

Universidade Federal de Juiz de Fora

Faculdade de Engenharia

Departamento de Circuitos Elétricos

**Gustavo Moraes Lopes**

**Desenvolvimento de um módulo virtual para o ensino de sistemas eletrônicos**

Juiz de Fora

2017

**Gustavo Moraes Lopes**

**Desenvolvimento de um módulo virtual para o ensino de sistemas eletrônicos**

Dissertação apresentada à Universidade Federal de Juiz de Fora, como requisito para obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Leandro Rodrigues Manso Silva

Juiz de Fora

2017

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF  
com os dados fornecidos pelo(a) autor(a)

Lopes, Gustavo.

Desenvolvimento de um módulo virtual para o ensino de sistemas eletrônicos / Gustavo Moraes Lopes. – 2017.

39 f. : il.

Orientador: Leandro Rodrigues Manso Silva

Dissertação (Graduação) – Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Departamento de Circuitos Elétricos, 2017.

1. Eletrônica. 2. Ensino de Engenharia. 3. PjBL. I. Manso, Leandro, orient. II. Título.

**Gustavo Moraes Lopes**

**Desenvolvimento de um módulo virtual para o ensino de sistemas eletrônicos**

Dissertação apresentada à Universidade Federal de Juiz de Fora, como requisito para obtenção do título de Bacharel em Engenharia Elétrica.

Aprovada em: 7 de Dezembro de 2017

**BANCA EXAMINADORA**

---

Prof. Dr. Leandro Rodrigues Manso Silva - Orientador  
Universidade Federal de Juiz de Fora

---

Prof D.Sc. Carlos Augusto Duque  
Universidade Federal de Juiz de Fora

---

Prof. D.Sc. Estêvão Coelho Teixeira  
Universidade Federal de Juiz de Fora

## AGRADECIMENTOS

Agradeço a meus amados pais, Valéria e Adolfo, e aos meus familiares, por me apoiarem e incentivarem durante toda esse período de estudos.

Agradeço aos meus grandiosos amigos do PET Elétrica, por terem estado comigo durante toda essa jornada da faculdade. Tenham certeza que a principal motivação de continuar no curso e hoje estar me formando foi porque vocês estavam lá para me dar a estrutura de que precisava.

Agradeço aos fiéis companheiros de estudos, Solano e Jhulliane, por estarem sempre dispostos a estudar e a ensinar. Obrigado pelo companheirismo de inúmeros trabalhos e provas.

Agradeço aos meus amigos João Tito e Diego, por me introduzirem no mundo da programação, área que pretendo seguir em minha vida profissional.

Agradeço aos meus sócios e amigos, Thiago e Rômulo, pelos anos de amizade e trabalho.

Agradeço aos meus professores, em especial ao meu orientador, Leandro Manso, por toda a dedicação no ensino e por sempre estar ajudando os alunos a alcançarem seus objetivos da melhor maneira possível. São esses os professores que motivam e impulsionam os alunos a serem ótimos profissionais e fazem da faculdade um lugar mais agradável e prazeroso de se estar.

Não obstante, agradeço a todos que já passaram ou ainda estão em minha vida e que torceram para que esse momento pudesse se concretizar.

“A imaginação é mais importante que o conhecimento.”  
(Albert Einstein)

## RESUMO

Este trabalho tem como objetivos o projeto, o desenvolvimento e a implementação de um novo módulo voltado para o ensino de eletrônica, mais especificamente para as matérias de Eletrônica Digital e Microcontroladores. Este, visa a integração entre software e hardware para um maior aproveitamento do aprendizado. Para isto, esse módulo foi pensado para se trabalhar utilizando a metodologia PjBL, onde o aluno tem maior envolvimento na matéria, uma vez que todo seu aprendizado é feito baseando-se num projeto previamente instituído e o curso da matéria é voltado para o mesmo. Paralelamente, pensando na comunidade que porventura venha a utilizar o software e até mesmo melhorá-lo, este foi concebido como um software FOSS, ou seja, gratuito e de código aberto, utilizando apenas ferramentas e linguagens que também seguem o conceito de gratuidade. Esse trabalho de conclusão de curso abordará, de maneira estendida, as motivações, objetivos e inspirações do mesmo. Em conjunto, ele exporá as lacunas que se pretende preencher com a proposta deste módulo. Em seguida, abordará revisões bibliográficas a respeito de PjBL e também a respeito de propostas de novos softwares na melhoria do ensino de Engenharia. Ademais, apresentará a proposta do módulo implementado, com todos os detalhes técnicos relevantes para o bom funcionamento do mesmo. Posteriormente, demonstrará os resultados obtidos com adição de exemplos de uso e funcionamento. Como complemento, este documento trará um tutorial de como implementar esse módulo em outros locais e até mesmo de como aprimorá-lo, incluindo discussões sobre melhorias e sugestões futuras. Esse trabalho termina concluindo os benefícios que esse módulo levará às universidades e aos estudantes que o utilizarem, tanto em caráter teórico quanto em caráter prático, aumentando e desenvolvendo habilidades transversais necessárias para boa prática de Engenharia Elétrica.

Palavras-chave: Eletrônica Digital. Ensino de Engenharia. PjBL. Proposta de novo software.

## **ABSTRACT**

This coursework has as final goals the project and the development of a new module directed towards the electronics education, more specifically towards Digital Electronics and Microcontrollers subjects. It aims software and hardware integration for stronger learning achievements. For this, this module was thought to work using PjBL methodology, where the student has bigger involvement on the subject, once all his or her learning will be based on a previously established project and the subject progress will follow it. In parallel, taking into consideration the community who may use this software, and maybe even improve it, this module was conceived as a FOSS software, that is, free and open source, using only tools and languages that was free as well. This coursework will address, in an extend manner, its motivations, goals and inspirations. Together, it will expose the gaps it intends to fill with this module proposition. Following, it will address bibliographic reviews regarding PjBL and also regarding other software propositions concerning an improvement on Engineering education. Furthermore, it will present the final module proposition, considering all the technical details relevant for a good operation of it. Then, it will demonstrate the obtained results in addition with use examples. In addition to this, this document will hold a tutorial on how to implement this module in other places and even on how to better it, including discussions on future improvements and suggestions. This coursework ends concluding the benefits this module brings to universities and students who are about to use it, both in theoretical and practical approaches, raising and developing transferable skills necessary for good Electrical Engineering practice.

Key-words: Digital Electronics. Engineering Education. PjBL. Proposal of a new software.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama do projeto . . . . .	16
Figura 2 – Conveyor Belt - componente do Multisim. . . . .	17
Figura 3 – Descrição do protocolo de comunicação. . . . .	18
Figura 4 – Tela inicial do Software Principal . . . . .	22
Figura 5 – Software Principal - Vista Frontal . . . . .	22
Figura 6 – Software Principal - Comparaçao com Conveyor Belt . . . . .	23
Figura 7 – Exemplo - Vista lateral com sensor ativado . . . . .	24
Figura 8 – Dimensões da esteira e da caixa . . . . .	26
Figura 9 – Exemplo - Módulo real implementado . . . . .	27
Figura 10 – Fluxograma das ações do Controller . . . . .	28
Figura 11 – Fluxograma da posição inicial . . . . .	29
Figura 12 – Fluxograma do sensor 1 . . . . .	29
Figura 13 – Fluxograma do sensor 2 . . . . .	30
Figura 14 – Exemplo - Bola caindo . . . . .	30
Figura 15 – Fluxograma do sensor 3 . . . . .	31
Figura 16 – Exemplo - Caixa parada no final . . . . .	31
Figura 17 – IntelliJ IDEA - Abrir Projeto . . . . .	36
Figura 18 – IntelliJ IDEA - Estrutura do Projeto . . . . .	37
Figura 19 – IntelliJ IDEA - Nova Biblioteca . . . . .	37
Figura 20 – IntelliJ IDEA - Arquivo <code>lib.xml</code> . . . . .	38
Figura 21 – IntelliJ IDEA - Configurar Classe Principal . . . . .	38
Figura 22 – IntelliJ IDEA - Run / Debug . . . . .	39

## **LISTA DE TABELAS**

Tabela 1 – Protocolo de envio Software Principal → Middleware . . . . .	19
Tabela 2 – Protocolo de envio Middleware → Software Principal . . . . .	19
Tabela 3 – Protocolo de envio Middleware → Controller . . . . .	20
Tabela 4 – Protocolo de envio Controller → Middleware . . . . .	20
Tabela 5 – Posição dos elementos, considerados a partir do início da esteira, o mais à esquerda . . . . .	25

## **LISTA DE ABREVIATURAS E SIGLAS**

FOSS	Free and Open Source Software (Software Gratuito e de Código Aberto)
LSB	Least Significant Bit (Bit Menos Significante)
MSB	Most Significant Bit (Bit Mais Significante)
PjBL	Project-Based Learning (Aprendizagem Baseada em Projetos)
UFJF	Universidade Federal de Juiz de Fora

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>11</b>
1.1	MOTIVAÇÃO . . . . .	11
1.2	OBJETIVOS . . . . .	12
1.3	ESTRUTURA DO TRABALHO . . . . .	13
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA . . . . .</b>	<b>14</b>
<b>3</b>	<b>MÓDULO PROPOSTO . . . . .</b>	<b>16</b>
3.1	CONVEYOR BELT . . . . .	16
3.2	SOBRE A COMUNICAÇÃO . . . . .	18
3.3	PROTOCOLOS DE COMUNICAÇÃO . . . . .	18
3.3.1	Software Principal e Middleware . . . . .	18
3.3.2	Middleware e Controller . . . . .	20
3.4	IMPLEMENTAÇÃO . . . . .	20
<b>4</b>	<b>RESULTADOS . . . . .</b>	<b>22</b>
4.1	MODELAGEM . . . . .	24
4.2	EXEMPLO DE FUNCIONAMENTO . . . . .	27
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>32</b>
5.1	SUGESTÕES FUTURAS . . . . .	32
	<b>REFERÊNCIAS . . . . .</b>	<b>34</b>
	<b>APÊNDICE A – TUTORIAL DE IMPLEMENTAÇÃO . . .</b>	<b>35</b>
A.1	SOFTWARES NECESSÁRIOS . . . . .	35
A.2	AQUISIÇÃO DO CÓDIGO . . . . .	35
A.3	INSTALAÇÃO . . . . .	35

## 1 INTRODUÇÃO

Há alguns anos, disciplinas relacionadas à eletrônica são parte fundamental da formação dos alunos de Engenharia Elétrica. Com o passar do tempo, a necessidade de aprimoramento em certas áreas, fez com que as universidades passassem a oferecer uma enorme gama de disciplinas. É considerado papel da escola de engenharia, possibilitar que os alunos desenvolvam suas carreiras em muitas direções possíveis e durante a mesma, ser capaz de absorver novos conhecimentos (GÓMEZ, 2000).

Atualmente, o ensino de Eletrônica Digital é comumente baseado em um modelo onde o professor transmite seu conhecimento e poucas ferramentas inovadoras são utilizadas para elucidar o assunto. O domínio de disciplinas como esta, requer um apelo prático somado à toda a teoria absorvida pelo aluno, o que nem sempre acontece. É possível listar alguns agentes como os principais empecilhos para esse modelo: o alto custo de softwares para simulações ou a incompletude dos mesmos e também o alto custo e complexidade de se montar experimentos práticos capazes de emular situações reais, como, por exemplo, uma linha de produção industrial.

Por se tratar de tecnologia, é aumentada a responsabilidade na formação de um aluno que seja capaz de se adaptar mais facilmente ao ambiente profissional. Alternativas vêm sendo testadas na área de Engenharia, como a que sugere a utilização da técnica de Aprendizagem Baseada em Projetos (do inglês Project-Based Learning - PjBL). O PjBL cria um ambiente onde o aluno pode aprender baseado no conhecimento prévio e, dentro de um contexto real, reforçar o conhecimento através de trabalho em pequenos grupos (GARCIA, 2014).

Considerando esse contexto, o estudante desempenha um papel de protagonista no seu próprio aprendizado, podendo contribuir, em alguns casos, com a melhora do próprio curso. Este trabalho pretende preencher uma lacuna, causada por motivos variados, financeiros ou institucionais, entre o que se aprende na sala de aula e sua correspondência no mundo real.

### 1.1 MOTIVAÇÃO

Um dos grandes desafios do século XXI, no que tange o ensino em Engenharia, é a adequação de um novo sistema de estudo onde os alunos sejam desafiados a se esforçar e a aprender a lidar com o novo, sejam experiências ou problemas, a fim de desenvolver competências transversais indispensáveis ao mercado de trabalho. Não há como negar que as mudanças ocorridas no final do século XX impactaram profundamente a maneira como é exercida a Engenharia (SILVEIRA, 2005).

A realização de atividades práticas, aliadas à toda a teoria ensinada em sala de

aula, é de fundamental importância para a formação de um profissional de Engenharia. Ainda assim, nota-se que as formas de ensino de inúmeras instituições mundo afora estão, há muito, ultrapassadas: os métodos de educação ainda são baseados em aprendizagem passiva e, não raras vezes, são transmitidos aos alunos por professores cuja instrução para lecionar se restringe, somente, às experiências vividas por este enquanto aluno (FELDER, 2005).

No caso do curso de Engenharia Elétrica/Eletrônica, em algumas disciplinas específicas, como Eletrônica Digital e Microcontroladores, é bastante comum a utilização de softwares de simulação como o Multisim e o Proteus, que possuem alguns modelos de simulação de motores, displays, e, inclusive, alguns mais complexos, como no caso do Multisim, que possui um modelo de correia de transporte e tanque de retenção.

Esses modelos são muito úteis no ambiente de simulação, pois, dessa forma, o aluno pode utilizar os conhecimentos para realizar o controle da operação dos mesmos. Porém, ainda há a necessidade de que o aluno implemente, na prática, utilizando circuitos eletrônicos “reais” essas funcionalidades, o que, com a utilização desses softwares, não é possível.

Surge, então, a necessidade de desenvolver sistemas ou módulos, a custo reduzido, capazes de emular os sistemas físicos para a implementação prática desses sistemas de controle. Uma possível solução seria desenvolver ambientes virtuais que simulem essas plantas e possuam comunicação com o mundo real, para que, dessa forma, o aluno possa aplicar os seus circuitos de maneira a controlar essa planta.

## 1.2 OBJETIVOS

Dentro do contexto citado acima, o objetivo desse trabalho é o desenvolvimento de um sistema que implemente a simulação virtual e gráfica de uma planta industrial e possua comunicação com circuitos físicos de maneira a interagir com essa simulação.

O sistema desenvolvido contará com dois módulos principais: o primeiro deles é um software FOSS, ou seja, gratuito e de código aberto, que realize a simulação da planta e tenha comunicação com o segundo módulo, que é composto de um microcontrolador capaz de realizar essa comunicação e externar para o circuito as entradas e saídas do modelo.

A escolha de se fazer uma aplicação FOSS vai ao encontro de se beneficiar toda a comunidade que por ventura venha a usufruir do mesmo, além de possibilitar uma melhora contínua, parte importante da ideia por detrás da plataforma.

O modelo escolhido para implementação foi o modelo de correia de transporte (Conveyor Belt) do Multisim e a comunicação com o microcontrolador será serial RS232. A interface entre o microcontrolador e o circuito de controle será dada de maneira paralela,

através de sinais digitais de tensão.

### 1.3 ESTRUTURA DO TRABALHO

Para atingir os objetivos propostos na Seção 1.2, esse trabalho foi estruturado sequencialmente da seguinte maneira:

**Capítulo 2** Revisão bibliográfica de diversos outros trabalhos segmentados nessa área, tanto na parte de criação de sistemas, quanto na parte de PjBL

**Capítulo 3** Módulo proposto para sanar os objetivos da Seção 1.2, contendo a implementação do projeto, discussões sobre como ele foi estruturado, quais os protocolos de comunicação ele segue e outras considerações a respeito do mesmo

**Capítulo 4** Módulo final alcançado

**Capítulo 5** Conclusão e sugestões futuras de melhorias

**Apêndice A** Tutorial de como repetir o projeto e/ou expandi-lo

## 2 REVISÃO BIBLIOGRÁFICA

Foram analisados vários trabalhos e artigos para uma fundamentação teórica e uma maior compreensão do tema.

Vários autores corroboram a necessidade de uma modificação nos estudos de engenharia. É comum acordo que é necessário a utilização de novas ferramentas que auxiliem o progresso da educação de um jovem engenheiro. O conhecimento deve ser construído através de experiências ativas do aluno, em que o mesmo necessita de mostrar proatividade. (OLIVEIRA, 2016).

Ademais, utilizam-se ainda, majoritariamente, projetos pedagógicos tradicionais (FELDER, 2003), com pouca relação com o contexto atual e fraca integração entre os componentes curriculares; são débeis as correlações entre teoria e prática, acarretando contato tardio com o ambiente profissional. Atualmente, além de conhecimento técnico, o profissional de engenharia deve possuir outras competências, denominadas transversais, representadas por atributos que não possuem relação direta e imediata com a capacidade técnica de um engenheiro, porém são fundamentais para o exercício profissional. Portanto, é natural que surjam medidas para alterar a educação em engenharia. (CARVALHO, 2014)

Dentre os métodos ativos de aprendizagem, o mais utilizado no nosso contexto atual é a Aprendizagem Baseada em Projetos - PjBL (THOMAS, 2000), (RAILSBACK, 2002) que constitui uma estratégia de organização da construção do conhecimento ao redor de projetos. O PjBL é sinônimo de “Aprendizado Profundo”, ou seja, fornece um enfoque baseado em indagações para engajar os alunos com questões e conflitos que sejam ricos, reais e relevantes às suas vidas. Um projeto bem planejado remete os alunos envolvidos a um encontro direto com os conceitos e princípios de uma disciplina. (CARVALHO, 2014)

Seguindo os caminhos propostos no Capítulo 1, o uso da metodologia PjBL é bastante incentivada por autores, além de mudanças nos métodos de aprendizagem. Necessita-se atualmente modificar a estrutura de formação utilizada nas universidades e repensar o modo como o ensino é praticado, para aproximar o estudante da realidade que encontrará no mercado de trabalho. (COMMITTEE ON ENGINEERING EDUCATION, 2005)

Simultaneamente a essa mudança do perfil do engenheiro buscado pelo mercado, ocorre a mudança do perfil do aluno ingressante nos cursos de engenharia. Tais alunos, já são muito inseridos numa sociedade muito desenvolvida tecnologicamente, gerando um conflito entre as visões dos estudantes e do modelo amplamente utilizado pelas universidades. O modelo pode se tornar desinteressante, assim desmotivando os portadores desse novo perfil durante a graduação. Cabe à ‘universidade atual’ a discussão, e resolução de tal conflito, para que a ‘universidade do futuro’ seja mais eficiente e motivadora, gerando

um profissional mais completo e adequado as demandas de mercado. (VIANNA, 2014)

Desta forma, torna-se fundamental o desenvolvimento de métodos ativos de aprendizagem, onde o estudante não é mais receptor da informação, mas sim modificador e criador desta, desenvolvendo o conhecimento técnico em sintonia com a demanda das indústrias, simultaneamente ao desenvolvimento das competências transversais (PRINCE, 2004)

No âmbito do uso de softwares no ensino, Sandro Jucá sustenta o uso de alguns softwares nos contextos de ensino-aprendizado, sejam estes específicos ou não.

(...) desenvolveram-se softwares específicos para serem utilizados em contextos de ensino-aprendizagem, o que não afasta o fato de que vários softwares desenvolvidos para outras finalidades, também são utilizados no processo de ensino-aprendizagem.

Neste momento histórico, as novas tecnologias mostram que, quando utilizadas adequadamente, auxiliam no processo da construção do conhecimento, tornando o processo de ensino-aprendizagem mais estimulante e mais eficaz. (JUCÁ, 2006).

Por fim, o uso de práticas embasadas na simulação de circuitos eletrônicos vem sendo cada vez mais difundido com o passar dos anos (SEDRA & SMITH, 2007), sendo o foco nestas, portanto, importante.

### 3 MÓDULO PROPOSTO

Como dito anteriormente, o presente trabalho propõe o desenvolvimento de um módulo de simulação de plantas industriais que seja capaz de se comunicar com o mundo externo através de sinais digitais de tensão. Um diagrama contendo a estrutura do módulo proposto é mostrado na Figura 1.

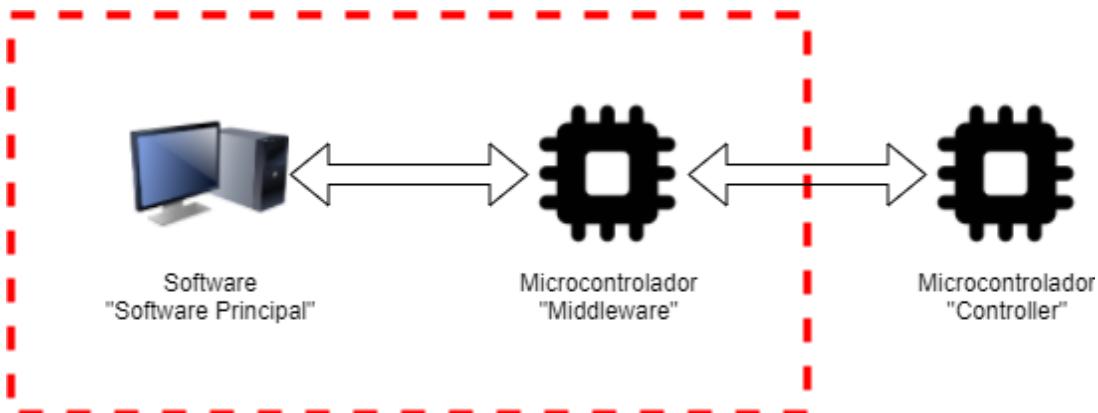


Figura 1 – Diagrama do projeto

Na Figura 1, a parte envolvida pelo retângulo vermelho é o módulo proposto. Pode-se perceber as partes constituintes do mesmo, que são: o Software principal, que será executado no PC e será responsável pela simulação da planta e o Middleware, que será executado em um microcontrolador e é responsável pela interface entre a simulação e o mundo real. A seguir, serão descritos com mais detalhes, as partes constituintes do projeto.

#### 3.1 CONVEYOR BELT

Primeiramente, será explicado o módulo escolhido para ser implementado na simulação, que será executado no computador. O modelo é o Conveyor Belt, do software Multisim. Ele apresenta 5 atuadores (entradas) e 4 sensores (saídas), e seu funcionamento é bastante simples.

Utilizando os atuadores, pode-se mover a esteira para frente, para trás, pará-la, alterar sua velocidade e liberar bolas de um compartimento. Em conjunto, as saídas fornecem valores dos 4 sensores da esteira. Seu esquemático é apresentado na Figura 2.

Este modelo é muito útil para o estudo de diversas disciplinas, como Eletrônica Digital, em que os conceitos a respeito de circuitos sequenciais devem ser empregados afim de controlar a movimentação da caixa sobre a esteira e a funcionalidade do dispensador de bolinhas.

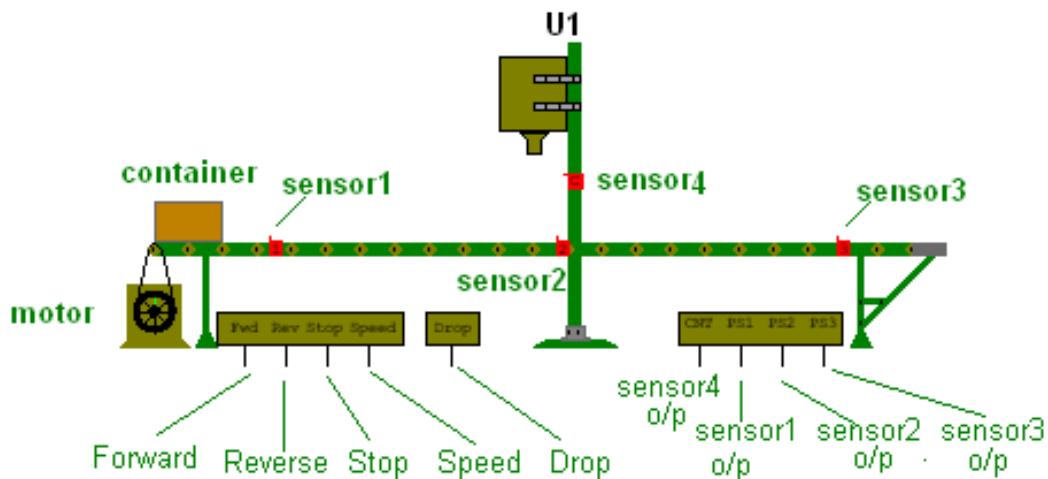


Figura 2 – Conveyor Belt - componente do Multisim.

Fonte: <https://www.engineersgarage.com/contribution/ambhatt/conveyor-belt-controller>

Os sinais relativos aos atuadores são mostrados abaixo. Tipo [D] para digital e [A] para analógico:

[D] **Fwd:** Ativo em '1'. Move a esteira para a direita;

[D] **Rev:** Ativo em '1'. Move a esteira para a esquerda;

[D] **Stop:** Ativo em '1'. Pára a esteira;

[A] **Speed:** Controla a velocidade da esteira;

[D] **Drop:** Ativo em '1'. Enquanto ativo, libera bolas do compartimento

Para os sensores, temos:

**CNT:** Pulso em '1' quando uma bola passa pelo sensor.

**PS1:** Ativo em '1' quando a caixa está sobre o sensor.

**PS2:** Ativo em '1' quando a caixa está sobre o sensor.

**PS3:** Ativo em '1' quando a caixa está sobre o sensor.

Dessa maneira, a comunicação entre o Software Principal e o Middleware é responsável por enviar/receber os valores de todos os sensores e todos os atuadores a cada instante de tempo.

### 3.2 SOBRE A COMUNICAÇÃO

Para a comunicação entre o Software Principal e o Middleware, escolheu-se por usar uma topologia de comunicação serial RS232, uma vez que os computadores modernos não apresentam interfaces de comunicação paralela.

Já entre os microcontroladores Middleware e Controller, escolheu-se por usar uma comunicação paralela, através de sinais digitais de tensão, uma vez que o circuito Controller será, na maioria das vezes, criado por estudantes que estão iniciando no mundo de Eletrônica Digital e não têm amplos conhecimentos de protocolos. Além disso, essa escolha faz com que o módulo aqui proposto seja mais parecido com o Conveyor Belt, elemento inspirador desse trabalho.

### 3.3 PROTOCOLOS DE COMUNICAÇÃO

Nesta seção, é explicado o protocolo de comunicação utilizado para a comunicação entre as partes. O esquemático do projeto pode ser visto na Figura 3, que mostra o protocolo de maneira reduzida. Essa figura é uma expansão, mais detalhada, da Figura 1.

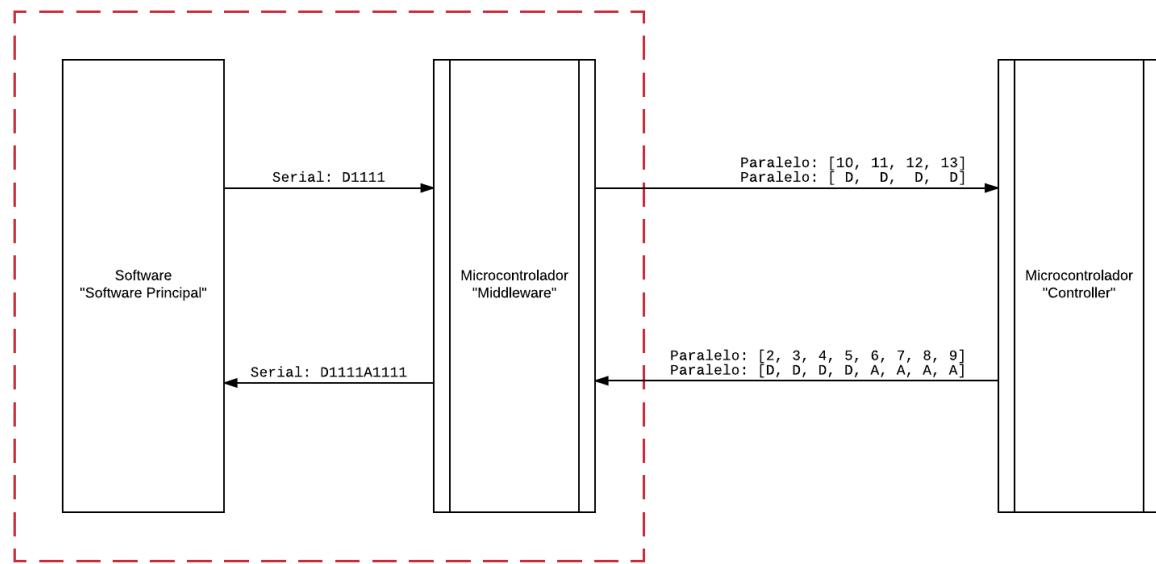


Figura 3 – Descrição do protocolo de comunicação.

#### 3.3.1 Software Principal e Middleware

O envio de informações do Software Principal para Middleware é feita de forma serial, enviando a seguinte string:

*Software Principal → Middleware:*

Dxxxx, [String de 5 posições]

em que 'D' significa que os valores que seguem são sinais digitais e 'x' é o valor de cada sensor, como explicado na Tabela 1.

Tabela 1 – Protocolo de envio Software Principal → Middleware

String	Posição na String	Significado
D	0	Caractere de controle
x	1	Sensor 1
x	2	Sensor 2
x	3	Sensor 3
x	4	Sensor 4

Já o envio de informações do Middleware para o Software Principal pode conter, adicionalmente, valores “analógicos”, relativos ao controle de velocidade da esteira. Neste caso, são precedidos pelo caractere 'A'.

***Middleware → Software Principal:***

DxxxxAyyyy, [String de 10 posições]

em que, 'D' significa que os valores que seguem são sinais digitais, 'x' é o valor de cada atuador, 'A' significa que os valores que seguem são analógicos e 'yyyy' é o valor analógico.

A Tabela 2 explicita o significado de cada bit da string.

String	Posição na String	Significado
D	0	Caractere de controle
x	1	Mover para frente
x	2	Mover para trás
x	3	Parar
x	4	Liberar bolas
A	5	Caractere de controle
y	6	Velocidade (MSB)
y	7	Velocidade
y	8	Velocidade
y	9	Velocidade (LSB)

Tabela 2 – Protocolo de envio Middleware → Software Principal

Esse protocolo de comunicação é simples, porém bem flexível para a utilização nesse contexto, pois permite o envio de qualquer número de sinais digitais e analógicos em sequência.

### 3.3.2 Middleware e Controller

A comunicação entre o Middleware e o Controller é feita de forma paralela. A pinagem utilizada para cada informação está mostrada nas Tabelas 3 e 4.

*Middleware → Controller:*

Pinos	Significado
10	Valor do Sensor 1
11	Valor do Sensor 2
12	Valor do Sensor 3
13	Valor do Sensor 4

Tabela 3 – Protocolo de envio Middleware → Controller

*Controller → Middleware:*

Pinos	Significado
2	Mover para frente
3	Mover para trás
4	Parar
5	Liberar bolas
6	Velocidade (MSB)
7	Velocidade
8	Velocidade
9	Velocidade (LSB)

Tabela 4 – Protocolo de envio Controller → Middleware

## 3.4 IMPLEMENTAÇÃO

O software foi, inicialmente, pensado para ser desenvolvido utilizando Processing<sup>1</sup>, que é uma linguagem utilizada para fáceis, porém complexas, implementações gráficas. O ambiente de programação e linguagem disponibilizado pela empresa, todavia, são bastante limitados e não tinham todos os recursos necessários para a implementação desse projeto.

Processing, contudo, também é um software FOSS. Isso possibilitou a apropriação de parte de seu código e a implementação deste em um projeto diferente, que é o caso desse trabalho.

Como Processing é escrito em Java, a linguagem escolhida para o software desenvolvido também foi Java, por simplicidade de importação do código fonte. Portanto, nesse software, Processing virou uma biblioteca e não mais um linguagem.

---

<sup>1</sup><https://processing.org/>

A escolha de se desenvolver tanto a parte do software quanto a parte do microcontrolador se dá por gerar facilidade para o aluno na criação do microcontrolador de controle. Caso o microcontrolador Middleware não fosse parte do projeto, o aluno ficaria na obrigação de saber todo o protocolo de comunicação fornecido pelo Software Principal, além de ter de implantar toda a comunicação do mundo externo ao software, comunicação essa que, uma vez implantada, não tem necessidades de alteração.

Além disso, torna-se desnecessário qualquer bloqueio na escolha do microcontrolador Controller, uma vez que a comunicação do Middleware com ele é feita de forma paralela, utilizando apenas sinais digitais de tensão.

## 4 RESULTADOS

Este capítulo apresentará os resultados obtidos com o módulo educacional proposto, através de exemplos ilustrativos. Na Figura 4 temos a tela inicial do Software Principal. É nele que configuramos a porta COM no qual o Middleware está conectado e seu Baud Rate. Além disso, configuramos também quantos dados o Software Principal e o Middleware irão trocar por segundo. Quanto maior esse número, mais preciso é o programa, porém maior é o número de operações que o microcontrolador Middleware e o processador do PC devem executar, o que pode acarretar num travamento do sistema.

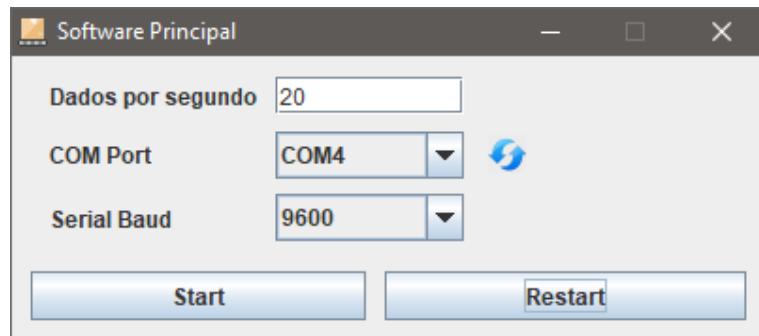


Figura 4 – Tela inicial do Software Principal

Na Figura 5 podemos ver a tela principal do Software Principal. Nela, pode-se notar um componente semelhante ao Conveyor Belt, do Multisim, que possui três sensores de posição da caixa e um sensor de passagem das bolas.

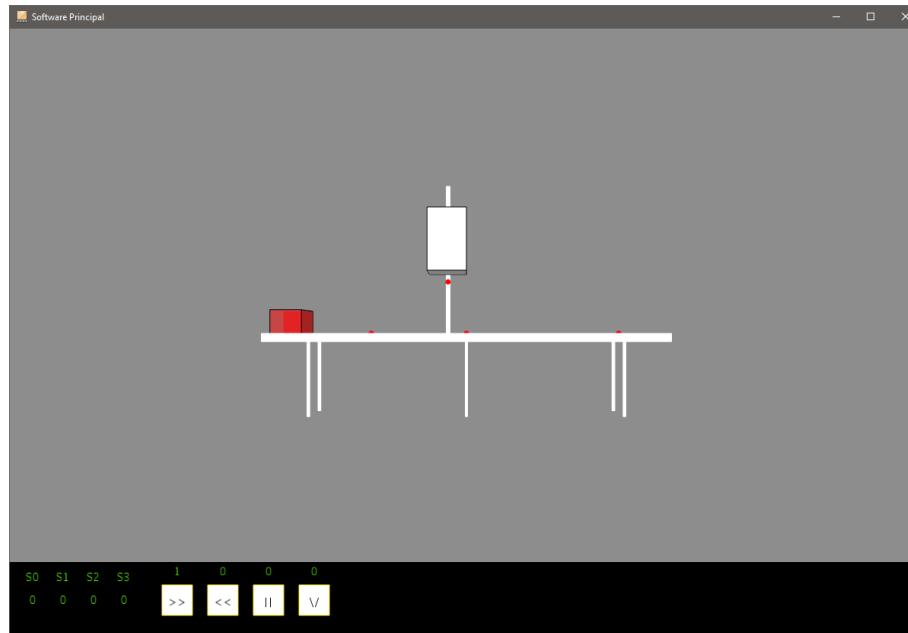


Figura 5 – Software Principal - Vista Frontal

Além disso, percebe-se na parte de baixo da tela, uma barra preta com algumas informações. Essas são as informações dos sensores da esteira e informações dos controles dos atuadores recebidos do Middleware, serialmente. Essa barra pode ser retirada caso assim se desejar.

Pode ser visto na Figura 6 o detalhamento do Software Principal, com os sensores e as informações contidas nele. O mesmo é bem parecido com o Conveyor Belt (Figura 2), tirando o fato desde ter entradas para controle e o Software Principal ter esse controle através de comunicação serial.

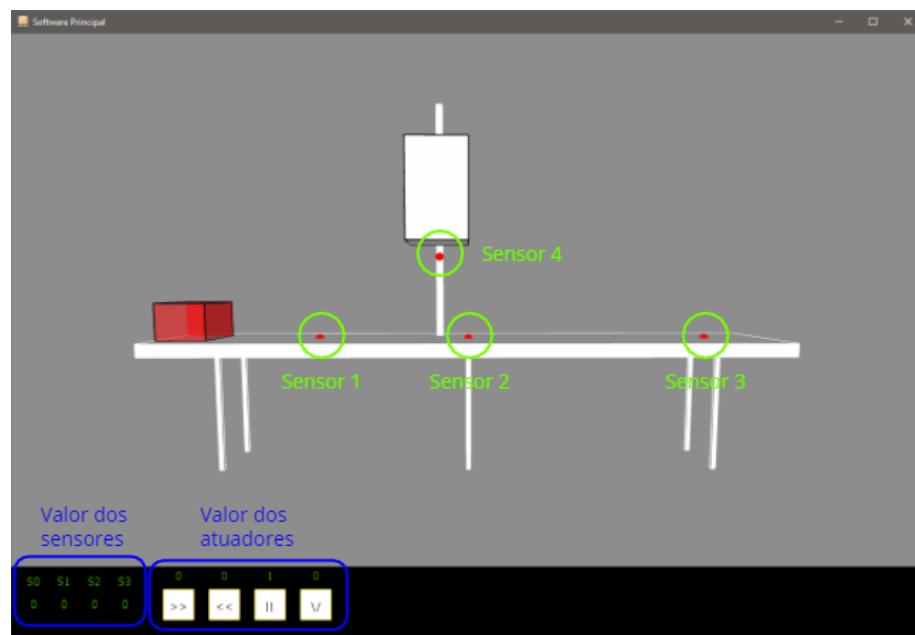


Figura 6 – Software Principal - Comparação com Conveyor Belt

A Figura 7 mostra um exemplo do software em funcionamento. O sensor 1 está ativado (a caixa está sobre ele e na barra preta aparece a informação “1” sob “S1”). Percebe-se também que a caixa está movimentando-se para frente, uma vez que na barra preta tem um número “1” sobre o comando “>>”.

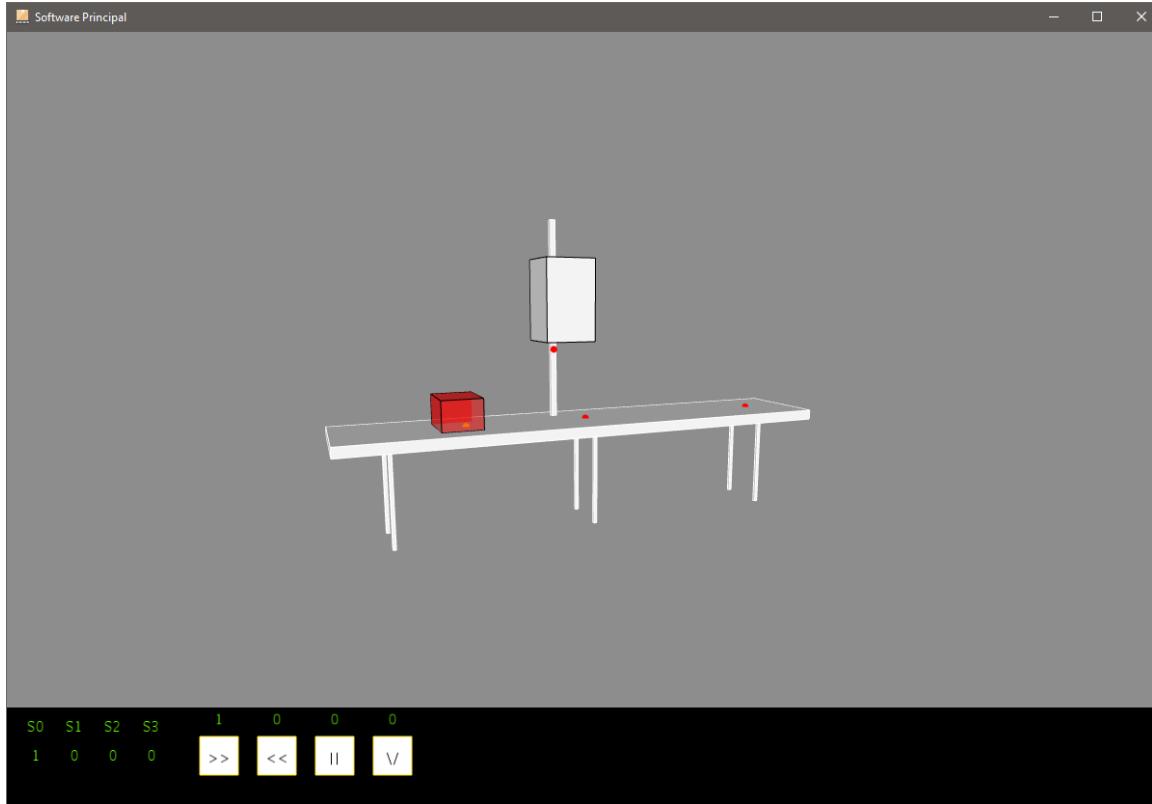


Figura 7 – Exemplo - Vista lateral com sensor ativado

#### 4.1 MODELAGEM

No caso de se querer implementar uma esteira física, para acompanhar a esteira do Software Principal, deve-se levar em conta a velocidade com que a esteira se movimenta.

Na versão atual, o movimento é regido de acordo com as equações abaixo:

$$v_{REAL} = \frac{V_C}{15} * FR * k \quad [\text{pixels/s}] \quad (4.1)$$

onde:

$V_C$  - Velocidade enviada pelo Controller, variando de 0 a 15, sendo 15 a velocidade máxima e 0 parado.

$FR$  - Frame rate que o Processing está rodando, configurado no código, no início do programa.  
No módulo implementado, o frame rate está configurado para 50 repetições por segundo.

$k$  - Fator de correção da velocidade, configurado no código, no início do programa. Elimina variações na velocidade por conta de variações no frame rate. No módulo implementado, o fator de correção da velocidade está configurado para  $30/\text{FRAME\_RATE}$ .

Ou seja, sua velocidade máxima é de 30 pixels/s.

Já a posição da caixa segue a equação abaixo:

$$s = s_0 + v_{REAL} * t \quad [\text{pixels}] \quad (4.2)$$

onde:

$s_0$  - Posição atual da caixa, em pixels.

$v_{REAL}$  - Velocidade, em pixels/s, explicada na Equação 4.1

$t$  - Tempo, em segundos.

As dimensões da esteira podem ser encontradas na Tabela 4.1 e estão discriminados abaixo (consideradas a partir do início da esteira, na posição mais à esquerda):

Tabela 5 – Posição dos elementos, considerados a partir do início da esteira, o mais à esquerda

Final da esteira	100 %	500 px
Sensor 1	25 %	125 px
Sensor 2	50 %	250 px
Sensor 3	90 %	450 px
Caixa	8 %	40 px

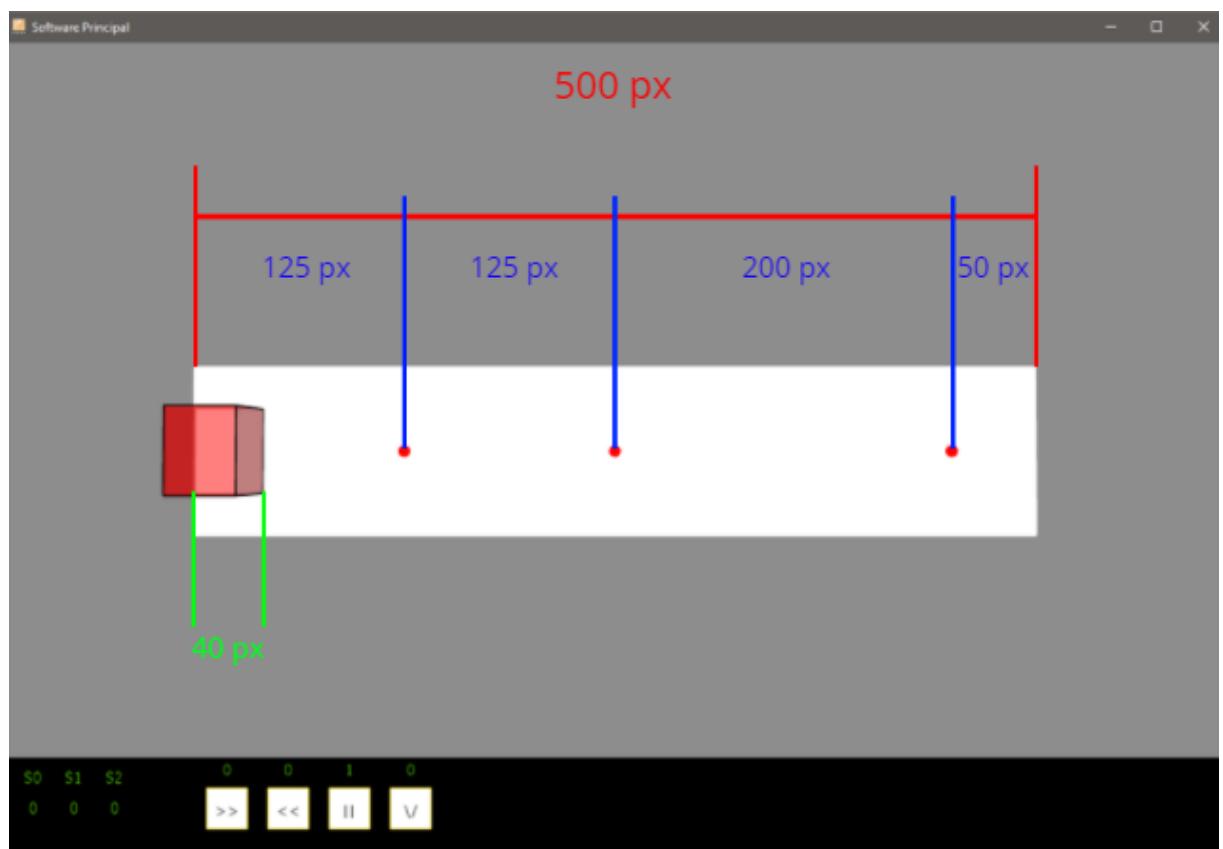


Figura 8 – Dimensões da esteira e da caixa

## 4.2 EXEMPLO DE FUNCIONAMENTO

Essa seção exemplificará a utilização do módulo proposto, detalhando a comunicação entre o Software Principal e Middleware e entre o Middleware e o Controller.

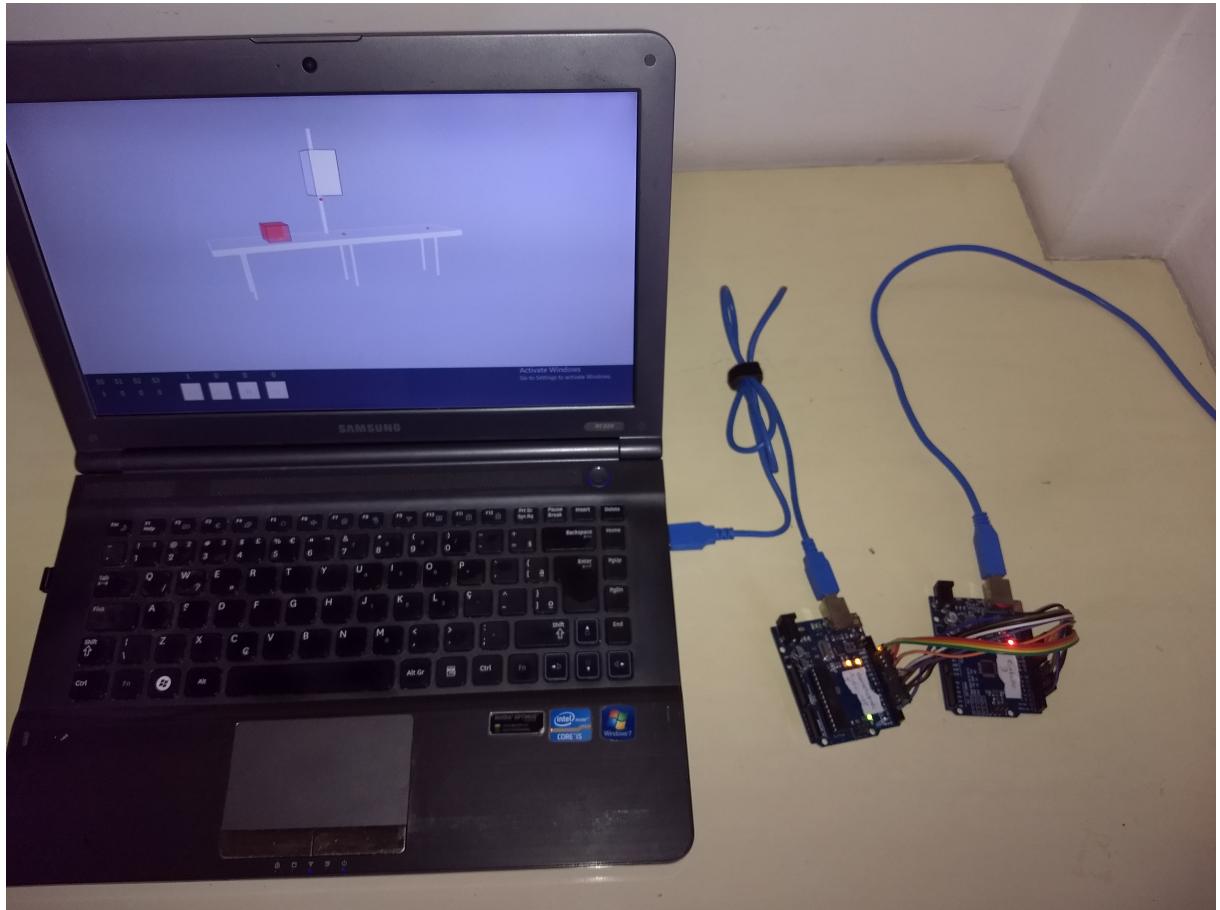


Figura 9 – Exemplo - Módulo real implementado

Na Figura 9 pode-se ver a configuração final real implementada. Nesta figura, vemos o uso de dois microcontroladores: o mais próximo ao computador é o microcontrolador Middleware, enquanto o mais à direita está emulando o Controller, que pode ser programado para executar qualquer controle de movimento da esteira. Vale a pena ressaltar que no lugar do microcontrolador Controller, qualquer tipo de circuito poderia ser utilizado.

A seguir, é mostrado a sequência de movimentos que a caixa deverá executar de acordo com o controle implementado:

1. A caixa inicia a movimentação com velocidade máxima
2. Ao passar pelo sensor 1, ela diminui sua velocidade para metade do valor máximo
3. Chegando no sensor 2, ela pára, recebe 4 bolas, e continua com velocidade máxima

4. Por fim, ao encontrar o sensor 3, ela pára completamente

O fluxograma da Figura 10 apresenta a sequência de passos implementados pelo Controller.

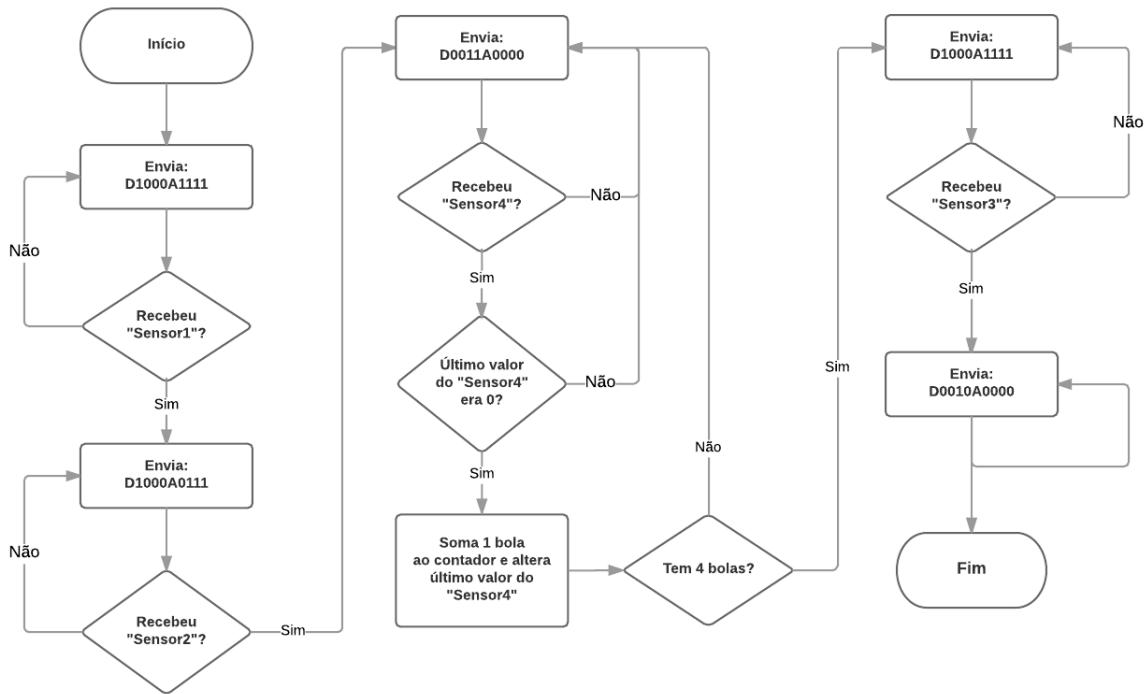


Figura 10 – Fluxograma das ações do Controller

Inicialmente, a caixa encontra-se parada, fora de qualquer sensor, portanto o Software Principal envia o comando “D0000” para o Middleware que, por sua vez, coloca os 4 pinos de saída em nível baixo.

O Controller, analisando essa situação, seta para nível alto o pino 2, relativo a ativação do atuador FWD, para iniciar a movimentação da caixa para a direita, e seta o valor da velocidade máxima, ou seja, coloca em nível alto os pinos 6, 7, 8 e 9. Desse forma, o Middleware envia o comando “D1000A1111” para o Software Principal que inicia o movimento da caixa. Essa sequência de passos é mostrada no fluxograma da Figura 11.

A caixa começa a se mover e, ao atinge o sensor 1, ela tem sua velocidade configurada em metade da velocidade máxima. Essa situação pode ser verificada no fluxograma da Figura 12.

Em seguida, a caixa atinge o sensor 2, local onde ela pára, recebe 4 bolas e continua com velocidade máxima. Essa comunicação é explicada no fluxograma da Figura 13. Nessa situação, porém, toda vez que uma bola cai, ela ativa o sensor 4 por uma fração de tempo,

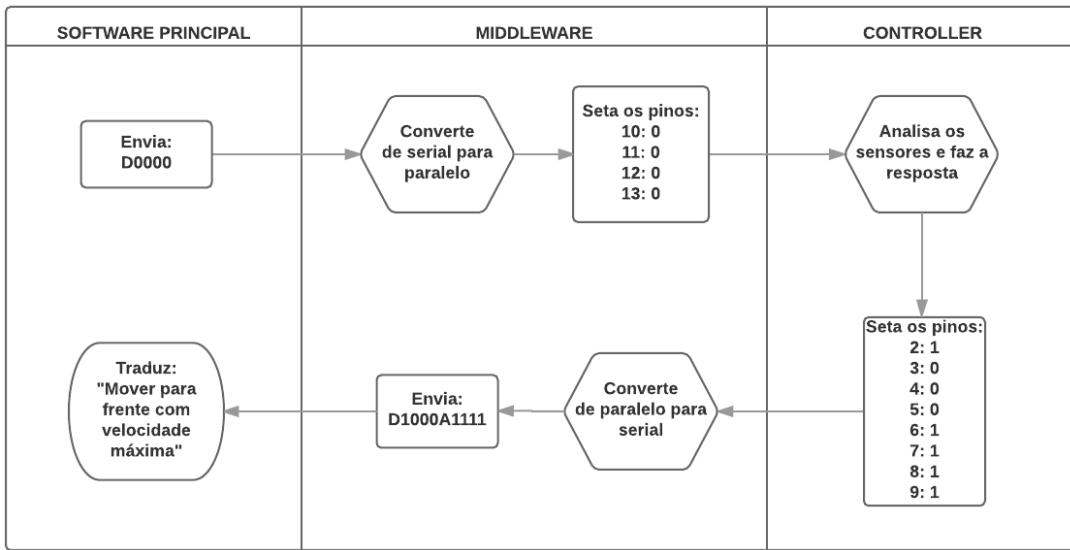


Figura 11 – Fluxograma da posição inicial

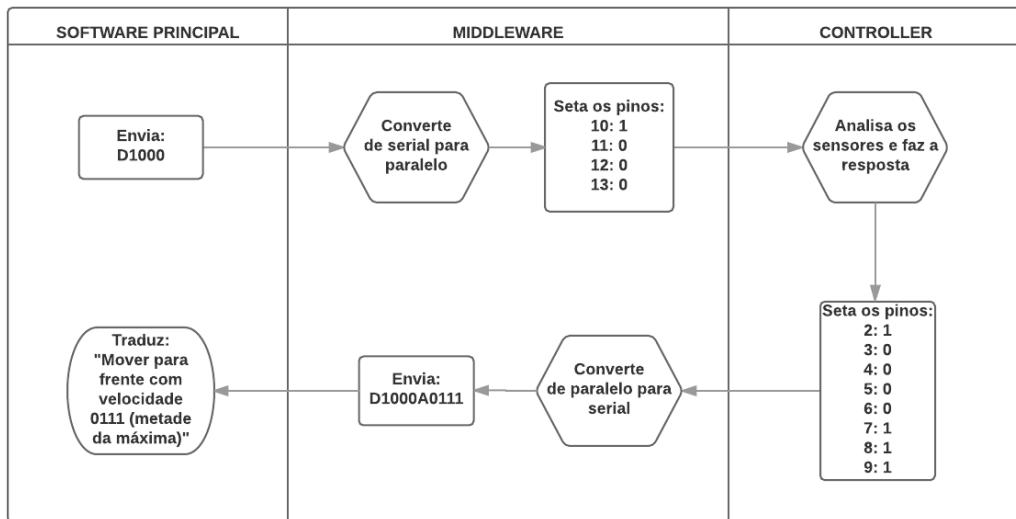


Figura 12 – Fluxograma do sensor 1

ou seja, a string enviada pelo Software Principal alterna entre D0100 e D0101. Pode-se observar uma captura de tela da bola caindo na Figura 14.

Por fim, ao atingir o sensor 3, a caixa pára completamente e a aplicação termina. Pode-se verificar a comunicação existente no fluxograma da Figura 15 e uma captura de tela dessa situação na Figura 16.

De acordo com as situações ilustradas pelo exemplo, pode-se perceber que o sistema proposto realmente consegue implementar essa planta virtual e atender aos comandos de

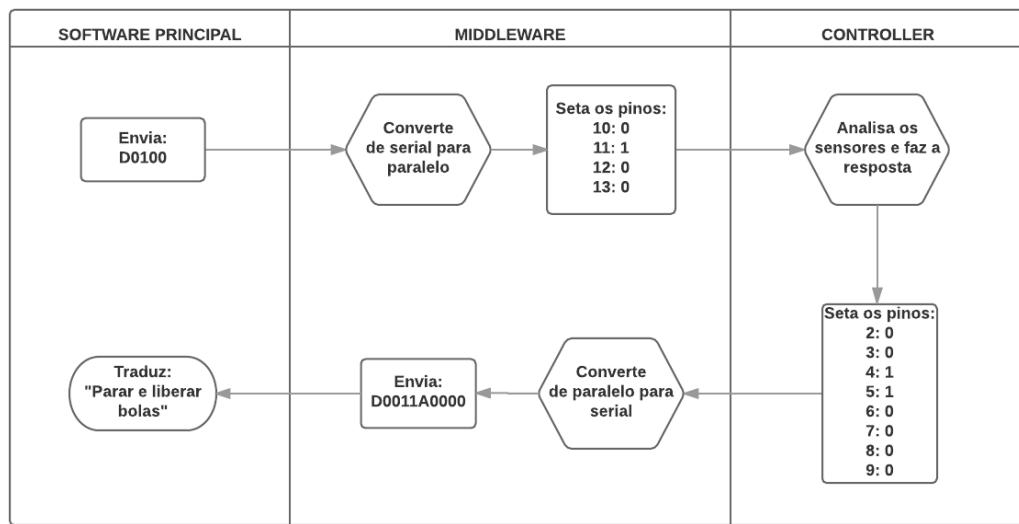


Figura 13 – Fluxograma do sensor 2

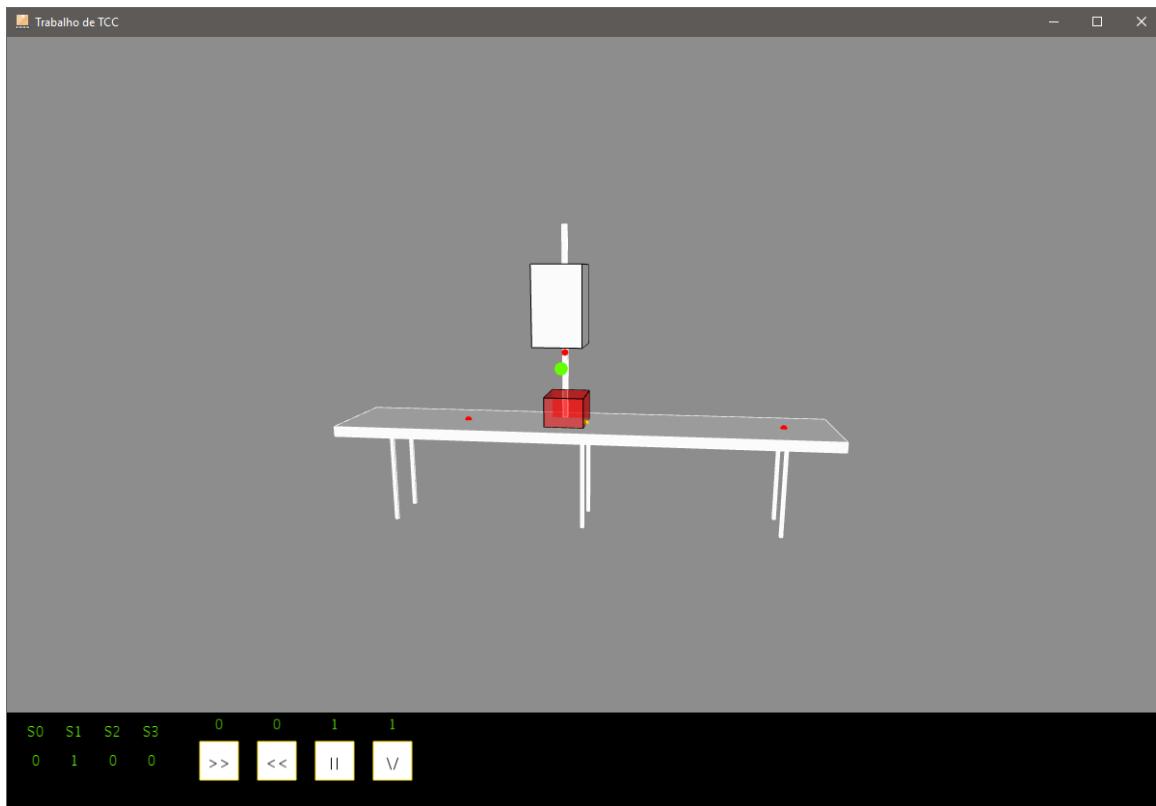


Figura 14 – Exemplo - Bola caindo

controle enviados pelo circuito físico. Com o aprimoramento dessa ferramenta, poderá conseguir-se uma plataforma educacional bastante interessante e de baixo custo para o ensino de eletrônica.

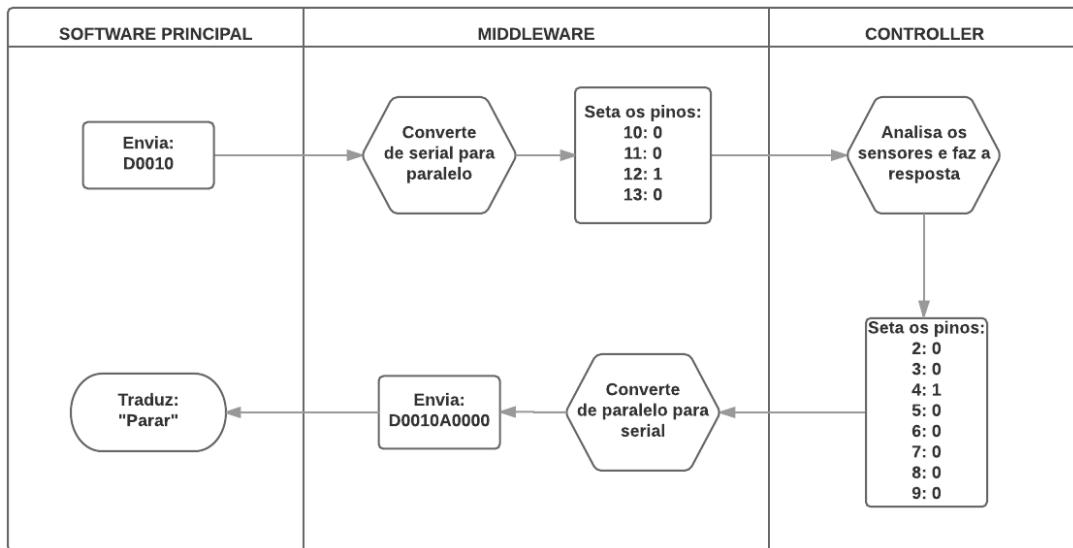


Figura 15 – Fluxograma do sensor 3

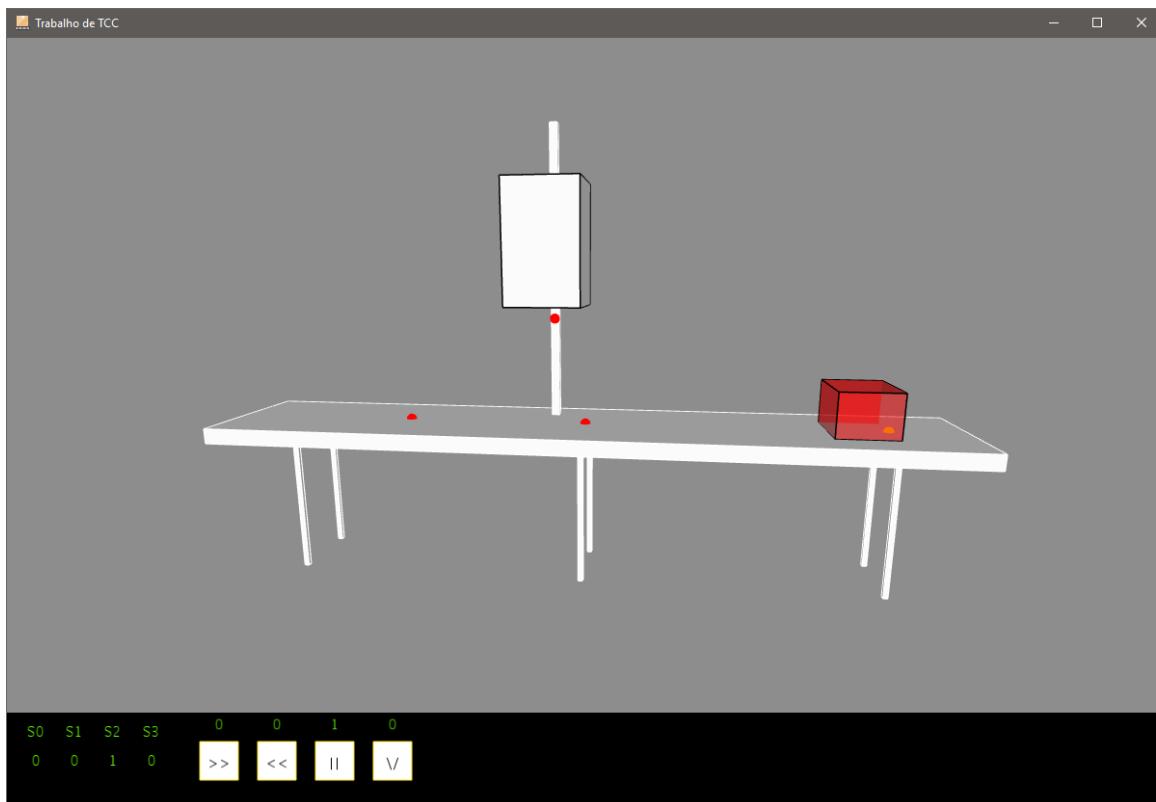


Figura 16 – Exemplo - Caixa parada no final

## 5 CONCLUSÃO

Após a implementação desse módulo, descrito no Capítulo 4, e os estudos da literatura, descritos no Capítulo 2, verificou-se uma enorme vantagem no uso do módulo proposto por esse trabalho.

Foram analisados diversos trabalhos que contemplam a metodologia PjBL e como ela influencia e beneficia o aprendizado dos alunos. Percebe-se que esse trabalho de conclusão de curso contemplou uma metalinguagem, uma vez que o mesmo seguiu a metodologia que ele propõe, o PjBL. Assim sendo, este é uma ratificação do benefício dessa metodologia, comprovado no aprendizado do próprio autor deste documento.

Quando comparado ao componente inspirador desse projeto, o “Conveyor Belt”, da Multisim, o módulo proposto funcionou de maneira igualitária, tendo o diferencial de ser controlado de maneira “física”, através de um microcontrolador. Esse controle mostrou-se bastante simples de implementar, utilizando apenas algumas linhas de código, o que o torna bastante factível para uso por estudantes iniciantes na área de Eletrônica Digital e Microcontroladores. Não obstante, o software abre possibilidades de um controle bem mais complexo, não sendo limitado, portanto, apenas para iniciantes.

Por fim, pode-se dizer que o software implementado foi um sucesso no quesito de utilizar apenas tecnologias gratuitas, tornando-o especialmente útil em universidades que não possuem altos recursos financeiros para aquisição de softwares caros, e, além disso, o software também não tem fins lucrativos, sendo disponibilizado com licença *Mozilla Public License*<sup>1</sup>, que garante total aquisição, modificação, incorporação e distribuição do software, desde que dado os devidos créditos aos desenvolvedores.

### 5.1 SUGESTÕES FUTURAS

Para aqueles que porventura venham a melhorar esse projeto, fica como sugestões futuras alguns pontos não abordados.

- Adicionar ao software a possibilidade de se criar novos componentes via interface gráfica
- Incluir controles de segurança no software como, por exemplo, um botão de “Parada Emergencial”
- Adicionar ao software a possibilidade de editar e/ou configurar o protocolo de comunicação diretamente via software como, por exemplo, caso se deseje aumentar o número de sensores ou alterar os controladores

---

<sup>1</sup><https://www.mozilla.org/en-US/MPL/2.0/>

- Adicionar ao software a possibilidade de criar, editar e subir o código do Middleware diretamente via software. Nessa versão do software, caso se deseje mudar algum parâmetro no Middleware, como, por exemplo, quais portas usar, é necessário utilizar o software proprietário Arduino IDE.
- Incluir um tutorial de uso e como funcionam os protocolos utilizados no próprio software (retirar a necessidade de um “manual de instruções”)
- Melhorar (consideravelmente) o layout do software

Para reproduzir o projeto, basta seguir os passos descritos no Apêndice A - TUTORIAL DE IMPLEMENTAÇÃO .

## REFERÊNCIAS

- [1] CARVALHO, D. A. et al. *Utilização da Estratégia PjBL no Desenvolvimento de um Heliodon como Ferramenta para Educação em Engenharia.* COBENGE 2014. Engenharia: Múltiplos Saberes e Atuações. Juiz de Fora, 2014.
- [2] COMMITTEE ON ENGINEERING EDUCATION. *Educating the Engineer of 2020: Adapting Engineering Education to the New Century.* Committee on the Engineer of 2020 – Phase II. National Academy of Engineering, 2005.
- [3] GARCIA, P. G. *O Ensino de Engenharia e o Método PBL.* Seminário Internacional de Educação Superior 2014: Formação e Conhecimentos. Sorocaba, 2014.
- [4] GÓMEZ RIBELLES, I. L. *Some ideas about the application of the project learning methodology in engineering education.* In: POUZADA, A. S. (ed.). Project based learning: project-led education and group learning. Guimarães: Editora da Universidade do Minho, 2000, p. 51-55.
- [5] FELDER, R. M. *Engineering Education in 2015 (or Sooner) Proceedings of the 2005.* Regional Conference on Engineering Education, December 12-13, Johor, Malaysia, 2005.
- [6] JUCÁ, S. C. S. *Avaliação dos Softwares Educativos na Educação Profissional.* Revista Educação e Tecnologia. Fortaleza, 2006.
- [7] OLIVEIRA, A. P. et al. *Utilização do PjBL na Educação em Engenharia: Projeto e Construção de um Drone.* XLIV Congresso Brasileiro de Educação em Engenharia. Natal, 2016.
- [8] PRINCE, M. *Does Active Learning Work? A Review of the Research Journal of Engineering Education.* Parallel Solution of Sparse Algebraic Equations. IEEE Transactions on Power Systems, Vol.9, No. 2, pp. 743- 799.
- [9] SEDRA, A. S.; SMITH, K. C. *Microeletrônica. 5a. ed..* São Paulo. 2007.
- [10] SILVEIRA, M. A. A. *Formação do Engenheiro Inovador: uma visão internacional.* Sistema Maxwell, PUC, Rio de Janeiro, 2005. The Engineer of 2020: Visions of Engineering in the New Century National Academy of Engineering. National Academies Press, 2004.
- [11] VIANNA, J. T. A. et al. *Utilização da Estratégia PjBL para Aprendizagem de Conversores DC/DC.* COBENGE 2014. Engenharia: Múltiplos Saberes e Atuações. Juiz de Fora, 2014.

## APÊNDICE A – TUTORIAL DE IMPLEMENTAÇÃO

Esse capítulo ensinará todo o processo para se rodar o software, desde a aquisição do código fonte até o modo de operação.

### A.1 SOFTWAREES NECESSÁRIOS

Os softwares utilizados nesse projeto são:

- IntelliJ IDEA <<https://www.jetbrains.com/idea/>>
- Java JDK <<http://www.oracle.com/technetwork/java/javase/overview/>>
- Arduino IDE <<https://www.arduino.cc>>

O sistema operacional testado foi Windows 10 Home, 64 bits.

### A.2 AQUISIÇÃO DO CÓDIGO

Todo o código necessário está hospedado na plataforma GitHub e pode ser acessado através do link:

<<https://github.com/lopes-gustavo/TCC/>>

ou através do comando git:

```
git clone https://github.com/lopes-gustavo/TCC.git
```

### A.3 INSTALAÇÃO

Para editar o código do Software Principal, vamos utilizar a ferramenta IntelliJ IDEA. Após sua instalação, é necessário abrir o projeto, utilizando o código adquirido na Seção A.2. Para tal, os seguintes passos são necessários:

1. Extraia os arquivos baixados para uma pasta vazia do sistema
2. Abra o software IntelliJ IDEA
3. Na janela que abriu, clicar em “Open”, como mostrado na Figura 17
4. Selecione a pasta na qual os arquivos foram extraídos

O projeto foi aberto para ser editado.

Porém, ainda é necessário ainda informar ao IntelliJ IDEA que vamos utilizar os arquivos Processing como bibliotecas. Para tal, siga os passos abaixo.

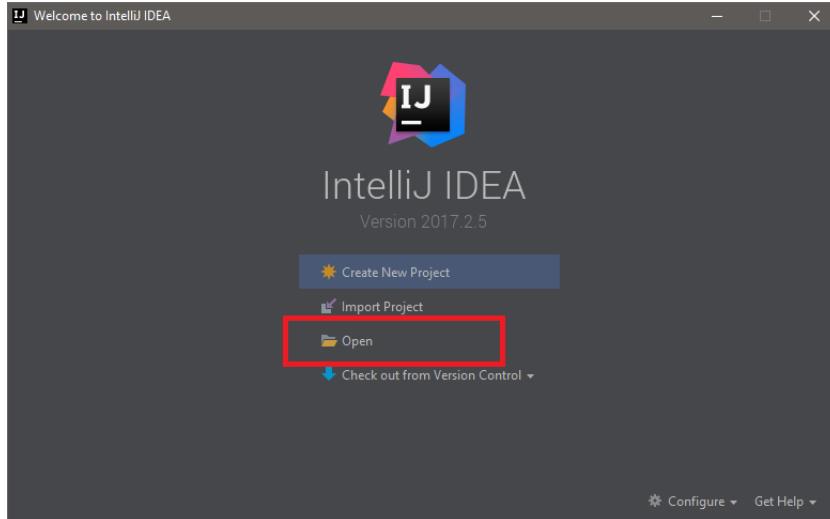


Figura 17 – IntelliJ IDEA - Abrir Projeto

1. Na barra de menu, aperte File > Project Structure (**Ctrl + Alt + Shift + S**)
2. Na janela que abriu, dentro de “Libraries”, clicar em “+” e, em seguida, no opção “Java”, como mostrado na Figura 19
3. Selecione a pasta “lib” dentro da pasta do projeto.
4. Confirme todas as perguntas.
5. Na página principal, abra o arquivo `/.idea/libraries/lib.xml`, de acordo com a Figura 20
6. Adicione a seguinte linha logo acima de `</library>`

```
<jarDirectory url="file://$PROJECT_DIR$/lib" recursive="true"/>
```

Por fim, para rodar o programa, é necessário dizer ao compilador qual a classe principal, seguindo os seguintes passos:

1. No topo e à direita do programa, clicar em “Edit Configurations”, de acordo com a Item A.3.
2. Clicar no “+” e em seguida na opção “Application”
3. Colocar o nome completo da classe principal. No caso, `app.MainApp`.
4. Selecionar que o programa é de instância única, uma vez que não se pode abrir mais de um programa ao mesmo tempo
5. Confirmar a caixa de diálogo

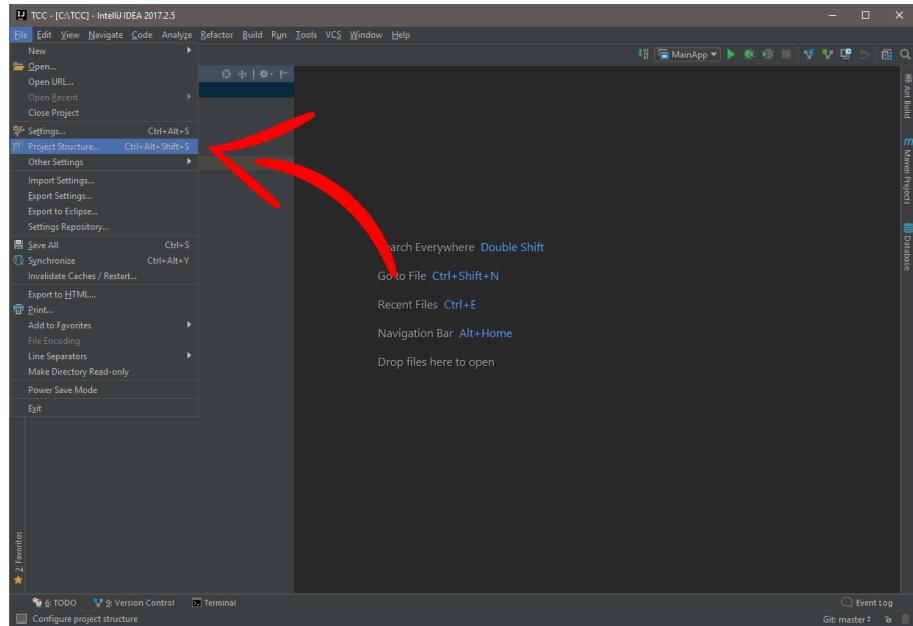


Figura 18 – IntelliJ IDEA - Estrutura do Projeto

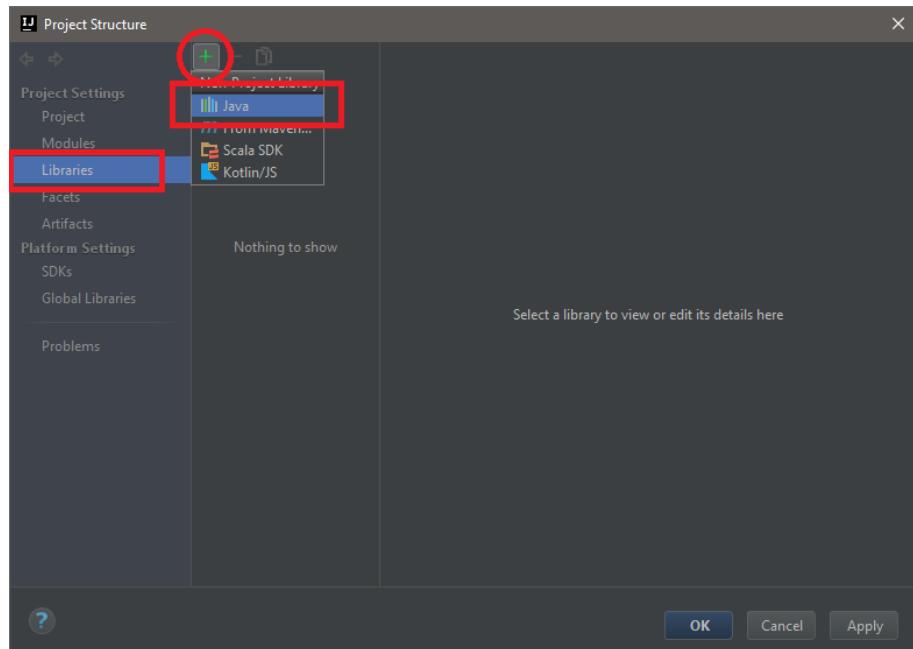


Figura 19 – IntelliJ IDEA - Nova Biblioteca

Para iniciar o programa, basta apertar o “Play” que aparecerá no local marcado na Item A.3

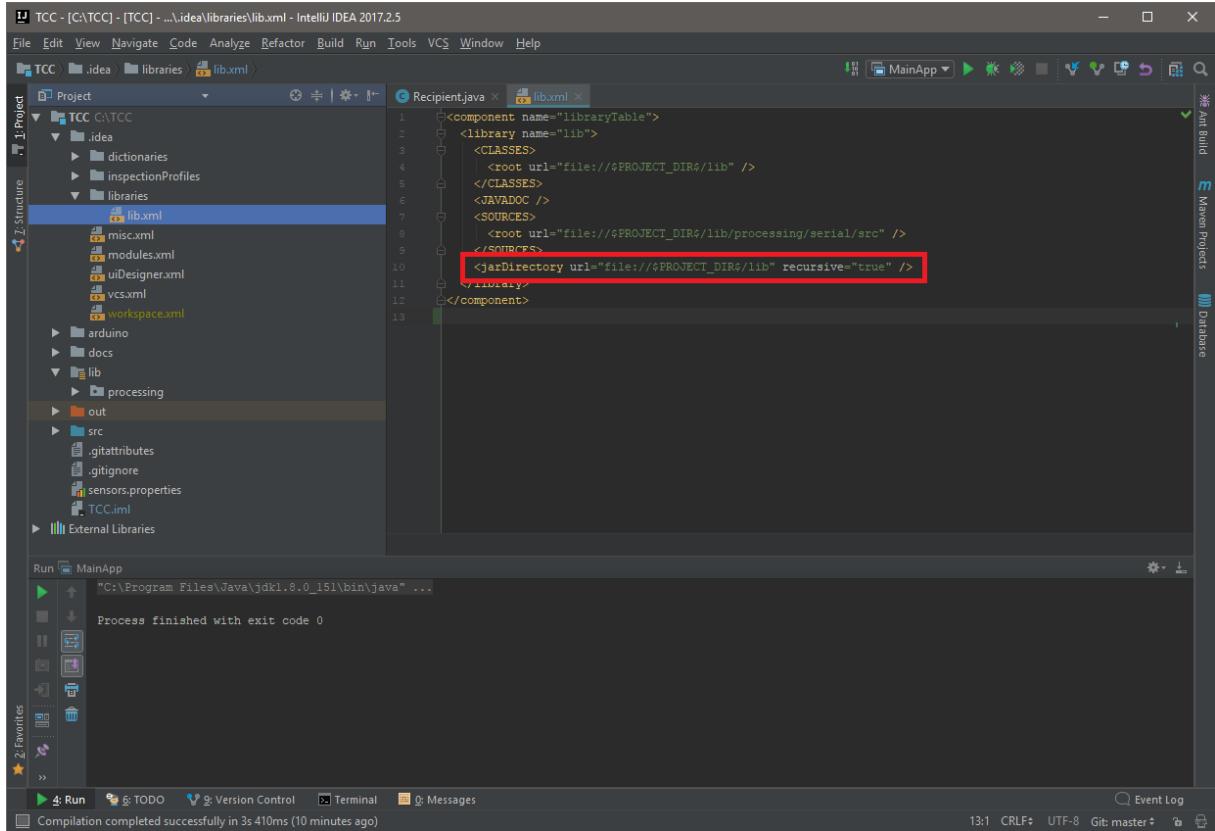


Figura 20 – IntelliJ IDEA - Arquivo lib.xml

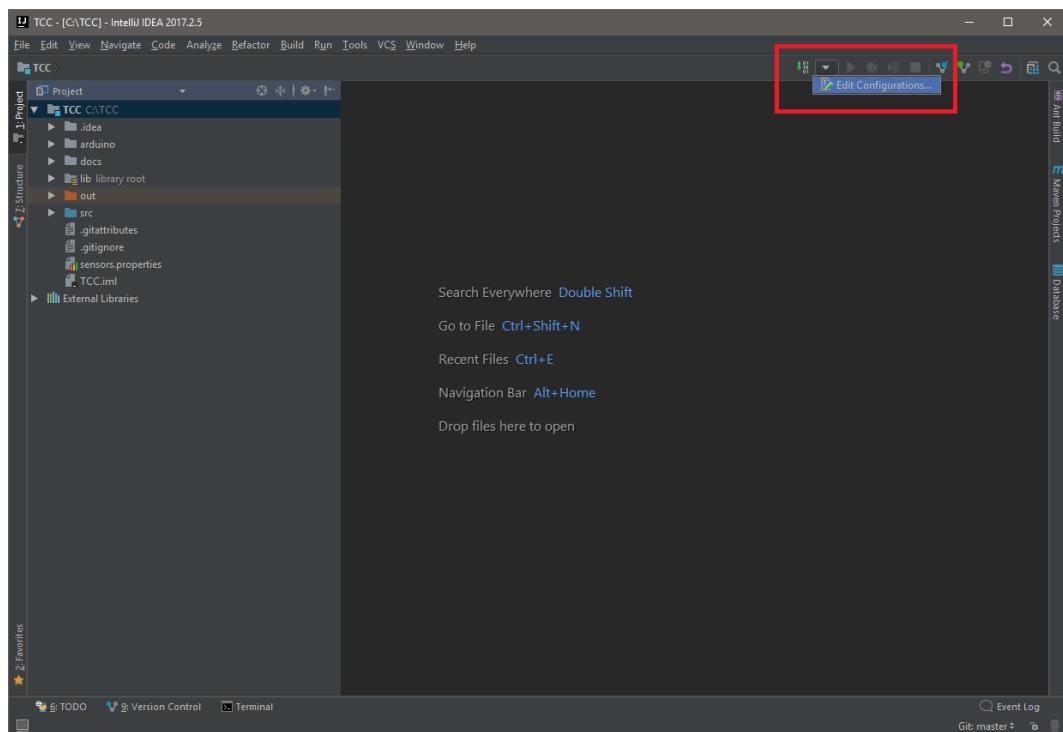


Figura 21 – IntelliJ IDEA - Configurar Classe Principal

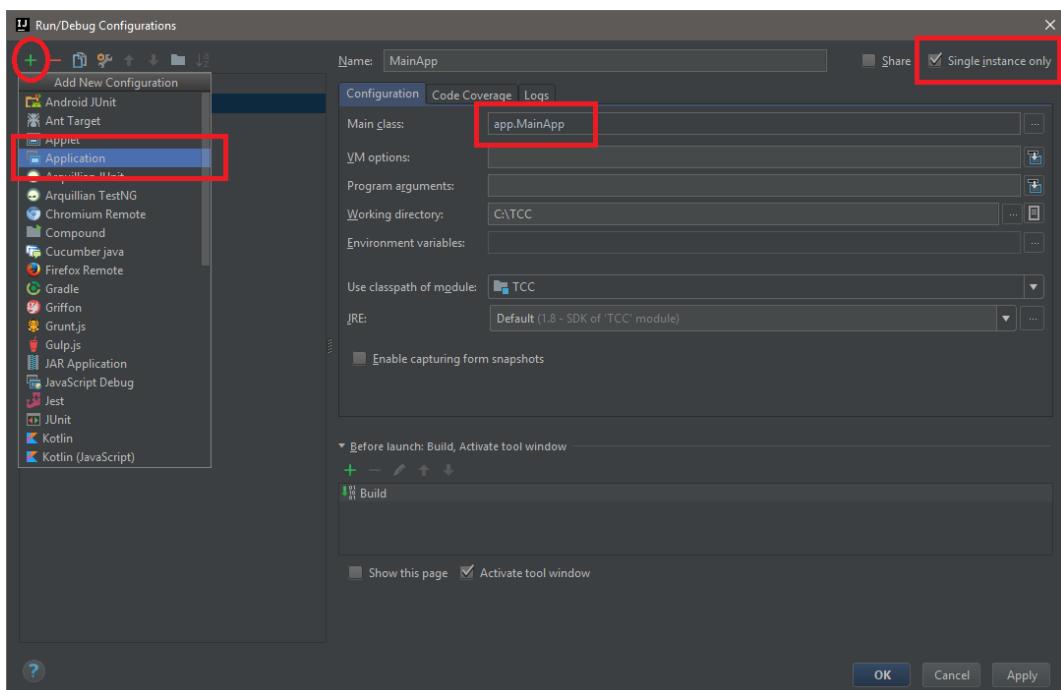


Figura 22 – IntelliJ IDEA - Run / Debug