

UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC

CENTRO DE EDUCAÇÃO DO PLANALTO NORTE - CEPLAN

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ANDRE SHADWICK PETERSEN

**PROTÓTIPO DE INTERFACE DE CONTROLE PARA
ROBÔS OU SISTEMAS DE AUTOMAÇÃO.**

SÃO BENTO DO SUL

2015

ANDRE SHADWICK PETERSEN

**PROTÓTIPO DE INTERFACE DE CONTROLE PARA ROBÔS OU SISTEMAS DE
AUTOMAÇÃO.**

Trabalho de conclusão de curso apresentado ao curso de Bacharelado em Sistemas de Informação como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Me. Alex Luiz de Souza.

SÃO BENTO DO SUL

2015



UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE EDUCAÇÃO DO PLANALTO NORTE – CEPLAN
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO Nº 66

Ata de defesa do Trabalho de Conclusão de Curso realizada em 01 de dezembro de 2015.

Acadêmico: **André Shadwick Petersen**

Orientação: Alex Luiz de Sousa

Banca Examinadora: Alex Luiz de Sousa, Me.

Luiz Cláudio Dalmolin, Dr.

Mário Ezequiel Augusto, Dr.

Título do Trabalho de Conclusão de Curso: **Protótipo de Interface de Controle para Robôs ou Sistemas de Automação**

Hora de Início: 16:00 h.

Local: UDESC / CEPLAN

Sala: 1ª fase BSI

Após submeter o Trabalho de Conclusão de Curso em forma escrita, o aluno efetuou a exposição oral, de cerca de 20 minutos, perante a banca examinadora. Logo após, os membros da banca efetuaram uma arguição oral. Terminada a arguição, a banca se reuniu e o Trabalho de Conclusão de Curso foi:

- Aprovado**
 Aprovado com restrições
 Reprovado

Terminada a sessão, foi lavrada a presente ata que é assinada pelos membros da Banca e pelo acadêmico.

São Bento do Sul, 01 de dezembro de 2015.

Acadêmico:

André Shadwick Petersen

Banca:

Alex Luiz de Sousa, Me.

Luiz Cláudio Dalmolin, Dr.

Mário Ezequiel Augusto, Dr.

RESUMO

O presente estudo demonstra a importância da criação de interfaces de controle de robôs ou sistemas automatizados. A robótica de modo geral busca trazer maior comodidade e praticidade para o dia a dia das pessoas, tanto em residências quanto no meio industrial. Pode-se dizer que interfaces de controle de mecanismos robóticos estão em alta no mercado e devem ser cada vez mais aperfeiçoados visando uma maior praticidade e segurança na hora de utilizar um robô ou sistema automatizado. É de suma importância o entendimento da interação entre o ser humano e a máquina, para que assim possa-se atingir o objetivo principal. Vale ressaltar que o projeto como um todo teve um foco muito grande no baixo custo e na otimização de código, sempre buscando obter processamento máximo dos computadores. O desenvolvimento do projeto foi feito com base em placas de prototipagem, bem como linguagens de programação de baixo e alto nível, além de conceitos fundamentais da robótica.

Palavras-chave: Robótica, Sistemas Automatizados, Interface homem-máquina, Interfaces de Controle.

ABSTRACT

The present study demonstrates the importance of control interfaces creation for robots or automatized systems. Robotics in general terms tries to bring among us more comfort, in residences and in the industrial sector. Robotics control interfaces are growing up at the market and looking for comfort and security when someone uses a robot or automatized system. The understanding of interaction of the human being and computers is fundamental to accomplish the main objective. It is important to register that the whole project focused in low costs and code optimization, always looking for the best results in terms of computer processing. The development of this project was done based on prototype boards, high and low level programming languages, besides fundamental concepts of robotics.

Key-Words: *Robotics, Automatized Systems, Human-machine Interface, Control Interfaces.*

SUMÁRIO

1 INTRODUÇÃO	8
1.1 MOTIVAÇÃO E JUSTIFICATIVA	9
1.2 OBJETIVO GERAL	10
1.3 OBJETIVOS ESPECÍFICOS	10
1.4 CONTRIBUIÇÕES	10
1.5 ORGANIZAÇÃO DO DOCUMENTO	11
2 REFERENCIAL TEÓRICO	12
2.1 CARACTERIZAÇÃO DO TRABALHO	12
2.2 AUTOMAÇÃO DE SISTEMAS	13
2.2.1 INTERFACE HOMEM-MÁQUINA	13
2.2.2 SISTEMAS HÁPTICOS	14
2.2.3 ROBÓTICA	14
2.3 <i>WEARABLE TECHNOLOGY</i>	15
2.4 PLATAFORMAS DE PROTOTIPAÇÃO	16
2.5 SENSORES	16
2.6 ARDUINO	17
2.7 COPPELIA V-REP	18
2.8 UNITY 3D	19
2.9 <i>DIY (DO IT YOURSELF)</i>	20
2.10 <i>VERMASON, VELOSTAT, LINQSTAT</i>	20
2.11 <i>ROBOTIC GLOVE</i>	21
2.12 CONCLUSÕES DO CAPÍTULO	22
3 DESENVOLVIMENTO DA INTERFACE	23
3.1 MONTAGEM E PROTOTIPAÇÃO	23
3.1.1 ETAPA 1 – PESQUISA	23
3.1.2 ETAPA 2 – PRIMEIRAS TENTATIVAS	24
3.1.3 ETAPA 3 – INVESTIMENTOS	25

3.1.4 ETAPA 4 – COPPELIA V-REP E UNITY3D	26
3.1.5 ETAPA 5 – PROGRAMAÇÃO E DESENVOLVIMENTO	27
3.2 CONCLUSÕES DO CAPÍTULO	31
4 VALIDAÇÃO E TESTES.....	33
4.1 CENÁRIO	33
4.2 TABELA DE TESTES	33
4.2.1 RESPOSTA DOS SENSORES E CALIBRAGEM	34
4.2.2 PROCESSAMENTO DO ARDUINO	35
4.2.3 PROCESSAMENTO DO UNITY	35
4.2.4 <i>DESIGN</i> DE INTERFACE.....	35
4.2.5 USABILIDADE DA LUVA “INTELIGENTE”	36
4.2.6 CONSIDERAÇÕES GERAIS DO SISTEMA.....	37
4.2.7 FINALIZAÇÃO DA FASE DE TESTES.....	37
4.8 CONCLUSÃO DO CAPÍTULO	38
5 CONCLUSÃO.....	40
REFERÊNCIAS	41
ANEXOS	44

1 INTRODUÇÃO

A Automação Industrial está diretamente relacionada com a mecanização de processos e pode ser definida como a tecnologia em que um procedimento é concluído sem intervenção humana (GROOVER, 2008) e, de acordo com Rosário (2005), desde o início dos anos 60 os robôs estão sendo cada vez mais utilizados na indústria.

Automação de Sistemas e Robótica são duas áreas que estão relacionadas entre si e, em um contexto amplo, pode-se definir automação como uma tecnologia que utiliza certos recursos (mecânicos, computacionais e eletroeletrônicos) para operação e controle da produção (PAZOS, 2002).

Hoje em dia pode-se dizer que a Robótica não está mais presente somente no setor industrial, já existem robôs atuando em algumas residências e cada vez mais pode-se usufruir da sua capacidade de trabalho para questões do cotidiano, como por exemplo: Cortar grama (Robô Landroid M), aspirar a casa (Deebot), limpar janelas (Winbot) e etc. Essas ações, que por muitas vezes podem ser simples para o ser humano, também podem prejudicar sua saúde ou apresentar perigos para sua integridade física, sendo assim, robôs (controlados ou não por humanos) vem para suprir essa demanda de um serviço bem feito e com o mínimo possível de risco à saúde humana, isso pode ser notado tanto nas residências, quanto nas indústrias.

Algumas situações em que os manipuladores industriais podem substituir o trabalho humano são: situações em que se exige alta repetibilidade, cuidados extremos, situações perigosas, desgaste físico e mental (CARRARA, [19--?]) e ainda, de acordo com Groover (2008), onde houver necessidade de manipulação de peças de difícil manuseio, situações desconfortáveis e na substituição de operadores em mais de um turno (um único robô pode substituir mais de um operador).

De acordo com Laus (2006), na Robótica estão presentes as contribuições das áreas de Mecânica (estudo das máquinas nas situações estáticas e dinâmicas), Matemática (desenvolvimento de ferramentas para descrever os movimentos do robô), Teoria de Controle (ferramentas que avaliam os algoritmos adotados para realização de movimentos e aplicação de forças), Elétrica (sensores e interfaces) e Informática (programar os manipuladores para que desenvolvam a tarefa desejada).

Os braços robóticos são um dos tipos de robos utilizados na indústria, que de acordo com, Braga (2009), devem conter os seguintes elementos: controle (que deve obter as informações de movimentação a partir de um teclado, *joystick* ou um computador), atuador

(deve movimentar o braço e a garra), sensores (para transmitir informações ao operador) e fonte de alimentação.

Os robôs, assim como os braços robóticos, são articulados ou seja, possuem movimento individual de algumas de suas partes, sendo assim, pode-se programá-los individualmente também. A programação é feita de forma pré-definida, e pode ser modificada através de interfaces.

Existem problemas que ocorrem nas instituições onde se faz necessário manipular itens perigosos ou realizar procedimentos de extrema precisão. No atual contexto do mercado, pode-se dizer que existem algumas soluções, porém, estas são limitadas, já que os mecanismos de controle apresentados não são práticos e requerem um treinamento avançado para serem manipulados. Existe também a questão do custo, onde muitas vezes nas soluções presentes tem-se o preço do equipamento em si, treinamento de funcionários que irão utilizar a ferramenta e também existe todo o custo de desenvolvimento, além é claro do custo de manutenção. É importante frizar que no caso do custo do desenvolvimento o que mais encarece é que muitas vezes o projeto precisa ser adaptado ou desenvolvido especificamente para uma determinada situação, isso tudo faz com que o próprio custo de manutenção, já citado, se torne mais alto.

Tendo esse cenário em vista, parte-se da idéia de que a solução existe, só precisa ser melhorada e barateada. A forma mais fácil encontrada para desenvolver algo neste sentido, está na criação de um objeto que seja prático e fácil de usar, e que também possibilitasse o manuseio com precisão e segurança.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

Com a finalidade de intensificar e tornar a interatividade homem-máquina cada vez mais natural e intuitiva, as pesquisas por sistemas hápticos vêm crescendo (CAETANO, 2008). De acordo com Zelek et al. (2003), háptico refere-se ao estudo do toque e das sensações da pele. Com a evolução da automação, esses sistemas estão cada vez mais complexos e tentam representar com perfeição os movimentos humanos. Caetano (2008) afirma que os sistemas hápticos têm sido utilizados em diversas áreas, como por exemplo, médica, militar, aero-espacial, robótica, submarina e exploração em lugares inóspitos.

Neste trabalho, a forma encontrada para desenvolver a interface de utilização e controle é através de *Wearable Technology*, que nada mais é do que tecnologia embarcada em roupas ou acessórios. Neste caso a ideia é projetar uma proposta de interface de controle em forma de luva que tenha a capacidade de manipular um robô de forma simples e intuitiva.

A ideia de criar uma luva “inteligente” veio de encontro com a questão da naturalidade, já que a utilização da interface se torna muito intuitiva tendo em vista que os dedos já fazem parte do corpo humano e são controlados com maior facilidade de precisão pelo cérebro, músculos e articulações. Também tem-se a redução do custo, já que não é necessário a utilização de muitos materiais e não será necessário investir tanto em treinamento e capacitação dos usuários.

1.2 OBJETIVO GERAL

Desenvolver um protótipo de uma interface de controle para robôs ou outros sistemas de automação utilizando os conceitos de *Wearable Technology*, afim de facilitar a manipulação ou controle de itens com maior precisão e segurança.

1.3 OBJETIVOS ESPECÍFICOS

- Avaliar sensores mais adequados ao projeto, para captar as informações do movimento das mãos de uma pessoa;
- Projeto e desenvolvimento da interface de controle para a luva “inteligente”;
- Identificar as formas de comunicação com o braço robótico, bem como suas limitações;
- Analisar programas ou ferramentas que permitam trabalhar com simulações de robôs comumente usados na indústria;
- Efetuar testes e validar a interface de controle proposta através da manipulação de um equipamento real ou virtual.

1.4 CONTRIBUIÇÕES

Além do objetivo geral, espera-se que este trabalho origine uma expressiva contribuição literária para futuros projetos de *hardware* livre, além de incentivar o uso de plataformas de prototipagem livre, linguagens de programação de alto e baixo nível, noções de robótica e também de sistemas hápticos, com outros desígnios no meio acadêmico ou no acréscimo desta pesquisa.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Este trabalho é composto por capítulos. O capítulo um expõe uma introdução que concede uma visão do problema da pesquisa e idealiza seu escopo. O capítulo dois revela o referencial teórico em que este trabalho foi fundamentado. O capítulo três apresenta a forma como o mesmo foi desenvolvido. O capítulo quatro apresenta os testes realizados no protótipo. Por fim, são estabelecidas as considerações finais e sugestões para futuras reproduções.

2 REFERENCIAL TEÓRICO

Este Capítulo apresenta a caracterização do trabalho, bem como a base teórica fundamental para o desenvolvimento do protótipo em questão.

2.1 CARACTERIZAÇÃO DO TRABALHO

O presente trabalho pode se enquadrar na área de Automação de Sistemas por meio de programação em plataformas e prototipação. Também pode ser considerado dentro da área de Robótica. Trabalhando com conceitos de: *Wearable Technology*, Robótica e Automação de Sistemas.

A robótica está cada dia mais presente, por conta disso, depende-se cada vez mais dela, tanto no trabalho quanto em tarefas do nosso cotidiano. Ao longo dos anos, novas tecnologias envolvendo robôs e automação de sistemas foram surgindo e com isso novos problemas para serem resolvidos.

Os robôs industriais podem possuir diversas interfaces de comunicação. Isto se exemplifica da seguinte forma (ROBÓTICA INDUSTRIAL, 2002):

- Interface homem-máquina: permite a operação e programação através dos operadores;
- Interface com sensores e atuadores: permite que os robôs atuem nos processos industriais de acordo com valores obtidos nos sensores; e
- Interface com controladores industriais: para que o robô possa interagir com os demais componentes dos sistemas de controle.

As Interfaces Homem-Máquina (IHM) são equipamentos que “traduzem” os comandos do homem para a máquina. São exemplos de interfaces para o controle de robôs:

- Teclados e *joysticks*: controle através de botões e alavancas que realizam ações programadas dentro do controlador do robô;
- Computador: envia comandos para o robô através de programas de comunicação;
- Dispositivos que captam impulsos elétricos emitidos pelo cérebro: interpretam e transmitem para o robô;
- Processamento de imagens: uma câmera filma uma determinada ação que processa as imagens, identifica os movimentos e transmite para o robô; e
- Sistemas hápticos: sistemas mecânicos e eletrônicos capazes de identificar ações através do “toque”, geralmente com dispositivos acoplados ao corpo ou sensíveis ao toque dos operadores.

A proposta básica de desenvolvimento é a criação de uma interface homem-máquina utilizando *wearable technology* a fim de desenvolver um sistema háptico que capture

informações da movimentação da mão humana e assim controle uma aplicação digital, robô ou sistema automatizado.

Existem alguns trabalhos correlatos na área, porém são poucos (*Instructables*, InMoove e *Robotic Glove* de *Harvard*), a grande maioria dos que têm alguma semelhança, trabalham com o desenvolvimento de tipos diferentes de interface e geralmente utilizam comunicação cabeada e local com o robô em questão.

2.2 AUTOMAÇÃO DE SISTEMAS

Historicamente, o surgimento da automação está ligado com a mecanização, sendo muito antigo, remontando da época de 3500 e 3200 a.C., com a utilização da roda. O objetivo era sempre o mesmo, o de simplificar o trabalho do homem, de forma a substituir o esforço braçal por outros meios e mecanismos, liberando o tempo disponível para outros afazeres, valorizando o tempo útil para as atividades do intelecto, das artes, lazer ou simplesmente entretenimento (Silveira & Santos, 1998).

Um sistema é um conjunto de elementos, dinamicamente relacionados, formando uma atividade para atingir um objetivo, operando sobre dados, energia e matéria, para fornecer informação, energia e matéria (CHIAVENATO, 1983).

Sendo assim, pode-se dizer que a automação de sistemas nada mais é do que a utilização de mecanismos que possam facilitar o esforço do homem na hora de manipular um sistema.

2.2.1 INTERFACE HOMEM-MÁQUINA

Também conhecida como Interface Homem-Computador é uma disciplina preocupada com o Design, avaliação, e implementação de sistemas interativos de computação para uso humano e com o estudo dos principais fenômenos em torno deles (HEWETT et al., 1992).

De acordo com Dumas (1993) a abordagem da IHM traduz-se em projetar produtos que facilitem a interação entre pessoas e computadores. Especialistas em IHC devem se preocupar com questões como: “O que constitui uma boa interface?” ou “Como posso distinguir uma interface boa de uma ruim?”.

Cada vez mais o IHM vem ganhando importância, pois as pessoas têm mais acesso aos computadores, não só em casa, mas também em seus locais de trabalho, sendo que muitas vezes eles dependem de computadores e outras máquinas computadorizadas para poder desempenhar sua função dentro de uma empresa. Sendo assim, caso haja uma interface não

entendível, isso pode resultar em sérios problemas para o funcionário em questão, mas também para a organização.

O termo “sistema”, empregado diversas vezes na descrição de IHM é derivado da teoria de sistemas e refere-se não apenas ao *hardware* e ao *software*, mas a todo o ambiente, seja ele uma organização ou mesmo a casa de uma pessoa, que utiliza ou é impactado pela tecnologia computacional em questão (Preece, 1994).

2.2.2 SISTEMAS HÁPTICOS

Pode-se dizer, em linhas gerais, que toda e qualquer atividade realizada por um computador, ou sistema computacional, geralmente imita alguma atividade ou característica da natureza humana. Sendo assim, quando fala-se em sistemas hápticos, logo pode-se notar uma semelhança grande com o sentido humano denominado tato. O termo háptico deriva-se da palavra grega “hapticos” que significa “tocar ou perceber” e relaciona-se à informação sensorial recebida através do toque ou contato físico (Burdea, 1999).

De acordo com Burdea e Coiffet (2003) dois diferentes conceitos são normalmente mencionados em relação à percepção do toque: tato e cinestesia. O tato está relacionado ao usuário sentir diferentes tipos de sensações como temperatura, pressão, vibração ou dor e depende da sensibilidade cutânea. Esta sensibilidade varia de acordo com a região utilizada para realizar o contato. A cinestesia, por sua vez, refere-se à percepção das tensões aplicadas aos músculos e juntas. Esta percepção é também chamada de propriocepção ou *force-feedback* (retorno de força) e permite identificar a rigidez de objetos.

Neste caso, os sistemas hápticos dependem de dois agentes importantes para conseguir fazer essa medição de toque e enviar alguma resposta ao usuário, seriam eles: os dispositivos, que basicamente recebem as informações do toque e as rotinas de controle, que enviam as informações.

Pode-se dizer que toda essa experiência que o ser humano tem com relação ao toque em objetos é transferida para o mundo digital através dos sistemas hápticos que é uma sub área de Realidade Virtual.

2.2.3 ROBÓTICA

Robô é um manipulador re-programável e multifuncional projetado para mover materiais, partes, ferramentas ou dispositivos especializados através de movimentos variáveis programados para desempenhar uma variedade de tarefas (R.I.A - *Robotics Industries Association*, 2015).

Logo, vê-se que os robôs são projetados para nos auxiliar em tarefas e automatizar processos antes feitos completamente através da força braçal humana, e deste contexto nasce a robótica, tida como “ciência dos sistemas que interagem com o mundo real, com pouco ou mesmo nenhuma intervenção humana” (MARTINS, 2006).

Assim sendo, pode-se dizer que a robótica é uma ciência multidisciplinar e que deve focar em desenvolver robôs que possuam autonomia para desempenhar tarefas, sendo elas sempre focadas em automatizar alguma ação humana. Logo, pode-se encontrar a robótica aplicada em nosso dia-a-dia, em equipamentos eletrônicos e eletrodomésticos, desde uma cafeteira elétrica até mesmo robôs autônimos que executam tarefas domésticas como já citado no Capítulo 1, mas também tem-se a robótica aplicada no meio industrial, onde ela mais cresce e se torna cada vez mais importante, pois muitas das coisas que tem-se hoje não seriam possíveis se não fossem por robôs, que realizam serviços muito repetitivos, pesados ou até mesmo nocivos para o homem. A indústria automobilística e eletro-eletrônica são exemplos de setores que fazem uso de robôs para estes tipos de trabalho.

Em 2006 as montadoras de automóveis respondiam por 36% dos robôs utilizados no planeta. Esse percentual caiu para 28% em 2010. O setor elétrico e eletrônico, que detinha 18% dos robôs, saltou para 26%. Também se destacam os fabricantes de plásticos, produtos químicos e cosméticos (Cruz, Renata. O Estado de S. Paulo, 2012).

2.3 WEARABLE TECHNOLOGY

O uso da tecnologia está cada vez mais presente, desde a chegada da internet até os dias de hoje, muita coisa mudou, houve o surgimento e o desaparecimento de diversas tecnologias, mas o fato é que cada vez mais busca-se aproximar a tecnologia das pessoas, auxiliando assim nas tarefas diárias e até mesmo em coisas que não imagináv-se precisar antigamente. Foi desta necessidade que surgiu o conceito de *Wearable Technology* (Tecnologia Vestível), onde o dispositivo eletrônico, independente do seu fim, vem embarcado em alguma peça de roupa e assim pode ser levado com o usuário para qualquer lugar. Alguns exemplos desse tipo de tecnologia são: *Smart Watches* (Relógios Inteligentes com tecnologia próxima a *smartphones*), *MEMS Sensors* (*MicroElectroMechanical Sensors* - Tecnologia embarcada em dispositivos pequenos que vão desde sensores a micro controladores), *Google Glass* (Óculos Inteligentes que trazem todas as tecnologias da Google Inc. em um óculos) e etc.

2.4 PLATAFORMAS DE PROTOTIPAÇÃO

Os sistemas digitais tornaram-se parte do cotidiano devido ao modo intenso pelo qual os circuitos digitais e as técnicas digitais passaram a ser utilizados em quase todas as áreas: computadores, automação, robôs, tecnologias e ciências médicas, transportes, telecomunicações, entretenimento, e assim por diante (Tocci et al, 2011).

Para que toda essa tecnologia pudesse estar envolvida na sociedade hoje, os sistemas computacionais (*hardwares* e *softwares*) evoluíram bastante desde o seu surgimento. Conforme novas tecnologias foram surgindo, elas também foram ficando cada vez mais baratas, e conseqüentemente mais acessíveis a todos. Tão acessível que hoje tem-se a possibilidade de montar computadores e desenvolver sistemas em residências comuns através de plataformas de prototipagem como é o caso do Arduino, por exemplo.

Essas plataformas não oferecem apenas um microprocessador (peça fundamental na construção de computadores e sistemas digitais) mas também oferece todo um aparato para realizar a comunicação com outros sistemas e computadores, como por exemplo: portas USB, Ethernet, componentes de microeletrônica básica (LEDs e botões) e etc.

No ambiente acadêmico, a construção de protótipos educacionais em disciplinas ou em trabalhos de iniciação científica ganha cada vez mais espaço, devido ao custo de aquisição dos protótipos das empresas que os comercializam, além da dependência tecnológica ao ser realizada esta aquisição, não se permitindo muitas vezes, a incorporação de novas tecnologias ou uso de novos algoritmos computacionais para testes comparativos (GOMES & SILVEIRA, 2007), (GOMES et al, 2011), (MAXIMO et al, 2011).

2.5 SENSORES

Em geral os sensores atuam transformando partes de uma grandeza física normalmente em um sinal elétrico, que por sua vez pode ser interpretado por certos equipamentos eletrônicos (BORGES & DORES, 2010). Em outras palavras, sensores são componentes que trabalham com a intenção de fazer com que o sistema computadorizado consiga se comunicar com o mundo real, recebendo e enviando informação.

Segundo Borges & Dores (2010), os sensores quando operam de forma direta, transformando uma forma de energia em outra são chamados de transdutores. Os sensores onde as operações ocorrem de forma indireta alteram suas propriedades, como a resistência,

capacitância ou indutância, sob a ação da grandeza de forma que essa alteração ocorre mais ou menos proporcional.

É através dos sensores que pode-se fazer com que os mecanismos implementados em um sistema computadorizado possa interagir com o usuário, por exemplo, se um robô não possuir sensores (de diversos tipos) ele não terá como saber se está próximo ou longe de um objeto, se está fazendo frio ou calor, se o ambiente está claro ou escuro e etc. Sem os sensores não é possível fazer qualquer tipo de comunicação com o mundo real e, sendo assim, a robótica e provavelmente boa parte da área da Computação não teria nenhuma utilidade prática real.

2.6 ARDUINO

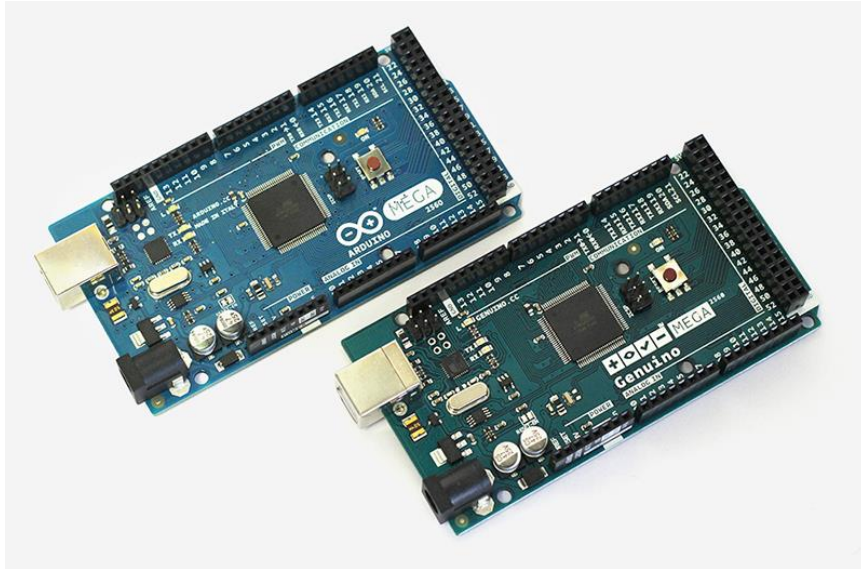
O Arduino nada mais é do que um microcontrolador, que segundo o site oficial do Arduino (arduino.cc):

“Arduino é uma plataforma de prototipagem de código aberto baseado em conceitos de *hardware* e *software* fáceis de usar. As placas Arduino são capazes de ler valores de entrada – sensores de luz, pressionamento de botões ou mensagens do *Twitter* – e transformar isso em uma informação de saída – ativar um motor, ligar um LED, publicar algo online. Você pode dizer para sua placa o que fazer enviando um conjunto de instruções para o microcontrolador na placa. Para fazer isso você utiliza a linguagem de programação do Arduino (baseada no *Wiring*) e o *software* do Arduino (IDE) baseada no *Processing*.”

Neste trabalho, o Arduino será programado para receber os sinais dos movimentos da mão do usuário e converter para código que será interpretado por uma interface digital que simula um robô um sistema automatizado.

Para o presente trabalho foi utilizado como base para início do desenvolvimento um Arduino Mega (Figura 1), posteriormente houve a troca por um Arduino Nano (Figura 2), também foi utilizada uma ProtoBoard comum (Figura 3).

Figura 1 – Arduino Mega.



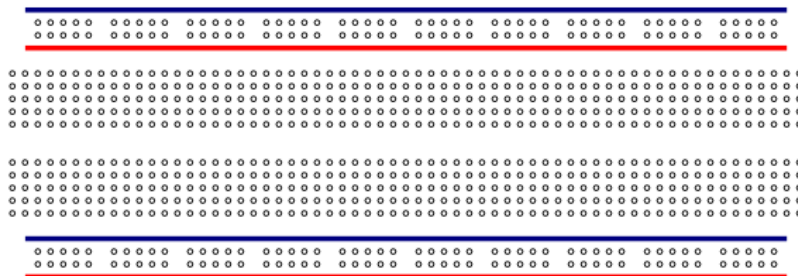
Fonte: Arduino Mega, 2015.

Figura 2 – Arduino Nano.



Fonte: Arduino Nano, 2015.

Figura 3 – ProtoBoard.



Fonte: ProtoBoard, 2015.

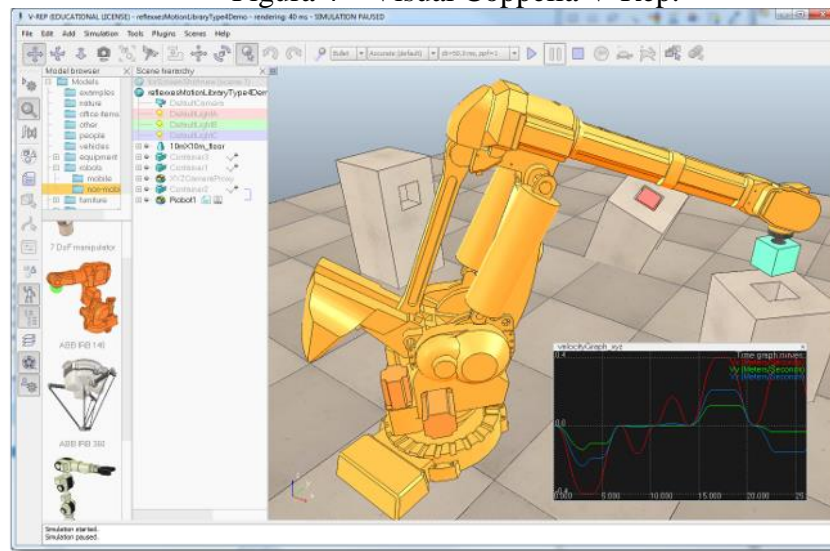
2.7 COPPELIA V-REP

O Coppelia V-rep é uma ferramenta capaz de simular robôs em modelos 3D para que sejam realizados testes em um ambiente virtual e controlado, antes de testar o sistema em um robô de verdade. Vale ressaltar que a ferramenta é extremamente parecida com o Unity 3D e

que por conta disso, muitos modelos de robôs (e até de outros objetos) que são desenvolvidos para serem utilizados no Unity 3D, podem ser aproveitados no Coppelia V-Rep, porém, a grande maioria dos modelos desenvolvidos especificamente para o Coppelia não podem ser aproveitados em nenhuma outra ferramenta, já que a mesma utiliza uma forma de modelagem própria, sendo assim, mesmo os objetos criados por ferramentas de modelagem comuns no mercado (Blender, Maya, 3DS Max e etc) precisam de pequenas adaptações para que possam ser usados no Coppelia V-Rep, além é claro de terem todas as funcionalidades de movimento implementadas previamente.

A linguagem de programação usada pelo Coppelia V-Rep é (Lua), que é similar a Python.

Figura 4 – Visual Coppelia V-Rep.

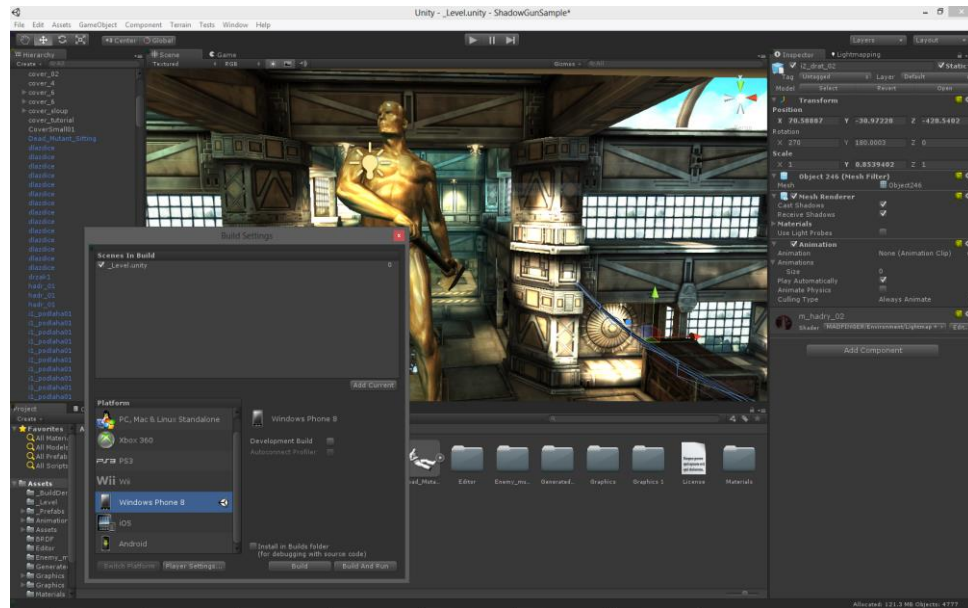


Fonte: Coppelia V-Rep, 2015.

2.8 UNITY 3D

Unity 3D é um software desenvolvido com o objetivo de ser uma plataforma de desenvolvimento de jogos digitais, utilizando-se de recursos de modelagem 3D para desenvolvimento de ambientes e personagens. A ferramenta também pode ser utilizada como ambiente virtual, da mesma forma que o Coppelia V-Rep, com o objetivo de simular situações e testá-las em ambiente virtual e controlado. O Unity 3D é baseado na criação de cenas para que possam ser então adicionados objetos, ou seja, cada tela que uma aplicação apresenta, na realidade nada mais é do que uma cena previamente criada e programada. A linguagem de programação utilizada pela ferramenta pode ser tanto o Javascript quanto o C#.

Figura 5 – Visual do Unity3D.



Fonte: Unity3D, 2012.

2.9 DIY (DO IT YOURSELF)

Do It Yourself (DIY) ou na tradução literal para o português, Faça Você Mesmo, é um conceito que consiste em desenvolver, reparar ou modificar mecanismos, sistemas, ferramentas, aplicações e etc, por conta própria, sem a necessidade de comprar nada profissional, apenas em alguns casos comprando itens básicos, para ter uma base por onde iniciar a criação.

No presente trabalho, trabalha-se o conceito de DIY juntamente com tecnologias já existentes, sendo que parte da construção do protótipo, como a base da luva e demais componentes de fixação de peças trazem o conceito de DIY, porém, os sensores e placas de prototipagem, não trazem este conceito.

2.10 VERMASON, VELOSTAT, LINQSTAT

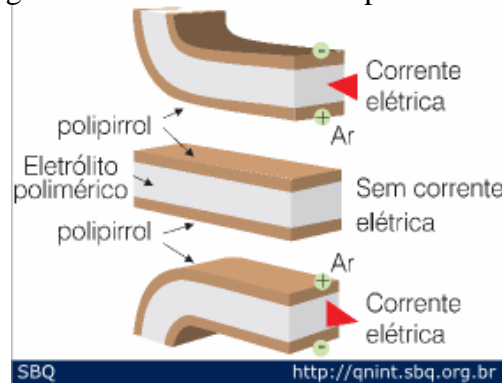
Vermason, *Velostat* e *Linqstat* são produtos que oferecem ao cliente uma solução quanto a descargas eletromagnéticas. Basicamente são produtos que têm uma propriedade condutiva de correntes elétricas, além de serem sensíveis a deslocamento mecânico e à pressão. Sendo assim, é possível desenvolver aplicações que necessitem deste tipo de interação. Em sua composição, são adicionadas substâncias que criam o fator de condução elétrica e de tensão.

O plástico condutor de eletricidade é chamado pelos pesquisadores de polianilina "dopada", por conter elementos adicionados à sua estrutura molecular - um mecanismo chamado de dopagem, largamente utilizado na fabricação de outros componentes eletrônicos, como transistores e diodos (Site Inovação e Tecnologia, 2007).

Como exemplo pode-se citar um item que será utilizado na parte prática deste mesmo trabalho que é o *Electrostatic Conductive Bag* da Marca *Vermason*, que nada mais é do que um plástico com as propriedades citadas.

A base de funcionamento deste tipo de material condutor pode ser exemplificado conforme a Figura 6.

Figura 6 – Funcionamento de plásticos condutores.



Fonte: De Paoli, Marco. 2001.

É importante frisar que materiais como este além de todas as propriedades citadas, ainda possuem o fator de serem facilmente maleáveis, o que torna o manuseio de aplicações e interfaces com alto envolvimento humano mais fáceis, como o de Luvas Robóticas, por exemplo.

2.11 ROBOTIC GLOVE

A luva robótica nada mais é do que a junção de vários dos itens e pontos discutidos. Esse mecanismos já apresentados, quando dispostos em uma interface onde a base é uma luva comum, podem nos trazer diversas opções de trabalho, como por exemplo, controlar um braço robótico, controlar uma aplicação virtual, controlar um robô inteiro através de movimentos pré-definidos que serão realizados com a mão e etc.

Vale resaltar que luvas robóticas nesse sentido, são uma boa opção para a criação de sistemas de controle que sejam de fácil manuseio, pois se baseiam em conceitos de sistemas hápticos, facilitam a IHM e em geral possuem um custo bem menor do que mecanismos

sofisticados de controles através de botões, alavancas ou até mesmo algo mais moderno como sistemas inteiramente digitais em telas com função de *touch-screen*.

Pode-se usar como exemplo a Figura 7. Não existe um padrão de desenvolvimento destas luvas, já que cada projeto é feito de forma independente.

Figura 7 – Luva Robótica.



Fonte: LUDOVISI, DAVIDE. 2011.

2.12 CONCLUSÕES DO CAPÍTULO

Dentro dos conceitos e tecnologias apresentadas, destacam-se o Arduino, Unity3D, *Wearable Technology* e *Robotic Glove*, pois os mesmos possuem importância enorme no desenvolvimento do protótipo da luva. Vale ressaltar que o conceito de desenvolvimento da luva tem como foco os Sistemas Hápticos onde o ser humano, com movimentos normais de seu corpo, consegue controlar aplicações robóticas ou automatizadas. Sendo assim, com esse referencial, pode-se ter uma base para a fase de desenvolvimento.

3 DESENVOLVIMENTO DA INTERFACE

Para realizar o desenvolvimento da interface foi utilizado o Arduino e o Intel Galileo (este foi usado mais para comparação e realização de alguns testes). Também foram utilizados programas para simular a interação com robôs, o Coppelia V-Rep e o Unity 3D (onde foi construída toda a base da aplicação, neste caso o Coppelia V-Rep serviu apenas para comparação). É válido citar a utilização dos sensores flexíveis, que são os responsáveis por captar os movimentos da mão.

Unindo essas três tecnologias (Arduino, Unity 3D e os sensores flexíveis) conseguiu-se então dar início ao desenvolvimento da proposta de interface que seria uma luva “inteligente”.

3.1 MONTAGEM E PROTOTIPAÇÃO

A Luva “Inteligente” neste caso foi a proposta de modelo mais adequado para ser usado como interface, já que seu formato é bastante usual e prático, pois não exige treinamento, já que o seu manuseio é absolutamente intuitivo. Em todo caso, a luva será apenas a carcaça do sistema, que contará com sensores flexíveis (que serão responsáveis por captar os movimentos dos dedos da mão) e contará também com o Arduino, que neste caso deve ser apresentado em uma versão menor, para que possa ser fixado na luva de forma que não atrapalhe na movimentação e manuseio da mesma.

O desenvolvimento da luva passou por várias etapas e também por vários problemas e ao longo do caminho, algumas ideias iniciais não puderam ser realizadas e foram substituídas por outras mais práticas. Desde o começo o projeto focou o baixo custo para o desenvolvimento do protótipo e isso foi mantido até então.

3.1.1 ETAPA 1 – PESQUISA

Inicialmente foi feita uma pesquisa na área de Robótica para consultar trabalhos similares e analisar as técnicas e recursos por eles utilizados, poucos resultados foram encontrados, dos quais podem-se citar: Conor Walsh e sua equipe de engenheiros dos laboratórios de robótica de Harvard possuem um protótipo de luva voltada para o auxílio de pessoas com deficiência motora. Existe também o projeto InMoov que é um projeto open source de criação de um robô humanóide com todas as características de um humano real, alguns de seus estudos iniciais começaram com o desenvolvimento de luvas robóticas. Pode-se ainda citar o site da

Instructables, onde existe um projeto que tenta auxiliar no desenvolvimento de uma luva robótica através do método DIY.

Todos os resultados encontrados anteriormente possuem alto custo ou seus códigos fechados, com isso pesquisas em outras áreas tiveram que ser realizadas. Porém, como o tema do trabalho é muito específico, a falta de resultados relevantes acabou virando um problema. Sendo assim, a base das ideias para criar o protótipo se tornou de certa forma empírica.

Segundo Demo (2000), “É a pesquisa dedicada ao tratamento da "face empírica e fatural da realidade; produz e analisa dados, procedendo sempre pela via do controle empírico e fatural”.

Foram utilizados conhecimentos prévios de projetos realizados em outras áreas e que tinham detalhes que podiam ser aproveitados para o desenvolvimento do trabalho em questão.

3.1.2 ETAPA 2 – PRIMEIRAS TENTATIVAS

Basicamente nesta etapa foram realizadas diversas tentativas de construir um *flex sensor* (sensor flexível) com materiais e recursos que pudessem ser encontrados facilmente em qualquer residência.

De início foi realizada a compra de um plástico condutivo (Figura 8), para que pudesse ser feita a medição da variação de tensão de quando um dedo é flexionado para os circuitos do arduino.

Figura 8 - *Electrostatic Conductive Bag* – Plástico condutor.



Fonte: VERMASON, 2015.

Apenas o plástico não foi o suficiente e a criação do sensor não funcionou, não foi possível medir as variações de tensão para poder calcular a dobra do dedo. Foram utilizados então fios de cobre na tentativa de aumentar a condutividade do sensor, obtendo sim uma melhora, porém a oscilação de valores e a margem de erro era muito grande então a ideia foi descartada.

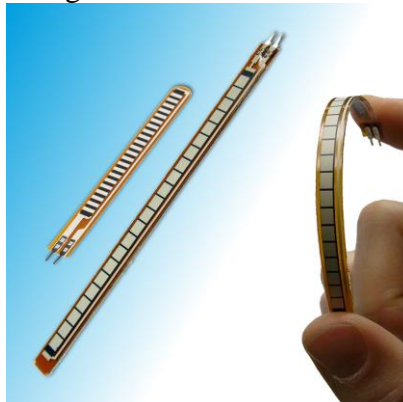
É importante ressaltar que as propriedades condutivas do plástico existem e funcionam, o problema se dá apenas pelo fato de que não há uma precisão muito grande, além de falhas constantes na transmissão da informação, o que impossibilitou a construção da interface, além da falta de tempo para que mais estudos pudessem ser feitos no plástico. Neste caso, optou-se pela compra de sensores prontos que pudessem ter o mesmo efeito e maior estabilidade.

Nesta etapa ainda, foram realizados os primeiros códigos e as primeiras tentativas de ligar o arduino com o sensor.

3.1.3 ETAPA 3 – INVESTIMENTOS

Como a criação do sensor através do método DIY não funcionou da maneira esperada, foi feita a compra de sensores flexíveis (Figura 9), os sensores chegaram dentro de 30 dias e não teve nenhuma complicação. Apesar da margem de valores apresentados por eles serem pequenas para maior precisão no mapeamento, foi possível utiliza-los para calcular a angulação que o dedo faz com extrema facilidade, o que resultou em um significativo avanço no desenvolvimento do protótipo. O código utilizado foi o mesmo da etapa anterior que já estava correto, apenas algumas correções foram realizadas para ajustar com a resposta do novo sensor.

Figura 9 – Sensor Flexível



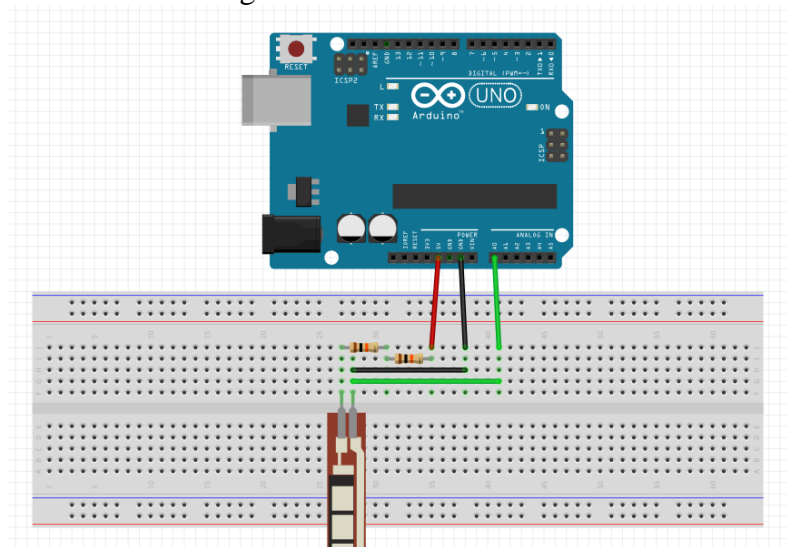
Fonte: SPECTRAL SYMBOL, 2015.

Pode-se citar como principais características do sensor, segundo site oficial da SpectraSymbol:

- Medição de deslocamento de ângulo;
- Dobra e flexiona fisicamente com dispositivos mecânicos;
- Construção simples;
- Baixo custo.

A ligação dos componentes, resistores e sensores são mostrados na (Figura 10). Seu complemento mostrando a ligação completa e a lista de componentes encontra-se no Anexo 1 e Anexo 2 respectivamente.

Figura 10 – Circuito Arduino.



Fonte: O Próprio Autor.

3.1.4 ETAPA 4 – COPPELIA V-REP E UNITY3D

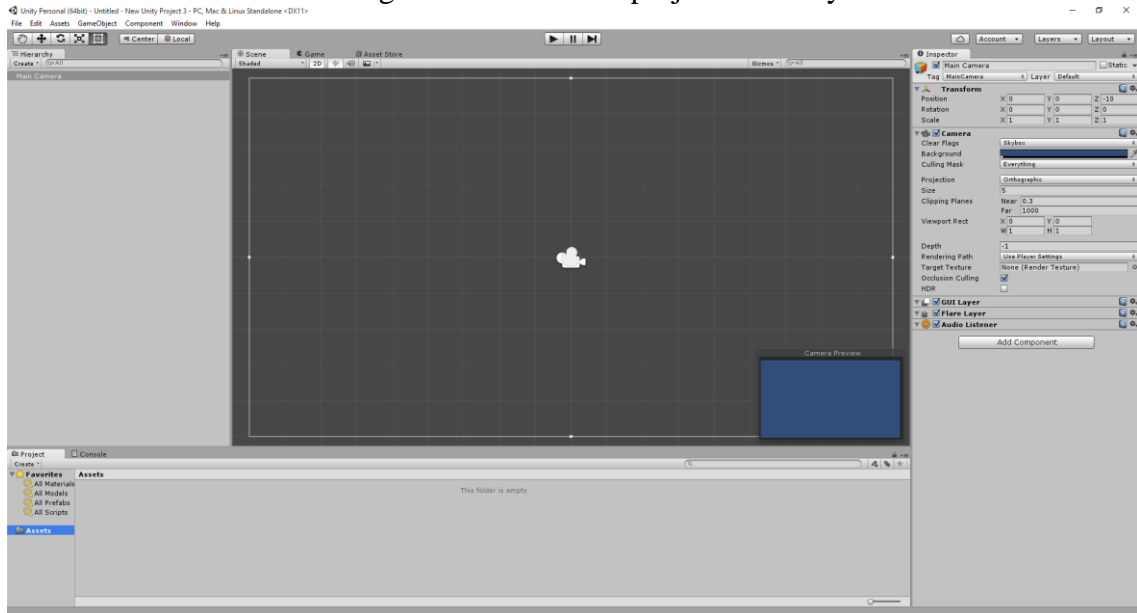
Após o sucesso com o sensores, foi dado início na conexão entre o arduino e o ambiente de testes Coppelia V-Rep.

A primeira tentativa de conexão foi utilizando portas seriais, onde não foi obtido êxito na conexão e na comunicação. Após muito tempo, chegou-se a conclusão de que o programa em questão demandaria muito tempo para que pudesse ser compreendido e assimilado, devido a enorme complexidade e necessidade de informações técnicas, pois só então poderia ser utilizado para o devido propósito do trabalho.

Foram analisadas alternativas mais “simples” para cumprir o mesmo propósito, que seria a utilização de um programa com interface visual gráfica que pudesse simular a interação entre o Arduino e um suposto Robô.

Após a tentativa de utilização do Coppelia não ter dado certo, ficou decidido que seria utilizado o Unity 3D (Figura 11), este que possui recursos de simulação parecidos com o Coppelia V-Rep (desde a forma em estabelecer conexão até os recursos de criação de cena e objetos).

Figura 11 – Início do projeto no Unity3D.



Fonte: O próprio Autor.

A conexão com o Unity aconteceu de forma tranquila, utilizando apenas de portas seriais, sem a necessidade de comunicar via sockets. Para que pudesse ser utilizado o Unity 3D para o propósito do trabalho, foi necessário criar retângulos na tela que representaria os dedos de um possível robô ou algum outro sistema de controle, já que neste caso, o Unity3D não possui modelos de robôs prontos como no Coppelia V-Rep. A mudança de *software* não interferiu nos objetivos do trabalho, que é testar a viabilidade de construção de um protótipo de controle de robôs ou sistemas automatizados com baixo custo e eficiência.

3.1.5 ETAPA 5 – PROGRAMAÇÃO E DESENVOLVIMENTO

Após várias tentativas de criar um código limpo e funcional, conseguiu-se chegar a uma situação favorável, com um código simples no Arduino, apenas capturando os dados dos sensores e os enviando para o Unity (Quadro 1).

Quadro 1 – Modelo do código utilizado no Arduino.

```

void setup(){
  Serial.begin(9600);
  //Inicia conexão serial
}

void loop(){
  int sensor0, sensor1, sensor2, sensor3, sensor4, degrees, val0, val1, val2,
  val3, val4;
  String valor0, valor1, valor2, valor3, valor4, valores;
  //Cria variáveis

  sensor0 = analogRead(0);
  sensor1 = analogRead(1);
  sensor2 = analogRead(2);
  sensor3 = analogRead(3);
  sensor4 = analogRead(4);
  //Atribui valor de uma porta analógica a uma variável

  val0 = degrees = map(sensor0, 177, 100, 0, 100);
  val1 = degrees = map(sensor1, 190, 115, 0, 100);
  val2 = degrees = map(sensor2, 247, 142, 0, 100);
  val3 = degrees = map(sensor3, 239, 158, 0, 100);
  val4 = degrees = map(sensor4, 280, 160, 0, 100);
  //Mapeia valores emitidos pelo sensor

  valor0 = String(val0);
  valor1 = String(val1);
  valor2 = String(val2);
  valor3 = String(val3);
  valor4 = String(val4);
  //Converte valores int em strings

  valores = valor0 + "," + valor1 + "," + valor2 + "," + valor3 + "," + valor4;
  //Concatena string criadas anteriormente

  Serial.println (valores);
  //Imprime valores

  delay(100);
}

```

Fonte: O Próprio Autor.

O código no Quadro 1 está capturando as informações dos sensores, fazendo um mapeamento desses valores, concatenando tudo em uma única string e assim, enviando para o Unity 3D.

Da mesma forma, procurou-se desenvolver um código simples e funcional no Unity, evitando assim que gerasse qualquer tipo de latência ou *delay* na execução do aplicativo (Quadro 2).

Quadro 2 – Modelo de código utilizado no Unity3D.

```

using UnityEngine;
using System.Collections;
using System.IO.Ports;
using System.Linq;
using System.Collections.Generic;
using System;
using System.Text.RegularExpressions;
//Importa bibliotecas

public class FlexCode1 : MonoBehaviour {

    SerialPort sp = new SerialPort("COM3", 9600);
//Cria variavel para realizar conexão com o Arduino

    public Dedo1 dedo1;
    public Dedo2 dedo2;
    public Dedo3 dedo3;
    public Dedo4 dedo4;
    public Dedo5 dedo5;
// Cria variáveis públicas para realizar a chamada dos scripts de movimentação
dos dedos

    // Inicialização
    public void Start () {
        sp.Open();
        sp.ReadTimeout = 1000;
    }

    // Update é chamado a cada frame
    public void Update () {
        if (sp.IsOpen) {
            MoveObject ((sp.ReadLine ());
//Se a conexão estiver aberta, realiza a leitura da linha
        }
    }

    public void MoveObject(String Serial){
        string[] lines = Regex.Split(Serial, ",");
        int valor0 = Convert.ToInt32(lines[0]);
        int valor1 = Convert.ToInt32(lines[1]);
        int valor2 = Convert.ToInt32(lines[2]);
        int valor3 = Convert.ToInt32(lines[3]);
        int valor4 = Convert.ToInt32(lines[4]);
//Quebra valores independentes de cada sensor que representa o dedo e insere em
ma variável única

        dedo1.MoverDedo(valor0);
        dedo2.MoverDedo(valor1);
        dedo3.MoverDedo(valor2);
        dedo4.MoverDedo(valor3);
        dedo5.MoverDedo(valor4);
// Chama os scripts de movimentação dos dedos
    }
}

```

Fonte: O Próprio Autor.

O código no Quadro 2 mostra o recebimento dos dados do Arduino através de uma porta serial. É feito um *split* na string que o Arduino envia e cada valor é colocado em um índice de um vetor de *strings*, depois, cada valor é enviado para o *script* de cada um dos “dedos” chamando diretamente uma única função desse novo *script* (Quadro 3).

Quadro 3 – Modelo de código para os “dedos” no Unity3D.

```
using UnityEngine;
using System.Collections;

public class Dedo1 : MonoBehaviour{
    public void MoverDedo(int Valor){
        int line = Valor;
        //Recebe valor do sensor vindo do outro script

        //Verifica o valor da variável "line" e transforma o tamanho do objeto na tela
        de acordo com cada possibilidade listada
        if (line > 0 && line <= 5){
            transform.localScale = new Vector3(1, 5.0f, 1);}

        if (line > 5 && line <= 10){
            transform.localScale = new Vector3(1, 4.8f, 1);}

        if (line > 10 && line <= 15){
            transform.localScale = new Vector3(1, 4.6f, 1);}

        if (line > 15 && line <= 20){
            transform.localScale = new Vector3(1, 4.4f, 1);}

        if (line > 20 && line <= 25){
            transform.localScale = new Vector3(1, 4.2f, 1);}

        if (line > 25 && line <= 30){
            transform.localScale = new Vector3(1, 4.0f, 1);}

        if (line > 30 && line <= 35){
            transform.localScale = new Vector3(1, 3.8f, 1);}

        if (line > 35 && line <= 40){
            transform.localScale = new Vector3(1, 3.6f, 1);}

        if (line > 40 && line <= 45){
            transform.localScale = new Vector3(1, 3.4f, 1);}

        if (line > 45 && line <= 50){
            transform.localScale = new Vector3(1, 3.2f, 1);}

        if (line > 50 && line <= 55){
            transform.localScale = new Vector3(1, 3.0f, 1);}

        if (line > 55 && line <= 60){
            transform.localScale = new Vector3(1, 2.8f, 1);}

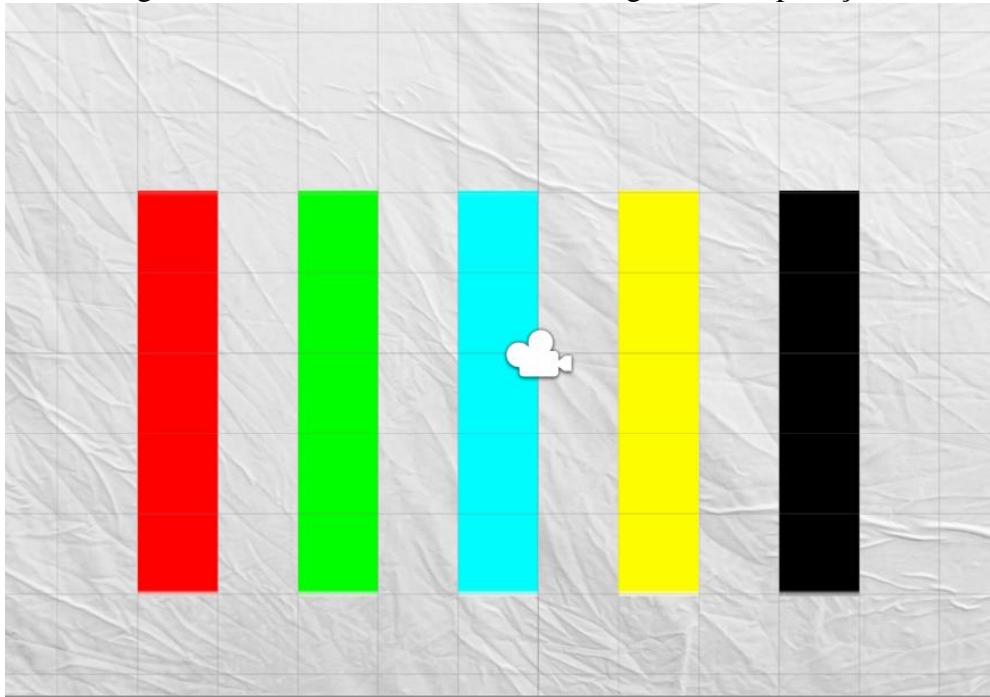
        if (line > 60 && line <= 65){
            transform.localScale = new Vector3(1, 2.6f, 1);}

        if (line > 65 && line <= 70){
            transform.localScale = new Vector3(1, 2.4f, 1);}}
```

Fonte: O Próprio Autor.

No código no Quadro 3, o *script* é replicado para cada um dos 5 dedos e o seu funcionamento é padrão em todos eles, ou seja, o *script* recebe o valor destinado ao dedo em específico e analisa qual é esse valor, atribuindo assim uma posição fixa para o objeto que representa o dedo, sendo que o valor tende a sofrer variação. Portanto é feito um *range* de aproximadamente 5 números para tentar estabilizar o movimento do dedo.

Figura 12 – Primeira forma de interface gráfica da aplicação.



Fonte: O Próprio Autor.

Na aplicação (Figura 12) os dedos da mão de um possível robô ou ainda partes de um sistema automatizado, são representados por barras coloridas, onde cada barra seria um dedo ou partes independentes de um sistema. Sendo assim, o objetivo da aplicação em questão é controlar essas barras através dos movimentos realizados naturalmente com a mão do usuário, movimentos estes que são capturados através da luva.

3.2 CONCLUSÕES DO CAPÍTULO

Ficou claro que houve falta de tempo para que pudesse ser realizada com precisão a comunicação entre plataformas de prototipagem (aqui no caso o Arduino) e programas computacionais (Coppelia V-Rep e Unity 3D), fato este que levou a algumas adaptações no roteiro e mudanças da ideia inicial no presente trabalho.

É mais do que óbvio de que a interface ficou longe do que foi imaginado no início do desenvolvimento, porém, já serve para que possa-se ter uma ideia de como funciona o

comportamento de uma tecnologia dessas. É importante frizar que a facilidade de trabalho com o Unity 3D é muito grande e que o Coppelia V-Rep, apesar de possuir aparentemente mais recursos, não é um programa intuitivo o que pode levar a uma demanda de tempo enorme para dominar o programa e só assim começar a desenvolver algo nele.

Pode-se concluir que o presente trabalho oferece um norte para onde seguir no caso de uma continuidade no desenvolvimento, os erros iniciais já não serão mais repetidos. É lógico que existe espaço para melhorias, porém, com o tempo escasso isso se torna impraticável no tempo destinado ao desenvolvimento do presente trabalho.

4 VALIDAÇÃO E TESTES

Segundo o dicionário Aurélio, teste é entendido como: “prova, verificação da eficiência ou do bom funcionamento (de máquinas, materiais, etc.)” (FERREIRA, 2010). Neste caso, será realizado o teste de um *software* que, para Pressman (2010), é um processo de execução de um programa com a finalidade de encontrar um erro.

No que diz respeito a testes de *software*, é importante que a atividade de teste seja planejada e desenvolvida através de uma estratégia definida e que estabeleça procedimentos a serem adotados como técnicas e métodos específicos que deverão ser seguidos, sempre devidamente documentados (HIRAMA, 2012).

A estratégia de testes utilizada no presente trabalho é a realização de uma espécie de *check-list* que contém os principais aspectos que são relevantes para serem testados na interface desenvolvida.

4.1 CENÁRIO

O cenário de teste estabelece um roteiro, onde são descritas as regras de negócio, o fluxo principal, ou seja, o fluxo que satisfaz o interesse do usuário, bem como os fluxos alternativos ou desvios, que descrevem condições alternativas ao fluxo principal. Os cenários de testes definem também como os casos de teste devem ser aplicados (RIBEIRO, 2010).

Neste caso o cenário será composto de diversas situações que devem ser avaliadas ao longo do processo de testes e a organização deste cenário será feita através de uma tabela.

4.2 TABELA DE TESTES

A tabela usada na fase de testes é composta de duas colunas, a primeira trata-se das situações que devem ser avaliadas, neste caso tem-se as seguintes situações: Resposta dos Sensores, Calibragem, Processamento do Arduino, Processamento do Unity, Design de Interface, Usabilidade e Considerações Gerais. A segunda coluna trata da avaliação referente a cada situação da coluna anterior. Sendo assim o modelo da tabela antes de seu preenchimento fica da seguinte forma (Tabela 1):

Tabela 1 – Modelo de Formulário de Controle de Testes

FORMULÁRIO DE CONTROLE DE TESTES	
Situação	Avaliação
1. Resposta dos sensores	
2. Calibragem	
3. Processamento do Arduino	
4. Processamento do Unity	
5. Design de Interface	
6. Usabilidade da Luva “Inteligente”	
7. Considerações gerais do sistema	

Fonte: O Própio Autor.

4.2.1 RESPOSTA DOS SENSORES E CALIBRAGEM

Este é o primeiro processo que deve ser averiguado na rotina de testes. A calibragem consiste basicamente em inicializar os valores dos sensores da maneira correta. Como cada pessoa possui um tamanho diferente para os dedos da mão, a tensão gerada no *Flex Sensor* será diferente também e por conta disso, deve-se recalcular os valores para poder chegar a angulação correta e assim manter a movimentação dos dedos sempre dentro de um mesmo padrão.

O primeiro passo para realizar a calibragem dos sensores é solicitar ao usuário que ele coloque as luvas. O segundo passo é analisar quais valores o Arduino está recebendo, qual o valor é transmitido quando o dedo está reto, ou seja, num ângulo de 0° e qual valor é transmitido quando o dedo está dobrado, ou seja, num ângulo de 90° . Tendo esses valores iniciais e finais, pode-se então substituí-los na variáveis que mapeiam a resposta do sensor, recarregar o código para o Arduino e assim iniciar a aplicação.

O processo de calibragem ainda está sendo feito completamente de forma arcaica, sendo necessário ter conhecimentos em programação e noção de como o código foi escrito para que só assim possa ocorrer a calibragem corretamente, porém, este processo pode ganhar uma melhoria significativa em trabalhos futuros.

4.2.2 PROCESSAMENTO DO ARDUINO

A capacidade de processamento do Arduino realmente surpreende pela velocidade e competência. Uma vez que tem-se uma quantidade significativa de código rodando no Arduino e ele consegue processar tudo com relativa facilidade.

Pode-se ressaltar aqui a questão da concatenação com *strings*, por algum motivo, em alguns casos ele gera um pequeno erro e a *string* é enviada completamente vazia. Num primeiro momento este erro não chegou a afetar o resto da aplicação, mas se existisse a possibilidade de aumentar a velocidade de tráfego de informações provavelmente teria que ser feito todo um tratamento em cima deste erro.

4.2.3 PROCESSAMENTO DO UNITY

O Unity 3D apresenta algumas complicações com relação a muito processamento ao mesmo tempo, é bem provável que um tratamento com *threads* resolveria um pouco a situação. Quando tem-se vários scripts rodando ao mesmo tempo e todos eles recebendo informações externas, ocorrem atrasos e em alguns momentos a execução chega a travar (por completo ou apenas algumas partes dela, já que a programação foi feita de forma independente).

O Unity3D consegue estabelecer conexão com o Arduino de forma bastante tranquila e dificilmente falha nesse quesito, aliás, é fato que em algumas execuções o Arduino atrasa por alguns instantes o envio da primeira linha de código e o Unity3D já está esperando por ela, gerando assim um pequeno erro, mas que não acarreta problemas na execução da aplicação.

4.2.4 DESIGN DE INTERFACE

Existe o problema de que muitos recursos inicialmente planejados para serem utilizados encontravam-se no Coppelia V-Rep e não há a possibilidade de exportar esses recursos (ao menos não sem conhecimentos de modelagem 3D), sendo assim, a parte visual da aplicação realmente carece um pouco de recursos mais chamativos e bem acabados.

Tendo isso em vista, pode-se considerar apenas a parte do propósito da aplicação, ou seja, a representação dos dedos através de barras coloridas funcionou muito bem e consegue passar tranquilamente a mensagem que é o foco do trabalho.

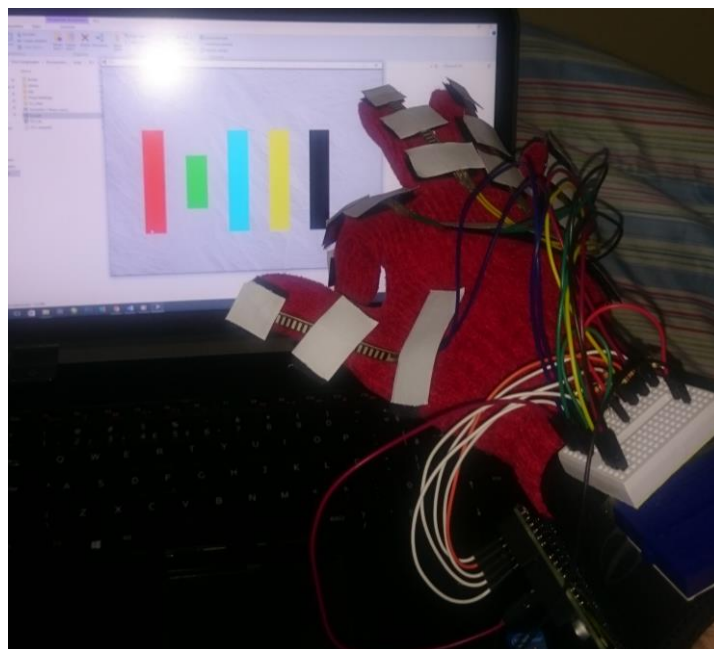
4.2.5 USABILIDADE DA LUVA “INTELIGENTE”

A luva possui diversas complicações, o que impossibilita a utilização do protótipo apresentado em uma ambiente empresarial comum. Antes disso ocorrer muitas correções devem ser feitas, entre elas cabe citar:

- Tamanho da luva: Por se tratar de um protótipo caseiro, o tamanho da luva é realmente muito pequeno para as mãos de um homem adulto. O correto seria ter uma luva que se ajustasse a diversos tamanhos de mão;
- Disposição dos sensores: Os sensores não se encaixam perfeitamente na luva, sendo que em alguns casos os cabos de conexão não possuem tamanho suficiente para alcançar o encaixe dos sensores e ligá-los no Arduino, isso ocorre pois os sensores não possuem todos o mesmo tamanho, mas também pelo fato de que a base da luva (que nada mais é do que uma luva de lã comum) não foi projetada para este propósito;
- Maleabilidade: Com todos os acessórios dispostos na luva (fios, plásticos, fitas adesivas, velcros, cabos de conexão, sensores e etc) um pouco da flexibilidade que a luva naturalmente trás é perdida, prejudicando assim a maleabilidade da mesma, não permitindo que certos movimentos como a dobra total dos dedos seja realizada corretamente.

A Figura 13 e a Figura 14 demonstram a versão final do protótipo da luva em funcionamento.

Figura 13 – Versão Final do Protótipo da Luva”Inteligente”.



Fonte: O Próprio Autor.

Figura 14 – Versão Final do Protótipo da Luva “Inteligente” 2.



Fonte: O Próprio Autor.

4.2.6 CONSIDERAÇÕES GERAIS DO SISTEMA

O protótipo em questão possui diversas limitações e a parte lógica da interface, ou seja, o que se passa no ambiente digital do mesmo, poderia ter mais recursos e um visual mais bem planejado.

Existem diversas falhas que precisam de correção assim como diversos pontos que não geram erros propriamente ditos, mas que podem ser melhorados. Uma vez que o foco é um projeto de baixo custo, e com curto prazo de desenvolvimento, pode-se dizer que o protótipo está cumprindo o que promete e rodando, dentro de suas limitações técnicas, com bastante tranquilidade

4.2.7 FINALIZAÇÃO DA FASE DE TESTES

Ao final da fase de testes, após realizar a avaliação de todas as situações apresentadas na Seção 4.2, a tabela de controle de testes preenchida (Tabela 2) fica da seguinte maneira:

Tabela 2 – Formulário de Controle de Testes Preenchido ao Final da Etapa.

FORMULÁRIO DE CONTROLE DE TESTES	
Situação	Avaliação
1. Resposta dos sensores	Respostas vindas dos sensores não são padronizadas e por isso cada sensor deve ter um mapeamento específico de seus valores.
2. Calibragem	Fazer a calibragem requer tempo e conhecimentos específicos do programa.
3. Processamento do Arduino	Apesar de conter grande quantidade de código, o arduino consegue processar tudo sem nenhum tipo de dificuldade.
4. Processamento do Unity	O processamento de código do Unity 3D é rápido, porém em alguns casos ocorre um certo atraso, principalmente nos casos de movimentação dos 5 dedos ao mesmo tempo.
5. Design de Interface	Os recursos de interface do Unity 3D são muito vastos, poderia ter sido aproveitado melhor para o desenvolvimento da aplicação. Mas o programa ficou fácil e intuitivo.
6. Usabilidade da Luva “Inteligente”	A luva é extremamente difícil de ser colocada sozinho e não se harmoniza perfeitamente com a mão, atrapalhando a movimentação em alguns casos. Mas trata-se apenas de um protótipo.
7. Considerações gerais do sistema	O sistema como um todo poderia fluir um pouco melhor, existem diversos pontos que deve ser melhorados e gargalos a serem eliminados, além da parte gráfica que poderia simular melhor a movimentação de um robô.

Fonte: O próprio Autor.

4.8 CONCLUSÃO DO CAPÍTULO

Pode-se concluir que a etapa de testes de qualquer produto ou serviço é de extrema importância para o bom desempenho do mesmo num ambiente empresarial, porém, não adianta apenas realizar testes, correções devem ser feitas para que possa-se trabalhar sempre com algo que tenha qualidade.

No trabalho em questão, todos os testes e análises realizadas foram feitas dentro de um ambiente controlado e com poucas variáveis, mas mesmo assim, diversos erros foram constatados, o processo de correção destes erros poderia demorar um longo período de tempo,

mas pode-se afirmar que, após as correções serem implementadas, o projeto iria funcionar com mais estabilidade, intuitividade e velocidade, o que iria auxiliar bastante numa possível implementação do protótipo no mercado de trabalho.

5 CONCLUSÃO

A robótica e a automação de sistemas são duas áreas que possuem uma afinidade muito grande, além de serem duas áreas que crescem muito a cada dia, tanto no ramo industrial, quanto do residências. Hoje em dia é cada vez mais comum empresas e residências possuírem robôs, conforme justificado no Capítulo 1. Sendo assim, pode-se concluir que a criação de novas interfaces e formas de comunicação e controle com esses robôs ou sistemas automatizados também é imprescindível, porém, é mais do que evidente que essas novas tecnologias de controle que serão encarregadas de realizar a interação entre o homem e a máquina, devem possuir o mínimo de ergonomia e intuitividade, sem deixar de lado a questão da simplicidade, procurando sempre focar no baixo custo e é exatamente esta a proposta do trabalho.

Com relação as considerações do protótipo desenvolvido, pode-se dizer que atendeu ao objetivo geral, a comunicação foi realizada, houve interação com o sistema e o baixo custo (que foi mantido desde o início) foi um grande diferencial para outros trabalhos que existem na área. Pode-se dizer, em linhas gerais que o protótipo é funcional e possui facilidade de uso, bem como o *software* desenvolvido que além de ter alta eficiência com seu tempo de resposta (o que também pode ser considerado um diferencial com relação a alguns trabalhos existentes), possui uma aparência bastante amigável e intuitiva, o que facilita a implementação em uma empresa, por exemplo e sua utilização pelo usuário.

Dentre as dificuldades encontradas, destacam-se a calibragem dos dedos, pois cada sensor possui um intervalo diferente de valores e devem ser mapeados separadamente. Pode-se citar também a conexão com o módulo Ethernet do Arduino, onde não obteve-se êxito e também a comunicação com o Coppelia V-Rep, que como já comentado no Capítulo 3, o mesmo demanda de muito conhecimento técnico devido a sua complexidade.

Dentre as possibilidade de trabalhos futuros, pode-se destacar principalmente o uso de tecnologias de comunicação remota, assim como o Zigbee (tecnologia de comunicação sem fio e de baixo custo para aplicações industriais), Bluetooth (tecnologia de comunicação sem fio através de pareamento de dispositivos), Wi-Fi (tecnologia de comunicação de rede sem fio com altas taxas de transferência), RFID (é uma tecnologia de comunicação sem fio baseada em identificação por ondas de rádio frequência) e etc, pode-se citar também, a melhoria no protótipo da luva, com a adesão de elementos mais maleáveis e de fácil manuseio, a interação com o Coppélia V-Rep ou até mesmo com algum robô ou sistema automatizado real e por fim a melhoria do processo de calibragem da luva.

REFERÊNCIAS

- ARDUINO. *Arduino Images*. Disponível em: <arduino.cc>. Acesso em: 08 de set de 2015.
- BARBOSA, Simone D. J.; BARBOSA, Bruno S. da Silva. **Interação Humano-Computador**. Rio de Janeiro: Elviesier, 2011.
- BARANIUK, Tui Alexandre Ono. **Garra robótica teleoperada com realimentação de força**. 2014. 60 f. Trabalho de Conclusão de Curso (Graduação) – Universidade Tecnológica Federal do Paraná, Curitiba, 2014.
- BAZZO, W. A., PEREIRA, L. T. do V. **Introdução à Engenharia**, 6. Ed. Florianópolis: Edufsc, 1997.
- BORGES, Carlos A.; CRESPO, Adalberto N.; JINO, Mario; JUNIOR, M.T.A.; SALVIANO, C. F.. **Uma metodologia para Teste de Software no Contexto de Melhoria de Processo**. São Paulo. UNICAMP, 2008.
- CARRARA, Valdemir. **Apostila de robótica**. Universidade de Braz Cubas. [19--?].
- CHIAVENATO, I. **Introdução à Teoria Geral da Administração**. S. Paulo: Mc Graw-Hill, 3ª edição, 1983.
- DAHLHOFF, H.. **Fundamentos da robótica: BSP70**. São Paulo: Festo Didactic, 1993.
- ECOVACS. **Aspirador de pó Deebot**. Disponível em: <Deebot - <http://www.ecovacs.com.br/deebot-ecovacs.asp#ad-image-0>>. Acessado em: 10 de out de 2015.
- ECOVACS. **Limpador de Janelas Winbot**. Disponível em: <Winbot - <http://www.ecovacs.com.br/winbot-ecovacs.asp#ad-image-0>>. Acessado em: 10 de out de 2015.
- ESTEVEES, Gustavo R. P., FERNANDES, Bruno J.T. **Estereoscopia no cálculo de distância e controle de plataforma robótica**. Disponível em: <http://www.decom.ufop.br/sibgrapi2012/e proceedings/wuw/102815_1.pdf> Acesso em: 01 nov 2015.
- FISHEL, Jeremy A.; LOEB, Gerald E. *Bayesian exploration for intelligent identification of textures*. *Frontiers In Neurobotics*, Los Angeles, 2012.
- GROOVER, M. P., *Automation, Production Systems, and Computer Integrated Manufacturing*. 3. Ed., Pearson Education Inc., Prentice Hall, 2008.
- HEWETT, T. T.; BAECKER, R.; CARD, S.; CAREY, T.; GASEN, J.; MANTEI, M.; PERLMAN, G.; STRONG, G.; VERPLANK, W. *Acm sigchi curricula for human-computer interaction*. Relatório Técnico, New York, NY, USA, 1992.

ISO. ISO 8373:1994 – *Manipulating industrial robots* – Vocabulary. 1994.

INC. *The Robotic Industries Association (RIA) definition of a robot*. Disponível em: <<http://www.inc.com/encyclopedia/robotics.html>> Acesso em: 20 out. 2015.

INSTRUCTABLES. *DIY Robotic Hand Controlled by a Glove and Arduino*. Disponível em: <<http://www.instructables.com/id/DIY-Robotic-Hand-Controlled-by-a-Glove-and-Arduino/>>. Acesso em: 08 de nov de 2015.

INMOOV. *Open Source 3D Printed Life-Size Robot*. Disponível em: <<http://inmoov.fr/>> Acesso em: 05 de nov de 2015.

HARVARD GAZETTE. *A New Grasp On Robotic Glove*. Disponível em: <<http://news.harvard.edu/gazette/story/2015/06/a-new-grasp-on-robotic-glove/>> Acesso em: 05 de nov de 2015.

HIRAMA, Kechi. **Engenharia de Software, Qualidade e Produtividade com Tecnologia**. Rio de Janeiro: Campus, 2012.

LAUS, L.P. **Apostila da disciplina de robótica**. UTFPR, 2006

MALAQUIAS, Fernanda Dorneles; ASSOLARI, Karina; NOGAWA, Marcos Likio. **Desenvolvimento de um sistema de controle de um braço robótico de cinco eixos**. 2012. 85 f. Trabalho de Conclusão de Curso (Graduação) – Universidade Tecnológica Federal do Paraná, Curitiba, 2012.

MARGOLIS, Michael. *Arduino Cookbook*. 2. ed. Sebastopol: O'Reilly, 2011.

MARTINS, Agenor. **O que é robótica**. São Paulo: Editora Brasiliense, 2006.

PAOLI, Marcos-A. **Plásticos Inteligentes**. Disponível em <<http://qnesc.sbq.org.br/online/cadernos/02/plastic.pdf>> Acesso em: 15 out. 2015.

PATOGLU, V.; SATICI, A. *Optimal design of haptics interfaces*. In: ZADEH, Mehrdad Hosseini. *Advances in Haptics*. Vukovar: Intech, 2010.

PAZOS, F. **Sistema e Robótica**, 1. Ed, Axcel Books, 2002.

PREECE, Jennifer; ROGERS, Yvonne; SHARP, Helen. **Design de Interação: além da interação homem-computador**. Porto Alegre. Editora Bookman, 2005. 3ª ed. Porto Alegre. Editora Bookman, 2013.

PRESSMAN, Roger S. **Engenharia de Software**. 6ª ed. Porto Alegre: AMG, 2010.

RIBEIRO, Leandro. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. 2010.

ROSÁRIO, João Maurício. **Princípios de Mecatrônica**. São Paulo: Pretience Hall, 2005.

SILVEIRA, P. R., SANTOS, W.E. **Automação e Controle Discreto**. Editora Érica, São Paulo, 1998.

SPECTRAL SYMBOL. *Flex Sensor Features*. Disponível em: <<http://www.spectrasymbol.com/wp-content/themes/spectra/images/datasheets/FlexSensor.pdf>>. Acesso em: 22 out. 2015.

THOMAZINI, Daniel. ALBUQUERQUE, Pedro U. B. **Sensores Industriais – Fundamentos e Aplicações**. 5ª ed. São Paulo: Érica, 2005.

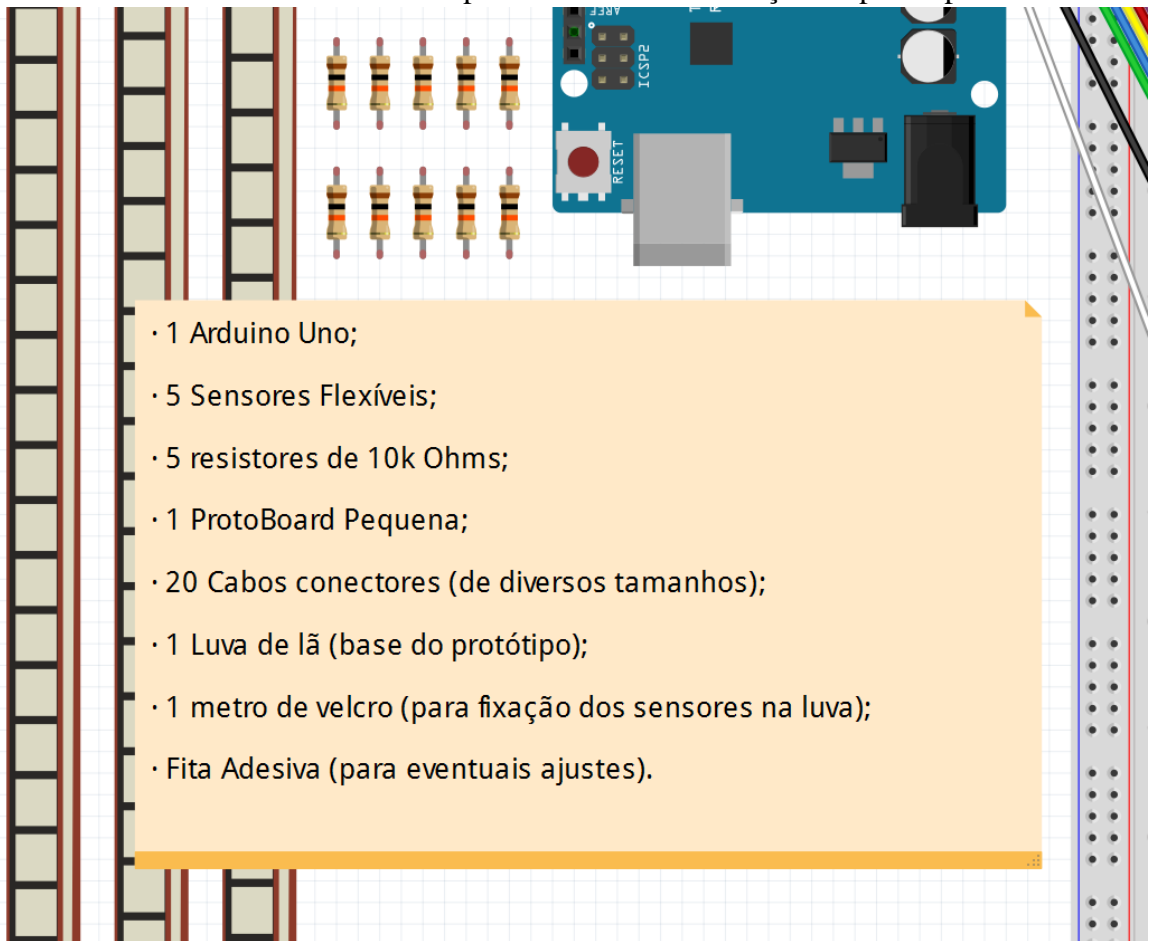
UNITY 3D BLOG. *Unity 3D demonstration*. Disponível em: <www.blogs.unity3d.com/2012/10/30/unity-windows-phone-8-demonstrated-at-microsoft-build-conference> Acesso em: 10 de out de 2015.

VERMASON. *Plastic Bag Features*. Disponível em: <<http://documents.vermason.co.uk/PDF/203531.pdf>> Acesso em: 10 out. 2015.

WORX. **Cortador de grama Robô Landroid M, modelo WG794E**. Disponível em: <<https://www.worxlandroid.com/pdf/instruction-manual-landroid-m-wg794e-en-da-fi-nb-sv.pdf>>. Acessado em: 10 de out. de 2015

ZADEH, Mehrdad Hosseini. *Advances in Haptics*. Vukovar: Intech, 2010. 732 p. Disponível em: <<http://www.intechopen.com/books/advances-in-haptics>>. Acesso em: 31 out. 2015.

Anexo 2 – Lista de componentes usados na criação do protótipo.



Fonte: O Próprio Autor.