

isec

Engenharia

MESTRADO EM ENGENHARIA
ELETROTÉCNICA

DEFINITIVO

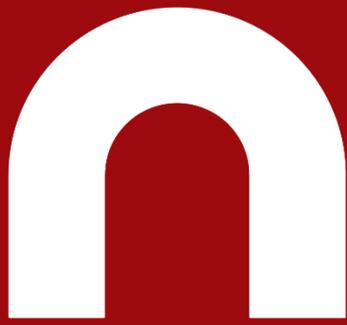
Desenvolvimento de uma aplicação para a reabilitação da mobilidade da mão humana com recurso a um *joystick* háptico

Autor

Belnick Monteiro da Silva

Orientador

João Paulo Morais Ferreira



isec

Engenharia

DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA

**Desenvolvimento de uma aplicação para a
reabilitação da mobilidade da mão humana com
recurso a um *joystick* háptico**

Relatório de Trabalho de Projeto para a obtenção do grau de
Mestre em Engenharia Eletrotécnica

Especialização em Automação e Comunicações em Sistemas
Industriais

Autor

Belnick Monteiro da Silva

Orientador

João Paulo Morais Ferreira

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, fevereiro de 2021

AGRADECIMENTOS

Agradeço ao meu orientador Doutor João Paulo Morais Ferreira pela disponibilidade, pelos conselhos e conhecimentos transmitidos que me ajudaram a trilhar o melhor caminho na concessão do projeto de tese.

Agradeço ao Sr. Eng. Paulo Mendes (ex-bolseiro da UC Pólo 2) pela experiência transmitida demonstrando ser de grande valia e muito esclarecedora na triagem do conhecimento.

Agradeço à minha família (meu pai, minha mãe, meus irmãos e os restantes familiares) pelo apoio emocional e monetário ao longo de todo percurso académico, que conduziu as realizações pessoais e profissionais. Sem eles nada disto seria possível, sendo que para um crescimento intelectual e psicológico, o suporte dos familiares é fulcral.

Agradeço aos meus amigos e colegas que sempre estiveram por perto para apoiar nos momentos difíceis.

Um obrigado a todos.

RESUMO

Dando seguimento à evolução do robô, este projeto almeja a utilização do *Geomagic Touch* no desenvolvimento de aplicações que promovam a mobilidade da mão Humana. Entre essas aplicações estão presentes a criação de exercícios físicos que consistem na reprodução da trajetória de um retângulo e um círculo, sendo que durante toda a reprodução da trajetória o utilizador é auxiliado pelo *joystick* háptico. Porém esses exercícios limitam-se apenas à interação háptica, que quando comparada à interação háptica-gráfica a imersão humana é inferior, por este motivo foi adicionado a interface gráfica. Para a criação da interface gráfica é utilizado a *Unity* e para programação da interface háptica o *Visual Studio*. O sistema criado consiste num ambiente gráfico criado especificamente para a escrita (em *Unity*), em que após ser feita a caligrafia pelo utilizador, esta é gravada. Uma vez gravada a imagem é importada para uma aplicação (*Visual Studio*) onde é processada por um algoritmo analítico que realiza operações tais como translação, rotação, acerto de escala e comparações entre as imagens do paciente antes e após a ocorrência (doença, lesão, etc...). No final dessa comparação são produzidos dados estatísticos que demonstram a evolução na mobilidade da mão humana em fase de reabilitação. Por fim os dados estatísticos são armazenados na base de dados em conjunto com os dados dos utilizadores.

O projeto tem a particularidade de focar-se na recuperação da mobilidade das mãos dos pacientes em fase de reabilitação, utilizando o dispositivo háptico denominado *Geomagic Touch* que se encontra no Instituto de Sistemas e Robótica de Coimbra.

Palavras-Chave: *Geomagic Touch*, háptico, *feedback*, reabilitação, mobilidade, interface gráfica, interface háptica, interface háptico-gráfico.

ABSTRACT

Following the evolution of the robot, this project aims to use Geomagic Touch in the development of applications that promote the mobility of the Human hand. Among these applications are the creation of physical exercises that consist of the reproduction of the trajectory of a rectangle and a circle, and during the entire reproduction of the trajectory the user is assisted by the haptic stick. However, these exercises are limited only to haptic interaction, which when compared to haptic-graphic interaction, human immersion is lower, for this reason the graphic interface was added. Unity is used to create the graphical interface and Visual Studio to program the haptic interface. The system created consists of a graphic environment created specifically for writing (in Unity), in which after the calligraphy is made by the user, it is recorded. Once the image is recorded, it is imported into an application (Visual Studio) where it is processed by an analytical algorithm that performs operations such as translation, rotation, scale adjustment and comparisons between the patient's images before and after the occurrence (disease, injury, etc...). At the end of this comparison, statistical data are produced that demonstrate the evolution in the mobility of the human hand in the rehabilitation phase. Finally, statistical data is stored in the database together with user data.

The project has the particularity of focusing on recovering patients hand mobility undergoing rehabilitation, using the haptic device called Geomagic Touch that is found at the Institute of Systems and Robotics, University of Coimbra.

Keywords: Geomagic Touch, haptic, feedback, rehabilitation, mobility, graphic interface, haptic interface, haptic-graphic interface

Índice

| | |
|---|----|
| CAPÍTULO 1 | 1 |
| Introdução | 1 |
| 1.1 - Enquadramento | 2 |
| 1.2 - Objetivos | 3 |
| 1.3 - Organização do projeto | 4 |
| CAPÍTULO 2 | 5 |
| Revisão da Literatura | 5 |
| 2.1 - Contextualização | 5 |
| 2.2 – Háptico | 6 |
| 2.3 – Dispositivo hápticos móveis | 8 |
| 2.4 – Dispositivos hápticos fixos | 9 |
| 2.4.1 Catalogação dos dispositivos hápticos fixos | 9 |
| CAPÍTULO 3 | 16 |
| Hardware e Software | 16 |
| 3.1 – <i>Geomagic Touch</i> | 16 |
| 3.2 - Computador | 20 |
| 3.3 - <i>Visual Studio</i> | 20 |
| 3.4 - <i>Unity</i> | 20 |
| CAPÍTULO 4 | 22 |
| Aplicações Desenvolvidas | 22 |
| 4.1 - Interface háptica | 22 |
| 4.1.1 - Controlo | 22 |
| 4.1.2 - Aplicações | 25 |
| 4.2 - Interface háptica – gráfica | 30 |
| 4.2.1 - Menu das opções | 30 |
| 4.2.2 - Menu do ambiente para escrita | 31 |
| 4.3 - Interface gráfica | 35 |
| 4.3.1 - SARA | 35 |
| 4.3.2 - Menu <i>login</i> | 36 |
| 4.3.3 - Menu cadastro | 37 |
| 4.3.4 - Menu análise | 38 |

| | |
|---|----|
| 4.3.4.1 - Menu dos registos..... | 38 |
| 4.3.4.2 - Menu para tratamento de imagens | 39 |
| 4.3.5 - Base de dados..... | 44 |
| CAPÍTULO 5 | 46 |
| Testes e resultados | 46 |
| 5.1 - Captação das assinaturas..... | 47 |
| CAPÍTULO 6 | 50 |
| Conclusões, e Sugestões de Melhoria | 50 |
| 6.1 - Conclusões | 50 |
| 6.2 - Sugestões de Melhoria | 51 |
| Bibliografia..... | 52 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1.1 - Interação homem-máquina | 1 |
| Figura 2.1 – <i>Unimate</i> (Wikipédia, Robô industrial, 2020). | 6 |
| Figura 2.2 – Comunicação háptica..... | 7 |
| Figura 2.3 - <i>CyberGlove II</i> (Systems C. , Cyberglove II, 2017)..... | 8 |
| Figura 2.4 – <i>CyberGrasp</i> (Systems C. , CyberGrasp, 2017). | 8 |
| Figura 2.5 - <i>TeslaSuit</i> (TeslaSuit, 2020). | 8 |
| Figura 2.6 - Volante háptico (PcComponentes, 2020)..... | 9 |
| Figura 2.7 - <i>Joystick</i> Háptico (Newegg, 2020). | 9 |
| Figura 2.8 – <i>Hapkit</i> (Univercity, 2019). | 11 |
| Figura 2.9 - <i>Omega 3</i> (FORCE DIMENSION OMEGA 3, 2020). | 11 |
| Figura 2.10 - <i>4-DOF-Haptic Device</i> (Payandeh, 2008)..... | 11 |
| Figura 2.11 - <i>Sigma 7</i> (Inition I. , 2020). | 11 |
| Figura 2.12 - <i>Geomagic Touch</i> (Systems 3. , 2020). | 12 |
| Figura 2.13 - Controlo do Robô UR3. (Soliwada, 2019). | 12 |
| Figura 2.14 - VR4Robot (B.V., 2020). | 12 |
| Figura 2.15 - Alfabeto Persa (Boroujen & Meghdari, 2009)..... | 12 |
| Figura 2.16 – <i>IDEA</i> (International, 2019). | 13 |
| Figura 2.17 - <i>Industrial Virtual Reality Immersive Touch</i> (Immersivetouch, 2019)..... | 13 |
| Figura 2.18 - Membros Superiores. | 14 |
| Figura 2.19 - Dados de ocorrência de AVC em Portugal. | 14 |
| Figura 3.1 – <i>Geomagic Touch</i> (Systems 3. , 2020). | 16 |
| Figura 3.2 - Posição inicial do <i>Geomagic Touch</i> | 18 |
| Figura 3.3 – Mesa. | 19 |
| Figura 3.4 - Acer Aspire..... | 20 |
| Figura 3.5 – <i>Visual Studio</i> logotipo. | 20 |
| Figura 3.6 – Unity Logotipo. | 20 |
| Figura 3.7 - <i>The Heretic</i> | 21 |
| Figura 4.1 - Controlo proporcional..... | 23 |
| Figura 4.2 - Controlo PID. | 24 |
| Figura 4.3 - Controlo P. | 25 |
| Figura 4.4 - Transição entre 4 vértices no retângulo. | 25 |
| Figura 4.5 - Transição no retângulo por pontos intermédios. | 26 |
| Figura 4.6 - Retângulo interpolado. | 27 |
| Figura 4.7- Sequência de deslocamento entre os pontos de referências..... | 27 |
| Figura 4.8 - Fluxograma do algoritmo. | 28 |
| Figura 4.9 Círculo interpolado. | 29 |
| Figura 4.10 - Menu opções..... | 30 |
| Figura 4.11 – Acessos proporcionados pelo menu. | 31 |
| Figura 4.12 - <i>Layout</i> para a escrita..... | 31 |

| | |
|---|----|
| Figura 4.13 - Fluxograma do funcionamento da escrita. | 32 |
| Figura 4.14 - Exemplo 1. | 33 |
| Figura 4.15 - Exemplo 2. | 33 |
| Figura 4.16 - Referência para escrita. | 33 |
| Figura 4.17 - Mão direita (A). | 34 |
| Figura 4.18 - Mão esquerda (A). | 34 |
| Figura 4.19 - Mão direita (B). | 34 |
| Figura 4.20 - Mão esquerda (B). | 34 |
| Figura 4.21 - 1º tarefa e o menu de teste. | 34 |
| Figura 4.22 - 2ª Tarefa. | 35 |
| Figura 4.23 - 3ª Tarefa. | 35 |
| Figura 4.24 - 4ª Tarefa. | 35 |
| Figura 4.25 - Resultado do treino. | 35 |
| Figura 4.26 – Esquema das camadas da SARA. | 36 |
| Figura 4.27 - Menu <i>Login</i> | 36 |
| Figura 4.28 – Cadastro. | 37 |
| Figura 4.29 - <i>Layout</i> inicial da SARA. | 38 |
| Figura 4.30 - Menu dos registos. | 39 |
| Figura 4.31 - Menu para tratamento das assinaturas. | 40 |
| Figura 4.32 - Assinatura inicial. | 42 |
| Figura 4.33 - Assinatura 0.5x. | 42 |
| Figura 4.34 - Assinatura 2x. | 42 |
| Figura 4.35 - Assinatura inicial. | 43 |
| Figura 4.36 - Assinatura centralizada e rodada. | 43 |
| Figura 4.37 - Assinatura após lesão. | 43 |
| Figura 4.38 - Conjugação com $\Theta = - 90^\circ$ | 44 |
| Figura 4.39- Conjugação com $\Theta = - 45^\circ$ | 44 |
| Figura 4.40 - Conjugação com $\Theta = 0^\circ$ | 44 |
| Figura 4.41 - Conjugação com $\Theta = 90^\circ$ | 44 |
| Figura 4.42 - Conjugação com $\Theta = 45^\circ$ | 44 |
| Figura 4.44 - Arquitetura ER1-N. | 45 |
| Figura 4.44 - Diagrama ER. | 45 |
| Figura 4.45 - Arquitetura E.R. | 45 |
| Figura 5.1 - Assinatura de referência (voluntário JL). | 47 |
| Figura 5.2 - 1º Ensaio do JL. | 47 |
| Figura 5.3 - 2º Ensaio do JL. | 47 |
| Figura 5.4 - 3º Ensaio do JL. | 47 |
| Figura 5.5 - 4º Ensaio do JL. | 47 |
| Figura 5.6- Assinatura de referência do voluntário BM. | 47 |
| Figura 5.7 - 1º Ensaio do BM. | 47 |
| Figura 5.8 - 2º Ensaio do BM. | 47 |
| Figura 5.9 - 3º Ensaio do BM. | 47 |
| Figura 5.10 - 4º Ensaio do BM. | 47 |

| | |
|--|----|
| Figura 5.11 - Assinatura de referência do voluntário MF..... | 48 |
| Figura 5.12 - 3º Ensaio MF..... | 48 |
| Figura 5.13 - 4º Ensaio MF..... | 48 |
| Figura 5.14 - 2º Ensaio do MF..... | 48 |
| Figura 5.15 - 1º Ensaio do MF..... | 48 |
| Figura 5.16 - Assinatura de referência do voluntário BMS. | 48 |
| Figura 5.17 - 1º Ensaio do BMS. | 48 |
| Figura 5.18 - 2º Ensaio do BMS. | 48 |
| Figura 5.19 - 3º Ensaio do BMS. | 48 |
| Figura 5.20 - 4º Ensaio do BMS. | 48 |
| Figura 5.21 - Desempenho dos utilizadores..... | 49 |

ÍNDICE DE QUADROS

| | |
|--|----|
| Tabela 2.1 - Dispositivo hápticos fixos. | 10 |
| Tabela 3.1 - Especificações Técnicas do <i>Geomagic Touch</i> (3D Systems, 2015). | 17 |
| Tabela 3.2 - Dimensões do dispositivo háptico. | 18 |
| Tabela 3.3 - Dimensões da mesa..... | 19 |
| Tabela 3.4 - Especificações do PC. | 20 |
| Tabela 4.1 - Dados dos voluntários..... | 33 |
| Tabela 5.1 - Dados dos voluntários para assinatura. | 46 |

SIMBOLOGIA E ABREVIATURAS

API – *Application Programming Interface*

AR – *Augmented Reality*

AVC – *Acidente Vascular Cerebral*

DOF – *Degrees of Freedom*

HDAPI – *Haptic Device API*

HLAPI – *Haptic Library API*

IDEA - *Individual Dental Education Assistant*

ISO – *International Organization for Standardization*

I&D – *Investigação e Desenvolvimento*

OpenGL – *Open Graphics Library*

P – *Proporcional*

PC – *Portable Computer*

PDD - *Phantom Device Driver*

PID – *Porporcional Integrated and Derivative*

RBG – *Red, Green and Blue*

RGB – *Vermelho, Verde e Azul*

SARA – *Software de Análise na Reabilitação da Assinatura*

SGBD – *Software de Gestão de Base de dados*

SQL - *Structured Query Language*

VR – *Virtual Reality*

CAPÍTULO 1

Introdução

O homem é um ser racional que desde os seus primórdios adaptou-se ao meio e como qualquer ser vivo este também possui necessidades básicas a serem saciadas. De modo a suprimir as suas necessidades o Homem cria técnicas (controlar as horas pela sombra, orientação pelas estrelas) e ferramentas (armas, utensílios, ciência, etc.) que as satisfaçam.

A criação do robô surge em resultado da necessidade da automatização de trabalhos a fim de se obter maior ritmo e qualidade de produção. Juntamente com a criação do robô, a necessidade de interação homem-máquina tornou-se um tópico de investigações e investimentos. Devido a essa lacuna surgiram *softwares* com a capacidade de produzir interfaces gráficas, utilizando diversas linguagens de programação tais como C, C++, C#, Python entre outras. Para o uso e comercialização de robôs são necessários normas e regras que garantem a saúde e segurança humana. Normas internacionais ajudam a clarificar conceitos, como interfaces que segundo ISO 9241-110 são “todas as partes de um sistema interativo (*software* ou *hardware*) que fornecem informações e controlos necessários para que o utilizador realize uma determine tarefa com o sistema interativo”. Uma interface tem por objetivo intermediar a comunicação homem-máquina o que implica que este seja o mais coerente, simples e intuitivo possível, dessa forma proporciona a segurança e a harmonia entre o equipamento e o humano. A figura 1.1 demonstra a interação homem-máquina.



Figura 1.1 - Interação homem-máquina

1.1 - Enquadramento

Este projeto é elaborado com o objetivo de obter o grau de mestre em Engenharia Eletrotécnica ministrado pelo Instituto Superior de Engenharia de Coimbra (ISEC), tendo como principal alvo a produção de um protótipo direcionado à medicina, mais concretamente à reabilitação de pacientes.

Os intelectuais possuem a mesma máxima, que o avanço do conhecimento, das técnicas e das tecnologias são pilares essenciais da ciência na compreensão da natureza. A área da saúde não é exceção, como pilar de uma sociedade é fundamental que esteja bem servida a nível de conhecimento e tecnologia. É um dos campos da ciência que mais requer atenção, sendo que a cada dia novas doenças e anomalias no homem são descobertas e para que se faça frente a essas demandas novas técnicas e tecnologias deverão ser criadas e investigadas. Grandes investimentos a nível de I&D nas componentes do *hardware* (sensores, robôs, câmaras...) e *software* têm sido feitas, para que cada vez mais estas se tornem indispensáveis ferramentas nas intervenções de saúde, aumentando assim a eficácia no ramo da hospitalar.

O presente projeto tem como finalidade a elaboração de uma solução de cariz prático e com argumentos substanciais, de modo dar uma possível resposta às necessidades da reabilitação da mobilidade da mão humana.

1.2 - Objetivos

Este projeto tem como principal objetivo a criação de exercícios e métodos analíticos que auxiliem na reabilitação da mobilidade da mão humana utilizando um *Geomagic Touch*. Para que o método analítico desenvolvido funcione corretamente este necessita de dados reais, esses dados são captados por intermédio de testes aos utilizadores. É de salientar que na construção do método analítico se criou uma interface gráfica que em conjunto com o dispositivo háptico permitiu criar um sistema capaz de produzir e captar a escrita elaborada pelo indivíduo. Após a captação da imagem da escrita são efetuadas operações matriciais à imagem, das quais resultam indicadores de erros que são guardados na base de dados.

De modo a atingir todos os objetivos, traçou-se um conjunto de etapas:

1º Etapa:

- Estudo e familiarização de conceitos em torno da temática: robô, interface, reabilitação; Revisão da literatura de interface homem-máquina;

2º Etapa:

- Compreensão do funcionamento do dispositivo háptico *Geomagic Touch*.
- Compreensão e utilização da lógica de programação do *Geomagic Touch*;

3º Etapa:

- Desenvolvimento da interface háptica;

4º Etapa:

- Compreensão do funcionamento da interface gráfica;

5º Etapa:

- Desenvolvimento da interface gráfica;

6º Etapa:

- Conjugação da interface háptica e interface gráfica;
- Desenvolvimento de algoritmos capazes de analisar imagens da assinatura;

7º Etapa:

- Compreensão e desenvolvimento da base de dados para o armazenamento dos dados resultantes do processamento da assinatura;

8º Etapa:

- Estudo dos testes da usabilidade; Interpretação e conclusões dos testes da usabilidade;

9º Etapa:

- Elaboração do relatório e conclusões.

1.3 - Organização do projeto

O projeto encontra-se estruturado em 6 capítulos, estando organizados da seguinte forma:

O primeiro capítulo ilustra e enquadra o tema, determina os objetivos e procura comunicar aos leitores uma perspetiva geral do projeto;

O segundo capítulo apresenta a revisão da literatura. Este capítulo expõe informações relacionadas aos dispositivos e interface hápticas existentes no mercado;

No terceiro capítulo, tópicos como hardware (*Geomagic Touch*) e software utilizados são abordados. Neste capítulo apresentam-se os modos de funcionamento, as características e a programação utilizada no *Geomagic Touch*;

O quarto capítulo ilustra a componente mais técnica e prática do projeto. Este capítulo foca-se na exibição dos processos de desenvolvimento das interfaces (háptica e gráfica): a construção dos algoritmos do *hardware* e *software*, o design do *software*.

O quinto capítulo, descreve e apresenta os testes de usabilidade e subsequentemente os seus resultados;

Por último o sexto capítulo, apresenta conclusões e as sugestões de melhorias para trabalhos futuros. Também nas secções seguintes são apresentadas as referências bibliográficas e os anexos.

CAPÍTULO 2

Revisão da Literatura

Destina-se a apresentação de informações introdutórias que demonstram o estado da arte sobre a temática abordada neste trabalho de projeto para a obtenção do grau de mestre, apresentando de forma breve a história e os conhecimentos produzidos ao longo da existência. Este capítulo é repartido em 2 secções a contextualização e o háptico. A contextualização ilustra a história a volta da temática robô e o háptico tendo como objetivo a apresentação de estudos e conhecimentos germinados em prol da ciência háptica.

2.1 - Contextualização

A terminologia robô (derivada de *robot/roboti* tem como raiz a palavra checa *robota*, a qual significa "trabalho forçado, servidão") foi usada pela primeira vez pelo Checo Karel Capek em 1921 na sua peça de teatro ("*Rossum's Universal Robots*") em Praga. Já o termo robótico foi popularizado pelo escritor de ficção científica Isaac Asimov no seu livro "*I, Robot*" em 1950. Nessa obra literária Isaac Asimov criou as Leis da Robótica, que segundo ele governariam os robôs no futuro. Já no séc. XX a construção de robôs popularizou-se devido às necessidades de grandes níveis de produtividade e qualidade dos produtos. Com níveis de exigências cada vez maiores, o mercado de robôs industriais tornou-se essencial à indústria mundial, tendo como pioneiros na robótica industrial George Devol e Joe Engelberg em conjunto com a empresa Unimation EUA em 1959. O 1º robô *Unimate* (fig. 2.1) foi instalado na Europa, em Metallverken, Uppsland Väsby, Suécia 1967 (Wikipédia, Robô industrial, 2020).

A evolução na indústria da robótica impulsionou o modo como os robôs atuam num ambiente de trabalho. Os robôs deixaram de desempenhar apenas papel ativo e pré-programado, tornaram-se capazes de agir e interagir com o ambiente (ativo e reativo). Por exemplo, num ambiente industrial em que um braço robótico aplica solda em estruturas metálicas, durante uma manobra o braço bate na estrutura metálica. Mesmo tendo danificado o metal e a ponta do braço este continua a exercer força sobre o local (papel ativo). Mas se este após embater no local interromper sua manobra ou então não efetuar o mesmo trajeto este desempenha um papel ativo e reativo em relação ao ambiente de trabalho.

Com passar dos anos, robôs físicos com a capacidade de ação e reação atraíram investidores, com novas ideias de aplicação e visões que favoreceram a inserção desses robôs em diversas outras áreas. Cada vez mais começaram a surgir interesses na capacidade de simular reações e interações com ambiente digital de

forma a produzir uma sensação háptica, impulsionando o desenvolvimento de sistemas hápticos.



Figura 2.1 – *Unimate* (Wikipédia, Robô industrial, 2020).

2.2 – Háptico

O termo háptico é proveniente do grego *haptikós* que significa “próprio para tocar, sensível ao tato” (Wikipédia, Háptico, 2020). No seio académico é definido como a ciência que incorpora o sentido do tato, cinestésica ou ambos na interação homem-máquina. A ciência háptica reúne conhecimento provenientes de ciências como a física, geometria computacional, neurofisiologia, biomecânica e ciências computacionais. Pesquisadores categorizaram campos de pesquisa relacionadas a ciência háptica em 4 campos de investigação (Abdulmtaleb El Saddik, 2011):

- Háptico Humano;
- Háptico aplicado à máquina;
- Háptico computacional;
- Háptico multimédia.

Háptico Humano – refere-se ao estudo da deteção e manipulação humana através de sensações táteis e cinestésicas.

Háptico aplicado à máquina – Envolve o design, a construção e o desenvolvimento de dispositivos mecânicos que possam substituir ou aumentar o tato humano.

Háptico computacional – está relacionada com o design e desenvolvimento de algoritmos e softwares que computam forças e simulam propriedades físicas de um objeto ao ser tocado, o que inclui a deteção de colisões e algoritmos que calculam as forças.

Háptica multimédia – envolve a integração e coordenação da apresentação do háptico numa aplicação multimédia.

Durante anos de interação homem-máquina, ideias sobre a possibilidade de aumentar o nível de imersão e a produtividade numa interação homem-máquina foram surgindo. Nessa caminhada para a produção de resultados, a ciência háptica demonstrou-se como uma das ferramentas fundamentais. Com isso, estudos e dispositivos com a capacidade háptica foram desenvolvidos ao longo dos tempos.

Um dispositivo háptico é um *hardware* com método de interação baseado no tato, o output é comandado por *software* que controla os inputs/outputs, os comandos produzidos têm a capacidade de simular o tato sobre um objeto virtual 3D. A fusão entre interação háptica e uma interface gráfica computadorizada (fig. 2.2) é uma das formas de obtenção de um elevado grau de imersão, sendo este um dos principais motivos na escolha deste tema.



Figura 2.2 – Comunicação háptica.

De modo que a junção da interface háptica e gráfica seja perfeita, a comunicação entre eles tem de ser atempada para que a relação entre o que o utilizador vê e sente seja sincronizado. O termo utilizado para identificar esse *timing* de resposta háptica e gráfica é o *feedback*. Os dispositivos hápticos de um modo geral encontram-se divididos em 2 grupos, essa divisão só é possível devido a um facto preponderante, a locomoção na interação com o utilizador. Os grupos são denominados de (Silva, 2014):

- Dispositivos hápticos Móveis;
- Dispositivos hápticos Fixos.

2.3 – Dispositivo hápticos móveis

Tem como fator diferenciador a sua locomoção quando interage com o utilizador, ou seja, acompanha os movimentos do utilizador para a produção de dados e normalmente possui a capacidade de ajustar a um determinado membro do corpo humano.

A fisionomia dos dispositivos hápticos móveis geralmente estão em concordância com o membro do corpo humano a que se destinam, o que é vantajoso nas vertentes de captação de dados cinemáticos, mantendo a liberdade de movimento. Alguns exemplos de interfaces hápticas móveis são citados abaixo.

CyberGlove II -São luvas equipadas com 22 sensores de flexão resistiva, para a captação de dados cinemáticos das mãos (Systems C. , Cyberglove II, 2017). Ilustração na Fig.2.3.



Figura 2.3 - CyberGlove II (Systems C. , Cyberglove II, 2017).

CyberGrasp – É uma versão otimizado da luva anterior, onde são adicionados alguns equipamentos (exosqueleto) que proporcionam uma capacidade reativa, ou seja, um feedback de força, o que confere maior imersão ao utilizador (Systems C. , CyberGrasp, 2017). Ilustração na Fig.2.4.



Figura 2.4 – CyberGrasp (Systems C. , CyberGrasp, 2017).

TeslaSuit – é um fato que incorpora *feedback* háptico pelo corpo, produz sensações de toque quando interage com ambiente virtual ou quando é dada a ordem via *software*. Também possui a capacidade de captação de dados do movimento e dados de indicadores fundamentais para a área da saúde (TeslaSuit, 2020). Ilustração na Fig.2.5.



Figura 2.5 - TeslaSuit (TeslaSuit, 2020).

2.4 – Dispositivos hápticos fixos

Como o próprio nome indica, no ato de realização de tarefas hápticas este permanece com a sua base fixa e estável. São os mais comuns e abundantes no mercado e no seio académico. Normalmente a componente a que o utilizador fica em contacto para a realização de tarefas denomina-se de *end-effector*. O *end-effector* do dispositivo háptico é o ponto mais extremo do dispositivo, é responsável pela captação e transmissão dos dados do utilizador à unidade de processamento. De entre os vários dispositivos hápticos fixos destacam-se 2, a nível comercial e académico, o *joystick* e o volante. Para meros utilizadores domésticos os gadgets com tecnologia háptica mais conhecidos são os presentes nas fig. 2.6 e 2.7, são utilizados como comandos de jogos de consola.



Figura 2.6 - Volante háptico (PcComponentes, 2020).



Figura 2.7 - Joystick Háptico (Newegg, 2020).

Tendo efetuado uma análise dos dispositivos hápticos existentes, começam a surgir informações úteis acerca do *hardware* a ser abordado neste trabalho. Segundo a classificação do tipo de dispositivo háptico o *hardware* para este trabalho é um dispositivo háptico fixo.

De seguida são apresentados um conjunto de dispositivos hápticos criados, sendo que alguns estão disponíveis no mercado. Além de serem dispositivos hápticos fixos, estes possuem características distintas como grau de liberdade, eixo de reação, gama de força e custo, estas características são importantes no momento da aquisição.

2.4.1 Catalogação dos dispositivos hápticos fixos

É imprescindível dizer que não são apenas os dados técnicos o único fator no momento da aquisição de um dispositivo, o preço é outro fator importante. De seguida apresenta-se a tabela 2.1, com as características dos *joysticks* hápticos. A organização na tabela 2.1 segue uma ordem ascendente segundo o grau de liberdade e de salientar que maioria dos valores associados aos *joysticks* estavam em Libras ou Dólares Americanos sendo necessária a conversão da moeda (data da conversão Abril de 2020).

Tabela 2.1 - Dispositivo hápticos fixos.

| Nome | Grau de Liberdade | Eixos de Reação | Gama de Força | Custo |
|--|-------------------|-----------------|---------------|------------------------|
| 1 DOF haptic device built with parts of recycled material (Gomes, Quintas, & Restivo, 2015) | 1 | 1 | 25 N | 80,00€ |
| Hapkit (Univercity, 2019) | 1 | 1 | 5 N | 47,00€ |
| The Pantograph (Christopher Rarnstein, 1994) | 2 | 2 | 10 N | - |
| Novint Falcon (Society, 2017) | 3 | 3 | Apro. 9 N | 182,00€ |
| Omega 3 (FORCE DIMENSION OMEGA 3, 2020) | 3 | 3 | 12 N | 17.000,00-23.000,00€ |
| Delta 3 (Inition, FORCE DIMENSION DELTA 3, 2020) | 3 | 3 | 20 N | 27.000,00-32.000,00€ |
| The HapticMaster (Delft Haptics Lab, s.d.) | 3 | 3 | 250 N | - |
| 4-DOF Haptic Device (Payandeh, 2008) | 4 | 3 | 3 N | - |
| Geomagic Touch (Systems 3. , 2020) | 6 | 3 | 3.3 N | 2.000,00€ |
| Phantom Premium (3dsystems, 2020) | 6 | 6 | 8.5 – 37.5 N | 30.000,00€ |
| Haption Virtuose 3D Descktop (Haption, 2019) | 6 | 3 | 10 N | 18.500,00€ |
| Haption Virtuose 6D (SA, 2019) | 6 | 6 | 35 N | 85.000,00€ |
| Omega 6 (Inition, FORCE DIMENSION OMEGA 6, s.d.) | 6 | 6 | 12 N | 21.000,00-46.000,00€ |
| Delta 6 (Inition, FORCE DIMENSION OMEGA 6, s.d.) | 6 | 6 | 20 N | 46.000,00 - 52.000,00€ |
| The Freedom 6S (J. G. S. Demers, 1998) | 6 | 3 | 2.5 N | 49.000,00€ |
| Omega 7 (Inition I. , 2020) | 7 | 4 | 12 N | 28.000,00 - 64.000,00€ |
| Sigma 7 (Inition I. , 2020) | 7 | 7 | 20 N | 64.000,00-127.000,00€ |

Não se limitando apenas a catalogar alguns dispositivos hápticos já criados, a seguir é exemplificado a área que estes se adequam. Instituições públicas/privadas manifestam interesse na tecnologia háptica, não só na melhoria do grau de imersão do homem, mas também na capacitação de jovens formandos para o mercado de trabalho.

O caso do *Haptkit*, é lecionado por uma instituição educacional que ensina aos jovens a sua construção.

Haptkit

Educação (projetos e pesquisas), Engenharia (Univercity, 2019). Ilustração na Fig. 2.8.



Figura 2.8 – *Haptkit* (Univercity, 2019).

Omega 3

Engenharia (robótica, design de produtos) (FORCE DIMENSION OMEGA 3, 2020). Ilustração na Fig. 2.9.



Figura 2.9 - *Omega 3* (FORCE DIMENSION OMEGA 3, 2020).

4-DOF Haptic Device

Medicina (Veterinária, Cirurgia), Educação (Payandeh, 2008). Exemplo Fig. 2.10.



Figura 2.10 - *4-DOF-Haptic Device* (Payandeh, 2008).

Sigma 7

Medicina (Cirurgia tele operada), Educação (Treinamento), Engenharia (Controle robótico, Design 3D, Tele Operação) (Inition I. , 2020). Exemplo Fig. 2.11.



Figura 2.11 - *Sigma 7* (Inition I. , 2020).

Geomagic Touch

Os seus exemplos de aplicação são: Medicina (Tele Operação, reabilitação), Entretenimento, Engenharia (Design produtos, Controle Robótico), Indústria (Modelação 3D) (Systems 3. , 2020). Exemplo Fig.2.12.



Figura 2.12 - Geomagic Touch (Systems 3. , 2020).

Neste projeto os esforços giram em torno de desenvolvimentos de aplicações com *Geomagic Touch*, é um hardware com diversas aplicabilidades e projetos já desenvolvidos. Os vários projetos desenvolvidos, a sua grande maioria concentra-se nas áreas da robótica e medicina pelo que de seguida serão apresentados alguns dos trabalhos.

Geomagic Touch na robótica:

Controlo braço robótico – Uso do *Geomagic Touch* no controlo do robô UR3 (Fig.2.13).



Figura 2.13 - Controlo do Robô UR3. (Soliwada, 2019).

VR4Robot – Controlo de robôs virtuais por intermédio do *Geomagic Touch* (Fig.2.14).



Figura 2.14 - VR4Robot (B.V., 2020).

Geomagic Touch na Educação:

Aplicação do *Geomagic Touch* escrita persa – consiste na prática de escrita persa utilizando o *Geomagic Touch*, sendo utilizados dois modos de escrita, o assistido e parcialmente assistido (Fig.2.15)

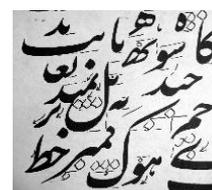


Figura 2.15 - Alfabeto Persa (Boroujen & Meghdari, 2009).

Geomagic Touch na Medicina:

Assistente Individual na Educação Odontológica (*IDEA*) – O sistema aumenta o realismo através do *feedback* da força e dos gráficos produzidos (International, 2019) (Fig.2.16).



Figura 2.16 – *IDEA* (International, 2019).

Industrial Virtual Reality Immersive Touch – O sistema integra dispositivo háptico em conjunto com localizador da cabeça, mãos e um sistema de alta resolução num monitor estereoscópico. Tudo isso com objetivo de simular uma operação (Immersivetouch, 2019) (Fig.2.17).



Figura 2.17 - *Industrial Virtual Reality Immersive Touch* (Immersivetouch, 2019).

A medicina possui vários ramos de especialidade, no entanto, nem todos os países têm especialistas a atuarem nesses ramos o que cria um défice. Mesmo havendo cooperações internacionais nem sempre os especialistas estão disponíveis ou dispostos a viajarem vários quilómetros para executar uma única intervenção ou apenas acompanhamento de doentes, ou seja, um problema devido à distância entre paciente e o médico ou até mesmo a falta de equipamentos adequados à intervenção cirúrgica. Ao longo dos tempos testemunhou-se avanços tecnológicos como a internet (4G, 5G), a robótica, displays (Led), sensores e atuadores. Todas essas inovações contribuem para uma vertente relativamente nova na medicina à Telemedicina, o que torna possível a cirurgia, o acompanhamento do doente, a reabilitação, a teleconferência, a teleconsulta entre outros, tudo isso à distância.

Algumas especialidades na medicina apresentam um potencial para a utilização de robôs no auxílio de intervenções cirúrgicas ou procedimentos médicos. As principais especialidades são: Urologia, Ginecologia, Cirurgia Torácica e Abdominal, Cirurgias Laparoscópicas, Neurocirurgia, Cirurgia geral, Cirurgia do Aparelho Digestivo, Cirurgia oral, Cirurgia Maxilofacial, Cirurgia Pediátrica, Cirurgia Cardiovascular, Patologia, Medicina Física e Reabilitação. Alguns destes ramos têm grande potencialidade para a integração do *Geomagic Touch*, no entanto para este projeto o ramo a ser focado é a Reabilitação.

Geomagic Touch e a reabilitação:

O *Geomagic Touch* com a sua arquitetura ergonómica de uma caneta, tem a capacidade de reprodução de forças, e de captação de dados empíricos relativos ao movimento da mão, demonstra ter requisitos atrativos para a sua exploração na reabilitação. A reabilitação também conhecida como Fisiatria é uma especialidade da medicina que tem por objetivo a promoção da função física e cognitiva das atividades comportamentais e da qualidade de vida. A reabilitação engloba muitas partes do corpo humano, porém tendo em conta que para o manuseio do *joystick* não só as mãos, mas os membros superiores em si podem ser reabilitados. Logo conclui-se que a secção do corpo a que direciona os estudos são os membros superiores (Fig. 2.18).

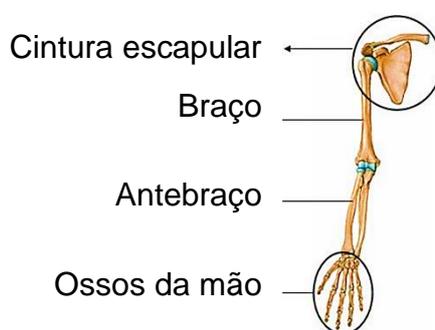


Figura 2.18 - Membros Superiores.

De uma forma genérica existem vários motivos que levam um indivíduo a reabilitação dos membros superiores tais como AVC (Acidente Vascular Cerebral), traumatismos, lesões entre outras. Um especial ênfase nos casos de AVC uma vez que apresenta números alarmantes a nível nacional, como mostram os dados do gráfico na fig. 2.19.

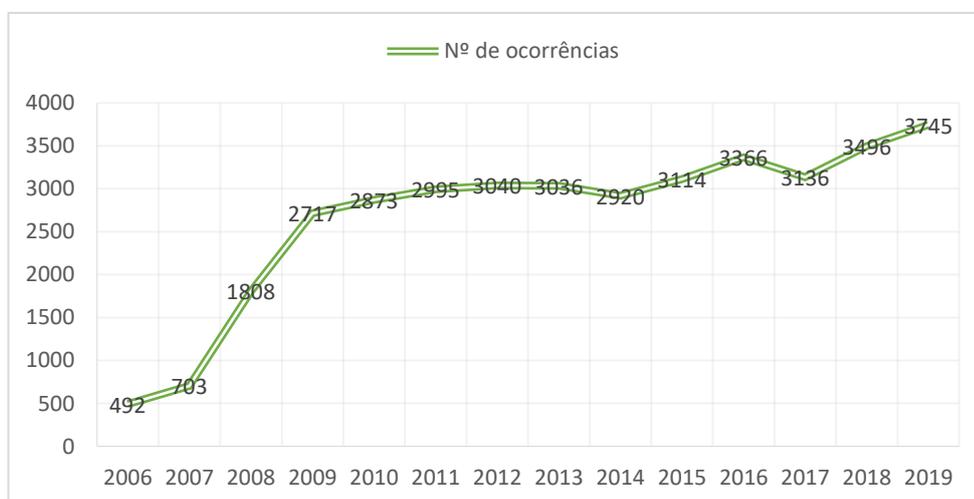


Figura 2.19 - Dados de ocorrência de AVC em Portugal.

Dados recolhidos neste gráfico levam em conta a população no geral, não demonstra quais as faixas etárias mais afetadas, mas demonstra uma progressão assustadora nos últimos 13 anos. A partir desses registos dos últimos 13 anos de 2006 a 2019 em média constata-se uma taxa de crescimento de 21,88% (INEM, s.d.). Sendo de referir que entre 2007 e 2009 houve um aumento superior.

Os números de ocorrências por AVC podem ser utilizados como referência na obtenção de uma amostra de pacientes, que podem ser encaminhados para reabilitação com o intuito de recuperar os movimentos. Com esses números é possível se ter mais ou menos uma noção de potenciais utilizadores com a necessidade da reabilitação da mobilidade da mão.

CAPÍTULO 3

Hardware e Software

Este capítulo engloba as diferentes ferramentas utilizadas na produção deste projeto, ferramentas como *Geomagic Touch*, computador, *Visual Studio* e a *Unity*. Dentro de cada tópico são descritos as principais características e o modo como este é implementado no produto criado.

3.1 – *Geomagic Touch*

É um instrumento que aumenta a produtividade e a eficiência em trabalhos que são necessários um elevado nível interação homem/computador, proporciona uma interação intuitiva, adicionando nessa relação homem-máquina a capacidade de resolver problemas através do toque.

O princípio de funcionamento da interação háptico provém da lógica de que quando o usuário manipula o *end-effector* do *Geomagic Touch* o sensor de posição converte e transmite esses dados ao computador.

Num intervalo de tempo de 1ms o computador analisa a existência ou não da colisão entre o *end-effector* e o objeto virtual. Em caso de colisão o sistema responsável pela modelação háptica calcula a reação (força/torque) a ser aplicada ao ponto de interação homem-máquina, o que conduz uma percepção tátil do objeto virtual (Abdulmaleb El Saddik, 2011).

O *feedback* de forças geradas pelo dispositivo háptico proporciona uma melhor e mais realista sensação *3D touch* para qualquer aplicação. Com o *end-effector* é possível captar e acompanhar todos os relevos e contornos dos objetos digitais. A tabela 3.1 apresenta as especificações técnicas do *Geomagic Touch*.



Figura 3.1 – *Geomagic Touch*
(Systems 3. , 2020).

Tabela 3.1 - Especificações Técnicas do *Geomagic Touch* (3D Systems, 2015).

| | |
|---|---|
| Espaço de Trabalho de Feedback de Força | >160 W x 120 H x 70 D mm |
| Área de Ocupação | ~ 168 W x 203 D mm |
| Massa do dispositivo | ~1.42 kg |
| Alcance do Movimento | Movimento da mão girando o pulso |
| Resolução de posição nominal | ~0.055 mm |
| Fricção de <i>Backdrive</i> | <0.26 N |
| Máxima força | 3.3 N |
| Força constante | > 0.88 N |
| Rigidez | Eixo X > 1.26 N/ mm Eixo Y > 2.31 N/mm Eixo Z > 1.02 N/mm |
| Inércia | ~45g |
| <i>Feedback</i> da força | XYZ |
| Sensor de Posição | XYZ (encoder digital) |
| Estilete de caneta | <i>Pitch, Roll, Yaw</i> (+/- 5% Potenciômetros de linearidade) |
| Interface | RJ45 compatível com Ethernet porta USB |
| Plataforma de Suporte | Intel ou AMD |
| <i>OpenHaptics</i> SDK compatibilidade | Sim |

Cada eixo do dispositivo háptico possui um limite físico, esse limite é notado quando o dispositivo háptico efetua um *stop* repentino no movimento. Sabe-se que o *Geomagic Touch* possui 6 de liberdade, mas no entanto só 3 é que possuem um motor. Uma vez interiorizada essa informação os cálculos para a análise do *gadget* são guiados tendo como principal enfoque os 3 eixos de rotação. A disposição das 3 articulações motorizadas é equivalente à configuração das articulações verticais ou antropomórficas (RRR). A seguir são demonstradas equações importantes no manuseamento e compreensão do dispositivo háptico. Baseando-se na representação dos parâmetros de Denavit – Hartenberg (Beckman, 2007) (seguindo Koul *et al*) é apresentada a cinemática direta e inversa do *Geomagic Touch*. A estrutura do *Geomagic Touch* é ilustrada pela fig. 3.2. e as suas dimensões são apresentadas na tabela 3.2. O *end-effector* na fig. 3.2 é representada pela letra “P”.

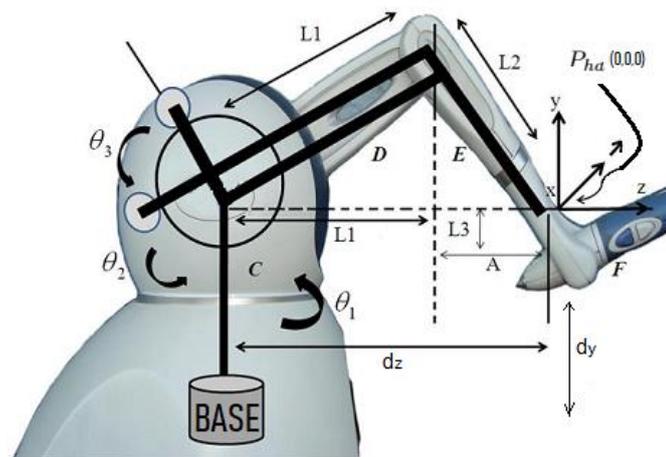


Figura 3.2 - Posição inicial do Geomagic Touch.

Tabela 3.2 - Dimensões do dispositivo háptico.

| | |
|----|------------|
| L1 | 133,35 mm |
| L2 | 133,35 mm |
| L3 | 30 mm |
| Dz | -168,35 mm |
| Dy | 133,35 mm |

Todas as dimensões apresentadas na tabela 3.2 são provenientes do posicionamento do *end-effector* na posição cartesiana $x=0$, $y=0$ e $z=0$. De seguida apresentam-se a matriz homogénea da base para o *end-effector* do dispositivo e a cinemática direta e inversa do dispositivo háptico (Majid H. Koul, 2010) (Paulo A. S. Mendes, 2020).

$$T_{End-effector}^{base} = \begin{bmatrix} c_1 & s_3 s_1 & -c_3 s_1 & -s_1(L_1 c_2 + L_2 s_3) \\ 0 & c_3 & s_3 & L_1 s_2 - L_2 c_3 \\ s_1 & -c_1 s_3 & -c_1 s_3 & c_1(L_1 c_2 + L_2 s_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Onde o c_1 e s_1 representam o coseno e o seno respetivamente do ângulo de rotação da 1ª articulação.

Tendo em consideração o valor da posição do *end-effector* dado pelo *software* do Geomagic Touch, é necessário adicionar as componentes d_y e d_z uma vez que a referência do espaço (mundo) é transacionada em d_y (segundo o eixo Y) e em d_z (segundo o eixo Z) em relação ao eixo da base.

Vindo assim a posição do *end-effector* dados pelas seguintes equações:

$$p_x = -s_1(L_1c_2 + L_2s_3) \quad (2)$$

$$p_y = -s_1(L_1c_2 + L_2s_3) \quad (3)$$

$$p_z = c_1(L_1c_2 + L_2s_3) + d_z \quad (4)$$

A cinemática inversa é dada pelas seguintes equações:

$$\theta_1 = \arcsin\left(-\frac{p_x}{L_1c_2 + L_2s_3}\right) \quad (5)$$

$$\theta_2 = \arcsin\left(\frac{p_y + L_2c_3 - d_y}{L_1}\right) \quad (6)$$

$$\theta_3 = \arcsin\left(\frac{(p_y - d_z)}{c_1} - L_1c_2\right) \quad (7)$$

Onde θ_1 , θ_2 e θ_3 são os ângulos de rotação das articulações correspondentes.

Em conjunto com o dispositivo háptico é utilizado uma mesa personalizada com a capacidade de manter estável o *Geomagic Touch* e prover uma superfície homogénea ao *end-effector*. A mesa é ilustrada pela fig. 3.3. e dados adicionais na tabela 3.3.

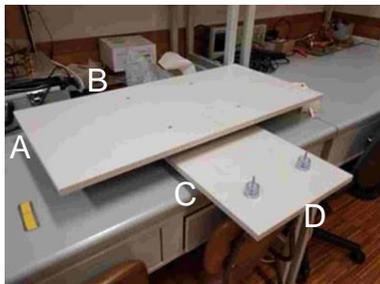


Figura 3.3 – Mesa.

Tabela 3.3 - Dimensões da mesa.

| Dimensão | Valor (mm) |
|----------|------------|
| A | 400 |
| B | 800 |
| C | 600 |
| D | 400 |

3.2 - Computador

Além do *Geomagic Touch* é necessária uma unidade de processamento externo para tratamento de dados, neste trabalho foi utilizado o PC apresentado na figura 3.6, com as especificações da tabela 3.4.

Tabela 3.4 - Especificações do PC.



Figura 3.4 - Acer Aspire.

| | |
|-------------------|-------------------------------|
| Sistema Operativo | Windows 10 Home |
| Fabricante | Acer |
| Modelo | Aspire E5-575G |
| Processador | i5-7200U CPU 2.50GHz 2.70 GHz |
| RAM | 8.00 GB |
| Tipo de Sistema | 64 bits |
| Placa Gráfica | NVidia GEFORCE 940MX 2GB |
| Ano Fabricação | 2017 |

3.3 - Visual Studio

Na programação do *Geomagic Touch* é necessário um ambiente de desenvolvimento em linguagem C++, C# e entre outros, a fim de ser possível a configuração e o controlo do dispositivo háptico. O *software* que melhor desempenha a função é o *Visual Studio* (fig. 3.5). É um software desenvolvido em várias edições pela *Microsoft*, porém para o desenvolvimento desse projeto é utilizada a versão *community*.



Figura 3.5 – Visual Studio logotipo.

3.4 - Unity

Para a produção gráfica a simplicidade e versatilidade de produção e utilização são fatores preponderantes na escolha e utilização de ferramentas para a produção gráfica. Por esses motivos foi escolhido a *Unity* (fig.3.6) como motor de produção gráfica. É um software desenvolvido pela empresa *Unity technologies*.



Figura 3.6 – Unity Logotipo.

É um motor de jogos que possibilita a criação e o desenvolvimento de jogos para sistemas operativos como *Windows*, *IOS* e *Linux*. A programação dos scripts do *Unity* é escrita no *Visual Studio* através linguagem *C#*, a fig.3.7 mostra um exemplo de aplicação da *Unity* na criação de conteúdo para o entretenimento.



Figura 3.7 - *The Heretic*.

As características principais da Unity são (Wikipédia, Unity, 2019):

- Suporte para o uso de *shaders*;
- Programação em *C#*;
- Suporte ao *PhysX*, incluindo detetor de colisão, *soft body* e *ragdoll*;
- Compatibilidade com os navegadores (via o *plugin Unity Web Player*): Internet Explorer, Firefox, Safari, Opera, Google Chrome e Camino;
- Compatibilidade com Blender, 3ds Max, Maya, Cinema 4D, Cheetah 3D, Softimage, modo, ZBrush, Lightwave, Photoshop, Fireworks e Substance.

As vantagens da Unity são:

- Interfaces esteticamente atrativas e intuitivas;
- Uso da *C#* ou *Java Script*;
- Alguns *scripts* pré-definidas;
- Grande comunidade de utilizadores e fóruns;
- Facilidade em encontrar tutoriais;
- Muitos modelos e acessos gratuitos;
- Motor de jogos muito realista;
- Facilidade de criar jogos para diversas plataformas;
- Criar jogos VR e AR;
- Criar jogos 2D, 3D e *multiplayer*.

As desvantagens da Unity são:

- Não executa modelação (caso necessário é importado de outros programas de modelação);
- Todas as configurações em Inglês;
- Versão grátis e versão paga, o que implica supressão de configurações;
- Necessários conhecimentos de programação para criar um jogo;
- Programa exige alguma memória e capacidade de processamento do computador.

É de salientar que todo o conhecimento e destreza adquirida acerca da *Unity* e de programação é tudo fruto da autoformação, tendo como principais ferramentas livros, trabalhos existentes e pesquisas na internet.

CAPÍTULO 4

Aplicações Desenvolvidas

Este capítulo dedica-se à apresentação de conteúdos resultantes de pesquisas e desenvolvimentos ao longo da elaboração deste projeto. Cada tópico neste capítulo aborda todo o trabalho desenvolvido a volta de conteúdos como Interface háptica, interface háptica-gráfica e interface gráfica.

Criou-se com a interface háptica exercícios físicos para reabilitação da mobilidade da mão humana, que consistem na reprodução de figuras geométricas. Para a interface háptica - gráfica desenvolveu-se ambientes gráficos destinados à escrita e adaptação ao dispositivo. E para completar foram também desenvolvidos conteúdos destinados à interface gráfica como a SARA (Software de Análise da Reabilitação da Assinatura).

4.1 - Interface háptica

A interface háptica é constituída por um dispositivo háptico e um *software* (baseado no mecanismo de controlo computacional), em conjunto proporcionam a comunicação homem-máquina através do tato (Abdulmotaleb El Saddik, 2011). Neste capítulo são especificadas todas as aplicações desenvolvidas viradas especificamente para a interface háptica.

4.1.1 - Controlo

Nas sessões de reabilitação, exercícios básicos como o de segurar um copo são executadas pelos pacientes sob a supervisão de agentes de saúde. Pouco a pouco o paciente recupera as funções motoras e com o tempo obtêm a sua independência uma vez perdida. A reabilitação da mobilidade das mãos possui a mesma linha de raciocínio, exercícios básicos e simples que normalmente são utilizadas para ensinar iniciantes a escrever.

Na primeira fase de reaprendizagem da escrita, tarefas simples são executadas, tais como a de refazer a trajetória de um círculo e um retângulo. Na reprodução das figuras há um auxílio do *Geomagic Touch*, esse auxílio é realizado por intermedio da aplicação de forças no *end-effector*. Essas forças aplicadas têm finalidade de corrigir o paciente encaminhando-o para a trajetória predefinida, dando a perceber ao paciente a sua performance.

De forma genérica, figuras geométricas são criadas a partir de linhas seja elas retas ou curvas, as linhas são um conjunto de pontos contínuos. Essas noções são

básicas, mas desmistificam e simplificam o modo de programar as figuras geométricas (círculo e retângulo), que a seguir são explicadas.

O deslocamento do *end-effector* é programado de forma que seja fruto do deslocamento entre 2 posições, posição atual (ponto preto) e a final (ponto cinza) como mostra a fig. 4.1. As coordenadas da posição atual variam ao longo do tempo consoante a força aplicada ao *end-effector*.

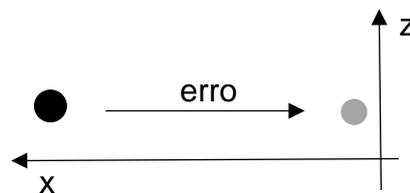


Figura 4.1 - Controlo proporcional.

O erro é o fator determinante no controlo da força (*output*). O erro consiste na diferença entre a posição atual (p_a) e a posição desejada (p_d) do dispositivo (Equação 8).

$$\text{erro} = p_d - p_a \quad (8)$$

O controlo de ganho proporcional utiliza apenas dados do presente para controlar o objeto. A equação do ganho proporcional gera um efeito mola no dispositivo háptico em que quanto mais se afasta da posição de repouso maior é a força reação (F_p) na direção oposta (Equação 9).

$$F_p = k_p * \text{erro} \quad (9)$$

Sendo o k_p a constante do ganho proporcional.

Também se ponderou a utilização do controlo PID (proporcional, integral e derivativo) para melhor controlar as forças aplicadas ao dispositivo háptico recorrendo as equações 10 e 11.

A fim de se atribuir uma certa estabilidade ao dispositivo háptico a componente integral é introduzida. O controlo integral consiste em manter registos dos acontecimentos do passado e produzir valores para estabilizar o dispositivo numa determinada posição (Equação 10).

$$F_i = k_i * \sum (\text{erro} * \tau) \quad (10)$$

Sendo, τ o período entre cada *loop* e k_i a constante do ganho integral.

E por último o controlo derivativo, tem por finalidade amortecer a paragem recorrendo a parâmetros que permitam a previsão de acontecimentos futuros (Equação 11). Tem como características:

- De forma prematura desacelerar o dispositivo;
- Analisar a variação da taxa do erro;
- Verificar o rácio de variação do erro.

$$F_d = kd * \frac{(erro - \text{último_erro})}{\tau} \quad (11)$$

Sendo, kd a constante do ganho derivativo.

Quando conjugadas estas 3 componentes no controlo do dispositivo háptico é obtido a força resultante como mostra a equação 12 e exemplificado na fig. 4.2.

$$F = F_p + F_i + F_d \quad (12)$$

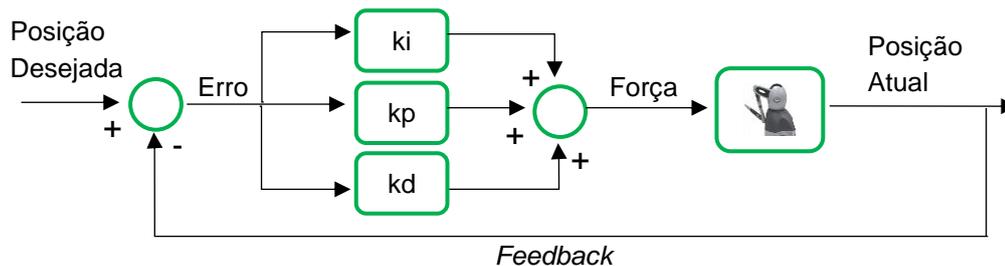


Figura 4.2 - Controlo PID.

Porém a opção de controlo PID foi descartada devido as seguintes situações:

- Quando aplicado o controlo PID ao dispositivo, este apresenta excessivas vibrações, o que coloca em risco tanto o utilizador como dispositivo háptico;
- Pouco tempo disponível para correção e melhoria da implementação do controlo PID;
- Pouco tempo de acesso ao dispositivo devido a situação atual do país.

Pelos motivos acima enunciados somente o controlo P. foi implementado e testado (fig. 4.3).

Porém o controlo de ganho proporcional possui desvantagens quando é aplicado a uma situação prática, este não leva em consideração a inércia da massa e a velocidade de transição entre pontos. A paragem da massa controlada pelo ganho proporcional é suscetível ao surgimento de vibrações no sistema, uma vez que está sempre a tentar corrigir o erro. Para colmatar essas dificuldades 2 medidas foram implementadas: Um limitador de força para o *end-effector* e um temporizador (cronometra o tempo que o *end-effector* fica dentro de um intervalo de posição

estabilizado). Estas medidas apresentaram uma melhoria no desempenho do dispositivo.

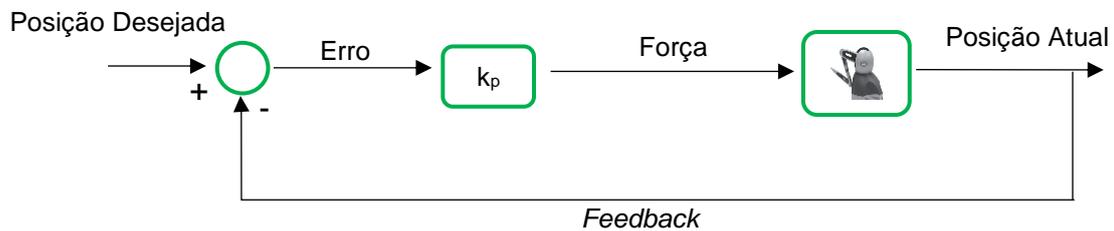


Figura 4.3 - Controlo P.

4.1.2 - Aplicações

Nesta secção demonstra-se como o desenho ou trajetória de um retângulo e um círculo são efetuados pelo utilizador recorrendo ao dispositivo háptico.

Retângulo

A ideia inicial era construir um retângulo constituído por 4 pontos posicionados sobre os 4 vértices do retângulo (fig.4.4). O desenho de um retângulo descrito pelo utilizador com *Geomagic Touch* seria no fundo mover o *end-effector* entre os pontos previamente criados. O algoritmo está presente no anexo A – Interface háptica.

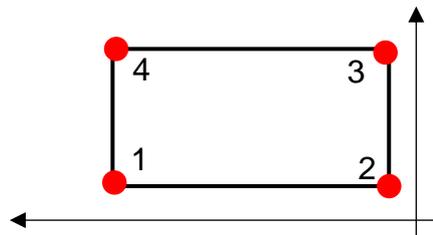


Figura 4.4 - Transição entre 4 vértices no retângulo.

Logo ao iniciar o programa o *end-effector* é deslocado para a posição 1, quando estabilizado repete o processo para as posições seguintes (2, 3 e 4) e por fim volta à posição 1 onde é terminado o processo de transição. É de salientar que só há transição para o próximo ponto quando o *end-effector* estiver estabilizado.

Na implementação do algoritmo são fulcrais 3 entradas:

- Localização dos vértices no espaço (p - coordenada tridimensional);
- Tempo entre pontos (t);
- Coeficiente proporcional de controlo (k_p).

Obs.: A partir deste ponto todo o algoritmo criado, destinado à interface háptica, foi implementado e testado no *Unity*.

Porém foi observado que quanto maior é a distância de transição maior é a possibilidade de ocorrerem desvios durante a trajetória efetuada pelo utilizador. A fim de reduzir o desvio, a solução criada é adicionar pontos intermédios localizados ao longo das arestas do retângulo (fig. 4.5)

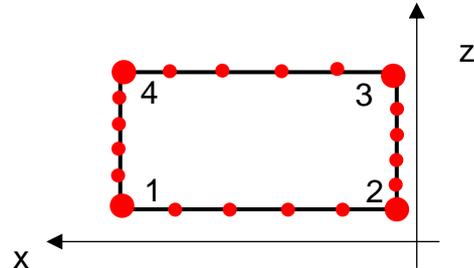


Figura 4.5 - Transição no retângulo por pontos intermédios.

Para o algoritmo de interpolação do retângulo, além das variáveis anteriormente anunciadas, é adicionada mais uma variável, o número de interpolações ou pontos intermédios (n).

A lógica por detrás desse algoritmo é a utilização da amplitude do distanciamento entre os vértices para a criação das posições dos pontos intermédios como mostra a equação 13.

$$d_c = p_i + c * \frac{p_f - p_i}{n} \quad (13)$$

Sendo,

c – Colocação do ponto intermedio no intervalo de 0 a n;

dc – Posição do ponto c;

pf – Posição final;

pi – Posição inicial;

n – Número de pontos intermédios (n>1).

Exemplo de aplicação:

Supondo que o *Geomagic Touch* está sobre o vértice número 1 (com n = 4), o *Geomagic Touch* desloca-se sequencialmente entre pontos até alcançar vértice número 2. Esta lógica é aplicada quando o *end-effector* do *Geomagic Touch* desloca entre os vértices do retângulo. Ao longo da transição ponto-a-ponto aplica-se um *timing* (de 0,5 a 1,5 segundos) para que o *end-effector* estabilize, a fig. 4.8 exemplifica a lógica algorítmica.

Nota: Os valores referentes tempo de espera são provenientes de ensaios efetuados com o dispositivo háptico.

Após ter ultrapassado essa fase foi constatado que seria uma boa prática ter pontos de inicialização e término do percurso junto à base do dispositivo háptico. Por esse motivo foram adicionados ao sistema 2 novos pontos de referência (fig.4.6).

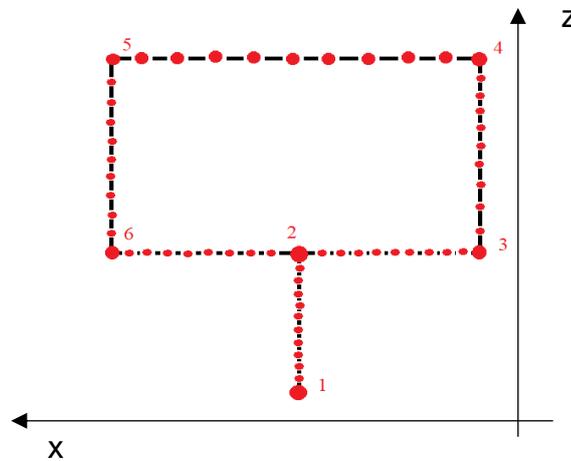


Figura 4.6 - Retângulo interpolado.

O programa passa a necessitar agora 6 conjunto de coordenadas tridimensionais, para que cada vez que o dispositivo atinge uma referência (1 a 6) o próximo ponto de referência da lista seja selecionado de forma sequencial como mostra a fig. 4.7. O processo de produção de pontos intermédios é equivalente ao processo anteriormente descrito.

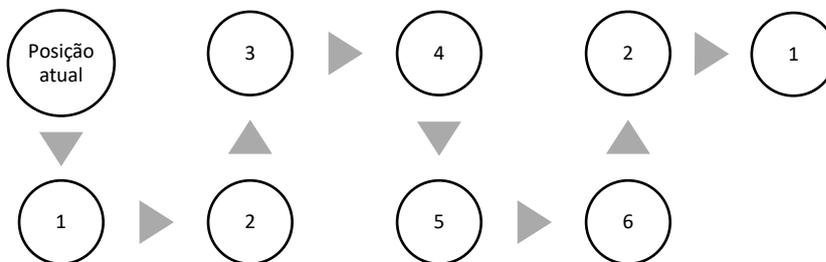


Figura 4.7- Sequência de deslocamento entre os pontos de referências.

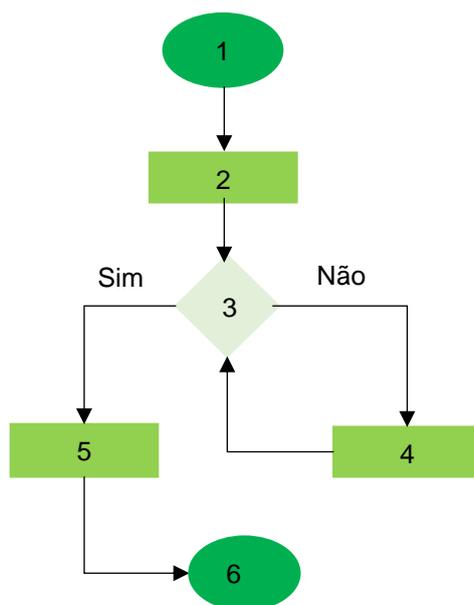


Figura 4.8 - Fluxograma do algoritmo.

| Etapas | Descrição |
|--------|--|
| 1 | Início |
| 2 | Captação de todas as variáveis necessárias ao controlo do dispositivo (n, pa, pd, t, kp) |
| 3 | Erro produzido inferior a +/- 5mm durante o período de estabilização |
| 4 | Medidas de correção do erro, produção de forças |
| 5 | Seleção da próxima posição |
| 6 | Fim |

Supondo que é aplicado um $n = 2$, quando é inicializado o programa (supondo que o utilizador tenha agarrado o *end-effector*), são aplicadas forças no sentido de guiar o *end-effector* à posição 1, após estabilizar, são criados automaticamente os pontos intermédios (desde a posição 1 até 2) aplicando a equação 13, após serem criados os pontos intermédios o *end-effector* desloca-se da posição 1 à posição d_1 (1º ponto intermedio). Com a chegada e estabilização na posição d_1 são aplicadas forças na direção à posição d_2 , que corresponde a posição 2. Todo o processo descrito anteriormente aplica-se às transições entre pontos de referência descrito pela fig. 4.7. A coordenada y é mantida constante segundo o plano da mesa. Na figura 4.8 é ilustrado o fluxograma que ilustra o mecanismo do algoritmo, aplicado a transição entre dois pontos.

Após experimentar este método, chegou-se à conclusão de que quanto maior o número de interpolações (n), menos vibrações existirão na trajetória. O programa com este algoritmo encontra-se no anexo A – Interface háptica (título – Retângulo interpolado).

Círculo

A criação do círculo (fig. 4.9) é feita também por interpolação, o que significa que a reprodução dos contornos de um círculo é formada por um conjunto de pontos descontínuos. A conjugação da transição entre esses pontos é uma aproximação de um círculo, esta interpolação é realizada ao longo do perímetro do círculo. A lógica por detrás desse algoritmo é a utilização de pontos cruciais de um polígono, fazendo sequencialmente a transição entre pontos. Ao longo da transição ponto-a-ponto aplica-se um *timing* (de 0,5 a 1,5 segundos) para que o *end-effector* estabilize.

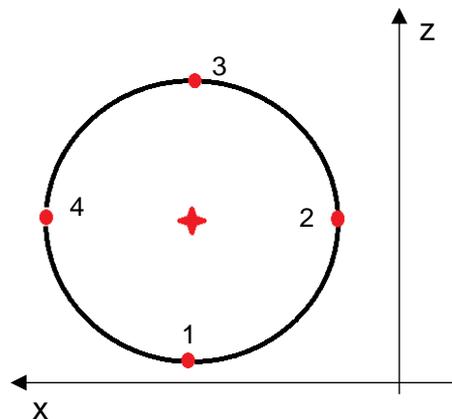


Figura 4.9 Círculo interpolado.

Para o algoritmo de interpolação do círculo são utilizadas 5 entradas:

- Número de interpolações (nc);
- Raio do círculo (r);
- Localização do centro do círculo (cc);
- Tempo entre pontos (tc)
- Coeficiente proporcional de controlo (kp).

As variáveis (Número de interpolação e Raio do círculo) são configuradas pelo utilizador. As restantes variáveis foram pré-definidas após um conjunto de experiências efetuadas (tentativa e erro). O fluxograma que melhor descreve a lógica algorítmica do círculo é equivalente à do retângulo (fig. 4.8). O programa que implementa o algoritmo encontra-se no anexo A – Interface háptica (tópico: Círculo Interpolado).

O algoritmo foi testado e implementado no Unity, tendo-se obtido com sucesso a transição por pontos ao longo da trajetória consoante os valores das variáveis. Após algumas experiências, chegou-se também à mesma conclusão de que quanto maior o número atribuído ao nc mais suave será a trajetória.

Nota: É aconselhável que o valor atribuído à variável nc seja par e superior a 4.

Foram feitas experiências preliminares envolvendo o controlo háptico, porém ficaram numa fase embrionária o que conduziu à sua não implementação na interface háptica-gráfica. Outros motivos que também conduziram à não implementação do controlo háptico na interface háptica-gráfica, foram os seguintes: O tempo necessário para adaptação e testes do controlo háptico nas interfaces; Aumento da exigência de processamento de dados produzidos a nível gráfico, uma vez que o computador utilizado já não corresponde à exigência. Dito isto, a escrita que é efetuada na interface háptica-gráfica não é auxiliada.

4.2 - Interface háptica – gráfica

Esta secção centraliza-se na descrição das aplicações criadas que envolvam a interação homem-máquina através do tato e da visão. Entre as aplicações desenvolvidas estão: menu de opções; *layouts* gráficos para a escrita da assinatura; e para teste do ambiente gráfico.

4.2.1 - Menu das opções

A figura 4.10, apresenta o menu de opções, que dá o acesso às funcionalidades principais do *software* como: *layout* para teste de adaptação, *layout* da escrita e *layout* para a análise da escrita.



Figura 4.10 - Menu opções.

O menu opções possui 4 botões: *Test* (*layout* para teste de adaptação), *Write* (*layout* da escrita), *Analyse* (*layout* para a análise da escrita), *Back* (Voltar). Botão *Test* direciona o utilizador a um ambiente onde efetua um conjunto de tarefas para se adaptar ao dispositivo háptico e ambiente virtual, o botão *Write* direciona o utilizador ao *layout* de escrita e aquisição de dados, o botão *Analyse* permite executar a interface direcionada ao especialista e o botão *Back* permite retornar ao menu principal.

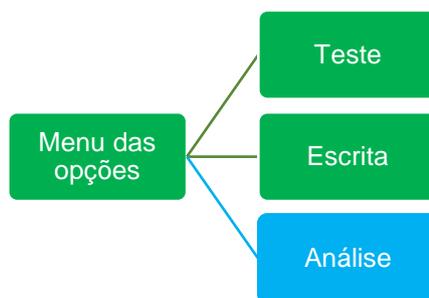


Figura 4.11 – Acessos proporcionados pelo menu.

Através do menu opções tem-se acesso às componentes da interface háptica – gráfica (Teste e Escrita) e também à interface gráfica (Análise), como ilustra a fig.4.11.

4.2.2 - Menu do ambiente para escrita

Na conceção de um produto 100% funcional para a reabilitação da escrita, com um ambiente esteticamente atrativo, a produção do *layout* manteve-se ancorado a atributos que chamassem atenção por ser simples, agradável, intuitivo e versátil. Após algumas pesquisas sobre trabalhos já realizados nessa ótica, muitos apresentavam bons desempenhos e viabilidade, mas a programação é complexa para programadores iniciantes em Unity. Após essas conclusões a missão era clara, desenvolver um programa simples e intuitivo capaz de reproduzir uma escrita, como mostra a fig. 4.12.



Figura 4.12 - Layout para a escrita.

A caneta com aparência azulada representa o dispositivo háptico, para aqueles que desejam controlar a espessura da linha recomenda-se a utilização do botão 0 e 1 do teclado, para diminuir e aumentar a espessura da linha respetivamente. A figura 4.13 ilustra o algoritmo que permite a escrita

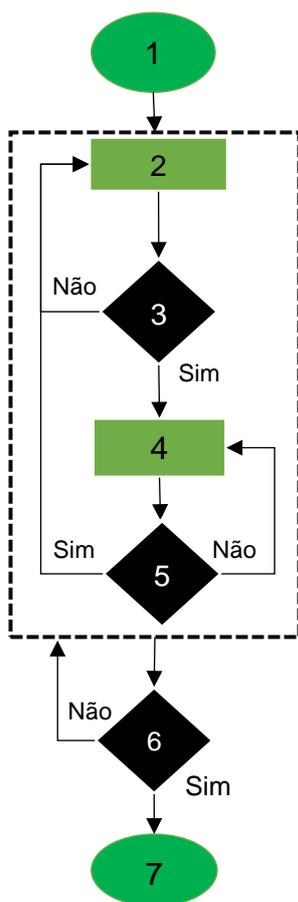


Figura 4.13 - Fluxograma do funcionamento da escrita.

| Etapa | Descrição |
|-------|---|
| 1 | Início – O dispositivo é iniciado, e é estabelecida a comunicação <i>Geomagic Touch-Unity</i> |
| 2 | A comunicação <i>Geomagic Touch-Unity</i> é feita continuamente para a troca de dados, nada é escrito na folha |
| 3 | <i>Geomagic Touch</i> entra em contacto com a folha? |
| 4 | É produzida uma linha que segue o trajeto da caneta (<i>Geomagic Touch</i>) quando o utilizador toca na folha |
| 5 | A caneta é levantada? |
| 6 | O botão <i>Exit</i> é acionado? |
| 7 | Fim – Termina a comunicação entre <i>Geomagic Touch-Unity</i> , desliga-se o <i>Geomagic Touch</i> |

Obs.: A linha tracejada significa que a qualquer instante durante a execução da aplicação o botão de saída tem a habilitação para encerrar o programa.

O algoritmo consiste em detetar o contacto entre caneta e a folha, quando esse contacto é detetado a tinta (*Brush*) é acionada e reproduzida várias vezes sobre a superfície.

Este menu para a escrita tem como funcionalidades adicionais:

- Apagar e guardar - caso pretender apagar o que foi escrito, o botão mais claro no *end-effector* tem essa funcionalidade de apagar tudo. E para que o registo da escrita seja feito o botão mais escuro tem essa funcionalidade indexada;
- Sair - possui a mesma funcionalidade que botão *Back*;
- Cor da tinta - é definida por uma mistura RGB presente no painel da direita.

Nota: Código de programação encontra-se no anexo B – Interface Háptica-Gráfica (secção Códigos de desenvolvimento para *Unity*).

Quando a velocidade da escrita é moderada a tal ponto que não ocorra o desfasamento entre a captação dos dados hápticos e gráficos, os resultados são satisfatórios como demonstram as figuras 4.14 e 4.15 (variação do tamanho das linhas).



Figura 4.14 - Exemplo 1.



Figura 4.15 - Exemplo 2.

Após ser produzido o *layout* procedeu-se a um teste de campo tendo como principal objetivo constar se o *layout* produzido é agradável e intuitivo para utilizadores que nunca utilizaram essa tecnologia. Foram utilizados 2 voluntários, caracterizados na tabela 4.1, que executaram a mesma tarefa (ver figura 4.16), com ambas as mãos.

Tabela 4.1 - Dados dos voluntários.

| Voluntários | A | B |
|--------------------------|----------|-----------|
| Género | Feminino | Masculino |
| Idade | 30 | 28 |
| Mão com que escreve | Direita | Direita |
| Doença | Nenhum | Nenhum |
| Lesão Membros Superiores | Nenhum | Nenhum |

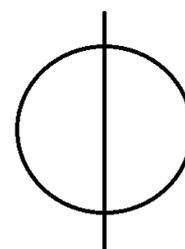


Figura 4.16 - Referência para escrita.

Uma constatação inicial é que tanto o voluntário A e o B, têm uma preferência pela escrita com a mão direita, possuindo maior controlo motor e da velocidade da mão direita do que da mão esquerda. Isto é demonstrado pelas imagens (fig. 4.17 a 4.20) captadas após a escrita em que na imagem da escrita da mão direita possui menos linhas descontínuas e rabiscos desnecessários do que a da mão esquerda. Salientando ainda que os voluntários nunca tinham tido contacto com esta tecnologia.



Figura 4.17 - Mão direita (A).



Figura 4.18 - Mão esquerda (A).



Figura 4.19 - Mão direita (B).



Figura 4.20 - Mão esquerda (B).

Posteriormente, o voluntário B teve um período de adaptação tendo efetuado ensaios no sistema criado, com isso instruiu-se acerca dos requisitos necessários para uma boa escrita, assim como o controlo e velocidades aceitáveis na escrita. O voluntário B, propôs a criação de um ambiente que proporcione ao utilizador inexperiente um teste *drive* no ambiente háptico-gráfico dando a oportunidade de se adaptar e melhorar a sua performance. Tendo em conta esta perceção procedeu-se à criação de um menu para testes (fig. 4.21).

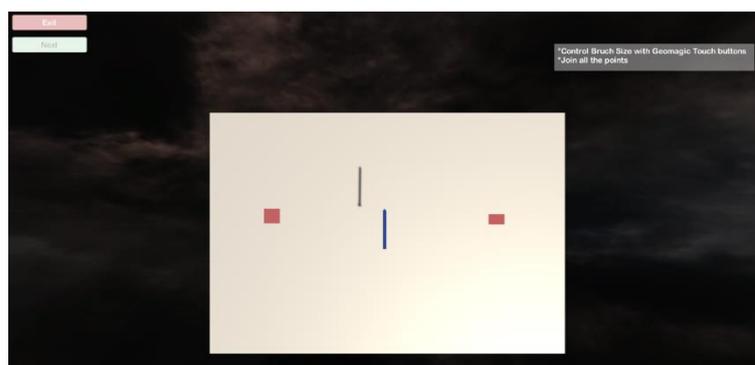


Figura 4.21 - 1ª tarefa e o menu de teste.

O menu possui 2 botões: *Next* (Próximo) e *Exit* (sair). Botão *Next* transita o utilizador para a próxima tarefa de forma sequencial, como mostram as figuras 4.22, 4.23 e 4.24. O *Exit* fecha o menu de teste e direciona o utilizador ao menu opções.

O menu de teste possui 2 botões: *Next* (Próximo) e *Exit* (sair). Botão *Next* transita o utilizador a próxima tarefa de forma sequencial, como mostram as figuras 4.22, 4.23 e 4.24. O *Exit* fecha o menu de teste e direciona o utilizador ao menu opções.

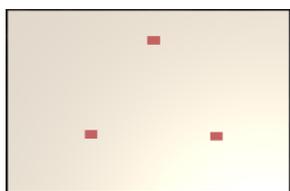


Figura 4.22 - 2ª Tarefa.

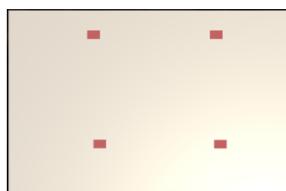


Figura 4.23 - 3ª Tarefa.

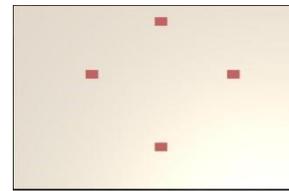


Figura 4.24 - 4ª Tarefa.

O utilizador após ultrapassar essas etapas, produz uma figura semelhante à que se apresenta na fig. 4.25, de forma genérica o paciente está habilitado para produção da assinatura.

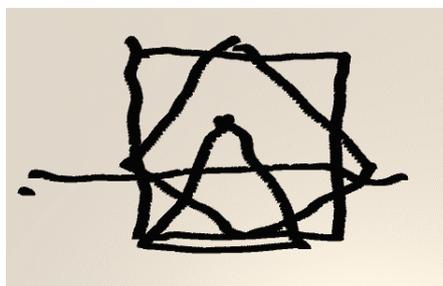


Figura 4.25 - Resultado do treino.

4.3 - Interface gráfica

Nesta secção são abordadas todas as aplicações criadas com a finalidade de captar, processar, analisar e armazenar os dados produzidos pelo utilizador sem recorrer a utilização do dispositivo háptica. Ao conjunto de *layouts* (interface gráfica) anteriormente designados por menu de análise, são denominados SARA (*Software de Análise a Reabilitação da Assinatura*).

4.3.1 - SARA

Aplicação gráfica criada com o objetivo central de captar, processar, analisar e armazenar os dados de uma imagem com escrita do utilizador. A fig. 4.26 mostra o funcionamento e as formas de interação da SARA.

Quando é inicializada a aplicação SARA, o utilizador deve fazer o *login*, para que possa aceder ao: Cadastro e Análise. Caso seja escolhido o Cadastro (fig. 4.28) o utilizador tem à sua disponibilidade a opção de voltar atrás ao *login* e campos de preenchimento que posteriormente são adicionados à base de dados. Caso seja

escolhida a *Análise* (fig. 4.29) o utilizador tem à sua disposição o *layout* dos Registos, o Tratamento de Imagens e a opção de fechar a aplicação.

Obs.: Todos os componentes ilustrados na figura estão ligados à base de dados.



Figura 4.26 – Esquema das camadas da SARA.

4.3.2 - Menu *login*

O menu *login* está conectado à base de dados para a verificação dos dados do utilizador, uma vez verificado que o utilizador está registado é consentida a sua progressão à próxima etapa que é o *layout* da análise. Caso o utilizador não esteja registado este tem a opção de efetuar registo na base de dados. Só é possível efetuar o *login* pela introdução correta de duas variáveis o nome e o NIF (fig. 4.27), caso contrário o acesso ao menu da análise é vetada. No campo destinado ao preenchimento do nome não se aceita números e no campo destinado ao preenchimento do NIF não se aceita letras. Todos os campos de preenchimento estão condicionados pelo número e tipo de caracteres, essas restrições visam salvaguardar as configurações estabelecidas na base de dados.

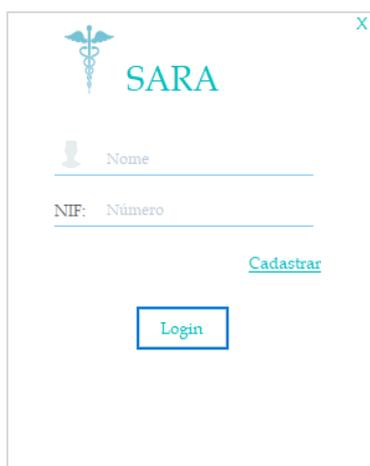


Figura 4.27 - Menu *Login*.

Obs.: Algoritmo ilustrativo das ações acima identificadas estão no anexo B (tópico: Login).

Se utilizador não estiver registado na base de dados este tem a opção de se cadastrar, clicando sobre o *link* “Cadastrar”. No menu *login* tem-se a capacidade de encerrar todo o programa clicando o botão “x”, localizado na extremidade superior direita.

4.3.3 - Menu cadastro

Para cadastrar um utilizador é necessário preencher todos os campos requisitados no *layout* do cadastro (fig. 4.28), caso contrário o registo do utilizador não é permitido. Os campos de preenchimento são: Nome, Apelido, NIF, Idade, Género, Telemóvel.

Os dados introduzidos pelo utilizador, são comparados aos dados já existentes na base de dados. Para efeitos de comparação o NIF é utilizado como variável chave. Se o número do NIF registado for igual ao já existente na base de dados o cadastro é vetado caso contrário o cadastro é viabilizado e os dados são adicionados à base de dados.



Figura 4.28 – Cadastro.

Todos os campos de preenchimento estão condicionados pelo número e tipo de caracteres, essas restrições visam salvaguardar as configurações estabelecidas na base de dados. Nos campos para o nome, apelido e género não aceitam caracteres numéricos e nos campos para o NIF, telemóvel, idade não aceitam caracteres alfabéticos.

Obs.: O algoritmo que implementa as funções descritas acima para registar dados e comparar, encontra-se no anexo B (tópico: Cadastro).

4.3.4 - Menu análise

O menu de análise é onde se encontra os *layouts* “Registos” e “Tratamento de Imagens”. Ao iniciar o menu de análise o utilizador visualiza a fig. 4.29. Para visualizar Registos ou Tratamento das Imagens clica-se num dos botões no painel a esquerda, também nesse *layout* o utilizador tem a opção de fechar a aplicação ou de voltar atrás (barra superior).

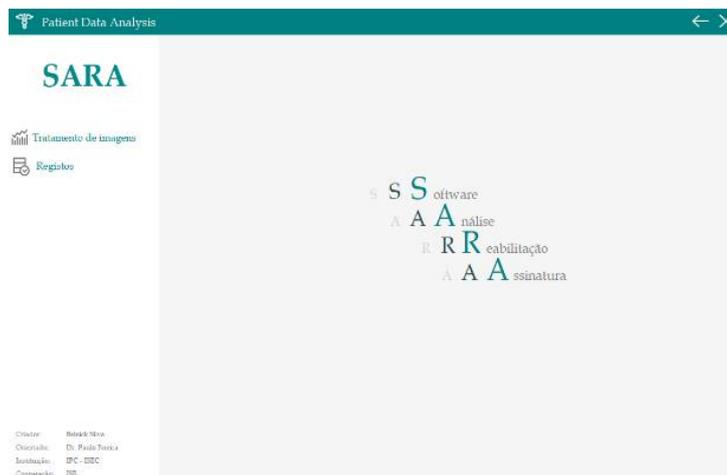


Figura 4.29 - *Layout* inicial da SARA.

4.3.4.1 - Menu dos registos

No *layout* do Registo (fig. 4.30) o operador tem a possibilidade de pesquisar e/ou eliminar dados de pessoas já registados na base de dados. Para pesquisar dados relacionados a um paciente, o utilizador deve preencher o campo NIF e clicar no botão “*Search*”. Caso esse número existir na base de dados, todos os dados associados ao NIF como N^o de paciente (gerado automaticamente) e o seu desempenho em cada escrita são exibidos, mas se o NIF introduzido não existir na base de dados o utilizador é notificado e nada é exibido ao utilizador. Quando clicado o botão “*Search*” as tabelas identificadas como “*Auxiliar Data*” exibem todos dados relacionados ao desempenho do paciente caso existam. Se houver a necessidade de apagar os dados associados ao NIF, deve-se preencher o campo destinados ao preenchimento com o número de paciente e pressionar o botão “*delete*”. Nesse caso apenas só são apagados os dados das tabelas *Auxiliar Data*, salvaguardando os dados da tabela “*Main Data*”. Se porventura o número de paciente não existir na base de dados é exibido imediatamente uma mensagem de notificação.

The screenshot shows the SARA application interface. At the top, there is a header 'Layout Análise' with a back arrow and a close button. Below the header, the application name 'SARA' is displayed. On the left side, there are two menu items: 'Processing Image' and 'Record'. The main content area is divided into several sections:

- Operation:** A section with a text input field for 'Nº Paciente' and a 'Delete' button.
- Main Data:** A table with columns: NIF, Name, NickName, Gender, Age, Phone. The data row shows: 100000000, xyz, lulu, Masculino, 51, 999988877.
- Auxiliar Data:** A section containing two tables.

| Eficiência | | | | | |
|-------------|---------|--------|--------|--------|--------|
| Nº Paciente | Anomaly | EFC1 1 | EFC1 2 | EFC1 3 | EFC1 4 |
| 5 | Lesão | 34,00 | 53,80 | 70,90 | 91,00 |
| 7 | Lesão | 31,00 | 34,90 | 27,90 | 29,90 |
| 8 | Lesão | 18,00 | 21,00 | 23,00 | 48,00 |
| 1007 | AVC | 59,00 | 15,00 | 89,00 | 5,00 |

| Eficácia | | | | | |
|-------------|---------|---------|---------|---------|---------|
| Nº Paciente | Anomaly | EFC A 1 | EFC A 2 | EFC A 3 | EFC A 4 |
| 5 | Lesão | 12,90 | 19,00 | 53,90 | 69,90 |
| 7 | Lesão | 16,90 | 39,90 | 65,70 | 92,70 |
| 8 | Lesão | 29,00 | 49,00 | 51,00 | 67,00 |
| 1007 | AVC | 15,00 | 65,00 | 100,00 | 86,00 |
- Search:** A section with a text input field for 'NIF' (containing '100000000') and a 'Search' button.

Figura 4.30 - Menu dos registos.

Obs.: O painel a esquerda está sempre disponível para se ter acesso ao *layout* para o processamento de imagens ou registos.

4.3.4.2 - Menu para tratamento de imagens

Por fim a caligrafia recolhida (imagem) tem de ser transformada em dados, recorrendo a ferramentas matemáticas e computacionais para que seja possível a análise do desempenho do paciente. O menu para tratamento de imagens é apresentado na fig. 4.31.

Este menu foi criado com os seguintes propósitos:

- Processamento das imagens de assinatura;
- Análise e ilustração do desempenho dos pacientes ao longo da reabilitação.

O menu para tratamento de imagens é constituído por 4 secções: dados recolhidos, gráfico dos índices de performance, caligrafia de referência (imagem), caligrafias executadas durante sessões de reabilitação (imagem). Para inserir as imagens nos respetivos blocos, basta clicar nos blocos que automaticamente é aberta uma janela que permite escolher as imagens.

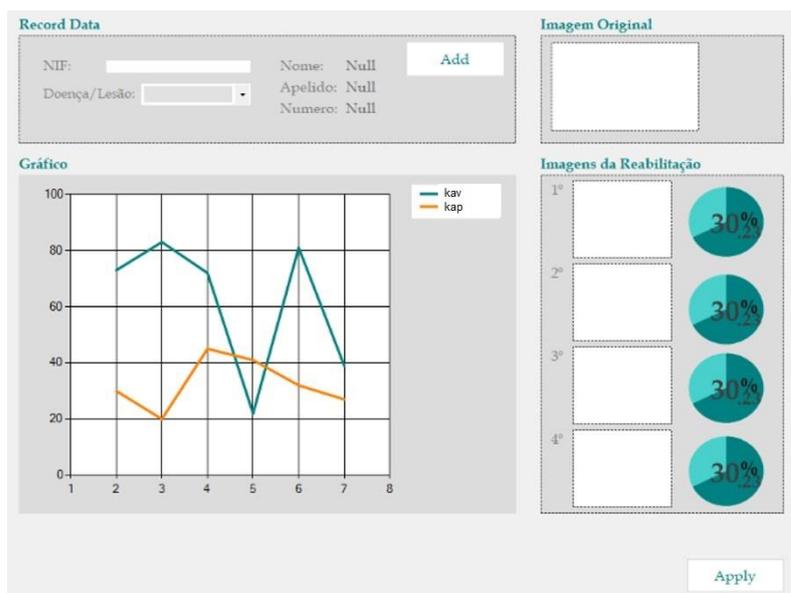


Figura 4.31 - Menu para tratamento das assinaturas.

Procedimentos para a utilização do menu:

1. Introduzir as imagens com caligrafia do paciente (antes e após o incidente) nos respetivos locais assinalados, tendo uma especial atenção na ordem com que são inseridas as imagens nos blocos “Imagens de Reabilitação”, porque cada imagem representa uma tentativa de reprodução da escrita;
2. Clicar no botão “Apply” para executar o algoritmo responsável pela captação e análise dos dados fornecidos pelas imagens das caligrafias;
3. A secção do gráfico apresenta os índices fornecidos pelo algoritmo;
4. Caso o utilizador deseje armazenar os dados produzidos pelo algoritmo este deve introduzir o número do NIF e seleccionar a causa dessa reabilitação na secção “Record Data”, e de seguida clicar no botão “Add”.

Nota: Uma vez clicado o botão “Add” os dados criados pelo algoritmo são associados a um número de paciente gerado automaticamente pela base de dados.

No tratamento de imagens, considerou-se que 3 situações podem ocorrer quando é executada uma assinatura:

- Escrita com uma certa rotação ou inclinação;
- Escrita em diferentes localizações na imagem;
- Escrita com uma escala diferente da imagem de referência.

Considerando estas potenciais situações, fica clara a necessidade de se proceder a uma translação, rotação e ajuste de escala na imagem. Para implementar a rotação, translação e ajuste de escala é necessária uma comparação entre as imagens.

Na elaboração de um algoritmo capaz de comparar duas imagens, algumas metodologias são adotadas:

1. Verificação da existência de imagens para processamento;
2. Acerto de escala na imagem do utilizador em recuperação (fig. 4.33 e 4.34);
3. Centrar todas as imagens (fig. 4.36);
4. Pintar a preto a imagem de referência (copiada de um documento) e a vermelho as imagens captadas após ocorrência;
5. Soma da caligrafia (imagem) de referência com cada uma das caligrafias (imagem) do paciente, enfatizando a amarelo os pixéis de sobreposição entre as duas imagens;
6. Captar os números de pixéis a preto, vermelho e amarelo;
7. Calcular rácios (equação 14 e 15) relacionados com a interceção entre as imagens (variáveis k_{av} e k_{ap});
8. Armazenamento dos ângulos de rotação, translação e escala das assinaturas do paciente e os rácios produzidos pela conjugação das imagens de caligrafias antes e após a ocorrência;
9. Repetição da etapa 2 a 7 desde os -90° a 90° ;
10. Após rodar a imagem até 90° é selecionado o ângulo de rotação que obteve o melhor valor de rácio;

$$K_{av} = \frac{\text{n}^\circ \text{ pixeis amarelo}}{\text{n}^\circ \text{ pixeis vermelho}} \quad (14)$$

$$K_{ap} = \frac{\text{n}^\circ \text{ pixeis amarelo}}{\text{n}^\circ \text{ pixeis preto}} \quad (15)$$

Nota: O rácio de referência para a seleção do melhor ângulo de rotação é a k_{av} , pois tem em sua essência a relação entre a escrita produzida e os pixéis sobrepostos.

Houve uma ponderação na criação de um indicador que relacione os pixéis a preto e a vermelho uma vez que as caligrafias nunca são iguais, ou gastam a mesma quantidade de pixéis.

Linguagens de programações atuais possuem várias técnicas e funções que proporcionam e facilitam o processamento de imagens. Com a linguagem C# diversas funções com diferentes abordagens como por exemplo o tratamento da imagem como uma matriz 2D (*Set/GetPixel*) ou como matriz unidimensional (*LockBits*).

Obs.: Normalmente as funções para o processamento de imagem (matriz 2D) abordam uma imagem pixel a pixel deslocando-se ao longo do comprimento e altura da imagem.

As equações 16 e 17 mostram como foi implementado o acerto de escala.

$$x = x_1/x_2 \quad (16)$$

$$y = y_1/y_2 \quad (17)$$

Onde,

x – Rácio entre os comprimentos das caligrafias;

x1 – Comprimento da caligrafia de referência;

x2 – Comprimento da caligrafia a efetuar acerto;

y – Rácio entre as alturas das caligrafias;

y1 – Altura da caligrafia de referência;

y2 – Altura da caligrafia a efetuar acerto.



Figura 4.32 - Assinatura inicial.

A assinatura representada pela fig. 4.32 após ser aplicada as equações de ajuste de escala de com a relação de 1:2 e 2:1, os resultados obtidos são representados pelas fig. 4.33 e 4.34 respetivamente.



Figura 4.33 - Assinatura 0.5x.



Figura 4.34 - Assinatura 2x.

Através das equações 18 e 19 é possível conjugar a rotação, centralização da assinatura. A aplicação das equações é demonstrada pelas figuras 4.35 e 4.36.

$$x = (p_x - m_x) * \cos \theta - (p_y - m_y) * \sin \theta + \frac{w}{2} \quad (18)$$

$$y = (p_x - m_x) * \sin \theta + (p_y - m_y) * \cos \theta + \frac{h}{2} \quad (19)$$

Onde:

x – Comprimento;

y – Altura;

Θ – Ângulo de rotação (radianos);

p_x – Posição do pixel em x;

p_y – Posição do pixel em y;

m_x – Ponto médio na figura na coordenada x;

m_y – Ponto médio na figura na coordenada y;

w – Comprimento da imagem;

h – Altura da imagem.

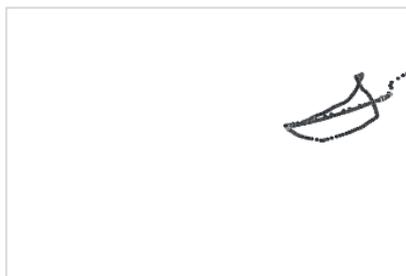


Figura 4.35 - Assinatura inicial.



Figura 4.36 - Assinatura centralizada e rodada.

A figura 4.37 mostra um exemplo de uma assinatura da pessoa após uma doença ou lesão e está em recuperação.



Figura 4.37 - Assinatura após lesão.

Utilizando as duas assinaturas do antes (fig.4.35) e após (fig.4.37) do incidente para a aplicação do algoritmo de centralização e rotação, o resultado obtido da conjugação das assinaturas resulta nas fig. 4.38 a 4.42. A assinatura antes da anomalia é enfatizada a preto, a assinatura após anomalia fica a vermelho e os pixéis sobrepostos na figura estão a amarelo. Essas utilizações de diferentes cores foram implementadas de modo a facilitar a visualização das assinaturas e as suas sobreposições.



Figura 4.38 - Conjugação com $\Theta = - 90^\circ$.



Figura 4.39- Conjugação com $\Theta = - 45^\circ$.



Figura 4.40 - Conjugação com $\Theta = 0^\circ$.



Figura 4.42 - Conjugação com $\Theta = 45^\circ$.



Figura 4.41 - Conjugação com $\Theta = 90^\circ$.

4.3.5 - Base de dados

A base de dados é uma ferramenta de recolha e organização de informação. Neste projeto os dados pessoais dos pacientes são adquiridos e organizados juntamente com os dados provenientes das imagens das caligrafias (assinatura) produzidas durante a recuperação da mobilidade das mãos.

Para a utilização da base de dados é necessário um *software* que permita a manipulação de dados assim como criação de conteúdos, uma SGBD (*Software* de Gestão de Base de Dados). Neste trabalho o *software* utilizado é o SQL Server.

Seguindo uma metodologia de concessão da base de dados o produto final obtido é o seguinte (Loureiro, 2018):

- Tipos de dados a serem recolhidos e tipologia:
 - INT – NIF, Idade, Telemóvel;
 - DECIMAL – Eficiência (k_{av}), Eficácia(k_{ap});
 - VARCHAR – Nome, Género, Doença ou Lesão.
- Número de tabelas necessárias: 4 quadros: Paciente, Género, Doença ou Lesão, Anomalia;
- Chaves Primarias: IDGenero (Género), NIF (Pacientes), ID_Doe_Les (Doença ou Lesão), Num_Paci (Anomalia);

- Chaves Estrangeiras: IDGenero (Pacientes), NIF (Anomalia), Num_Paci (Anomalia).

Nota: Passando o quadro de referência por um processo de normalização foram obtidas apenas 4 tabelas (ou entidades), com cardinalidade 1-N.

Após elucidar as informações resultantes da aplicação da metodologia o próximo passo é ilustrar o diagrama ER (Relação entre as Entidades), como mostra a fig. 4.44.

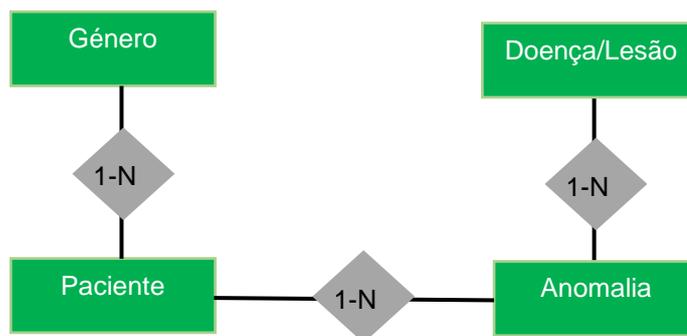


Figura 4.44 - Diagrama ER.

Empregou-se números como identificadores nas entidades, Género e Doença/Lesão pelo simples fato deste proporcionar maior rapidez na procura de dados e poupança no armazenamento. Quando implementado no *SQL Server* o diagrama E.R. fica com o aspeto apresentado na fig.4.45.

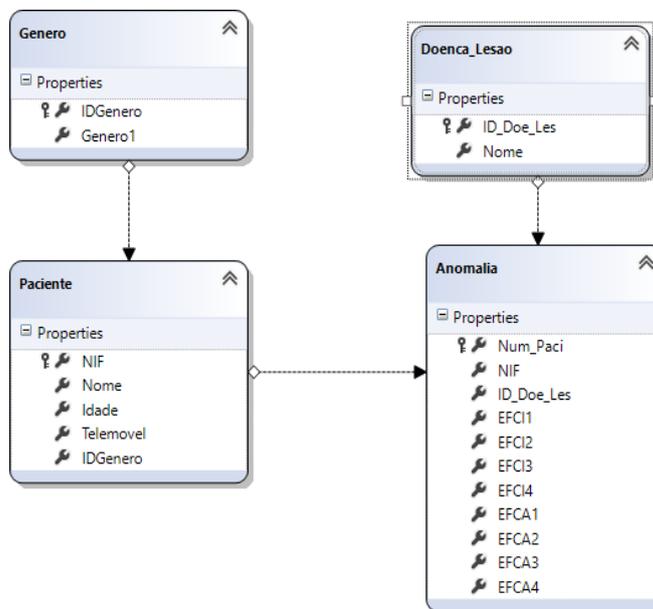


Figura 4.45 - Arquitetura E.R.

CAPÍTULO 5

Testes e resultados

Na realização de um projeto, tecnologias e metodologias criadas precisam de ser postas à prova. Neste projeto são realizados testes de campo para que todo trabalho realizado demonstre o seu real valor aos olhos do consumidor. Os testes foram feitos com 4 voluntários em perfeita saúde, na faixa etária dos 23 aos 29 anos, apresentados na tabela 5.1.

Os testes têm a finalidade de criar um valor médio aceitável de acerto de uma pessoa ao tentar reproduzir uma assinatura inicial. O valor médio obtido será usado como um valor de referência para os indicadores de desempenho de pacientes em reabilitação.

Na elaboração de testes de campo é necessária uma metodologia para que todo conjunto de resultados e informações sintetizados estejam organizados de forma sistemática e clarividente.

Metodologia para a recolha e processamento da caligrafia dos voluntários:

1. Recolha de dados pessoais do voluntário;
2. Produção da caligrafia de cada voluntário, utilizando o *Geomagic Touch* em conjunto com a *Unity*; a. Produção de uma assinatura inicial; b. Quatro assinaturas suplementares na tentativa da reprodução da assinatura inicial;
3. Captação da imagem da assinatura transformada em imagem png;
4. Processamento e análise dos dados provenientes da imagem da assinatura por intermédio da SARA;
5. Organização e elaboração de dados estatísticos fornecidos pela SARA;
6. Extração de valores de referência para a comparação com outros tipos de paciente;
7. Reflexão conclusiva sobre os resultados obtidos.

Nota: Por motivos de proteção de dados dos voluntários, os nomes são fictícios.

Tabela 5.1 - Dados dos voluntários para assinatura.

| Nome | Idade | Género | Saúde |
|------|-------|-----------|----------|
| JL | 23 | Masculino | Conforme |
| BM | 29 | Feminino | Conforme |
| MF | 25 | Feminino | Conforme |
| BMS | 24 | Masculino | Conforme |

5.1 - Captação das assinaturas

Durante os experimentos as caligrafias recolhidas foram divididas em 2 partes, a assinatura de referência e assinaturas de ensaio. São 4 os ensaios efetuados porque supondo que o sistema venha a ser testado com pacientes com real necessidade de reabilitação e a cada semana do mês necessite de avaliar do seu progresso na recuperação da mobilidade das mãos. Com essa possível situação, SARA tem a capacidade de dar resposta a avaliação mensal do paciente na reabilitação da mobilidade da mão humana.

As assinaturas do voluntário JL, são apresentadas nas figuras 5.1 a 5.5.:

Figura 5.1 - Assinatura de referência (voluntário JL).

Figura 5.2 - 1º Ensaio do JL.

Figura 5.3 - 2º Ensaio do JL.

Figura 5.4 - 3º Ensaio do JL.

Figura 5.5 - 4º Ensaio do JL.

As assinaturas do voluntário BM, são apresentadas nas figuras 5.6 a 5.10.

Figura 5.6- Assinatura de referência do voluntário BM.

Figura 5.7 - 1º Ensaio do BM.

Figura 5.8 - 2º Ensaio do BM.

Figura 5.9 - 3º Ensaio do BM.

Figura 5.10 - 4º Ensaio do BM.

As assinaturas do voluntário MF, são apresentadas nas figuras 5.11 a 5.15.



Figura 5.11 - Assinatura de referência do voluntário MF.



Figura 5.15 - 1º Ensaio do MF.



Figura 5.14 - 2º Ensaio do MF.



Figura 5.12 - 3º Ensaio MF.



Figura 5.13 - 4º Ensaio MF.

As assinaturas do voluntário BMS, são apresentadas nas figuras 5.16 a 5.20.



Figura 5.16 - Assinatura de referência do voluntário BMS.



Figura 5.17 - 1º Ensaio do BMS.



Figura 5.18 - 2º Ensaio do BMS.



Figura 5.19 - 3º Ensaio do BMS.



Figura 5.20 - 4º Ensaio do BMS.

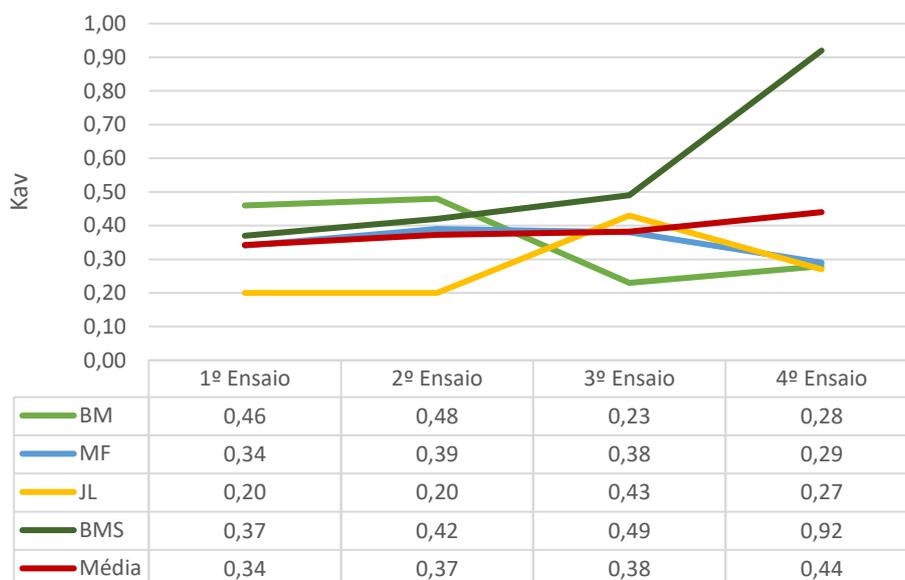


Figura 5.21 - Desempenho dos utilizadores.

Nota: O valor k_{av} de acerto obtido tem a possibilidade de não ser tão assertivo devido aos contornos da escrita, linhas descontínuas com a aparência de pontos.

$$\text{Média genérica} = \frac{\sum \text{Media por semana}}{n^{\circ} \text{ de semanas}} \quad (20)$$

O valor da média genérica (Equação 20) calculada é de 0,38, este valor será utilizado como padrão de comparação do desempenho dos utilizadores futuros, vale ressaltar que o valor obtido é proveniente da assinatura de utilizadores com boa saúde. Mas como se sabe quanto mais testes forem executados mais dados são produzidos e mais acertos à média genérica são implementadas, concluindo assim que o valor resultante desta experiência poderá ser considerado como preliminar. Em projetos futuros que venham a dar continuidade a este conteúdo, o valor da média genérica está sujeito a ajustes.

Utilizadores com lesões e doenças de grande severidade o valor de referência está sujeito a ajustes por parte do responsável pela reabilitação do paciente, uma vez que cada caso requer um cuidado e cada paciente é um quadro médico. Uma possível melhoria para caso venha a ser implementado o sistema num ambiente de reabilitação, seria a determinação de uma média específica para cada quadro clínico.

Conclui-se que mesmo um utilizador com boa saúde dificilmente iria executar duas assinaturas 100% idênticas, a possibilidade de ter essa ocorrência são escassas, porém o resultado obtido pela conjugação das imagens das caligrafias enfatiza o quão próximo um utilizador está de repetir a caligrafia de referência, o que demonstra ser um indicador de avaliação. Já com um utilizador com a saúde debilitada a percentagem de acerto teoricamente é inferior à de um utilizador de boa saúde.

CAPÍTULO 6

Conclusões, e Sugestões de Melhoria

Este capítulo dedica-se à apresentação de conclusões após o término deste projeto, e sugestão de melhorias que possam ser implementadas de modo a melhorar futuras teses que venham a derivar deste projeto.

6.1 - Conclusões

Neste projeto o principal objetivo é a aquisição e produção de conhecimentos, que contribuam para a criação de soluções viáveis para a reabilitação da mobilidade da mão humana com auxílio do *Geomagic Touch*.

No decorrer do processo de aquisição e produção de conhecimentos, foram criadas diversas interfaces hápticas, háptica-gráficas e gráficas. Recorrendo a um conjunto de ferramentas de baixo custo, com linguagens de programação de grande variedade de aplicação. O protótipo criado é dinâmico, aberto à adição de outras tecnologias. Segundo o *feedback* de alguns utilizadores, o protótipo é "...intuitivo, com menus de fácil compreensão e manuseio".

Os testes e resultados comprovaram a potencialidade e aplicabilidade do protótipo desenvolvido, permitindo assim afirmar que foi concluído com sucesso o objetivo traçado.

6.2 - Sugestões de Melhoria

Espera-se que no futuro melhorias venham a ser implementados a nível do *software*, que possam melhor desempenho no processamento das imagens e na comunicação entre o computador e dispositivo háptico. Assim sugiro as seguintes melhorias:

- Otimização do algoritmo para a diminuição do tempo de processamento da imagem;
- Teste de campo com dados de pacientes em reabilitação;
- Adição de outros exercícios a serem executados;
- Criação de base de dados que interliguem exercícios de recuperação com desempenho na escrita;
- Criação de um algoritmo que por intermédio de uma figura a preto e branco seja possível ser criado automaticamente um plano de trajetória para o dispositivo háptico;
- Inteligência artificial para a escolha da melhor definição para a compatibilidade entre imagens
- Limitar a quantidade de tinta produzida por contacto do *Geomagic Touch* no ambiente digital;
- Melhor a velocidade de captação dos dados hápticos e gráficos;
- Utilização de um computador com excelente placa gráfica e processador.

Por fim, este sistema deve ser testado nos centros de reabilitação e assim demonstrar o seu real contributo na reabilitação da mobilidade da mão humana.

Bibliografia

- 3D Systems, I. (2015). *Openhaptic Toolkit*. 333 Three D Systems Cir: 3D Systems, Inc.
- 3dsystems. (2020). *3D Systems Phantom Premium*. (3dsystems) Obtido em 14 de 04 de 2020, de <https://www.3dsystems.com/haptics-devices/3d-systems-phantom-premium>
- Abdulmotaleb El Saddik, M. O. (2011). *Haptics Technologies*. Berlin: Springer.
- Abdulmtaleb El Saddik, M. O. (2011). *Haptics Technologies*. Berlin: Springer.
- B.V., T. C. (18 de 04 de 2020). *Tree C Technology*. Obtido de Tree C Technology B.V.: <http://www.vr4robots.com>
- Beckman, J. A. (2007). *The PHANTOM Omni as an under-actuated robot*. Iowa State University.
- Boroujen, M. M., & Meghdari, A. (2009). Haptic Application in Persian Calligraphy. *International Conference on Computer and Automation Engineering*. Iran.
- Christopher Rarnstein, V. h. (24-28 de 04 de 1994). The PANTOGRAPH. *The PANTOGRAPH: a Large Workspace Haptic Device for a Multi-Modal Human-Computer Interaction*, p. 2.
- Delft Haptics Lab, D. U. (s.d.). *Delft haptics lab*. (Moog FCS Robotics) Obtido em 14 de 04 de 2020, de <https://delfthapticslab.nl/device/hapticmaster/>
- FORCE DIMENSION OMEGA 3. (2020). (Inition) Obtido em 14 de 04 de 2020, de <https://www.inition.co.uk/product/force-dimension-omega-3/>
- Gomes, P. U., Quintas, M. R., & Restivo, M. T. (13 de 12 de 2015). 1 DOF haptic device build with parts of recycle material. *Design and Test of a 1 DOF Haptic Device for Online Experimentation*, p. 3.
- Haption. (2019). *Área de trabalho 3D Virtuouse*. (Haption SA) Obtido em 14 de 04 de 2020, de <https://www.haption.com/fr/products-fr/virtuose-3d-desktop-fr.html>
- IEE. (13 de 04 de 2020). *Unimate*. Obtido de ROBOTS YOUR GUIDE TO THE WORLD OF ROBOTICS: <https://robots.ieee.org/robots/unimate/>
- Immersivetouch. (2019). *Medical Virtual Reality Solutions*. Obtido de ImmersiveTouch, Inc.: <http://www.immersivetouch.com>
- INEM. (s.d.). *N.º registos AVC por NÚMERO DE CASOS POR ANO - 2019*. (INEM) Obtido em 18 de 04 de 2020, de http://oldsite.inem.pt/stats_avc/stats.asp?stat=0&CODU=&DISTRITO=&MES=&ANO=2019
- Inition. (2020). *FORCE DIMENSION DELTA 3*. (FORCE DIMENSION) Obtido em 14 de 04 de 2020, de Inition: <https://www.inition.co.uk/product/force-dimension-delta-3/>

- Inition. (s.d.). *Force Dimension Delta 6*. (Force Dimension) Obtido em 14 de 04 de 2020, de <https://www.inition.co.uk/product/force-dimension-delta-6/>
- Inition. (s.d.). *FORCE DIMENSION OMEGA 6*. (FORCE DIMENSION) Obtido em 14 de 04 de 2020, de <https://www.inition.co.uk/product/force-dimension-omega-6/>
- Inition, I. (2020). *Force Dimension Omega 7*. (Force Dimension) Obtido em 14 de 04 de 2020, de Inition: <https://www.inition.co.uk/product/force-dimension-omega-7/>
- Inition, I. (2020). *FORCE DIMENSION SIGMA 7*. (FORCE DIMENSION) Obtido em 14 de 04 de 2020, de FORCE DIMENSION: <https://www.inition.co.uk/product/force-dimension-sigma-7/>
- International, I. (01 de 11 de 2019). *THE DENTIST'S FUTURE EDUCATION*. Obtido de IDEA International: <http://www.ideadental.com/>
- J. G. S. Demers, J. M. (1998). Freedom 6S Force Feedback Hand Controller. *Freedom 6S Force Feedback Hand Controller*, p. 6.
- Loureiro, H. (2018). *C# 7.0*. Lisboa: FCA - Informática.
- Majid H. Koul, P. K. (25-27 de Maio de 2010). Gravity Compensation for PHANToMTM Omni® Haptic Interface. *The 1st Joint International Conference on Multibody System Dynamics*, p. 10.
- Newegg. (16 de 04 de 2020). *PC Joystick, YF2009 USB Gaming Controller with Vibration Feedback and Throttle, Wired Flight Stick for PC Computer Laptop*. Obtido de Newegg: <https://www.newegg.com/p/1B4-06RX-02B11>
- P. U. Gomes, M. R. (13 de 12 de 2015). 1 DOF haptic device build with parts of recycle material. *Design and Test of a 1 DOF Haptic Device for Online Experimentation*, 3.
- Paulo A. S. Mendes, A. P. (Janeiro de 2020). Hand Exercise Using a Haptic Device. p. 14.
- Payandeh, A. M. (2008). *Analysis and Experimentation of a 4-DOF Haptic Device*. Reno, NE, USA: IEEE.
- PcComponentes. (16 de 04 de 2020). *Logitech G29 Driving Force para PS4/PS3/PC*. Obtido de PcComponentes: https://www.pccomponentes.pt/logitech-g29-driving-force-para-ps4-ps3-pc?kk=a4c6361-1766d1473df-1c8de&gclid=CjwKCAiA_eb-BRB2EiwAGBnXXi3QkQwcLGLMNHg3FEiNS7Hgeqk3lfrpCkYtqV_UDQ5529H8OPpsWBoCIMYQAvD_BwE&utm_source=kelkoopt&utm_medium=cpc&utm_campaign=kelkoocl
- SA, H. (2019). *Virtuose™ 6D*. (Haption SA) Obtido em 14 de 04 de 2020, de <https://www.haption.com/fr/products-fr/virtuose-6d-fr.html>

- Silva, J. M. (2014). *Estudo e desenvolvimento de um dispositivo háptico de dois graus de liberdade baseado em atuadores de corrente contínua*. Porto: FEUP.
- Society, V. R. (2017). *The Novint Falcon Haptic System*. (Novint Technologies, Inc.) Obtido em 04 de 14 de 2020, de <https://www.vrs.org.uk/virtual-reality-gear/haptic/novint-falcon.html>
- Soliwada, D. (18 de 03 de 2019). *Using Geomagic Touch to control UR3 robot*. Obtido de <https://www.youtube.com/watch?v=804NdWn56o4>
- Systems, 3. (2020). *Touch*. (3dsystems) Obtido em 14 de 04 de 2020, de 3d System: <https://www.3dsystems.com/how-to-buy/geomagic-touch>
- Systems, C. (2017). *Cyberglove II*. Obtido em 14 de 04 de 2020, de Cyberglove System: <http://www.cyberglovesystems.com/cyberglove-ii/>
- Systems, C. (2017). *CyberGrasp*. Obtido em 14 de 04 de 2020, de Cybergrasp System: <http://www.cyberglovesystems.com/cybergrasp>
- TeslaSuit. (2020). *TeslaSuit*. Obtido em 14 de 04 de 2020, de VR Electronics Ltd: <https://teslasuit.io/the-suit/>
- Univercity, S. (28 de 10 de 2019). *hapkit*. Obtido de <https://hapkit.stanford.edu/>
- Wikipédia. (05 de 05 de 2019). *Medicina Física e Reabilitação*. Obtido de Wikipédia: https://pt.wikipedia.org/wiki/Medicina_f%C3%ADsica_e_reabilita%C3%A7%C3%A3o
- Wikipédia. (09 de 2019). *Unity*. (Wikipédia) Obtido em 18 de 04 de 2020, de <https://pt.wikipedia.org/wiki/Unity>
- Wikipédia. (13 de 02 de 2020). *Háptico*. Obtido em 06 de 03 de 2020, de Wikipédia: <https://pt.wikipedia.org/wiki/H%C3%A1ptico>
- Wikipédia. (13 de 04 de 2020). *Robô industrial*. Obtido de https://pt.wikipedia.org/wiki/Rob%C3%B4_industrial

ÍNDICE DOS ANEXOS

| | |
|---|----|
| Anexo A – Interface Háptica..... | 57 |
| Anexo B – Interface Háptico – Gráfica | 60 |
| Anexo C – Base de dados..... | 76 |

Anexo A – Interface Háptica

Retângulo 4 pontos

```

if (ii == 0) {
    fx = 0.4;
    fz = 0.3;
    fy = 1.8;}
else {
    fx = 0.3;
    fz = 0.3;
    fy = 0.893413;}
const HDdouble pre[4][3] = { { 24, -10,-10 },
                             { -24, -10,-10 },
                             {-24, -10,10},
                             {24,-10,10} };

hdBeginFrame(hdGetCurrentDevice());
hduVector3Dd position;
hduVector3Dd desiredPos, quant;
desiredPos[0] = pre[ii][0];
desiredPos[1] = pre[ii][1];
desiredPos[2] = pre[ii][2];
hdGetDoublev(HD_CURRENT_POSITION, position);
hduVecSubtract(quant, desiredPos, position);
hduVecScale(desiredPos, quant, kp);
if (ii != 0) { fx = fz = 0.5; }
if (sqrt(pow(desiredPos[0], 2)) > 0.4)desiredPos[0] =
desiredPos[0]/sqrt(pow(desiredPos[0], 2))*fx;
if (sqrt(pow(desiredPos[1], 2)) > 0.9)desiredPos[1] =
desiredPos[1]/sqrt(pow(desiredPos[1], 2))*fy;
if (sqrt(pow(desiredPos[2], 2)) > 0.4)desiredPos[2] =
desiredPos[2]/sqrt(pow(desiredPos[2], 2))*fz;
if ((position[0] > pre[ii][0] - 3) && (position[0] < pre[ii][0] + 3)) {
    if ((position[1] > pre[ii][1] - 5) && (position[1] < pre[ii][1] + 2)) { if
((position[2] > pre[ii][2] - 2) && (position[2] < pre[ii][2] + 2)) {
        fx = fz = 0.3;
        ii++;
        if (ii > 3)ii = 0;}}}

```

Retângulo Interpolado

```
hdBeginFrame(hdGetCurrentDevice());
if (state== 5){
    aux = 1;
    aux1 = 1;}
pri2 = fin[aux, 0] - fin[state, 0];
sec2 = fin[aux, 1] - fin[state,1];
ter2 = fin[aux, 2] - fin[state,2];
for (int i = 0; i < linha; i++){
    step[i, 0] = fin[state, 0] + pri2 / linha * (i);
    step[i, 1] = fin[state, 1] + sec2 / linha * (i);
    step[i, 2] = fin[state, 2] + ter2 / linha * (i);}
vx = mass * (step[count, 0] - position[0]);
vy = mass * (step[count, 1] - position[1]);
vz = mass * (step[count, 2] - position[2]);
float [,] v={vx, vy, vz};
if ((position[0] >= step[count, 0] - 0.4f) && (position[0] <= step[count, 0] + 0.4f)) {
    if ((position[1] >= step[count, 1] - 0.4f) && (position[1] <= step[count, 1] + 0.4f)) {
        if ((position[2] >= step[count, 2] - 0.4f) && (position[2] <= step[count, 2] + 0.4f)){
            count++;
            if (count > 9){
                state++;
                count = 0;}
            if (count_1 == 1){
                state = 0;
                count_1 = 0;}
            if (state > 5){
                state = 1;
                count_1 = 1;}}}}
//-----temporizador e ordem de movimento
timer -= Time();
if (timer< 0) {
    hduVecScale(v,v,speed*time);
    hdSetDoublev(HD_CURRENT_FORCE,v)
    timer = .5f;
}
hdEndFrame(hdGetCurrentDevice());
```

Círculo Interpolado

```

void Start(){
    rb = this.GetComponent<Rigidbody>();
    xp = new float[divisao];
    zp = new float[divisao];
    float alfa = 0f;
    xp[0]=x+ raio * Mathf.Cos(alfa);
    zp[0]=z+ raio * Mathf.Sin(alfa);
    for (int i = 1; i < divisao; i++){
        alfa =(float)(alfa + 2 * 3.14 /(float) divisao);
        xp[i] = x + raio * Mathf.Cos(alfa);
        zp[i] = z + raio * Mathf.Sin(alfa);
    }
}
void FixedUpdate(){
    timer -= Time.deltaTime;
    if (timer < 0){
        rb.MovePosition(transform.position + ((new Vector3(xp[j], 0, zp[j])) *
0.5f*Time.deltaTime));
        j++;
        if (j >=divisao){
            j = 0;}
        timer = 0.25f;}}

```

Anexo B – Interface Háptico – Gráfica

Código desenvolvimento para *Unity*

Botão Desliga

```
public void QuitGame()
{
    Application.Quit();
}
```

Botão análise

```
public void winform()
{
    Process.Start(@"C:\Users\Belnick
Silva\Desktop\C++\unity\Demo\App1\Al
ternativas Set e
GetPixel\LockBits\teste_0\teste_0.sl
n");
}
```

Escrita

```
if (ObjNome.touching.name == "Plane"){
    Brush.transform.position = Grabber.position;
    var Ray = CameraF.ScreenPointToRay(Grabber.position
RaycastHit hit;
    if(Physics.Raycast(Ray, out hit))
    var go = Instantiate(Brush, new Vector3(Grabber.position.x,0.1f,Grabber.position.z),
Quaternion.identity, transform);
    go.transform.localScale = Vector3.one * BrushSize;
    go.name =(i++).ToString();
}
}
```

Controlar o tamanho da tinta

```
if ((Input.GetKey(KeyCode.Alpha1)){
    BrushSize += 0.001f;
    if (BrushSize > 0.1f)
    {
        BrushSize = 0.1f;
    }
}
else if
(Input.GetKey(KeyCode.Alpha0)){
    BrushSize -= 0.001f;
    if (BrushSize < 0.0f){
        BrushSize = 0.0f;}
}
```

Botão Escrita

```
public void PlayGame()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene
()).buildIndex + 2);
}
```

Botões *Back*(teste e escrita)

```
public void Sair(){
    SceneManager.LoadScene(SceneManager.GetActiveScene
()).buildIndex - 1);}
public void Sair_test(){
    SceneManager.LoadScene(SceneManager.GetActiveScene
()).buildIndex - 2);}
}
```

Configuração dos botões do *Geomagic Touch*

```
int bot1 = ObjNome.Buttons[0];
int bot2 = ObjNome.Buttons[1];
if (bot1==1)
{
    Salvar();
}
if (bot2==1)
{
    SceneManager.LoadScene("Scene_1");
}
```

Salvar Assinatura

```

public void Salvar(){
    StartCoroutine(CoSave());}
private IEnumerator CoSave(){
    yield return new WaitForEndOfFrame();
    Debug.Log("Salvar");
    Debug.Log(Application.dataPath + "/saveImage.png");
    RenderTexture.active = Foto;
    var texture2D = new Texture2D(Foto.width, Foto.height);
    texture2D.ReadPixels(new Rect(0, 0, Foto.width, Foto.height), 0, 0);
    texture2D.Apply();
    var data = texture2D.EncodeToPNG();
    string fileName = Snap();
    File.WriteAllBytes(fileName, data);
}
string Snap()
{
    ii++;
    return string.Format("{0}/Scenes/Imagens/Cam/saveImage_{1}_{2}.png",
Application.dataPath,ii, System.DateTime.Now.ToString("yyyy-MM-dd_HH-mm-ss"));}

```

Controlar a Cor

```

void Start(){
    rend = GetComponent<Renderer>();
}
void Update(){
    rend.material.SetColor("_Color",new Vector4(red,green,blue,1.0f));}
public void AjRed(float newRed){
    red = newRed;}

public void AjGreen(float newGreen){
    green = newGreen;}
public void AjBlue(float newBlue){
    blue = newBlue;}
}

```

Código de desenvolvimento da Interface de Análise

Cadastro

```
public Form1(){
    InitializeComponent();}
private void Nome2_Click(object sender, EventArgs e){
    Nome2.Clear();
    Nome2.ForeColor = Color.Black;
    Nome2.MaxLength = 20;}
private void Apelido2_Click(object sender, EventArgs e){
    Apelido2.Clear();
    Apelido2.ForeColor = Color.Black;
    Apelido2.MaxLength = 20;}
private void NIF2_Click(object sender, EventArgs e){
    NIF2.Clear();
    NIF2.ForeColor = Color.Black;
    NIF2.MaxLength = 9;
}
private void Tel_Click(object sender, EventArgs e){
    Tel.Clear();
    Tel.ForeColor = Color.Black;
    Tel.MaxLength = 9;}
private void Back_Click(object sender, EventArgs e){
    Form2 form2 = new Form2();
    form2.Show();
    this.Close();}
private void Idade_Click_1(object sender, EventArgs e){
    Idade.Clear();
    Idade.ForeColor = Color.Black;
    Idade.MaxLength = 2;}
private void Tel_KeyPress(object sender, KeyEventArgs e){
    char ch = e.KeyChar;
    if (!char.IsDigit(ch) && ch != 8){
        e.Handled = true;}}
private void Idade_KeyPress(object sender, KeyEventArgs e){
    char ch = e.KeyChar;
    if (!char.IsDigit(ch) && ch != 8){
        e.Handled = true;}}
private void Nome2_KeyPress(object sender, KeyEventArgs e){
    char ch = e.KeyChar;
    if(!char.IsLetter(ch) && !char.IsWhiteSpace(ch) && !char.IsControl(ch)){
        e.Handled = true;}}
private void Apelido2_KeyPress(object sender, KeyEventArgs e){
    char ch = e.KeyChar;
    if(!char.IsLetter(ch) && !char.IsWhiteSpace(ch) && !char.IsControl(ch)){
        e.Handled = true;}}
private void NIF2_KeyPress(object sender, KeyEventArgs e){
    char ch = e.KeyChar;
    if(!char.IsDigit(ch) && ch != 8){
        e.Handled = true; }}
```

Login

```
private void Cadastrar_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e){
    Form1 form1 = new Form1();
    form1.Show();}
private void Close_Click(object sender, EventArgs e){
    Nome.Clear();
    NIF.Clear();
    this.Close();}
private void Nome_Click(object sender, EventArgs e){
    Nome.Clear();
    Nome.ForeColor = Color.Black;
    Nome.MaxLength = 50;}
private void NIF_Click(object sender, EventArgs e){
    NIF.Clear();
    NIF.ForeColor = Color.Black;
    NIF.MaxLength = 9;}
private void Nome_KeyPress(object sender, KeyPressEventArgs e){
    if(!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) &&
!char.IsWhiteSpace(e.KeyChar)){
        e.Handled = true;}
    }
private void NIF_KeyPress(object sender, KeyPressEventArgs e){
    if (!Char.IsDigit(e.KeyChar)&& e.KeyChar!=8){
        e.Handled = true;}}
```

Registos

```
private void search_nif_Click(object sender, EventArgs e){
    search_nif.Clear();
    search_nif.ForeColor = Color.Black;
    search_nif.MaxLength = 9;}
private void search_nif_KeyPress(object sender, KeyPressEventArgs e){
    if (!Char.IsDigit(e.KeyChar) && e.KeyChar != 8 ){
        e.Handled = true;}}
private void np_delete_Click(object sender, EventArgs e){
    np_delete.Clear();
    np_delete.ForeColor = Color.Black;
    np_delete.MaxLength = 9;}
private void np_delete_KeyPress(object sender, KeyPressEventArgs e){
    if (!Char.IsDigit(e.KeyChar) && e.KeyChar != 8){
        e.Handled = true;}}
private void delete_button_Click(object sender, EventArgs e){
    query = "Select * from Anomalia where Num_paci='" + np_delete.Text.Trim() + "'";
    SqlDataAdapter sda = new SqlDataAdapter(query, sqlcon);
    DataTable dtbl = new DataTable();
    sda.Fill(dtbl);
    if (dtbl.Rows.Count == 1){
        delete_np();
        label_update.Visible = true;
        update_pic.Visible = true;}
    Else{
        MessageBox.Show("Number not found");}
    sqlcon.Close();
    sda.Dispose();
    dtbl.Dispose();}
```

Código de Conexão a Base de Dados

Cadastro

```
string ConnetionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=Projeto;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSub
netFailover=False";
private void registrar_Click(object sender, EventArgs e){
    if (NIF2.Text == "Número" || Idade.Text == "Número" || Nome2.Text == "Nome" ||
Apelido2.Text == "Último Nome" || Tel.Text == "Número" || Genero.Text == "" || NIF2.Text
== "" || Nome2.Text == "" || Apelido2.Text == "" || Idade.Text == "" || Tel.Text == ""){
    MessageBox.Show("Fill every field");}
    else{string IDG="0";
        if (Genero.Text == "Masculino") IDG = "1";
        else if (Genero.Text == "Femenino") IDG = "2";
        else IDG = "3";
        using (SqlConnection sqlcon = new SqlConnection(ConnetionString))
        {sqlcon.Open();
            SqlCommand sqlCmd = new SqlCommand("DataAdd",sqlcon);
            sqlCmd.CommandType = CommandType.StoredProcedure;
            sqlCmd.Parameters.AddWithValue("@NIF",NIF2.Text.Trim());
            sqlCmd.Parameters.AddWithValue("@Nome", Nome2.Text.Trim());
            sqlCmd.Parameters.AddWithValue("@Apelido", Apelido2.Text.Trim());
            sqlCmd.Parameters.AddWithValue("@Idade", Idade.Text.Trim());
            sqlCmd.Parameters.AddWithValue("@Telemovel", Tel.Text.Trim());
            sqlCmd.Parameters.AddWithValue("@IDGenero", IDG.Trim());
            sqlCmd.ExecuteNonQuery();
            MessageBox.Show("Successfull Registration");
        }
        this.Close();
        Form2 form = new Form2();
        form.Show();}}
```

Registo

```
static private SqlConnection sqlcon = new SqlConnection(@"Data
Source=(localdb)\MSSQLLocalDB;Initial Catalog=Projeto;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Mul
tiSubnetFailover=False");
private SqlCommand oCmd;
private string query;
private void update(){
    sqlcon.Open();
    query = "Select Paciente.NIF, Paciente.Nome,Paciente.Apelido,Genero.Genero1,
Paciente.Idade, Paciente.Telemovel from Paciente inner join Genero on
Paciente.IdGenero=Genero.IdGenero";
    main_list.Items.Clear();
    oCmd = new SqlCommand(query, sqlcon);
    using (SqlDataReader reader = oCmd.ExecuteReader()){
        while (reader.Read()){
            ListViewItem item = new ListViewItem(reader.GetInt32(0).ToString());
            item.SubItems.Add(reader.GetString(1).ToString());
            item.SubItems.Add(reader.GetString(2).ToString());
            item.SubItems.Add(reader.GetString(3).ToString());
            item.SubItems.Add(reader.GetInt32(4).ToString());
```

```

        item.SubItems.Add(reader.GetInt32(5).ToString());
        main_list.Items.Add(item);}}
    sqlcon.Close();
    oCmd.Dispose();}
private void list_main(){
    sqlcon.Open();
    query = "Select Paciente.NIF, Paciente.Nome,Paciente.Apelido,Genero.Genero1,
Paciente.Idade, Paciente.Telemovei from Paciente inner join Genero on
Paciente.IdGenero=Genero.IdGenero where NIF='" + search_nif.Text.Trim() + "'";
    main_list.Items.Clear();
    oCmd = new SqlCommand(query, sqlcon);
    using (SqlDataReader reader = oCmd.ExecuteReader()){
        while (reader.Read()){
            ListViewItem item = new ListViewItem(reader.GetInt32(0).ToString());
            item.SubItems.Add(reader.GetString(1).ToString());
            item.SubItems.Add(reader.GetString(2).ToString());
            item.SubItems.Add(reader.GetString(3).ToString());
            item.SubItems.Add(reader.GetInt32(4).ToString());
            item.SubItems.Add(reader.GetInt32(5).ToString());
            main_list.Items.Add(item);}}
            sqlcon.Close();
            oCmd.Dispose();}
private void list_efci(){
    sqlcon.Open();
    query = "Select Anomalia.Num_paci,
Doe_les.Nome,Anomalia.efci1,Anomalia.efci2,Anomalia.efci3,Anomalia.efci4 from
Anomalia inner join Doe_les on Anomalia.Id_doe_le=Doe_les.Id_doe_les where NIF='" +
search_nif.Text.Trim() + "'";
    oCmd = new SqlCommand(query, sqlcon);
    using (SqlDataReader reader = oCmd.ExecuteReader()){
        while (reader.Read()){
            ListViewItem item = new ListViewItem(reader.GetInt32(0).ToString());
            item.SubItems.Add(reader.GetString(1).ToString());
            item.SubItems.Add(reader.GetDecimal(2).ToString());
            item.SubItems.Add(reader.GetDecimal(3).ToString());
            item.SubItems.Add(reader.GetDecimal(4).ToString());
            item.SubItems.Add(reader.GetDecimal(5).ToString());
            efci_list.Items.Add(item);}}
        sqlcon.Close();
        oCmd.Dispose();}
private void list_efca(){
    sqlcon.Open();
    query = "Select Anomalia.Num_paci,
Doe_les.Nome,Anomalia.efca1,Anomalia.efca2,Anomalia.efca3,Anomalia.efca4 from
Anomalia inner join Doe_les on Anomalia.Id_doe_le=Doe_les.Id_doe_les where NIF='" +
search_nif.Text.Trim() + "'";
    oCmd = new SqlCommand(query, sqlcon);
    efca_list.Items.Clear();
    using (SqlDataReader reader = oCmd.ExecuteReader()){
        while (reader.Read()){
            ListViewItem item = new ListViewItem(reader.GetInt32(0).ToString());
            item.SubItems.Add(reader.GetString(1).ToString());
            item.SubItems.Add(reader.GetDecimal(2).ToString());
            item.SubItems.Add(reader.GetDecimal(3).ToString());
            item.SubItems.Add(reader.GetDecimal(4).ToString());
            item.SubItems.Add(reader.GetDecimal(5).ToString());
            efca_list.Items.Add(item);}}

```

```
sqlcon.Close();
oCmd.Dispose();}
private void delete_np(){
sqlcon.Open();
query = "Delete from Anomalia where Num_paci='" + np_delete.Text.Trim() + "'";
efci_list.Items.Clear();
oCmd = new SqlCommand(query, sqlcon);
oCmd.ExecuteNonQuery();
sqlcon.Close();
oCmd.Dispose();}
```

Algoritmo para o processamento de imagens

```
private void roda_c_Lock
{Bitmap ima0 = new Bitmap(sec_pic.ImageLocation);
Bitmap ima1 = new Bitmap(third_pic.ImageLocation);
Bitmap ima2 = new Bitmap(fourth_pic.ImageLocation);
Bitmap ima3 = new Bitmap(fifth_pic.ImageLocation);
Bitmap ima_orig = new Bitmap(orig_pic.ImageLocation);

int i = 0;
principal rd = new principal();
auxiliar aux = new auxiliar();

memoria_0 memo_0 = new memoria_0();
memo_0.a_efci = new float[5];
memo_0.a_efca = new float[5];
memo_0.a_ang = new float[5];
memoria_1 memo_1 = new memoria_1();
memo_1.a_efci = new float[5];
memo_1.a_efca = new float[5];
memo_1.a_ang = new float[5];
memoria_2 memo_2 = new memoria_2();
memo_2.a_efci = new float[5];
memo_2.a_efca = new float[5];
memo_2.a_ang = new float[5];
memoria_3 memo_3 = new memoria_3();
memo_3.a_efci = new float[5];
memo_3.a_efca = new float[5];
memo_3.a_ang = new float[5];
Bitmap[] scal=mestre(ima_orig, ima0, ima1, ima2, ima3);

ima0 = scal[0];
ima1 = scal[1];
ima2 = scal[2];
ima3 = scal[3];

rd.roda_LB(ima_orig, 0);
orig_pic.SizeMode = PictureBoxSizeMode.StretchImage;
orig_pic.Image = ima_orig;

for (float a = -90; a <= 90; a += 45) {
rd.roda_LB(ima0, a);
rd.roda_LB(ima1, a);
rd.roda_LB(ima2, a);
rd.roda_LB(ima3, a);
```

```

rd.pinta_LB(ima_orig, ima0, a, memo_0.a_efci, memo_0.a_efca, memo_0.a_ang, i);
rd.pinta_LB(ima_orig, ima1, a, memo_1.a_efci, memo_1.a_efca, memo_1.a_ang, i);
rd.pinta_LB(ima_orig, ima2, a, memo_2.a_efci, memo_2.a_efca, memo_2.a_ang, i);
rd.pinta_LB(ima_orig, ima3, a, memo_3.a_efci, memo_3.a_efca, memo_3.a_ang, i);

        ima0 = scal[0];
        ima1 = scal[1];
        ima2 = scal[2];
        ima3 = scal[3];
        i++;}

rd.roda_LB(ima0, aux.escolha(memo_0.a_efci, memo_0.a_ang));
rd.roda_LB(ima1, aux.escolha(memo_1.a_efci, memo_1.a_ang));
rd.roda_LB(ima2, aux.escolha(memo_2.a_efci, memo_2.a_ang));
rd.roda_LB(ima3, aux.escolha(memo_3.a_efci, memo_3.a_ang));

sec_pic.SizeMode = PictureBoxSizeMode.StretchImage;
third_pic.SizeMode = PictureBoxSizeMode.StretchImage;
fourth_pic.SizeMode = PictureBoxSizeMode.StretchImage;
fifth_pic.SizeMode = PictureBoxSizeMode.StretchImage;

sec_pic.Image = rd.pinta_LB_final(ima_orig, ima0);
third_pic.Image = rd.pinta_LB_final(ima_orig, ima1);
fourth_pic.Image = rd.pinta_LB_final(ima_orig, ima2);
fifth_pic.Image = rd.pinta_LB_final(ima_orig, ima3);

edita_grafico(memo_0.a_efci,          memo_0.a_efca,          memo_1.a_efci,
memo_1.a_efca, memo_2.a_efci, memo_2.a_efca, memo_3.a_efci, memo_3.a_efca);}

private void edita_grafico(float[] efc0, float[] efca0, float[] efc1, float[]
efca1, float[] efc2, float[] efca2, float[] efc3, float[] efca3){
    auxiliar aux = new auxiliar();
    graphic_1.Series["Eficiência"].Points.AddXY("1º", efc0.Max());
    graphic_1.Series["Eficiência"].Points.AddXY("2º", efc1.Max());
    graphic_1.Series["Eficiência"].Points.AddXY("3º", efc2.Max());
    graphic_1.Series["Eficiência"].Points.AddXY("4º", efc3.Max());
    graphic_1.Series["Eficácia"].Points.AddXY("1º", efca0[aux.esc_efica(efc0)]);
    graphic_1.Series["Eficácia"].Points.AddXY("2º",
efca1[aux.esc_efica(efc1)]);
    graphic_1.Series["Eficácia"].Points.AddXY("3º",
efca2[aux.esc_efica(efc2)]);
    graphic_1.Series["Eficácia"].Points.AddXY("4º",
efca3[aux.esc_efica(efc3)]);

    label1.Text= efc0.Max().ToString();
    label2.Text = efc1.Max().ToString();
    label3.Text = efc2.Max().ToString();
    label4.Text = efc3.Max().ToString();
}

public Bitmap[] mestre(Bitmap ima_orig, Bitmap ima0, Bitmap ima1, Bitmap
ima2, Bitmap ima3){
    List<int> orig_x = new List<int>();
    List<int> orig_y = new List<int>();
    Bitmap[] i = new Bitmap[4];
    float[] info_orig = meio_fig(ima_orig, orig_x, orig_y);
    float[] info_0 = meio_fig(ima0, null, null);

```

```
float[] info_1 = meio_fig(ima1, null, null);
float[] info_2 = meio_fig(ima2, null, null);
float[] info_3 = meio_fig(ima3, null, null);

y = {(float)(info_orig[7] / info_1[7])});
    ima0 = scaling((float)(info_orig[3] / info_0[3]), (float)(info_orig[7] /
info_0[7]), ima0);
    ima1 = scaling((float)(info_orig[3] / info_1[3]), (float)(info_orig[7] /
info_1[7]), ima1);
    ima2 = scaling((float)(info_orig[3] / info_2[3]), (float)(info_orig[7] /
info_2[7]), ima2);
    ima3 = scaling((float)(info_orig[3] / info_3[3]), (float)(info_orig[7] /
info_3[7]), ima3);
    orig_x.Clear();
    orig_y.Clear();
    i[0] = ima0;
    i[1] = ima1;
    i[2] = ima2;
    i[3] = ima3;
    return i;}

public Bitmap scaling(float x, float y, Bitmap im){
    Image image = resizeImage((Image)im, (int)(x * 1024), (int)(y * 1024));
    Bitmap branco = new Bitmap(
Bitmap(@"C:\Users\belni\Desktop\Demo\Assets\Scenes\Imagens\Cam\branco.png");
    return paste(branco, (Bitmap)image);}

public static Image resizeImage(Image image, int new_width, int new_height){
Bitmap new_image = new Bitmap(new_width, new_height);
Graphics g = Graphics.FromImage((Image)new_image);
g.InterpolationMode = InterpolationMode.High;
g.DrawImage(image, 0, 0, new_width, new_height);
return new_image;}

public Bitmap paste(Bitmap branco, Bitmap ima){
    List<int> x = new List<int>();
    List<int> y = new List<int>();
    for (int i = 0; i < ima.Width; i++){
        for (int j = 0; j < ima.Height; j++){
            var p = ima.GetPixel(i, j);
            if (((p.R < 250) || (p.G < 250) || (p.B < 250)) && (p.A > 240)){
                x.Add(i);
                y.Add(j);}}
    Graphics g = Graphics.FromImage((Image)branco);
    Rectangle cloneRect = new Rectangle(x.Min(), y.Min(), x.Max() - x.Min(),
y.Max() - y.Min());
    PixelFormat format = ima.PixelFormat;
    Bitmap cloneBitmap = ima.Clone(cloneRect, format);
    g.DrawImage(cloneBitmap, 0, 0);
    x.Clear();
    y.Clear();
    return branco;}

public float[] meio_fig(Bitmap ima, List<int> dim_x, List<int> dim_y){
    List<int> box_y = new List<int>();
    List<int> box_x = new List<int>();
```

```

        int seg, cap = -1;
        seg = -2;
        BitmapData bdata = ima.LockBits(new Rectangle(0, 0, ima.Width,
        ima.Height), ImageLockMode.ReadWrite, ima.PixelFormat);
        int bpp = Bitmap.GetPixelFormatSize(ima.PixelFormat) / 8;
        int bcount = bdata.Stride * ima.Height;
        byte[] pixel = new byte[bcount];
        IntPtr ip = bdata.Scan0;
        Marshal.Copy(ip, pixel, 0, pixel.Length);
        int hp = ima.Height;
        int wp = bpp * ima.Width;

        for (int i = 0; i < hp; i++){
            int cline = i * wp;
            for (int j = 0; j < wp; j += bpp){
                var blue = pixel[cline + j];
                var green = pixel[cline + j + 1];
                var red = pixel[cline + j + 2];
                var alfa = pixel[cline + j + 3];
                if (((blue < 250) || (green < 250) || (red < 250))){
                    if (cap != seg){
                        box_y.Add(cline);
                        seg = box_y.Last();}
                    box_x.Add(j / 4);
                    if (chave == 0){
                        dim_y.Add(cline);
                        dim_x.Add(j / 4);}}
                    cap++;}}
        Marshal.Copy(pixel, 0, ip, pixel.Length);
        ima.UnlockBits(bdata);
        float[] soma = new float[8];
        //-----casa do x
        soma[0] = box_x.Min();
        soma[1] = box_x.Max();
        soma[2] = (int)((box_x.Min() + box_x.Max()) / 2); //centro
        soma[3] = -box_x.Min() + box_x.Max(); //amplitude
        soma[4] = box_y.Min();
        soma[5] = box_y.Max();
        soma[6] = (int)(box_y.Min() + box_y.Max()) / 2;
        soma[7] = -box_y.Min() + box_y.Max();
        chave = 1;
        box_x.Clear();
        box_y.Clear();
        return soma;}}

    public class principal{
        public void roda_LB(Bitmap ima, float ang){
            List<int> x = new List<int>();
            float angulo = (float)(ang * Math.PI) / 180;
            auxiliar meio = new auxiliar();
            meio.armazena(ima, ang, x);
            BitmapData bdata = ima.LockBits(new Rectangle(0, 0, ima.Width,
            ima.Height), ImageLockMode.ReadWrite, ima.PixelFormat);
            int bpp = Bitmap.GetPixelFormatSize(ima.PixelFormat) / 8;
            int bcount = bdata.Stride * ima.Height;
            byte[] pixel = new byte[bcount];
            IntPtr ip = bdata.Scan0;
            Marshal.Copy(ip, pixel, 0, pixel.Length);

```

```
int hp = ima.Height;
int wp = bpp * ima.Width;

for (int i = 0; i < hp; i++){
    int cline = i * wp;
    for (int j = 0; j < wp; j += bpp) {
        var blue = pixel[cline + j];
        var green = pixel[cline + j + 1];
        var red = pixel[cline + j + 2];
        var alfa = pixel[cline + j + 3];
        if ((blue < 200) || (green < 200) || (red < 200)){
            pixel[cline + j]=(byte)255;
            pixel[cline + j + 1] = (byte)255;
            pixel[cline + j + 2] = (byte)255;}}}

foreach(int d in x){
    pixel[d] = (byte)0;
    pixel[d + 1] = (byte)0;
    pixel[d + 2] = (byte)0;}
Marshal.Copy(pixel, 0, ip, pixel.Length);
ima.UnlockBits(bdata);}

public Bitmap pinta_LB(Bitmap ima, Bitmap ima_d1, float ang, float[]
efci,float[] efca,float[] angulo,int i){
    int v_orig = 0, v_doent = 0, v_res = 0;
    BitmapData bitmapData = ima.LockBits(new Rectangle(0, 0, ima.Width,
ima.Height), ImageLockMode.ReadWrite, ima.PixelFormat);

    int bytesPerPixel = Bitmap.GetPixelFormatSize(ima.PixelFormat) / 8;
    int byteCount = bitmapData.Stride * ima.Height;
    byte[] pixels = new byte[byteCount];
    IntPtr ptrFirstPixel = bitmapData.Scan0;
    Marshal.Copy(ptrFirstPixel, pixels, 0, pixels.Length);
    int heightInPixels = bitmapData.Height;
    int widthInBytes = bitmapData.Width * bytesPerPixel;

    BitmapData bitmapData1 = ima_d1.LockBits(new Rectangle(0, 0,
ima_d1.Width, ima_d1.Height), ImageLockMode.ReadWrite, ima_d1.PixelFormat);
    int bytesPerPixel1 = Bitmap.GetPixelFormatSize(ima_d1.PixelFormat) / 8;
    int byteCount1 = bitmapData1.Stride * ima_d1.Height;
    byte[] pixels1 = new byte[byteCount1];
    IntPtr ptrFirstPixel1 = bitmapData1.Scan0;
    Marshal.Copy(ptrFirstPixel1, pixels1, 0, pixels1.Length);
    int heightInPixels1 = bitmapData1.Height;
    int widthInBytes1 = bitmapData1.Width * bytesPerPixel1;
    Bitmap pngfinal = new
Bitmap(@"C:\Users\belni\Desktop\Demo\Assets\Scenes\Imagens\Cam\branco.png");
    BitmapData bitmapData2 = pngfinal.LockBits(new Rectangle(0, 0,
pngfinal.Width, pngfinal.Height), ImageLockMode.ReadWrite, pngfinal.PixelFormat);
    int bytesPerPixel2 = Bitmap.GetPixelFormatSize(pngfinal.PixelFormat) /
8;

    int byteCount2 = bitmapData2.Stride * pngfinal.Height;
    byte[] pixels2 = new byte[byteCount2];
    IntPtr ptrFirstPixel2 = bitmapData2.Scan0;
    Marshal.Copy(ptrFirstPixel2, pixels2, 0, pixels2.Length);
    int heightInPixels2 = bitmapData2.Height;
    int widthInBytes2 = bitmapData2.Width * bytesPerPixel2;
```

```

for (int y = 0; y < heightInPixels; y++){
    int currentLine = y * bitmapData.Stride;
    for (int x = 0; x < widthInBytes; x = x + bytesPerPixel){
        int oldBlue = pixels[currentLine + x];
        int oldGreen = pixels[currentLine + x + 1];
        int oldRed = pixels[currentLine + x + 2];
        int oldAlfa = pixels[currentLine + x + 3];
        if (((oldBlue < 200) || (oldGreen < 200) || (oldRed < 200))){
            pixels2[currentLine + x] = (byte)0;
            pixels2[currentLine + x + 1] = (byte)0;
            pixels2[currentLine + x + 2] = (byte)0;
            v_orig++;}}
for (int y = 0; y < heightInPixels1; y++){
    int currentLine = y * bitmapData1.Stride;
    for (int x = 0; x < widthInBytes1; x = x + bytesPerPixel1){
        int oldBlue = pixels2[currentLine + x];
        int oldGreen = pixels2[currentLine + x + 1];
        int oldRed = pixels2[currentLine + x + 2];
        //Esta pintado preto
        int oldB = pixels1[currentLine + x];
        int oldG = pixels1[currentLine + x + 1];
        int oldR = pixels1[currentLine + x + 2];
        int oldA = pixels1[currentLine + x + 3];
        if (((oldB < 200) || (oldG < 200) || (oldR < 200))){
            v_doent++;
            if ((oldBlue < 200) || (oldGreen < 200) || (oldRed < 200)){
                // calculate new pixel value
                pixels2[currentLine + x] = (byte)0;
                pixels2[currentLine + x + 1] = (byte)255;
                pixels2[currentLine + x + 2] = (byte)255;
                v_res++;}
            else{
                pixels2[currentLine + x] = (byte)0;
                pixels2[currentLine + x + 1] = (byte)0;
                pixels2[currentLine + x + 2] = (byte)200;}}}}
Marshal.Copy(pixels, 0, ptrFirstPixel, pixels.Length);
ima.UnlockBits(bitmapData);
Marshal.Copy(pixels1, 0, ptrFirstPixel1, pixels1.Length);
ima_d1.UnlockBits(bitmapData1);
Marshal.Copy(pixels2, 0, ptrFirstPixel2, pixels2.Length);
pngfinal.UnlockBits(bitmapData2);
auxiliar aux = new auxiliar();
aux.calcula(v_orig, v_doent, v_res, ang, efci, efca, angulo,i
return pngfinal;}

public Bitmap pinta_LB_final(Bitmap ima, Bitmap ima_d1){
    BitmapData bitmapData = ima.LockBits(new Rectangle(0, 0, ima.Width,
ima.Height), ImageLockMode.ReadWrite, ima.PixelFormat);
    int bytesPerPixel = Bitmap.GetPixelFormatSize(ima.PixelFormat) / 8;
    int byteCount = bitmapData.Stride * ima.Height;
    byte[] pixels = new byte[byteCount];
    IntPtr ptrFirstPixel = bitmapData.Scan0;
    Marshal.Copy(ptrFirstPixel, pixels, 0, pixels.Length);
    int heightInPixels = bitmapData.Height;
    int widthInBytes = bitmapData.Width * bytesPerPixel;
    BitmapData bitmapData1 = ima_d1.LockBits(new Rectangle(0, 0,
ima_d1.Width, ima_d1.Height), ImageLockMode.ReadWrite, ima_d1.PixelFormat);
    int bytesPerPixel1 = Bitmap.GetPixelFormatSize(ima_d1.PixelFormat) / 8;

```

```
int byteCount1 = bitmapData1.Stride * ima_d1.Height;
byte[] pixels1 = new byte[byteCount1];
IntPtr ptrFirstPixel1 = bitmapData1.Scan0;
Marshal.Copy(ptrFirstPixel1, pixels1, 0, pixels1.Length);
int heightInPixels1 = bitmapData1.Height;
int widthInBytes1 = bitmapData1.Width * bytesPerPixel1;
Bitmap pngfinal = new
Bitmap(@"C:\Users\belni\Desktop\Demo\Assets\Scenes\Imagens\Cam\branco.png");
BitmapData bitmapData2 = pngfinal.LockBits(new Rectangle(0, 0,
pngfinal.Width, pngfinal.Height), ImageLockMode.ReadWrite, pngfinal.PixelFormat);
int bytesPerPixel2 = Bitmap.GetPixelFormatSize(pngfinal.PixelFormat) /
8;

int byteCount2 = bitmapData2.Stride * pngfinal.Height;
byte[] pixels2 = new byte[byteCount2];
IntPtr ptrFirstPixel2 = bitmapData2.Scan0;
Marshal.Copy(ptrFirstPixel2, pixels2, 0, pixels2.Length);
int heightInPixels2 = bitmapData2.Height;
int widthInBytes2 = bitmapData2.Width * bytesPerPixel2;
for (int y = 0; y < heightInPixels; y++){
    int currentLine = y * bitmapData.Stride;
    for (int x = 0; x < widthInBytes; x = x + bytesPerPixel){
        int oldBlue = pixels[currentLine + x];
        int oldGreen = pixels[currentLine + x + 1];
        int oldRed = pixels[currentLine + x + 2];
        int oldAlfa = pixels[currentLine + x + 3];
        if (((oldBlue < 200) || (oldGreen < 200) || (oldRed < 200))){
            pixels2[currentLine + x] = (byte)0;
            pixels2[currentLine + x + 1] = (byte)0;
            pixels2[currentLine + x + 2] = (byte)0;}}}
for (int y = 0; y < heightInPixels1; y++){
    int currentLine = y * bitmapData1.Stride;
    for (int x = 0; x < widthInBytes1; x = x + bytesPerPixel1){
        int oldBlue = pixels2[currentLine + x];
        int oldGreen = pixels2[currentLine + x + 1];
        int oldRed = pixels2[currentLine + x + 2];
        int oldB = pixels1[currentLine + x];
        int oldG = pixels1[currentLine + x + 1];
        int oldR = pixels1[currentLine + x + 2];
        int oldA = pixels1[currentLine + x + 3];
        if (((oldB < 200) || (oldG < 200) || (oldR < 200))){
            if ((oldBlue < 200) || (oldGreen < 200) || (oldRed < 200)){
                pixels2[currentLine + x] = (byte)0;
                pixels2[currentLine + x + 1] = (byte)255;
                pixels2[currentLine + x + 2] = (byte)255;}
            else{
                pixels2[currentLine + x] = (byte)0;
                pixels2[currentLine + x + 1] = (byte)0;
                pixels2[currentLine + x + 2] = (byte)200;}}}}}
Marshal.Copy(pixels, 0, ptrFirstPixel, pixels.Length);
ima.UnlockBits(bitmapData);
Marshal.Copy(pixels1, 0, ptrFirstPixel1, pixels1.Length);
ima_d1.UnlockBits(bitmapData1);
Marshal.Copy(pixels2, 0, ptrFirstPixel2, pixels2.Length);
pngfinal.UnlockBits(bitmapData2);
return pngfinal;}}
public class auxiliar{
```

```

static int c, h, v=0;
public int[] p_meio_fig(Bitmap ima){
    List<int> box_b = new List<int>();
    List<int> box_x = new List<int>();
    int cap = 0;
    //PixelFormat can be "Format24bpp", "Format32bpp" ... etc
    BitmapData bdata = ima.LockBits(new Rectangle(0, 0, ima.Width,
ima.Height), ImageLockMode.ReadWrite, ima.PixelFormat);
    int bpp = Bitmap.GetPixelFormatSize(ima.PixelFormat) / 8;//32bpp/8=4
    int bcount = bdata.Stride * ima.Height;//4*1024*1024
    byte[] pixel = new byte[bcount];
    IntPtr ip = bdata.Scan0;//the first line to scan on the image
    Marshal.Copy(ip, pixel, 0, pixel.Length);
    int hp = ima.Height;
    int wp = bpp * ima.Width;
    for (int i = 0; i < hp; i++){
        int cline = i * wp;
        for (int j = 0; j < wp; j += bpp){
            var blue = pixel[cline + j];
            var green = pixel[cline + j + 1];
            var red = pixel[cline + j + 2];
            var alfa = pixel[cline + j + 3];
            if (cap > 1023){
                cap = 0;}
            if (((blue < 200) || (green < 200) || (red < 200))) {
                box_b.Add(cline + j);
                box_x.Add(cap);}
            cap++;}}
    Marshal.Copy(pixel, 0, ip, pixel.Length);
    ima.UnlockBits(bdata);
    int[] soma = new int[2];
    soma[0] = box_b.Min() + box_b.Max();
    soma[0] /= 4;
    soma[0]= (int)(soma[0]/2);
    soma[1] = (int)((box_x.Min() + box_x.Max())/2);
    box_b.Clear();
    box_x.Clear();
    return soma;}

public int[] meio_fig_set_get(Bitmap ima, List<int> disc_x, List<int>
disc_y){
    int[] meio = new int[2];
    List<int> box_y = new List<int>();
    List<int> box_x = new List<int>();
    int veri = 0, verj = 0;
    for (int i = 0; i < ima.Width; i++){
        for (int j = 0; j < ima.Height; j++){
            var cor = ima.GetPixel(i, j);
            if (((cor.B < 200) || (cor.G < 200) || (cor.R < 200))){
                disc_x.Add(i);
                disc_y.Add(j);
                if (veri != i){
                    box_x.Add(i);
                    veri = i;}
                if (veri != j){
                    box_y.Add(j);
                    verj = j;}}}}
    meio[0] = (int)((box_x.Max() + box_x.Min())/2);

```

```
        meio[1] = (int)((box_y.Max() + box_y.Min())/2);
        box_x.Clear();
        box_y.Clear();
        return meio;}

    public void calcula(int v_orig,int v_doent, int v_resul,float ang, float[]
efci, float[] efca, float[] a_ang,int i){
        efci[i] =(float) v_resul / v_doent * 100; //eficiencia
        efca[i] = (float)v_resul / v_orig * 100; //eficácia
        a_ang[i]=ang;
        i++;
        if (i > 4){
            i = 0;}}

    public int escolha(float[] efci,float[] a_ang){
        int b_ang = Array.IndexOf(efci, efci.Max());
        return (int)a_ang[b_ang];}

    public void armazena(Bitmap ima, float ang, List<int> mono){
        int x, y;
        List<int> disc_x=new List<int>();
        List<int> disc_y = new List<int>(); ;
        //posição do meio da figura em x=p[0] e y=p[1]
        int[] pos=meio_fig_set_get(ima, disc_x,disc_y);
        int i = 0;
        foreach (int d in disc_x){
            x = ((int)((d - pos[0]) * Math.Cos(Math.PI * ang / 180) - (disc_y[i]
- pos[1]) * Math.Sin(Math.PI * ang / 180) + ima.Width / 2));
            y = ((int)((d - pos[0]) * Math.Sin(Math.PI * ang / 180) + (disc_y[i]
- pos[1]) * Math.Cos(Math.PI * ang / 180) + ima.Height / 2));
            mono.Add(4 * (y * ima.Width + x));//mono dimensional
            i++;}
        disc_y.Clear();
        disc_x.Clear();}

    public int esc_efica(float[] efci){
        int b_ang = Array.IndexOf(efci, efci.Max());
        return b_ang;}}
```

Produção gráfica

```
private void edita_grafico(float[] efci0, float[] efca0, float[] efci1, float[]
efca1, float[] efci2, float[] efca2, float[] efci3, float[] efca3){
    auxiliar aux = new auxiliar();
    graphic_1.Series["Eficiência"].Points.AddXY("1º", efci0.Max());
    graphic_1.Series["Eficiência"].Points.AddXY("2º", efci1.Max());
    graphic_1.Series["Eficiência"].Points.AddXY("3º", efci2.Max());
    graphic_1.Series["Eficiência"].Points.AddXY("4º", efci3.Max());

    graphic_1.Series["Eficácia"].Points.AddXY("1º",efca0[aux.esc_efica(efci0)]);
    graphic_1.Series["Eficácia"].Points.AddXY("2º",
efca1[aux.esc_efica(efci1)]);
    graphic_1.Series["Eficácia"].Points.AddXY("3º",
efca2[aux.esc_efica(efci2)]);
    graphic_1.Series["Eficácia"].Points.AddXY("4º",
efca3[aux.esc_efica(efci3)]);}}
```


Anexo C – Base de dados

Algoritmo de configuração da base de dados

Paciente

```
CREATE TABLE [dbo].[Paciente] (  
    [NIF] INT NOT NULL,  
    [Nome] VARCHAR (15) NOT NULL,  
    [Apelido] VARCHAR (15) NOT NULL,  
    [Idade] INT NOT NULL,  
    [Telemovel] INT NOT NULL,  
    [IdGenero] INT NOT NULL,  
    PRIMARY KEY CLUSTERED ([NIF] ASC),  
    CONSTRAINT [Genero_Paciente] FOREIGN KEY ([IdGenero]) REFERENCES [dbo].[Genero]  
    ([IdGenero])  
);
```

Gênero

```
CREATE TABLE [dbo].[Genero] (  
    [IdGenero] INT IDENTITY (1, 1) NOT NULL,  
    [Genero1] VARCHAR (13) NOT NULL,  
    PRIMARY KEY CLUSTERED ([IdGenero] ASC)  
);
```

Anomalia

```
CREATE TABLE [dbo].[Anomalia] (  
    [Num_paci] INT IDENTITY (1, 1) NOT NULL,  
    [NIF] INT NOT NULL,  
    [Id_doe_le] INT NOT NULL,  
    [efci1] DECIMAL (5, 2) NOT NULL,  
    [efci2] DECIMAL (5, 2) NOT NULL,  
    [efci3] DECIMAL (5, 2) NOT NULL,  
    [efci4] DECIMAL (5, 2) NOT NULL,  
    [efca1] DECIMAL (5, 2) NOT NULL,  
    [efca2] DECIMAL (5, 2) NOT NULL,  
    [efca3] DECIMAL (5, 2) NOT NULL,  
    [efca4] DECIMAL (5, 2) NOT NULL,  
    PRIMARY KEY CLUSTERED ([Num_paci] ASC),  
    CONSTRAINT [Anomalia_doe_le] FOREIGN KEY ([Id_doe_le]) REFERENCES  
    [dbo].[Doe_les] ([Id_doe_les]),  
    CONSTRAINT [Anomalia_Paciente] FOREIGN KEY ([NIF]) REFERENCES [dbo].[Paciente]  
    ([NIF])  
);
```

Doenças e Lesões

```
CREATE TABLE [dbo].[Doe_les] (  
    [Id_doe_les] INT IDENTITY (1, 1) NOT NULL,  
    [Nome] VARCHAR (15) NOT NULL,  
    PRIMARY KEY CLUSTERED ([Id_doe_les] ASC)  
);
```