



**INSTITUTO POLITÉCNICO DE COIMBRA**

**INSTITUTO SUPERIOR DE ENGENHARIA DE  
COIMBRA**

**Interfaces Homem-Máquina para controlador  
NX100 do Robô HP3L da Motoman**

**Relatório de Projeto para a obtenção do grau de Mestre  
em**

**Engenharia Eletrotécnica**

**Autor**

**Inês da Silva Simões**

**Orientação**

**Doutor João Paulo Morais Ferreira**

**Coimbra, Novembro de 2019**





Mestrado em Engenharia Eletrotécnica

---

# **Interfaces Homem-Máquina para controlador NX100 do Robô HP3L da Motoman**

Relatório de Projeto apresentado para a obtenção do grau de Mestre  
em Engenharia Eletrotécnica – Área de Especialização em  
Automação e Comunicações em Sistemas Industriais

**Autor**  
**Inês da Silva Simões**

**Orientador**  
**Doutor João Paulo Morais Ferreira**  
Professor do Departamento de Engenharia Eletrotécnica  
Instituto Superior de Engenharia de Coimbra

**Coimbra, Novembro de 2019**



## RESUMO

Com a expansão dos robôs, os locais e ambientes industriais alteraram-se. Os robôs industriais conseguem efetuar tarefas pesadas e repetitivas a uma velocidade elevada com maior resistência e força, tarefas outrora realizadas pelo homem e pelas ferramentas manuais. Eles são dotados de programação que os configuram para executar uma ou várias tarefas específicas. Contudo, essa programação apresenta-se limitada, e sem destreza motora e visual igual ao ser humano, os robôs necessitam de ser supervisionados. Assim, adaptar a máquina ao homem para se atingir um objetivo comum é o pensamento num mundo em constante mudança. Erros, mensagem de alertas, reinicializações, paragens por tempo indefinido com necessidade que alguém reponha as definições iniciais ou ainda alguém que faça manualmente a sua tarefa aquando das situações apresentadas são alguns dos problemas encontrados nos robôs. O uso de interfaces através de combinação de diferentes *hardware* e *software* tenta responder a estas necessidades. Interfaces que permitam uma interação amigável e intuitiva, que ultrapassem as diferentes condições do utilizador, do local e do ambiente, que garantam segurança e alternativas a reprogramação e a gastos excessivos são soluções procuradas pela indústria.

Neste sentido o presente projeto resume-se no estudo da interação homem-máquina com o desenvolvimento e avaliação de interfaces para manipular um robô industrial presente no laboratório de automação do Instituto Superior de Engenharia de Coimbra (ISEC), o robô HP3L com controlador NX100 da Motoman, através de comunicação RS232. São descritas quatro interfaces, constituídas por diferentes objetos e componentes tais como o uso de botões e comunicação por cabos até objetos mais atuais como o uso de um telemóvel e comunicação Bluetooth.

A avaliação consiste na aplicação de testes de usabilidade aos alunos do mesmo Instituto para se obter o seu feedback na interação destes com o robô HP3L, a fim de se comparar e tirar ilações sobre os diferentes sistemas em estudo.

**Palavras-chave:** Interface, Interface Homem-Máquina, robô, Motoman, *hardware*, *software*, interação, testes de usabilidade



## ABSTRACT

With an expansion of robots, industrial environments have changed. Industrial robots can perform heavy and repetitive tasks at a high speed with greater resistance and strength whilom made by man and hand tools. They are programmatically configured to perform one or more specifically tasks. However, those programming are limited and without motor and visual ability like human being, robots need to be supervised. Then adapting the machine to the man to achieve a common goal is the thought in a constantly changing world. Errors, alerts message, restarts, indefinite stops with the need for someone to reset the initial settings or even someone who does their task manually are some of the problems encountered in robots. The use of different hardware and software in interfaces tries to respond to these needs. Interfaces that allows a friendly and intuitive interaction that surpass the different conditions of user, place and environment which guarantee security and alternatives to reprogramming and over-spending are solutions sought by the industry.

In this sense the present project is summarized in the study of human-machine interaction with the development and evaluation of interface to manipulate an industrial robot present in the automation laboratory of the ISEC, the Motoman HP3L with NX100 controller with RS232 communication. Four interfaces are described consisting of different objects and components such as the use of buttons and wired communication until objects equipped with more current technologies, such as the use of a mobile phone and Bluetooth communication.

The evaluation consists in the application of usability tests to the students of the same Institute to obtain their feedback in their interaction with the HP3L robot, in order to compare and draw conclusions about the different systems under study.

**Key word:** Interface, *Human Machine Interface*, robot, Motoman, *hardware*, *software*, interaction, usability tests



## AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer ao meu orientador, Doutor João Ferreira, pela oportunidade de desenvolver um projeto na área da robótica e por me orientar, ajudar e aconselhar ao longo da elaboração do mesmo.

Aos meus pais Carlos e Clara por tudo o que me transmitiram ao longo dos anos e pelo apoio incondicional nesta caminhada.

Um especial agradecimento à minha irmã Lara pelo apoio incansável, pela motivação e por ser um exemplo para mim e à minha avó Fernanda.

Agradeço também aos meus amigos e em particular à minha colega e amiga Jéssica Pinto, pelo apoio e companheirismo em todo o mestrado.

Por último, agradeço a todos os alunos e aqueles que, embora não referidos, de alguma maneira contribuíram para o desenvolvimento deste projeto.



## ÍNDICE

1.	Introdução .....	1
1.1	Motivação e Enquadramento .....	2
1.2	Apresentação do Projeto .....	3
1.3	Objetivos do Projeto .....	4
1.4	Organização dos Capítulos do Projeto .....	5
2	Revisão da Literatura .....	6
3	Robô HP3L com controlador NX100 da Motoman .....	12
3.1	Descrição .....	12
3.2	Sistema de Coordenadas .....	13
3.3	Cinemática Direta .....	15
3.4	Arquivos de trabalho para comunicação RS232 .....	18
3.5	Consola do Robô HP3L com controlador NX100 da Motoman .....	21
4	Interfaces .....	24
4.1	Interface Cubo.....	24
4.1.1	Arduíno Uno.....	25
4.1.2	<i>Hardware</i> .....	27
4.1.3	<i>Software</i> .....	29
4.1.4	Modo de utilização .....	35
4.2	Interface Telemóvel .....	37
4.2.1	<i>Hardware</i> .....	38
4.2.2	<i>Software</i> .....	39
4.2.3	Modo de utilização .....	41
4.3	Interface Joystick .....	42
4.3.1	<i>Hardware</i> .....	43
4.3.2	<i>Software</i> .....	45
4.3.3	Modo de utilização .....	46
5	Testes de Usabilidade e Resultados .....	48
5.1	Teste de Usabilidade .....	48
5.2	Resultados dos Testes de Usabilidade.....	50
6	Conclusões e trabalho futuro .....	53
	Bibliografia.....	55
	Anexos.....	57



## ÍNDICE DE FIGURAS

Figura 1.1 - Ilustração do conceito interação homem-máquina.....	1
Figura 1.2 - Esquema das interfaces homem-máquina descritas para o Robô HP3L da Motoman com controlador NX100.....	3
Figura 2.1 - Robô <i>Unimate</i> [7] .....	6
Figura 2.2 - Robô instalado na indústria cerâmica. [8].....	6
Figura 2.3 - Robô <i>bin-picking</i> da YASKAWA. [9].....	7
Figura 2.4 - Dispositivo háptico. [10].....	8
Figura 2.5 - Gráfico do tipo de programação utilizado nos robôs industriais. [11] .....	8
Figura 2.6 - Gráfico da distribuição por tipo de colaboradores na utilização de robôs na indústria.[11] .....	8
Figura 2.7 - Consola de programação do Motoman DX100. [12] .....	9
Figura 2.8 - <i>Flexpendant</i> do controlador IRC5 <i>Compact</i> da ABB. [13].....	10
Figura 2.9 - Interface homem-máquina através de realidade virtual [16].....	11
Figura 3.1 - Robô HP3L com controlador NX100 da Motoman presente no Laboratório de Automação do ISEC. ....	13
Figura 3.2 - Nome das estruturas e sistema de eixos <i>Standard Job</i> . [18] .....	13
Figura 3.3 - Utilização de cada sistema de eixos de trabalho. [20].....	14
Figura 3.4 - Sistema de coordenadas cartesianas de referência. [20] .....	14
Figura 3.5 - Disposição dos eixos nas articulações do robô HP3L.....	16
Figura 3.6 - Simulação do robô HP3L da Motoman no <i>Software Motosim</i> .....	18
Figura 3.7 - Mensagem da posição inicial do robô, em milésimas de milímetro e centésimas de grau. ....	19
Figura 3.8 - Esquema do processamento de uma mensagem enviada pelas interfaces, em programação <i>IMFORM</i> . ....	20
Figura 3.9 - <i>Layout</i> gráfico da consola. ....	21
Figura 3.10 - Consola do controlador NX100 para o robô da Motoman com indicação das teclas referidas. ....	22
Figura 3.11 - Utilizador a usar a consola.....	23
Figura 4.1 - Interação entre o utilizador e o robô com a interface cubo. ....	24
Figura 4.2 - <i>Layout</i> da interface cubo.....	25
Figura 4.3 - <i>Layout</i> da mensagem enviada para o robô da interface cubo.....	25
Figura 4.4 - Arduino Uno. ....	26
Figura 4.5 - Arquitetura interno do Arduino Uno. [25] .....	26
Figura 4.6 - Projeto da estrutura da interface cubo com as respetivas dimensões. ....	27
Figura 4.7 - <i>Hardware</i> da interface cubo.....	27
Figura 4.8 - Sistema de coordenadas cartesianas correspondente a cada botão.....	28
Figura 4.9 - Correspondência de cada botão à entrada/saída digital ou analógica do Arduino. ....	28
Figura 4.10 - Esquema elétrico do <i>hardware</i> da interface cubo. ....	29
Figura 4.11 - Fluxograma algoritmo velocidade. ....	30
Figura 4.12 - Fluxograma do algoritmo translação.....	31
Figura 4.13 - Fluxograma do algoritmo rotação. ....	32
Figura 4.14 - Algoritmo garra em linguagem corrente. ....	33
Figura 4.15 - Função de escrita na memória <i>EEPROM</i> .....	34
Figura 4.16 - Função de leitura na memória <i>EEPROM</i> .....	35
Figura 4.17 - a) Parte frontal da interface Cubo, b) Parte Traseira da interface Cubo, c) Parte Direita da interface Cubo, e d) Parte Esquerda da interface Cubo. ....	36
Figura 4.18 - Interação entre o utilizador e o robô com a interface telemóvel. ....	37
Figura 4.19 - <i>Layout</i> da interface telemóvel.....	38
Figura 4.20 - <i>Layout</i> da mensagem enviada para o robô da interface telemóvel.....	38
Figura 4.21 - <i>Hardware</i> da interface telemóvel.....	38
Figura 4.22 - Disposição dos eixos no acelerómetro do telemóvel. ....	39

Figura 4.23 - <i>Layout do App Inventor Designer</i> .....	40
Figura 4.24 - <i>Layout do App Inventor Blocks Editor</i> .....	40
Figura 4.25 - Interface gráfica da aplicação Motoman .....	41
Figura 4.26 - Interação entre o utilizador e o robô com a interface joystick .....	42
Figura 4.27 - <i>Layout</i> da interface joystick .....	43
Figura 4.28 - Mensagem enviada pela interface joystick .....	43
Figura 4.29 - <i>Hardware</i> utilizado para a interface joystick .....	44
Figura 4.30 - Caracterização da função dos seus manipuladores e botões do joystick .....	44
Figura 4.31 - Caracterização da função dos botões do joystick .....	45
Figura 4.32 - Sistema de coordenadas cartesianas no joystick .....	45
Figura 4.33 - GUI da interface joystick .....	46
Figura 5.1 - Modelo <i>Usefulness, Satisfaction and Ease of use</i> . [29] .....	49
Figura 5.2 - Gráfico dos Resultados dos Testes de Usabilidade por média de cada pergunta .....	50
Figura 5.3 - Gráfico dos Resultados dos Testes de Usabilidade por média de parâmetros .....	52



## ÍNDICE DE TABELAS

Tabela 1.1- Cronograma do plano de trabalho do presente projeto. ....	5
Tabela 3.1– Valores dos parâmetros <i>Denavit-Hartenberg</i> retirados dos eixos. ....	16
Tabela 3.2 – Valores dos parâmetros de X e Y . ....	17



## **SIMBOLOGIA E ABREVIATURAS**

*ARM - Advanced Robot Motion*

*BPS – Bits por segundo*

*EEPROM - Electrically-Erasable Programmable Read-Only Memory*

*HMI - Human Machine Interface*

*GUI - Graphical User Interface*

*ISEC - Instituto Superior de Engenharia de Coimbra*

*MIT - Massachusetts Institute of Technology*

*PWM - Pulse Width Modulation*

*SO - Sistema Operativo*

*UART - Universal Asynchronous Receiver/Transmitter*

*USE - Usefulness, Satisfaction and Ease of Use*





## 1. Introdução

Com o desenvolvimento das indústrias e a constante procura do avanço tecnológico, a interação homem-máquina está constantemente a ser aprimorado. [1] O aumento da capacidade de execução de tarefas de forma autónoma e com grande rapidez, obriga a pensar nas interações que os humanos têm com as máquinas. [2]

A acentuada robotização industrial levou a uma readaptação e criação de estratégias de integração do operador no mundo automático. Assim, o conceito de interface começou a ganhar força e hoje em dia são procuradas maneiras inovadoras de se estabelecer e melhorar esta relação. Os robôs mesmo executando a maior parte das tarefas industriais, necessitam de uma supervisão e ajuda humana.

A ISO 9241-110 define interface como "todas as partes de um sistema interativo (de software ou hardware) que fornecem informações e controlos necessários para que o utilizador realize uma determinada tarefa com o sistema interativo." [3] Como exemplos de *hardwares* conta-se com os botões, os teclados, os joysticks, as alavancas, os sensores, os sistemas de visão e os sistemas de movimento corporal. Como exemplos de *softwares* inclui-se programação em Labview, Matlab, C++, Java, Python, entre outros. Para interligação e comunicação dos mesmos com o sistema interativo conta-se com protocolos de comunicação Bluetooth, RS232, Modbus, PROFINET, Ethernet.

A aplicação de interfaces, possibilita ao homem familiarizar-se com a máquina e aprender a manipula-la, a parametriza-la, a controla-la, e a responder rapidamente às suas necessidades, comunicando-se. Ocorre assim, uma aprendizagem e uma troca de informação, que se inicia quando o homem ativa o mecanismo que permite efetuar um movimento, uma tarefa ou uma ação na máquina. A simplicidade de utilização, aplicação, aprendizagem, satisfação e um nível de segurança para o operador são requisitos para o sucesso de interfaces, e conseqüentemente para a indústria. Saber a opinião dos utilizadores aquando da manipulação de diversas tecnologias, possibilita encontrar o *hardware* e o *software* ideal para uma maximização da interação *HMI*. A figura 1.1 ilustra o conceito de interação homem-máquina.

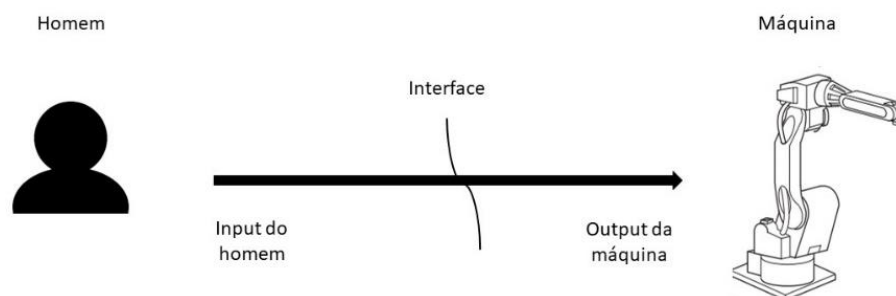


Figura 1.1 - Ilustração do conceito interação homem-máquina

## 1.1 Motivação e Enquadramento

O presente projeto para obtenção do grau de mestre em Engenharia Eletrotécnica ministrado no Instituto Superior de Engenharia de Coimbra (ISEC), tenciona interagir com a área da robótica mais especificamente com os robôs industriais.

A área da robótica encontra-se numa fase de inovação explosiva o que resultará na produção de robôs para qualquer tarefa comercial, industrial e de consumo. A disponibilidade de módulos acopláveis e padronizados de *software* e *hardware* com fácil adaptação a novas aplicações está na origem da referida explosão. [4] Numa indústria, como homem e robô coexistem e cooperam no mesmo espaço e tempo, é relevante questionar como se relacionam, como se comunicam, como se entendem e que soluções existem para que esta relação seja harmoniosa, segura e eficaz, visto que é reconhecido que as capacidades humanas aliadas aos robôs trazem vantagens na produção como, otimização de trabalho, melhoria da qualidade e melhor relação custo-eficácia. [5] Não obstante, a indústria do futuro tenciona que pessoas do chão da fábrica criem as suas próprias interfaces para desenvolverem produtos mais personalizados e específicos consoante as exigências dos clientes. [6]

O presente projeto pretende ir ao encontro destas investigações e tirar conclusões sobre o comportamento e as preferências do homem na manipulação de objetos para interagir com os robôs.

## 1.2 Apresentação do Projeto

O presente projeto, como referido nos pontos anteriores, aborda interfaces entre o homem e a máquina.

Nesta interação, homem é qualquer utilizador que manipule a máquina existente no Laboratório de Automação do Instituto Superior de Engenharia de Coimbra (ISEC), o robô HP3L da Motoman com controlador NX100. O robô, incorpora uma consola para interface entre este e o utilizador. A consola é o padrão para as restantes interfaces: a interface cubo, a interface telemóvel e a interface joystick. Cada uma é composta por um objeto diferente com combinações distintas de *hardware* e *software*.

A interface cubo, é um objeto em forma de cubo, com botões incorporados nas suas faces e prolongamento. Cada botão, ou combinação de botões, é responsável por efetuar uma determinada ação. A informação é processada pela plataforma eletrónica Arduíno e enviada para o robô por comunicação série RS232 ou por comunicação Bluetooth. A interface telemóvel apresenta-se como uma aplicação para sistema operativo Android, que tira partido do acelerómetro e da comunicação Bluetooth do telemóvel. A interface joystick, como o nome indica, permite a monitorização do robô através de um joystick e dos seus botões. Disponibiliza também uma interação virtual do robô. A comunicação deste sistema pode ser feita através de comunicação série RS232 ou comunicação Bluetooth.

A figura 1.2 apresenta os objetos que servirão de interfaces homem-máquina para controlo do referido robô.

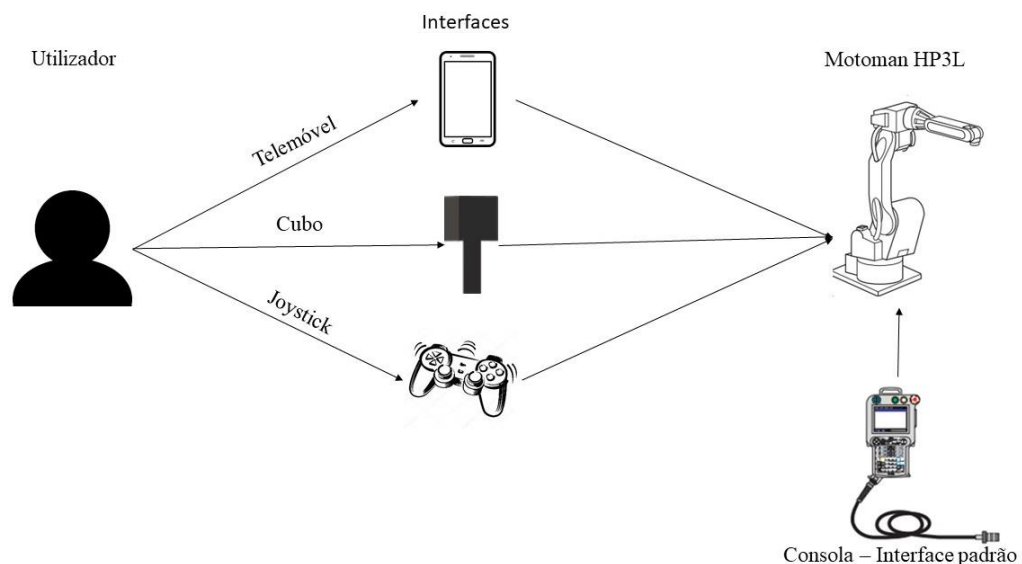


Figura 1.2 - Esquema das interfaces homem-máquina descritas para o Robô HP3L da Motoman com controlador NX100.

As interfaces mencionadas são sujeitas, através da aplicação de testes de usabilidade, à avaliação por parte de vários utilizadores. Os testes de usabilidade são testes em forma de questionário, com escala quantificada, que avaliam parâmetros de performance de um produto ou objeto.

### 1.3 Objetivos do Projeto

Vários objetivos foram delineados para o presente projeto, além do objetivo final que assenta na aplicação dos testes de usabilidade para retirar conclusões das interfaces, está também presente o objetivo de combinar *hardwares* e *softwares* para desenvolver objetos que realizem algumas funções da interface padrão, a consola. Todas as interfaces, e face à consola, devem: permitir a translação ou rotação do robô, fechar ou abrir a sua garra, gravar posições para criar trajetórias, reproduzi-las e variar a velocidade dos seus movimentos.

Para se alcançar todos os objetivos mencionados, considerou-se as diferentes etapas:

#### Etapa 1

Estudo e familiarização de conceitos: interface, utilizador, robô; Revisão de literatura de interfaces homem-máquina.

#### Etapa 2

Compreensão do funcionamento do robô HP3L da Motoman com controlador NX100; Utilização e compreensão da programação utilizada nos robôs Motoman; Utilização da consola *HMI*.

#### Etapa 3

Desenvolvimento da interface cubo.

#### Etapa 4

Desenvolvimento da interface telemóvel.

#### Etapa 5

Adaptação da interface joystick.

#### Etapa 6

Estudo de testes de usabilidade; Aplicação dos testes de usabilidade a diferentes turmas; Interpretação e conclusões dos resultados dos testes.

#### Etapa 7

Escrita de relatório e conclusões.

O cronograma abaixo pretende temporizar as etapas descritas.

Tabela 1.1- Cronograma do plano de trabalho do presente projeto.

Mês	Mar.18	Abril.18	Mai.18	Jun.18	Jul.18	Set.18	Out.18	Nov.18	Dez.18	Jan.19	Fev.19	Mar.19	Abril.19	Mai.19
Etapa 1	x	x	x											
Etapa 2			x	x	x									
Etapa 3				x	x	x	x							
Etapa 4								x	x	x				
Etapa 5								x	x	x	x			
Etapa 6												x	x	
Etapa 7										x	x	x	x	x

#### 1.4 Organização dos Capítulos do Projeto

O projeto estrutura-se em seis capítulos. O primeiro capítulo apresenta e enquadra o tema e os conceitos, caracteriza os objetivos e transmite ao leitor a visão geral do projeto.

Prossegue com o segundo capítulo que apresenta uma revisão de literatura, onde se expõem interfaces praticadas na indústria, interfaces em fase de estudo ou em desenvolvimento.

Seguidamente, o terceiro capítulo, compreende o robô Motoman HP3L, o controlador NX100 e a sua interface, a consola. Neste capítulo, apresentam-se os seus modos de funcionamento, as suas características e explicam-se a programação envolvida.

O quarto capítulo contém a parte mais técnica e prática do projeto. É neste capítulo que se exibem os processos de desenvolvimento das três interfaces (cubo, telemóvel e joystick): o design e a construção do *hardware*, os algoritmos do *software* e o modo de utilização de cada uma.

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

Por último, o sexto capítulo, apresenta as conclusões e os trabalhos futuros, sugerindo também algumas melhorias. São ainda encontrados nas secções seguintes, as referências bibliográficas e o anexo. A programação desenvolvida no Matlab, no Arduino e no Android, encontram-se num documento anexo a este relatório de projeto.

## 2 Revisão da Literatura

Neste capítulo são apresentadas algumas interfaces existentes e outras em estado de desenvolvimento na robótica industrial.

A robótica industrial surgiu na década de 50, com a entrada do primeiro robô na produção industrial numa fábrica da General Motors, o *Unimate*, representado na figura 2.1. O *Unimate* tinha como função levantar peças de metal quentes de uma máquina de fundição e embalá-las, tarefa bastante perigosa para os trabalhadores. Este robô abriu caminho à revolução dos robôs na indústria. [7]



Figura 2.1 - Robô *Unimate* [7]

Numa indústria cerâmica, os operadores têm o trabalho árduo e monótono de colocar peças de cerâmica de elevado peso nos tapetes. Aliado a isso, cometem-se erros de posicionamento e de seleção no tipo de peças. Para auxiliar e eliminar erros, foi desenvolvido e implementado um robô industrial. O robô tem um sistema de segurança que impede que os operadores entrem no seu espaço sem autorização protegendo-os, sendo a interface efetuada a partir de um painel de comando, onde o operador seleciona o modelo de peças em operação, o tipo de palete, verifica a produção, visualiza avisos e erros, entre outros. A juntar a esta interface, o operador tem à sua disposição um joystick para conduzir o robô de forma manual. [8] A figura 2.2 representa o robô implementado na referida indústria.



Figura 2.2 - Robô instalado na indústria cerâmica. [8]

Numa linha de montagem, um operador tem a função de selecionar, escolher, separar e pegar peças com diferentes dimensões e geometrias para colocar num tapete em constante movimento. As peças encontram-se frequentemente amontoadas em contentores, colocadas arbitrariamente, oclusas em parte, ou metidas nos espaços de outras peças, apresentando um problema de perceção e atuação para o operador que se pode atrasar a colocar a peça pretendida no tapete. De forma a melhorar uma boa seleção de peças e um trabalho menos repetitivo e mais descontraído para o operador, surgiu o conceito de *bin-picking*.

O *bin-picking* consiste na utilização de sensores e sistemas de visão, para perceção e reconstrução 3D ou 2D de objetos de modo a alterar o movimento do robô para recolher os objetos em causa, um exemplo de um robô *bin-picking* encontra-se representado na figura 2.3. Contudo, pode haver resistência à extração do objeto ou podem ser recolhidos objetos interligados. Neste caso, o robô necessita de reavaliar os procedimentos, e ter mesmo que parar e reiniciar o processo. Surge assim a necessidade de os robôs “sentirem” os objetos manipulados, monitorizando as suas forças e os seus momentos impostos, abrindo portas para a área da háptica. A área da háptica tira partido dos sentidos do operador humano permitindo grande interação entre eles e os robôs. Um dispositivo háptico deteta o tato, o toque e a força dos utilizadores podendo também retribuir ao mesmo sensações da manipulação dos objetos. Este tipo de interface afirma-se como uma solução para que qualquer pessoa que não saiba programar consiga por demonstração através de dispositivos hápticos ensinar os robôs a executar tarefas. Um dispositivo háptico está representado na figura 2.4. [8]



Figura 2.3 - Robô *bin-picking* da YASKAWA. [9]



Figura 2.4 - Dispositivo háptico. [10]

Nos finais da década de 90, nos Estados Unidos, um estudo divulgou o tipo de programação mais utilizada nos robôs industriais. A figura 2.5 apresenta o gráfico com os resultados desse estudo, em que consta que cerca de 63% da programação de robôs é feita com recurso às consolas de programação, seguido da programação *offline* com 25%. [11]

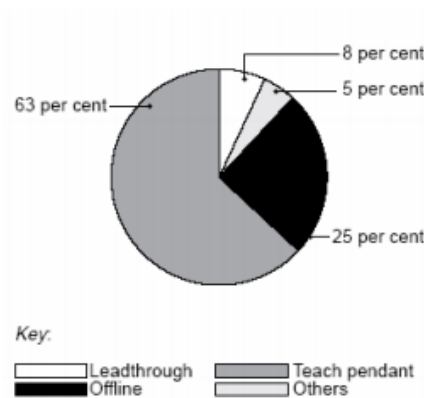


Figura 2.5 - Gráfico do tipo de programação utilizado nos robôs industriais. [11]

Outro estudo paralelo, permitiu saber qual a distribuição na utilização de robôs pelos colaboradores numa indústria. A figura 2.6 apresenta o gráfico com os resultados do estudo, onde consta que 11% da manipulação dos robôs é feita por programadores, 22% por técnicos e 24% por operadores no chão da fabrica. [11]

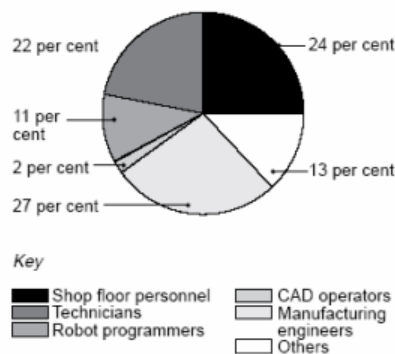


Figura 2.6 - Gráfico da distribuição por tipo de colaboradores na utilização de robôs na indústria.[11]

Constata-se assim, que a maior percentagem de manipulação dos robôs pertence aos operadores no chão da fábrica e a técnicos, que face à percentagem por parte dos programadores, não estão habilitados a programar os robôs ou não tem noções das suas linguagens de programação. A manipulação dos robôs ainda depende muito dos programadores. A interface entre os operadores e as máquinas ocorre maioritariamente pelas consolas de programação.

As consolas de programação tradicionais apresentam-se como consolas de grandes dimensões, ligeiramente pesadas, não ergonómicas, com inúmeras teclas alfanuméricas, pequenos ecrãs táteis, pouco intuitivas e que criam barreiras com o operador no chão da fábrica. A figura 2.7 apresenta uma consola de programação desse tipo, a consola do Motoman DX100. [12]



Figura 2.7 - Consola de programação do Motoman DX100. [12]

Atualmente, para responder aos aspetos negativos destas consolas, os fabricantes apostam em consolas mais intuitivas, ergonómicas e fáceis de controlar. Exemplo disso é o *Flexpendant* do controlador IRC5 Compact da ABB visualizado na figura 2.8. O *Flexpendant* do controlador IRC5 Compact da ABB apresenta-se como um computador autónomo, que pode ser usado por operadores destros e canhotos, com um ecrã tátil agradável e com integração de um joystick para interação intuitiva. Usufriui de um ambiente poderoso de suporte à customização de aplicações, possibilitando personalizar ecrãs à medida de cada operador. [13]



Figura 2.8 - Flexpendant do controlador IRC5 Compact da ABB. [13]

A programação *offline*, que representa uma grande fatia na programação de robôs, visualizado no estudo anterior, apresenta vantagens em relação à programação online. Contrariamente à segunda programação, a utilização de programação *offline* não implica a paragem da célula robótica e consequentemente a perda de produção e dinheiro, além de que previne colisão dos robôs com outros equipamentos.

A realidade virtual é identificada como uma interface homem-máquina avançada e em crescimento para aplicação nas indústrias. Ela permite simular e criar diversos ambientes realísticos, em 3D, com o objetivo de o participante interagir com eles. A interface pode ser efetuada por vários dispositivos como, um capacete com sensores, óculos de realidade virtual, dispositivos de força, emissores de odores, sensores musculares, ou a combinação deles. O ambiente deve interpretar as ações do homem. [14] A universidade de Coimbra em parceria com a empresa alemã *DOLL Engineering* e utilizando o equipamento *HoloLens* da Microsoft, óculos de realidade aumentada, desenvolveu uma ferramenta robótica que permite que o utilizador visualize, manipule e explore, em ambiente real, uma peça em 3D, com as mãos, transmitindo todas as informações ao robô que seguidamente as executa. Este sistema visa ir ao encontro de novas interações homem-máquina para que qualquer pessoa ou técnico manuseie-o e programe-o sem saber nada sobre ele. A figura 2.9 representa a interface mencionada com recurso à realidade virtual. [15]

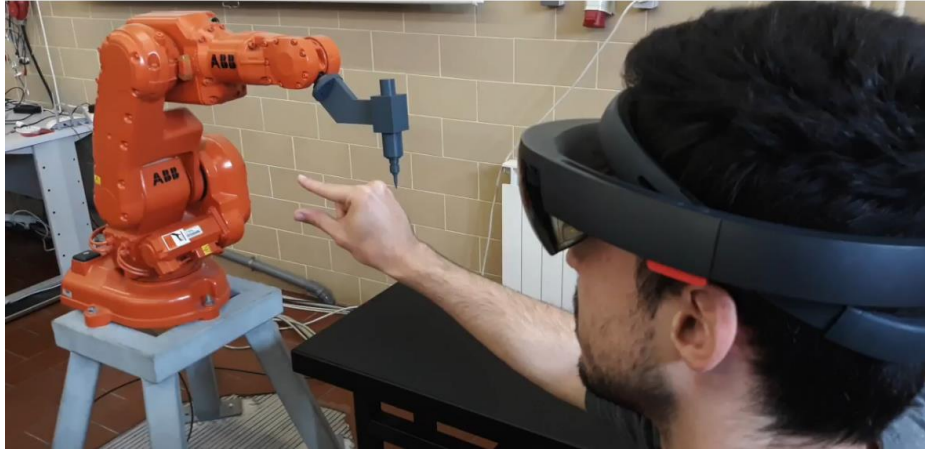


Figura 2.9 - Interface homem-máquina através de realidade virtual [16]

Além da realidade virtual, as interfaces basear-se-ão em dispositivos móveis, como por exemplo, um telemóvel, na captação e interpretação de grandes quantidades de dados em nuvem e inteligência artificial. [16]

### 3 Robô HP3L com controlador NX100 da Motoman

O presente capítulo é dedicado à caracterização do robô HP3L da Motoman com controlador NX100, ao sistema de coordenadas aplicado, à cinemática direta das articulações do robô e à programação envolvida para comunicação RS232 com as interfaces.

#### 3.1 Descrição

O robô HP3L da Motoman com controlador NX100, é o robô industrial existente, como já referido, no Laboratório de Automação do Instituto Superior de Engenharia de Coimbra (ISEC). O robô é caracterizado por uma elevada precisão aliada a uma alta velocidade de deslocamento, reduzindo os tempos de posicionamento no processo onde é aplicado. É dotado de 6 graus de liberdade, ideal para montagem, distribuição, embalamento e manuseamento de materiais. [17]

O controlador NX100 é um controlador avançado de alto desempenho, que coordena e alimenta o robô, processa os sinais de entrada e saída e os diferentes dados numéricos e sistemas de coordenadas. Fornece também funções lógicas, dados constantes e dados variáveis, programação e memória. O controlador suporta até 60 000 etapas e 10 000 instruções, conseguindo controlar simultaneamente quatro robôs, num total de 36 eixos. Incorpora um controlador *Advanced Robot Motion (ARM)*. Acrescenta ainda uma consola de ecrã tátil a cores e cursor de navegação que reduz em 30% o tempo de aprendizagem do operador. [18] A figura 3.1 mostra o robô HP3L com controlado NX100 do Laboratório de Automação do Instituto Superior de Engenharia de Coimbra (ISEC).



Figura 3.1 - Robô HP3L com controlador NX100 da Motoman presente no Laboratório de Automação do ISEC.

### 3.2 Sistema de Coordenadas

O robô HP3L possui seis eixos individuais disponíveis em dois sistemas de eixos de trabalho, o *Standard Job*, representado na figura 3.2, e o *Relative Job*. O *Standard Job*, atribui a cada posição números de pulsos, por sua vez, o *Relative Job*, representa cada posição num sistema de coordenadas XYZ. [19]

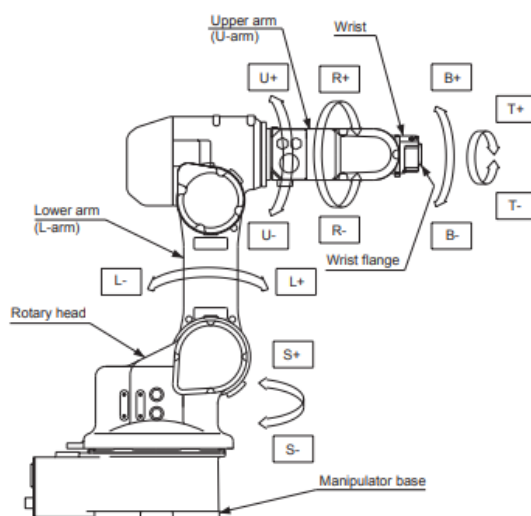


Figura 3.2 - Nome das estruturas e sistema de eixos *Standard Job*. [18]

Ambos os sistemas podem-se converter um no outro. Através da cinemática direta, o *Standard Job* converte-se em *Relative Job* e através da cinemática inversa realiza-se o processo contrário. A figura 3.3 mostra o processo mencionado. [19]

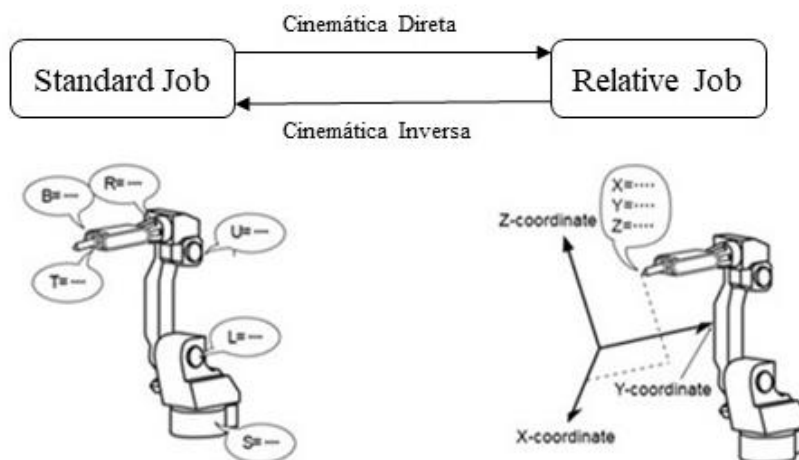


Figura 3.3 - Utilização de cada sistema de eixos de trabalho. [20]

Neste projeto, o sistema de trabalho utilizado é o sistema *Relative Job*, com o uso de coordenadas cartesianas uma vez que, do ponto de vista do utilizador, o sistema cartesiano é de simples utilização. Escolhido o sistema de trabalho e o tipo de coordenadas a usar, foi necessário selecionar e fixar um sistema de coordenadas que atue como referência para todos os elementos do robô. A figura 3.4 apresenta o sistema de coordenadas cartesianas de referência selecionado, fixado na base do robô HP3L.

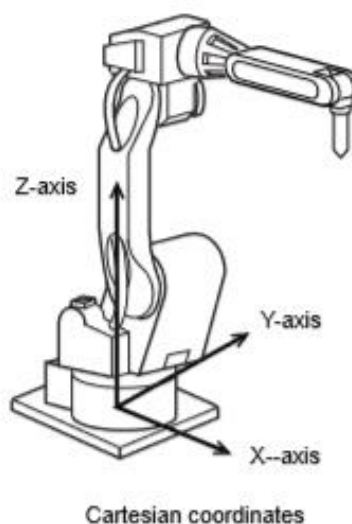


Figura 3.4 - Sistema de coordenadas cartesianas de referência. [20]

Um sistema de coordenadas cartesianas de três dimensões, consiste em três eixos de coordenadas: eixo X, eixo Y e eixo Z, perpendiculares entre si interseçados num ponto

chamado origem. Com nomenclatura (X, Y, Z), os eixos permitem localizar um ponto no espaço, especificando o valor da distância entre a sua localização e a origem do referencial. Os eixos podem também sofrer rotações, rotação em X, rotação em Y e rotação em Z, assumindo a nomenclatura (Rx, Ry, Rz). [20]

### 3.3 Cinemática Direta

Amplamente utilizada na área da robótica, a cinemática direta permite obter a posição de vários elementos de um robô em relação à sua posição base. É representada por um conjunto de matrizes, as matrizes de *Denavit-Hartenberg*,  ${}^{i-1}T_i$ , que transforma as coordenadas do sistema i para o sistema i-1. A matriz de *Denavit-Hartenberg*,  ${}^{i-1}T_i$ , encontra-se deduzida a seguir. [21]

$$\begin{aligned}
 {}^{i-1}T_i &= T_{x, a_i} * T_{x, \alpha_i} * T_{z, \theta_i} * T_{z, d_i} \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_i \\ \cos(\alpha_i)\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i) & -\sin(\alpha_i)d_i \\ \sin(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\cos(\theta_i) & \cos(\alpha_i) & \cos(\alpha_i)d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)
 \end{aligned}$$

onde  $a_i$ ,  $\alpha_i$ ,  $\theta_i$  e  $d_i$  correspondem a parâmetros de *Denavit-Hartenberg*.

O parâmetro  $a_i$  e o parâmetro  $\alpha_i$  correspondem ao deslocamento e à rotação no eixo X, respetivamente, o parâmetro  $\theta_i$  e o parâmetro  $d_i$  correspondem à rotação e ao deslocamento no eixo Z, respetivamente. Para se obter as matrizes das diferentes articulações do HP3L foi representado ao longo do mesmo a disposição dos eixos cartesianos para se retirar os parâmetros de *Denavit-Hartenberg*. A figura 3.5 apresenta a configuração dos eixos ao longo das articulações do robô, e a tabela 3.1 apresenta o valor dos parâmetros de *Denavit-Hartenberg* retirados da mudança do sistema i-1 para o sistema i.

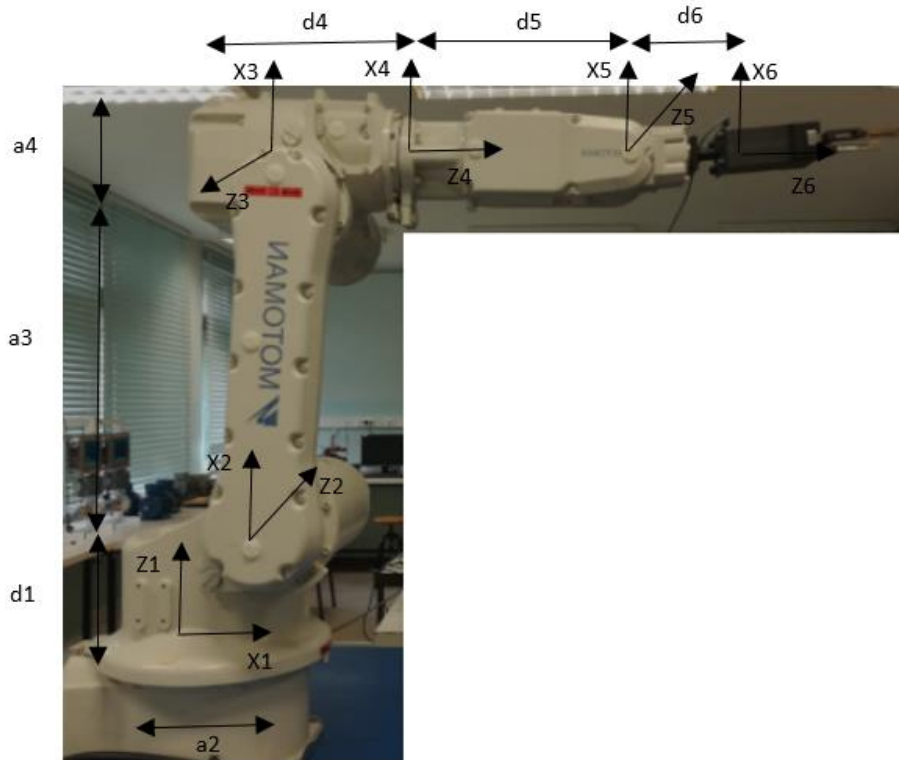


Figura 3.5 - Disposição dos eixos nas articulações do robô HP3L.

Tabela 3.1– Valores dos parâmetros *Denavit-Hartenberg* retirados dos eixos.

Sistema de eixo $i-1 \rightarrow i$	Nº da articulação	X		Z	
		$a_i$	$\alpha_i$ (°)	$\theta_i$ (°)	$d_i$
0→1	1	0	0	$\theta_1$	$d_1$
1→2	2	$a_2$	-90	$\theta_2-90$	0
2→3	3	$a_3$	180	$\theta_3$	0
3→4	4	$a_4$	90	$\theta_4$	$d_4 + d_5$
4→5	5	0	90	$\theta_5$	0
5→6	6	0	-90	$\theta_6$	$d_6$

O parâmetro  $a_i$  do eixo X, e o parâmetro  $d_i$  do eixo Z, correspondem ao comprimento das articulações do robô. Os seus valores encontram-se representados na tabela 3.2.

Tabela 3.2 – Valores dos parâmetros de X e Y .

<b>X</b>	<b>Z</b>
<b>a<sub>i</sub> (mm)</b>	<b>d<sub>i</sub> (mm)</b>
a2=100	d1=300
a3=370	d4=191.5
a4=85	d5=188.5
	d6=90

Depois de definido o valor dos parâmetros, deduz-se as matrizes de *Denavit-Hartenberg* dos sistemas. Elas possibilitam o movimento entre o sistema anterior e o sistema seguinte. As matrizes finais encontram-se apresentadas em seguida.

$${}^0T_1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^1T_2 = \begin{bmatrix} \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 \\ 0 & 0 & 1 & 0 \\ \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$${}^2T_3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3 \\ -\sin(\theta_3) & -\cos(\theta_3) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$${}^3T_4 = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & a_4 \\ 0 & 0 & -1 & -d_4 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$${}^4T_5 = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$${}^5T_6 = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ -\sin(\theta_6) & -\cos(\theta_6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

A cinemática inversa não é deduzida no presente projeto, devido a não ser aplicada diretamente nas interfaces, no entanto, ela é utilizada internamente pelo robô.

### 3.4 Arquivos de trabalho para comunicação RS232

Os arquivos de trabalho designados por *jobs*, são arquivos que contêm a programação das ações ou tarefas a realizar pelo robô, em linguagem de programação *INFORM*, linguagem característica dos robôs da marca Motoman. Os *jobs* contêm todo o *software*: declaração de endereços, variáveis, instruções de movimentos, instruções de interação com outros equipamentos, etc, que são enviados para o banco de trabalho do robô, processados e executados. [22]

Os robôs Motoman podem também ser programados em *offline* através do seu *software* Motosim. O *software* Motosim permite simular células de robôs em 3D, otimizar os espaços de trabalho, calcular colisões, melhorar performance, fazer cálculos de ciclo e utilizar linguagem *INFORM* de forma mais fácil, entre outras funcionalidades. [22] A figura 3.6 apresenta uma simulação, no *software* Motosim, do robô HP3L da Motoman a executar o problema das torres de Hanói com layout do laboratório. Como se pode verificar, é possível criar ambientes e objetos que simulam perfeitamente um sistema real.

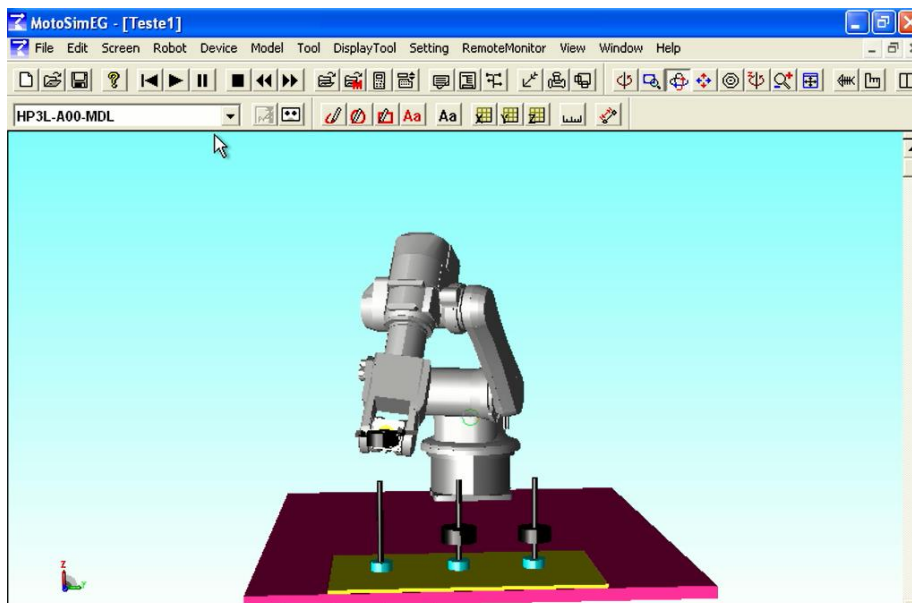


Figura 3.6 - Simulação do robô HP3L da Motoman no *Software* Motosim.

No presente projeto, utilizou-se um arquivo de trabalho para comunicação RS232 que possibilita ao robô receber e interpretar a mensagem das interfaces.

A mensagem é definida por um padrão, X, Y, Z, Rx, Ry, Rz, 0/1: onde as três primeiras coordenadas correspondem ao valor dos 3 eixos cartesianos (milésimas de milímetro), seguida do valor das coordenadas de rotação dos mesmos eixos (centésimas de grau), do valor da garra que pode estar no modo fechada, (0), ou aberta, (1). Os símbolos presentes na mensagem têm a função de sinalizar o estado dela durante a sua leitura no arquivo de trabalho. O caracter “,” informa que a mensagem está incompleta e que há mais valores para serem lidos, o caracter “:” sinaliza o fim de uma mensagem e a letra “f” existente no *job* sinaliza o fim do programa. A comunicação série é configurada a uma taxa de transmissão de 9600 *bps*, 8 *bits* e 1 *stop bit*. Além da interpretação da mensagem, o arquivo de trabalho, retorna os ângulos dos movimentos do robô para o PC utilizando a sua cinemática inversa interna e analisa a variação do estado de três botões externos. A figura 3.7 apresenta a mensagem utilizada com o valor da posição inicial (P020) do robô e a figura 3.8 apresenta o fluxograma da programação do arquivo de trabalho para comunicação RS232 de forma a simplificar e se obter uma visão geral do processo. A programação do arquivo de trabalho encontra-se no Anexo I.

+

X	Y	Z	Rx	Ry	Rz	Garra
724189	,-75536	,165660	,-1607	,-8102	,-17009	,0

Figura 3.7 - Mensagem da posição inicial do robô, em milésimas de milímetro e centésimas de grau.

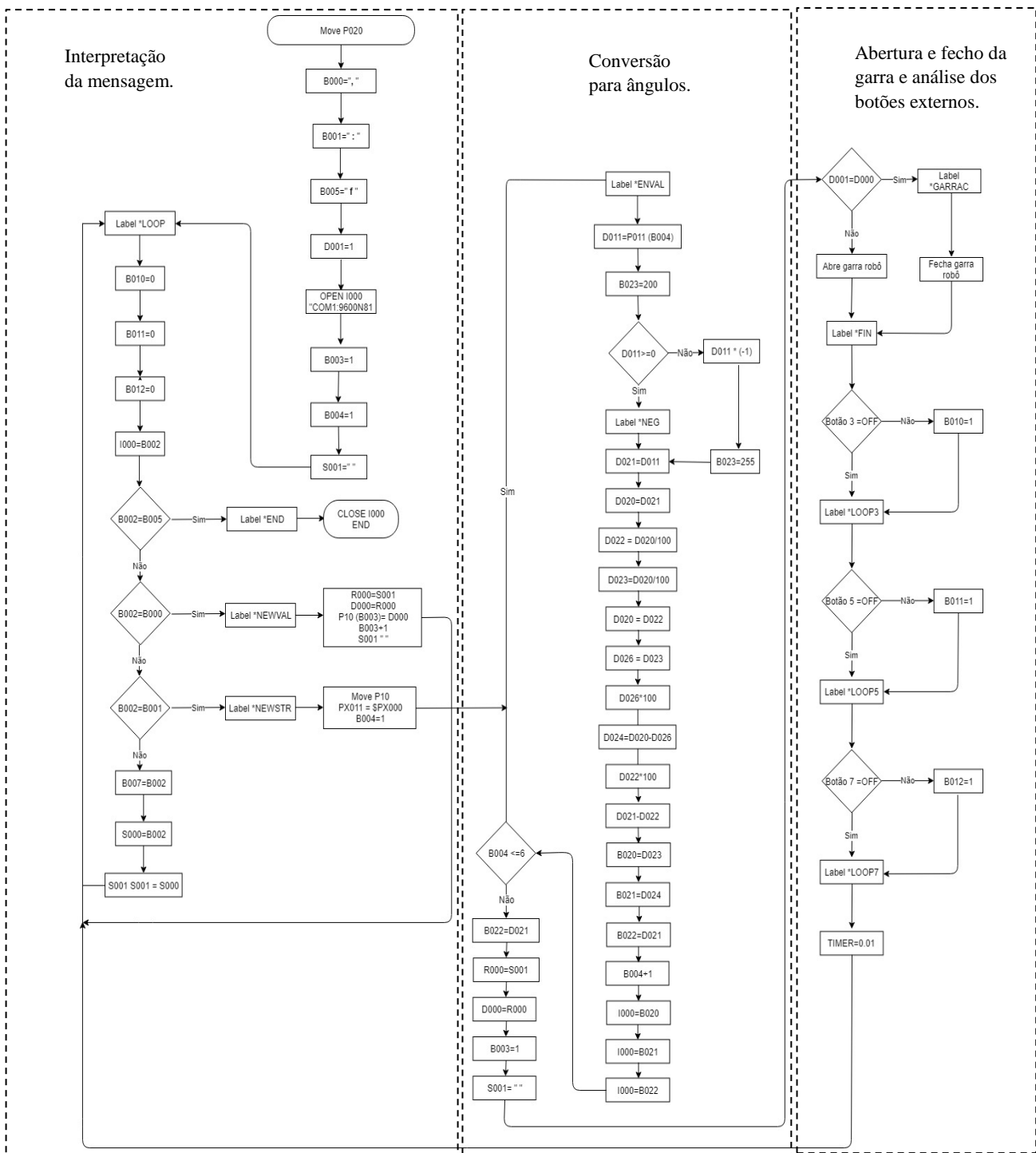


Figura 3.8 - Esquema do processamento de uma mensagem enviada pelas interfaces, em programação *IMFORM*.

### 3.5 Consola do Robô HP3L com controlador NX100 da Motoman

A consola que acompanha o controlador NX100 para o robô HP3L da Motoman é uma consola com *touchscreen* de 6.5 polegadas que possibilita ao utilizador, através das suas chaves e botões, manipular e realizar todas as ações no robô, incluindo editar, reproduzir e eliminar arquivos de trabalho.

Possibilita também melhorar o tempo de programação, facilitar a aprendizagem do robô pela mão do utilizador, comunicar erros e alertas, ajuda na sua manutenção e também efetuar interrupções de emergência, entre outras funcionalidades. [23] A figura 3.9 apresenta o *layout* gráfico da consola.



Figura 3.9 - *Layout* gráfico da consola.

Como referido, algumas funções da consola servem de padrão para as restantes interfaces. As funções passam por movimentar, abrir e fechar a garra, gravar posições, reproduzi-las e alterar a velocidade do Motoman.

Para o utilizador interagir através dela com o robô deve, em primeiro lugar, seleccionar o modo de aprendizagem, o *teach mode* presente no botão de seleção da consola. Seguidamente, para se sinalizar a intenção de mexer o Motoman, é necessário pressionar o botão *enable* até se ouvir um terceiro *click* e clicar no botão [*SERVO ON READY*]. Nestas condições, pode-se pressionar os botões dos eixos que se pretende mover e caso se queira aumentar a velocidade de deslocação pressionar ao mesmo tempo os botões de velocidade. Existem quatro níveis de velocidade; lenta, média, rápida e progressiva. Sempre que é pressionado o botão [*FAST*] a velocidade muda na ordem: lenta, média, rápida e progressiva, e quando pressionado o botão [*SLOW*] a velocidade é alterada no sentido contrário. Ainda assim, a consola disponibiliza o botão [*HIGH SPEED*].

Para gravar e reproduzir posições é preciso criar um arquivo de trabalho. O arquivo de trabalho é criado ao se selecionar o ícone “JOB” no *touchscreen* e depois o sub-ícone “Create New JOB”, dando-se um nome ao arquivo. O sistema de coordenadas é escolhido através do botão [COORD]. Para se gravar uma posição do robô, ativa-se o *insert* e clica-se na tecla [ENTER] para que seja registado a nova linha de movimento. Para abrir ou fechar a garra é necessário selecionar o valor da variável *Dout* presente no arquivo para 1 ou para 0, respetivamente.

Quando todas as posições estiverem gravadas, move-se o cursor para a linha inicial do *job* e pressiona-se a tecla [INTERLOCK] e a tecla [TEST START]. O processador lê e reproduz todas as etapas do arquivo. Para modificar ou eliminar alguma linha de movimento, move-se, através da tecla [FWD], até ao número da linha alvo e pressiona-se a tecla [MODIFY] ou [DELETE], respetivamente. Outra maneira de reproduzir uma trajetória, é modificar o botão de seleção de *teach mode* para *play mode* clicando no botão *enable* e no botão [START]. [24] A figura 3.10 apresenta a consola e a indicação das teclas mencionadas.

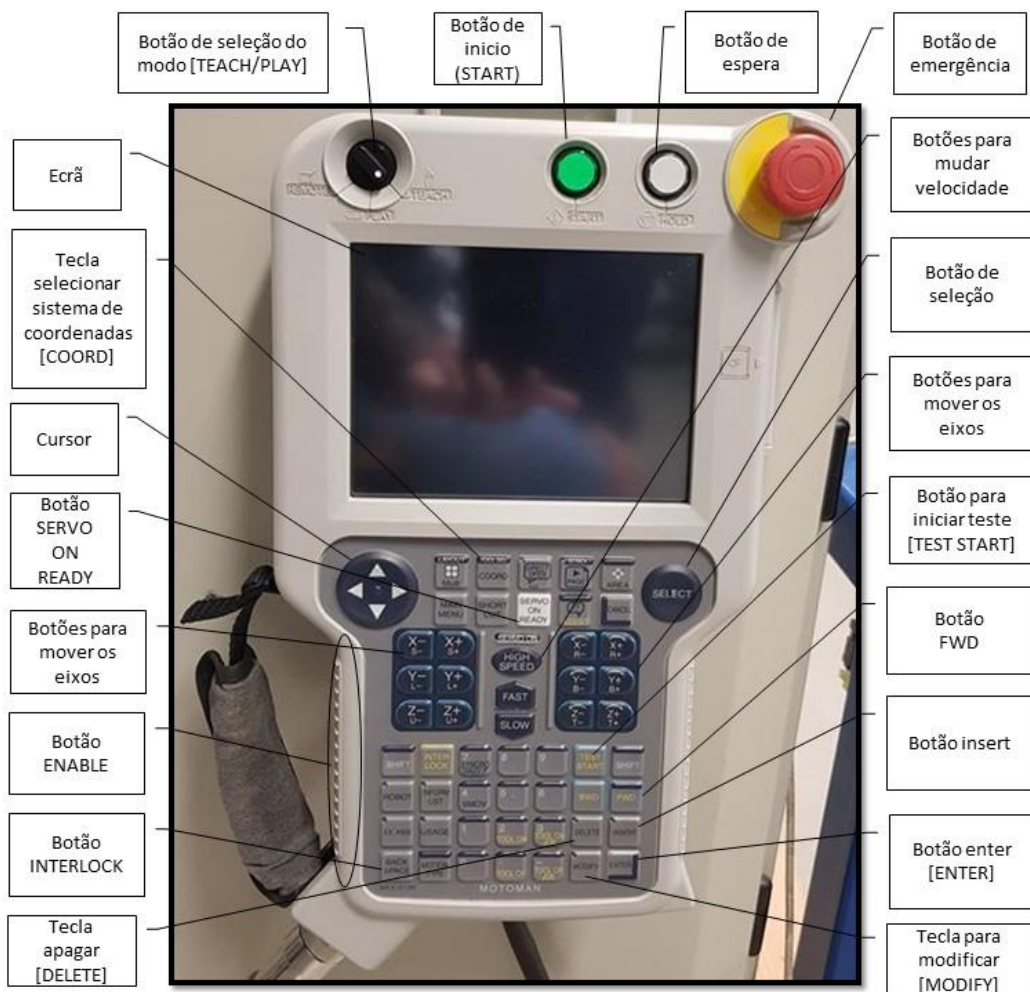


Figura 3.10 - Consola do controlador NX100 para o robô da Motoman com indicação das teclas referidas.

Na consola para se realizar as ações mencionadas anteriormente é necessário segurá-la de forma firme e clicar, em certas alturas, em quatro botões ao mesmo tempo dificultando a tarefa para o utilizador, como mostra a figura 3.11, existindo também a necessidade de alguns conhecimentos de programação.

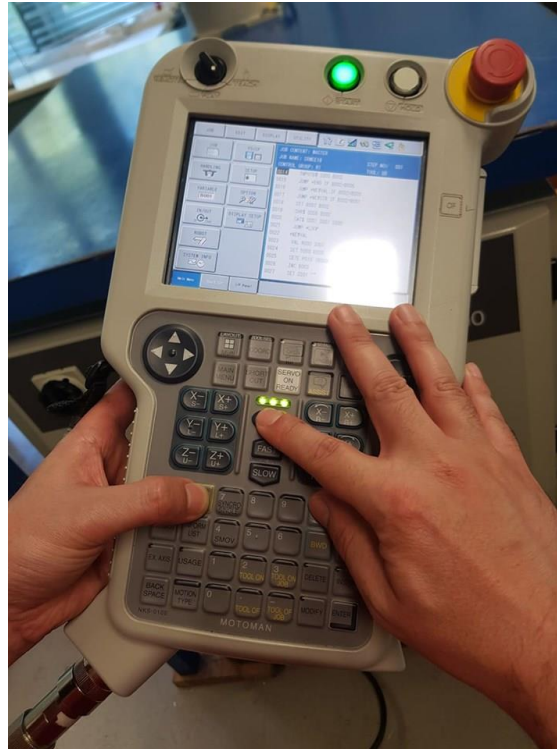


Figura 3.11 - Utilizador a usar a consola

## 4 Interfaces

O capítulo 4 representa o capítulo mais prático do presente projeto. É nele que se apresenta e se explica as interfaces desenvolvidas, dividindo-se em secções de *hardware*, *software* e modo de utilização. Na secção *hardware*, são mencionados os componentes físicos e os conversores de comunicação utilizados: o adaptador série Bluetooth-RS232 e o módulo RS232 Serial Port. No *software*, explica-se as linguagens de programação utilizadas, os programas, os algoritmos desenvolvidos e as interfaces gráficas. Por último, é descrito, passo a passo, o modo de funcionamento de cada interface para que qualquer utilizador consiga interagir com o robô através delas.

### 4.1 Interface Cubo

A interface cubo foi desenvolvida para permitir uma interação intuitiva e fácil entre o homem e o robô HP3L da Motoman, apresentada na figura 4.1.



Figura 4.1 - Interação entre o utilizador e o robô com a interface cubo.

A interface cubo é composta por 5 elementos: pelo objeto cubo que tem o propósito de simular o espaço tridimensional para ajudar o utilizador na orientação dos eixos cartesianos; pelo prolongamento que além de suportar o cubo permite a colocação da mão do utilizador e a colocação de botões; pelos botões e interruptores que representam mecanismos de controlo utilizados em larga escala na indústria mais precisamente nos painéis de comando *HMI*; pelo Arduino que processa todos os dados e pelo conversor RS232 Serial Port ou pelo adaptador série Bluetooth-RS232, que permitem a conversão do sinal do processador para comunicação RS232. A interação inicia-se quando o utilizador pressiona um botão correspondente à ação que quer efetuar no robô. A mudança do estado do botão é detetada, processada e é enviada a respetiva mensagem. O envio pode ser feito por cabo RS232 ou por Bluetooth, caso o utilizador queira tornar a interface totalmente portátil. A figura 4.2 apresenta o *layout* da interface e a figura 4.3 apresenta o *layout* da mensagem enviada para o robô.

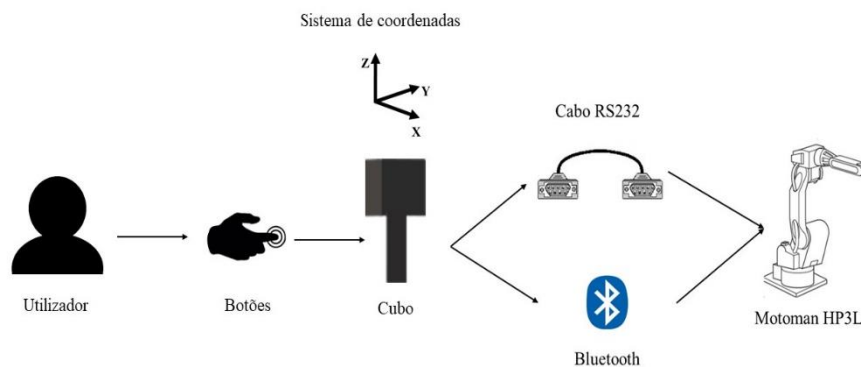


Figura 4.2 - *Layout* da interface cubo.

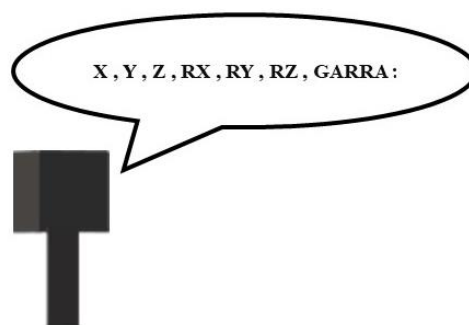


Figura 4.3 – *Layout* da mensagem enviada para o robô da interface cubo.

#### 4.1.1 Arduino Uno

O Arduino uno, apresentado na figura 4.4, é uma plataforma eletrónica *open-source* que permite uma fácil e flexível utilização de *hardware* e *software*. Da gama de Arduino existentes, o Arduino uno é o mais usado e o mais documentado. Incorpora 14 entradas/saídas digitais (0-13), onde 6 podem ser usadas como saída de sinal *PMW*, 6 entradas analógicas (A0-A5), uma conexão USB, uma entrada para alimentação externa

do tipo *jack*, um botão *reset*, uma *UART* (*Universal Asynchronous Receiver/Transmitter*) e um oscilador de cristal de 16 MHz. [24] A tensão de operação é de 5 Volts, mas quando alimentado por fonte externa pode receber uma tensão entre os 7 Volts e os 12 Volts. O microcontrolador presente no Arduino é o microcontrolador ATmega 328P com uma *EEPROM* de 1kB, sendo programado na plataforma IDE Arduino através de linguagem C++. A *UART* é responsável pelo *upload* do código da plataforma IDE para o microcontrolador e pela comunicação com outros dispositivos, representados pelo pino 0 (Rx- Recepção) e pino 1 (Tx-Transmissão). Esta comunicação pode ser realizada em diferentes taxas de transmissão, sendo que neste projeto é utilizada uma taxa de transmissão de 9600 *bps*. [26] A figura 4.5 mostra a sua arquitetura interna.



Figura 4.4 - Arduino Uno.

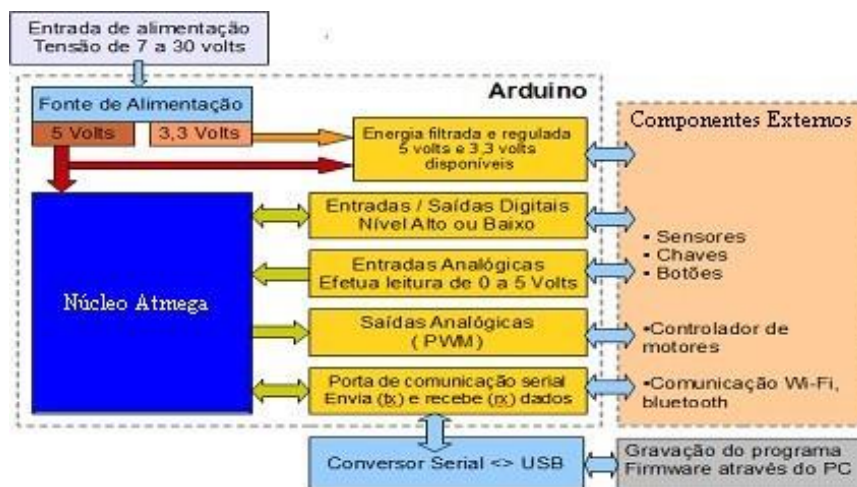


Figura 4.5 - Arquitetura interno do Arduino Uno. [25]

4.1.2 Hardware

Para a conceção do objeto elaborou-se um esboço com a forma e as dimensões que este deveria apresentar para que cumprisse o requisito de acomodar no seu interior todo o *hardware* necessário. A estrutura e as suas dimensões, em mm, assim como a estrutura real visualizam-se na figura 4.6.

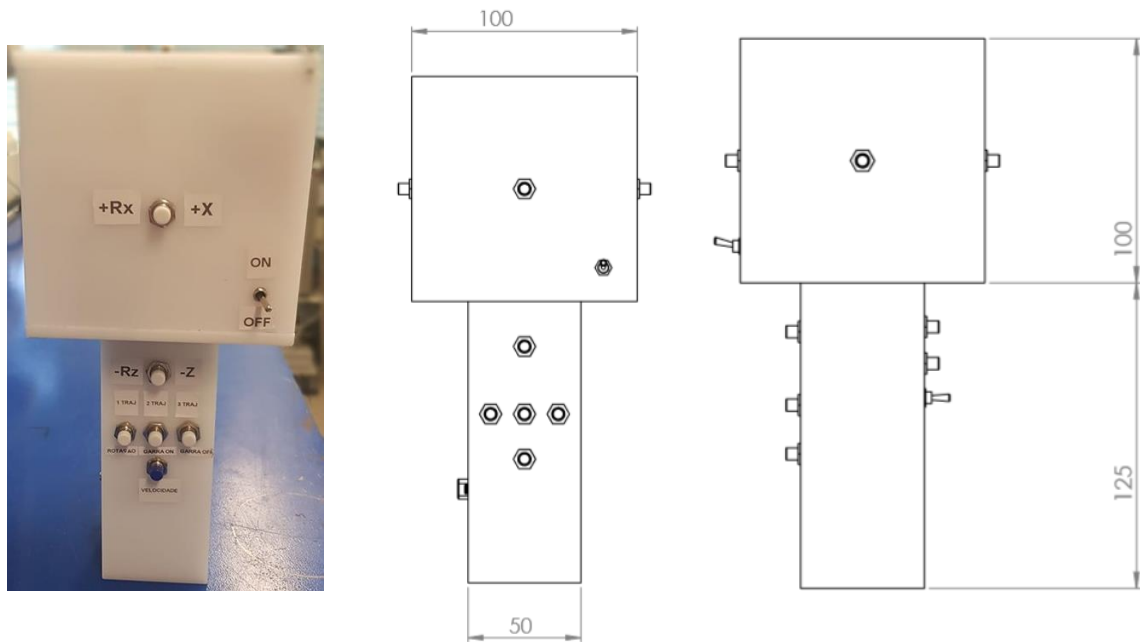


Figura 4.6 - Projeto da estrutura da interface cubo com as respetivas dimensões.

Vários são os elementos que permitem obter as funcionalidades traçadas. A figura 4.7 apresenta a sequência de componentes utilizados e necessários.

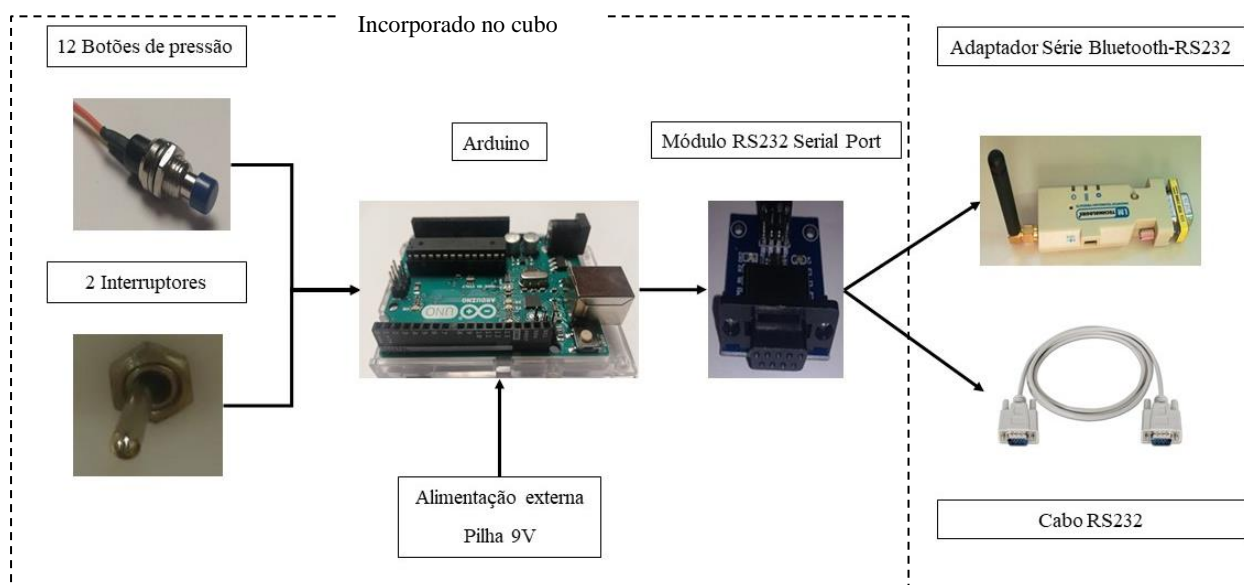


Figura 4.7 - Hardware da interface cubo.

Cada botão presente no cubo corresponde a um eixo cartesiano, possibilitando a translação ou rotação do mesmo. Saber a orientação dos eixos é essencial para uma boa interação. A figura 4.8 apresenta a configuração e o sentido dos eixos cartesianos e a correspondência dos botões do cubo.

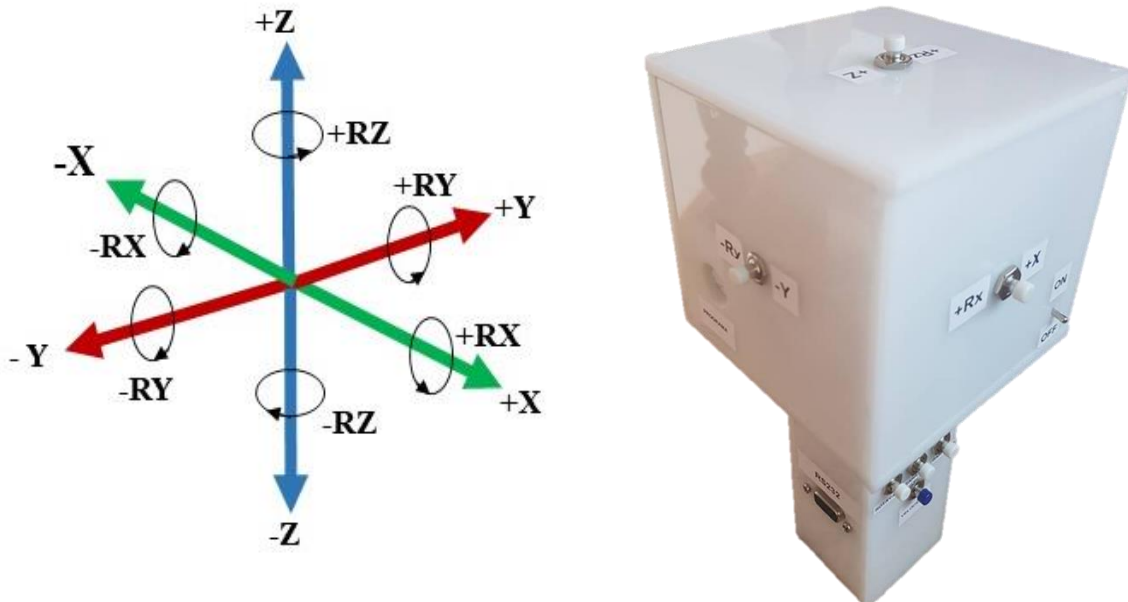
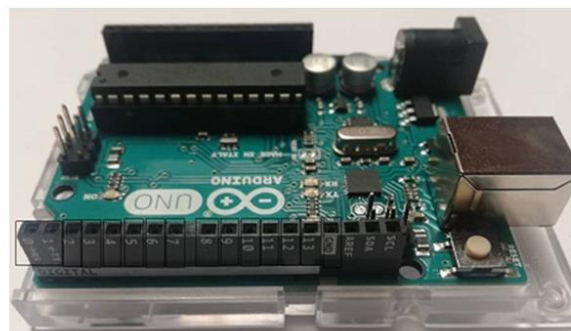


Figura 4.8 - Sistema de coordenadas cartesianas correspondente a cada botão.

Os botões encontram-se distribuídos e ligados às entradas digitais e analógicas do Arduino. A cada botão é atribuído uma entrada, e definido a sua função. A figura 4.9 apresenta a distribuição e a função atribuída a cada um nas entradas/saídas digitais e analógicas do Arduino.



0 → Canal série Rx	7 → -Y / -RY
1 → Canal série Tx	8 → +Y / +RY
2 → Fechar garra/Selecionar 2ª trajetória	9 → -X / -RX
3 → Gravar Trajetória	10 → +X / +RX
4 → Selecionar velocidade	11 → Seleciona rotação/Selecionar 1ª trajetória
5 → -Z / -RZ	12 → Abrir garra/Selecionar 3ª trajetória
6 → +Z / +RZ	13 → Limpar trajetórias
A0 → Selecionar modo Teach/Play	GND → Terra

Figura 4.9 - Correspondência de cada botão à entrada/saída digital ou analógica do Arduino.

O esquema elétrico da interface cubo está representado na figura 4.10. Os botões são acionados a 0 Volts.

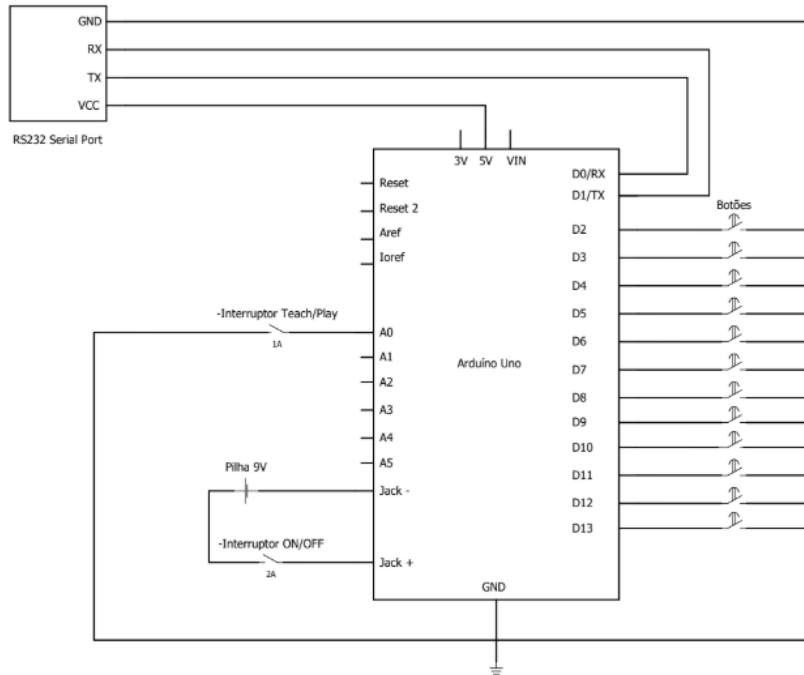


Figura 4.10 - Esquema elétrico do *hardware* da interface cubo.

#### 4.1.3 Software

Todo o *software* foi desenvolvido no IDE Arduíno com programação C++. A programação é baseada em algoritmos: o algoritmo velocidade, o algoritmo translação, o algoritmo rotação, o algoritmo garra e o algoritmo memória. Para efetuar as translações e rotações no robô, foi definido um valor constante que é somado ou subtraído, dependendo do eixo em questão, ao movimento anterior para gerar o movimento seguinte. Este valor é chamado de passo e para a translação o valor é de 1000 e para a rotação o valor é de 100. De forma a explicar os algoritmos desenvolvidos, é apresentado neste ponto, os respectivos fluxogramas.

Algoritmo Velocidade

O algoritmo velocidade, como o nome indica, permite modificar a velocidade dos movimentos. Estão disponíveis quatro velocidades que se permutam consoante o número de vezes que se pressiona o respetivo botão. A figura 4.11 representa o fluxograma deste algoritmo.

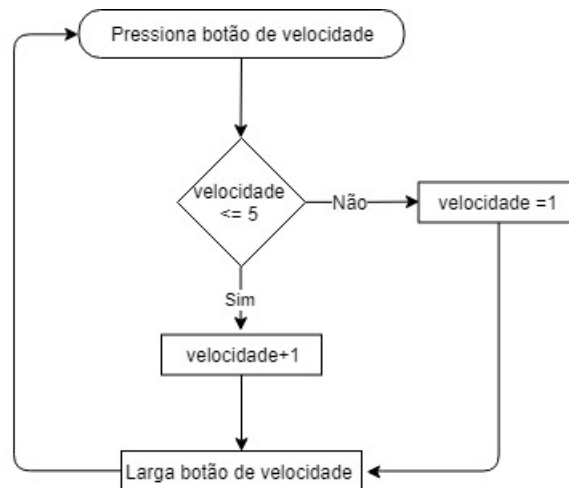


Figura 4.11 - Fluxograma algoritmo velocidade.

Algoritmo Translação

O algoritmo translação corresponde à translação dos eixos do robô quando se pressiona um dos botões: +X, -X, +Y, -Y, +Z, -Z. A figura 4.12 apresenta o seu fluxograma.

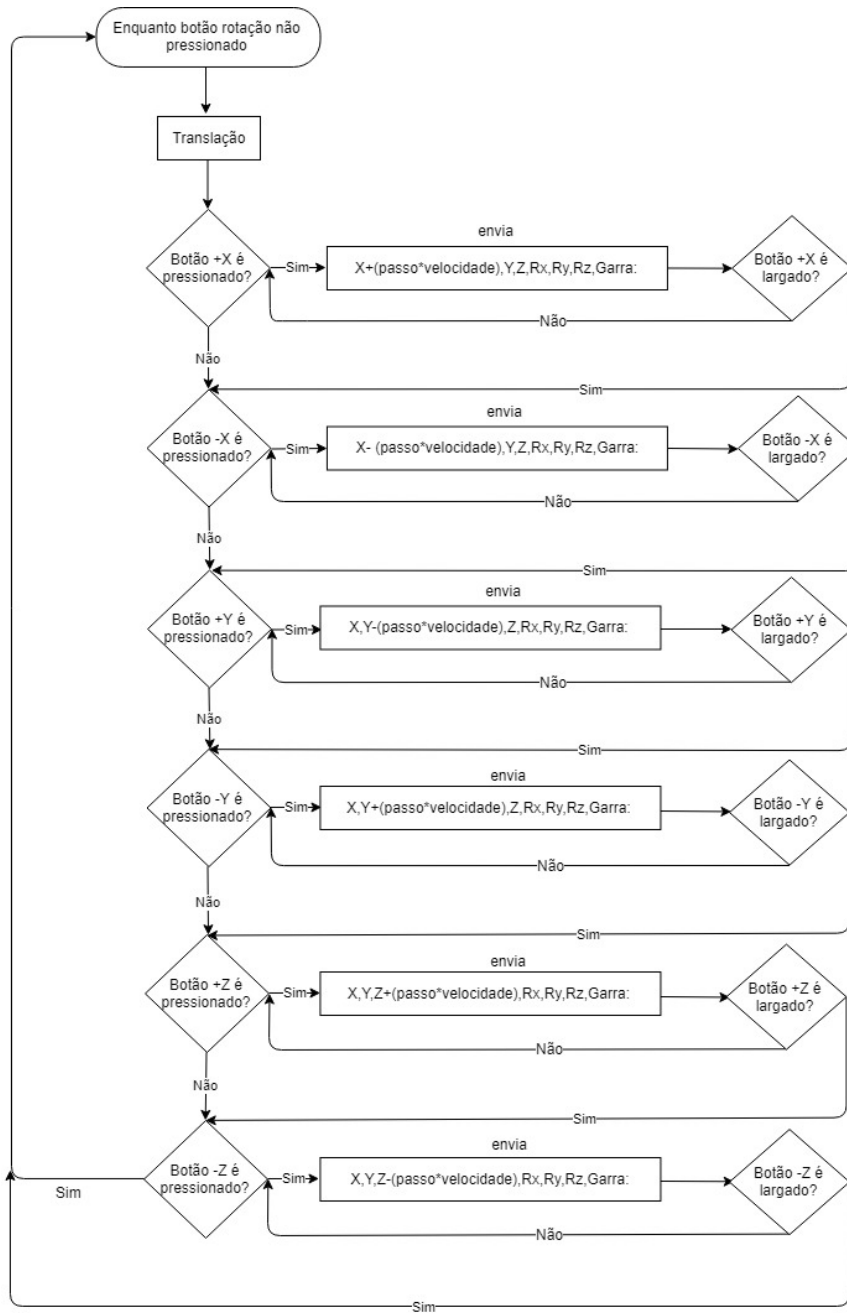


Figura 4.12 - Fluxograma do algoritmo translação.

Algoritmo Rotação

O algoritmo rotação corresponde à rotação dos eixos do robô quando se pressiona um dos botões: +Rx, -Rx, +Ry, -Ry, +Rz, -Rz. Em simultâneo, o botão que habilita a rotação deve ser pressionado constantemente. A figura 4.13 apresenta o fluxograma do algoritmo.



### Algoritmo Garra

O algoritmo garra possibilita a abertura ou o fecho da garra, consoante a pressão de um dos botões para esse efeito. A figura 4.14 apresenta o fluxograma do algoritmo.

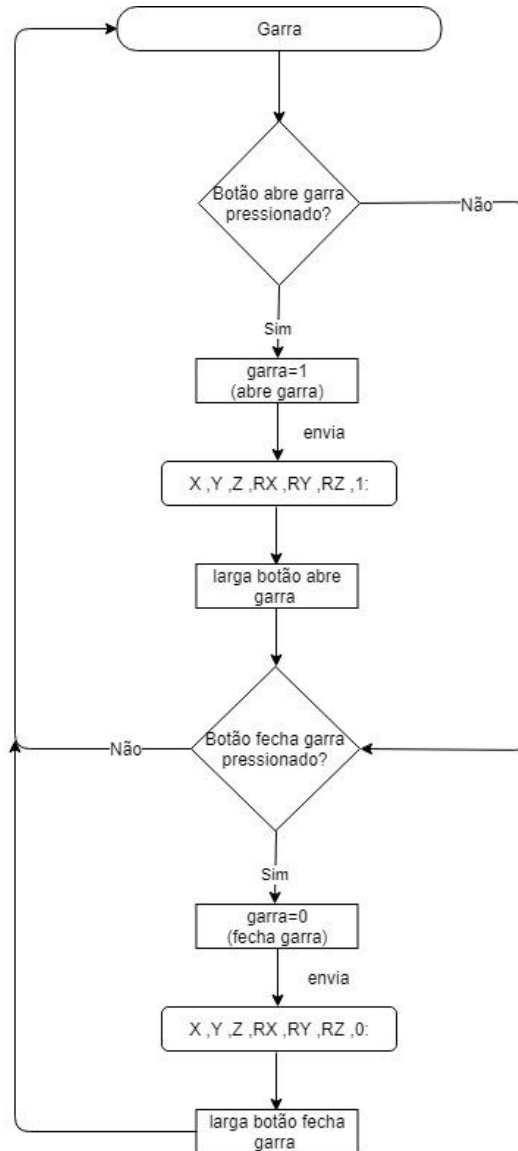


Figura 4.14 - Algoritmo garra em linguagem corrente.

### Algoritmo Memória

Para escrever ou ler dados na memória do Arduíno, *EEPROM* com capacidade de 1kB, é necessário especificar o endereço onde se aloca os valores da posição gravada. Assim, o *address* é o endereço ou posição da memória entre 0 e 1023, e o *value* é o valor inteiro armazenado entre 0 e 255, correspondente a 1 *byte*. Como o valor de cada eixo é um número maior que 255, é necessário disponibilizar na memória 4 *bytes* para cada. O algoritmo responsável pela escrita na memória é o *EEPROMWritelong* e o algoritmo

responsável pela leitura na memória é o *EEPROMReadlong*. Os algoritmos recebem os *address* e os *values* das posições do robô por passagem por valor.

*EEPROMWritelong*

O algoritmo *EEPROMWritelong* permite a escrita na memória do Arduino. Como são reservados 4 *bytes* para cada valor, é escrito em primeiro o *byte* menos significativo, o quarto *byte*, deslocando-se sempre 8 *bits* até chegar ao *byte* mais significativo, o primeiro *byte*. A figura 4.15 representa o fluxograma desta função.

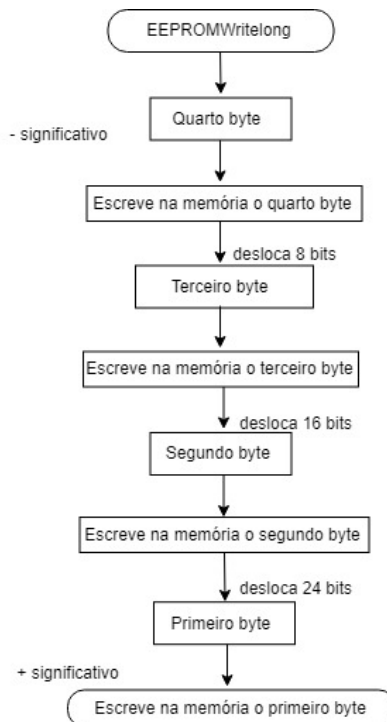


Figura 4.15 - Função de escrita na memória *EEPROM*.

*EEPROMReadlong*

A função *EEPROMReadlong* lê acedendo aos respetivos endereços de memória os valores guardados. A leitura inicia-se no *byte* menos significativo até ao *byte* mais significativo, reconstruindo cada valor. A figura 4.16 representa o fluxograma desta função.

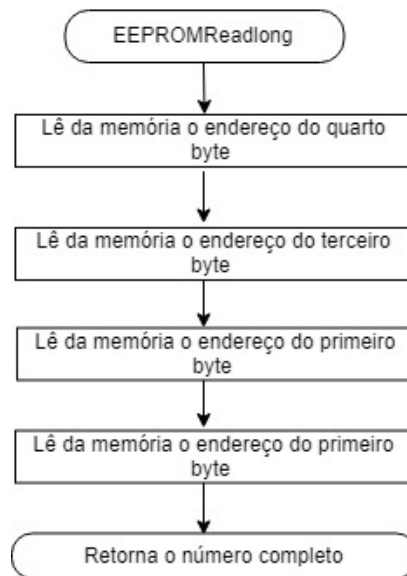


Figura 4.16 - Função de leitura na memória *EEPROM*.

#### 4.1.4 Modo de utilização

Este ponto 4.1.4 chamado de modo de utilização pretende explicar de forma detalhada o modo de funcionamento da interface.

O primeiro passo para iniciar a interação com o cubo é colocar na ranhura identificada como “RS232” localizado no prolongamento, o cabo RS232 ou o conversor Bluetooth, consoante o desejo do utilizador. Caso se opte pela comunicação sem fios, deve-se “emparelhar” os dois conversores de Bluetooth, um na ranhura e outro no robô. A interface possui um interruptor para o ligar/desligar. Ao posicionar o interruptor na posição ON, o cubo está apto a funcionar e o robô move-se para a sua posição inicial. No prolongamento, encontra-se outro interruptor com a opção do modo *teach/play*. O modo *teach* permite ensinar o HP3L, sistema semelhante ao utilizado na consola. Ao seleccionar este modo, o utilizador pode pressionar os botões das suas faces para movimentar o Motoman, pressionar o botão velocidade para duplicar, triplicar ou quadruplicar a velocidade, o botão de abrir a garra, o botão de fechar a garra, o botão de gravar as posições pretendidas para efetuar trajetórias e o botão de as apagar, estes últimos localizados no prolongamento do cubo. Uma pressão num botão corresponde apenas a uma ação. Para efetuar o movimento de rotação, deve-se pressionar constantemente o botão da rotação aquando da seleção dos botões dos eixos. Os botões dos eixos podem também ser pressionados constantemente se se pretender um movimento contínuo. A gravação de cada posição é sequencial. Uma trajetória é um conjunto de até sete posições, ao fim deste número, a posição seguinte pertencerá à próxima trajetória. A interface está habilitada para gravar até três trajetórias. O modo *play* permite reproduzir as trajetórias. Assim, o utilizador deve mudar para este modo apenas quando quiser efetuar essa ação,

pressionando um dos botões: trajetória 1, trajetória 2 ou trajetória 3. Para finalizar a interação basta desligar o cubo.

A interface cubo disponibiliza também uma entrada para fazer o *upload* de outros programas para o Arduíno ou até para efetuar alimentação por cabo. Nas figuras 4.17 a, 4.17 b, 4.17 c, 4,17 d visualiza-se todos os planos da interface e todos os botões referidos.

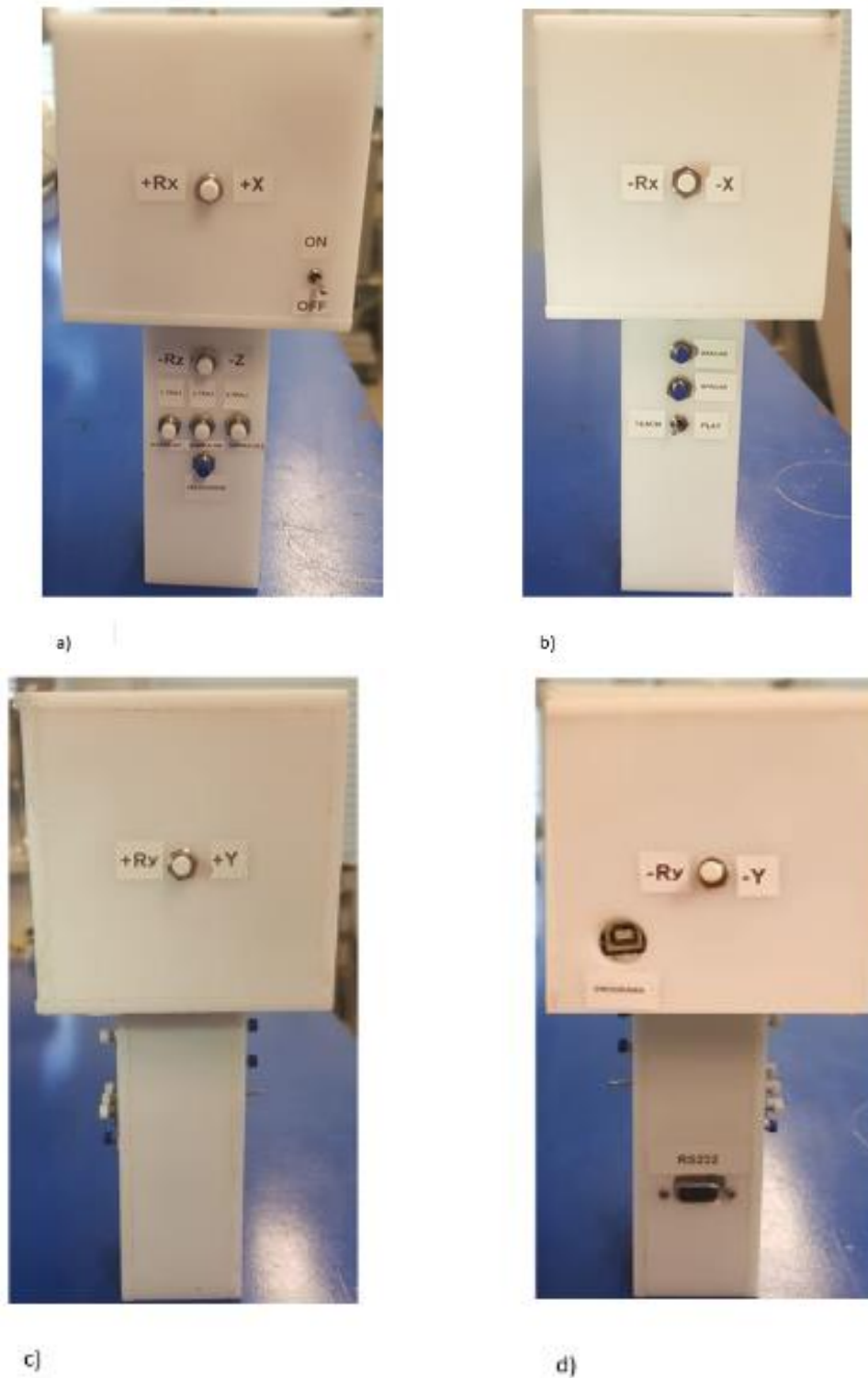


Figura 4.17 - a) Parte frontal da interface Cubo, b) Parte Traseira da interface Cubo, c) Parte Direita da interface Cubo, e d) Parte Esquerda da interface Cubo.

## 4.2 Interface Telemóvel

A interface telemóvel, apresentada na figura 4.18, foi desenvolvida para explorar e interatuar com tecnologias atuais. O telemóvel com aplicações e a comunicação sem fios são alvos de interesse nas interações da indústria.



Figura 4.18 - Interação entre o utilizador e o robô com a interface telemóvel.

Apresenta-se como uma aplicação para sistema operativo (SO) *Android* baseado no acelerómetro do telemóvel. A alteração da sua posição é o mecanismo responsável pelos movimentos. A interface gráfica apresenta-se com itens de seleção para escolher o eixo a modificar, alterar velocidade, gravar e reproduzir trajetórias. O envio da mensagem efetua-se por comunicação Bluetooth sendo que o emissor e o recetor necessitam de se conectar. A aplicação foi desenvolvida numa plataforma online *open-source App Inventor for Android*, mantida atualmente pela *Massachusetts Institute of Technology* (MIT). A *App Inventor* utiliza uma linguagem de blocos visuais numa ação de arrastar e largar (*drag-and-drop*). O *layout* desta interação está representado na figura 4.19 e o *layout* da mensagem enviada por Bluetooth está representado na figura 4.20.

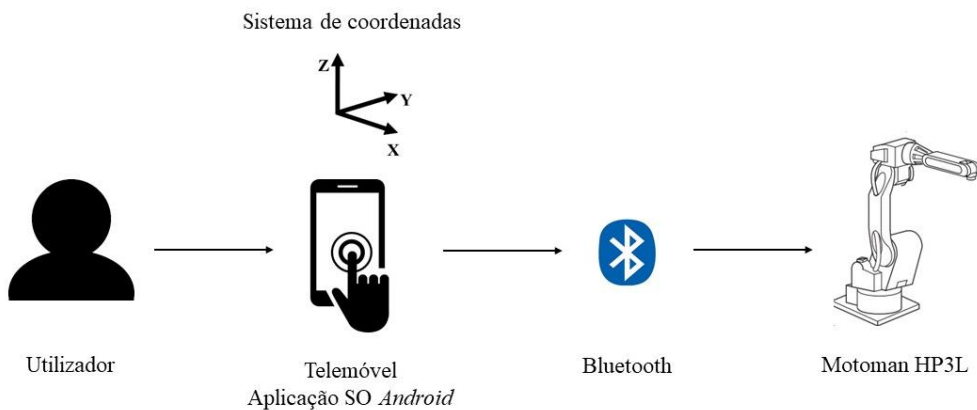


Figura 4.19 - *Layout da interface telemóvel.*



Figura 4.20 - *Layout da mensagem enviada para o robô da interface telemóvel.*

#### 4.2.1 Hardware

A interface é composta por diferentes componentes que constituem o seu *hardware*, estes encontram-se visualizados na figura 4.21.



Figura 4.21 - *Hardware da interface telemóvel.*

Para uma boa utilização do telemóvel e da aplicação, é essencial saber a disposição dos eixos no acelerómetro do telemóvel. A figura 4.22, apresenta essa disposição dos eixos.

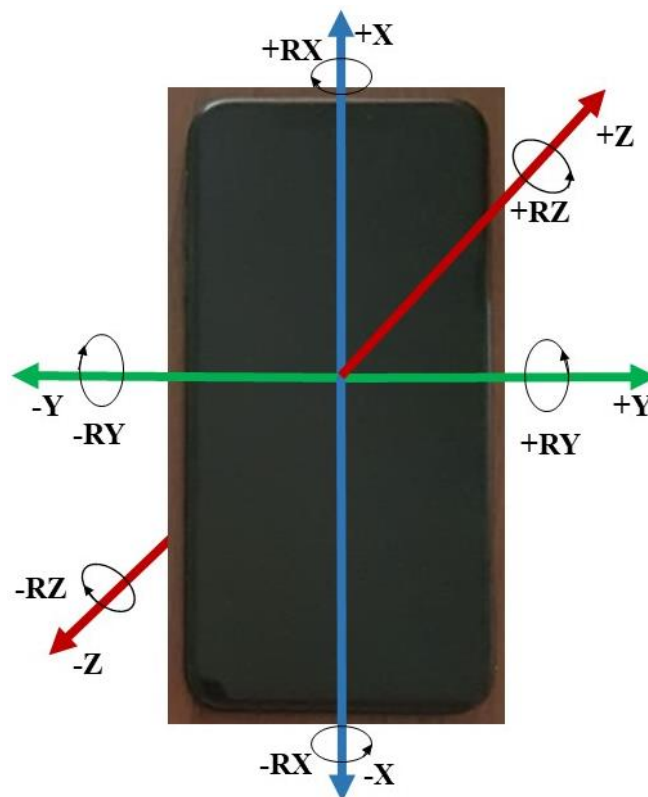


Figura 4.22 - Disposição dos eixos no acelerómetro do telemóvel.

#### 4.2.2 Software

O *software* foi desenvolvido na *App Inventor2 for Android*, como referido anteriormente. O *App Inventor2* é dividido em duas partes, o *App Inventor Designer* e o *App Inventor Blocks Editor*. O *App Inventor Designer* tem como objetivo desenhar a interface gráfica da aplicação: escolher a configuração e a posição dos ícones, botões, imagens, *droplists*, *checkboxes*, texto, etc. A figura 4.23 apresenta o *layout* do *App Inventor Designer*. [26]

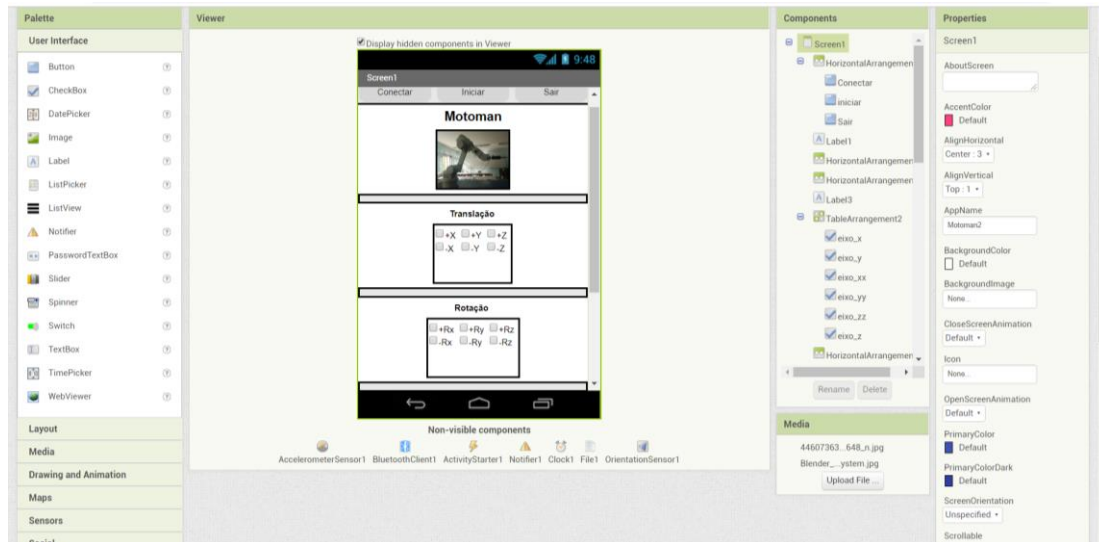


Figura 4.23 - Layout do App Inventor Designer.

A segunda parte, o *App Inventor Blocks Editor* é a secção onde se programam os ícones, botões, *slider*, *checkboxes*, etc, colocadas na *App Inventor Designer*. É apresentada por blocos com diferentes funções, que se interligam através de condições, estados ou variáveis para formar algoritmos. É uma programação onde apenas as funções compatíveis se encaixam. [26] O passo de rotação e translação é igual à utilizada na interface cubo. As funções são ativadas consoante o ícone que é selecionado na interface gráfica. A programação baseia-se: na função que lê as redes Bluetooth, na função do acelerómetro (função pré-existente no *App Inventor*), na função de abrir e fechar garra, na função que cria uma página de texto para gravar as posições, na função da velocidade e nas funções que habilitam e desabilitam cada ícone. A figura 4.24 apresenta o *layout* do *App Inventor Blocks Editor*. [26]

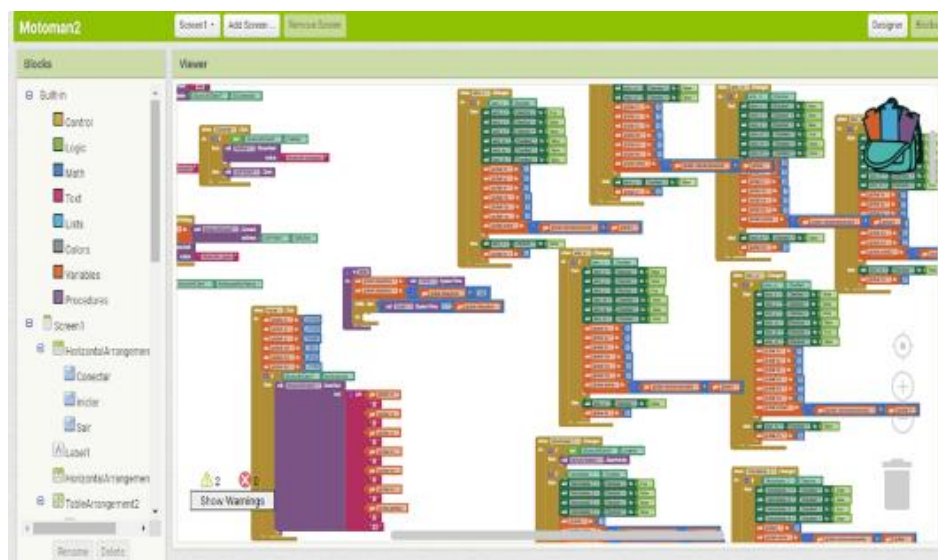


Figura 4.24 - Layout do App Inventor Blocks Editor.

### 4.2.3 Modo de utilização

Este ponto 4.2.3, modo de utilização, pretende explicar de forma detalhada o modo de funcionamento da interface telemóvel. A aplicação Motoman para sistema operativo *Android* encontra-se representada na figura 4.25.

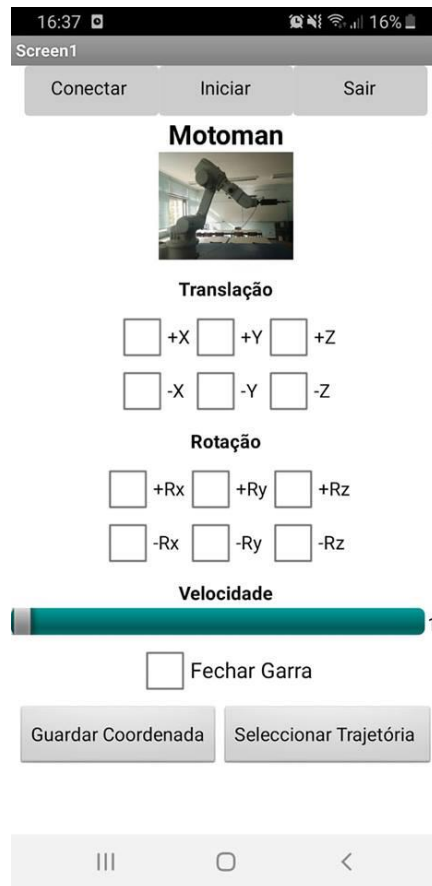


Figura 4.25 - Interface gráfica da aplicação Motoman.

O primeiro passo para se utilizar a aplicação é estabelecer comunicação com o robô. A comunicação via Bluetooth é feita através do ícone “Conectar”. Ao selecionar este ícone é disponibilizada uma lista de redes sem fio. É necessário selecionar a rede do adaptador série Bluetooth-RS232 e colocar a sua palavra-passe, presente no conversor. A conexão é estabelecida quando aparecer no ecrã a mensagem “Bluetooth Ligado”. Seguidamente, deve-se selecionar o ícone “Iniciar” para colocar o robô na sua posição inicial. O ícone “Sair” finaliza a qualquer momento a interação. Quando iniciada, existe uma série de ícones que correspondem aos eixos que se pretende mover, divididos em translação e rotação. Dependendo do eixo selecionado, o utilizador deve agitar o telemóvel na direção do mesmo, daí ser importante saber-se a sua configuração dos eixos cartesianos. Apenas pode ser escolhido um eixo de cada vez, e para alterá-lo o utilizador deve selecionar outro ícone. A velocidade dos movimentos altera-se através do slide “Velocidade” que por defeito é iniciado na velocidade mais baixa (1), mas que pode ser

preenchido até ao valor 2 para duplicá-la, valor 3 para triplicá-la ou valor 4 para quadruplicá-la. Para fechar a garra do Motoman seleciona-se o ícone “Fechar Garra”, esta por defeito encontra-se aberta. O ícone “Guardar Coordenada” possibilita guardar as posições do robô. Todas as posições são guardadas num documento de texto (.txt) e para as reproduzir basta carregar no ícone “Selecionar Trajetória”. A presente interface não tem limite de posições guardadas. Quando se carregar no ícone “Iniciar”, a trajetória é apagada.

### 4.3 Interface Joystick

A interface joystick, apresentada na figura 4.26, é uma interface desenvolvida ao longo dos anos letivos pelos alunos da cadeira de Sistemas Robóticos do curso de Engenharia Eletrotécnica do Instituto Superior de Engenharia de Coimbra (ISEC) e foi adaptada e melhorada para o presente projeto.

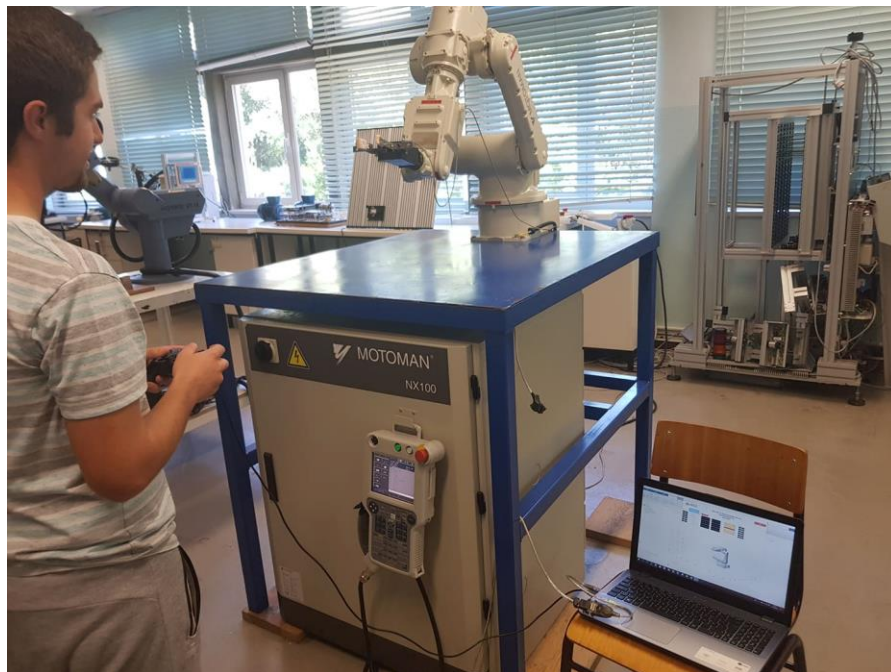


Figura 4.26 - Interação entre o utilizador e o robô com a interface joystick.

A adaptação deveu-se ao joystick ser um dispositivo leve e pequeno, de fácil manipulação, muito utilizado em videojogos e no controlo e monitorização de diferentes máquinas.

A interface consiste na utilização desse objeto para controlar o robô HP3L da Motoman, com recurso ao *software* matlab. Existem duas opções de interação, a nível físico e a nível virtual. O modo virtual indica interagir em modo *offline*, ou seja, manipular o robô através de uma interface gráfica, sem estar fisicamente presente, à semelhança do que oferece o *software* Motosim. O modo físico permite controlar o robô ao enviar as instruções por porta série RS232. Uma particularidade deste modo é poder visualizar-se

graficamente os movimentos que ocorrem em tempo real, ou por outras palavras, o controlo físico é também reproduzido no ambiente virtual. Existe assim, uma comunicação bidirecional para troca de diferentes informações. Uma delas, desenvolvida neste trabalho, é o robô converter as coordenadas cartesianas recebidas em ângulos e retorna-las na interface gráfica. Isto ocorre devido à sua cinemática inversa interna e aos cálculos presentes no arquivo de trabalho. A comunicação pode ser efetuada por cabo RS232 ou por adaptador série Bluetooth-RS232.

Além das opções referidas, a interface tem a opção de abrir ou fechar a garra, e foi também desenvolvido as opções de guardar posições, limpar e reproduzir trajetórias. A figura 4.27 apresenta o *layout* da interface joystick e a figura 4.28 apresenta o *layout* da mensagem enviada por Bluetooth.

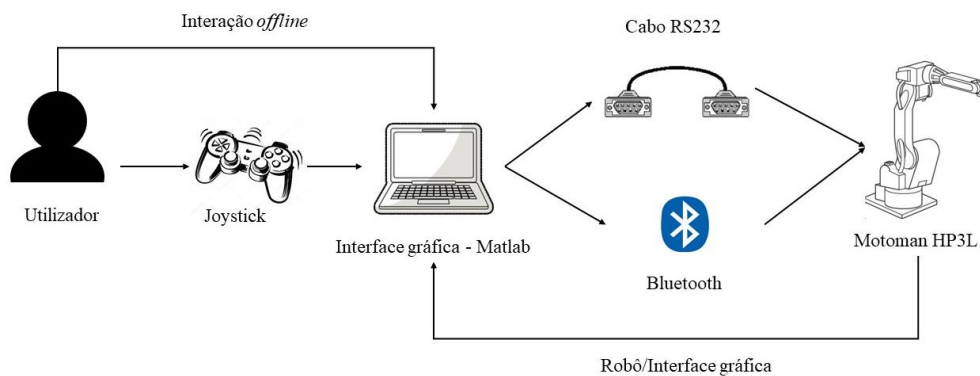


Figura 4.27 - *Layout* da interface joystick.

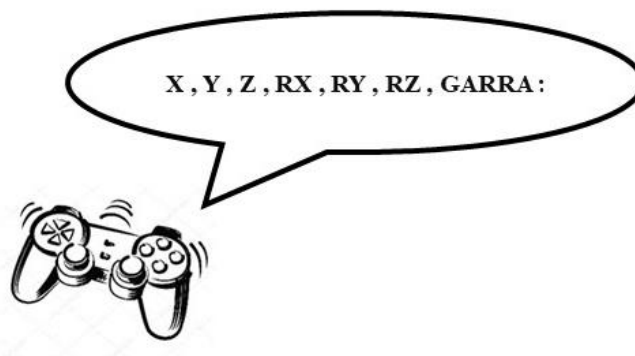


Figura 4.28 - Mensagem enviada pela interface joystick.

#### 4.3.1 Hardware

O *hardware* desta interface consiste no joystick, na interface gráfica/PC e nos diferentes conversores. A figura 4.29 apresenta a sequência dos componentes necessários para a interação.

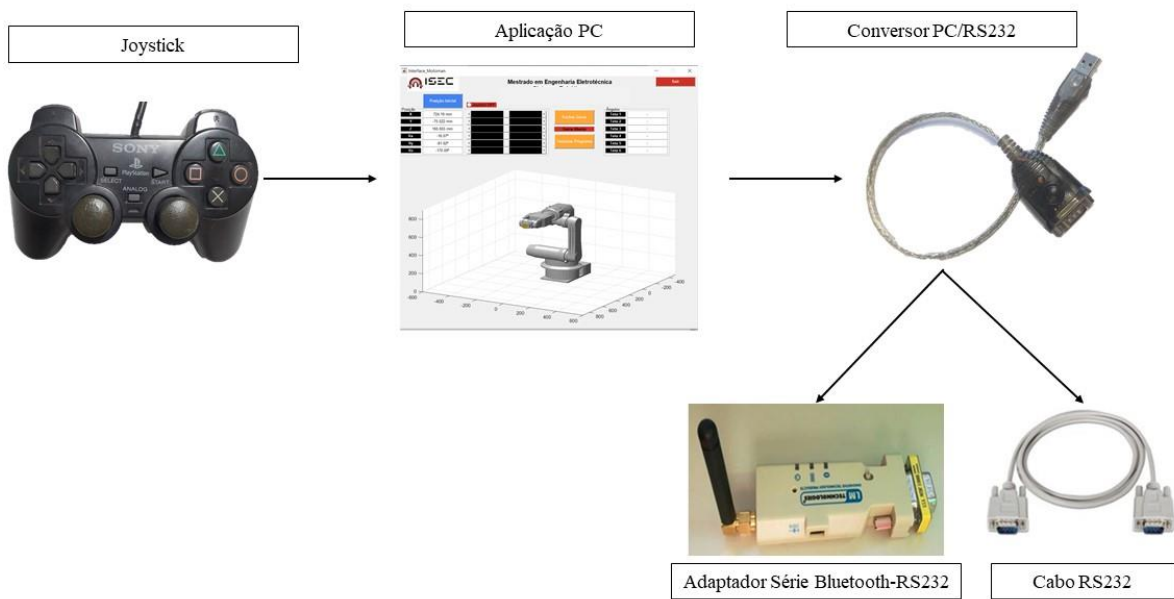


Figura 4.29 - Hardware utilizado para a interface joystick.

O joystick da *PlayStation* da *Sony*, possui catorze botões e dois manipuladores analógicos. A figura 4.30 e a figura 4.31 ilustram o joystick e a função dada a cada botão e seus manipuladores.



Figura 4.30 - Caracterização da função dos seus manipuladores e botões do joystick.



Figura 4.31 - Caracterização da função dos botões do joystick

A orientação dos eixos é essencial para uma boa interação. A figura 4.32 apresenta a configuração e o sentido dos eixos cartesianos no joystick.



Figura 4.32 - Sistema de coordenadas cartesianas no joystick.

#### 4.3.2 Software

A interface joystick foi desenvolvida no *software* Matlab, como indicado anteriormente. Os valores do passo de rotação e translação são os mesmos das restantes interfaces. No Matlab está presente a cinemática direta do robô deduzida na secção 3.3 para permitir interligar os elos do robô virtual e assim move-lo, as funções de construção e envio das mensagens por RS232, as funções de conversão entre os dois modos, as funções de abrir e fechar garra, a função de guardar posições, limpar e reproduzir trajetórias e o código do ambiente gráfico, a *GUI (Graphical User Interface)*. Na programação a porta série está configurada na COM 6 pelo que esta deve ser alterada consoante a porta que se utilize.

A *GUI* apresenta um robô semelhante ao HP3L da Motoman e um conjunto de blocos com diferentes funções. Possui o bloco de “Posição Inicial” para colocar o Motoman na sua posição inicial, o bloco “Fechar Garra”, para fechar ou abrir a garra, o bloco “Terminal Programa” para finalizar a interação, e a opção de ligar o joystick. É esta

última que permite a permutação entre o modo físico e o modo virtual. No canto esquerdo do ambiente gráfico encontram-se os slides de cada eixo, com indicação do seu valor. No canto direito encontra-se indicado o valor dos ângulos retribuídos pelo robô. Os valores Teta1, Teta2, Teta3, Teta4, Teta5, Teta6 correspondem à conversão, falada anteriormente, às coordenadas X, Y, Z, Rx, Ry, Rz presentes no canto direito. A figura 4.33 apresenta a GUI mencionada.

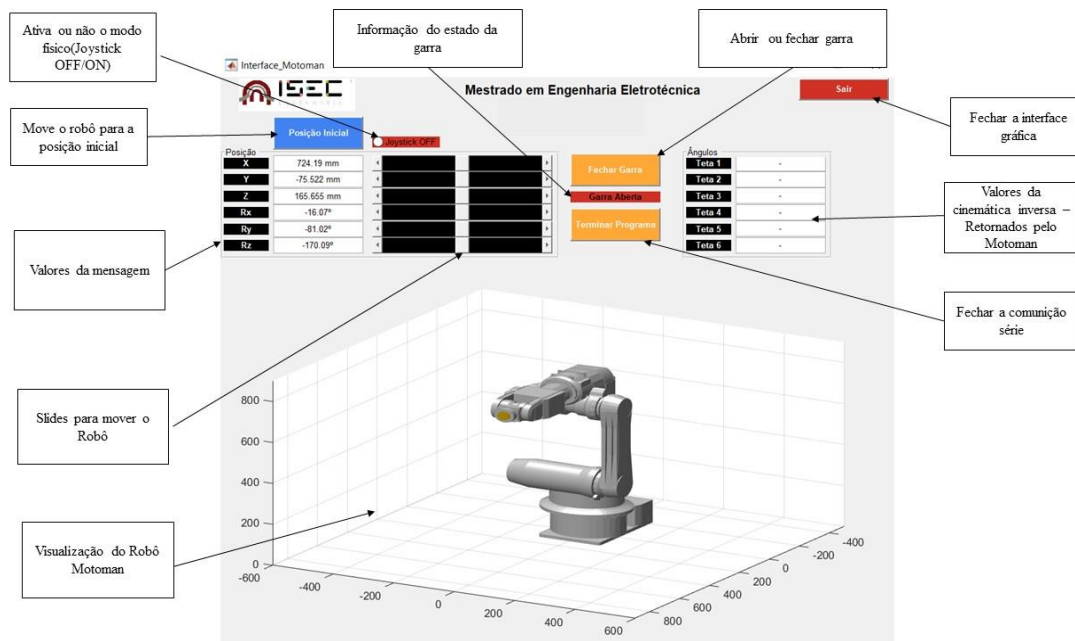


Figura 4.33 - GUI da interface joystick.

### 4.3.3 Modo de utilização

Este ponto 4.2.3, modo de utilização, pretende explicar de forma detalhada o modo de funcionamento da interface joystick.

Os primeiros passos para se utilizar a interface, é abrir o Matlab, ligar o joystick ao Pc e ordenar o início do programa. Com esta ação a GUI é aberta com o *layout* indicado anteriormente. O utilizador pode escolher interagir com o robô no modo físico ou no modo virtual. Se o utilizador desejar utilizar o modo físico deve colocar na COM o conversor PC/RS232, e o cabo RS232 ou o conversor Bluetooth, e seleccionar a opção “Joystick OFF”, onde a label mudará para “Joystick ON”, estando assim todas as condições reunidas para enviar as ações por porta RS232. As translações são ativadas com a pressão constante do botão L2 e com o movimento no manipulador analógico esquerdo para cima ou para baixo para translações em X, para a direita ou para a esquerda para translações em Y e com o movimento do manipulador analógico direito na vertical para translações em Z. Assim como na translação, para efetuar a rotação é preciso

pressionar constantemente o botão R2. Os movimentos dos manipuladores das coordenadas X, Y e Z são iguais para as coordenadas Rx, Ry e Rz, respetivamente.

No joystick estão designados botões de abrir (círculo) e fechar a garra (x), o botão de gravar (triângulo), o botão de reproduzir trajetórias (quadrado) e o botão R1 se a intenção for apagá-las. A interface está habilitada para gravar uma trajetória com posições ilimitadas. Caso o utilizador opte por uma interação *offline* têm à sua disposição as ações da GUI e pode utilizar o joystick para mover o robô virtual ou alterar manualmente os slides.

## 5 Testes de Usabilidade e Resultados

Os testes de usabilidade são testes que pretendem quantificar e qualificar a interação entre um utilizador e um certo produto ou serviço. É maioritariamente aplicado na avaliação de *softwares*, *hardwares*, *websites* e documentos. Representam um elevado contributo no desenvolvimento de produtos uma vez que permitem detetar em fase de desenvolvimento problemas de usabilidade, para que sejam corrigidos e melhorados até ao seu lançamento final e produção em massa. Assim, os testes de usabilidade costumam ser aplicados em protótipos com diferentes detalhes e acabamentos, para se chegar a versões mais detalhadas e de alta fidelidade. Um típico teste consiste em fornecer tarefas a um/vários utilizador(es) para executar as ações que o produto final fará, observar e analisar os resultados do questionário, para destacar os problemas encontrados pelo utilizador, opiniões, pontos fracos e pontos fortes, entre outros. Quanto maior a amostragem dos testes mais se consegue criar produtos bem-sucedidos. [27]

### 5.1 Teste de Usabilidade

O modelo adaptado para realizar os testes de usabilidade é o modelo *Usefulness, Satisfaction and Ease of use (USE)*. Este modelo baseia-se na avaliação de quatro parâmetros: a usabilidade, a facilidade de utilização, a facilidade de aprendizagem e a satisfação. Estes parâmetros são classificados numa escala de 1 a 7, onde 1 representa a escala inferior, discordo completamente, e o 7 representa a escala superior, concordo completamente. [28]

A aplicação de cada teste baseou-se em 5 procedimentos:

1. Apresentação das diferentes interfaces e explicação do modo de funcionamento de cada uma a cada participante;
2. Explicação dos testes de usabilidade;
3. Definição da tarefa a executar pelo participante: utilizar a interface de modo a manipular o robô, e deslocar a peça de um local A para um local B, gravar e reproduzir esse deslocamento, adequar a velocidade;
4. Preenchimento do teste pelo participante;
5. Avaliação geral de cada interface, apontamento de sugestões e melhorias;

O questionário do modelo *Usefulness, Satisfaction and Ease of use* encontra-se representado na figura 5.1.

Interface: \_\_\_\_\_

		Discordo plenamen te						Concordo plenamen te	
	<b>Usabilidade</b>	1	2	3	4	5	6	7	NA
1	Ajuda-me a ser mais eficaz.								
2	Ajuda-me a ser mais produtivo.								
3	É útil.								
4	Permite-me um melhor controlo das atividades da minha vida.								
5	Torna mais fácil concretizar as coisas que pretendo.								
6	Poupa-me tempo quando o uso.								
7	Satisfaz as minhas necessidades.								
8	Faz o que esperaria que fizesse.								
	<b>Facilidade de Utilização</b>	1	2	3	4	5	6	7	NA
9	É fácil de usar.								
10	É simples de usar.								
11	É intuitivo.								
12	Requer o mínimo possível de passos para alcançar o objetivo.								
13	É flexível.								
14	O seu uso não requer esforço.								
15	Conseguo utiliza-lo sem instruções.								
16	Não há nenhuma inconsistência quando o uso.								
17	Utilizadores regulares ou ocasionais irão gostar dele.								
18	Conseguo corrigir erros de forma rápida e fácil.								
19	Conseguo utilizar sempre com sucesso								
	<b>Facilidade de Aprendizagem</b>	1	2	3	4	5	6	7	NA
20	Aprendi rapidamente a usá-lo.								
21	Lembro-me facilmente quando o volto a usar.								
22	É fácil aprender a usá-lo.								
23	Conseguo usá-lo todas as vezes com sucesso.								
	<b>Satisfação</b>	1	2	3	4	5	6	7	NA
24	Estou satisfeito em utilizá-lo.								
25	Eu recomendaria-o a um amigo.								
26	É divertido de usar-se.								
27	Funciona como eu quero que funcione.								
28	É incrível.								
29	Gostava de o ter.								
30	É agradável de usar.								

Figura 5.1 - Modelo *Usefulness, Satisfaction and Ease of use*. [29]

Este modelo é composto por trinta perguntas, a serem respondidas de acordo com uma escala de classificação de um a sete pontos, que vão desde “discordo fortemente” a “concordo plenamente”, respetivamente, e N/A “Não Aplicável”. [28]

No presente projeto foram aplicados 4 testes de usabilidade, um por cada interface, a cada aluno, num total de 18 estudantes do Instituto Superior de Engenharia de Coimbra (ISEC). A interface cubo e interface joystick foram utilizadas o cabo RS232.

## 5.2 Resultados dos Testes de Usabilidade

O resultado da aplicação dos testes, num total de 72, aos 18 alunos do ISEC, visualiza-se na figura 5.2, onde é apresentada a média dos valores da escala de classificação de cada pergunta, divididas em usabilidade (1), facilidade utilização (2), facilidade de aprendizagem (3) e satisfação (4).

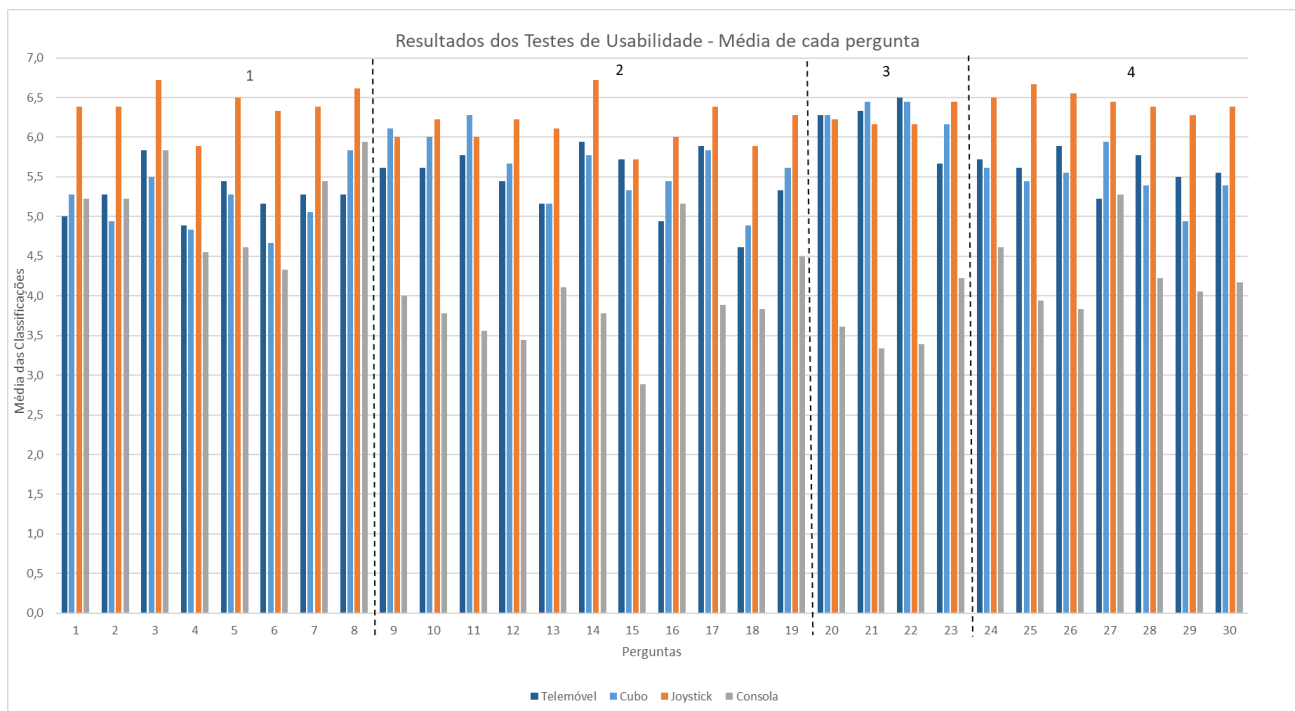


Figura 5.2 - Gráfico dos Resultados dos Testes de Usabilidade por média de cada pergunta.

Ao analisar o gráfico da figura 5.2 a primeira evidência dos resultados é a baixa classificação da consola na maioria das perguntas.

A pergunta 15 “Consigno utiliza-lo sem instruções” com média de 2,9, a pergunta 21 “Lembro-me facilmente quando o volto a utilizar” com média de 3,3, a pergunta 14 “O seu uso não requer esforço” com média de 3,8 e a pergunta 22 “É fácil aprender a usa-lo” com média de 3,4 apresentam-se como as perguntas com mais baixas classificações que permitem aferir e confirmar os problemas apresentadas anteriormente na consola como, conter muitas instruções, ser necessário manual de instruções, ter muitos ícones e teclas para monitorizar o robô, haver necessidade de noções de programação *INFORM* e ter que se pressionar várias teclas ao mesmo tempo.

A interface cubo registou melhor classificação nas perguntas 9, 10 e 11 e nas perguntas 20, 21 e 22. Através destes resultados pode-se afirmar que a sua forma cúbica,

a marcação da função de todos os botões e a forma como se encontram distribuídos ajudam o utilizador a ter uma interação intuitiva e com facilidade de aprendizagem. O resultado mais baixo verifica-se na pergunta 6 “Poupa-me tempo quando o uso”, com média de 4,7, isto dever-se ao seu *hardware* ser constituído por botões de pressão ON/OFF que torna os movimentos do robô mais demorados uma vez que o sinal é enviado apenas no fecho do contacto. Consequentemente a esta análise, os valores das perguntas 1 “Ajuda-me a ser mais eficaz” e da pergunta 2 “Ajuda-me a ser mais produtivo”, apresentam-se também baixos. Outro aspeto a salientar é o formato do prolongamento da interface cubo, que se apresenta como um paralelepípedo podendo dificultar a colocação da mão do utilizador. Este ponto fraco é justificado pela pergunta 13 “é flexível”, a pergunta 30 “É agradável de usar” e pela pergunta 29 “Gostava de o ter” ambas com classificações abaixo das outras interfaces.

A interface telemóvel registou o valor mais baixo na pergunta 18 “Consigo corrigir erros de forma rápida e fácil” e na pergunta 4 “Permite-me um melhor controlo das atividades da minha vida” ambas com média de 4,9 isto justifica-se uma vez que se verificou que o acelerómetro é muito sensível à mudança do seu estado o que leva a movimentos não desejados pelo utilizador aquando de um descuido. Já a melhor classificação verificou-se na pergunta 22 “é fácil aprender a usá-lo”, em parte porque o telemóvel é um objeto utilizado diariamente.

A interface Joystick revelou ser a melhor classificada. Os valores mais altos verificaram-se na pergunta 3 “É útil”, na pergunta 14 “O seu uso não requer esforço” e na pergunta 25 “Recomendaria a um amigo” todas com média de 6,7, os valores mais baixos registam-se nas perguntas de melhor classificação das interfaces cubo e telemóvel.

Além dos resultados por média de cada pergunta, foram também calculados a média de cada parâmetro: usabilidade, facilidade de utilização, facilidade de aprendizagem e satisfação. Os resultados encontram-se representados na figura 5.3.

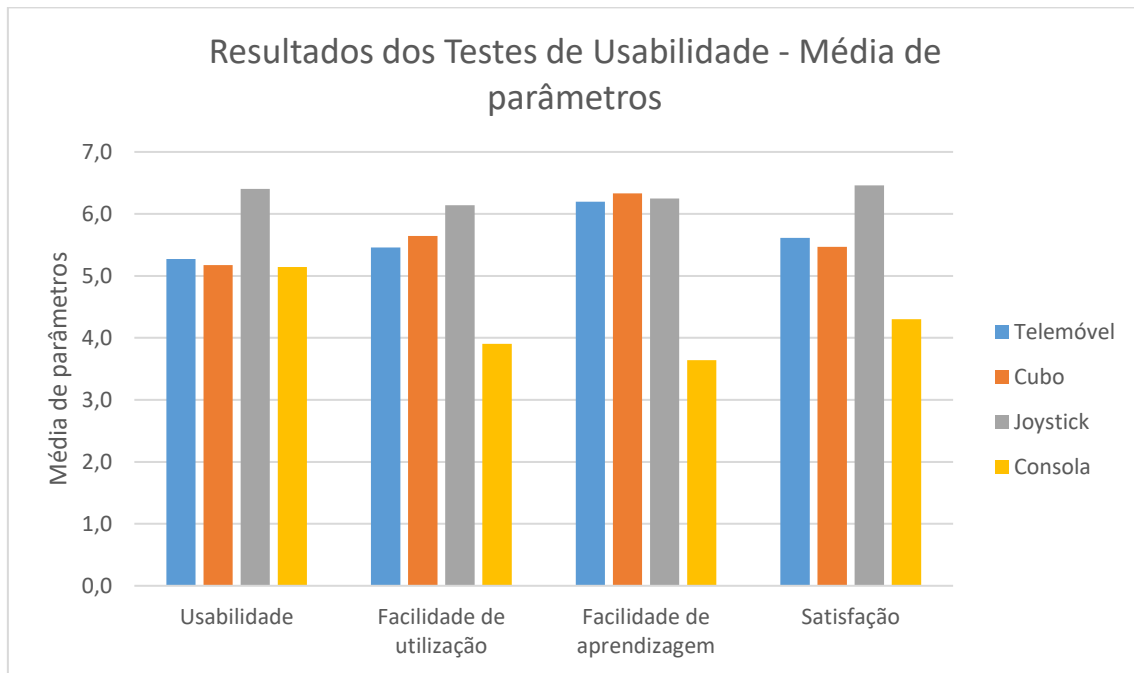


Figura 5.3 - Gráfico dos Resultados dos Testes de Usabilidade por média de parâmetros.

Ao visualizar o gráfico da figura 5.3, conclui-se que o joystick é a interface com mais usabilidade, facilidade de utilização e satisfação. Já o cubo é a interface que apresenta maior facilidade na aprendizagem. Assim, o joystick é o objeto que mais facilita e simplifica a concretização de uma tarefa, o objeto mais fácil de utilizar pelo seu design e capacidade de responder a eventos não esperados e o objeto que durante a sua utilização gera maior satisfação nos utilizadores. O cubo é a interface que tem as características ideais para ensinar o utilizador a manipular o robô.

## 6 Conclusões e trabalho futuro

Um tema atual e em ascensão nas indústrias onde ao longo do seu desenvolvimento é percebido a importância do mesmo para a obtenção de uma boa, fácil e simples colaboração entre os homens e os robôs, assim como a necessidade de explorar e dar relevância a parâmetros como os mencionados neste projeto: usabilidade, facilidade de utilização, facilidade de aprendizagem e satisfação das interfaces, que muitas vezes são ignoradas pelos fabricantes de máquinas e equipamentos e pelos gestores de topo. O desenvolvimento do presente projeto mostrou ser um grande desafio para a autora. A familiarização com o robô HP3L com controlador NX100 da Motoman e a sua consola apresentou-se como o primeiro desafio, seguindo-se a elaboração de interfaces com as mesmas funções e a aplicação dos testes de usabilidade. Este processo requereu a leitura de manuais do robô, a utilização e desenvolvimento de várias linguagens de programação, da escolha e criação de objetos, da seleção dos componentes mais adequados para as interações e a visitas regulares a várias turmas.

Todos os objetivos traçados para o presente projeto foram alcançados. As interfaces descritas realizam todas as funções propostas e de forma clara reúnem condições para simplificar e até substituir a interação entre o utilizador e o robô HP3L. No entanto, através dos resultados atingidos e de opiniões por parte dos utilizadores, evidenciaram-se melhorias às interfaces.

Na interface cubo poder-se-á modificar o seu *software* para quando da pressão instantânea de um botão o movimento do robô se mantenha constante até uma segunda pressão no botão que o fará parar, prevendo-se que esta modificação poupe tempo na execução da tarefa. O prolongamento do cubo, que se apresenta em formato de paralelepípedo, ser substituído por um prolongamento em formato de cilindro, formato esse que possibilitará maior conforto para a mão do utilizador.

A interface telemóvel poderá ser melhorada através da colocação de um ícone que permita parar, a qualquer momento, o movimento do acelerómetro do telemóvel naquele eixo, uma vez que este mostrou-se muito sensível na utilização descuidada do objeto. Outra solução seria a utilização conjunta do giroscópio com o acelerómetro.

Verificou-se que a interface joystick é a interface que satisfaz mais parâmetros na interação com o robô, em parte por ser constituído por manipuladores com eixos livres que possibilita uma permutação rápida entre os mesmos, por ser agradável na sua utilização e também ergonómico. Não esquecendo que quase todos os utilizadores já tinham manipulado uma vez na sua vida um objeto igual. Contudo, verifica-se como melhoria que cada manipulador devia ter identificado o sistema de eixos aplicado.

Como trabalho futuro é proposto efetuar as melhorias nas interfaces e aplicar novamente os testes de usabilidade para se avaliar a influência das mesmas. Se se verificar

resultados equivalentes é proposto colocar os manipuladores do joystick, eleitos como um sistema útil para a manipulação do robô, como *hardwares* nas interfaces cubo e telemóvel, potencializando-as. Possibilita assim, juntar as melhores características de todas as interfaces.

Por fim, salienta-se que o presente projeto permitiu à aluna enriquecer os seus conhecimentos, aumentar o seu espírito crítico, a sua autonomia, a resolução de problemas, interatuar com diferentes conceitos e melhorar os seus conhecimentos de programação. A aluna também se sente orgulhosa pelo desenvolvimento do projeto, pelos objetivos alcançados, pelas conclusões retiradas e por ter enriquecido o seu saber na área da robótica industrial. Deste relatório de projeto foi também possível à aluna redigir um artigo científico, para a sua submissão a um congresso científico internacional da área.

## Bibliografia

- [1] Santos, A. P. (9 de Fevereiro de 2012). *A importância da interação Humano-Computador*. Obtido de TIqx: <http://tiqx.blogspot.com/2012/02/compreenda-importancia-da-interacao.html>. Consultado em:6/03/2018
- [2] Scholtz, J. (2003). Theory and evaluation of human robot interactions. *36th Annual Hawaii International Conference on System Sciences, 2003.* . USA.
- [3] ISO. (s.d.). *Ergonomics of human-system interaction*. Organisation Internationale de Normalisation.
- [4] Ford, M. (2016). *Robôs e a ameaça de um futuro sem emprego*. Bertrand Editora.
- [5] FFonseca. (29 de Agosto de 2017). *Soluções de segurança para colaboração inteligente homem-robô*. Obtido de FFonseca: <https://www.ffonseca.com/pt/novidades-solucoes-de-seguranca-para-colaboracao-inteligente-homem-roboto>. Consultado:21/04/2019.
- [6] *A Indústria 4.0 e a remodelação da interação homem-máquina em análise*. (25 de Janeiro de 2018). Obtido de Centro de Computação Gráfica: <http://www.ccg.pt/industria4-0-hmi/>. Consultado em:20/04/2018
- [7] Robots. (s.d.). *Unimate*. Obtido de Robots: <https://robots.ieee.org/robots/unimate/>. Consultado em:15/06/2018
- [8] Pires, J. (2017). Os desafios da robótica industrial. *Revista Robótica n°83*, 48-49-50 51-52.
- [9] DEMAITRE, E. (12 de Fevereiro de 2018). *Yaskawa deep learning bin picking robot at iREX 2017*. Obtido de Rbr: <https://www.roboticsbusinessreview.com/manufacturing/deep-learning-factory-automation/attachment/yaskawa-deep-learning-bin-picking-robot-at-irex-2017/>. Consultado em:10/11/2018
- [10] Machado, L. S. (Novembro de 2010). *Dispositivo Háptico PHANToM Omni da SensAble Technologies Inc.* . Obtido de ResearchGate: <https://www.researchgate.net/figure/Figura-1-Dispositivo-Haptico-PHANToM-Omni-da-SensAble-Technologies-Inc-fig1-228579672>. Consultado em: 10/11/2018
- [11] Cancela, R. (2007). *Extensão e flexibilização da interface de controlo*.
- [12] Mecalux. (s.d.). Consola (Motoman DX100). Obtido de Mecalux: <https://www.logismarket.pt/yaskawa-iberica/consola/1790745897-1792160439-p.html> Consultado em:15/06/2018
- [13] Mikaelsson, M. C. (2010). O novo robô irb 120 da abb e o seu controlador irc5 compact.
- [14] Latta, J., & Oberg, D. (Janeiro de 1994). *A conceptual virtual reality model*.
- [15] HoloLens, M. (7 de Março de 2019). *Integrar a Realidade Virtual na programação de robôs industriais*.

- [16] *A Indústria 4.0 e a remodelação da interação homem-máquina em análise*. (25 de Janeiro de 2018). Obtido de Centro de Computação Gráfica: <http://www.ccg.pt/industria4-0-hmi/>. Consultado em: 14/01/2019
- [17] RobotWork. (s.d.). Obtido de Motoman HP3: <https://www.robots.com/robots/motoman-hp3>. Consultado em: 8/04/2018
- [18] Incorporated, M. (2007). *Motoman NX100 Controller - HP3 Manipulator Manual*.
- [19] HW0481924, Y. (s.d.). *NX100 INSTRUCTION for Relative Job Functions*.
- [20] Nykamp, D. Q. (s.d.). *Cartesian coordinates*. Obtido de Math Insight: [https://mathinsight.org/cartesian\\_coordinates](https://mathinsight.org/cartesian_coordinates). Consultado em: 15/05/2018
- [21] J. Ferreira (2018). Apontamentos de Sistemas Robóticos do DEE/ISEC
- [22] Yaskawa. (s.d.). *MOTOSim EG-VRC (Powerful Simulation Software)*.
- [23] Motoman. (2013). MOTOMAN NX100 BASIC PROGRAMMING.
- [24] Souza, F. (21 de Janeiro de 2014). *Arduino - Comunicação Serial*. Obtido de Embarcados: <https://www.embarcados.com.br/arduino-comunicacao-serial/>. Consultado em: 20/09/2018
- [25] Arduino. (s.d.). *ARDUINO UNO REV3*. Obtido de Arduino: <https://store.arduino.cc/arduino-uno-rev3>. Consultado em: 19/06/2018
- [26] Cordeiro, F. (s.d.). *App Inventor: Guia de Criação de Apps*. Obtido de Android Pro: <https://www.androidpro.com.br/blog/desenvolvimento-android/app-inventor/>. Consultado em: 23/03/2019
- [27] Usability Testing Obtido de: <https://www.interaction-design.org/literature/topics/usability-testing> Consultado em: 23/07/2019
- [28] Lund, A. (. (2001). *Measuring Usability with the USE Questionnaire: Usefulness, Satisfaction, and Ease of use*.

## Anexos

**Anexo I - Arquivo de trabalho**

```
/JOB
//NAME SRMEE18
//POS
///NPOS 0,0,0,3,0,0
///TOOL 0
///POSTYPE BASE
///RECTAN
P0020=611.000,-62.000,488.000,-17.00,-80.00,-173.00 //Posição Inicial(P0020)
//INST
///DATE 2018/06/14 20:19
///ATTR SC,RW
///GROUP1 RB1
NOP
MOVJ P020 VJ=5.00
ASC B000 ","
ASC B001 ":"
ASC B005 "f"
SET D001 1
OPEN I000 "COM1:9600N81"
SET B003 1
SET B004 1
SET S001 ""
*LOOP
SET B010 0
SET B011 0
SET B012 0
INPUTB# I000 B002
JUMP *END IF B002=B005
JUMP *NEWVAL IF B002=B000
JUMP *NEWSTR IF B002=B001
SET B007 B002
CHR$ S000 B002 S000
CAT$ S001 S001 S000
```

---

```
JUMP *LOOP
*NEWVAL
VAL R000 S001 R000
SET D000 R000
SETE P010 (B003) D000 D000
INC B003 P010
SET S001 ""
JUMP *LOOP
*NEWSTR
MOVJ P010 VJ=10.00
GETS PX011 $PX000
SLURBT
SET B004 1
*ENVAL
GETE D011 P011 (B004)
SET B023 200
JUMP *NEG IF D011 >= 0
MUL D011 -1
SET B023 255
*NEG
SET D021 D011
SET D020 D021
DIV D020 100
SET D022 D020
DIV D020 100
SET D023 D020
SET D020 D022
SET D026 D023
MUL D026 100
SUB D020 D026
SET D024 D020
MUL D022 100
SUB D021 D022
SET B020 D023
SET B021 D024
```

```
SET B022 D021
INC B004
OUTPUT# I000 B023
OUTPUT# I000 B020
OUTPUT# I000 B021
OUTPUT# I000 B022
JUMP *ENVAL IF B004<=6
VAL R000 S001
SET D000 R000
SET B003 1
SET S001 ""
JUMP *GARRAC IF D000=D001
DOUT OT#(1) ON
JUMP *FIN
*GARRAC
DOUT OT#(1) OFF
*FIN
JUMP *LOOP3 IF IN#(3)=OFF
SET B010 1
*LOOP3
JUMP *LOOP5 IF IN#(5)=OFF
SET B011 1
*LOOP5
JUMP *LOOP7 IF IN#(7)=OFF
SET B012 1
*LOOP7
TIMER T=0.01
JUMP *LOOP
*END
CLOSE I000
END
```