como, desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas, importa salientar, que não foram tidos muito em conta, devido à existência de requisitos funcionais, com maior prioridade de implementação. No entanto, explicita-se o que se considera essencial para cada um deles.

3.2.1 Requisitos funcionais

Relativamente aos requisitos funcionais, que podem ser consultados no apêndice C com maior detalhe, pretende-se implementar um conjunto de funcionalidades genéricas relativamente a todo o processo de gestão e aquisição de dados de utilizadores que necessitam de uma prótese. Estes encontram-se sintetizados de seguida.

1. RF 01 – Adicionar novo paciente

Descrição: Este requisito permite que o utilizador do sistema consiga registar um novo paciente no sistema informático, introduzindo alguns dados pessoais, assim como, as medidas do seu coto.

2. RF 02 – Remover paciente

Descrição: Este requisito permite que o utilizador do sistema consiga remover um paciente do sistema informático.

3. RF 03 – Alterar paciente

Descrição: Este requisito permite que o utilizador do sistema altere qualquer informação de um paciente já registado no sistema informático.

4. RF 04 – Listagem de pacientes

Descrição: Este requisito permite ao utilizador do sistema visualizar uma lista com todos os pacientes registados no sistema informático.

5. RF 05 – Pesquisar paciente

Descrição: Este requisito permite ao utilizador pesquisar um determinado paciente registado no sistema, por campos como o seu nome, endereço eletrónico, ou número de utente.

6. RF 06 – Adicionar nova aquisição de dados

Descrição: Este requisito permite ao utilizador do sistema iniciar uma nova aquisição de dados para um determinado paciente, registando os valores dos sensores na base de dados, por forma a serem acedidos mais tarde.

7. RF 07 – Apagar aquisição

Descrição: Este requisito permite ao utilizador do sistema remover uma aquisição de dados para um determinado paciente.

8. RF 08 – Registar descrição para aquisição de dados

Descrição: Este requisito permite ao utilizador do sistema registar uma descrição para a aquisição de dados.

9. RF 09 – Fazer a descrição de aquisição

Descrição: Este requisito permite ao utilizador do sistema alterar a descrição associada a uma aquisição de dados.

10. RF 10 – Inserir notas

Descrição: Este requisito permite ao utilizador do sistema inserir notas associadas a pacientes registados no sistema informático.

11. RF 11 – Listagem de notas por paciente

Descrição: Este requisito permite ao utilizador do sistema listar todas as notas associadas a pacientes registados no sistema informático.

12. RF 12 – Visualização do *socket* sob a forma de uma imagem tridimensional

Descrição: Este requisito permite ao utilizador do sistema visualizar o *socket* sob a forma de uma imagem tridimensional.

3.2.2 Requisitos não funcionais

Os requisitos não funcionais consistem em atributos de qualidade que serão especificados nesta secção.

A nível de desempenho, a solução deve ser o mais rápida possível na receção, processamento e armazenamento de dados provenientes dos sensores, assim, é requerido que o sistema consiga adquirir os dados provenientes dos sensores, processá-los e apresentar um modelo do *socket* num prazo de tempo aceitável.

A nível de usabilidade, a solução deve ser de fácil utilização para qualquer protésico ou profissional com formação na área, deste modo, pretende-se que a solução tenha coerência nos menus, cores tamanhos e tipos de letra, fácil navegação e um *software* intuitivo, de fácil aprendizagem.

A nível de segurança, como solução lida com dados e informações de pacientes, é pretendido que os mesmos dados estejam protegidos e inacessíveis a pessoas externas ao consórcio. Assim os dados devem ser guardados numa base de dados protegida com credencias de acesso.

A nível de disponibilidade, a solução deve responder sempre aos utilizadores em tempo útil, não devendo esta estar indisponível.

3.3 Casos de Uso

Baseado nos requisitos necessários a um bom desenvolvimento do *software* em questão, elaborou-se o diagrama de casos de uso (Figura 3-7). Este diagrama de casos de uso tem como objetivo descrever o comportamento do sistema na perspetiva do seu utilizador, utilizando ações e reações. Os casos de uso focam primordialmente a utilização do sistema. Desta forma, consegue-se observar de uma maneira mais simples e evidente a função do sistema, ou seja, a sua utilidade, bem como, uma sequência de iterações com elementos da interface para atingir um resultado útil.

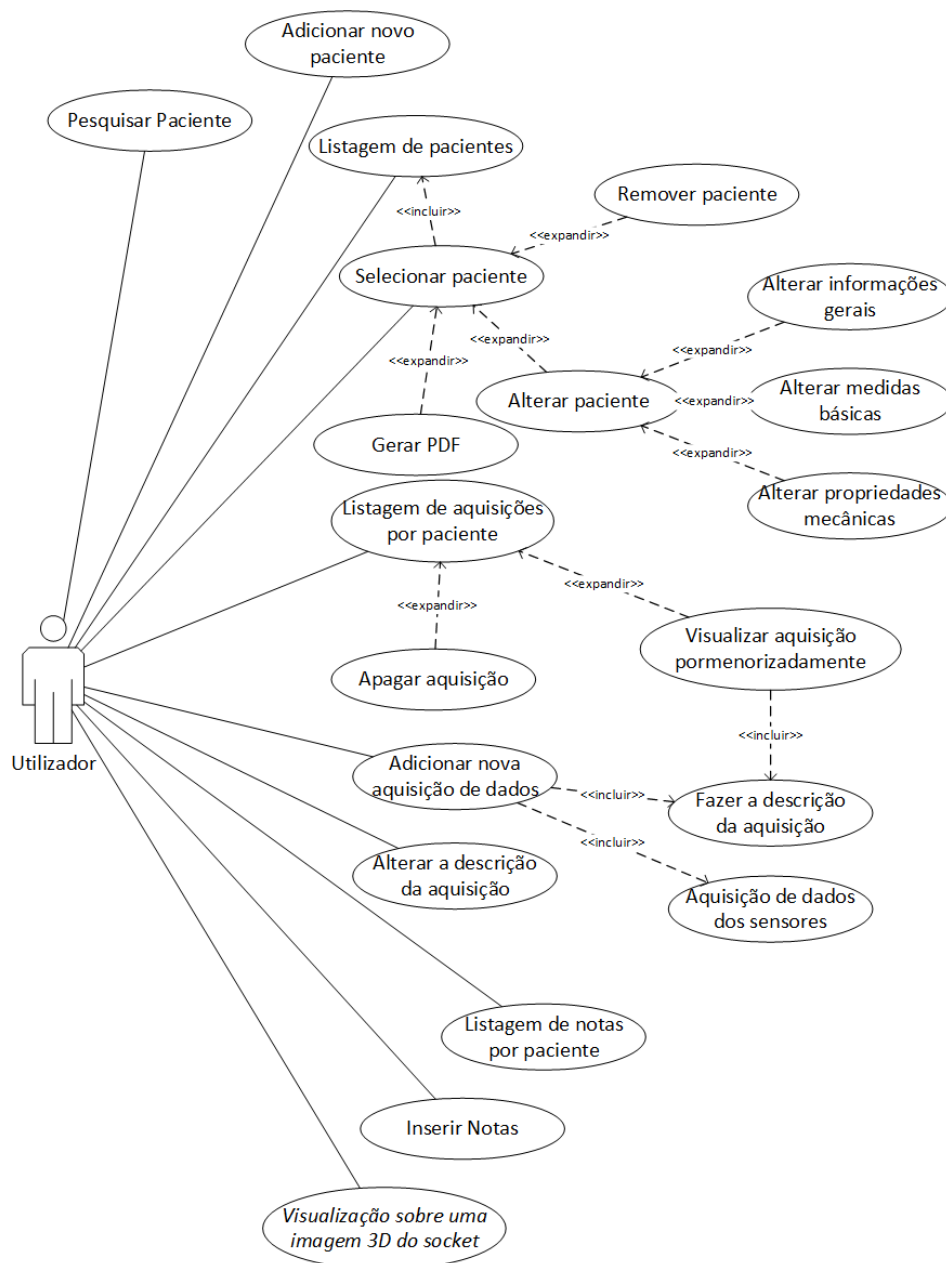


Figura 3-7 - Diagrama de Casos de Uso (Fonte: Elaborada pelo autor)

Neste diagrama tem-se representado o único utilizador que diretamente interage com o sistema, bem como as utilizações que este pode fazer. De salientar que, no processo de levantamento de requisitos foi questionado ao consórcio que tipos de utilizadores deveriam existir, e se existiria a necessidade de vários tipos de utilizadores com permissões diferentes. Por exemplo, um administrador de uma eventual clínica conseguir fazer todas as ações, um protésico conseguir fazer toda a gestão dos pacientes, e por exemplo um auxiliar apenas conseguir inserir novos pacientes e editar os seus dados. Porém, no momento em que estas questões foram feitas, debateu-se que para uma primeira fase do projeto existiriam outros aspetos prioritários não existindo a necessidade de ter utilizadores registados e com

permissões específicas, dessa forma não se contemplou sequer a existência de mais atores no sistema, nem um mecanismo de autenticação na aplicação.

Os casos de uso ilustrados anteriormente traduzem os requisitos funcionais da aplicação. Alguns têm propriedades específicas como por exemplo, aquando da iniciação do utilizador ao realizar o caso de uso “Alterar paciente”, poderá optar por uma ou mais das três opções facultativas: “Alterar informações gerais”, “Alterar medidas básicas” e “Alterar propriedades mecânicas”. Outro exemplo de um caso de uso com propriedades específicas é o “Adicionar nova consulta”. Ao iniciar este caso de uso é feita automaticamente uma aquisição de dados dos sensores, e quando a consulta terminar o utilizador tem de fazer o caso de uso correspondente a “Fazer a descrição da consulta”.

4 Arquitetura do Sistema

Neste capítulo é apresentada a arquitetura do sistema – *Socket Master*, através da qual será possível descrever o sistema e compreender o seu comportamento e fluxos de informação.

A arquitetura é um meio de se obter uma análise antecipada para garantir que a abordagem de *design* apresentada, produzirá um sistema capaz de alcançar os objetivos exigidos pelo cliente, tais como: desempenho, segurança, disponibilidade. Uma arquitetura de *software* consiste na definição das suas camadas, podendo estas camadas ser componentes de *software*, aplicações externas ou componentes de *hardware* (Mariotti, 2014).

4.1 Visão geral

O sistema desenvolvido é composto por três grandes módulos: o *hardware*, o *software* e a comunicação entre os dois anteriores.

Na Figura 4-1 pode observar-se que o sistema é composto por dois grandes sistemas, a parte física (*hardware*, ilustrado com tracejado verde), e um módulo para processamento de dados e gestão de informação dos pacientes (*software* – ilustrado com linha vermelha).

Existe também um módulo de comunicação para que os vários módulos do sistema comuniquem entre si de uma forma automática.

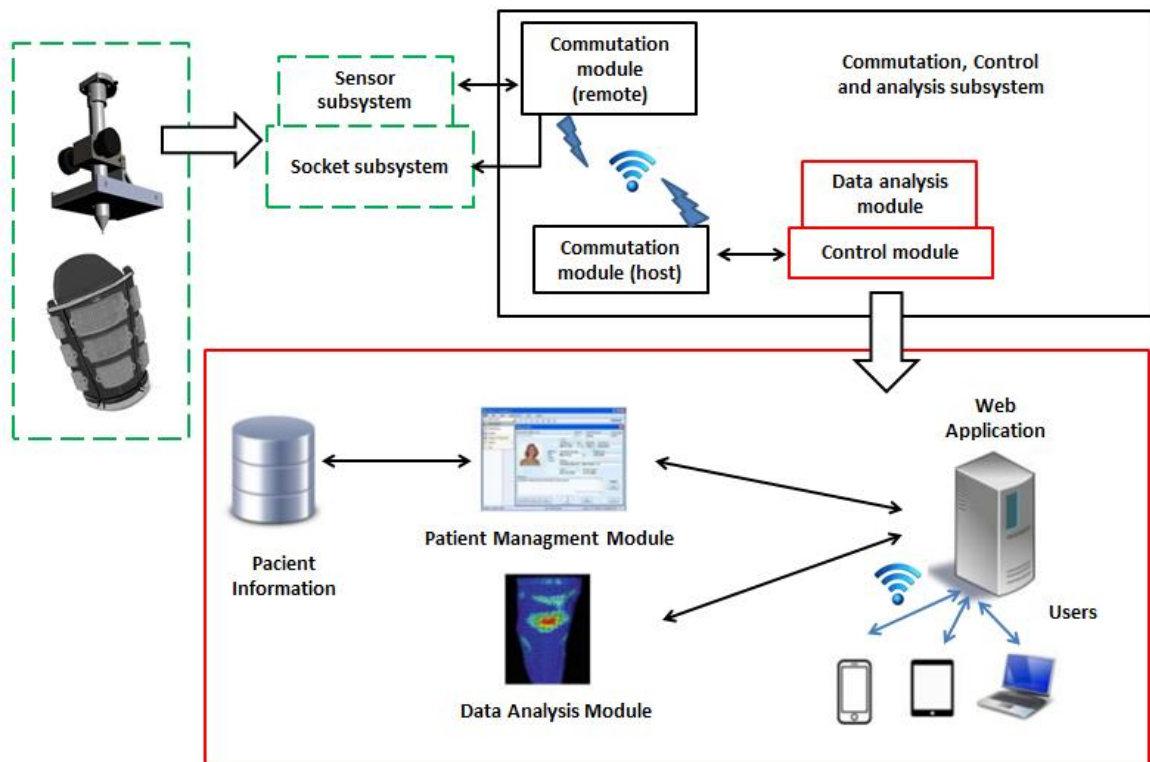


Figura 4-1 - Arquitetura Geral do Projeto (Fonte: Elaborada pelo autor)

4.1.1 Módulo de *Hardware*

O *hardware* do sistema é composto por um *socket* físico (*Socket Master*) ajustável ao coto do paciente, este *socket* contém uma malha de sensores composta por sensores de pressão e temperatura, que recolhem dados enquanto o paciente está a utilizar o *socket*. Para além destes dois dispositivos integrados num único, existe também um *Indenter*, dispositivo este capaz de adquirir as propriedades mecânicas do coto do paciente, através de uma técnica não invasiva.

4.1.2 Módulo de *Software*

O *software* do sistema é composto por dois módulos principais, um para gestão e recolha de dados dos pacientes (Patient Management Module) e outro para análise de dados recolhidos (Data Analysis Module) pelo *hardware*.

O módulo para gestão da informação dos pacientes, bem como das recolhas de dados de cada paciente é composto também por um motor de base de dados (Patient Information) para guardar toda a informação necessária.

Foi desenvolvido um algoritmo para fazer a análise e interpretação dos dados recolhidos. Este algoritmo tem como *input's* os valores provenientes dos sensores (pressão, temperatura, coordenadas da posição), sendo o *output* do mesmo o desenho otimizado do *socket* para determinado paciente, para que a prótese seja o mais confortável possível.

4.2 Conceção da base de dados

Por forma a conseguir guardar todos os dados que a aplicação necessita, foi necessário a construção de uma base de dados. Com esse objetivo, e por forma a melhor compreender as tabelas necessárias para o sistema trabalhar corretamente, bem como todas as relações existentes entre elas, desenharam-se vários diagramas, entre os quais: o modelo entidade relacionamento; o modelo conceptual de dados; e o modelo físico de dados.

4.2.1 Modelo entidade relacionamento

Mediante os problemas a solucionar com o desenvolvimento desta aplicação, e os requisitos atrás referenciados, concebeu-se o diagrama entidade relacionamento, que se pode observar na Figura 4-2.

Após uma breve análise sobre o modelo, verifica-se que a base de dados se encontra muito centrada nos pacientes. Praticamente todas as entidades têm uma relação com a entidade paciente (Patient). Na listagem após a figura, pode visualizar-se a lista de entidades e qual a sua finalidade na aplicação.

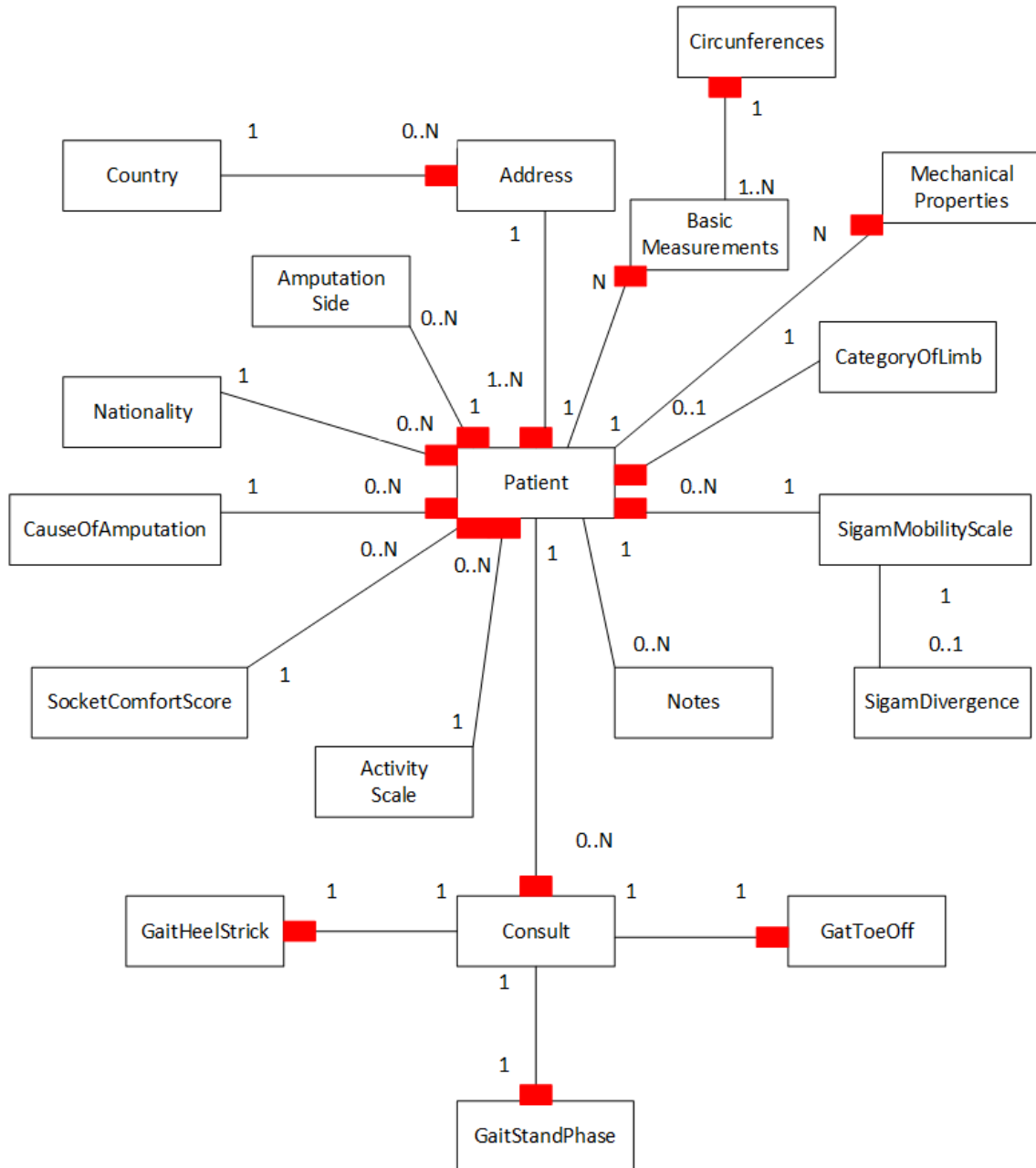


Figura 4-2 – Modelo entidade-relacionamento (Fonte: Elaborada pelo autor)

- *Patient* – Entidade com todos os pacientes registados no *software*;
- *Country* – Entidade com todos os países existentes no mundo;
- *Address* – Entidade com todos os dados referentes à morada dos pacientes;
- *Nationality* – Entidade com todas às nacionalidades existentes no mundo;
- *CauseOfAmputation* – Entidade com todas as possíveis causas de amputação (acidente de viação, motivos de saúde, ...);
- *SocketComfortScore* – Entidade com uma escala de conforto, na qual o paciente deverá avaliar a sua prótese quanto ao nível de conforto;

- *MechanicalProperties* – Entidade com todos os dados relativos às propriedades mecânicas do coto do paciente;
- *CategoryOfLimb* – Entidade para identificar qual o tipo de amputação, como por exemplo, distinguir amputações transfemorais de amputações transtibiais;
- *SigamMobilityScale* – Entidade com a escala de mobilidade de determinado paciente, para saber por exemplo se o paciente não consegue andar ou se anda no máximo 2 horas, entre outras possibilidades;
- *SigamDivergence* – Esta entidade relaciona-se com a *SigamMobilityScale* e tem como função guardar dados específicos de determinadas opções da *SigamMobilityScale*, por exemplo, um paciente que ande com o auxílio de moletas, é necessário saber se necessita de uma, duas, ou até mesmo de um andador;
- *BasicMeasurements* – Entidade com todos os dados relativos a medições, como medidas do coto do paciente, assim, algumas medidas como a distância do coto ao joelho, ou à anca serão guardados nesta entidade;
- *Circunferences* – Entidade que se relaciona com a *BasicMeasurements*, e permite identificar o diâmetro do coto do paciente;
- *ActivityScale* – Entidade com uma escala de valores de um a dez, permitindo avaliar numa escala finita qual o nível de atividade do paciente;
- *AmputationSide* – Entidade com três valores possíveis: esquerda, direita ou ambos, permite identificar qual o lado da amputação do paciente;
- *Notes* – Entidade com todas as notas associadas aos pacientes;
- *Consult* – Entidade com todos os dados das várias consultas feitas aos pacientes;
- *GaitHeelStrick* – Entidade com todos os movimentos quando o pé estiver levantado apenas com o calcanhar no chão. Esta entidade guarda os dados provenientes dos sensores;
- *GaitStandPhase* – Entidade com todos os movimentos quando o pé estiver completamente assente no chão. Esta entidade guarda os dados provenientes dos sensores;
- *GaitToeOff* – Entidade com todos os movimentos quando o pé estiver com o calcanhar levantado e a ponta no chão. Esta entidade guarda os dados provenientes dos sensores.

A descrição detalhada de todas estas tabelas pode ser consultada no apêndice D.

4.2.2 Modelo conceptual de dados

O modelo conceptual de dados permite identificar todas as entidades existentes no sistema, identificando os atributos de cada uma delas bem como as relações entre elas.

O modelo conceptual de dados do presente projeto pode ser visualizado na Figura 4-3.

4.2.3 Modelo físico de dados

Para concluir a modelação da base de dados, foi gerado o modelo físico de dados (Figura 4-4) partindo do modelo conceptual de dados.

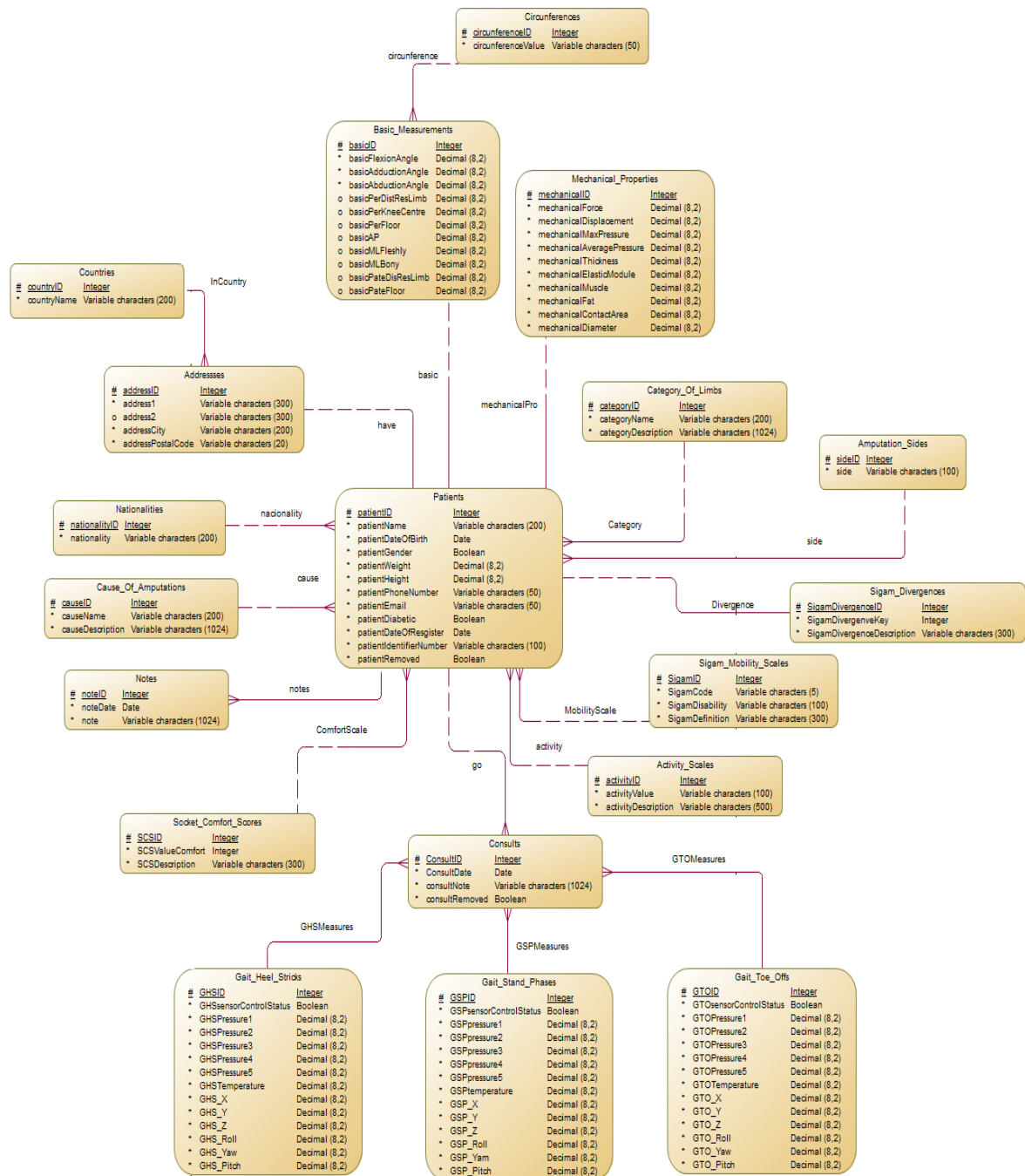


Figura 4-3 - Modelo Conceptual de Dados (Fonte: Elaborada pelo autor)

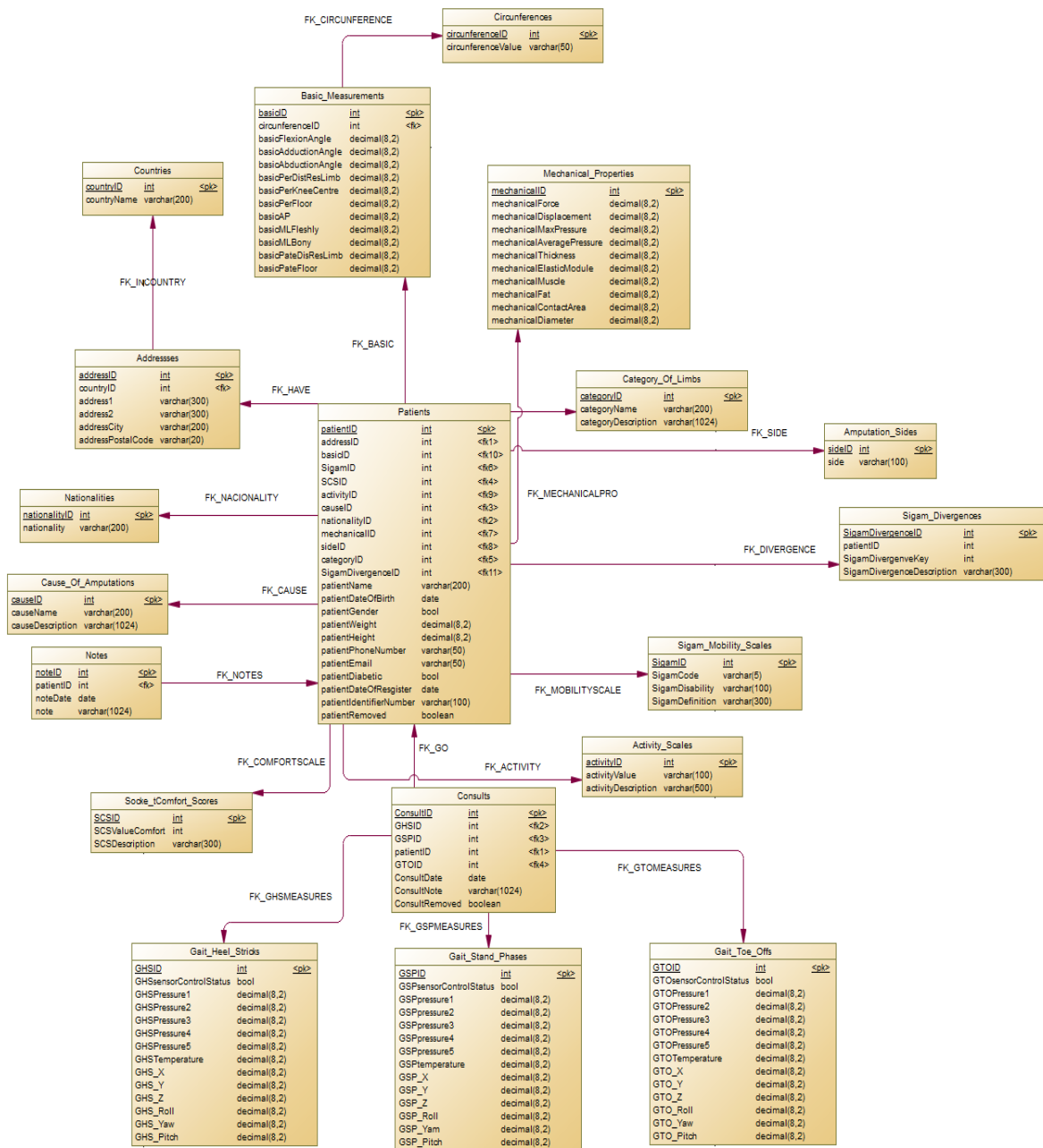


Figura 4-4 - Modelo Físico de Dados (Fonte: Elaborada pelo autor)

4.3 Linguagens e Tecnologias

Após uma primeira análise sobre o sistema, definição de requisitos, desenho da arquitetura, e modelação de base de dados, deu-se início ao processo de desenvolvimento. Porém, este processo não pode ser tido em conta levemente, sendo necessário definir as tecnologias e linguagens de programação que irão ser utilizadas, bem como as *frameworks* já existentes no mercado e que possam ajudar na implementação do projeto, entre outros fatores como a sua interligação, compatibilidade e adequabilidade.

Assim, antes de dar início ao processo de desenvolvimento do projeto, foi necessário estabelecer alguns pontos, entre eles responder à questão “Que tecnologias e linguagens de programação devem ser utilizadas para dar resposta aos requisitos deste projeto?” Desde a análise do projeto que se definiu que este seria desenvolvido sobre uma arquitetura web, tendo sido elaborada uma pesquisa de entre várias tecnologias para o desenvolvimento web, para desta forma optar pela que melhor se adapte ao projeto.

Assim sendo, analisaram-se três tecnologias para o desenvolvimento web: .NET da Microsoft, J2EE da Oracle e o PHP que é uma linguagem *open source*. Ponderou-se também desenvolver uma aplicação mobile através de *frameworks* que permitissem publicar o projeto em suportes com vários sistemas operativos, tais como *Android*, *iOS* e *Windows*. No entanto esta possibilidade foi descartada, uma vez que se definiu que o *software* seria utilizado principalmente em computadores pessoais, como portáteis com sistemas operativos *Windows*, *Linux* ou *MAC OSX*. Embora a utilização do *software* também possa ocorrer em *tablets*, a sua utilização em *smartphones* com dimensões reduzidas não será desejável.

Face ao exposto, neste subcapítulo é feita uma abordagem sobre as diversas tecnologias analisadas: .NET, J2EE e PHP. Para cada uma delas, serão identificadas as suas principais funcionalidades e características, assim como, as suas principais vantagens e desvantagens.

4.3.1 .NET

.Net é uma *framework* de *software* desenvolvida pela Microsoft (ASP.NET, 2015), que inclui uma enorme biblioteca de classes e permite ao programador trabalhar com várias linguagens de programação, entre elas o C#, o *Visual Basic* e o C++. Os programas desenvolvidos com esta *framework* são transcritos para uma linguagem intermédia, CLR (*Common Language Runtime*) que depois é interpretada por uma máquina virtual. É esta linguagem intermédia que permite a programação com distintas linguagens de programação (Microsoft, 2016).

Algumas das vantagens desta *framework* são a interdependência entre linguagens, o CLR permite aos programadores escolherem em que linguagem pretendem programar a sua aplicação de entre um leque alargado de linguagens disponíveis. Outro fator importante é a portabilidade fornecida por esta *framework*. Ao utilizar uma máquina virtual onde o programa é executado, os utilizadores podem aceder a esta plataforma através de vários sistemas operativos diferentes.

Um dos fatores negativos desta tecnologia é o facto de não ser *open source*, ou seja, para desenvolver projetos com esta *framework* é necessária a aquisição de licenças.

4.3.2 J2EE

Java Enterprise Edition (JEE), é uma plataforma para desenvolvimento de *software* para a *web*, sendo maioritariamente programada através da linguagem de programação *Java*.

Através desta plataforma, é possível interligar algumas das tecnologias existentes no desenvolvimento *web* (HTML 5, CSS 3, *JavaScript*) com o *Java*, permitindo o desenvolvimento de aplicações *web*, com a rapidez, segurança e a organização que o *Java* oferece (ORACLE, 2014).

Alguns fatores que salientam o *JEE* como uma boa plataforma para o desenvolvimento de aplicações *web* são a sua robustez, o facto de possuir um vasto leque de API's, podendo estas serem reutilizáveis pelas aplicações a ser desenvolvidas. Outro fator a favor é o de ser uma linguagem *open source* e possuir uma enorme comunidade que a utiliza, desta forma as dúvidas que possam surgir com a linguagem facilmente serão resolvidas. A programação orientada a objetos é outro dos fatores chave. Desta forma trabalhar com um modelo de dados específico é muito mais rápido e fácil.

4.3.3 PHP

O PHP é uma linguagem de programação baseada em *scripts* direcionada para o desenvolvimento *web*. Esta tecnologia evidencia-se devido à sua facilidade de aprendizagem, velocidade e flexibilidade (PHP, 2016).

Alguns dos pontos que favorecem a utilização desta linguagem são o facto de esta ser *open source*, ágil, possuir muitos mecanismos para trabalhar com base de dados, facilidade no seu alojamento na *web*, uma vez que praticamente todos os serviços de *hosting* são compatíveis com o PHP.

Estes aspetos proporcionam um rápido desenvolvimento para projetos de pequenas dimensões, sendo adicionalmente de fácil aprendizagem e manuseamento (Praciano, 2014).

Porém existem também desvantagens associadas a esta linguagem e essas recaem principalmente na robustez, segurança e o facto de não ser orientada a objetos. Este aspeto torna mais difícil o trabalho com um modelo de dados, tornando mais complexo e menos expedito o acesso a dados e suas alterações, fatores que seriam facilitados através de uma linguagem orientada a objetos.

Outro fator negativo do PHP é o facto de as variáveis não serem tipificadas (em inteiro, decimal, ou outro tipo). Este facto poderá trazer problemas na altura da implementação de algoritmos, podendo este estar mais propenso à ocorrência de erros.

4.3.4 Análise comparativa entre as tecnologias apresentadas

Todas as três tecnologias têm como principal foco o desenvolvimento de aplicações para a *web*, no entanto todas elas têm propriedades distintas, importando definir qual será mais vantajosa para o projeto em questão. Construiu-se a Tabela 4-4 por forma a aglomerar as vantagens e desvantagens das tecnologias atrás referenciadas.

Após a sua análise, constata-se que .NET e JEE são tecnologias mais robustas, seguras, e o facto de possuírem variáveis tipificadas permite uma melhor deteção de erros no ato da compilação do projeto. Estas duas tecnologias possuem também um vasto leque de API's e bibliotecas.

Porém a escolha da tecnologia recaiu sobre o PHP, que embora não tenha algumas das características das outras tecnologias, seria mais ágil e fácil de trabalhar, proporcionando um rápido desenvolvimento, o que traduz um aspeto muito importante dada as características de um projeto desta natureza. Também se verificou que embora esta tecnologia não permita a programação orientada a objetos, já existem algumas *frameworks* que auxiliam os programadores nestas tarefas, como é o caso do Laravel que será abordado no subcapítulo 4.4.6.

Tabela 4-4 - Comparação das tecnologias para desenvolvimento web

	.NET	JEE (JAVA)	PHP
Vantagens	• Programação orientada a objetos • Bastante robusta • Escalável • Segura • Variáveis tipificadas	• <i>Open Source</i> • Programação orientada a objetos • Robusta • Escalável • Segura • Variáveis tipificadas	• <i>Open Source</i> • Ágil • Desenvolvimento rápido para projetos de pequenas dimensões • Fácil de comunicar com base de dados • Fácil de colocar num servidor • Facilidade de aprendizagem
Desvantagens	• Tem custos • Dificuldade em colocar no servidor • Necessidade de escrever muito código para fazer uma operação simples	• Dificuldade em colocar no servidor • Necessidade do conhecimento de muitas API's e Classes externas	• Pouco robusta • Pouco segura • Fraca programação orientada a objetos • Código visível no servidor • Variáveis não tipificadas

4.4 Bibliotecas e Frameworks

Desenvolver este projeto todo de raiz, seria uma tarefa bastante desafiadora, face aos tempos definidos e prazos a cumprir. Adicionalmente, a existência de bibliotecas e *frameworks* que se encontrassem bem testadas, com API's bem documentadas poderiam facilitar a implementação de determinadas funcionalidades da aplicação tornando-se numa situação muito vantajosa.

Por forma a agilizar o processo de desenvolvimento, fez-se uma pesquisa sobre bibliotecas e *frameworks* que poderiam ajudar a implementar algumas das funcionalidades pretendidas.

Neste subcapítulo são, de seguida, abordadas algumas das bibliotecas e *frameworks* utilizadas.

4.4.1 Bootstrap

Um dos principais paradigmas com que os programadores web se deparam é, a apresentação da aplicação desenvolvida num dispositivo, com dimensões diferentes das que o programador está habitualmente a trabalhar. Uma interface com muito conteúdo, que seja implementada sem ter em conta o tamanho do ecrã, pode ser perfeita para um monitor de 14 polegadas, no entanto ao abrir a mesma aplicação num *tablet* de 9 polegadas por exemplo, pode tornar impossível tarefas como clicar em botões, preencher formulários, devido a estes ficarem demasiado pequenos tendo em conta a dimensão do ecrã.

O Bootstrap é uma *framework front-end* que visa ajudar o trabalho dos programadores. Com a ajuda desta *framework*, é possível desenvolver desde um simples *website* a uma aplicação web complexa e robusta, tendo como principal objetivo a responsividade (Figura 4-5), quer isto dizer, a forma como a informação é mostrada ajustando-se a diferentes tamanhos de ecrã.

Esta *framework* baseia-se numa grande quantidade de classes CSS, tendo o programador de as conhecer e desta forma criar interfaces amigáveis de uma forma rápida e com pouco código. Desta forma torna simples a implementação de certas funcionalidades como: *tooltip*, *menu-dropdown*, *carousel*, *slideshow*, entre outros (Costa, 2014).



Figura 4-5 - Exemplo de responsividade – Bootstrap (Fonte: Hayaran, 2014)

O Bootstrap funciona com um sistema de *grids* (grelhas) para posicionar os elementos na página. Este sistema funciona como uma tabela abstrata que se adapta de acordo com a dimensão do ecrã, sendo ajustada sempre que o ecrã muda de tamanho ou de orientação e é *Mobile First*, o que significa que assume, inicialmente, que o ecrã é o de um dispositivo móvel, com tamanho pequeno. Por outras palavras, este mecanismo adapta todos os conteúdos para tamanhos pequenos, e de seguida analisa o tamanho real, e ajusta os itens de forma a ficarem posicionados corretamente, tendo em conta o tamanho e a resolução do dispositivo.

Existem alguns conceitos importantes para desenvolver uma aplicação com esta *framework*, tais como:

- *container* – É uma *div* (hierarquia DOM que divide a página numa porção) que garante que o *layout* irá ficar sempre alinhado na página, centralizando o conteúdo no *browser* e criando margens laterais. Desta forma o elemento principal de uma página web deverá conter esta classe;
- *row* – Em português linha, e tal como o nome indica, esta classe servirá para fazer divisões horizontais na página, duas *div's* numa página ambas com a classe *row*, irão posicionar-se um por baixo da outra;
- *column* – Em português coluna, esta terá a função de fazer divisões na vertical. As *columns* devem estar sempre dentro de *row's*.

Existem prefixos para as colunas, que têm como função indicar ao *layout* até que tamanho de ecrã as colunas se vão manter colunas, quer isto dizer que duas colunas poderão fazer sentido ser colunas num ecrã de 14 polegadas, e passar a ser linhas num dispositivo de 5 polegadas, facilitando desta forma a visualização dos conteúdos em dispositivos menores. Na Tabela 4-5

podem-se observar os vários prefixos existentes no Bootstrap, identificando qual o tamanho mínimo do ecrã para ser visível como coluna.

Tabela 4-5 - Prefixos das colunas

<i>Smartphones</i>	<i>Tablets</i>	<i>Desktops</i>	<i>Ecrãs grandes</i>
.col-xs-*	.col-sm-*	.col-md-*	.col-lg-*

O * deve ser substituído pelo tamanho da coluna (de 1 a 12).

Por exemplo, duas colunas definidas com o prefixo `.col-md-6` são vistas num ecrã de 14 polegadas, como duas colunas ocupando cada uma 50% da largura do ecrã. No entanto se a mesma página for visitada num *smartphone*, as mesmas colunas são visíveis como duas linhas, isto porque, o prefixo `.col-md-6` garante que a coluna é visível como coluna no mínimo em ecrãs de *desktops*, em ecrãs mais pequenos, irá comportar-se como linhas.

Anteriormente foi referido que o * deve ser substituído pelo tamanho da coluna, tamanho este compreendido entre 1 e 12. Resta agora compreender o significado destes valores. Cada linha pode ter no máximo 12 colunas, e no mínimo 1. Sabe-se que para as colunas devem estar declaradas dentro de uma linha, e a soma dos * das colunas deve totalizar 12. Desta forma se se pretender desenhar 3 colunas todas com a mesma largura bastará dividir 12 por 3 e obter o número 4 ($12/3 = 4$). Desta forma devem ser declaradas três colunas tendo o $* = 4$.

A primeira linha contém doze colunas de tamanho unitário, a segunda tem três colunas de tamanho quatro, a terceira tem duas colunas de tamanho seis, a quarta tem apenas uma coluna de tamanho doze, a quinta e última linha tem apenas uma coluna de tamanho três, isto porque se a soma das colunas for menor que doze, o Bootstrap ajusta da esquerda para a direita, não necessitando de somar doze obrigatoriamente.

Neste projeto foi utilizado para implementar as interfaces gráficas da aplicação. Com esta biblioteca foi possível implementar uma aplicação responsiva. Também os vários ícones da aplicação pertencem a esta biblioteca, o que permite utilizar ícones semelhantes aos existentes noutros *software*, evitando a dúvida no utilizador.

4.4.2 JQuery

JQuery é uma biblioteca do JavaScript com o objetivo de facilitar a manipulação dos elementos de uma página HTML, permitindo programar eventos de botões, criar animações e trabalhar com tecnologia AJAX de uma forma menos complexa que utilizando JavaScript puro.

O principal objetivo desta *framework client side* é o de simplificar os *scripts* que integram o HTML. Esta biblioteca é *open source*, apresentando como principais vantagens, o facto de resolver incompatibilidades entre *browsers*, possibilitar um código mais reduzido com a reutilização de código através de *plugins* e utilização de *plugins* de terceiros, o facto de

trabalhar com AJAX e DOM e possibilitar a implementação segura de recursos de estilos (CSS).

Para utilizar esta *framework* basta fazer o *download* da mesma, e incluí-la no projeto a ser desenvolvido. Estão disponíveis duas versões do JQuery para *download*, a versão integral e a versão minimizada.

A função principal do JQuery é a `$()`, sendo um atalho para a função principal do JQuery. Ao analisar-se código escrito com esta biblioteca, notar-se-á que praticamente todas as linhas começam com a chamada a esta função. Esta função permite selecionar um determinado elemento de uma página HTML para o trabalhar, ou seja, aplicar um efeito de animação, mudar algum estilo associado a este, como por exemplo aumentar o tamanho da fonte de um texto quando se passa o rato por cima (Belem, 2010).

Analise-se agora a sintaxe mais abundante num *script* JQuery, e qual o seu significado:

- `$()` – Utilizado em todas as funções que referenciam o JQuery;
- `document` – Expressão para identificar o documento HTML;
- `ready()` – Indica ao *script* para ser executado enquanto a página está a ser carregada;
- `function{ }` – Dentro das chavetas deverá conter os comandos a serem interpretados pelo *browser* (Emmanuele, 2008).

Na prática esta biblioteca surgiu para apoiar o trabalho dos programadores. O lema do JQuery é “Escrever menos e fazer mais”, para isso o JQuery está munido de um vasto leque de funções para trabalhar os elementos de uma página HTML, a sintaxe básica do JQuery é: `$(seletorHTML).ação()`, seguem-se algumas das funcionalidades desta poderosa ferramenta:

- `click()` – Permite, por exemplo, programar o evento do clique do rato em determinado botão;
- `hide()` – Função para esconder determinado elemento da página web;
- `show()` – Função para mostrar determinado elemento da página web;
- `toggle()` – Função que permite alterar a visibilidade de elementos HTML que usam a função `show()/hide()`, desta forma os elementos que estiverem a `show()` irão mudar para `hide()` e vice-versa;
- `slideDown(speed, callback) / slideUp(speed, callback) / slideToggle(speed, callback)` – Função que permite alterar gradualmente a altura dos elementos selecionados, dando-lhes movimento verticalmente;
- `fadeIn(speed, callback) / fadeout(speed, callback) / fadeTo(speed, opacity, callback)` – permite alterar gradualmente a opacidade dos elementos selecionados, proporcionando-lhes um efeito de desvanecimento;
- `css()` – Função que permite definir propriedades CSS para os elementos;
- `load(url, data, callback)` – Função AJAX simples mas muito poderosa para comunicar com o servidor.

Neste projeto o JQuery foi utilizado principalmente para elaborar algumas animações e também para comunicar com o servidor recorrendo ao AJAX.

4.4.3 DataTables

DataTables é um forte *plugin* do JQuery para criar tabelas e adicionar certas funcionalidades à mesma. Através deste *plugin* é possível prover a tabela com mecanismos para pesquisa, ordenação por colunas, paginação, tudo isto é conseguido com poucas linhas de configuração.

A utilização desta ferramenta não é complexa, basta apenas criar a tabela em HTML identifica-la com o ID próprio, e de seguida num *script* JQuery associar este *plugin* à tabela, configurando alguns parâmetros facultativos por forma a criar as funcionalidades desejadas para a tabela.

Entre as funcionalidades possíveis destacam-se as que foram aproveitadas para implementar nas tabelas deste projeto:

- *bPaginate* – Propriedade booleana para definir se a tabela deve ser particionada em várias páginas (paginação);
- *pageLength* – Propriedade inteira para definir quantos registos estão visíveis por página da tabela;
- *bFilter* – Propriedade booleana para ativar a ordenação por colunas da tabela bem como a pesquisa de dados da tabela.

Esta ferramenta tornou-se uma mais valia para o projeto, pois facilitou a implementação de requisitos tais como: pesquisa de paciente e ordenação por diferentes dados do paciente.

4.4.4 AJAX

O AJAX (Asynchronous Javascript and XML) é uma tecnologia que tem como objetivo tornar as páginas web mais interativas com o utilizador, fazendo atualizações assíncronas de informação. Apesar de na sua designação conter a palavra XML, também é frequente a utilização da tecnologia JSON.

No funcionamento do AJAX (Figura 4-6), após ocorrer um estímulo através da interface web, é feita uma chamada a um método do Javascript (1 e 2), o javascript faz uma solicitação ao servidor (3), que normalmente responderá ao pedido, retornando apenas os dados em formato XML ou JSON (4). Após a obtenção da resposta o javascript fará a sua renderização na página HTML (5 e 6). Esta tecnologia permite que apenas algumas partes de uma página sejam atualizadas, evitando o consumo de dados necessário para recarregar uma página inteira a cada iteração do utilizador.

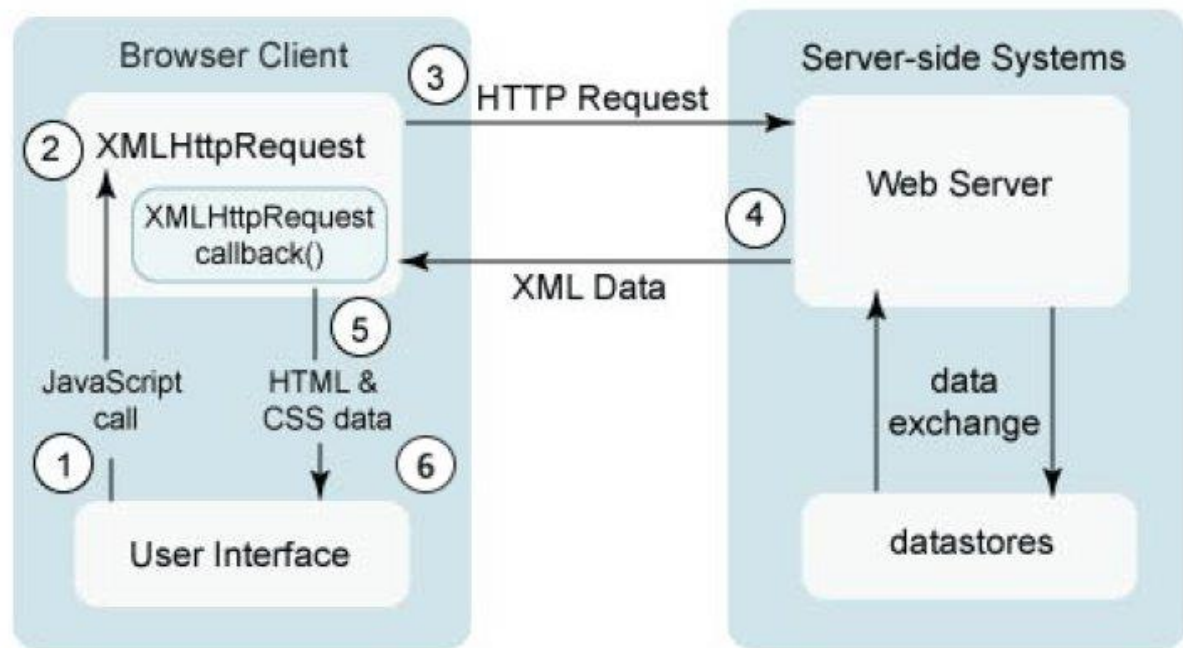


Figura 4-6 - Funcionamento da tecnologia Ajax (Fonte: adaptada de Magento, 2014)

Esta tecnologia destaca-se pelo facto de comunicar com o servidor de uma forma assíncrona, permitindo desta forma ao utilizador continuar a interagir com a página enquanto a comunicação ao servidor é concluída. Adicionalmente destacam-se como vantagens: o aumento da velocidade das interações com uma aplicação web, a aplicação com melhor usabilidade por parte dos utilizadores, e a maior flexibilidade na escolha da linguagem para o servidor. Por outro lado, apresenta algumas desvantagens como o facto dos botões de avançar e retroceder do *browser* serem inúteis, pois o *browser* não consegue navegar entre as comunicações através desta tecnologia (Medeiros, 2014).

A título de exemplo, considere-se a situação na qual o utilizador visualiza uma lista de pacientes e pretende remover um deles. Através desta tecnologia é possível comunicar com o servidor para remover esse paciente e atualizar apenas a sua listagem, evitando atualizar a página na sua totalidade.

No projeto em questão, o AJAX foi utilizado para comunicar assincronamente com o servidor, permitindo por exemplo, atualizar listas de pacientes, sem existir a necessidade de recarregar a totalidade da página.

4.4.5 Three.js

A Three.js é uma biblioteca escrita em Javascript que permite criar e visualizar modelos 3D num *browser*. Esta biblioteca permite criar animações 3D através de WebGL disponibilizando uma API para trabalhar com este de uma forma mais acessível.

São apresentados de seguida os conceitos inerentes a esta biblioteca:

- Scene – Uma cena (em português) conterá todos os objetos a serem visualizados na aplicação;
- Camera – Câmera (em português) será um ponto na cena 3D pelo qual o utilizador poderá visualizar os objetos tridimensionais, analogamente a uma câmara de filmar, também esta será o ponto de visualização da cena;
- Renderer – Mapeamento (em português) será a forma de mapear os objetos presentes na cena na página HTML;
- Geometry – Contém as faces e os vértices do elemento a criar. A própria biblioteca disponibiliza algumas geometrias já criadas, entre elas: BoxGeometry, CircleGeometry, CylinderGeometry. A biblioteca disponibiliza também ferramentas para importar geometrias externas;
- Material – É um objeto para colorir os objetos criados;
- Mesh – É um objeto que recebe uma geometria para lhe aplicar um material. Este objeto também pode sofrer transformações como por exemplo, rotações e translações (Amarante, 2015).

Three.js foi fundamental para o desenvolvimento do projeto, uma vez que um dos requisitos da aplicação passava pela visualização de modelos tridimensionais, pelo que para ajudar na implementação desta funcionalidade foi integrada esta biblioteca.

4.4.6 Laravel

O Laravel é uma *framework* escrita em PHP utilizada para o desenvolvimento web, utilizando a arquitetura MVC (*Model View Componente*). Através desta, é possível desenvolver aplicações seguras, com melhor desempenho, permitindo uma programação com código descomplicado. São apresentados de seguida algumas das características mais relevantes:

- *Blade* – É um *template* para criar interfaces gráficas fornecendo uma enorme variedade de ferramentas para ajudar a criar interfaces amigáveis e funcionais de forma rápida evitando replicação de código;
- *ActiveRecord* – Também conhecido por *Eloquent Object-relacional mapping (ORM)* é uma ferramenta que facilita a comunicação com uma base de dados, esta ferramenta aglomera várias funcionalidades para simplificar as funções de inserir, atualizar, excluir e procurar registos numa base de dados;
- *Composer* – Ferramenta para gerir as dependências de projetos, sendo possível definir todas as dependências de um projeto num ficheiro de configurações, e automaticamente são feitos todos os *downloads* necessários assim como a preparação do ambiente do projeto.

Alguns dos conceitos inerentes ao Laravel são: as rotas, ficheiro onde são definidos os caminhos (URL) da aplicação bem como os controladores que irão tratar determinados pedidos; os *Controllers*, classes onde serão implementadas a lógica da aplicação, onde se

encontram os pedidos dos utilizadores e onde são geradas as respostas; e o *View*, classes onde se define as interfaces visíveis para o utilizador. No Laravel existem também os modelos, onde são criadas as várias classes para o correto funcionamento da aplicação (Adriel, 2015).

Neste projeto o Laravel foi utilizado para organizar o projeto, assentando numa arquitetura MVC (*Model View Controller*), permitindo assim, organizar melhor as suas dependências, evitar replicação de código bem como organizar a estrutura da base de dados, fornecendo mecanismos de migração e de *backups*.

5 Desenvolvimento e Implementação

Antes de dar início ao desenvolvimento de uma aplicação robusta, intuitiva e capaz de responder a todos os requisitos propostos, foi necessário desenvolver algumas interfaces. Desta forma, o processo de desenvolvimento das interfaces do projeto teve as seguintes etapas:

1. Desenho dos *mockups* da aplicação;
2. Apresentação à empresa;
3. Obtenção de *feedback*;
4. Redesenho segundo os contributos da empresa.

Após a conclusão dos *mockups* iniciou-se o processo de implementação dos mesmos.

5.1 Mockups do projeto

Numa primeira fase optou-se por desenhar protótipos de baixa fidelidade, os *mockups* para a aplicação de forma manuscrita (Figura 5-1), assim foi possível simular as várias interações possíveis com o sistema, obtendo *feedback* por parte da empresa sobre a sua adequabilidade.

The image shows a hand-drawn mockup of a patient registration form. The form is titled "New Patient" and contains several input fields and sections. On the left, there is a "General Measurements" section for a patient named "Robin" (45 years, 82.7 kg) and a "Sensors Data" section with dates for first and last consultations and a total of 2 consultations. The main form area includes fields for "Id Patient" (130,337), "Name", "weight" (50.0), "height" (1.70), "gender" (Male), "Birth Date" (1/1/19), "Nationality" (Portuguese), "Category" (Limb), and a "Note" field. The bottom right corner has a "Cancel" button and a save icon.

Figura 5-1 - Mockups desenhados à mão (Fonte: Elaborada pelo autor)

Após uma primeira apresentação dos *mockups* e feita uma ilustração das interações possíveis com os mesmos, foram retidas algumas informações relevantes, levadas em conta para o melhoramento dos *mockups*. De seguida são listados os pontos a ter em conta, resultantes dessas discussões, para o desenvolvimento das interfaces:

- Os conteúdos deverão ser em inglês;
- Deverá ser criada uma página inicial (HOME) que integrará a funcionalidade de pesquisa de pacientes, e botões para inserir um novo paciente, listar todos os pacientes e, fazer novas aquisições de dados;
- Os botões referentes às funcionalidades de adicionar, editar e remover pacientes deverão ficar na coluna da direita;
- O formulário de inserção de paciente deverá estar dividido em três níveis (informações gerais, medidas básicas e propriedades mecânicas);
- Os dados do paciente também deverão estar separados em vários níveis, destacando, as informações gerais, medidas básicas, propriedades mecânicas, dados de aquisições e notas do paciente.

Após a análise de todas estas sugestões foram redefinidos os *mockups* e desenhados novamente recorrendo ao *software* Balsamiq Mockups, estes podem ser consultados na sua totalidade no apêndice E. Na Figura 5-2 pode visualizar-se, a título de exemplo, o *mockup* da página principal da aplicação.

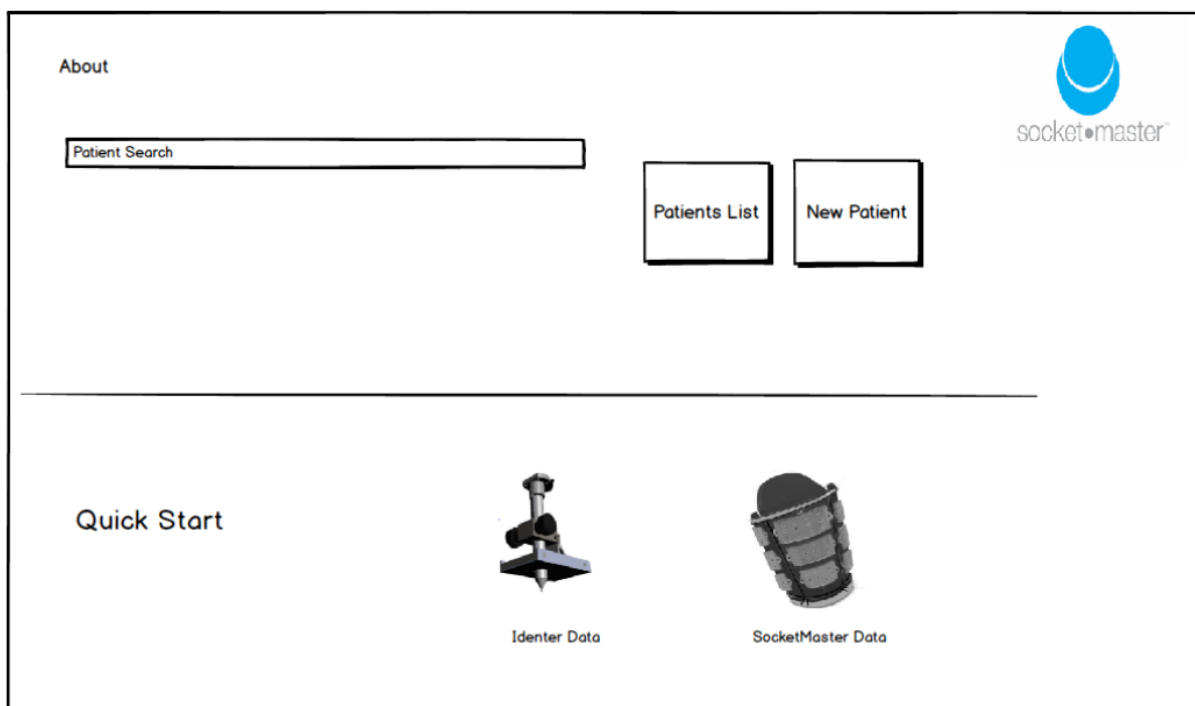


Figura 5-2 - Mockup da página principal da aplicação (Fonte: Elaborada pelo autor)

5.2 Interfaces do Sistema

Após a validação dos mockups, procedeu-se à implementação das interfaces respetivas. De notar que, após a conclusão dos mockups, ainda foram sugeridas mais duas alterações que dizem respeito à localização do logótipo do projeto no canto superior esquerdo e à colocação dos logótipos das entidades pertencentes ao consórcio no rodapé da aplicação. Na Figura 5-3

pode visualizar-se a página principal da aplicação desenvolvida já com estas últimas sugestões implementadas.

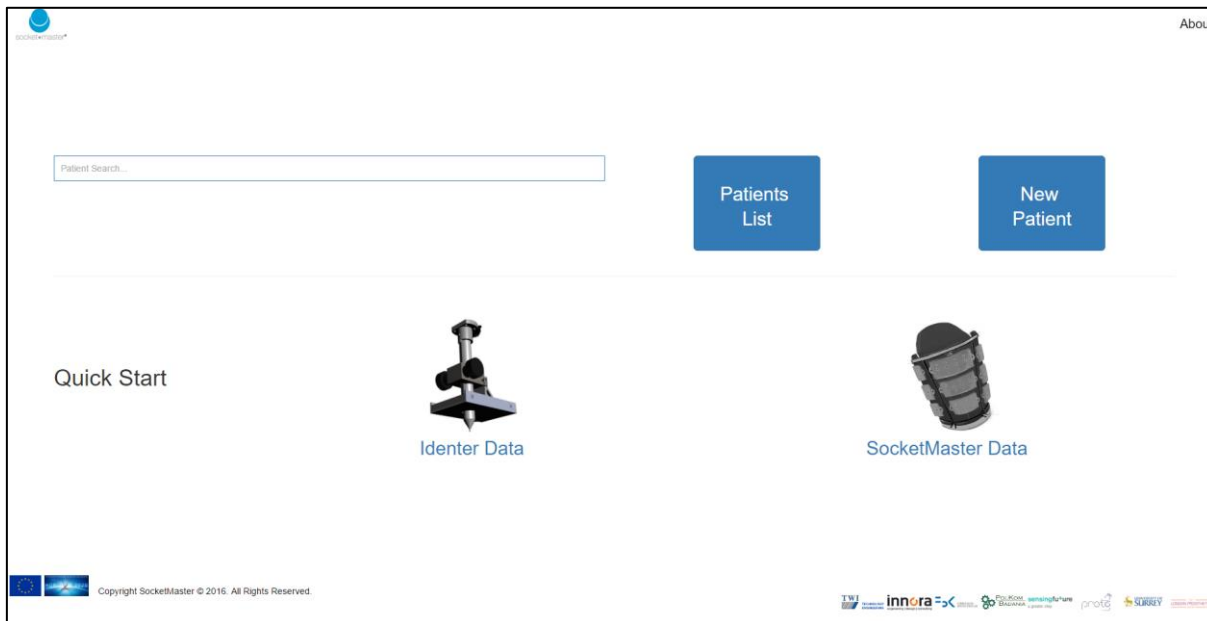


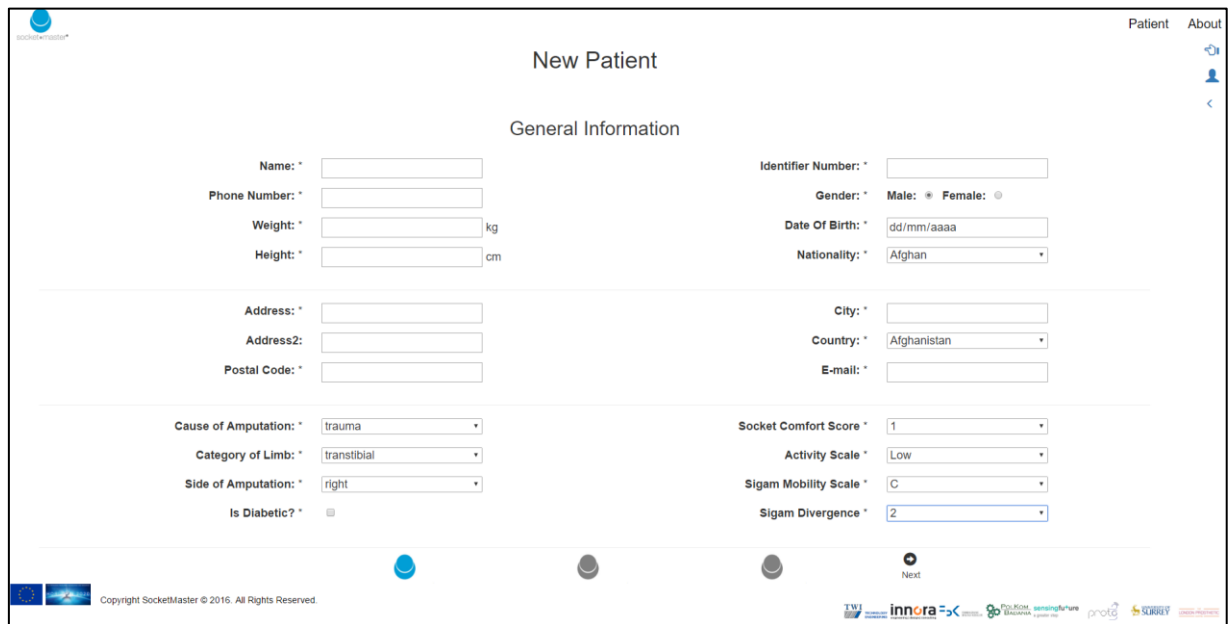
Figura 5-3 - Interface da página principal do sistema (Fonte: Elaborada pelo autor)

Seguidamente são apresentadas as interfaces mais importantes do sistema:

- Interface para inserção de pacientes;
- Interface para listagem de pacientes;
- Interface para mostrar dados de um paciente;
- Interface para gestão das consultas de um paciente;
- Interface para visualizar a imagem tridimensional do coto do paciente.

5.2.1 Interface para inserção de pacientes

Uma das sugestões após o primeiro contato da empresa com os *mockups* do sistema foi a de criar três etapas para o processo de inserção de um paciente. Desta forma existem três interfaces distintas para registar as diferentes características de cada paciente: informações gerais (Figura 5-4), medidas básicas (Figura 5-5) e propriedades mecânicas Figura 5-6).



New Patient

General Information

Name: *

Phone Number: *

Weight: * kg

Height: * cm

Identifier Number: *

Gender: * Male: ☒ Female: ☐

Date Of Birth: * dd/mm/aaaa

Nationality: *

Address: *

Address2:

Postal Code: *

City: *

Country: *

E-mail: *

Cause of Amputation: *

Category of Limb: *

Side of Amputation: *

Is Diabetic? * ☐

Socket Comfort Score *

Activity Scale *

Sigam Mobility Scale *

Sigam Divergence *

Next

Copyright SocketMaster © 2016. All Rights Reserved.

Figura 5-4 - Interface para inserção de paciente (Informações gerais) (Fonte: Elaborada pelo autor)



New Patient

Basic Measurements

Circumference *

Flexion Angle: * °

Patellar tendon distance to Rest of Limb: mm

Adduction Angle: * °

Abduction Angle: * °

Patellar tendon distance to Floor: mm

Previous

Next

Copyright SocketMaster © 2016. All Rights Reserved.

Figura 5-5 – Interface para inserção de paciente (Medidas básicas) (Fonte: Elaborada pelo autor)

SocketMaster

Patient About

New Patient

Mechanical Properties

Force: N

Max Pressure: Pa

Thickness: mm

Muscle: mm

Contact Area: m²

Displacement: mm

Average Pressure: Pa

Elastic Module: mm

Fat: mm

Diameter: cm

Notes:

Add Patient

Previous

Copyright SocketMaster © 2016. All Rights Reserved.

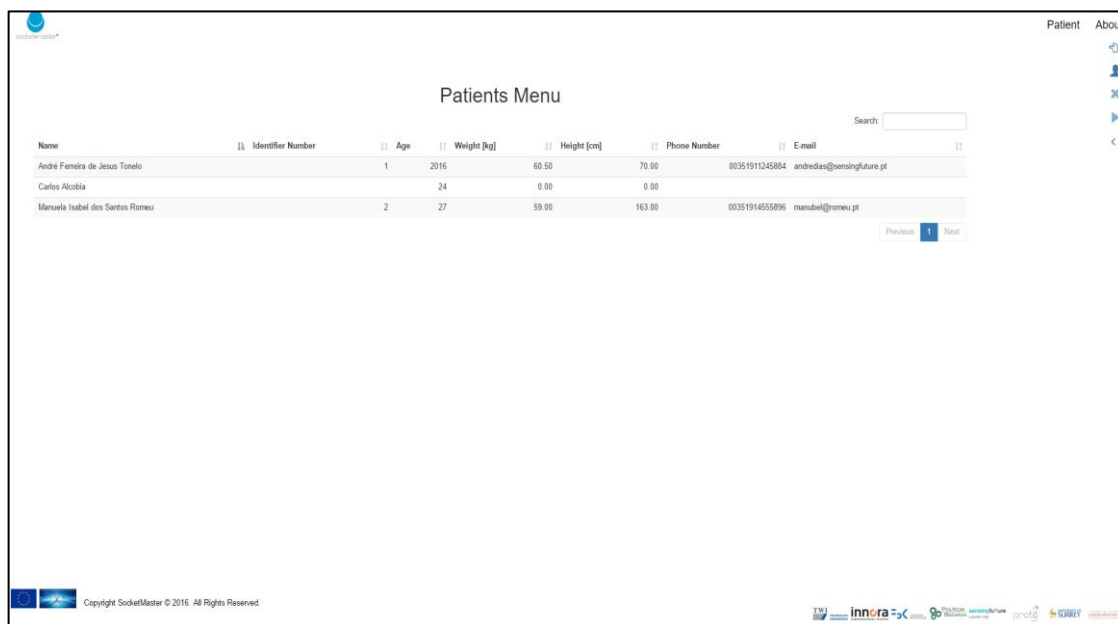
T&T innora

Figura 5-6 – Interface para inserção de paciente (Propriedades mecânicas) (Fonte: Elaborada pelo autor)

Como é visível nas figuras anteriores, o processo de inserção de novos pacientes encontra-se dividido em três etapas. Por forma a implementar uma interface amigável para o utilizador, foi desenhada uma barra de progresso, evidenciando a evolução do processo de inserção.

5.2.2 Interface para listagem de pacientes

A interface para listar os pacientes registados no sistema é de grande importância, porquanto é através desta que é possível visualizar várias informações dos pacientes como por exemplo: nome, número de utente, idade, peso, altura, número de telefone e endereço de e-mail. Através desta listagem é possível selecionar um paciente para posteriormente fazer ações sobre o mesmo, de entre as quais, visualizar todos os dados associados ao paciente ou removê-lo do sistema. A interface para a listagem de pacientes encontra-se representada na Figura 5-7.



Patients Menu

Search:

Name	Identifier Number	Age	Weight [kg]	Height [cm]	Phone Number	E-mail
André Ferreira de Jesus Tonelo	1	2016	60.50	70.00	00351911245804	andredias@sensingfuture.pt
Carlos Alcobia		24	0.00	0.00		
Manuela Isabel dos Santos Romeu	2	27	59.00	163.00	00351914555896	manubel@romeu.pt

Previous 1 Next

Copyright SocketMaster © 2016. All Rights Reserved.

Figura 5-7 - Interface para listagem de pacientes (Fonte: Elaborada pelo autor)

Esta interface é provida com funcionalidades de pesquisa de pacientes, ordenação da tabela pelas colunas da mesma (por ordem crescente e decrescente), paginação da tabela, mostrando apenas 15 registos por página e possibilidade de seleccionar um registo da tabela. Todas estas funcionalidades foram implementadas recorrendo ao *plugin DataTables* descrito no subcapítulo 4.4.3.

5.2.3 Interface para mostrar dados de um paciente

De modo a apresentar as informações guardadas de um paciente, implementou-se a interface presente na Figura 5-8. Esta interface encontra-se dividida em cinco zonas, assim, é possível demonstrar as informações separadamente de acordo com o processo de inserção (informações gerais, medidas básicas e propriedades mecânicas), as restantes duas zonas expõem um resumo das consultas do paciente e as notas associadas ao mesmo.

The screenshot shows a web application interface for patient data management. At the top, the patient's name "André Ferreira de Jesus Tonelo" is displayed. Below this, there are five main sections, each with a header bar and a list of data points:

- General Information:** Name: André Ferreira de Jesus Tonelo, Age: 2016, Height: 70.00 cm, Weight: 60.50 kg, Nationality: Portuguese, Gender: Male.
- Basic Measurements:** Circumference: 10 cm, Flexion Angle: 11.20°, Adduction Angle: 11.20°, Abduction Angle: 10.00°, Perineum to distal residual limb end: 10.00 mm, Perineum to Knee centre: 11.00 mm, Perineum to floor: 12.00 mm, A.P.: 11.00 mm, M.L. fleshy: 10.00 mm, M.L. bony: 10.00 mm.
- Mechanical Properties:** Force: 11.00 N, Displacement: 12.00 mm, Max Pressure: 12.00 Pa, Average Pressure: 13.00 Pa, Thickness: 21.00 mm, Elastic Module: 22.00 mm, Muscle: 21.00 mm, Fat: 12.00 mm, Contact Area: 12.00 m², Diameter: 12.00 cm.
- Notes:** A table with columns for Date and Description. The first entry is dated 2016-05-19 with the description "Primeira consulta". Navigation buttons "Previous", "1", and "Next" are visible.
- Sensors Data:** First Consult: 2016-05-19, Last Consult: 2016-05-19, Total of 1 Consults. Below this, there are two icons: "Indenter Data" (a mechanical indenter) and "SocketMaster Data" (a prosthetic socket).

At the bottom of the interface, there is a copyright notice: "Copyright SocketMaster © 2016. All Rights Reserved." and a row of logos for various partners and sponsors.

Figura 5-8 - Interface para mostrar informações dos pacientes (Fonte: Elaborada pelo autor)

As três secções que mostram dados registados do paciente têm nesta interface um botão que permite a sua edição. O campo relativo às notas registadas para o paciente permite ainda inserir novas notas, bastando para tal clicar no ícone com o sinal de mais (+). A última secção desta interface permite visualizar a data da primeira e da última consulta do paciente, assim como o número total de consultas de determinado paciente. Neste campo existem ainda dois botões que permitem fazer novas aquisições quer do *Indenter* quer do *Socket Master*.

5.2.4 Interface para gestão das consultas de um paciente

Para fazer a gestão das consultas dos pacientes criou-se uma interface para listar todas as consultas associadas a um paciente. Através desta interface é possível visualizar a data e a descrição associada a cada uma delas. A listagem é feita recorrendo a uma tabela, também esta construída com as potencialidades do *plugin* DataTables mencionado já anteriormente. Assim, os registos presentes podem ser seleccionados para remoção, edição ou visualização

das informações de forma mais detalhada. Na Figura 5-9 encontra-se a interface para gestão das consultas de um paciente.

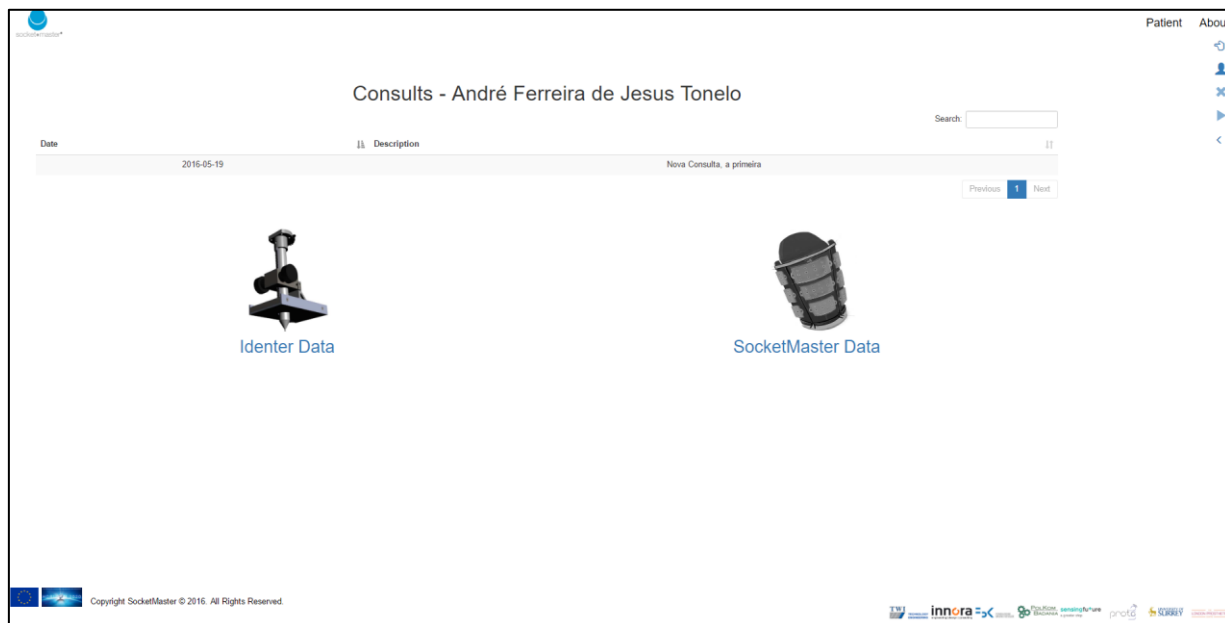


Figura 5-9 - Interface para gestão de consultas de um paciente (Fonte: Elaborada pelo autor)

Ao analisar-se esta interface, constata-se que esta página é provida com dois botões para realizar aquisições rápidas do *Indenter* e do *SocketMaster* respetivamente.

5.2.5 Interface para visualizar a imagem tridimensional do coto do paciente

Para posteriormente ser visível a imagem tridimensional do coto do paciente, foi necessário implementar uma interface para visualizar um modelo tridimensional guardado com a extensão STL. No entanto as funcionalidades de edição ainda não se encontram implementadas, será estudada futuramente uma forma de integrar o *Matlab* no projeto para que, seja possível não só visualizar o modelo, mas também manuseá-lo.

5.3 Visualização tridimensional do coto do paciente

Para auxiliar o desenvolvimento da funcionalidade para visualização e manipulação de um modelo tridimensional do coto do paciente foi necessário definir quais os dados da estrutura. Esta será composta por três partes mais simples de desenhar, parte inferior (semiesfera), parte do meio (cilindro) e parte superior (cilindro com algumas modificações). Na Figura 5-10 encontram-se ilustradas as três divisões propostas para representar o modelo.

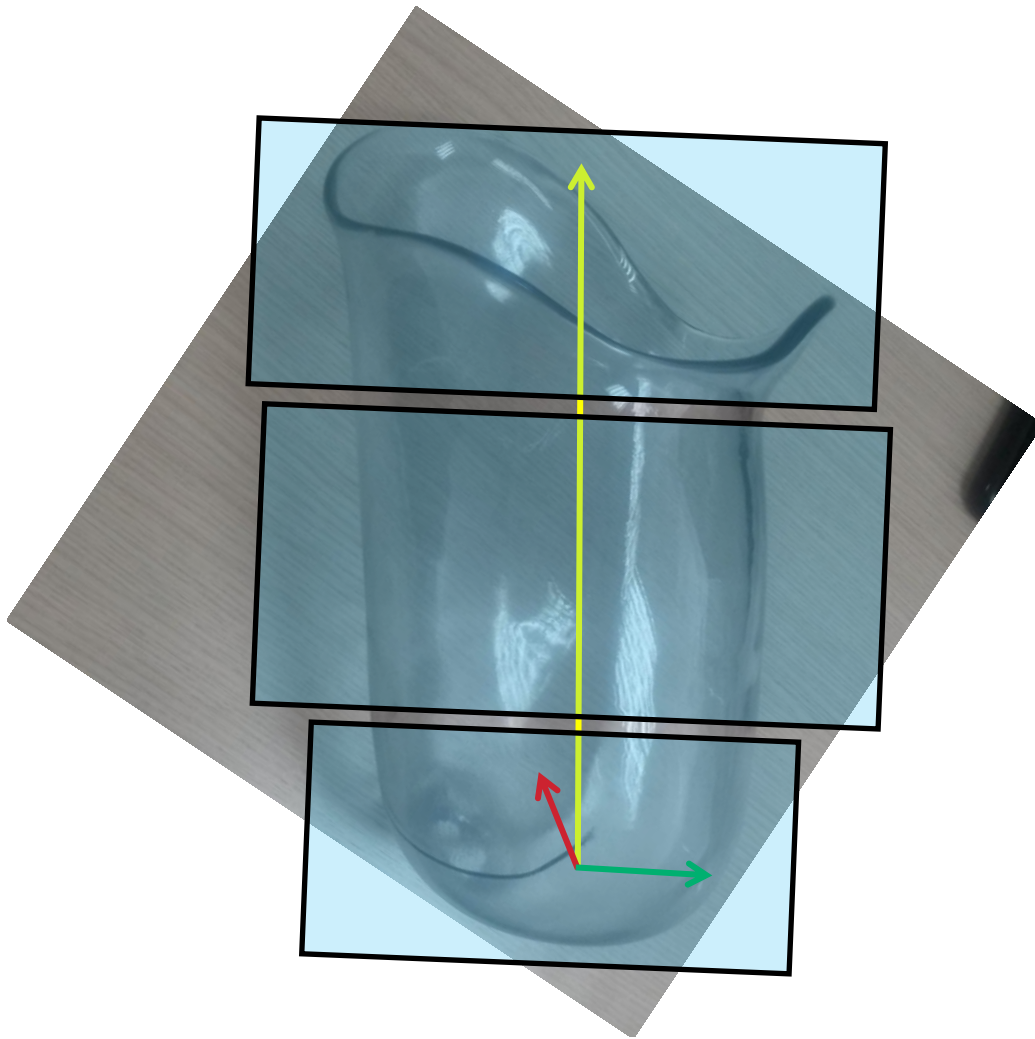


Figura 5-10 - Particionamento do socket (Fonte: Elaborada pelo consórcio)

5.3.1 Definição matemática das divisões do coto

Para a parte inferior é assumida uma semiesfera, sendo os dados dispostos recorrendo a coordenadas cilíndricas (latitude, longitude e altura), ou seja (θ, ϕ, ρ) , onde,

[1]

$$\theta = i \times \Delta\theta$$

Assumindo $\Delta\theta = 10^\circ$ e o i correspondendo a uma sequência de 0 até 35 inclusive, conseguem-se alcançar as coordenadas do perímetro de uma circunferência com um determinado raio p .

[2]

$$\phi = j \times \Delta\phi$$

Assumindo $\Delta\phi = 10^\circ$ e o j corresponde a uma sequência de 1 até 8 inclusive, é possível alcançar as coordenadas da parte inferior de uma esfera, obtendo assim a semiesfera inferior. Aplicando as fórmulas, para $\phi = 10^\circ$ irão ser obtidos 36 pontos correspondendo às longitudes $0^\circ, 10^\circ, 20^\circ, \dots, 350^\circ$, todos estes pontos para uma mesma latitude 10° , na segunda iteração

com $\phi = 20^\circ$ obtém-se mais 36 pontos mas agora correspondendo à latitude de 20° , iterando desta forma até ao $\phi = 80^\circ$ ir-se-ão obter as coordenadas correspondentes à latitude de 80° . No total serão calculados 288 pontos para desenhar uma semiesfera.

As seguintes fórmulas permitirão obter as coordenadas cartesianas da semiesfera:

[3]

$$x = x_0 + \rho \times \cos(i\Delta\phi) \times \cos(j\Delta\theta)$$

[4]

$$y = y_0 + \rho \times \cos(i\Delta\phi) \times \sin(j\Delta\theta)$$

[5]

$$z = z_0 + \rho \times \sin(i\Delta\phi)$$

As coordenadas x_0 , y_0 e z_0 correspondem às coordenadas do centro da esfera.

Para calcular os valores das coordenadas da parte central (cilindro), é necessário definir uma linha vertical contendo os centróides, assim, será possível determinar as coordenadas da superfície do cilindro sabendo o seu centro respetivo. Iterando o cilindro na sua altura, será possível determinar todos os pontos. O seu respetivo centro será composta por 11 pontos ($C_0 \dots C_{10}$). O espaçamento dos centróides será calculado tendo por base a seguinte equação:

[6]

$$\Delta h = \frac{(h_{10} - h_0)}{10}$$

A altura de cada centróide será calculada por:

[7]

$$h = h_0 + i\Delta h$$

Sendo i uma sequência de 0 até 10 inclusive. Para determinar as coordenadas da superfície é necessário iterar o cilindro na sua altura. Para cada altura irão ser calculados 36 pontos. É crucial conhecer o raio do cilindro a cada altura, uma vez que se está perante um cilindro não perfeito. As fórmulas para calcular as coordenadas cartesianas são descritas de seguida:

[8]

$$x = x_{ci} + r \times \cos(j\Delta\theta)$$

[9]

$$y = y_{ci} + r \times \sin(j\Delta\theta)$$

[10]

$$z = z_{ci}$$

Nas fórmulas anteriores r representa o raio da respetiva circunferência. A variável i representa uma sequência de 0 até 10 inclusive, a variável j representa uma sequência de 0 até 35 inclusive. Para desenhar esta parte serão necessários 288 pontos.

Para obter os pontos da parte superior (cilindro modificado), é necessário considerar a altura de cada um dos pontos, assim deve ter-se em conta quer a altura do ponto quer o raio. Serão considerados 360 pontos, 36 por cada camada considerando um total de 10 camadas. A primeira camada coincidirá com a circunferência superior do cilindro da parte do meio, e a décima camada coincidirá com a superfície mais alta do *socket*. Note-se que a altura das várias camadas não é constante como no cilindro, uma vez que, esta parte superior é irregular, assim, para a mesma camada, existirão pontos com alturas diferentes, dependendo da orientação a que este se encontre (longitude).

5.3.2 Modelo 3D do coto em *Matlab*

A tradução do modelo matemático definido anteriormente foi uma tarefa bastante desafiadora e complexa, sendo que consumiu muito tempo, sendo necessárias bastantes tentativas para a representação do modelo 3D. Com o objetivo de construir este modelo foram pesquisadas algumas ferramentas entre as quais o *Matlab*.

Analisaram-se as capacidades do *Matlab* para construir gráficos tridimensionais, tendo sido utilizadas as fórmulas x , y e z das três partes do *socket*, e obtendo-se o respetivo modelo. Através da ferramenta do *Matlab* pode-se criar um modelo 3D com mapeamento de cores pois os dados do modelo são trabalhados como uma matriz de valores, que depois assumem uma forma 3D. Assim, analisando-se os pontos de pressão adquiridos pelos sensores, é possível redefinir esta matriz por forma a construir uma correta representação de cores como é visível na Figura 5-11.

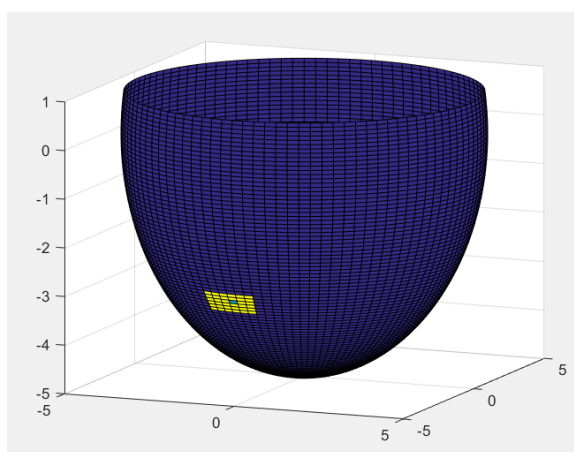


Figura 5-11 - Visualização do modelo 3D do Socket com um ponto de pressão simulado (Fonte: Elaborada pelo autor)

A título de conclusão observa-se que criar o modelo teórico do *socket* é uma tarefa um pouco complicada. Através das potencialidades do MatLab, é possível representar o modelo teórico, no entanto para representar o modelo real de cada paciente é necessário avaliar as

coordenadas dos 36 sensores, e através de regressões lineares obter o modelo real do paciente. Note-se que o *Socket Master* será composto por 36 sensores, e o modelo 3D deve ser criado tendo como ponto de partida as coordenadas de cada um destes sensores.

5.3.3 Integração do *software* com o módulo de *hardware*

A integração do módulo de *software* com o de *hardware* ainda não aconteceu, uma vez que estes dois módulos encontram-se a ser desenvolvidos por entidades distintas. Importa salientar que quer o desenvolvimento do *hardware* quer o desenvolvimento do *software* estão a seguir um planeamento redigido pelo projeto e retificado sempre que necessário através de reuniões presenciais ou via *Skype* por forma a agilizar o processo de comunicação entre as várias entidades do consórcio.

Os testes de integração já foram planeados (ver secção 6.2), de forma a quando a integração do módulo de *hardware* e *software* estiver concluída, se possam implementar os testes, por forma a testar as várias funcionalidades do sistema.

6 Testes e Validação

Neste capítulo pretende-se expor os testes a que foi sujeita a aplicação para que a implementação dos requisitos funcionais e não funcionais fosse verificada e validada. Primeiramente são detalhados os testes unitários, seguindo-se os testes de integração, e no final os testes de usabilidade.

Um ponto relevante aquando do desenvolvimento de *software* consiste em decidir quando testar e como testar o *software* ao longo do seu desenvolvimento. A escolha do tipo de testes e quais os momentos para os aplicar é, então, um aspeto muito importante a ter em conta.

Na Figura 6-1 exemplifica-se como relacionar os vários tipos de testes com as várias fases de desenvolvimento de *software*.

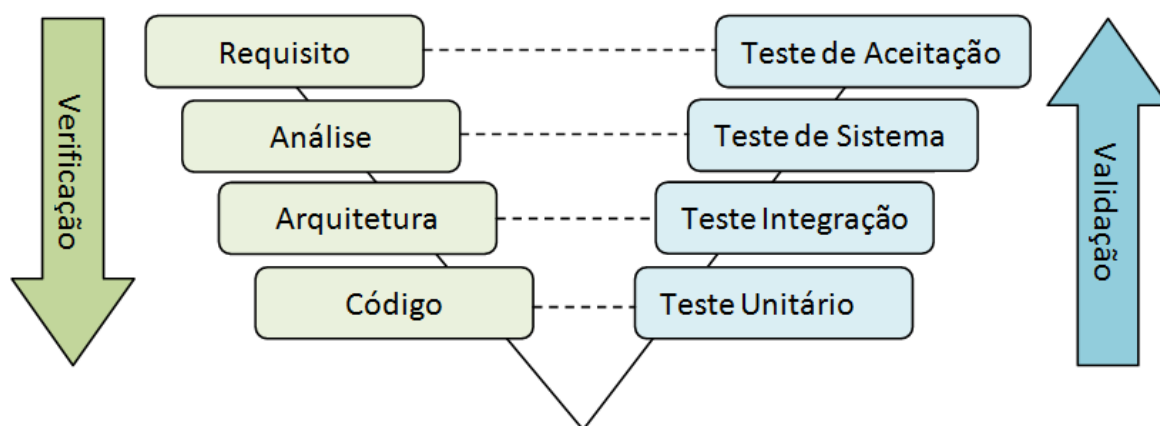


Figura 6-1 - Modelo "V" (Fase de Desenvolvimento Vs. Fase de Testes) (Fonte: Aparecido, 2015)

Através da figura anterior é possível visualizar que os vários tipos de testes, verificam uma certa fase do desenvolvimento do *software*. Desta forma para validar o código, existem os testes unitários, os testes de integração tem como objetivo verificar a arquitetura do sistema (as diversas camadas, o seu fluxo e interligação), através dos testes de sistema é possível verificar o sistema como um todo, e por fim os testes de aceitação visam verificar os requisitos do sistema (Aparecido, 2015). De seguida serão abordados cada um dos quatro tipos de testes e feita uma descrição superficial dos mesmos.

Testes Unitários: têm como objetivo verificar cada unidade que compõe o *software* de uma forma isolada;

Testes de Integração: têm como objetivo testar vários módulos, validando a correta comunicação entre eles. Este tipo de testes visa encontrar falhas provenientes de interfaces ou dependências entre módulos;

Testes de Sistema: têm como objetivo testar o sistema na sua totalidade;

Testes de Aceitação: têm como objetivo determinar se o *software* funciona da maneira esperada, conforme consta na especificação dos requisitos. Este tipo de testes valida que cada um dos requisitos se encontra a funcionar com o comportamento definido.

Os testes de sistema ainda não foram feitos nem planeados uma vez que o projeto ainda se encontra desenvolvido em módulos separados.

6.1 Testes Unitários

Os testes unitários concentram-se na verificação das unidades mais pequenas do *software*, como por exemplo em testar pequenas funções ou métodos, através deste tipo de testes é possível verificar e testar as várias funções do *software*.

Este tipo de testes apresenta várias vantagens, das quais se destaca: a prevenção para o aparecimento de “*Bug’s*” provenientes de código mal escrito, em especial com erros lógicos; aumento do nível de confiança do *software*; teste de situações quer de sucesso quer de falha; teste dos requisitos do projeto.

Os testes unitários encontram-se organizados da seguinte forma, existem vários *Test Case’s* (um para cada classe do sistema), possuindo cada um destes, vários testes para diferentes métodos da mesma classe. Os *Test Case’s* são agrupados num *Test Suite*, tendo como função agrupar vários *Test Case’s*, e executá-los todos partindo de uma única invocação.

Por forma a criar testes para o sistema de uma forma imparcial ao mesmo, estes deveriam ser escritos e executados por uma equipa destinada a testar a aplicação, normalmente designada por equipa de teste, no entanto a não existência desta mesma equipa no projeto em questão, levou a que o próprio Estagiário, criasse e executa-se os vários testes. Desta forma, o Estagiário adquiriu um maior controlo sobre o sistema, permitindo a imediata correção de *bug’s* sempre que estes sejam detetados pelo conjunto de testes implementados.

Os testes unitários devem ser capazes de testar todas as funcionalidades do sistema, devendo-se iniciar por implementar os mais simples, e progressivamente ir implementando testes mais complexos, implementando o mínimo suficiente para testar uma determinada função, por exemplo, não se vão implementar dez condições de testes se três forem suficientes (Pimentel, 2014). Estas foram as diretrizes genericamente seguidas para a realização dos testes unitários.

Ao longo do projeto foi definido um plano de testes por forma a validar e aumentar o grau de confiabilidade do sistema desenvolvido, na Tabela 6-6 podem se analisar os vários testes planeados para o sistema. Estes podem também ser consultados no apêndice F com um grau de detalhe mais elevado.

Tabela 6-6 - Casos de Testes Unitários

ID	Casos de Teste	Resultado
CT-001	Registo de paciente	✓
CT-002	Remover paciente	✓
CT-003	Alterar dados do paciente	✓
CT-004	Listar pacientes registados	✓
CT-005	Pesquisa de pacientes	
CT-006	Adicionar nova aquisição de dados	✓
CT-007	Remover aquisição de dados	✓
CT-008	Registar descrição na aquisição de dados	✓
CT-009	Alterar descrição na aquisição de dados	✓
CT-010	Inserir notas	✓
CT-011	Remover notas	✓
CT-012	Visualização da prótese numa imagem 3D	

Importa salientar que apenas o caso de teste CT-005 e o CT-012 não foram implementados para correr automaticamente, no entanto estes foram executados de uma forma manual pelo testador.

6.2 Testes de Integração

Quando todos os componentes de um *software* forem testados, surge uma pergunta: quando estiverem todos integrados, continuará tudo a funcionar como pretendido? Os testes unitários contribuem para assegurar a correção dos componentes a nível individual, não sendo capazes de garantir que as dependências funcionais entre componentes estejam perfeitamente implementadas. Para fazer este tipo de testes, utilizaram-se testes de integração.

Os testes de integração, são os testes executados aquando da integração dos vários módulos de *software*, com o objetivo de se testarem as interações entre módulos. Esta fase de testes inicia-se com os vários módulos já testados individualmente pelos testes unitários e visa testar as várias interações entre módulos, por forma a garantir uma correta comunicação. Através deste tipo de testes, é possível descobrir erros nas interfaces entre componentes (Testes de Software, 2016).

Uma vez que o projeto tem por base um consórcio, estando vários parceiros a trabalhar para o mesmo objetivo, no entanto em tarefas distintas, um plano de testes de integração é inevitável, para o sucesso do projeto. Apesar dos vários módulos serem testados individualmente, recorrendo aos testes unitários, é imprescindível recorrer aos testes de integração, para garantir que após os vários módulos se encontrarem integrados num único sistema, a comunicação necessária entre módulos irá funcionar de acordo com o previsto.

Dessa forma, foi planeado um conjunto de testes de integração que visam o correto funcionamento entre módulos. Na Tabela 6-7 encontra-se listado o planeamento de testes de integração, estes podem ser consultados com maior detalhe no apêndice G.

Tabela 6-7 - Casos de Testes de Integração

ID	Casos de Teste de Integração	Resultado
CTI-001	Aplicar uma força num sensor de pressão e validar que é lida e inserida na BD corretamente	
CTI-002	Colocar um coto artificial no <i>Socket Master</i> e validar que os valores de pressão adquiridos pelos vários sensores são adquiridos e registados na BD	
CTI-003	Efetuar medidas de indentação num coto artificial e validar que os valores são registados na BD	
CTI-004	Iniciar uma aquisição de dados com o <i>Socket Master</i> , desligar a sua conexão e validar que o <i>software</i> avisa a perda de comunicação	
CTI-005	Iniciar uma aquisição de dados com o <i>Identer</i> , desligar a sua conexão e validar que o <i>software</i> avisa a perda de comunicação	
CTI-006	Colocar um sensor numa determinada posição, verificar que os dados das suas coordenadas são registadas na BD corretamente	
CTI-007	Iniciar uma aquisição de dados sem nada estar em contato com o <i>Socket Master</i> , validar que são registados valores nulos de pressão na BD	
CTI-008	Iniciar uma aquisição de dados do <i>Identer</i> sem nenhuma amostra, validar que são registados valores nulos na BD	

Os testes acima identificados ainda não foram implementados devido à integração do sistema ainda não ter ocorrido. Os módulos físicos “*Socket Master*” e “*Identer*” são desenvolvidos por duas entidades distintas pertencentes ao consórcio, no entanto os seus desenvolvimentos ainda não se encontram concluídos. Por esse motivo ainda não se terem efetuado testes de integração, permanecendo estes apenas planeados, podendo ainda ser alterados de acordo com os novos desenvolvimentos.

6.3 Testes de Usabilidade

O objetivo dos testes de usabilidade é o de verificar o que está a funcionar mal no sistema e o que necessita de ser melhorado, de forma a permitir uma navegação: segura, eficaz, eficiente e que gere satisfação. Para tal existem diferentes tipos de testes e metodologias, genericamente classificados em Métodos Analíticos e os Métodos Empíricos. De entre os Métodos Analíticos, e com particular interesse para este trabalho destacam-se a Avaliação Heurística e a Avaliação Preditiva. Dos Métodos Empíricos destacam-se a Avaliação com utilizadores.

6.3.1 Métodos Analíticos

A Avaliação Heurística consiste num método para avaliar a interface de forma rápida, barata e simples. Rápida, sendo normalmente suficiente um dia ou menos para as aplicar; Barata pois não precisa de laboratórios ou equipamento especializados; Simples, sendo fácil de aprender. No entanto, quanto mais experientes forem os avaliadores melhores resultados se obterão. Normalmente na sua aplicação utilizam-se as normas de usabilidade de Jakob Nielsen (Nielsen, 1992). Esta técnica desenvolvida por Jakob Nielsen, é útil para encontrar problemas de interação, utilizando para tal um pequeno conjunto de avaliadores (≤ 5) que examinam a interface e eventuais interações com o utilizador, verificando a aderência a heurísticas de usabilidade. Neste tipo de avaliação, diferentes avaliadores poderão detetar problemas diferentes, mas em regra bons avaliadores detetarão problemas similares que serão consolidados em relatório. Esta é uma técnica que tanto pode ser aplicada a produtos finais como a protótipos funcionais ou esboços. Neste projeto, e não tendo peritos de usabilidade, desde uma fase muito precoce que se tentaram seguir as normas de Nielsen de forma a elaborar os protótipos atendendo às seguintes recomendações:

Visibilidade do sistema (H2.1). Tentou-se, sempre que possível manter o utilizador informado sobre onde estava ou o que estava a acontecer no sistema, através de respostas (textuais, ou outras) apropriadas e dentro de um tempo razoável.

Correspondência entre o sistema e o mundo real (H2.2). Fez-se um esforço para que o sistema “falasse” a linguagem do utilizador, com palavras, frases e conceitos que lhe são familiares, em vez de usar a linguagem do sistema, e de uma forma lógica e natural que evite termos técnicos.

Controlo e liberdade do utilizador (H2.3). Tentou-se que o utilizador se sentisse livre ao utilizar a aplicação desenvolvida, de forma a poder realizar as ações pretendidas da forma desejada. Para tal, tentou-se que o sistema fornecesse formas do utilizador sair de estados não desejados sem ter que passar por diálogos extensos ou caminhos inflexíveis. Este aspeto foi conseguido através da utilização de opções como “Back”, “Cancelar” ou “Sair”.

Consistência e normas (H2.4). Tentou-se que a aplicação fosse consistente em todas as suas vertentes, grafismo, terminologia, ações e especialmente em relação às expectativas do utilizador. Em termos de grafismo, tentou-se definir, cores, localização, forma e posicionamento de elementos de maneira consistente em todos os ecrãs, respeitando esta definição em toda a interface. Em termos de terminologia, evitando termos, situações e ações diferentes com o mesmo significado, seguindo-se as convenções das plataformas, normas e regras.

Prevenção de erros (H2.5). Tentou-se desenhar o sistema de forma cuidadosa prevenindo os problemas antes de acontecerem, pois melhor que uma boa mensagem de erro é evitar o erro. A prevenção de erros esteve patente, por exemplo na forma de entrada de dados, evitando, sempre que possível, a entrada de dados de forma manual, através do teclado e verificando os valores introduzidos.

Reconhecer em vez de lembrar (H2.6). Tentou-se, sempre que possível, tornar os objetos, ações e opções visíveis e fáceis de identificar, para que o utilizador não tenha que recordar

informação de uma parte do sistema para outra. Teve-se algum cuidado na escolha dos ícones e dos nomes para os diversos elementos da interface para que estes fossem facilmente reconhecidos e associados às suas funções.

Flexibilidade e eficiência na utilização (H2.7). De forma a acomodar diferentes níveis de experiência fornecendo “aceleradores” para os utilizadores experientes que são invisíveis para os novatos. Houve o cuidado de posicionar os elementos na interface de forma a permitir aos utilizadores adotarem facilmente e de forma expedita ações frequentes.

Desenho estético e minimalista (H2.8). Mesmo não havendo *designers* envolvidos, tentou-se seguir algumas regras de forma a obter interfaces com um bom desenho. O princípio basilar seguido foi o da simplicidade, seguindo o princípio do “Menos é mais” (menos para aprender, para perceber mal, para distrair...), suprimindo toda a informação irrelevante ou raramente necessária. Para tal tentaram-se diálogos com textos concisos. Cada unidade extra de informação num diálogo compete com unidades relevantes de informação e diminui a sua visibilidade relativa. Também foi feito um esforço para que a ordem de acesso à informação estivesse de acordo com as expectativas do utilizador (destaque-se, por exemplo, o preenchimento dos dados de endereço) ou que a informação relacionada estivesse graficamente agrupada.

Ajudar os utilizadores a reconhecer, diagnosticar e recuperar erros (H2.9). As possibilidades de ocorrência de erros não devem nunca ser descuradas, pelo que foi planeado qualquer tipo de ocorrência de erro para as mais diversas situações. Para tal, planearam-se mensagens de erro expressas em linguagem simples (sem códigos), indicando de forma precisa o problema, e sugerindo uma solução construtiva.

Documentação e Ajuda (H2.10). Apesar de as regras indicarem, em geral, que um sistema em que a usabilidade esteja presente não necessite de ajuda ou documentação e que o sistema não deve depender dela para ser utilizado, esta deve estar disponível para quando necessária. Porém, deve estar concentrada nas tarefas dos utilizadores, e deve ser fácil de pesquisar e interpretar, concisa, focada nas tarefas, facilmente acessível, listando os passos concretos para concretizar a tarefa.

Os métodos de avaliação preditiva medem o desempenho do utilizador sem o testarem realmente. São particularmente úteis quando não é possível ou aceitável efetuar testes com utilizadores. Isto porque o custo de construir uma interface e testá-la com vários utilizadores até descobrir todos os problemas pode ser inaceitável não apenas do ponto de vista económico como temporal. Este método utiliza procedimentos rápidos e informais que permitem avaliar produtos sem envolver os utilizadores, utilizando modelos cognitivos dos utilizadores para testar o desenho. A grande desvantagem deste método é o facto de ser apenas útil para sistemas com tarefas previsíveis. Neste caso, a tarefa divide-se em duas fases: A determinação da sequência pela qual a atividade é realizada e a análise dos passos da sequência para determinar medidas de usabilidade. Dependendo do método e objetivos pretendidos podem-se analisar aspetos como o tempo para realizar cada passo ou os passos onde podem ocorrer erros.

Apesar de existirem diferentes métodos e variantes, nomeadamente os Modelos Hierárquicos que representam a estrutura da tarefa e dos objetivos (GOMs - *Goals, Operators, Methods, Selection*), os Modelos Físicos e de Dispositivo que representam capacidades psicomotoras (KLM - *Keystroke-Level Model*), os Modelos Linguísticos que representam a gramática de interação utilizador-sistema, considera-se que os percursos cognitivos são os de maior utilidade neste projeto. Os percursos cognitivos, fundamentados na Engenharia Cognitiva, objetivam fazer uma correspondência entre a conceptualização de uma tarefa por parte dos utilizadores e dos projetistas. Este método serve então para identificar problemas de formação de objetivos, identificar problemas de especificação da ação ou identificar problemas de execução da ação. A verificação de aspetos como a escolha adequada (ou inadequada) de termos (vocabulário utilizado) e do *feedback* adequado (ou inadequado) para as consequências de uma ação serão também alvo de análise. Assim, o avaliador necessita para a sua execução de fazer suposições sobre a população de utilizadores bem como do contexto em que executará as tarefas na aplicação, de descrição das tarefas representativas a serem executadas e da lista das ações necessárias para executar cada uma das tarefas. Assim, executa cada tarefa passo a passo, revendo as ações necessárias para executá-la, prevendo como os utilizadores agiriam e os problemas que teriam. Desta forma, o avaliador constrói histórias plausíveis sobre a interação de um utilizador típico com a interface, com base nos cenários de tarefas selecionados. Posteriormente, simulam a execução da tarefa, efetuando uma série de perguntas sobre cada passo, anotando pontos-chave, como: o que o utilizador precisa de saber antes de realizar a tarefa ou o que o utilizador deve aprender ao realizar a tarefa. Durante essa análise, o avaliador deteta eventuais problemas que o utilizador irá enfrentar, propondo soluções típicas para as falhas mais comuns que poderão passar por eliminar determinada ação, combinando-a com outra ou deixando o sistema assumi-la; fornecer uma instrução ou indicação de que a ação precisa ser realizada; modificar determinada parte da tarefa para que o utilizador entenda a necessidade desta ação; atribuir um operador (ícone, botão, ...) mais óbvio para a ação; entre outros.

Neste projeto, e desde uma fase muito precoce se tentou seguir, de uma forma aligeirada o percurso cognitivo. Assim, na fase de definição dos *mockups* estiveram sempre presentes suposições sobre os utilizadores e o contexto em que executaria as tarefas na aplicação, decidindo sobre os diversos elementos de interação com base na construção de histórias plausíveis sobre a interação de um utilizador típico com a interface.

6.3.2 Métodos Empíricos

Na realidade só é possível afirmar que uma IU é boa quando for usada. O problema dos métodos que recorrem a peritos é que estes podem saber demais, podem não saber o suficiente sobre as tarefas, podem não saber o suficiente sobre os utilizadores tornando muitas vezes difícil prever o que utilizadores reais fazem. Os testes com utilizadores objetivam avaliar a Usabilidade, nomeadamente, a Operabilidade, Aprendizagem e Compreensibilidade da aplicação através da perspetiva do utilizador. Pretende-se essencialmente averiguar se as necessidades do utilizador foram levadas em conta; se o sistema demonstra um comportamento estável; se é apazível, gratificante, frustrante ou cansativo navegar no

sistema; se o sistema tem uma apresentação atraente; se a navegabilidade é fácil, simples, eficiente e intuitiva; se possibilita a ajuda ao utilizador; se as informações/ajuda fornecidas são suficientes; se todos os aspetos são de fácil compreensão ou se existem frases, instruções ou esquemas de navegação que possam confundir ou iludir o utilizador; se são necessários conhecimentos prévios; se estão disponíveis os comandos necessários, entre outros.

Existem várias técnicas para realizar testes com utilizadores, nomeadamente a observação, entrevistas ou aplicação de questionários. Na observação, observam-se os utilizadores a executar as atividades previamente definidas como roteiro. Esta observação pode ser realizada de forma mais ou menos detalhada, com mais ou menos intervenientes, num ambiente controlado (laboratório de usabilidade) ou no contexto habitual de execução de tarefas. Nestas observações pretende-se colher indicadores quantitativos para análise, como o número de ações incorretas, consultas a ajuda, erros repetidos, atividades concluídas, comandos e controlos utilizados, facilidade de movimentação e/ou navegabilidade, questões relacionadas com o aspeto gráfico, facilidade de aprendizagem de utilização do sistema, cansaço provocado, satisfação geral na utilização do produto, entre outros aspetos considerados de interesse. Considera-se que os testes com utilizadores apesar de poderem ser demorados e caros são imprescindíveis neste projeto, pelo que estão já planeados algumas possibilidades no apêndice H, porém questões como o número de testes, as técnicas envolvidas bem como as tarefas a testar são questões a ser discutidas futuramente no consórcio.

As entrevistas podem ter um carácter mais ou menos estruturado ou até mesmo contextual. Normalmente consistem numa conversa com o utilizador baseada num conjunto de perguntas (semi) preparadas, sendo um método mais informal, subjetivo e relativamente barato, podendo ser guiadas para se adaptar ao contexto, para explorar alguns assuntos específicos podendo também ajudar a identificar problemas não previstos. Porém, as entrevistas têm muitas vezes um carácter subjetivo (depende do rumo) e consomem normalmente muito tempo para a sua realização e análise. As entrevistas contextuais são outro exemplo de avaliação normalmente utilizadas para perceber as necessidades dos utilizadores no seu trabalho ou ambiente, de maior utilidade para definir requisitos, planejar e desenhar. Embora o espaço apresente um papel importante para a equipa, pois é lá que se desenrolam as atividades, não se planeia a sua utilização neste projeto.

Frequentemente aplicam-se também questionários pós-teste, para a sondagem da satisfação subjetiva do utilizador (em termos de facilidade de navegação, controlo do sistema, funcionalidades, linguagem utilizada, ...) analisar experiências ou emoções de interação. No apêndice I, encontra-se o modelo proposto para o questionário pós-teste adaptado do Questionário SUMI (*Software Usability Measurement*).

7 Outras atividades desenvolvidas

Para além do principal projeto descrito ao longo deste documento, e como algumas das atividades estão condicionadas temporalmente por atividades que ainda estão em curso no consórcio, foram realizadas outras atividades, como por exemplo atualização e criação de *websites*, ou manutenção de aplicações de diversos produtos, entre outras.

No presente capítulo serão apresentadas as várias atividades extra projeto *Socket Master* que foram desenvolvidas no âmbito do presente estágio.

7.1 *Smartherapy*²

O *Smartherapy* é uma aplicação informática para registo eletrónico baseado na CIF (Classificação Internacional de Funcionalidade, Incapacidade e Saúde). Esta aplicação foi desenvolvida pela *Sensing Future Technologies*, com a tecnologia .NET da Microsoft, recorrendo ao WPF (*Windows Presentation Foundation*) e a uma base de dados implementada em *SQL Server*.

A nível organizacional, o *Smartherapy* está dividido em três áreas: problema, intervenção e reavaliação. O problema conta com quatro campos que provém da própria organização da CIF: funções e estruturas do corpo, atividades e participação, fatores ambientais e fatores pessoais. A intervenção conta com quatro campos: principais problemas e facilitadores, diagnóstico funcional, objetivos e prognóstico, e intervenção. Na reavaliação pretende-se fazer uma comparação entre os itens avaliados inicialmente, com os objetivos que foram definidos através de ações na área da intervenção e com a avaliação que será efetuada no final da implementação do plano de intervenção definido.

Qualquer registo introduzido no *Smartherapy* pode ser a qualquer momento exportado para um formato portátil de documento, normalmente designado de PDF (*Portable Document Format*) ou formato de folha de cálculo, normalmente CSV (*Comma-Separated values*) de modo a permitir trabalhar e partilhar os dados introduzidos.

Esta aplicação conta com um *website* para divulgar o produto, assim como entrar em contacto com a empresa para fazer a sua aquisição. Este *website* encontra-se desenvolvido sobre a *framework* Joomla. Foi sobre este *website* que recaiu uma das tarefas, tendo consistido na reestruturação de todo o *website*, e tendo sido ainda implementada uma versão *trial* do *software*. Nesta versão é possível ao utilizador trabalhar apenas com a área do problema, sendo possível registar os problemas dos pacientes, tendo por base a organização CIF.

A implementação da versão *trial* (www.smartherapy.pt/smt/index.php/pt/registo) (Figura 7-1), teve como base a análise da aplicação já existente, fazendo-se assim uma replicação da mesma, mas numa versão web. Porém, não foi necessária a implementação de persistência de dados numa base de dados, partilhando a base de dados *SQL Server* já existente para aceder

² www.smartherapy.pt

aos vários grupos, capítulos e qualificadores inerentes à CIF. Também nesta versão *trial* é possível exportar os dados para o formato pdf para o utilizador posteriormente poder trabalhar e aceder aos dados.

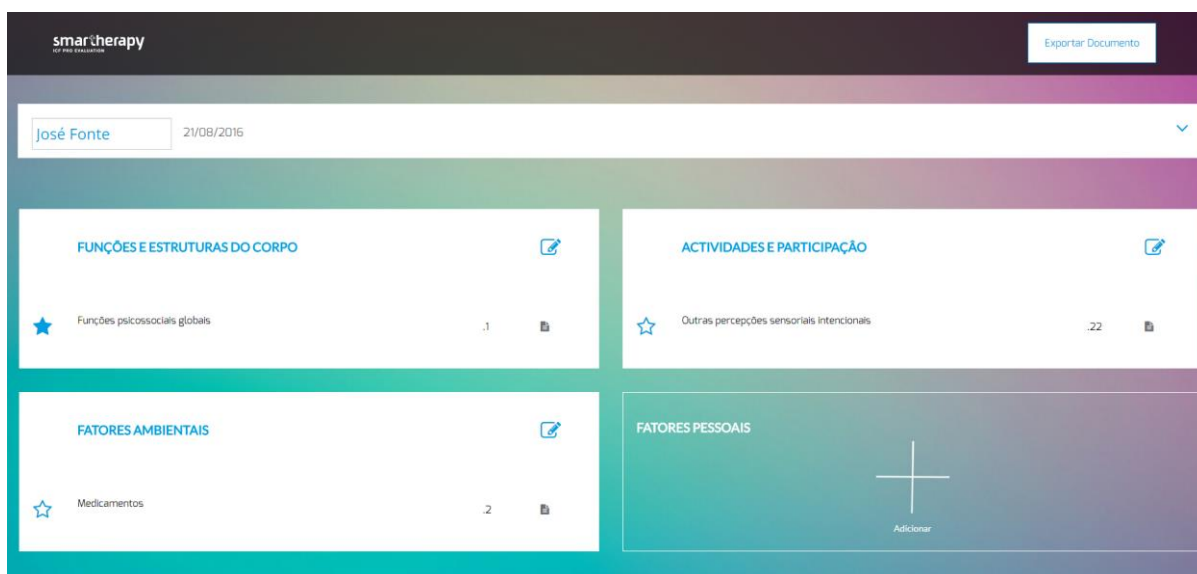


Figura 7-1 - Versão trial implementada no website (Fonte: Elaborada pelo autor)

7.2 FallSensing³

O projeto *FallSensing* é um projeto que se encontra em fase de desenvolvimento (Figura 7-2), financiado pelo programa Portugal2020 e cujo consórcio é composto por três entidades: a *Sensing Future Technologies* (promotor líder), a Escola Superior de Tecnologia da Saúde de Coimbra, e o Instituto *Fraunhofer* Portugal. Este projeto permitirá a avaliação de múltiplos fatores de risco associadas a quedas e a implementação de planos de exercícios para a prevenção de quedas.



Figura 7-2- Projeto FallSensing (Fonte: www.fallsensing.com)

No âmbito deste projeto foi desenvolvido um *website*, disponível em português e inglês, para divulgação e apresentação do projeto, tendo como principais funcionalidades: formulário de contacto, *backend* para introdução de datas e locais de recolhas, rastreios de risco de queda, visualização dos próximos rastreios, visualização dos *tweets* mais recentes feitos com a conta

³ www.fallsensing.com

do *FallSensing*, disponibilização de apresentações, notícias e artigos científicos relacionados com o projeto.

A nível da aplicação do projeto, foi estruturada e implementada a base de dados em SQL Server, bem como implementado um sistema de autenticação recorrendo à ferramenta KeyCloak⁴. Este projeto consiste em várias aplicações, todas elas a partilhar a mesma base de dados, listadas de seguida:

- Aplicação para rastreios (testes funcionais);
- Cenário clínico (Apenas testes funcionais e exercícios terapêuticos com *biofeedback*);
- Cenário de atividades de grupo (Jogos e alguns testes funcionais);
- Cenário de utilização individual (Jogos e alguns testes funcionais).

Irá existir ainda uma plataforma *web* para visualizar os resultados dos testes e jogos realizados nestas quatro aplicações. Aqui nesta plataforma um fisioterapeuta responsável por N pacientes, poderá acompanhar a evolução dos mesmos, analisando os resultados dos exercícios realizados, uma vez que, os pacientes poderão fazer os exercícios através de casa, bastando apenas ter a aplicação para o efeito. Neste portal também será feita toda a gestão de utilizadores registados nas aplicações, e as suas permissões (Figura 7-3).

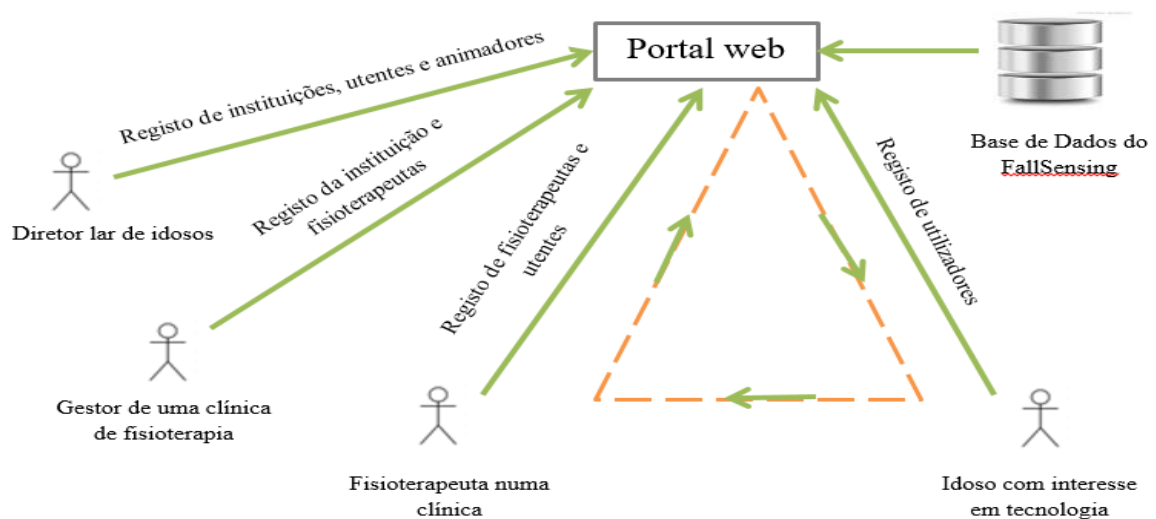


Figura 7-3- Esquema a identificar os vários tipos de utilizadores e interações possíveis no sistema (Fonte: Elaborada pelo autor)

⁴ API para autenticação de aplicações www.keycloak.org

7.3 Website – Sensing Future Technologies⁵

Foram também realizadas algumas tarefas de manutenção e atualização do *website* da própria empresa, também este desenvolvido sobre a *framework* Joomla. As principais tarefas consistiram em: atualização de produtos da empresa; atualização de projetos da empresa e funcionalidade de contato através de formulário.

A nível desta atividade as dificuldades maiores residiram na análise de toda a estrutura do *website* e na compreensão do funcionamento do Joomla. Após esta primeira tarefa, de analisar a arquitetura do *website*, e compreendê-la, as tarefas posteriores tornaram-se menos complicadas.

⁵ www.sensingfuture.pt

8 Conclusão e Trabalho Futuro

Atualmente, os pacientes amputados de membros inferiores estão sujeitos a um processo demorado, pouco económico que por vezes, podem criar consequências psicologicamente negativas para o paciente. Produzir um *socket* e garantir que o mesmo seja confortável e adequado ao paciente é um dos grandes problemas inerentes a este processo, frequentemente são desconfortáveis podendo causar ferimentos no coto.

Perante a situação apresentada, o projeto desenvolvido pretende diminuir o ciclo de problemas associado à produção de *sockets*, tendo como objetivo fundamental o desenvolvimento de uma plataforma informática que tenha como principais características a aquisição e apresentação de parâmetros físicos do paciente, para posteriormente, se proceder à personalização do *socket* do membro inferior.

Numa fase inicial houve necessidade de perceber quais os métodos e técnicas atualmente utilizadas no desenvolvimento de próteses. Verificaram-se que, apesar dos avanços tecnológicos, hoje em dia o método tradicional é ainda o mais utilizado, apesar de apresentar várias desvantagens pois é pouco rigoroso e demorado.

Os pacientes começam a ser mais exigentes e tendem a procurar técnicas menos invasivas, com melhor qualidade e mais segurança. No mercado já existem várias tecnologias que contemplam estas vantagens, recorrendo por exemplo a Imagens 3D captadas com laser, CAD/CAM *software*, entre outros, no entanto nenhum dos métodos encontrados e na posição normal do dia-a-dia, tem em conta a análise da pressão exercida pelo paciente em todos os pontos da prótese. Assim surge o *Socket Master*, uma estrutura com a forma de um *socket* com uma rede de sensores integrada, para fazer a recolha dos pontos de pressão e temperatura.

No entanto, apesar de um projeto desta natureza ser muito interessante também levanta problemas de vária ordem. Destaca-se, a título de exemplo, a dificuldade de definição da base de dados. Esta, foi uma tarefa um pouco complexa uma vez que as reuniões presenciais com o consórcio foram escassas, teve-se então que recorrer a reuniões “*Skype*” com membros do consórcio, assim como troca de e-mails, este último não tão ágil devido ao tempo de espera por vezes causado. Outro ponto que dificultou esta tarefa foi a necessidade de trocar informações com várias empresas distintas, como por exemplo, contactar a empresa responsável pelo desenvolvimento dos sensores, e perceber que dados seriam guardados na base de dados, contactar a clínica para identificar quais as medidas relevantes no processo de desenvolvimento de próteses, entre outros.

A visualização e edição de modelos tridimensionais foi um processo bastante desafiador, sendo necessário fazer uma pesquisa exaustiva de soluções existentes, e sobre a forma de integrá-las neste projeto. Através do *Matlab*, foi possível visualizar o modelo teórico do coto do paciente, no entanto para o projeto em questão pretende-se obter o modelo real. Este modelo tem de ser baseado nas coordenadas dos 36 sensores integrados no *socket master*. Partindo das suas coordenadas e recorrendo a regressões lineares será possível embora com

algum erro devido ao número limitado de sensores, alcançar a representação real do coto do paciente.

Embora a plataforma desenvolvida já permita carregar modelos tridimensionais para visualização, como este projeto ainda se encontra em desenvolvimento, existem algumas funcionalidades que serão futuramente adicionadas ao projeto. Destaque-se, a integração de uma biblioteca para edição de modelos tridimensionais, a integração do *Socket Master* com a plataforma desenvolvida ou o teste de todo o sistema junto dos pacientes pilotos da clínica.

De salientar que algumas destas tarefas foram impossíveis de concretizar, pois encontram-se dependentes da conclusão de tarefas por parte do resto do consórcio. Assim que a estrutura física estiver concluída e a rede de sensores integrada, será possível implementar a aquisição de dados por parte da plataforma, assim como efetuados os testes de integração já planeados.

Em suma, e tendo em conta o desenvolvimento da plataforma informática, considere-se que os objetivos do estágio foram atingidos nos tempos previstos, no entanto ainda existirão certamente algumas funcionalidades a melhorar em função dos testes e integrações dos diversos módulos a realizar. Estas tarefas irão prosseguir o seu desenvolvimento, uma vez que o projeto só termina em Janeiro de 2018.

Referências Bibliográficas

- Adriel, W. (12 de 07 de 2015). *Introdução ao Laravel Framework PHP*. Obtido de DEVMEDIA: <http://www.devmedia.com.br/introducao-ao-laravel-framework-php/33173>
- Amarante, R. (18 de 02 de 2015). *Iniciando com Three.js - 3D nos navegadores*. Obtido de TABLELESS: <http://tableless.com.br/iniciando-com-three-js-3d-nos-navegadores/>
- Amputee Coalition. (07 de Dezembro de 2014). *Prosthetic Design and Fitting*. Obtido de Military in-step: <http://www.amputee-coalition.org/military-instep/prosthetic-design-fitting.html>
- Aparecido, J. (2015). *Teste de integração na prática*. Obtido de DEVMEDIA: <http://www.devmedia.com.br/teste-de-integracao-na-pratica/31877>
- Araújo, D. P. (20 de 02 de 2010). *Guia Prático - Amputação Membro Superior / Inferior*. Obtido de Pro Reabilitação: <http://www.proreabilitacao.com.br/includes/paginas/papo-e-cafezinho/demetrio-praxedes-araujo/pdf/proreabilitacao-guia-pratico-amputacao-parte-2.pdf>
- ASP.NET. (31 de 12 de 2015). *ASP.NET*. Obtido de ASP.NET: <http://www.asp.net/>
- Belem, T. (02 de 05 de 2010). *O que é e como funciona o JQuery*. Obtido de ThiagoBelem: <http://blog.thiagobelem.net/o-que-e-e-como-funciona-o-jquery>
- Brendler, C. F., Muller, M. S., Silva, A. R., & Teixeira, F. G. (2 de 10 de 2014). *Academia*. Obtido de Digitalização 3D utilizando Kinect e sistemas CAD e CAM para confecção de órtese de membro inferior: http://www.academia.edu/14261167/Digitaliza%C3%A7%C3%A3o_3D_utilizando_Kinect_e_sistemas_CAD_e_CAM_para_confec%C3%A7%C3%A3o_de_%C3%B3rtese_de_membro_inferior
- Caldwell, R., & Sanders, J. (01 de Fevereiro de 2013). *Socket Technology and Design: Future of an Evolving Practice*. Obtido de HEALIO: <http://www.healio.com/orthotics-prosthetics/prosthetics/news/print/o-and-p-news/%7B9954eb7e-5231-400c-a927-ff8fbd0c8532%7D/socket-technology-and-design-future-of-an-evolving-practice>
- Costa, G. (29 de 04 de 2014). *Conhecendo o Bootstrap*. Obtido de webdesign: <http://www.tutorialwebdesign.com.br/o-que-e-bootstrap/>
- Curralo, A., Queijo, L., & Rocha, J. (22-27 de Julho de 2012). Manufacturing of lower-limb custom fit prosthetics socket usin reverse engineering. 2. Porto, Porto, Portugal.
- Emmanuele. (18 de 01 de 2008). *JQuery: o que é e como usar?* Obtido de Código Fonte: https://www.codigofonte.net/dicas/javascript/310_jquery-o-que-e-e-como-usar
- Fourroux Prosthetics. (01 de Janeiro de 2014). *Prosthetic Socket Design*. Obtido de Fourroux Prosthetics: <http://www.fourrouxprosthetics.com/our-approach/prosthetic-socket-design.html>
- Kira. (09 de 09 de 2015). *3ders*. Obtido de MIT lab's FitSocket measures human limb tissue to 3D print better-fitting prosthetics: <http://www.3ders.org/articles/20150909-mit-labs-fitsocket-measures-human-limb-tissue-to-3d-print-better-fitting-prosthetics.html>

- Leal, N. E. (2011). *Desenvolvimento do processo de fabrico de próteses humanas em silicone para substituição de órgãos em tecidos moles*. Porto: FEUP.
- Limites. (27 de 01 de 2011). *Principais causas da Amputação*. Obtido de Limites do Homem: <http://limitesdohomem.blogs.sapo.pt/8389.html>
- Mariotti, F. S. (2014). *Como documentar a Arquitetura*. Obtido de Linha de Código: <http://www.linhadecodigo.com.br/artigo/3343/como-documentar-a-arquitetura-de-software.aspx>
- Marques, S. (2013). *Fabrico, alinhamento e avaliação de marcha com próteses e ortóteses de membros inferiores*. Porto: FEUP.
- Materialise. (01 de Abril de 2014). *A step Forward in Below Knee Prosthesis Socket Design*. Obtido de Software & Services for Biomedical Engineering: <http://biomedical.materialise.com/cases/step-forward-below-knee-prosthesis-socket-design>
- Medeiros, H. (12 de 05 de 2014). *Ajax básico: Introdução*. Obtido de Linha de Código: <http://www.linhadecodigo.com.br/artigo/3585/ajax-basico-introducao.aspx>
- Meyer, J. F. (2012). *Superfícies Funcionais Aplicadas a Biomateriais Inteligentes*. Coimbra: UC.
- Nesi, C. (2005). *Principais causas e níveis das amputações de membros inferiores nos pacientes do instituto de reabilitação do Hospital Erasto Gaertner*. Obtido de TC Online: <http://tcconline.utp.br/wp-content/uploads//2012/09/PRINCIPAIS-CAUSAS-E-NIVEIS-DAS-AMPUTACOES-DE-MEMBROS-INFERIORES-NOS-PACIENTES-DO-INSTITUTO-DE-REABILITACAO-DO-HOSPITAL-ERASTO-GAERTNER.pdf>
- Nielsen, J. (1992). Finding usability problems through heuristic evaluation. *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 373-380). California: Proceeding of CHI.
- ORACLE. (10 de 01 de 2014). *ORACLE*. Obtido de Java EE: <http://www.oracle.com/technetwork/java/javaee/overview/index.html>
- PHP. (10 de 01 de 2016). *PHP*. Obtido de PHP: <http://secure.php.net/>
- Pimentel, M. (2014). *Implementando testes unitários*. Obtido de DEVMEDIA: <http://www.devmedia.com.br/junit-implementando-testes-unitarios-em-java-parte-i/1432>
- Praciano, E. (26 de 02 de 2014). *Os benefícios e as vantagens do PHP*. Obtido de Elias Praciano: <http://elias.praciano.com/2014/02/15-beneficios-e-vantagens-do-php/>
- Queiroz, W. F. (2008). *Desenvolvimento de métodos construtivos e de novos materiais em pregados na confecção de cartuchos de próteses de membros inferiores*. Natal: Universidade do Rio Grande.
- Ricardo, F. d. (2012). *Engenharia de Requisitos*. Obtido de Universidade Federal do Espírito Santo.
- Sengeh, D. M. (2013). A Variable-Impedance Prosthetic Socket for a Transtibial Amputee Designed from Magnetic Resonance Imaging Data. *Journal of Prosthetics and Orthotics*, 129-137.
- Sensing Future Technologies. (01 de 08 de 2016). *Sensing Future Technologies*. Obtido de Sensing Future Technologies: www.sensingfuture.pt
- Shopping Ortopédico. (2012). *Centro de Reabilitação*. Obtido de Shopping Ortopédico: <http://shoppingortopedico.com.br/area-do-amputado/centro-de-reabilitacao>

- Tekscan. (2012). *F-Socket System*. Obtido de TEKSCAN: <https://www.tekscan.com/products-solutions/systems/f-socket-system?tab=description>
- Testes de Software. (29 de 06 de 2016). *Teste de integração*. Obtido de Testes de Software: <http://testesdesoftware.com/teste-de-integracao/>
- XSENSOR. (2012). *WheelChair Seating*. Obtido de XSENSOR Techonology Corporation: <http://www.xsensor.com/ForeSiteSS>

PROPOSTA DE ESTÁGIO

Ano Lectivo de 2015/2016

em Mestrado em Informática e Sistemas (Desenvolvimento de Software)

TEMA

Desenvolvimento de plataforma informática para colocação de próteses em amputados de membro inferior

SUMÁRIO

O desenvolvimento de uma prótese e as afinações posteriores para proporcionar um melhor conforto ao utente, é na maioria das vezes um processo moroso e dispendioso, quer para o utente quer para a entidade que desenvolve a mesma.

Desta forma, a empresa Sensing Future Technologies integra um projeto que tem como objetivos integrar sistemas elétricos e mecânicos com vários sensores de pressão, temperatura, humidade e fricção, para tornar o processo de desenhar e fabricar uma prótese mais confortável para os membros inferiores que permita que o processo seja menos moroso e dispendioso.

O projeto proposto ao aluno de Mestrado em Informática e Sistemas, ramo de desenvolvimento de *software*, consiste em desenvolver uma aplicação informática que receba os dados captados por um módulo eletrónico e faça uma análise dos mesmos e apresente de uma forma intuitiva os dados que estão a ser recolhidos para ajudar o profissional de saúde na tarefa de adaptação da prótese ao utente.

1. Âmbito

A Sensing Future concebe, desenvolve e implementa dispositivos médicos tecnológicos para terapia e reabilitação física. A Sensing Future apresenta *Know-how* no domínio da tecnologia, muito em concreto: Engenharia Mecânica, Engenharia Informática e Engenharia Eletrotécnica. A estes conhecimentos alia uma vasta rede sólida de contactos com profissionais de saúde.

O estágio proposto pretende implementar um *software* capaz de analisar e mostrar em tempo real que terá o objetivo de auxiliar o profissional de saúde no processo de colocação da prótese no utente.

Este estágio está inserido num projeto internacional de IDI, integrado no Horizon 2020 (Tópico: ICT-02-2014 - Smart System Integration) e que tem a Sensing Future Technologies

como membro do consórcio, fazendo parte da estratégia da empresa como afirmação na área de desenvolvimento de tecnologia para a área da saúde.

2. Objetivos

O presente projeto pretende atingir o seguinte objetivo:

Desenvolvimento de plataforma informática para aquisição e apresentação de parâmetros físicos para personalização de próteses de membro inferior.

E os seguintes objetivos específicos:

1. Implementação de um algoritmo para processamento dos dados;
2. Implementação de um *software* para mostrar os dados recolhidos em tempo real.
3. Desenvolvimento de interface para apresentação de formato 3D para construção de prótese otimizada para cada paciente.
4. Integração de *software e hardware* e apresentação do sistema final.

3. Programa de trabalhos

O estágio consistirá nas seguintes atividades e respetivas tarefas:

A1 – Análise ao estado da arte.

T1 – Pesquisa do processo de construção e colocação de prótese

T2 – Pesquisa sobre as meias de prótese atuais

T3 – Pesquisa de sensores e instrumentos de medida usados na colocação de prótese

T4 – Pesquisa de projetos e produtos que utilizem instrumentação na colocação ou uso de próteses

T5 – Pesquisa e estudo de *software* de instrumentação em próteses de membro inferior.

A2 – Problema e levantamento de requisitos

T1 – Levantamento dos requisitos para a solução a desenvolver

A3 – Arquitetura do sistema: estudo e abordagem à tecnologia a utilizar

T1 – Pesquisa e estudo de ferramentas a utilizar e identificação de linguagem de programação de acordo com os requisitos identificados

T2 – Apresentação da arquitetura global do sistema

A4 – Desenvolvimento e implementação da solução

T1 – Desenvolvimento de algoritmo

T2 – Desenvolvimento de *software* e interfaces

T3 – Integração do *software* com o módulo de hardware

A5 – Teste do sistema

T1 – Definição de testes a realizar

T2 - Teste e verificação do sistema

4. Calendarização das tarefas

As Tarefas acima descritas, incluindo os testes de validação de cada módulo, serão executadas de acordo com a seguinte calendarização:

O plano de escalonamento dos trabalhos é apresentado em seguida:

		2015			2016						
		10	11	12	1	2	3	4	5	6	7
A1	Análise ao estado da arte.										
T1	Pesquisa do processo de construção e colocação de prótese										
T2	Pesquisa sobre as meias de prótese atuais										
T3	Pesquisa de sensores e instrumentos de medida usados na colocação de prótese										
T4	Pesquisa de projetos e produtos com utilizem instrumentação na colocação ou uso de próteses										
T5	Pesquisa e estudo de software de instrumentação em próteses de membro inferior.										
A2	Problema e levantamento de requisitos										
T1	Levantamento dos requisitos para a solução a desenvolver										
A3	Arquitetura do sistema: estudo e abordagem à tecnologia a utilizar										
T1	Pesquisa e estudo de ferramentas a utilizar e identificação de linguagem de programação de acordo com os requisitos identificados										
T2	Apresentação da arquitetura global do sistema										
A4	Desenvolvimento e implementação da solução										
T1	Desenvolvimento de algoritmo										
T2	Desenvolvimento de software e interfaces										
T3	Integração do software com o módulo de hardware										
A5	Teste do sistema										
T1	Definição de testes a realizar										
T2	Teste e verificação do sistema										

5. Resultados

Os resultados do estágio serão consubstanciados num conjunto de documentos a elaborar pelo estagiário de acordo com o seguinte plano:

M1 Estudo do estado da arte

R1.1: Relatório do estudo da arte

M2: Requisitos do sistema

R2.1: Listagem de todos os requisitos do sistema

M3: Arquitetura do sistema

R3.1: Apresentação de arquitetura do sistema e ferramentas a utilizar no desenvolvimento

M4: Desenvolvimento e implementação da solução

R4.1: Apresentação da implementação do algoritmo

R4.2: Apresentação das interfaces em funcionamento com o algoritmo

R4.3: Apresentação do sistema final: software integrado com hardware

M5: Teste de sistema

R5.1: Listagem de testes realizados e resultados de cada teste

M6: Sistema final

R6.1: Apresentação do sistema final em funcionamento

R6.2: Entrega de relatório de estágio e apresentação de documentação

6. Local de Trabalho

Sede da Sensing Future: Rua Pedro Nunes (IPN – Incubadora), 3030-199 Coimbra.

7. Metodologia

A metodologia de trabalho será organizada de forma temporal através do programa de trabalho definido para este projeto. Para a realização de cada uma das atividades propostas será utilizada a metodologia SCRUM, implementada na entidade.

Organização de um Dossier de Projeto e reuniões.

8. Orientação

ISEC:

Nome: Anabela Gomes (anabela@isec.pt)

Categoria: Professor Adjunto

Entidade de Acolhimento:

Entidade: Sensing Future Technologies

Nome: Luís Ferreira (luisferreira@sensingfuture.pt)

Cargo: Responsável Técnico

9. Caracterização e Remuneração

- Data de início: 26 de Outubro de 2015
- Data de fim: 1 de Julho de 2016
- Horário: 9h30 às 18h30
- Tipo de regalias oferecidas: Estágio em regime de contratado
- Tipo de formação oferecida aos estagiários: Formação interna
- Outras condições

Apêndice B

SCRUM

SCRUM

O *Scrum* é uma metodologia ágil para desenvolvimento de projetos. Esta tem como objetivos organizar e gerir trabalhos complexos, muito utilizado em projetos de software.

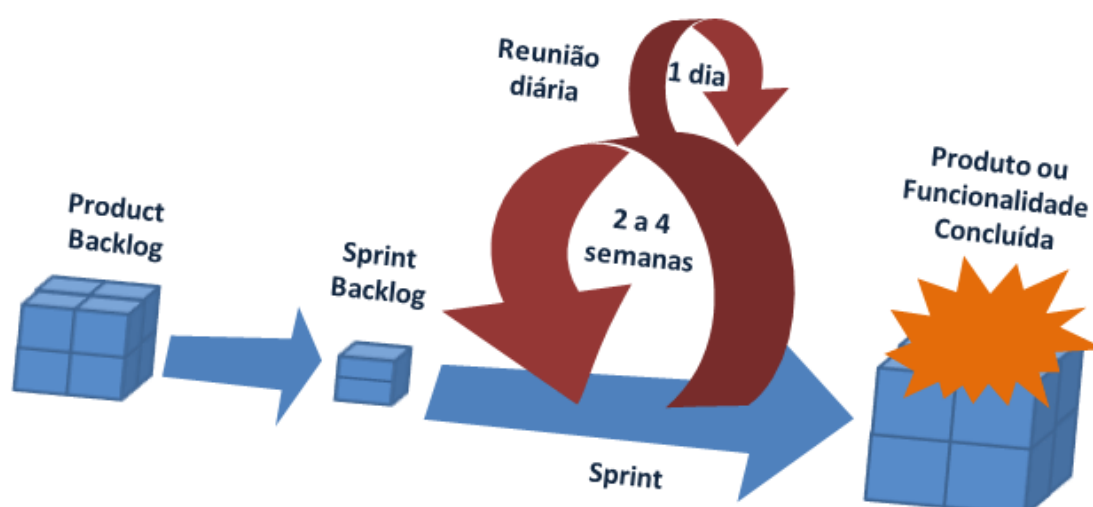
As equipas de desenvolvimento baseadas em *Scrum* são compostas por pessoas enquadradas em três tipos de papéis diferentes, entre eles o *product owner*, *scrumMaster* e equipa de desenvolvimento.

O *product owner* é um membro com poderes de liderança sobre o produto a desenvolver. Ele é responsável por definir e decidir quais os recursos e funcionalidades vão ser implementadas e qual a sua ordem. Ele é responsável pelo sucesso global da solução. O *product owner* deve estar a trabalhar diretamente com o *scrumMaster* e com a equipa de desenvolvimento, para que quando surjam dúvidas, este as possa esclarecer prontamente.

O *scrumMaster* é responsável por ajudar todos os intervenientes a entender e realizar corretamente os princípios e as práticas do *Scrum*. Este deve ajudar a equipa a resolver problemas, bem como ajudar a fazer melhorias na utilização desta metodologia. Este tem como funções perceber o que está a correr bem ou mal durante o desenvolvimento, e encaminhar a equipa de desenvolvimento ao rumo certo.

A equipa de desenvolvimento, é constituída por membros com várias valências, sendo responsáveis por desenvolver o produto bem como testá-lo. A ideia principal é que a equipa se auto-organize de forma a determinar a melhor maneira de realizar o trabalho por forma a atingir a meta estabelecida pelo *product owner*.

Na figura seguinte, representa-se as interações entre as várias atividades no processo.



O *product backlog* é uma lista única composta por todas as funcionalidade que o *product owner* pretende ver implementadas no produto final. A sprint tem uma duração média entre

duas a quatro semanas de trabalho, inicia-se com uma reunião designada de planeamento de sprint, onde são escolhidas de entre o *product backlog* quais as funcionalidade que devem ser desenvolvidas durante a sprint que se inicia. Ao longo da sprint são feitas reuniões diárias, com o objetivo de acompanhar o trabalho de cada um dos membro da equipa, esta reunião é designada de *daily scrum*. No final da *sprint* existe a *sprint review* e a *sprint retrospectiva*, a *sprint review* tem como objetivo verificar e adaptar o produto que está a ser desenvolvido. A *sprint retrospectiva* tem como objetivo verificar as necessidades de adaptação no processo de trabalho. Esta reunião é feita após a *sprint review* e antes da reunião de planeamento da próxima *sprint*.

Apêndice C

Definição dos Casos de Uso

UC	01
Nome	Adicionar novo paciente
Descrição	Permite ao utilizador inserir novos pacientes no <i>software</i> .
Ator	Utilizador genérico
Pré-condição	
Fluxo principal	1. Clicar no botão para inserir novo paciente; 2. Introduzir dados relativos à informação geral do paciente; 3. Introduzir dados relativos às medidas básicas do coto do paciente; 4. Introduzir dados relativos às propriedades mecânicas do paciente.
Fluxo alternativo	
Pós-condição	
Fonte	Equipa de Requisitos

UC	02
Nome	Remover paciente
Descrição	Permite ao utilizador remover pacientes registados no <i>Software</i> .
Ator	Utilizador genérico
Pré-condição	Paciente a remover tem de estar registado no <i>Software</i> .
Fluxo principal	1. Clicar no botão para remover paciente; 2. Certificar que deseja mesmo remover o paciente; 3. Paciente removido do <i>software</i>.
Fluxo alternativo	(2) Não valida a ação de remover o paciente a. Se o utilizador responder não à pergunta “Tem a certeza que deseja remover o paciente?”, será apresentado a página com os dados do paciente.
Pós-condição	
Fonte	Equipa de Requisitos

UC	03
Nome	Alterar paciente
Descrição	Permite ao utilizador alterar dados de um paciente registado no <i>Software</i> .
Ator	Utilizador genérico
Pré-condição	O paciente está selecionado.
Fluxo principal	1. Clicar no botão para editar dados do paciente
Fluxo alternativo	(1) Podem ser alterados três tipos de dados: a. Alterados dados relativos a informação geral do paciente; b. Alterar dados relativos às medidas básicas do coto do paciente; c. Alterar dados relativos às propriedades mecânicas do coto do paciente.
Pós-condição	
Fonte	Equipa de Requisitos

UC	04
Nome	Visualizar lista de pacientes
Descrição	Permite ao utilizador visualizar a lista de pacientes existentes no <i>Software</i>
Ator	Utilizador genérico
Pré-condição	
Fluxo principal	1. Clicar no botão para visualizar a lista de pacientes. 2. Visualiza a lista de todos os pacientes registados no <i>Software</i> .
Fluxo alternativo	(2) Não existem pacientes registados a. Se o <i>software</i> não tiver pacientes registados, deverá ser apresentada uma mensagem informativa, dizendo que ainda não existem pacientes registados.
Pós-condição	
Fonte	Equipa de Requisitos

UC	05
Nome	Pesquisar paciente por nome, endereço de email, número de saúde, ...
Descrição	Este requisito permite ao utilizador pesquisar um determinado paciente registado no sistema, por campos como o seu nome, endereço de email, número de saúde, entre outros campos.
Ator	Utilizador genérico
Pré-condição	O paciente a pesquisar tem de estar registado no <i>Software</i>
Fluxo principal	1. Clicar na caixa de texto para introduzir dados a pesquisar, como nome, e-mail, nif, número de telefone, entre outros campos. 2. O <i>Software</i> deverá filtrar os pacientes tendo em conta a pesquisa do utilizador, e assim apenas mostrar na tabela os pacientes desejados.
Fluxo alternativo	(1) Nenhum paciente possui o nome inserido na caixa de pesquisa a. Deverá ser apresentada uma mensagem a informar o utilizador, que não existe nenhum registo coincidente com a pesquisa feita.
Pós-condição	
Fonte	Equipa de Requisitos

UC	06
Nome	Visualizar histórico de consultas de um paciente
Descrição	Permite ao utilizador visualizar todas as consultas de determinado paciente
Ator	
Pré-condição	O paciente está selecionado
Fluxo principal	1. Clicar no botão para visualizar consultas do paciente; 2. Apresentada lista com todas as consultas de determinado paciente.
Fluxo alternativo	(1) O paciente pode ainda não ter consultas registadas. a. Deverá ser apresentada uma mensagem informativa de que ainda não existem consultas para determinado paciente.
Pós-condição	
Fonte	Equipa de Requisitos

UC	07
Nome	Remover aquisição de dados
Descrição	Permite ao utilizador remover uma aquisição de dados de um determinado paciente.
Ator	Utilizador genérico
Pré-condição	A aquisição de dados a remover deve existir no <i>Software</i> .
Fluxo principal	1. Clicar no botão para remover aquisição de dados; 2. Confirmar que pretende remover a aquisição de dados.
Fluxo alternativo	(1) Não valida a ação de remover a aquisição de dados a. Se o utilizador responder não à pergunta “Tem a certeza que deseja remover a aquisição de dados?”, será apresentado a página com os dados das aquisições de determinado paciente.
Pós-condição	
Fonte	Equipa de Requisitos

UC	08
Nome	Registrar descrição para aquisição de dados
Descrição	Permite ao utilizador registar uma descrição para a aquisição de dados.
Ator	Utilizador genérico
Pré-condição	Aquisição de dados concluída.
Fluxo principal	1. Preencher o campo relativo à descrição da aquisição; 2. Clicar no botão para guardar a descrição.
Fluxo alternativo	
Pós-condição	
Fonte	Equipa de Requisitos

UC	09
Nome	Alterar a descrição da aquisição de dados
Descrição	Permite ao utilizador alterar a descrição associada a uma aquisição de dados.
Ator	Utilizador genérico
Pré-condição	Aquisição de dados já registados no <i>Software</i> .
Fluxo principal	1. Clicar no botão para editar a descrição da aquisição; 2. Preencher o campo relativo à descrição da aquisição; 3. Clicar no botão para guardar a nova descrição.
Fluxo alternativo	
Pós-condição	
Fonte	Equipa de Requisitos

UC	10
Nome	Inserir notas associadas a pacientes
Descrição	Permite ao utilizador inserir notas associadas a pacientes registados no <i>Software</i> .
Ator	Utilizador genérico
Pré-condição	Paciente já registado no <i>Software</i> .
Fluxo principal	1. Clicar no botão para introduzir nova nota ao paciente; 2. Preencher o campo relativo à nota; 3. Clicar no botão para guardar a nova nota.
Fluxo alternativo	
Pós-condição	
Fonte	Equipa de Requisitos

UC	11
Nome	Remover notas associadas a pacientes
Descrição	Permite ao utilizador remover notas associadas a pacientes.
Ator	Utilizador genérico
Pré-condição	Nota já registada no <i>Software</i>
Fluxo principal	1. Clicar no botão para remover nota do paciente; 2. Confirmar que pretende remover a nota.
Fluxo alternativo	(1) Não valida a ação de remover a nota a. Se o utilizador responder não à pergunta “Tem a certeza que deseja remover a nota?”, será apresentado a página com os dados do paciente, assim como, as suas notas.
Pós-condição	
Fonte	Equipa de Requisitos

UC	12
Nome	Visualização sobre uma imagem tridimensional do <i>socket</i>
Descrição	Permite ao utilizador visualizar o <i>socket</i> sobre a forma de uma imagem tridimensional.
Ator	Utilizador genérico
Pré-condição	Aquisição de dados já registada no <i>Software</i>
Fluxo principal	1. Clicar no botão para visualizar uma aquisição de dados.
Fluxo alternativo	
Pós-condição	
Fonte	Equipa de Requisitos

Apêndice D

Descrição das tabelas da Base de Dados

Patient
Entidade com todos os pacientes registados no sistema. • id: campo responsável pela identificação da entidade; • name: campo responsável por guardar o nome do paciente; • dateOfBirth: campo responsável por guardar a data de nascimento do paciente; • gender: campo responsável por guardar o género do paciente; • weight: campo responsável por guardar o peso do paciente; • height: campo responsável por guardar a altura do paciente; • phoneNumber: campo responsável por guardar o número de telemóvel do paciente; • email: campo responsável por guardar o endereço eletrónico do paciente; • diabetic: campo responsável por guardar se o paciente é diabético; • dateOfRegister: campo responsável por guardar a data de registo do paciente; • identifierNumber: campo responsável por guardar o número de identificação do paciente; • removed: campo responsável por definir se determinado paciente se encontra removido.
Relacionamento com outras entidades: • Basic Measurements: Relaciona-se com esta tabela, por forma a guardar os valores das medidas básicas associadas ao paciente; • Mechanical Properties: Relaciona-se com esta tabela, por forma a guardar os valores das propriedades mecânicas associadas ao paciente; • Category Of Limb: Relaciona-se com esta tabela, por forma a guardar o tipo de amputação; • Amputation Side: Relaciona-se com esta tabela, por forma a guardar o lado da amputação; • Sigam Divergence: Relaciona-se com esta tabela, por forma a guardar o tipo de ajuda na mobilidade; • Sigam Mobility Scale: Relaciona-se com esta tabela, por forma a guardar o nível de mobilidade do paciente; • Activity Scale: Relaciona-se com esta tabela, por forma a guardar o nível de atividade do paciente; • Consults: Relaciona-se com esta tabela, por forma a guardar as consultas associadas ao paciente; • Socket Comfort Score: Relaciona-se com esta tabela, por forma a guardar o nível de conforto da prótese; • Notes: Relaciona-se com esta tabela, por forma a guardar as notas associadas ao paciente; • Cause Of Amputation: Relaciona-se com esta tabela, por forma a guardar a causa da amputação; • Nationalities: Relaciona-se com esta tabela, por forma a guardar a nacionalidade do paciente; • Addresses: Relaciona-se com esta tabela, por forma a guardar a morada do paciente.

Circumferences
Entidade que se relaciona com a “Basic Measurements”, e permite identificar o diâmetro do coto do paciente. • id: campo responsável pela identificação da entidade; • value: campo obrigatório que indica o valor do diâmetro do coto.

Basic Measurements
Entidade com todos os dados relativos a medições, como medidas do coto do paciente, assim, algumas medidas como a distância do coto ao joelho, ou à anca serão guardados nesta entidade. • id: campo responsável pela identificação da entidade; • flexionAngle: valor do ângulo formado entre o coto do paciente e o seu tronco; • adductionAngle: valor do ângulo máximo alcançado ao juntar as pernas; • abductionAngle: valor do ângulo mínimo alcançado ao afastar as pernas; • perDistResLimb: valor da distância da anca até à extremidade do coto; • perKneeCentre: valor da distância do coto até ao joelho; • perFloor: valor da distância do coto até ao chão; • mlFleshly: propriedade específica da prótese; • mlBony: propriedade específica da prótese. Relacionamento com outras entidades: • Circumferences: Relaciona-se com esta tabela, por forma a guardar o diâmetro do coto.

Country
Entidade com todos os países existentes no mundo. • id: campo responsável pela identificação da entidade; • name: campo obrigatório que indica o país do paciente.

Adress
Entidade com todos os dados referentes à morada dos pacientes. • id: campo responsável pela identificação da entidade; • adress1: campo responsável por guardar a primeira linha da morada; • adress2: campo opcional responsável por guardar a segunda linha da morada; • city: campo obrigatório que indica a cidade do paciente; • postalCode: campo obrigatório que indica o código postal do paciente. Relacionamento com outras entidades: • Country: Relaciona-se com esta tabela, por forma a guardar o país da morada.

Nationality
Entidade com todas às nacionalidades existentes no mundo. • id: campo responsável pela identificação da entidade; • nationality: campo obrigatório que indica a nacionalidade do paciente.

Cause of Amputation
Entidade com todas as possíveis causas de amputação. • id: campo responsável pela identificação da entidade; • name: campo responsável por guardar a causa da amputação; • description: campo obrigatório que indica a descrição da causa de amputação.

Socket Comfort Score
Entidade com uma escala de conforto, na qual o paciente deverá avaliar a sua prótese quanto ao nível de conforto. • id: campo responsável pela identificação da entidade; • valeuComfort: campo responsável por guardar o valor de conforto; • description: campo obrigatório que indica a descrição de conforto.

Mechanical Properties
Entidade com todos os dados relativos às propriedades mecânicas do coto do paciente. • id: campo responsável pela identificação da entidade; • force: campo responsável por guardar o valor da força exercida pelo Identer; • displacement: campo responsável por guardar o valor da deformação da pele; • maxPressure: campo responsável por guardar o valor máximo de pressão; • averagePressure: campo responsável por guardar o valor médio de pressão; • thickness: campo responsável por guardar o valor da espessura; • elasticModule: campo responsável por guardar o valor do módulo da elasticidade; • muscle: campo responsável por guardar o valor de músculo; • fat: campo responsável por guardar o valor da gordura; • contactArea: campo responsável por guardar o valor da área de contato; • diameter: campo responsável por guardar o valor do diâmetro do Identer.

Category of Limb
Entidade para identificar qual o tipo de amputação, como por exemplo, distinguir amputações transfemorais de amputações transtibiais. • id: campo responsável pela identificação da entidade; • name: campo responsável por guardar o tipo de amputação; • description: campo obrigatório que indica a descrição do tipo de amputação.

Sigam Mobile Scale
Entidade com a escala de mobilidade de determinado paciente, para saber por exemplo se o paciente não consegue andar ou se anda no máximo 2 horas, entre outras possibilidades. • id: campo responsável pela identificação da entidade; • code: campo responsável por guardar o valor da escala de mobilidade; • disability: campo responsável por guardar o grau de mobilidade; • definition: campo responsável por definir o grau de mobilidade.

Sigam Divergence
Esta entidade relaciona-se com a Sigam Mobility Scale e tem como função guardar dados específicos de determinadas opções da Sigam Mobility Scale, por exemplo, um paciente que ande com o auxílio de moletas, é necessário saber se necessita de uma, duas, ou até mesmo de um andador. • id: campo responsável pela identificação da entidade; • key: campo responsável por guardar o valor da ajuda na mobilidade; • description: campo responsável por guardar a descrição da ajuda na mobilidade.

Activity Scale
Entidade com uma escala de valores de um a dez, permitindo avaliar numa escala finita qual o nível de atividade do paciente. • id: campo responsável pela identificação da entidade; • value: campo responsável por guardar o nível de atividade; • description: campo responsável por guardar a descrição da ajuda na mobilidade.

Amputation Side
Entidade com três valores possíveis: esquerda, direita ou ambos, permite identificar qual o lado da amputação do paciente. • id: campo responsável pela identificação da entidade; • side: campo responsável por guardar o lado da amputação.

Notes
Entidade com todas as notas associadas aos pacientes, desta forma será possível registar notas para cada paciente. • id: campo responsável pela identificação da entidade; • date: campo responsável por guardar a data da nota; • note: campo responsável por guardar a nota.

Consult
Entidade com todos os dados das várias consultas feitas aos pacientes. • id: campo responsável pela identificação da entidade; • date: campo responsável por guardar a data da consulta; • note: campo responsável por guardar uma nota relativa à consulta; • removed: campo responsável por definir se determinada consulta se encontra removida.
Relacionamento com outras entidades: • Gait Heel Strick: Relaciona-se com esta tabela, por forma a guardar os valores dos sensores adquirido com o calcanhar no chão; • Gait Stand Phases: Relaciona-se com esta tabela, por forma a guardar os valores dos sensores adquirido com a totalidade do pé no chão; • Gait Toe Offs: Relaciona-se com esta tabela, por forma a guardar os valores dos sensores adquirido com a ponta do pé no chão.

Gait Heel Strick

Entidade com todos os movimentos quando o pé estiver levantado apenas com a calcanhar no chão.

- **id:** campo responsável pela identificação da entidade;
- **sensorControlStatus:** campo responsável por guardar o estado do sensor;
- **pressure1:** campo responsável por guardar o valor de pressão do sensor 1;
- **pressure2:** campo responsável por guardar o valor de pressão do sensor 2;
- **pressure3:** campo responsável por guardar o valor de pressão do sensor 3;
- **pressure4:** campo responsável por guardar o valor de pressão do sensor 4;
- **pressure5:** campo responsável por guardar o valor de pressão do sensor 5;
- **temperature:** campo responsável por guardar o valor da temperatura;
- **X:** campo responsável por guarda o valor da coordenada X do sistema eletrônico;
- **Y:** campo responsável por guardar o valro da coordenada Y do sistema eletrônico;
- **Z:** campo responsável por guardar o valor da coordenada Z do sistema eletrônico;
- **roll:** campo responsável por guardar a propriedade *roll* do sistema eletrônico;
- **yaw:** campo responsável por guardar a propriedade *yaw* do sistema eletrônico;
- **pitch:** campo responsável por guarda a propriedade *pitch* do sistema eletrônico.

Gait Stand Phases

Entidade com todos os movimentos quando o pé estiver completamente assente no chão.

- **id:** campo responsável pela identificação da entidade;
- **sensorControlStatus:** campo responsável por guardar o estado do sensor;
- **pressure1:** campo responsável por guardar o valor de pressão do sensor 1;
- **pressure2:** campo responsável por guardar o valor de pressão do sensor 2;
- **pressure3:** campo responsável por guardar o valor de pressão do sensor 3;
- **pressure4:** campo responsável por guardar o valor de pressão do sensor 4;
- **pressure5:** campo responsável por guardar o valor de pressão do sensor 5;
- **temperature:** campo responsável por guardar o valor da temperatura;
- **X:** campo responsável por guarda o valor da coordenada X do sistema eletrônico;
- **Y:** campo responsável por guardar o valro da coordenada Y do sistema eletrônico;
- **Z:** campo responsável por guardar o valor da coordenada Z do sistema eletrônico;
- **roll:** campo responsável por guardar a propriedade *roll* do sistema eletrônico;
- **yaw:** campo responsável por guardar a propriedade *yaw* do sistema eletrônico;
- **pitch:** campo responsável por guarda a propriedade *pitch* do sistema eletrônico.

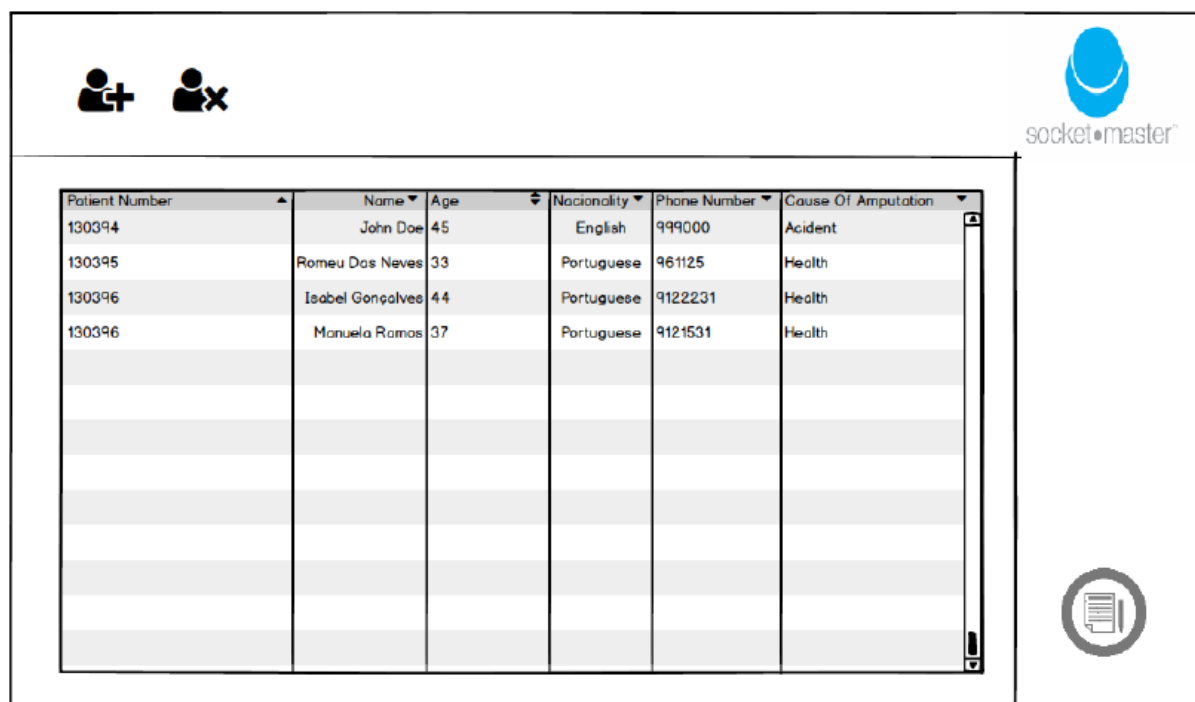
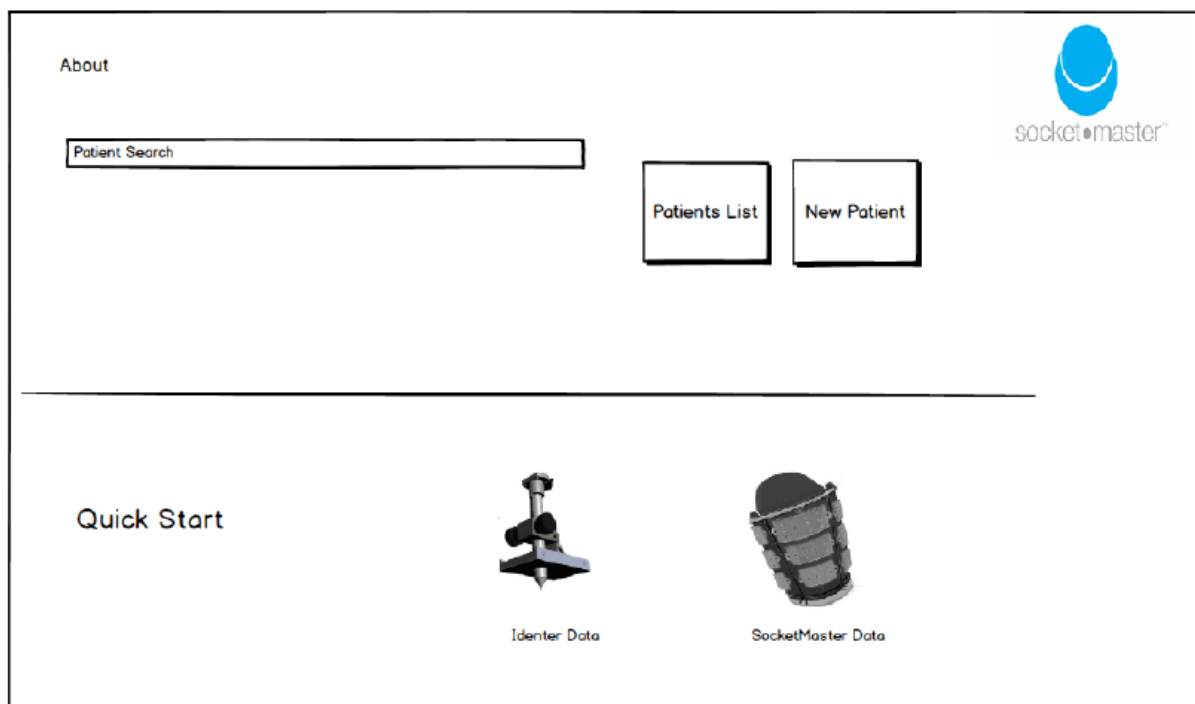
Gait Toe Offs

Entidade com todos os movimentos quando o pé estiver com o calcanhar levantado e a ponta no chão.

- **id:** campo responsável pela identificação da entidade;
- **sensorControlStatus:** campo responsável por guardar o estado do sensor;
- **pressure1:** campo responsável por guardar o valor de pressão do sensor 1;
- **pressure2:** campo responsável por guardar o valor de pressão do sensor 2;
- **pressure3:** campo responsável por guardar o valor de pressão do sensor 3;
- **pressure4:** campo responsável por guardar o valor de pressão do sensor 4;
- **pressure5:** campo responsável por guardar o valor de pressão do sensor 5;
- **temperature:** campo responsável por guardar o valor da temperatura;
- **X:** campo responsável por guardar o valor da coordenada X do sistema eletrônico;
- **Y:** campo responsável por guardar o valor da coordenada Y do sistema eletrônico;
- **Z:** campo responsável por guardar o valor da coordenada Z do sistema eletrônico;
- **roll:** campo responsável por guardar a propriedade *roll* do sistema eletrônico;
- **yaw:** campo responsável por guardar a propriedade *yaw* do sistema eletrônico;
- **pitch:** campo responsável por guardar a propriedade *pitch* do sistema eletrônico.

Apêndice E

Mockups do projeto





New Patient

General Information


socket•master

Patient ID:	130397	City:	Coimbra
Name:	Joana Santo	Phone Number:	+351910528500
Weight:	60.3 kg	Category Limb:	
Height:	175 m	Gender:	☐ Male ☒ Female
Nationality:	Portuguese	Birthday Date:	13/12/1980 
Address:	Rua General Humberto Delg	Cause Of Amputation:	Acident

Notes:

Some Notes





New Patient

Basic Measurements


socket•master

Diameter of Limb near of hip:	20.0 mm
Diameter of Limb in end of stump:	5.0 mm
Distance between hip and end of stump:	25.0 mm





New Patient

Mechanical Properties



Force: N

Max Pressure: Pa

Thickness: mm

Muscle: mm

Contact Area: m²

Displacement: mm

Average Pressure: Pa

Elastic Module: mm

Fat: mm

Diameter: mm







Patient Number	Name	Age	Nationality	Phone Number	Cause Of Amputation
130394	Robin	45	English	999000	Acident
130395	Romeu	33	Portuguese	961125	Health
130396	Isabel Gonçalves	44	Portuguese	9122231	Health
130396	Manuela Ramos	37	Portuguese	9121531	Health
130937	Joana Santo	36	Portuguese	+35191052.8500	Acident








socket•master

Patient Number	Name	Age	Nationality	Phone Number	Cause Of Amputation
130394	Robin	45	English	999000	Acident
130395	Romeu	33	Portuguese	961125	Health
130396	Isab				Health
130396	Ma				Health
130937				528500	Acident

Alert

Are you sure that you want to delete the patient selected?

Yes
No





John Doe - 130394



socket•master

General Information 

John Doe 45 Years

 178m

 88.7Kg

 London



 999000

Basic Measurements 

20cm

←→



22cm

←→

5cm

Sensors Data 

State Complete: ☐ Complete

First Consult: 01/02/2016

Last Consult: 08/02/2016

Total of 2 Consults

Mechanical Properties 

Force: 10N Displacement: 5mm

Max Pressure: 15N Fat: 8mm

Thickness: 10mm Muscle: 20mm





Edit - 130397

General Information


socket•master

Patient ID:	130394	City:	London
Name:	John Doe	Phone Number:	999000
Weight:	88.7 kg	Category Limb:	
Height:	178 m	Gender:	☒ Male ☐ Female
Nacionality:	English	Birthday Date:	13/12/1980 
Address:	Waterloo Road	Cause Of Amputation:	Acident

Notes:

Some Notes





Edit - 130397

Basic Measurements


socket•master

Diameter of Limb near of hip:	20.0 mm
Diameter of Limb in end of stump:	5.0 mm
Distance between hip and end of stump:	25.0 mm





Edit - 130397

Mechanical Properties



Force:	10	N	Displacement:		mm
Max Pressure:		Pa	Average Pressure:		Pa
Thickness:		mm	Elastic Module:		mm
Muscle:		mm	Fat:		mm
Contact Area:		m	Diameter:		mm



[illegible]

Sensors Data

John Doer - 130394

Consult ID	Date	Description
130	01/02/2016	First Consult
196	08/02/2016	Evolution Analysis

Alert

Are you sure that you want to delete the consult selected?

Yes

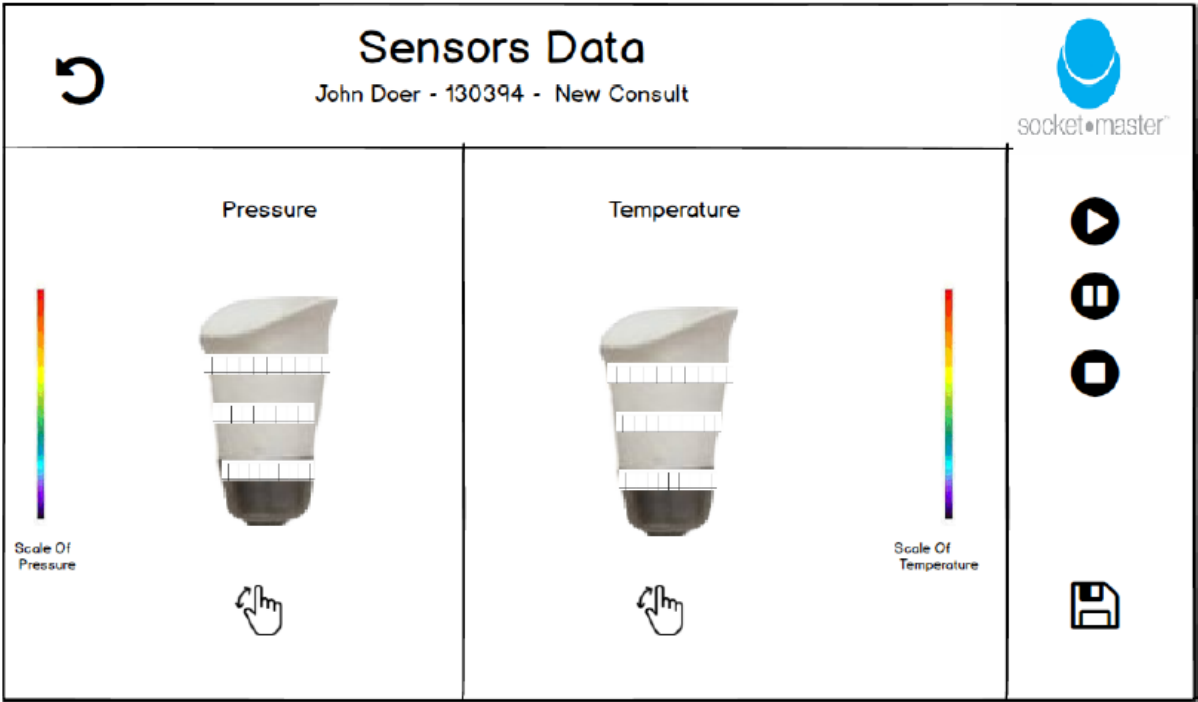
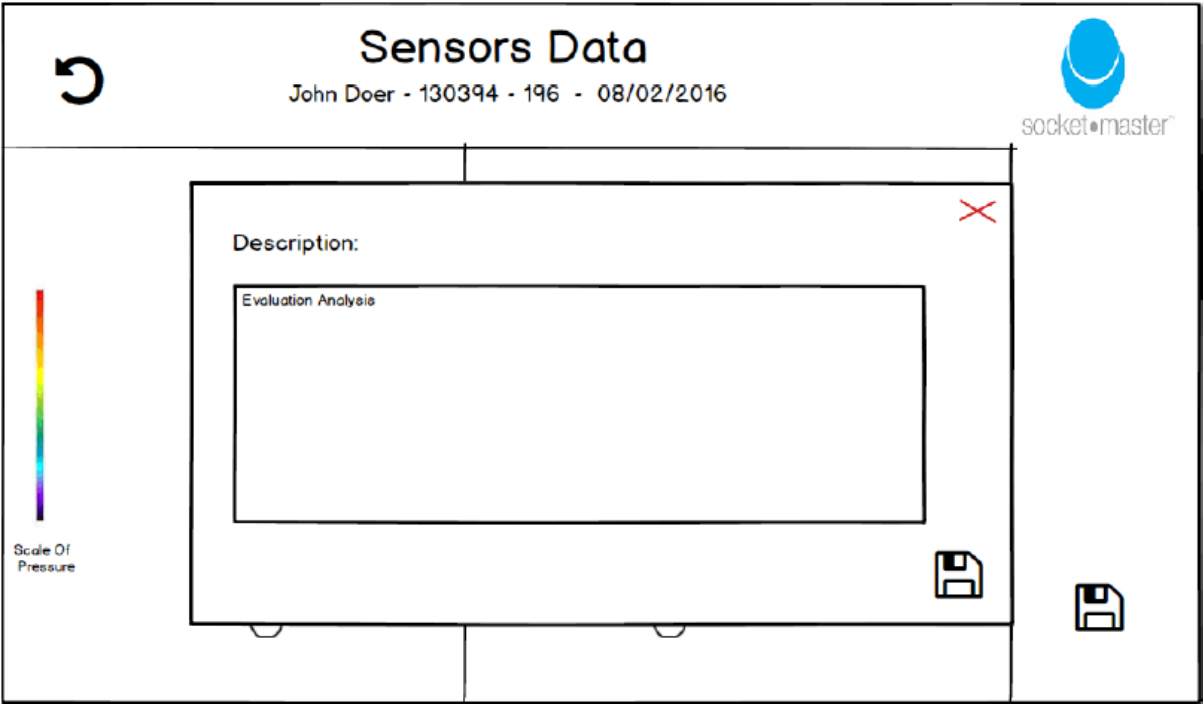
No

Sensors Data

John Doer - 130394 - 196 - 08/02/2016

Pressure

Temperature



The screenshot shows the Socket Master application window. The title bar displays 'Sensors Data' and 'John Doer - 130394 - New Consult'. The main window has a 'Description:' label and a text area containing 'The Patient is adapt well to prosthesis'. Below the text area is a 'Complete' checkbox and a save icon. A color scale for pressure is visible on the left side of the window.

[illegible]



John Doe - 130394


socket•master

General Information 

John Doe 45 Years

 1.78m 88.7Kg

 London   999000

Sensors Data 

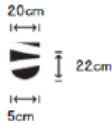
State Complete:  Complete

First Consult: 01/02/2016

Last Consult: 10/02/2016

Total of 3 Consults

Basic Measurements 



Mechanical Properties 



Apêndice F

Testes unitários

Testar – Registo de pacientes	
ID	CT-001
Caso de Teste	Registrar paciente no sistema.
Pré-Condição	
Procedimento	1. Fazer select à tabela “Patient” e guardar o número de pacientes registados no sistema; 2. Clicar em “New Patient”; 3. Introduzir os dados do paciente; 4. Clicar em adicionar paciente; 5. Voltar a fazer select à tabela “Patient” e guardar o número de pacientes.
Resultado Esperado	Validar que o número de pacientes no final é igual ao número de pacientes no início mais um.
Resultado	✓

Testar – Remover paciente	
ID	CT-002
Caso de Teste	Remover paciente no sistema.
Pré-Condição	Paciente registado no sistema.
Procedimento	1. Fazer select à tabela “Patient” e guardar o número de pacientes registados no sistema; 2. Selecionar o paciente a remover; 3. Clicar em “Remove”; 4. Voltar a fazer select à tabela “Patient” e guardar o número de pacientes.
Resultado Esperado	Validar que o número de pacientes no final é igual ao número de pacientes no início menos um.
Resultado	✓

Testar – Alterar dados do paciente	
ID	CT-003
Caso de Teste	Alterar dados do paciente no sistema.
Pré-Condição	Paciente registado no sistema.
Procedimento	1. Fazer select à tabela “Patient” com o id do paciente a alterar; 2. Fazer as devidas alterações ao paciente; 3. Voltar a fazer select à tabela “Patient” com o id do paciente alterado;
Resultado Esperado	Validar que os dados do paciente foram alterados na base de dados.
Resultado	✓

Testar – Listar pacientes registados	
ID	CT-004
Caso de Teste	Listar pacientes registados no sistema.
Pré-Condição	Paciente registado no sistema.
Procedimento	1. Fazer select à tabela “Patient” e guardar o número de pacientes registados no sistema. 2. Clicar em “Patient List”;
Resultado Esperado	Validar que o número de pacientes listados é igual ao número proveniente do select.
Resultado	✓

Testar – Pesquisa de pacientes	
ID	CT-005
Caso de Teste	Pesquisa de pacientes registados no sistema.
Pré-Condição	Paciente registado no sistema.
Procedimento	
Resultado Esperado	
Resultado	

Testar – Adicionar nova aquisição de dados	
ID	CT-006
Caso de Teste	Adicionar nova aquisição de dados no sistema.
Pré-Condição	Paciente registado no sistema.
Procedimento	1. Fazer select à tabela “Consult” e guardar o número de consultas registadas no sistema. 2. Clicar em “Add Consult”; 3. Iniciar aquisição de dados; 4. Voltar a fazer select à tabela “Consult” e guardar o número de consultas registadas no sistema.
Resultado Esperado	Validar que o número de consultas retornadas após a inserção é igual ao número anterior mais um.
Resultado	✓

Testar – Remover aquisição de dados	
ID	CT-007
Caso de Teste	Remover aquisição de dados no sistema.
Pré-Condição	Paciente registado no sistema; Aquisição de dados existente.
Procedimento	1. Fazer select à tabela “Consult” e guardar o número de consultas registadas no sistema. 2. Selecionar a consulta a remover; 3. Clicar em “Remove”; 4. Voltar a fazer select à tabela “Consult” e guardar o número de consultas registadas no sistema.
Resultado Esperado	Validar que o número de consultas após a remoção é igual à anterior menos um.
Resultado	✓

Testar – Registrar descrição na aquisição de dados	
ID	CT-008
Caso de Teste	Registrar descrição na aquisição de dados no sistema.
Pré-Condição	Paciente registado no sistema; Aquisição de dados existente.
Procedimento	1. Preencher a descrição da aquisição; 2. Clicar em “Save”; 3. Fazer select à tabela “Consult”.
Resultado Esperado	Validar que a descrição da consulta está preenchida.
Resultado	✓

Testar – Alterar descrição na aquisição de dados	
ID	CT-009
Caso de Teste	Alterar descrição na aquisição de dados no sistema.
Pré-Condição	Paciente registado no sistema; Aquisição de dados existente; Descrição na aquisição de dados existente.
Procedimento	1. Fazer select à tabela “Consult” com o id da consulta a alterar; 2. Fazer as devidas alterações à descrição; 3. Voltar a fazer select à tabela “Consult” com o id da consulta alterada.
Resultado Esperado	Validar que a consulta foi alterada na base de dados.
Resultado	✓

Testar – Inserir notas	
ID	CT-010
Caso de Teste	Inserir notas no sistema, associadas a um paciente
Pré-Condição	Paciente registado no sistema.
Procedimento	1. Fazer select à tabela “Notas” e guardar o número de notas registadas no sistema; 2. Preencher a nota; 3. Clicar em “Save”; 4. Voltar a fazer select à tabela “Notas”, e guardar o número de notas registadas no sistema.
Resultado Esperado	Validar que o número de notas após a inserção é igual ao anterior mais um.
Resultado	✓

Testar – Remover notas	
ID	CT-011
Caso de Teste	Remover notas do sistema, associadas a um paciente
Pré-Condição	Paciente registado no sistema.
Procedimento	1. Fazer select à tabela “Notas” e guardar o número de notas registadas no sistema. 2. Selecionar a nota a remover; 3. Clicar em “Remove”; 4. Voltar a fazer select à tabela “notas” e guardar o número de notas registadas no sistema.
Resultado Esperado	Validar que o número de notas após a remoção é igual à anterior menos um.
Resultado	✓

Testar – Visualização do <i>socket</i> numa imagem 3D	
ID	CT-012
Caso de Teste	Visualizar o <i>socket</i> sob forma de uma imagem tridimensional, no sistema.
Pré-Condição	Paciente registado no sistema.
Procedimento	
Resultado Esperado	
Resultado	

Apêndice G

Testes de Integração

Teste de Integração	
ID	CTI-001
Caso de Teste	Aplicar uma força num sensor de pressão e validar que é lida e inserida na BD corretamente.
Pré-Condição	Sensores ligados e conectados ao <i>software</i>
Procedimento	1. Aplicar uma força num sensor de pressão 2. Iniciar aquisição de dados no <i>software</i> 3. Fazer select à base de dados
Resultado Esperado	Validar que a força aplicada no sensor foi registada na BD
Resultado	

Teste de Integração	
ID	CTI-002
Caso de Teste	Colocar um coto artificial no <i>Socket Master</i> e validar que os valores de pressão adquiridos pelos vários sensores são adquiridos e registados na BD.
Pré-Condição	Sensores ligados e conectados ao <i>software</i>
Procedimento	1. Colocar um coto artificial no <i>Socket Master</i> 2. Iniciar aquisição de dados no <i>software</i> 3. Fazer select à base de dados
Resultado Esperado	Validar que a força aplicada pelo coto artificial foi registada na BD
Resultado	

Teste de Integração	
ID	CTI-003
Caso de Teste	Efetuar medidas de indentação num coto artificial e validar que os valores são registados na BD.
Pré-Condição	Indenter ligado e conectado ao <i>software</i>
Procedimento	1. Iniciar medidas de indentação num coto artificial 2. Iniciar aquisição de dados no <i>software</i> 3. Fazer select à base de dados
Resultado Esperado	Validar que as medições foram registadas no <i>software</i>
Resultado	

Teste de Integração	
ID	CTI-004
Caso de Teste	Iniciar uma aquisição de dados com o <i>Socket Master</i> , desligar a sua conexão e validar que o <i>software</i> avisa a perda de comunicação
Pré-Condição	Sensores ligados e conectados ao <i>software</i>
Procedimento	1. Iniciar aquisição de dados no <i>software</i> 2. Desligar a conexão entre o <i>socket master</i> e o <i>software</i>
Resultado Esperado	Validar que o <i>software</i> informa o utilizador da perda de conexão
Resultado	

Teste de Integração	
ID	CTI-005
Caso de Teste	Iniciar uma aquisição de dados com o <i>Identer</i> , desligar a sua conexão e validar que o <i>software</i> avisa a perda de comunicação.
Pré-Condição	Sensores ligados e conectados ao <i>software</i>
Procedimento	1. Iniciar aquisição de dados no <i>software</i> 2. Desligar a conexão entre o <i>Identer</i> e o <i>software</i>
Resultado Esperado	Validar que o <i>software</i> informa o utilizador da perda de conexão
Resultado	

Teste de Integração	
ID	CTI-006
Caso de Teste	Colocar um sensor numa determinada posição, verificar que os dados das suas coordenadas são registadas na BD corretamente.
Pré-Condição	Sensores ligados e conectados ao <i>software</i>
Procedimento	1. Colocar o <i>Socket Master</i> numa determinada posição inicial 2. Iniciar aquisição de dados no <i>software</i> 3. Fazer select à base de dados
Resultado Esperado	Validar que as coordenadas dos vários sensores são registadas na BD corretamente
Resultado	

Teste de Integração	
ID	CTI-007
Caso de Teste	Iniciar uma aquisição de dados sem nada estar em contato com o <i>Socket Master</i> , validar que são registados valores nulos de pressão na BD.
Pré-Condição	Sensores ligados e conectados ao <i>software</i>
Procedimento	1. Colocar o <i>socket master</i> sem nada em contacto 2. Iniciar aquisição de dados no <i>software</i> 3. Fazer select à base de dados
Resultado Esperado	Validar que a bd fica preenchida com dados a zero
Resultado	

Teste de Integração	
ID	CTI-008
Caso de Teste	Iniciar uma aquisição de dados do <i>Identer</i> sem nenhuma amostra, validar que são registados valores nulos na BD.
Pré-Condição	Sensores ligados e conectados ao <i>software</i>
Procedimento	1. Colocar o <i>Identer</i> sem nada em contacto 2. Iniciar aquisição de dados no <i>software</i> 3. Fazer select à base de dados
Resultado Esperado	Validar que a bd fica preenchida com dados a zero
Resultado	

Apêndice H

Testes de Usabilidade com Utilizadores

Tests with users

Considering Socket Master Software, to help us to evaluate their functions at the level of usability, performance and simplicity, we plan these list of tests to run with users.

1. Insert a patient;
2. See the list of patients;
3. See data from a specific patient
4. Update basic information of a patient;
5. Update general information of a patient;
6. Update mechanical properties of a patient;
7. Remove a patient;
8. Start a new acquisition to a patient;
9. See the Socket proposed by an acquisition;
10. Remove an acquisition;
11. Register a note to a patient;
12. Remove notes from patients;
13. See the list of notes from patients.

These list of tests should be done with a minimum of 5 users, and for each user should be filled the table below (Note: 1-> Very Bad Navigation / Very tired 10 -> Very good navigation / Not tired):

Test Number	Test Description	Number of:				In a scale between 1 and 10 classify:	
		Incorrect actions	Help requests	Repetition of errors	Activities finished	Navigation Facility	Tiredness
1	Insert a patient						
2	See the list of patients						
3	See data from a specific patient						
4	Update basic information of a patient						
5	Update general information of a patient						
6	Update mechanical properties of a patient						
7	Remove a patient						
8	Start a new acquisition to a patient						
9	See the Socket proposed by an acquisition						
10	Remove an acquisition						
11	Register a note to a patient						
12	Remove notes from patients						
13	See the list of notes from patients						

Apêndice I

Questionário

Quiz

Considering Socket Master Software, evaluate their functions at the level of usability and performance.

1. Answer the following questions with an X:
1) Very Bad; 2) Bad; 3) Reasonable; 4) Good; 5) Very Good

Questions	1	2	3	4	5
Intuitiveness					
Complexity					
Software response time / Agility					
Aesthetics					

2. In the home page, the image suggestive of the type of data acquisition are intuitive?
Yes ☐
No ☐

3. If you answered No, in the previous question, please give your suggestions.

--

4. Answer the following questions with an X:
1)Very Bad; 2) Bad; 3) Reasonable; 4) Good; 5) Very Good

Classify	1	2	3	4	5
The icons insert remove and update patient					
Navigation menus in the level of consistency					
Navigation and interaction with Software					

5. Register your suggestions / comments.

--