



UNIVERSIDADE DO ESTADO DO AMAZONAS  
ESCOLA SUPERIOR DE TECNOLOGIA  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

**FELIPE AKIO VIEIRA MAKIHARA**

**SISTEMA SUPERVISÓRIO COM APLICAÇÃO DA INTERNET DAS  
COISAS NO SISTEMA MODULAR DE PRODUÇÃO**

Manaus

2016

**FELIPE AKIO VIEIRA MAKIHARA**

**SISTEMA SUPERVISÓRIO COM APLICAÇÃO DA INTERNET DAS  
COISAS NO SISTEMA MODULAR DE PRODUÇÃO**

Trabalho de Conclusão de Curso submetido à  
Coordenação do curso de Engenharia de Controle  
e Automação da Universidade do Estado do  
Amazonas como parte dos requisitos necessários  
para a obtenção do grau de Engenheiro em  
conformidade com as normas ABNT.

Orientador: Almir Kimura Junior, Me.  
Coorientador: Giovanni Ribeiro Caldeira.

Manaus

2016

**FELIPE AKIO VIEIRA MAKIHARA**

**SISTEMA SUPERVISÓRIO COM APLICAÇÃO DA INTERNET DAS  
COISAS NO SISTEMA MODULAR DE PRODUÇÃO**

Trabalho de Conclusão de Curso submetido à  
Coordenação do curso de Engenharia de Controle  
e Automação da Universidade do Estado do  
Amazonas como parte dos requisitos necessários  
para a obtenção do grau de Engenheiro em  
conformidade com as normas ABNT.

Trabalho aprovado. Manaus, 12 de janeiro de 2017:

---

Giovanni Ribeiro Caldeira.

Co-orientador

---

Me. Charles Luiz Silva de Melo.

Professor

---

André Dias Machado

Convidado

Manaus

2016

## **AGRADECIMENTOS**

Aos meus pais, Arlindo e Marta Makihara, que me apoiaram a minha vida inteira, independente das dificuldades e me incentivaram na escolha da minha área profissional, fazendo o possível para eu seguir meus sonhos. Ao meu irmão, Rodrigo, por me proporcionar conhecimentos relacionados ao curso. À minha namorada, Andrea Nagai, por me apoiar sempre que possível. E à toda a minha família por me ajudar sempre que necessário.

Aos meus amigos, por proporcionarem conhecimento e companheirismo durante este período de aprendizado.

À Universidade do Estado do Amazonas, que oportunizou a minha formação como profissional. À coordenação de Engenharia de Controle e Automação e seu quadro docente, em especial ao coordenador vigente, Me. Charles Melo, pelo auxílio constante no decorrer do curso como professor, coordenador e amigo tanto nas salas de aula quanto fora delas incentivando sempre meu conhecimento e minha formação como profissional da área.

Ao meu Orientador, Me. Almir Kimura Junior, que foi o professor que mais me acompanhou durante todos os anos, me trazendo crescimento tanto pessoal quanto profissional me proporcionando conhecimento da área de Engenharia como professor de disciplinas e como Coordenador de projetos de Pesquisa & Desenvolvimento tanto no GEAR (Grupo de Estudos Aplicados à Robótica) quanto no GRAEST (Grupo de Robótica e Automação), pela oportunidade e auxílio no desenvolvimento da monografia, além de um grande amigo. Ao meu Co-orientador, Giovanni Ribeiro Caldeira, pelo esclarecimento de dúvidas, auxílio e amizade durante os anos de curso e no decorrer do projeto de pesquisa.

*“Uma longa jornada começa com um único passo”  
(Lao-Tsé)*

## RESUMO

Com o avanço da tecnologia, as empresas buscam cada vez mais concorrência entre si tanto no produto final, quanto na forma como é produzida de acordo com a implementação de novos ideais. A quarta revolução industrial introduziu o conceito da Internet das Coisas, possibilitando uma autonomia melhor e, a partir da coleta de dados através da própria Internet ou de determinados sistemas, uma tomada de decisões inteligente. Como a maior parte das aplicações deste estão voltadas à automação residencial, o presente projeto de pesquisa demonstra a utilização da mesma em automação residencial, através do Sistema Modular de Produção sendo ela monitorada em tempo real por um Sistema Supervisório onde tem início em uma página Web que recebe informações relacionadas à compra de produtos, em seguida sua produção é iniciada automaticamente e as mesmas são gravadas em Tags *RFID* encontradas nas matérias-primas com o auxílio do microcontrolador *Raspberry Pi 2*.

**Palavras-Chave:** Internet das Coisas, Sistema Supervisório, Processo Industrial, *RFID*, *Raspberry Pi*.

## **ABSTRACT**

With the advancement of technology, as companies seek more and more competition between themselves in both the final product and the way it is produced according to an implementation of new ideals. The fourth industrial revolution, the concept of the Internet of Things, enabling a better autonomy, from the collection of data through the Internet itself or systems of determination, an intelligent decision making. As a major part of the applications that are geared towards residential automation, the current research project is a part of it in home automation through the Modular Production System to be monitored in real time by a Supervisory System where to start on a page Web that has images related to the purchase of products, then its production starts automatically and how they are recorded in RFID Tags found in the raw materials with the aid of the microcontroller Raspberry Pi 2.

**Key-words:** Internet of Things, Supervisory Systems, Industrial Process, RFID, Raspberry Pi.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>11</b>
1.1	Problema de Pesquisa .....	11
1.2	Motivação.....	11
1.3	Justificativa.....	12
1.4	Objetivo .....	13
1.4.1	Objetivos Específicos.....	13
1.5	Metodologia .....	13
1.6	Organização do Trabalho .....	14
<b>2</b>	<b>REFERENCIAL TEÓRICO.....</b>	<b>16</b>
2.1	Internet das Coisas .....	16
2.2	Indústria 4.0 .....	17
2.3	RFID.....	18
2.4	Rede de Petri .....	18
2.5	Controlador Lógico Programável .....	19
2.5.1	Linguagem Ladder .....	20
2.6	Microcontroladores .....	21
2.6.1	Raspberry Pi .....	22
2.7	Sistema Supervisório .....	23
2.8	UML.....	23
2.9	Banco de Dados .....	25
2.10	Linguagem Java.....	26
2.11	Página Web .....	27
2.12	Linguagem HTML .....	28
2.13	Servidor OPC.....	28
<b>3</b>	<b>MATERIAIS E MÉTODOS .....</b>	<b>29</b>
3.1	Estações de Sistema Modular de Produção.....	29
3.1.1	Módulo de distribuição .....	30
3.1.2	Módulo de teste .....	32
3.1.3	Módulo de classificação.....	34

3.2	CLP CPX-CEC FESTO .....	35
3.2.1	CoDeSys FESTO .....	36
3.3	Modelagem Rede de Petri .....	37
3.3.1	Compra de produtos .....	37
3.3.2	Estação de Distribuição .....	37
3.3.3	Estação de Teste .....	39
3.3.4	Estação de Classificação .....	40
3.4	RFID.....	41
3.4.1	Tag RFID.....	41
3.4.2	Leitor/Gravador RFID.....	41
3.5	Raspberry Pi .....	42
3.6	Sistema Supervisório .....	42
3.6.1	Requisitos Funcionais .....	43
3.6.2	Modelagem do Banco de Dados.....	44
3.6.3	JavaFX Scene Builder (Layout das Telas).....	45
3.6.4	Netbeans 8.1 (Programação Java e JavaWeb) .....	46
3.7	Adobe DreamWeaver (Layout do Site em HTML) .....	47
3.8	CoDeSys OPC Server (Servidor OPC).....	48
3.9	Rede de Comunicação .....	48
<b>4</b>	<b>RESULTADOS OBTIDOS .....</b>	<b>50</b>
4.1	Página Web .....	50
4.2	Transferência de informações para tag RFID .....	52
4.2.1	Programação em Python para os microcontroladores <i>Raspberry Pi</i> .....	53
4.3	Sistema Supervisório .....	54
4.3.1	Controle de Acesso.....	54
4.3.2	Tela Principal .....	55
4.3.3	Controle de Estoque .....	56
4.3.4	Gerenciamento de Usuários .....	59
4.3.5	Acompanhamento de Produção .....	63
4.3.6	Supervisão do Processo .....	65
<b>5</b>	<b>CONCLUSÕES .....</b>	<b>70</b>
5.1	Trabalhos Futuros.....	70
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>72</b>

<b>APÊNDICE A – MODELAGEM EM REDE DE PETRI .....</b>	<b>76</b>
<b>APÊNDICE B – PROGRAMAÇÃO DO MICROCONTROLADOR RASPBERRY PI77</b>	
<b>B.1      Escrever informação nas Tags <i>RFID</i> .....</b>	<b>77</b>
<b>B.2      Ler informação nas Tags <i>RFID</i> .....</b>	<b>79</b>
<b>APÊNDICE C – PROGRAMAÇÃO PARA COMUNICAR O SISTEMA SUPERVISÓRIO COM O SISTEMA MODULAR DE PRODUÇÃO.....</b>	<b>80</b>

# 1 INTRODUÇÃO

A internet está se tornando cada vez mais presente no dia-a-dia das pessoas, seja nos computadores, nos celulares ou até mesmo nas televisões. Um conceito pouco explorado no Brasil é o da Internet das Coisas, a aptidão de controlar objetos utilizando redes sem fio para o mesmo; em residências, objetos que antes eram de controle manual poderão ter um chip capaz de controlá-lo sem a necessidade de estar no mesmo cômodo; em ambientes industriais, máquinas independentes podem interligar-se com outras utilizando o mesmo processo, basta manter-se conectado à mesma rede e a mesma conectada à internet.

Lima (2015) desenvolveu um Sistema Supervisório para o MPS® (Modular Production System) do laboratório da E.S.T. (Escola Superior de Tecnologia), porém o mesmo não está integrado atualmente. Segundo o autor, sua proposta era a produção de um produto específico conforme sua demanda, cujas informações eram providas de uma página de internet, onde a única intervenção humana no processo é do cliente que realiza a compra da peça. Na simulação da produção, foram implementados leitores e gravadores RFIDs (Radio Frequency IDentification) para que o produto final tenha as informações desejadas pelo cliente.

A proposta de melhoria envolve em otimizar a simulação de produção utilizando diferentes métodos de comunicação entre o *Raspberry Pi*, os leitores/gravadores *RFID*, os CLPs e o Sistema Supervisório, também foram desenvolvidas uma nova estrutura da Página Web e do Sistema Supervisório em Java, tais quais suas comunicações com o banco de dados.

## 1.1 Problema de Pesquisa

Ausência da implantação do Sistema Supervisório para o Sistema Modular de Produção do Laboratório de Engenharia de Controle e Automação da Escola Superior de Tecnologia utilizando o conceito da Internet das Coisas.

## 1.2 Motivação

Segundo Marão (2014), “até 2011 a Internet das Coisas era um conceito inexistente no Brasil, salvo nichos particulares”. Visto que é um conceito novo no

mundo todo, era de se esperar que o mesmo ainda possuísse poucos estudos no Brasil.

Apesar de já existirem esforços iniciais na criação de cidades inteligentes com conceitos de ambientes digitais e tentativas na direção de se implementar a internet do futuro, o Brasil se encontra longe da aplicação generalizada do conceito por diversos fatores, entre eles, a inexistência de uma agenda estratégica de pesquisa e desenvolvimento, pouco financiamento para projetos estruturantes e a falta de condições para estabelecimento de projetos colaborativos interdisciplinares com formação de consórcio entre empresas, academia e usuários finais, em que o conceito de Internet das Coisas se reverta em benefícios para a sociedade em geral (MARÃO, 2014).

A proposta de aplicação do conceito em um sistema que representa a funcionalidade de uma indústria em escala laboratorial tem a finalidade de demonstrar uma das diversas possibilidades de utilização de Internet das Coisas.

### 1.3 Justificativa

O presente projeto de pesquisa se justifica pela otimização do Sistema Modular de Produção implementando diferentes métodos de comunicação em toda a produção além de uma reestruturação da Página Web e do Sistema Supervisório.

A pesquisa se justifica pelo cunho acadêmico, acrescentando o uso do conceito de Internet das Coisas e Aplicativos Móveis para o laboratório de Controle e Automação, possibilitando um auxílio no estudo da próxima geração de discentes, além de utilizar conceitos vistos nas disciplinas de Engenharia de Controle e Automação como Linguagem de Programação I e II, Computação para Automação, Circuitos Elétricos I e II, Controle e Automação I, II e III, Informática Industrial, Instrumentação Industrial, Manufatura Integrada por Computador e Robótica, Tecnologia WEB, Modelagem e Simulação de Sistemas Discretos, Comandos Eletropneumáticos e Hidráulico, Automação e Supervisão de Processos e Tópicos Especiais em Engenharia de Controle e Automação I e II, explorando uma vasta área de conhecimento.

## 1.4 Objetivo

Desenvolver um Sistema Supervisório do MPS® possibilitando o controle e acompanhamento de sua produção utilizando o conceito de Internet das Coisas em conjunto com uma Página Web capaz de simular a compra dos produtos a serem produzidos.

### 1.4.1 Objetivos Específicos

- Implementar o uso do microcontrolador Raspberry Pi em conjunto com o Leitor/Gravador RFID como servo enviando e recebendo dados para o CLP.
- Desenvolver uma Página Web responsável por simular a compra de produtos;
- Desenvolver um Sistema Supervisório em Java capaz de controlar e supervisionar as estações do Sistema Modular de Produção;
- Realizar a comunicação entre o Sistema Supervisório e o Sistema Modular de Produção
- Testar o Sistema e avaliar a funcionalidade e eficiência do mesmo.

## 1.5 Metodologia

O Trabalho apresentado será uma Pesquisa Aplicada, e terá como objetivo a realização de Pesquisa Exploratória sobre o material bibliográfico e de laboratório sobre o assunto. Serão utilizados os procedimentos técnicos de pesquisa bibliográfica e experimental. Serão utilizados os métodos de abordagem hipotético-dedutivo e de procedimento monográfico em sua elaboração.

À priori, pesquisas bibliográficas serão realizadas nas áreas de Internet das Coisas, RFID, Controladores Lógicos Programáveis, Linguagem Ladder e Java, além do funcionamento das bancadas no MPS®, assim como seus componentes e do projeto de pesquisa do autor Renan Araújo de Lima.

Com base nessa pesquisa, serão analisados os requisitos funcionais do MPS® e de seu Sistema Supervisório, assim poderão ser reinterpretadas todas as atividades realizadas anteriormente (LIMA, 2015) a fim de integrar melhorias ao sistema. Será

definido também, os melhores componentes para desenvolver os dois braços robóticos que irão manipular o produto rejeitado e a esteira que irá transportá-los.

Após tais análises e definições, o desenvolvimento e testes do sistema foram realizados de acordo com a sequência a seguir:

- I. Estudo de Uso dos Programas de Simulações (Lima, 2015);
- II. Compra dos materiais necessários de acordo com o projeto de hardware adicional ao MPS®;
- III. Teste da comunicação entre os leitores/gravadores *RFID* e o Sistema Supervisório em *Python* com auxílio do hardware *Raspberry Pi 2*;
- IV. Programação da Página Web em Java Web com auxílio do software *Netbeans 8.1*;
- V. Programação do Sistema Supervisório em Java com auxílio do software *Netbeans 8.1*;
- VI. Teste da comunicação *Ethernet* entre o Sistema Supervisório e o Sistema Modular de Produção utilizando o protocolo *Modbus IP*;
- VII. Integração de todos os componentes nas estações do MPS®;
- VIII. Testar o Sistema e avaliar a funcionalidade e eficiência do mesmo.

## 1.6 Organização do Trabalho

Este trabalho está dividido em seis capítulos para melhorar a compreensão do projeto. Sendo assim, o Capítulo 1 contém uma breve introdução sobre o assunto a ser explanado no trabalho, sua motivação, justificativa para estudar as áreas mencionadas e um resumo da metodologia empregada no desenvolvimento do projeto.

O Capítulo 2 contém as principais áreas de conhecimento e suas teorias para que o leitor tenha pleno entendimento das técnicas e abordagens adotadas no detalhamento de execução do projeto e até estar apto a analisar os resultados de forma objetiva.

O Capítulo 3 detalha os materiais e técnicas empregados na execução do projeto de modo que seja possível para o leitor replicá-los para obter resultados similares aos do projeto implementado.

O Capítulo 4 contém a análise dos resultados obtidos após execução da metodologia e os comenta de maneira detalhada, para melhor entendimento do leitor.

O Capítulo 5 contém as conclusões que podem ser afirmadas após execução do projeto, fazendo análise de qualidade dos resultados obtidos e os confronta com os objetivos propostos pelo projeto.

Por fim, seguem os Apêndices e Anexos para uma compressão mais detalhada das programações e requisitos que são necessários para o entendimento completo de dados informados no corpo do projeto.

## 2 REFERENCIAL TEÓRICO

Para o desenvolvimento do presente projeto, foram necessárias pesquisas bibliográficas relacionadas à Internet das Coisas, Estações de Distribuição, Teste e Classificação do Sistema Modular de Produção, Controlador Lógico Programável, RFID, Sistema Supervisório e Linguagens de Programação Java e Ladder.

### 2.1 Internet das Coisas

A Internet das Coisas, do inglês Internet of Things, é uma “[...] nova infraestrutura que integra a Internet convencional com os objetos permitindo a comunicação entre o mundo virtual e o mundo real” (MARÃO, 2014). Seguindo o pensamento de Vermesan e Friess (2013), pode-se dizer que é um conceito que considera a presença generalizada de coisas/objetos no ambiente, que através de conexões com ou sem fio e um esquema de endereçamento único (IP) é possível a interação e cooperação com outras coisas/objetos para a criação de novas aplicações ou serviços para atingir um objetivo.

Neste contexto, a noção de convergência de rede utilizando o endereçamento único (IP) é fundamental, e depende de uma rede comum que suporte a utilização de vários IPs para a realização de diversas aplicações e serviços.

Segundo Vermesan e Friess (2014), as características fundamentais da Internet das Coisas são as seguintes:

- Interconectividade: em relação ao I.o.T., qualquer coisa pode ser interconectada à infraestrutura global de informação e comunicação (Internet);
- Serviços relacionados a coisas: a Internet das Coisas é capaz de proporcionar serviços dentro dos limites impostos pelas coisas, assim como, proteção de privacidade e consistência semântica entre a parte físico e a associação virtual das coisas. Para que os serviços sejam fornecidos de maneira correta, as tecnologias tanto no mundo físico quanto no virtual devem mudar conjuntamente;
- Heterogeneidade: os dispositivos dentro da I.o.T. são heterogêneos, ou seja, são baseados em diferentes plataformas de hardware e rede. Eles podem interagir com outros dispositivos ou serviços por meio de diferentes redes;

- Mudanças dinâmicas: o estado dos dispositivos muda dinamicamente, que pode ser tanto o status, como ligado e desligado, quanto no número de dispositivos, que pode aumentar ou diminuir com o passar do tempo.

A Internet das Coisas proporciona soluções baseadas na integração entre tecnologia de informação, que se refere ao hardware e software utilizados para armazenar e processar dados, e a tecnologia de comunicação, que inclui sistemas eletrônicos utilizados para comunicação entre dispositivos. A rápida evolução das tecnologias de informação e comunicação converge para três grandes áreas de inovação tecnológica: nuvem, redes de comunicação e dispositivos (WORLD ECONOMIC FORUM, 2012).

## 2.2 Indústria 4.0

De acordo com Silva, Filho e Miyagi (2015), a quarta revolução industrial, chamada de indústria 4.0, é considerada o futuro paradigma da produção. Novas tecnologias devem ser empregadas para integrar máquinas e humanos em cadeias de valor compondo uma rede de entidades (plantas industriais) localizadas em posições geograficamente distribuídas (dispersas), e que devem fornecer serviços e produtos de forma autônoma.

As três revoluções industriais passadas foram desencadeadas por inovações tecnológicas: a invenção das máquinas a vapor, a divisão de trabalho e a introdução dos CLPs na automação de manufatura. De acordo com especialistas em indústria e pesquisa, a futura quarta revolução industrial será desencadeada pela Internet, que permite a comunicação entre humanos e máquinas em extensas redes de comunicação (BRETTEL et al., 2014).

A indústria 4.0 também envolve o uso de dispositivos do tipo “smart” no chão de fábrica. Isto permite mais interação entre clientes e fornecedores, e aumenta ainda mais a personalização dos produtos para atender as demandas individuais (SCHMITT; ORFGEN; ZÜHLKE, 2015).

Segundo Brettel et al. (2014), a Internet tem sido identificada como um poderoso instrumento para gerenciar sistemas e tecnologias distribuídos, utilizando tecnologias como o RFID (Radio Frequency Identification) para realizar a localização individual de produtos através de uma linha de processo.

O principal foco da indústria 4.0 é realizar a imersão de produtos inteligentes nos processos de produção (VYATKIN et al., 2007). Com a aquisição de dados, os sistemas poderão realizar tomadas de decisão baseadas nas informações recebidas e exibir o que acontece no processo produtivo para o usuário por meio de Interfaces Humano Máquina.

## 2.3 RFID

RFID (Radio Frequency Identification), ou Identificação por Rádio Frequência em português, é uma tecnologia utilizada para identificar, rastrear e gerenciar desde produtos e documentos até mesmo indivíduos, sem contato e sem a necessidade de um campo visual. A possibilidade de ser aplicada a inúmeras situações tornou a tecnologia RFID objeto de diversos projetos-piloto em diferentes lugares no mundo (RFID CENTER OF EXCELLENCE, 2013).

Segundo Sarma, Weis e Engels (2003), todo sistema RFID deve possuir três componentes principais:

- Tag RFID, ou transponder: se localiza no objeto a ser identificado, e é o dispositivo que carrega os dados do sistema RFID;
- Leitor/Gravador RFID, ou transciever: possui a capacidade de ler e escrever dados em uma tag RFID;
- Subsistema de processamento de dados: utiliza os dados obtidos do leitor/gravador RFID para alguma aplicação.

## 2.4 Rede de Petri

A rede de Petri é uma ferramenta gráfica e matemática que se adapta bem a um grande número de aplicações em que as noções de eventos e de evoluções simultâneas são importantes.

Segundo Belan (2007), a definição de uma rede de petri pode ser enumerada em três elementos básicos, que são os seguintes:

Lugar: é representado por um círculo e pode ser interpretado como uma condição, um estado parcial, uma espera, um procedimento, um conjunto de recursos, um estoque, uma posição geográfica num sistema de transporte, etc. De modo geral,

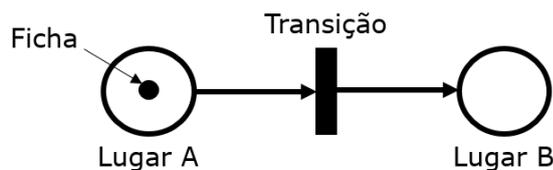
os lugares representam o estado do sistema e são considerados como elementos passivos;

Transição: representada por uma barra ou retângulo é associada a um evento que ocorre no sistema, como por exemplo, o acionamento de um botão, a detecção de que a temperatura foi alcançada e a detecção de presença de peças no magazine. Em geral representam elementos ativos;

Ficha: representada por um ponto num lugar é um indicador de que a condição associada ao lugar é verificada. A Ficha pode representar um objeto (recurso ou peça) numa certa posição geográfica (num determinado estado), ou ainda uma estrutura de dados que se manipula. Por exemplo, se em lugar com o significado de “peças em espera” houvesse três fichas, indicaria que existem três peças em espera.

Na figura 1 estão representados na forma gráfica os três elementos básicos da Rede de Petri:

Figura 1 – Elementos Básicos da Rede de Petri



Fonte: (LIMA, 2015)

## 2.5 Controlador Lógico Programável

Os Controladores Lógico Programáveis ou CLPs são equipamentos eletrônico-digitais programáveis para aplicações industriais. Os CLPs foram construídos a partir da necessidade de uma tecnologia capaz de substituir os painéis de relés, pela pouca flexibilidade que existia na utilização dos mesmos.

Nascia assim um equipamento bastante versátil e de fácil utilização, que vem se aprimorando constantemente, diversificando cada vez mais os setores industriais e suas aplicações, o que, atualmente, justifica um mercado mundial estimado em 4 bilhões de dólares anuais (FILHO, 1998).

Segundo Filho (1998), “os sinais de entrada e saída dos CLPs podem ser digitais ou analógicos. Existem diversos tipos de módulos de entrada e saída que se adequam as necessidades do sistema a ser controlado”.

Os sinais dos sensores são aplicados às entradas do controlador e a cada ciclo (varredura) todos esses sinais são lidos e transferidos para a unidade de memória

interna denominada memória imagem de entrada. Estes sinais são associados entre si e aos sinais internos. Ao término do ciclo de varredura, os resultados são transferidos à memória imagem de saída e então aplicados aos terminais de saída.

Para uma aplicação que procura representar a funcionalidade de uma indústria de grande porte, é necessária a escolha de um equipamento robusto e de grande utilização nas fábricas. Por esse motivo o CLP foi escolhido para a pesquisa, sendo utilizado um para cada estação.

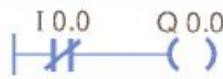
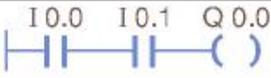
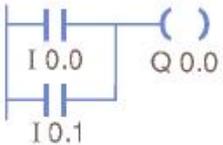
### 2.5.1 Linguagem Ladder

De acordo com o que diz Filho (1998), a lógica matemática ou simbólica visa superar as dificuldades e ambiguidades de qualquer língua, devido a sua natureza vaga e equívoca das palavras usadas e do estilo confuso que poderia atrapalhar o rigor lógico do raciocínio. Para evitar essas dificuldades, criou-se uma linguagem lógica artificial.

A lógica binária possui apenas dois valores que são representados por: 0 e 1. A partir desses dois símbolos constrói-se então uma base numérica binária. Por meio desses conceitos foram criadas as portas lógicas, que são circuitos utilizados para combinar níveis lógicos digitais de formas específicas. As portas lógicas básicas são: AND, OR e NOT.

Tabela 1 – Representação das Portas lógicas básicas em Simbologia, Expressão e na Linguagem Ladder

**T2** Funções lógicas básicas

Portas Lógicas	Símbolo	Expressão	Ladder
NOT	A  S	$S = \bar{A}$	
AND	A  B  S	$S = A \cdot B$	
OR	A  B  S	$S = A + B$	

Fonte: BRANDÃO, 2007

A linguagem Ladder permite que se desenvolvam lógicas combinacionais, sequenciais ou ambas, utilizando como operadores para estas lógicas: entradas,

saídas, estados auxiliares e registros numéricos. Os 3 principais símbolos de programação são os seguintes:

Tabela 2 – Principais símbolos da programação Ladder

**T1** Principais símbolos da programação Ladder

<i>Tipo</i>	<i>Símbolo</i>	<i>Equipamento Elétrico</i>
<i>Contato aberto</i>		
<i>Contato fechado</i>		
<i>Saída</i>		

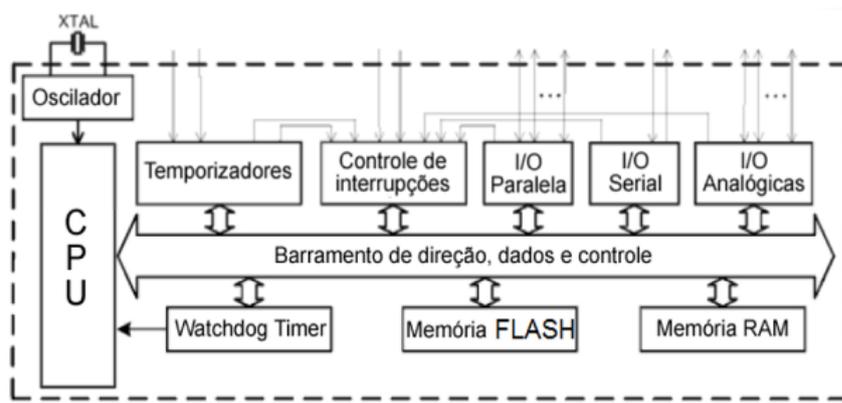
Fonte: BRANDÃO, 2007

## 2.6 Microcontroladores

De acordo com Foresti (2013), os microcontroladores são compostos de microprocessadores que podem ser programados para funções específicas. Em geral, eles são usados para controlar circuitos e, por isso, são comumente encontrados dentro de outros dispositivos, sendo conhecidos como "controladores embutidos". A estrutura interna de um microcontrolador apresenta um processador, bem como circuitos de memória e periféricos de entrada e saída.

Os microcontroladores se diferenciam dos processadores, pois além dos componentes lógicos e aritméticos usuais de um microprocessador de uso geral, o microcontrolador integra elementos adicionais em sua estrutura interna, como memória de leitura e escrita para armazenamento de dados, memória somente de leitura para armazenamento de programas, EEPROM para armazenamento permanente de dados, dispositivos periféricos como conversores analógico/digitais (ADC), conversores digitais/analógicos (DAC), Portas de entrada e saída digitais (I/O) para propósito geral.

Figura 2 – Diagrama de blocos geral de um Microcontrolador



Fonte: (BASTOS, 2010)

Para uma melhor ideia de quais são e como se comunicam os componentes de um microcontrolador, a figura 2 apresenta um diagrama de blocos geral de um microcontrolador.

### 2.6.1 Raspberry Pi

Segundo Raspberry Pi Foundation (2013), o microcontrolador Raspberry Pi é um pequeno computador do tamanho de um cartão de crédito, composto apenas de uma única placa e que foi desenvolvido no Reino Unido pela Raspberry Pi Foundation. Este microcontrolador foi criado com a intenção de estimular o ensinamento do básico de ciência da computação nas escolas.

Os componentes do computador foram selecionados de modo a otimizar o custo. Trata-se de um instrumento bastante simples, porém capaz de realizar várias tarefas específicas. De acordo com as informações de Paiva e Moreira (2014), o Raspberry Pi não possui disco rígido e o sistema operacional, assim como todos os arquivos, são armazenados em um cartão SD. É dotado de uma saída de vídeo de alta definição com suporte para monitor de até 2,2 megapixels (HDMI, 1920 × 1200 pixels) e de uma segunda saída de vídeo RCA. Possui duas ou quatro entradas USB, uma saída de áudio e uma saída para conectores RJ45

(Ethernet) para acesso à internet. Existem oito pinos programáveis de GPIO (General Purpose Input/Output), que podem ser, por exemplo, ligados a sensores, motores ou relés (interruptores controlados por eletricidade) para controle de outros equipamentos.

## 2.7 Sistema Supervisório

Unindo os pensamentos de Mott (2012) e Kondrasovas (2013), pode-se dizer que o sistema supervisório destina-se a capturar e armazenar em um banco de dados, informações sobre um processo de produção. Possibilitando através do acesso a esses dados, a representação de objetos estáticos e animados de forma amigável, que, juntos, representam todo o processo de uma planta em uma IHM (Interface Humano-Máquina).

Segundo Paiola (2012), “com a evolução da tecnologia, foram incorporadas funções nunca antes imaginadas para um sistema supervisório. Sempre baseadas nas necessidades e requisições dos usuários finais, são implementadas ferramentas para as mais diversas finalidades”. E utilizando-se deste pensamento, o sistema supervisório que será desenvolvido nesta pesquisa procura trazer inovações através da implementação de novas funcionalidades por meio da aplicação do conceito de Internet das coisas.

Logo, com o desenvolvimento desta ferramenta em Java, o sistema será esquematizado da seguinte maneira:

Figura 3 – Diagrama de um sistema automatizado com sistema supervisório



Fonte: (LIMA, 2015).

## 2.8 UML

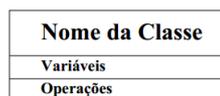
A apresentação gráfica de um programa orientado a objetos é um artifício muito utilizado para facilitar a visualização das entidades e suas relações. Dentre as

diversas linguagens gráficas disponíveis, a mais sistematicamente elaborada, sendo, também, a mais aceita, é a Unified Modelling Language (UML).

A UML proporciona uma forma padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de banco de dados e componentes de software reutilizáveis (BOOCH; RUMBAUGH; JACOBSON, 2006).

A figura 4 mostra um diagrama de classe UML. O diagrama é dividido em três campos. O campo superior contém o nome da classe; no campo abaixo se declaram as variáveis daquela classe, enquanto que no último campo se declaram os operadores (métodos) dessa classe.

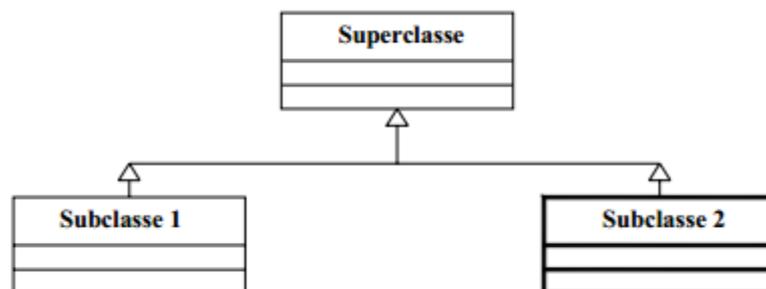
Figura 4 – Diagrama de classe em UML



Fonte: (ALMEIDA, 2005)

A figura 5 mostra um diagrama de herança, no qual pode-se visualizar duas subclasses derivando da superclasse.

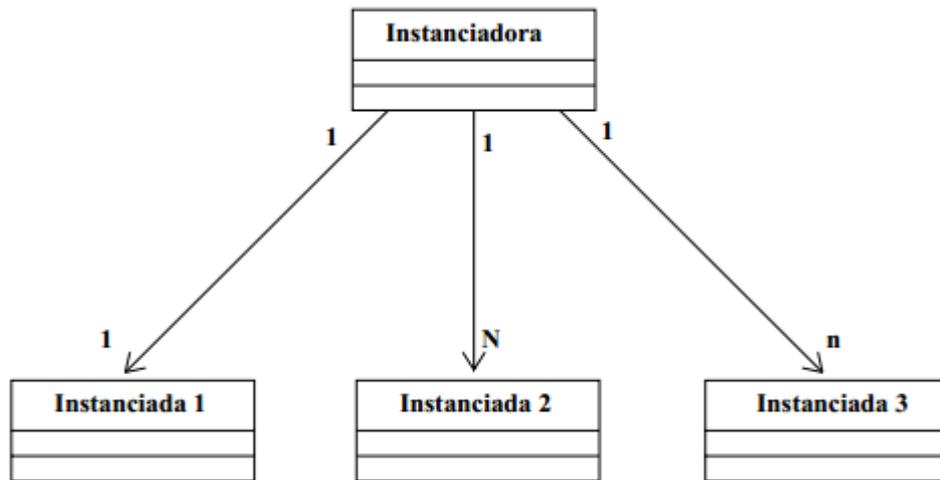
Figura 5 – Diagrama de herança em UML



Fonte: (ALMEIDA, 2005)

A figura 6 mostra um diagrama de instâncias de uma dada interface. Ao lado das linhas anota-se o número de relações entre as classes.

Figura 6 – Diagrama de instâncias em UML



Fonte: (ALMEIDA, 2005)

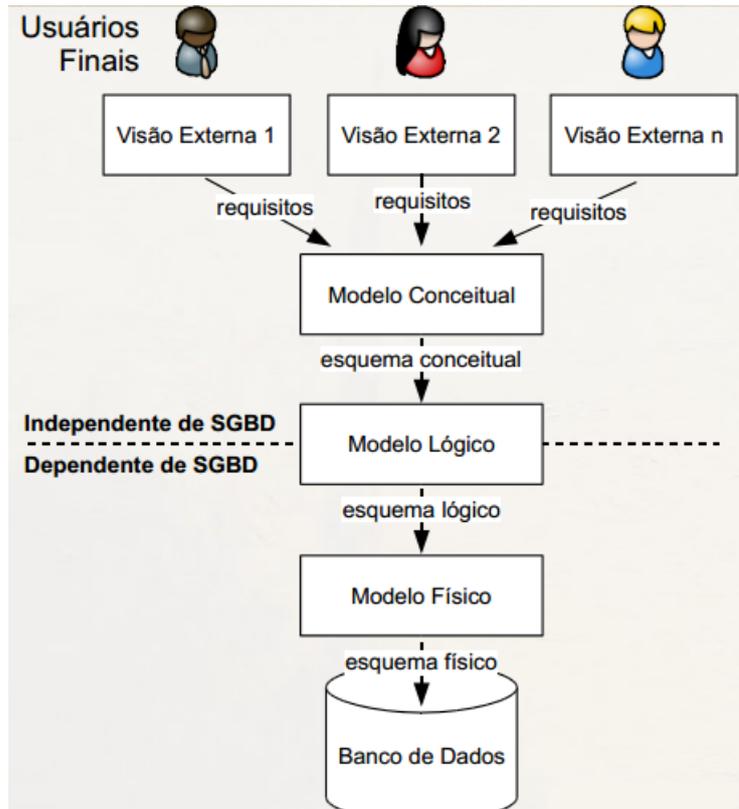
## 2.9 Banco de Dados

De acordo com Ramakrishnan e Gehrke (2008), “um banco de dados é uma coleção de dados que, tipicamente, descreve as atividades de uma ou mais organizações relacionadas”. Um exemplo poderia ser o banco de dados de uma universidade, que poderia conter informações sobre entidades, como alunos, professores, cursos e turmas; e relacionamento entre entidades, como a matrícula dos alunos nos cursos ministrados pelos professores e o uso de salas por curso.

É necessário realizar o gerenciamento dos dados salvos no banco, para esse fim, existem os Sistemas de Gerenciamento de Banco de Dados, ou SGBD, que são “softwares projetados para auxiliar a manutenção e utilização de vastos conjuntos de dados” (RAMAKRISHNAN; GEHRKE, 2008).

Para realizar uma modelagem de qualidade é necessário seguir uma sequência de passos importantes, que são apresentados na figura 7:

Figura 7 – Passos para o projeto de um banco de dados



Fonte: (SANTANCHÉ; JR, 2014)

Como visto na figura 7, deve-se primeiramente realizar o levantamento dos requisitos do banco com os usuários finais do sistema, em seguida, deve-se fazer um modelo conceitual que independe do SGBD utilizado no projeto. A partir do modelo conceitual pode-se então desenvolver um modelo lógico de acordo com a linguagem do sistema de gerenciamento de banco de dados utilizado. Este modelo lógico pode gerar então, o modelo e esquema físicos do banco de dados, o qual já pode ser utilizado.

## 2.10 Linguagem Java

Para o desenvolvimento do sistema supervisorio foi escolhido a linguagem Java, pois, diferente de outras linguagens de programação, sua execução não está diretamente relacionada com o Sistema Operacional, pois o código é executado em uma máquina virtual chamada de JVM (Java Virtual Machine), possibilitando assim a portabilidade do software (ROMANATO, 2015). Além disso, a linguagem Java é orientada a objetos, o que torna mais simples e rápida a realização de alterações no sistema (CAELUM, 2003).

Segundo Almeida (2005), dentre as linguagens que suportam o paradigma de programação orientada a objetos, as mais utilizadas são C++ e Java. Alguns estudos indicam que Java é mais apropriada devido aos seguintes aspectos: independência de sistema operacional; performance numérica; capacidade de reutilização do software e suporte à persistência dos dados.

Programadores Java concentram-se na elaboração de novas classes e reutilização de classes existentes. Existem muitas bibliotecas de classe e outras estão sendo desenvolvidas em todo o mundo. O software é, então, construído a partir de componentes amplamente disponíveis, portáteis, bem-documentados, cuidadosamente testados e bem definidos. Esse tipo de capacidade de reutilização de software acelera o desenvolvimento de programas poderosos e de alta qualidade (DEITEL; DEITEL, 2003).

## 2.11 Página Web

De acordo com Souto et al. (2001), conhece-se por página Web qualquer documento que faça parte de um sítio web e que costuma conter ligações (igualmente chamadas hiperligações ou links) para facilitar a navegação entre os conteúdos. As páginas Web são desenvolvidas com linguagens de marcação como o HTML, as quais podem ser interpretadas pelos navegadores. Desta forma, as páginas podem apresentar informação em diferentes formatos (texto, imagens, sons, vídeos, animações), estar associadas a dados de estilo ou conter aplicações interativas.

Segundo Furquim (2004), “de acordo com o pensamento de marketing, o conhecimento das necessidades, percepções e satisfação dos consumidores permitirá a melhor estruturação da oferta”. Por meio desta afirmação fica claro o porquê de a comunicação de sistemas com páginas Web estar presente no conceito de Internet das Coisas. Neste projeto será desenvolvida uma página Web para simulação das compras dos produtos de um processo de manufatura. Esta será implantada em um servidor local, com o objetivo de redução de custos, porém, ela pode ser implantada futuramente em um servidor Web e ser acessada remotamente.

## 2.12 Linguagem HTML

HTML é uma sigla que representa HyperText Markup Language, que traduzido para o português significa Linguagem de Marcação de Hipertexto. Segundo Graham e Sullivan (1998), consiste em uma linguagem de marcação utilizada para produção de páginas na web, que permite a criação de documentos que podem ser lidos em praticamente qualquer tipo de computador e transmitidos pela internet.

De acordo com Silva (2008), para escrever documentos HTML é apenas um editor de texto simples e conhecimento dos códigos que compõem a linguagem. Os códigos (conhecidos como tags) servem para indicar a função de cada elemento da página Web. As tags funcionam como comandos de formatação de textos, formulários, links (ligações), imagens, tabelas, entre outros. Os navegadores identificam as tags e apresentam a página conforme está especificada.

## 2.13 Servidor OPC

Em 1995, algumas empresas se reuniram com o objetivo de desenvolver um padrão baseado na tecnologia OLE/DCOM para acesso à dados de tempo real dentro do sistema operacional Windows. Este grupo sem fins lucrativos é formado por diversas empresas e é gerenciado pela organização OPC Foundation (FONSECA, 2002).

Basicamente, o padrão OPC estabelece as regras para que sejam desenvolvidos sistemas com interfaces padrões para comunicação dos dispositivos de campo (CLPs, sensores, balanças, etc.) com sistemas de monitoração.

### 3 MATERIAIS E MÉTODOS

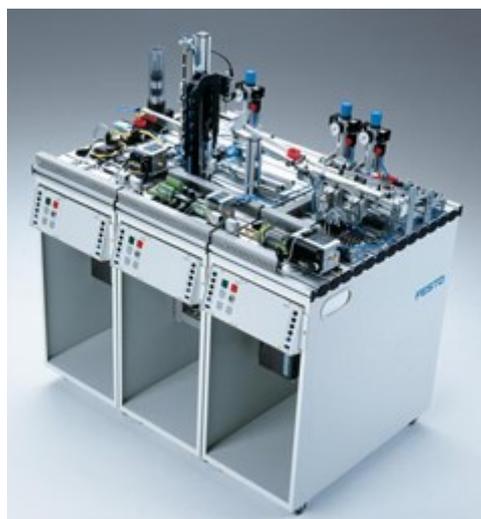
Neste Capítulo serão abordados as especificações e características dos materiais escolhidos para desenvolvimento da metodologia, assim como quais os métodos utilizados para obtenção dos resultados.

#### 3.1 Estações de Sistema Modular de Produção

Uma linha de produção de uma fábrica pode ser composta de células individuais, onde cada unidade possui uma função específica no processo (distribuição, teste, processamento, manipulação, montagem, armazenamento, classificação). Através das Estações de Sistema Modular de Produção é possível gerar combinações para atingir aos requisitos de múltiplas aplicações, as quais podem representar funcionalidades de um processo real em grande escala (FESTO DIDADIC, 2005).

A figura 8 ilustra como se interligam as três estações de sistema modular de produção que se encontram no laboratório de automação da Escola Superior de Tecnologia da UEA, que são os módulos de Distribuição, Teste e Classificação.

Figura 8 – Estações de sistema modular de produção



Fonte: (FESTO DIDADIC, 2005)

Para um melhor entendimento sobre as estações de sistema modular de produção utilizadas neste projeto, é apresentado em seguida o funcionamento detalhado de cada uma.

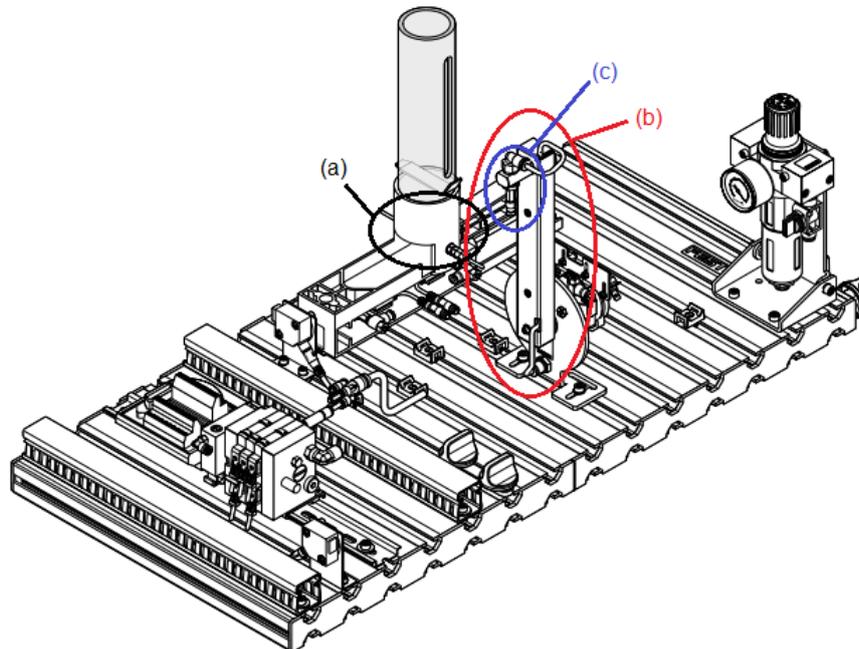
### 3.1.1 Módulo de distribuição

A estação de distribuição é um dispositivo de alimentação. Dispositivos de alimentação são unidades que cumprem o papel de abastecimento, ordenação e fornecimento de componentes. Por meio dessa funcionalidade, estes dispositivos podem facilitar a separação de dispositivos de acordo com suas características como forma, peso, etc. A função da estação de distribuição é selecionar as peças que estão no compartimento e transferi-las por meio de um dispositivo com movimento rotacional utilizando uma ventosa (FESTO DIDADIC, 2006a).

A figura 9 ilustra o modelo 3D da estação de distribuição, que esclarece a localização dos atuadores que se encontram na seguinte ordem:

- (a) Empurrador;
- (b) Transportador;
- (c) Ventosa;

Figura 9 – Estação de sistema modular de produção - Distribuição



Fonte: (FESTO DIDADIC, 2006a)

O módulo utiliza um CLP que realiza a leitura dos sensores e a atuação dos atuadores do sistema, as entradas e saídas estão configuradas conforme a tabela 3.

Tabela 3 – Entradas e Saídas do CLP da Estação de Distribuição

BIT	Cartão	Input	Descrição	Endereço	Output	Descrição	Endereço
0	PAINEL DE CONTROLE	DI0	Botão START	%IX2.0	DO0	LED START	%QX0.0
		DI1	Botão STOP	%IX2.1	DO1	LED RESET	%QX0.1
		DI2	Botão Manual/Autom.	%IX2.2	DO2	Auxiliar Q1	%QX0.2
		DI3	Botão RESET	%IX2.3	DO3	Auxiliar Q2	%QX0.3
		DI4	Auxiliar Detection_T	%IX2.4	DO4	Auxiliar Q4	%QX0.4
		DI5	Auxiliar I5	%IX2.5	DO5	Auxiliar Q5	%QX0.5
		DI6	Auxiliar I6	%IX2.6	DO6	Auxiliar Q6	%QX0.6
		DI7	Auxiliar I7	%IX2.7	DO7	Auxiliar Q7	%QX0.7
1	SENSORES E ATUADORES DO MÓDULO	DI0	-	%IX4.0	DO0	OUT - Estander Empurrador	%QX2.0
		DI1	IN - Empurrador Retraído	%IX4.1	DO1	OUT - Acionar Vácuo (Ventosa)	%QX2.1
		DI2	IN - Empurrador Estendido	%IX4.2	DO2	OUT - Acionar Ar (Ventosa)	%QX2.2
		DI3	IN - Ventosa Ligada	%IX4.3	DO3	OUT - Transportador Pos. Inicial	%QX2.3
		DI4	IN - Transportador Pos. Inicial	%IX4.4	DO4	OUT - Transportador Pos. Final	%QX2.4
		DI5	IN - Transportador Pos. Final	%IX4.5	DO5	-	%QX2.5
		DI6	IN - Detector de Peças	%IX4.6	DO6	-	%QX2.6
		DI7	IN - Sensor de Barreira - IP-FI	%IX4.7	DO7	OUT - Sensor de Barreira - IP-FI	%QX2.7

Fonte: Autor

O intuito desta pesquisa é deixar o sistema mais autônomo, portanto, ao ser realizada uma compra na página Web, o sistema supervisor envia um sinal via ethernet que aciona a memória que inicia a produção.

O início do funcionamento da estação de distribuição se dá pelo acionamento do empurrador pneumático, que possui dois sensores de final de curso, os quais são utilizados para saber se o mesmo está retraído ou estendido. O empurrador é acionado se o sensor de peças no magazine está ativado, ou seja, se existe peça no magazine. Ao ser acionado, o empurrador impulsiona uma peça para fora do magazine, que então para na posição de coleta.

Quando a peça chega na posição de coleta, o leitor/gravador RFID grava a informação de qual cor a peça deve ter e para onde ela deve ir. O próximo passo é o acionamento do transportador pneumático, o qual possui dois sensores de fim de curso, um acionado quando o mesmo está na posição inicial junta à estação seguinte, e o outro quando está na posição final, no ponto de coleta de peças.

O transportador pneumático possui uma ventosa que possui dois estados: sucção e expiração de ar. Quando o transportador chega na posição final, a ventosa entra no estado de sucção, e permanece nesse estado até o transportador voltar à posição inicial, deixando a peça na estação seguinte.

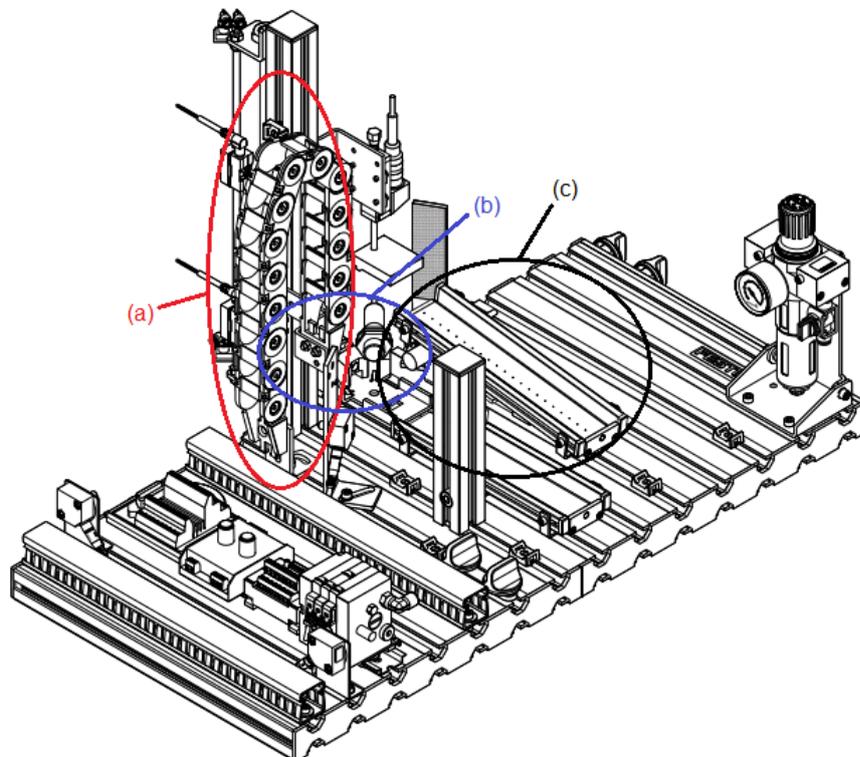
### 3.1.2 Módulo de teste

Realizar testes, assim como medir, é parte integrante da verificação de produtos. Aquisição da informação do teste em comparação com características específicas resultam na decisão da rejeição ou aprovação de certo produto. Um componente importante da medição é a comparação de características reais com a desejada. A função da Estação de Teste é determinar os atributos do material de uma peça, checar a altura da mesma, e então, rejeitá-la ou repassá-la para a próxima estação (FESTO DIDADIC, 2006c).

A figura 10 ilustra o modelo 3D da estação de teste, que esclarece a localização dos atuadores que se encontram na seguinte ordem:

- (a) Elevador;
- (b) Empurrador;
- (c) Air Conveyor;

Figura 10 – Estação de sistema modular de produção - Teste



Fonte: (FESTO DIDADIC, 2006c)

Como na estação de distribuição, o módulo também utiliza um CLP para a leitura dos sensores e a atuação dos atuadores do sistema, as entradas e saídas estão configuradas conforme a tabela 4.

Tabela 4 – Entradas e Saídas do CLP da Estação de Teste

BIT	Cartão	Input	Descrição	Endereço	Output	Descrição	Endereço
0	PAINEL DE CONTROLE	DI0	Botão START	%IX4.0	DO0	LED START	%QX2.0
		DI1	Botão STOP	%IX4.1	DO1	LED RESET	%QX2.1
		DI2	Botão Manual/Autom.	%IX4.2	DO2	Auxiliar Q1	%QX2.2
		DI3	Botão RESET	%IX4.3	DO3	Auxiliar Q2	%QX2.3
		DI4	Auxiliar Detection_T	%IX4.4	DO4	Auxiliar Detection_T	%QX2.4
		DI5	Auxiliar I5	%IX4.5	DO5	Auxiliar Q5	%QX2.5
		DI6	Auxiliar I6	%IX4.6	DO6	Auxiliar Q6	%QX2.6
		DI7	Auxiliar I7	%IX4.7	DO7	Auxiliar Q7	%QX2.7
1	SENSORES E ATUADORES DO MÓDULO	DI0	IN - S. Capacitivo	%IX2.0	DO0	OUT - Descer Elevador	%QX0.0
		DI1	IN - S. Óptico	%IX2.1	DO1	OUT - Subir Elevador	%QX0.1
		DI2	IN - S. Reflexivo	%IX2.2	DO2	OUT - Estender Empurrador	%QX0.2
		DI3	IN - S. Analógico	%IX2.3	DO3	OUT - Acionar Air Conveyor	%QX0.3
		DI4	IN - S. Elevador UP	%IX2.4	DO4	-	%QX0.4
		DI5	IN - S. Elevador DOWN	%IX2.5	DO5	-	%QX0.5
		DI6	IN - Empurrador Retraído	%IX2.6	DO6	-	%QX0.6
		DI7	IN - Sensor de Barreira - IP-FI	%IX2.7	DO7	OUT - Sensor de Barreira - IP-FI	%QX0.7

Fonte: Autor

Ao término das ações da estação de distribuição, é enviado um sinal para a estação de teste, que representa que uma peça foi deixada no início do módulo, que é composto por um dispositivo que possui duas funções: elevador, com sensores que indicam os dois estados possíveis, topo ou base; e empurrador, a qual também possui sensores de fim de curso.

O início do funcionamento da estação de teste se dá pela leitura do chip RFID realizada pelo leitor/gravador RFID da estação, o qual verifica qual cor deveria ter a peça. Ocorre então leitura dos sensores óptico e capacitivo, e por meio da união das duas informações, pode-se definir qual é a cor real da peça (a lógica desta definição se encontra na tabela 5); se esta for a mesma da leitura do chip, o elevador é acionado e em seguida o empurrador, o que faz com que a peça siga para o Air Conveyor, terminando então na próxima estação. Porém se as cores forem diferentes, somente o empurrador é acionado, que impulsiona a peça para os rejeitos da bancada de teste.

Tabela 5 – Lógica para definição de cor real da peça

Sensor Capacitivo	Sensor Óptico	Cor Real da Peça
0	0	Preta
0	1	Vermelha
1	1	Metálica

Fonte: Autor

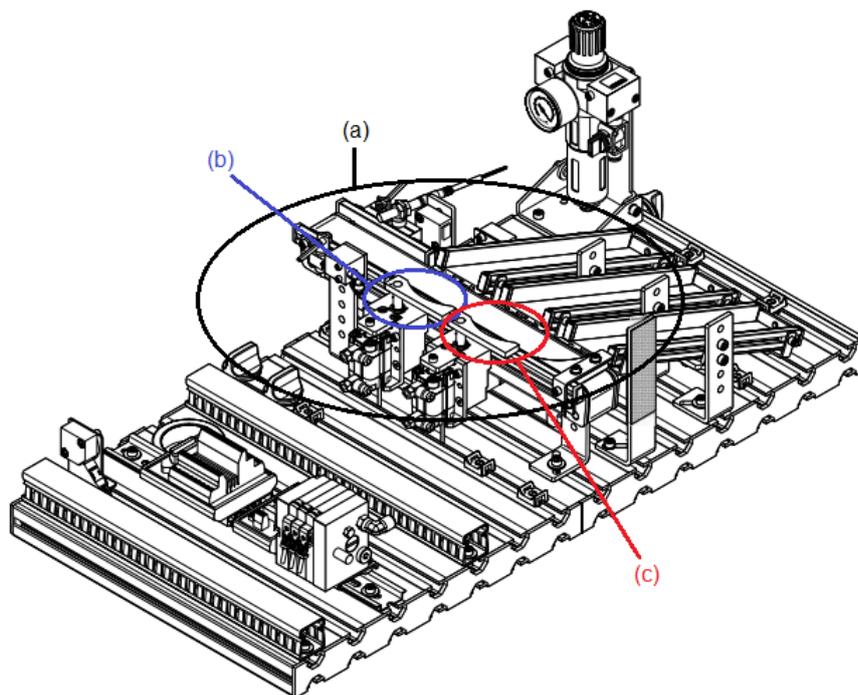
### 3.1.3 Módulo de classificação

Classificação faz parte da função de mudança de quantidades. A seção do transportador pode ser ramificada para a divisão, onde os ramos desta área são acionados de acordo com a peça, e devem proceder individualmente sem comprometer a funcionalidade dos mesmos. A função da estação de classificação é dividir as peças em três escorregadores (FESTO DIDADIC, 2006b).

A figura 11 ilustra o modelo 3D da estação de classificação, que esclarece a localização dos atuadores que se encontram na seguinte ordem:

- (a) Esteira;
- (b) Desvio 1;
- (c) Desvio 2;

Figura 11 – Estação de sistema modular de produção - Classificação



Fonte: (FESTO DIDADIC, 2006b)

Como nas estações anteriores, o módulo também utiliza um CLP para a leitura dos sensores e a atuação dos atuadores do sistema, as entradas e saídas estão configuradas conforme a tabela 6.

Tabela 6 – Entradas e Saídas do CLP da Estação de Classificação

BIT	Cartão	Input	Descrição	Endereço	Output	Descrição	Endereço
0	PAINEL DE CONTROLE	DI0	Botão START	%IX4.0	DO0	LED START	%QX2.0
		DI1	Botão STOP	%IX4.1	DO1	LED RESET	%QX2.1
		DI2	Botão Manual/Autom.	%IX4.2	DO2	Auxiliar Q1	%QX2.2
		DI3	Botão RESET	%IX4.3	DO3	Auxiliar Q2	%QX2.3
		DI4	Auxiliar I4	%IX4.4	DO4	Auxiliar Detection_T	%QX2.4
		DI5	Auxiliar I5	%IX4.5	DO5	Auxiliar Q5	%QX2.5
		DI6	Auxiliar I6	%IX4.6	DO6	Auxiliar Q6	%QX2.6
		DI7	Auxiliar I7	%IX4.7	DO7	Auxiliar Q7	%QX2.7
1	SENSORES E ATUADORES DO MÓDULO	DI0	IN - S. Óptico Presença	%IX2.0	DO0	OUT - Acionar Esteira	%QX0.0
		DI1	IN - S. Indutivo Material	%IX2.1	DO1	OUT - Acionar Desvio 1	%QX0.1
		DI2	IN - S. Óptico Cor	%IX2.2	DO2	OUT - Acionar Desvio 2	%QX0.2
		DI3	IN - S. Final de Curso	%IX2.3	DO3	OUT - Desacionar Trava	%QX0.3
		DI4	IN - S. Desvio 1 Recuado	%IX2.4	DO4	-	%QX0.4
		DI5	IN - S. Desvio 1 Acionado	%IX2.5	DO5	-	%QX0.5
		DI6	IN - Desvio 2 Recuado	%IX2.6	DO6	-	%QX0.6
		DI7	IN - Desvio 2 Acionado	%IX2.7	DO7	OUT - Sensor de Barreira - IP-FI	%QX0.7

Fonte: Autor

Os sensores de detecção de cores (óptico e indutivo) não serão utilizados para esta classificação. Ao chegar no início da estação de classificação, o sensor óptico de presença é acionado, que segue então para a leitura do chip RFID feita pelo leitor/gravador RFID da estação, que verifica para qual guia a peça deve ir.

A partir da informação do chip, os desvios são acionados para que as peças caiam em suas respectivas guias. O acionamento dos desvios pode ser expressado pela tabela 7.

Tabela 7 – Acionamento dos desvios da Estação de Classificação

Guia	Desvio 1	Desvio 2
1	1	0
2	0	1
3	0	0

Fonte: Autor

### 3.2 CLP CPX-CEC FESTO

Os Controladores Lógico Programáveis (CLPs) da família CPX-CEC da FESTO permitem ser programados nas linguagens de programação de acordo com o padrão IEC 61131-3. É possível realizar a comunicação com o CLP por meio de conexão Ethernet com os protocolos Modbus/TCP, EasyIP ou TCP/IP. Além disso, é possível

a comunicação com um servidor OPC que permite a criação de diferentes Interfaces Humano Máquina

(FESTO, 2015). O uso deste CLP é a melhor opção para este projeto pois todos os componentes

das estações de sistema modular de produção são do mesmo fabricante, o que facilita a comunicação entre eles e diminui o trabalho de realizar adaptações nos dispositivos. O mesmo também possui diversas formas de comunicação, que são vitais para o correto funcionamento do sistema supervisorio.

As três estações de sistema modular de produção do laboratório de automação da EST-UEA utilizam o mesmo CLP, o qual possui expansões tanto de entrada quanto saída, totalizando em quarenta e oito (48) entradas e quarenta e oito (48) saídas disponíveis, que são mais que o suficiente para a implantação do projeto.

A figura 11 mostra o CLP CPX-CEC com as expansões de entrada e de saída.

Figura 12 – CLP CPX-CEC da FESTO



Fonte: (FESTO, 2015)

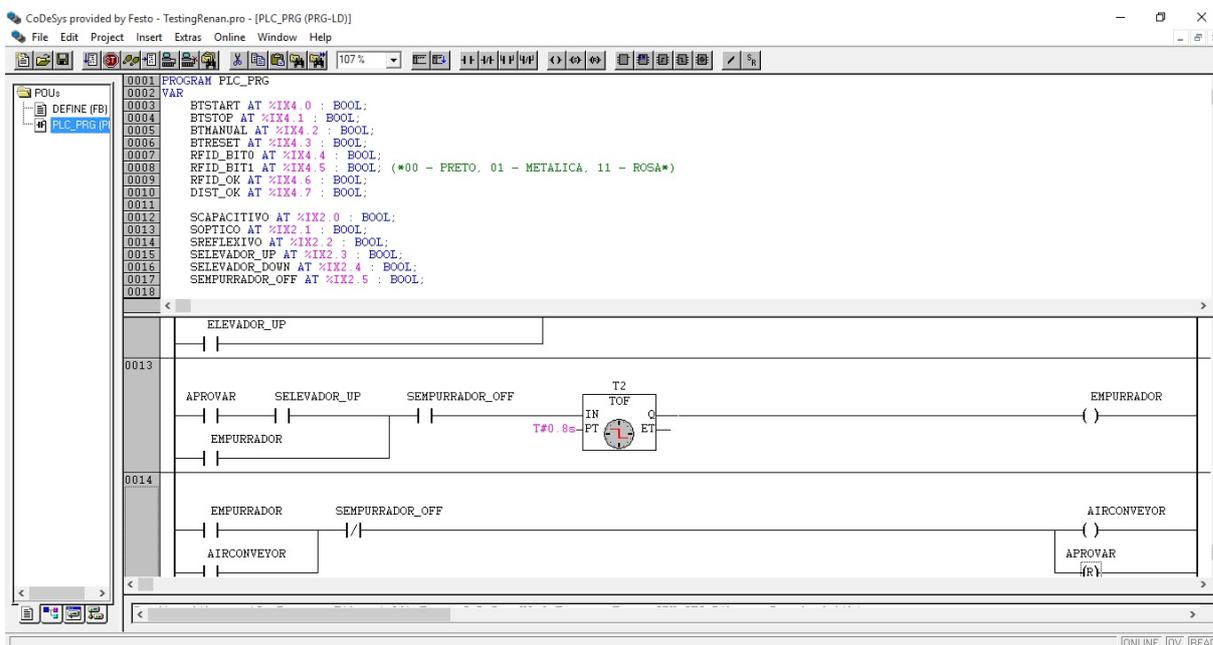
### 3.2.1 CoDeSys FESTO

De acordo com o manual do usuário, CoDeSys é um ambiente completo de programação para o CLP do projeto. CoDeSys vem da sigla de Controlled Development System, que em português significa Sistema de Desenvolvimento Controlado. Este software traz uma abordagem simples para as linguagens IEC que estão disponíveis para o programador do CLP. O uso de editores e de funções de debug são baseados em programas comprovados de linguagens de programação avançada (3S - SMART SOFTWARE SOLUTIONS GMBH, 2010).

Este software foi escolhido pois já possui diversas configurações para a programação do CLP, além de ser bastante didático e possuir diversos tutoriais

disponíveis na internet. A figura 12 mostra o ambiente para programação do CoDeSys 2.3.

Figura 13 – Ambiente de Programação do CoDeSys 2.3



Fonte: (LIMA, 2015)

### 3.3 Modelagem Rede de Petri

Devido ao tamanho da modelagem Rede de Petri, ela foi dividida em 4 partes, a serem apresentadas nas subseções a seguir. A modelagem completa pode ser encontrada no Apêndice A.

#### 3.3.1 Compra de produtos

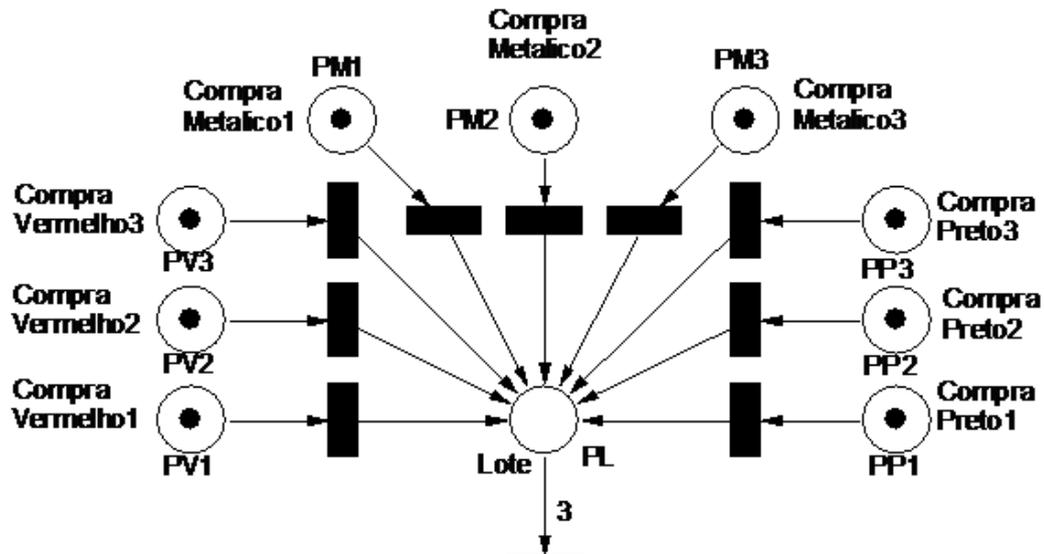
O início de todo o processo é feito na página Web, esta possui 9 produtos que são combinações entre as cores das peças e as guias as quais elas irão ser classificadas. Quando o usuário efetua uma compra, a informação é lida pelo sistema supervisorio, que espera a formação de um lote de pelo menos três peças para iniciar a produção. Este processo está representado na figura 14.

#### 3.3.2 Estação de Distribuição

A modelagem apresentada na figura 15 se inicia com a formação de um lote que ocorre conforme mostrado na subseção anterior. Ao analisar a figura 15 é possível

compreender que a modelagem pode ser dividida em três subsistemas: empurrador, pick-up e ventosa. Cada um destes subsistemas pode assumir dois estados possíveis: o empurrador pode estar avançado ou recuado, o pick-up pode estar na posição inicial ou na posição final, e a ventosa pode estar ativada ou desativada.

Figura 14 – Modelagem em Rede de Petri - Formação de Lote



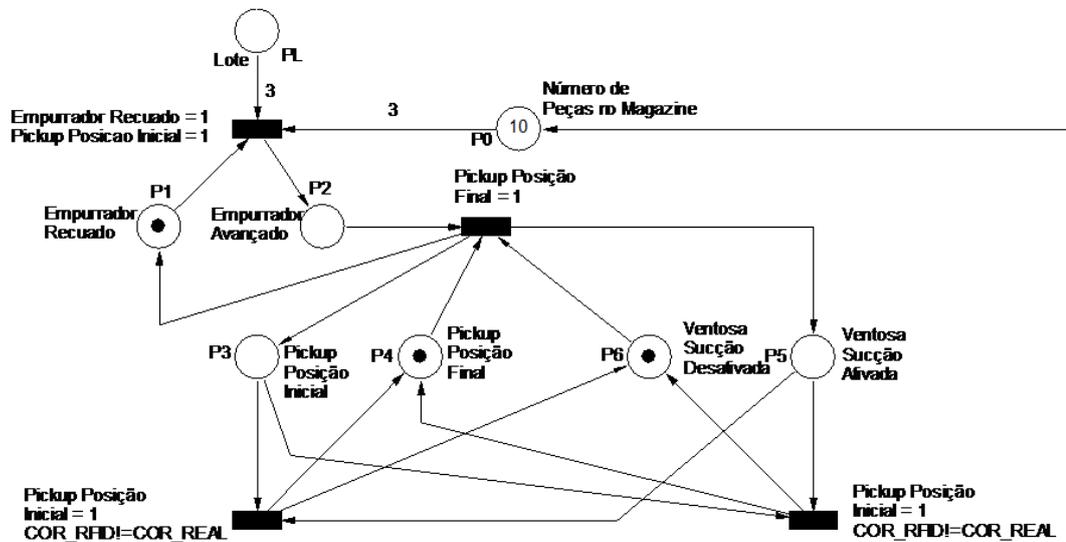
Fonte: (LIMA, 2015)

Tão importante quanto os estados são as transições para estes estados. Sendo assim, para o empurrador mudar do estado recuado para avançado, é necessário que o pick-up esteja na posição inicial e que o sensor do empurrador recuado estejam acionados, e para retornar ao estado recuado, é necessário que o sensor do pick-up posição final seja acionado.

O pick-up só pode ir de uma posição a outra se o mesmo chegar completamente na posição solicitada, por esse motivo, em suas transições estão os extremos das posições do transportador.

A ventosa possui transições similares ao do pick-up, pois eles trabalham em conjunto para o transporte da peça da estação de distribuição para a estação de teste.

Figura 15 – Modelagem em Rede de Petri - Estação de Distribuição



Fonte: (LIMA, 2015)

### 3.3.3 Estação de Teste

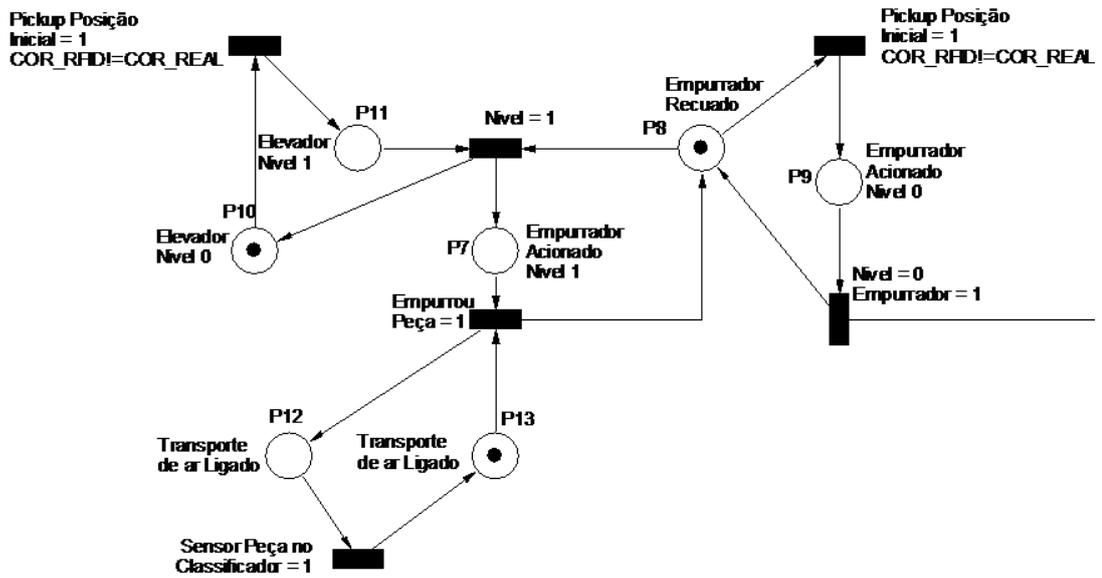
A modelagem da estação de teste pode ser dividida em três subsistemas: empurrador, elevador e transportador de ar, porém neste caso, os subsistemas possuem um número diferente de estados: o elevador possui dois estados, nível 1 e nível 0, o empurrador possui três estados, recuado, avançado no nível 0 e avançado no nível 1 e o transportador de ar possui dois estados, ligado e desligado.

As transições do elevador dependem da leitura do chip RFID, pois se a cor lida for igual à cor da peça o elevador vai para o nível 1, e retorna para o nível zero ao atingir o extremo do nível 1 e acionar o sensor de fim de curso.

O empurrador, assim como o elevador, também depende da leitura do chip RFID, pois ele só estará no estado de acionado no nível 0 se a cor lida no chip RFID for diferente da cor real da peça, o que empurra a peça para o rejeito da estação de teste, a peça é então levada de volta para o magazine da estação de distribuição. Para que o empurrador seja acionado no nível 1 é necessário que o elevador esteja no nível 1, e para que o empurrador retorne para o estado recuado é necessário que ele esteja completamente avançado.

Para que o transportador de ar passe do estado desligado para o ligado, o empurrador no nível 1 deve ter empurrado uma peça e para retornar ao estado desligado o sensor que indica que a peça chegou na próxima estação deve ser acionado, como é possível ver na figura 16.

Figura 16 – Modelagem em Rede de Petri - Estação de Teste

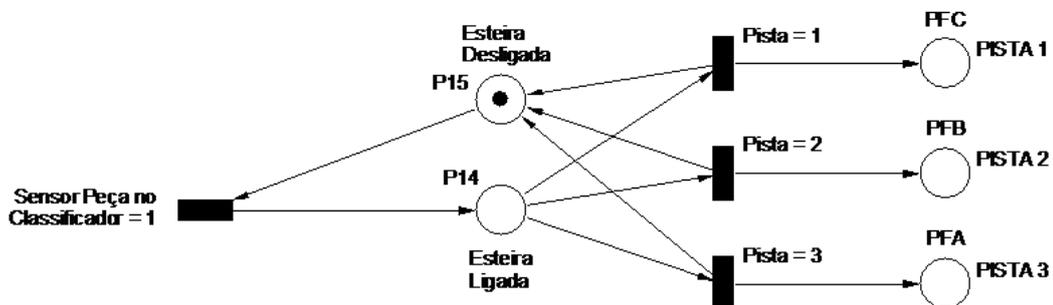


Fonte: (LIMA, 2015)

### 3.3.4 Estação de Classificação

Nesta estação, o acionamento dos desvios foi representado como as três pistas nas quais a peça pode ser classificada. O subsistema da esteira tem apenas dois estados, ligado e desligado, no qual possui a transição de desligado para ligado no momento em que é feita a leitura de para qual pista a peça deve ir. A esteira volta ao estado desligado quando a peça é corretamente classificada, como mostrado na figura 17.

Figura 17 – Modelagem em Rede de Petri - Estação de Classificação



Fonte: (LIMA, 2015)

### 3.4 RFID

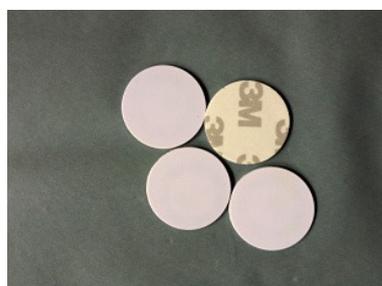
Como apresentado na seção 2.3, um sistema RFID deve apresentar três componentes principais: tag RFID, leitor/gravador RFID e um subsistema de processamento de dados, o qual será representado pelo microcontrolador Raspberry Pi, que irá processar os dados provenientes do leitor/gravador RFID e enviará a informação processada ao CLP.

#### 3.4.1 Tag RFID

As Tags RFID são dispositivos que carregam a informação sem a necessidade de uma fonte de alimentação e podem ter variados tamanhos, capacidade de memória, distâncias e frequências de trabalho. A tag escolhida para o projeto deve ser capaz de identificar os produtos do sistema modular de produção, portanto foi escolhida uma etiqueta adesiva de tamanho reduzido, com capacidade de 1024 bytes de memória, faixa de leitura entre 0 e 100mm e frequência de trabalho de 13.56MHz.

A figura 18 ilustra como são as tags RFID escolhidas para o projeto.

Figura 18 – Tags RFID escolhidas para a pesquisa



Fonte: (VIZION2000, 2015)

#### 3.4.2 Leitor/Gravador RFID

Por meio do leitor/gravador RFID é possível gravar pequenas informações em tags RFID. Existem vários leitores/gravadores no mercado, os quais possuem diferentes faixas de frequência de trabalho, a frequência escolhida foi 13,56 MHz devido aos diversos tipos de tags existentes no mercado que também trabalham nesta frequência.

A figura 19 mostra o módulo de leitura RFID escolhido para o projeto, o qual deve ser conectado a um microcontrolador para processar as informações lidas.

Figura 19 – Leitor/Gravador RFID Mifare 13.56MHz RC522



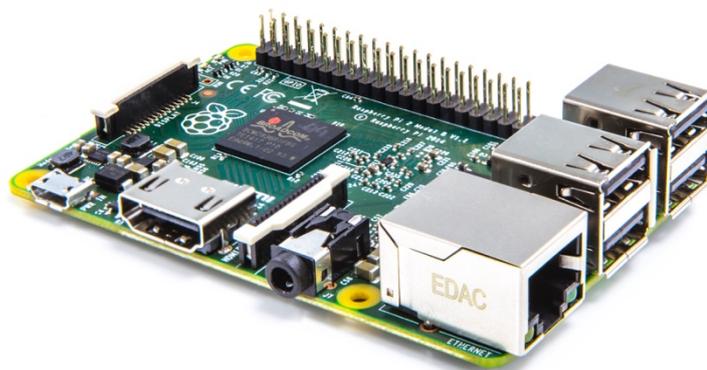
Fonte: (THAOYU ELETRONICS, 2015)

### 3.5 Raspberry Pi

Existem até o momento, alguns modelos de microcontroladores Raspberry Pi no mercado. Os modelos que possuem possibilidade de comunicação por Ethernet são os modelos B e B+, portanto, como o projeto possui a necessidade de comunicação em rede com endereçamento único para cumprir os pré-requisitos da aplicação do conceito de Internet das Coisas, foi escolhido o modelo Raspberri Pi 2, que além de possuir essas possibilidades, também possui mais memória e melhor processador, dando mais rapidez ao processo.

A figura 20 mostra o modelo escolhido para o presente projeto.

Figura 20 – Raspberry Pi 2



Fonte: (RASPBERRY PI FOUNDATION, 2015)

### 3.6 Sistema Supervisório

Para se ter um software completo e de qualidade, alguns passos devem ser seguidos. O começo da modelagem de um sistema se dá pelo levantamento dos pré-requisitos, no qual são listadas todas as funcionalidades das telas do sistema. Em

seguida, deve-se realizar a modelagem do banco de dados, o qual deve ser capaz de suprir todas as necessidades de salvamento de dados do sistema.

Como a linguagem escolhida é orientada à objetos, é possível antes de se iniciar a programação, desenvolver uma modelagem UML, que já possui todas as classes necessárias para o sistema, auxiliando assim, no último passo, a programação do sistema em si.

### 3.6.1 Requisitos Funcionais

Serão apresentados nesta subseção os requisitos funcionais para o desenvolvimento do sistema supervisorio.

#### 3.6.1.1 Controle de Acesso

Para manter a integridade das informações do sistema supervisorio, é necessário um controle de acesso. Para este projeto foram escolhidos dois tipos de usuários: administrador e operador.

O primeiro tipo (administrador) tem permissão para acessar todas as telas do sistema, e é o único que pode realizar o cadastro de novos usuários. O segundo tipo (operador) tem a capacidade apenas de visualizar o processo por meio da interface gráfica.

#### 3.6.1.2 Controle de Estoque

De acordo com Santos e Rodrigues (2006), “a utilização de estoques, seja de segurança ou de cobertura para atender à demanda média durante o lead time é extremamente importante porque possibilita um melhor nível de atendimento ao cliente e melhora a competitividade da empresa em relação aos concorrentes”.

O controle de estoque incorporado ao sistema deve possibilitar o cadastro e visualização dos materiais utilizados no processo. Além disso, deve também realizar uma tomada de decisão automática caso os materiais estejam se esgotando.

### 3.6.1.3 Supervisão do Processo

O sistema deve possibilitar ao usuário uma completa visualização do processo por meio de uma interface de fácil entendimento das três estações de sistema modular de produção.

A supervisão do processo deve ser dividida em quatro telas, uma de visualização completa e as outras três devem representar cada uma das estações de sistema modular de produção. A tela de visualização completa mostra onde cada peça está no processo juntamente com as informações de quem foi o comprador e qual será o produto final a ser feito a partir dela. As outras três telas de supervisão devem apresentar os níveis dos sensores e atuadores das três estações.

### 3.6.1.4 Acompanhamento de Produção

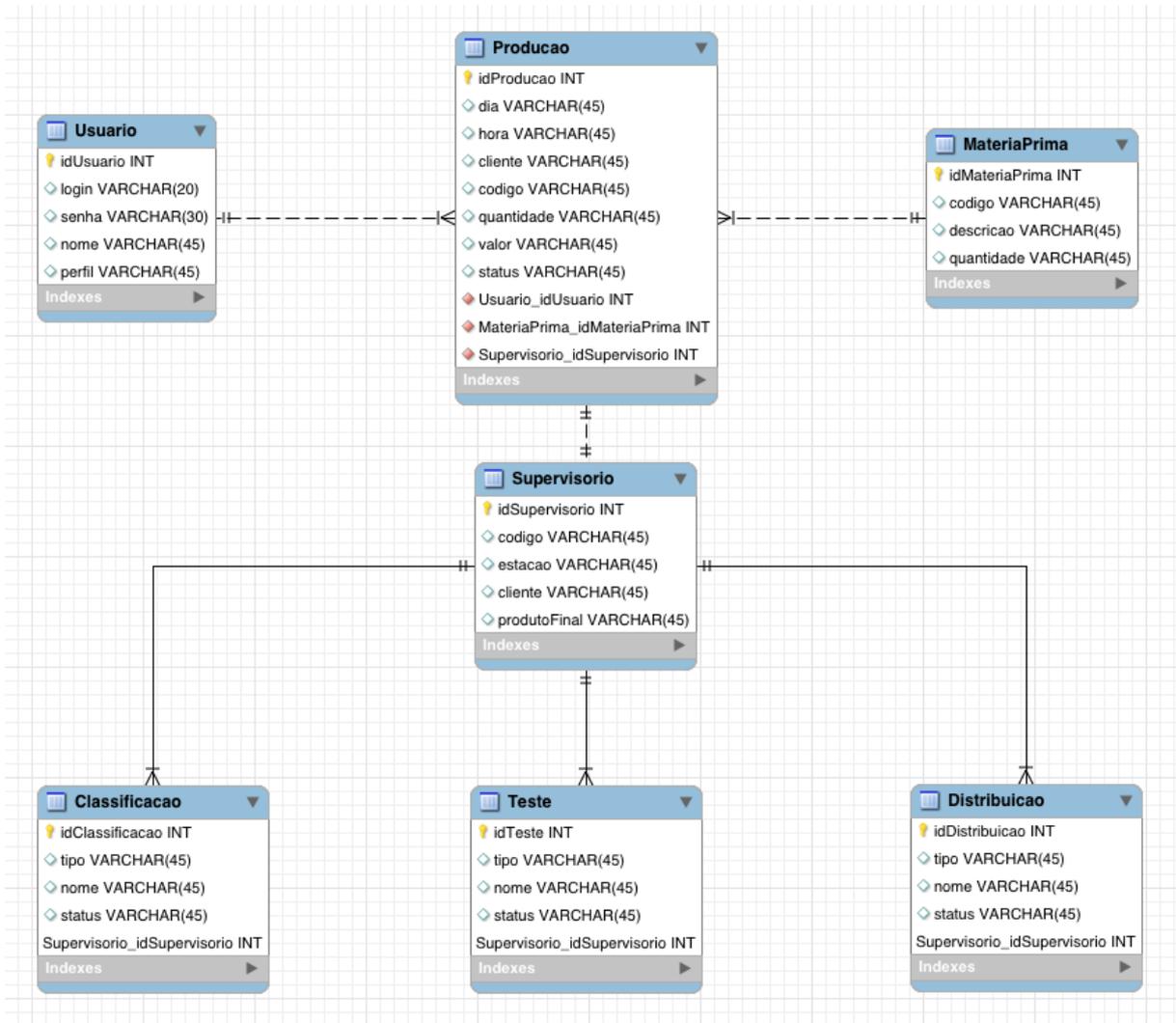
O acompanhamento de produção deve mostrar todo o histórico de compras e fabricação de produtos, mostrando as informações de quando foi a compra, quem foi o comprado, o código do produto, o valor e o status da produção.

## 3.6.2 Modelagem do Banco de Dados

O banco de dados foi modelado a partir das informações do levantamento de pré-requisitos do sistema, de forma a que todos os dados do sistema pudessem ser salvos, alterados e visualizados. O Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido para a pesquisa foi o MySQL, o qual fornece algumas ferramentas para auxiliar na modelagem do banco de dados, que é o caso do MySQL Workbench, software escolhido para a modelagem do banco do projeto.

A figura 21 mostra a modelagem do banco de dados para o projeto em questão.

Figura 21 – Modelagem do banco de dados



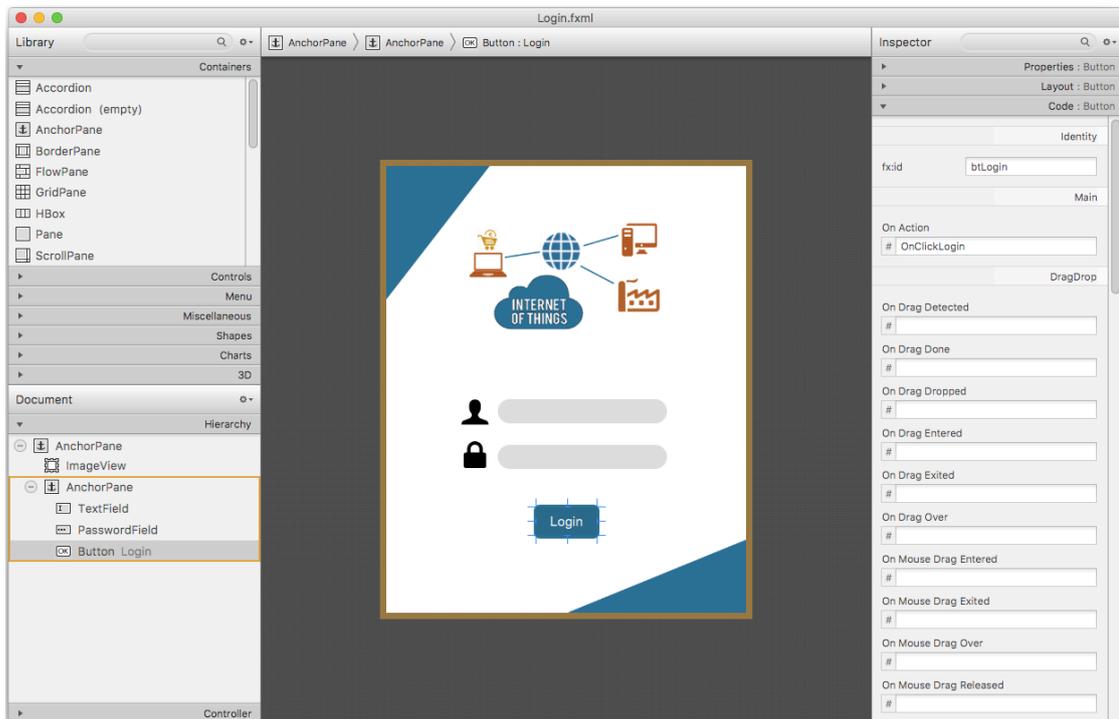
Fonte: Autor

### 3.6.3 JavaFX Scene Builder (Layout das Telas)

O JavaFX Scene Builder é uma ferramenta que facilita a criação de telas em formato FXML. Por meio deste mecanismo é possível clicar e arrastar componentes para a tela a ser criada, e ao fim o arquivo .fxml será gerado automaticamente e poderá ser utilizado nas aplicações Java.

A figura 22 mostra o ambiente de desenvolvimento do JavaFX Scene Builder.

Figura 22 – Ambiente de desenvolvimento do JavaFX Scene Builder



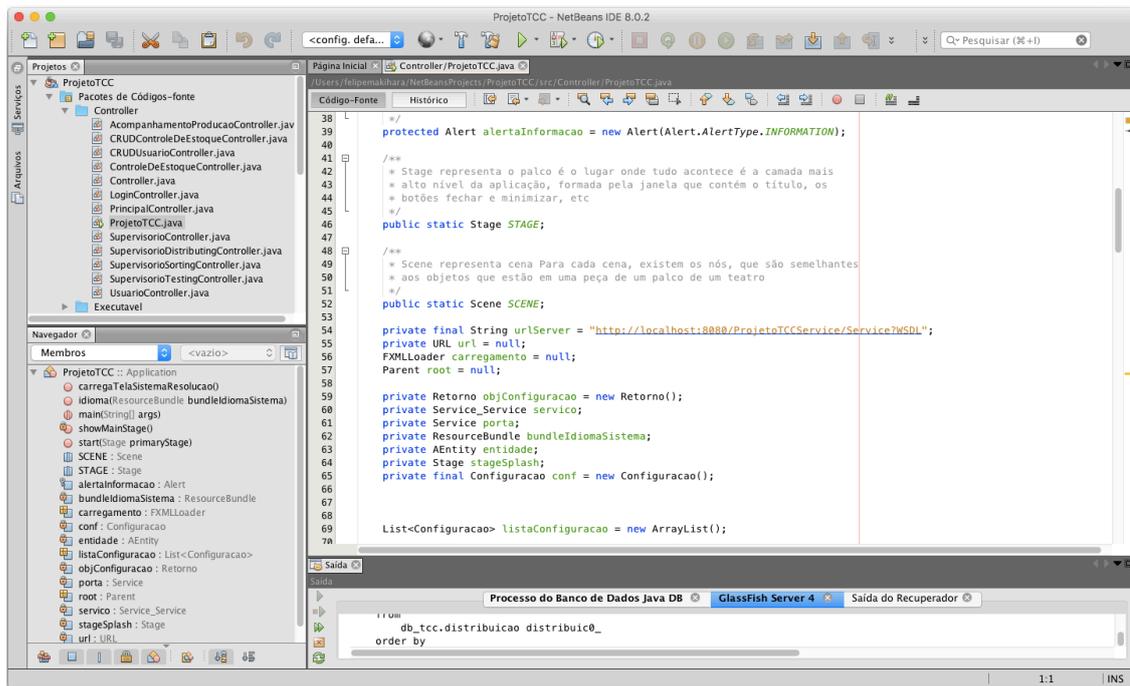
Fonte: Autor

### 3.6.4 Netbeans 8.1 (Programação Java e JavaWeb)

Segundo o livro de Boudreau et al. (2011), Netbeans é uma IDE, ou Integrated Development Environment, que se traduz para o português como um ambiente integrado de desenvolvimento. É uma ferramenta poderosa, útil, extensível e em código aberto.

O objetivo do software é auxiliar o programador por meio de ferramentas avançadas, pois o mesmo oferece suporte para diferentes linguagens e proporciona diversos mecanismos, como avisos de erros, inspeção de componentes, navegador de objetos, assistente de diálogo e depurador. A figura 23 mostra o ambiente de desenvolvimento do Netbeans versão 8.1.

Figura 23 – Ambiente de desenvolvimento Netbeans 8.1



Fonte: Autor

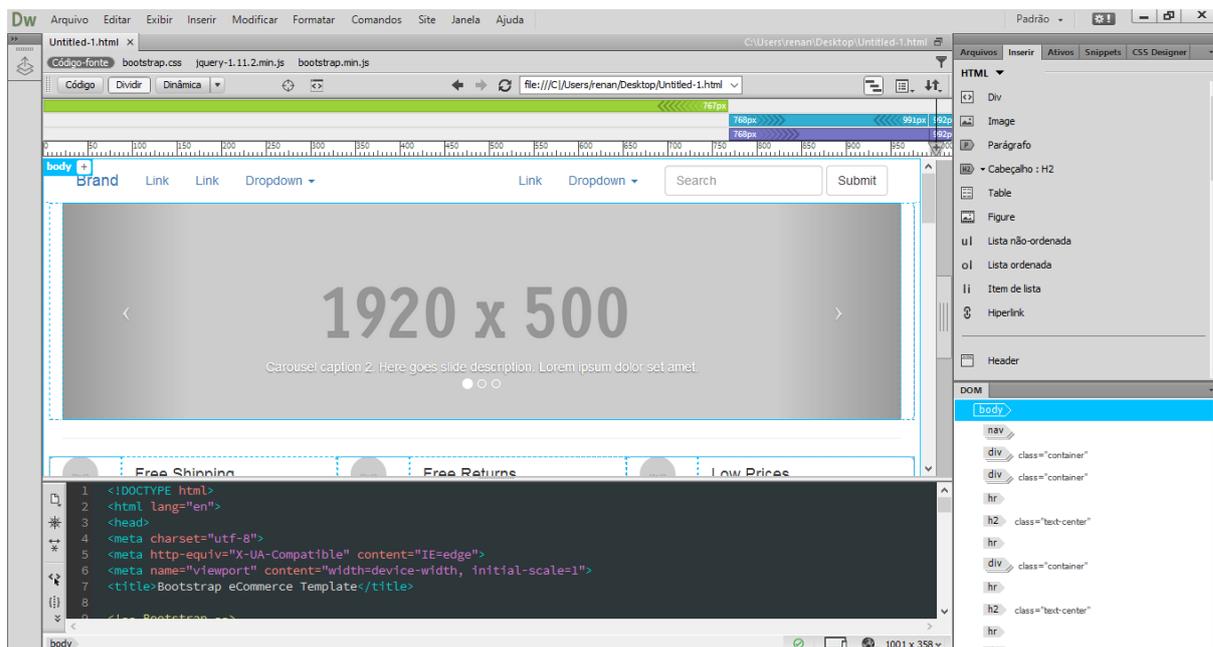
### 3.7 Adobe DreamWeaver (Layout do Site em HTML)

De acordo com Team (2012), o Adobe DreamWeaver é um criador de sites dinâmicos que possui um conjunto completo de ferramentas de criação para Web e ajuda a criar sites que se adaptem a qualquer tamanho de tela, com a possibilidade de aprimorar a aparência com a adição de imagens.

O software oferece uma poderosa combinação de ferramentas para modelos visuais, funções de desenvolvimento de aplicativos e suporte à edição por código, permitindo que desenvolvedores de qualquer nível criem sites com um bom visual e aplicativos rapidamente.

A figura 24 ilustra como é o ambiente de desenvolvimento do Adobe DreamWeaver.

Figura 24 – Ambiente de desenvolvimento do Adobe DreamWeaver



Fonte: (LIMA, 2015)

### 3.8 CoDeSys OPC Server (Servidor OPC)

Segundo as informações do manual da 3S - Smart Software Solutions GmbH (2015), o principal objetivo do CoDeSys OPC Server é o intercâmbio de dados (leitura / escrita) entre, por exemplo, um sistema supervisor para a visualização ou registro de dados.

Dentre as funcionalidades do software, se destacam:

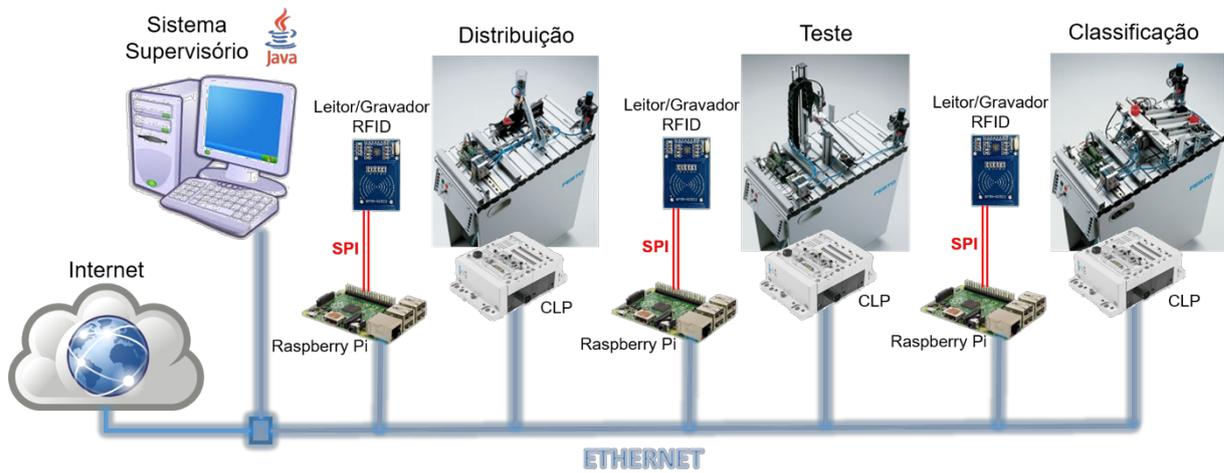
- Desenvolvimento de projetos no formato IEC 61131-3 no CoDeSys;
- Configuração de símbolos: criação de objetos que representam os valores das variáveis no processo via OPC;
- Realizar login no CLP e executar as aplicações;
- Fornecimento automático de símbolos via CoDeSys OPC Server;
- Iniciar o cliente OPC no computador e estabelecer conexão entre o CLP e o CoDeSys OPC Server.

### 3.9 Rede de Comunicação

Como a intenção da pesquisa é a aplicação do conceito de Internet das Coisas no processo, um dos pilares do conceito foi levado em conta ao desenvolver a rede

de comunicação: o endereçamento único de todos os dispositivos na rede. Sendo assim, conforme mostrado na figura 25, todos os CLPs e microcontroladores estão conectados por cabos Ethernet a um switch, que está conectado ao computador em que se encontra o sistema supervisor e também à Internet.

Figura 25 – Rede de comunicação do sistema



Fonte: (LIMA, 2015)

## 4 RESULTADOS OBTIDOS

A partir da utilização da metodologia apresentada no Capítulo 3, foram produzidos os resultados que serão apresentados a seguir neste capítulo.

### 4.1 Página Web

Foi desenvolvida uma página Web para simular a compra dos produtos a serem fabricados no Sistema Modular de Produção. A página principal, como mostrado na figura 26, possibilita a compra de 9 (nove) produtos diferentes onde existem 3 (três) tipos para cada modelo de peça, sendo elas a peça preta, a peça vermelha e a peça metálica e seu tipo é responsável por dizer em qual saída a peça será designada na estação de Classificação.

Figura 26 – Página Web - Compra de Produtos

Internet of Things Login/Registrar-se Carrinho

Trabalho de Conclusão de Curso - Felipe Akio Vieira Makihara  
Site para simulação de compras

Pesquisar

MENU

Acompanhamento de Produtos  
Carrinho  
Produtos

Produtos

1 R\$1,00 Preto Tipo 1 COMPRAR	2 R\$1,00 Preto Tipo 2 COMPRAR	3 R\$1,00 Preto Tipo 3 COMPRAR
1 R\$2,00 Vermelho Tipo 1 COMPRAR	2 R\$2,00 Vermelho Tipo 2 COMPRAR	3 R\$2,00 Vermelho Tipo 3 COMPRAR
1 R\$3,00 Metálico Tipo 1 COMPRAR	2 R\$3,00 Metálico Tipo 2 COMPRAR	3 R\$3,00 Metálico Tipo 3 COMPRAR

Trabalho de Conclusão de Curso  
Engenharia de Controle e Automação  
Universidade do Estado do Amazonas  
Escola Superior de Tecnologia

Felipe Akio Vieira Makihara  
felipe.makihara@gmail.com

Fonte: Autor

Após a compra ser realizada, é possível acompanhar sua produção pela seção de “Acompanhamento de Produtos” apresentando em qual etapa a peça se encontra ou se ela já foi finalizada, assim como a data e a hora em que seu status foi alterado.

As etapas são “Em Produção (Distribuição)” para a estação de Distribuição, “Em Produção (Teste)” para a estação de Testes, “Em Produção (Classificação)” para a estação de Classificação e “Finalizada” quando a peça termina de ser produzida, na figura 27 é mostrada o acompanhamento realizado pelo Administrador e na figura 28 é mostrada o acompanhamento realizado pelos Clientes.

Figura 27 – Página Web - Acompanhamento de Produtos (Administrador)

**Internet of Things** Login/Registrar-se Carrinho

**Trabalho de Conclusão de Curso - Felipe Akio Vieira Makihara**  
Site para simulação de compras

Pesquisar

**MENU**

- Acompanhamento de Produtos
- Carrinho
- Produtos

**Acompanhamento de Produtos**

Data	Hora	Cliente	Código	Quantidade	Valor	Status
08/12/16	09:19:30	Almir Kimura Junior	001003	2	R\$2,00	Finalizado
08/12/16	09:27:12	Almir Kimura Junior	002001	3	R\$6,00	Finalizado
08/12/16	09:33:39	Almir Kimura Junior	003002	1	R\$3,00	Finalizado
12/12/16	20:02:11	Andrea Thiyomi Nagai	001002	4	R\$4,00	Finalizado
12/12/16	20:16:50	Andrea Thiyomi Nagai	002003	2	R\$4,00	Finalizado
12/12/16	20:25:28	Andrea Thiyomi Nagai	003001	2	R\$6,00	Finalizado
18/12/16	13:46:18	Giovanni Ribeiro Caldeira	001001	3	R\$3,00	Em Produção (Distribuição)
18/12/16	13:55:22	Giovanni Ribeiro Caldeira	002002	2	R\$4,00	Em Produção (Teste)
18/12/16	14:01:56	Giovanni Ribeiro Caldeira	003003	5	R\$15,00	Em Produção (Classificação)

**Trabalho de Conclusão de Curso**  
Engenharia de Controle e Automação  
Universidade do Estado do Amazonas  
Escola Superior de Tecnologia

**Felipe Akio Vieira Makihara**  
felipe.makihara@gmail.com

Fonte: Autor

Figura 28 – Página Web - Acompanhamento de Produtos (Clientes)

**Internet of Things** Login/Registrar-se Carrinho

**Trabalho de Conclusão de Curso - Felipe Akio Vieira Makihara**  
Site para simulação de compras

Pesquisar

**MENU**

- Acompanhamento de Produtos
- Carrinho
- Produtos

**Acompanhamento de Produtos**

Data	Hora	Cliente	Código	Quantidade	Valor	Status
08/12/16	09:19:30	Almir Kimura Junior	001003	2	R\$2,00	Finalizado
08/12/16	09:27:12	Almir Kimura Junior	002001	3	R\$6,00	Finalizado
08/12/16	09:33:39	Almir Kimura Junior	003002	1	R\$3,00	Finalizado

**Trabalho de Conclusão de Curso**  
Engenharia de Controle e Automação  
Universidade do Estado do Amazonas  
Escola Superior de Tecnologia

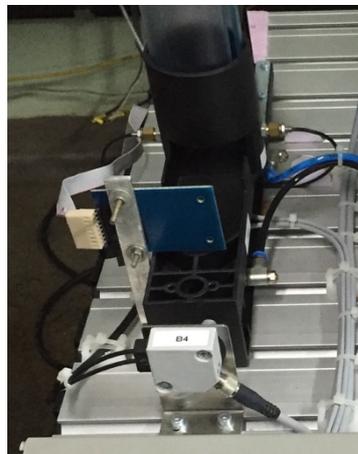
**Felipe Akio Vieira Makihara**  
felipe.makihara@gmail.com

Fonte: Autor

## 4.2 Transferência de informações para tag RFID

As informações relacionadas aos 3 (três) tipos de peças são lidas/escritas através dos leitores/gravadores *RFID* espalhados através do Sistema Modular de Produção possuindo uma unidade em cada estação, onde na Estação de Distribuição é responsável por escrever a informação da peça desejada, na Estação de Testes é responsável por ler se a informação está correta e na Estação de Classificação é responsável por ler a informação e separar cada tipo em uma saída diferente representados nas figuras 29, 30 e 31, respectivamente.

Figura 29 – Leitor/Gravador *RFID* na Estação de Distribuição



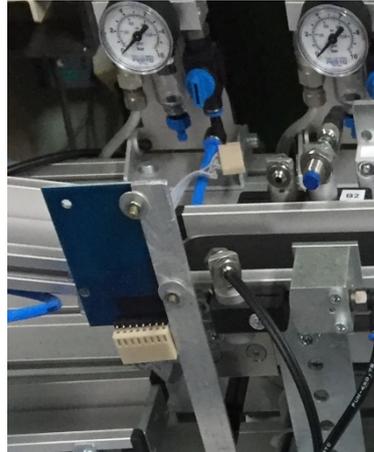
Fonte: Autor

Figura 30 – Leitor/Gravador *RFID* na Estação de Testes



Fonte: Autor

Figura 31 – Leitor/Gravador *RFID* na Estação de Classificação



Fonte: Autor

Estes leitores/gravadores *RFID* são controladas por um microcontrolador *Raspberry Pi* capaz de armazenar e gravar as informações demandadas pela produção nas tags *RFID* acopladas nas peças a serem produzidas como mostra na figura 32.

Figura 32 – Posicionamento da Tag *RFID* em uma peça do processo



Fonte: (LIMA, 2015)

#### 4.2.1 Programação em Python para os microcontroladores *Raspberry Pi*

Para comunicar os leitores/gravadores *RFID* com os microcontroladores *Raspberry Pi* foi utilizada a linguagem de programação Python pela facilidade das bibliotecas com a comunicação SPI e com o módulo MFRC522 e a partir delas foram desenvolvidos o código da leitura das tags *RFID* que se encontra no apêndice B, seção B.1, e o código da gravação das tags *RFID* que se encontra no apêndice B, seção B.2.

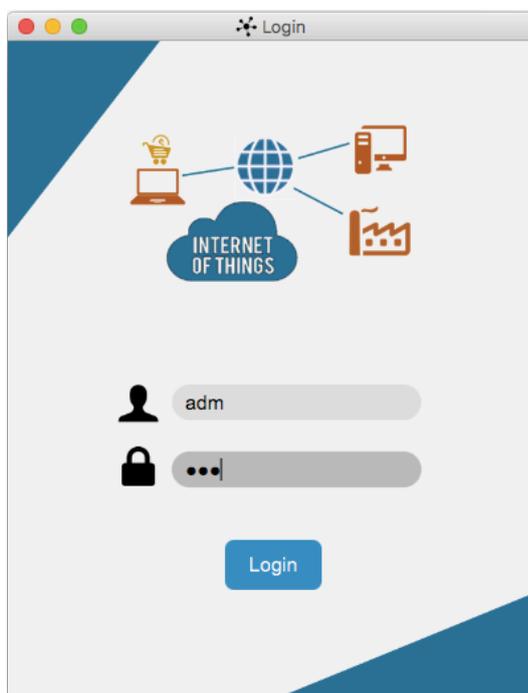
### 4.3 Sistema Supervisório

Nesta subseção serão apresentadas as telas do sistema que foram desenvolvidas com o objetivo de cumprir os pré-requisitos apresentados na subseção 3.6.1.

#### 4.3.1 Controle de Acesso

Para o Controle de Acesso, foi feita uma tela de login ilustrada na imagem 33 que realiza a verificação do usuário cadastrado e de acordo com o perfil do mesmo, o sistema libera ou bloqueia o acesso conforme descrito na subseção 3.6.1.1.

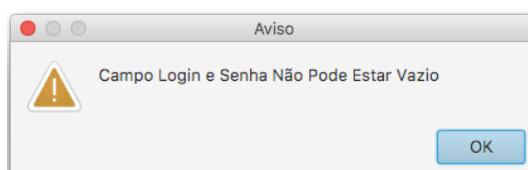
Figura 33 – Sistema Supervisório - Tela de Login



Fonte: Autor

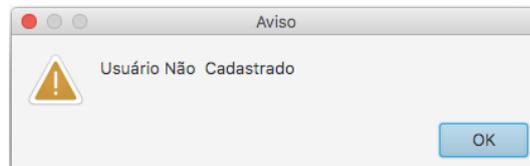
Em determinadas situações surgem telas de erros, como campos vazios, usuários inexistentes e senhas incorretas como mostram as figuras 34, 35 e 36, respectivamente.

Figura 34 – Janela de Erro de Campos Vazios



Fonte: Autor

Figura 35 – Janela de Erro de Usuário Inexistente



Fonte: Autor

Figura 36 – Janela de Erro de Senha Incorreta



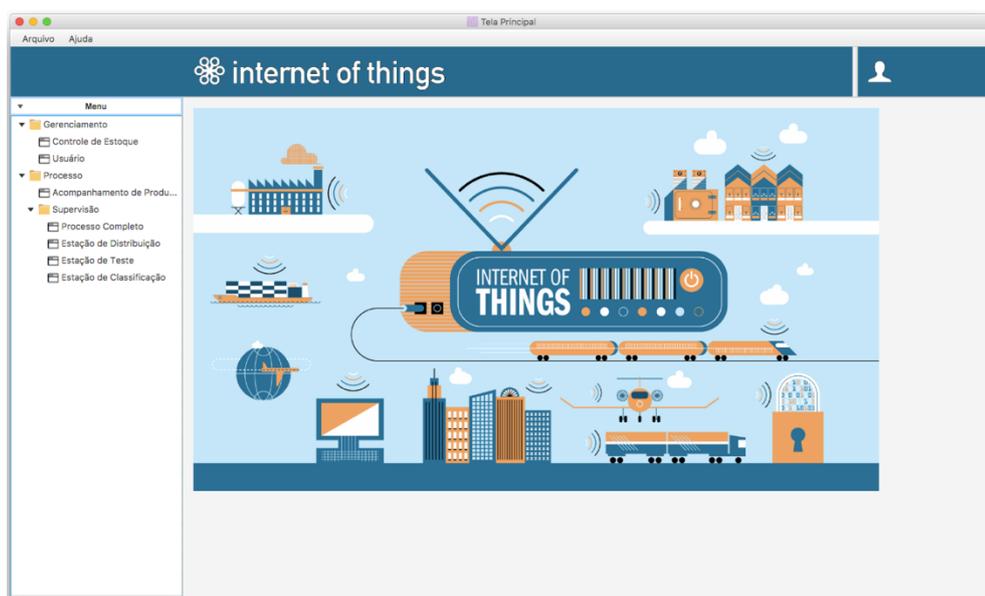
Fonte: Autor

#### 4.3.2 Tela Principal

A Tela Principal é responsável por possibilitar a interação entre todas as telas do sistema onde na pasta de “Gerenciamento” possui a entrada das telas de “Controle de Estoque” e de “Usuário”, na pasta de “Processo” possui a entrada das telas de “Acompanhamento de Produção” e uma subpasta “Supervisão” com a entrada das telas de “Processo Completo”, “Estação de Distribuição”, “Estação de Testes” e “Estação de Classificação”.

A figura 37 mostra a Tela Principal do Sistema.

Figura 36 – Sistema Supervisório - Tela Principal

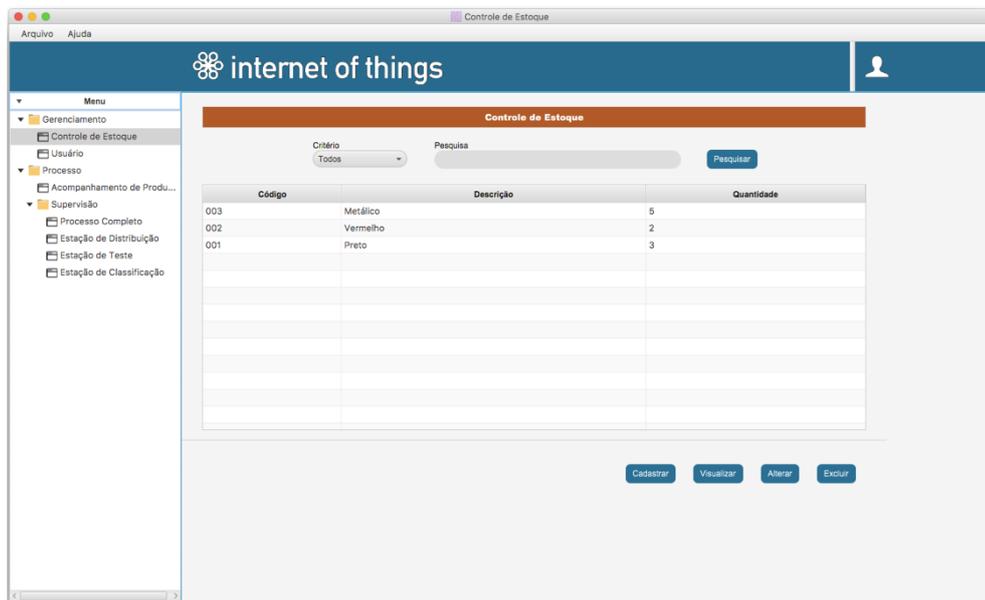


Fonte: Autor

### 4.3.3 Controle de Estoque

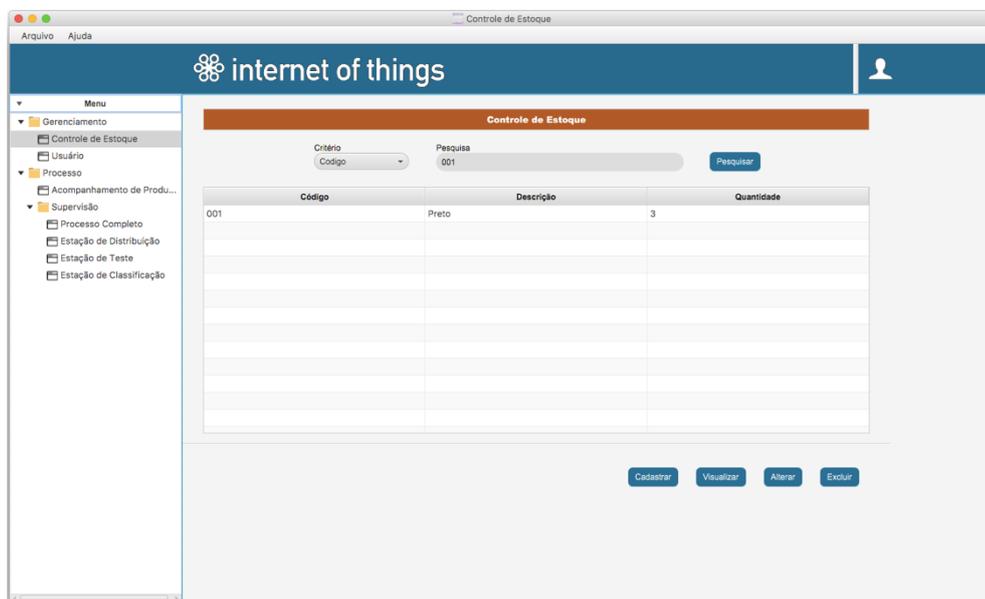
Para realizar o Controle de Estoque foi desenvolvida uma tela onde mostra em uma tabela as matérias primas organizadas em “Código”, “Descrição” e “Quantidade” possibilitando a pesquisa pelos critérios: “Todos”, “Código” e “Descrição”, assim como nas figuras 37, 38 e 39, respectivamente.

Figura 37 – Sistema Supervisório - Tela de Controle de Estoque - Pesquisa por Todos



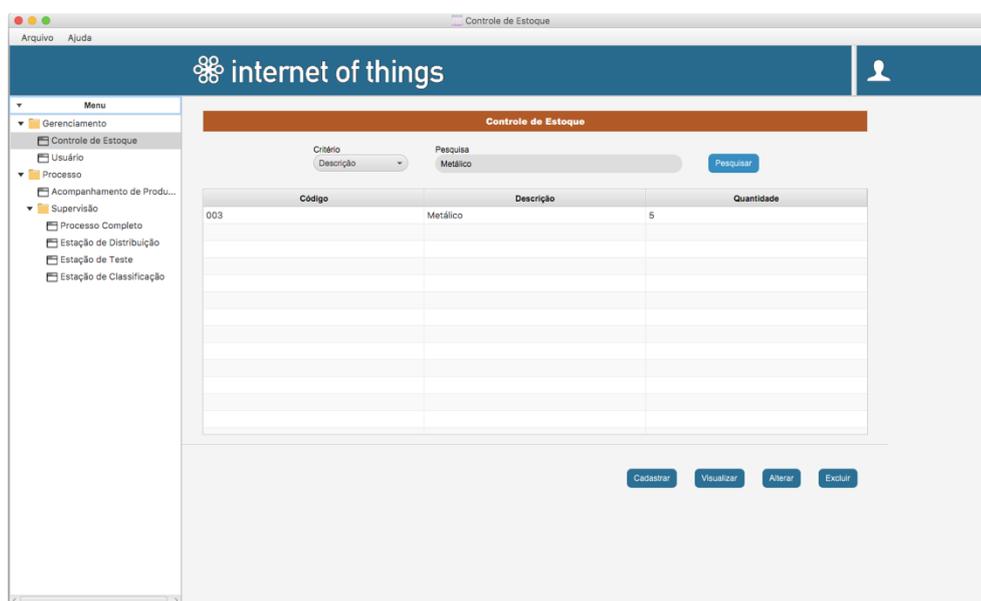
Fonte: Autor

Figura 38 – Sistema Supervisório - Tela de Controle de Estoque - Pesquisa por Código



Fonte: Autor

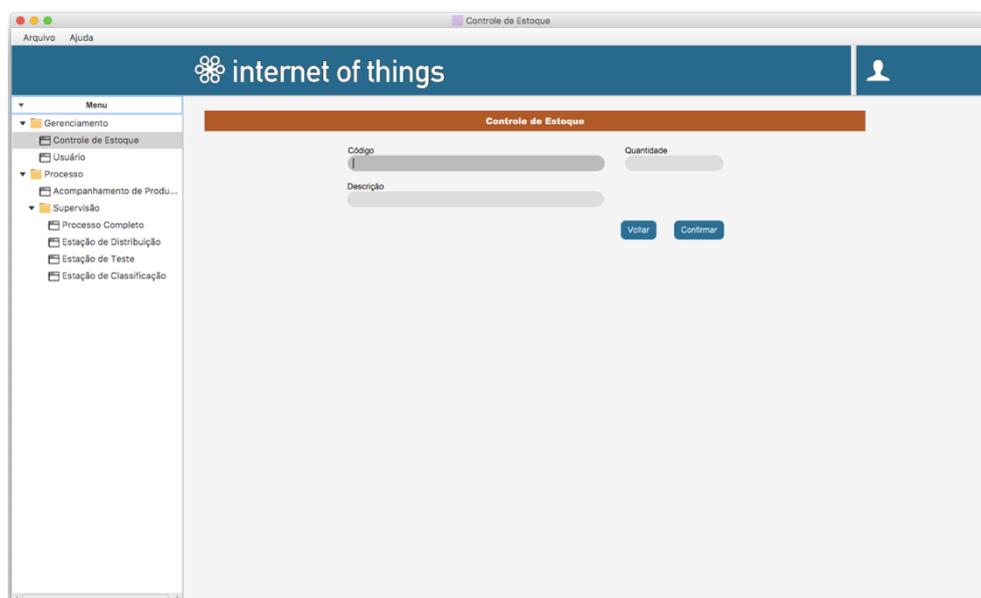
Figura 39 – Sistema Supervisório - Tela de Controle de Estoque - Pesquisa por Descrição



Fonte: Autor

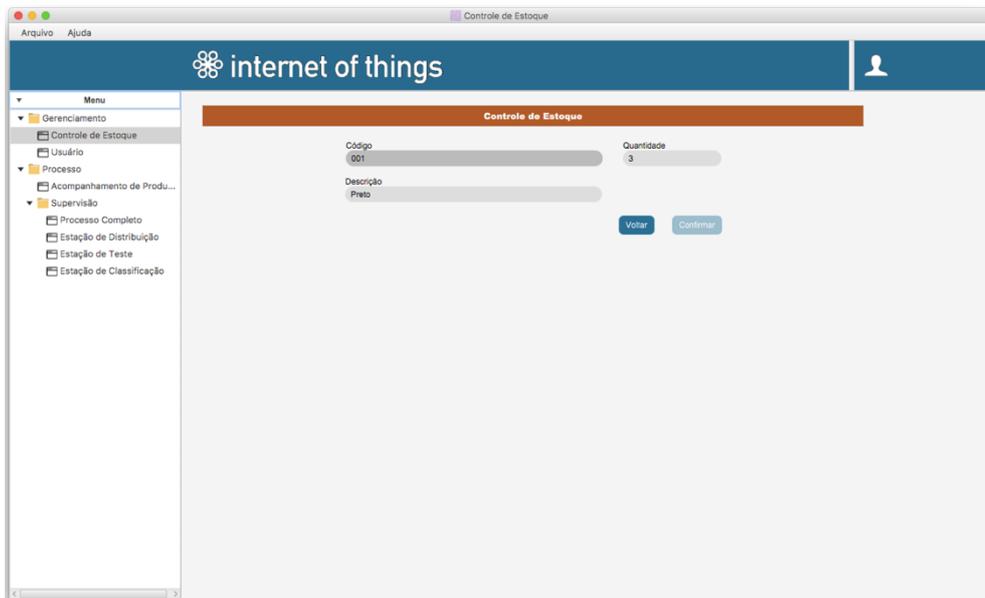
Para o Cadastro, Visualização e Alteração das Matérias-Primas, foi desenvolvida uma tela de CRUD (Create, Read, Update & Delete) possibilitando as ações assim como nas figuras 40, 41 e 42, respectivamente, onde na tela de Cadastro possibilita a inserção de uma nova matéria-prima no banco de dados, a tela de Visualização desabilita todos os campos com exceção do botão “Voltar”, impossibilitando a alteração dos mesmos e a tela de Alteração possibilita reescrever uma matéria-prima já existente no banco de dados.

Figura 40 – Sistema Supervisório - Tela de Cadastro do Controle de Estoque



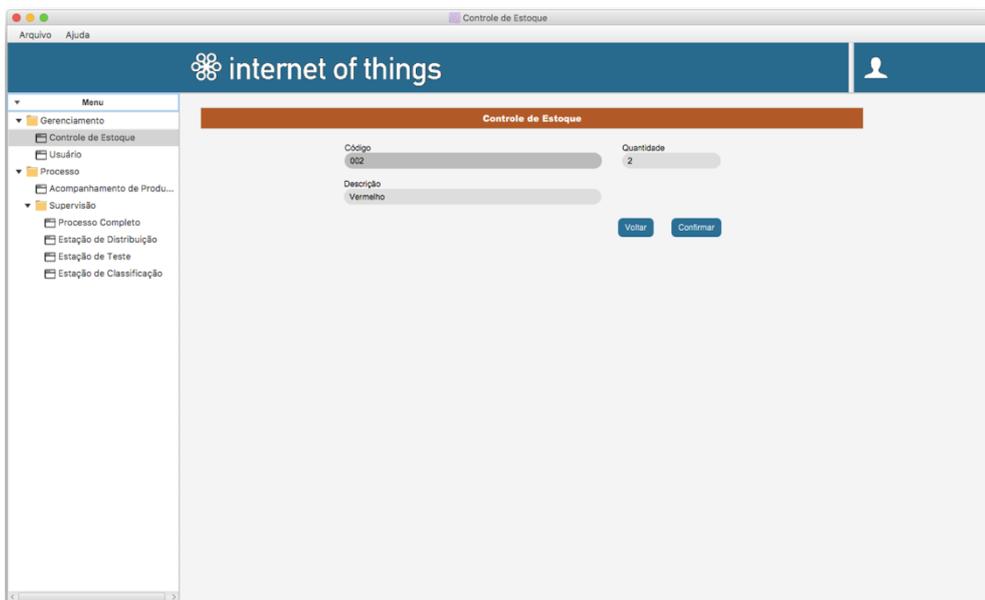
Fonte: Autor

Figura 41 – Sistema Supervisório - Tela de Visualização do Controle de Estoque



Fonte: Autor

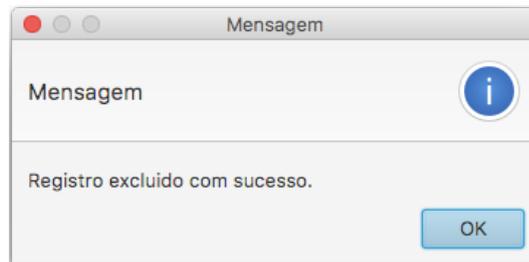
Figura 42 – Sistema Supervisório - Tela de Alteração do Controle de Estoque



Fonte: Autor

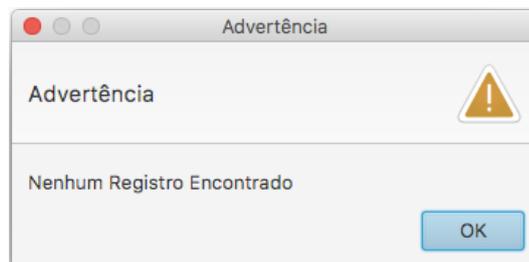
Por fim, foram feitas janelas de aviso para quando um registro for excluído e quando for pesquisar por critérios e o registro não existe, como mostram as figuras 43 e 44, respectivamente.

Figura 43 – Janela de Aviso de Exclusão de Registro



Fonte: Autor

Figura 44 – Janela de Aviso de Registro não encontrado

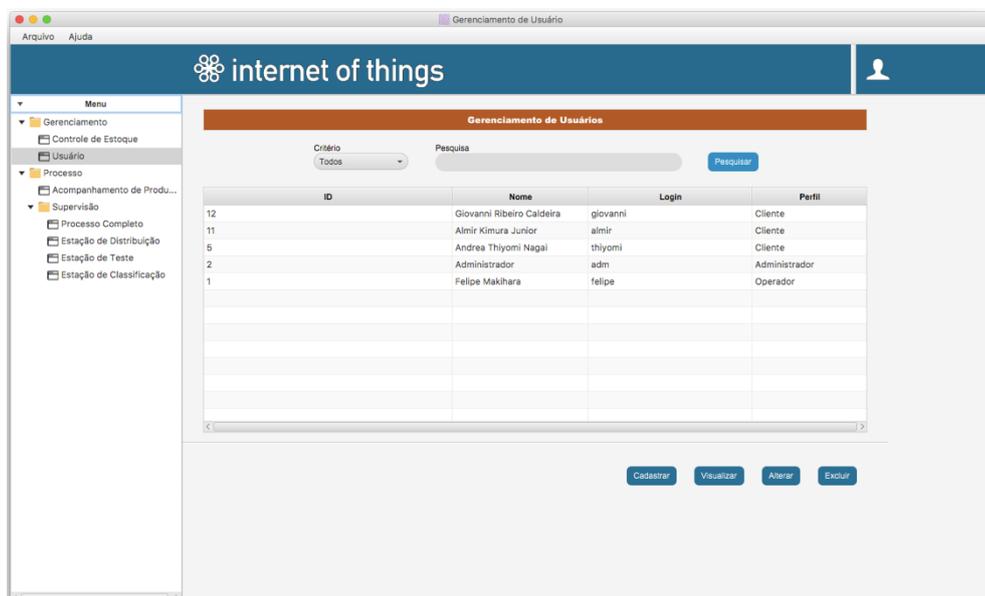


Fonte: Autor

#### 4.3.4 Gerenciamento de Usuários

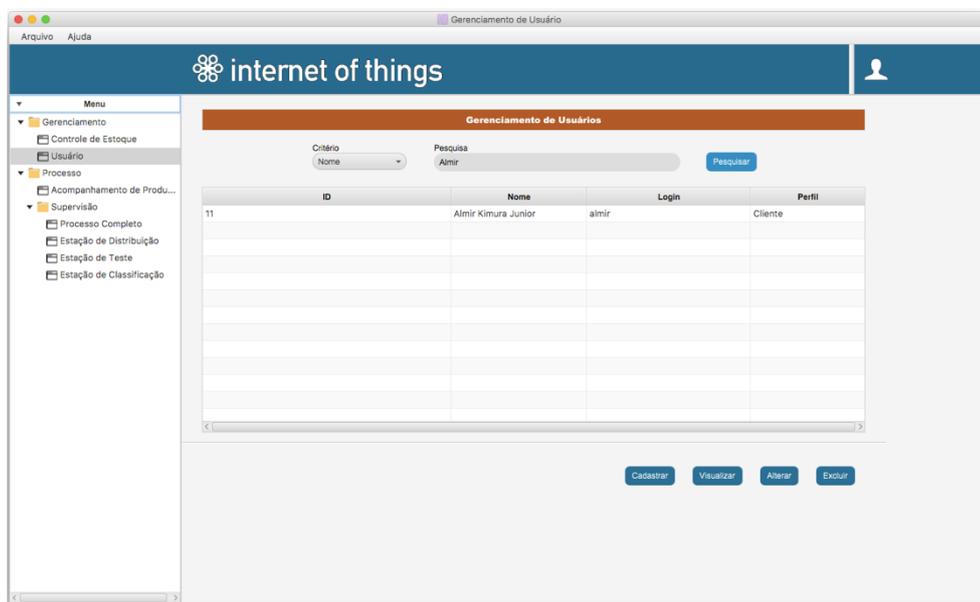
Quando o Administrador acessar o Sistema, a tela de Gerenciamento de Usuários se torna visível para acesso mostrando uma tabela com as colunas "ID", "Nome", "Login" e "Perfil" possibilitando a pesquisa pelos critérios: "Todos", "Nome", "Login" e "Perfil", assim como nas figuras 45, 46, 47 e 48, respectivamente.

Figura 45 – Sistema Supervisório - Tela de Gerenciamento de Usuários - Pesquisa por Todos



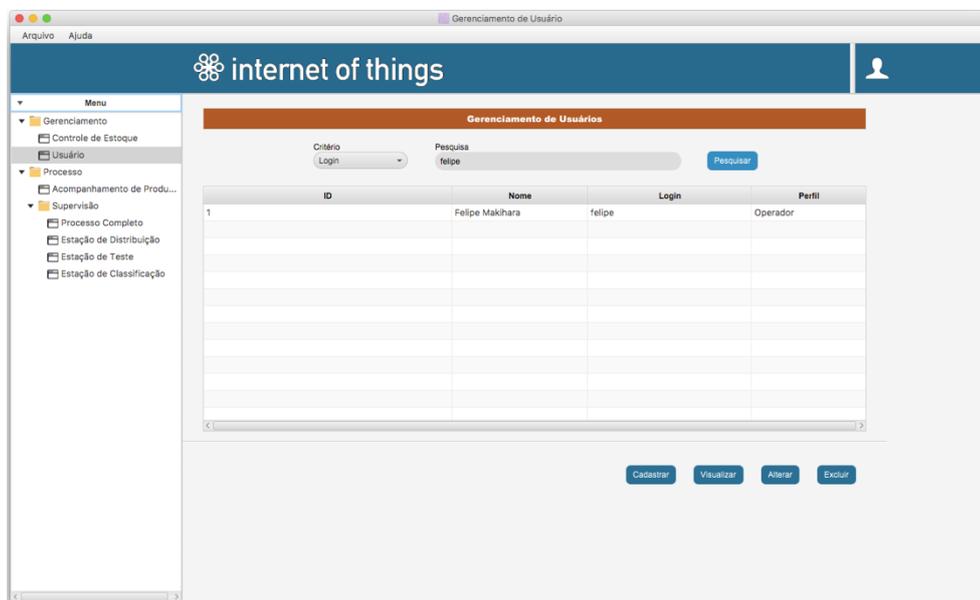
Fonte: Autor

Figura 46 – Sistema Supervisório - Tela de Gerenciamento de Usuários - Pesquisa por Nome



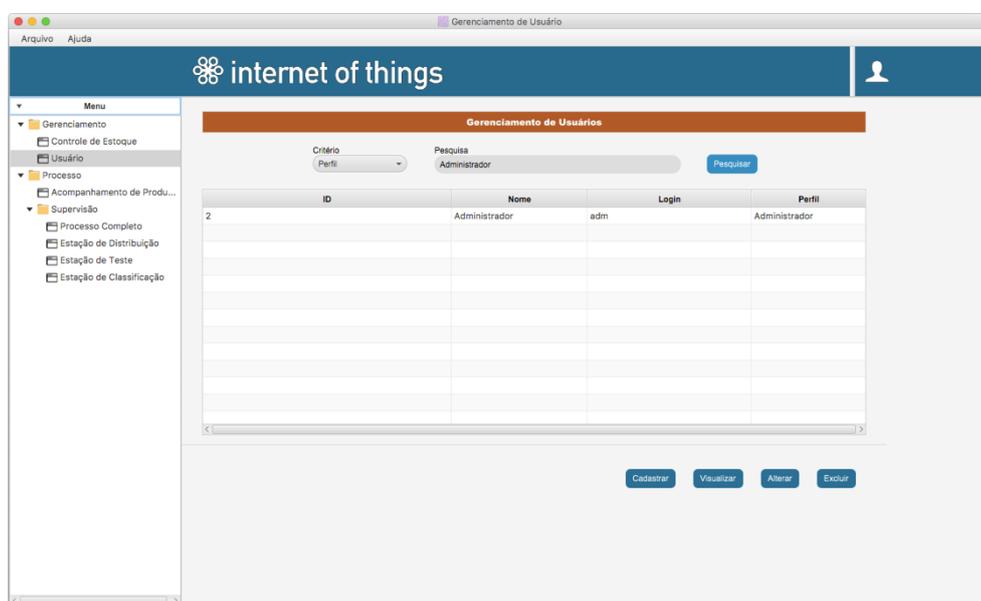
Fonte: Autor

Figura 47 – Sistema Supervisório - Tela de Gerenciamento de Usuários - Pesquisa por Login



Fonte: Autor

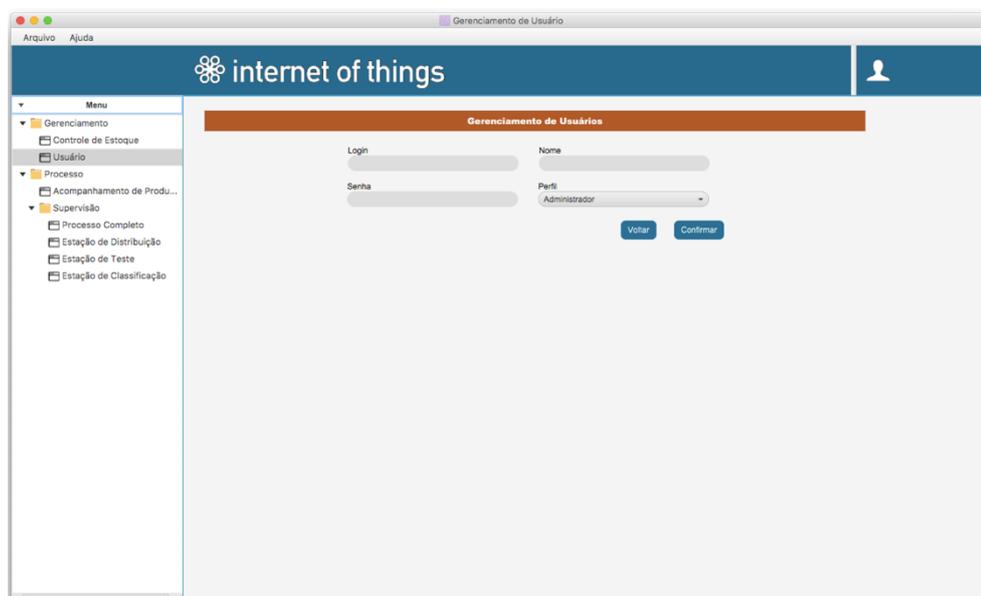
Figura 48 – Sistema Supervisório - Tela de Gerenciamento de Usuários - Pesquisa por Perfil



Fonte: Autor

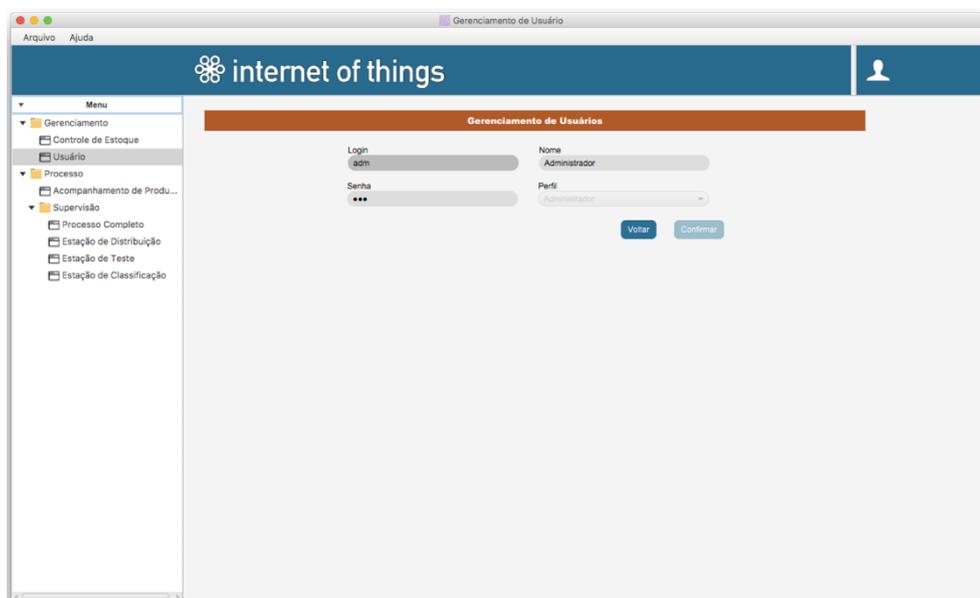
Para o Cadastro, Visualização e Alteração dos Usuários, foi desenvolvida uma tela de CRUD (Create, Read, Update & Delete) possibilitando as ações assim como nas figuras 49, 50 e 51, respectivamente, onde na tela de Cadastro possibilita a inserção de um novo cadastro no banco de dados, a tela de Visualização desabilita todos os campos com exceção do botão “Voltar”, impossibilitando a alteração dos mesmos e a tela de Alteração possibilita reescrever um usuário já existente no banco de dados.

Figura 49 – Sistema Supervisório - Tela de Cadastro do Gerenciamento de Usuários



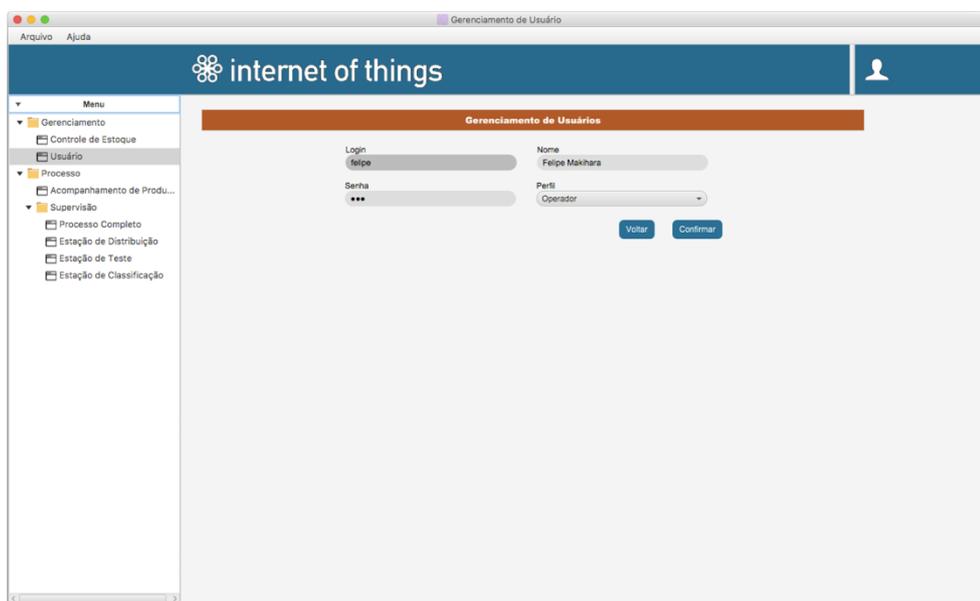
Fonte: Autor

Figura 50 – Sistema Supervisório - Tela de Visualização do Gerenciamento de Usuários



Fonte: Autor

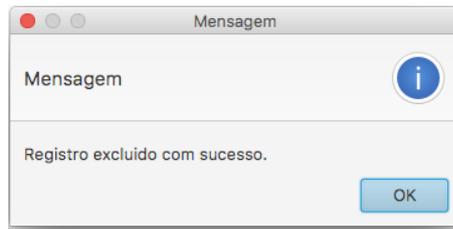
Figura 51 – Sistema Supervisório - Tela de Alteração do Gerenciamento de Usuários



Fonte: Autor

Por fim, foram feitas janelas de aviso para quando um registro for excluído e quando for pesquisado por critérios e o registro não existir, como mostram as figuras 52 e 53, respectivamente.

Figura 52 – Janela de Aviso de Exclusão de Registro



Fonte: Autor

Figura 53 – Janela de Aviso de Registro não encontrado

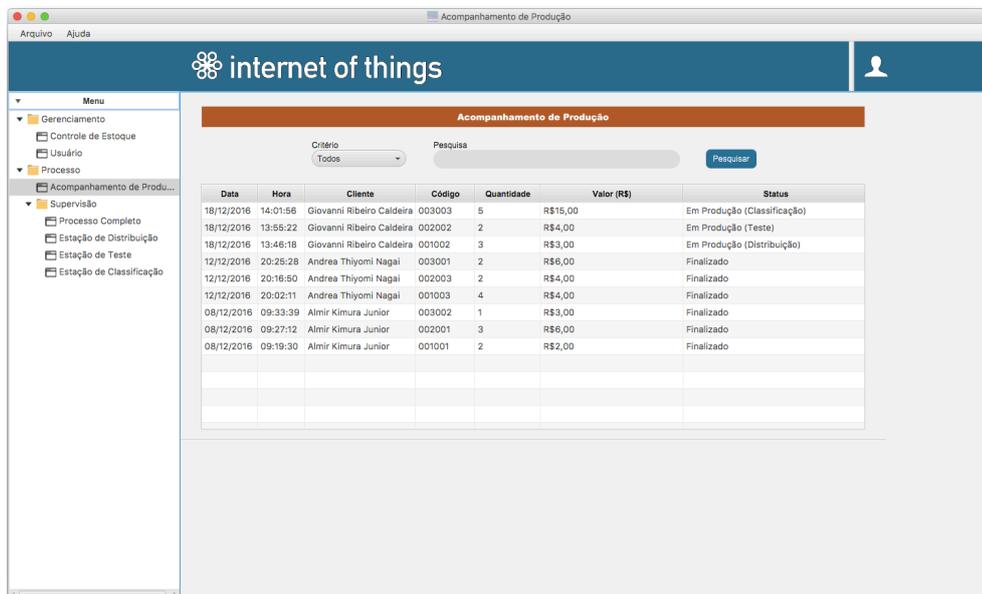


Fonte: Autor

#### 4.3.5 Acompanhamento de Produção

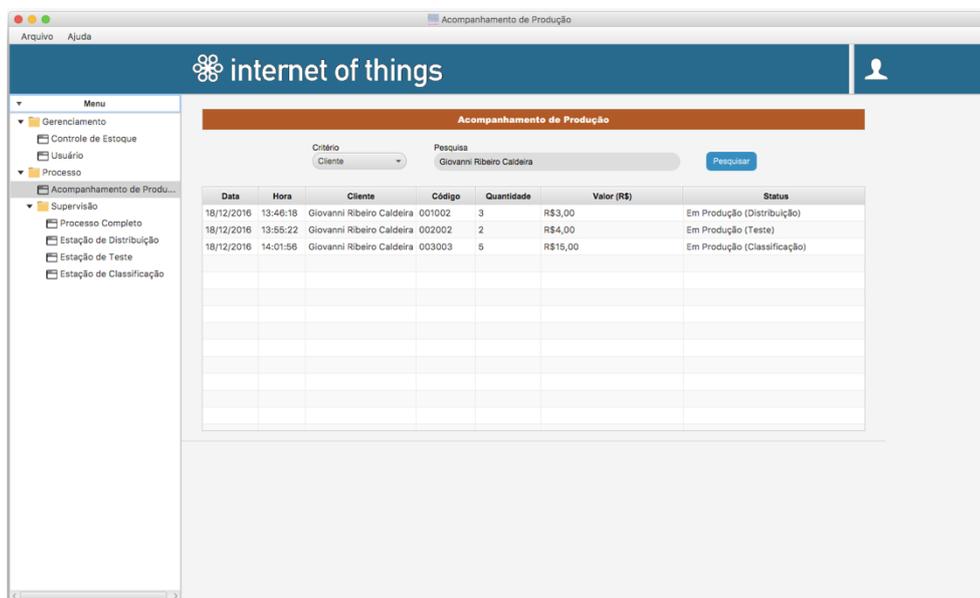
Na tela de Acompanhamento de Produção é possível visualizar todas as informações relacionadas à produção de um determinado pedido, assim como distribuído na tabela com as colunas “Data”, “Hora”, “Cliente”, “Código”, “Quantidade”, “Valor” e “Status”, possibilitando a pesquisa pelos critérios: “Todos”, “Cliente”, “Código” e “Status”, assim como mostram as figuras 54, 55, 56 e 57, respectivamente.

Figura 54 – Sistema Supervisório - Tela de Acompanhamento de Produção - Pesquisa por Todos



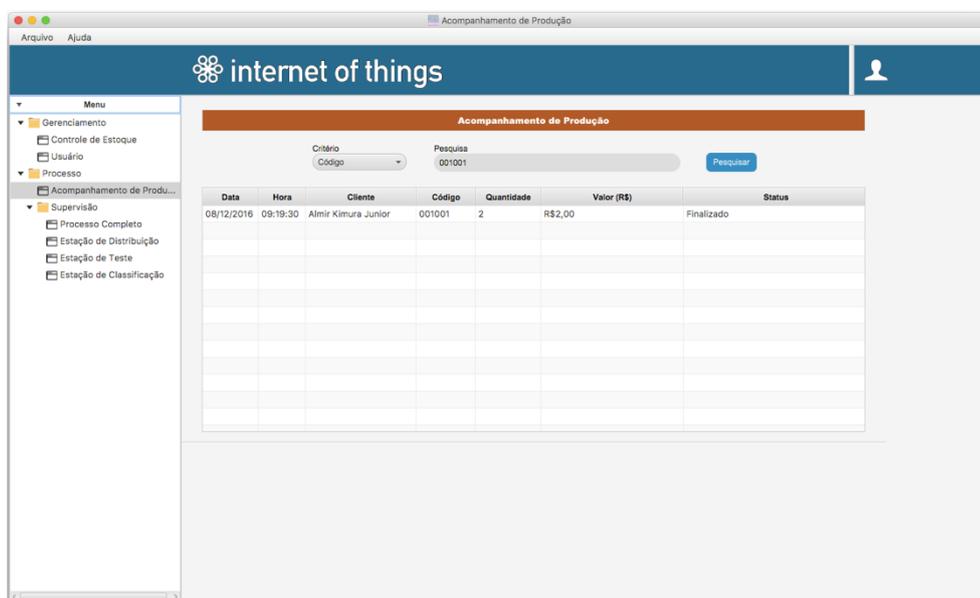
Fonte: Autor

Figura 55 – Sistema Supervisório - Tela de Acompanhamento de Produção - Pesquisa por Cliente



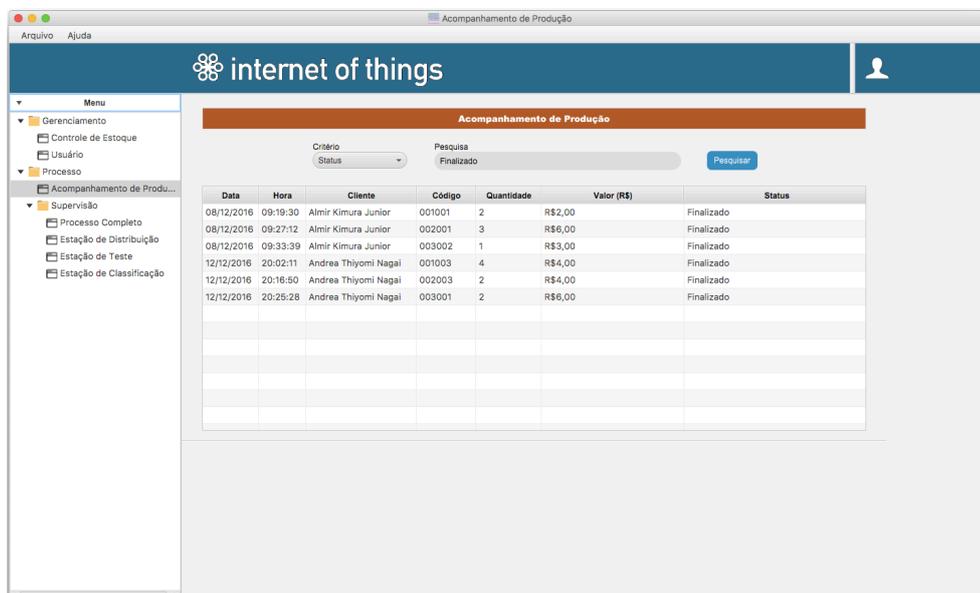
Fonte: Autor

Figura 56 – Sistema Supervisório - Tela de Acompanhamento de Produção - Pesquisa por Código



Fonte: Autor

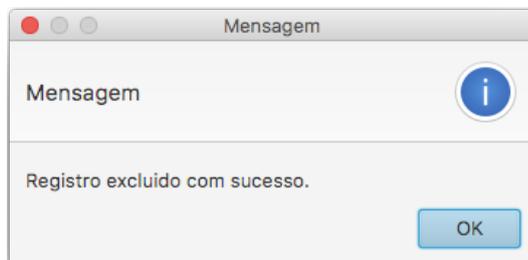
Figura 57 – Sistema Supervisório - Tela de Acompanhamento de Produção - Pesquisa por Status



Fonte: Autor

Foi feita uma janela de aviso para quando for pesquisar por critérios e o registro não existe como mostra a figura 58.

Figura 58 – Janela de Aviso de Exclusão de Registro



Fonte: Autor

#### 4.3.6 Supervisão do Processo

A Supervisão de Processo foi dividida em “Processo Completo”, “Estação de Distribuição”, “Estação de Testes” e “Estação de Classificação” como especificado na subseção 3.6.1.4.

##### 4.3.6.1 Comunicação entre o Sistema Supervisório e o MPS®

Para realizar a comunicação entre o Sistema Supervisório e o Sistema Modular de Produção, foi desenvolvido um código em Java utilizando o protocolo Modbus IP que se encontra no apêndice C.

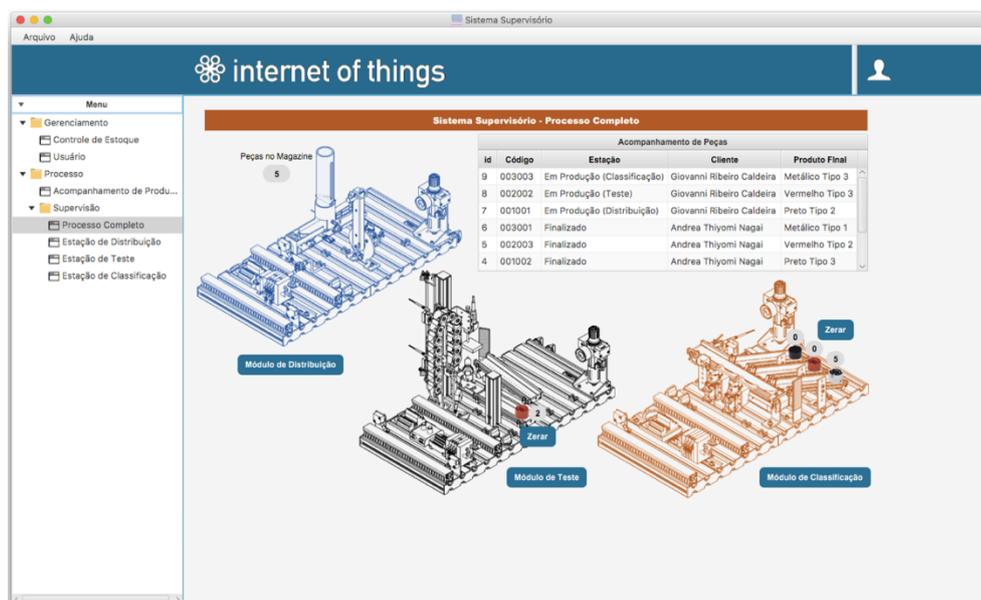
Este serve para, através do cabo *Ethernet*, recolher informações dos sensores e atuadores controlados pelo CLP e mostrar nas telas de Supervisão do Processo se os mesmos estão acionados ou não e em que etapa se encontra a produção da peça.

#### 4.3.6.2 Processo Completo

Na tela do Processo Completo é possível visualizar o modelo das 3 (três) estações em 3D lado a lado assim como a quantidade de peças no magazine inicial na Estação de Distribuição, quantas foram rejeitadas na Estação de Testes e quantas peças se encontram em cada saída da Estação de Classificação. Também é possível visualizar uma tabela com as informações da produção das peças em tempo real com as colunas “id”, “Código”, “Estação”, “Cliente” e “Produto Final”.

A figura 59 mostra a tela do Processo Completo.

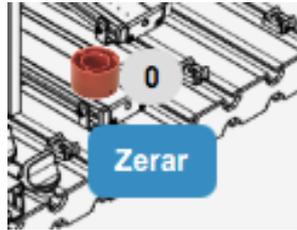
Figura 59 – Sistema Supervisório - Tela do Processo Completo



Fonte: Autor

Foram adicionados 5 (cinco) botões à tela, o botão “Módulo de Distribuição” acessa a tela da Estação de Distribuição, o botão “Módulo de Teste” acessa a tela da Estação de Teste, o botão “Módulo de Classificação” acessa a tela da Estação de Classificação, o botão Zerar que se encontra no modelo da Estação de Testes, serve para zerar a contagem de peças rejeitadas como mostra a figura 60 e o botão zerar que se encontra no modelo da Estação de Classificação serve para zerar a contagem das peças separadas nas saídas quando finalizadas como mostra na figura 61.

Figura 60 – Botão de Zerar as Peças Rejeitadas



Fonte: Autor

Figura 61 – Botão de Zerar as Peças Classificadas

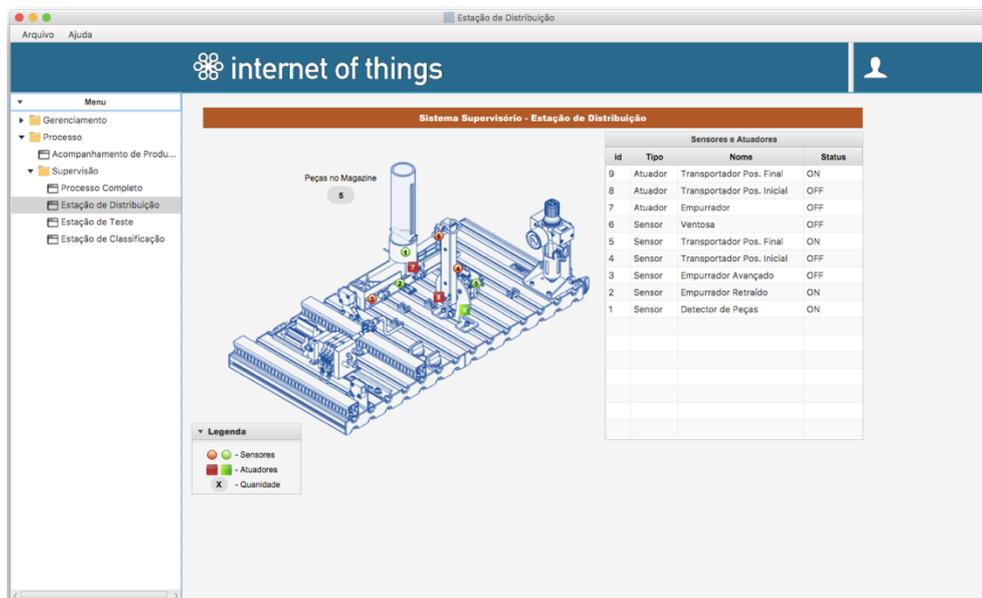


Fonte: Autor

#### 4.3.6.3 Estação de Distribuição

A tela da Estação de Distribuição possibilita visualizar o modelo em 3D da mesma mostrando em círculos os sensores e em quadrados os atuadores e na cor verde caso estejam acionados e na cor vermelha se não estiverem acionados e também é possível visualizar estas informações na tabela com as colunas "id", "Tipo", "Nome" e "Status", também é possível visualizar quantas peças se encontram no magazine inicial assim como são mostrados na figura 62.

Figura 62 – Sistema Supervisório - Tela da Estação de Distribuição

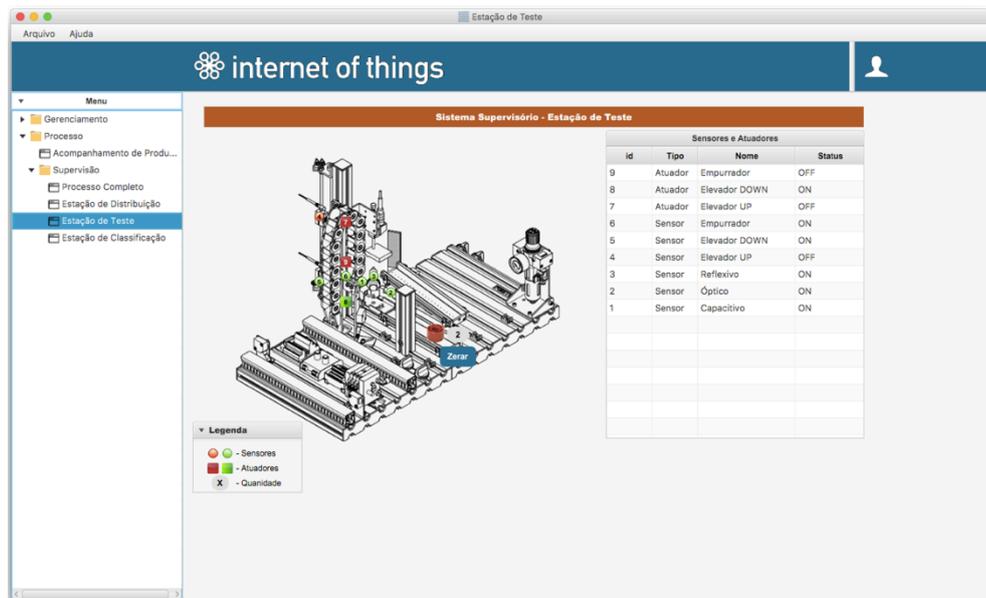


Fonte: Autor

#### 4.3.6.4 Estação de Teste

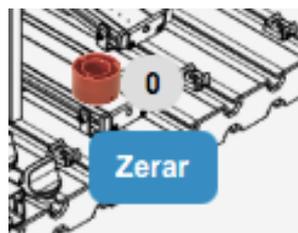
A tela da Estação de Testes possui as mesmas funcionalidades da Estação de Distribuição, porém com as informações da primeira, assim como seu modelo 3D representados na figura 63 com a adição do número de peças rejeitadas e com a funcionalidade do botão “Zerar” como mostra a figura 64.

Figura 63 – Sistema Supervisório - Tela da Estação de Teste



Fonte: Autor

Figura 64 – Botão de Zerar as Peças Rejeitadas

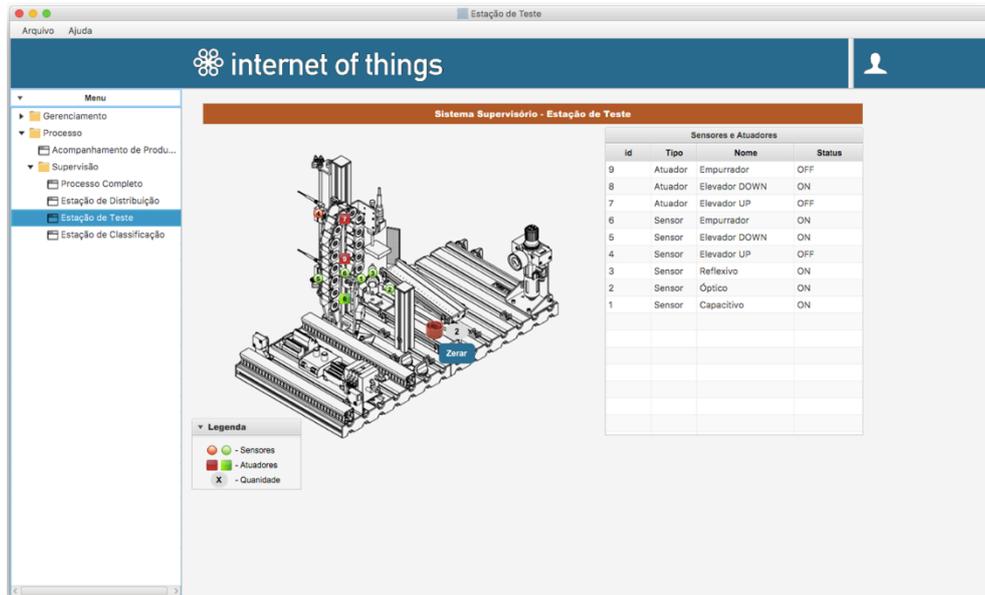


Fonte: Autor

#### 4.3.6.5 Estação de Classificação

A tela da Estação de Classificação possui as mesmas funcionalidades da Estação de Distribuição e de Teste, porém com as informações da primeira, assim como seu modelo 3D representados na figura 65 com a adição do número de peças em cada saída da estação e com a funcionalidade do botão “Zerar” como mostra a figura 66.

Figura 65 – Sistema Supervisório - Tela da Estação de Classificação



Fonte: Autor

Figura 66 – Botão de Zerar as Peças Classificadas



Fonte: Autor

## 5 CONCLUSÕES

O projeto e implantação de um sistema supervisório capaz de tomar a decisão de início de produção a partir da informação de demanda de produto obtida de uma página Web, seguindo com a transferência destes dados para os chips RFID da matéria prima, que carregam os dados de como o produto deve ser produzido, para a leitura de cada posto de trabalho subsequente até a finalização da fabricação, se caracteriza como um exemplo de aplicação no ramo industrial do principal conceito trazido pelos ideais da quarta revolução industrial, o conceito de Internet das Coisas.

Para integrar todas as estações do processo juntamente com os microcontroladores ao sistema supervisório, estabeleceu-se uma conexão Ethernet entre os dispositivos utilizando um switch. Para a comunicação entre o sistema supervisório e os microcontroladores, foi utilizado um padrão de comunicação chamado SPI, definindo os Raspberrys Pi como “servidores” de informação e o sistema supervisório como o “cliente”. A comunicação entre estes dois dispositivos obteve resultado satisfatório, pois por meio desta arquitetura é possível enviar e receber informações de ambos os lados. Para a comunicação entre os CLPs e o sistema supervisório foi desenvolvido um programa em Java utilizando o protocolo Modbus IP. Por meio deste, é possível definir todas as variáveis a serem monitoradas e manipuladas, e estas serão constantemente modificadas de acordo com o funcionamento do sistema, refletindo o estado real da variável no processo.

Foi de suma importância seguir a sequência correta para o desenvolvimento do sistema supervisório, desde o levantamento dos requisitos funcionais, seguindo então para a modelagem do banco de dados, finalizando com a programação em si do software, pois ao seguir a sequência, a necessidade de retrabalho devido a erros é diminuída consideravelmente. Além disso, o resultado obtido alcançou o objetivo do desenvolvimento de um sistema supervisório capaz de tanto prover a visibilidade de todas as variáveis importantes do processo quanto tomar decisões importantes como o início de produção e compra de materiais de forma automática.

### 5.1 TRABALHOS FUTUROS

Este projeto trouxe um grande diferencial ao implementar um novo conceito da quarta revolução industrial, porém não foi possível demonstrar todo o potencial da

indústria 4.0 apesar dos diversos recursos apresentados nessa monografia devido às limitações da pesquisa, o que possibilita as seguintes implementações futuras:

- Implantar um conjunto de dispositivos que permita que as peças automaticamente sejam transportadas do rejeito da estação de teste para o início da estação de distribuição;
- Agregar outros equipamentos eletromecânicos como, braços robóticos ou esteiras, ou até outras estações de sistema modular de produção aos módulos utilizados na pesquisa situados no laboratório de automação da EST-UEA;
- Hospedar o site desenvolvido na pesquisa em um servidor online, permitindo o acesso de qualquer dispositivo a longas distâncias;
- Substituir o computador com o sistema supervisorio por um servidor conectado à Internet, permitindo o monitoramento do processo não tenha a necessidade de ser in loco.
- Desenvolver um aplicativo para celular ou tablet com as mesmas funções do sistema supervisorio.

## REFERÊNCIAS BIBLIOGRÁFICAS

3S - SMART SOFTWARE SOLUTIONS GMBH. **User Manual fo PLC Programming with Codesys 2.3**. Kempten, 2010.

3S - SMART SOFTWARE SOLUTIONS GMBH. CODESYS OPC Server - **Standard Access to the Variables of a CODESYS PLC**. Kempten, 2015. Disponível em: <<https://www.codesys.com/products/codesys-runtime/opc-server.html>>.

ALMEIDA, M. L. d. **Elementos finitos paramétricos implementados em java**. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, 2005. Disponível em: <<http://www.bibliotecadigital.ufmg.br/dspace/handle/1843/BUD-B-8C7LMP>>.

ASHTON, K. That “Internet of Things” Thing. **RFID Journal**, Orlando, 2009. Disponível em: <<http://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf>>.

ATZORI, L.; IERA, A.; MORABITO, G.. The Internet of Things: A survey. **Elsevier Journal**. Amsterdam, 2010. Disponível em <<http://cs.uwaterloo.ca/~brecht/courses/854-Emerging-2014/readings/iot/iot-survey.pdf>>.

BASTOS, A. V. **Microcontroladores**. 2010. Disponível em: <[http://www.decom.ufop.br/alex/arquivos/sof\\_bas\\_ECA/Microcontroladores.pdf](http://www.decom.ufop.br/alex/arquivos/sof_bas_ECA/Microcontroladores.pdf)>.

BELAN, H. C. **Formalização da rede de petri canal/agência para projeto de equipamentos industriais**. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, 2007. Disponível em: <<https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/89820/241204.pdf?sequence=1&isAllowed=y>>.

BOARETTO, N. **Sistemas Supervisórios**. Joinville: IFSC, 2008.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: **Guia do Usuário**. Rio de Janeiro: Elsevier Brasil, 2006.

BOUDREAU, T. et al. **NetBeans: the definitive guide**. Sebastopol: O’Reilly Media, Inc., 2011.

BRANDÃO, J. A. R.; **Linguagem Ladder para Basic Step M8 e M16**. [Editorial]. Mecatrônica Fácil, v. 6, n. 35, p. xxxv, jul./ago., 2007.

BRETTEL, M. et al. **How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0 perspective**. International Journal of Science, Engineering and Technology, v. 44, 2014. 18

CAELUM. **Java e Orientações a Objetos**. São Paulo, 2003. Disponível em: <<http://www.caelum.com.br/apostila-java-orientacao-objetos.html>>.

CORREIA, J.; et al. **Sistemas de Recuperação de Informação**. Barcarena: Universidade Atlântica, 2011.

DEITEL, H. M.; DEITEL, P. J. **Java: Como Programar**. 4. ed. Porto Alegre: Bookman, 2003.

FABIO, I. S. **SISTEMA SCADA PARA CÁLCULO DE INDICADORES DE DESEMPENHO EMPREGANDO FERRAMENTAS DE CONFIABILIDADE EM UM SISTEMA MODULAR DE PRODUÇÃO**. Universidade do Estado do Amazonas, Manaus, 2015.

FESTO DIDADIC. **The stations in the Modular Production System at a glance**. Denkendorf, 2006.

FESTO DIDADIC. **Distributing Station Manual**. Denkendorf, 2006.

FESTO DIDADIC. **Sorting Station Manual**. Denkendorf, 2006.

FESTO DIDADIC. **Testing Station Manual**. Denkendorf, 2006.

FILHO, B. S. d. S. **Curso de controladores lógicos programáveis**. LEE-UERJ, Rio de Janeiro, RJ, 1998.

FONSECA, M. de O. **Comunicação opc - uma abordagem prática**. In: IV Seminário de Automação de Processos, Associação Brasileira de Metalurgia e Materiais. Vitória: [s.n.], 2002. Disponível em: <<http://www.cpdee.ufmg.br/~seixas/PaginaSDA/Download/DownloadFiles/OPCMarcosFonseca.PDF>>.

FORESTI, H. B. **Microcontroladores**. 2013. Disponível em: <[http://robolivre.org/cont\\_eudo/microcontroladores](http://robolivre.org/cont_eudo/microcontroladores)>.

FURQUIM, T. de A. **Fatores motivadores de uso de site web: um estudo de caso**. *Ci. Inf*, SciELO Brasil, v. 33, n. 1, 2004

GERMANY TRADE AND INVESTMENT. **Integrated Industry**. Berlim, 2014. Disponível em: <<http://www.gtai.de/GTAI/Navigation/EN/Invest/Service/Publications/markets-germany-,did=993284.html>>.

GRAHAM, I. S. **HTML 4.0 Sourcebook**. [S.l.]: Wiley, 1998.

GRUPO IBRACE-ICBr. **Brasil ocupa 17º lugar entre os países mais preparados a usar IoT na indústria**. Campinas, 2015. Disponível em: <<http://www.grupoibrace-icbr.org.br/brasil-ocupa-17o-lugar-entre-os-paises-mais-preparados-usar-iot-na-industria.html>>.

KONDRASOVAS, I. **Sistemas Supervisórios e Protocolos de Comunicação Industrial**. Joinville: UDESC, 2013.

LECHETA, R. R., **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 2 ed. São Paulo: Novatec, 2010, 607 p.

LIMA, R. A. de. **Sistema Supervisório com Conceito de Internet das Coisas em Escala Laboratorial**. *Universidade do Estado do Amazonas*, Manaus, 2015.

MAFRA, M. A.; MEDEIROS, J. **Curso de Controladores Lógicos Programáveis**. Rio de Janeiro: UERJ, 2014. Disponível em: <<http://www.lee.eng.uerj.br/downloads/cursos/clp/clp.pdf>>.

MARÃO, G. A.. **Internet das Coisas: Oportunidades e Perspectivas da Nova Revolução Digital para o Brasil**. In: *SEMINÁRIO INTERNACIONAL BNDES*, Rio de Janeiro: [s.n.], 2014. p. 14 .Disponível em: <[http://www.bndes.gov.br/SiteBNDES/export/sites/default/bndes\\_pt/Galerias/Arquivos/conhecimento/seminario/seminario\\_internet\\_das\\_coisas/Gabriel\\_Marao.pdf](http://www.bndes.gov.br/SiteBNDES/export/sites/default/bndes_pt/Galerias/Arquivos/conhecimento/seminario/seminario_internet_das_coisas/Gabriel_Marao.pdf)>.

MOTT, A. **O que são Sistemas Supervisórios?**. São Paulo, 2012. Disponível em: <<http://www.automacaoindustrial.info/o-que-sao-sistemas-supervisorios.html>>.

PAIOLA, C. E. G. **O papel do supervisório no atual contexto tecnológico**. São Paulo, 2014. Disponível em: <[http://www.aquarius.com.br/Boletim/InTech132\\_artigo.pdf](http://www.aquarius.com.br/Boletim/InTech132_artigo.pdf)>.

PAIOLA, C. E. G. **O papel do supervisório no atual contexto tecnológico**. InTech, v. 132, p. 8, 2012.

PAIVA, A. O.; MOREIRA, R. O. **Raspberry pi: dispositivo de 35 dólares para visualização de imagens dicom**. *Radiol Bras*, SciELO Brasil, 2014.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de gerenciamento de banco de dados**. 3. ed. Porto Alegre: McGraw Hill Brasil, 2008.

RASPBERRY PI FOUNDATION. **Raspberry Pi**. Cambridgeshire, 2013. Disponível em: <<http://www.sheffieldlearningcommunity.com/sites/default/files/uploads/Raspberry%20Pi.pdf#page=2>>.

RFID CENTER OF EXCELLENCE. **O que é RFID?**. Sorocaba, 2013. Disponível em: <[http://www.rfid-coe.com.br/\\_Portugues/OqueERFID.aspx](http://www.rfid-coe.com.br/_Portugues/OqueERFID.aspx)>.

ROUSE, M. **RFID Tagging**. Newton, MA, 2010. Disponível em: <<http://whatis.techtarget.com/-/definition/RFID-tagging.html>>.

SILVA, L. A. **Apostila de Android – Programando Passo a Passo**. Rio de Janeiro, 2015. Disponível em: <<http://othonbatista.com/arquivos/android/apostila-android.pdf>>.

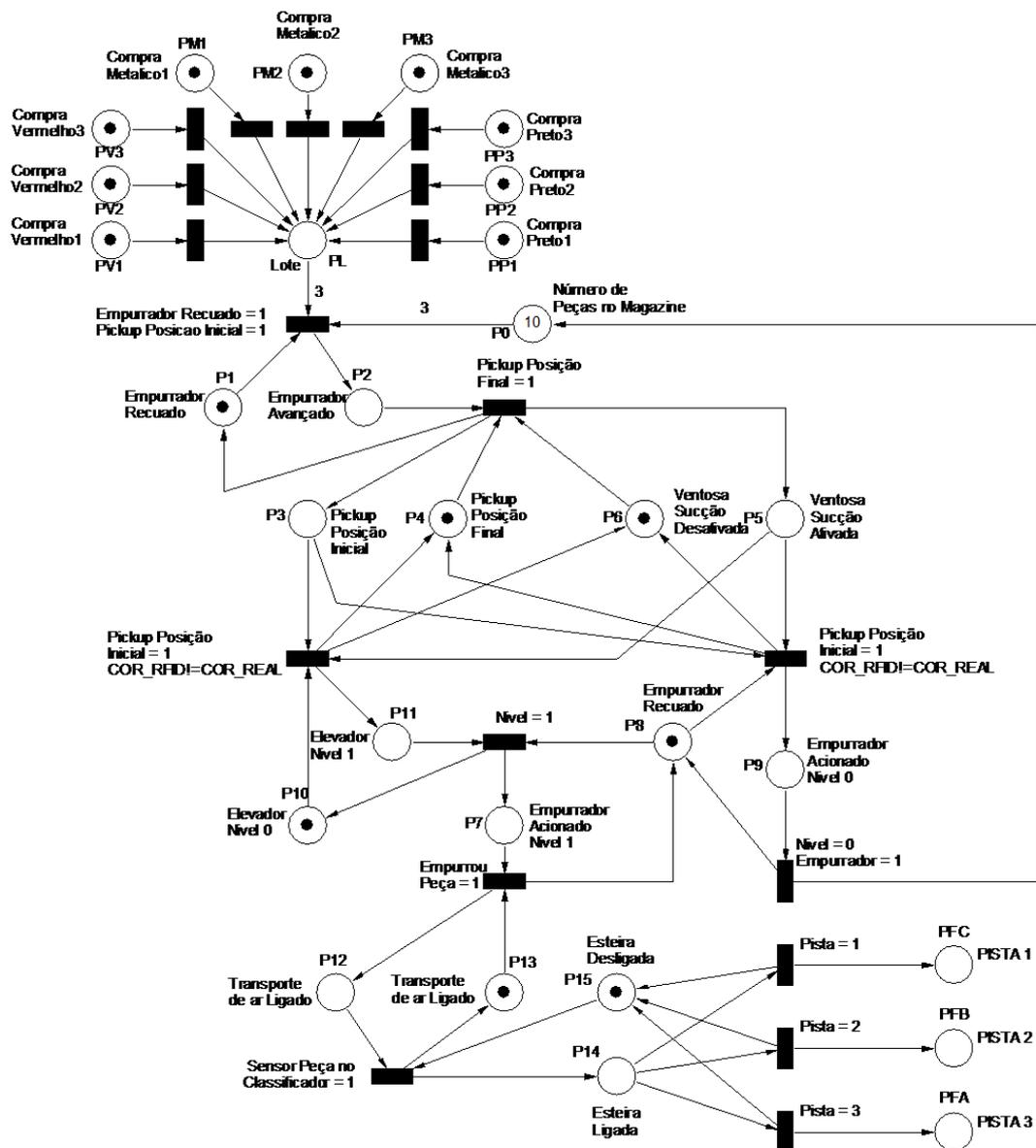
SILVA, M. S. **Criando sites com HTML: sites de alta qualidade com HTML e CSS**. Novatec Editora, 2008.

SOUTO, M. A. M.. et al. **Ferramentas de suporte a monitoração do aluno em um ambiente inteligente de ensino na web**. In: *Anais do VII Workshop de Informática na Escola, Fortaleza, Brasil*. [s.n.], 2001. Disponível em: <[http://www.researchgate.net/profile/Rosa\\_Maria\\_Vicari/publication/267822393\\_Ferramentas\\_de\\_Suporte\\_a\\_Monitorao\\_do\\_Aluno\\_em\\_um\\_Ambiente\\_Inteligente\\_de\\_Ensino\\_na\\_Web/links/5461f6540cf2c1a63c017dc9.pdf](http://www.researchgate.net/profile/Rosa_Maria_Vicari/publication/267822393_Ferramentas_de_Suporte_a_Monitorao_do_Aluno_em_um_Ambiente_Inteligente_de_Ensino_na_Web/links/5461f6540cf2c1a63c017dc9.pdf)>.

VERMESAN, Ovidiu; FRIESS, Peter. **Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems**. Aalborg: River, 2013. Disponível em: <[http://www.internetofthingsresearch.eu/pdf/Converging\\_Technologies\\_for\\_Smart\\_Environments\\_and\\_Integrated\\_Ecosystems\\_IERC\\_Book\\_Open\\_Access\\_2013.pdf](http://www.internetofthingsresearch.eu/pdf/Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems_IERC_Book_Open_Access_2013.pdf)>.

## APÊNDICE A – MODELAGEM EM REDE DE PETRI

Figura 67 – Modelagem em Rede de Petri do Sistema



Fonte: (LIMA, 2015)

## APÊNDICE B – PROGRAMAÇÃO DO MICROCONTROLADOR RASPBERRY PI

### B.1 Escrever informação nas Tags *RFID*

Figura 68 – Código para escrever informações nas Tags RFID parte 1

```
#!/usr/bin/env python
# -*- coding: utf8 -*-

import RPi.GPIO as GPIO
import MFRC522
import signal

continue_reading = True

# Capture SIGINT for cleanup when the script is aborted
def end_read(signal,frame):
    global continue_reading
    print "Ctrl+C captured, ending read."
    continue_reading = False
    GPIO.cleanup()

# Hook the SIGINT
signal.signal(signal.SIGINT, end_read)

# Create an object of the class MFRC522
MIFAREReader = MFRC522.MFRC522()

# This loop keeps checking for chips. If one is near it will get the UID and authenticate
while continue_reading:

    print "Aproxime a TAG"

    # Scan for cards
    (status,TagType) = MIFAREReader.MFRC522_Request(MIFAREReader.PICC_REQIDL)

    # If a card is found
    if status == MIFAREReader.MI_OK:
        print "Tag detectada"
        j = input("Qual o modelo?\n")

    # Get the UID of the card
    (status,uid) = MIFAREReader.MFRC522_Anticoll()

    # If we have the UID, continue
    if status == MIFAREReader.MI_OK:

        # Print UID
        print "Card read UID: "+str(uid[0])+", "+str(uid[1])+", "+str(uid[2])+", "+str(uid[3])

        # This is the default key for authentication
        key = [0xFF,0xFF,0xFF,0xFF,0xFF,0xFF]

        # Select the scanned tag
        MIFAREReader.MFRC522_SelectTag(uid)

        # Authenticate
        status = MIFAREReader.MFRC522_Auth(MIFAREReader.PICC_AUTHENT1A, 8, key, uid)
        print "\n"

        # Check if authenticated
        if status == MIFAREReader.MI_OK:

            # Variable for the data to write
            data = []
```

Fonte: Autor

Figura 69 – Código para escrever informações nas Tags RFID parte 2

```
# Fill the data with 0xFF
for x in range(0,16):
    if j == 1:
        data.append(0x01)
    elif j == 2:
        data.append(0x02)
    elif j == 3:
        data.append(0x03)
    elif j == 4:
        data.append(0x04)
    elif j == 5:
        data.append(0x05)
    elif j == 6:
        data.append(0x06)
    elif j == 7:
        data.append(0x07)
    elif j == 8:
        data.append(0x08)
    elif j == 9:
        data.append(0x09)
    else:
        data.append(0x00)

print "Sector 8 looked like this:"
# Read block 8
MIFAREReader.MFRC522_Read(8)
print "\n"

print "Sector 8 will now be filled with 0xFF:"
# Write the data
MIFAREReader.MFRC522_Write(8, data)
print "\n"

print "It now looks like this:"
# Check to see if it was written
MIFAREReader.MFRC522_Read(8)
print "\n"

data = []

# Stop
MIFAREReader.MFRC522_StopCrypto1()

# Make sure to stop reading for cards
#continue_reading = False
else:
    print "Authentication error"
```

Fonte: Autor

## B.2 Ler informação nas Tags *RFID*

Figura 70 – Código para ler informações das Tags RFID

```
#!/usr/bin/env python
# -*- coding: utf8 -*-

import RPi.GPIO as GPIO
import MFRC522
import signal

continue_reading = True

# Capture SIGINT for cleanup when the script is aborted
def end_read(signal, frame):
    global continue_reading
    print "Ctrl+C captured, ending read."
    continue_reading = False
    GPIO.cleanup()

# Hook the SIGINT
signal.signal(signal.SIGINT, end_read)

# Create an object of the class MFRC522
MIFAREReader = MFRC522.MFRC522()

# Welcome message
print "Welcome to the MFRC522 data read example"
print "Press Ctrl-C to stop."

# This loop keeps checking for chips. If one is near it will get the UID and authenticate
while continue_reading:

    # Scan for cards
    (status, TagType) = MIFAREReader.MFRC522_Request(MIFAREReader.PICC_REQIDL)

    # If a card is found
    if status == MIFAREReader.MI_OK:
        print "Card detected"

    # Get the UID of the card
    (status, uid) = MIFAREReader.MFRC522_Anticoll()

    # If we have the UID, continue
    if status == MIFAREReader.MI_OK:

        # Print UID
        print "Card read UID: "+str(uid[0])+", "+str(uid[1])+", "+str(uid[2])+", "+str(uid[3])

        # This is the default key for authentication
        key = [0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF]

        # Select the scanned tag
        MIFAREReader.MFRC522_SelectTag(uid)

        # Authenticate
        status = MIFAREReader.MFRC522_Auth(MIFAREReader.PICC_AUTHENT1A, 8, key, uid)

        # Check if authenticated
        if status == MIFAREReader.MI_OK:
            MIFAREReader.MFRC522_Read(8)
            MIFAREReader.MFRC522_StopCrypto1()
        else:
            print "Authentication error"
```

Fonte: Autor

## APÊNDICE C – PROGRAMAÇÃO PARA COMUNICAR O SISTEMA SUPERVISÓRIO COM O SISTEMA MODULAR DE PRODUÇÃO

Figura 71 – Código para comunicar o Sistema Supervisório com o MPS®

```
package app.modbus.ip;

import com.serotonin.modbus4j.ModbusFactory;
import com.serotonin.modbus4j.ModbusMaster;
import com.serotonin.modbus4j.exception.ErrorResponseException;
import com.serotonin.modbus4j.exception.ModbusInitException;
import com.serotonin.modbus4j.exception.ModbusTransportException;
import com.serotonin.modbus4j.ip.IpParameters;
import java.io.*;

public class AppModbusIP {

    public static void main(String args[]) throws IOException, InterruptedException, ErrorResponseException, ModbusTransportException {
        ModbusFactory factory = new ModbusFactory();
        IpParameters params = new IpParameters();
        String pasta = "/User/felipemakihara/NetBeansProjects/MODBUS_CODES/Modbus_IP/configuracao.ini";
        File config = new File(pasta);
        if (!config.exists()) {
            System.out.println("Não Existe");
            return;
        }
        String parametros[] = Arquivo.leitor(pasta);
        params.setHost(parametros[0]);
        params.setPort(Integer.parseInt(parametros[1]));
        params.setEncapsulated(false);
        int slaveID = Integer.parseInt(parametros[2]);
        int tempo = Integer.parseInt(parametros[6]);
        ModbusMaster master = factory.createTcpMaster(params, false);
        master.setTimeout(2000);
        master.setRetries(1);
        Data timestamp = new Data();
        String pathSensorDist = parametros[3];
        String pathAtuadorDist = parametros[4];
        String pathSensorTeste = parametros[3];
        String pathAtuadorTeste = parametros[4];
        String pathSensorClass = parametros[3];
        String pathAtuadorClass = parametros[4];
        boolean erro;
        do {
            do {
                try {
                    master.init();
                    erro = false;
                } catch (ModbusInitException e) {
                    System.out.println("Erro ao iniciar o Mestre: " + e.getMessage());
                    erro = true;
                }
            } while (erro);

            CLP distribuicao = new CLP(master, slaveID);
            Arquivo.escriptor(pathSensorDist, timestamp.dataFormatada() + "," + distribuicao.getSensor());
            Arquivo.escriptor(pathAtuadorDist, timestamp.dataFormatada() + "," + distribuicao.getAtuador());

            CLP teste = new CLP(master, slaveID);
            Arquivo.escriptor(pathSensorTeste, timestamp.dataFormatada() + "," + teste.getSensor());
            Arquivo.escriptor(pathAtuadorTeste, timestamp.dataFormatada() + "," + teste.getAtuador());

            CLP classificacao = new CLP(master, slaveID);
            Arquivo.escriptor(pathSensorClass, timestamp.dataFormatada() + "," + classificacao.getSensor());
            Arquivo.escriptor(pathAtuadorClass, timestamp.dataFormatada() + "," + classificacao.getAtuador());

            master.destroy();
            Thread.sleep(1000 * tempo);
        } while (true);
    }
}
```

Fonte: Autor