

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**JOÃO ALVAREZ PEIXOTO**

**SISTEMA MINIMAMENTE INVASIVO BASEADO EM  
AGENTES APLICADO EM CONTROLADORES LÓGICOS  
PROGRAMÁVEIS**

Porto Alegre

2016

**JOÃO ALVAREZ PEIXOTO**

**SISTEMA MINIMAMENTE INVASIVO BASEADO EM  
AGENTES APLICADO EM CONTROLADORES LÓGICOS  
PROGRAMÁVEIS**

Tese de doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica.  
Área de concentração: Controle e Automação

ORIENTADOR: Carlos Eduardo Pereira

COORIENTADOR: José Antônio Barata de Oliveira

Porto Alegre

2016

**JOÃO ALVAREZ PEIXOTO**

**SISTEMA MINIMAMENTE INVASIVO BASEADO EM  
AGENTES APLICADO EM CONTROLADORES LÓGICOS  
PROGRAMÁVEIS.**

Esta tese foi julgada adequada para a obtenção do título de Doutor em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. Dr. Carlos Eduardo Pereira, UFRGS

Doutor pela Stuttgart University - Stuttgart, Alemanha

Co-orientador: \_\_\_\_\_

Prof. Dr. José Antônio Barata de Oliveira, UNL

Doutor pela Universidade Nova de Lisboa - Lisboa, Portugal

**BANCA EXAMINADORA:**

Prof. Dr. Rafael Heitor Bordini - Faculdade e Informática - PUC-RS (Relator)  
Doutor pela University College London, Inglaterra

Prof. Dr. Marcelo Ricardo Stemmer - Departamento de Automação e Sistemas - UFSC  
Doutor pelo Instituto WZL da RheinischWestfälische Technische Hochschule-Aachen, Alemanha

Prof. Dr. João Cesar Netto - PPGC - UFRGS  
Doutor pela Université Catholique de Louvain, França

Prof. Dr. Walter Fetter Lages - DELET-UFRGS  
Doutor pela Instituto Tecnológico de Aeronáutica – ITA – Brasil

Prof. Dr. Edison Pignaton de Freitas - PPGEE - UFRGS  
Doutor pela Universidade Federal do Rio Grande do Sul e a Universidade de Halmstad, Brasil e Suécia

Coordenador do PPGEE: \_\_\_\_\_

Prof. Dr. Valner João Brusamarello, UFRGS

Doutor pela Universidade Federal de Santa Catarina, Brasil

Porto Alegre, dezembro de 2016

## **DEDICATÓRIA**

Dedico este trabalho a todos aqueles que não medem esforços para estudar, aprender e tentar fazer um mundo melhor. Àqueles que não são egoístas a ponto de guardar o conhecimento e descoberta para si, e que são humanos a ponto de aprender para auxiliar os outros.

A minha família, minha esposa Solange e meus filhos, Taise e João, que respeitaram minha escolha, com todas as privações e toda ausência que provoquei, e que entenderam e aceitaram minha vontade de aprender mais. Que possam seguir este caminho e querer mudar o mundo, pelo menos o mundo a sua volta.

## **AGRADECIMENTOS**

Ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela oportunidade de realização de trabalhos em minha área de pesquisa.

Aos colegas do PPGEE, pelo auxílio nas tarefas desenvolvidas durante o curso e apoio na revisão deste trabalho.

Aos colegas Armando Rial Michalski e Gustavo Enedir Hennemann, que auxiliaram na montagem de cenários para aplicação do método, acreditando no conceito e se envolvendo na pesquisa, contribuindo com suas críticas, conhecimentos e descobertas e trazendo um toque técnico a uma tese acadêmica.

Ao Instituto SENAI de Inovação Soluções Integradas em Metalmeccânica, que aceitou e investiu em meu trabalho, minhas pesquisas, meu empenho e meus resultados.

Ao orientador, Carlos Eduardo Pereira, por acreditar no meu empenho e dedicação, e, com suas orientações técnicas, conduzir as atividades para que eu pudesse apresentar os melhores resultados.

Ao co-orientador, José Antônio Barata de Oliveira, que, com seu carisma e preocupação, me propiciou as melhores informações para dar um toque de qualidade ao trabalho.

A Deus, por ter me concedido o dom de aprender.

## RESUMO

A indústria carece de sistemas produtivos que atendam a diversidade de demanda do mercado, com eficiência e capacidade de adaptação rápida. As novas soluções para estas necessidades remetem ao uso de novas tecnologias de automação. Porém, a maioria dos sistemas integrados de manufatura implantados em indústria possui seu gerenciamento por controladores lógicos programáveis, assim como suas interligações elétricas e lógicas de controle. A tese proposta faz uso de Sistemas Multiagentes em controladores lógicos programáveis, para que estes possam controlar seus sistemas de manufatura, propiciando-lhes os requisitos de diversidade, agilidade e auto-organização no meio produtivo. A comprovação desta tese se dá pelo método desenvolvido para que tal implementação aconteça, bem como pela análise de abrangência deste método, a fim de verificar em quais cenários ele é eficaz. O método proposto analisa o controlador lógico programável e descreve uma rotina de ações a serem seguidas para que ele se insira em Sistemas Multiagentes, valendo-se das funcionalidades que um agente pode oferecer. Sua análise de aplicação ocorreu em três cenários distintos, onde a abordagem por sistemas auto-organizáveis apresentou melhores resultados para atingir os requisitos de diversidade, agilidade e auto-organização, a partir de sistemas multiagentes interagindo com os controladores lógicos programáveis e suas lógicas locais.

**Palavras-chave: Sistemas Multiagentes, manufatura integrada, auto-organizável, CLP.**

## **ABSTRACT**

The industry needs production systems that attend the diversity of market demand, with efficiency and fast adaptability. The new solutions to these needs refer to the use of new automation technologies. However, most integrated manufacturing systems deployed in industry have its management by programmable logic controllers as well as their electrical interconnections and logics of control. The thesis proposal indicates the use of multi-agent systems on programmable logic controllers, so that they are able to control their manufacturing systems, addressing the requirements of diversity, self-organizing and agility to the productive environment. The proof of this thesis is given by the method developed for such implementation to happen, as well as analysis of the scope of this method in order to verify in what scenarios it is effective. The proposed method analyzes the programmable logic controller and describes a routine of actions to follow so that it may enter into a multi-agent system, using the features that an agent can offer. Its application analysis took place in three different scenarios where the approach by self-organizing systems showed better results for achieving the requirements of diversity, agility and self-organization from multi-agent systems interacting with programmable logic controllers and their local logic.

**Keywords: multi-agent systems, integrated manufacturing, self-adapting systems, PLC.**

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>14</b>
1.1 PROBLEMÁTICA.....	20
1.2 HIPÓTESE.....	23
1.3 OBJETIVOS.....	25
1.4 ESTRUTURA DO TRABALHO .....	26
<b>2. FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>27</b>
2.1 AGENTE EM SOFTWARE .....	27
2.2 SISTEMAS MULTIAGENTES E ARQUITETURA ORIENTADA A SERVIÇOS .....	30
2.3 PLATAFORMA DE AGENTES .....	35
2.3.1 A plataforma JADE.....	40
2.3.2 Agente em JADE.....	43
2.4 COMUNICAÇÃO ENTRE AGENTES .....	47
2.4.1 Protocolos especificados para troca de mensagens.....	51
2.4.2 Protocolo FIPA <i>Request</i> . .....	53
2.4.3 Protocolo FIPA <i>Contract Net</i> . .....	54
2.4.4 Serviço de páginas amarelas em JADE. ....	56
<b>3 PRINCIPAIS SISTEMAS DE GESTÃO DA MANUFATURA</b> .....	<b>60</b>
3.1 MANUFATURA INTEGRADA POR COMPUTADOR - CIM ( <i>COMPUTER INTEGRATED MANUFACTURING</i> ) .....	60
3.2 O PLANEJAMENTO DE RECURSOS COLABORATIVOS – (ERP - <i>ENTREPRISE RESOURCE PLANNING</i> ) .....	64
3.3 SISTEMA DE GERENCIAMENTO DA PRODUÇÃO (MES – <i>MANUFACTURING EXECUTION SYSTEMS</i> ) .....	69
3.4 SISTEMAS AUTO-ORGANIZÁVEIS COMO SISTEMA DE GESTÃO DA MANUFATURA .....	74
3.5 COMPARAÇÃO ENTRE OS SISTEMAS DE GESTÃO DA MANUFATURA .....	76
<b>4 DISCUSSÃO SOBRE PESQUISAS COM A UTILIZAÇÃO DE SISTEMAS MULTIAGENTES PARA ATENDER ÀS NECESSIDADES DE MANUFATURA INDUSTRIAL POR DIVERSIDADE, AGILIDADE E AUTO-ORGANIZAÇÃO. ...</b>	<b>80</b>
4.1 DIVERSIDADE NOS REQUISITOS DE PRODUÇÃO .....	80
4.2 AGILIDADE EM SISTEMAS PRODUTIVOS .....	94
4.3 AUTO-ORGANIZAÇÃO EM SISTEMAS PRODUTIVOS .....	105
4.4 ANÁLISE DOS RESULTADOS SOBRE AS ABORDAGENS.....	120
4.5 IDENTIFICAÇÃO DE LACUNAS NA ABORDAGEM DE SISTEMAS MULTIAGENTES PARA SOLUÇÃO DE PROBLEMAS DE DIVERSIDADE, AGILIDADE E AUTO-ORGANIZAÇÃO.....	126
<b>5 PROPOSTA DE TESE</b> .....	<b>129</b>
5.1 INEDITISMO DA PROPOSTA DA TESE.....	131
5.2 METODOLOGIA.....	132
5.3 DETALHAMENTO DA METODOLOGIA .....	148
5.4 SIMULAÇÕES E RESULTADOS .....	155
5.4.1 Cenário proposto para aplicação da metodologia. ....	155
5.4.2 Ensaio do método no cenário proposto.....	158
<b>6 ESTUDO DE CASO PARA COMPARAÇÃO DE ABORDAGENS</b> .....	<b>173</b>
6.1 MÉTRICAS A SEREM ANALISADAS .....	173
6.2 PLANTA COM GESTÃO POR MES .....	175
6.2.1 Cenário na gestão por MES.....	175
6.2.2 Avaliação na gestão por MES.....	182
6.3 PLANTA COM GESTÃO POR CIM.....	184
6.3.1 Cenário na gestão por CIM .....	184

<b>6.3.2 Avaliação na gestão por CIM .....</b>	<b>187</b>
<b>6.4 PLANTA COM GESTÃO POR SISTEMAS AUTO-ORGANIZÁVEIS .....</b>	<b>188</b>
<b>6.4.1 Cenário na gestão por sistemas auto-organizáveis .....</b>	<b>188</b>
<b>6.4.2 Avaliação na gestão por sistemas auto-organizáveis .....</b>	<b>195</b>
<b>6.5 PLANTA COM GESTÃO POR SISTEMAS AUTO-ORGANIZÁVEIS, SEGUNDO CENÁRIO EXPERIMENTADO .....</b>	<b>197</b>
<b>6.5.1 Cenário na gestão por sistemas auto-organizáveis, segundo experimento.....</b>	<b>198</b>
<b>6.5.2 Avaliação na gestão por sistemas auto-organizáveis no segundo cenário experimentado.....</b>	<b>200</b>
<b>6.6 AVALIAÇÃO DOS RESULTADOS DA APLICAÇÃO DO MÉTODO.....</b>	<b>202</b>
<b>7 CONCLUSÕES.....</b>	<b>208</b>
<b>REFERÊNCIAS .....</b>	<b>213</b>
<b>ANEXO I: PROGRAMAS DO CLP PARA UMA SEQUÊNCIA PRODUTIVA.....</b>	<b>222</b>

## LISTA DE ILUSTRAÇÕES

Figura 1 Exemplo de necessidade de diversificação de produtos. ....	14
Figura 2 Sobreposição no ciclo de vida de produtos. ....	16
Figura 3 Oito fases do ciclo de vida de um produto em blocos.....	18
Figura 4 Necessidade de sistemas produtivos. ....	18
Figura 5 Soluções da academia e necessidade da indústria.....	21
Figura 6 Perspectiva de investimentos em CLP comparando 2011 com previsão para 2016. .	22
Figura 7 Hipótese para soluções da problemática elencada. ....	24
Figura 8 Composição de um agente. ....	29
Figura 9 Metamodelo de agente e artefatos (A&A). ....	30
Figura 10 Estrutura de um SOA. ....	31
Figura 11 Interação entre os agentes em um MAS.....	33
Figura 12 Tela de acesso do usuário no ambiente JADE. ....	41
Figura 13 Exemplo de estrutura da plataforma JADE.....	42
Figura 14 Agente programado em JAVA.....	44
Figura 15 Formato de execução dos comportamentos no JADE.....	45
Figura 16 Fluxograma de execução de um agente. ....	45
Figura 17 Código exemplo de envio de mensagem FIPA-ACL.....	50
Figura 18 Código exemplo de recepção de mensagens FIPA-ACL.....	51
Figura 19 Estrutura do protocolo FIPA-ACL.....	53
Figura 20 Protocolo FIPA <i>Request</i> . ....	54
Figura 21 Protocolo FIPA <i>Contract Net</i> . ....	56
Figura 22 Directory Facilitator (DF) em JADE.....	57
Figura 23 Estrutura de postagem no DF. ....	57
Figura 24 Código exemplo para postagem no DF. ....	58
Figura 25 Código exemplo para busca de serviços no DF. ....	59
Figura 26 Módulos que compõe um CIM. ....	62
Figura 27 Estrutura CIM no sistema produtivo. ....	64
Figura 28 Módulos básicos que compõe um ERP.....	65
Figura 29 Estrutura do sistema ERP no meio produtivo. ....	69
Figura 30 Arquitetura de um MES em um sistema produtivo.....	72
Figura 31 Pirâmide da automação segundo ISA-95. ....	75
Figura 32 Estrutura de um sistema Auto-organizável em um sistema produtivo.....	75
Figura 33 Comparação entre sistema convencional e sistema auto-organizável de manufatura. .....	78
Figura 34 Método proposto para modelar um recurso mecatrônico como um objeto.....	82
Figura 35 Modelo de interação em três camadas para o ambiente dos agentes. ....	85
Figura 36 Modelo de configuração dos agentes para formar um arranjo produtivo. ....	87
Figura 37 Modelo de interação dos agentes em nuvem.....	93
Figura 38 Composição do agente genérico com o agente de interface com a manufatura.....	96
Figura 39 Composição de um agente utilizando JADE e PROTÉGÉ.....	96
Figura 40 Etapas propostas para modelar um agente mecatrônico. ....	97
Figura 41 Modelo de diagnóstico e predição da propagação de falha.....	98
Figura 42 Arquitetura do agente holômico na proposta da Rockwell Automação.....	104
Figura 43 Proposta para envelopar as funcionalidades de um dispositivo.....	106
Figura 44 Arquitetura para Controle de Sistemas Reconfiguráveis de Manufatura.....	108
Figura 45 Interface entre o HRMS e o componente físico por XML.....	109
Figura 46 Proposta para proceder as coalizões entre agentes para formar habilidades específicas.....	111

Figura 47	Conceito do modelo para um <i>holon</i> na plataforma ADACOR. ....	117
Figura 48	Planta com sistemas Multiagentes interagindo por conexão sem fio.....	119
Figura 49	Identificação da lacuna na abordagem de sistemas que atendem a diversidade, agilidade e auto-organização. ....	127
Figura 50	Foco de abordagem da proposta de tese.....	130
Figura 51	Exemplo de um sistema integrado de manufatura com a sequência de manufatura definida pela disposição das estações. ....	132
Figura 52	Propósito do trabalho para interligar as estações de manufatura. ....	134
Figura 53	Princípio do compartilhamento de variáveis através de rede industrial.....	136
Figura 54	Alternativas para interfaceamento do agente com o CLP, sem alteração do programa. ....	138
Figura 55	Rotina de manufatura em uma máquina em um processo produtivo. ....	139
Figura 56	Interação do operador e máquina em um processo automatizado.....	140
Figura 57	Interação entre o operador e a máquina. ....	143
Figura 58	Interação entre o computador e a máquina. ....	144
Figura 59	Diagrama de classe com a relação entre os agentes para formar o PLCAgent. ....	145
Figura 60	Modelo da interface do PLCAgent, configurado para Agente Peça, Agente Máquina e Agente Transporte.....	146
Figura 61	Plataforma multiagente em JADE para dispor a infraestrutura necessária. ....	147
Figura 62	Sequência de passos do método proposto. ....	148
Figura 63	Estrutura de implantação dos agentes da plataforma a partir do PLCAgent. ....	154
Figura 64	Cenário para ensaio da manufatura controlada por CLP. ....	156
Figura 65	Conexões lógicas entre os controladores lógicos programáveis. ....	157
Figura 66	Configuração dos agentes para a manufatura proposta.....	164
Figura 67	Serviços da plataforma JADE. ....	166
Figura 68	Configuração da plataforma JADE para o sistema de manufatura. ....	167
Figura 69	Implantação do PLCAgent Transporte.....	168
Figura 70	Implantação do PLCAgent Máquina de usinagem e manipulação. ....	168
Figura 71	Implantação do PLCAgent Peça. ....	170
Figura 72	Sistema integrado de manufatura em que o cenário foi ensaiado. ....	170
Figura 73	Ocorrência de <i>deadlock</i> e sua solução. ....	172
Figura 74	Estande da Manufatura Avançada na FEIMEC2016. ....	176
Figura 75	Produto gerado na planta de Manufatura Avançada. ....	177
Figura 76	Ilustração da planta de Manufatura Avançada. ....	177
Figura 77	Imagem da planta real de Manufatura Avançada.....	179
Figura 78	Diagrama da estrutura da planta demonstrativa da Manufatura Avançada.....	180
Figura 79	Hierarquia da comunicação e interação com MES. ....	182
Figura 80	Hierarquia da comunicação e interação com CIM. ....	184
Figura 81	Programação em blocos do CLP. ....	185
Figura 82	Exemplo de comunicação entre o agente e a máquina.....	188
Figura 83	a) agente de comunicação da máquina; b) agente que simula a máquina.....	189
Figura 84	Agente Máquina. a) Escaneamento, b) Programação, c) Usinagem, d) Inspeção, e) Montagem, f) Armazenamento, g) Entrega, f) Transporte. ....	191
Figura 85	PLCAgente, executa o lançamento dos agentes Peças. a) PLCAgente lançador, b) agente peça em execução. ....	192
Figura 86	Hierarquia da comunicação e interação com Sistemas Auto-organizáveis.....	193
Figura 87	Diagrama de Sequência com a interação entre peça, máquina e sistema de transporte. ....	194
Figura 88	Agente com a função de sistema supervisorio. ....	196
Figura 89	Planta de manufatura industrial exposta na MERCOPAR.....	197

Figura 90 Hierarquia da comunicação e interação com Sistemas Auto-organizáveis no segundo cenário de ensaio. ....	198
Figura 91 Diagrama de contatos da lógica local que comanda a máquina. a) sem a interferência do Agente; b) com a interferência do agente. ....	200

## LISTA DE TABELAS

Tabela 1	Análise da problemática e proposta de abordagem. ....	26
Tabela 2	Análise comparativa entre SOA e MAS. ....	34
Tabela 3	Comparação entre as plataformas para desenvolvimento de Sistemas Multiagentes estudados. ....	39
Tabela 4	Lista de software ERP. ....	68
Tabela 5	Comparativo entre os sistemas de manufatura industrial. ....	78
Tabela 6	Contribuições das pesquisas estudadas para o desenvolvimento desta tese. ....	120
Tabela 7	Relação de abordagens, resultados e contribuição dos trabalhos pesquisados. ....	122
Tabela 8	Análise das características do processo de manufatura. ....	142
Tabela 9	Características observadas para máquina de montagem. ....	159
Tabela 10	Características observadas para máquina de usinagem. ....	159
Tabela 11	Características observadas para o sistema de transporte. ....	160
Tabela 12	Características observadas para o produto a ser produzido. ....	161
Tabela 13	Relação de métricas utilizadas na validação experimental. ....	174
Tabela 14	Análise da planta de Manufatura Avançada a partir das métricas propostas. ....	183
Tabela 15	Análise da planta de Manufatura Avançada com gestão por CIM. ....	187
Tabela 16	Análise da planta de Manufatura Avançada com gestão por Sistemas Auto-organizáveis. ....	196
Tabela 17	Memória ocupada no controlador lógico programável considerando o programa da lógica local. ....	201
Tabela 18	Análise da planta de Manufatura Avançada com gestão por Sistemas Auto-organizáveis. ....	202
Tabela 19	Resultados da avaliação dos sistemas de gestão. ....	206

## LISTA DE ABREVIATURAS

- ACL – *Agent Communication Language* (Linguagem de Comunicação entre Agentes)
- AID – *Agent Identifiers Definition* (Definição do Identificador do Agente)
- AMS – *Agent Management System* (Gerenciamento do Sistema de Agentes)
- ATP – *Agent Transfer Protocol* (Protocolo de Comunicação entre Agentes)
- CLP – Controlador Lógico Programável
- DF – *Directory Facilitator* (Postagem de Serviços)
- DSC – *Discentred System Control* (Sistema de Controle Descentralizado)
- EAS - *Evolvable Assembly System* (Sistemas Evolutivos de Manufatura)
- FBD - *Function Blocks Diagram* (Diagrama de Blocos Funcionais)
- FIPA – *Fundation for Intelligent Physical Agents* (Fundação para Agentes Físicos)
- FIPA-ACL - *FIPA-Agent Communication Language* (FIPA–Linguagem de Comunicação de Agentes)
- FMS – *Flexible Manufacturing Systems* (Sistemas Flexíveis de Manufatura)
- HTTP - *HyperText Transfer Protocol* (Protocolo de Transferência de Hipertexto)
- IEC – *International Electrotechnical Commission* (Comissão Internacional de Eletrotécnica)
- IHM – *Interface human Machine* (Interface Homem Máquina)
- IL - *Instruction List* (Lista de Instruções)
- JADE – *JAVA Agent DEvelopment* (Desenvolvimento de Agentes em JAVA)
- JVM – *JAVA Virtual Machine* (Máquina Virtual JAVA)
- LD - *Ladder Diagram* (Diagrama Ladder)
- MAS – *Multi-Agent System* (Sistema Multiagente)
- MMS – *Manufacturing Messaging Services* (Manufatura por Serviços de Mensagens)
- OEE – *Overall Equipament Effectivences* (Eficiência Geral de Equipamento ou Máquinas)
- RMS – *Reconfigurable Manufacturing System* (Sistemas Reconfiguráveis de Manufatura)
- SFC - *Sequential Function Chart* (Sequenciamento Gráfico de Funções)
- SOA – *Service Oriented Architecture* (Arquitetura Orientada a Serviços)
- ST - *Struct Text* (Texto Estruturado)
- UML - *Unified Modeling Language* (linguagem gráfica padrão para modelagem de dados)
- XML – *Extensible Markup Language* (Linguagem Extensível de Marcação)
- WWW - *world wide web* (Rede de alcance mundial)

## 1. INTRODUÇÃO

A indústria de manufatura tem se deparado com uma necessidade de mercado que remete ao consumo de produtos altamente personalizados. Durante vários anos o conceito de produção em massa, caracterizada pela produção do mesmo produto em grande escala, foi amplamente utilizado. Hoje em dia este conceito é incapaz de tratar as variações no tipo de produto, e não consegue mais responder aos desafios de modernidade e dinamismo. Grandes lotes de produção, linhas de produção com máquinas e processos idênticos e a padronização de produtos tendem a deixar de existir. A produção em grandes quantidades continua a ser demandada, porém, como aponta (MENDES et al., 2008), há uma tendência à produção em massa de produtos altamente personalizados. A Figura 1 apresenta um exemplo de necessidade de diversificação de produtos demandados pelo mercado.



**Figura 1 Exemplo de necessidade de diversificação de produtos.**

Fonte: <http://www.fotosearch.com.br>

Nesta Figura 1 se pode observar aparelhos refrigeradores e automóveis que possuem funcionalidades semelhantes entre si, mas com um alto grau de personalização, requerendo tratamento específico na produção para atender às individualidades.

Qualidade e funcionalidade deixaram de ser os únicos atributos de um produto. A eles incorporaram-se diversidade, atualização, possibilidade de agregar funcionalidades, dentre outros. Atributos que remetem a um sistema de produção cada vez mais ágil, flexível e eficaz. Em (JAMES; SMIT, 2005) são descritos alguns dos requisitos da chamada fábrica do futuro:

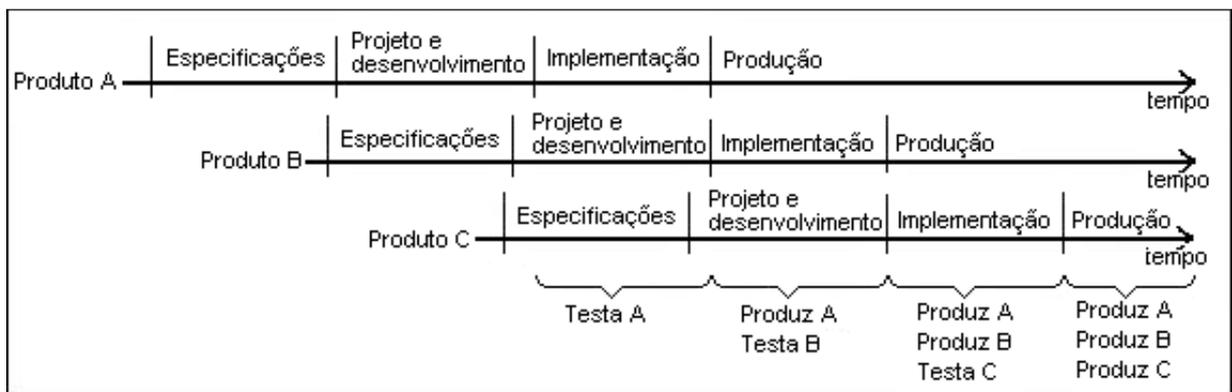
- a) Capacidade de interação;
- b) Cooperação entre as empresas;
- c) Suporte de *hardware* heterogêneo interoperável;
- d) Escalabilidade, adicionando ou retirando o número de recursos sem interromper as operações;
- e) Agilidade através da adaptabilidade e reconfiguração;
- f) Tolerância à falha e recuperação dela.

A flexibilidade nos sistemas produtivos (BEACH et al., 2000) é o fator que leva este sistema a ser capaz de se modificar, seja estruturalmente ou em sua sequência de operação, de forma a mudar a regra do fluxo produtivo já estabelecido. Um sistema produtivo segue uma regra de fluxo de produção definida, mas deve ser flexível o suficiente para modificar esta regra em função de ocorrências no processo (mudança de requisitos, falta de um componente, sobrecarga de tarefas em um ponto do processo, entre outros).

Quanto ao ciclo de vida de um produto, detendo-se nas etapas do desenvolvimento à produção, percebe-se a necessidade de desenvolvimento e produção de uma grande variedade de produtos, visando atender um mercado ávido por inovações e serviços customizados, o que acaba implicando em um aumento na diversidade de produtos. Ciclos de produtos distintos necessitam ocorrer concomitantemente, como apontado em (OLIVEIRA, 2003). O desenvolvimento de um produto é definido pelas etapas de especificação, projeto e desenvolvimento, implementação e produção propriamente dita. Na etapa de especificação ocorre o levantamento de requisitos, a delimitação das especificações em função das funcionalidades requeridas e a viabilidade de produção. O desenvolvimento do projeto diz respeito ao protótipo com as funcionalidades requeridas e seus métodos de fabricação. A implementação é o ensaio de produção, onde o produto é testado no processo produtivo, sendo validados os processos de manufatura e seus parâmetros. Então o produto começa a ser

produzido em escala industrial, com os parâmetros, requisitos e insumos definidos no plano de processo. Nota-se então que a etapa de implementação requer os recursos de manufatura tanto quanto a produção para produzir um protótipo. Este fato já era apontado em (KOREN et al., 2003), que estudou a etapa de *ramp up*<sup>1</sup>, necessária a cada início de um novo produto e que, na eminência de haver dois produtos ou mais sendo manufaturados na mesma linha de produção, estas etapas de *ramp up* ocorrerão para cada início de produção de cada produto.

Em um sistema produtivo moderno é comum que, no momento em que um produto entra no ciclo de produção, um novo produto já deve estar em fase de implementação. Isto exige um sistema de manufatura que possa ser reconfigurado para atender aos dois processos. A Figura 2 ilustra graficamente os ciclos de três produtos sendo desenvolvidos concomitantemente, atendendo à diversidade exigida pelo mercado.



**Figura 2 Sobreposição no ciclo de vida de produtos.**

Fonte: autor.

O mercado necessita atender as demandas de produto de uma população que prima pela diversidade, tanto de aparência quanto de funcionalidades. No entanto, para um produto ser concebido, requer um conjunto de fases, as quais envolvem desde sua idealização até seu descarte. Todo seu ciclo de vida deve estar programado afim de que ao final de sua vida útil não seja mais um produto desnecessário descartado no meio ambiente.

<sup>1</sup> Etapa necessária para configurar as máquinas e equipamentos para se adequar ao novo produto que está sendo solicitado em sua produção.

A análise do Ciclo de Vida de um Produto compreende desde a extração dos recursos naturais ou matérias primas, necessários à sua produção, até a disposição final do produto ao fim de sua vida útil (VALLE, 2002, p. 145).

E o próprio mercado para este novo produto deve ser pensado de forma a respeitar as etapas com que sua inserção ocorre. Considerando os aspectos mercadológicos, Kotler (1999) define quatro fases, considerando as vendas, e cinco fases, considerando os lucros obtidos com as vendas. Estas fases são:

- I. Desenvolvimento do produto: começa quando a empresa encontra e desenvolve a ideia de um novo produto a partir de uma demanda do mercado. Durante esse desenvolvimento, as vendas são iguais a zero e os custos do investimento são crescentes;
- II. Introdução do produto no mercado: período de lento crescimento das vendas à medida que o produto é introduzido no mercado. Não há lucros nesse estágio devido aos altos custos da introdução, especialmente com propaganda e distribuição;
- III. Crescimento de vendas: período de rápida aceitação no mercado e de lucros crescentes. Isto supondo que o produto foi aceito pelo mercado;
- IV. Maturidade: período em que o crescimento das vendas diminui. As vendas começam a diminuir, pois outros novos produtos concorrentes estão se projetando. Gasta-se muito dinheiro com propaganda para enfrentar a concorrência;
- V. Declínio: período em que as vendas e os lucros caem. Isto ocorre por obsolescência ou devido produtos concorrentes. Produto deverá ser descontinuado.

Portanto, é sabido que todo produto, por mais inovador que seja, terá fases de desenvolvimento, ascensão, maturidade e declínio, cabendo a um novo produto lhe substituir no mercado.

Implementar um sistema de manufatura significa definir máquinas, equipamentos, ordem de execução, insumos, matéria-prima, arranjo logístico e programação dos controladores, segundo a especificação do projeto do produto. A Figura 3 apresenta uma versão em blocos das oito fases do ciclo de vida do produto, descrita por Kotler (2007).



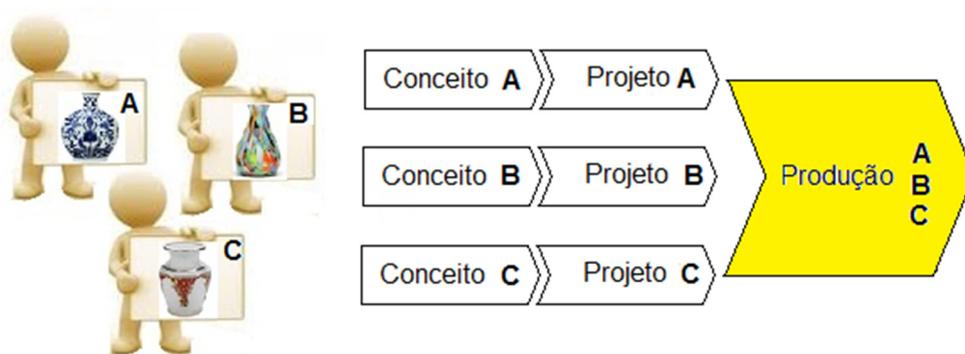
**Figura 3 Oito fases do ciclo de vida de um produto em blocos.**

Fonte: autor

Da forma posta por Kotler, em termos de planejamento de produção, a alternativa seria a implementação de uma linha de produção para cada produto a ser desenvolvido, de sua fabricação até seu desuso. Mas para a indústria isto não é viável, pois se trata do dispêndio de recursos que irão onerar em demasia o produto final.

Se de um lado há consumidores que demandam produtos diversificados com uma velocidade cada vez maior, de outro a indústria demanda sistemas produtivos que possam dar conta de problemáticas que o meio produtivo enfrenta, sem precisar duplicar ou criar novas linhas fabris.

A Figura 4 descreve este fato e a necessidade da indústria.



**Figura 4 Necessidade de sistemas produtivos.**

Fonte: autor.

A alternativa então é a concepção de sistemas de manufatura que atendam a diversidade, a ponto de produzir produtos diferentes no mesmo meio fabril, e que sejam flexíveis a ponto de atender as variações de processo que cada produto requer. Que sejam auto-organizáveis a ponto de modificar seu arranjo e sequência de produção sem a necessidade de reprogramar controladores ou modificação de leiautes.

Para uma indústria se manter no mercado é necessário que oferte novos produtos e atinja novos mercados, mas isto só é possível se seu sistema produtivo atender aos novos requisitos para manufatura, são eles:

- a) Diversidade: capacidade de manufaturar produtos com características distintas no mesmo meio produtivo;
- b) Agilidade: capacidade de um sistema de manufatura ser ágil a ponto de responder rapidamente a uma nova necessidade de forma eficaz, ser flexível a ponto de absorver componentes heterogêneos e de absorver a manufatura de produtos diferentes no mesmo meio de produção;
- c) Auto-organização: capacidade de compor novos arranjos de seus componentes para formar novos meios produtivos, de ter escalabilidade a ponto de absorver a inserção ou retirada de recursos sem parar o fluxo produtivo. De ser tolerante à falha de forma que outro componente assuma funcionalidades na eventual falta de um componente do sistema, de integrar-se a outros componentes e de provocar a cooperação e compartilhamento de recursos.

Em síntese, para indústria manter-se no mercado ou atingir mercados novos deverá ter sistemas produtivos que atendam à diversidade, agilidade e auto-organização.

A pesquisa acadêmica tem contribuído para que soluções sejam apresentadas aos problemas citados. Elas buscam desenvolver ou aprimorar métodos e técnicas de forma a

mostrar abordagens de cunho científico que possam contribuir para solução de problemas nos meios produtivos. Dentre estas pesquisas é apresentada esta tese, onde a proposta é investigar e apresentar método, tecnologia ou arranjos tecnológicos que se apresente com proposta de solução a problemas que o meio produtivo enfrenta.

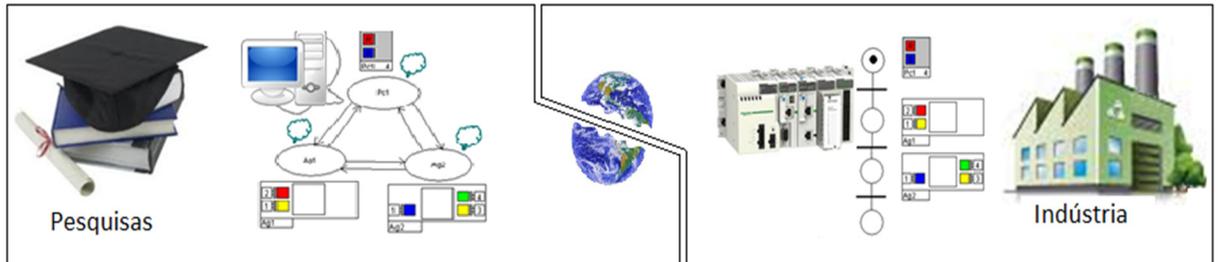
## 1.1 PROBLEMÁTICA

Para que se possa atingir os novos mercados, os sistemas produtivos passam a preocupar-se com necessidade de diversificação, agilidade e auto-organização. Surgem então os Sistemas Evolutivos de Manufatura (EAS), mencionados no trabalho de (CAVALCANTE; PEREIRA; BARATA, 2010) e (BARATA; RIBEIRO; ONORI, 2007), com suas funcionalidades de autoconfiguração, reorganização, autonomia de gerenciamento do processo produtivo, disponibilidade de serviços ao sistema (utilizando conceitos de “arquiteturas orientadas a serviço” ou em inglês “*service oriented architectures* - SOA) e disposição para negociação e agendamento de processos entre seus agentes (usando conceitos de “Sistemas Multiagentes” ou em inglês – “*multi-agent systems*” - MAS). Eles se apresentam como uma tecnologia promissora para atender aos requisitos de mercado de produtividade em massa e diversidade de produção, dispondo da flexibilidade de um FMS e a autonomia de um sistema descentralizado (DSC), como consta em (LEITÃO; RESTIVO, 2001).

Estas soluções com sistemas autoadaptáveis remetem a uma gestão descentralizada dos processos de manufatura industrial. Esta gestão é então executada em computadores ou controladores industriais com grande disponibilidade de capacidade computacional (NAMBIAR, 2010), capazes de executar aplicativos em *software* em sua estrutura operacional. São alternativas que as pesquisas propõem para que a indústria possa evoluir seus sistemas de

gestão para obter a diversidade, agilidade e auto-organização desejadas. Porém isto pode implicar na implementação de sistemas de manufatura novos ou alterar os já existentes.

A Figura 5 apresenta uma ideia deste paradoxo.

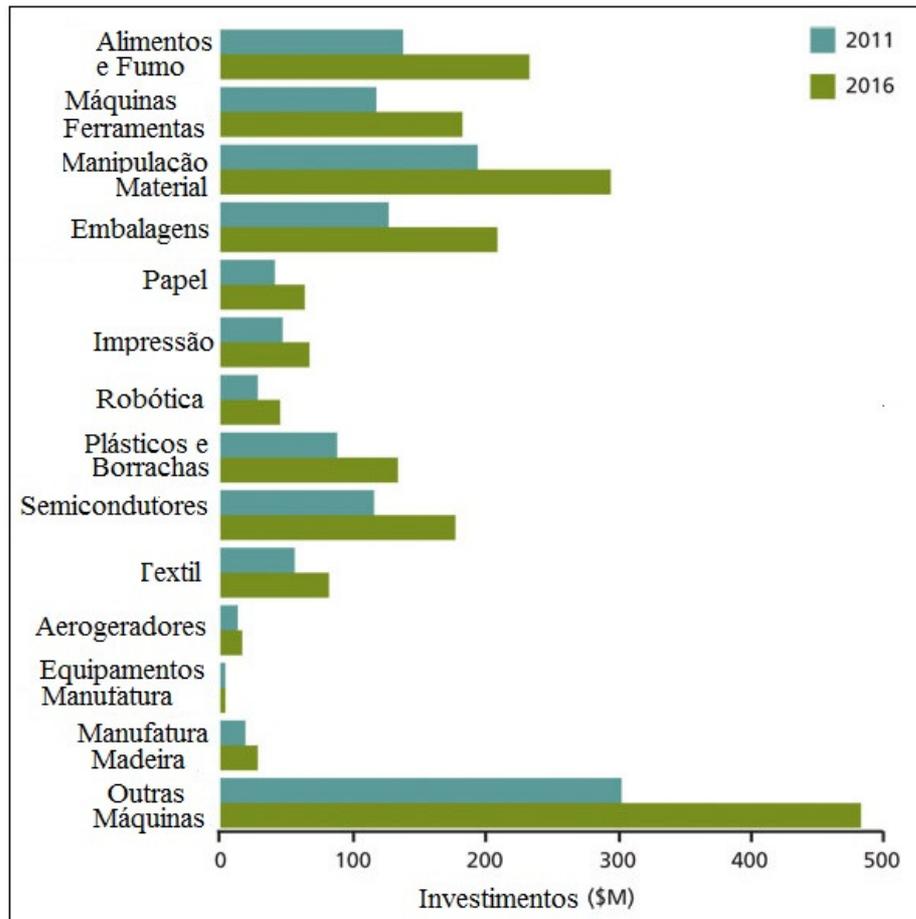


**Figura 5 Soluções da academia e necessidade da indústria.**

Fonte: autor.

Se de um lado as pesquisas apresentam a solução para que os sistemas de manufatura industrial possam atender aos novos requisitos de mercado, de outro a indústria possui seus sistemas de manufatura já implementados, de forma centralizada, com controladores lógicos programáveis e uma infraestrutura de conexão de entradas e saídas. Estes conceitos e tecnologias utilizadas na indústria atual são frutos de pesquisas acadêmicas anteriores, mas que não se propunham a provocar a interação entre os sistemas. E talvez desfazer-se desta infraestrutura não seja viável.

A revista (CONTROL, 2013) apontava em 2013 que mesmo que o mercado de venda do controlador lógico programável (CLP) estivesse em um crescimento modesto, o mercado nas Américas continuaria a crescer até 2016. A Figura 6 aponta um gráfico que mostra este crescimento previsto por áreas de atuação na indústria.



**Figura 6** Perspectiva de investimentos em CLP comparando 2011 com previsão para 2016.

Fonte: Adaptado de (CONTROL, 2013).

Neste gráfico se prevê um crescimento no investimento em CLP para todas as áreas industriais.

Haja vista a grande utilização de CLP como controladores da manufatura dentro das indústrias, estes então deverão ser os responsáveis em fazer com que o sistema de manufatura tenha a capacidade de atender aos novos requisitos desejados.

Portanto, a problemática que será abordada diz respeito a:

**Problemática:**

Fazer com que os controladores lógicos programáveis controlem sistemas de manufatura de forma a atender requisitos de diversidade, agilidade e auto-organização nos sistemas produtivos, usando uma abordagem minimamente invasiva.

**1.2 HIPÓTESE**

Uma tecnologia que pode fazer com que um sistema de manufatura opere os requisitos de diversidade, agilidade e auto-organização são os Sistemas Multiagentes (MAS), definidos em (WOOLDRIDGE, 2002) e descrito em (BELLIFEMINE et al., 2007). MAS é um sistema em que os atores do processo produtivo podem ser modelados como agentes, onde um agente é uma entidade de software que pode gerenciar cada elemento da manufatura, dispondo suas funcionalidades na forma de serviços. Cada agente pode então oferecer ou solicitar serviços dentro de uma plataforma onde os agentes se comunicam uns com os outros.

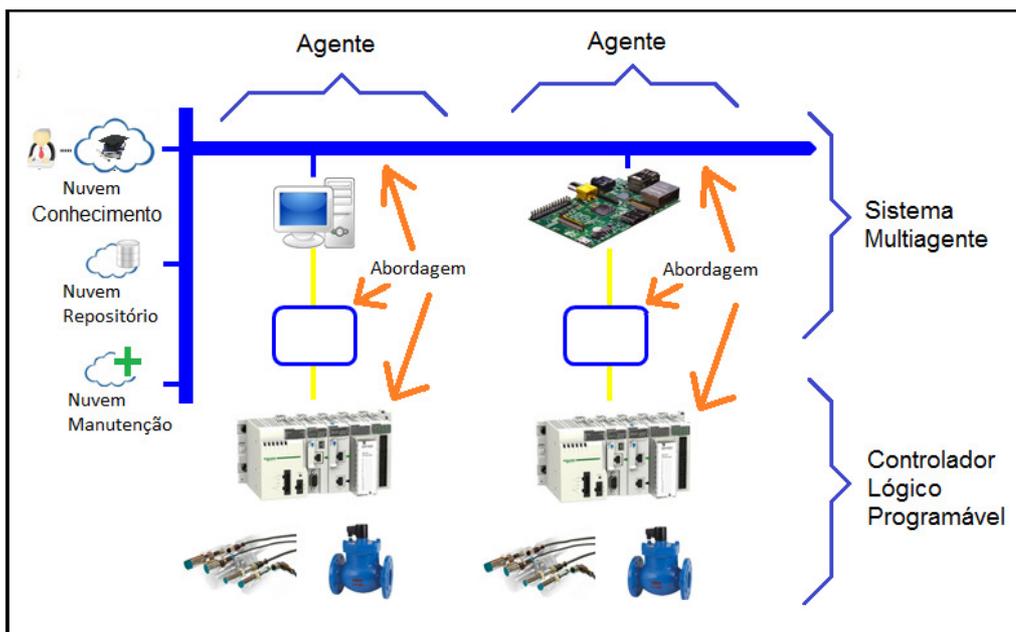
Com estas características os Sistemas Multiagentes podem fazer com que o sistema produtivo, controlado por controladores lógicos programáveis, atue com funcionalidades para atender a diversidade, a agilidade e a auto-organização requeridas (MARIK; McFLARLANE, 2005).

A hipótese prevista para este trabalho é:

### Hipótese

É possível a utilização de Sistemas Multiagentes junto aos Controladores Lógicos Programáveis para gerir o sistema de manufatura, de forma a atender os requisitos de diversidade, agilidade e auto-organização do meio produtivo, sem alterar significativamente os tempos envolvidos no processo produtivo.

A Figura 7 demonstra a hipótese levantada, onde se percebe o ambiente do Sistema Multiagente, os controladores lógicos programáveis e um foco de abordagem que se refere ao meio com que as funcionalidades do CLP irão comunicar-se com os agentes do Sistema Multiagente.



**Figura 7 Hipótese para soluções da problemática elencada.**

Fonte: autor.

A composição do controlador lógico programável e do sistema computacional (computador ou um processador embarcado) formará então o agente, que atuará no Sistema Multiagente, comunicando-se com outros agentes para consulta, disposição e alocação de serviços. Também há a possibilidade de acesso dos agentes aos recursos postados em “ambientes em nuvem”, mas esta funcionalidade não foi implementada neste trabalho.

### 1.3 OBJETIVOS

Com o objetivo de solucionar a problemática definida através da hipótese que se apresentou, o trabalho se propõe a:

#### **Objetivo Geral:**

Possibilitar que sistemas de manufatura controlados por controladores lógicos programáveis possam participar de Sistemas Multiagentes, valendo-se das funcionalidades destes para atender as necessidades de diversidade, agilidade e auto-organização de manufaturas industriais.

#### **Objetivos específicos:**

1. Desenvolver um método que agregue a controladores lógicos programáveis as funcionalidades de Sistemas Multiagentes;
2. Avaliar a funcionalidade do método num estudo de caso em planta industrial piloto;
3. Avaliar a abrangência do método para sistemas de manufatura distintos.

Estes objetivos, sendo alcançados, apresentam solução para problemática definida, pois possibilitam aos controladores lógicos programáveis atuarem em um Sistemas Multiagentes, atendendo assim aos requisitos de diversidade, agilidade e auto-organização.

A Tabela 1 apresenta uma síntese da análise da problemática, envolvendo de forma sistemática a necessidade, a problemática em si, a hipótese e os objetivos para atender um sistema de manufatura industrial.

**Tabela 1 Análise da problemática e proposta de abordagem.**

<b>Necessidade</b> Qual necessidade?	<b>Problemática</b> Qual o problema?	<b>Hipótese</b> Qual a solução?	<b>Objetivo</b> Como será feito?
Diversidade, agilidade e auto-organização no meio produtivo.	Fazer controladores lógicos programáveis controlarem sistemas de manufatura de forma a atender requisitos desejados na necessidade destacada.	Utilizar Sistemas Multiagentes junto aos controladores lógicos programáveis.	Desenvolver o método que agregue aos controladores lógicos programáveis as funcionalidades de Sistemas Multiagentes; Avaliar a funcionalidade do método e sua abrangência.

Fonte: Autor.

#### **1.4 ESTRUTURA DO TRABALHO**

Este documento está dividido em seis capítulos, que descrevem cada etapa do trabalho. Além da introdução no Capítulo 1, o documento traz no Capítulo 2 a fundamentação teórica dos tópicos principais para que haja a compreensão das demais etapas do trabalho.

O Capítulo 3 apresenta um estudo sobre os sistemas de gestão da manufatura aplicáveis na indústria, com um comparativo entre eles, na busca de lacunas entre as abordagens.

O Capítulo 4 traz a discussão sobre o estado da arte de sistemas auto-organizáveis em manufatura industrial com as visões da academia e de grupos de pesquisa.

O Capítulo 5 apresenta a proposta de tese propriamente dita, com o método proposto e desenvolvido.

O Capítulo 6 apresenta a aplicação do método em três cenários distintos, para a análise dos resultados.

E, por fim, o Capítulo 7 apresenta as conclusões deste estudo.

## 2. FUNDAMENTAÇÃO TEÓRICA

Nesta etapa os conceitos mais relevantes são abordados de forma a esclarecer a forma com que um componente ou uma tecnologia funciona, suas características, funcionalidades e sua aplicação. Os fundamentos citados a seguir são os considerados mais importantes para a compreensão da proposta.

### 2.1 AGENTE EM SOFTWARE

Para que um Sistema Multiagente agregue um controlador lógico programável é necessário que este último seja modelado por um agente, de forma que seus serviços possam ser negociados com outros agentes de um sistema. O estudo de agentes é um fundamento necessário para a compreensão deste trabalho.

A FIPA (*Foundation for Intelligent Physical Agents*) (BELLIFEMINE; CAIRE; GREENWOOD, 2007) define agente como uma entidade que reside em um ambiente onde interpretam dados através de sensores, refletem eventos no ambiente e executam ações que produzem efeitos no ambiente. Um agente pode ser *software* ou *hardware* puro, ou a combinação entre eles.

No geral, agentes são entidades de *software* autônomas que atuam em determinados ambientes, de forma a interagir com outros agentes. Além de produzir ações e percepções sem requerer intervenções humanas constantes.

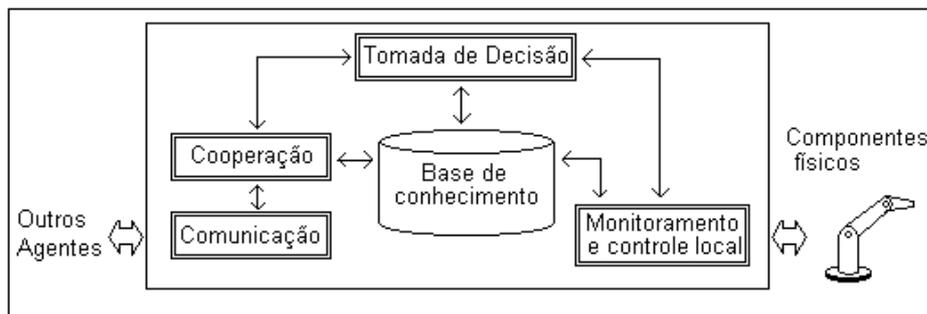
Em (WOOLDRIDGE, 2009) se descreve uma classificação dos tipos de Agentes, como segue:

- a) Agentes móveis: são agentes que têm a mobilidade como característica principal, isto é, uma capacidade de mover-se por uma rede interna local ou distribuída em domínios;

- b) Agentes situados ou estacionários: são aqueles opostos aos móveis. São fixos em um mesmo ambiente ou plataforma;
- c) Agentes competitivos: são agentes que “competem” entre si para a realização de seus objetivos ou tarefas. Não há colaboração entre os agentes;
- d) Agentes coordenados ou colaborativos: agentes com a finalidade de alcançar um objetivo maior realizam tarefas específicas, porém coordenando-as entre si de forma que suas atividades se complementem;
- e) Agentes reativos: são agentes que reagem a estímulos sem ter memória do que já foi realizado no passado e nem previsão da ação a ser tomada no futuro;
- f) Agentes Cognitivos: ao contrário dos agentes reativos, eles podem raciocinar sobre as ações tomadas no passado e planejar ações a serem tomadas no futuro.

Tem-se o agente como um sistema capaz de, não somente agir autonomamente, mas também, em alguns casos, interagir com outros agentes, os quais, assim como ele, objetivam alcançar suas metas. Um agente pode ainda ter algumas características que o tornem mais ou menos útil para uma determinada tarefa, tais como: autonomia, interatividade, adaptabilidade, sociabilidade, mobilidade, representatividade, pró-atividade, inteligência, racionalidade, imprevisibilidade, continuidade temporal, transparência, coordenação, cooperação, competição, robustez e confiabilidade.

A arquitetura de um agente genérico, segundo (LEITÃO; RESTIVO, 2001), é baseada em quatro módulos e um banco de dados local, que contém todas as informações pertinentes sobre o comportamento do agente, demonstrado na Figura 8.



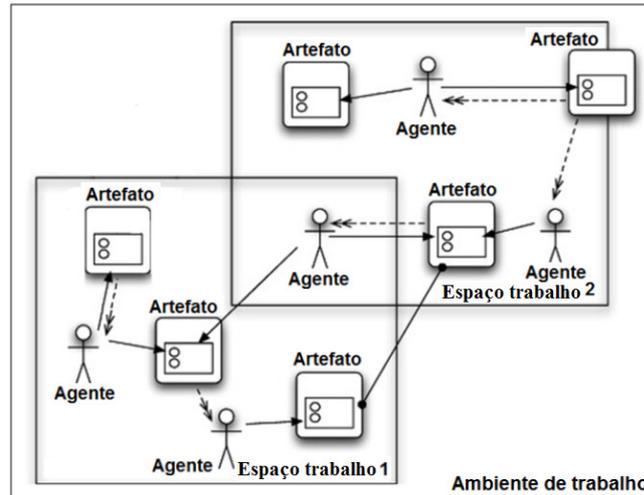
**Figura 8 Composição de um agente.**

Fonte: adaptado de (LEITÃO, 2009).

Pode-se descrever cada módulo da Figura 8 como:

- a) Módulo de tomada de decisão: este módulo controla todas as atividades do agente e inclui a resolução de problemas e tomada de decisão;
- b) Módulo de cooperação: este módulo gerencia a cooperação com agentes exteriores, solicitando a cooperação de outros agentes, coletando as respostas e enviando para o módulo de tomada de decisão;
- c) Módulo de comunicação: este módulo trata da necessidade de padronizar a interação entre agentes distribuídos e define uma comunicação linguística;
- d) Módulo de monitoramento e controle local: este módulo pretende controlar e monitorar a execução operacional do agente;
- e) Módulo de base de conhecimento: o banco de dados armazena todos os conhecimentos sobre o comportamento do agente.

Há vários conceitos que tratam de arranjo entre os agentes, dentre os quais se cita o metamodelo de agentes e artefatos (OMICINI; RICCI; VIROLI, 2008). Este modelo é caracterizado em termos de três abstrações: agentes, artefatos e ambiente de trabalho. Segundo este modelo é projetado um Sistema Multiagente dentro de um ambiente comum, onde agentes não se comunicam apenas em uma linguagem de alto nível, mas também ao utilizarem diferentes tipos de artefatos organizados no ambiente de trabalho (THOMASI, 2014). A Figura 9 apresenta o metamodelo de agente e artefatos (A&A).



**Figura 9 Metamodelo de agente e artefatos (A&A).**

Fonte: Adaptado de (THOMASI, 2014).

Nesta figura podem-se observar os agentes, os artefatos e suas interações no espaço de trabalho.

Os artefatos são a representação de componentes passivos do sistema, concebidos para uma função específica. São entidades que não possuem autonomia e são reativas em termos de controle, porém eles podem ser considerados como ferramentas aos agentes, auxiliando em suas tarefas. Os artefatos disponibilizam operações que modificam seu estado, provocando efeitos no ambiente. Eles exercem um papel importante no auxílio aos agentes cognitivos para executar suas tarefas, permitindo uma interação indireta entre os agentes.

Os espaços de trabalho são containers utilizados para estruturar e organizar o conjunto de agentes e artefatos, permitindo que sistemas multiagentes complexos possam ser organizados em áreas de trabalho distribuídas entre vários nós da rede.

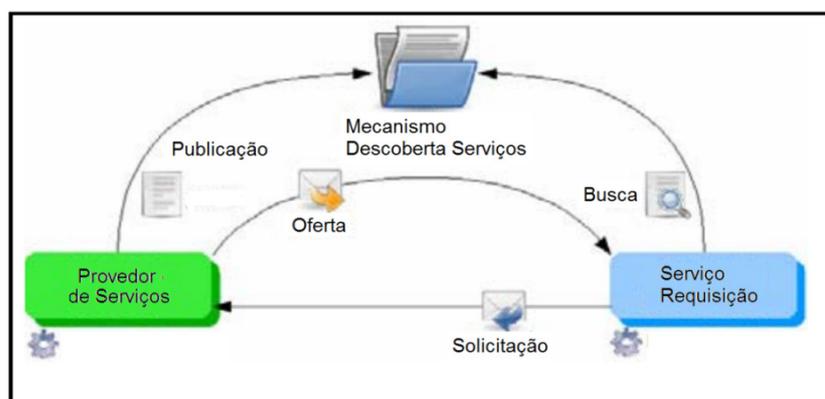
## 2.2 SISTEMAS MULTIAGENTES E ARQUITETURA ORIENTADA A SERVIÇOS

A Arquitetura Orientada a Serviços (*Service Oriented Architecture* - SOA) (CANDIDO et al., 2009) e os Sistemas Multiagentes (*Multi-Agent System* - MAS) (HERRERA et al., 2008)

são conceitos que se propõem a atender a agilidade e flexibilidade aos sistemas de produção. Em referenciais teóricos encontram-se diversos artigos que apontam soluções aos problemas de automação utilizando os princípios de SOA e outros artigos apontam MAS como instrumento. A dúvida que surge diz respeito à aplicação de MAS e SOA.

Segundo (JAMES; SMIT, 2005) SOA é o conjunto de princípios arquitetônicos para a construção de sistemas interoperáveis autônomos. Eles sistemas são criados independentes uns dos outros e operam independentemente do seu meio, fornecendo funcionalidades próprias (podem funcionar sozinhos). Esta autonomia é ressaltada por (OLIVEIRA; CAMARINHAMATOS; CÂNDIDO, 2008) que também relaciona SOA a um conjunto de arquitetura para a construção de sistemas interoperáveis autônomos, que suportam sistemas autônomos e oferecem funcionalidades de autosuficiência, funcionamento independente e flexibilidade.

A arquitetura orientada a serviços é um conceito onde uma manufatura pode ser vista como um conjunto de sistemas que dispõem serviços que se complementam para formar um processo produtivo (MENDES et al., 2008). Cada sistema dentro de um SOA presta serviços independente dos demais e, ao serem conectados ao conjunto, eles dispõem seus serviços aos outros sistemas. Uma vez solicitados, os sistemas de manufatura são alocados e disponibilizam seus serviços. A Figura 10 apresenta o princípio de SOA, onde há uma publicação e oferta de serviços, bem como uma solicitação destes.



**Figura 10 Estrutura de um SOA.**

Fonte: adaptado de (MENDES et al., 2008).

Neste exemplo, os serviços são dispostos uns aos outros e cada um deles vale-se de atividades que realizam para dispor seus serviços aos demais. Um serviço pode implementar mais de uma atividade, ou mesmo compor duas ou mais atividades para formar um serviço único.

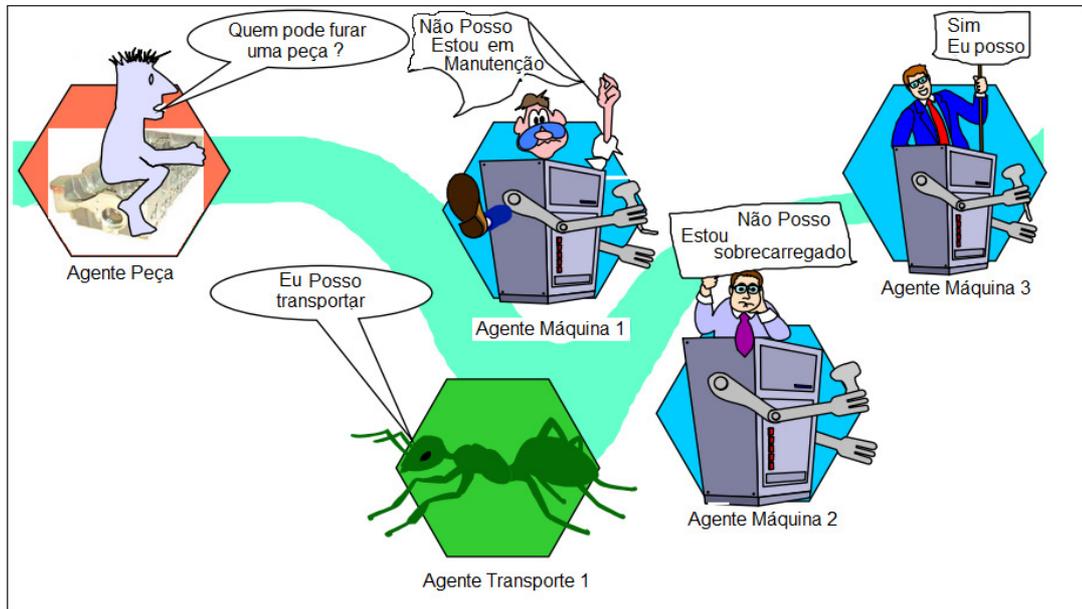
Fica claro no exemplo que cada sistema é autônomo e que o gerenciamento do fluxo de processo só depende da disponibilidade dos serviços, podendo outros sistemas serem agregados ou retirados, em função da necessidade de funcionalidades diversificadas. Mas o sincronismo deste gerenciamento não é trivial. A obra de (MENDES et al., 2008) já apontava o desafio de como descrever os processos que regulamentam o comportamento do sistema e como sincronizar e coordenar a execução dos serviços oferecidos pelas entidades distribuídas para atingir o comportamento desejado.

O Sistema Multiagente (FERBER, 1999) é uma tecnologia de *software* que é capaz de modelar e implementar comportamentos individuais e sociais em sistemas distribuídos. Como relatado também em (RIBEIRO; BARATA; MENDES, 2008).

No conceito MAS (PECHOUCEK; MARIK, 2008) os sistemas, agora chamados de agentes, são ativos, não apenas podem postar seus serviços e se submetem a serem solicitados em suas funcionalidades, mas assumem papel proativo para iniciar comunicação com outros agentes, propor negociação e alocar serviços. Em (LEITÃO, 2009) um Sistema Multiagente é descrito como a junção de componentes autônomos que representam objetos físicos ou lógicos no sistema a fim de atingir seus objetivos, e ser capaz de interagir com outros componentes quando ele não possuir capacidade de atingi-los sozinho.

Em MAS há métodos bem estabelecidos para descrever o comportamento de um agente (COLOMBO, 2005). O fato de que os agentes são regulados por normas internas que apoiem a implementação do comportamento social é uma clara vantagem. Isto é de grande importância

quando se considera sistemas que sofrem mudanças de execução dinâmica. A Figura 11 representa uma estrutura com a interação entre os agentes.



**Figura 11 Interação entre os agentes em um MAS.**

Fonte: adaptado de (COLOMBO, 2005).

Na figura representada fica evidente a interação entre os agentes. Como quando o “Agente Peça” solicita um serviço e os demais agentes respondem, seja com respostas negativas ou com aceite, ou assim como o agente de transporte, necessário para a movimentação da peça entre os “Agentes Máquinas”.

Assim, SOA e MAS se complementam. Em ambos casos cada participante no processo de montagem (robô, pinça, transportador, armazém de ferramentas, entre outros) pode ser identificado como um agente que pode oferecer diversos serviços, e que interage na realização de tarefas cooperativas.

A Tabela 2 apresenta uma comparação entre os sistemas SOA e MAS.

**Tabela 2 Análise comparativa entre SOA e MAS.**

<b>Característica</b>	<b>SOA</b>	<b>MAS</b>
Unidade básica	Serviço.	Agente.
Autonomia	Ambas as entidades denotam autonomia já que a funcionalidade fornecida é autossuficiente	
Descrição do comportamento	Em SOA o foco está no detalhamento da interface pública, ao invés de descrever detalhes de execução.	Existem métodos bem estabelecidos para descrever o comportamento de um agente.
Habilidade social	Habilidade social não é definida para SOA, no entanto, a utilização de um serviço implica a aceitação das regras definidas na descrição de interface.	Os agentes denotam a capacidade social regulada por normas internas ou ambientais.
Complexidade do encapsulamento	A natureza autossuficiente das funcionalidades previstas permite esconder os detalhes. Em SOA este encapsulamento é explícito.	
Infraestrutura de comunicação	SOA são suportados pela WEB e tecnologias relacionadas que podem perfeitamente funcionar na internet.	A maioria das implementações são otimizadas para uso de rede local.
Suporte para arquiteturas dinamicamente reconfiguráveis em tempo de execução	Reconfiguração muitas vezes exige uma reprogramação, a fim de indicar a nova configuração que se implementou.	A natureza flexível dos agentes torna reativo às mudanças no ambiente, sem necessidade de reprogramação.
Interoperabilidade	Assegurada pela utilização das tecnologias da WEB de propósito geral.	Fortemente dependente do cumprimento padrões FIPA.
Requisito computacional	Implementações leves com DPWS tem garantia de alto desempenho, sem restrições de interoperabilidade.	A maioria das implementações tem pesados requisitos computacionais.

Fonte: adaptada de (RIBEIRO; BARATA; MENDES, 2008).

Pelo fato de seus agentes possuírem capacidades sociais para interagir, prestarem serviços e negociar a alocação destes, através da interação entre os agentes, o sistema escolhido como conceito para o desenvolvimento deste trabalho é o Sistema Multiagente – MAS. Uma vez que o controlador lógico programável adquira as funcionalidades de um agente, este poderá se comunicar com outros agentes para ofertar e solicitar serviços. Isto ocorre através de plataformas de agentes, estudadas agora como fundamento importante ao trabalho.

## 2.3 PLATAFORMA DE AGENTES

Para que os agentes se comuniquem entre si, trocando mensagens, postando suas habilidades e requisitando serviços, é necessário que haja uma plataforma que suporte à instalação e execução dos mesmos. Estas plataformas são *softwares* que se comunicam em rede, garantindo um meio físico e uma arquitetura para que os agentes possam ser instalados e executados.

Há disponíveis diversas plataformas já desenvolvidas para este propósito, tais como: 2APL, A-Globe , ABLE , ACT-R, ADK, AgentBuilder , Agent Factory , AgentScape, Aglets, April , CArtAgO, CHAP, CORMAS, Cougaar, Cybele, DIET Agents, Emerald, FIPA-OS, GOAL, Golem , Grasshopper , INDUS , JaCaMo , JACK , Jackdaw , JADE , JADEx , Janus , Jason , JAM , JIAC , Lisa , Lost Wax, LS/TS, MadKit , Magentix , Micro-PSI, Moise , Nuin , Open PRS, Pogamut , Practionist , RETSINA , Soar , SAGE e Semoa , SPARK.

Destas plataformas listadas foram escolhidas as seguintes para um estudo mais detalhado:

- a) Golem: (BROMURI; STATHIS, 2008) é uma plataforma de agente de código aberto que pode ser usado para criar aplicativos de sistemas multiagente. Permite a programação de agentes cognitivos com recursos do ambiente que estão estruturados como objetos em *containers*. A plataforma permite um desenvolvedor do sistema para a construção de uma aplicação, através da implantação de três entidades principais: Agentes (entidades ativas envolvidas em interações físicas e sociais, dentro do ambiente); Objetos (recursos passivos com os quais os agentes podem interagir) e Containers (uma parte do ambiente que possa alocar agentes e objetos com suas interações). Uma característica importante do Golem é que ele permite que os agentes de software e o ambiente em que estão situados sejam

especificados simbolicamente, usando abordagens declarativas e baseadas em lógica. Isso faz com que a implantação de um tipo específico de agentes cognitivos seja mais fácil;

- b) Jason: (BORDINI et al., 2007) é uma plataforma para o desenvolvimento de sistemas multiagente, que incorpora uma linguagem de programação orientada a agentes. A linguagem é baseada na arquitetura BDI<sup>2</sup> inspirada na linguagem *AgentSpeak*. Uma extensão da linguagem de programação orientada a agentes *AgentSpeak* é usada para programar o comportamento dos agentes individuais. Jason é desenvolvido em JAVA e permite a personalização da maioria dos aspectos de um agente ou de um Sistema Multiagente. Jason é um software tipo código aberto;
- c) Grasshopper: (MAGEDANZ et al., 2009) é uma plataforma comercial de agentes, em que a comunicação entre os componentes da plataforma é do tipo cliente/servidor. Fornece um apoio para que os agentes interajam com ambientes computacionais realizadas como objetos JAVA, recebendo alguns benefícios da integração Agente&Ambiente (A&E). Uma vez que um agente pode fornecer ou utilizar alguns serviços de outros componentes, pode ser um servidor, bem como um cliente, ou ainda os dois simultaneamente. A ideia por trás do processo de comunicação no Grasshopper é que os clientes não se referem aos servidores correspondentes, mas eles trabalham com *proxies* em vez disso.
- d) JADE: (BELLIFEMINE, 2007) (WOOLDRIDGE, 2009) é uma plataforma<sup>3</sup> implementada em linguagem JAVA. Ela simplifica a implementação de

---

<sup>2</sup> Modelo de software que implementa os principais aspectos da teoria do raciocínio prático humano de Michael Bratman, também conhecida como Crença-Desejo-Intenção (Belief – Desire – Intention).

<sup>3</sup> Ambiente pré-existente em que um pedaço de software é projetado para ser executado internamente, obedecendo as suas limitações e fazendo uso de suas instalações.

sistemas multiagente através de classes que cumprem com as especificações FIPA (*Foundation for Intelligent Physical Agents*) e através de um conjunto de ferramentas gráficas que suporta as fases de depuração e implantação. Os agentes podem migrar ou clonar a si mesmos dentro desta plataforma, usando diferentes *hosts*. JADE é um produto de código aberto distribuído pela TILAB. A estrutura da mensagem de JADE é realizada através de FIPA, que permite a definição de ontologias, linguagem, parâmetros, entre outros, informações úteis para descrever uma mensagem;

- e) Lana: (BRYCE; RAZAFIMAHEFA; PAWLAK, 2002) é uma plataforma de agentes móveis a serem executados em JADE. A principal tarefa de Lana é fornecer uma base de desenvolvimento, bem como ambiente de tempo de execução para aplicativos baseados em agentes móveis. Lana tem a sua própria linguagem de máquina virtual e orientada a objeto, que é derivado da linguagem JAVA. O *kernel*<sup>4</sup> do sistema agente de Lana dá a possibilidade de usar este sistema em dispositivos embarcados. Lana carece de padronização (não suporta FIPA, Masif, bem como línguas de comunicação do Agente), mas a intenção geral dos desenvolvedores LANA era criar uma plataforma rápida e fácil de implementar;
- f) Aglets: (LANGE; MITSURU, 1998) é um ambiente de programação para Internet de agentes em JAVA. Aglets são objetos JAVA que podem se mover de um *host* na *Internet* para outro usando a protocolo ATP (*Agent Transfer Protocol*), que tem como modelo o protocolo HTTP. A plataforma implementa a norma MASIF. A principal vantagem da plataforma é sua agilidade na execução dos códigos e disponibilidade destes (código aberto).

---

<sup>4</sup> Kernel em computação, o núcleo ou cerne, é o componente central do sistema operativo da maioria dos computadores; ele serve de ponte entre aplicativos e o processamento real de dados feito a nível de hardware.

Dentre as plataformas citadas, JADE foi a escolhida para ser aplicada no desenvolvimento deste trabalho por ser a plataforma que melhor atende aos requisitos de:

- a) Facilidade na instalação;
- b) Facilidade na operação;
- c) Disponibilidade de literatura;
- d) Forma mais clara para troca de mensagens;
- e) Facilidade de inserção e retirada de agentes da plataforma;
- f) Disponibilidade de recursos de monitoramento da plataforma.

A Tabela 3 apresenta uma comparação entre as plataformas estudadas e suas características, destacando a plataforma JADE selecionada.

**Tabela 3 Comparação entre as plataformas para desenvolvimento de Sistemas Multiagentes estudados.**

Fonte: adaptada de (PABADIS, 2006)

Características	Plataformas					
	Golem	Jason	Grasshopper	JADE	Lana	Aglets
Disponibilidade	Código aberto	Código aberto	Software livre	<b>Código aberto</b>	Código aberto	Software livre
Complexidade	Média	Média	Média	<b>Média</b>	Baixa	Baixa
Mobilidade	Não suporta	Não suporta	Suporta	<b>Suporta</b>	Não suportada	Não o suficiente
Padronização	FIPA, RMI	SACI	MAFIS, FIPA, XML, CORBA	<b>RMI, FIPA, XML</b>	RMI	Http, RMI
Segurança	Suficiente	Suficiente	Complexa	<b>Suficiente</b>	Suficiente	Não suficiente
Comunicação	Invocação de métodos	Passagem de mensagens	Invocação de métodos	<b>Passagem de mensagens</b>	Invocação de métodos	Passagem de mensagens
Múltiplas mensagens	Atendida	Atendida	Atendida	<b>Atendida</b>	Não atendida	Atendida
Linguagem de comunicação	Prosoc	AgentSpeak	JAVA	<b>JAVA</b>	JAVA	JAVA
Interface com Usuário	Não possui	Não possui	Suportada	<b>Suportada</b>	Não possui	Suportada

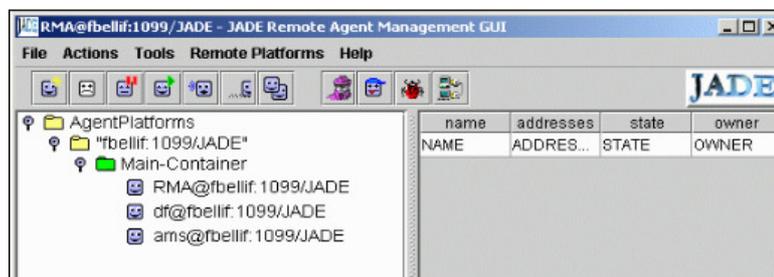
### 2.3.1 A plataforma JADE

JADE (*Java Agent DEvelopment*) é implementada em linguagem JAVA (BELLIFEMINE et al., 2007). A plataforma de agentes pode ser distribuída através de diversos computadores, sem a necessidade de compartilhar o mesmo sistema operacional e a configuração pode ser controlada através de uma interface gráfica remota. A configuração pode ser alterada, mesmo em tempo de execução, e agentes podem migrar de uma máquina para outra.

O objetivo do JADE é simplificar o desenvolvimento de Sistemas Multiagentes, através de um conjunto abrangente de serviços e agentes do próprio sistema, em conformidade com as especificações FIPA, tais como:

- a) Serviço de identificação dos agentes: reconhece cada agente que se insere no domínio e o identifica com um nome único, disponibilizando esta lista de nomes para outros agentes que desejarem consultar quais estão atuando na plataforma;
- b) Serviços de páginas amarelas: local onde os agentes podem postar seus serviços e também consultar quais agentes realizam um determinado serviço desejado;
- c) Transporte de mensagens: permite que se interfira na troca de mensagens entre os agentes, servindo como um bom recurso para identificação de problemas na implementação;
- d) Serviços de análise da comunicação: é um verificador da troca de mensagens entre os agentes que estão na plataforma;
- e) Biblioteca de protocolos de interação FIPA: disponibiliza classes com requisitos de interação, facilitando a implementação dos agentes.

A Figura 12 apresenta a interface humano-computador do sistema JADE, que auxilia na monitoração dos componentes que estão na plataforma.



**Figura 12 Tela de acesso do usuário no ambiente JADE.**

Fonte: (BELLIFEMINE et al., 2007).

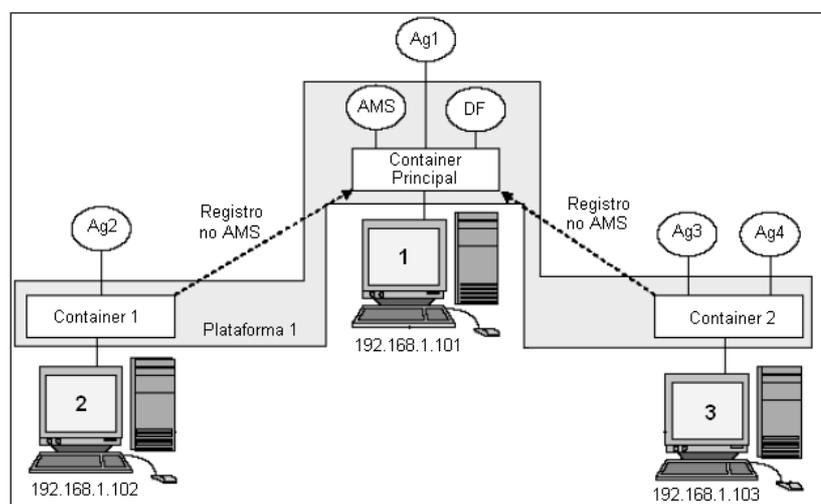
Ao instalar o JADE, este já dispõe dos recursos de interconectividade, bastando as máquinas estarem ligadas em rede, comunicando-se por TCP/IP. Quando executado o JADE em um computador, este se instalará em um “container<sup>5</sup> principal”. Dentro deste container são criados outros dois agentes (AMS e DF), sendo descritas suas funções a seguir:

- a) *Agente*: é uma instância da classe *Agent* que representa a base para a definição de agentes. Já oferece todas as interações básicas da plataforma (registro, configuração, etc.) e oferece um conjunto de métodos para a implementação do comportamento do agente JADE. No modelo computacional, um agente é multitarefa, onde os serviços são executados concorrentemente;
- b) *Agent Management System* (AMS): agente que exerce o controle sobre o acesso e o uso da plataforma e mantém a lista de identificadores dos agentes (AID) que estão na plataforma. Todo agente deve se registrar no AMS;
- c) *Directory Facilitator* (DF): oferece o serviço de páginas amarelas na plataforma, onde os agentes postam suas habilidades e onde podem buscar o endereço de outros agentes que possuem habilidades requeridas.

Somente o *container* principal possui, além do próprio agente, o AMS e o DF. Os demais agentes serão executados em “*containers*”, dentro da plataforma, podendo haver mais

<sup>5</sup> Um ambiente de execução onde os agentes instalados na plataforma JADE são hospedados e executados.

de um agente por *container*. A Figura 13 mostra um exemplo de estrutura de uma plataforma JADE.



**Figura 13 Exemplo de estrutura da plataforma JADE.**

Fonte: autor.

Nesta figura tem-se o container principal com a execução da plataforma JADE, os agentes AMS, o agente Ag1 e o agente DF. O agente Ag2 está sendo executado no container 1 e o agente Ag3 e Ag4 estão sendo executados no container 2.

A execução de um agente ocorre quando se executa uma instância dele. Neste momento o identificador do agente (AID) é informado da sua presença na plataforma. Para o exemplo da Figura 18 os comandos de instância para execução dos quatro agentes na plataforma 1 seriam:

- a) Computador 1: `java jade.Boot -gui Ag1:nome_da_classe`
- b) Computador 2: `java jade.Boot -host 192.168.1.101 -container Ag2:nome_da_classe`
- c) Computador 3: `java jade.Boot -host 192.168.1.101 -container Ag3:nome_da_classe`
- d) Computador 4: `java jade.Boot -host 192.168.1.101 -container Ag4:nome_da_classe`

Descrevendo o significado de cada comando tem-se:

- a) <java> comando para execução da JVM no computador;
- b) <jade.Boot> programa a ser executado pela JVM;
- c) <-gui> executa o *host* para hospedar recursos de AMS e DF, além dos demais agentes que se inserirem na plataforma;
- d) <-host numero\_ip> identifica onde está localizado o *host* da plataforma;
- e) <-container> cria um ambiente para hospedar um agente que não está no *host*;
- f) <nome\_agente> o *nome\_agente* é um identificador que será atribuído a este agente e que será postado no AMS, sendo-lhe agregado o endereço origem (*nome\_agente@numero\_ip:1099/JADE*);
- g) <nome\_da\_classe> o *nome\_da\_classe* identifica qual classe está sendo instanciada. Esta classe estende uma classe específica chamada *agent*, que possui as funcionalidades para implementação de agentes.

### 2.3.2 Agente em JADE

Do ponto de vista de programação, um agente JADE é uma instância da classe *Agent* (BELLIFEMINE et al., 2007), na qual os programadores ou desenvolvedores deverão escrever seus próprios agentes como extensões da classe *Agent*, adicionando comportamentos específicos de acordo com a necessidade e objetivo da aplicação. Estes comportamentos são adicionados através de um conjunto básico de métodos, utilizando as capacidades herdadas que a classe *Agent* dispõe, tais como mecanismos básicos de interação com a plataforma de agentes (registro, configuração, gerenciamento remoto, etc.). A estrutura básica de um agente programado em JAVA é apresentada na Figura 14.

```

import jade.core.Agent;
public class AgenteTeste extends Agent {
    protected void setup() {
        System.out.println("Sou o agente " + getAID().getName());
        // inserir as demais tarefas.
        doDelete();
    }
    protected void takeDown() {
        System.out.println("Agente " + getAID().getName() + "terminando");
    }
}

```

**Figura 14 Agente programado em JAVA.**

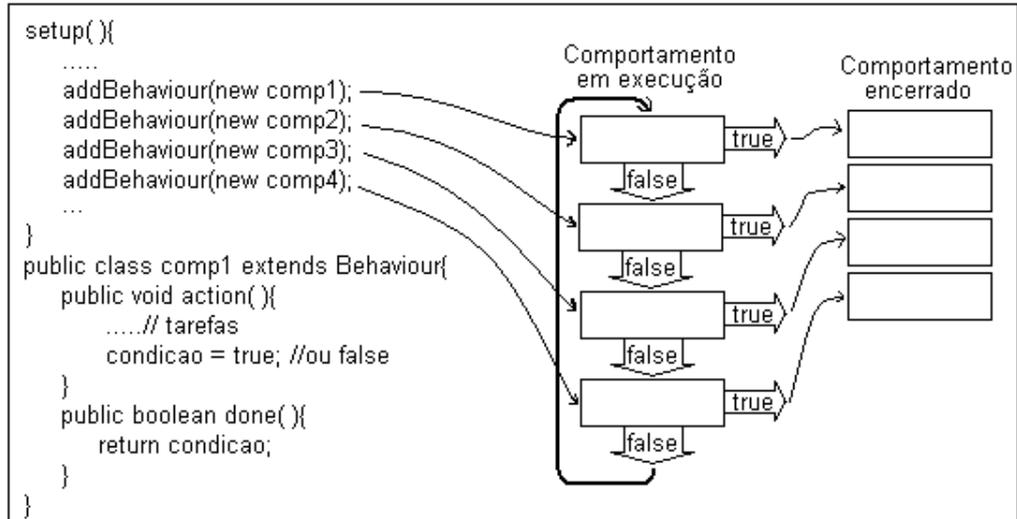
Fonte: autor.

Quando executado o agente pelo comando “*java jade.Boot -gui AgI:AgenteTeste*”, o método *setup()* será executado, e é neste ponto que as tarefas deverão ser inseridas. Um agente, mesmo que não esteja executando algo, estará ativo. Para encerrar o agente o método *doDelete()* deve ser invocado, que chamará o método *takeDown()*, encerrando o agente.

Um agente pode agregar em seu programa comportamentos específicos (*behaviours*). Os comportamentos são métodos cíclicos de execução, onde são descritas as ações que devem ser executadas, e são criados através do método *addBehaviour(new nome\_classe\_comportamento)*. Comportamentos são definidos como extensões da classe *Behaviour*, e nelas são definidos dois métodos principais, que são:

- a) *void action()*: define as ações a serem executadas em um ciclo;
- b) *boolean done()*: método que, se retornar *true*, encerra a execução do comportamento, se retornar *false*, coloca novamente o comportamento na fila de execução.

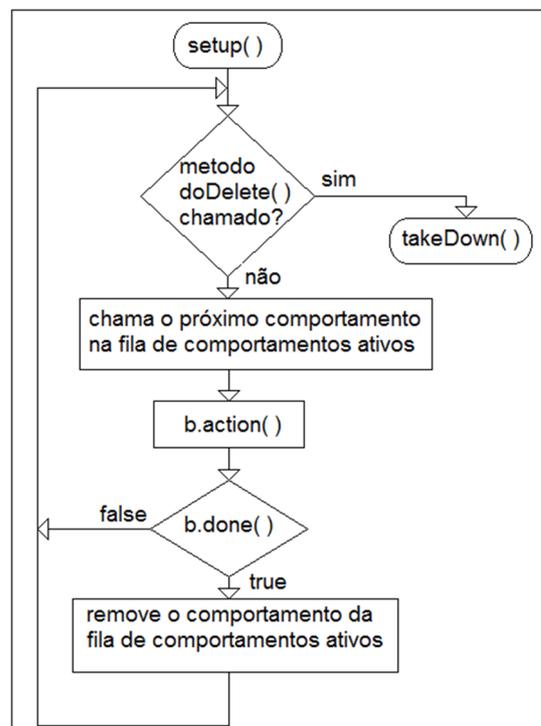
Quando um comportamento é adicionado, ele é encaminhado para uma fila de comportamentos em execução. Ao chegar sua vez, ele é executado e ao chegar ao final da execução o método *done()* é monitorado. Se ele retornar *false*, o comportamento se mantém na fila para ser executado na próxima vez que o fluxo de processos passar por ele. Caso o método retorne *true*, então é deslocado para a pilha de comportamentos encerrados. A Figura 15 demonstra este processo.



**Figura 15** Formato de execução dos comportamentos no JADE.

Fonte: autor.

Na Figura 16 está representado o fluxograma de execução de um agente. Nota-se que um comportamento é executado por vez, dentro do método de *setup()*.



**Figura 16** Fluxograma de execução de um agente.

Fonte: autor.

Na implementação dos comportamentos tem-se a classe *Behaviour*, que é uma classe abstrata do JADE, disponível no pacote *jade.core.behaviours*. Uma classe abstrata é uma classe que possui alguns métodos implementados e outros não, e não pode ser instanciada diretamente. Ela tem como finalidade prover a estrutura de comportamentos para os agentes JADE os implementarem (SILVA, 2003). Nela encontram-se disponíveis comportamentos com funcionalidades próprias, onde se destacam os seguintes:

- a) *SimpleBehaviour*: comportamento que estende a classe *Behaviour* (classe *jade.core.behaviours.Behaviour*), composta dos métodos *action()*, que promove as chamadas de tarefas, e *done()* que define se o comportamento irá continuar ou será encerrado;
- b) *OneShotBehaviour*: comportamento que estende a classe *OneShotBehaviour* (classe *jade.core.behaviours.OneShotBehaviour*), composta somente do método *action()*. O método *done()* retorna *true*, fazendo com que o comportamento seja executado uma única vez;
- c) *CyclicBehaviour*: comportamento que estende a classe *CyclicBehaviour* (classe *jade.core.behaviours.CyclicBehaviour*), composta somente pelo método *action()*. O método *done()* retorna *false*, fazendo com que o seja repetido à medida que o *setup()* chamar a execução do comportamento;
- d) *WakerBehaviour*: comportamento que estende a classe *WakerBehaviour* (classe *jade.core.behaviours.WakerBehaviour*), comportamento que espera um determinado período de tempo (em milisegundos) para efetivamente executar a tarefa;
- e) *TickerBehaviour*: comportamento que estende a classe *TickerBehaviour* (classe *jade.core.behaviours.TickerBehaviour*), comportamento que executa uma tarefa periodicamente em intervalos de tempo constantes (em

milissegundos). Este comportamento não possui término previsto, pois seu método *done()* retorna *false*.

## 2.4 COMUNICAÇÃO ENTRE AGENTES

A fim de que o controlador lógico programável possa negociar seus serviços com outros controladores, agora já incorporados a agentes, são necessários então modelos de comunicação. A comunicação entre os agentes passa a ser um fundamento muito importante para o trabalho em si.

A *Foundation for Intelligent Physical Agents* (FIPA) foi criada em 1996 por uma associação internacional sem fins lucrativos para desenvolver uma coleção de normas relativas à tecnologia de agentes de *software*. As composições iniciais, um grupo de acadêmicos e organizações industriais, elaboraram um conjunto de estatutos para orientar a criação de um conjunto de especificações padrão para tecnologias de agentes de *software*. Naquela época os agentes de *software* já estavam muito bem conhecidos na comunidade acadêmica, mas, até a data, só recebiam atenção limitada de empresas comerciais, além de uma perspectiva exploratória. O consórcio concordou em produzir normas que formariam a base de uma nova indústria, sendo utilizáveis através de um vasto número de aplicações.

O padrão FIPA é subdividido em um conjunto de 25 especificações. Além dessas especificações, o padrão busca fornecer uma linguagem para a comunicação entre os agentes com rigorosas fórmulas semânticas. *FIPA Agent Communication Language* (FIPA-ACL) inclui cerca de 20 tipos de comunicação e todos eles são baseados na forma como os seres humanos se comunicam. Basicamente, trata-se de um formato onde uma mensagem ou comando é enviada/recebida por um agente. Isto permite que dois agentes, mesmo que com concepções diferentes (plataforma, protocolo, camada física, entre outras), possam se comunicar.

Segundo consta em BELLIFEMINE (2007) o padrão FIPA encapsula uma mensagem, sendo dividido em camadas, em que cada uma destas camadas é responsável por uma parte no processo de comunicação. As subcamadas do padrão FIPA são:

- a) Subcamada 1 (Transportes): responsável pelo protocolo de transporte. FIPA definiu protocolos de transporte de mensagens para IIOP, WAP e HTTP;
- b) Subcamada 2 (Codificação): em vez de enviar simples *byte* em mensagens codificadas, FIPA define várias representações de mensagens para o uso de estruturas de nível mais elevado de dados, incluindo XML *String*, e-Bit Eficiente. O último é destinado a ser utilizado quando comunicando através de conexões com baixa largura de banda;
- c) Subcamada 3 (Conteúdo de mensagens): a estrutura da mensagem é especificada independente da codificação particular, para incentivar a flexibilidade;
- d) Subcamada 4 (Ontologia): o termo individual contido na carga ou no conteúdo de uma mensagem FIPA pode ser explicitamente referenciado a um modelo de aplicação específica conceitual ou ontologia;
- e) Subcamada 5 (Conteúdo): o teor real de mensagens FIPA pode ser de qualquer forma, mas FIPA definiu orientações para a utilização de fórmulas gerais lógicas e operações algébricas. A linguagem mais usada para expressar conteúdo é FIPA-SL;
- f) Subcamada 6 (Ato comunicativo): a classificação de uma mensagem simples em termos de ação ou ato performativo<sup>6</sup>. Exemplos incluem informar, solicitar ou concordar;

---

<sup>6</sup> Diz-se de um enunciado que se dá ao mesmo tempo em que a ação por ele apresentada; palavra e ato coincidem em seu significado.

- g) Subcamada 7 (Protocolo de interação): raramente as mensagens são trocadas de forma isolada, mas sim fazem parte de alguma sequência de interação. FIPA define protocolos de interação, especificando as sequências de mensagens, como um pedido que descreve uma parte de outro pedido, que por sua vez devem concordar ou recusar-se a cumprir.

A Linguagem ACL, que agrega ao padrão FIPA a troca de mensagens, é assíncrona e usa alguns campos principais, citados a seguir:

- a) *Sender*: AID do agente que envia a mensagem;
- b) *Receivers*: lista de AID dos agentes destinos da mensagem;
- c) *Performative*: ato de fala que indica o significado da mensagem (uma informação, um questionamento, uma resposta, um aviso, entre outras);
- d) *Content*: o real conteúdo da mensagem;
- e) *Language*: sintaxe usada para expressar o conteúdo;
- f) *Ontology*: denota a semântica dos itens do conteúdo;
- g) Outros campos de controle de conversação como: *conversation-id*, *reply-with*, *in-reply-to*, *reply-by*.

Toda a troca de mensagens realizada no JADE é feita através de métodos próprios e com o uso de instâncias da classe *ACLMessage*. Esta classe possui um conjunto de atributos que estão em conformidade com as especificações FIPA, implementando o padrão FIPA-ACL. Assim, um agente que pretenda enviar uma mensagem deve instanciar um objeto da classe *ACLMessage*, preenchê-lo com as informações necessárias e chamar o método *send()* da classe *Agent*. Caso for receber mensagens, o método *receive()* da classe *Agent* deve ser chamado.

Para preencher as informações necessárias na instância de um objeto *ACLMessage*, os métodos seguintes devem ser utilizados:

- a) *public void addReceiver(AID idAgente)*: adiciona o AID de um agente como receptor ou destinatário da mensagem. Determina quem receberá a mensagem;
- b) *public void removeReceiver(AID idAgente)*: remove o AID do agente da lista de receptores;
- c) *public ACLMessage createReply()*: cria uma nova mensagem ACL de resposta à determinada mensagem. Assim, o programador só necessita definir o ato comunicativo (*communicate-act*) e o conteúdo da mensagem;
- d) *public static String[] getAllPerformativeNames()*: retorna uma lista de *strings* com a lista de todos os atos performativos;
- e) *public Iterator getAllReceiver()*: retorna um objeto *iterator* contendo todos os AIDs dos agentes receptores da mensagem;
- f) *public String getContent()*: retorna uma *string* contendo o conteúdo da mensagem;
- g) *public void setContent(String conteúdo)*: define o conteúdo da mensagem a ser enviada;
- h) *public void setOntology (String ontologia)*: define a ontologia da mensagem.

A Figura 17 apresenta um código exemplo de envio de mensagem FIPA-ACL com JADE.

```

public class AgenteEnvia extends Agent{
    protected void setup( ){
        ACLMessage msg = new ACLMessage( ACLMessage.INFORM );
        msg.addReceiver( new AID( "AgenteRecebe", AID.ISLOCALNAME ) );
        msg.setContent( "Oi amigo agente!" );
        send(msg);
    }
}

```

**Figura 17 Código exemplo de envio de mensagem FIPA-ACL.**

Fonte: autor.

A Figura 18 apresenta um código exemplo de recepção de mensagem FIPA-ACL com JADE.

```

public class AgenteRecebe extends Agent{
    protected void setup(){
        addBehaviour(new ReceberMsg( ));
    }
    public class ReceberMsg extends CyclicBehaviour{
        public void action() {
            MessageTemplate mt;
            mt = MessageTemplate.MatchPerformative(ACLMessage.INFORM);
            ACLMessage msg = myAgent.receive(mt);
            if(msg != null) {
                System.out.println(msg.getContent());
            } else block();
        }
    }
}

```

**Figura 18 Código exemplo de recepção de mensagens FIPA-ACL.**

Fonte: autor.

#### 2.4.1 Protocolos especificados para troca de mensagens

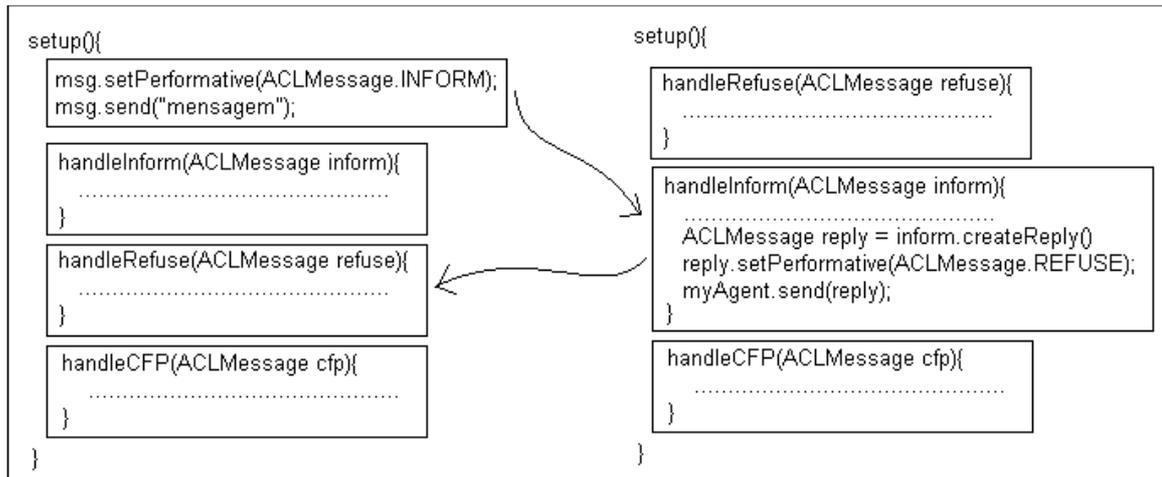
Outro meio de enviar ou receber mensagens em JADE é através do uso das classes de comportamentos *SenderBehaviour* e *ReceiveBehaviour*. Fato que torna possível que as trocas de mensagens sejam escalonadas como atividades independentes de um agente.

As interações se dão de forma estanque, onde cada método é executado segundo o ato performativo da mensagem que chega. Estes atos podem ser descritos, segundo *FIPA-ACL Communicative Act Library Specification SC00037*, como:

- a) *Accept Proposal* – indica o aceite de uma proposta de uma mensagem anterior;
- b) *Agree* – indica a concordância com uma ação proposta;
- c) *Cancel* – indica a um agente que não tem a intenção de executar alguma ação;

- d) *Call for Proposal* – indica a um agente uma proposta para realização de uma tarefa, com um convite;
- e) *Confirm* – indica a um agente a confirmação de algo que foi proposto;
- f) *Disconfirm* – indica a um agente que o que for proposto não está confirmado;
- g) *Failure* – indica que a ação proposta por um agente falhou;
- h) *Inform* – indica uma mensagem de informação a outro agente;
- i) *Not Understood* – indica que uma mensagem recebida não foi devidamente entendida;
- j) *Propagate* – indica a um agente que uma mensagem deve ser propagada a outro agente;
- k) *Propose* – indica que está sendo emitida uma proposta para realização de uma ação;
- l) *Refuse* – indica que uma proposta de ação foi abandonada;
- m) *Reject Proposal* – indica que uma proposta de ação foi rejeitada;
- n) *Request* – indica a requisição de alguma ação a um agente;
- o) *Subscribe* – indica uma notificação a um agente.

A Figura 19 apresenta a estrutura do protocolo FIPA-ACL especificado por SC00037, em que cada método é executado de forma independente, segundo o ato performativo da mensagem que é lida.



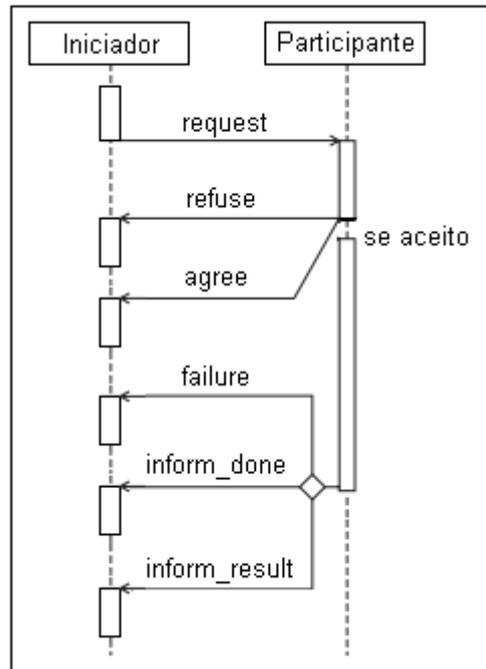
**Figura 19** Estrutura do protocolo FIPA-ACL.

Fonte: autor.

Há alguns protocolos descritos no padrão FIPA, sendo que os mais usuais são os protocolos *FIPA Request Interaction Protocol Specification (SC00026)* e *FIPA Contract Net Interaction Protocol Specification (SC00026)*.

#### 2.4.2 Protocolo FIPA *Request*.

Representado pelo diagrama de sequência na Figura 20, o protocolo de interação FIPA *Request* permite um agente (o iniciador) solicitar a outro (o participante) a execução de uma ação. O participante processa o pedido e toma uma decisão de aceita-lo ou recusa-lo. Se as condições indicam que um acordo pode ocorrer, então o participante comunica um acordo (*AGREE*), caso contrário uma notificação de declínio é enviada (*REFUSE*).



**Figura 20 Protocolo FIPA Request.**

Fonte: (BELLIFEMINE, 2007).

Uma vez que o participante aceitou a requisição, ele processa a ação solicitada e envia uma resposta final, que pode resultar em três situações: se uma falha ocorrer ao realizar a ação, uma mensagem *FAILURE* será enviada; se tudo ocorrer bem, uma mensagem *INFORM\_DONE* é enviada para indicar tão somente que a ação foi realizada; e há o caso onde não só a informação de ação realizada é enviada, mas também o resultado desta ação, enviando uma mensagem *INFORM\_RESULT*.

#### **2.4.3 Protocolo FIPA Contract Net.**

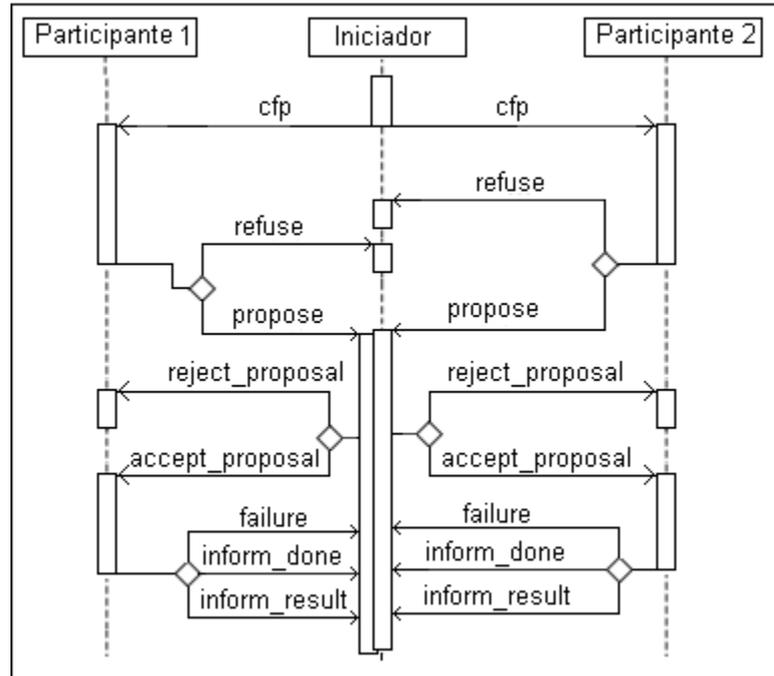
Como um exemplo de um protocolo de interação mais complexo, o protocolo FIPA *Contract Net* descreve o caso de um agente (o Iniciador), que deseja ter alguma tarefa realizada por um ou mais agentes (os participantes), para otimizar uma função que caracteriza a tarefa. Esta característica é geralmente expressa como um custo, mas também poderia ser o tempo mais rápido para a conclusão, a distribuição das tarefas, a ordem de retorno da resposta, entre outras. Para uma dada tarefa, qualquer número de participantes pode responder com uma proposta. No parágrafo seguinte este procedimento é descrito em detalhes.

O iniciador solicita propostas a outros agentes através da emissão de um convite à apresentação de propostas (*Call for Propose - CFP*), que especifica a tarefa e as condições dos locais sobre a execução desta.

Os participantes que recebem o convite à apresentação de propostas são vistos como potenciais candidatos a serem contratados e são capazes de gerar respostas. As propostas dos participantes incluem as pré-condições do participante, que pode ser o preço, o tempo de execução, o tempo até o início da realização da tarefa, quantidades de tarefas realizadas concomitantemente, entre outras. Os participantes também podem se recusar a enviar uma proposta.

As tratativas continuam com os participantes que apresentaram proposta para realização (*PROPOSE*), os que enviaram *REFUSE* deixam a tratativa. Uma vez que o prazo cessa, o iniciador avalia as propostas e seleciona os agentes para executar a tarefa, com base em seus critérios previamente definidos. Um, vários ou nenhum agente podem ser escolhidos. Aos agentes da proposta selecionada será enviado um aceite à proposta (*ACCEPT\_PROPOSAL*) e os agentes restantes receberão uma rejeição à proposta (*REJECT\_PROPOSAL*). As propostas são obrigatórias para o participante, de modo que uma vez que o Iniciador aceita a proposta do participante, este adquire o compromisso de executar a tarefa. Uma vez que o participante tenha concluído a tarefa, ele envia uma mensagem de conclusão para o iniciador, na forma de uma mensagem de tarefa realizada (*INFORM\_DONE*), ou em uma versão mais explicativa na forma de uma mensagem de tarefa realizada com os devidos resultados (*INFORM\_RESULT*). No entanto, se o participante não concluir a tarefa, uma mensagem de falha é enviada (*FAILURE*).

A Figura 21 apresenta o diagrama de um protocolo FIPA *Contract Net* com dois participantes, na sua forma geral.



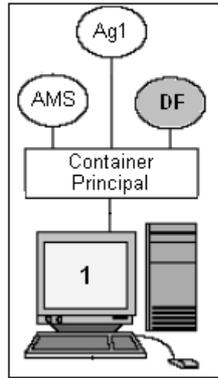
**Figura 21 Protocolo FIPA *Contract Net*.**

Fonte: (BELLIFEMINE, 2007).

Este protocolo requer que o iniciador saiba quando ele recebeu todas as respostas. No caso de um participante não responder ou propor uma mensagem errada, o iniciador pode ficar esperando indefinidamente. Para se proteger contra isso, o protocolo FIPA *Contract Net* inclui um prazo para que as respostas sejam recebidas pelo iniciador. As propostas recebidas após o prazo serão automaticamente rejeitadas. O prazo é especificado pelo parâmetro de resposta no objeto *ACLMensagem*.

#### 2.4.4 Serviço de páginas amarelas em JADE.

Durante a criação de um agente (instância) no container principal, mais dois agentes são criados pelo JADE, que são o *Agent Management System* - AMS e o *Directory Facilitator* – DF. A Figura 22 apresenta esta estrutura.

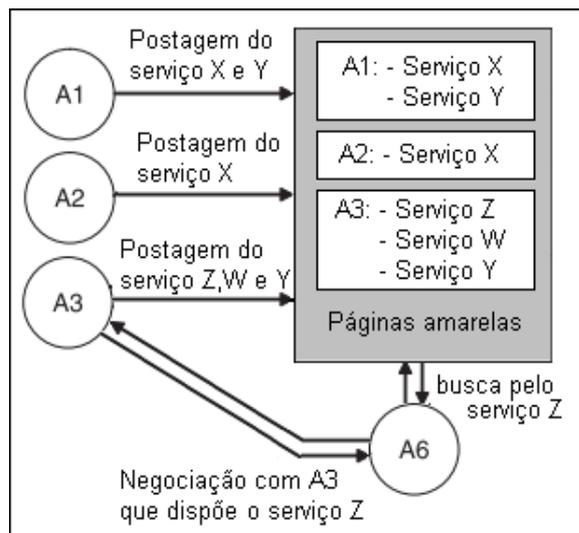


**Figura 22 Directory Facilitator (DF) em JADE.**

Fonte: (BELLIFEMINE, 2007).

O *Directory Facilitator* oferece o serviço de páginas amarelas na plataforma, onde os demais agentes postam suas habilidades e onde eles podem buscar o endereço de agentes que possuem as habilidades desejadas.

O serviço de páginas amarelas permite aos agentes publicar descrições de um ou mais serviços que prestam, a fim de que outros agentes possam facilmente descobri-los e explorá-los. Isto é representado na Figura 23. Qualquer agente pode tanto registrar (publicar) serviços quanto procurar (descobrir) serviços. As inscrições, modificações, exclusão de registros e pesquisas pode ser realizadas em qualquer momento durante a vida de um agente.



**Figura 23 Estrutura de postagem no DF.**

Fonte: (BELLIFEMINE, 2007).

Um agente que pretenda publicar um ou mais serviços deve fornecer ao DF uma descrição, que inclui uma lista de serviços prestados e, opcionalmente, a lista de idiomas e ontologias que outros agentes devem usar para interagir com ele. As classes *DFAgentDescription*, *ServiceDescription* e *Property*, incluídas no *jade.domain.FIPAAgent*, tem a função de gerenciar os pacotes e representar as abstrações.

Para publicar um serviço o agente deve criar uma descrição e chamar o método *register()*, método estático da classe *DFService*. A Figura 24 apresenta um código exemplo da operação de registro de serviço no DF.

```
DFAgentDescription dfd = new DFAgentDescription();
dfd.setName(getAID());
ServiceDescription sd1 = new ServiceDescription();
sd1.setType("montagem");
sd1.setName("montar azul");
dfd.addServices(sd1);
try{DFService.register(this,dfd);
  } catch (FIPAException fe) {
    fe.printStackTrace();
  }
```

**Figura 24 Código exemplo para postagem no DF.**

Fonte: Autor.

A busca de serviços ocorre de forma semelhante, onde as características do serviço são preenchidas na instância da classe *ServiceDescription*, e o DF retorna um objeto *Iterator* com a lista de agentes que são capazes de realizar o serviço solicitado. A Figura 25 apresenta um código exemplo para busca de um serviço no DF.

```

DFAgentDescription template = new DFAgentDescription();
ServiceDescription sd = new ServiceDescription();
sd.setType("montagem");
template.addService(sd);
try{DFAgentDescription[ ] result = DFService.Search(this, template);
    for( int i = 0; i < result.length ; i++) {
        String out = result[ i ].getName().getLocalName() + " Faz ";
        Iterator iter = result[ i ].getAllServices();
        while( iter.hasNext() ) {
            ServiceDescription SD = (ServiceDescription) iter.next();
            out += " " + SD.getName();
        }
        System.out.println(out);
    }
} catch (FIPAException e) {e.printStackTrace();}

```

**Figura 25 Código exemplo para busca de serviços no DF.**

Fonte: autor.

Sabido os conceitos tecnológicos importantes para o entendimento do trabalho, faz necessário compreender previamente os conceitos de gestão de sistemas produtivos, uma vez que o controlador lógico programável a que se destina a interação do agente, estará atuando neste meio produtivo, participando destes conceitos de sistemas de produção industrial.

Manufatura Integrada por Computador, Planejamento de Recurso Corporativo, Sistema de Gerenciamento da Produção e Sistemas Auto-organizáveis de produção são conceitos em que regem a produção industrial, produção esta comandada por controladores lógicos programáveis, portanto, o entendimento destes conceitos de gestão de sistemas produtivos é importante para o entendimento do trabalho.

É o que será visto no capítulo seguinte.

### **3 PRINCIPAIS SISTEMAS DE GESTÃO DA MANUFATURA**

A gestão da produção leva em conta sistemas que possam servir como ferramentas para organizar o processo produtivo, atender aos imprevistos operacionais da produção, propiciar o melhor fluxo de processo para otimizar os custos de produção e melhorar a utilização dos recursos disponíveis.

A história dos sistemas produtivos industriais aponta várias teorias para sistemas de gestão da produção. Este trabalho será focado em 4 sistemas importantes e bastante usuais no meio industrial: Manufatura Integrada por Computador, Planejamento de Recursos Colaborativos, Sistema de Gerenciamento da Produção e Sistemas Auto-organizáveis.

#### **3.1 MANUFATURA INTEGRADA POR COMPUTADOR - CIM (*COMPUTER INTEGRATED MANUFACTURING*)**

Com a visão sobre o processo de manufatura, entende-se um CIM como a união dos dispositivos que possuem um gerenciamento local, interligados a um computador central que gerencia o fluxo do processo e a ordem com que os algoritmos de controle serão executados em cada máquina.

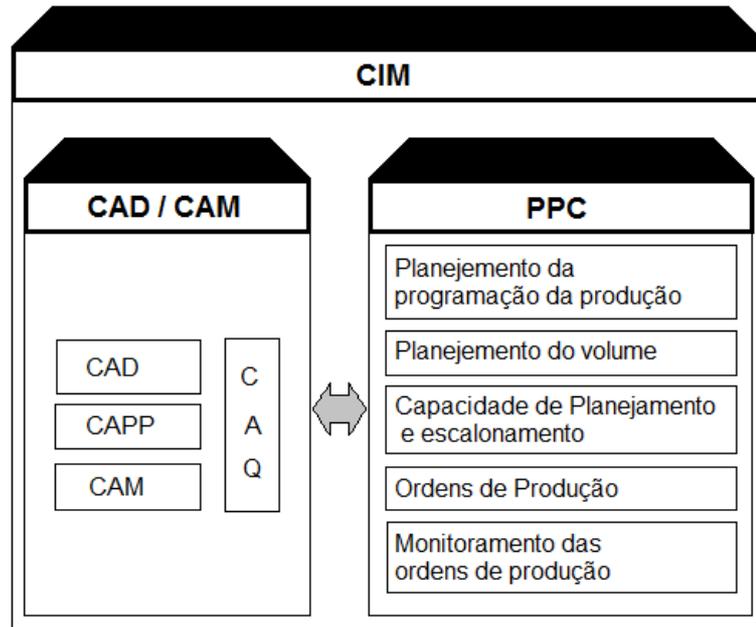
O conceito de CIM é mais amplo e visa à integração entre todas as etapas do processo: vendas, suprimentos, projeto e desenvolvimento, produção, expedição e pós-vendas. Segundo (LEITÃO et al., 2001), o paradigma CIM consiste na integração de todas as atividades da empresa por meio da utilização das tecnologias da informação, como: bancos de dados, redes, aplicativos, entre outros, que permitem a troca e o compartilhamento de dados entre as unidades da empresa e suas aplicações. Para (FLEISCHHAUER, 1996) a manufatura integrada por computador é o eficiente uso da tecnologia de informação em manufatura para aumentar a produtividade e eficiência de empresas modernas.

Numa visão global e genérica, pode-se estabelecer que uma Manufatura Integrada por Computador tem seu início com o planejamento da produção (projeto do produto, do processo e da estimativa de quantidades a produzir), continua com a programação (definição precisa de produtos a produzir no período, cálculo de necessidades de material, estabelecimento de prazos e capacidades e sequenciamento), aciona a produção e termina no controle, através de módulos de captação de dados de quantidade e qualidade da produção, havendo, ainda, a possibilidade de trabalhar com funções de controle de qualidade.

Assim, segundo (SCHEER, 2012), um CIM é composto pelos módulos seguintes:

- a) Projeto Assistido por Computador (CAD – *Computer Aided Design*);
- b) Planejamento Assistido por Computador (CAPP – *Computer Aided Process Planning*);
- c) Manufatura Assistida por Computador (CAM – *Computer Aided Manufacturing*);
- d) Controle de Qualidade Assistida por Computador (CAQ – *Computer Aided Quality*);
- e) Controle de Produção e Planejamento (PPC – *Production Planning and Control*).

Estes módulos podem ser observados na Figura 26 a seguir.



**Figura 26 Módulos que compõe um CIM.**

Fonte: adaptado de (SCHEER, 2012)

O módulo Projeto Assistido por Computador (CAD) representa a atividade de projeto que envolvem o uso do computador para criar, modificar ou documentar um projeto de engenharia. O CAD não deve ser visto como instrumento restrito ao projeto ou desenho, sendo uma forma de integração entre projetistas e respectivas equipes e demais setores da empresa, inclusive clientes e fornecedores.

Já o Planejamento do Processo Assistido por Computador (CAPP) tem a função de estabelecer o roteiro ou o processo de fabricação de um produto. É a determinação sistemática dos métodos, através dos quais um produto deve ser fabricado. Trata-se de determinar e selecionar máquinas, ferramentas, instruções de trabalho e demais condições necessárias à transformação dos pedidos em produtos finais. No CAPP são listados os processos que são capazes de serem realizados, a sequência de operação que o produto vai seguir, a distribuição do trabalho pelas máquinas, o cálculo dos tempos de fabricação, o dimensionamento das sobras de materiais e a programação das máquinas para a execução do processo estabelecido.

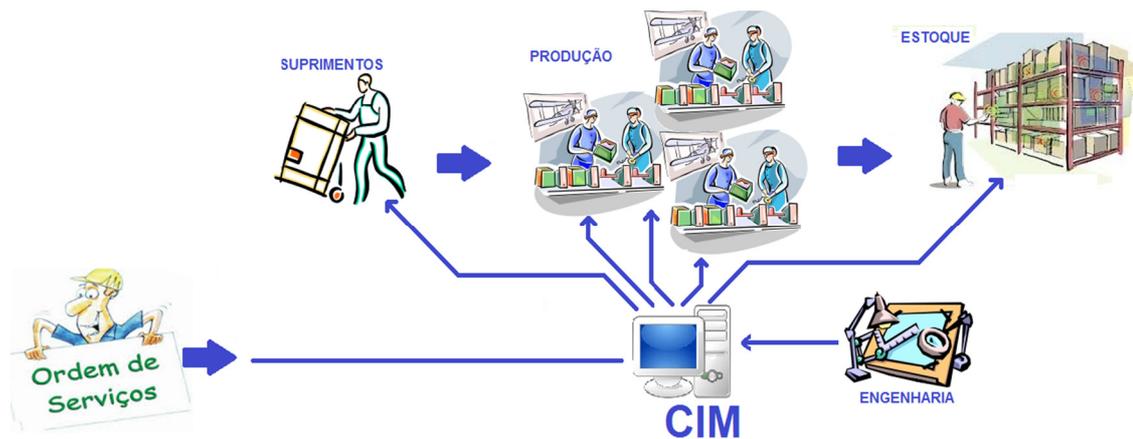
O módulo de Manufatura Assistida por Computador (CAM) é a aplicação da informática e da tecnologia de informação e comunicações ao sistema de produção, no sentido de eliminar a perda de tempo inerentes à manipulação e decisão do ser humano. O CAM, enquanto auxílio à produção, pode se restringir ao controle ou ser extensivo ao planejamento, e ainda auxiliar na monitoração dos recursos de produção.

O módulo de Qualidade Assistida por Computador (CAQ) constitui-se de um acompanhamento desde a chegada dos insumos, passando pelo processo produtivo, estendendo-se até a saída do produto acabado. Este acompanhamento garante a qualidade do produto através de seus pontos de inspeção e permite que um processo de manufatura possa ser monitorado desde as suas etapas iniciais.

Os módulos que compõem um CIM interagem, permitindo a troca de informações do processo de manufatura programado. O sistema de produção inicia-se pela elaboração do projeto mediante o auxílio de sistemas CAE e CAD. Pelo PPC tem-se a geração de lista de materiais e seus custos. O CAPP trata do roteiro de produção. E por fim o CAM implementa a manufatura, segundo informações armazenadas em seu banco de dados relativas ao produto.

Um sistema integrado de manufatura traz o controle sobre o processo bem definido e a possibilidade de variação do produto no mesmo sistema de controle da produção, uma vez definida no CAPP. Suas estações são autônomas, o que permite que o sistema continue em funcionamento mesmo com a retirada de uma estação.

A Figura 27 apresenta a estrutura CIM em um sistema produtivo.



**Figura 27 Estrutura CIM no sistema produtivo.**

Fonte: Autor.

A partir da ordem de serviço o CIM organiza a produção, garantindo os suprimentos, o arranjo produtivo, os programas para as máquinas que se fizerem necessários e o local de estoque para saída do produto acabado.

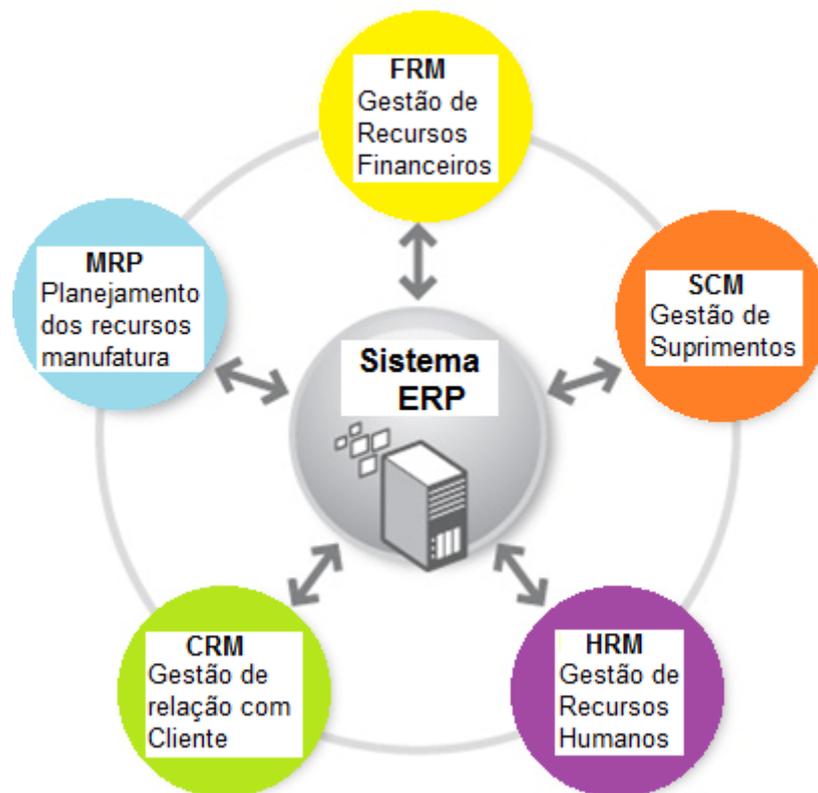
Algumas características levantadas por (BUSSMANN, 1998), como: flexibilidade, possuir em uma hierarquia de controle fixa, a reconfiguração e ampliação dos sistemas existentes são difíceis de serem alcançadas no CIM. O desempenho da produção não é mantido quando o sistema está fora das condições normais. Mas a capacidade de integrar todas as partes de um projeto, da sua concepção a sua manufatura, leva o CIM a um conceito de manufatura amplamente aceito no meio industrial.

### **3.2 O PLANEJAMENTO DE RECURSOS COLABORATIVOS – (ERP - *ENTREPRISE RESOURCE PLANNING*)**

O Planejamento de recurso corporativo é um sistema de informação que integra todos os dados e processos de uma organização em um único sistema. A integração pode ser vista sob a perspectiva funcional (sistemas de finanças, contabilidade, recursos humanos, fabricação, marketing, vendas, compras, entre outros) e sob a perspectiva sistêmica (sistema de

processamento de transações, sistemas de informações gerenciais, sistemas de apoio a decisão, entre outros).

Segundo aponta SLACK; CHAMBERS; JOHNSTON, (2009) o conceito ERP abrange módulos básicos que se relacionam, como podem ser observados na Figura 28 a seguir.



**Figura 28** Módulos básicos que compõe um ERP.

Fonte: adaptado de (SLACK; CHAMBERS; JOHNSTON, 2009).

A proposta de um ERP é que todos os recursos que a produção de um determinado produto requeira sejam planejados de forma a serem requisitados de forma escalonada. Além da disposição do recurso, o planejamento de produto também obtém informações quanto ao custo de cada recurso, sua disposição e prazos de entrega. Assim, a manufatura de um produto pode sofrer modificações em seu processo em função da disponibilidade de recursos e obter minimização dos custos.

O conceito ERP é implementado através de plataformas de software que interagem com os diversos departamentos da empresa, o que possibilita a automação e armazenamento de todas as informações do negócio.

A fim de entender melhor como o conceito funciona, o ERP pode ser visto como um grande banco de dados, com informações que interagem e se realimentam. Assim, o dado inicial sofre uma mutação de acordo com seu estado, como a ordem de vendas que se transforma no produto final alocado no estoque da companhia. Outro fator é a tomada de decisões, que também ganha outra dinâmica. Imagine uma empresa que por alguma razão, talvez uma mudança nas normas de segurança, precise modificar aspectos da fabricação de um de seus produtos. Com o ERP, todas as áreas corporativas são informadas e se preparam de forma integrada para o evento, das compras à produção, passando pelo almoxarifado e chegando até mesmo à área de *marketing*, que pode assim ter informações para mudar algo nas campanhas publicitárias de seus produtos. E tudo realizado em menos tempo do que seria possível sem a presença do sistema.

Os sistemas integrados dão às empresas a flexibilidade para responder rapidamente as solicitações dos clientes e, ao mesmo tempo, produzir e manter em estoque apenas o necessário para atender aos pedidos existentes. Sua capacidade de tornar a expedição mais veloz e precisa, minimizar os custos e aumentar a satisfação do cliente também gera mais lucratividade às empresas.

Algumas das vantagens da implementação de um ERP em uma empresa podem ser listadas como:

- a) Qualidade e eficácia para produção;
- b) Redução de custos com a otimização dos processos;
- c) Eliminação do uso de recursos humanos para a troca de informações entre departamentos;

- d) Melhoria do processo de tomada de decisão;
- e) Eliminação da redundância de atividades;
- f) Redução do tempo de resposta às necessidades de mercado;
- g) Redução de estoque de materiais;
- h) Redução da carga de trabalho, pois atividades repetitivas podem e devem ser automatizadas.

Como desvantagens se apresentam:

- a) A utilização do ERP por si só não torna uma empresa verdadeiramente integrada;
- b) Altos custos, que muitas vezes não atendem a relação custo/benefício;
- c) Dependência do fornecedor do pacote de software;
- d) A adoção de melhores práticas aumenta o grau de imitação e padronização entre as empresas de um segmento;
- e) Torna os módulos dependentes uns dos outros, pois cada departamento depende das informações do módulo anterior. Logo, as informações têm que ser constantemente atualizadas, uma vez que são em tempo real, ocasionando maior trabalho;
- f) Inserção de dados não confiáveis, quando é necessário o *input* pelo usuário;
- g) Dificuldade de repasse da cultura organizacional aos funcionários.

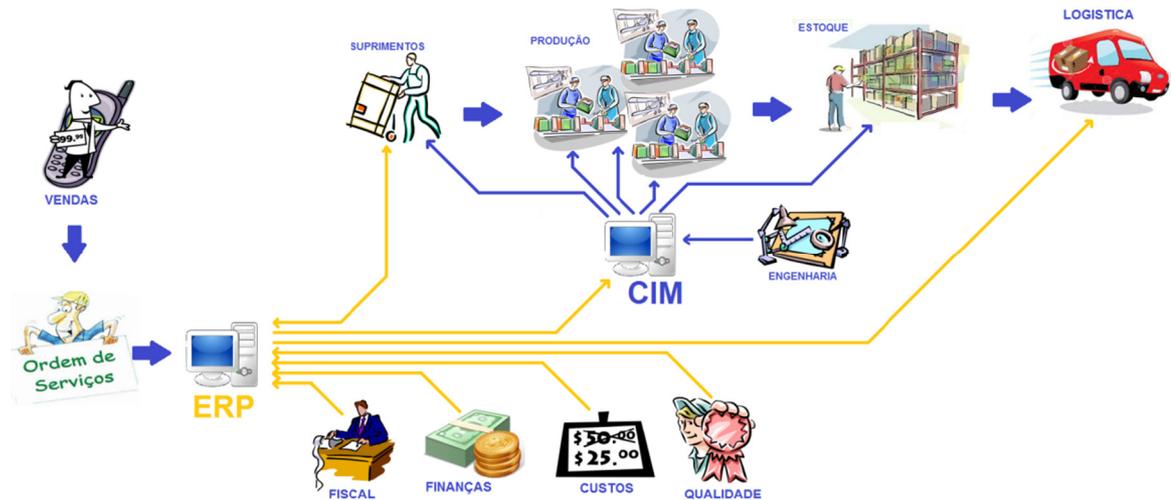
No mercado há vários softwares que implementam o conceito ERP no processo produtivo, a destacar os que são listados na Tabela 4, que apresenta uma lista dos softwares ERP e suas características:

Tabela 4 Lista de software ERP.

Software	Licença	Origem	Homepage
KCMS Intelligent Solutions	Proprietária	Brasil	<a href="http://www.kcms.com.br/">http://www.kcms.com.br/</a>
Mega Sistemas	Proprietária	Brasil	<a href="http://www.mega.com.br/home">http://www.mega.com.br/home</a>
CIGAM	Proprietária	Brasil	<a href="http://www.cigam.com.br/">http://www.cigam.com.br/</a>
DON Corporate ERP	Proprietária	Brasil	<a href="http://portal.vilessoft.com/">http://portal.vilessoft.com/</a>
ERP Cloud	Proprietária	Brasil	<a href="http://old.betalabs.com.br/site">http://old.betalabs.com.br/site</a>
ERP5	GPL	França	<a href="https://www.erp5.com/">https://www.erp5.com/</a>
FreedomERP	GPL	Brasil	<a href="http://www.freedom.org.br/">http://www.freedom.org.br/</a>
Openbravo	GPL	Espanha	<a href="http://www.openbravo.com/">http://www.openbravo.com/</a>
Odoo	GPL	EUA	<a href="https://www.odoo.com/">https://www.odoo.com/</a>
TOTVS Protheus	Proprietária	Brasil	<a href="https://www.totvs.com.br">https://www.totvs.com.br</a>
SAP AG	Proprietária	Alemanha	<a href="http://go.sap.com/index.html">http://go.sap.com/index.html</a>
Stoq	GPL	Brasil	<a href="http://www.stoq.com.br/">http://www.stoq.com.br/</a>

Fonte: *sites* destacados na tabela.

Um sistema produtivo dentro do conceito ERP consegue agregar informações que dizem respeito aos recursos necessários, sejam de materiais, insumos, máquinas, equipamentos, capacidade de vendas e, inclusive, recursos humanos. Com isto, durante o processo produtivo pode-se modificar o que fora inicialmente planejado em função das informações que se possui e refazer um novo planejamento. A Figura 29 apresenta um modelo de sistema ERP em um meio produtivo.



**Figura 29** Estrutura do sistema ERP no meio produtivo.

Fonte: Autor

A figura mostra a busca do ERP pelos aspectos que envolvem o meio produtivo, levando em consideração os custos, finanças, requisitos de qualidade, suprimentos, as variáveis que possam interferir no processo de produção do produto, de forma que possa gerar o melhor planejamento para uma demanda de produção. Este planejamento é então posto em execução pelo CIM.

### 3.3 SISTEMA DE GERENCIAMENTO DA PRODUÇÃO (MES – *MANUFACTURING EXECUTION SYSTEMS*)

A partir de uma nova necessidade de mercado, com a redução de custos e melhor otimização dos recursos, surge então uma abordagem chamada Sistema de Gerenciamento da Produção (*Manufacturing Execution Systems - MES*), para os sistemas de automação da produção (McCLELLAN, 1997). A premissa se resume a proporcionar a troca de informações a partir dos sistemas de chão de fábrica com os sistemas de planejamento de negócios e fornecer aplicativos integrados em tempo real de produção.

O termo MES foi originalmente criado em 1990 pela AMR – (*Advance Manufacturing Research*), uma empresa de pesquisas para a área industrial. Este primeiro conceito

caracterizou o MES como uma classe de sistemas de informação que reside na camada de software que fica entre os sistemas de automação no chão de fábrica e os sistemas corporativos da companhia denominados ERP – (*Enterprise Resource Planning*).

Um sistema MES é um sistema de controle para gerenciar e monitorar o processo de trabalho no chão de fábrica. O MES mantém o controle de todas as informações de produção em tempo real, recebendo dados de robôs, máquinas e funcionários. O objetivo de um sistema MES é melhorar a produtividade e reduzir os tempos de ciclo, enquanto continua produzindo produtos de qualidade.

Um sistema produtivo implementado com MES permite alguns benefícios, podendo ser citados os seguintes:

- a) Redução de restos e de desperdícios na produção;
- b) Aumentar o tempo em atividade de cada equipamento;
- c) Livrar-se de estoques intermediários;
- d) Reduzir o custo de problemas críticos;
- e) Reagir mais cedo às mudanças do mercado;
- f) Tomar decisões com mais embasamento sobre a produção;
- g) Melhoria da qualidade e rendimento produtivo.

Em um sistema produtivo sem MES, a coleta e a distribuição de informações são um processo complicado em termos de organização da produção. O envio de instruções de trabalho para uma planta de forma eficaz é fundamental para manter a produção funcionando com eficiência máxima. A ordem de produção é enviada para o chão de fábrica e é dividida em instruções de trabalho que são impressas e entregues a cada operário para executar no chão de fábrica. Algumas ordens de serviço podem ser concluídas em paralelo, enquanto outras precisam aguardar que outras sejam finalizadas para elas serem iniciadas. Como o empregado “B” poderia saber que o empregado “A” terminou suas instruções de trabalho para que ele possa

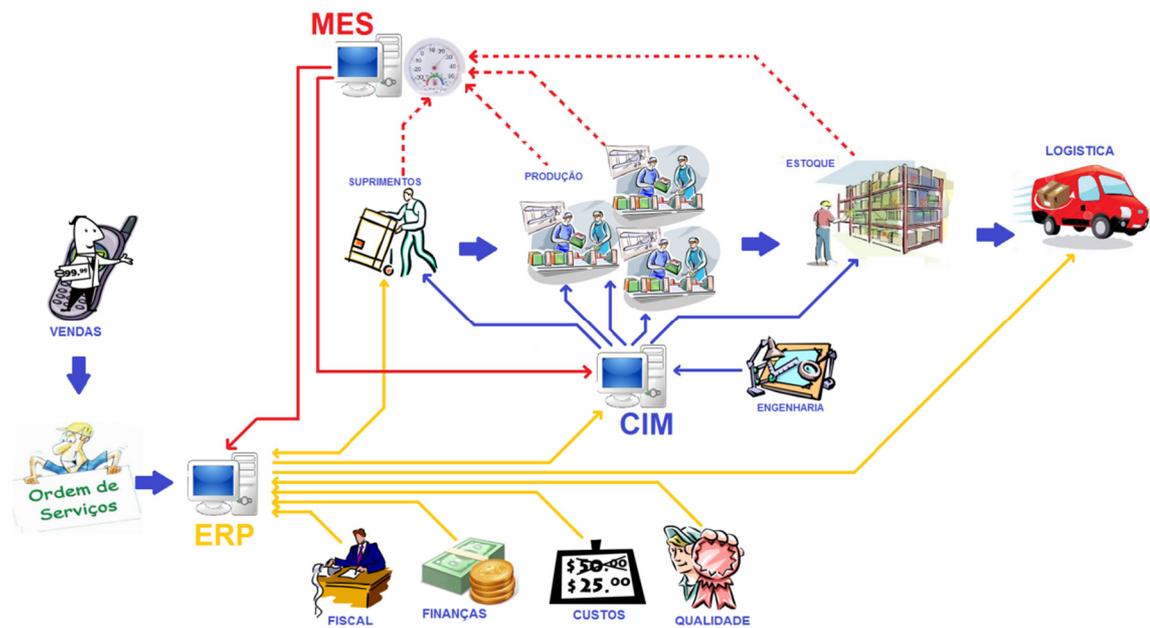
iniciar seu serviço? Quando o empregado deve começar a trabalhar nas instruções de trabalho? Quando o empregado deve parar de trabalhar sobre as instruções de trabalho? Esta informação não está prontamente disponível para a gestão. Alguém precisa coletar manualmente essa informação e criar um relatório. Identificar gargalos é um processo mais longo e o tempo de reação é lento.

Um sistema MES é um sistema de controle para gerenciar e monitorar o processo de trabalho no chão de fábrica (QIU; ZHOU, 2016). Um MES mantém o controle de todas as informações de produção em tempo real, recebendo dados de robôs, máquinas e funcionários. O objetivo de um sistema MES é melhorar a produtividade e reduzir o tempo de ciclo, o tempo total para produzir uma ordem de produção. Ao integrar um MES com software ERP, os gerentes da fábrica podem ser proativos em garantir a entrega da produção com boa qualidade em tempo hábil e baixo custo.

Um sistema MES gerencia todo o estoque de material automaticamente. Quando os materiais são recebidos, eles são inseridos no MES. Quando as instruções de trabalho são emitidas, os materiais consumidos em uma ordem de serviço são automaticamente removidos do inventário. Gerentes de fábrica podem ver em tempo real os materiais restantes no estoque. Gerentes de plantas podem, em seguida, determinar a quantidade de produto que pode ser produzido, se há matéria-prima suficiente para iniciar um pedido de produção, a quantidade de material necessário a ser encomendada e assim por diante.

Quando uma instrução de trabalho é finalizada, o MES notifica as instruções de trabalho relacionadas que podem ser iniciadas. O status em tempo real da ordem de trabalho está disponível para os gerentes de planta monitorarem se a produção está de acordo com o planejado. No final do dia, relatórios das ordens de serviços podem ser gerados para verificar os resultados das instruções de trabalho.

A Figura 30 apresenta a arquitetura de um MES em um sistema produtivo.



**Figura 30** Arquitetura de um MES em um sistema produtivo.

Fonte: autor.

Nesta figura pode-se observar que o MES é um conceito que inspeciona os dados de suprimento, produção e estoque em tempo de produção e fornece estas informações para gerir o controle do sistema produtivo. Também interagindo com o ERP a fim de buscar novos planejamentos da produção em caso de dificuldades detectadas no meio produtivo.

A capacidade de um sistema MES de buscar as informações de um processo produtivo no tempo em que a produção ocorre permite que realize uma análise de desempenho, disponibilidade e qualidade da manufatura. Estes três parâmetros formam o Indicador de Eficiência Geral de Equipamento ou Máquinas ou *Overall Equipment Effectiveness* (OEE) (JONSSON; LESSHAMMAR, 1999). O OEE é um indicador que permite mensurar a eficiência de um sistema produtivo e avalia-lo comparativamente com outros sistemas. O OEE é definido a partir dos parâmetros:

- a) **Disponibilidade:** corresponde ao quanto a máquina está disponível para ser utilizada. Se a máquina opera por 2 turnos de 6 horas ao dia, mas para durante o intervalo do operador que é de 1 hora por turno, então sua disponibilidade

total será de 10 horas. Se no dia ela operar por 10 horas das 12 totais, então seu índice de disponibilidade será de 83,33%;

- b) Desempenho: Representa o quanto a máquina produz em relação a capacidade de produção desta mesma máquina. Por exemplo, em uma máquina injetora onde o tempo padrão definido para produção é de 200 peças por minuto para um determinado produto, se a produção for inferior a esse valor o desempenho não será de 100%;
- c) Qualidade: corresponde à produção sem refugos, que os produtos estejam dentro do padrão de qualidade definido e não sejam descartados. Se uma máquina produz 20 peças sem defeito algum, sua qualidade é 100%, se produz 20 peças, mas somente 19 estão dentro do padrão de qualidade, então seu índice de qualidade é 95%.

O índice OEE é calculado como o produto entre o índice de disponibilidade, desempenho e qualidade.

$$\text{Índice}_{OEE} = \text{Índice}_{Disponibilidade} \times \text{Índice}_{Desempenho} \times \text{Índice}_{Qualidade} \quad (1)$$

Segundo o JIPM, *Japan Institute of Plant Maintenance*<sup>7</sup>, os índices para definir uma empresa considerada de classe mundial são:

- a) Índice<sub>Disponibilidade</sub> = Acima de 90%;
- b) Índice<sub>Desempenho</sub> = no mínimo 95%;
- c) Índice<sub>Qualidade</sub> = pelo menos 99%.

Isto resulta em um índice de OEE de 85% para uma empresa ser considerada de classe mundial segundo o JIPM.

---

<sup>7</sup> JIPM – Japan Institute of Plant Maintenance, disponível em [www.jipm.or.jp/en/](http://www.jipm.or.jp/en/)

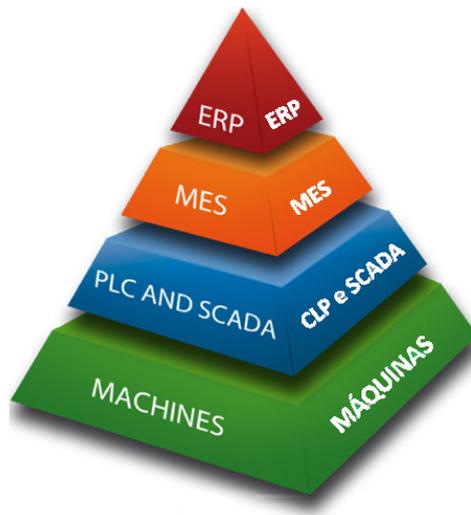
A grande dificuldade é obter fidedignamente estes dados para poder calcular o índice. O sistema MES tem capacidade de proceder estas medições e fornecer os índices de uma forma dinâmica, à medida que a produção ocorre, fornecendo parâmetros instantâneos para tomada de decisão aos gerentes de planta.

### **3.4 SISTEMAS AUTO-ORGANIZÁVEIS COMO SISTEMA DE GESTÃO DA MANUFATURA**

Um sistema auto-organizável traz algumas vantagens aos sistemas produtivos. Com ele os elementos que afetam um sistema produtivo interagem entre si e buscam em conjunto a melhor solução para atender uma demanda de produção. Isto ocorre de forma dinâmica, a cada produto a ser produzido, a cada etapa da produção, tornando as decisões e arranjos produtivos mais assertivos. Esta evolução visa atender a uma necessidade de rápidas respostas a novas demandas de mercado, onde sistemas com hierarquia normalmente não dão conta (ONORI; SEMERE; LINDBERG, 2011).

Em um sistema auto-organizável não há esta hierarquia definida. Todos os componentes do sistema interagem e formam suas coalizões, seus fluxos produtivos, suas restrições, seus monitoramentos, e tomam decisões em meio ao processo produtivo.

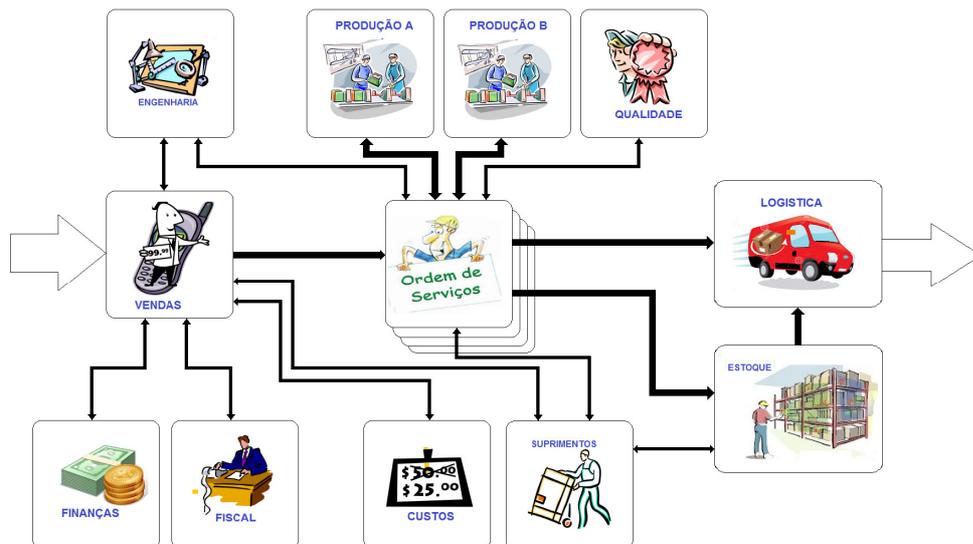
Ao analisar os sistemas de gestão da produção mencionados neste trabalho se verifica certa hierarquia na pirâmide da automação, que pode ser observada na Figura 31, onde se percebe o CLP e sistemas supervisórios na camada de controle, o MES está então em uma camada acima, coletando dados e *interfaceando* com o ERP que está no topo da pirâmide.



**Figura 31 Pirâmide da automação segundo ISA-95.**

Fonte: Adaptado de <https://www.isa.org/isa95/>.

Já nos sistemas auto-organizáveis a abordagem é diferente. A hierarquia é definida pela peça a ser produzida. Quando um produto requer uma manufatura, e aqui consideram-se todos os aspectos envolvidos, ele interage diretamente com os componentes da manufatura para então definir a melhor forma de ser produzido. A Figura 32 apresenta a estrutura de um sistema auto-organizável em um sistema produtivo.



**Figura 32 Estrutura de um sistema Auto-organizável em um sistema produtivo.**

Fonte: Autor.

Neste sistema, o setor de “vendas” ao propor uma oferta interage com “engenharia” para ter conhecimento se o que o cliente está solicitando é possível ser desenvolvido, interage com “suprimentos”, “produção” e “estoque” para definir os prazos de produção, interage com “custos” e “finanças” para definir o preço atual de venda, e, no momento da venda, já pode disparar o pedido “produção”. Todo este processo é dinâmico, pois na negociação estão sendo buscados dados atuais do processo, permitindo um cenário real dele.

Este sistema não se limita a atender a produção formatada de forma mais otimizada e contornando problemas do processo produtivo. Mas busca entender a necessidade do produto e organizar o sistema produtivo para atender a esta necessidade da forma mais otimizada possível, sem ter a preocupação de que se trate de um produto já produzido anteriormente. Em síntese, a auto-organização atende a diversidade de produção, com a pretensão de atender à necessidade de produtos com lote único.

### **3.5 COMPARAÇÃO ENTRE OS SISTEMAS DE GESTÃO DA MANUFATURA**

Comparando os sistemas de manufatura citados conclui-se que não se trata de sistemas competitivos, mas de sistemas que se complementam para atender as necessidades do mercado. O diferencial está na capacidade de atender a diversidade no sistema produtivo.

Um sistema CIM tem a preocupação de garantir que o elemento vinculado diretamente ao sistema produtivo tenha os insumos e as interconexões necessárias para fabricar um dado produto. Armazena os programas das máquinas e as ordens de produção na forma de receitas. Seu objetivo é orquestrar o fluxo de produção. Em caso de algum imprevisto nas máquinas produtivas, o sistema CIM tem dificuldade de apresentar uma solução. Assim como a entrada de um pedido de produto não fabricado anteriormente.

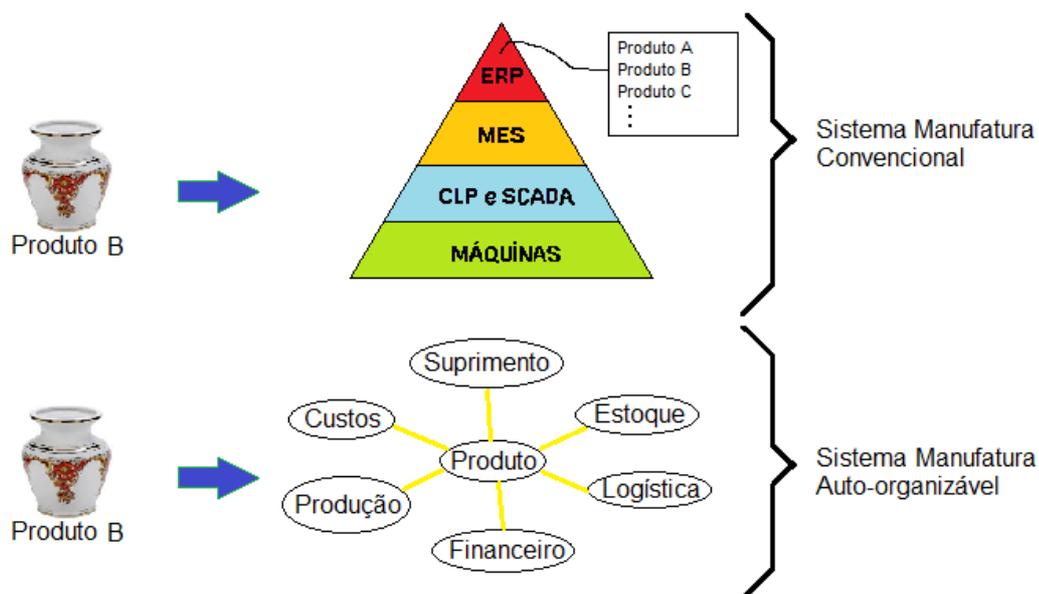
Já o sistema ERP tem a preocupação de levar em consideração todos os fatores que envolvem a produção, não se detendo somente ao aspecto produtivo, mas também aos aspectos

financeiros, custos, logística, entre outros. A proposta é ter uma análise completa dos impactos e benefícios na fabricação de um produto desejado, fornecendo dados mais assertivos para tomada de decisão quanto à produção. Porém ainda mostra dificuldade em tratar as diversidades da produção, pois é um sistema que analisa situações que já ocorreram, portanto, um produto novo terá que ser programado no sistema para ser levado em consideração.

O sistema MES tem a preocupação de coletar dados de tempo de produção no meio produtivo, afim de verificar se o que foi planejado pelo ERP está acontecendo a contento, fornecendo informações importantes para a tomada de decisões em caso de imprevistos em alguma parte do processo. Não tem a preocupação de tratar a diversidade, mas pode dar conta dela opinando quanto a fluxos alternativos do processo. Mas ainda apresenta grande dificuldade para o produto novo, inédito, que foi requerido ao sistema.

Já o sistema Auto-organizável tem a preocupação de promover a interação entre todos os integrantes do sistema de manufatura, fornecendo-lhes a capacidade de dispor e alocar serviços, formando o processo produtivo. Como o fluxo do processo está vinculado ao produto, é este que rege o sistema de manufatura, negociando e alocando recursos em tempo de produção, permitindo que recursos entrem ou saiam do sistema durante este processo, seja por falta ou por alguma outra necessidade.

A Figura 33 a seguir apresenta uma interpretação do requisito de produção em um sistema convencional e um sistema auto-organizável.



**Figura 33 Comparação entre sistema convencional e sistema auto-organizável de manufatura.**

Fonte: Autor.

A Tabela 5 faz um comparativo entre os sistemas de manufatura industrial.

**Tabela 5 Comparativo entre os sistemas de manufatura industrial.**

<b>Sistema de Manufatura Industrial</b>	<b>Características importantes</b>	<b>Objetivo</b>
CIM	Garante que o elemento vinculado diretamente ao sistema produtivo tenha os insumos e as interconexões necessárias para fabricar um dado produto.	Orquestrar o fluxo de produção.
ERP	Leva em consideração todos os fatores que envolvem a produção, não se detendo somente ao aspecto produtivo, mas também aos aspectos financeiros, custos, logística, entre outros.	Analisar completamente os impactos e benefícios na fabricação de um produto desejado.
MES	Coleta dados de tempo de produção no meio produtivo, afim de verificar se o que foi planejado pelo ERP.	Fornecer informações importantes do processo para a tomada de decisões em caso de imprevistos em alguma parte do processo.
Auto-Organizável	Promove a interação entre todos os integrantes do sistema de manufatura, a fim de alocar e disponibilizar serviços.	Gerenciar o fluxo produtivo a partir do produto, de forma dinâmica, em tempo de produção.

Fonte: Autor.

Mesmo com o planejamento abrangente do sistema ERP, com a dinâmica do sistema MES e o controle integrado do sistema CIM, a lista de produtos que o sistema produz deve estar programada neste sistema, com todas as suas peculiaridades. A inserção de uma demanda de produto que ainda não ocorreu no sistema é um problema, e necessita de uma reprogramação do sistema para poder atender a esta nova necessidade.

Com sistema de manufatura sendo implementado no conceito de sistemas auto-organizáveis, o produto novo irá interagir com os atores do processo produtivo e buscar a organização necessária para sua produção, isto em tempo de processo. Assim, não há necessidade do fluxo de processos, uso de insumos ou uso de recursos estarem previamente descritos no ambiente de planejamento da produção. Estas variáveis serão descobertas e alocadas conforme a negociação entre os atores forem ocorrendo.

Portanto, pode-se dizer que um sistema de manufatura concebido com o conceito de sistemas auto-organizáveis se apresenta como uma boa solução para a problemática de atender a diversidade de mercado por produtos novos, diferenciados e com lotes pequenos de produção.

Uma vez conhecidas as tecnologias e os conceitos produtivos pertinentes ao trabalho, faz-se necessário investigar o que outros autores e grupos de pesquisa estudaram sobre a problemática apontada e suas soluções apresentadas, de forma a verificar pontos importantes de seus trabalhos, sugestões para o desenvolvimento desta tese e, principalmente, identificar as lacunas, ou seja, os aspectos que ainda não foram abordados e que carecem de um estudo específico, que formará a tese proposta.

Estas pesquisas são apontadas no próximo capítulo, bem como a identificação das lacunas a serem investigadas.

## **4 DISCUSSÃO SOBRE PESQUISAS COM A UTILIZAÇÃO DE SISTEMAS MULTIAGENTES PARA ATENDER ÀS NECESSIDADES DE MANUFATURA INDUSTRIAL POR DIVERSIDADE, AGILIDADE E AUTO-ORGANIZAÇÃO.**

A problemática abordada diz respeito à necessidade da indústria de que seus sistemas produtivos possam atender a diversidade de produção, a agilidade em seus processos produtivos e a auto-organização de seus recursos de produção. Neste capítulo serão então descritos trabalhos de pesquisa que se propõe a solucionar estas necessidades, com o intuito de identificar as problemáticas e formas de abordagem propostas pelos autores. Este estudo será importante para identificar pontos comuns e pontos divergentes entre pesquisas realizadas e o trabalho aqui proposto.

### **4.1 DIVERSIDADE NOS REQUISITOS DE PRODUÇÃO**

A diversidade nos requisitos produtivos refere-se à necessidade de alteração do fluxo de produção, ou do ajuste nos recursos em função de um novo requisito que se apresentou ao produto, ou até mesmo a um novo produto que necessita ser ensaiado/produzido, mas em pequeno lote. Assim, a mudança no fluxo produtivo passa ser uma constante e o sistema de produção deve atender a estas mudanças. E com foco neste problema alguns pesquisadores desenvolveram estudos em busca de alternativas que resolvam ou minimizem esta necessidade de diversidade.

Em sua tese de doutorado, **André Cavalcante** (CAVALCANTE, 2012) propõe uma plataforma multiagente que usa o conceito de agente mecatrônico, afim de permitir o desenvolvimento de sistemas auto-organizados e que também possuam a capacidade de otimizar autonomamente algum recurso interno ao sistema. Ele parte da necessidade de uma rápida configuração (ou reconfiguração) para permitir a produção de uma grande gama de produtos customizados.

O trabalho do autor aponta algumas características que o ambiente deve ter para atender a diversidade de produção, que se apresenta pela necessidade de produtos customizados, tais como:

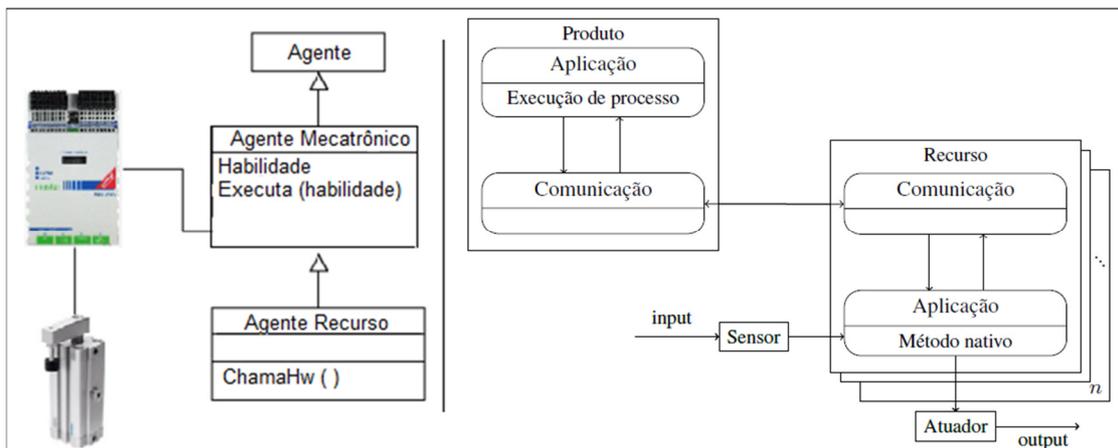
- a) Robustez: sistemas de montagem necessitam funcionar mesmo em presença de falhas, ou ter capacidade de suportar fadigas;
- b) Repetibilidade: sistemas de montagem devem ser capazes de funcionar da mesma forma por um longo tempo;
- c) Integração: o sistema de montagem deve estar integrado no sistema produtivo;
- d) Interoperabilidade: capaz de operar com sistemas externos, que podem não ter sido descritos no momento do seu desenvolvimento;
- e) Compatibilidade: sistemas de montagem devem ser capazes de interoperar com outros sistemas que já existem na produção (sistemas legados);
- f) Adaptabilidade: o sistema de montagem deve ser capaz de alterar a si mesmo, ou ser alterado, a fim de adaptar-se às mudanças ambientais;

O método proposto por Cavalcante faz uso da arquitetura CoBaSa (*Coalition Based Approach for Shop Floor Agility*), originalmente proposta por (OLIVEIRA; CAMARINHA-MATOS, 2003). A arquitetura CoBASA propõe vários tipos de agentes e suas interações para o desenvolvimento de sistemas evolutivos, tais como:

- a) MRA (Mechatronic Resource Agent): que modela um recurso em um agente;
- b) AMI (Agent Machine Interface): que está conectado a um módulo físico e é uma interface padrão para o acesso ao hardware pelo MRA;
- c) BA (Broker Agent): responsável pela criação de coalizões entre os agentes;

- d) CMgA (Cluster Manager Agent): responsável pelo gerenciamento de agrupamentos entre os agentes;
- e) CA (Coordination Agent): coordena outros agentes, isto é, representa uma coalizão.

A proposta de Cavalcante para atender a esta necessidade é a implementação de uma arquitetura híbrida para o sistema, a qual é formada por agentes em duas camadas, onde uma camada realiza as deliberações necessárias e a outra contém os agentes que devem reagir rapidamente aos eventos de organização. A Figura 34 apresenta o modelo proposto para formação do objeto em duas camadas.



**Figura 34 Método proposto para modelar um recurso mecânico como um objeto.**

Fonte: (CAVALCANTE, 2012)

Na figura se observa que há um agente responsável pela ação direta sobre os elementos físicos do sistema e outro responsável pela deliberação das ações e busca de informações. Assim, a ação física mais comum, que já possui a lógica de execução definida, é então realizada com mais rapidez, tornando o sistema mais eficiente.

A proposta foi implementada usando a plataforma JADE, sendo validada em um sistema de produção que incluía diversos sistemas de transporte de produtos por caminhos e rotas

distintas. A implementação dos agentes ocorreu através de eletrônica embarcada e computadores.

O resultado da implementação foi um sistema que apresentou as características de auto-organização. À medida que novos produtos entravam em produção, eles passavam a concorrer pelos recursos existentes, atendendo assim a necessidade de reconfiguração. A implementação mostrou-se capaz de fornecer as funcionalidades do agente mecatrônico como um objeto a ser instanciado dentro da aplicação.

A contribuição deste trabalho foi a apresentação de uma proposta para uma arquitetura híbrida, baseada em agentes, que possibilita tanto deliberação quanto reação, especificamente desenvolvida para suportar o conceito de agente mecatrônico e, portanto, específica para a criação de aplicações na manufatura, em especial para aplicações que levam à customização em massa.

O trabalho de **Mário Roloff**, em sua tese de doutoramento (ROLOFF, 2014), identifica a necessidade de Produção de Pequenas Séries (PPS), que é um tipo de manufatura caracterizado pela alta diversidade de produtos a serem produzidos associada a tamanhos de lotes reduzidos, possivelmente unitários. Neste sistema de produção as tecnologias empregadas para controle do processo devem priorizar a produção sem defeitos, inclusive no primeiro item do lote, que pode ser o único. Falhas neste sistema de produção ou defeitos inseridos nos produtos inviabilizam economicamente todo o lote. A necessidade que se apresentou foi a de ter um modelo de referência para Sistemas Multiagentes aplicado ao controle de produção de pequenas séries (MAS4SSP – *Multi-agent System for Small Series Production*).

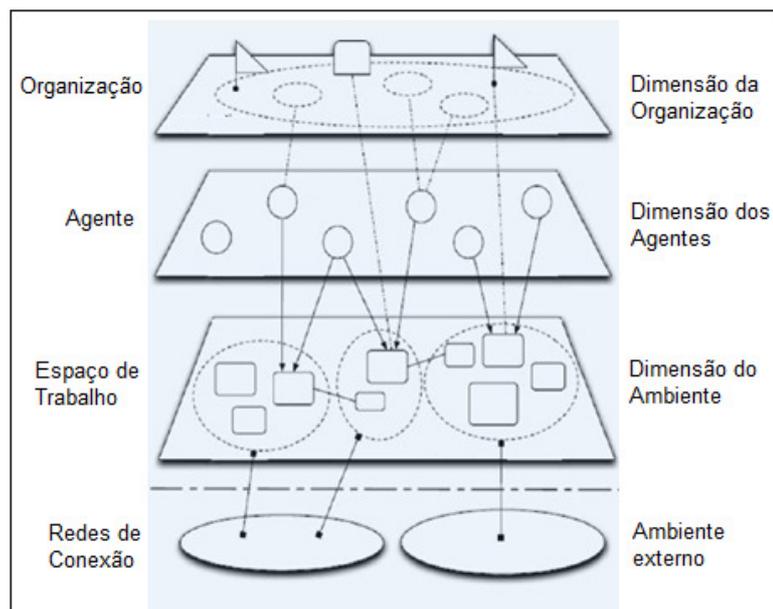
O método empregado vale-se do paradigma vogal (DEMAZEAU, 1995) onde a abordagem ao Sistema Multiagente se divide nas seguintes dimensões:

- a) Dimensão dos Agentes (*A – Agent*): que trata das lógicas com que os agentes serão concebidos;
- b) Dimensão do Ambiente (*E – Environment*): onde os agentes estarão sendo executados;
- c) Dimensão da Interação (*I – Interaction*): que trata das interações entre os elementos do Sistema Multiagente;
- d) Dimensão da Organização (*O – Organization*): define os modelos e a intenção nas interações entre os agentes, a partir de metas e objetivos do sistema como um todo.

O método empregado utiliza a plataforma JaCaMo que integra outras três plataformas, onde cada uma atua em uma dimensão distinta. Na dimensão dos agentes a plataforma Jason foi utilizada, com os agentes sendo programados na linguagem AgentSpeak. Na dimensão do ambiente foi utilizada a plataforma CArtaGO e como dimensão de organização foi utilizada a plataforma Moise+.

O objetivo é obter um ambiente de programação que seja uniforme e consistente no sentido de simplificar a combinação das dimensões para a programação de Sistemas Multiagentes voltados à Produção em Pequenas Séries.

A Figura 35 apresenta o modelo de tratamento do ambiente em três camadas interligadas.



**Figura 35 Modelo de interação em três camadas para o ambiente dos agentes.**

Fonte: (ROLOFF, 2014)

Fator tecnológico interessante neste trabalho foi a utilização de um sistema SCADA (*Supervisory Control And Data Acquisition*) atuando como um sistema de aquisição de dados (ALBUQUERQUE et al., 2007). Este sistema permitiu a interconexão dos agentes com os controladores da planta, sem a necessidade de intervenção na lógica de controle local. A partir dos sistemas SCADA as variáveis foram então disponibilizadas, permitindo monitoramento e intervenção.

Outro fator tecnológico foi a utilização de *Web Services*<sup>8</sup> para integração de sistemas e na comunicação entre aplicações diferentes (BENSLIMANE; CAIRE; GREENWOOD, 2008). Com esta tecnologia é possível que novas aplicações possam interagir com outras já existentes e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Assim, foi possível comunicar o sistema SCADA com os Agentes.

O modelo obtido com a utilização de uma aplicação do Sistema Multiagente para controle de Produção em Pequenas Séries utiliza como referência a linha de produção de

<sup>8</sup> São componentes de software que permitem às aplicações enviar e receber dados em formato XML. Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal.

pequenas séries do laboratório-fábrica LabElectron da Fundação CERTI (ROLOFF, 2014). Este sistema é composto de carregador de placas de circuito impresso, inserção de pasta de solda, impressora, forno de refusão, inspetora de objeto e descarregador de placas de circuito impresso. Sua implementação focou a interação do agente com as máquinas citadas, controladas por sistema de processamento dedicados, sem o uso de controladores lógicos programáveis, valendo-se do SCADA para obter informações.

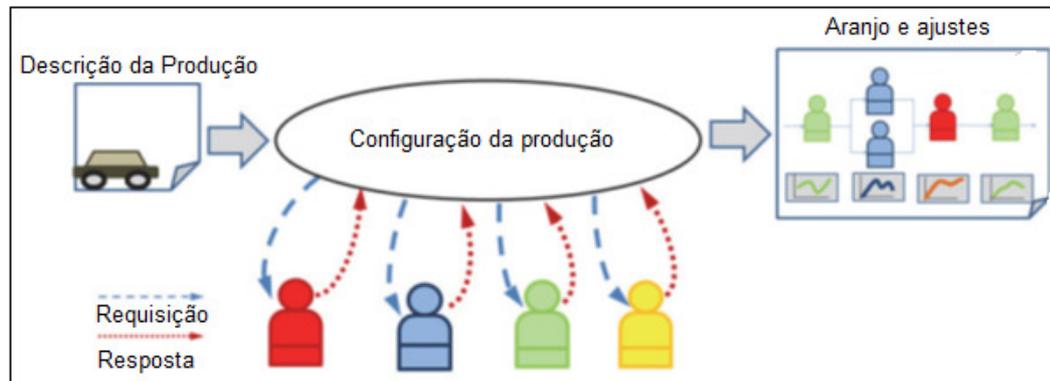
O objetivo geral da tese de Roloff foi apresentar uma nova abordagem para a implementação de Sistema Multiagente para o controle da Produção em Pequenas Séries. Como resultado, a nova abordagem do sistema implementado forneceu uma ferramenta que diminuiu o tempo de *setup* da linha de produção e também fornece auxílio aos operadores do sistema, assumindo a gestão de algumas atividades de controle da produção.

A contribuição do trabalho de Roloff é um conceito de modelo de implementação de Sistema Multiagente para Produção em Pequenas Séries, que vai em direção aos requisitos desejáveis para este tipo de necessidade da indústria moderna.

No trabalho de **Michael Peschl** (PESCHL, 2014) é abordada a importância de atender às mudanças rápidas das necessidades do consumidor, com o incremento do número de novos produtos e sua variedade. A manufatura do futuro terá que dar conta de uma variedade de produtos sendo produzidos na mesma linha de produção, assim como a flexibilidade de alterar a rota previamente estipulada.

O objetivo deste trabalho é desenvolver uma arquitetura de manufatura flexível de sistemas, que permite a fácil aceitação na indústria. Baseia-se em uma abordagem de baixo para cima (*bottom up*), com um novo tipo de componentes de chão de fábrica inteligentes e flexíveis, denominados “Manufactrons”.

O autor propõe um método que gera a configuração de cada agente a partir da descrição do produto, com objetivo de formar um arranjo produtivo e suas necessidades de configurações. A Figura 36 ilustra o modelo de configuração dos agentes para formar um arranjo produtivo.



**Figura 36 Modelo de configuração dos agentes para formar um arranjo produtivo.**

Fonte: (PESCHL, 2014).

O sistema vale-se de componentes chamados “*Manuclatron*”, que são dispositivos dotados de habilidades de cada agente (coalizão) para realizar tarefas. A definição é realizada com base em quatro pontos de vista diferentes: visão do componente, funcionalidade, hierarquia e resposta do mestre. Um “*Manufactron*” é uma entidade autosuficiente, capaz de encapsular funcionalidades e interagir com seu ambiente pela troca de mensagens padronizadas e sincronizadas.

A tecnologia empregada na validação utiliza como meio de comunicação a linguagem ACL (*Agent Communication Language*) e KQML (*Knowledge Query and Manipulation Language*), além de FIPA-ACL.

O método foi testado em quatro situações de manufatura industrial:

- a) em estudo de caso na indústria automotiva o resultado foi uma redução do tempo de rampa de partida para sistemas de manufatura, especialmente os sistemas de soldagem robotizada;
- b) na indústria de componentes elétricos as aplicações são caracterizadas por ciclos de produção curtos e o envolvimento de uma grande quantidade de

- diferentes processos, tais como o transporte, a fixação e a manipulação de solda, tudo em máquinas de dimensões reduzidas. A aplicação do método nestas situações foi capaz de reduzir o tempo de comutação para configurar a máquina com uma nova variação do produto de forma significativa;
- c) aplicação na indústria aeroespacial apresentou redução nos tempos de ajuste dos sistemas, valendo-se também de comissionamento virtual;
  - d) aplicação em geração de energia solar que culminou em redução do tempo de rampa de implantação pela identificação de *Manufactrons*, adequadas para tarefas de produção por meio das capacidades de autodescrição e simulação distribuídas em grandes instalações de produção.

Como resultado, o Sistema Multiagente foi testado nos quatro ambientes citados nas seguintes características:

- a. Diminuir o tempo para pôr em marcha a produção;
- b. Fornecer uma metodologia de fácil utilização para a produção de múltiplas variantes;
- c. Permitir uma fácil implementação na indústria;
- d. Gerar e usar conhecimentos de fornecedores de componentes e sistemas;
- e. Documentar detalhadamente os dados de produção.

E nos quatro ambientes as características avaliadas mostraram resultados que o autor considerou adequados, apresentando ganho qualitativo e quantitativo.

Como contribuição, o trabalho apresentou um método de implementação de uma arquitetura flexível, que atende as diversidades do mercado, como previsto na problemática inicial.

O projeto **SOPRO** (*Self-Organizing Production*) (SOPRO PROJECT, 2014), desenvolvido pela Fraunhofer IZM – Alemanha, desenvolveu soluções para ambientes de

produção dinâmicos, em que são possíveis decisões rápidas sobre o processamento de pedidos e respostas rápidas em termos de produção para fluxos produtivos imprevisíveis. Aborda a necessidade dos processos de fabricação, que não serão apenas eficientes, mas poderão ser flexíveis o suficiente para atender a diversidade das necessidades do produto, capazes de coordenar e controlar a si mesmos. O projeto SOPRO propõe uma arquitetura com agentes móveis descentralizados, os quais negociam com outros elementos da rede de produção, realizam coleta de dados e tem a capacidade de se auto-organizar para otimizar os processos de fabricação. Utilizam Sistemas Multiagentes para implementar sua plataforma e programar seus agentes específicos.

Apesar de uma boa proposta para atender aos sistemas de manufatura, o projeto não apresenta muita informação para consulta. Mostra as diretrizes, mas não apresenta mais explicitamente o método nem a forma de abordagem, onde o Sistema Multiagente atuará.

O projeto **EuPASS** (*Evolvable Ultra-Precision Assembly Systems*) (EUPASS PROJECT, 2014) coordenado pela Universidade Nova de Lisboa – Portugal tem como objetivo desenvolver soluções rentáveis e sustentáveis de produção em montagem de ultra precisão, oferecendo serviços de montagem com destaque na rápida demanda. Isto soluciona a necessidade de atendimento a demandas de montagem muito específicas, quase customizadas. Isto é alcançado através do desenvolvimento e fornecimento de uma série de tecnologias inovadoras e soluções, incluindo:

- a) Infraestrutura europeia piloto com amplo depósito de micromódulos de montagem e integração de software, permitindo a configuração e implantação de sistemas flexíveis de montagem de precisão com custo mínimo de investimento rápido;

- b) Desenvolvimento de dispositivos de montagens de ultraprecisão, incluindo sistemas modulares de alta precisão, como: manipuladores, pinças e alimentadores;
- c) Junção de novas técnicas, incluindo juntas micromecânicas, nanodistribuição e solda a laser;
- d) Metodologia de conhecimento robusta, modelos de custos e ferramentas de software para apoiar a oferta de serviços de montagem em ultraprecisão, de acoplamento rápido com baixo custo de investimento, elevada utilização da capacidade e melhoria da capacidade de reutilização de equipamentos;
- e) Novas normas para integração de módulos de montagem de precisão e sistemas de controle utilizando a abordagem de arquitetura aberta.

O projeto faz uma abordagem em plataformas de Sistemas Multiagentes para reconfiguração de equipamentos e módulos, dirigidos por requerimentos de produção definidos, em uma ontologia de projeto. O projeto contribui então com um modelo utilizando Sistemas Multiagentes para atender a sistemas de montagem de ultra precisão, em uma demanda quase customizada.

Dentro da pesquisa os resultados apresentados remetem ao atendimento das necessidades de montagem de precisão, com soluções em Sistemas Multiagentes.

No artigo apresentado por **Regina Frei, Giovanna Serugendo e José Barata** (FREI; SERUGENDO; BARATA, 2008) são abordados o problema da customização em massa de produtos e a necessidade dos meios produtivos de atuarem com lotes de baixos volumes, muitas variantes na produção e rápida mudança nos requisitos do produto.

Apresenta uma abordagem com Sistemas Evolutivos de Manufatura (*Evolvable Assembly Systems – EAS*), definido como um sistema de montagem dinâmico, auto-organizado e evolutivo, formado por módulos interligados e capazes de mudar sua estrutura de acordo com

as mudanças ambientais relevantes, ou com a capacidade de atender às flutuações de demanda de processos produtivos. Os sistemas de montagem evolutivos estão mais próximos do gerenciamento do processo distribuído, onde cada equipamento de montagem tem seu gerenciamento próprio, capaz de interagir, apresentar suas funcionalidades e de reconhecer demandas de funcionalidades que requerem uma reconfiguração (BI et al., 2007) do sistema de montagem. No sistema EAS se um módulo falhar, alguma funcionalidade pode ser perdida, mas o sistema pode se ajustar e formar novas coalizões ou diferentes formatações de constituição, suprimindo o módulo perdido (CAVALCANTE; PEREIRA; BARATA, 2010). Assim, o sistema pode apresentar propriedades que não são encontradas em partes. Estas propriedades ou comportamentos surgem para auto-organizar o sistema.

A arquitetura genérica para sistemas de auto-organização e autoadaptação aborda especificamente os sistemas trabalhando sob condições de mudança dinâmica. Ela engloba a coordenação do trabalho entre os componentes autônomos, além da inserção ou remoção de componentes a qualquer momento, mantendo o nível de confiabilidade do sistema produtivo.

Como tecnologia, o trabalho utiliza a plataforma JADE e os agentes implementados na linguagem JAVA.

O principal resultado apontado é a implementação da estratégia *plug-and-produce*, que garante a entrada e saída de estações nos sistemas, provocando assim sua reconfiguração, e, por conseguinte, o atendimento dos requisitos de diversidade do produto.

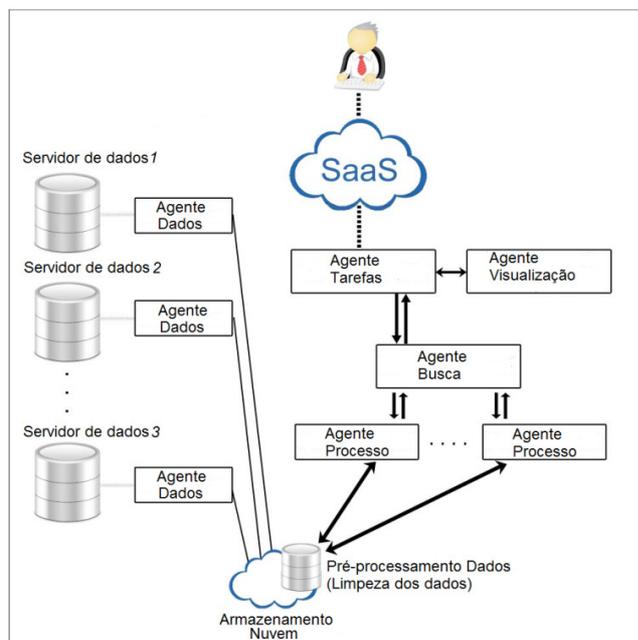
Mas o trabalho apresentado não considera uma forma de medir a qualidade da solução obtida. Questiona-se se a solução obtida com a autoadaptação dos componentes do sistema produtivo é realmente a solução com a máxima precisão e recursos mínimos. Também se questiona uma maior investigação do mecanismo para elaboração das políticas de interação e

coalizão entre os agentes, a fim de evitar situações conflitantes. Sugere-se aqui a consulta ao usuário humano como uma opção, tornando este uma parte do sistema, um agente humano.

A contribuição do trabalho apresentado é um Sistema Multiagente implementando os conceitos de EAS, evidenciando os benefícios das características de *plug-and-produce* a um sistema produtivo que necessita de diversidade em seus requisitos de produto.

O artigo apresentado por **Benyoucer Othmane** e **Rahal Sidi Ahmed Hebri** (OTHMANE; HEBRI, 2012) visou atender a diversidade nos sistemas produtivos em função dos requisitos de produção de um sistema focando o incremento do volume comercial de produtos, requisitando novas funcionalidades para o meio de fabricação. Hoje em dia com o aumento na conectividade de nossas redes de dados, mudou-se a forma de pensar de sistemas centralizados para pensar em arquiteturas distribuídas, de modo que à busca de um dado específico em um sistema distribuído tornou-se um tema de pesquisa importante.

Os autores demonstram um modelo a partir do conceito de nuvem (JADEJA; MODI, 2012) que se refere à utilização de memória e capacidade de computação de computadores e servidores em todo o mundo, ligados por uma rede, tal como a Internet. Os usuários da nuvem podem ter um recurso computacional flexível e considerável capacidade computacional. Neste modelo os agentes buscam dados, os quais foram compartilhados por outros agentes na nuvem, de forma a obter as regras para a melhor otimização de seus recursos e atender assim aos requisitos de produção, que agora necessitam que o sistema tenha uma nova configuração. A Figura 37 apresenta o modelo proposto com as interações entre os agentes.



**Figura 37 Modelo de interação dos agentes em nuvem.**

Fonte: adaptado de (OTHMANE; HEBRI, 2012).

A figura apresenta uma arquitetura baseada em nuvem como um SaaS (*Software as a Service*), que usa Sistemas Multiagentes para a execução de diferentes tarefas do sistema. Esta arquitetura tem componentes, tais como: interface com o usuário; agente de tarefas; agente de dados; nuvem de armazenamento temporário; agente de busca; agente de processo e agente de visualização.

A contribuição deste trabalho é uma nova abordagem para a busca de dados distribuídos com base em sistemas de computação em nuvem e Sistema Multiagente. Esta abordagem visa aumentar a eficiência com uma arquitetura modular.

O resultado apresentado foi um sistema eficaz da busca por informações que levaram o sistema a se autoconfigurar para atender as necessidades de reconfiguração do Sistema Multiagente.

## 4.2 AGILIDADE EM SISTEMAS PRODUTIVOS

Agilidade é um requisito que se apresenta aos novos meios de manufatura, em face da necessidade dos meios de produção de atenderem o mercado de forma rápida e eficaz. A flexibilidade torna-se fator importante, pois através dela os sistemas podem adequar-se para atender às mudanças requeridas. Fator também importante é a heterogeneidade dos recursos, que faz com que o ambiente produtivo tenha que ser flexível a ponto de atender especificidades de cada recurso que surge e até mesmo a substituição destes. Neste foco algumas pesquisas foram realizadas e relatadas a seguir.

Na tese de doutoramento de **José Antônio Barata de Oliveira** (OLIVEIRA, 2003) o autor aborda o problema da necessidade de agilidade no chão de fábrica, a fim de lidar com os distúrbios e incertezas que as empresas de fabricação enfrentam em seus cenários de negócios. A diversidade de produtos e a agilidade na produção e entregas fazem com que os dispositivos mecatrônicos existentes não consigam atender a estes novos requisitos. A capacidade de meios produtivos de serem melhorados rapidamente, alterados ou tornarem-se facilmente modificáveis dá ao meio produtivo a agilidade necessária para atender aos novos requisitos.

A problemática apontada pelo autor refere-se as necessidades de métodos e ferramentas para o processo produtivo que tenham em seus controles componentes reutilizáveis e que possam ser rapidamente modificáveis.

Como hipótese o autor cita que o controle e supervisão do meio de produção é conseguido através da arquitetura multiagente baseado em CoBASA, criada para apoiar a adaptação e mudanças de arquiteturas de controle de chão de fábrica, com rapidez e mínimo esforço. Os elementos da manufatura são reunidos em coligações, constituídas de componentes de fabricação agentificado (módulos), cujas relações dentro das coalizões são regidas por contratos entre os elementos da coalizão, que estão configurados sempre que uma coalizão está

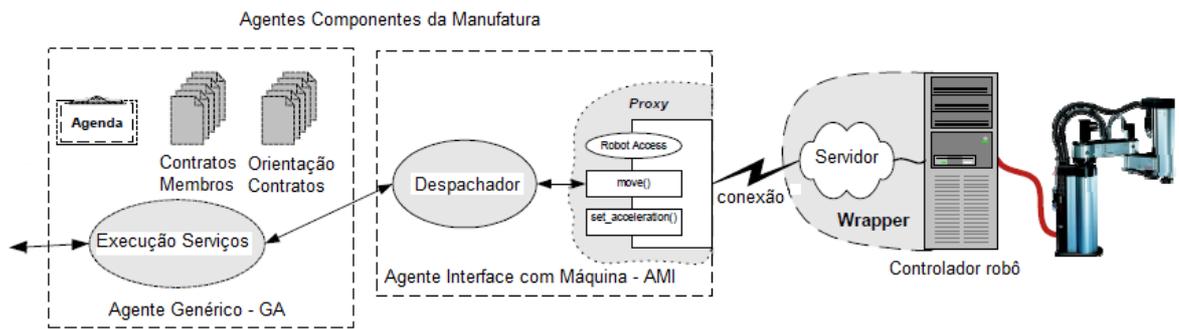
estabelecida. Criação e alteração de uma coalizão requer esforço mínimo, pois a programação requer apenas alterações no contrato que a regula. A arquitetura CoBASA já foi mencionada no trabalho de André Cavalcante, descrito anteriormente. CoBASA se compõe de agentes como: BA (*Broker Agent*), responsável interface com usuário, com outros agentes e com outros *clusters*<sup>9</sup>; CMgA (*Cluster Manager Agent*) que regula a interação entre os agentes e a formação dos *clusters*; MRA (*Manufacturing Resource Agent*) responsável por estender as habilidades do agente de manufatura; AMI (*Agent Manufacturing Interface*) responsável pelo interfaceamento do agente com o meio físico e GA (*Generic Agent*) capaz de assumir um comportamento desejado.

Um destaque especial para o *Agent Machine Interface* (AMI) que é o agente que está diretamente ligado ao controlador físico. Ele atua como uma espécie de driver de dispositivo. Para cada controlador diferente deve haver um AMI. Basicamente o AMI é um invólucro do agente de um componente industrial que exporta as funcionalidades já existentes em seu controlador físico.

Mas o trabalho não deixa claro como que esta conexão física ocorre, nem que protocolo foi utilizado. A Figura 38 mostra como o AMI se comporta na composição com o agente genérico.

---

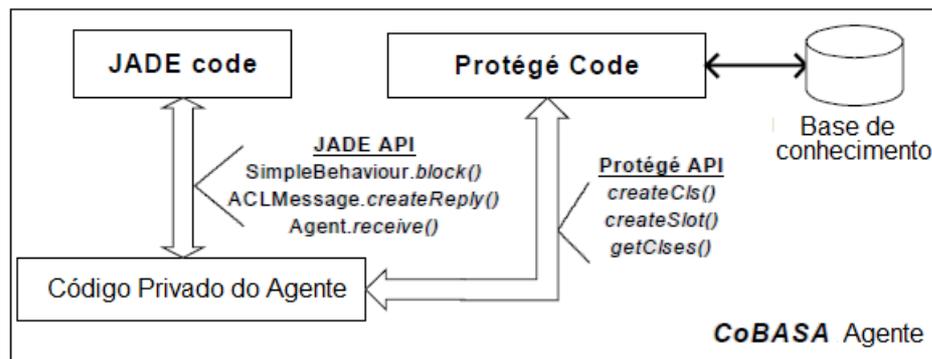
<sup>9</sup>Cluster é definido como um grupo de coisas ou de atividades semelhantes que se desenvolvem conjuntamente. Entende-se a ideia de junção, união, agregação, integração. Neste contexto são agentes como funcionalidades complementares que se unem para formar uma funcionalidade desejada.



**Figura 38** Composição do agente genérico com o agente de interface com a manufatura.

Fonte: Adaptado de (OLIVEIRA, 2003).

Como recurso tecnológico o autor vale-se da plataforma JADE para implementação dos agentes e a ela agrega o aplicativo para modelagem do conhecimento *PROTÉGÉ*, que atua sobre a base de conhecimento do agente. A Figura 39 apresenta este modelo de composição.



**Figura 39** Composição de um agente utilizando JADE e PROTÉGÉ.

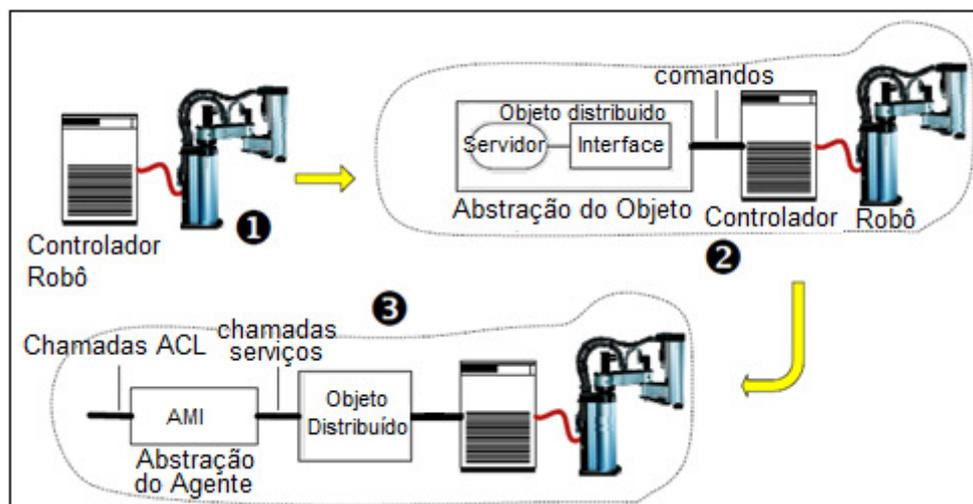
Fonte: adaptado de (OLIVEIRA, 2003).

Para a modelagem dos agentes o autor propõe um método que visa estruturar os sistemas de manufatura mecatrônicos em agentes de manufatura, possibilitando sua interação e disposição de serviços para serem consultados, agendados e executados. O modelo para implementar os agentes mecatrônicos segue a seguinte sequência proposta pelo autor:

1. Criar o *Agent Machine Interface* - AMI, ligação entre o agente genérico e o meio físico;

2. Integrar o componente de manufatura, as funcionalidades de fabricação com o controlador de componente, que são modelados utilizando mecanismos de abstração baseados em computador;
3. Criar o agente de manufatura – agentificação dos componentes de manufatura;
4. Criar o Agente de Recursos - MRA (*Manufacturing Resource Agent*);
5. Executar os agentes.

A Figura 40 apresenta uma ilustração de como o modelo funciona em suas etapas.



**Figura 40** Etapas propostas para modelar um agente mecânico.

Fonte: adaptado de (OLIVEIRA, 2003).

Em “1” o componente mecânico é reconhecido, em “2” é construído o objeto que abstrai as funcionalidades do componente mecânico e em “3” é construído o objeto que abstrai o agente para se comunicar com os demais agentes de um sistema.

O resultado foi ensaiado em um laboratório com sistemas robóticos procedendo a manufatura junto com um sistema transportador. Nele o autor observou o atendimento das necessidades de agilidade das tomadas de decisões e composição de recursos formando *clusters*.

A principal contribuição científica deste trabalho foi uma arquitetura baseada em multiagentes, chamada CoBASA, criada para apoiar o processo de reengenharia do chão de

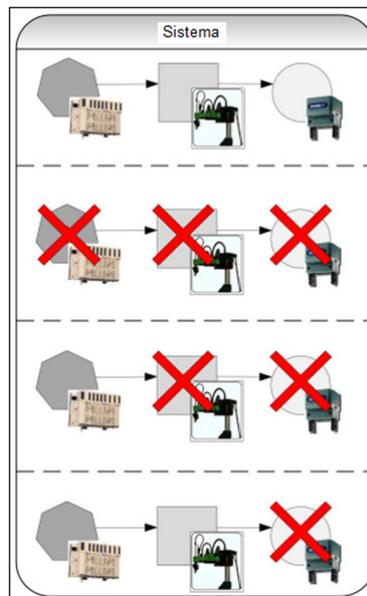
fábrica, com arquiteturas de controle / supervisão que utilizam contratos para reger as relações entre as coligações membros.

O trabalho de **Luis Domingos Ferreira Ribeiro** (RIBEIRO, 2012) aborda a problemática da execução de diagnóstico em instalações industriais complexas, que são compostas por um elevado número de componentes que interagem entre si no âmbito do processo de produção. Neste ambiente o diagnóstico de falha é dificultado no que diz respeito à detecção da origem da mesma.

A abordagem apresentada se aplica a problemas em geral e a várias classes de sistemas ligados em rede, mas neste trabalho específico o autor utilizou como base a abordagem dos sistemas evolutivos de produção e os sistemas evolutivos de montagem (EPS/EAS).

O autor traz um modelo em que Sistemas Multiagentes interagem para detectar e prever a propagação de falhas, buscando diagnosticar a assim origem da falha e suas consequências.

A Figura 41 indica uma ilustração do modelo do processo.



**Figura 41 Modelo de diagnóstico e predição da propagação de falha.**

Fonte: (RIBEIRO, 2012).

Como tecnologia o autor executou Sistemas Multiagentes programados em JAVA, executados na plataforma JADE.

Os resultados foram obtidos através da implantação de uma célula de manufatura robotizada composta de um transportador, dois robôs industriais e paletes de transporte. Os resultados sugerem que é de fato possível explorar as interações e as informações locais para promover uma resposta ao diagnóstico de auto-organização em um sistema EPS/EAS.

As principais contribuições que o autor traz com seu trabalho são: um método formalizado para diagnóstico de falhas e um banco de diagnóstico em ambiente de auto-organização em rede.

Projeto **XPRESS** (*Flexible Production Experts for Reconfigurable Assembly Technology*)\_ (XPRESS PROJECT, 2004) foi desenvolvido pela Universidade de OULU, junto com o Instituto Fraunhofer IFF, Universidade do Porto e Fraunhofer IPA, contando com empresas parceiras como: Airbus, AWL Critical, Brotje Automation, FIAT, GAMAX, Zentrum, Swantec, Karlsruhe Institute of Technology- KIT, Aerotec e Technax.

Este projeto visa atender à necessidade dos fabricantes de serem capazes de responder às rápidas mudanças dos requisitos dos consumidores, oferecendo produtos de alta qualidade em quantidades adequadas a custos reduzidos, agindo assim com agilidade em seus sistemas produtivos.

O projeto descreve o objetivo de estabelecer um avanço na fabricação de um novo conceito de produção flexível. O conceito descrito baseia-se na ideia genérica de um componente de manufatura, que o autor chama de “*Manufactrons*”.

O método utilizado no projeto foca a abordagem de três seções dentro dos níveis de produção industrial:

- 1) Configuração e simulação da produção: que permite simulação e otimização de uma linha de montagem durante a fase de planejamento, iniciando o fluxo de trabalho para diferentes variantes do produto e avalia a qualidade global para cada produto;

- 2) Condução do fluxo de produção: controla o fluxo de produção de uma ou várias variantes do produto e coordena a sequência de etapas de produção. Com estas características fundamentais do meio produtivo o controle deve prover: uma produção multivariante, reação dinâmica de mudanças nos requisitos de produção, o encaminhamento dinâmico no fluxo de produção e avaliação global da qualidade;
- 3) Controle dos elementos de campo e interação: relacionado com o nível de campo, que visam a integração dos seres humanos no processo de produção automatizado e a implementação de máquinas inteligentes.

A descrição do projeto não deixa claro a tecnologia que será utilizada para implementar os *Manufactrons*, se em um nível conceitual ou uma implementação em *software* propriamente dita.

Como resultado, o autor descreve que com a implementação dos *Manufactrons* os fabricantes serão capazes de responder às rápidas mudanças das necessidades dos consumidores, oferecendo produtos de alta qualidade em quantidades adequadas a custos reduzidos.

A contribuição deste projeto aponta para benefícios como a comunicação entre tarefas e resultados de qualidade, devido ao encapsulamento de conhecimento especializado. Preservação e fornecimento de conhecimento especializado é possível mesmo em ambientes em que o perito na operação fabril não esteja presente. O tempo para a implementação de novos processos de produção é reduzido ao incorporar os *Manufactrons* a uma rede de aprendizagem.

O projeto **PRIME** (*Plug and Produce Intelligent Multi Agent Environment based on Standard Technology*) (PRIME PROJECT, 2014) é coordenado pela Universidade Nova de Lisboa e Universidade de Nottingham, e conta com empresas parceiras como SIEMENS, SIMPLAN, TQC, Asyрил, AHAW, CSEM, Technology Transfer System e Introsys. Ele visa

atender à necessidade dos sistemas produtivos de reconfigurar seus sistemas em função dos requisitos de mercado, com agilidade.

O objetivo descrito no projeto PRIME é uma mudança de paradigma do processo de configuração e integração de sistemas convencional, uso maximizado de recursos, e, em grande parte, em sistemas de montagem automatizada, com capacidade de autoconhecimento e adaptação, metodologia e ferramentas. O objetivo apresentado no projeto é apoiado pelos objetivos fundamentais, descritos como:

- a) Desenvolvimento de uma visão e arquitetura para produção em volume variável de sistemas de montagem multiproduto;
- b) Desenvolvimento de métodos de configuração rápida e sistema de otimização que envolva necessidade de adaptação;
- c) O desenvolvimento de ambiente de produção com abordagem de controle multiagente para integração dos módulos;
- d) Desenvolvimento de uma ferramenta baseada na tecnologia padrão e linguagem de integração e ligação em rede de sistemas de controle heterogêneos de diferentes fornecedores de equipamentos dentro de uma linha de produção;
- e) Desenvolvimento de modelo de comportamento do sistema e metodologia de escolha em tempo real, para apoiar a evolução do sistema, ligado ao desempenho do processo e do produto e a variação de volume de produção;
- f) Integração e sistemas de produção com capacidade de produção autônoma e interfaces para integração contínua;
- g) Apoio à transformação da montagem em uma linha de produção, automatizando a implantação de serviços de fabricação na linha.

Neste projeto os agentes são executados em controladores industriais, que possuem capacidade e disponibilidade para execução do agente em *software*. Esta abordagem tecnológica facilita a implementação, pois coloca o agente em execução no mesmo componente em que são executadas as lógicas que relacionam as entradas e saídas da planta a ser controlada.

Como resultado o projeto prevê que as funcionalidades de *plug-and-produce* resolvam a necessidade do processo produtivo de se reconfigurar e formar novos arranjos de recursos disponíveis para produção. Assim, se prevê que o desenvolvimento de uma ferramenta baseada em Sistemas Multiagentes possa resolver a necessidade de que as heterogeneidades dos equipamentos impliquem na reconfiguração do sistema produtivo.

A contribuição que se destaca neste projeto é a experimentação do Sistema Multiagente que integra os módulos em demonstradores reais de processos produtivos, focando a aplicação industrial.

O artigo publicado por **ZAHNG** e **SHARIFI** (ZAHNG, SHARIFI, 2007) a turbulência e incerteza no ambiente de negócios é indicada como a principal causa de falhas do setor de manufatura industrial, sendo necessária a consideração da necessidade de agilidade como uma estratégia na gestão da manufatura. E como tal, o artigo analisado apresenta um estudo das estratégias de agilidade dentro de empresas através de método que classifica numericamente os fatores que afetam a agilidade, através de resultado de pesquisa realizada junto a 900 empresas do Reino Unido. Traz resultados importantes quanto às estratégias de agilidade que a empresa adota e a forma como o tema é desenvolvido no ambiente da empresa.

O método estrutural proposto prevê quatro elementos principais da agilidade como estratégia de fabricação, que se inter-relacionam de forma a definir as tarefas e escolhas da manufatura:

- a) Drive de Agilidade: interface com as dimensões de negócios de mercado, critério de concorrência, necessidades dos clientes, tecnologia e fatores sociais. Detectam as mudanças no ambiente de negócios;
- b) Capacidade de Agilidade: dispõe das capacidades competitivas de proatividade, resposta rápida, competência, flexibilidade, rapidez, foco no consumidor e formação de parcerias;
- c) Provedores de Agilidade: fornecem características para os sistemas serem ágeis, através de tecnologia, pessoas, sistemas e estratégias de agilidade;
- d) Escolha de Manufatura: dispõe a escolha da manufatura adequada para atender a agilidade solicitada.

O artigo apontou, por uma abordagem taxonômica, três tipos distintos de estratégias de agilidade e explorou como essas estratégias foram moldadas por mudanças no ambiente de negócios:

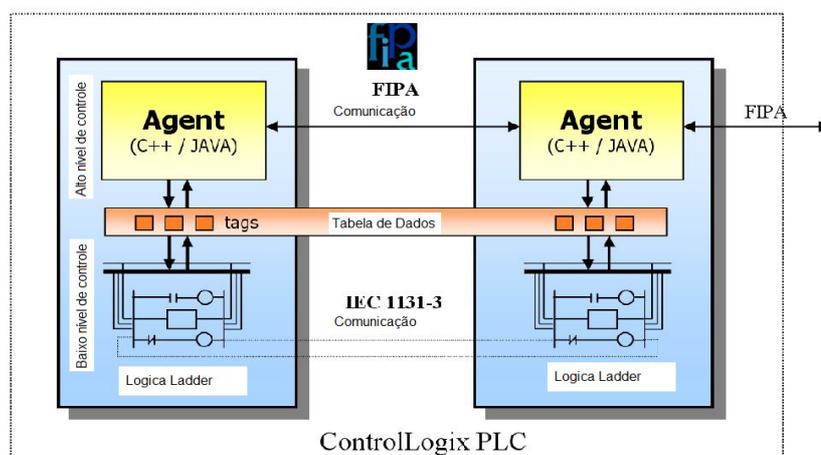
- a) Agilidade como habilidade de satisfazer e estar próximo ao consumidor;
- b) Agilidade como capacidade de prosperar em mudanças que podem ser antecipadas;
- c) Agilidade como habilidade de lidar com mudanças inesperadas.

As três estratégias de agilidade parecem ser comuns em vários setores industriais. Forte evidência para sugerir que as três estratégias foram moldadas por pressões do ambiente de negócios. O resultado do trabalho foi um estudo com a identificação de estratégias de agilidade em empresas, relacionando estas como: empresas responsivas, empresas de resposta rápida e empresas proativas. A caracterização destas empresas determina o grau de agilidade que manifestarão e o quanto estão prontas para as novas demandas de mercado.

Trata-se de um trabalho conceitual que foca a necessidade de agilidades das empresas frente aos novos desafios de mercado. Não trata de uma tecnologia específica, mas aborda os conceitos de forma a caracterizar as indústrias frente ao requisito de agilidade.

No artigo proposto por **Vladimir Marik, Pavel Vrba, Ken Hall e Francisco Maturana** (MARIK et al., 2005) apresenta uma abordagem do centro de pesquisa da Rockwell Automação, em que são utilizados agentes para manufatura como forma de obter flexibilidade e reconfiguração, a fim de resolver os problemas de robustez no processo industrial. Visa atender então a necessidade de agilidade para os sistemas produtivos através da implementação de agentes holônicos (BABICEANU; CHEN, 2006).

Estes agentes são compostos de um sistema de controle de alto nível, onde são executadas as lógicas dos agentes e suas interatividades com outros agentes e um sistema de controle de baixo nível, onde são executadas as lógicas do controlador lógico programável e seu controle local. O interfaceamento entre estes dois sistemas ocorre através de uma tabela de dados, que é compartilhada entre as duas camadas. Nesta tabela encontram-se as variáveis (*tags*) que indicam o estado da planta física ou inferem situações sobre esta planta. A Figura 42 mostra as duas camadas de controle.



**Figura 42** Arquitetura do agente holômico na proposta da Rockwell Automação.

Fonte: Adaptado de (MARIK et al., 2005).

Nesta figura fica claro o compartilhamento entre as duas camadas através da tabela de dados e suas *tags*. Na camada de baixo nível ocorrem as lógicas em *ladder*. Já na camada de alto nível ocorre a execução do agente.

No desenvolvimento da Rockwell Automação a implementação física destas duas camadas ocorrem na forma de um *slot* que é inserido no controlador lógico programável, fazendo com que o agente tenha acesso à tabela de dados do controlador lógico programável. Fato então que se trata de um hardware dedicado para comunicação do agente com o CLP.

Como resultados o artigo apresenta o sistema de manufatura com as características de *plug-and-produce*, assim como a eficiência do Sistema Multiagente frente às necessidades de agilidade para manufatura.

A contribuição destes autores é um sistema de agente holônico que permite uma lógica do controlador lógico programável atuar em um Sistema Multiagente, e com estas funcionalidades possam atender aos requisitos de agilidade para os sistemas produtivos.

#### **4.3 AUTO-ORGANIZAÇÃO EM SISTEMAS PRODUTIVOS**

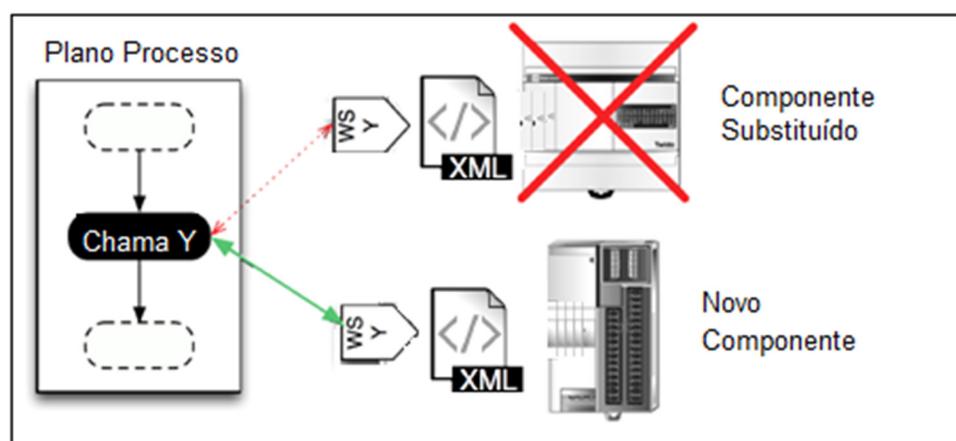
A capacidade de sistemas produtivos provocarem novos arranjos de seus recursos é um fator determinante para se atingir a auto-organização. Em um sistema produtivo a inserção ou retirada de recursos passa a ser um problema de escalabilidade que deve ser atendido. Assim como a tolerância a falha, onde a falta de um componente não deve provocar a parada do sistema produtivo, mas sim provocar a auto-organização do fluxo de processos, para recursos similares ao recurso que entrou em falta supram a funcionalidade que deixou de existir. A auto-organização também remete à capacidade dos sistemas se integrarem formando coalizões,

compartilhando recursos e provocando funcionalidades que até então individualmente não possuíam. Nesta linha algumas pesquisas foram realizadas são relatadas a seguir.

Em sua tese de doutorado, **Gonçalo Cândido** (CÂNDIDO, 2013) aborda questões relacionadas à necessidade de substituição de componentes de sistemas automatizados, onde o componente substituto não cumpre de imediato os requisitos solicitados para o sistema.

A problemática abordada diz respeito ao suporte ao longo do ciclo de vida ao nível do dispositivo. Afinal, um sistema de automação concebido requer que haja dispositivos de reposição, que em função das novas tecnologias, podem ser difíceis de encontrar reposição, o que implicaria em ter que reprogramar todo o sistema para agregar este novo componente.

O método propõe o uso de *web services*<sup>10</sup> para integrar os componentes de automação com os serviços a serem disponibilizados. Os componentes de automação devem oferecer suas funcionalidades através destes serviços, os quais permitem uma abstração dos componentes físicos e seus detalhes de implementação e configuração, facilitando assim a substituição de componentes. A Figura 43 demonstra o princípio da proposta.



**Figura 43 Proposta para envelopar as funcionalidades de um dispositivo.**

Fonte: adaptado de (CÂNDIDO, 2013).

<sup>10</sup> São componentes que permitem às aplicações enviar e receber dados em formato XML. Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal.

A figura apresenta um plano de processo em que há uma chamada para um serviço chamado “Y”. Tanto o componente substituído quanto o componente novo executam o mesmo serviço, porém, são componentes de tecnologias distintas.

Mesmo que o dispositivo físico mude, ele está encapsulado de tal forma que para o sistema a ação é idêntica. Para a produção o que importa é descobrir um serviço, não importa onde ele esteja.

Para implementar este método o autor utiliza uma arquitetura orientada a serviços, modelando os sistemas de manufatura com foco nos serviços disponibilizados. Utiliza *web services* para comunicar os serviços com o dispositivo físico, valendo-se da linguagem “xml” para a descrição dos serviços.

Este método foi validado com a implementação de sistemas de manufatura integrada dentro do conceito de SOA (*Service Oriented Architecture*), com os componentes modelados como agentes de serviços, abstraída as características físicas de cada componente.

Como resultado o trabalho apresentou capacidade de suportar intervenções de reengenharia ao longo do ciclo de vida de um sistema de automação industrial orientada a serviços, a partir de uma arquitetura de referência modular, composta por um conjunto de ferramentas de interoperabilidade e serviços.

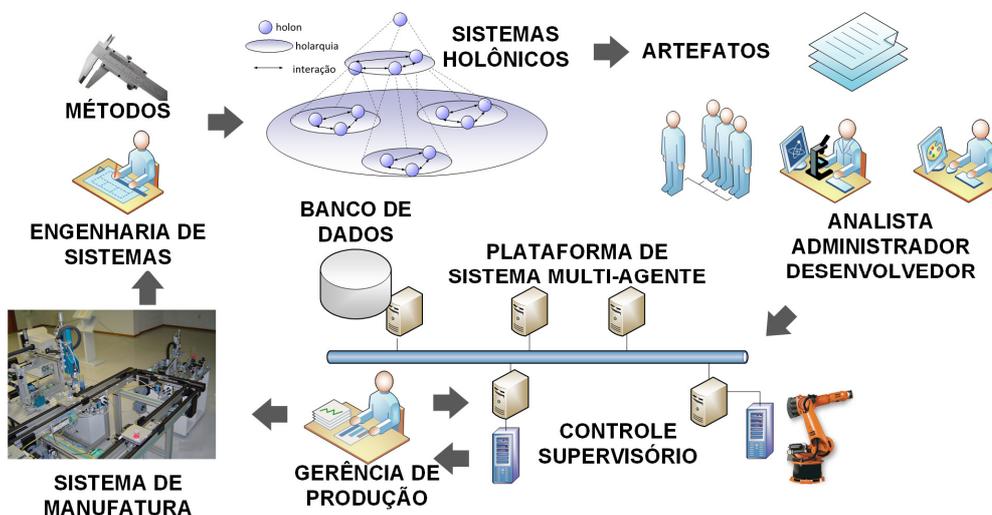
A contribuição do trabalho foi o método que padroniza o componente na forma de serviço, abstraindo as características físicas (elétricas, mecânicas, etc.) e focando nas funcionalidades que serão traduzidas em serviços a serem disponibilizados.

O trabalho de **Robson Marinho da Silva** (SILVA, 2016), na sua tese de doutoramento, é motivado pela necessidade de reconfiguração que os sistemas de manufatura de empresas

demandam, de forma a atender diversidade de produtos através da reorganização do fluxo produtivo, auto-organizando os processos de produção.

O autor propõe uma abordagem com o uso de Sistemas Multiagentes e arquitetura de Holons, o que chama de HMAS (*Holonic Multi-Agent System*), com sinergia aos Sistemas Orientados a Serviços (SOA – Service-Oriented Architecture) no que diz respeito a postagem de serviços, disponibilizando-os a outros agentes de um sistema de manufatura. A arquitetura proposta baseia-se em um sistema para criar e modificar processos produtivos, abstraindo a estrutura e a operação do sistema real a partir de modelos adequados. Por meio da arquitetura de controle especifica os componentes do sistema, suas funções e suas interações.

A Figura 44 apresenta a arquitetura proposta pelo autor para o controle do Sistema Reconfigurável de Manufatura.

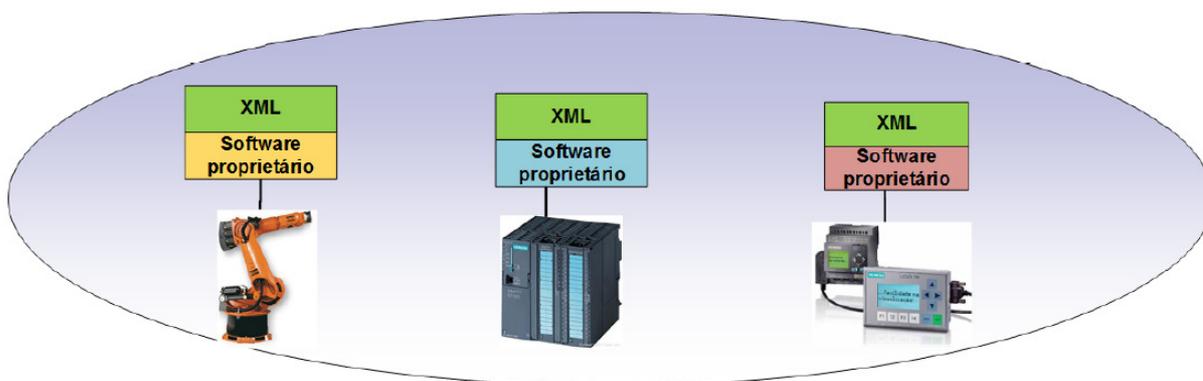


**Figura 44** Arquitetura para Controle de Sistemas Reconfiguráveis de Manufatura.

Fonte: (SILVA, 2016).

Nesta arquitetura de controle, quando um processo produtivo solicita um recurso, na verdade está solicitando aos *holons* para verificar as funcionalidades requeridas, escolhendo aquela que, de acordo com o plano de produção requerido, atenda aos objetivos planejados.

A interface entre o sistema de controle virtualizado e o meio real é realizado através de linguagem XML, com códigos dedicados a cada componente específico, como consta na Figura 45 a seguir, adaptada da figura do autor.



**Figura 45 Interface entre o HRMS e o componente físico por XML.**

Fonte: adaptado de (SILVA, 2016).

Para implementação das funções de controle o autor utilizou a linguagem JAVA com a plataforma JADE e sua extensão WADE, formando uma plataforma de código aberto para o desenvolvimento de sistemas multiagentes, onde o conhecimento não é alocado em um único servidor, mas distribuído para aumentar a robustez do sistema.

São então definidos *holons* especializados para tratar de funcionalidades específicas, como constam:

- a) *Holon* de produto (PrH – *Product Holon*): que representa a matéria prima, produtos intermediários e os produtos finais;
- b) *Holon* de estratégia (TH – *Task Holon*): que são as estratégias planejadas para atender uma ordem de cliente, monitorando e requisitando serviços disponíveis;
- c) *Holon* supervisor (SuH – *Supervisor Holon*): que contém o conhecimento necessário para coordenar as operações que compõe os processos produtivos,

registrando a inclusão de novos *holons* de acordo com as habilidades de cada componente.

Como resultado o trabalho apresentou uma plataforma que permite análise da reconfiguração de um sistema de manufatura e seu impacto no processo produtivo, através da virtualização do processo e utilização de *holons* especializados para tratar cada parte do processo produtivo.

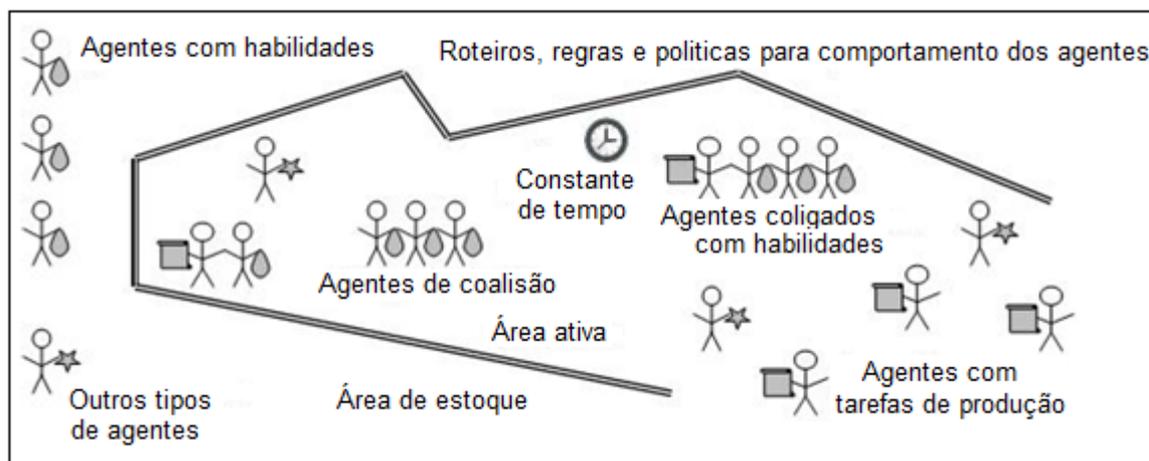
Este trabalho traz como contribuição científica uma plataforma que agrega os conceitos de *holons* e de Sistemas Multiagentes para modelagem de sistemas reconfiguráveis de manufatura.

A tese de **Regina Maria Frei Santos Barbosa** (BARBOSA, 2010) é motivada pela necessidade de encontrar soluções inovadoras para lidar com os desafios postos pelos mercados de hoje e do futuro.

A autora propõe que as propriedades de auto-organização podem aliviar os problemas causados pela necessidade operar com equipamentos complexos, permitindo ao projeto a manufatura tornar a complexidade do equipamento transparente ao usuário.

Para a fase de implementação o modelo da reação química foi utilizado como mecanismo de auto-organização, onde os módulos agentificados formam coalizões segundo regras e reação.

A Figura 46 demonstra a configuração dos agentes e as coalizões que se formam.



**Figura 46 Proposta para proceder as coalizões entre agentes para formar habilidades específicas.**

Fonte: (BARBOSA, 2010).

No trabalho a autora aponta que no sistema de produção há quatro diferentes níveis de sofisticação no controle, diferenciados como:

- a) Controle Reativo: baseado somente no estado corrente;
- b) Controle Retrospectivo: considera os estados passados e trajetórias realizadas pelo corrente estágio;
- c) Controle Preditivo: faz pequenas previsões das várias alternativas pra tomada de decisão e continuar a operação;
- d) Controle Proativo: tenta guiar o sistema através de estados que já obtiveram sucesso.

A coalizão de agentes permite formar as habilidades necessárias para proceder aos devidos controles que um sistema produtivo requer.

Na tecnologia empregada para implementar este modelo, os módulos são programados em linguagem XML, listando suas especificações e utilizando a ferramenta computacional Maude, mecanismo de reescrita que oferece execução e suporte de análise para reescrever especificações lógicas, que se referem ao encadeamento dos processos para formar o produto.

Os resultados foram obtidos a partir de uma implementação em uma planta demonstradora composta de três robôs industriais, cinco esteiras transportadoras, três alimentadores de peças e três garras robóticas. Os resultados mostram que os mecanismos de auto-organização são capazes de encontrar uma solução viável para a concepção de um sistema de montagem.

A contribuição principal desta tese é uma proposta de estrutura para sistemas evolutivos de montagem e auto-organização.

O projeto **COSMOS** (COSMOS PROJECT, 2013) é desenvolvido em um consórcio entre a Universidade King's College London e empresas, como: Gamesa, Ibermática, Tekniker, EDAG, Phoenix Contact, Holos, Ascolab, Reis Robotics, OST, Zema, STT e Universidade de Aachen. O principal objetivo do projeto é o desenvolvimento e a implementação de um sistema de controle distribuído para a gestão de uma fábrica, com uma abordagem de automação flexível, modular e evolutiva, capaz de aumentar a produtividade da fábrica de montagem, sem comprometer a organização dos recursos do sistema fabril. Assim, atende a necessidade de auto-organização do sistema produtivo em função de demandas que se modificam ao longo do processo.

Os principais objetivos do projeto são alcançados, satisfazendo as seguintes ações técnicas:

- a) Criar um conceito de organização da fábrica com base em unidades fabris inteligentes que facilitem a autoadaptação às mudanças no produto / produção em uma base de configuração de automação flexível e modular;
- b) Projetar e desenvolver uma arquitetura de sistema de controle distribuído, que leva em conta a organização da unidade de fábrica concebida e sua configuração modular, e, além disso, capaz de sincronizar diferentes unidades fabris;

- c) Projetar e desenvolver a infraestrutura de camada de serviço entre o sistema de controle e os equipamentos / dispositivos envolvidos nas atividades de produção. Esta camada de serviço permitirá:
- a. A interoperabilidade com os novos dispositivos nas unidades fabris modulares de um modo *plug-and-produce* contínuo, facilitando a escalabilidade da solução;
  - b. A integração de subsistemas automatizados orientados para a sequência de tarefa a ser executada, uma vez que está previsto para ser o caso para uma variedade de operações de montagem, teste e inspeção complexos;
  - c. Definir modelos de custos para otimizar a configuração da unidade da fábrica em termos de automação e modularização. Tais modelos deverão considerar também o potencial das capacidades de adaptação com relação a novas demandas de produtos.

O resultado deste projeto é o desenvolvimento de uma arquitetura de sistema de controle distribuído, que levará em conta a organização da unidade de fábrica concebida e sua configuração modular. E como contribuição o projeto prevê então a disposição de um conceito de organização da fábrica com base em unidades fabris inteligentes que facilitem a autoadaptação.

O projeto **IDEAS** (*Instantly Deployable Evolvable Assembly Systems*) (IDEAS PROJECT, 2010) foi desenvolvido pela Universidade Nova de Lisboa – Portugal e Universidade de Nottingham – Reino Unido, além dos departamentos de pesquisa da MASMEC, Elsest, TEKS, Karlsruhe Institute of Technology- KIT, FIAT, Eletrolux, FESTO e KTH. Ele visou atender a necessidade de adaptação dos componentes de um sistema de manufatura às necessidades do meio produtivo de se organizar, a fim de atender a entrada e/ou saída de recursos no sistema produtivo. O projeto apresentou a proposta de desenvolver

sistemas com autoconfiguração, adaptáveis, com autodiagnóstico, com os componentes orientados a processos e mudança do foco tecnológico de sistemas de manufatura, que agora se vale de Sistemas Multiagentes.

Enquanto tecnologia empregada, o projeto propôs embarcar ambientes de Sistemas Multiagentes em controladores lógicos industriais com capacidade e disponibilidade de execução dos agentes em *software*, para explorar aplicações reais e experimentação no domínio industrial, ao nível de dispositivo. Tais controladores possuem sistema operacional embarcado que permite a execução dos agentes e acesso aos periféricos de entrada e saída. Portanto, com o controlador industrial com esta capacidade, o agente é executado diretamente no componente que tem acesso aos periféricos do sistema físico, o que facilita o controle das lógicas locais direto pelo agente.

Como contribuição o projeto se propôs a fornecer uma aplicação do sistema de montagem industrial em protótipo, como para evidenciar as novas abordagens de controle.

O artigo proposto por **Pavel Vrba** e **Vladimir Marik** (VRBA;MARIK, 2010) se concentra em aspectos de reconfiguração de sistemas de controle industrial baseados em agentes e apresenta a reconfiguração como a capacidade do sistema de controle de modificar seu comportamento, para lidar com condições incomuns e mudanças do ambiente de produção, como quebra de máquinas, inovações de produtos, pedidos urgentes, entre outros. Buscando assim que o sistema produtivo se auto-organize.

Os autores apontam que, do ponto de vista do fabricante de controlador lógico programável (CLP), há a necessidade de integrar as novas tecnologias de controle baseados em agentes como este. Sugerem a abordagem que baseia-se em acoplar o componente de agente de software de alto nível com o de baixo nível em um controlador lógico programável, formando

uma única entidade, definida como *holon*<sup>11</sup> (BABICEANU; CHEN, 2006) (LEITÃO; RESTIVO, 2006) (VRBA et al., 2011).

Neste trabalho é apresentado um Sistema Multiagente dedicado para controle de fluxo de matéria, ao qual se intitula *Manufacturing Agent Simulation Tool* (MAST), também abordado em (PECHOUCEK; MARIK, 2008), que é uma ferramenta de demonstração de forma transparente ao usuário e baseada em agentes aplicado ao domínio da automação industrial. O objetivo foi implementar uma solução baseada em agentes para manipulação de material, mas o sistema MAST desenvolvido pela Rockwell Automation (VBRA, 2006) evoluiu para uma simulação fundamentada em agentes com um sistema de controle em tempo de execução, seguindo uma interface dedicada com o CLP. Esta interface produzida para o CLP permite que os agentes interajam, ou com o subsistema de simulação ou com o sistema físico, através do compartilhamento de valores dos sensores e atuadores na memória do padrão CLP.

O trabalho discute em detalhes como agentes lidam de forma eficiente com a reconfiguração dinâmica, analisando dois cenários: a reconfiguração automática dos caminhos de transporte por parte dos agentes em caso de falhas e a reconfiguração estrutural que permita adição, remoção ou religamento de equipamentos de transporte ou caminhos no chão de fábrica.

Os resultados apresentados foram: a distribuição completa do conhecimento e tomada de decisão, relacionada com a busca e encaminhamento do produto, sem dependência de qualquer unidade de controle central; o algoritmo de caminho de busca baseado em troca de mensagens, concebido como uma parte integral do comportamento social de agentes; o apoio à reconfiguração dinâmica eficiente do controle do sistema em caso de detecção de uma falha ou dentro da modificação do leiaute da produção.

---

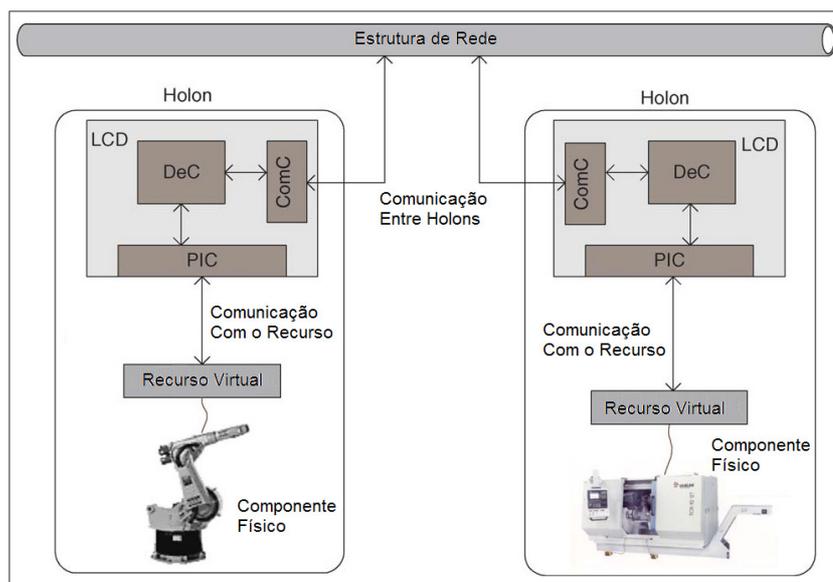
<sup>11</sup> Entidade autônoma e cooperativa que é uma parte identificável de um sistema (de fabricação), que tem uma identidade única, por sua vez faz parte de um todo maior.

A contribuição deste estudo foi a aplicação do conceito MAST em sistemas de transporte, valendo-se de Sistemas Multiagentes para negociação e definição de rotas. Este conceito pode ser útil à proposta desta tese, uma vez que haverá negociação entre agentes e estes terão que interagir com o meio físico, que será um CLP, assim como na proposta deste trabalho. Fato que no estudo de Pavel Vrba e Vladimir Marik está sendo utilizado um CLP com um hardware dedicado, em que já está previsto pelo fabricante o acesso desta interface (hardware) com as memórias, entradas e saídas do CLP.

O artigo publicado por **Paulo Leitão e Francisco Restivo** (LEITÃO; RESTIVO, 2006) relata aspectos econômicos, demanda de mercado e a necessidade de produtos de qualidade e a baixo custo. Nesta problemática os autores identificam então que, para atender ao desejo de customização em massa dos produtos, os sistemas de manufatura devem ser auto-organizáveis, para dar conta dos requisitos que se apresentam.

O trabalho aborda os Sistemas de Manufatura Holônicas (*Holon Manufacturing Systems - HMS*) (BUSSMANN, 1998) em uma arquitetura intitulada ADACOR (*ADaptative holonic COntrol aRchitecture*), que tem a intenção de contribuir para a melhoria do desempenho dos sistemas de controle de processos de fabricação em termos de reação à emergência e mudança, aumentando a agilidade e flexibilidade da empresa, funcionando em ambientes voláteis, caracterizados por frequentes ocorrências de distúrbios. O foco da arquitetura ADACOR é a fabricação em sistemas flexíveis de produção, caracterizado por processos simultâneos e assíncronos, operações não preditivas e rotas alternativas.

A Figura 47 mostra o modelo de uma estrutura para o *holon* na plataforma ADACOR.



**Figura 47** Conceito do modelo para um *holon* na plataforma ADACOR.

Fonte: adaptado de (LEITÃO; RESTIVO, 2006).

Nesta figura se percebe então o módulo de Controle Lógico do Componente (LCD), o módulo de Decisão (DeC), o módulo de Comunicação (ComC) com a rede e o módulo de Interface com o Componente Físico (PIC). Este último traz um aspecto a destacar: a comunicação entre o holon e o componente físico é realizada através de uma conexão dedicada, onde tanto o módulo PIC quanto o componente físico possuem códigos definidos para este propósito, de estabelecer a comunicação e troca de informações.

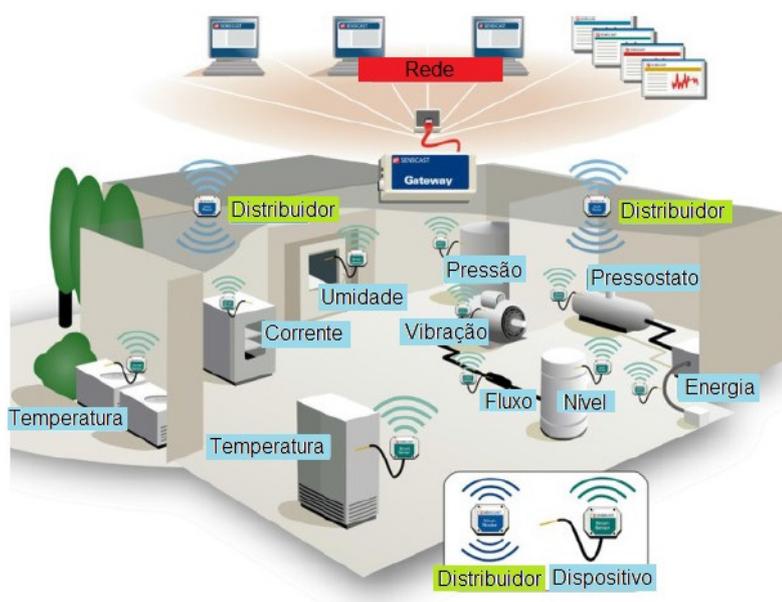
A proposta apresentada pelos autores melhorou o desempenho do sistema em cenários industriais caracterizados pela ocorrência frequente de perturbações inesperadas, com destaque a abordagem colaborativa que manifestaram.

A contribuição deste trabalho foi o *case*<sup>12</sup> de uso da plataforma ADACOR em cenário industrial, com desempenho adequado às necessidades de auto-organização dos sistemas produtivos.

<sup>12</sup> Case é uma situação de aplicação de um conceito ou método, em que as variáveis de ambiente são as mais reais possíveis, retratando um caso de uso prático, com resultados válidos para análise.

O artigo publicado por **Edison Freitas, Ivan Muller, Rodrigo Allgayer, Carlos Pereira, André Cavalcante, Marco Marinho, João Costa e Rafael Júnior** (FREITAS et al., 2013) trata os problemas relacionados com o dimensionamento e especialmente na adição de novos dispositivos para reorganização do ambiente de produção fabril, recomendando o uso de conexões sem fio entre as diferentes unidades de produção. Nesta problemática os autores identificam que, para atender ao desejo de auto-organização do sistema produtivo, o uso de conexões sem fio que permitam que os agentes migrem de posições dentro do processo e mantenham sua conexão podem apresentar problemas quanto a determinação temporal dos sinais, principalmente quando os processos envolverem malhas de controle. O trabalho aponta que num ambiente industrial, em que os produtos são diferentes, processados na mesma linha de produtos, a detecção distribuída corrobora como alternativa para as interações entre unidades de processamento e produção, a fim de fornecer o necessário suporte para processar corretamente cada produto. E neste cenário, uma abordagem com sistemas multiagentes pode ajudar a proceder a configuração necessária.

Este estudo propõe o um sistema produtivo com uso de sistemas multiagentes, onde as interações entre eles ocorrem através de conexões sem fio, como forma de garantir a mobilidade dos componentes na planta. A Figura 48 apresenta uma planta configurada com sistemas multiagentes, onde as interações ocorrem por conexões sem fio.



**Figura 48** Planta com sistemas Multiagentes interagindo por conexão sem fio.

Fonte: Adaptado de (FREITAS et al., 2013).

Neste artigo foi analisado o comportamento de uma malha de controle de posicionamento robótico, uma das estações da planta sugerida como agente, face a necessidade de interagir com os demais integrantes do sistema de manufatura, através de suas mensagens pela rede de conexão sem fio. Fator importante a ser levado em consideração no momento de implementar um sistema auto-organizável.

O estudo apresentado pelos autores mostrou a aplicabilidade de utilizar sistemas auto-organizáveis baseados em multiagentes, utilizando conexões sem fio para proceder suas interações, mas com a ressalva de que há sim uma interferência no controle, devido as variações temporais que este meio de comunicação provoca, cabendo um melhor aprimoramento desta tecnologia específica.

A contribuição deste trabalho foi o estudo do comportamento e interferência da utilização de redes sem fio para interação entre os agentes do sistema auto-organizável.

#### 4.4 ANÁLISE DOS RESULTADOS SOBRE AS ABORDAGENS

Analisando as pesquisas nota-se que há um interesse pelo tema e que soluções já foram desenvolvidas, porém ainda são muito modestas as iniciativas em termos de quantidade. Há abordagens em comum utilizando tecnologias similares para solução de problemas distintos, assim como o uso de tecnologias similares com abordagens distintas.

Além das formas de abordagem e dos resultados atingidos, as pesquisas estudadas também trazem meios tecnológicos importantes que podem ser utilizados no desenvolvimento desta proposta. A Tabela 6 descreve a contribuição tecnológica que cada pesquisa investigada ofereceu para o desenvolvimento desta tese.

**Tabela 6 Contribuições das pesquisas estudadas para o desenvolvimento desta tese.**

Referência da Pesquisa	Contribuição Conceitual ou Tecnológica para a tese.
CAVALCANTE, 2012	Agente dissociado: um para deliberação e outro para ações diretas.
ROLOFF, 2014	Paradigma Vogal (AEIO) - SCADA para interface com componentes físicos.
PESCHL, 2014	Agentes <i>manufactron</i> – entidade autossuficiente capaz de interagir.
SOPRO PROJECT, 2014	Aplicação industrial de agentes móveis.
EUPASS PROJECT, 2014	Aplicação industrial de agentes móveis.
FREI; SERUGENDO; BARATA, 2008	Aplicação de EAS como forma de conseguir a reconfiguração.
OTHMANE; HEBRI, 2012	Utilização de MAS em nuvem, provocando a interação entre os agentes.
OLIVEIRA, 2003	<i>Agent Machine Interface</i> (AMI) que é o agente que está diretamente ligado ao controlador físico.
RIBEIRO, 2012	Aplicação de EAS como forma de conseguir a predição de comportamento.
XPRESS PROJECT, 2004	Agentes <i>manufactron</i> – entidade autossuficiente capaz de interagir.
PRIME PROJECT, 2014	Multiagentes como meio de atender a heterogeneidade dos componentes.
ZAHNG, SHARIFI, 2007	Identificação de estratégias de agilidade em empresas.
MARIK et al., 2005	Interfaceamento entre o CLP e o agente através de uma tabela de <i>Tags</i> no mesmo <i>hardware</i> .
CÂNDIDO, 2013	<i>Web services</i> como meio de interfacear os componentes físicos com o processo.
SILVA, 2016	<i>Holons</i> em uma estrutura de RMS para gestão da reconfiguração.
BARBOSA, 2010	Formação de coalizões entre os agentes.
COSMOS PROJECT, 2013	Dispositivos nas unidades fabris modulares um modo <i>plug-and-produce</i> .
IDEAS PROJECT, 2010	Execução do agente diretamente no componente físico.
VBRA; MARIK, 2010	MAST, demonstração de forma transparente e baseada em agentes aplicados ao domínio automação industrial.

LEITÃO; RESTIVO, 2006	Arquitetura intitulada ADACOR, com comunicação dedicada do holon com o componente físico.
FREITAS et al.,2013	Estudo do efeito das conexões sem fio como meio de interação entre os agentes.

Fonte: Autor

Percebem-se aqui grandes contribuições que as pesquisas podem fornecer ao desenvolvimento desta tese, seja como conceito orientativo à forma de abordagem ou sendo uma tecnologia de aplicação direta.

Com relação às abordagens das problemáticas iniciais, as 20 pesquisas foram relacionadas de forma a observar a relação delas comparadas à problemática a ser atendida, a forma de abordagem do problema, os resultados alcançados com a pesquisa e a contribuição científica que o trabalho deixou para a Ciência. A Tabela 7 apresenta um resumo das atividades de pesquisa descritas.

Tabela 7 Relação de abordagens, resultados e contribuição dos trabalhos pesquisados.

Referência da Pesquisa	Problemática apresentada	Abordagem ao problema	Resultados da Pesquisa	Contribuição Científica
CAVALCANTE, 2012	<b>Diversidade.</b> Necessidade de reconfiguração para uma grande gama de produtos customizados.	Arquitetura CoBaSa, com agentes com duas camadas: deliberação e ação. Sistema <i>plug-and-produce</i> . Uso de PC e eletrônica embarcada (ARM7) para os agentes.	O sistema apresentou auto-organização. A medida que novos produtos, entravam em produção eles passaram a concorrer pelos recursos existentes.	Modelo de uma arquitetura híbrida, baseada em agentes, que possibilita tanto deliberação quanto reação, para aplicações na manufatura industrial.
ROLOFF, 2014	<b>Diversidade.</b> Necessidade de atender a Pequenos Lotes de Produção (SSP).	Paradigma Vogal, implementando os ambientes do Agente (Jason), do Ambiente (CArtAgO) e da Organização (Moise+). Comunicação com ambiente físico por SCADA.	MAS para SSP se adequou de forma efetiva, diminuindo o tempo de <i>setup</i> na linha e assumindo atividades no controle de produção.	Modelo de implementação de SMA para PPS que vai à direção dos requisitos desejáveis para um SMA para PPS.
SOPRO PROJECT, 2014	<b>Diversidade.</b> Sistema de produção dinâmica, os requisitos de produção mudam ao longo do processo.	Utilização de plataforma multiagentes para desenvolvimento do sistema. Implementam em sistemas embarcados.	Implantação de agentes móveis que atuam para atender à necessidade dinâmica das mudanças no fluxo produtivo.	Utilização de Sistemas Multiagentes para propor solução atender a necessidade de mudança no fluxo de um processo produtivo.
EUPASS PROJECT, 2014	<b>Diversidade.</b> Atendimento a demandas de montagem muito específicas, quase customizadas.	Plataformas de Sistemas Multiagentes para reconfiguração de equipamentos e módulos dirigidos por requerimentos de produção definidos em uma ontologia.	Atendimento às necessidades de montagem de precisão com soluções em Sistemas Multiagentes.	Modelo utilizando Sistemas Multiagentes para atender a sistemas de montagem de ultraprecisão, em uma demanda quase customizada.
PESCHL, 2014	<b>Diversidade.</b> Atender as mudanças do consumidor, com o incremento de novos produtos e sua variedade.	Gerar a configuração de cada agente a partir da descrição do produto. Utiliza como meio de comunicação a linguagem ACL e KQML, além de FIPA-ACL	Diminuição do tempo para pôr em marcha à produção multivariante; implementação fácil na indústria; geração e uso dos conhecimentos.	Método de implementação de uma arquitetura flexível que atende as diversidades do mercado.
FREI; SERUGENDO; BARATA, 2008	<b>Diversidade</b> Customização em massa, muitos lotes com baixo volume de produção.	Sistemas Evolutivos de Manufatura – EAS, com os conceitos de autoadaptação dos componentes, possibilitando <i>plug-and-produce</i> . Utiliza JADE e em JAVA.	Implementação da estratégia <i>plug-and-produce</i> , que garantiu a entrada e saída de estações nos sistemas, provocando assim sua reconfiguração.	Sistemas Multiagentes implementando os conceitos de EAS, evidenciando os benefícios das características de <i>plug-and-produce</i> a um sistema produtivo que necessita de diversidade.

OTHMANE; HEBRI, 2012	<b>Auto-organização</b> Reconfigurar os sistemas produtivos em função dos requisitos de produção.	Modelo a partir do conceito de nuvem, implementando o MAS. Agentes buscam dados na nuvem, compartilhados por outros agentes, visando a melhor forma de otimização.	Eficácia da busca por informações que levaram o sistema a se autoconfigurar para atender as necessidades de reconfiguração do Sistema Multiagente.	Nova abordagem para a busca de dados distribuídos com base em sistemas de computação em nuvem e Sistema Multiagente, em uma arquitetura modular.
OLIVEIRA, 2003	<b>Agilidade</b> Lidar com os distúrbios e incertezas, que refletem os cenários de negócios enfrentados pelas empresas	Arquitetura multiagente baseada em CoBASA, criada para apoiar a adaptação e mudanças de arquiteturas de controle. Utiliza a plataforma JADE junto com aplicativo Protégé.	Atendimento das necessidades de agilidade das tomadas de decisões e composição de recursos, formando <i>clusters</i> .	Uma arquitetura baseada em multiagentes, chamada CoBASA, criada para apoiar o processo de reengenharia do chão de fábrica.
RIBEIRO, 2012	<b>Agilidade</b> Necessidade de agilidade na execução de diagnóstico em instalações industriais complexas.	Abordagem dos EPS/EAS. Utilização de Sistemas Multiagentes programados em JAVA dentro da plataforma JADE.	Exploração das interações e informações locais para promover uma resposta ao diagnóstico de auto-organização em um sistema EPS/EAS.	Método formalizado para diagnóstico de falhas e um banco de diagnóstico em ambiente de auto-organização em rede.
XPRESS PROJECT, 2014	<b>Agilidade</b> Fabricante responder às rápidas mudanças das necessidades dos consumidores	Conceito de “ <i>Manufactrons</i> ”, unidades de processo inteligentes. Atua na configuração da produção, controle de fluxo e na interação com os componentes de campo.	Prevê que a implementação dos dispositivos <i>Manufactrons</i> sejam capazes de responder às mudanças das necessidades dos consumidores.	Desenvolvimento de um dispositivo que permite ganhos ao processo produtivo no tempo para a implementação de novos processos e na preservação do conhecimento.
PRIME PROJECT, 2014	<b>Agilidade</b> Necessidade dos sistemas produtivos de reconfigurar seus sistemas em função dos requisitos de mercado.	Desenvolvimento de ambiente de produção por MAS para integração dos módulos, incluindo equipamentos legados, com implantação de agentes em controladores industriais.	Funcionalidades de <i>plug-and-produce</i> do Sistema Multiagente resolve a necessidade de reconfiguração, face a heterogeneidade.	Experimentação do Sistema Multiagente com as funcionalidades de integração entre os módulos em demonstradores de reais de processos produtivos.
ZAHNG, SHARIFI, 2007	<b>Agilidade</b> Agilidade como uma estratégia para evitar falhas em manufatura e instabilidade no negócio.	O método estrutural proposto prevê quatro elementos principais da agilidade como estratégia de fabricação. Não emprega diretamente uma tecnologia.	Estudo com a identificação de estratégias de agilidade: empresas responsivas, empresas de resposta rápida e empresas proativas, determinando o grau de agilidade.	Um estudo em empresa, destacando a importância da agilidade no processo produtivo, fundamentando o tema.
MARIK et al., 2005	<b>Agilidade</b> Necessidade de robustez e agilidade no processo de manufatura industrial.	Implementação de agentes holônicos. CLP com o agente, compartilhando tabela de variáveis com hardware é dedicado para a implementação da Rockwell Automação.	Sistema de manufatura com as características de <i>plug-and-produce</i> . Atendendo às necessidades de agilidade na produção.	Sistema de agente holônico que permite uma lógica para o controlador lógico programável atuar em um Sistema Multiagente.

SILVA, 2016	<b>Auto-organização</b> Proceder a auto-organização do sistema produtivo a partir de demandas diversificadas	Desenvolvimento de uma plataforma HRMS para atender à necessidade de reconfiguração, valendo-se de agentes <i>holômicos</i> .	Plataforma que permite análise da reconfiguração de um sistema de manufatura e seu impacto no processo produtivo, através da virtualização.	Plataforma que agrega os conceitos de <i>holons</i> e Sistemas Multiagentes para modelagem de sistemas reconfiguráveis de manufatura.
BARBOSA, 2010	<b>Auto-organização</b> Responder rapidamente aos pedidos dos clientes, com volumes de produção variáveis e distúrbios.	Propriedades de auto-organização, permitindo tornar a complexidade do utilizador transparente ao usuário. Linguagem XML com ferramenta Maude.	A tecnologia programada em linguagem XML, uso de Maude, como mecanismo de reescrita que oferece execução e suporte de análise das especificações lógicas.	Proposta de estrutura para Sistemas Evolutivos de Montagem e Auto-organização, visando a abstração da tecnologia do componente e focando somente nos serviços que ele presta.
CÂNDIDO, 2013	<b>Auto-organização</b> Necessidade de substituição de componentes de sistemas automatizados.	Uso de <i>web services</i> para integrar os componentes de automação com os serviços a serem disponibilizados, linguagem "xml" para comunicar os serviços com o componente físico.	Capacidade de suportar intervenções de reengenharia ao longo do ciclo de vida de um sistema de automação industrial orientada a serviços.	Método que padroniza o componente na forma de serviço, abstraindo as características físico-elétricas e focando nas funcionalidades que serão traduzidas em serviços.
COSMOS PROJECT, 2013	<b>Auto-organização</b> Atender a necessidade de criar unidades fabris inteligentes que facilitem a auto-organização.	Comportamento autônomo; controle descentralizado multicamada; interconectados; conectividade interoperável; local; utilização de Sistemas Multiagentes.	Arquitetura de sistema de controle distribuído, que levará em conta a organização da unidade de fábrica concebida e sua configuração modular.	Disposição de um conceito de organização da fábrica com base em unidades fabris inteligentes que facilitem a autoadaptação.
IDEAS PROJECT, 2010	<b>Auto-organização</b> Necessidades do meio produtivo de se organizar, para atender a entrada e/ou saída de recursos.	Embarcar ambientes de MAS em controladores industriais para explorar aplicações reais e validação no domínio industrial ao nível de dispositivo.	Desenvolvimento de arquitetura de sistema de controle distribuído, que levará em conta a organização da unidade de fábrica concebida e sua configuração modular.	Fornecimento de uma aplicação do sistema de montagem industrial em protótipo, para evidenciar as novas abordagens de controle.
VBRA;MARIK, 2010	<b>Auto-organização</b> Atender a condições incomuns e mudanças do ambiente de produção.	Multiagentes com conceito MAST, implementando soluções para transportadores de material. Utiliza agente com CLP, mas com interface dedicada no hardware.	Distribuição do conhecimento e tomada de decisão, com a busca e encaminhamento do produto, sem dependência de unidade de controle central.	Aplicação do conceito MAST em sistemas de transporte, valendo-se de Sistemas Multiagentes para negociação e definição de rotas.
LEITÃO; RESTIVO, 2006	<b>Auto-organização</b> Organizar o sistema produtivo para atender a necessidade de produtos de qualidade e a baixo custo.	HMS com o uso da plataforma ADACOR, caracterizada a atender a processos simultâneos e assíncronos, operações não preditivas e rotas alternativas.	Melhora do desempenho do sistema em cenários industriais caracterizados pela ocorrência frequente de perturbações inesperadas na produção.	Case de uso da plataforma ADACOR em cenário industrial, com desempenho adequado às necessidades de auto-organização dos sistemas produtivos.

FREITAS et al., 2013	<b>Auto-organização</b> Interagir componentes móveis no sistema produtivo.	Utilização de rede de conexão sem fio para promover a interação entre os agentes do sistema produtivo.	A utilização de conexão sem fio como meio de interação é aplicável, porém, provoca alterações temporais na comunicação.	Estudo da interferência temporal na comunicação entre agentes em um sistema de manufatura, quando utilizam rede de conexão sem fio para tal.
----------------------	---	--	---	--

Fonte: Autor

#### **4.5 IDENTIFICAÇÃO DE LACUNAS NA ABORDAGEM DE SISTEMAS MULTIAGENTES PARA SOLUÇÃO DE PROBLEMAS DE DIVERSIDADE, AGILIDADE E AUTO-ORGANIZAÇÃO.**

Analisando cada abordagem, foi identificado que há soluções para que sistemas de manufatura industrial possam atuar dentro do conceito de Sistemas Multiagentes e que dão conta dos requisitos de diversidade, agilidade e auto-organização. Porém, são soluções que remetem à implantação em dispositivos com grande capacidade de processamento, tais como: computador, computador industrial ou sistema embarcado, ou então a inserção de módulos dedicados no hardware da máquina, o que em alguns casos não é viável, já que o fabricante do equipamento físico pode não disponibilizar esta alternativa. Estas soluções levam a indústria a modificar seus sistemas de controle para esta tecnologia.

O fato é que a indústria já possui sistema de controle de seus sistemas produtivos, normalmente controladores lógicos programáveis (CLP) e desfazer-se destes sistemas para implementar o novo conceito pode não ser a melhor alternativa.

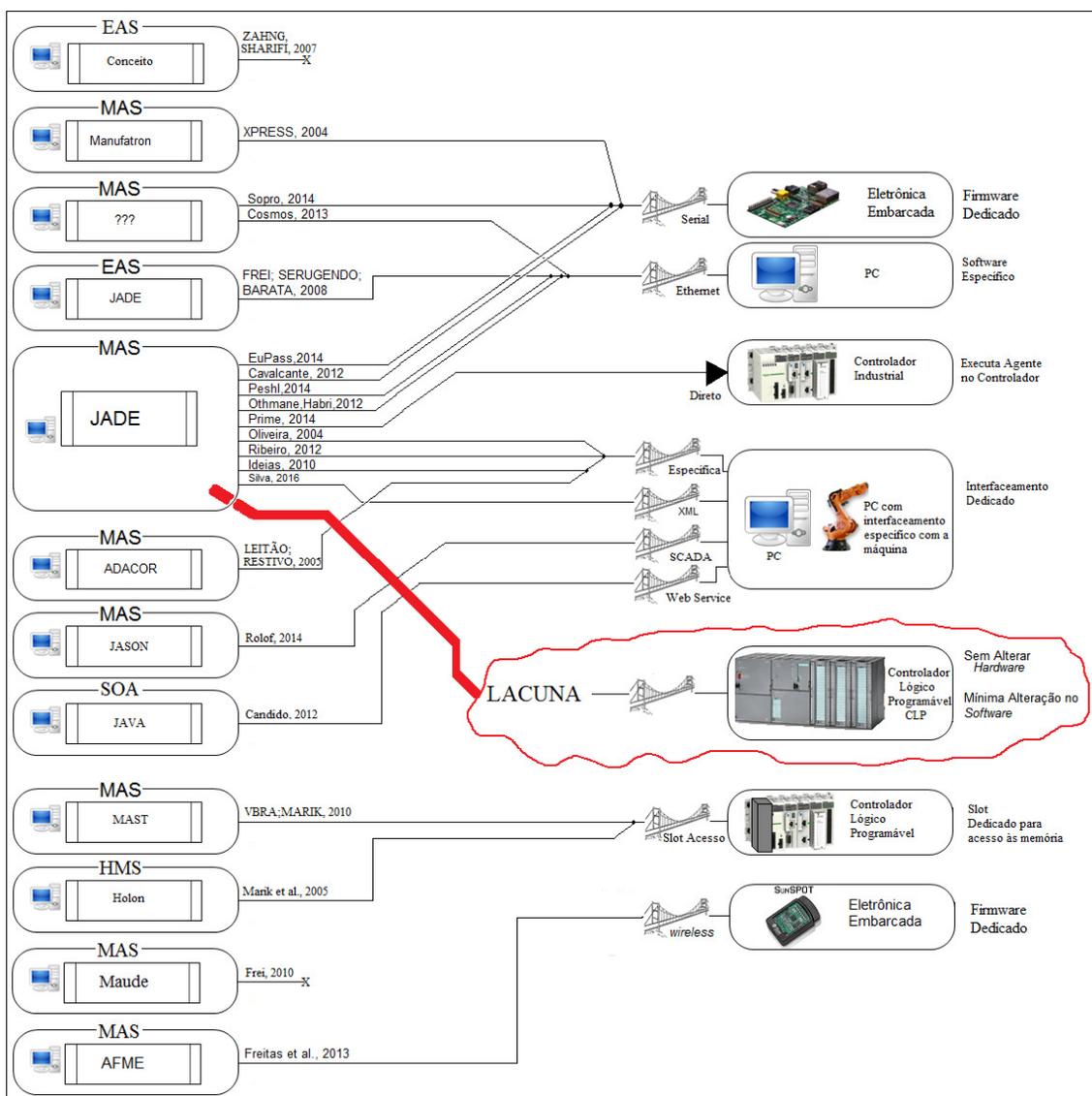
Com o objetivo de construir um mapa que retrate os estudos e pesquisas realizados, foi considerado então os aspectos:

- a) Utilização de sistemas multiagentes para aplicação em manufatura industrial;
- b) Plataforma utilizada para execução dos agentes;
- c) Meio de interação com o componente de gestão da máquina ou equipamento;
- d) Tipo de controlador da máquina ou equipamento a que se está interagindo.

A lembrar que o propósito é atender às necessidades dos sistemas produtivos industriais por diversidade, agilidade e auto-organização, e que a problemática remete

que o uso de sistemas multiagentes junto ao controlador lógico programável possa ser a solução. Portanto, a busca é por pesquisas ou estudos que abordem a utilização de sistemas multiagentes em controladores lógicos programáveis sem interferir na execução de sua lógica local, atendendo os requisitos da manufatura industrial.

De posse destes critérios o mapa da Figura 49 foi montado, apresentando as abordagens de cada projeto de pesquisa e a identificação da lacuna que persiste, alvo de abordagem desta tese.



**Figura 49** Identificação da lacuna na abordagem de sistemas que atendem a diversidade, agilidade e auto-organização.

Fonte: autor.

A Figura 49 identifica os conceitos de manufatura, a tecnologia empregada, o meio de comunicação com o meio físico e o meio físico utilizado. E nela identifica-se uma lacuna nas abordagens estudadas de uma proposta que trate da implantação das funcionalidades dos Sistemas Multiagentes em sistemas de manufatura industrial, aproveitando a estrutura de controlador lógico industrial, sem a necessidade de modificar o seu hardware e alterando minimamente em seu software de controle, sem alterar sua lógica local que rege seu processo de manufatura.

Sob esta ótica que a pesquisa da tese desenvolve, justificando-se a abordagem de controladores lógicos programáveis por serem largamente utilizados no controle de manufatura industrial e por haver uma carência de pesquisas quanto à utilização de Sistemas Multiagentes com este componente.

## 5 PROPOSTA DE TESE

A julgar pela demanda dos sistemas de manufatura de se adequarem às necessidades de agilidade, auto-organização e atendimento à diversidade de produção e pelas soluções que se apresentam, e remeter à necessidade de alterar o controlador e o software instalado, é compreensível que as indústrias relutem tanto para migrar seus sistemas de gestão da manufatura para o conceito de sistemas auto-organizáveis. Mas se houver uma forma de proceder esta migração sem a necessidade de se desfazer do que já existe, do que já está implantado, então os benefícios se tornam mais atrativos.

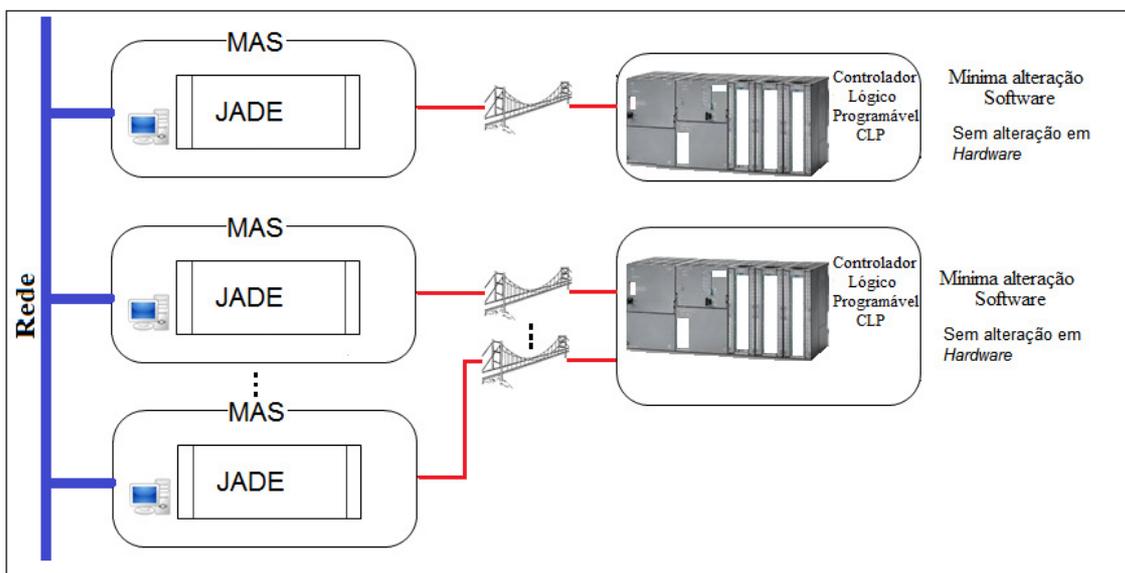
É sabido que no futuro os controladores que são utilizados atualmente serão substituídos por computadores industriais ou sistemas embarcados, e que a preocupação com a capacidade computacional ou outros requisitos de infraestrutura de software e hardware diminuirão. Mas acredita-se que esta mudança não ocorrerá em uma única etapa, de um instante para o outro. Por este motivo a tese proposta neste trabalho tem o objetivo de ser o primeiro passo para que os sistemas de gestão da manufatura migrem de um sistema convencional para um sistema auto-organizável. Um passo importante para que outros possam ser dados.

Cabe às pesquisas acadêmicas buscar soluções para as problemáticas que possam ser aplicadas às necessidades cotidianas, se apresentando como um conceito ou como uma solução de aplicação direta. Neste trabalho a pesquisa busca investigar conceitos e tecnologias que possam gerar um método científico que se proponha conceitualmente a dar solução às problemáticas definidas. E que por fim possam ser aplicados no ambiente produtivo, dando solução a alguns problemas lá existentes.

Motivada pela lacuna identificada no capítulo anterior, uma hipótese é então formulada: “Utilização de Sistemas Multiagentes junto a controladores lógicos programáveis para gerir sistemas de manufatura, de forma a atender os requisitos de

diversidade, agilidade e auto-organização do meio produtivo, sem alterar significativamente os tempos envolvidos no processo produtivo”. Afinal, fora identificado que há pesquisas que visam a utilização de Sistemas Multiagentes no meio produtivo, mas não abordam o fato de que a gestão do meio produtivo é feita, em grande parte dos casos, através de controladores lógicos programáveis (CLP).

O foco deste trabalho passa a ser então uma abordagem da problemática sendo atendida pela utilização de controladores lógicos programáveis com as funcionalidades de sistemas. A Figura 50 apresenta o foco de abordagem da proposta de tese.



**Figura 50** Foco de abordagem da proposta de tese.

Fonte: Autor.

E dentro deste foco é proposta a tese seguinte:

**Tese:**

A aplicação de Sistemas Multiagentes em controladores lógicos programáveis como forma de gestão da manufatura industrial propicia características de diversidade, agilidade e auto-organização ao sistema produtivo.

Caberá ao desenvolvimento da pesquisa provar que esta tese se sustenta, através de argumentos e ensaios práticos, de forma a obter dados que apontem para sua comprovação.

### **5.1 INEDITISMO DA PROPOSTA DA TESE**

A pesquisa inicial já apontou alguns trabalhos que se propõem a atender as necessidades descritas na problemática, utilizando conceitos e métodos distintos. Mas conforme apontado no Capítulo 3, há uma lacuna quanto a abordagens que envolvem a utilização de Sistemas Multiagentes sobre controladores lógicos programáveis. Na investigação até surgiram abordagens que envolvem este tipo de controladores, mas que provocam uma alteração no hardware e no software do controlador. A proposta deste trabalho é que tais alterações sejam as mínimas possíveis, tanto em software como em hardware, procedendo somente a alteração na configuração do controlador lógico programável. Fato este em função da alta demanda deste tipo de dispositivo no meio produtivo industrial como forma de controle.

Portanto, o ineditismo desta tese se sustenta soluções que atendam a sistemas em que o dispositivo de controle é um controlador lógico programável, abordando a inserção das características de Sistemas Multiagentes neste controlador, com o mínimo de alteração em hardware e software. Esta abordagem se apresentou inédita gera um método que atenda conceitualmente ao propósito de tornar a manufatura ágil, auto-organizada e capaz de suprir diversidade de produção no mesmo ambiente de manufatura.

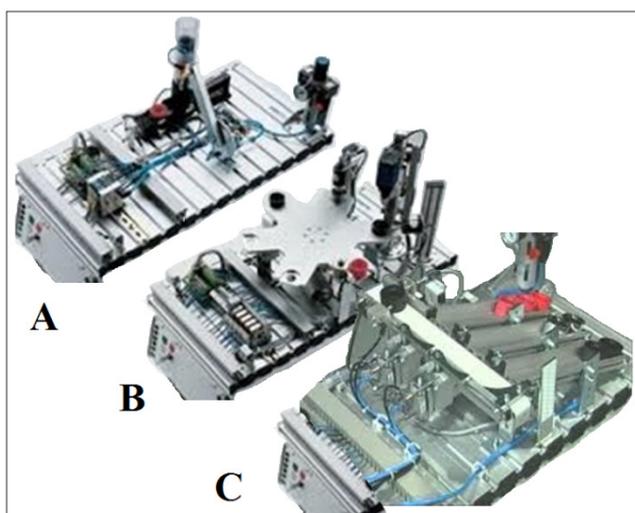
## 5.2 METODOLOGIA

Sistemas de manufatura controlados por controladores lógicos programáveis se caracterizam por ter seu sensoriamento e elementos de comando ligados na entrada do controlador e por ter os atuadores e elementos de sinalização ligados à sua saída. E relacionando as entradas com as saídas existe então um programa, desenvolvido em uma linguagem específica para CLP (regrada pela norma IEC61131), que indica a relação dos eventos de entrada com os eventos de saída.

E para formar um arranjo produtivo as máquinas são interconectadas segundo uma sequência de processos definidos no planejamento de produção.

A este sistema produtivo deverá ser observada algumas características elementares (PEIXOTO, 2012), que irão caracterizar a manufatura e indicar sua disposição a se tornar um agente de serviços, que irá dispor suas funcionalidades para que outros agentes a requisitem.

A Figura 51 apresenta três estações de manufatura, que quando interconectadas elétrica e mecanicamente, podem formar um arranjo produtivo.



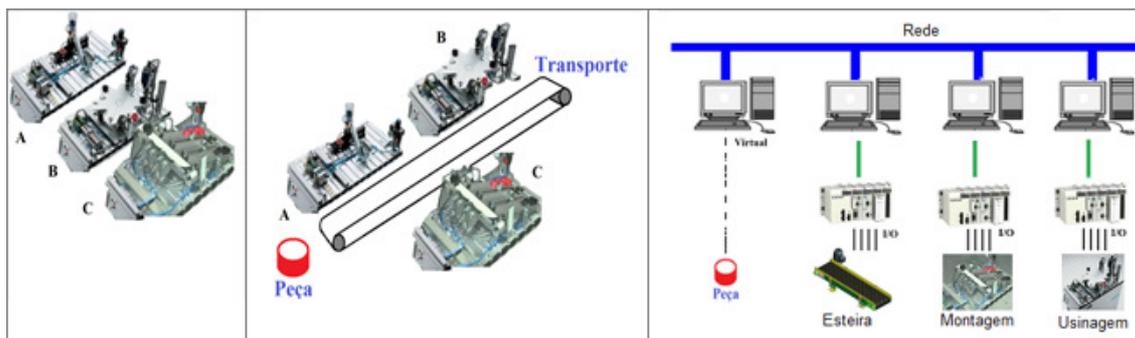
**Figura 51 Exemplo de um sistema integrado de manufatura com a sequência de manufatura definida pela disposição das estações.**

Fonte: Adaptado de <http://www.festo-didactic.com>

Neste modelo tem-se a estação “A” fazendo o papel de entrada de materiais no sistema de manufatura e passagem da peça para a estação “B”, que por sua vez executa o processamento da peça conforme seu tipo, deslocando esta para estação “C” que é responsável pela seleção da peça conforme sua natureza. Neste modelo a sequência de operação será  $A \rightarrow B \rightarrow C$ , definida no planejamento do conjunto. Cada CLP de cada estação controla os periféricos da sua estação, dando-lhe independência em relação às demais, mas trocam informações com os demais controladores lógicos programáveis para que as informações do estágio de produção da peça sejam entendidas pela estação seguinte.

Caso o produto necessite mudar suas características, então terá que se refazer etapas da programação de cada máquina, mesmo que seja só uma questão de redistribuição de leiaute. Já para o atendimento dos requisitos de agilidade, recuperar-se de falhas por exemplo, o sistema de manufatura depende das três estações estarem produzindo, se uma falhar todo o processo falha. E por fim, a necessidade de redistribuição do fluxo de processo não é contemplada, pois o sistema não consegue mudar um componente de posição no fluxo produtivo já definido para o sistema.

As características de um Sistema Multiagente, em tese, levam o sistema a atender as necessidades descritas nos parágrafos anteriores. Mas para isto as estações devem ser interpretadas como elementos dissociados, que haja um meio de transporte da peça entre todas as estações e que haja uma rede de comunicação entre cada estação. Este é o propósito do trabalho, que é apresentado na Figura 52.



**Figura 52 Propósito do trabalho para interligar as estações de manufatura.**

Fonte: Autor.

Aqui se nota pela figura que as estações A, B e C estão dissociadas, podendo atuar independentes e atender ao fluxo de produção, agora definido pela peça, que negociará os serviços de transporte e produção com as estações.

Então, como estudo inicial é necessário entender a manufatura industrial e seus serviços, assim como identificar os sinais que fazem a manufatura ser executada e os sinais que indicam que a manufatura se encerrou. Estas informações são importantes para definir a classe de agentes que serão implementados.

Dentro do paradigma vogal, definido por (DEMAZEAU, 1995), esta proposta visa atuar sobre a dimensão A (*Agent*), através da plataforma JADE, e a imensão I (*Interaction*), através das negociações que ocorrerão entre os agentes dentro da plataforma JADE. As demais dimensões E (*Environment*) e O (*Organization*) não serão abordadas por não ser o objetivo da tese proposta.

Como contribuição do trabalho de (CAVALCANTE, 2012) tem-se a definição das classes de agentes como agentes mecatrônicos, onde é apontado:

- a) RA – *Resource Agent* (Agente de recurso): são os equipamentos que irão dispor seus serviços aos demais agentes;
- b) PA – *Product Agent* (Agente de produto): são as peças/produtos que serão manufaturados. Neste agente é que se encontrará o fluxo produtivo que deverá ser seguido para sua confecção;

- c) TUA – *Transport Unit Agent* (Agente Unidade de transporte): são os sistemas de transportes que efetuarão o destolamento das peças/produtos de uma estação a outra.

Neste trabalho estes agentes não terão contato com o meio físico (neste caso o CLP), será implementado um agente que propicie esta troca de informações. Aqui entra a figura do AgentCom, uma classe desenvolvida para auxiliar o PLCAgent que trata da especificidade da interação com o componente físico. É uma das contribuições deste trabalho para ciência, pois apresenta o modelo de classe que torna os serviços do componente físico transparente ao Agente, tratando as características específicas de interação.

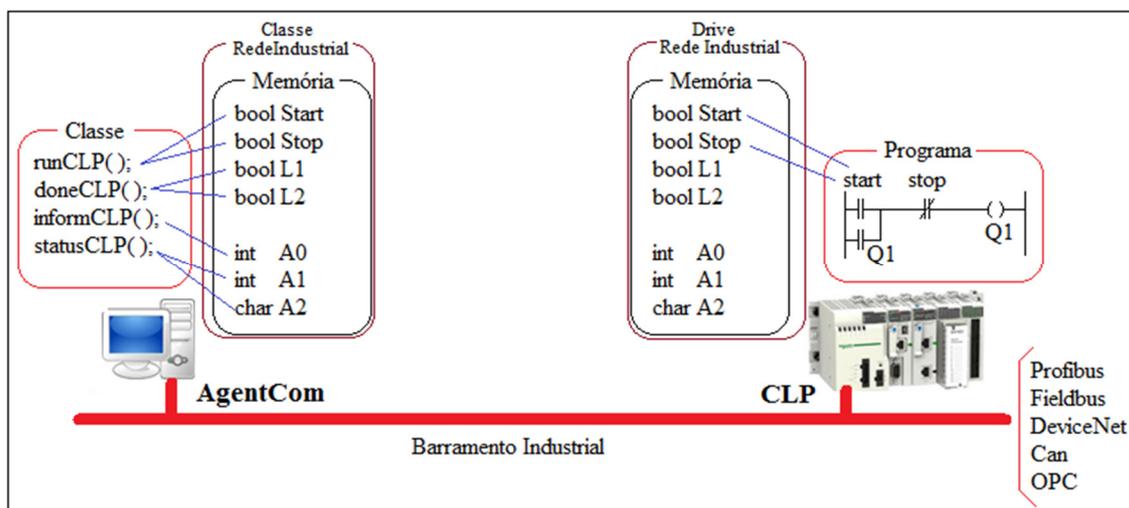
Este agente se diferencia do AMI (*Agent Manufacturing Interface*) que fora abordado em outros trabalhos, pelo fato de não ser dedicado, específico e programado para uma dada manufatura. Diferente do AMI, o AgentCom irá buscar e inferir no estado de variáveis no CLP, sem interferir na programação original ou acoplar algum hardware específico.

O AgentCom utilizará os seguintes métodos para se comunicar com o CLP:

- a) runCLP( ): COMANDO, provoca a partida da máquina para executar o serviço desejado;
- b) doneCLP( ): RESPOSTA, retorna a informação quanto ao estado de execução do serviço;
- c) informCLP(par1, par2, par3, par4): INFORMAÇÃO, envia ao CLP informações relativas ao serviço que quer ser executado;
- d) statusCLP( ): ESTADO, retorna o status da máquina, informando dados relativos à execução do serviço solicitado.

Estes métodos estarão disponíveis para o agente de manufatura, sendo que o meio com que o *AgenteCom* irá se comunicar com o CLP será definido em uma classe desenvolvida conforme a técnica que será utilizada, e que diz respeito a estrutura do CLP que estará atuando, acessando suas variáveis de memória sem alterar o programa da lógica local do CLP. Esta técnica pode ir além da questão de promover uma comunicação entre CLP, mas poderá se comunicar com qualquer outro equipamento de controle que disponibilize variáveis de memória para acesso externo.

Basicamente o que se propõe nesta etapa é a utilização de rede industrial como meio físico, onde um dos princípios básicos é o compartilhamento de variáveis entre dispositivos (TANENBAUM; WETHERALL, 2011). Este compartilhamento ocorre com um mínimo de alteração na lógica local do controlador lógico programável, pois o *AgenteCom* irá acessar as variáveis de memória do CLP que estão vinculadas diretamente à lógica local. A Figura 53 mostra o princípio desta técnica.



**Figura 53 Princípio do compartilhamento de variáveis através de rede industrial.**

Fonte: Autor.

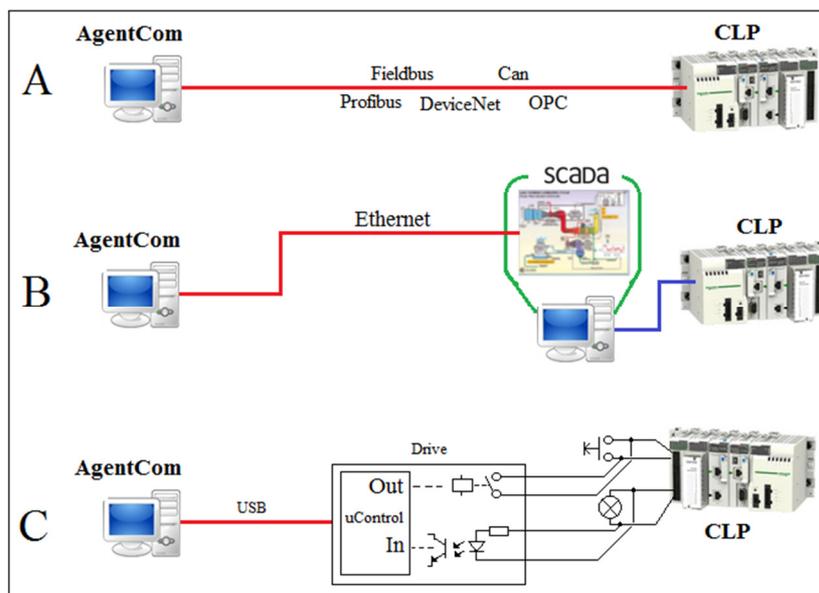
Nesta figura se percebe então a ação do agente *AgenteCom*, que se comunica com o CLP através do meio físico de uma rede industrial. As memórias do CLP são então compartilhadas pela rede com o agente de comunicação, que dispõe destes dados para os métodos que serão invocados para gestão do processo.

Há outra técnica que permite a interação com as variáveis de um controlador lógico programável, sem a necessidade de alterar o programa que se está executando. É através do conceito SCADA (*Supervisory Control And Data Acquisition*). Trata-se de um sistema supervisorio que se propõe a comunicar-se com o CLP ou outro dispositivo a fim de monitorar variáveis do processo e até mesmo interagir, divulgando as informações ao operador do sistema de forma gráfica, com representações alusivas ao processo (ALBUQUERQUE et al., 2007). Cabe então ao agente acessar o sistema SCADA através de rede de dados e compartilhar as variáveis de interesse. Isto pode ocorrer através de *web services* ou através de padrões como OPC.

Em muitas aplicações esta técnica pode ser bastante viável, uma vez que os sistemas SCADA já estejam instalados e operando no sistema de manufatura, ou seja, já há um sistema SCADA em funcionamento, então cabe ao agente acessá-lo.

Na hipótese de que o controlador lógico programável não possua conexão alguma com rede externa, então haverá necessidade de implementar um dispositivo (*drive*), que se conecte diretamente aos dispositivos de entrada e saída do CLP, coletando sinais pertinentes para as ações de comando que o agente necessita executar, e ainda assim não se está alterando as características do programa que está sendo executado no CLP. Esta técnica requer a implementação de um dispositivo de uma interface de comunicação, que basicamente consiste de relés para através de seus contatos enviar dados e dispositivos optoacopladores destinados a receber dados.

A Figura 54 apresenta as três alternativas para o interfaceamento do agente com o CLP, sem a necessidade de intervenção no software ou implementação de hardware específico incorporado ao CLP.



**Figura 54 Alternativas para interfaceamento do agente com o CLP, sem alteração do programa.**

Fonte: Autor.

Em “A” se observa uma conexão do AgentCom com o CLP através de barramento industrial, em “B” a conexão ocorre através de um sistema SCADA e em “C” a conexão ocorre através do acesso direto aos periféricos de entrada e saída. Estas três técnicas permitem o acesso do AgentCom às variáveis do CLP sem que haja necessidade de alteração significativa da lógica do programa local, que nele é executado.

Agora um olhar na manufatura. Um sistema de manufatura se propõe a transformar uma matéria-prima em produto manufaturado. A **manufatura** (do latim, *manu*, mão e *factura*, feito) descreve a transformação de matérias primas em produtos terminados para sua comercialização. Nesta transformação as máquinas e equipamentos realizam o processo de recebimento de matéria-prima em sua entrada, realizam algum tipo de processo sobre esta matéria, valendo-se de insumos se necessário, e entrega em sua saída um produto ou parte de um produto acabado.

Neste processo as máquinas recebem comandos e informações do operador para realização de sua atividade, e fornecem a ele informações quanto o estado do processo e

a informação de término, a fim que o produto possa passar para uma próxima etapa. Ocorre então uma troca de mensagens e comandos entre operador e máquina durante o processo produtivo.

Analisando os aspectos do processo produtivo, se podem determinar quais são estes comandos e quais são estas informações necessárias. Mas há uma ressalva quanto a estes processos. Eles dizem respeito a processos discretos, típicos de produção em batelada. Os processos contínuos possuem características que requerem uma análise mais aprofundada quanto a sua atuação em um sistema auto-organizável. Neste trabalho o universo definido se restringe aos processos discretos.

A Figura 55 mostra uma rotina de manufatura em uma máquina de um processo produtivo.



**Figura 55 Rotina de manufatura em uma máquina em um processo produtivo.**

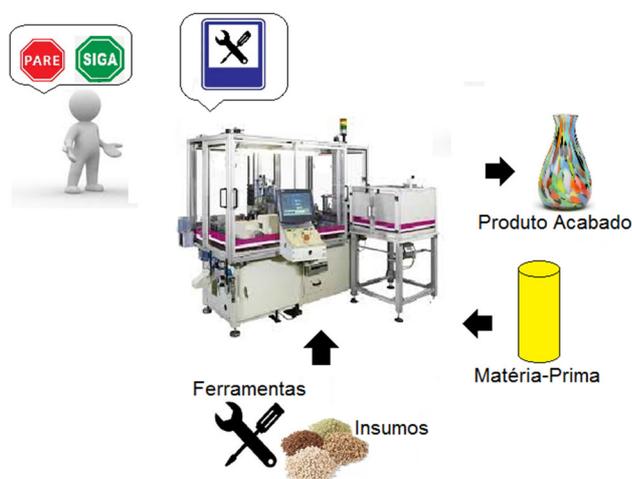
Fonte: Autor.

Na figura se percebe a rotina de manufatura em uma máquina em um processo produtivo, a fim de que máquina receba a matéria-prima e transforma em produto acabado. O operador acessa a máquina lhe fornecendo os insumos e as ferramentas necessárias ao processo em questão.

Cabe aqui salientar que se está verificando um processo em que a máquina é comandada por um operador (ser humano), mas que poderia ser comandada por outra máquina ou processo, de forma automatizada. Este fato não interfere na metodologia, pois

de igual forma há sinais de interação entre as máquinas. Por uma questão didática é abordado somente o caso em que há um operador que comanda a máquina.

Mas numa visão de automação do processo este cenário requer uma mudança. A máquina necessita ter certa autonomia em seus insumos, ferramentas e programas, a fim de que o operador interaja com a máquina lhe enviando comandos e recebendo informações da mesma. A Figura 56 apresenta este cenário desejado para um processo automatizado.



**Figura 56 Interação do operador e máquina em um processo automatizado.**

Fonte: Autor.

Neste cenário, uma máquina com um processo automatizado já gerencia suas ferramentas, gerencia seus insumos necessários, absorve a matéria-prima em posições pré-definidas e entrega o produto acabado. O operador interage com a máquina lhe enviando COMANDO (início de processo, parada do processo, entre outros) e recebe da máquina a RESPOSTA do processo de manufatura (fim do processo, quantidade produzida, qualidade produzida, entre outros).

Cabe salientar que a matéria-prima não necessita ser necessariamente o suprimento de material para primeira etapa do processo, mas sim o próprio produto já com etapas precedentes manufaturadas, necessitando outras etapas de manufatura para sua conclusão. O produto acabado também não necessita ser o produto final, mas o

produto com esta tarefa agregada e que já pode ser encaminhado para o próximo passo do seu processo fabril.

Para este grau de atingimento de automação, o controle da máquina recairá sobre um Controlador Lógico, podendo ser um CLP, um CNC, um microcontrolador, um computador industrial ou mesmo um computador. Fato é que se faz necessário recurso computacional para processar esta interação da máquina com o operador. Este é o grau de automação desejado para que um sistema de manufatura se candidate a interagir dentro de um sistema mais evoluído, onde as máquinas e peças interagem umas com as outras.

Agora é necessário analisar que mensagens e comandos são estes que as máquinas e operadores trocam dentro de um processo fabril. A Tabela 8 apresenta uma relação de processos industriais e seus comandos e informações envolvidas, que auxiliarão nesta análise. Estes processos foram definidos por serem os mais usuais em termos de manufatura industrial e listado a partir de entrevistas a profissionais que atuam neste meio.

**Tabela 8 Análise das características do processo de manufatura.**

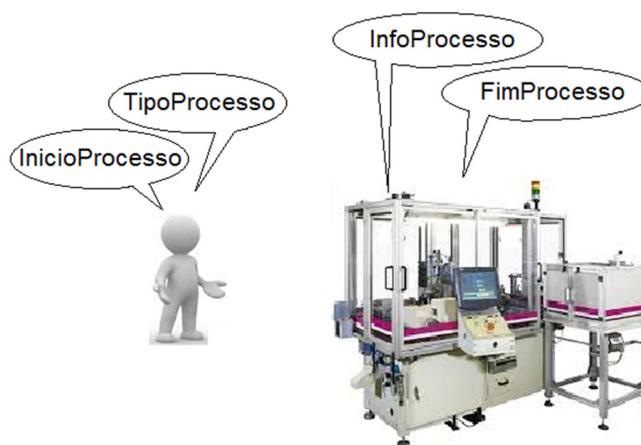
Processo de Manufatura	Característica do processo	Comando de partida	Informações para a produção	Informações do estado da produção	Informação de término	Insumo/ferramenta necessária	Produto da saída
Usinagem	Material é retirado do produto, dando-lhe uma geometria desejada.	Partida da usinagem.	Folha de processo ou programa CNC.	Tempo para execução da manufatura.	Peça usinada.	Ferramentas, folha de processo.	Peça usinada.
Montagem	Peças são agregadas ao produto, dando-lhe o conceito de conjunto.	Iniciar a montagem.	Sequência de montagem a ser executada.	Estado atual da montagem.	Montagem executada.	Peças a serem montadas.	Conjunto montado.
Pintura	O produto é submetido a uma alteração de cor ou textura.	Iniciar a pintura.	Trajectoria da pintura a ser executada.	Qualidade da pintura realizada.	Fim de pintura.	Tinta, pistola de aplicação da tinta.	Peça pintada.
Transporte	O produto é deslocado de uma posição a outra dentro de uma planta industrial.	Iniciar o transporte.	Posição de origem e posição de destino.	Localização atual do transporte.	Fim do transporte.	Dispositivo de acondicionamento do transporte.	Peça entregue.
União	Através de soldagem, fricção ou outro processo, partes de peças são unidas, formando uma peça única.	Início da operação de União.	Posição onde se encontram as peças a serem unidas.	Tempo para execução da união.	Fim da união das peças.	Insumos para a união dos materiais.	Peça soldada.
Dobradura	O produto é modificado em seu formato através de forças aplicadas em partes específicas.	Início da operação de dobradura.	Folha de processo com indicação da geometria desejada.	Tempo para execução da dobragem.	Fim da dobradura.	Ferramentas de dobradura.	Peça dobrada.
Corte	Partes do produto são seccionadas.	Início da operação de corte.	Plano de corte.	Estado da operação de corte.	Fim do corte.	Ferramentas de corte.	Peça cortada.
Inspeção	As características de um produto são medidas e estes dados fornecidos.	Início da operação de inspeção.	Variáveis a serem inspecionadas.	Valores medidos nas variáveis.	Fim do processo de inspeção.	Instrumentos de medição conforme varável a ser mensurada.	Inspeção realizada.

Fonte: Autor.

A partir desta tabela uma análise geral é realizada e podem-se concluir fatos importantes. Entre eles, que a comunicação entre o operador e a máquina pode se resumir a 4 tipos de mensagens:

- a) COMANDO de início de operação (Operador → Máquina);
- b) INFORMAÇÃO de como o processo deve ser realizado (Operador → Máquina);
- c) RESPOSTA da situação da máquina (máquina → Operador);
- d) SITUAÇÃO da operação realizada (máquina → Operador).

A Figura 57 apresenta a interação do operador com a máquina, na sua troca de mensagens e comandos.



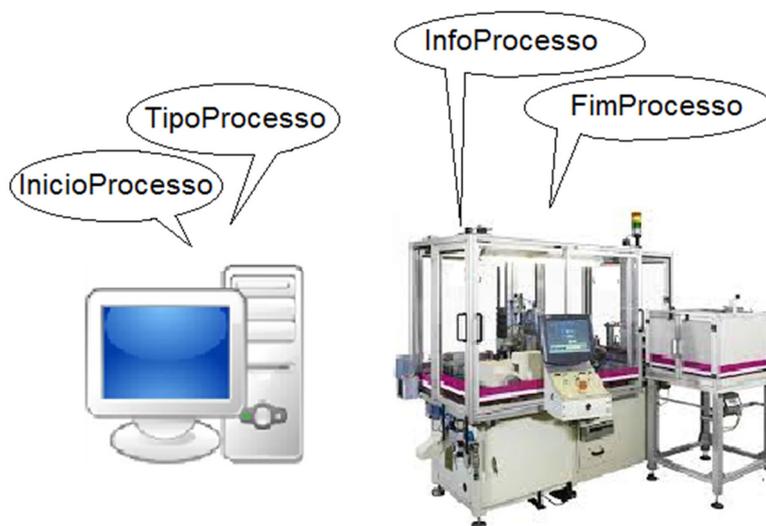
**Figura 57 Interação entre o operador e a máquina.**

Fonte: Autor.

O operador envia o COMANDO para iniciar o processo (InicioProcesso) e também as INFORMAÇÕES quanto ao processo que quer realizar (TipoProcesso). A máquina envia ao operador a mensagem de RESPOSTA do fim de processo (FimProcesso) e as informações quanto a SITUAÇÃO do processo que foi realizado (InfoProcesso).

Agora, analisando sob o aspecto da máquina poder participar de um sistema auto-organizável, obtendo a capacidade de interagir com outras máquinas e peças, cabe então propor a troca das funções do operador por um computador, que, através de software, efetuará os

comandos e receberá as mensagens da máquina. O computador por sua vez representará a máquina na interação com as outras. A Figura 58 mostra o processo de interação da máquina com o computador.

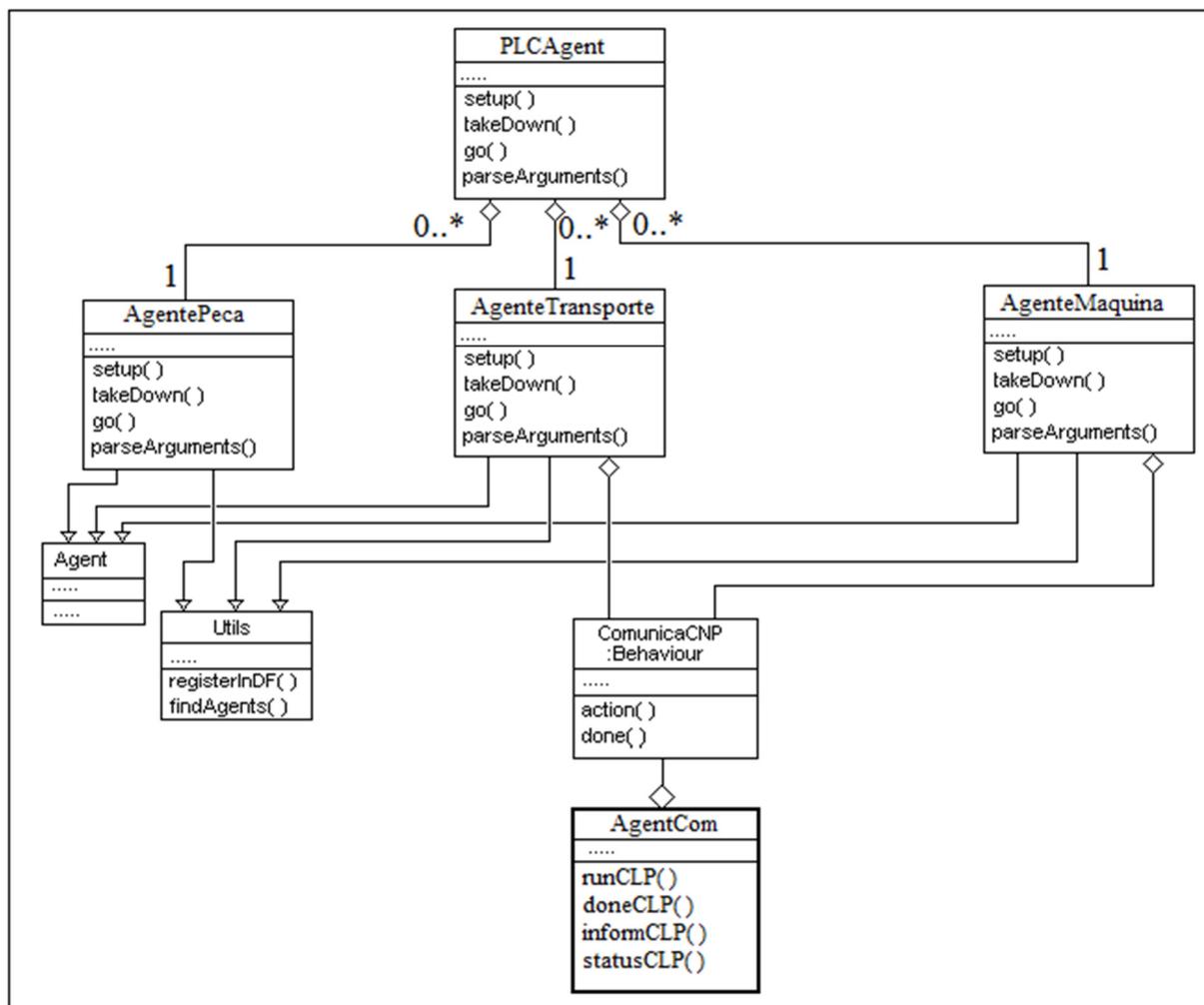


**Figura 58 Interação entre o computador e a máquina.**

Fonte: Autor.

Partindo então para o olhar sobre os agentes que irão compor os modelos de máquinas, eles são implementados através de um componente agente desenvolvido para assumir o comportamento de cada classe.

A Figura 59 a seguir mostra o diagrama de classe que representa a relação entre os agentes que irão compor a manufatura.

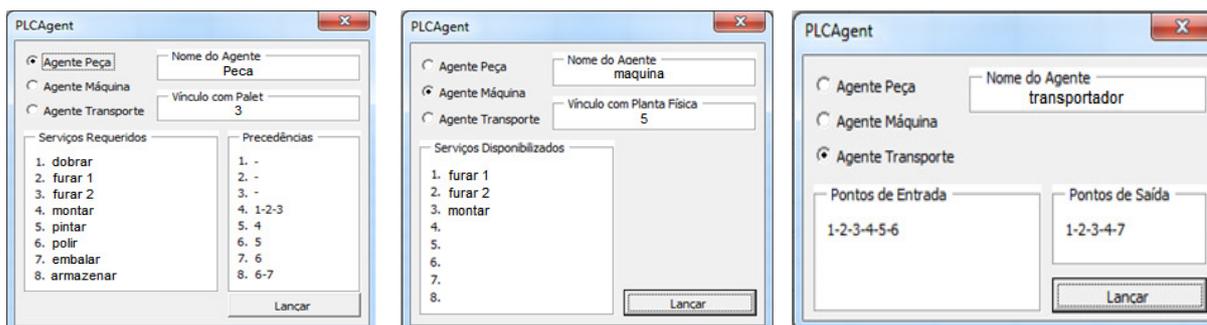


**Figura 59** Diagrama de classe com a relação entre os agentes para formar o PLCAgent.

Fonte: Autor.

Nesta figura se observa as relações de herança e agregação dentre as classes, de forma a compor o PLCAgent, que assumirá o comportamento de cada elemento da manufatura, segundo suas características e parâmetros definidos quando da sua geração de instância.

O PLCAgent, disponibiliza ao operador a inserção de parâmetros, que irão definir o comportamento do agente segundo sua classe. A Figura 60 apresenta o modelo da interface do PLCAgent, com seus parâmetros de inserção para a execução do agente, onde o “Agente Peça” se refere ao agente de produto, o “Agente Máquina” se refere ao agente de recurso e o “Agente Transporte” se refere ao agente unidade de transporte, descritos em seção anterior.



**Figura 60 Modelo da interface do PLCAgent, configurado para Agente Peça, Agente Máquina e Agente Transporte.**

Fonte: Autor.

Quando o PLCAgent é definido como Agente Peça, então os campos a serem inseridos serão o nome do agente, vínculo inicial com *palet*, o que propicia uma relação com algum transportador físico, e a lista de serviços executados, bem como sua indicação de precedência. A precedência estabelece quais serviços podem ser executados sem depender de uma tarefa anterior. Quando o PLCAgent é definido como Agente Máquina então é necessário indicar o nome do agente, o vínculo com a posição física na planta e a lista de serviços a serem disponibilizados. Quando o PLCAgent é definido como Agente de Transporte então deverão ser inseridos o nome do agente, os pontos que tem capacidade de buscar peças e que tem possibilidade de entregar.

Este trabalho dispõe então estes agentes como um recurso para a implementação de sistemas de manufatura gerenciados por controladores lógicos programáveis.

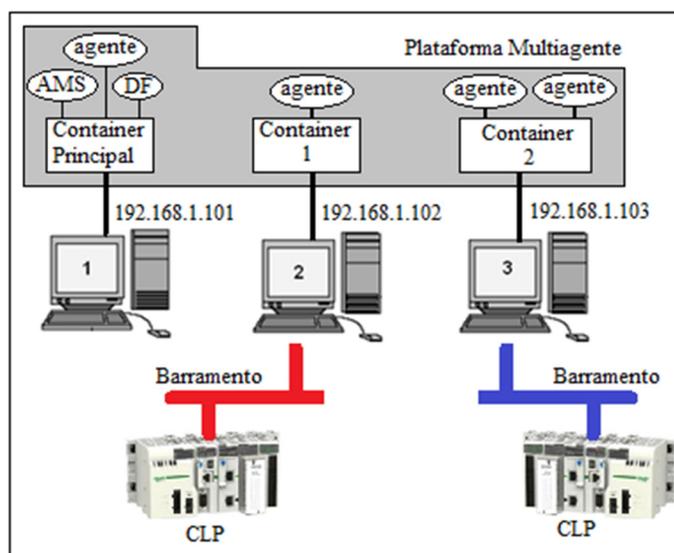
Para que estes agentes possam interagir uns com os outros, uma plataforma multiagente se faz necessária. Resgatando o que foi descrito no Capítulo 2, JADE foi à escolhida para ser aplicada no desenvolvimento deste trabalho por ser a plataforma que:

- a) Cumpre os requisitos técnicos de facilidade na instalação e facilidade na operação;
- b) Oferece uma boa estrutura para agentes móveis;

- c) Está disponível gratuitamente e o código da plataforma pode ser alterado sem restrições;
- d) Melhor atende aos requisitos de disponibilidade de literatura e suporte técnico;
- e) Disponibilidade de recursos de monitoramento da plataforma.

A plataforma JADE ao ser executada instala uma infraestrutura de software que permite a identificação de outros agentes que entrarem no segmento de rede que a plataforma estiver instalada. Ela é então instalada em um segmento de rede ethernet, de forma que outros agentes a serem executados neste segmento serão identificados pela plataforma e lhes serão dispostos os serviços de infraestrutura de software para que eles interajam com os demais agentes, além de dispor os recursos dos agentes DF e AMS.

A Figura 61 apresenta então a plataforma multiagente servindo como infraestrutura para suportar a execução dos agentes, que por sua vez estarão conectados aos Controladores lógicos programáveis através de barramento industrial.



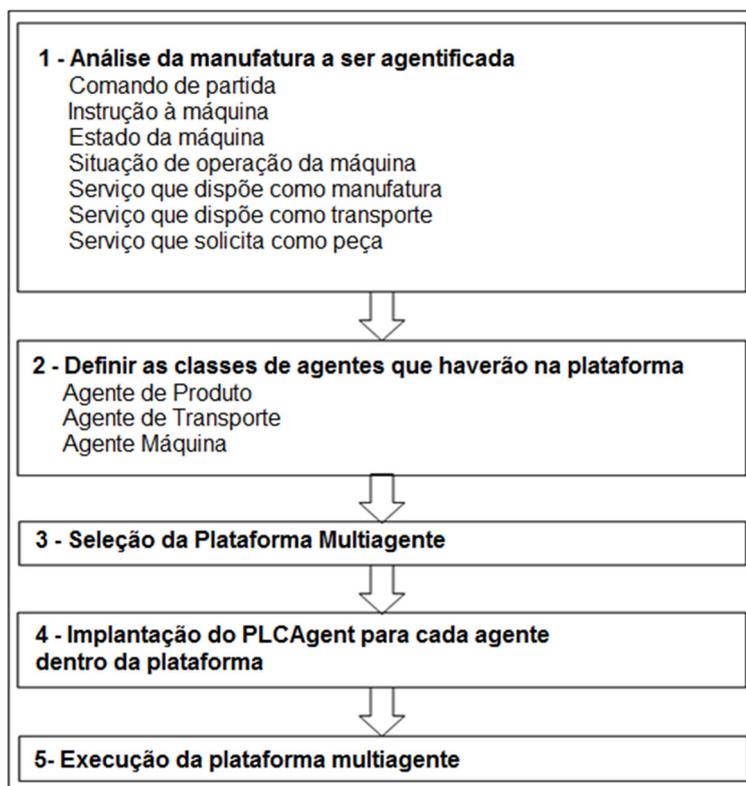
**Figura 61** Plataforma multiagente em JADE para dispor a infraestrutura necessária.

Fonte: Autor.

A execução da plataforma JADE foi descrita no Capítulo 2 deste trabalho.

### 5.3 DETALHAMENTO DA METODOLOGIA

A descrição do caminho para levar o CLP a ter as características dos Sistemas Multiagentes caracteriza o método proposto por este trabalho. A Figura 62 representa em um diagrama com a sequência de passos do método aqui proposto.



**Figura 62 Sequência de passos do método proposto.**

Fonte: Autor.

Nesta figura observa-se o caminho a ser seguido para que o controlador lógico programável tenha as características dos Sistemas Multiagentes, visando a atender a diversidade, agilidade e auto-organização da manufatura, caracterizando assim a metodologia.

Detalhando cada passo do método se tem os seguintes tópicos:

### Passo 1 – Análise da manufatura a ser agentificada.

Neste passo o método prevê a análise da manufatura a que se deseja incorporar as características multiagentes. Esta análise é importante para definir quais meios e técnicas serão utilizadas nas demais etapas para implementação total do método.

Uma análise abrangente sobre todos os elementos da manufatura é necessária, e primeiramente separando a manufatura em grupos controlados por um CLP, pois ele será o candidato a se tornar um agente. Nesta análise observa-se então o CLP para determinar qual o meio de comunicação do qual ele dispõe com componentes externos. Com o olhar no processo analisam-se os serviços que a manufatura dispõe, agrupados segundo o CLP a que pertencem. Uma manufatura controlada por um CLP pode dispor mais de um serviço e a indicação de qual serviço estará sendo solicitado geralmente é procedido através de uma *flag*. Para cada serviço é necessário então analisar quais sinais no programa do CLP que fazem o serviço ser executado, quais indicam que o serviço acabou e quais indicam o estado do serviço.

A proposta é isolar a manufatura de forma a observá-la como uma entidade autônoma, identificando as 4 mensagens/comandos que devem ocorrer entre a máquina e o elemento de controle, respondendo as seguintes perguntas:

- a) Qual sinal de comando para dar início a operação?
- b) Qual instrução de execução deverá ser passada para a máquina?
- c) Qual sinal de aviso que a máquina emite para indicar o estado de sua operação?
- d) Qual instrução de estado da manufatura que a máquina indica?

A resposta a estas 4 perguntas indicam se a máquina é passível de agentificação, ou seja, de se tornar um agente.

Na sequência se define quais serviços a máquina a ser agentificada disponibiliza, sejam serviços de manufatura propriamente dita ou serviços de transporte. Alguns exemplos de serviço:

- a) Furação de peças;
- b) Pintura de peças;
- c) Transporte da posição 1 para 2;
- d) Montagem peças vermelhas e pretas.

Por fim, se aplica agora um olhar na peça ou nas peças a serem fabricadas, observando quais operações necessitam ocorrer para sua fabricação, assim como a precedências que estas operações deverão ocorrer. Esta análise deverá ocorrer a cada solicitação de alguma peça que ainda não foi produzida neste sistema de manufatura, pois como a proposta indica, o sistema deve ser versátil a ponto de atender a diversidade. A sequência e precedência serão importantes para que o agente peça possa definir o fluxo de processo durante as negociações.

Em uma síntese, pode-se entender a etapa 1 como sendo a análise da manufatura visando identificar os seguintes dados:

- a) Qual COMANDO para partida;
- b) Qual INFORMAÇÃO ao equipamento;
- c) Qual aviso de RESPOSTA do equipamento;
- d) Qual a SITUAÇÃO da operação realizada;
- e) Se for uma máquina, quais serviços ela dispõe;
- f) Se for um transportador, quais serviços de transporte realiza;
- g) Se for um peça, quais serviços necessita e suas precedências.

#### Passo 2 – Definir as classes de agentes que existirão na plataforma.

Neste passo serão definidas, a partir da análise da manufatura, quais classes de agentes haverão no Sistema Multiagente.

As classes irão determinar os comportamentos dos agentes dentro da plataforma e sua interação na manufatura. Para cada tipo de agente foi criado um agente genérico que assume o comportamento da classe desejada, valendo-se dos parâmetros inseridos e já realiza os comportamentos necessários para a manufatura, como descrito a seguir:

- a) *Agente Peça (PA-Product Agent)*: um agente que quando configurado no PLCAgent receberá a lista de serviços a serem executados e suas precedências. A partir de um algoritmo ele consulta na sua lista de serviços a próxima atividade a ser executada e inicia um protocolo de alocação de serviços com os “agentes máquina”, a fim de contratar o agente que satisfaça um critério definido, como o agente que mais atividades puder realizar no seu ambiente de manufatura. Após esta alocação fechada, o agente peça irá alocar um “agente transporte” que execute o deslocamento até a posição do “agente máquina” alocado. Esta operação se repete até que a lista de serviços seja atendida, quando então o último serviço é executado, que trata-se da saída da peça do sistema;
- b) *Agente Máquina (MA-Manufacturing Agent)*: este agente quando configurado no PLCAgent irá estender uma classe que fará a comunicação com o CLP. Nesta classe será definida a técnica de comunicação, a forma de troca de informações e o meio com o qual irá ocorrer, podendo ser:
  - a. *Rede Industrial*: neste caso deverá ser implementada uma classe que execute o mesmo protocolo de comunicação que o CLP possui. Como exemplo, se o CLP possui o protocolo Modbus instalado, então a classe do AgentCom terá que se comunicar também neste protocolo. Uma vez definido o protocolo na classe então parte-se para o reconhecimento das variáveis de interesse no mapa de memória compartilhada pelo

protocolo. O AgentCom, um agente criado para controlar a comunicação, irá estender esta classe gerada para executar o protocolo definido, e atuará sobre as variáveis de interesse, executando seus métodos para a interação do PLCAgent com o CLP.

- b. SCADA: neste caso o AgentCom irá se comunicar com o sistema SCADA através de rede de dados, compartilhando variáveis de interesse para execução do agente. Como o sistema SCADA já está comunicando com o CLP, e já possui as variáveis de interesse bem definidas. O processo então é comunicar com o AgentCom através de web services ou outro meio e trocar os dados.
- c. Conexão por I/O: trata-se de um CLP que não possui forma alguma de comunicação com os elementos externos via rede. Neste caso o AgenCom irá comunicar-se com um dispositivo de drive, implementado para este fim, que será conectado diretamente aos periféricos de entrada e saída do controlador, não interferindo no programa, mas interagindo como se fosse um periférico externo, mas trata-se do agente comandando o CLP. Exemplo de atuação de comando é a interface de comunicação possuir um sinal paralelo ao botão de partida da máquina, que uma vez acionado a lógica local irá interpretar que o botão de partida da máquina foi acionado.

Definida a comunicação, então o PLCAgent irá postar os serviços definidos pelo passo 1, a fim de que outros agentes possam saber os serviços a que está disposto. E aguarda então o chamado de algum “agente peça” para proceder as negociações.

- c) Agente de Transporte (*TUA-Transport Unit Agent*): este agente também irá estender o AgentCom e terá o mesmo tratamento para definição do modo de comunicação como CLP. Já o PLCAgent, ao ser configurado para “agente de transporte” irá postar seus serviços, que dizem respeito a quais posições físicas na planta ele pode recolher peças e quais ele pode entregar peças. Feito isto, ele se submete a alocação de seus serviços, aguardando algum “agente peça” lhe solicitar proposta para então proceder as tratativas para alocação do recurso. Acertado o acordo, o “agente de transporte” executa o traslado.

### Passo 3 – Seleção da plataforma multiagente.

Neste caso cabe a escolha da plataforma multiagente a ser implantada. Esta escolha irá definir a forma com que os agentes serão codificados, em função da linguagem de programação que utiliza. A plataforma trata-se de um recurso de software que permite que outras entidades de software possam interagir umas com as outras, estando no mesmo domínio.

No Capítulo 2 foram sugeridas algumas plataformas multiagentes que podem ser utilizadas, sendo lá descritas suas características.

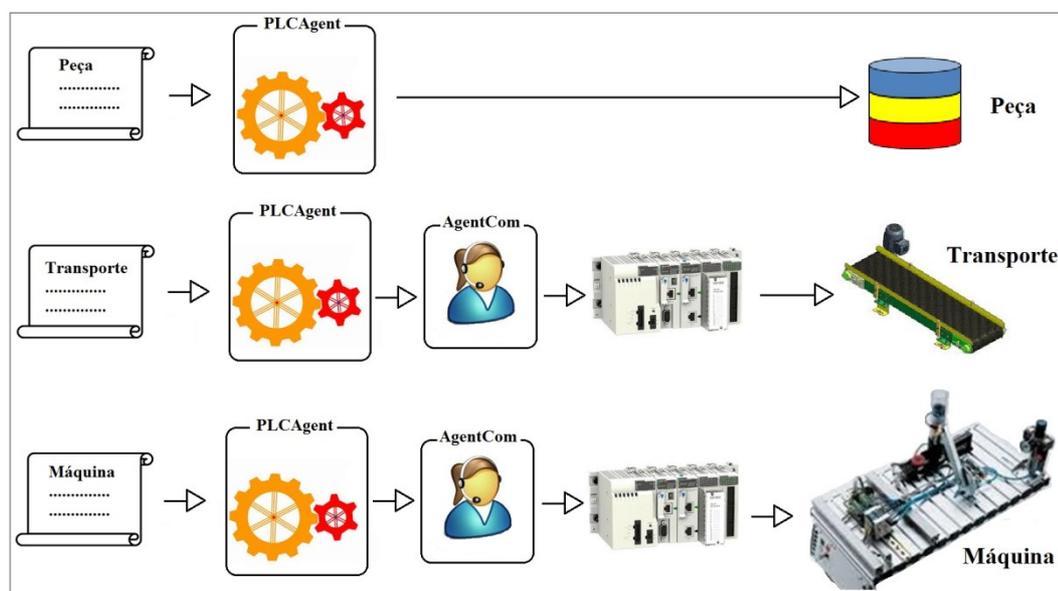
### Passo 4 – Implantação do PLCAgente para cada agente, dentro da plataforma.

Definido no passo 2 os agentes que irão compor a plataforma, neste passo cabe a execução destes agentes como instâncias. Aqui é proposto o agente PLCAgent, desenvolvido para assumir o comportamento da categoria de agente a que se destina (Agente Peça, Agente Máquina ou Agente Transporte).

Durante a análise da manufatura os requisitos foram então definidos e estes servirão como parâmetros para execução do agente. Os agentes já terão dentro do seu código o protocolo de comunicação para estabelecer contratos (ContractNet) e alocação de tarefas (Request):

- a) Implantação do PLCAgent Transporte: será configurado para receber solicitações de serviços, recebendo as informações de posição de retirada e a entrega da peça. Há um método que irá postar os serviços na plataforma multiagente e se submeterá a participar das negociações com o agente peça. Terá uma classe que se comunicará com o CLP através do AgentCom, provocando o comando para o transportador físico;
- b) Implantação dos PLCAgent Máquina: será configurado para receber solicitações de serviços, os quais executará e indicará suas execuções. Há um método que irá postar os serviços na plataforma multiagente e se submeterá a participar das negociações com o agente peça. Terá uma classe que irá interagir com o CLP através da classe AgentCom, provocando o comando para a máquina física.

A Figura 63 mostra a estrutura de implantação dos agentes da plataforma segundo suas características definidas na análise da estrutura da manufatura.



**Figura 63** Estrutura de implantação dos agentes da plataforma a partir do PLCAgent.

Fonte: Autor.

Nesta figura se observa a geração dos agentes a partir das especificações definidas, valendo-se do PLCAgent.

### Passo 5 - Execução da plataforma multiagente.

Como passo final cabe a execução do sistema completo, com os agentes peça sendo lançados para solicitar a produção e os agentes máquina e transporte se ofertando para proceder a manufatura desejada:

- a) Implantação dos PLCAgent Peça: será configurado como agente peça através da informação lista de serviços e suas relações de precedência. O agente começará uma tratativa para alocação de recurso com os agentes de máquina, a partir da lista de serviços cuja precedência esteja resolvida. Alocará o agente máquina necessário e negociará com o agente de transporte para proceder o deslocamento da peça até a posição da máquina contratada.

## **5.4 SIMULAÇÕES E RESULTADOS**

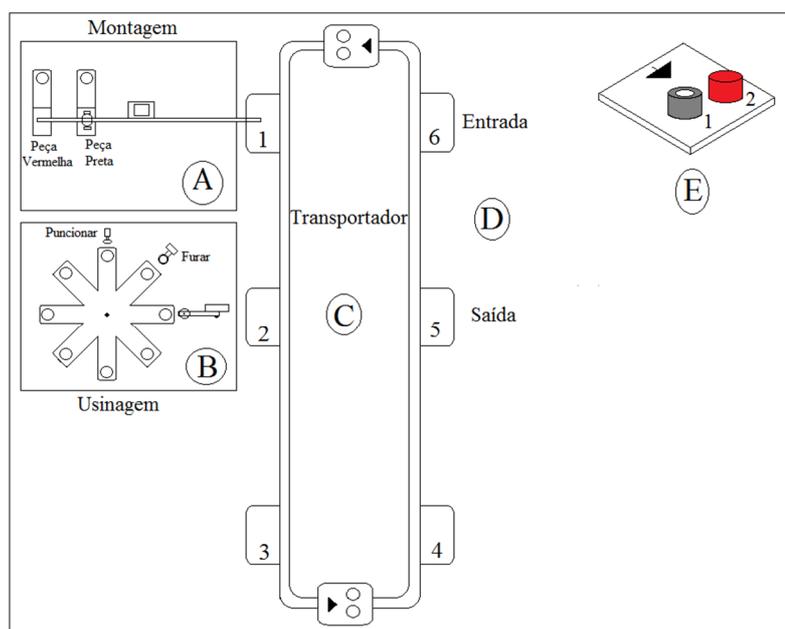
Com o objetivo de testar o método proposto, faz-se necessário promover ensaios em cenários que traduzam a realidade fabril e a realidade tecnológica dos sistemas de manufatura. Como a proposta deste trabalho é apresentar soluções em termos de pesquisa para os problemas de diversidade, agilidade e auto-organização, os ensaios realizados irão aplicar o método sobre o sistema de manufatura, para então comparar com outros sistemas de gestão da manufatura.

### **5.4.1 Cenário proposto para aplicação da metodologia.**

Para proceder o ensaio de aplicação da metodologia é proposto um cenário de manufatura integrada, composto de uma estação de usinagem, uma estação de montagem e uma estação de transporte. O produto a ser produzido é um palete com duas peças, uma na posição 1 e outra na posição 2, sendo que cada posição pode ter peças das cores vermelha ou preta. No caso da posição 1 esta peça pode passar pelo processo de usinagem, podendo ser furada,

puncionada ou ambas operações. A estação de montagem tem capacidade de montar peças pretas ou vermelhas tanto na posição 1 como na posição 2 do palete. Já a estação de usinagem tem capacidade de proceder às operações de puncionamento e de furação sobre a peça que estiver na posição 1 do palete. Além destas duas estações, há a estação de transporte, que tem a função de levar os paletes até as estações e também de gerenciar a manufatura.

A Figura 64 apresenta o leiaute deste cenário de manufatura.



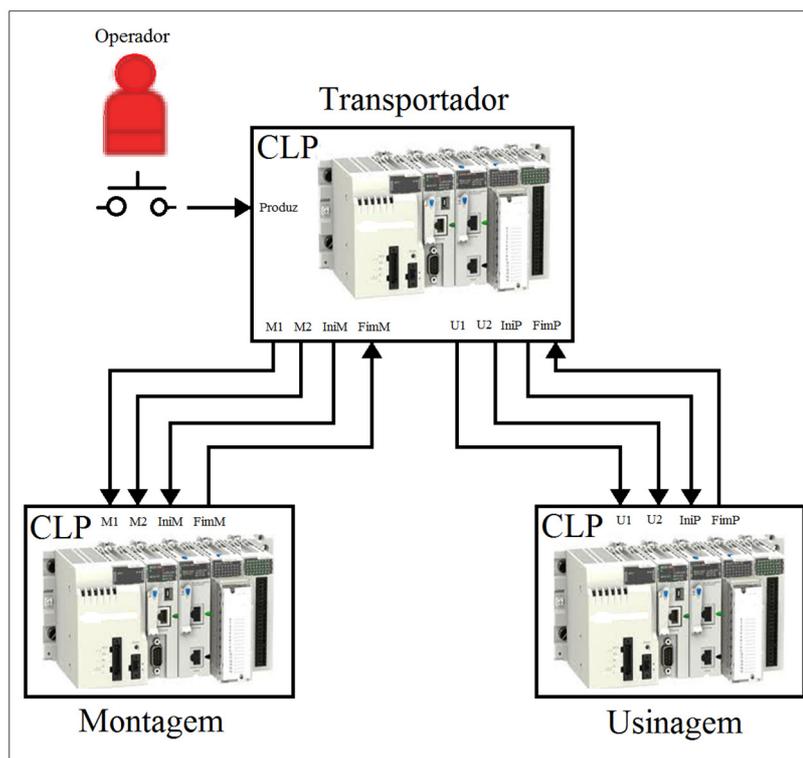
**Figura 64 Cenário para ensaio da manufatura controlada por CLP.**

Fonte: Autor.

Nesta figura observa-se então em “A” a estação de montagem, que por meio de um manipulador pneumático, irá pegar peças nas rampas abastecedoras e montar sobre o paleta na posição definida. Em “B” está a estação de usinagem, onde serão realizadas as operações de puncionamento e furação. Em “C” se observa o sistema transportador, que fará o deslocamento do paleta dentre suas seis posições de alocação de estação. A estação transportadora também é a responsável pelo gerenciamento da produção do sistema, ela é que regulará a sequência de produção de cada paleta. Em “D” se observa na estação de transporte a posição de entrada de paleta para ser manufaturado, assim como a saída deles depois de realizada a manufatura. Em

“E” tem-se então o produto manufaturado, onde se pode observar a montagem de uma peça preta usinada na posição 1 e uma peça vermelha na posição 2.

Cada uma das três estações é composta de um controlador lógico programável, que irão se comunicar entre si através das entradas e saídas digitais. A Figura 65 apresenta um diagrama das conexões lógicas entre os controladores lógicos programáveis.



**Figura 65 Conexões lógicas entre os controladores lógicos programáveis.**

Fonte: Autor.

Cada vez que o operador envia um sinal na entrada “produz” do CLP do transportador, este iniciará a sequência de produção de um palete com uma peça vermelha montada na posição 2 e uma peça preta usinada na posição 1.

Quando o CLP do transportador aciona o sinal de “IniM”, ele dispara o início da estação de montagem, que atenderá aos parâmetros estabelecidos pelos sinais M1(montar preta) e M2(montar vermelha). Ao encerrar a montagem esta estação envia um sinal “FimM” para indicar que a montagem se encerrou.

A mesma lógica se aplica para a estação de usinagem, sendo que os sinais de U1 (puncionamento) e U2 (furação) é que determinam o tipo de usinagem que será realizada.

O controle do transporte dos paletes entre as estações se dá pelo CLP da estação transportadora.

Este será então o cenário de um sistema industrial que será utilizado para desenvolver um processo de manufatura real, a fim de aplicar-se a metodologia aqui proposta. No anexo I estão descritos os programas utilizados nos controladores lógicos programáveis em uma programação para uma sequência produtiva.

#### **5.4.2 Ensaio do método no cenário proposto**

Uma vez definido o cenário, então o método é aplicado, com o objetivo de empregar as funcionalidades dos Sistemas Multiagentes no controlador lógico programável. Sabido então que neste conceito não haverá mais um sistema que gerencie o fluxo de manufatura, mas sim o produto é que passa a ter capacidade de processamento para interagir com os demais componentes do sistema de manufatura e definir seu fluxo de produção. O Anexo 1 apresenta o diagrama desta planta de ensaio com seus devidos programas originais dos controladores, concebidos para serem executados de forma tradicional, com a ordem de produção pré-definida.

Aplicando cada passo do método se tem então:

##### Passo 1 – Análise da manufatura a ser agentificada.

Analisando o sistema de manufatura proposto no cenário se observa então duas estações de manufatura, uma estação de transporte e a peça a ser produzida. Separando cada estação se obtém as características de cada processo:

- a) Máquina de montagem: Equipamento com a função de montar peças de cores distintas sobre um palet de acondicionamento. A Tabela 9 mostra as características observadas.

**Tabela 9 Características observadas para máquina de montagem.**

Aspectos de comunicação	Característica observada
Controlador Lógico	CLP Siemens S7-300.
Meio de Comunicação	Rede industrial Profinet, com meio físico <i>ethernet</i> .
Qual COMANDO para partida	Acionamento da <i>flag</i> “IniM” que está vinculado ao botão de partida da máquina.
Qual INFORMAÇÃO ao equipamento	Através da <i>flag</i> “M1” se indica a realização da operação montar peça preta ou vermelha na posição 1 do palete (M1=0 → preta; M1=1 → vermelha). Através da <i>flag</i> “M2” se indica a realização da operação montar peça preta ou vermelha na posição 1 do palete (M2=0 → preta; M2=1 → vermelha).
Qual aviso de RESPOSTA do equipamento	Monitoramento da <i>flag</i> “FimM” que indica se a operação chegou ao fim.
Qual a SITUAÇÃO da operação realizada	Esta máquina não indica situação da operação realizada.
Se for uma máquina, quais serviços ela dispõe	Montar peça preta na posição 1 do palete; Montar peça preta na posição 2 do palete; Montar peça vermelha na posição 1 do palete; Montar peça vermelha na posição 2 do palete.
Se for um transportador, quais serviços de transporte realiza	
Se for um peça, quais serviços necessita e suas precedências	

Fonte: autor.

- b) Máquina de usinagem: Equipamento com a função de usinar peças nas operações de puncionamento e furação. A Tabela 10 mostra as características observadas.

**Tabela 10 Características observadas para máquina de usinagem.**

Aspectos de comunicação	Característica observada
Controlador Lógico	CLP Siemens S7-300.
Meio de Comunicação	Rede industrial Profinet, com meio físico <i>ethernet</i> .
Qual COMANDO para partida	Acionamento da <i>flag</i> “IniP” que está vinculado ao botão de partida da máquina.
Qual INFORMAÇÃO ao equipamento	Através da <i>flag</i> “U1” se indica a realização da operação de puncionar na peça. Através da <i>flag</i> “U2” se indica a realização da operação de furar na peça.
Qual aviso de RESPOSTA do equipamento	Monitoramento da <i>flag</i> “FimP” que indica se a operação chegou ao fim.
Qual a SITUAÇÃO da operação realizada	Esta máquina não indica situação da operação realizada.
Se for uma máquina, quais serviços ela dispõe	Furar peça da posição 1 do palete; Puncionar peça da posição 1 do palete;
Se for um transportador, quais serviços de transporte realiza	
Se for um peça, quais serviços necessita e suas precedências	

Fonte: autor.

- c) Sistema de transporte: Equipamento com a função de transportar peças de uma posição para outra dentro da estação. Em se tratando da estação de transporte,

aqui se percebe que ela é, até então, o sistema de gerenciamento da manufatura, o qual ordena o fluxo de processos. Normal perceber então que este sistema em si terá modificações no programa, a fim de deixar de ser o gerenciador da manufatura e ser então um prestador de serviços de transporte. Então, os sinais que até então gerenciavam os outros CLP do sistema podem ser utilizados para receber comando do agente. A Tabela 11 apresenta as características observadas no sistema de transporte.

**Tabela 11 Características observadas para o sistema de transporte.**

Aspectos de comunicação	Característica observada
Controlador Lógico	CLP Siemens S7-300.
Meio de Comunicação	Rede industrial Profinet, com meio físico <i>ethernet</i> .
Qual COMANDO para partida	Acionamento da <i>flag</i> “IniT” que está vinculado ao botão de partida da máquina.
Qual INFORMAÇÃO ao equipamento	Através da <i>flag</i> “PosInicio” se indica a posição em que se encontra a peça, indicada por um número inteiro de 1 a 6. Através da <i>flag</i> “PosFim” se indica a realização em que posição se deseja entregar a peça, indicada por um número inteiro de 1 a 6.
Qual aviso de RESPOSTA do equipamento	Monitoramento da <i>flag</i> “FimT” que indica se a operação chegou ao fim.
Qual a SITUAÇÃO da operação realizada	Esta máquina não indica situação da operação realizada.
Se for uma máquina, quais serviços ela dispõe	
Se for um transportador, quais serviços de transporte realiza	Transportar peça da posição 1 para posição 2; Transportar peça da posição 1 para posição 3; Transportar peça da posição 1 para posição 4; Transportar peça da posição 1 para posição 5; Transportar peça da posição 1 para posição 6; Transportar peça da posição 2 para posição 1; Transportar peça da posição 2 para posição 3; Transportar peça da posição 2 para posição 4; Transportar peça da posição 2 para posição 5; Transportar peça da posição 2 para posição 6; Transportar peça da posição 3 para posição 1; Transportar peça da posição 3 para posição 2; Transportar peça da posição 3 para posição 4; Transportar peça da posição 3 para posição 5; Transportar peça da posição 3 para posição 6; Transportar peça da posição 4 para posição 1; Transportar peça da posição 4 para posição 2; Transportar peça da posição 4 para posição 3; Transportar peça da posição 4 para posição 5; Transportar peça da posição 4 para posição 6; Transportar peça da posição 5 para posição 1; Transportar peça da posição 5 para posição 2; Transportar peça da posição 5 para posição 3; Transportar peça da posição 5 para posição 4; Transportar peça da posição 5 para posição 6; Transportar peça da posição 6 para posição 1;

	Transportar peça da posição 6 para posição 2; Transportar peça da posição 6 para posição 3; Transportar peça da posição 6 para posição 4; Transportar peça da posição 6 para posição 5;
Se for um peça, quais serviços necessita e suas precedências	

Fonte: Autor.

- d) Produto a ser produzido: o produto pode ser vinculado a um sistema computacional através de uma indexação em seu palete de transporte. Neste caso há uma identificação física no palete que é sensoriada em cada ponto de estação do transportador, que fornece o número do palete em questão. Para 5 pinos metálicos há a referência ao palete 5, para 2 pinos metálicos há a referência ao Palete 2, e assim para os demais. Para este produto tem-se então um mapeamento para verificar os serviços que cada um ocupa e quais suas precedências. A Tabela 12 apresenta o resultado da análise dos serviços que compõe o produto.

**Tabela 12 Características observadas para o produto a ser produzido.**

Aspectos de comunicação	Característica observada	
Controlador Lógico	Computador com a indexação por pinos metálicos no palete, que traduzem uma contagem relativa ao palete em questão.	
Meio de Comunicação	Rede <i>ethernet</i> entre computadores.	
Qual COMANDO para partida	A solicitação de produção.	
Qual INFORMAÇÃO ao equipamento	Tipo de montagem e usinagem que se deseja.	
Qual aviso de RESPOSTA do equipamento	Mensagem de fim da solicitação de produção.	
Qual a SITUAÇÃO da operação realizada	Mensagens de local em que a produção se encontra.	
Se for uma máquina, quais serviços ela dispõe		
Se for um transportador, quais serviços de transporte realiza		
Se for um peça, quais serviços necessita e suas precedências	Produto 1	Montar peça preta posição 1
		Montar peça preta posição 2
	Produto 2	Montar peça preta posição 1
		Montar peça vermelha posição 2
	Produto 3	Montar peça vermelha posição 1
		Montar peça preta posição 2
	Produto 4	Montar peça vermelha posição 1
		Montar peça vermelha posição 2
Produto 5	Montar peça preta posição 1	
	Montar peça preta posição 2	
	Furar peça da posição 1	
Produto 6	Montar peça preta posição 1	

		Montar peça vermelha posição 2
		Furar peça da posição 1
	Produto 7	Montar peça vermelha posição 1
		Montar peça preta posição 2
		Furar peça da posição 1
	Produto 8	Montar peça vermelha posição 1
		Montar peça vermelha posição 2
		Furar peça da posição 1
	Produto 9	Montar peça preta posição 1
		Montar peça preta posição 2
		Puncionar peça da posição 1
	Produto 10	Montar peça preta posição 1
		Montar peça vermelha posição 2
		Puncionar peça da posição 1
	Produto 11	Montar peça vermelha posição 1
		Montar peça preta posição 2
		Puncionar peça da posição 1
	Produto 12	Montar peça vermelha posição 1
		Montar peça vermelha posição 2
		Puncionar peça da posição 1
	Produto 13	Montar peça preta posição 1
		Montar peça preta posição 2
		Furar peça da posição 1
		Puncionar peça da posição 1
	Produto 14	Montar peça preta posição 1
		Montar peça vermelha posição 2
		Furar peça da posição 1
		Puncionar peça da posição 1
	Produto 15	Montar peça vermelha posição 1
		Montar peça preta posição 2
		Furar peça da posição 1
		Puncionar peça da posição 1
	Produto 16	Montar peça vermelha posição 1
		Montar peça vermelha posição 2
		Furar peça da posição 1
		Puncionar peça da posição 1

Fonte: autor.

Obs: as operações de furar e puncionar necessitam das operações de montagem como precedência.

## Passo 2 – Definir as classes de agentes que haverão na plataforma.

Definem-se aqui os tipos de agentes que atuarão no Sistema Multiagente.

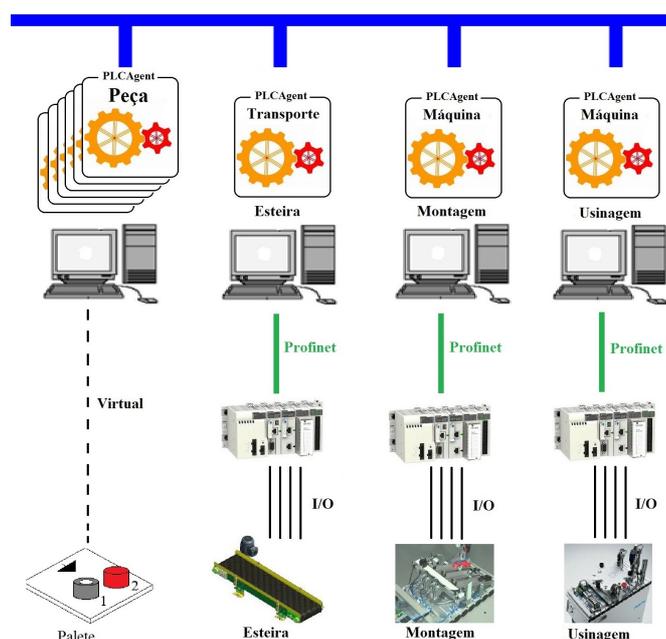
- a) Agente Peça: a manufatura de cada produto será um agente peça, cuja função será negociar com os recursos sua disponibilidade e gerenciar a ordem de manufatura segundo as precedências definidas. Este agente representará um palete, mas não necessariamente estará embarcado neste. Ele será executado em um computador que fará um vínculo com o palete por indexação. Cada

palete tem um número específico, definido como um código que será reconhecido pelos pontos de abastecimento de estações.

b) Agente Máquina: haverão 2 agentes máquina, um responsável pela montagem e outro responsável pela usinagem. Estes agentes necessitam comunicar-se com os seus respectivos Controladores lógicos programáveis. E esta aplicação farão por meio de rede industrial Profinet, que disponibilizará o mapa de memória do CLP para ser compartilhado. Bastando então reconhecer o local de cada memória desejada no arquivo que é fornecido. Esta tarefa é executada pelo *AgentCom*, um agente especializado para proceder esta comunicação. E se valerá de uma classe específica para comunicação no protocolo Profinet para JAVA.

c) Agente de Transporte: haverá 1 agente de transporte, que terá a função de transportar paletes de um ponto a outro dentro da esteira transportadora, onde estão definidas 6 posições para alocar estações. Este agente, assim como o agente máquina, também necessitará se comunicar com o CLP e fará isto através do *AgentCom* e se valerá também da classe de comunicação no protocolo Profinet.

A Figura 66 apresenta o modelo de agentes definido com a aplicação do método.



**Figura 66 Configuração dos agentes para a manufatura proposta.**

Fonte: Autor.

Nesta figura percebe-se o agente peça com uma relação virtual por indexação ao palete. Também se vê a conexão via protocolo Profinet do agente com o CLP para os agentes máquina e agentes de transporte. Aqui é utilizada uma biblioteca desenvolvida por Davide Nardella (NARDELLA, 2016), que contém as classes necessárias para comunicar-se em linguagem JAVA com controlador lógico programável da SIEMENS série S7, utilizado neste cenário. Os códigos desenvolvidos nesta biblioteca são de tipo LGPL (*Lesser General Public License* – licença de uso público) e nesta classe os agentes podem, através da conexão profinet, ter interação com as variáveis de entrada e saída do controlador lógico programável.

Do CLP para os periféricos a conexão é via entradas e saídas.

### Passo 3 – Implantação da plataforma multiagente.

Neste passo cabe a execução da plataforma JADE, escolhida e justificada nos capítulos anteriores. Como infraestrutura para execução desta plataforma, é necessário que os computadores

sejam ligados a uma rede Ethernet, com protocolo TCP/IP, normalmente conseguido através de um *switch* ou *hub* e configuração de rede local pelo sistema operacional do computador.

A implantação da plataforma multiagente consiste em instalar a plataforma JADE em um computador que esteja conectado em um domínio de rede em que os outros agentes serão conectados. Isto ocorrerá através do comando:

```
java jade.Boot -gui <nome_agente>:<nome_da_classe>
```

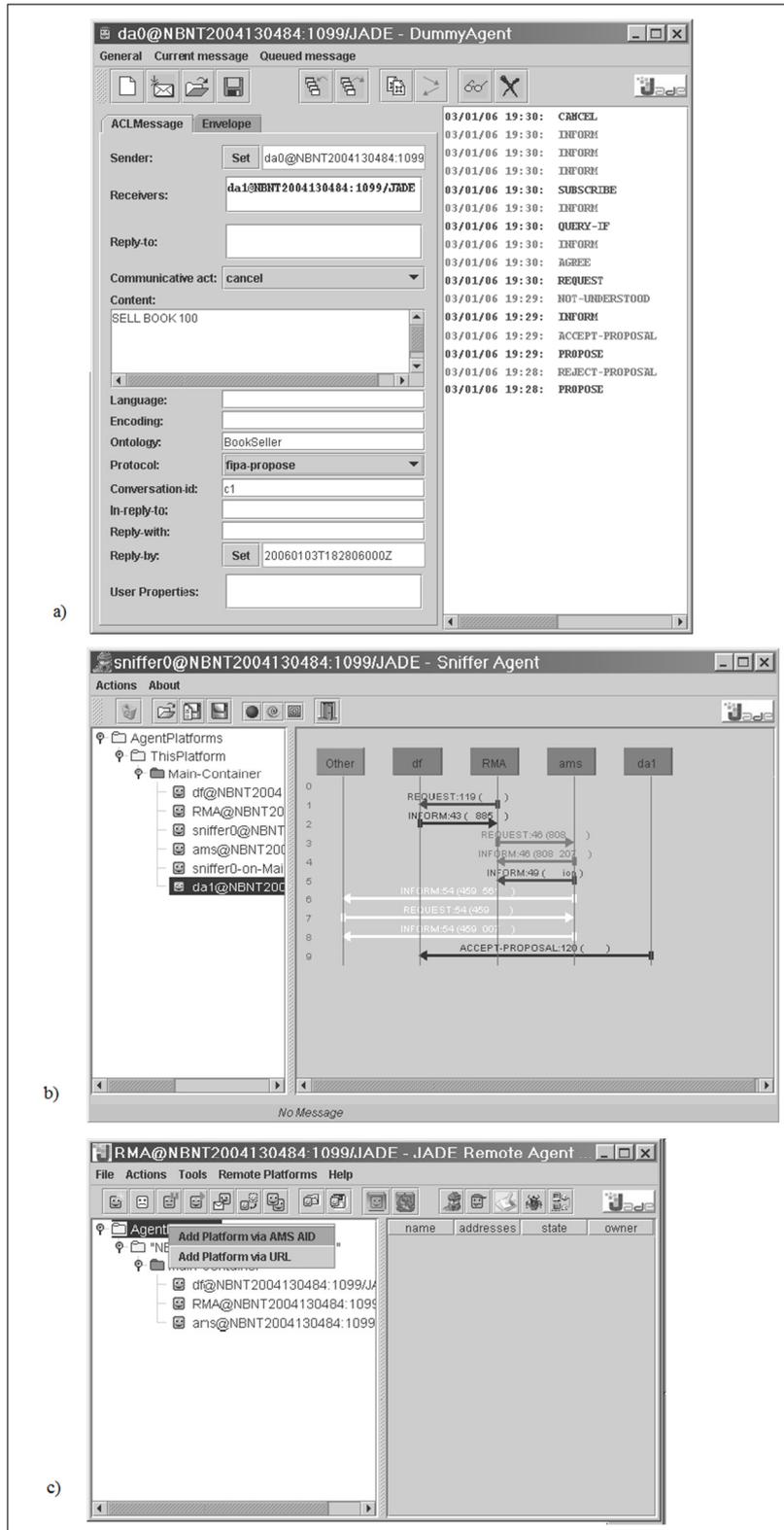
Os demais agentes que entrarão na plataforma serão executados pelo comando:

```
java jade.Boot -container <nome_agente>:<nome_da_classe>
```

A diretiva “-gui” indica que se está instalando a plataforma, ao passo que a diretiva “-container” indica que se está instanciando um agente na plataforma.

A instalação da plataforma JADE irá dispor, além do serviço de interconexão entre os agentes, serviço de identificação dos agentes que estão na plataforma (AMS), serviço de postagem e busca de funcionalidades através da identificação dos agentes (DF), serviço de monitoramento de troca de mensagens entre os agentes e serviço de envio e recebimento de mensagens, com fins de teste da plataforma e dos agentes.

A Figura 67 representa as interfaces da plataforma JADE como serviços à disposição do usuário.



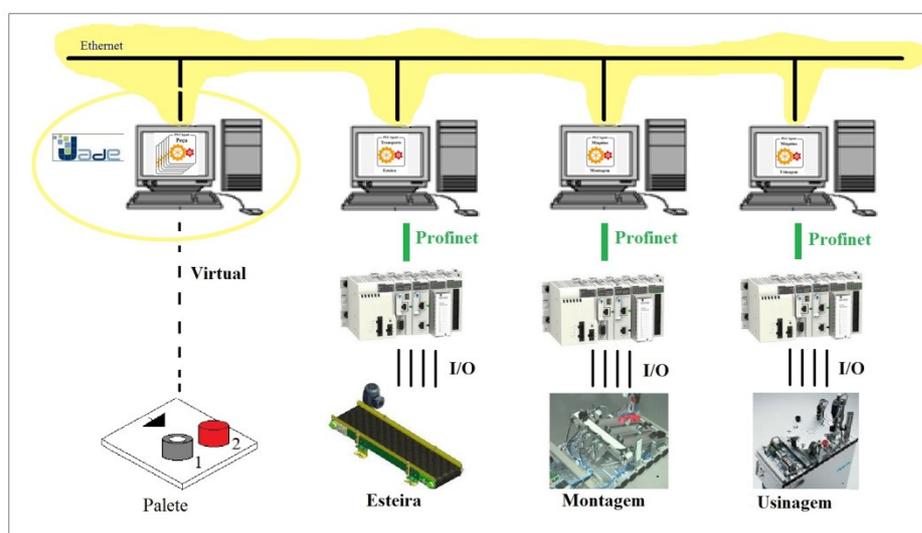
**Figura 67** Serviços da plataforma JADE.

Fonte: adaptado de (BELLIFEMINE et al., 2007).

Na figura, em “a” se percebe a interface para proceder a troca de mensagens entre os agentes. Em "b" se percebe o agente *Sniffer*, destinado a monitorar a troca de mensagens. Em “c” há o serviço identificador de agentes que estão atuando na plataforma.

Uma vez os computadores conectados em rede, basta então executar a plataforma JADE, que neste cenário de uso será utilizado o mesmo computador que estará executando os “agentes peças”. Não há restrição quanto a utilizar um computador específico para uso da plataforma, mas por motivo de economia de recurso se utilizará o mesmo computador.

A Figura 68 apresenta a estrutura de rede para uso da plataforma JADE.



**Figura 68 Configuração da plataforma JADE para o sistema de manufatura.**

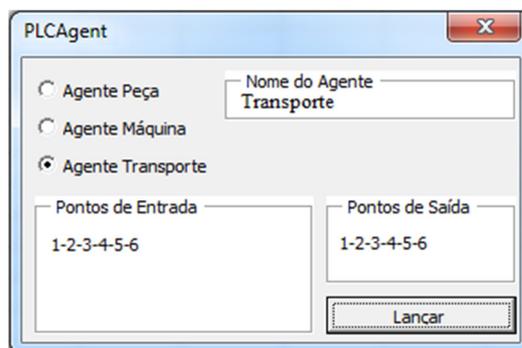
Fonte: Autor.

A plataforma JADE deverá ser instalada no computador em destaque, conforme orientações da referência em (BELLIFEMINE et al., 2007).

#### Passo 4 – Implantação do PLCAgent para cada agente, dentro da plataforma.

Com a plataforma JADE em execução é a vez de executar os agentes a partir do PLCAgent. Este agente irá se comportar como o tipo selecionado e atenderá aos parâmetros a ele configurados. Estes parâmetros foram definidos no passo 1:

- a) Implantação do PLCAgent Transporte: este agente deverá ser lançado para execução com as configurações definidas no passo 1 e que estão descritas na interface do PLCAgent, demonstrado na Figura 69.

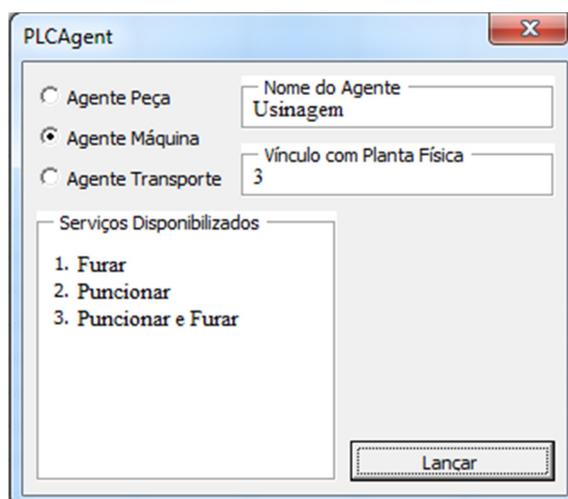


**Figura 69 Implantação do PLCAgent Transporte.**

Fonte: Autor.

O agente de transporte, que foi configurado para atender pelo nome de “Transporte”, irá buscar paletes nos pontos 1, 2, 3, 4, 5 e 6 da esteira e entregará paletes nos pontos 1, 2, 3, 4, 5 e 6 da esteira.

- b) Implantação dos PLCAgent Agente Máquina: este agente deverá ser inicializado com as configurações definidas no passo 1 e que estão descritas na interface do PLCAgent, demonstrado na Figura 70, sendo lançado para o agente máquina de usinagem e agente máquina de montagem.



**Figura 70 Implantação do PLCAgent Máquina de usinagem e manipulação.**

Fonte: Autor.

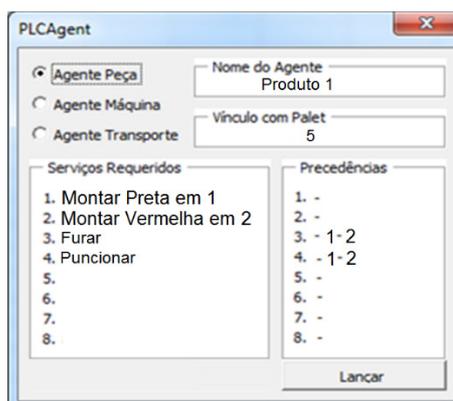
O agente máquina configurado com o nome “Montagem” estará vinculado fisicamente com a posição 2 da esteira de transporte e prestará os serviços de montagem de peça preta nas posições 1 e 2 do palete, assim como os serviços de montagem de peça vermelha nas posições 1 e 2 do palete.

#### Passo 5 - Execução da plataforma multiagente.

A execução da plataforma multiagente consiste então em lançar os agentes peça para a produção.

Este lançamento ocorre através da implantação do PLCAgent Peça. Este agente tem a função de negociar com os agentes do sistema a fim de obter os recursos necessários para manufaturar cada tarefa desejada. Em termos de fluxo, a primeira função deste agente é buscar no DF da plataforma JADE o nome dos agentes capazes de executar os primeiros serviços que não requerem precedência ou que as precedências já estão resolvidas. Uma vez encontrados estes agentes começam a tratativa para definir qual deles atenderá. Definido o agente, cabe agora negociar com o transportador o deslocamento na esteira da posição atual para a posição do agente que irá atender ao serviço solicitado. Esta rotina irá se repetir até que todos os serviços da lista sejam encerrados. Uma outra técnica que será abordada na continuidade deste trabalho é um descompromisso do agente peça para com o sistema de transporte, de forma que o agente máquina que se ofertar a atender ao pedido do agente peça é que terá que dar conta de negociar com o agente de transporte, fornecendo um serviço completo: transporte e manufatura.

A Figura 71 apresenta a interface e configuração do PLCAgent para o agente peça.



**Figura 71 Implantação do PLCAgent Peça.**

Fonte: Autor.

Para esta solicitação de produção o agente peça que foi configurado para atender ao nome de “Produto1”, estará vinculado ao palete numero 5, necessita que seja montado uma peça preta na posição 1 do palete, uma peça vermelha na posição 2 do palete, que tenha a peça da posição 1 sendo furada e puncionada.

E para dar continuidade na produção é só ir lançando os agentes peça conforme as características desejadas para cada produto.

A Figura 72 apresenta um o sistema integrado de manufatura no qual o cenário foi ensaiado.



**Figura 72 Sistema integrado de manufatura em que o cenário foi ensaiado.**

Fonte: Autor.

A figura mostra um sistema integrado de manufatura desenvolvido pela empresa FESTO Didatic e que dispõe suas estações de forma modular, sendo cada estação controlada por um controlador lógico programável independente.

Nesta pequena implementação se comprova que a aplicação da metodologia funciona e é viável, propiciando aos controladores lógicos programáveis as características e funcionalidade de sistemas multiagente, que organizados em um sistema produtivo atende aos requisitos de diversidade, agilidade e auto-organização.

Fato importante a ser salientado é a ocorrência de *Deadlock*<sup>13</sup>, fator que poderá ocorrer nesta abordagem devido ao modo com que as relações entre os agentes ocorrem.

Se admitir a situação em que uma peça “P1” que está na estação A necessita dos serviços da estação B, mas uma peça “P2” que se encontra na estação B necessita justamente dos serviços da estação A. Neste caso a peça “P1” receberá da Estação B a resposta que está ocupada, ao passo que a peça “P2” receberá também a resposta da estação A de que esta também se encontra ocupada. Nesta condição de *deadlock* o sistema se bloqueia e não consegue avançar seus passos, a menos que uma intervenção externa ocorra.

Como o objetivo deste trabalho é avaliar as interações entre os agentes que representam máquina, transporte e peças, e a forma com que os sistemas multiagentes podem trazer um controlador lógico programável para participar desta abordagem de sistemas auto-organizáveis, então o problema do *deadlock* é encarado como uma consequência.

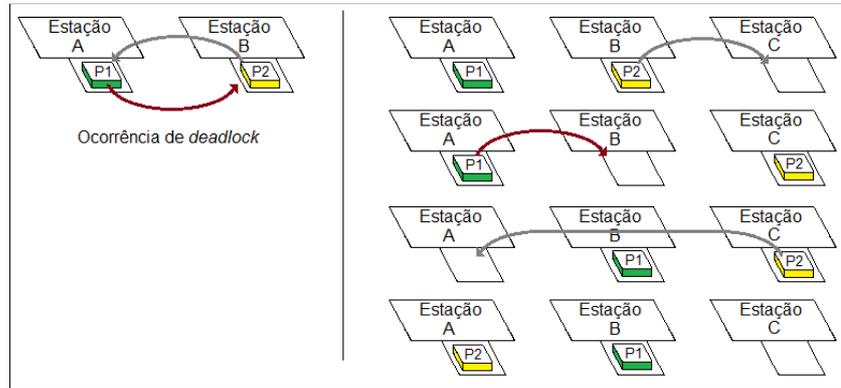
A solução para este fato é a peça que estiver solicitando um serviço de uma estação e receber uma resposta de que esteja ocupada por uma quantidade definida de tentativas, então esta peça solicita ao transportador a ida para uma posição livre, de uma estação desocupada ou

---

<sup>13</sup> *DeadLock* é um termo utilizado para um fator que ocorre em sistemas dinâmicos, em que a condição de operação de um sistema A depende da ocorrência do sistema B, que por sua vez depende da ocorrência do sistema A. Esta relação de interdependência irá provocar uma situação em que não há como haver segmento sem uma intervenção externa. Este fato, parada do sistema, dá-se o nome de *deadlock*.

ponto de entrada ou saída, de forma a liberar a estação que já lhe manufaturou. Esta ação resolve o problema de *deadlock* e permite que o fluxo de processos volte a dar segmento.

A Figura 73 apresenta um exemplo da ocorrência de *deadlock* e sua solução.



**Figura 73 Ocorrência de *deadlock* e sua solução.**

Fonte: Autor.

Nesta figura se observa à esquerda a ocorrência do *deadlock* e na direita a forma com que o *deadlock* se resolve, uma vez que a peça “P2” toma a iniciativa de se deslocar para posição da estação C, liberando a estação B para tratativas de alocação.

## **6 ESTUDO DE CASO PARA COMPARAÇÃO DE ABORDAGENS**

Nesta fase do trabalho, o ensaio da metodologia aplicada necessita ser comparado com outros sistemas produtivos, a fim de que se estabeleça uma análise crítica quanto sua aplicabilidade e eficácia. A aplicabilidade foi observada no capítulo anterior e se mostrou factível, quando a metodologia foi ensaiada em uma situação com duas máquinas e um sistema de transporte.

Para análise de aplicação se opta agora por uma planta com mais elementos de manufatura. Aproveitando uma demanda da ABIMAQ<sup>14</sup> de implantar uma planta de demonstração da Manufatura Avançada, com as características de produção de produtos customizados e de lote único, valeu-se desta implementação na qual o autor participou para obter dados de implementação, organização da produção e estrutura de interação entre os componentes do sistema. Com isto se tem uma aplicação real com 9 estações que oferecem serviços para a produção de um produto. Uma aplicação bem característica de processo fabril.

E para realizar a devida comparação se faz necessário definir as métricas a serem analisadas, analisar uma estrutura fabril com gestão por Sistema de Gerenciamento da Produção (MES), analisar esta mesma estrutura com gestão por Sistema Integrado de Manufatura (CIM) e analisar também esta mesma manufatura com gestão por Sistemas Auto-Organizáveis (Self-x). Com estas três comparações se pode ter uma boa análise da aplicação de sistemas auto-organizáveis e a aplicação do método proposto para comprovação desta tese.

### **6.1 MÉTRICAS A SEREM ANALISADAS**

Para poder efetuar uma análise de desempenho do sistema de manufatura, quando aplicado o método, faz-se necessário estabelecer métricas para comparação. As métricas

---

<sup>14</sup> Associação Brasileira da Indústria de Máquinas e Equipamentos foi fundada em 1975, com o objetivo de atuar em favor do fortalecimento da Indústria Nacional, mobilizando o setor, realizando ações junto às instâncias políticas e econômicas, estimulando o comércio e a cooperação internacionais e contribuindo para aprimorar seu desempenho em termos de tecnologia, capacitação de recursos humanos e modernização gerencial.

dimensionam ou qualificam características de requisitos de um sistema, como descrito em (PEIXOTO, 2012), permitindo análise comparativa. As métricas podem ser medidas diretamente (custo, esforço, linhas de código, velocidade de execução, memória, número de erros, entre outras) ou indiretamente (funcionalidade, qualidade, complexidade, eficiência, confiabilidade, manutenibilidade<sup>15</sup>, modularidade, entre outras).

As necessidades da indústria por diversidade, agilidade e auto-organização requer que o desempenho do sistema de manufatura seja observado por características mais qualitativas do que quantitativas. Isto implica na escolha de métricas qualitativas para poder avaliar os sistemas produtivos no atendimento às necessidades da indústria.

As métricas definidas como adequadas e pertinentes para a análise devem remeter aos aspectos da necessidade da manufatura, abordadas na introdução deste trabalho, ditas como necessidade de atender a diversidade de produção, agilidade e auto-organização.

As métricas sugeridas para analisar o cenário proposto são apresentadas na Tabela 13, junto com sua justificativa técnica, forma de obtenção e critério de avaliação.

**Tabela 13 Relação de métricas utilizadas na validação experimental.**

<b>Métricas [requisito]</b>	<b>Justificativa técnica</b>	<b>Forma de obtenção</b>	<b>Critério para avaliação</b>
<b>[Diversidade]</b> Manter a produção com a mudança nos requisitos.	Atender a diversidade de mercado que solicita produtos distintos.	Alterar a especificação do produto e solicitar sua produção.	O sistema deverá manter a produção sem necessitar parar o sistema pela mudança de requisitos do produto.
<b>[Diversidade]</b> Manter a produção com inserção de um único produto em meio a um lote.	Durante a produção de um lote de um produto há a necessidade de testar a manufatura de protótipos.	Solicitar uma quantidade de produtos idênticos e inserir um produto distinto neste lote.	A produção deve se manter mesmo com a necessidade de fazer um único produto, diferente do produto que se estava produzindo.
<b>[Agilidade]</b> Manter a produção com a retirada de estações.	Em caso de falha em uma estação, o sistema de montagem deve continuar o processo, cabendo a outra estação	Montar o sistema com estações em que tenham a mesma habilidade. Requerer esta habilidade no produto e, durante o	Sendo o sistema modular, saindo uma das estações, outra deve assumir a tarefa, admitindo ter a mesma funcionalidade,

<sup>15</sup> Manutenibilidade é uma característica inerente a um projeto de sistema ou produto, e se refere à facilidade, precisão, segurança e economia na execução de ações de manutenção nesse sistema ou produto (BLANCHARD, 2003). Em engenharia de software, manutenibilidade é um aspecto da qualidade de software que se refere à facilidade de um software de ser modificado a fim de corrigir defeitos, adequar-se a novos requisitos, aumentar a suportabilidade ou se adequar a um ambiente novo.

	absorver a tarefa faltante.	processo, desligar uma das estações.	sem requerer alteração na programação.
<b>[Agilidade]</b> Mudar a ordem de processo em função da não disponibilidade do recurso.	Os sistemas produtivos podem gerar acúmulos de tarefas em um dado recurso, assim, a produção deve ter capacidade de modificar a ordem de execução de tarefas que não requeiram precedência.	Solicitar peças com características idênticas, mas sem precedência entre os recursos, verificando se ocorre a alteração da ordem do processo pela peça.	A peça deve ser capaz de modificar sua sequência de produção em função da não disponibilidade do recurso.
<b>[Auto-organização]</b> Manter a produção com a inserção de estações.	Em caso de formação de gargalo, o sistema deve permitir que mais estações fossem inseridas em auxílio ao sistema de montagem.	Montar o sistema com estações com habilidades distintas. Requerer produto e, durante o processo, inserir mais uma estação com a mesma habilidade.	O sistema deve permitir expansão. A inserção de uma estação não deve requerer alterações na programação, nem demandar uma parada do sistema.
<b>[Auto-organização]</b> Manter a produção mesmo com a mudança da estação de posição física na planta.	Um sistema de manufatura deve se adaptar com as mudanças de leiaute no meio produtivo.	Com o sistema em marcha retirar uma estação de uma posição no transportador e colocar em outra.	O sistema deve manter a produção mesmo com a mudança de posição da estação na planta.

Fonte: Autor.

Como já mencionado que haverá interferência na lógica local do CLP para interagir através dos sinais de comando, informação, situação e resposta, admite-se aqui que uma interferência percentual de até 10% é tolerável para um sistema produtivo industrial.

## 6.2 PLANTA COM GESTÃO POR MES

### 6.2.1 Cenário na gestão por MES.

Para o estudo de um cenário real semelhante foi utilizado uma aplicação desenvolvida pela ABIMAQ, junto com 22 empresas<sup>16</sup>, que formaram um consórcio para implementar uma planta demonstrativa de manufatura avançada, com o objetivo de expor em um evento intitulado FEIMEC – Feira Mundial de Máquinas e Ferramentas (FEIMEC, 2016), realizado em São

<sup>16</sup> Relação das 22 empresas: ABB, AutoDesk, Baltec, Festo, Hexagon, Instituto Mauá de Tecnologia, Instituto SENAI de Inovação em Metalmeccânica, MCK, Metal Work, Mult-e, NCSsystems, Phoenix Contact, PLMX, Pollux, KUKA, Rexroth, Romi, Sick, Siemens, SMC, Stäubli e Westcon.

Paulo no mês de maio de 2016. A Figura 74 mostra uma foto do estande onde esta planta foi implementada para demonstração dos aspectos da manufatura avançada.



**Figura 74 Estande da Manufatura Avançada na FEIMEC2016.**

Fonte: autor.

A importância deste modelo a ser utilizado como referência é que ele também se propõe a resolver os problemas de atender a um produto customizado e ser um lote pequeno de produção. Neste cenário valeu-se então do conceito da Indústria 4.0<sup>17</sup> (HERMANN;PENTEK;OTTO, 2015), onde havia interação entre os elementos produtivos para dar o melhor arranjo ao fluxo de produção.

A proposta do sistema de manufatura era a produção de um objeto porta canetas e celulares, para ser colocado sobre mesa de escritório, onde o consumidor poderia optar por uma inscrição de frase com até 12 caracteres, a escolha de três cores de canetas diferentes e, através do fornecimento do seu telefone celular para escaneamento, uma cavidade no objeto era usinada para que encaixasse perfeitamente o telefone celular do cliente. A Figura 75 mostra o produto customizado gerado na planta de Manufatura Avançada em estudo.

---

<sup>17</sup> Conhecida como Indústria 4.0 na Alemanha, Smart Manufacturing nos Estados Unidos e Manufatura Avançada no Brasil, são termos que tratam do mesmo conceito que visa combinar modernos recursos de automação industrial com as ferramentas computacionais para permitir a sistemas produtivos troquem entre si informações ao longo do processo, interagindo, alocando recursos e tomando decisões quanto ao fluxo produtivo a ser tomado, em tempo de execução.

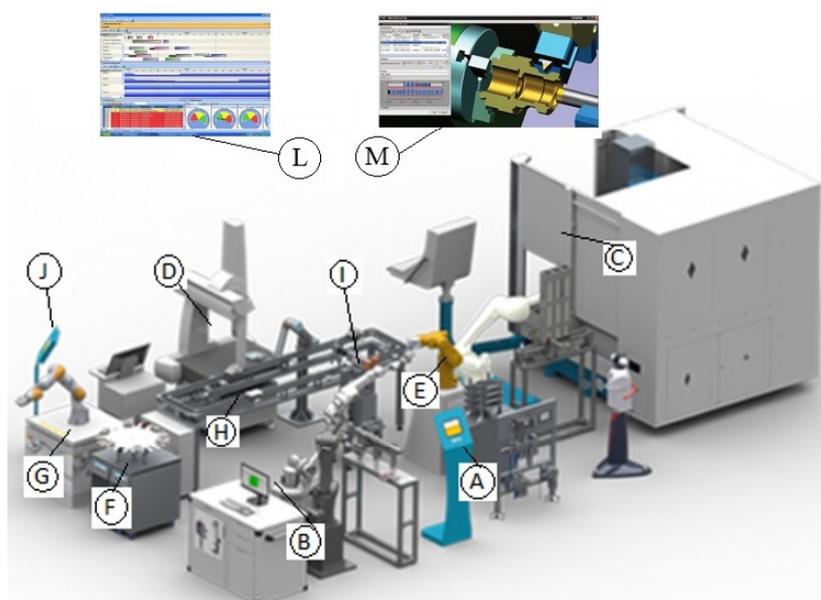


**Figura 75 Produto gerado na planta de Manufatura Avançada.**

Fonte: Autor.

Aqui fica evidente a proposta de atender à necessidade de um produto customizado, que muda de cliente para cliente, e uma produção de lote único, para cada cliente. E o sistema de manufatura deve dar conta desta necessidade.

A manufatura fora proposta como um conjunto de recursos que se dispõe a serviços distintos, como pode ser observado na Figura 76 e detalhado na sequência.



**Figura 76 Ilustração da planta de Manufatura Avançada.**

Fonte: Autor.

O detalhamento de cada recurso na manufatura é descrito como:

A – Toten de Entrada: aplicativo de software cuja função é de interpretar uma mensagem codificada em QRCode, interagir com o usuário para customizar

- a mensagem a ser inscrita na peça e permitir ao usuário a escolha das cores dos lápis que serão montados no produto;
- B – Escaneamento: conjunto de escâner ótico manipulado por um robô articulado, com a função de identificar as dimensões do celular do cliente, em sua largura e espessura;
- C – Usinagem: centro de usinagem de 5 eixos com a função de dar a forma ao produto a partir de um programa CNC (SOUZA; ULBRICH,2013);
- D – Inspeção: máquina de medir coordenada com um robô articulado, com a função de medir a peça e verificar se ela se encontra dentro das dimensões desejadas, resguardado a devida tolerância dimensional;
- E – Montagem: robô articulado com um magazine gravitacional de capacidade para quatro tipos de lápis, diferenciados pelo tipo de cor. Tem a função de montar os lápis conforme a solicitação do cliente;
- F- Almoxarifado: eixo elétrico rotativo com a função de armazenar as peças sendo produzidas para posterior entrega aos clientes;
- G – Entrega: robô articulado com a função de entregar a peça desejada ao cliente;
- H- Esteira: sistema de transporte de peças a partir de paletes, com identificadores de RFID que identificam as peças em cada uma das seis estações de parada. Gerenciada por um controlador lógico programável;
- I – Descarte: manipulador cartesiano com a função de descartar as peças que não forem consideradas adequadas pela inspeção dimensional;
- J – Toten Saída: aplicativo de software cuja função é interpretar uma mensagem via QRCode e interagir com a estação de entrega e almoxarifado para entregar a peça ao cliente;

L – MES: software com a função de hospedar as ordens de serviço e identificar a situação de cada equipamento inserido no sistema;

M – PLM-NX: software com a função de gerar um programa CNC a partir dos dados customizados do cliente.

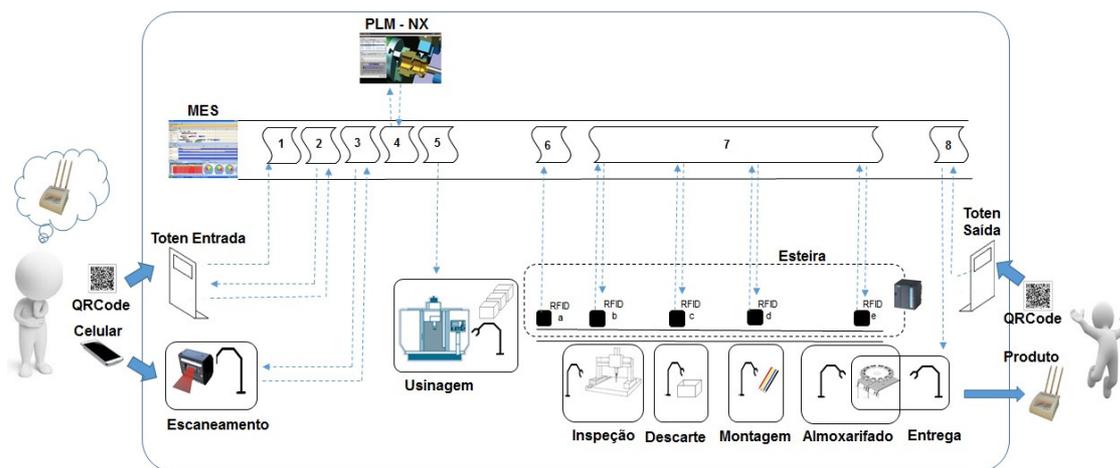
A Figura 77 a seguir pode ser observada a planta real da Manufatura Avançada.



**Figura 77 Imagem da planta real de Manufatura Avançada.**

Fonte: Autor.

O processo prevê uma manufatura em que a ordem com que o produto será executado é definido na ordem de serviço [OS], a partir de então a cada estação que o produto passa é recorrido à ordem de serviço para saber se há alguma operação para ser executada naquela estação. A Figura 78 a seguir mostra o diagrama da estrutura desta planta de manufatura em estudo.



**Figura 78 Diagrama da estrutura da planta demonstrativa da Manufatura Avançada.**

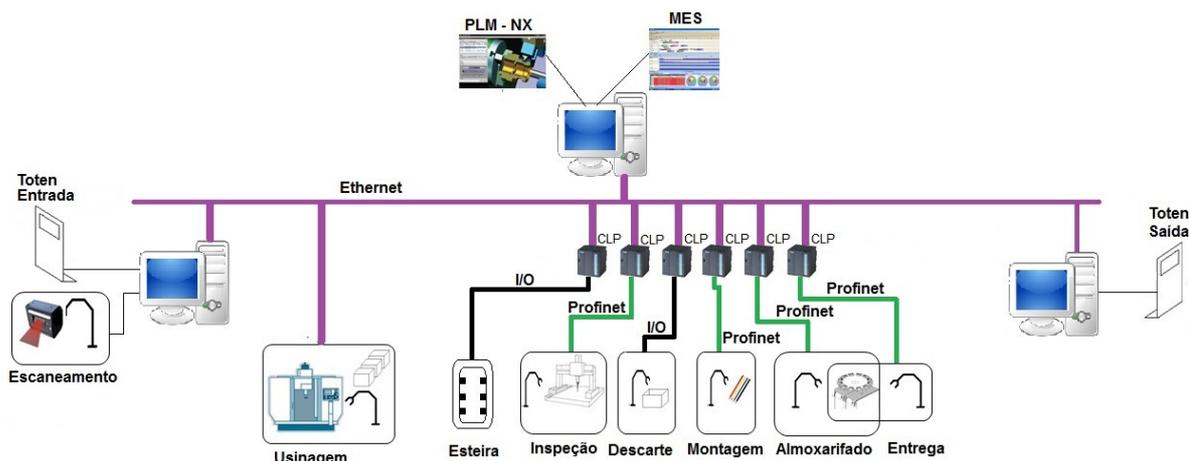
Fonte: autor.

A sequência de operação começa com o usuário inserindo um QRCode com um código identificador de seu pedido no *totem entrada*. Neste instante é gerada a ordem de serviço no MES, identificado na ilustração como passo 1. O passo 2 consiste na a ordem de serviço interagir com o usuário através do *totem entrada* para definir a mensagem que será inscrita (até 12 caracteres) e a escolha de três cores de lápis distintas para a montagem. No passo seguinte, passo 3, a ordem de serviço então interage com o *escaneamento* para detectar a espessura e largura do celular do cliente. No passo 4 a ordem de serviço então interage com o software *PLM-NX* para elaboração do programa CNC para a confecção desta peça com uma geometria pré-definida, a inscrição que o cliente solicitou e a cavidade para suportar o celular nas dimensões que foram medidas pelo escâner. No passo 5 gera-se então a ordem para a *usinagem* proceder a fabricação desta peça a partir de uma matéria prima disponibilizada. Cada bloco de matéria prima é identificado com uma etiqueta de RFID e, uma vez usinada a peça, o robô articulado vinculado à máquina de usinagem leva a peça para a *esteira* e vincula o código do RFID da peça à ordem de serviço, identificado aqui como passo 6. A partir deste ponto começa a capacidade deste sistema de modificar seu fluxo de produção segundo a rotina que está descrita na ordem de serviço (passo 7). A *esteira* gira continuamente e a cada vez que o palete passa pela frente de uma estação ele é reconhecido pela etiqueta de RFID da peça, onde é então

questionado ao MES se aquela ordem de serviço tem alguma operação a ser realizada. Caso negativo a peça segue para próxima estação, caso positivo, a peça é retirada e inserida na estação para que a operação descrita na ordem de serviço seja então realizada. Assim, a ordem com que as operações serão realizadas a partir daqui está definida na ordem de serviço e pode mudar de produto para produto. A última operação de cada peça é ser acondicionada no almoxarifado, e isto é realizado por um robô articulado, interagindo com o almoxarifado rotativo, que tem capacidade de armazenar até 12 peças. Uma vez que o cliente se dirija ao totem de saída (passo 8) ele insere o código de QRCode, que é verificado na ordem de serviço se já está com todos processos finalizados e aguardando no almoxarifado. Caso positivo a ordem de serviço emite um comando ao robô articulado na estação de entrega, que interage com o almoxarifado para entregar a peça ao cliente.

Nesta planta não havia o objetivo de produzir rapidamente, mas de produzir conforme a customização do cliente e num lote único. As máquinas e equipamentos de manufatura, manipulação e inspeção foram implementados com uma velocidade reduzida, para servir como demonstração.

Um fato a observar são as relações de comunicação entre os equipamentos, pois haviam controladores de fabricantes distintos. Foi necessário estabelecer uma hierarquia para que a comunicação fluísse. O diagrama da Figura 79 apresenta a hierarquia da comunicação e interação entre os elementos do sistema.



**Figura 79 Hierarquia da comunicação e interação com MES.**

Fonte: Autor

Neste diagrama se pode observar que, em termos de comunicação, há uma centralização dos elementos de manufatura no CLP e este é que se comunica com o MES. Porém, a forma com que é concebida a programação, faz com que a sequência de produção seja definida pela peça em produção, através da ordem de produção no MES, o que garante a flexibilidade ao sistema. Mas também cada recurso está conectado diretamente a um CLP, logo a uma lógica de programa, o que impede a inserção ou retiradas de estações sem a necessidade de programar o CLP.

### 6.2.2 Avaliação na gestão por MES.

Avaliando as características que o processo produtivo apresentou com a utilização do MES como sistema de gestão, se observou que há o atendimento de dois requisitos importantes: diversidade de produto e lote pequeno de produção. Isto permitiu a produção de produtos customizados e lote único de produção, requerido pelo cliente.

Porém, se observa que ainda há uma dependência muito forte de um programador especializado para proceder alterações na planta, em termos de software, a cada vez que há uma

inserção, relocação ou retirada de estações de trabalho da planta. A tabela 14 a seguir mostra o resultado desta experimentação, segundo as métricas que foram definidas.

**Tabela 14 Análise da planta de Manufatura Avançada a partir das métricas propostas.**

Métricas	Planta demonstrativa de Manufatura Avançada gerenciada por MÊS
Manter a produção com a mudança nos requisitos.	Uma vez que os novos requisitos possuam recursos na planta para atendê-lo, são então descritos na Ordem de Serviço e passam a definir o fluxo de produção para tal. Isto ocorre sem a necessidade de parada da produção nem reprogramação dos controladores.
Manter a produção com inserção de um único produto em meio a um lote.	O sistema atende ao requisito de lote único, pois cada solicitação do usuário corresponde a uma Ordem de Serviço que é um lote de uma única peça.
Manter a produção com a retirada de estações.	A retirada de uma estação requer uma reprogramação no CLP que gerencia o sistema de transporte e uma reprogramação no planejamento do MES, implicando consequentemente na parada do sistema para <i>download</i> da nova programação.
Mudar a ordem de processo em função da não disponibilidade do recurso.	A cada vez que a peça chega a uma máquina, é solicitado ao MES a disponibilidade desta máquina e a operação a ser realizada. Assim, a ordem do processo se encontra no MES e pode ser alterada em função da não disponibilidade do recurso, desde que não haja precedência.
Manter a produção com a inserção de estações.	A inserção de uma estação requer uma reprogramação no CLP que gerencia o sistema de transporte e uma reprogramação no planejamento do MES, implicando consequentemente na parada do sistema para <i>download</i> da nova programação.
Manter a produção mesmo com a mudança da estação de posição física na planta.	Necessária a reprogramação do CLP da esteira e mudança física dos componentes de sensoriamento, implicando consequentemente na parada do sistema para <i>download</i> da nova programação.

Fonte: Autor.

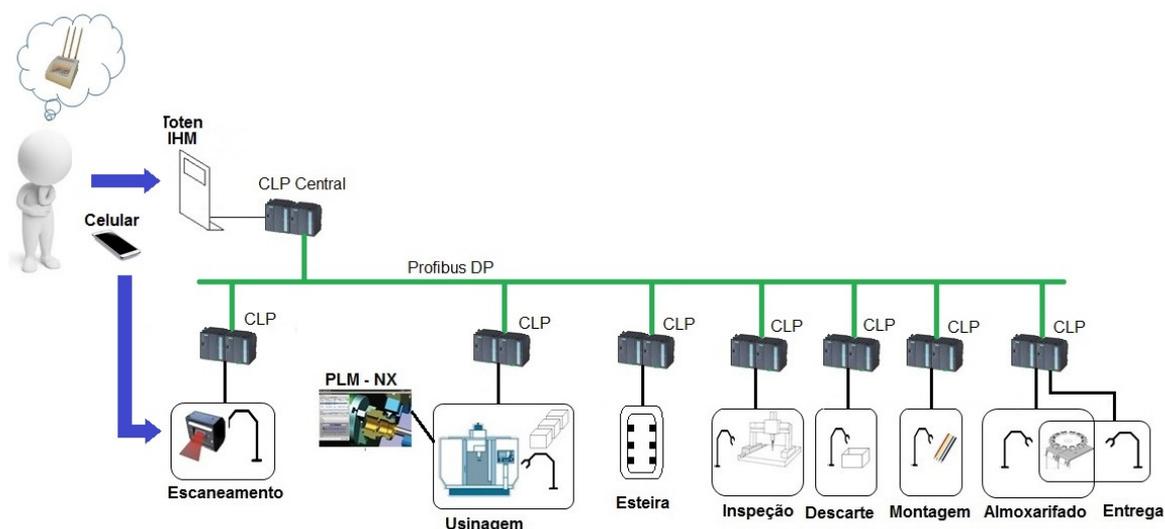
Assim sendo, o MES consegue dar conta de manter a produção com a mudança nos requisitos, manter a produção com inserção de um único produto em meio a um lote e de mudar a ordem de processo em função da não disponibilidade do recurso. Nas demais métricas o MES não se mostrou adequado para dar conta da necessidade.

## 6.3 PLANTA COM GESTÃO POR CIM

### 6.3.1 Cenário na gestão por CIM

Tomando como base a mesma planta que fora ensaiada no item 6.2 e o mesmo produto a ser confeccionado, se pode modelar a estrutura de gestão a partir do conceito de Manufatura Integrada por Computador (CIM), através da programação em controladores lógicos programáveis, o que é muito comum nas indústrias como forma de automação do processo fabril.

Para esta análise, se podem programar os controladores lógicos, mas a operação da máquina propriamente dita será simulada, o que não é prejuízo à análise, pois o foco é na forma com que o sistema produtivo se organiza e não no tempo que demanda para produzir a peça. Portanto, as entradas e saídas de cada CLP em relação ao seu hardware foram simuladas no controlador lógico programável. A Figura 80 apresenta o diagrama de conexão entre os controladores lógicos programáveis que compõe o Sistema Integrado de Manufatura (CIM).

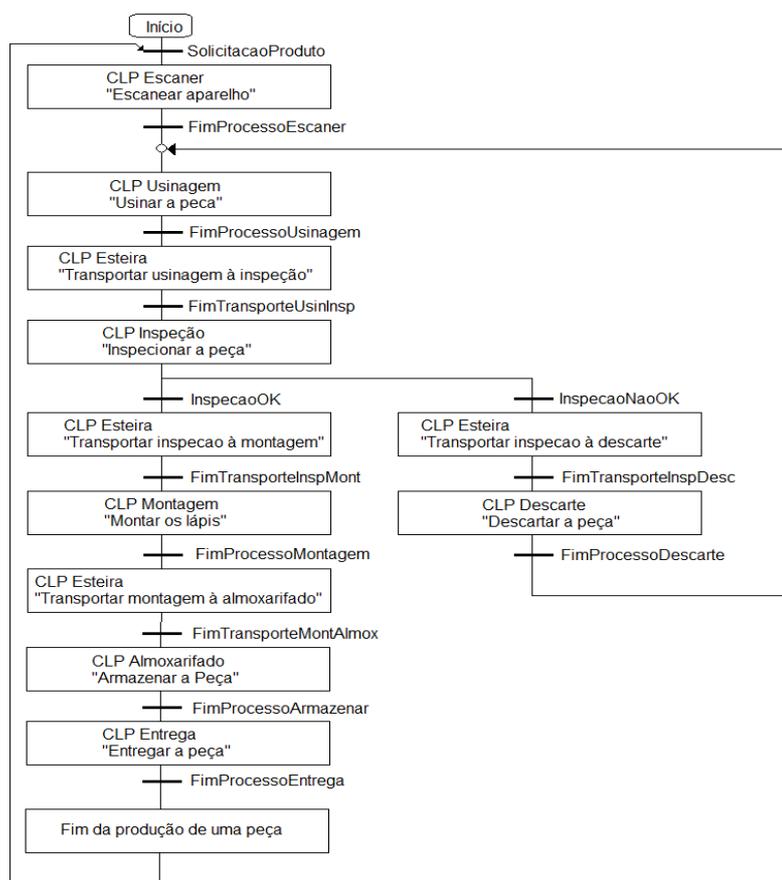


**Figura 80 Hierarquia da comunicação e interação com CIM.**

Fonte: autor.

Há um CLP central que se comunica com uma Interface Humano Máquina (IHM) para interação com o usuário, que através de telas programadas, insere os fatores de customização

do produto, neste caso a escolha das cores dos lápis e a inserção dos caracteres que devem ser impressos. Este CLP central possui um programa para comandar a sequência com que os demais Controladores lógicos programáveis do sistema serão acionados. Os Controladores lógicos programáveis de cada estação possuem uma programação local para gerenciar a sua manufatura, comandando seus periféricos através de entradas e saídas. Estes Controladores lógicos programáveis são acionados pelo CLP central através de uma rede industrial Profibus DP, que permite que variáveis sejam compartilhadas entre os Controladores lógicos programáveis ligados a esta rede. A Figura 81 apresenta o programa em linguagem Grafcet<sup>18</sup> do CLP central, descrevendo a sequência de produção.



**Figura 81 Programação em blocos do CLP.**

Fonte: autor.

<sup>18</sup> O diagrama funcional Grafcet permite descrever os comportamentos de um automatismo em função das informações que recebe. O Grafcet não pretende minimizar as funções lógicas que representam a dinâmica do sistema, pelo contrário o seu potencial reside na imposição de um funcionamento rigoroso, evitando desta forma incoerências, bloqueios ou conflitos durante o funcionamento do mesmo.

O processo se inicia com a solicitação de um produto customizado pelo usuário, onde o CLP central corresponde ao acionamento da *flag* “SolicitacaoProduto”. Neste momento, já se tem as informações do conjunto de caracteres e da escolhas de cores de canetas como parâmetros. Agora o CLP central aciona o CLP Escaner para proceder a leitura dimensional do aparelho celular, com interesse na largura e espessura, que são dois parâmetros repassados ao programa. Uma vez encerrado este processo, *flag* “FimProcessoEscaner” acionada, o CLP central aciona o CLP Usinagem para proceder a usinagem da peça segundo um modelo pré-definido e parâmetros customizados. O programa CNC para descrever as trajetórias é editado por um *software* vinculado à máquina de usinagem para operar a manufatura. Encerrado a usinagem então o CLP central aciona o CLP esteira para executar o transporte da estação de usinagem até a estação de inspeção. Uma vez acionado o CLP inspeção, a peça é medida em suas coordenadas para garantir o controle dimensional do produto. Neste passo há dois fluxos a serem seguidos, segundo resultado da inspeção. Se ocorrer uma “InspecaoNaoOK”, significa que a peça deve ser descartada, e isto ocorre com o CLP central solicitando ao CLP esteira para transportar a peça até a posição de descarte e na sequência solicitar ao CLP descarte para que proceda a exclusão desta peça considerada não conforme, retornando então o fluxo para posição anterior a usinagem a fim de produzir uma outra peça, com as mesmas características customizadas pelo usuário. Se ocorrer uma “InspecaoOK”, significa que a peça foi aprovada no controle de qualidade, devendo o CLP central acionar o CLP esteira pra transportar a peça da estação de inspeção até a estação de montagem. Nesta estação o CLP montagem, acionado pelo CLP central, procede então a montagem dos lápis segundo as cores escolhidas pelo cliente no momento de sua solicitação. Após este passo o CLP central solicita ao CLP esteira para transportar a peça até a estação de armazenamento, quando então o CLP almoxarifado realiza o armazenamento da peça em uma posição escolhida e informada ao programa. Quando o usuário recorre ao totem para solicitar a entrega da peça, o CLP Central aciona o CLP entrega

que faz a busca da peça em questão no almoxarifado e a entrega ao cliente. O programa volta ao início e fica aguardando uma nova solicitação de produto.

Para ocorrer a produção simultânea de mais de um produto, o CLP central deve ser programado com dois ou mais programas similares a este que fora relatado, sendo executados concomitantemente. A quantidade de programas concomitantes indica a quantidade de peças que podem ser produzidas ao mesmo tempo no sistema, resguardada a disponibilidade de recursos de produção.

### 6.3.2 Avaliação na gestão por CIM

Utilizando as mesmas métricas, aqui já definidas, se procede então a análise deste sistema de manufatura, a fim de comparar com os demais sistemas alvos deste trabalho. O resultado desta análise se encontra na Tabela 15 a seguir.

**Tabela 15 Análise da planta de Manufatura Avançada com gestão por CIM.**

<b>Métricas</b>	<b>Planta demonstrativa de Manufatura Avançada gerenciada por CIM</b>
Manter a produção com a mudança nos requisitos.	Uma vez que os programas das estações já estão prevendo que podem ter requisitos distintos de uma peça para outra, este requisito é então contemplado, pois o CLP central passa o requisito customizado para a manufatura em questão.
Manter a produção com inserção de um único produto em meio a um lote.	Uma vez que os programas das estações já estão prevendo que podem ter requisitos distintos de uma peça para outra, a inserção de um lote único não é problema, pois dada peça já é interpretada como um lote único.
Manter a produção com a retirada de estações.	A retirada de uma estação implica na parada do sistema e reprogramação do CLP central, o que não condiz com a necessidade desta métrica.
Mudar a ordem de processo em função da não disponibilidade do recurso.	A mudança da ordem de processo não é possível, pois já está definida no programa do CLP central. Uma mudança nesta ordem significa necessidade de reprogramar o CLP.
Manter a produção com a inserção de estações.	A inserção de uma estação implica na parada do sistema e reprogramação do CLP central, o que não condiz com a necessidade desta métrica.
Manter a produção mesmo com a mudança da estação de posição física na planta.	A mudança de uma estação de posição no sistema implica na necessidade de se parar o sistema e reprogramar o CLP central para indicar a nova posição dos periféricos.

Fonte: Autor.

A partir destas métricas se conclui que um Sistema Integrado de Manufatura (CIM) em alguns momentos atende aos requisitos de diversidade e suporte a lotes pequenos de produção, quando bem programado e bem implementado. Mas nos quesitos que requerem a inserção,

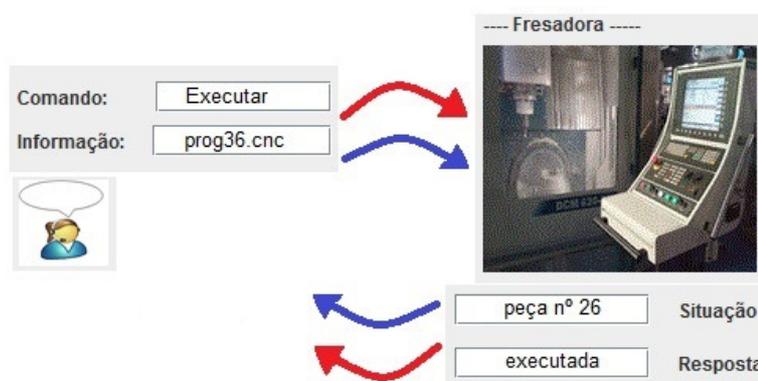
retirada, mudança de ordem no fluxo ou mudança de leiaute na planta o CIM deixa a desejar. Não atende a contento.

#### 6.4 PLANTA COM GESTÃO POR SISTEMAS AUTO-ORGANIZÁVEIS

Nesta implementação o objetivo é realizar a produção de um produto customizado de um lote único, sendo que as máquinas utilizadas como recursos se integram em um sistema de manufatura, dispondo seus serviços e negociando com a peça sua alocação à atividade. A peça é um agente do sistema que detém a sequência de tarefas a serem executadas e propõe a cada tarefa uma tratativa para alocação de recursos com as máquinas para sua alocação ao serviço.

##### 6.4.1 Cenário na gestão por sistemas auto-organizáveis

Neste cenário, com a proposta de implementar a solução para produção de um produto customizado e lote pequeno de produção, especificamente lote único, foi então abordado um cenário em que cada estação de manufatura seja uma máquina dotada de uma lógica local, autônoma, que aceita e responde a comando, recebendo informações e informando seu estado de produção. A Figura 82 representa esta comunicação entre o “agente máquina” e o agente que simula a máquina.



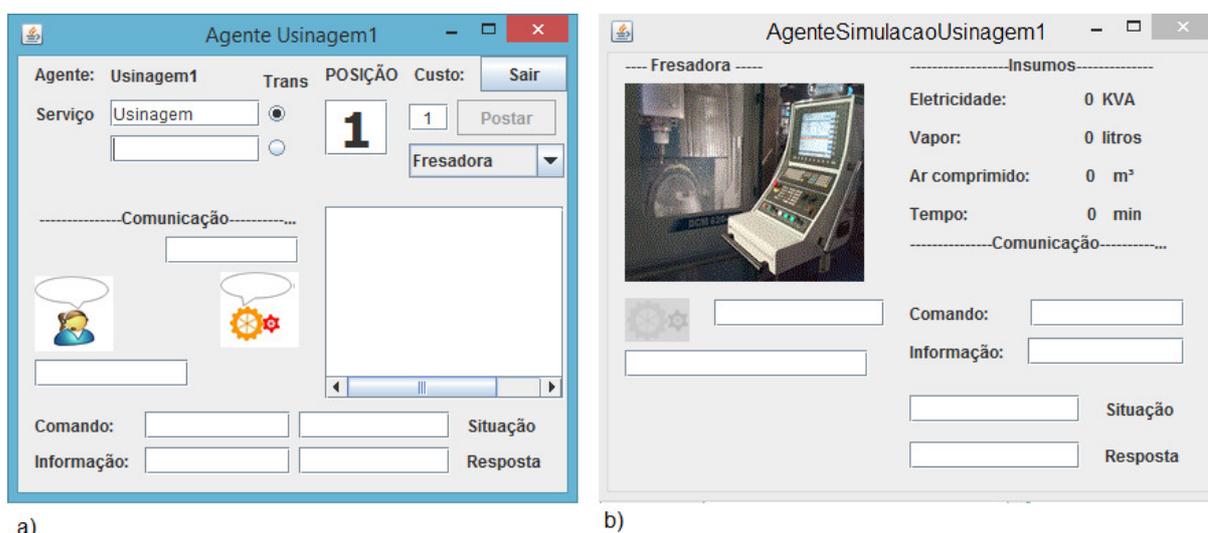
**Figura 82 Exemplo de comunicação entre o agente e a máquina.**

Fonte: Autor.

Neste modelo o agente envia a “Informação” para máquina que diz respeito a indicação do arquivo a ser executado e envia o “Comando” para executar. A máquina executa o programa indicado e responde com o “Estado”, indicando que o número desta peça agora é identificada como número 26, e envia a “Resposta” indicando que a peça foi executada. São as quatro mensagens que indicam uma comunicação entre um operador e máquina, neste caso entre o “agente peça” e a máquina.

O sistema completo de manufatura compreende então um ambiente com até 9 alocações de máquinas de manufatura, 1 estação de transporte com capacidade de interfacear até 6 máquinas como meio de transporte e 7 slots para produção de peças.

Como o objetivo é testar a comunicação entre as estações de trabalho visando sua auto-organização, não foi implementada a máquina fisicamente, mas sim um aplicativo que simula a execução da máquina. Assim, há o elemento computacional que representa o agente da máquina e outro elemento computacional que simula sua execução. A Figura 83 representa este modelo de “agente máquina” e o agente simulador da máquina.



**Figura 83 a) agente de comunicação da máquina; b) agente que simula a máquina.**

Fonte: autor

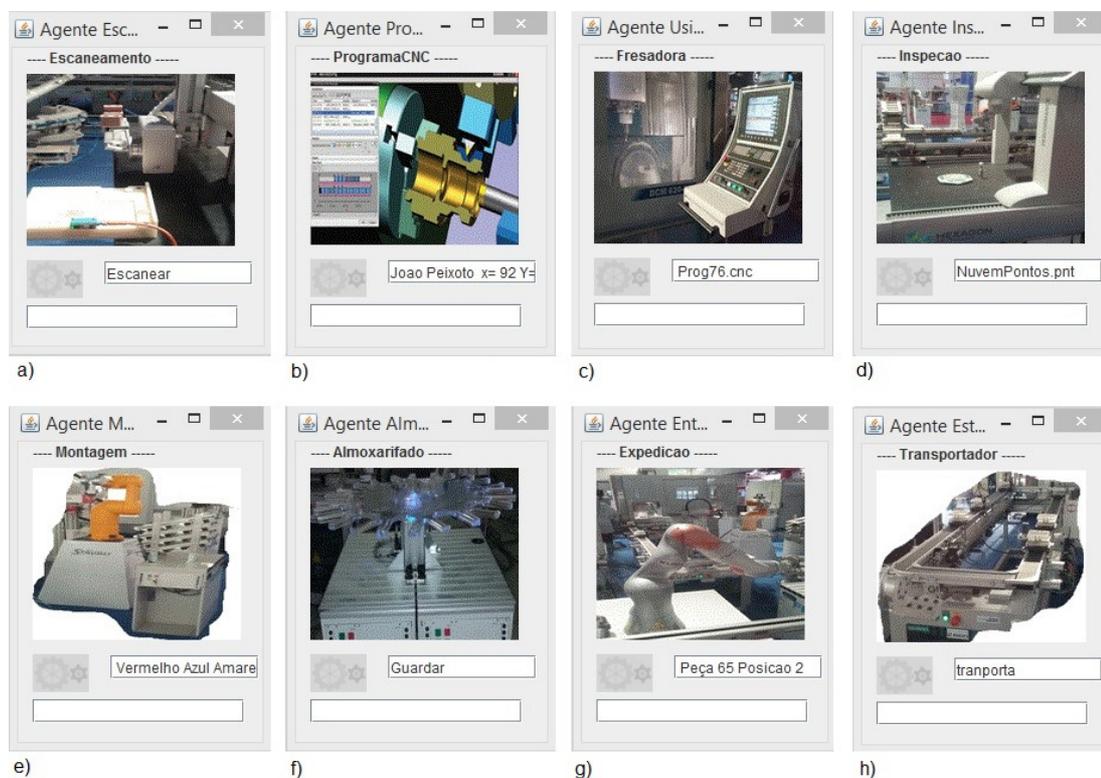
Na Figura 83 está representado o agente que simula a máquina, que recebe comando e informação do agente de comunicação da máquina e responde com situação e resposta. Dentro

de seu modelo de simulação, o agente de simulação da máquina também dá conta de mensurar a energia demandada em eletricidade, ar comprimido e vapor para a operação exigida pela peça, bem como o tempo de produção, permitindo à peça totalizar estes insumos ao final de sua produção.

As operações que o produto demanda são:

- a) Digitalização: para escanear o aparelho celular do cliente e definir sua largura e profundidade, a fim de servir como parâmetro para usinagem do produto customizado às dimensões do celular do cliente;
- b) Programação: para elaborar um programa CNC para usinagem do produto conforme as dimensões do celular do cliente definidas na estação de digitalização e conforme os caracteres definidos como mensagem na parametrização inicial realizada pelo cliente;
- c) Usinagem: para usinar a peça segundo o programa CNC definido na etapa de programação;
- d) Inspeção: para inspecionar as dimensões da peça usinada, tendo como referência o dimensional do produto definido na parametrização do cliente;
- e) Montagem: para realizar a montagem dos três lápis por um manipulador robótico, cuja sequência de cores foi definida pelo cliente ao solicitar o produto;
- f) Armazenamento: para armazenar a peça montada a fim de esperar a vinda do cliente para buscar o seu produto final, realizado por um manipulador robótico e um almoxarifado rotativo, aqui representado por um aplicativo de software;
- g) Entrega: para entregar o produto ao cliente, quando este solicitar, valendo-se de uma interação com a estação de armazenamento.

A Figura 84 apresenta as estações citadas, que irão simular as operações destas máquinas.

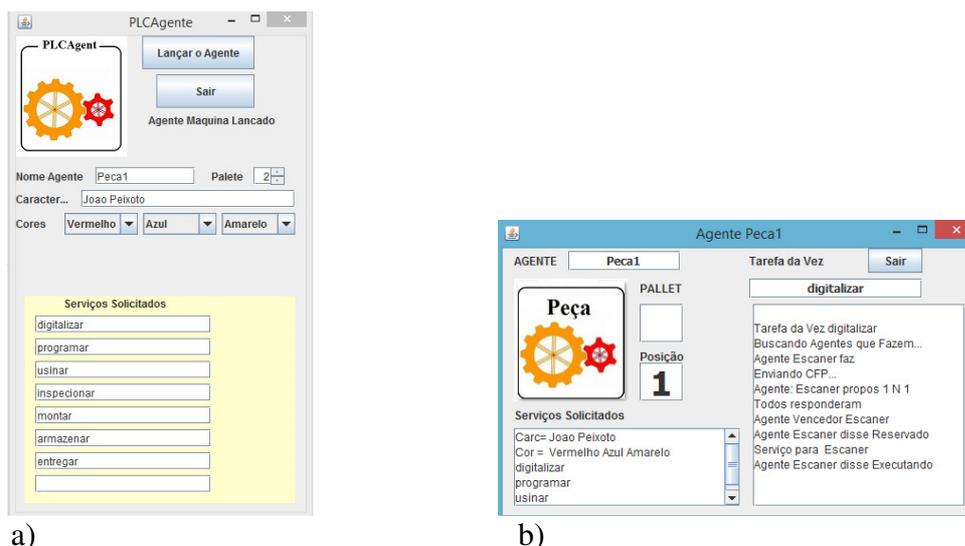


**Figura 84 Agente Máquina. a) Escaneamento, b) Programação, c) Usinagem, d) Inspeção, e) Montagem, f) Armazenamento, g) Entrega, f) Transporte.**

Fonte: Autor.

Nesta figura há então, além dos agentes máquina que serão demandados pela peça, o agente de “Descarte” que é responsável pelo descarte da peça em caso de reprovação no controle dimensional realizado pela estação de “Inspeção”. Há também o agente de “Transporte”, responsável pelo transporte das peças de uma estação para outra.

Neste cenário, a ordem com que os serviços serão demandados é função da peça, que é modelada como um agente, que irá procurar os serviços junto aos demais agentes. A Figura 85 apresenta o agente lançador de peças e o próprio agente peça em execução.

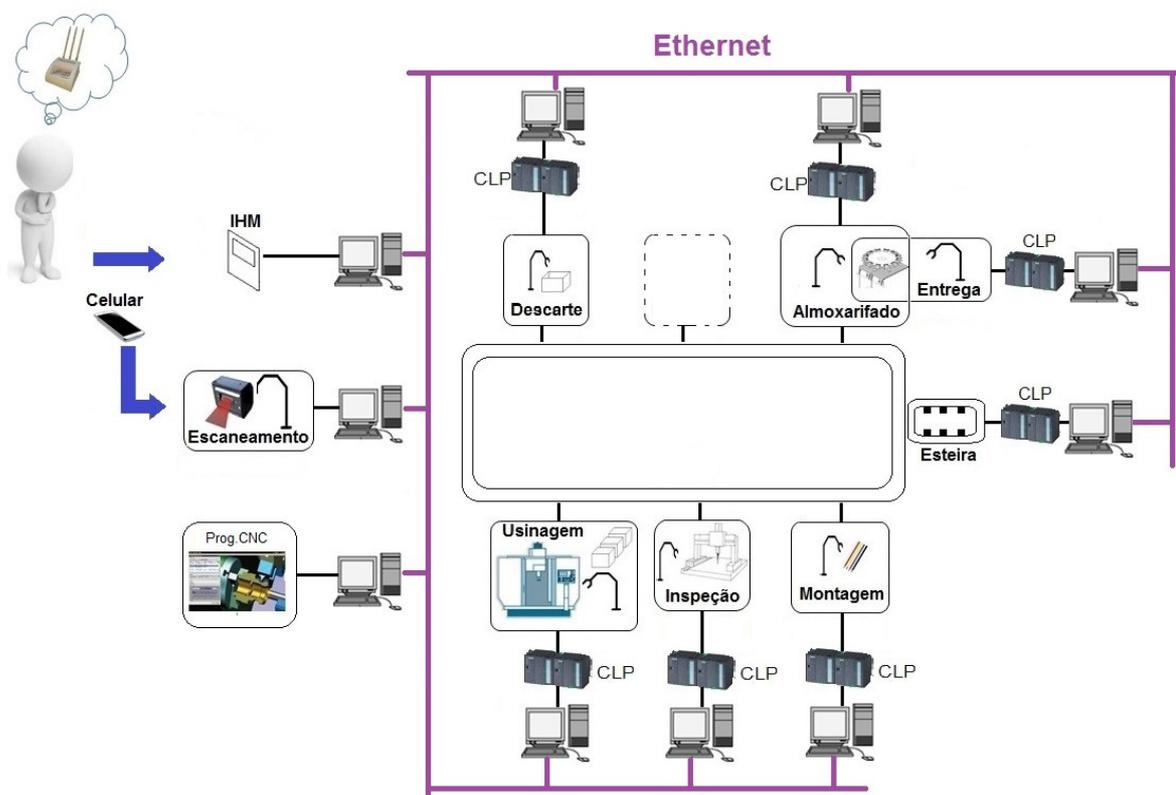


**Figura 85 PLCAgente, executa o lançamento dos agentes Peças. a) PLCAgente lançador, b) agente peça em execução.**

Fonte: Autor

A estrutura dos agentes para compor o cenário de produção do produto exigido com seu grau de customização e lote único compreende os agentes interligados pela plataforma JADE, utilizando o meio físico *ethernet*, onde cada agente máquina irá postar seus serviços na plataforma JADE e disponibilizá-lo para alocação e execução. A solicitação de peça, ao ser executada, instiga o usuário a inserir os caracteres que deseja imprimir na peça, a escolher a sequência de cores dos lápis a serem montados. Nesta peça são definidas então as operações que se tenciona realizar, na sequência desejada. A cada tarefa que a peça solicita ocorre uma tratativa para alocação de recursos com as máquinas para alocar a melhor delas para executar o serviço. Se esta máquina escolhida requer transporte da peça, então a máquina fica reservada e passa a ocorrer uma tratativa para alocação de recursos com os sistemas de transporte para providenciar o deslocamento da peça, para então voltar a uma requisição à máquina selecionada para que execute a operação desejada. Este processo se dá a cada serviço necessário para a produção da peça, permitindo assim que mais estações participem das tratativas, bastando sua entrada na plataforma. Assim como permite que estações saiam da plataforma, passando a não participar da próxima tratativa para alocação de recurso.

A Figura 86 mostra a hierarquia de comunicação com um sistema multiagente, sem um mestre central ou um escalonador, onde todos os agentes (máquina, transporte ou peça) interagem entre si para que a melhor otimização da produção ocorra.



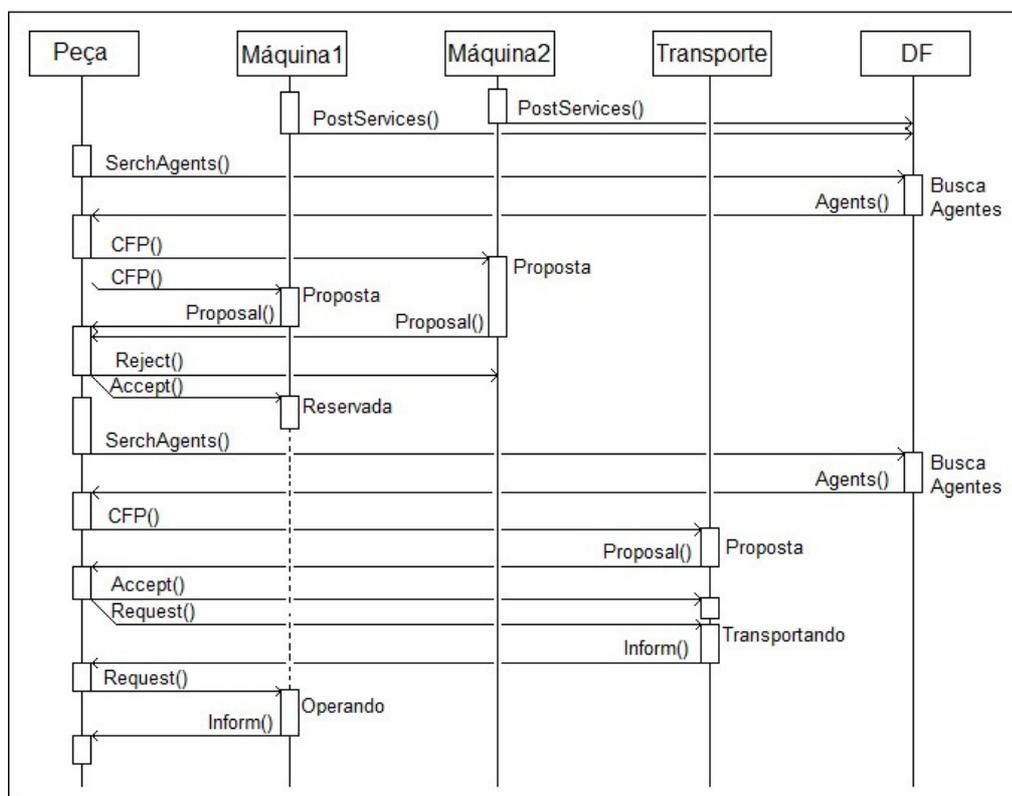
**Figura 86 Hierarquia da comunicação e interação com Sistemas Auto-organizáveis.**

Fonte: Autor

A lembrar que neste cenário as máquinas em si e o sistema de transporte são simulados por aplicativos em software, pois o objetivo aqui é verificar a interação entre as máquinas, transporte e peças, desprezando os aspectos de produção da peça, como: parâmetros de usinagem, tempo de transporte, método de escaneamento, entre outros.

Nesta aplicação houveram 8 agentes máquina, 1 agente de transporte e mais os agentes peças que foram sendo solicitados como requisito de produção. Com uma solicitação de 6 peças se chega ao número de 15 agentes sendo executados na plataforma JADE. E aqui também vale o adendo de que o objetivo era verificar a interação entre as máquinas, desconsiderando os aspectos de tempo de comunicação entre os agentes na plataforma.

Analisando a Figura 87 que transcreve um diagrama de sequência de comunicação da peça com uma máquina e sistema de transporte pode-se entender a forma com que a interação ocorre.



**Figura 87 Diagrama de Sequência com a interação entre peça, máquina e sistema de transporte.**

Fonte: Autor.

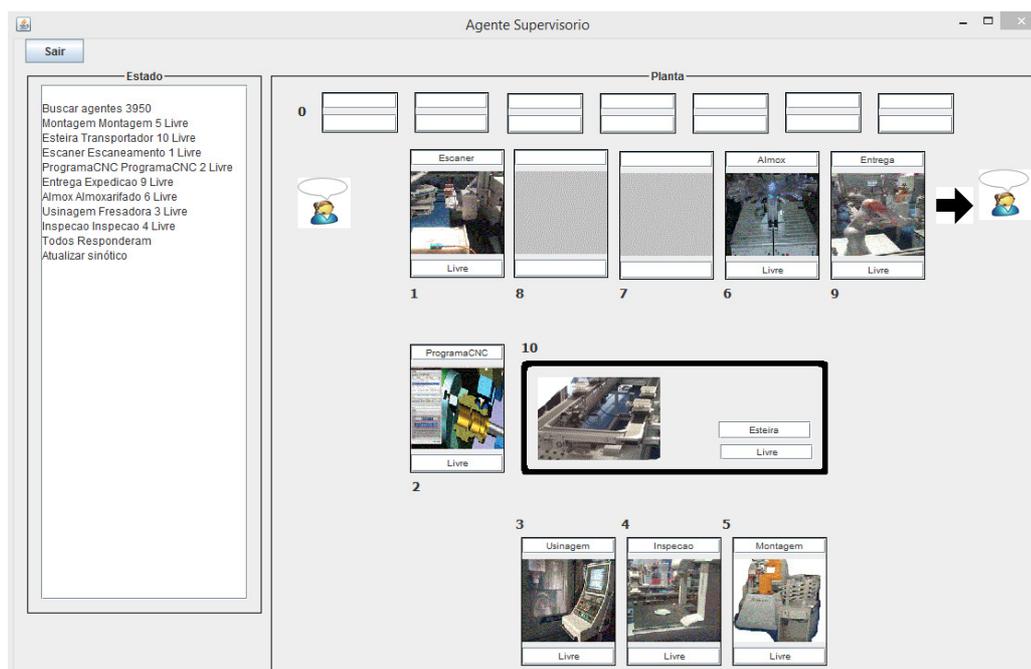
Cada máquina que entra na plataforma postas seus serviços no agente DF (*Directory Facilitator*). O primeiro passo da peça para executar seu primeiro serviço é buscar no DF os agentes que oferecem o serviço desejado, através de *SearchAgents()*. O DF responde com os agentes que realizam este serviço desejado. Então a peça envia um CFP(*Call For Propose*) aos agentes indicados. Os agentes que recebem o CFP indicam suas propostas através de um *Proposal()*. A peça avalia as propostas e responde com um *Accept()* ou *Reject()* aos agentes que propuseram. O agente máquina que foi aceito entra em estado de reserva, pois o agente peça agora deve buscar no DF os agentes que fazem o transporte de sua posição atual para posição do agente máquina escolhido, deve enviar uma solicitação de proposta CFP() e receber propostas dos agentes de transporte, quando então envia um *Accept()* e um *Request()* a fim de

que o transporte ocorra. Ao término do transporte, este agente envia um *Inform()* para indicar seu término de atividade. Neste instante o agente peça envia um *Request()* ao agente máquina que estava reservado para que o processo de execução do serviço então ocorra. Ao final da execução do serviço o agente máquina envia um *Inform()* para indicar o final do processo e a peça tem então seu serviço requisitado terminado, podendo agora buscar um novo serviço nos seus requisitos e repetir o processo novamente.

Cabe salientar que, em alguns serviços que a peça solicita pode não ocorrer a necessidade de transporte. Neste caso, esta etapa de busca e solicitação de transporte não é realizada.

#### **6.4.2 Avaliação na gestão por sistemas auto-organizáveis**

Para melhor avaliação deste cenário, em que fora simulado em computadores os aspectos de usinagem, mas que eram executados em computadores distintos, fora implementado um agente supervisorio, que, ao entrar na plataforma JADE, busca todos agentes que nela se encontravam, envia uma solicitação de estado de operação a cada um deles, e à medida que recebe as respostas, plota em um ambiente gráfico a posição e estado de cada agente, seja ele peça, máquina ou transporte. A Figura 88 apresenta o ambiente gráfico deste agente supervisorio.



**Figura 88 Agente com a função de sistema supervisorio.**

Fonte: Autor.

Neste ambiente se consegue perceber a entrada de agentes na estação de manufatura, a saída destes agentes, seu deslocamento de posição dentro da plataforma, a inserção de duplicidade de recursos e o fluxo de processos que a peça está trilhando.

Analisando os fatores que servem como referência para este estudo, tem-se a Tabela 16 que indica as métricas analisadas para este cenário.

**Tabela 16 Análise da planta de Manufatura Avançada com gestão por Sistemas Auto-organizáveis.**

Métricas	Planta demonstrativa de Manufatura Avançada gerenciada por Sistemas Auto-organizáveis (self-x)
Manter a produção com a mudança nos requisitos.	A mudança dos requisitos a serem atendidos para o produto é definida através da sequência de serviços listados na própria peça, portanto, sua mudança não implica na alteração da programação das máquinas e mantém o processo produtivo, mesmo esta mudança ocorrendo de uma peça para outra no mesmo lote de produção.
Manter a produção com inserção de um único produto em meio a um lote.	A inserção de uma peça com requisitos diferenciados das demais terá um tratamento segundo sua sequência de serviços que nela está listado, sem interferir nos processos das demais peças do lote.
Manter a produção com a retirada de estações.	Como a peça faz a tratativa para alocação de serviços a cada serviço necessário, a saída de estação implica que esta não participará da próxima tratativa de alocação de serviço, mas não altera o fluxo produtivo nem requer reprogramação das demais estações ou de um sistema de gerenciamento central.
Mudar a ordem de processo em função da não disponibilidade do recurso.	A ordem e a concomitância dos serviços necessários, bem como suas procedências estão descritos na peça, assim, a indisponibilidade de um recurso exige a que a peça busque a execução de outro serviço

	que não seja precedente, mudando a ordem do processo, sem necessidade de reprogramação dos dispositivos.
Manter a produção com a inserção de estações.	A inserção de novas estações faz com que mais recursos sejam disponíveis para negociar serviços com a peça, não requerendo reprogramação de dispositivos, mantendo o fluxo de produção.
Manter a produção mesmo com a mudança da estação de posição física na planta.	No momento da postagem dos serviços no DF pelo agente máquina sua posição na planta é lá postada, assim, a mudança de posição na planta por uma máquina significa refazer a postagem e indicar sua nova posição.

Fonte: Autor.

Nesta avaliação, se percebe que o atendimento em 100% das métricas definidas para avaliar o atendimento do sistema produtivo aos requisitos de diversidade, agilidade e auto-organização, e que a metodologia proposta fornece aos controladores das máquinas, com suas lógicas locais, a capacidade de interação e participação de um sistema auto-organizável, valendo-se de sistemas Multiagentes, como retrata o tema da tese.

## 6.5 PLANTA COM GESTÃO POR SISTEMAS AUTO-ORGANIZÁVEIS, SEGUNDO CENÁRIO EXPERIMENTADO

Como segundo cenário implementado foi montado uma planta de manufatura industrial a ser exposta no evento Feira de Subcontratação e Inovação Industrial – MERCOPAR, realizada em outubro de 2016 na cidade de Caxias do Sul - Brasil. A Figura 89 mostra a planta implementada dentro do espaço Salão de Inovação, patrocinado pelo SEBRAE-RS.



**Figura 89** Planta de manufatura industrial exposta na MERCOPAR.

Fonte: Autor

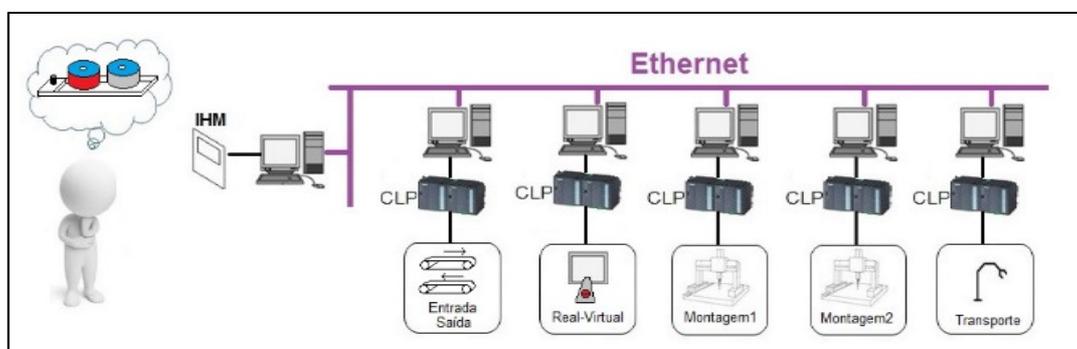
Esta planta se propôs a executar a montagem de microcilindros, compostos de base, êmbolo, mola e tampa. O usuário ao solicitar a produção tem a possibilidade de escolher duas combinações de cilindros que deseja ser montado entre três opções distintas. Há o cilindro cor prata, o cilindro cor preta e o cilindro cor vermelha. Portanto, trata-se de uma produção diversificada, pois o mesmo sistema produtivo irá atender a produção de três tipos de peças distintas.

### 6.5.1 Cenário na gestão por sistemas auto-organizáveis, segundo experimento.

Neste cenário fora aplicado a metodologia proposta, o que resultou na implementação de um agente para interagir com cada controlador lógico programável, segundo sua característica. Haviam na planta os seguintes agentes:

- a) Agente máquina de montagem de base preta e base prata;
- b) Agente máquina de montagem de base preta e base vermelha;
- c) Agente máquina de montagem virtual de êmbolo, mola e tampa;
- d) Agente de transporte entre as cinco posições;
- e) Agente máquina de entrada e saída de produtos;
- f) Agentes peças, lançados pelo usuário por interface IHM.

A Figura 90 apresenta a hierarquia da comunicação entre os agentes neste sistema auto-organizável.



**Figura 90 Hierarquia da comunicação e interação com Sistemas Auto-organizáveis no segundo cenário de ensaio.**

Fonte: Autor

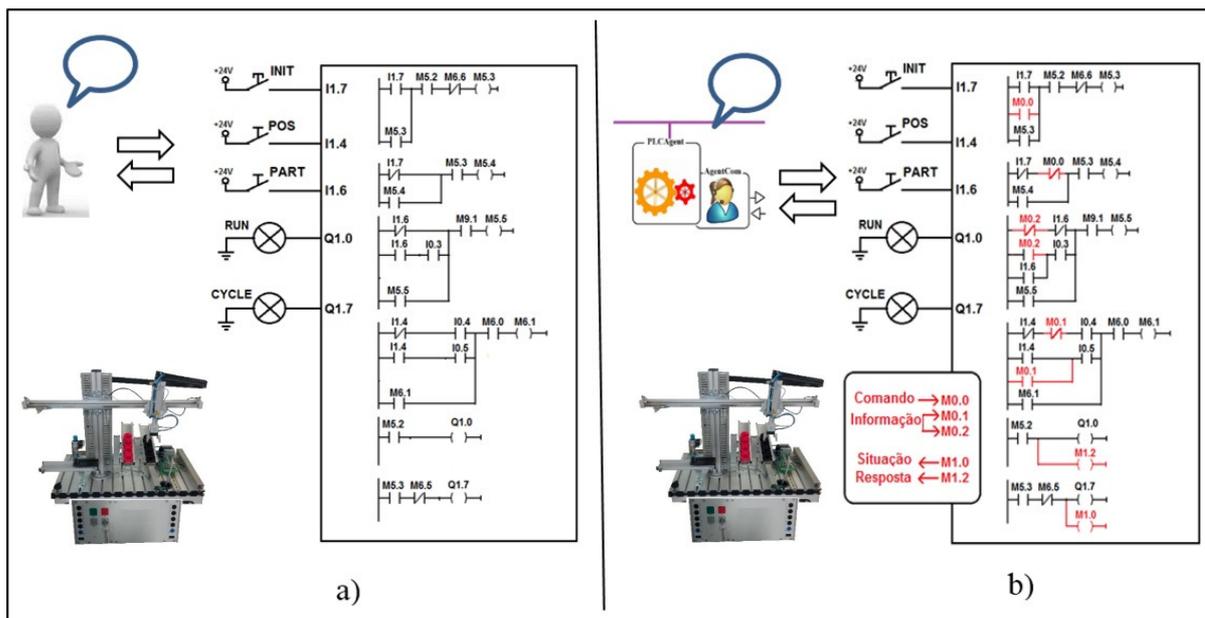
Uma vez que o usuário solicitava um produto com as características de seu desejo, o produto parte de seu primeiro serviço demandado e começa uma tratativa para alocação de serviços com as máquinas que possuem esta capacidade. Uma vez escolhida a máquina que realizará o serviço, então ocorre um reserva desta máquina, para então começar uma tratativa para alocação de serviços com o agente de transporte, de forma a levar a peça até a posição da máquina reservada. Realizada esta tarefa, o produto volta a interagir com a máquina reservada para provocar a execução do serviço solicitado.

Isto acontece para cada serviço demandado pelo produto e na ordem com que eles são solicitados. Este fato permite que a cada início de tratativa de alocação de serviços um agente máquina se insira ou saia do ambiente de manufatura, evidenciando assim a auto-organização do sistema produtivo.

A interação entre o agente e o controlador lógico programável se dá através da classe AgentCom, implantada junto ao PLCAgent, e que trata da especificidade da interação e na forma com que a interação ocorrerá. Nesta aplicação se traz um exemplo de como a interação ocorre de forma minimamente invasiva.

O princípio da interação é fazer com que o Agente acesse as áreas de memória dentro do CLP que tenham relação com os sinais de comando, informação, situação e resposta. Desta forma, a lógica de programação local que gerencia os periféricos da máquina não é alterada, mas passa a receber e informar sinais como se estivesse interagindo com o operador, mas na realidade é um agente que está provocando esta interação.

A Figura 91 a seguir apresenta a diferença entre os programas, antes e após a interação com o agente, para uma máquina considerada de baixa complexidade, devido a ter 8 sinais de entradas para serem monitorados e 7 sinais de saída para comandar.



**Figura 91 Diagrama de contatos da lógica local que comanda a máquina. a) sem a interferência do Agente; b) com a interferência do agente.**

Fonte: Autor.

Em “a” está representado a lógica de programa por diagrama de contatos que possuem relação direta com os sinais de comando, informação, situação e resposta. As demais lógicas que comandam este equipamento não estão representadas neste diagrama. Em “b” há a representação da lógica de programa com a inserção das memórias que fornecerão os sinais ao agente, de forma a simular o controle do operador sobre a máquina. Este acesso se dá através da conexão por rede industrial, item descrito na metodologia.

### **6.5.2 Avaliação na gestão por sistemas auto-organizáveis no segundo cenário experimentalado.**

Avaliando os resultados desta implementação, pode-se se observar o aspecto de a metodologia ser minimamente invasiva. A Tabela 17 traz um comparativo de área de memória ocupada pelo controlador lógico programável para o programa da lógica local original da máquina e para esta mesma lógica com a interação com o agente.

**Tabela 17 Memória ocupada no controlador lógico programável considerando o programa da lógica local.**

Memória ocupada	Programa da lógica local original da máquina	Programa da lógica local com a interação do agente	Acréscimo percentual
Memória de Dados	26 bytes	26 bytes	0%
Códigos	644 bytes	700 bytes	8,70%
Memória local	796 bytes	852 bytes	7,04%
Memória de trabalho	680 bytes	736 bytes	8,24%
TOTAL	2.146 bytes	2.314 bytes	7,83%

Fonte: Aplicativo TIA Portal SIEMENS monitorando a execução do programa.

Para esta aplicação em questão se observa que para provocar a interação entre o agente e o controlador lógico programável, o acréscimo de memória ocupada foi de 7,83%. Considerando que se trata de uma aplicação onde há 8 entradas e 7 saídas, algo bem sucinto em termos de aplicação industrial. Se avaliar esta aplicação com mais complexidade o que ocorrerá é que haverá mais lógicas de monitoramento e controle, porém as lógicas que tratam diretamente dos sinais de comando, informação, situação e resposta serão as mesmas. Assim, o número absoluto de posições de memórias acrescido continuará o mesmo, mas a quantidade total de memória ocupada será maior. Portanto, o percentual de memória ocupada (ocupação relativa) tende a ser menor.

Assim sendo, pode-se afirmar categoricamente que na aplicação da metodologia a interferência no programa da lógica local do controlador lógico programável será de no máximo 8%. Isto comprova a característica da proposta ser minimamente invasiva.

Com relação ao atendimento das métricas estipuladas para o ensaio, se verificou o comportamento do sistema auto-organizável com a aplicação de cada uma delas e registrou-se sua reação. A Tabela 18 apresenta o resultado deste ensaio.

**Tabela 18 Análise da planta de Manufatura Avançada com gestão por Sistemas Auto-organizáveis.**

<b>Métricas</b>	<b>Planta demonstrativa de Manufatura Avançada gerenciada por Sistemas Auto-organizáveis (self-x) no segundo cenário implementado.</b>
Manter a produção com a mudança nos requisitos.	Novos requisitos passados por parâmetros ao agente, sem necessidade de reprogramação ou parada da produção.
Manter a produção com inserção de um único produto em meio a um lote.	Novos requisitos passados por parâmetros ao agente, sem necessidade de reprogramação ou parada da produção.
Manter a produção com a retirada de estações.	A estação que fica, com mesma funcionalidade, assume a tarefa e a produção não para.
Mudar a ordem de processo em função da não disponibilidade do recurso.	A falta de um recurso faz com que a peça negocie outra tarefa não precedente, se auto-organizando e mantendo o fluxo produtivo.
Manter a produção com a inserção de estações.	A estação que entra no sistema passa a participar das negociações de tarefas, sem necessidade de reprogramação ou parada da produção.
Manter a produção mesmo com a mudança da estação de posição física na planta.	A estação vincula sua posição no transportador através de parâmetro, quando da sua instância, evitando a necessidade de reprogramação ou parada da produção.

Fonte: Autor.

Nesta avaliação, se percebe que o atendimento em 100% das métricas definidas para avaliar o atendimento do sistema produtivo aos requisitos de diversidade, agilidade e auto-organização, e que a metodologia proposta fornece aos controladores das máquinas, com suas lógicas locais, a capacidade de interação e participação de um sistema auto-organizável.

## 6.6 AVALIAÇÃO DOS RESULTADOS DA APLICAÇÃO DO MÉTODO

Para efetiva comparação entre as três abordagens foram aplicadas condições semelhantes, a fim de observar resultados a partir das mesmas premissas, sabendo previamente quais resultados se esperariam e quais realmente se comprovaram.

As condições aplicadas e seus resultados foram:

Manter a produção com a mudança nos requisitos: para avaliar esta métrica é solicitada a produção de um tipo de produto e depois se faz a mudança nos seus requisitos, acompanhando seus efeitos no meio produtivo. O esperado que a produção não pare ou seja desligada para reprogramação.

No Sistema Integrado de Manufatura, a mudança de requisitos do produto requereu a reprogramação do CLP que gerencia o fluxo produtivo, para que nele fosse colocada a nova sequência de produção. Necessidade de uma nova compilação do programa e *download* para o CLP. Este processo necessitou a parada e desligamento do sistema produtivo para instalação do novo programa.

No Sistema de Gestão com MES a mudança de requisitos implicou na alteração no MES da sequência de produção, não sendo necessário o desligamento de *download* de programas, mas a reprogramação do sistema de gestão central.

No sistema de manufatura Auto-organizável (Self-x) os novos requisitos foram passados para o “agente peça” através de parâmetros e este passou a negociar o novo fluxo produtivo com os demais agentes do sistema. Não houve necessidade de desligamento do sistema nem a parada do mesmo. Também não houve necessidade de reprogramação dos controladores lógicos programáveis, assim, não houve necessidade de conhecimentos e habilidades em programação de controladores lógicos ao operador do processo.

Manter a produção com inserção de um único produto em meio a um lote: nesta métrica é então solicitada uma produção de produtos idênticos simulando um lote de produção e, em meio a este lote, se solicita a produção de um único produto com especificações diferentes. O que se espera é que não ocorra a parada do sistema produtivo, nem sua necessidade de reprogramação para aceitar a inserção deste produto distinto.

No Sistema Integrado de Manufatura, o desenvolvimento de um programa para o CLP que atendesse as lógicas para este produto único evitou a reprogramação, não havendo a necessidade de parar o sistema e proceder o *download* deste programa para execução do lote único.

No Sistema de Gestão com MES é implementada em sua programação a situação de lote único, podendo este sistema produzir lotes maiores com as mesmas características e peculiaridades de um lote único de produção.

No Sistema Auto-organizado de manufatura não foi necessário parar o sistema, pois bastou configurar os parâmetros deste lote único na execução do agente, e este já passou a negociar os serviços para execução desta peculiaridade do lote único.

Manter a produção com a retirada de estações: neste caso foram instaladas duas estações com funcionalidades semelhantes, mesmos serviços prestados, e foi solicitado um produto que requer estes serviços. Em meio à produção, uma das estações foi retirada do processo produtivo. O que se espera é que, sendo o sistema modular, com a saída de uma das estações a outra deve assumir a tarefa, admitindo ter a mesma funcionalidade, sem requerer alteração na programação.

No Sistema Integrado de Manufatura a retirada de uma das estações necessitou a parada do sistema e reprogramação do CLP para que o fluxo produtivo se deslocasse para a estação que permaneceu no sistema.

No Sistema de Gestão com MES a retirada da estação implicou na reprogramação do MES para considerar esta ausência de estação.

No Sistema Auto-organizável de manufatura a retirada da estação não implicou na parada do sistema, pois durante as novas negociações a estação que permaneceu passou a absorver estas tarefas.

Mudar a ordem de processo em função da não disponibilidade do recurso: para análise desta métrica é solicitada a produção de dois produtos requerendo serviços que podem ser realizados em concomitância e que não possuam precedência. Após o início do processo fora então retirado um dos recursos comum aos dois produtos. É esperado que a ausência de um

recurso faça com que o produto busque a execução de outros serviços que estejam presentes no sistema e que não haja precedência.

No Sistema Integrado de Manufatura a mudança na ordem de produção só foi possível com a reprogramação das estações de trabalho para considerar a ausência desta estação que fora retirada, a fim de que a ordem do processo fosse modificada.

No Sistema de Gestão com MES a mudança no fluxo do processo com a ausência do recurso já estava concebida na sua programação, podendo ocorrer sem transtornos.

No Sistema Auto-organizável de manufatura a ordem com que os serviços ocorrem é definida na peça, logo é ela quem organiza o próximo serviço a ser realizado, permitindo assim que o sistema mude a ordem de produção conforme a disponibilidade de recursos.

Manter a produção com a inserção de estações: agora o sistema é montado com estações que atendem aos serviços solicitados para confecção de um produto definido. Ao longo da produção mais uma estação idêntica em termos de habilidades é inserida no sistema. O que se espera é que o sistema permita a expansão. A inserção de uma estação não deve requerer alterações na programação, nem demandar uma parada do sistema.

No Sistema Integrado de Manufatura a inserção de uma nova estação requereu a parada do sistema e a reprogramação do programa do CLP, bem como sua nova compilação e novo *download*, a fim de inserir as lógicas desta nova estação.

No Sistema de Gestão por MES a inserção de uma nova estação necessitou que houvesse a reprogramação do MES, a fim de contemplar agora esta nova estação.

No Sistema Auto-organizável de manufatura a inserção da nova estação não necessitou de parada do sistema, pois ao entrar em operação esta nova estação já passou a participar das negociações de serviços e assumir tarefas.

Manter a produção mesmo com a mudança da estação de posição física na planta: para esta análise uma vez montado o sistema de manufatura e solicitada a produção de produtos,

uma das estações é deslocada de sua posição e colocada em outra, ao longo da mesma esteira. O que se espera é que o sistema mantenha a produção mesmo com a mudança de posição da estação na planta, com o mínimo de intervenção no programa que está sendo executado no CLP.

No Sistema Integrado de Manufatura a mudança de leiaute provocou a parada do sistema, seu desligamento, mudança das ligações do sensoriamento da estação, reprogramação do CLP, compilação e *download* do novo programa. Isto devido ao fato que a sequência de produção no programa do CLP já estava definida em sua ordem física na planta.

No Sistema de Gestão por MES a mudança de posição da estação na planta implicou na alteração no MES dos parâmetros que apontavam para os dispositivos de entradas e saídas desta estação, para poder indicar a nova posição na planta.

No Sistema Auto-organizável de manufatura a mudança do leiaute não requereu a parada do sistema, nem sua reprogramação. Fato que ao ser instanciado o agente máquina, este requer como parâmetro seu vínculo com a posição na esteira transportadora. Uma vez definido este parâmetro, os demais agentes saberão em que posição esta estação se encontra.

Comparando as métricas definidas para os três cenários deste capítulo de aplicação do método e mais o cenário que foi aplicado no desenvolvimento do método no capítulo anterior, onde se propôs uma abordagem centralizada no controlador lógico programável. Em seus resultados tem-se o levantamento de atendimentos, como descreve a tabela 19.

**Tabela 19 Resultados da avaliação dos sistemas de gestão.**

<b>Métricas</b>	<b>CLP</b>	<b>CIM</b>	<b>MES</b>	<b>CLP+MAS</b>
Manter a produção com a mudança nos requisitos.	Não atende	Não atende	Atende	Atende
Manter a produção com inserção de um único produto em meio a um lote.	Não atende	Atende	Atende	Atende
Manter a produção com a retirada de estações.	Não atende	Não atende	Não atende	Atende
Mudar a ordem de processo em função da não disponibilidade do recurso.	Não atende	Não atende	Atende	Atende

Manter a produção com a inserção de estações.	Não atende	Não atende	Não atende	Atende
Manter a produção mesmo com a mudança da estação de posição física na planta.	Não atende	Não atende	Não atende	Atende

Fonte: Autor.

Pela tabela apresentada e os resultados obtidos se observa que em comparação com o sistema gerenciado por CIM, MES e Auto-organizável (CLP+MAS), os Sistemas Auto-organizáveis não requerem a parada, desligamento ou reprogramação do sistema caso haja necessidade de alterar os requisitos do produto, inserir novas estações, retirar estações ou modificar sua posição no sistema de manufatura.

A essência do controle de cada estação continuou a mesma, pois continuaram com um controlador lógico programável e seu programa de controle não foi modificado, mas a ele foram agregadas as características de Sistemas Multiagentes com aplicação da metodologia proposta, o que levou o sistema de manufatura atender a diversidade, ser adaptável e ser auto-organizado, uma vez que o fluxo do processo produtivo se encontra descrito no agente que modela a peça e este é que determina a ordem com que os recursos serão utilizados.

Os resultados mostram que o método é aplicável e leva o sistema com controlador lógico programável a atender às necessidades de diversidade, agilidade e auto-organização do meio produtivo.

## 7 CONCLUSÕES

Para projeto de sistemas de manufatura para plantas novas, que ainda não haviam sido implantadas, já há soluções em nível de pesquisa que dão conta de atender às funcionalidades requeridas. Porém necessitam que os sistemas de gestão sejam a partir de controladores industriais que possuam capacidade e disponibilidade de executar agentes em *software*.

Os sistemas de manufatura já implantados, que possuem controladores lógicos instalados e executando suas lógicas locais, encontram uma dificuldade em migrar para um sistema auto-organizável, pois não é factível descartar os controladores existentes em substituição a computadores industriais, que possuem suporte computacional para a execução de Sistemas Multiagentes.

A solução passa então à busca de um método que permita à sistemas que já possuem controladores lógicos programáveis migrar para o novo conceito sem que haja este descarte. Fazendo com que os Controladores lógicos programáveis tenham as características de Sistemas Multiagentes.

O método proposto como tese atendeu ao objetivo ao possibilitar que uma planta de demonstração que já possuía um controlador lógico migrasse para o conceito de Sistemas Multiagentes, passando a atender as características de absorver produtos diversificados, e atender os requisitos de agilidade e auto-organização do sistema. Neste método há a instalação de um sistema computacional que executa o agente, que tem a função de comandar o controlador lógico e interagir com os demais agentes no sistema de manufatura.

O método foi ensaiado em um cenário composto de estações de manufatura, onde cada estação possuía um CLP de controle. E neste cenário o método foi aplicado, gerando assim dois ambientes: um em que a gestão do fluxo produtivo estava definida em um CLP e outro onde o Sistema Multiagente que definia o fluxo produtivo, estando este descrito no próprio agente de

produto. Neste cenário o método se mostrou eficaz ao atender as seis métricas propostas e remeteu ao atendimento dos requisitos de diversidade, agilidade e auto-organização.

A implantação dos PLCAgent - agentes que transformam um CLP em um agente – se deu através de uma comunicação via rede industrial *Profinet* entre o CLP e o agente. Junto a classe AgentCom que promoveu a interação com o controlador lógico programável, formam então uma grande contribuição deste trabalho para comunidade científica, comprovando sua funcionalidade e estando a disposição para outras implementações.

A problemática apresentada remete a necessidade de que o controlador lógico programável interagir com outros controladores, através das funcionalidades de sistemas multiagentes. Uma vez que o processo produtivo estiver em marcha, a partir de um conceito auto-organizável, cada produto é que irá gerenciar seu fluxo produtivo. Este fato pode provocar problemas na coordenação das atividades, no momento em que os produtos concorrem por recursos. Este trabalho não abordou este fato, pois o foco era levar o CLP à interação, mas é um fator que pode ocorrer e se apresenta como uma limitação do trabalho.

Assim como a ocorrência de *deadlock*, assunto abordado no trabalho com proposta de solução, mas que não foi implementada. Cabendo então esta ressalva de que a ocorrência deste fator deve ser tratada em sistemas produtivos maiores, com maior concorrência e negociações entre recursos.

Este trabalho focou solução a partir do uso de controladores lógicos programáveis, componente de gestão da manufatura largamente utilizado na indústria. E para a execução do agente, que irá interagir com o controlador, faz-se necessário a utilização de um computador ou outro recurso computacional. E o uso de computador em ambiente industrial enfrenta uma certa resistência por parte da indústria, por questões de robustez estrutural e também cultural, pois a indústria não costuma reagir bem a implantação de novidades em seu sistema fabril sem um

extensivo período de testes e validação. Este fato pode ser um limitante a aplicação prática do que é proposto como tese.

Outro fator limitante a proposta deste trabalho é quanto à forma com que a lógica local do controlador lógico programável foi implementada. A fase 1 da metodologia prevê a identificação no controlador lógico dos sinais de comando, informação, situação e resposta. Se o programa da lógica local no controlador não estiver implementado de uma forma estruturada, pode ser difícil localizar tais sinais dentro do programa e suas relações com as demais lógicas. Portanto, o método pressupõe que os programas das lógicas locais nos controladores lógicos programáveis estejam implementados de forma estruturada.

Como trabalho futuro, em segmento a este trabalho, se propõe experimentar e avaliar o uso de outras plataformas e paradigmas, como é o caso da plataforma JAISON (<http://jason.sourceforge.net/>) e Camel Apache (<http://camel.apache.org/>), onde outros protocolos de comunicação entre agentes podem serem ensaiados e comparados quanto do seu desempenho, em termos de comunicação entre os agentes. Trabalho que se julga importante, pois no cenário da manufatura com sistemas auto-organizáveis aqui implementada ocorreu a execução de 15 agentes simultâneos dentro da plataforma JADE. A análise de desempenho pode apresentar bons resultados indicativos para caminhos futuros e aplicação fabril desta tese.

A metodologia proposta nesta tese apontou a interação física do agente com o controlador lógico programável através de rede industrial, sugerindo a utilização de sistema SCADA ou drive físico para tal interação. Como trabalho futuro é interessante um estudo quando a eficácia destas outras duas propostas, verificando sua forma de implementação e sua interferência no processo interativo.

Nesta tese, ao aplicar a metodologia, o sistema produtivo passou a operar no conceito de sistemas auto-organizáveis, onde o produto é que faz a gestão do fluxo de processo produtivo. Mas este agente não tem conhecimento das operações dos demais agentes de

produto, portanto, a otimização do sistema produtivo não foi abordada neste trabalho. Fica a sugestão para trabalho futuro de um estudo para implementação de um agente de otimização, com a função de monitorar os fluxos de processos de todos os agentes da plataforma e propor alterações de fluxo de forma a otimizar o processo produtivo.

A metodologia proposta visa o atendimento a processos em sistemas produtivos discretos. Isto não significa que a metodologia não se aplique a processos contínuo, porém, em uma análise prévia a aplicação do conceito de auto-organização faz mais sentido quando há concorrência entre recursos, e os processos contínuos já possuem um fluxo fisicamente definido, havendo poucos desvios de fluxo de seu processo. Mas cabe um estudo mais aprofundado para verificar a pertinência de aplicação da metodologia também nos processos contínuos, e isto pode ocorrer em um trabalho futuro.

Em termos de estado da arte há uma tendência forte na aplicação de sistemas “cyber-físicos”, componentes que agregam a parte física e lógica de um elemento de trabalho, dotados de capacidade computacional para execução de sua lógica de controle e interação com os demais componentes de um sistema produtivo. Neste contexto o sistema multiagente já possui o meio para interação com tais componentes, cabendo sua execução e atuação como um agente. Esta será uma grande revolução dentro do meio industrial, e a tese aqui proposta já é um primeiro passo para chegar neste grau de tecnologia.

Há também uma forte tendência de os processos produtivos evoluírem para atuar de forma interativa, atendendo os preceitos da indústria 4.0, que aponta para virtualização dos processos, interoperabilidade, descentralização, adaptação da produção em tempo real, orientação a serviços e sistemas modulares dos equipamentos. E para isto é necessário que os recursos de um sistema produtivo tenham a capacidade de interação. Esta tese traz uma metodologia que leva o controlador lógico programável a ter a capacidade de interação, a partir das funcionalidades dos sistemas multiagentes que a ele fora disponibilizado. Com esta

interação a tese leva os recursos de produção a interagirem entre si, ao encontro da tendência dos sistemas produtivos em atuarem conforme os preceitos da indústria 4.0.

O que se observa é que o trabalho proposto foca uma abordagem e a ela apresenta seus resultados em termos de funcionalidades. Mas possibilita diversos trabalhos futuros, várias linhas de pesquisa que surgem a partir desta abordagem. E se apresenta em consonância com as tendências de futuro a sistemas produtivos industriais.

Ao final deste trabalho fica definido uma metodologia e avaliado seu desempenho para situações de aplicação em manufatura industrial, com resultados publicados e divulgados à comunidade científica, contribuindo para a Ciência e para a solução dos problemas da manufatura industrial.

## REFERÊNCIAS

ALBUQUERQUE, Pedro Urbano Braga de; ALEXANDRIA, Auzuir Ripardo de. **Redes Industriais: Aplicações em Sistemas Digitais de Controle Distribuídos**. Fortaleza: Ensino Profissional, 2009.

BABICEANU, Radu F.; CHEN, F. Frank. Development and Applications of Holonic Manufacturing Systems: A Survey. **International journal of intelligent manufacturing**, Cham, v. 17, n. 1, p.111-131, Feb. 2006. Springer Science + Business Media. <http://dx.doi.org/10.1007/s10845-005-5516-y>.

BARATA, José; RIBEIRO, Luis; ONORI, Mauro. Diagnosis on Evolvable Production Systems. In: IEEE INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS, 2007., 2007, Vigo. **Proceedings...** [S.l.]: IEEE, 2007. p. 3221 - 3226. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4375131&isnumber=4374555>>. Acesso em: 13 ago. 2015.

BARBOSA, Regina Maria Frei Santos. **Self-organisation in evolvable assembly systems**. 2010. 273 f. Tese (Doutorado) - Curso de Engenharia Electrotécnica e de Computadores - Robótica e Manufactura Integrada, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, 2010. Disponível em: <<https://docs.google.com/file/d/0B4y8mLzahQT6TU1Qd2ZBOS0tNnM/edit>>. Acesso em: 12 abr. 2015.

BEACH, Roger et al. A review of manufacturing flexibility. **European Journal Of Operational Research**, Amsterdam, v. 122, n. 1, p.41-57, Apr. 2000. Elsevier BV. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221799000624>>. Acesso em: 12 jan. 2016.

BELLIFEMINE, Fábio; CAIRE, Giovanni; GREENWOOD, Dominic. **Developing multi-agent systems with JADE**. Chichester: John Wiley & Son, 2007. (Wiley series in agent technology).

BENSLIMANE, Djamal; DUSTDAR, Schahram; SHETH, Amit. Services Mashups: The New Generation of Web Applications. **IEEE Internet Computing**, Los Alamitos, v. 12, n. 5, p.13-15, Sept. 2008. Institute of Electrical & Electronics Engineers (IEEE). Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4620089>>. Acesso em: 15 jan. 2015.

BI, Zhuming et al. Reconfigurable manufacturing systems: the state of the art. **International Journal Of Production Research**, London, v. 46, n. 4, p.967-992, 17 Nov. 2007. Informa UK Limited. Disponível em: <<http://dx.doi.org/10.1080/00207540600905646>>. Acesso em: 12 mai 2016.

BLANCHARD, Benjamin S. **Logistics Engineering & Management**. 6. ed. New Jersey: Pearson, 2003.

BORDINI, Rafael H.; HÜBNER, Jomi Fred; WOOLDRIDGE, Michael. **Programming Multi-Agents Systems in AgentSpeak using Jason**. Chichester: John Wiley, 2007. (Wiley series in agent technology).

BROMURI, Stefano; STATHIS, Kostas. Situating Cognitive Agents in GOLEM. **Engineering Environment-mediated Multi-agent Systems**, Berlin, v. 5049, p.115-134, 5 Oct. 2007. Springer Science + Business Media. [http://dx.doi.org/10.1007/978-3-540-85029-8\\_9](http://dx.doi.org/10.1007/978-3-540-85029-8_9). Disponível em: <[http://link.springer.com/chapter/10.1007/978-3-540-85029-8\\_9](http://link.springer.com/chapter/10.1007/978-3-540-85029-8_9)>. Acesso em: 26 fev. 2015.

BRYCE, Ciarán; RAZAFIMAHEFA, Chrislain; PAWLAK, Michel. Lana: An Approach to Programming Autonomous Systems. In: ECOOP 2002 OBJECT-ORIENTED PROGRAMMING, 16., 2002, Málaga. **Proceedings...** Berlin: Springer Berlin Heidelberg, 2002. v. 2374, p. 281 - 308. Disponível em: <[http://link.springer.com/chapter/10.1007/3-540-47993-7\\_13#page-1](http://link.springer.com/chapter/10.1007/3-540-47993-7_13#page-1)>. Acesso em: 24 abr. 2015.

BUSSMANN, Stefan. An Agent-Oriented Architecture for Holonic Manufacturing Control. In: OPEN WORKSHOP, IMS EUROPE, 1., 1998, Lausanne. **Proceedings...** Keele: IFAC, 1998. p. 1 - 12. Disponível em: <<http://stefan-bussmann.de/downloads/ims98.pdf>>. Acesso em: 12 abr. 2015.

CÂNDIDO, Gonçalo Moreira. **Service-oriented architecture for device lifecycle support in industrial automation**. 2013. 244 f. Tese (Doutorado) - Curso de Engenharia Electrotécnica e de Computadores - Robótica e Manufatura Integrada, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Uninova, Lisboa, 2013. Disponível em: <<http://hdl.handle.net/10362/11222>>. Acesso em: 15 set. 2014.

CÂNDIDO, Gonçalo et al. SOA in reconfigurable supply chains: A research roadmap. **Engineering Applications Of Artificial Intelligence**, Amsterdam, v. 22, n. 6, p.939-949, Sept. 2009. Elsevier BV. <http://dx.doi.org/10.1016/j.engappai.2008.10.020>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0952197608001735#>>. Acesso em: 24 mar. 2015.

CAVALCANTE, André Luiz Duarte. **Arquitetura baseada em agentes e auto-organizável para a manufatura**. 2012. 140 f. Tese (Doutorado) - Curso de Programa de Pós-graduação em Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012. Disponível em: <<http://hdl.handle.net/10183/75873>>. Acesso em: 3 mar. 2014.

CAVALCANTE, Andre L.; PEREIRA, Carlos E.; BARATA, José. Component-Based Approach to the Development of Self-X Automation Systems. **IFAC Proceedings Volumes**, Amsterdam, v. 43, n. 4, p.222-227, 2010. Elsevier BV. <http://dx.doi.org/10.3182/20100701-2-pt-4011.00039>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1474667015301452>>. Acesso em: 16 dez. 2014.

COLOMBO, Armando W. Industrial agents: towards collaborative production, automation, management and organization. **IEEE Industrial Electronics Society Newsletter**, New Jersey, v. 52, n. 3, p.17-18, Nov. 2005.

CONTROL DESIGN. **Global PLC Market Coasts to 2016**. 2013. Online Journal. Disponível em: <<http://www.controldesign.com/industrynews/2012/global-plc-market-coasts-to-2016/>>. Acesso em: 15 jun. 2015.

CORREA, Henrique L, GIANESI, Irineu G. N. e CAON, Mauro. **Planejamento, Programação e Controle da Produção: MRP II / ERP**. São Paulo: Atlas. 1999.

COSMOS PROJECT. **Flexible Modular Automation**. 2013. Disponível em: <<http://www.cosmosproject.eu>>. Acesso em: 12 jan. 2015.

DEITEL, Paul; DEITEL, Harvey. **Java how to program**. 7. ed. Upper Saddle River: Pearson Prentice Hall, 2007.

DEMAZEAU, Yves. From interactions to collective behaviour in agent-based systems. In: EUROPEAN CONFERENCE ON COGNITIVE SCIENCE, 1., 1995, Saint-malo. **Proceedings...** Saint-malo: ECCS, 1995. p. 117 - 132. Disponível em: <[https://www.researchgate.net/publication/2782790\\_From\\_Interactions\\_To\\_Collective\\_Behaviour\\_In\\_Agent-Based\\_Systems](https://www.researchgate.net/publication/2782790_From_Interactions_To_Collective_Behaviour_In_Agent-Based_Systems)>. Acesso em: 12 ago. 2015.

EUPASS PROJECT. **Evolvable Ultra-Precision Assmby Systems**. 2014. Disponível em: <[http://cordis.europa.eu/result/rcn/51017\\_en.html](http://cordis.europa.eu/result/rcn/51017_en.html)>. Acesso em: 22 dez. 2014.

FEIMEC. **Feira Mundial de Máquinas e Ferramentas**. 2016. Disponível em: <<http://www.feimec.com.br>>. Acesso em: 12 jun. 2016.

FERBER, Jacques. **Multi-agent systems: an introduction to distributed artificial intelligence**. 1. ed. Boston: Addison-Wesley Professional, 1999.

FLEISCHHAUER, Luciana Irene Amaral. **O uso da tecnologia de agentes na integração da programação da produção**. 1996. 68 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Engenharia de Produção, Departamento da Engenharia da Produção, Universidade Federal de Santa Catarina, Florianópolis, 1996. Disponível em: <<https://repositorio.ufsc.br/xmlui/handle/123456789/158092>>. Acesso em: 18 jan. 2012.

FREI, Regina; SERUGENDO, Giovanna di Marzo; OLIVEIRA, Jose Antônio Barata de. Designing Self-Organization for Evolvable Assembly Systems. In: IEEE INTERNATIONAL CONFERENCE ON SELF-ADAPTIVE AND SELF-ORGANIZING SYSTEMS, 2., 2008, Venezia. **Proceedings...** New York: IEEE, 2008. p. 97 - 106. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4663414&isnumber=4663391>>. Acesso em: 3 mai. 2015.

FREITAS, Edison Pignaton de et al. Intelligent and Flexible Manufacturing Product Line Supported by Agents and Wireless Sensor and Actuator Network. **7th Ifac Conference On Manufacturing Modelling, Management, And Control**. Saint Petersburg, p. 222-227. jun. 2013. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1474667016342896>>. Acesso em: 25 set. 2015.

GEORGINI, Marcelo. **Automação aplicada: descrição e implementação de sistemas sequenciais com PLCs**. 6. ed. São Paulo: Érica, 2004.

GUIMARÃES, Hugo Casati Ferreira. **Norma IEC 61131-3 para programação de controladores programáveis**: estudo e aplicação. 2005. 86 f. Monografia (Especialização) - Curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, 2005. Disponível em: <<http://docslide.com.br/documents/normas-para-programacao.html>>. Acesso em: 22 dez. 2014.

HERRERA, Vladimir Villasenor et al. Integration of Multi-Agent Systems and Service-Oriented Architecture for industrial automation. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS, 6., 2008, Daejeon. **Proceedings...** New York: IEEE, 2008. p. 768 - 773. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4618205&isnumber=4618046>>. Acesso em: 23 out. 2014.

HERMANN, Mario; PENTEK, Tobias; OTTO, Boris. **Design Principles for Industrie 4.0 Scenarios**: A Literature Review. 2015. Disponível em: <[http://www.snom.mb.tu-dortmund.de/cms/de/forschung/Arbeitsberichte/Design-Principles-for-Industrie-4\\_0-Scenarios.pdf](http://www.snom.mb.tu-dortmund.de/cms/de/forschung/Arbeitsberichte/Design-Principles-for-Industrie-4_0-Scenarios.pdf)>. Acesso em: 29 fev. 2016.

IDEAS PROJECT. **FP7 - Instantly Deployable Evolvable Assembly Systems**. 2010. Disponível em: <<http://www.ideas-project.eu/>>. Acesso em: 20 dez. 2014.

IEC. **International Electrotechnical Commission**. 2003. Disponível em: <<http://www.iec.ch/>>. Acesso em: 10 jul. 2016.

JADEJA, Yashpalsinh; MODI, Kirit. Cloud computing - concepts, architecture and challenges. In: INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRONICS AND ELECTRICAL TECHNOLOGIES (ICCEET), 2012, Kumaracoil. **Proceedings...** New York: IEEE, 2012. p. 877 - 880. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6203873&isnumber=6203726>>. Acesso em: 7 maio 2012.

JAMMES, Francois; SMIT, Harm. Service-Oriented Paradigms in Industrial Automation. **IEEE Transactions On Industrial Informatics**, New Jersey, v. 1, n. 1, p.62-70, Feb. 2005. Institute of Electrical & Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tii.2005.844419>. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1411764>>. Acesso em: 13 set. 2015.

JONSSON, Patrik; LESSHAMMAR, Magnus. Evaluation and improvement of manufacturing performance measurement systems - the role of OEE. **International Journal of Operations & Production Management**, Bradford, v. 19, n. 1, p.55-78, Jan. 1999. Emerald. <http://dx.doi.org/10.1108/01443579910244223>. Disponível em: <<http://www.emeraldinsight.com/doi/pdfplus/10.1108/01443579910244223>>. Acesso em: 23 mar. 2016.

KOTLER, Philip; REIN, Irving J.; STOLLER, Martin. **Marketing de Alta Visibilidade**. 1.ed. São Paulo: Makron, 1999.

KOTLER, Philip; ARMSTRONG, Gary. **Princípios de marketing**. 12. ed. São Paulo: Prentice Hall Brasil, 2007.

KOREN, Y. et al. Reconfigurable Manufacturing Systems. In: DASHCHENKO, Anatoli I. **Manufacturing Technologies for Machines of the Future: 21st Century Technologies**. Berlin: Springer Berlin Heidelberg, 2003. Cap. 19. p. 627-665. Disponível em: <[http://link.springer.com/chapter/10.1007/978-3-642-55776-7\\_19](http://link.springer.com/chapter/10.1007/978-3-642-55776-7_19)>. Acesso em: 14 mar. 2015.

LANGE, Danny; MITSURU, Oshima. **Programming and Deploying Java Mobile Agents Aglets**. 1. ed. Boston: Addison-Wesley Longman Publishing, 1998.

LEITÃO, Paulo et al. Trends in agile and cooperative manufacturing. In: PROCEEDINGS OF THE LOW COST AUTOMATION SYMPOSIUM, 2001. Berlin. **Proceedings...** Berlin: Elsevier, 2001. p. 156 - 165. Disponível em: <<https://bibliotecadigital.ipb.pt/handle/10198/1426>>. Acesso em: 12 abr. 2016.

LEITÃO, Paulo. Agent-based distributed manufacturing control: A state-of-the-art survey. **Engineering Applications Of Artificial Intelligence**, New York, v. 22, n. 7, p.979-991, Oct. 2009. Elsevier BV. <http://dx.doi.org/10.1016/j.engappai.2008.09.005>. Disponível em: <<http://dl.acm.org/citation.cfm?id=1618624.1618834>>. Acesso em: 27 ago. 2014.

LEITÃO, Paulo; RESTIVO, Francisco. An agile and cooperative architecture for distributed manufacturing systems. In: INTERNATIONAL CONFERENCE ROBOTICS AND MANUFACTURING - IASTED, 2001. Cancun. **Proceedings...** Anaheim: IASTED, 2001. p. 21-24. Disponível em: <<http://bibliotecadigital.ipb.pt/bitstream/10198/1467/1/2001-RM.pdf>>. Acesso em: 12 nov. 2015.

LEITÃO, Paulo; RESTIVO, Francisco. ADACOR: A holonic architecture for agile and adaptive manufacturing control. **Computers In Industry**, Amsterdam, v. 57, n. 2, p.121-130, Feb. 2006. Elsevier BV. <http://dx.doi.org/10.1016/j.compind.2005.05.005>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0166361505001077>>. Acesso em: 03 out. 2014.

McCLELLAN, Michael, **Applying Manufacturing Execution Systems**. 1. ed. Boca Raton: St. Lucie Press, 1997.

MAGEDANZ, Thomas et al. **Grasshopper: A Universal Agent Platform Based on OMG MASIF and FIPA Standards**. 2009. Disponível em: <<http://cordis.europa.eu/infowin/acts/analysys/products/thematic/agents/ch4/ch4.htm>>. Acesso em: 26 mar. 2016.

MARIK, Vladimír; MCFARLANE, Maynard D. Industrial Adoption of Agent-Based Technologies. **IEEE Intelligent Systems**, Los Alamitos, v. 20, n. 1, p.27-35, Jan. 2005. Institute of Electrical & Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/mis.2005.11>. Disponível em: <[http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1392671&url=http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1392671](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1392671&url=http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1392671)>. Acesso em: 1 out. 2014.

MARIK, Vladimír et al. Rockwell Automation Agents for Manufacturing. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, 4., 2005, New York. **Proceedings...** New York: Acm, 2005. p. 107 - 113. Disponível em: <<http://dl.acm.org/citation.cfm?id=1082812>>. Acesso em: 8 maio 2015.

MENDES, J. Marco et al. Service-Oriented Control Architecture for Reconfigurable Production Systems. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS -INDIN2008, 2008, Daejeon. **Proceedings...** New York: IEEE, 2008. p. 744 - 749. Disponível em: <<http://repositorio-aberto.up.pt/bitstream/10216/25257/2/27473.pdf>>. Acesso em: 9 maio 2015.

NAMBIAR, Arun N. Modern Manufacturing Paradigms: A Comparison. In: INTERNATIONAL MULTICONFERENCE OF ENGINEERS AND COMPUTER SCIENTISTS, 3., 2010, Hong Kong. **Proceedings...** Hong Kong: IMECS, 2010. p. 1662 - 1667. Disponível em: <[http://www.iaeng.org/publication/IMECS2010/IMECS2010\\_pp1662-1667.pdf?origin=publication\\_detail](http://www.iaeng.org/publication/IMECS2010/IMECS2010_pp1662-1667.pdf?origin=publication_detail)>. Acesso em: 22 set. 2015.

NARDELLA, Dave. **Snap7 multi-plataform Ethernet S7 PLC**. 2016. Disponível em: <<https://sourceforge.net/u/davenardella/profile/>>. Acesso em: 13 jan. 2016.

OTHMANE, Benyoucef; HEBRI, Rahal Sidi Ahmed. Cloud computing & multi-agent systems: A new promising approach for distributed data mining. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY INTERFACES (ITI), 34., 2012, Cavtat. **Proceedings...** New York: IEEE, 2012. p. 111 - 116. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6307989&isnumber=6307955>>. Acesso em: 23 abr. 2015.

OLIVEIRA, José António Barata de. **Coalition Based Approach for Shop Floor Agility: A Multiagent Approach**. 2003. 329 f. Tese (Doutorado) - Curso de Engenharia Electrotécnica e de Computadores - Robótica e Manufatura Integrada, Faculdade de Ciências da Computação, Engenharia Elétrica e Matemática, Universidade Nova de Lisboa Uninova, Lisboa, 2003. Disponível em: <[https://run.unl.pt/bitstream/10362/2483/1/Oliveira\\_2004.pdf](https://run.unl.pt/bitstream/10362/2483/1/Oliveira_2004.pdf)>. Acesso em: 6 jan. 2015.

OLIVEIRA, Jose Antônio Barata de; CAMARINHA-MATOS, Luis M. Coalitions of manufacturing components for shop floor agility: The CoBASA architecture. **International journal of networking and virtual organisations**, Genebra, v. 2, n. 1, p.50-77, Jan. 2003. Inderscience Publishers. Disponível em <<http://dx.doi.org/10.1504/ijnvo.2003.003518>>. Acesso em fev 2016.

OLIVEIRA, José Antônio Barata de; CAMARINHA-MATOS, Luís; CÂNDIDO, Gonçalo. A multiagent-based control system applied to an educational shop floor. **Robotics And Computer-integrated Manufacturing**, New York, v. 24, n. 5, p.597-605, Oct. 2008. Elsevier BV. Disponível em <<http://dx.doi.org/10.1016/j.rcim.2007.09.008>>. Acesso em nov 2016.

OMICINI, Andrea; RICCI, Alessandro; VIROLI, Mirko. Artifacts in the A&A meta-model for multi-agent systems. **Autonomous Agents And Multi-agent Systems**, Norwell, v. 17, n. 3, p.432-456, 15 May 2008. Springer Science + Business Media. <http://dx.doi.org/10.1007/s10458-008-9053-x>. Disponível em: <<http://link.springer.com/article/10.1007/s10458-008-9053-x>>. Acesso em: 15 dez. 2015.

ONORI, Mauro; SEMERE, Daniel; LINDBERG, Bengt. Evolvable systems: an approach to self-X production. **International Journal of Computer Integrated Manufacturing**, London, v. 24, n. 5, p.506-516, May 2011. Informa UK Limited. Disponível em <<http://dx.doi.org/10.1080/0951192x.2011.566282>>. Acesso em mar 2016.

PABADIS. **State of the Art in Factory Control and Requirement Specification for most advanced, most flexible, and most adaptable control of manufacturing systems.** 2006. Disponível em: <[www.pabadis-promise.org](http://www.pabadis-promise.org)>. Acesso em: 13 dez. 2014.

PECHOUCEK, Michal; MARIK, Vladimír. Industrial deployment of multi-agent technologies: review and selected case studies. **Autonomous Agents And Multi-agent Systems**, Norwell, v. 17, n. 3, p.397-431, 14 May 2008. Springer Science + Business Media. <http://dx.doi.org/10.1007/s10458-008-9050-0>. Disponível em: <<http://link.springer.com/article/10.1007/s10458-008-9050-0>>. Acesso em: 17 ago. 2014

PEIXOTO, João Alvarez. **Desenvolvimento de sistemas de automação da manufatura usando arquiteturas orientadas a serviço e sistemas multiagentes.** 2012. 134 f. Dissertação (Mestrado) - Curso de Pós-graduação em Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012. Disponível em: <<http://hdl.handle.net/10183/61385>>. Acesso em: 13 fev. 2014.

PESCHL, Michael. **An architecture for flexible manufacturing systems based on taskdriven agents.** 2014. 192 f. Tese (Doutorado) - Curso de Faculdade de Tecnologia da Informação e Engenharia Elétrica, Departamento de Ciências da Computação e Engenharia, Universidade de Oulu, Oulu, 2014. Disponível em: <<http://herkules.oulu.fi/isbn9789526203669/isbn9789526203669.pdf>>. Acesso em: 20 out. 2014.

PRIME PROJECT. **Plug and Produce Intelligent Multi Agent Environment based on Standard Technology.** 2014. Disponível em: <<http://www.prime-eu.com>>. Acesso em: 05 dez. 2014.

QIU, Robin Guanghua; ZHOU, Mengchu. Mighty MESS: State of the art and future manufacturing execution systems. **IEEE Robotics & Automation Magazine**, New York, v. 11, n. 1, p.19-40, May. 2004. Institute of Electrical & Electronics Engineers (IEEE). Disponível em: <<http://dx.doi.org/10.1109/mra.2004.1275947>>. Acesso em 6 de março de 2016.

RIBEIRO, Luís Domingos Ferreira. **Diagnosis in evolvable production systems.** 2012. 230 f. Tese (Doutorado) - Curso de Engenharia Elétrica e Computação, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, 2012. Disponível em: <<http://hdl.handle.net/10362/7074>>. Acesso em: 12 set. 2014.

RIBEIRO, Luís; BARATA, José; MENDES, Pedro. MAS and SOA: Complementary Automation Paradigms. In: IFIP ADVANCES IN INFORMATION AND COMMUNICATION TECHNOLOGY, 8., 2008, Porto. **Proceedings...** Cham: Springer, 2008. p. 259 - 268. Disponível em: <<http://www.springerlink.com/content/g75735487077g73x/fulltext.pdf>>. Acesso em: 28 mar. 2014.

ROLOFF, Mário Lucio. **Uma nova Abordagem para a Implementação de Sistemas Multiagentes para o Controle da Produção de Pequenas Séries.** 2014. 294 f. Tese (Doutorado) - Curso de Programa de Pós-graduação em Engenharia de Automação e Sistemas, Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina,

Florianópolis, 2014. Disponível em:

<<https://repositorio.ufsc.br/xmlui/handle/123456789/129497>>. Acesso em: 16 jun. 2016.

SCHEER, August-Wilhelm. **CIM Computer Integrated Manufacturing: Towards the Factory of the Future**. 3. ed. Dordrecht: Springer Science & Business Media, 2012.

SLACK, Niguel; CHAMBERS, Stuart; JOHNSTON, Robert. **Administração da Produção**. 3. ed. São Paulo: Atlas, 2009.

SILVA, Leonardo Ayres de Moraes. **Estudo e Desenvolvimento de Sistemas Multiagentes usando JADE: Java Agent Development framework**. 2003. 97 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Bacharel em Informática, Centro de Ciências Tecnológicas, Universidade de Fortaleza, Fortaleza, 2003. Disponível em: <<http://jade.cselt.it/papers/2003/monografia.pdf>>. Acesso em: 22 dez. 2014.

SILVA, Robson Marinho da. **Controle de sistemas reconfiguráveis de manufatura**. 2016. 157 f. Tese (Doutorado) - Curso de Ciências, Escola Politécnica, Universidade de São Paulo, São Paulo, 2016.

SOPRO PROJECT. **Self-Organizing Production**. 2014. Disponível em: <<http://www.izm.fraunhofer.de/en/abteilungen/>>. Acesso em: 13 nov. 2014.

SOUZA, Adriano Fagali; ULBRICH, Cristiane Brasil Lima. **Engenharia Integrada por Computador e Sistemas CAD/CAM/CNC**. 2. ed. São Paulo: Artliber. 2013.

TANENBAUM, Andrew S.; WETHERALL, David. **Redes de Computadores**. 5. ed. Rio de Janeiro: Campus, 2011.

THOMASI, Camila Dasso. **OrIAs: Uma infraestrutura de nível micro-organizacional baseada em Artefatos para Sistemas Multiagentes**. 2014. 121 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Computação, Departamento de Ciências da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014. Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/90439/000915240.pdf?sequence=1>>. Acesso em: 5 maio 2015.

VALLE, Cyro Eyer do. **Qualidade ambiental - ISO 14000**. 5. ed. São Paulo: Editora Senac, 2002.

VRBA, Pavel. Simulation in agent-based control systems: MAST case study. **International Journal Of Manufacturing Technology And Management**, Genova, v. 8, n. 1/2/3, p.175-187, 2006. Inderscience Publishers. <http://dx.doi.org/10.1504/ijmtm.2006.008794>. Disponível em: <<http://www.inderscienceonline.com/doi/pdf/10.1504/IJMTM.2006.008794>>. Acesso em: 25 abr. 2015.

VRBA, Pavel; MARIK, Vladimír. Capabilities of Dynamic Reconfiguration of Multiagent-Based Industrial Control Systems. **IEEE Transactions On Systems, Man, And Cybernetics - Part A: Systems and Humans**, New York, v. 40, n. 2, p.213-223, March. 2010. Institute of Electrical & Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tsmca.2009.2034863>. Disponível em: <[http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5353631&url=http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5353631](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5353631&url=http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5353631)>. Acesso em: 8 maio 2015.

VRBA, Pavel et al. Rockwell Automation's Holonic and Multiagent Control Systems Compendium. **IEEE Transactions On Systems, Man, And Cybernetics, Part C (applications And Reviews)**, [s.l.], v. 41, n. 1, p.14-30, Jan. 2011. Institute of Electrical & Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tsmcc.2010.2055852>. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5545420&isnumber=5669216>>. Acesso em: 3 out. 2014

WOOLDRIDGE, Michael. **An Introduction to MultiAgent Systems**. 1 ed. Chichester: John Wiley & Sons, 2002.

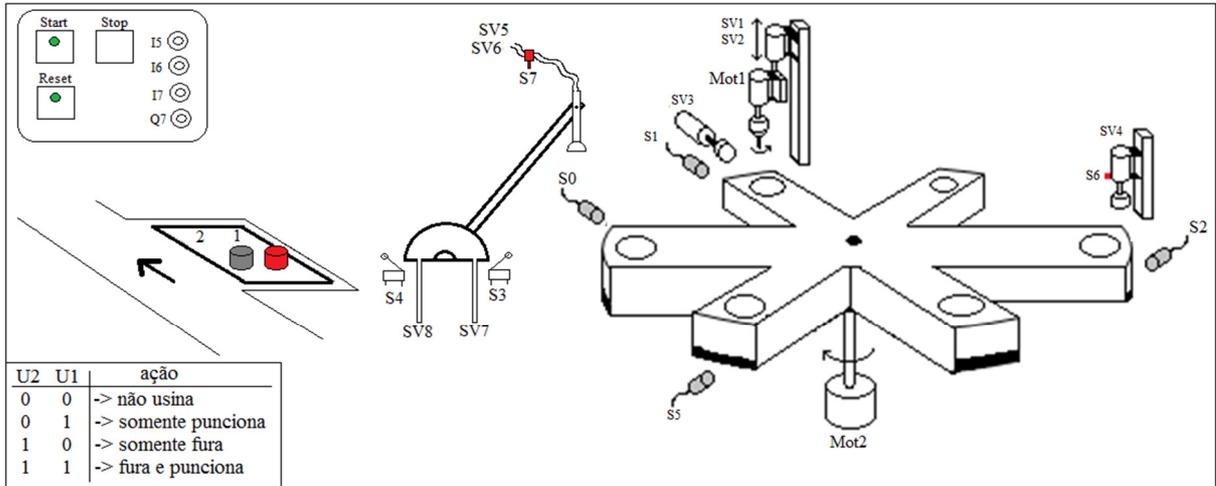
WOOLDRIDGE, Michael. **An Introduction to MultiAgent Systems**. 2 ed. Chichester: John Wiley & Sons, 2009.

XPRESS PROJECT. **Flexible production experts for reconfigurable assembly technology**. 2014. Disponível em: <<http://www.xpress-project.eu/>>. Acesso em: 19 nov. 2014.

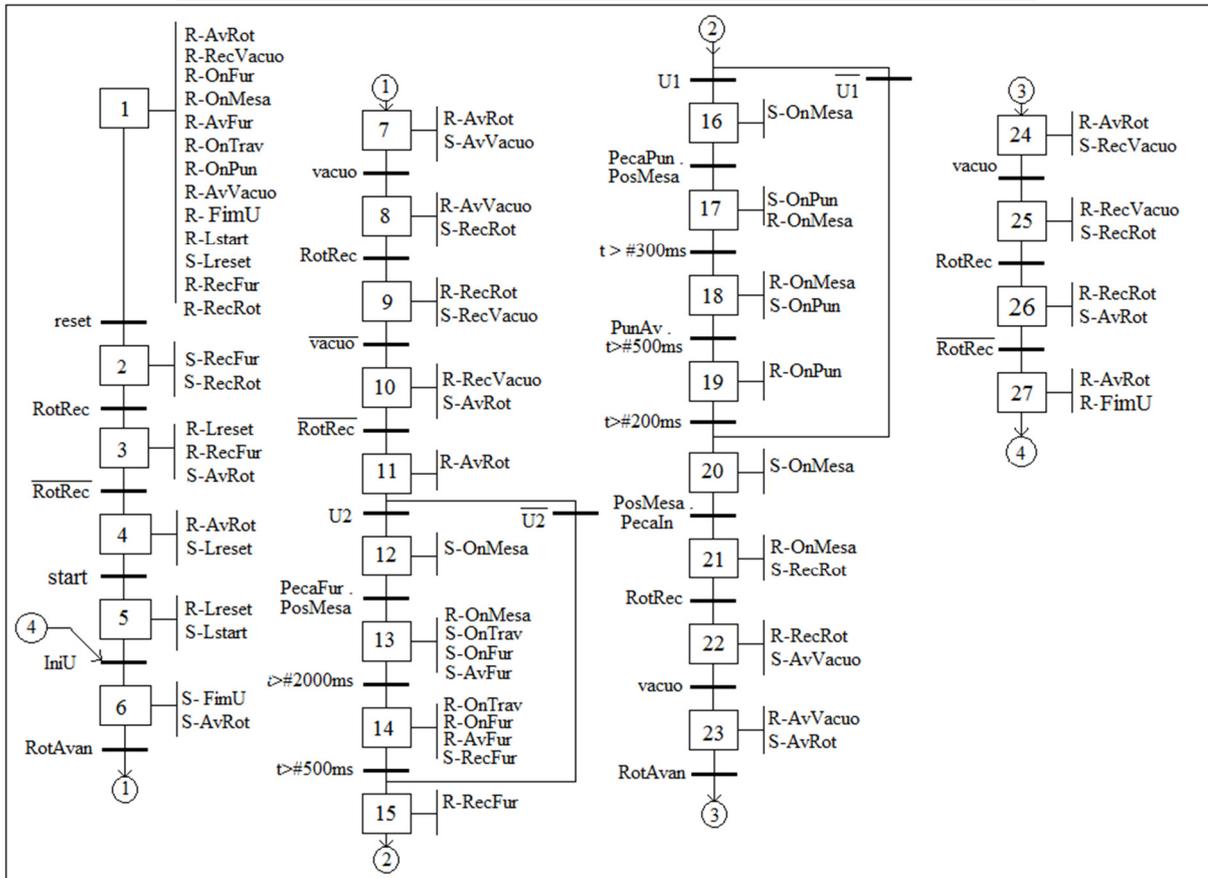
ZHANG, Zhengwen; SHARIFI, H. Towards Theory Building in Agile Manufacturing Strategy: a Taxonomical Approach. **IEEE Transactions On Engineering Management**, New York, v. 54, n. 2, p.351-370, May 2007. Institute of Electrical & Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tem.2007.893989>. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4160195&isnumber=4160146>>. Acesso em: 10 ago. 2015.

# ANEXO I: Programas do CLP para uma sequência produtiva.

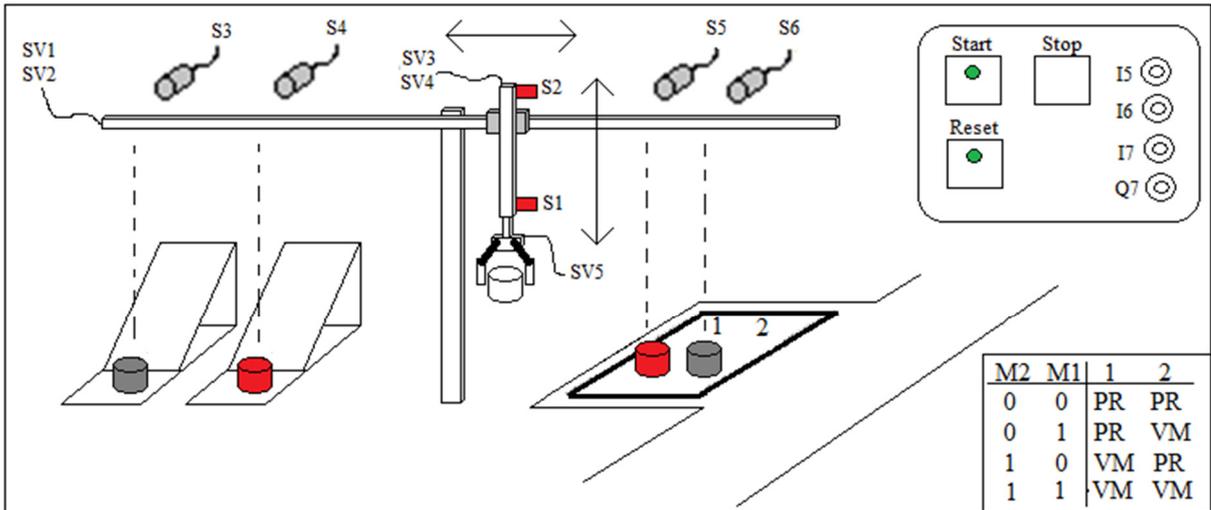
## a) Estação de Usinagem



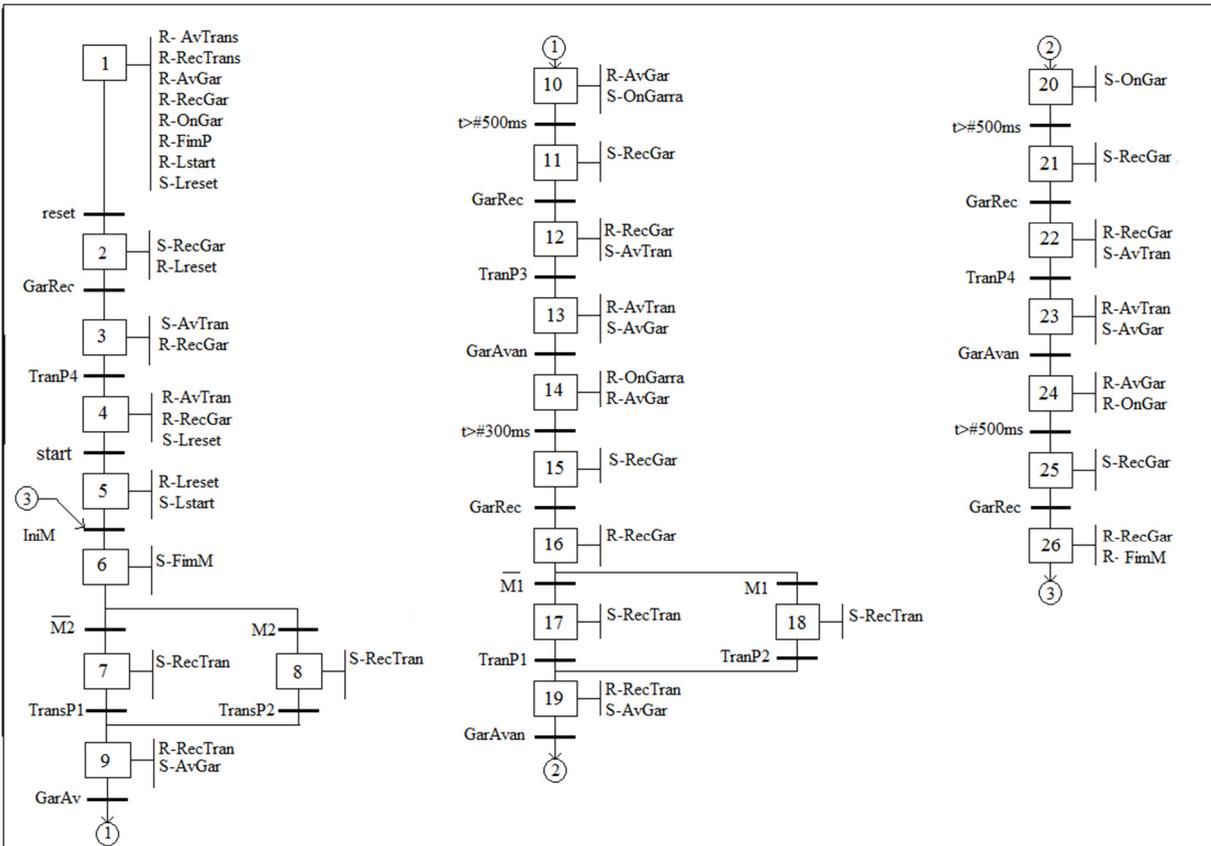
S0 - I0.0 - PecaIn	b1 - I1.0 - start	Mot1 - Q0.0 - OnFur	SV7 - Q1.4 - AvRot
S1 - I0.1 - PecaFur	b0 - I1.1 - stop	Mot2 - Q0.1 - OnMesa	SV8 - Q1.5 - RecRot
S2 - I0.2 - PecaPun	b2 - I1.3 - reset	SV1 - Q0.2 - AvFur	Q7 - Q1.7 - FimU
S3 - I0.3 - RotRec		SV2 - Q0.3 - RecFur	- Q1.0 - Lstart
S4 - I0.4 - RotAvan	15 - I1.5 - U1	SV3 - Q0.4 - OnTrav	- Q1.1 - Lreset
S5 - I0.5 - PosMesa	16 - I1.6 - U2	SV4 - Q0.5 - OnPunc	
S6 - I0.6 - PunAvan	17 - I1.7 - IniU	SV5 - Q0.6 - AvVacuo	
S7 - I0.7 - Vacuo		SV6 - Q0.7 - RecVacuo	



b) Estação de Montagem.



S1 - I0.0 - GarAvan	b1 - I1.0 - start	SV1 - Q0.0 - AvTran	- Q1.0 - Lstart - Q1.1 - Lreset Q7 - Q1.7 - FimM
S2 - I0.1 - GarRec	b0 - I1.1 - stop	SV2 - Q0.1 - RecTran	
S3 - I0.2 - TranP1	b2 - I1.3 - reset	SV3 - Q0.2 - AvGar	
S4 - I0.3 - TranP2	I5 - I1.5 - M1	SV4 - Q0.3 - RecGar	
S5 - I0.4 - TranP3	I6 - I1.6 - M2	SV5 - Q0.4 - OnGarra	
S6 - I0.5 - TranP4	I7 - I1.7 - IniM		



c) Estação de Transporte.

