

Integração vertical de um ERP em ambiente de fabricação virtual

Miguel Miragaia Gomes Inácio

Dissertação do MIEM

Orientador: Prof. António José Pessoa de Magalhães (FEUP)

Coorientador: Eng.º Bruno Teófilo Vigário (Real Games Lda.)



Mestrado Integrado em Engenharia Mecânica

Ramo de Automação

Setembro de 2015

*“Lord grant me strength to change the things i can,
the courage to accept the things i can’t
and the wisdom to know the difference”*

Reinhold Niebuhr

Resumo

A crescente complexidade e dimensão dos modernos sistemas industriais torna difícil lecionar as matérias relativas à automação e controlo dos mesmos. Com o intuito de alcançar um melhor método de ensino dos futuros engenheiros de automação, recorre-se por vezes a modelos didáticos capazes de demonstrar as atuais e futuras tecnologias aplicadas na indústria. Estes modelos didáticos são essenciais para a aquisição de sensibilidade prática e espírito crítico na análise e construção de sistemas automáticos.

A presente dissertação pretende ser um importante contributo no ensino prático das características e comportamentos de sistemas integrados de controlo e gestão, consistindo no desenvolvimento, em ambiente de sala de aula, de uma integração vertical de aplicações de controlo com aplicações de gestão, para fins didáticos. Como base para o desenvolvimento desta dissertação foi utilizado um simulador de chão de fábrica, o Factory IO, que proporciona, na sala de aula, um ambiente industrial seguro e controlado.

O presente trabalho inicia-se pela análise extensa dos sistemas integrados de controlo e gestão na indústria, caracterizando a sua estrutura, tecnologias aplicadas, normas e conceitos base. O levantamento de todas as especificidades permitiu definir a estrutura de aplicações a implementar, sendo estas posteriormente selecionadas.

Após a análise das aplicações escolhidas para implementação na sala de aula, procedeu-se à integração das mesmas, estabelecendo comunicações de modo a integrar as atividades realizadas por cada uma.

Finalmente a integração foi testada para validação do resultado final. Este revelou-se satisfatório apresentado a dissertação uma integração que permite aos alunos desenvolver ambientes fabris que comunicam com aplicações MOM (*Manufacturing Operations Management*) e ERP (*Enterprise Resource Planning*) possibilitando o ensino destas tecnologias na sala de aula.

Abstract

Vertical Integration of an ERP into a Virtual Factory Simulator

The growing complexity and dimension of industrial automated systems complicates the education of this subject. With the objective of guaranteeing the best training for the future automation engineers, physical models capable of demonstrating the current and future industrial technologies are used in the classroom. This didactic models let the students acquire practical experience and critical thinking in the analysis and construction of automated systems.

The present dissertation intends to be a valuable contribution in the practical education of the characteristics and behaviours of integrated control and enterprise systems, through the development of a integrated control and enterprise system in a classroom. In order to recreate a safe and controlled industrial environment the integration used a virtual factory simulator, Factory IO.

The present report starts with an extensive analysis of the integrated control and enterprise systems currently used in the industry, describing their structure, technologies, standarts and basic concepts. This study allowed the specification of the structure of applications to implement in the classroom.

After analyzing the selected applications, communications where established in order to integrate each application's activities together.

Lastly the integration was tested. The final result of the dissertation was satisfactory, having created a integration that allows students to develop production systems that comunicate with MOM applications (*Manufacturing Operations Management*) and ERP applications (*Enterprise Resource Planning*), facilitating the teaching of this tecnologies in the classroom.

Agradecimentos

Gostaria em primeiro lugar, de agradecer ao professor António Pessoa de Magalhães por me ter dado a oportunidade de desenvolver um tema de especial interesse para mim, pelo constante apoio, disponibilidade e profissionalismo.

Um agradecimento especial também ao meu orientador, Eng.º Bruno Teófilo Vigário, pelo apoio, disponibilidade, e exemplo que me marcou.

Agradeço também a todos os intervenientes da Critical Manufacturing, em especial ao CEO Francisco Almada Lobo e ao Eng. Manuel Brandão que me auxiliaram no decorrer da dissertação, pelo profissionalismo, apoio e disponibilidade.

Sou grato ao professor Francisco Jorge Teixeira de Freitas, pelo acompanhamento que prestou ao organizar as reuniões de atualização de progresso e a disponibilidade prestada.

Dirijo ainda a minha gratidão a todos os meus colegas que me apoiaram ao longo destes longos meses, em especial ao Vitor Veiga, Luis Santos e Luis Máximo.

E principalmente à minha família que sempre estiveram lá para mim, à minha mãe por ter feito de mim quem eu sou hoje, ao meu pai pelo gosto da indústria e ao meu irmão Pedro pelo exemplo que sempre foi para mim, quer como pessoa quer como profissional.

Índice de Conteúdos

1	Introdução	1
1.1	Objetivos	2
1.2	Organização da tese	3
2	Sistemas Integrados de Controlo e Gestão.....	5
2.1	Integração Vertical	6
2.1.1	Estrutura da produção.....	6
2.1.2	Automação na indústria	6
2.1.3	Computer Integrated Manufacturing.....	7
2.1.4	Normas aplicáveis à integração de sistemas	7
2.1.5	Tendências tecnológicas	9
2.1.6	Iniciativas.....	9
2.2	Estrutura de um sistema integrado	11
2.2.1	Processo.....	12
2.2.2	Controlo da Unidade	13
2.2.3	Controlo da célula	15
2.2.4	Sistema de apoio à produção.....	17
2.2.5	Gestão da Empresa.....	21
2.2.6	Gestão das operações de produção	23
2.3	Considerações finais	25
3	Especificação da integração experimental	26
3.1	Estrutura da integração	26
3.2	Objetivos da integração	27
3.3	Simulação do ambiente fabril.....	27
3.3.1	Factory I/O	28
3.4	Ambiente de programação de PLC	30
3.5	Aplicação SCADA	30
3.6	Aplicação MES.....	31
3.7	Aplicação ERP.....	32
4	Desenvolvimento, experimentação e validação	33
4.1	Opentaps	33
4.1.1	Ciclo de processamento da encomenda	33
4.2	cmNavigo	38
4.2.1	Controlo de receitas do produto	39

4.2.2	Configuração da aplicação	40
4.2.3	Automação das atividades do cmNavigo	42
4.3	Factory I/O	54
4.3.1	Emitter	55
4.3.2	Transportador	55
4.3.3	Máquina.....	56
4.3.4	Remove	58
4.4	CODESYS.....	59
4.4.1	Comunicação	59
4.4.2	Programação.....	59
4.5	Vijeo Citect.....	62
4.5.1	Programação modular das unidades e da GUI	62
4.6	Resultados obtidos	67
5	Conclusões	69
5.1	Trabalhos Futuros	70
5.2	Balanço Pessoal	71
6	Referências Bibliográficas	73
Anexos		75
	Aplicações MES.....	75
	Aplicações ERP <i>Open-Source</i>	77

Índice de Figuras

Figura 1.1 - Modelo físico de ambiente fabril da SMC International Training [6]	2
Figura 2.1 - Estrutura da produção	6
Figura 2.2 - Convergência dos sistemas de informação e a automação industrial [7]	7
Figura 2.3 - Modelo simplificado da hierarquia da automação segundo PRM	8
Figura 2.4 - Separação entre receita e equipamento [13].....	8
Figura 2.5 - SmartFactory KL [15].....	10
Figura 2.6 - Relação entre a hierarquia física e de controlo [17]	11
Figura 2.7 - Modelo do processo	12
Figura 2.8 - <i>Radio Frequency Identification</i> [5].....	13
Figura 2.9 - Rota do material	13
Figura 2.10 - Esquema de um controlador.....	14
Figura 2.11 - Exemplo de HMI [5]	15
Figura 2.12 - Unidade com Controlo central	15
Figura 2.13 - Unidade com controlo distribuído.....	16
Figura 2.14 - Áreas do apoio à produção	17
Figura 2.15 - Esquema de sistema <i>Pull</i> e <i>Push</i>	18
Figura 2.16 - Atividades de controlo da produção a realizar no decorrer do processo.....	20
Figura 2.17 - Modelo de comunicação entre o ERP e o controlo da produção	22
Figura 2.18 - Módulos do MOM.....	23
Figura 2.19 - Atividades do MES	24
Figura 2.20 - Componentes da hierarquia da automação.....	25
Figura 3.1 - Estrutura hierárquica da aplicação	26
Figura 3.2 - Simulação fabril do Factory I/O.....	28
Figura 3.3 - Materiais do Factory I/O	29
Figura 3.4 - Unidades de Processamento e Assembly	29

Figura 3.5 - Produtos variados	29
Figura 3.6 - Regra DEE do cmNavigo	31
Figura 4.1 - Troca de informação entre andares	33
Figura 4.2 - Atividades de produção do Opentaps	34
Figura 4.3 - <i>Web-Store</i> e ordem de venda criada	34
Figura 4.4 - Página da <i>Production Run</i> no Opentaps	35
Figura 4.5 - Página final da <i>Production Run</i> no Opentaps.....	35
Figura 4.6 - Informação do BOM no Opentaps	36
Figura 4.7 - Tarefa no Opentaps	36
Figura 4.8 - Rota no Opentaps	36
Figura 4.9 - Comunicação por <i>WebService</i>	37
Figura 4.10 - Estados do Material no Passo	39
Figura 4.11 - Relação entre a <i>Facility</i> , <i>Area</i> e <i>Resource</i>	40
Figura 4.12 - Relação entre <i>Bill of Materials</i> e o <i>Product</i>	40
Figura 4.13 - Relação entre materiais e produtos	41
Figura 4.14 - Esquema do Flow e Step.....	41
Figura 4.15 - Relação entre o <i>Service</i> , <i>Recipe</i> e <i>Parameter</i> e <i>Recursos</i>	42
Figura 4.16 - Comunicação por regra DEE.....	42
Figura 4.17 - Comunicação OPC [35].....	43
Figura 4.18 - Exemplo da falha da ação	43
Figura 4.19 - Ordem de venda	44
Figura 4.20 - <i>Grafcet</i> comportamental da leitura da base de dados.....	44
Figura 4.21 - Linha de texto da ligação à base de dados.....	45
Figura 4.22 - <i>Grafcet</i> comportamental da leitura da base de dados do Opentaps	45
Figura 4.23 - Relatório da comunicação.....	46
Figura 4.24 - Comunicação entre MES e SCADA	47
Figura 4.25 - <i>Grafcet</i> comportamental da Escrita na base de dados.....	48
Figura 4.26 - <i>Grafcet</i> comportamental da escrita na tabela <i>Dispatch</i>	49
Figura 4.27 - <i>Grafcet</i> comportamental da escrita da receita	50
Figura 4.28 - <i>Grafcet</i> comportamental do <i>TrackIn</i>	51

Figura 4.29 - Informação da Instância do <i>Timer</i> do <i>TrackIn</i>	52
Figura 4.30 - <i>Grafcet</i> comportamental do <i>TrackOut</i>	53
Figura 4.31 - <i>Grafcet</i> comportamental do <i>Dispatch</i>	53
Figura 4.32 - Estrutura de teste	54
Figura 4.33 - Unidade de criação do material	55
Figura 4.34 - Sistema de transporte	55
Figura 4.35 - Unidade Máquina	56
Figura 4.36 - <i>Grafcet</i> da Unidade de Processamento.....	56
Figura 4.37 - Material em processamento	57
Figura 4.38 - <i>Grafcet</i> do equipamento.....	58
Figura 4.39 - Projeto do Codesys.....	59
Figura 4.40 - POU do equipamento e código no seu interior.....	60
Figura 4.41 - CFC do cenário de simulação.....	60
Figura 4.42 - POU do equipamento de transporte	61
Figura 4.43 - POU da Unidade de Produção "Máquina".....	61
Figura 4.44 - <i>Template</i> do <i>Equipment</i> Editor e criação de novo equipamento	62
Figura 4.45 - Informação da <i>Tag</i> do equipamento	63
Figura 4.46 - Campos indexados no <i>Genie</i>	63
Figura 4.47 - Criação de um módulo do <i>Genie</i> através do <i>template</i>	63
Figura 4.48 - HMI do SCADA usando <i>Genies</i>	64
Figura 4.49 - Driver ODBC do MySQL	64
Figura 4.50 - <i>Grafcet</i> comportamental da leitura da Receita	65
Figura 4.51 - <i>Grafcet</i> comportamental das funções da máquina.....	65
Figura 4.52 - <i>Grafcet</i> comportamental do Dispatch	66
Figura 4.53 - <i>Grafcet</i> comportamental do TrackOut à esquerda e do TrackIn à direita	67
Figura 4.54 - Estrutura da Integração final.....	67
Figura 4.56 - Sobre carregamento do processador do computador	68

Lista de Tabelas

Tabela 2.1 - Características dos sistemas de controlo e das aplicações de gestão [24]	23
Tabela 4.1 - Colunas da tabela de <i>Dispatch</i>	47
Tabela 4.2 - Tabela Receita	47

Abreviaturas

ANSI	American National Standards Institute
BOM	Bill of Materials
CIM	Computer Integrated Manufacturing
CPS	Cyber Physical System
CRM	Customer Relationship Management
DEE	Dynamic Execution Engine
ECSEL	Electronic Components and Systems for European Leadership
FEUP	Faculdade de Engenharia da Universidade do Porto
FIFO	First In First Out
IEC	International Electrotechnical Commission
ISA	International Society of Automation
LIFO	Last In First Out
MES	Manufacturing Execution Management
MOM	Manufacturing Operations Management
ODBC	Open Database Connectivity
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
PRM	Purdue Reference Model
SCADA	Supervisory Control and Data Acquisition
SCM	Supply Chain Management

1 Introdução

A internet e as redes sociais vieram alterar a maneira como os consumidores compram os seus bens. O fácil acesso à informação, quer de produtos, quer de tendências de mercado fazem com que o consumidor procure a solução que melhor satisfaça todos os seus gostos pessoais [1].

Assim, as empresas defrontam-se com o desafio de responder a um mercado totalmente diferente do existente há duas décadas atrás. Um mercado guiado pelo consumidor, onde não basta uma abordagem tradicional à produção, caracterizada por uma forte procura de eficiência aliada a baixos custos de produção.

A produção eficiente continuará a ser importante no futuro, mas em segundo plano, cedendo a prioridade à capacidade de cumprir os requisitos do cliente através de um nível elevado de agilidade e flexibilidade [1].

Para adquirir estas características, as empresas terão que adotar novas estratégias e tecnologias na sua estrutura organizacional e produtiva. Um ponto fulcral desta nova abordagem à produção é a troca de informação entre sistemas de controlo e a gestão da empresa [2].

A integração dos sistemas de controlo com as aplicações de gestão permite um controlo preciso da produção e fornece aos órgãos de gestão uma visão em tempo real do estado da produção. Este controlo preciso da produção possibilita estratégias baseadas na procura do consumidor (*Make-on-demand*), novas oportunidades de negócio originadas da flexibilização da produção, a otimização das operações da fábrica, redução de desperdício devido a atividades como a manutenção preditiva e preventiva, o rastreio automático da encomenda, desde a matéria-prima proveniente dos fornecedores até à entrega ao cliente [2, 3].

É assim fundamental introduzir estas tecnologias e práticas no mundo académico, com o intuito de adequar e melhorar a formação dos futuros profissionais de engenharia. O ensino ligado à área de automação industrial necessita de estar de acordo com os requisitos do ambiente fabril, pelo que é imprescindível a utilização de modelos próximos da realidade na indústria que possibilitem a introdução de problemas reais para estudo [4].

Modelos físicos que imitam ambientes fabris como o representado na Figura 1.1 são considerados os meios mais interessantes de ensino na área de automação industrial pois permitem aos alunos enfrentar problemas reais, com equipamentos encontrados na indústria, com os quais podem interagir, adquirindo experiência prática no desenvolvimento de soluções automáticas. No entanto, estes são elementos dispendiosos e ocupam demasiado espaço [4].

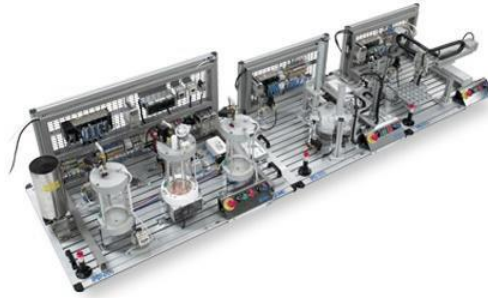


Figura 1.1 - Modelo físico de ambiente fabril da SMC International Training [6]

Assim, em alternativa surge a simulação de um ambiente fabril em computador. É neste contexto que surgiu a ideia de realizar uma aplicação de integração vertical num simulador de ambiente fabril, com o objetivo de lecionar estas novas estratégias e tecnologias num laboratório ou sala de aula.

1.1 Objetivos

Esta dissertação consiste no desenvolvimento, na sala de aula, de uma integração de aplicações de automação industrial com aplicações de gestão da produção, semelhante ao encontrado numa moderna fábrica automatizada.

Pretende-se com esta integração demonstrar, de uma forma didática e interativa, as características dos sistemas integrados de controlo e gestão, recorrendo para tal a um simulador de ambiente fabril, o Factory I/O, em vez de um modelo físico.

Para que a estrutura da integração seja semelhante aos sistemas reais, esta deverá ser constituída por uma aplicação de controlo de autómatos, controlo de supervisão, gestão da produção e gestão da empresa.

A integração deve tirar partido dos recursos existentes na sala de aula e obter os recursos em falta, onde o critério principal de escolha deve ser o custo da solução.

1.2 Organização da tese

O presente texto encontra-se dividido em 5 partes, incluindo esta secção introdutória.

No capítulo 2 é exposto o estado da arte dos sistemas integrados de controlo e gestão na indústria, onde são expostos todos os conceitos técnicos necessários para a compreensão da presente dissertação.

O capítulo 3 é dedicado à apresentação da estrutura da integração na sala de aula, onde são expostas as aplicações informáticas implementadas. Neste capítulo é incluído também uma breve introdução a simuladores, em geral, e o seu interesse no sector industrial.

No capítulo 4 é apresentada a integração desenvolvida na dissertação, onde são expostos os métodos de comunicação utilizados para integração das aplicações, a informação trocada, a configuração realizada em cada aplicação e resultados obtidos.

Finalmente, o capítulo 5 corresponde à sumula das conclusões pertinentes de todo o trabalho desenvolvido, bem como algumas propostas para enriquecer futuramente o trabalho desenvolvido.

2 Sistemas Integrados de Controlo e Gestão

A integração vertical na indústria é uma área multidisciplinar abrangente, que une a gestão à automação industrial através de diversas aplicações informáticas. Consequentemente, o estudo e ensino da integração vertical coloca grandes dificuldades e desafios ao meio académico.

Adicionalmente fontes bibliográficas que descrevam, de uma forma educativa, a globalidade do sistema integrado, atualmente aplicado na indústria, são aparentemente muito raras, sendo a informação existente bastante especializada e orientada para a indústria. Esta informação é de difícil compreensão, sendo vocacionada para profissionais muito experientes, e muitas vezes restrita no sentido em que só está disponível a membros de determinadas organizações.

Muito provavelmente por isso, sente-se uma falta de engenheiros com conhecimento e visão global do projeto de integração, capazes de retirar a maior vantagem dos sistemas integrados e criar soluções inovadoras na indústria através da integração das várias áreas.

O ponto de partida da dissertação foi o estudo exaustivo da integração vertical aplicada atualmente na indústria de modo a apresentar de forma clara em que consiste um sistema integrado de controlo e gestão.

Alguns da informação recolhida na dissertação encontrava-se reservada como os documentos da MESA (*Manufacturing Enterprise Solutions Association*). O contacto com esta entidade permitiu o livre acesso aos seus artigos através de uma *Non-Profit Educational Premium Membership* atribuída à Universidade do Porto [5].

No próximo ponto é exposta uma visão global dos sistemas de controlo com aplicações de gestão, aplicados atualmente na indústria. As temáticas abordadas estão organizadas da seguinte forma:

- Integração Vertical - definição de uma integração vertical, sua evolução e características;
- Estrutura dos sistemas integrados - apresentação dos constituintes dos sistemas integrados e respetivas características.

2.1 Integração Vertical

2.1.1 Estrutura da produção

A produção pode ser definida como o **processo** de combinar e transformar matéria-prima com o objetivo de criar um **produto** com valor acrescentado[6]. A estrutura da produção é representada na Figura 2.1. Ao conjunto de equipamentos e recursos humanos que realizam fisicamente o processo dá-se o nome de **sistema de produção** e ao local onde estes se encontram de **fábrica** [6].

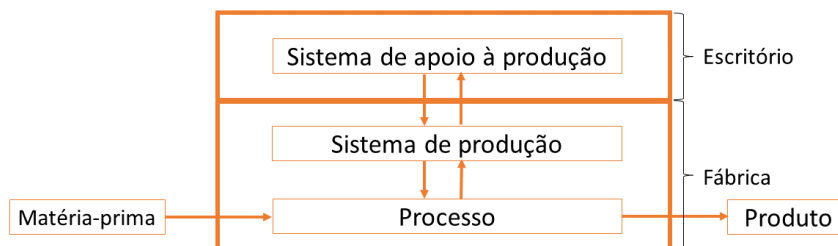


Figura 2.1 - Estrutura da produção

Para que a produção ocorra com sucesso é necessário proporcionar aos sistemas de produção a **informação** e conhecimento necessário para estes realizem as suas atividades. Ao conjunto de equipamentos e recursos humanos que proporcionam essa informação dá-se o nome de **sistemas de apoio à produção** e estes encontram-se em ambientes de **escritório** [6].

2.1.2 Automação na indústria

A **automação** é definida como a tecnologia pela qual um processo é realizado sem intervenção humana [1].A automação de uma fábrica é tradicionalmente separada em duas áreas [6]:

- A **automação industrial**, na qual se automatizam sistemas de produção.
- Os **sistemas de informação**, que informatizam os sistemas de apoio à produção.

Nos sistemas de produção modernos as duas áreas sobrepõem-se, como se demonstra na Figura 2.2, visto os sistemas de produção incluírem, na sua estrutura, dispositivos capazes de comunicar com aplicações informáticas e as aplicações dos sistemas de informação realizarem atividades em tempo real que influenciam a produção [7].

Uma integração vertical não é nada mais do que o estabelecimento de uma comunicação entre estas duas áreas permitindo a interação e cooperação entre estas.

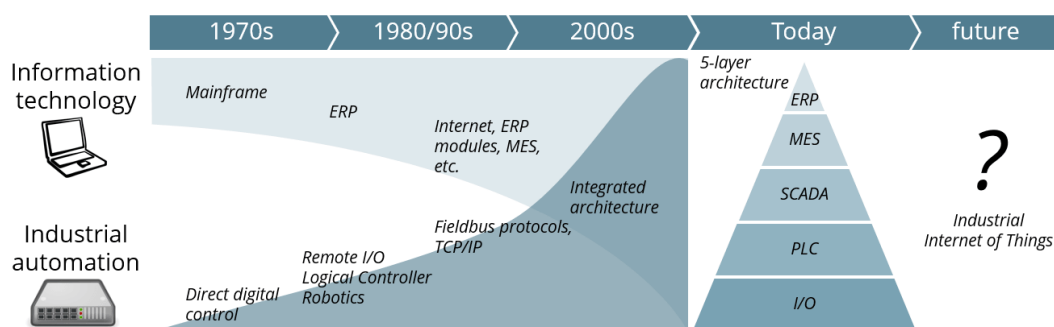


Figura 2.2 - Convergência dos sistemas de informação e a automação industrial [7]

2.1.3 Computer Integrated Manufacturing

A ideia da integração dos sistemas de produção automatizados com os sistemas de informação origina do conceito **CIM (Computer integrated manufacturing)**. Uma definição dada pela Digital Equipment Corporation de CIM é “a aplicação da tecnologia de computadores na estrutura da empresa de forma a dar a informação certa no lugar certo à hora certa cumprindo os objetivos da produção do processo e do negócio” [8].

As primeiras implementações de CIM, nas décadas de 1980 e 1990, criaram fábricas altamente automatizadas de onde se previa um processo completamente autónomo com melhorias significativas aos níveis da produtividade, qualidade e rendimento.

No entanto, muitas destas implementações, posteriormente denominadas de “ilhas de automação” falharam, devido a limitações nas tecnologias de informação e comunicação nas fábricas e a uma flexibilidade limitada dos sistemas de produção [9].

2.1.4 Normas aplicáveis à integração de sistemas

Em 1988, o Dr. Theodore Williams da universidade de *Purdue* criou o modelo de referência de *Purdue (Purdue Reference Model)*, que propõe uma estrutura modelo de aplicações de controlo e de sistemas de informação de modo a flexibilizar e modularizar o CIM.

Uma das maiores contribuições do PRM é a representação hierárquica da automação na empresa apresentada na Figura 2.3, onde o controlo dos equipamentos automáticos encontra-se nos níveis inferiores da hierarquia, a gestão da produção num nível intermédio e as aplicações de gestão na empresa no nível superior [9].

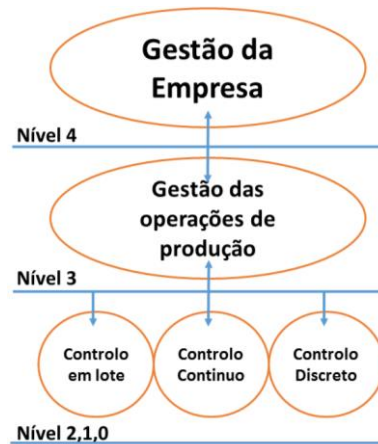


Figura 2.3 - Modelo simplificado da hierarquia da automação segundo PRM

O modelo PRM e o CIM são as bases para os modelos atuais de integração existentes na norma ISA-95 (*ANSI/ISA-95 Enterprise-Control System Integration*) e na IEC 62264 (*IS/IEC 62264 Enterprise-Control System Integration*)[10]. Esta norma, referida frequentemente apenas como ISA-95, visa normalizar a comunicação entre os níveis de **gestão da empresa** e **gestão das operações de produção** (*Manufacturing Operations Management*). Foi desenvolvida para ser aplicada em todo o tipo de processos (em lotes, contínuos e discretos).

A norma ISA-95 veio complementar a ISA-88 (*Batch-Control*), que apresenta as bases para uma solução de automação flexível para os sistemas de controlo em lotes (*Batch*) na indústria [11].

Um dos conceitos base introduzidos na norma ISA-88 é a necessidade de separar a programação do equipamento, das instruções de produção [11]. Através desta separação representada na Figura 2.4 é possível produzir produtos variados sem necessitar de parar o equipamento para ser reprogramado pois a informação do produto é externa ao equipamento e enviada em tempo-real [12].

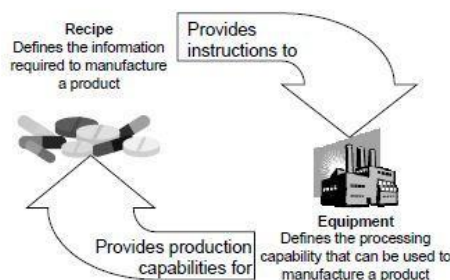


Figura 2.4 - Separação entre receita e equipamento [13]

A informação acerca de cada produto originada pelos níveis superiores da hierarquia de automação, sobre a forma de uma receita (*Recipe*), descreve o que é necessário à máquina para fabricar o produto. Os níveis superiores recolhem também, através da comunicação com a máquina, relatórios do estado da produção, em tempo real.

Esta abordagem nova à programação do equipamento foi impulsionada pela norma IEC 61131-3 que introduziu novas linguagens de programação e apresentou conceitos essenciais para a programação de PLCs (*Programmable Logic Controller*) [13].

2.1.5 Tendências tecnológicas

Aliada a uma evolução dos modelos CIM e à introdução de normas assistidas também a uma evolução nas tecnologias aplicadas na indústria; equipamentos e instrumentação inteligente, que possuem microprocessadores na sua estrutura sendo capazes de realizar tarefas autonomamente, recolher mais e melhor informação do processo, realizar autodiagnóstico e previsão de avarias. Estes também são capazes de comunicar digitalmente, permitindo, assim, unir toda a troca de dados na mesma rede de comunicação [14].

As normas industriais abertas padronizaram os interfaces de comunicação de dados em tempo real possibilitando a comunicação entre dispositivos variados e de fabricantes diversos. Estas comunicações abertas permitem, assim, integrar todos os equipamentos existentes na fábrica numa rede industrial que é independente de fabricantes dos equipamentos sendo possível desenvolver soluções de automação com a melhor relação custo-benefício [14].

A informação recolhida pela rede industrial é aproveitada por aplicações de apoio à decisão, que armazenam e processam essa informação. Estas aplicações especializadas permitem a visualização da informação recolhida do processo, do inventário e do plano de produção em tempo real a gestores dando a estes uma visão verdadeira e em tempo real do estado da produção de uma forma compreensível, facilitando a tomada de decisões [14].

2.1.6 Iniciativas

As novas tecnologias aplicadas na indústria aliadas aos novos modelos de integração distanciam os sistemas automáticos integrados atuais das antigas implementações CIM. Existem atualmente várias iniciativas a nível nacional e internacional de empresas e governos que procuram desenvolver as fábricas do futuro, fábricas capazes de comunicar com clientes e fornecedores automaticamente, controlar e supervisionar todo o seu processo, através da integração do seu sistema de produção com as aplicações de gestão da empresa [15].

Estas iniciativas incluem a *Advanced manufacturing Partnership 2.0*, *Industrial Internet* e *CPS* nos Estados Unidos da América. *Industry 4.0* na Alemanha. *ECSEL, Factories of the Future* na União Europeia, e “*Monozukuri*” no Japão. Outras iniciativas incluem *Wise-ShopFloor* para tecnologias de sensores *web-based* e o *Cyber Physical European Roadmap and Strategy*. [16]

A título de exemplo, na área da medicina atualmente utilizam-se próteses de joelho produzidas em massa que requerem vários dias para personalizar cada prótese ao paciente. A Siemens implementou uma solução de produção que permite integrar a informação das medições realizadas no diagnóstico médico com as instruções de produção das próteses permitindo que a produção do implante demore entre 3 a 4 horas [15].

No conjunto destas iniciativas foram também desenvolvidas demonstrações de fábricas que permitem apresentar as vantagens da aplicação do estado de arte das tecnologias da informação na indústria como a SmartFactory KL, apresentada na Figura 2.5, desenvolvida pelo *German Research Center For Artificial Intelligence* [15].



Figura 2.5 - SmartFactory KL [15]

Esta fábrica de demonstração produz um sabão líquido personalizado, num sistema de produção por lotes, em que cada produto é personalizado de acordo com as especificações de cada cliente. Esta aplicação demonstra um sistema de produção flexível e modular no qual os componentes do produto são transportados, processados e montados automaticamente e autonomamente. Todo o processo de produção é controlado pela informação do produto.

Pretendemos, com esta dissertação, desenvolver uma integração na sala de aula, para fins educativos, capaz de apresentar as mesmas características de um sistema de produção flexível e modular, completamente automatizado no qual cada produto produzido esteja de acordo com o especificado pelo cliente.

2.2 Estrutura de um sistema integrado

Após uma introdução aos sistemas de produção integrados vamos expor a estrutura destes de acordo com a norma ISA-95. Serão apresentados os constituintes de cada nível de automação, as suas características e atividades desempenhadas, bem como conceitos fundamentais para a compreensão destes sistemas.

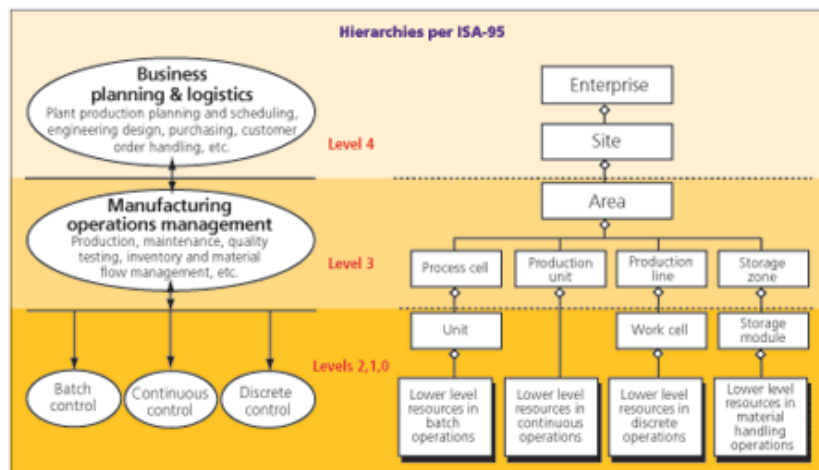


Figura 2.6 - Relação entre a hierarquia física e de controlo [17]

A hierarquia de controlo, segundo a norma ISA-95, estabelece 5 níveis de controlo na empresa. Estes apresentam uma correlação com a estrutura física da empresa, como o apresentado na Figura 2.6 e são os seguintes:

- O nível 0 é o nível do **processo** de produção - é o nível a ser controlado e é constituído por dispositivos de atuação e sensorização;
- O nível 1 é o controlo da **unidade** - é o nível que controla cada unidade de produção, sendo esta entendida como o menor equipamento para o qual é gerada uma receita por parte da gestão da produção;
- O nível 2 é o controlo da **célula** - é o nível que coordena e supervisiona todas as unidades dentro de uma célula de produção, que é definida como o agrupamento lógico de unidades fisicamente próximas;
- O nível 3 é a gestão das operações de produção ao nível da **fábrica** - é o nível que controla e monitoriza em tempo real as operações de produção das várias células presentes na fábrica;
- O nível 4 é a gestão ao nível da **empresa** - é o nível das aplicações informáticas que realizam todas as funções assíncronas (que não decorrem em tempo real) necessárias para a produção.

De seguida são expostos os conceitos fundamentais de cada nível da hierarquia necessários para a compreensão das características de um sistema integrado de controlo e gestão.

2.2.1 Processo

Segundo a norma ISA-88 o processo é dividido em várias secções como apresentado na Figura 2.7 [11]. Analisando, como exemplo, o processo de produção de uma palete de madeira esta pode ser dividida em várias etapas, designadamente a etapa secagem da matéria-prima, a etapa de corte da matéria-prima em tábuas e a etapa de empilhamento das várias tábuas de modo a criar a palete.

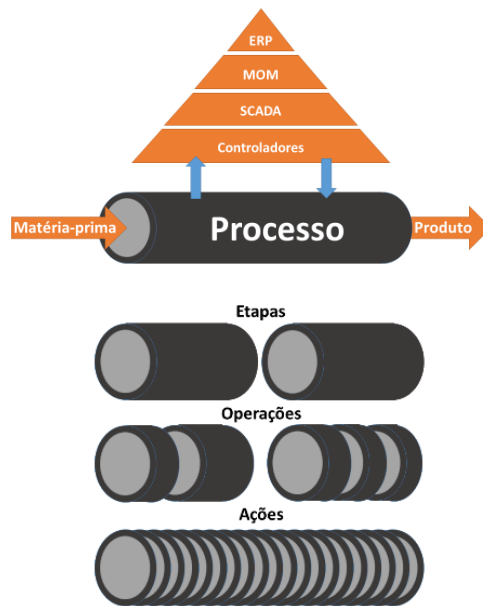


Figura 2.7 - Modelo do processo

Cada etapa do processo envolve um conjunto de **operações** no material. Estas operações podem ser [6]:

- **Operações de processamento** físicas ou químicas para alterar geometria, propriedades e/ou aparência do material;
- **Operações de montagem** (*Assembly*) de vários componentes num só.

Existem também outras operações necessárias para a realização da produção mas que não afetam o produto, sendo estas **operações de manipulação** do material. Operações de manipulação do material incluem: o **transporte**, o **armazenamento** e o **posicionamento** do material ao longo do processo.

Para a realização de cada operação é necessário realizar um conjunto de **ações** como por exemplo: a operação de pregar numa tábuas da paletes que envolve as ações de agarrar na tábuas, posicionar a tábuas, aproximar a ferramenta, pregar e afastar a ferramenta.

2.2.1.1 Automação das operações

O nível do processo é constituído por dispositivos que realizam sobre o processo de produção atividades de recolha de informação e de manipulação das operações de produção.

2.2.1.1.1 Atuadores

Atuadores são os elementos que executam fisicamente a **ação de controlo**, podendo estes ser elétricos, pneumáticos e hidráulicos. Exemplos de atuadores são: motores elétricos, resistências térmicas, válvulas, cilindros pneumáticos, etc.

2.2.1.1.2 Sensores

Sensores são dispositivos que respondem, com um sinal elétrico, a um estímulo. Os estímulos são propriedades físicas, quantidades ou condições de natureza acústica, biológica, química, eletromagnética, ótica, mecânica, radiação ou térmica [17].

Um tipo de sensor essencial nos sistemas integrados são os leitores da identificação do material. Estes elementos são leitores de códigos de barras. Recentemente têm vindo a ser substituídos pela identificação por rádio frequência RFID (*Radio Frequency Identification*), apresentada na Figura 2.8, por ser mais adequada à leitura automática da informação, uma vez que pode ser lida sem a necessidade de contacto direto com o material e sem problemas com a orientação da etiqueta [18].

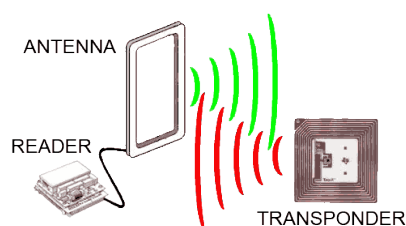


Figura 2.8 - *Radio Frequency Identification* [5]

2.2.2 Controlo da Unidade

Toda a matéria-prima que dá entrada no processo de produção, mas cuja produção está incompleta (*Work in progress*), será denominada ao longo da dissertação de **material**.

Uma unidade realiza um ou mais operações sobre o material, ao conjunto de operações realizadas numa unidade dá-se o nome de passo. Ao percurso de passos pelo qual o material transita ao longo do seu processamento dá-se o nome de rota, como ilustra a Figura 2.9.

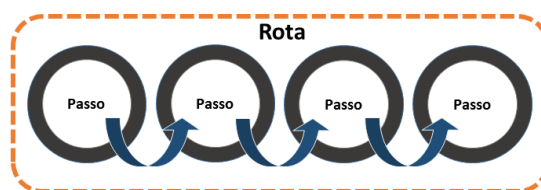


Figura 2.9 - Rota do material

Ao contrário dos computadores normais estes são robustos e construídos para suportar ambientes industriais, sendo resistentes a condições adversas como: vibrações mecânicas, humidade, sujidade ou interferências magnéticas [19]. No entanto, é possível desempenhar as mesmas funções de um PLC num computador normal, sendo este definido como um **SoftPLC**.

2.2.2.2 HMI

Um Interface Homem Máquina ou **HMI** (*Human Machine Interface*) é qualquer dispositivo visual com uma aplicação que permita a comunicação de um humano com o equipamento ou máquina, através de uma **GUI** (*Graphic User Interface*) como o apresentado na Figura 2.11. As HMIs podem-se encontrar em todos os níveis da hierarquia e à medida que aumenta a troca de informação na fábrica, estes dispositivos ganham um papel relevante.



Figura 2.11 - Exemplo de HMI [5]

2.2.3 Controlo da célula

De modo a que o processo ocorra sem imprevistos é necessário um sistema de controlo que coordene, comunique e monitorize o processo de várias unidades, em **tempo-real**.

Uma solução para esta necessidade é a ligação de várias unidades a um só controlador, como representado na Figura 2.12, denominando-se **controlo central**. No entanto, o controlo central requer um dispositivo com elevada capacidade de processamento capaz de controlar todas unidades. Esta solução torna-se muito dispendiosa em virtude do custo de equipamento e cablagem, ocorrendo também problemas de ruído na comunicação [20].

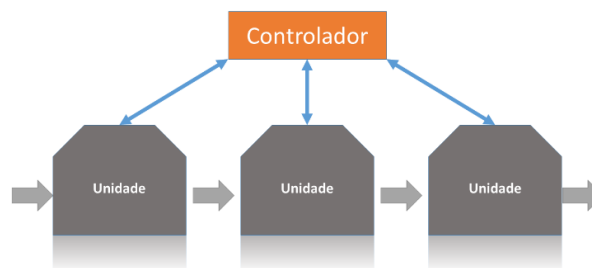


Figura 2.12 - Unidade com Controlo central

Surge, então, a solução de um **sistema de controlo distribuído (DCS)** no qual o processo da célula é controlado por um equipamento supervisor, representado na Figura 2.13, existindo também um controlador por cada unidade. Esta solução apresenta uma estrutura facilmente

expansível, devido à sua modularidade, permitindo a criação de redundâncias da comunicação em rede dos equipamentos [20].

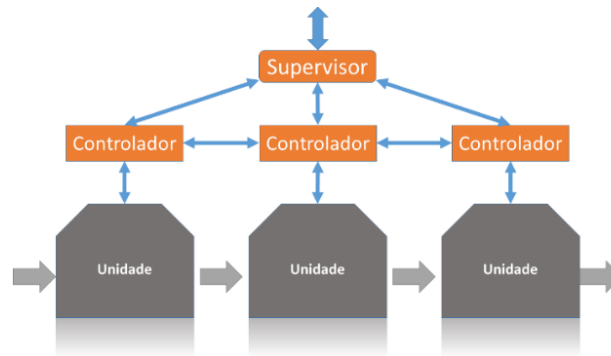


Figura 2.13 - Unidade com controlo distribuído

Associados ao controlo de supervisão encontram-se as aplicações **SCADA** (*Supervisory Control and Data Acquisition*)

As funcionalidades base destes sistemas são as seguintes [20]:

- Aquisição de informação da instrumentação e equipamentos no chão de fábrica;
- Processamento da informação do chão de fábrica para detetar alarmes ou alterações significativas no processo;
- Proporcionar uma base de dados com informação do processo;
- Apresentar a informação através de GUIs que apresentam relatórios, estatísticas, alarmes de uma forma fácil de entender;
- Realizar controlo remoto sobre dispositivos no chão de fábrica;
- Realizar monitorização e diagnóstico do sistema de produção;
- Arquivar o histórico de informação para análise;
- Transferir informação em tempo real para/e de aplicações de nível superior.

Em termos práticos as aplicações SCADA permitem a um **supervisor**, situado numa sala de controlo, **monitorizar** o processo, em tempo-real, e realizar alterações remotamente como: abrir e fechar válvulas e responder a situações de emergência que já ocorreram ou tenham sido detetadas.

2.2.4 Sistema de apoio à produção

O sistema de apoio à produção é constituído pelas atividades do ciclo de processamento de informação que suporta as atividades físicas de produção na fábrica. Estas podem ser agrupadas em 4 áreas de acordo com a Figura 2.14 [6].

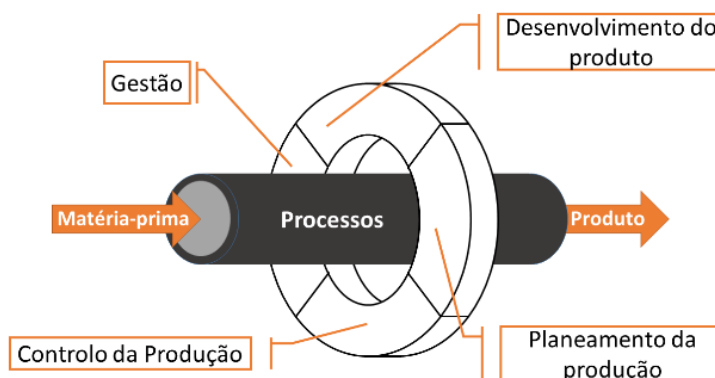


Figura 2.14 - Áreas do apoio à produção

2.2.4.1 Gestão

A área de gestão engloba os departamentos de vendas, contabilidade, finanças, compras e marketing. Esta apresenta-se como o início e o fim do ciclo de processamento de informação, possuindo os meios de comunicação com os clientes e fornecedores. As atividades realizadas na área de gestão incluem:

- Gestão financeira e fiscal da empresa;
- Tomada de decisões estratégicas que afetam toda a estrutura da empresa;
- Encomenda de matéria-prima e qualquer outro recurso a fornecedores;
- Divulgação dos produtos e serviços realizados pela empresa;
- Receção de encomendas por parte dos clientes;

É por aqui que dá entrada, no sistema de apoio à produção, a encomenda do cliente sob a forma de uma **ordem de venda**, no departamento de vendas [6]. Se para satisfazer a encomenda ou as previsões de venda for necessário a produção de produtos adicionais é criada uma **ordem de produção**.

As estratégias adotadas para a criação de encomendas de produção pode variar, tendo essas estratégias repercussões no sistema de produção.

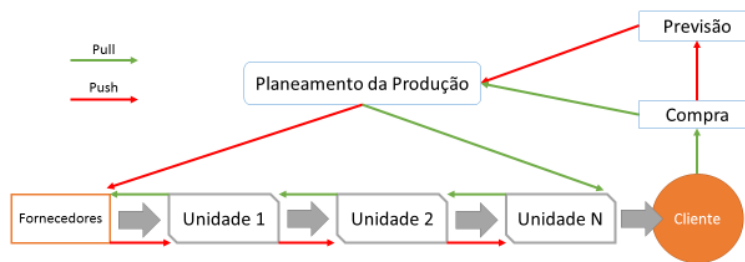


Figura 2.15 - Esquema de sistema *Pull* e *Push*

Num sistema *push* as ordens de produção são planeadas em função das previsões de mercado, originado no fabricante e “puxando” o produto para o mercado. São adequados para produtos com maior garantia de procura. Pode ser visto como produção *make-to-stock* [21].

Num sistema *pull* as ordens de produção tem origem num pedido do cliente, ou seja a informação vem do cliente e puxa o produto pela produção, no sentido contrário ao fluxo de material. Podem ser vistos como produção *make-to-order*. São adequadas para produtos com elevada incerteza de procura [21].

2.2.4.2 Desenvolvimento do produto

Nesta área encontra-se todo o “*Know-how*” da empresa em relação à fabricação dos seus produtos, cujas atividades incluem[22]:

- Conceptualização do produto - onde se elaboram conceitos de novos produtos e se verifica a viabilidade tecnológica e financeira destes conceitos, os mais promissores originam propostas de produto;
- Projeto do produto e processo - onde, após a aprovação da proposta de produto, são definidas as características técnicas do produto e, simultaneamente, o seu método de produção;
- Fabricação de protótipos para o teste e validação dos produtos e processos, antes de lançar o produto para a produção;
- Lançamento do produto para a produção, após aprovação da empresa ou do cliente que requisitou o produto. Fornecimento de toda a informação técnica necessária para o fabrico do produto, ao sistema de produção, como: **desenhos dos componentes, especificações, lista de materiais**, etc;
- Melhoria contínua do produto - onde, através do feedback da produção da empresa ou do cliente, se realizam alterações ao produto de modo a melhorar o seu desempenho ou solucionar problemas não detetados anteriormente.

Esta área, durante o processo de produção, fornece toda a informação necessária do produto para cumprir as ordens de venda provenientes da área de gestão.

2.2.4.3 Planeamento da produção

Com a informação do produto proveniente da área de desenvolvimento e a ordem de produção, são realizadas as funções de planeamento de produção, que envolve os departamentos de engenharia de produção, engenharia do processo e planeamento da produção [6].

As atividades que se realizam nesta área incluem:

- **Planeamento do processo** - consiste em determinar a sequência de passos de produção do produto. Ao realizar esta função é criado o **documento da rota** que lista os vários passos necessários, assim como as unidades associadas a estes passos, para cada material, do produto final;
- **A autorização da produção** - introduz o produto no plano de produção mestre;
- **O plano de produção mestre** - é uma lista constituída por todos os produtos a serem fabricados, contendo as datas de entrega e as respetivas quantidades. A unidade temporal usada no plano mestre é, normalmente, o mês;
- **A requisição de matéria-prima** - processo pelo qual se encomendam os componentes a fornecedores. Todos os componentes e matérias-primas têm de estar disponíveis no momento em que são necessários no chão de fábrica;
- **O planeamento da capacidade** - visto o plano de produção não poder listar uma quantidade de produtos maior do que aquela que a fábrica é capaz de produzir em virtude do seu dado número de máquinas e mão-de-obra;

Esta atividade é uma das principais razões para a integração dos sistemas de controlo, visto ser impossível definir a verdadeira capacidade de produção de uma fábrica sem recolher uma elevada quantidade de informação do processo, periodicamente. A grande quantidade de informação do processo proveniente de um sistema de produção integrado resolve esse problema, otimizando o planeamento e, conseqüentemente, a produção.

2.2.4.4 Controlo da produção

As funções da área de controlo da produção tratam do controlo das operações de produção da fábrica de modo a cumprir o plano de produção. Estas atividades, como é ilustrado na Figura 2.16, realizam-se ao mesmo tempo que decorre a produção do produto (*online*).

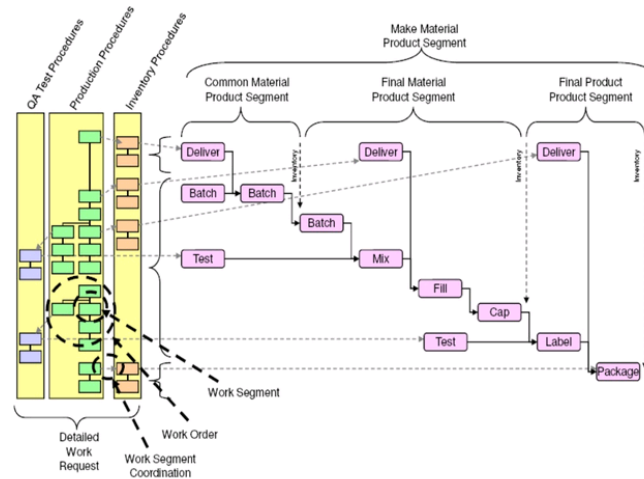


Figura 2.16 - Atividades de controlo da produção a realizar no decorrer do processo

As atividades realizadas por esta área são:

- Um **plano de produção detalhado** diferente do plano de produção mestre, sendo a sua unidade temporal, normalmente, turnos. É assim planeada a produção de cada produto alocando-a aos equipamentos, de acordo com a informação proveniente do documento da rota criada no planeamento da produção;
- O **controlo do sistema de produção** envolve a monitorização do progresso do produto à medida que este vai sendo produzido, montado, movido e inspecionado na fábrica;
- **Ordens de trabalho** baseadas no plano de produção estático, para os operadores das máquinas, onde é indicado o trabalho (operações) que eles vão realizar na peça;
- **Controlo de inventário** que procura otimizar o processo encontrando um balanço entre o risco de ter pouco inventário na fábrica e os custos de ter inventário a mais. Inventário na fábrica também se aplica a peças a serem processadas;
- **Controlo de qualidade** de modo a assegurar que os componentes estão de acordo com as especificações do cliente. Para cumprir esta missão o controlo de qualidade tem que realizar atividades no chão de fábrica em alturas variadas. Também é necessário analisar a matéria-prima que é recebida;
- **Controlo da manutenção** que procura otimizar as operações de manutenção dos equipamentos guardando a informação de todas as operações de manutenção e peças sobresselentes existentes na fábrica. Cria um plano de manutenção dos equipamentos marcando operações de manutenção de acordo com durabilidade dos seus componentes de modo a os substituir antes que a máquina avarie, realizando uma manutenção preventiva.

As atividades do sistema de apoio à produção podem ser automatizadas. No entanto estas não se encontram todas no mesmo nível da hierarquia de controlo, segundo a norma ISA-95, estando distribuídas da seguinte forma:

- As áreas de gestão, desenvolvimento do produto e planeamento da produção realizadas por aplicações do nível de gestão da empresa, sendo estas atividades *offline*;
- A área de **controlo da produção** encontra-se no nível 3 da hierarquia e é realizada por aplicações do nível de **gestão da produção, que correm em tempo real**.

2.2.5 Gestão da Empresa

A integração das atividades dos vários departamentos de gestão numa única aplicação informática traz várias vantagens uma vez que [6]:

- Elimina a necessidade de sincronizar alterações entre vários sistemas, consolidando os departamentos de finanças, marketing, vendas, recursos humanos;
- Traz legitimidade e transparência a cada dado estatístico;
- Facilita a integração ao substituir aplicações diferentes com formatos de comunicação variados;
- Permite uma visão compreensiva do estado da empresa, disponibilizando aos gestores da empresa, informação em tempo real, a qualquer momento, de modo a possibilitar melhores decisões e reagir mais facilmente a mudanças;
- Ao conter toda a informação numa só aplicação melhora a segurança da informação confidencial e o controlo do acesso à mesma;
- Facilita a comunicação não só dentro da fábrica, mas também com aplicações externas.

Esta aplicação, denominada de **ERP** (*Enterprise resource planning*), é usada para gerir todos os recursos da fábrica intangíveis, financeiros, humanos ou materiais [23].

A estrutura do ERP é modular. A norma ISA-95 apresenta um modelo base da troca de informação entre os módulos do ERP com a área de controlo da produção que, na Figura 2.17, se encontra dentro da linha a tracejado. Aqui é possível ver a quantidade elevada de interações entre o controlo da produção e o ERP e, conseqüentemente, a elevada quantidade de informação trocada e processada.

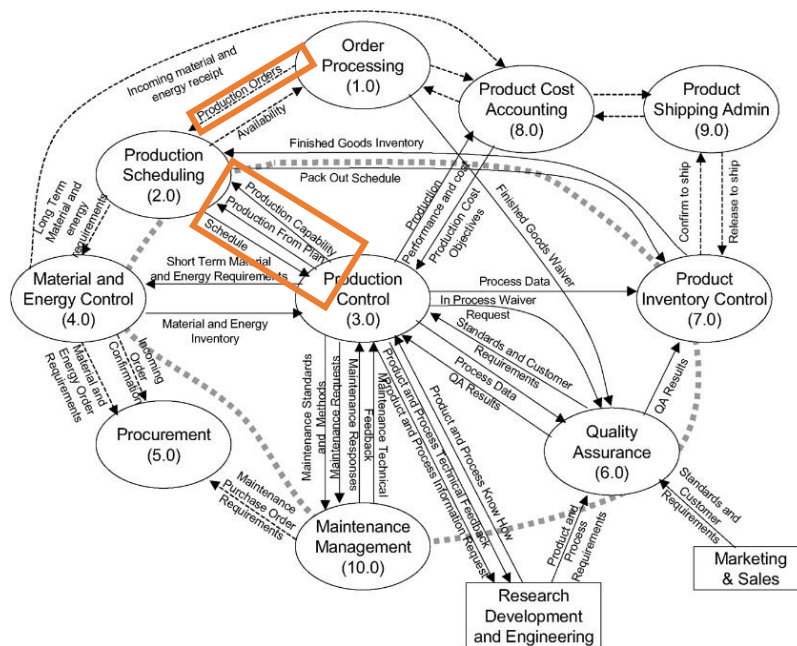


Figura 2.17 - Modelo de comunicação entre o ERP e o controlo da produção

Sendo este um modelo demasiado completo de trocas de informação, a integração a realizar na dissertação irá tratar apenas das trocas de ordens de produção destacadas na Figura 2.17. Esta figura serve apenas como ilustração de modo a melhor se compreender a escala e a complexidade do sistema de produção completamente integrado.

Os módulos representados na norma não apresentam a mesma denominação que os módulos vendidos pelas aplicações comerciais ERP, visto a norma ISA-95 não abranger as aplicações ERP, mas apenas as aplicações de gestão das operações de produção e as suas comunicações com o ERP.

Os módulos relevantes que se encontram no nível de gestão da empresa e que vão para além das atividades base do ERP são [3]:

- O PLM (*Product Lifecycle Management*) trata de toda a informação do produto, ou seja, da área de desenvolvimento do produto.
- O SCM (*SupplyChain Management*) trata de todas as atividades com fornecedores e logística.
- O CRM (*Customer Relationship Management*) trata de todas as atividades com clientes.

2.2.6 Gestão das operações de produção

A razão pelo qual é necessário uma aplicação interface, entre as atividades de gestão e dos sistemas de controlo, é que os requisitos de troca de dados, para as duas áreas, são muito diferentes como representa a Tabela 2.1. As aplicações de gestão da empresa realizam atividades que geram grandes quantidades de informação, cada vez que correm, sendo executadas periodicamente, enquanto os sistemas de controlo necessitam de responder, em tempo real, a variações no sistema.

Tabela 2.1 - Características dos sistemas de controlo e das aplicações de gestão [24]

	Sistemas de controlo	Gestão da Empresa
Operação em tempo-real	Sim	Não
Necessidade de fiabilidade	Elevada	Moderada
Necessidade de Segurança	Não é crítica	Critica
Tarefas realizadas	Controlo de equipamento	Transações entre bases de dados.
Necessidade de comunicação	Comunicação rápida	Processamento de grandes quantidades de informação

A aplicação de gestão das operações de controlo, denominada de MOM (*Manufacturing Operations Management*), age como uma ponte entre estes dois sistemas incompatíveis, comunicando grandes quantidades de informação com as aplicações de gestão e realizando também algumas das suas atividades em tempo real, de forma a conseguir comunicar com os sistemas de controlo e respondendo rapidamente a alterações. Deste modo permite controlar efetivamente a produção e aquisição de informação do processo, que posteriormente é transmitida ao ERP [25]. O MOM, analogamente ao ERP, distribui as suas atividades por módulos, como demonstra a Figura 2.18.

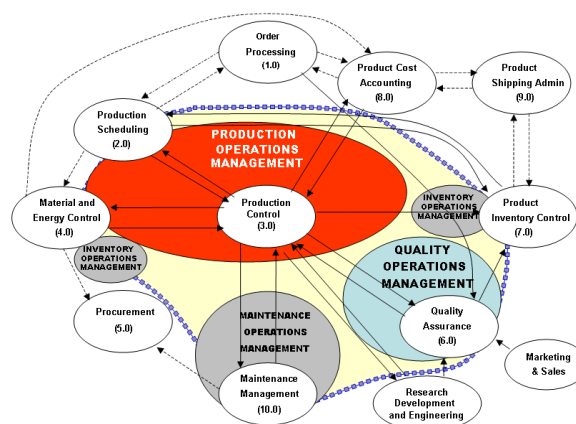


Figura 2.18 - Módulos do MOM

Os módulos definidos pela norma ISA-95, para o nível 3, são os seguintes:

- Gestão das operações de manutenção;
- Gestão das operações de produção;
- Gestão das operações de qualidade;
- Gestão das operações de inventário.

Na presente dissertação focamo-nos no módulo de **gestão das operações de produção**, denominado de MES, ficando as questões de gestão das operações de qualidade, manutenção e inventário fora do seu alcance.

A Figura 2.19 apresenta as atividades do MES, segundo a norma ISA-95 onde as atividades superiores estão mais próximas das atividades de nível 4, processando maior quantidade de informação, mas de forma assíncrona. Do mesmo modo, as atividades inferiores estão mais próximas de atividades de nível 2, decorrendo em tempo-real e comunicando com os níveis inferiores.

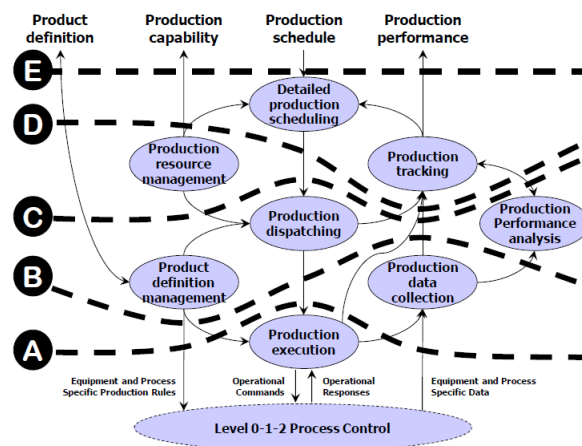


Figura 2.19 - Atividades do MES

O MES comunica com o ERP de modo a receber o plano de produção mestre e criando o plano de produção detalhado (Detailed Production Plan). Através desse plano são criadas ordens de trabalho para cada unidade (Production Dispatching). Essas ordens de trabalho são convertidas em receitas (*Production Execution*) que são enviadas para as unidades, em tempo real, comandando as operações destes.

Os Equipamentos enviam as respostas que o MES guarda na aplicação (*Production data collection*) para atualizar o estado do processo (*Production Tracking*) e realizar análise estatística da performance da produção (*Production Performance Analysis*).

Esta informação em tempo real, do estado da produção, aliada à informação da base de dados do MES, das unidades existentes na fábrica (*Production Resource Management*) e dos produtos fabricados (*Product Definition Management*), realimentam as atividades de controlo

das operações garantindo assim um controlo da produção com a informação mais atualizada possível, proporcionando as melhores decisões para a produção

O MES é uma aplicação interface, e como tal tem que se adaptar aos sistemas que está a integrar, por isso a linha interface entre as atividades do ERP e o MES pode variar como é demonstrado na Figura 2.19 pelas linhas a tracejado A-B-C-D-E.

A integração entre os vários níveis não é sempre igual, tendo esta de se adaptar às aplicações que estão a ser integradas e às suas atividades.

2.3 Considerações finais

Após a análise do estado de arte dos sistemas integrados de produção e gestão é possível definir os componentes de uma fábrica real indispensáveis para um sistema integrado de controlo e gestão, como representa a Figura 2.20.

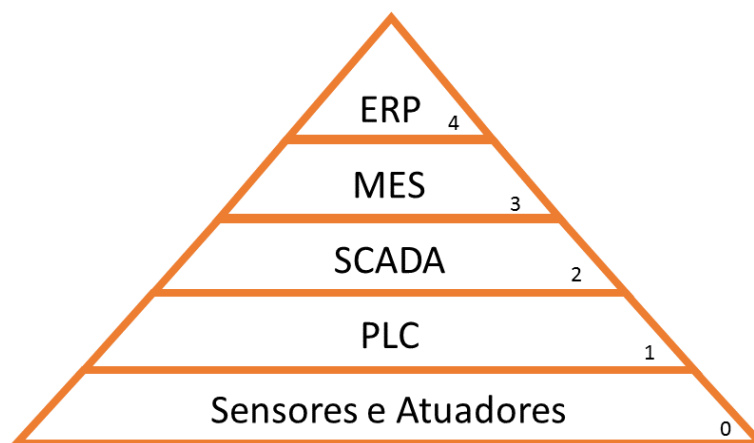


Figura 2.20 - Componentes da hierarquia da automação

Os equipamentos que realizam o processo de produção necessitam de sensores e de atuadores, de forma a controlar o processo.

Este controlo é realizado diretamente nos equipamentos por controladores, sendo os mais comuns os PLCs.

Os controladores de todas as unidades comunicam entre si e são supervisionados por um SCADA.

A informação que especifica as operações a realizar, em cada unidade, tem origem de uma aplicação MES, cumprindo o plano de produção criado pela aplicação ERP, que entre outras atividades interage com os clientes e fornecedores.

3 Especificação da integração experimental

O presente capítulo será dedicado à apresentação da estrutura proposta para a implementação do sistema integrado na sala de aula utilizando um ambiente virtual para simular o sistema automatizado de produção. Também serão apresentadas as respectivas aplicações a serem integradas.

3.1 Estrutura da integração

O plano da estrutura hierárquica da aplicação, em sala de aula, é similar ao de uma fábrica real, diferindo somente nos níveis inferiores.

Uma vez que o ambiente fabril será simulado, os dispositivos presentes na fábrica serão também virtuais (sensores e atuadores).

A utilização de um PLC na sala de aula seria concretizável, mas acarretava custos financeiros. Optou-se pela utilização de um *SoftPLC* num computador da sala, sendo em tudo semelhante a um PLC que esteja de acordo com a norma IEC 61161-3.

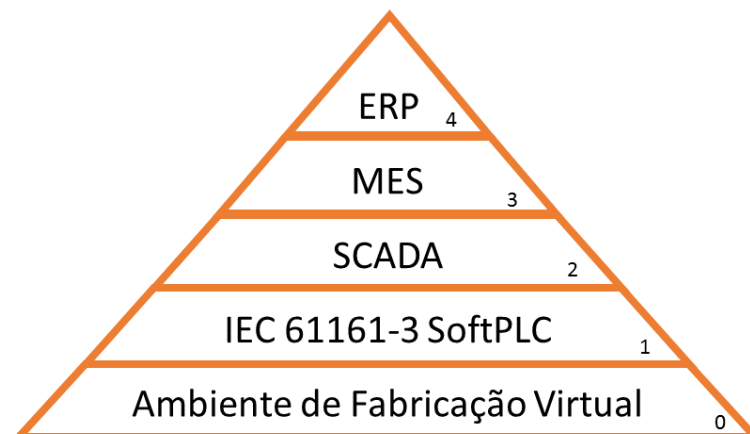


Figura 3.1 - Estrutura hierárquica da aplicação

3.2 Objetivos da integração

O objetivo desta dissertação é desenvolver uma integração vertical na sala de aula recorrendo a um simulador de ambiente fabril.

Por meio desta integração os alunos poderão desenvolver sistemas automáticos de produção que tiram vantagem da comunicação com níveis superiores de controlo e, deste modo, podem adquirir prática na programação de PLCs, na área de automação industrial e na de produção.

Esta integração deve:

1. Permitir a encomenda de produtos através de uma *Web-Store* - Por meio de um catálogo *online* na aplicação ERP, os alunos poderão encomendar produtos, para o ambiente fabril virtual produzir, realizando o interface com os clientes;
2. Introduzir novos produtos sem alterar o controlo direto dos equipamentos - Programar o controlo das unidades para que estas comuniquem, por receitas, com os sistemas de produção, tirando partido da programação de PLCs segundo a norma IEC-61131-3;
3. Produção automática das encomendas - Automação completa da produção, integrando as atividades necessárias para tal, em cada aplicação;
4. Apresentar um controlo da produção flexível - Tirar partido das capacidades do MES de forma a controlar a produção em tempo-real;
5. Garantir o menor custo possível do projeto.

3.3 Simulação do ambiente fabril

Um simulador de treino é uma aplicação gráfica que imita um equipamento ou cenário real onde o utilizador controla o comportamento dos objetos simulados e assiste às suas reações, aprendendo deste modo a controlar o objetos simulados [26].

Um exemplo clássico de uso de simuladores de treino à na área da aviação, onde através de tecnologias usadas nos videojogos é possível aos alunos pilotar aviões simulados. Desta forma é possível ganhar experiência a um custo reduzido, sem pôr em risco a vida do utilizador ou de terceiros, sem causar danos materiais e sem necessitar de grandes espaços para a simulação.

A aplicação de tecnologias de simulação na área da automação industrial permite simular ambientes e equipamentos fabris em computadores, como demonstra a Figura 3.2, onde é possível programar os equipamentos e assistir ao seu comportamento em tempo real [26].



Figura 3.2 - Simulação fabril do Factory I/O

De igual modo, estes simuladores aplicados na área de gestão da produção permitem criar ambientes de produção ideais para testar e desenvolver aplicações de controlo de produção, eliminando o perigo de danos pessoais ou materiais, fornecendo ao programador o feedback visual da programação desenvolvida permitindo assim detetar, rápida e eficiente, erros e conjecturar possíveis soluções para os problemas encontrados.

3.3.1 Factory I/O

O simulador utilizado para a integração é o *Factory I/O* da empresa portuguesa Real Games, capaz de simular sistemas que realizam operações de transporte, *sorting* e manuseamento de materiais em 3 dimensões, para fins didáticos [27].

Este é extremamente interativo e apresenta uma *sandbox*, que permite ao utilizador construir os seus próprios sistemas de automação industriais, possibilitando a criação de uma multiplicidade de cenários, apenas limitados pelos componentes existentes no programa. Adicionalmente a inclusão de um cliente OPC no simulador permite também o seu controlo através de um *SoftPLC*.

As desvantagens desta aplicação é que não apresenta a totalidade dos componentes vulgarmente encontrados em ambientes fabril, oferecendo apenas os equipamentos que realizam operações de manipulação do material, como transportadores, armazéns automáticos, *Pick&Place*. Por outro lado as simulações realizadas pela aplicação estão mais vocacionadas para sistemas de transporte e *sorting*, do que para sistemas de produção. Por último é o facto de só possuir materiais na forma de caixas, apresentadas na Figura 3.3.

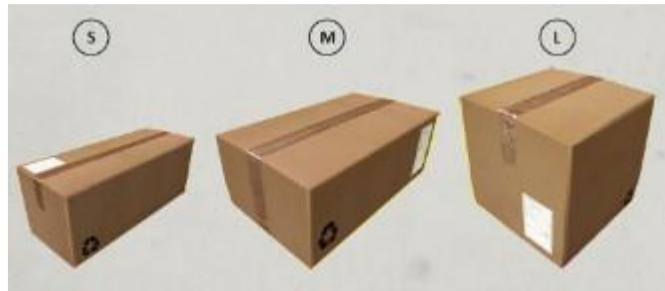


Figura 3.3 - Materiais do Factory I/O

Apesar das limitações referidas, este simulador constitui uma boa ferramenta para a aplicação, visto permitir construir, com bastante liberdade, os seus equipamentos automáticos o que aliado a um pouco de imaginação possibilita o desenvolvimento de unidades de produção capazes de realizarem operações de processamento e *Assembly* fictícias nos materiais, como fornos ou estações de *assembly* apresentadas na Figura 3.4.



Figura 3.4 - Unidades de Processamento e Assembly

Tendo em conta que o objetivo da aplicação é demonstrar a flexibilidade no fabrico de produtos e, embora Factory I/O esteja limitada a apenas 3 modelos de produtos a solução proposta foi ilustrar diferentes variações de produto através das diferentes posições e orientações do conjunto dos três objetos permite recrear diferentes variações de produto como os três exemplos apresentados na Figura 3.5.



Figura 3.5 - Produtos variados

3.4 Ambiente de programação de PLC

O uso de ambientes de programação baseados na norma IEC61131-3 para a programação de PLCs é prática atual na indústria e continuará a ser a nos próximos 5 a 10 anos [28].

Tendo em conta esta premissa faz sentido a aplicação de um ambiente de programação concordante com a norma, pelo que se escolheu o CODESYS da 3S Softwares.

O CODESYS é a aplicação mais usada no mercado de programação independente e baseada na norma possuindo as seguintes características apelativas à aplicação [29]:

- Ferramentas de teste (*debug*) permitindo o controlo e monitorização, em tempo-real, de variáveis, o que facilita a deteção de possíveis erros e a sua correção;
- Contém um *SoftPLC* permitindo a utilização dos computadores na sala de aula como controladores reais;
- Uso das linguagens de programação da norma;
- Permite a estruturação do código em módulos, POUs (*Program Organization Unit*).

Nos vários problemas de controlo foi possível verificar que certos aspetos lógicos se repetem, conduzindo à repetição do seu algoritmo.

A simples cópia de código não é uma boa prática de programação, podendo estes códigos ser inseridos em módulos, POUs, que podem ser chamados quando necessários. Para além do mais, caso seja detetado algum *bug* na secção de código repetida, basta corrigir a POU e não em todos os lugares onde o código se encontra, evitando-se erros humanos que podem custar milhões de euros à empresa [29]. Estas POUs também permitem o encapsulamento do código no seu interior com garantia de que este não interfere com outras secções de código.

Para além de responder a todas as necessidades do projeto esta aplicação é gratuita e já se encontrava na sala de aula.

3.5 Aplicação SCADA

Vijeo Citect é uma aplicação SCADA que permite a um supervisor gerir e monitorizar sistemas de produção ou processos através de uma HMI. Esta aplicação tem a capacidade de comunicar com vários PLCs através de vários protocolos industriais.

As variáveis comunicadas com estes equipamentos intitulam-se de *Tags*. Ao referenciar estas *Tags* na HMI é possível apresentar a informação destes equipamentos ou alterar essa informação controlando assim o equipamento através da HMI ou das funções lógicas do programa. É possível desenhar a GUI, de modo a representar a unidade de trabalho, através da ferramenta *Citect Graphics Builder*.

Apresenta uma linguagem de programação própria, o CICODE, com uma estrutura semelhante a Pascal. Esta aplicação apresenta uma versão gratuita.

Uma vez que esta aplicação se encontrava instalada na sala de aula e satisfaz bem os requisitos exigidos foi eleita para a integração.

3.6 Aplicação MES

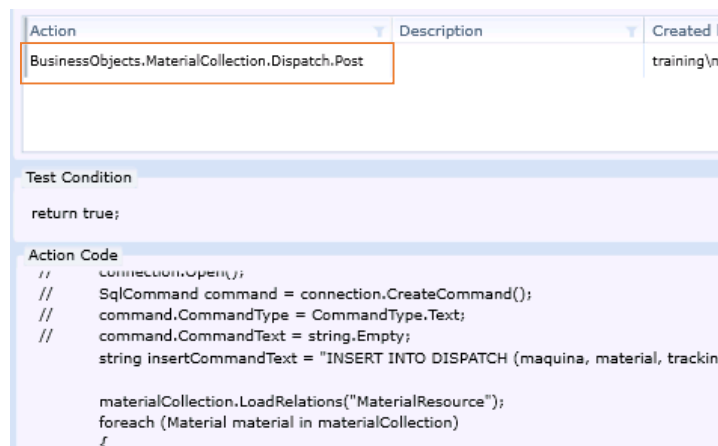
O estudo de mercado das aplicações MES iniciou-se pela procura de soluções gratuitas ou *open-source*, de modo a satisfazer o requisito de baixo custo da integração em sala de aula. As aplicações *open-source*, por definição, disponibilizam publicamente o seu código. Desta forma abrem as portas à colaboração dos interessados [30].

Atualmente a única aplicação MES funcional é o qcadoo, contudo a informação disponibilizada acerca da sua utilização e/ou instalação é muito reduzida. Em alternativa, foram encontradas 73 soluções comerciais que se anexam ao presente trabalho [31].

Na análise das soluções existentes deparamo-nos com a aplicação cmNavigo da Critical Manufacturing que se localiza na Maia. Devido à sua proximidade geográfica realizaram-se contactos com esta empresa, para apresentação do presente projeto. Em resultado deste contacto e respetiva apresentação a Critical Manufacturing cedeu generosamente a aplicação, toda a informação sobre a mesma e apoio técnico, que se revelou imprescindível.

O cmNavigo é uma aplicação bastante completa que pode ser considerada uma aplicação MOM, pois realiza atividades que vão para além de um MES.

Esta aplicação permite a sua personalização às especificações do cliente, visto ser fácil a sua personalização através de Ações DEE (*Dynamic Execution Engine*). Estas Ações DEE são criadas pelo utilizador e são essencialmente código em C#, que corre após atividades do cmNavigo (apresentada no retângulo na Figura 3.6) e permitindo ao utilizador programar a aplicação de forma simples. Deste modo é possível realizar integrações personalizadas com outras aplicações e automação das atividades do cmNavigo.



Action	Description	Created i
BusinessObjects.MaterialCollection.Dispatch.Post		training\m

```
Test Condition
return true;

Action Code
// connection.Open();
// SqlCommand command = connection.CreateCommand();
// command.CommandType = CommandType.Text;
// command.CommandText = string.Empty;
string insertCommandText = "INSERT INTO DISPATCH (maquina, material, trackin

materialCollection.LoadRelations("MaterialResource");
foreach (Material material in materialCollection)
{
```

Figura 3.6 - Regra DEE do cmNavigo

3.7 Aplicação ERP

A análise de mercado dos ERPs revelou a existência de três grandes fornecedores destas soluções. Em primeiro lugar aparece a SAP, seguida da ORACLE e depois a Microsoft. Os maiores projetos dos diferentes setores económicos utilizam uma aplicação destes fornecedores [30].

No entanto, os competidores oferecem soluções mais especializadas e buscam nichos de mercado. São cerca de 20 fornecedores de ERPs, de nível médio, tais como a Epicor, Sage, Infor [30].

Por último existem vendedores que se focam nas empresas pequenas, vendendo soluções mais baratas. Este é um mercado que tem vindo a crescer à medida que mais empresas procuram informatizar as suas atividades de gestão. Entre estes vendedores encontram-se por exemplo a Exact, Globe, Syspro, NetSuite [32].

Analisando o mercado das soluções ERP *open-source* o cenário difere das aplicações MES, uma vez que existe uma vasta gama de aplicações. O levantamento das aplicações ERP *open-source* existentes é apresentado no Anexo 2.

Foi selecionada a aplicação *Opentaps* pela sua maturidade e boa documentação, comparativamente com outras, e pelos seus *case studies* apresentados com a sua aplicação na Toyota e Honeywell [33].

A aplicação *Opentaps* apresenta uma *Web-store* onde é possível realizar encomendas, podendo ser acedida em qualquer computador na sala de aula e guardando a sua informação numa base de dados MySQL *open-source*.

Ao longo do trabalho viemos a constatar que a questão da documentação se revelou um fator preponderante numa aplicação que se pretenda implementar sem o auxílio técnico de empresas.

Concluído o processo de seleções de todas as aplicações necessárias para a integração, passou-se à sua instalação na sala de aula e procedeu-se ao desenvolvimento da integração.

4 Desenvolvimento, experimentação e validação

Após a definição da estrutura da integração e seleção das aplicações constituintes, passou-se à criação das comunicações entre os níveis da hierarquia, de modo a integrar as suas atividades. No capítulo 2 especificou-se os requisitos de informação que cada aplicação requer, estando estes ilustrados na Figura 4.1.

O Opentaps fornece a informação do plano de produção ao cmNavigo que com essa informação cria as receitas de controlo dos equipamentos. Estas receitas são transferidas, pelos sistemas de supervisão SCADA, para os respetivos equipamentos que executam as instruções na receita. Concluído este processo é enviada a informação da operação para os níveis superiores.

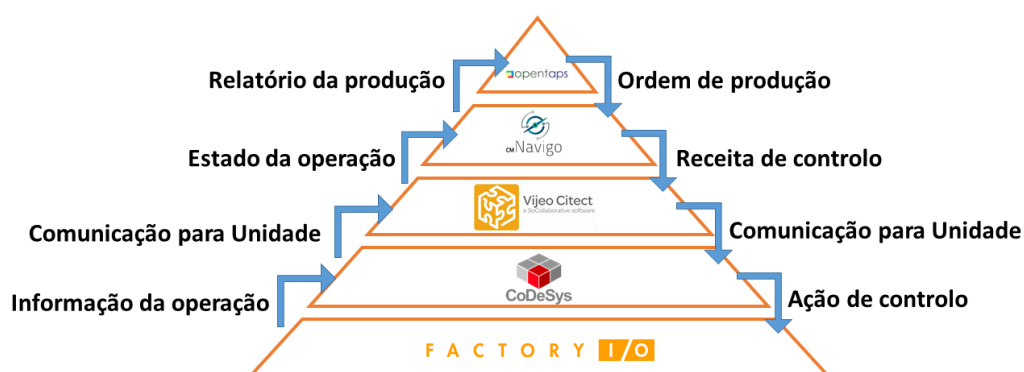


Figura 4.1 - Troca de informação entre andares

4.1 Opentaps

Cada integração varia em função dos requisitos do projeto, das aplicações e comunicação utilizadas. Antes de integrar as aplicações na sala de aula realizou-se uma análise prévia das características das aplicações escolhidas para os níveis superiores, o ERP e o MES, pois sem o conhecimento básico de cada aplicação a integrar seria impossível definir, em concreto, a troca de informação entre estes níveis.

4.1.1 Ciclo de processamento da encomenda

A Figura 4.2. explicita o seguimento do ciclo de processamento da informação de uma encomenda no Opentaps e as atividades que envolvem este processamento, na aplicação. O ciclo tem início com a criação da encomenda, por parte do cliente, na *Web-Store*. Nesta são selecionados os produtos a serem adicionados à lista de compras e quando o cliente decidir realizar o *CheckOut*, esta lista transforma-se numa ordem de venda.

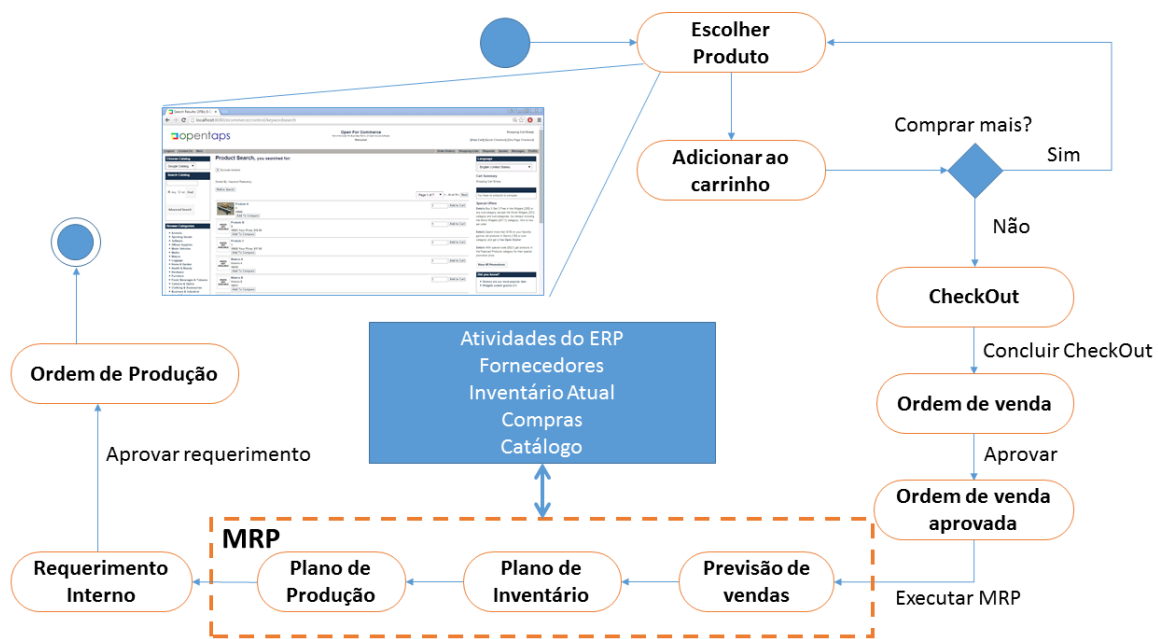


Figura 4.2 - Atividades de produção do Opentaps

Esta ordem de venda, apresentada na Figura 4.3, faz parte do módulo de finanças e aguarda pela aprovação de um responsável das finanças. Este ao concretizar a aprovação da ordem de venda possibilita que esta seja processada pelo MRP (*Material Requirements Planning*), que automatiza as atividades de planeamento da produção.

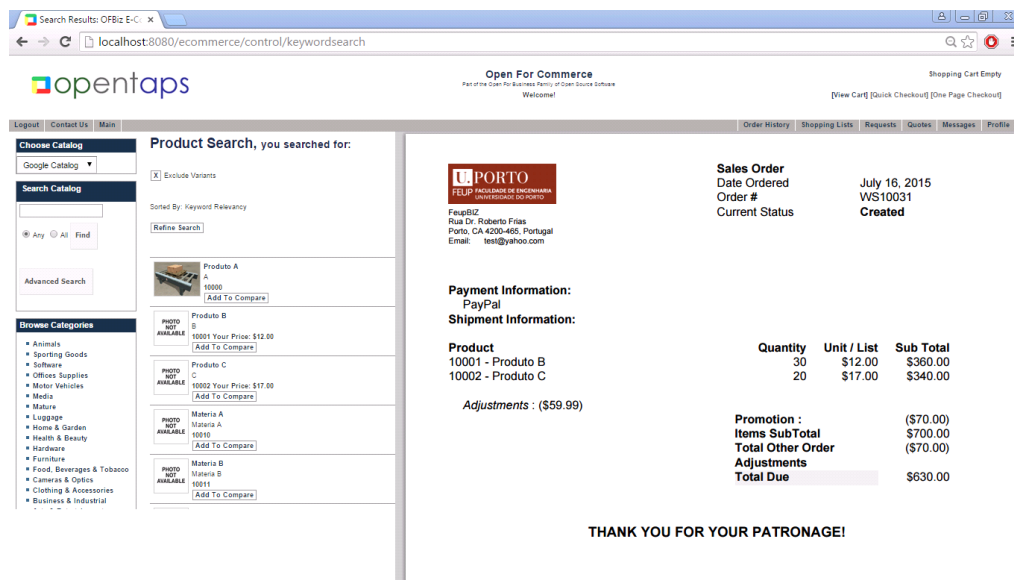


Figura 4.3 - Web-Store e ordem de venda criada

No MRP são introduzidos os produtos para os quais se pretende planear a respetiva produção realizando as seguintes tarefas [34]:

1. Análise da previsão de venda e das ordens de venda existentes para o produto;
2. Estabelecimento dos prazos de entrega de cada ordem de venda no plano de produção mestre;
3. Verificação da existência de produtos suficientes para a satisfação das ordens de produção e previsões de vendas, por meio do plano de inventário. No caso de existir falta de produtos no inventário da fábrica são gerados os requerimentos:
 - a. De transporte, se existir material disponível em armazém externo à fábrica;
 - b. De produto, se for necessário e vantajoso encomendar o produto a fornecedores;
 - c. Internos de produção, caso seja necessário produzir o material na fábrica;
4. Para cada componente do produto que se encontra indicado no BOM (Lista de materiais) é realizado o passo 3.

Os requerimentos criados pelo MRP necessitam de ser aprovados de modo a transformarem-se em ordens de produção, em pedidos de transporte e em encomendas aos fornecedores, respetivamente.

A ordem de produção cria *Production Runs*, exemplificadas na Figura 4.4 e na Figura 4.5, no módulo de produção do Opentaps, que contêm toda a informação acerca do processamento da ordem. Esta informação pode ser posteriormente usada para análise estatística.

Production Run ID: 10050 Imprimir Revert Cancel

Product Name	A (10000)
Estado	Confirmed
Qty To Produce	30
Estimated Start Date	20-07-2015 16:24:00
ManufacturingActualStartDate	
Calculated Completion Date	20-07-2015 16:29:01
Completion Date	
Production Run Name	Produto A-Peça A
Description	

Tasks

Número de Sequência	Routing Task Name	Estado	Fixed Asset	Calculated Completion Date	Actual Time	Produced
1	Assembly Peça A [10051]	Scheduled	00001	20-07-2015 16:29:01	0	

Figura 4.4 - Página da *Production Run* no Opentaps

Production Run ID: 10050 Imprimir Close

Product Name	A (10000)
Estado	Completed
Qty To Produce	30
Estimated Start Date	20-07-2015 16:24:00
ManufacturingActualStartDate	20-07-2015 16:28:59
Calculated Completion Date	20-07-2015 16:29:01
Completion Date	20-07-2015 16:33:03
Production Run Name	Produto A-Peça A
Description	

Produced	0
Rejected	0
Stock in	<input type="text" value="30"/>
Inventory Item Type	<input type="text" value="Non-Serialized"/>
Lote	<input type="text"/> <input type="button" value="Criar Novo"/>
	<input type="button" value="Adicionar"/>

Figura 4.5 - Página final da *Production Run* no Opentaps

4.1.1.1 Configuração da aplicação

A realização das atividades de processamento da encomenda requer a configuração prévia de informação na aplicação Opentaps, pelo que foi inserida informação acerca da empresa fictícia, da fábrica, das células e unidades da fábrica. Criaram-se também produtos e matérias-primas no sistema assim como a informação acerca da produção destes.

Esta informação é guardada em três documentos:

- O BOM que apresenta todas as matérias-primas e quantidades respetivas necessárias para fabricar um produto como se apresentada na Figura 4.6;
- Os passos da produção, definidos no Opentaps como tarefas (*routing tasks*), que descrevem as operações a realizar e onde se anexam os custos e tempos de cada passo, apresentado na Figura 4.7;
- A rota (*routing*) que define a sequência das várias tarefas desde o início até ao fim da produção, apresentado na Figura 4.8.

Components of this Product									
Product ID	Product Name	Data Origem	Data Final	Número de Sequência	Quantidade	Scrap Factor %	Formula	Routing Task	Only for Routing
10010	Materia A	13-04-2015		1	2			10000	

This Product is a Component of				
Product ID	Product Name	Data Origem	Data Final	Quantidade

Figura 4.6 - Informação do BOM no Opentaps

Task ID	Routing Task Name	Descrição	Task Type	Fixed Asset Id	Estimated Setup Time	Estimated Unit Run Time
10000	Assembly Peça A		Assembling		1.500	10.000

Figura 4.7 - Tarefa no Opentaps

Routing Task Name	Número de Sequência	Data Origem	Data Final	Estimated Setup Time	Estimated Unit Run Time
[TASK01] Stock out	10	2004-09-24 15:09:38.000		0	600000
[TASK02] Assembly	20	2004-09-24 15:09:47.000		0	900000
[TASK03] Test	30	2004-09-24 15:09:55.000		20000	450000

Figura 4.8 - Rota no Opentaps

4.1.1.2 Automação das atividades do ERP

No decurso da análise da aplicação verificou-se que a aprovação da ordem de venda e do requerimento interno necessitavam de ser executadas manualmente. Para que o Opentaps gerasse automaticamente a ordem de produção, essencial ao cmNavigo, era necessário que as aprovações da ordem de venda, do requerimento interno e das atividades do MRP fossem executadas automaticamente.

Num primeiro momento a solução foi a escrita de código adicional no Opentaps para aprovação da ordem de venda e execução do MRP para todos os produtos da ordem de venda. No entanto, esta não era uma solução desejável devido à ausência de apoio técnico ou informação acerca deste tópico (pois embora exista alguma informação acerca da aplicação, esta não referencia a automação destas atividades) o que implicaria grande dispêndio de tempo de programação sem garantia de sucesso.

Tendo em conta a especificidade da integração que se está a desenvolver na sala de aula, a não existência de fornecedores e o facto de os materiais serem criados na hora faz com que os requerimentos (MRP) se revelem supérfluos à aplicação porque não existem fornecedores nem armazéns externos. Deste modo todas as ordens de venda geram obrigatoriamente uma ordem de produção.

Com o apoio da Critical, concluiu-se que a informação requerida pelo cmNavigo podia ser retirada diretamente da ordem de venda, que é gerada automaticamente no Opentaps após o cliente realizar o *CheckOut*.

4.1.1.3 Comunicação do Opentaps com o cmNavigo

De modo a concluir a integração entre as duas aplicações era necessário definir o método de comunicação entre o Opentaps e o cmNavigo. O método *standard* da comunicação do interface das aplicações de gestão com as aplicações MES, realiza-se através de *Webservices*, ou seja, o acesso de conteúdos via internet, num formato de troca de mensagens padronizado sendo o mais comum o formato XML (*Extensible Markup Language*)[9].

Estes *Webservices*, como ilustra a Figura 4.9, permitem o envio e receção de dados automaticamente ou com interação humana, para atualização de base de dados, recolher relatórios de atividades ou qualquer outra ação de uma aplicação via internet [22].

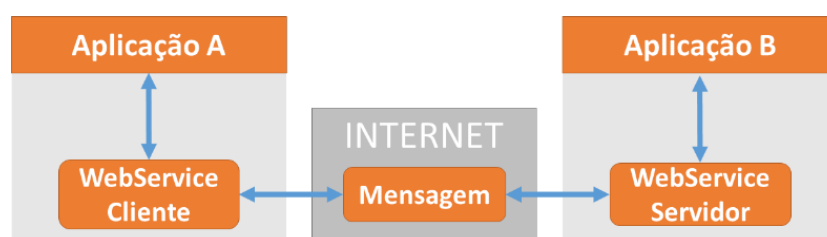


Figura 4.9 - Comunicação por *Webservice*

O Opentaps é capaz de gerar documentos XML, contudo este processo não é automático, não existindo no Opentaps um interface de comunicação dotado de um conjunto de serviços agregados, o qual seja capaz de despoletar atividades como o MRP, criando e recebendo mensagens automaticamente. A solução para este problema seria desenvolver esse interface. No entanto, como já foi referido anteriormente, é uma solução inadequada devido à ausência de apoio técnico do Opentaps ou de informação relevante para a realização desta tarefa.

Com o intuito de ultrapassar esta barreira estudou-se, conjuntamente com a Critical Manufacturing, a possibilidade de automatizar esta comunicação externamente ao Opentaps, em virtude de no cmNavigo existir apoio técnico disponível e a escrita de código na aplicação ser facilitada pelas regras DEE, descritas anteriormente na secção 3.6.

Do estudo resultaram duas soluções. A primeira opção seria do cmNavigo chamar as atividade de geração do código XML do Opentaps por meio de *WebServices*. No entanto, esta solução não era viável, em virtude da falta do interface de comunicação.

O outro método estudado consistia no cmNavigo aceder, periodicamente, à base de dados do Opentaps e retirar diretamente a informação necessária, permitindo, deste modo uma integração simples e direta, uma vez que existia o acesso total à base de dados do Opentaps, optou-se por esta solução. No entanto, a escrita de informação do cmNavigo para o Opentaps requeria que o Opentaps processasse a informação escrita na base de dados para atualizar o seu estado. Esta atualização remota envolvia novamente chamar os serviços para atualizar a aplicação externamente.

Na análise das características do projeto a desenvolver concluiu-se que as vantagens da comunicação, no sentido do cmNavigo para o Opentaps, destinadas à aplicação na sala de aula não justificavam o tempo despendido na sua realização, visto que todas as atividades que beneficiavam da informação do processo como análise estatística, custos de produção, manutenção, podiam ser desempenhadas no cmNavigo. Apenas ficariam de fora as atividades relacionadas com fornecedores e clientes, não existentes na sala, e contabilidade, sem interesse para o ensino da integração de sistemas de gestão com sistemas de controlo.

Concluída a seleção do método de integração entre o cmNavigo e Opentaps, tendo sido decidido que a troca de informação seria desenvolvida na aplicação cmNavigo, concluiu-se a configuração do Opentaps e transitou-se para a aplicação cmNavigo.

4.2 cmNavigo

A metodologia usada para descrever a integração do cmNavigo é semelhante à realizada no Opentaps tendo início numa descrição das atividades da aplicação, seguida da informação necessária na aplicação para esta desempenhar as suas atividades e por último são apresentadas as integrações realizadas com as aplicações adjacentes.

4.2.1 Controlo de receitas do produto

As atividades realizadas pelo cmNavigo sobre o produto iniciam-se pela entrada da informação proveniente do Opentaps sobre a forma de ordens de produção. Estas ordens de produção requerem matéria-prima, normalmente requisitada a armazéns adjacentes ao sistema de produção, às quais o cmNavigo (MES) realiza a requisição. Como já foi referido anteriormente não é necessária a requisição, na integração da sala de aula, visto os materiais serem virtuais e poderem ser criados instantaneamente.

O material ao dar entrada no sistema de produção apresenta já uma rota de passos pré-definida. Em cada passo apresentam-se quatro estados diferentes, como o exposto na Figura 4.10.

O estado inicial do material no passo é em lista de espera (*Queued*). Aqui o material aguarda pela disponibilidade de uma unidade capaz de processar o material. Quando o cmNavigo é informado que uma unidade está disponível é gerada uma ordem de transporte (*Dispatch*) destinado ao material no topo da lista de espera.

Essa ordem de transporte transita o estado do material de *Queued* para *Dispatched*. Este estado constitui o transporte do material para a unidade onde será processado. Quando o material dá entrada na unidade de produção, esta comunica ao cmNavigo o evento *TrackIn*, que faz com que o material transite para o estado em que vai ser processado.

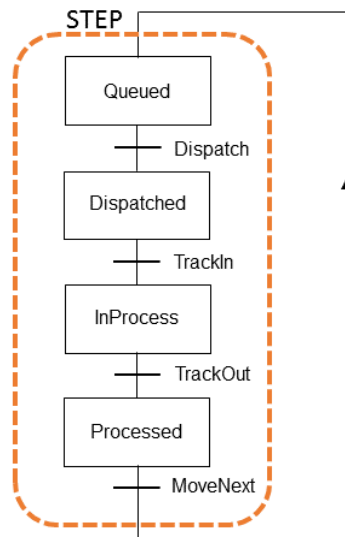


Figura 4.10 - Estados do Material no Passo

No estado *InProcess*, o cmNavigo envia uma receita com as ordens de produção para a unidade. Quando a unidade completa o processamento do material comunica ao cmNavigo que o material se encontra à saída da unidade (*TrackOut*) e pronto a avançar para o próximo passo (*MoveNext*).

O material dá então saída da unidade e transita para o passo seguinte, onde se reinicia o ciclo. É desta forma que o material é processado, passando por todos os passos, até atingir o passo final onde é concluída a produção transformando-se em produto.

4.2.2 Configuração da aplicação

O cmNavigo requer a introdução de informação base para a realização das suas atividades. De seguida são apresentadas as várias entidades relevantes à configuração da aplicação.

O passo inicial na configuração é a criação da instância da fábrica (**Facility**) que vai conter a informação sobre a fábrica e também estabelece a ligação a todas as entidades que pertencem a este espaço físico.

De acordo com a Figura 4.11, a fábrica pode ser dividida em áreas (**Area**). Dentro de cada área existem todas as unidades de produção definidas no cmNavigo como recursos (**Resources**) que realizam o processamento do material. É, então, necessário criar as instâncias das várias áreas e das unidades dentro de cada área.

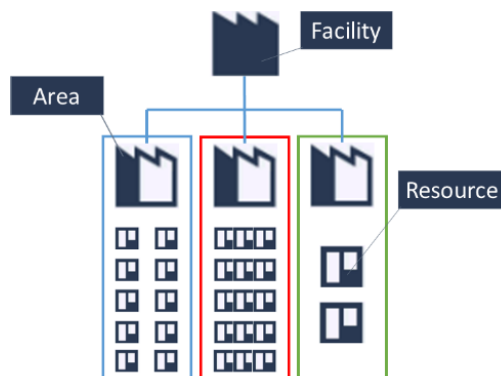


Figura 4.11 - Relação entre a **Facility**, **Area** e **Resource**

De seguida são criados os vários produtos e componentes na classe produtos (**Product**), não existindo diferença entre componentes de cada produto e o produto final. A relação entre cada produto e os seus componentes é descrita na lista de materiais (**BOM**), que relaciona o produto com os seus componentes, como revela a Figura 4.12, sendo necessário também criar a BOM de cada produto.

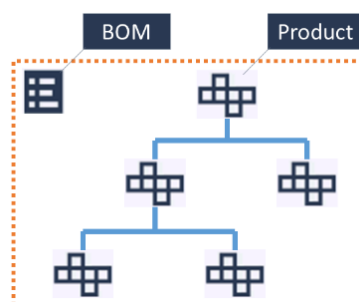


Figura 4.12 - Relação entre **Bill of Materials** e o **Product**

A classe de produtos é uma classe abstrata. Cada material físico existente na fábrica é uma instância da classe material (**Material**) que tem uma identificação específica. Cada instância de material está relacionada com uma instância da classe produto herdando as características dessa classe como o representado na Figura 4.13.

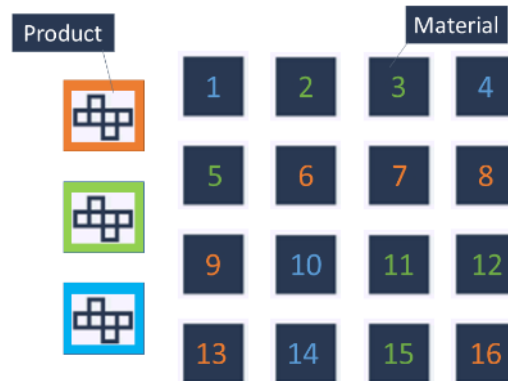


Figura 4.13 - Relação entre materiais e produtos

São criadas instâncias dos vários passos (**Step**) de produção que se realizam na fábrica bem como as instâncias das rotas (**Flow**) pelos quais os materiais podem transitar no decorrer do processo como demonstrado na Figura 4.14.

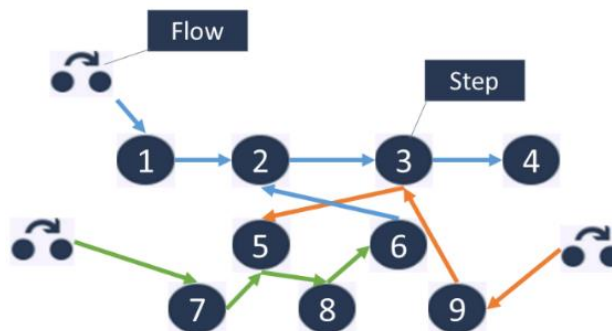


Figura 4.14 - Esquema do Flow e Step

Em cada passo é realizado um serviço (**Service**) no material. Esta entidade permite agrupar uma gama de diferentes operações no mesmo passo dependendo a operação do recurso, tipo de produto a fabricar, rota que o material está a seguir, dando assim maior flexibilidade à criação de receitas.

Quando um material dá entrada no recurso, é gerada uma instância da receita (**Recipe**) como descrito na Figura 4.15.

Esta instância é específica à interseção do serviço a realizar no material com o recurso que o vai realizar e contém toda a informação necessária ao recurso, para que este realize as operações no material, sobre a forma de parâmetros de produção (**Parameter**) que podem ser reutilizados nas várias receitas. Estes podem ser de vários formatos como *strings*, variáveis binárias ou equações, podendo variar no tempo.

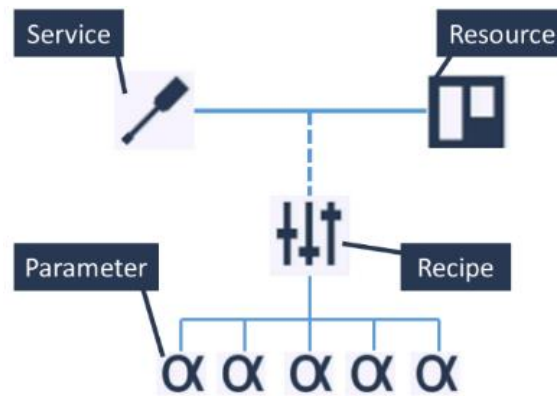


Figura 4.15 - Relação entre o *Service*, *Recipe* e *Parameter* e *Recursos*

4.2.3 Automação das atividades do cmNavigo

A recolha de informação oriunda do Opentaps, como ilustra a Figura 4.16, vai ser realizada através de uma regra DEE que lê diretamente a base de dados MySQL, do Opentaps, recolhendo periodicamente as novas ordens de venda, de modo a que o cmNavigo consiga realizar as suas atividades automaticamente.



Figura 4.16 - Comunicação por regra DEE

Um outro aspeto que requer a automação são as transições dos estados e dos passos do material que no cmNavigo são operações realizada manualmente. O cmNavigo precisa da informação de *TrackIn* e *TrackOut* para atualizar o seu estado e envia as ordens de transporte (*Dispatch*) do material para os recursos, indicando para onde deve ser enviado cada material, assim como as receitas para o processamento do material.

Os sistemas MES e SCADA são integrados através de redes cliente-servidor usando protocolos de comunicação OPC DA, como ilustra na Figura 4.17 [3].

Estas redes especificam, em tempo real, a comunicação de informação do processo entre equipamentos de controlo de vendedores variados. O objetivo da comunicação aberta é reduzir o esforço necessário por parte dos fornecedores de soluções de automação, em permitir que os seus equipamentos sejam interoperáveis, e comuniquem com equipamentos variados de outras marcas. No entanto basta o equipamento ou aplicação informática possuir um servidor OPC para permitir a comunicação com aplicações de níveis superiores ou inferiores.



Figura 4.17 - Comunicação OPC [35]

Em virtude do tempo limitado para a execução da dissertação e tendo em conta que para se realizar a integração de duas aplicações, através de uma comunicação OPC, são necessários conhecimentos avançados sobre OPC, optou-se pela aplicação de uma comunicação simples.

Uma vez que a comunicação a trocar, na forma de receitas, é informação tipicamente constante no tempo, não sendo necessário um controlo ao nível do milissegundo, optou-se por desenvolver uma comunicação semelhante à da troca de informação com o Opentaps.

Adotou-se então uma comunicação por base de dados. Esta comunicação é executada por códigos escritos, quer no cmNavigo quer no Vijeo Citect, que são despoletados por temporizadores (*Timers*). Deste modo, aproveitou-se o código da programação com o nível superior.

4.2.3.1 Códigos DEE

Os códigos realizados para a automação no cmNavigo serão expostos com o auxílio de *Grafjets* que descrevem o comportamento dos códigos.

Um aspeto a ter em conta em todos os *Grafjets* apresentados é que, no caso de ocorrência de erro ao realizar a ação descrita na transição entre estados, é desencadeado o retorno ao estado inicial, sendo revertidas (*Rollback*) as ações realizadas até ao momento, e despoletada uma mensagem de erro como apresentado na Figura 4.18.

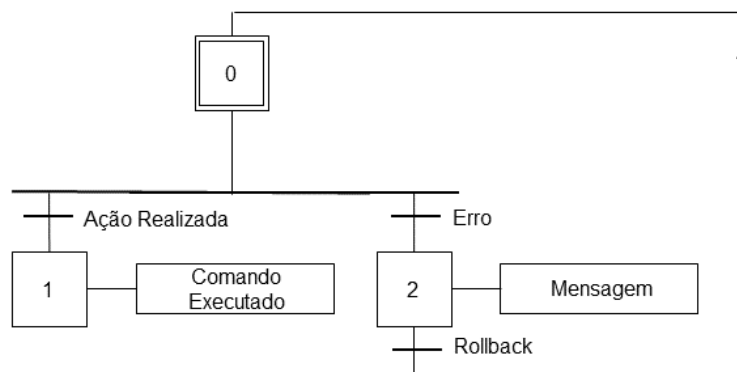


Figura 4.18 - Exemplo da falha da ação

De modo a simplificar a exposição do comportamento do código nestes *Grafcets* foram omissas as etapas de erro. Contudo, é de realçar a importância destas etapas, uma vez que são essenciais para deteção de erros e para o sucesso da execução da comunicação.

Estes códigos são constituídos por métodos do programa cmNavego e da base de dados MySQL que realizam a ligação à base de dados e as respetivas *queries*.

Uma Query é um pedido de informação à base de dados que retorna posteriormente a informação requerida [36].

4.2.3.2 Comunicação com Opentaps

O cmNavego vai recolher a informação apresentada na Figura 4.19 diretamente da base de dados para criar as ordens de produção a partir das ordens de venda do cliente.

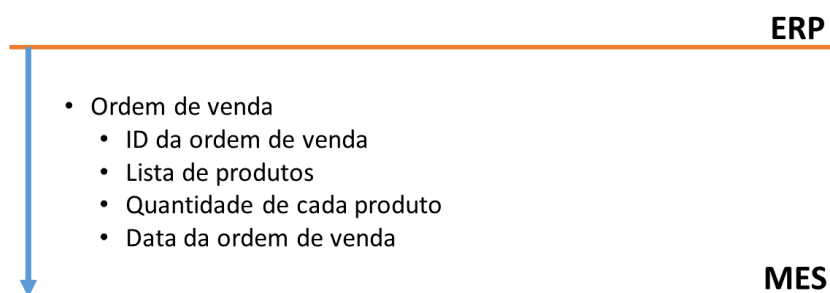


Figura 4.19 - Ordem de venda

A recolha da informação necessária para a produção, oriunda do Opentaps, constitui o primeiro passo para automação das atividades do cmNavego. Esta recolha ocorre através de uma regra DEE, cujo código apresenta um segmento genérico, usado em todas as comunicações de leitura à base de dados do projeto, por parte do cmNavego, exemplificado na Figura 4.20.

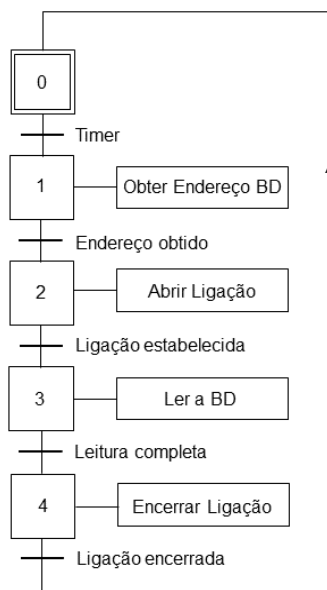


Figura 4.20 - Grafcet comportamental da leitura da base de dados

As regras DEE são iniciadas por uma ação. No caso do código de leitura, essa ação é um *Timer*, que envia um sinal de minuto a minuto. Ao iniciar o código é recolhida a informação de acesso à base de dados. Esta informação encontra-se guardada, numa linha de texto, nas configurações do cmNavigo, como mostra a Figura 4.21. Caso seja necessário alterar algum parâmetro de acesso à base de dados, uma vez que esta informação é externa à regra não é necessário alterar a regra DEE.

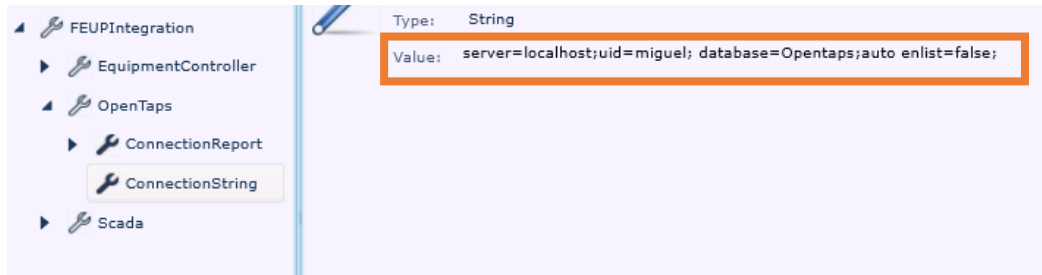


Figura 4.21 - Linha de texto da ligação à base de dados

Na etapa 2 é realizada a ligação à base de dados através da instância *MySQLConnection* criada que estabelece a ligação à base de dados. Esta instância também encerra, de igual modo, ligação na etapa 4. Na etapa 3 do código genérico é executado o segmento apresentado na Figura 4.22 que descreve o que ocorre especificamente na leitura da base de dados do Opentaps.

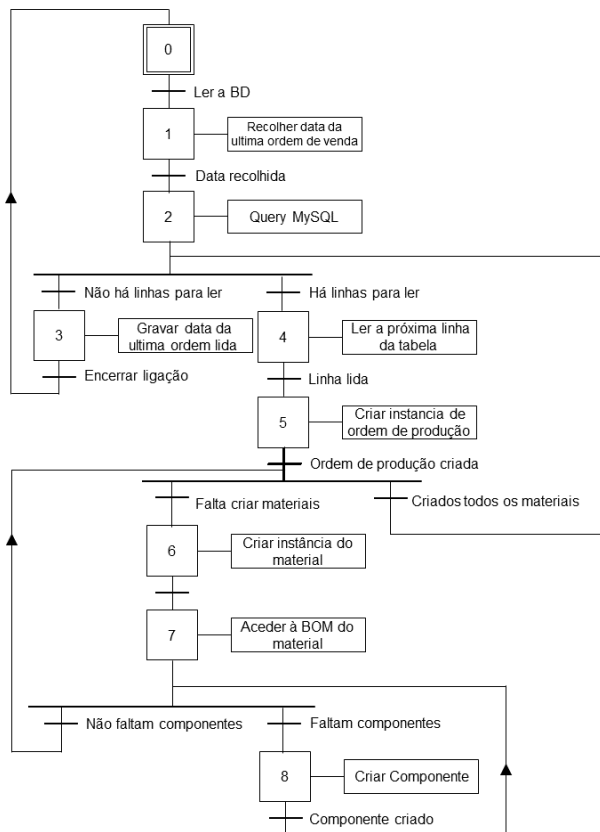


Figura 4.22 - Grafcet comportamental da leitura da base de dados do Opentaps

Neste segmento é recolhido, inicialmente a data da última ordem lida, que se encontra novamente na configuração do cmNavigo, junto ao endereço de ligação. Seguidamente é realizada uma *query*, onde é recolhida a informação das ordens de venda, ainda não lidas, pelo cmNavigo da base de dados do Opentaps. O texto executado pela *query* é apresentado de seguida. Se a *query* não retornar informação, quer dizer que não existem novas ordens de venda, pelo que termina a leitura da base de dados e, conseqüentemente, a etapa 3 do código genérico de leitura.

Se a *query* retornar linhas de informação, é executada a leitura da próxima linha, visto a leitura do MySQL iniciar na linha 0, que não existe. Após a leitura da linha seguinte é criada a instância da ordem de produção. Também são geradas as instâncias dos materiais e respetivos componentes dessa mesma ordem de produção. Este processo é repetido para todas as linhas presentes na *query*. Após a conclusão de toda a leitura é encerrada a ligação com a base de dados e o código conclui a sua operação.

4.2.3.2.1 Teste

Após a criação da regra DEE, esta foi testada para deteção de erros. O método de teste do código foi a introdução de *input* sobre a forma de ordens de venda no Opentaps, após o qual se realiza uma única execução do código. De seguida é analisado o *output* através de uma mensagem criada no final da execução do código, mensagem na qual são referidas todas as atividades realizadas na execução, inserindo uma linha na mensagem após cada atividade. Esta mensagem também apresenta erros ocorridos na comunicação MySQL.

Após a deteção dos erros do código, por este método, foi realizado o mesmo teste, mas com a execução do código através do *Timer*, não sendo exibida a mensagem no final da execução, para não sobrecarregar o programa com mensagens.

A informação da mensagem foi anexada ao objeto *Facility*, esta indica os resultados da execução ao lado da data da última leitura da ordem de produção, como representado na Figura 4.23. Estas mensagens podem ser comentadas no código de modo a não sobrecarregar a informação, podendo ser inseridos de novo na eventualidade de algum erro.

Name	Value
ConnectionReport	Connection Created Connection Opened 4/27/2015 3:34:48 PM Command Created ***** ***** Connection Close END
LastPOModifiedDate	4/27/2015 3:34:48 PM
poReport	

Figura 4.23 - Relatório da comunicação

4.2.3.3 Integração e comunicação com o Vijeo Citect

Tendo sido definida a informação necessária proveniente do processo para realizar as atividades do cmNavigo, de acordo com a Figura 4.24, foi possível definir a troca de informação entre o cmNavigo e o Vijeo Citect, que será denominado como SCADA no seguimento da presente dissertação.

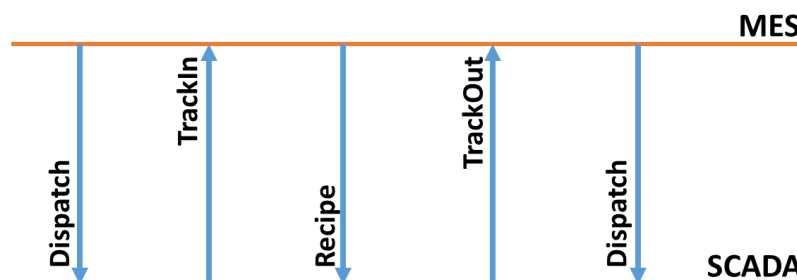


Figura 4.24 - Comunicação entre MES e SCADA

A fim de integrar o SCADA com o cmNavigo foram criadas duas tabelas na base de dados, sendo uma para o envio de ordens de transporte (*Dispatch*) dos materiais.

Na tabela *Dispatch* o cmNavigo escreve o Equipamento para o qual o produto tem que ser transportado, assim como a identificação do material a ser transportado, sendo esta a informação essencial para os equipamentos de transporte realizarem a operação de transporte.

Concluído o transporte do material o SCADA escreve o valor *TrackIn*=1 na respetiva linha da ordem de transporte, enviando o sinal de *TrackIn* para o cmNavigo.

Tabela 4.1 - Tabela *Dispatch*

Maquina	Material	TrackIn
Maquina A	Material00123	
Maquina B	Material00241	1
Maquina C	Material00136	

Na tabela Receita, ilustrada na Tabela 4.2, é enviada a receita de controlo para os recursos da fábrica. Nesta é indicado o recurso para o qual a receita de controlo deve seguir, como também o material no qual se vão realizar as ações.

Tabela 4.2 - Tabela Receita

Maquina	Material	Field1	Field2	...	FieldN	TrackOut
Maquina A	Material00123	“Ação1”	1		125	1
Maquina B	Material00241	“Ação8”	0		352	
Maquina C	Material00136	“Ação5”	1		124	1

A receita é constituída por parâmetros diversos que contêm a informação essencial ao recurso para este processar o material. Existem N campos para as variáveis que são preenchidos

por ordem crescente, permitindo enviar, de forma flexível, a informação para os diferentes recursos. Quando o equipamento conclui o processamento envia o sinal de *TrackOut* na linha específica da receita.

4.2.3.4 Escrita do *Dispatch*

O código da escrita na tabela das ordens de transporte (*Dispatch*) encontra-se numa regra DEE, cujo código apresenta um segmento genérico, usado em todas as comunicações de escrita à base de dados do projeto, por parte do cmNavego, como exemplifica a Figura 4.25. Este segmento genérico é, em tudo, semelhante ao código de leitura apresentado anteriormente na Figura 4.20, mas neste caso é despoletado por uma ação, em vez de um *Timer*. A ação que despoleta este código é a criação de uma ordem de transporte na aplicação cmNavego (*Dispatch.Post*).

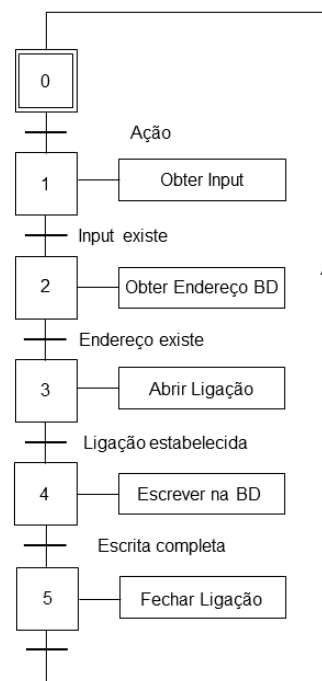


Figura 4.25 - *Grafcet* comportamental da Escrita na base de dados

Quando a regra DEE é iniciada, esta recolhe a informação da ordem de transporte que a despoletou, mais concretamente, a lista de materiais. O restante segmento de código é semelhante ao código de leitura com a exceção do estado 4, no qual se realizam as ações descritas na Figura 4.26.

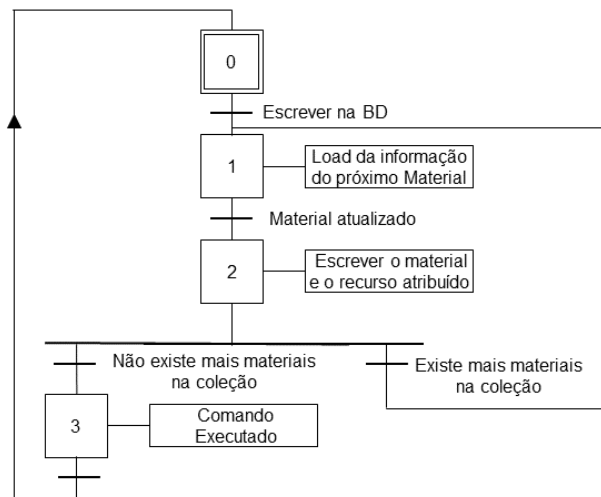


Figura 4.26 - Grafset comportamental da escrita na tabela *Dispatch*

Para cada um dos elementos da lista de materiais é recolhida a informação referente a esse elemento, para uma instância de material gerada no código. Essa instância é usada para escrever uma linha na tabela *Dispatch* com a identificação do elemento e do recurso para o qual o elemento tem que ser transportado. Concluída esta operação, para todos os materiais da lista, é terminada a escrita na tabela e é encerrada a ligação à base de dados, terminando assim o código.

4.2.3.5 Teste

Após a criação da regra DEE, esta foi testada, separadamente, para deteção de erros. O teste aplicado foi a inicialização da regra automaticamente ao transitar manualmente o material do estado de *Queued* para *Dispatch*, na aplicação cmNavigo. Esta atividade despoleta a regra DEE, que no fim da sua execução cria uma mensagem semelhante da comunicação com o Opentaps.

Após a deteção dos erros do código, por este método, foi desativada a exibição da mensagem no final da execução, para não sobrecarregar o programa com mensagens. Na eventualidade de algum erro é possível analisar a causa do erro através da mensagem criada pelo código de leitura do *Dispatch*, referindo-se mais à frente que automatiza a transição do estado do material.

Novamente estas mensagens podem ser comentadas no código de modo a não sobrecarregar o processamento da comunicação, podendo ser inseridos de novo na eventualidade de algum erro.

4.2.3.6 Escrita da Receita

Quando o material dá entrada no recurso, na aplicação cmNavigo, é gerada automaticamente uma instância da receita (RecipeInstance).

A criação desta instância despoleta um código de escrita na Tabela Recipe que apresenta um segmento de escrita semelhante ao da Figura 4.25, mas neste, a ação que inicia a regra, é a criação da instância da receita (RecipeInstance.Post) e a informação recolhida do *input* é a instância da receita.

O segmento da regra específico é apresentado na Figura 4.27.

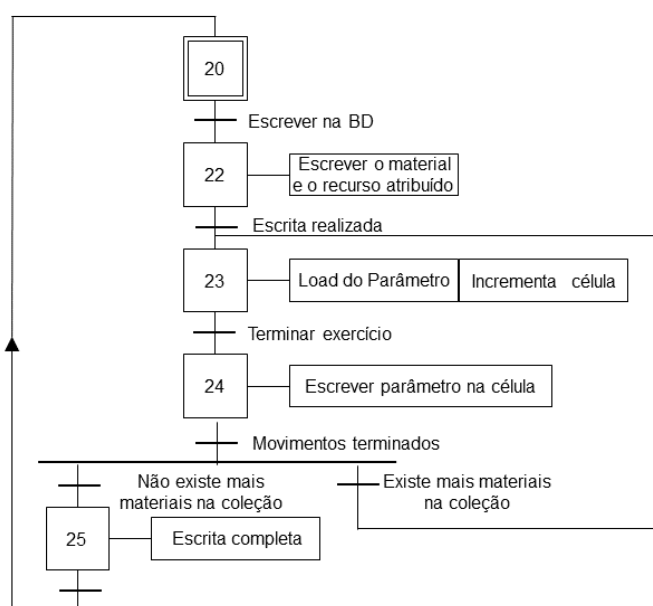


Figura 4.27 - Grafset comportamental da escrita da receita

Na etapa da escrita na base de dados são inseridos, inicialmente, a identificação do material e do recurso. Após a inserção do material e do recurso da receita são escritos todos os parâmetros existentes na instância da receita. Concluída a inserção de todos os parâmetros é concluída a escrita e encerrada a ligação e a regra DEE.

4.2.3.7 Teste

A regra de escrita da receita foi testada separadamente, de forma semelhante à escrita do *Dispatch*, sendo o teste inicializado manualmente ao transitar o material do estado de *Dispatch* para *InProcess*, na aplicação cmNavigo. Esta atividade despoleta a regra DEE, que no fim da sua execução cria uma mensagem. Após a deteção dos erros do código por este método foi desativada a exibição da mensagem no final da execução, mas, na eventualidade de algum erro, é possível analisar a causa do erro através da mensagem criada pelo código de leitura do *TrackIn*, referido mais à frente que automatiza a transição do estado do material.

4.2.3.8 Leitura do *TrackIn*

Quando o material dá entrada na máquina, esta envia o sinal de *TrackIn* para o SCADA, alertando a tabela *Dispatch*, no campo *TrackIn*. De modo a que o cmNavigo atualize o estado do material no sistema é necessário realizar a leitura da tabela *Dispatch*. Para tal foi criado um código de leitura com um segmento igual ao código da Figura 4.20, mas com um *Timer* de envio de sinal com a periodicidade de 3 segundos, devido à necessidade da comunicação ocorrer o mais próximo possível do tempo-real.

Este código contém o segmento específico apresentado na Figura 4.28.

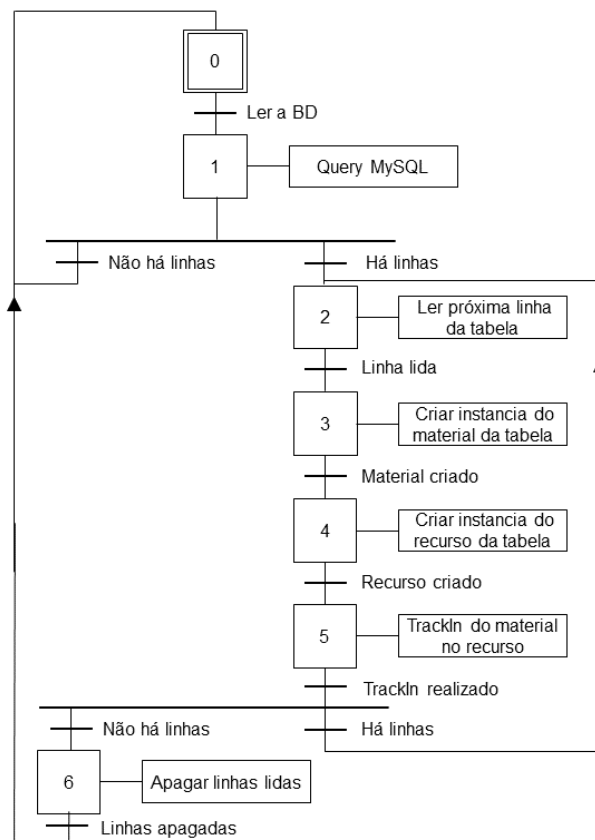


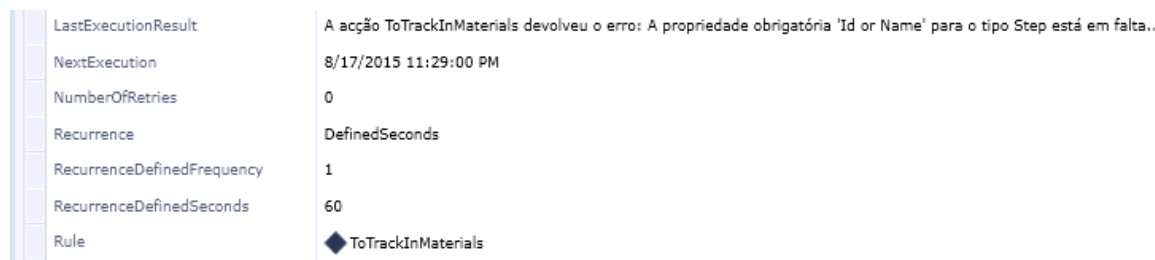
Figura 4.28 - Grafset comportamental do *TrackIn*

Neste segmento é realizada a seguinte *query*, onde se procura na tabela do *dispatch* as ordens de transportes concluídas (*TrackIn*=1). Se a *query* retornar linhas, quer dizer que há ordens de transporte concluídas. Procedem-se então à leitura linha a linha, em que em cada leitura é gerada a instância do material e a instância do recurso presentes na linha lida. Com estas duas instâncias é realizada automaticamente a atividade *TrackIn* (material, recurso) do cmNavigo, onde o cmNavigo atualiza sua informação, dando entrada do material no recurso.

Após a leitura de todas as linhas, estas são apagadas da base de dados. É concluída então a leitura e encerrada a ligação e a regra.

4.2.3.9 Teste

A automação da atividade *TrackIn* foi testada separadamente, sendo o teste inicializado, criando materiais no cmNavego, no estado *Dispatched*, e alterando manualmente a *Flag* de *TrackIn* na linha respetiva do material na tabela *Dispatch* e executando o código uma única vez. Concluída a regra é criada uma mensagem com o relatório da execução. Após a deteção dos erros do código, por este método, foi desativada a exibição da mensagem, no final da execução, mas caso ocorra algum erro no código é possível ver a mensagem do erro, na instância do *Timer* para o código de *TrackIn*, como o representado na Figura 4.29.



LastExecutionResult	A acção ToTrackInMaterials devolveu o erro: A propriedade obrigatória 'Id or Name' para o tipo Step está em falta..
NextExecution	8/17/2015 11:29:00 PM
NumberOfRetries	0
Recurrence	DefinedSeconds
RecurrenceDefinedFrequency	1
RecurrenceDefinedSeconds	60
Rule	◆ ToTrackInMaterials

Figura 4.29 - Informação da Instância do *Timer* do *TrackIn*

4.2.3.10 Leitura do *TrackOut*

Quando o recurso conclui o processamento do material envia o sinal *TrackOut* para o SCADA que altera a Tabela *Recipe* no campo *TrackOut*. De modo a que o cmNavego atualize o estado do material no sistema é necessário realizar a leitura da Tabela *Recipe*.

Para tal foi criado uma regra DEE de leitura com um segmento igual ao código da Figura 4.20, mas com um *Timer* de envio de um sinal de 3 em 3 segundos, do mesmo modo que a regra do *TrackIn*. Este código contém depois o segmento específico apresentado na Figura 4.30.

Neste segmento é realizada uma *query* onde se procura na Tabela *Recipe* as receitas concluídas (*TrackIn*=1). Caso a *query* retorne linhas, quer dizer que há receitas concluídas. Proceder-se então à leitura, linha a linha, em que em cada leitura é gerada a instância do material e a instância do recurso, presentes na linha lida. Com estas duas instâncias é realizada a atividade *TrackOut* (material, recurso) do cmNavego atualizando o estado do material no sistema e dando a saída do material do recurso.

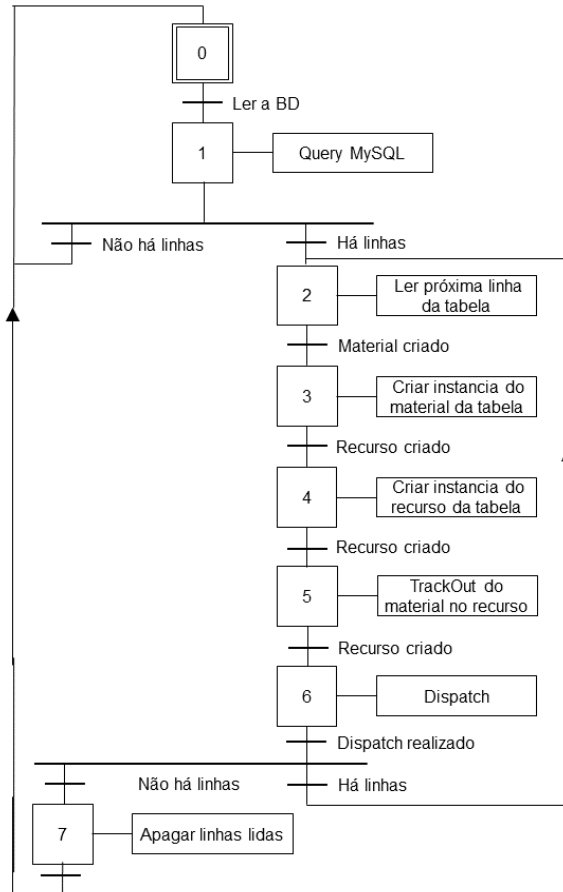


Figura 4.30 - Grafset comportamental do TrackOut

Após dar saída do recurso é necessário transitar o material para o próximo passo, definir o próximo recurso do material e realizar a ação de *Dispatch* do cmNavigo para o material que vai despoletar a escrita da ordem de *Dispatch*, já referida. A etapa 6 da Figura 4.30 é descrita em detalhe na Figura 4.31.

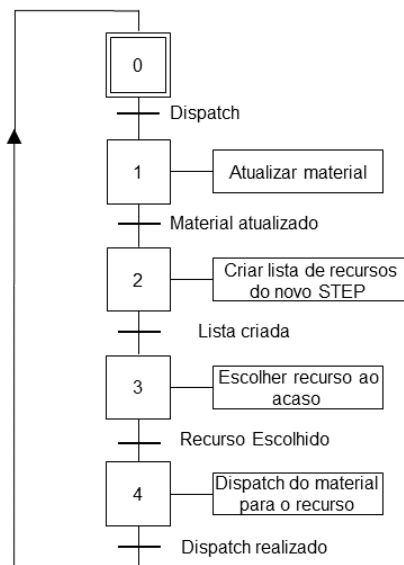


Figura 4.31 - Grafset comportamental do Dispatch

Após o *Trackout* é atualizada a instância do material, na regra DEE, visto o sistema ter sido atualizado enquanto a regra corria e de modo a que a instância do material apresente, agora, a informação do passo seguinte, sendo recolhida a lista de recursos capazes de realizar o novo passo do material, escolhido um ao acaso e realizada a função *Dispatch* do material para esse recurso concluindo, assim a automação do ciclo do passo e do *cmNavigo*.

4.2.3.11 Teste

Esta regra DEE foi testada da mesma forma que foi testada a regra DEE do *TrackIn*. Neste caso o teste é iniciado alterando manualmente a *Flag* de *TrackOut* na respetiva linha do material a transitar de estado na tabela *Recipe* e as mensagens de erro da execução do código encontram-se no *Timer* para o código *TrackOut*.

4.3 Factory I/O

Entrando no domínio da automação industrial da integração vertical é necessário, antes da configuração do controlo direto e de supervisão das unidades criar as unidades a controlar. A estrutura apresentada na Figura 4.32 tem como objetivo o teste completo da integração criada, sendo constituída por 3 unidades e o transporte entre cada uma.

Na primeira unidade exponho o elemento na simulação que cria o material a ser processado, permitindo assim apresentar um modelo para o passo inicial de qualquer sistema de produção que se pretenda simular com a integração.

O segundo elemento é um modelo de unidade, que vai ser denominado de “Máquina”, no qual se pretende controlar as operações de produção que realiza sobre o material. Permite testar todas as atividades realizadas no controlo da produção da unidade, desde a ordem de transporte da unidade anterior até libertar a peça para o passo seguinte.

O último elemento é bastante semelhante ao primeiro, realizando a operação de remoção do material da simulação, permitindo, assim, apresentar um modelo para o passo final de qualquer sistema de produção que se pretenda simular com a integração.

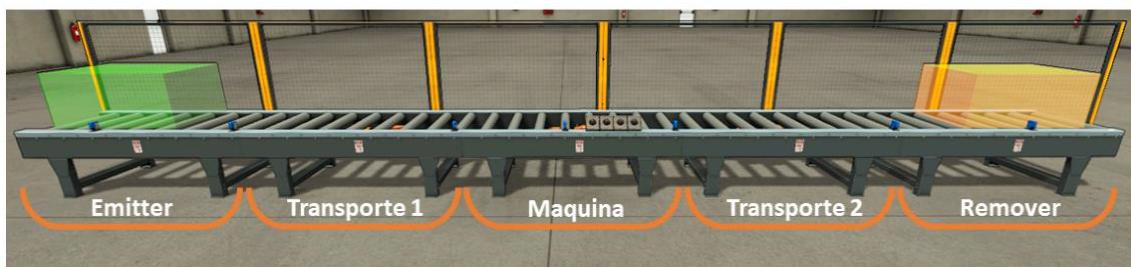


Figura 4.32 - Estrutura de teste

4.3.1 Emitter

A primeira unidade, representada na Figura 4.33, inclui o equipamento *Emitter* do Factory I/O, que cria, na simulação, os materiais a serem processados. Um *Emitter* deverá ser o STEP inicial de cada sistema de produção simulado pois embora o MES crie o material na sua aplicação também é necessário criar este no Factory I/O.

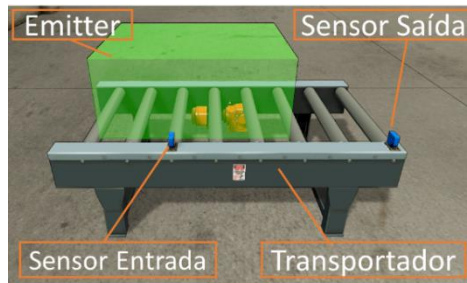


Figura 4.33 - Unidade de criação do material

Esta unidade é constituída também por dois sensores e um atuador. O atuador, que é um transportador, desloca o material dentro da unidade. O sensor de entrada deteta a criação do material. O sensor de saída permite detetar que o material se encontra na saída da unidade pronta para ser aceite pelo sistema de transporte (*TrackOut*) e quando este material sai completamente do equipamento.

4.3.2 Transportador

Quando o material sai do *Emitter* entra no sistema de transporte constituído por um transportador e sensorização, como o apresentado na Figura 4.34. O sistema de transporte recebe a informação da ordem de *Dispatch* do SCADA, sobre a forma do ID do material a transportar e do ID da unidade de destino.

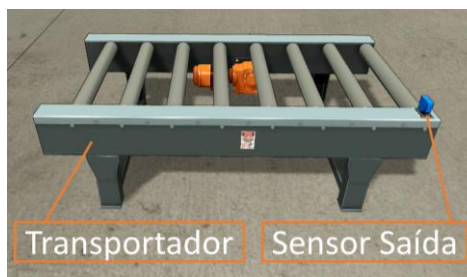


Figura 4.34 - Sistema de transporte

O transporte entre o passo 1-2 é igual ao transporte entre o passo 2-3. A simulação de teste da integração poderia apresentar um sistema de transporte para múltiplas unidades permitindo testar a definição flexível da unidade seguinte do material. No entanto, como foi ensaiada essa componente no teste do código de transição do passo no cmNativo, qualquer erro detetado na definição flexível originaria erros de programação ao nível do PLC. Tratando-se de uma característica supérflua ao teste e capaz de interferir, esta não foi implementada.

4.3.3 Máquina

A unidade Máquina apresenta um passo genérico que permite testar todos os estados do material, no decorrer do processamento desta. Nesta é também proposta uma estrutura de código base para todas as unidades de produção na simulação. Estrutura possível de ser implementada nos vários passos apresentando características de flexibilidade e modularidade na sua programação.

A unidade, como representado na Figura 4.35, é constituída por um transportador, sensores para o posicionamento correto do material no equipamento e indicadores luminosos que apresentam as operações de processamento realizadas, de acordo com a informação transmitida pela receita.

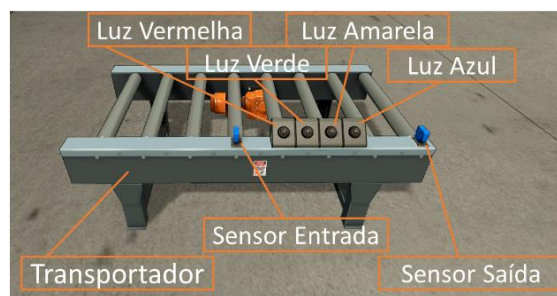


Figura 4.35 - Unidade Máquina

A estrutura de código proposta para as unidades de processamento, na Figura 4.36, deve apresentar um estado inicial onde aguarda por material.

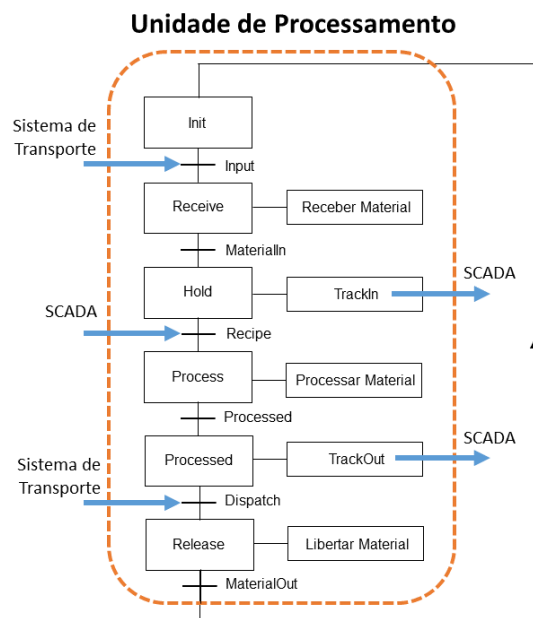


Figura 4.36 - Grafcet da Unidade de Processamento

Quando o equipamento de transporte, à entrada, lhe comunica que tem um material, este deve receber esse material, indicar ao SCADA que recebeu a peça, quando esta se encontra dentro da máquina, enviando o sinal *TrackIn*.

Quando o SCADA responde com a receita, este muda o seu estado para *Process*, onde se executam as operações de produção que a receita especifica.

Dentro do estado do processo encontra-se todo o código das operações de processamento que a unidade é capaz de realizar e a receita indica quais as operações a realizar, a sua ordem e parâmetros. Estas operações são visualizadas através de 4 luzes na máquina como demonstrado na Figura 4.37, podendo estas operações ser simples combinações de cores ou seqüências de comutações despoletadas pela receita.



Figura 4.37 - Material em processamento

Quando é completado o processamento do material a unidade transita para o estado *Processed*, comunicando ao SCADA o sinal de *TrackOut*. Após o envio do *Dispatch* para o sistema de transporte e se este se encontra disponível, a unidade liberta o material e retorna para o estado inicial aguardando novo material para processar.

Com a introdução, no sistema de transporte, de filas de espera como LIFOs (*Last In First Out*) e FIFOs (*First In First Out*) na entrada e saída de máquinas é possível que a máquina nunca pare por ter um material no interior, otimizando o uso da máquina.

4.3.4 Remover

Ao completar o fabrico do material, tendo sido este transformando num produto, é necessário remover o produto da simulação, sendo esta operação realizada pelo equipamento que inclui o Remover do Factory I/O, apresentado na Figura 4.38. Este equipamento semelhante ao *Emitter* serve como o passo final do processo de produção.

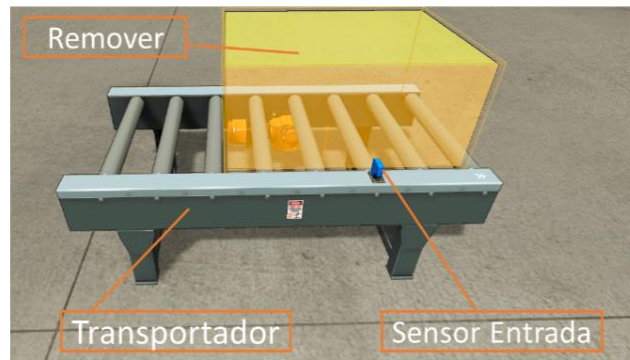


Figura 4.38 - Gráfico do equipamento

A estrutura de código para o *Emitter* e o Remover difere ligeiramente do código geral da unidade de processamento. O *Emitter* não apresenta o estado *Receive*, pois não existe unidade anterior, e *TrackIn*, pois o SCADA realiza um Dispatch fictício para o cmNavego sinalizando logo o *TrackIn* no passo inicial. O Remover não possui o *Release* por ser o passo final.

Com este teste pretendeu-se validar que a integração consegue processar todos os produtos ao longo de cada passo, através da comunicação de receitas e ordens de *Dispatch*. Demonstrando, não só uma maior flexibilidade nas operações de produção como também retirar conclusões de como os métodos de programação apresentados pela norma IEC61131-3 favorecem a implementação de sistemas automáticos integrados.

4.4 CODESYS

Sobre a simulação do sistema de produção teste foi implementado o controlo direto no Codesys de modo a que este apresente um comportamento de acordo com o descrito anteriormente no modelo da unidade. Como demonstrado na Figura 4.39 o controlo das unidades é realizado por um *SoftPLC*, denominado de “MyPLC”, que corre num computador da sala e comunica com o SCADA por Modbus TCP.

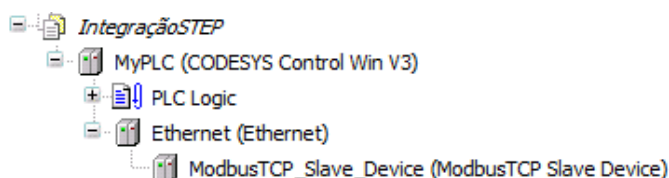


Figura 4.39 - Projeto do Codesys

4.4.1 Comunicação

O protocolo Modbus TCP permite a troca de informação por um cabo de rede Ethernet presente em todos os computadores da sala de aula, destacando-se pela sua simplicidade. As variáveis transmitidas podem ser valores numéricos, como temperatura e pressão, ou binários, para ligar e desligar um motor por exemplo.

A comunicação entre o CODESYS e o FACTORY I/O realizou-se por protocolo OPC, devido às duas aplicações conterem interfaces OPC sendo, deste modo, a sua integração direta, bastando garantir que as variáveis a comunicar apresentem a mesma identificação em ambas as aplicações.

4.4.2 Programação

A arquitetura de programação adotada para cada unidade de processamento utiliza as POU's segundo a norma IEC-61131-3, de forma a estruturar o código de cada equipamento num único elemento de acordo com a Figura 4.40. Deste modo a programação do equipamento encontra-se encapsulada, precavendo interferência entre variáveis de equipamentos diferentes sendo só possível alterar o comportamento do equipamento através das variáveis de entrada (*input*), e ler os valores das variáveis de saída (*output*).

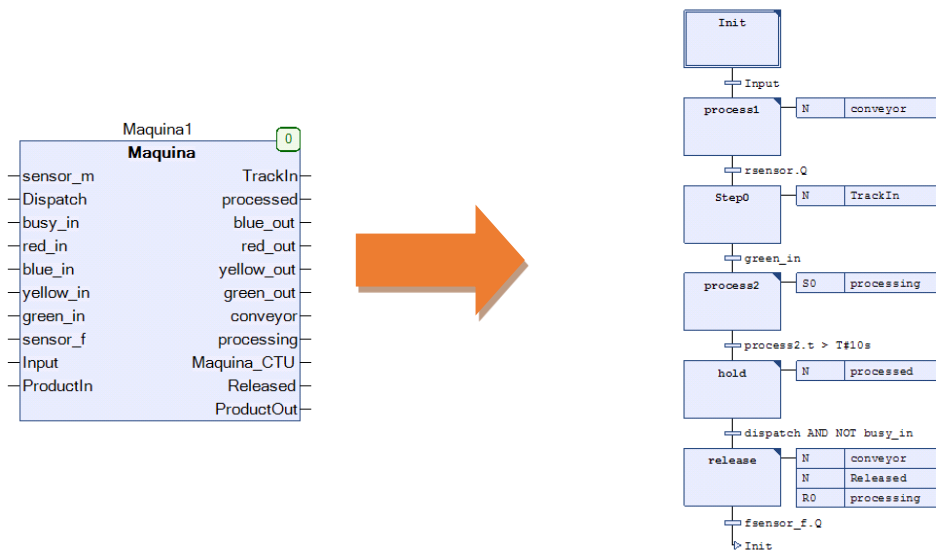


Figura 4.40 - POU do equipamento e código no seu interior

Adicionalmente o ambiente Codesys permite separar visualmente a lógica de cada equipamento, utilizando linguagens de programação como o CFC, demonstrado na Figura 4.41, definindo de forma clara as trocas de informação que ocorrem, facilitando a integração horizontal dos equipamentos,

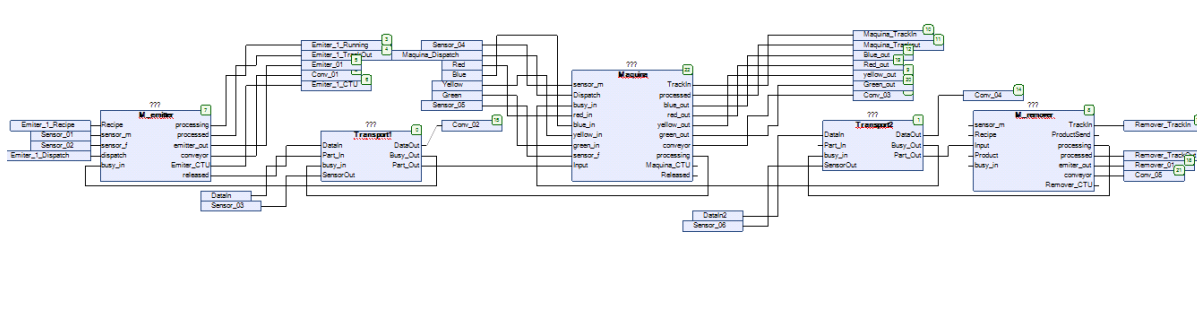


Figura 4.41 - CFC do cenário de simulação

O controlo das operações de produção por receitas e o transporte flexível entre equipamentos exige que a informação do material seja transmitida, em simultâneo, com o seu transporte, ao longo de todo o processo. Nas fábricas reais esta transmissão de informação é realizada através de códigos de barras ou RFID.

Na simulação, a informação do material (*DataIn*) é transferida quando o material físico dá entrada no equipamento (*PartIn*). O material só dá entrada no equipamento se este indicar que está disponível (*BusyOut*). Nos sistemas de transporte, como apresentado na Figura 4.42, transita também a informação da unidade de destino do material (*MaqIn*). Quando o material dá saída do equipamento, este verifica se o equipamento seguinte está disponível para receber a peça (*BusyIn*). Caso esteja disponível, sinaliza o envio do material (*PartOut*), em conjunto com a informação do material (*DataOut*) e do equipamento de destino (*MaqOut*).



Figura 4.42 - POU do equipamento de transporte

No caso das unidades de produção apresentada na Figura 4.43, para além de apresentarem variáveis que enviam a ação de controlo para os equipamentos físicos (*Conveyor*, *Blue_Out*, *Yellow_out*), variáveis de estado da unidade de produção (*Processing*, *TrackIn*, *TrackOut*) e as variáveis de informação do processo (*Maquina_CTU* que conta os materiais processados), recebem também as ordens de Dispatch e os parâmetros na receita que vão controlar as operações de processamento (*Red_in*, *Blue_in*, *Yellow_in*).

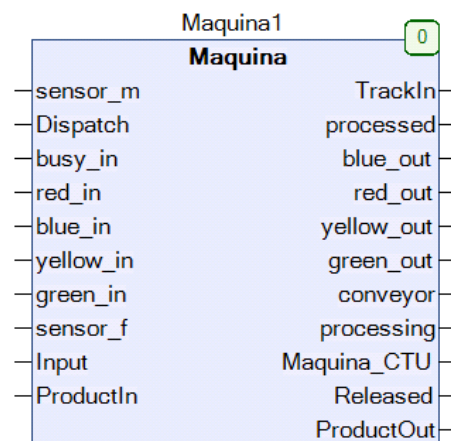


Figura 4.43 - POU da Unidade de Produção "Maquina"

4.5 Vijeo Citect

Concluída a programação do controlo direto da aplicação, temos como passo final a programação do controlo de supervisão. Esta vai transmitir a informação entre o controlo direto e o controlo da produção, ao mesmo tempo que supervisiona toda a operação. Oferece também aos operários ou supervisores a capacidade de visualizar o estado da operação, ao apresentar a informação trocada através de uma GUI.

Definida a estrutura foi então criado um projeto no SCADA, no qual se criou uma GUI de modo a supervisionar as unidades automáticas e realizar o rastreamento do material. Para além da GUI foi necessário criar os códigos CICODE para comunicação entre o cmNavigo e automação do envio da informação de receitas e Dispatch para os respetivas unidades e sistema de transporte. Foram também configuradas todas as variáveis *Tag* para as comunicações com o cmNavigo e o Codesys e eventos no sistema para despoletar a escrita de alterações na produção para o cmNavigo.

4.5.1 Programação modular das unidades e da GUI

De modo a facilitar a programação e a expansão do sistema de produção foram utilizadas ferramentas do Citect SCADA que permitiram a programação da aplicação de uma forma modular. Foram criados *templates* de cada unidade através do **equipment editor**, apresentado na Figura 4.44, que permite indexar toda a informação das unidades no sistema em módulos, denominados de “Equipamentos”.

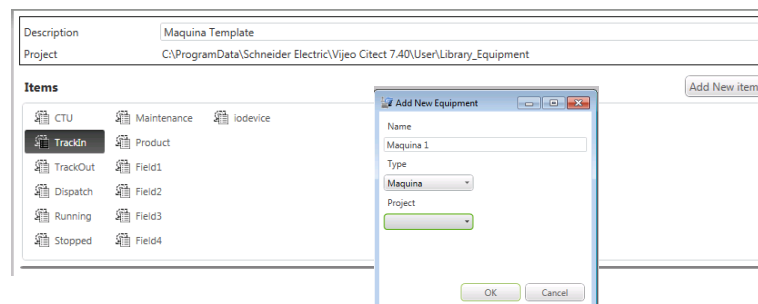


Figura 4.44 - *Template* do *Equipment Editor* e criação de novo equipamento

Como apresentado na Figura 4.45, os *templates* permitem indexar várias entidades do SCADA, desde *Tags*, alarmes, a configurações de comunicação. Deste modo, estas entidades são também criadas automaticamente com a criação de um novo equipamento, sendo possível obter menores tempos de programação da aplicação.

Var_1 (Variable Tags)	
Tag Name	{equipment.tagprefix}_TrackIn
Data Type	DIGITAL
I/O Device	{equipment.iodevice}
Address	{equipment.tagprefix}_TrackIn
Cluster Name	{equipment.cluster}
Comment	Flag TrackIn

Figura 4.45 - Informação da *Tag* do equipamento

Graficamente também foi adotada uma estrutura modular através de *templates* gráficos denominados de *Genies* que apresentam as mesmas características dos equipamentos, mas são usados na HMI. Estes *Genies* permitem indexar toda a informação do objeto gráfico, como demonstrado na Figura 4.46.

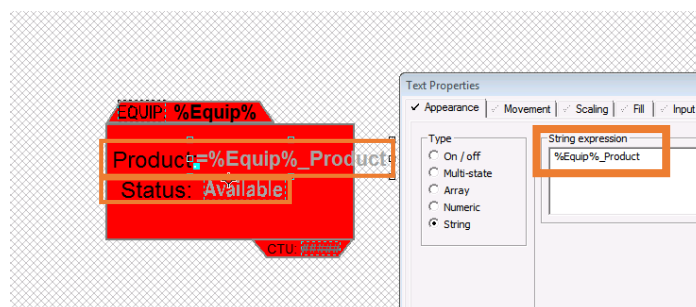


Figura 4.46 - Campos indexados no *Genie*

Deste modo, ao criar um novo objeto, como mostra a Figura 4.47 basta inserir o nome do equipamento para que todos os elementos presentes no objeto gráfico, como por exemplo *Tags*, sejam relacionados com as variáveis internas do objeto. No entanto, É necessário que o *template* do equipamento possua atributos com o mesmo nome que os valores indexados do *genie*.



Figura 4.47 - Criação de um módulo do *Genie* através do *template*

Esta metodologia modular que apresenta uma relação direta com as unidades físicas a controlar reduz erros humanos e tempos de programação. Com recurso a estas ferramentas foi implementada uma GUI do sistema de teste, apresentada na Figura 4.48, para rastreamento da informação dos materiais e estados das unidades.

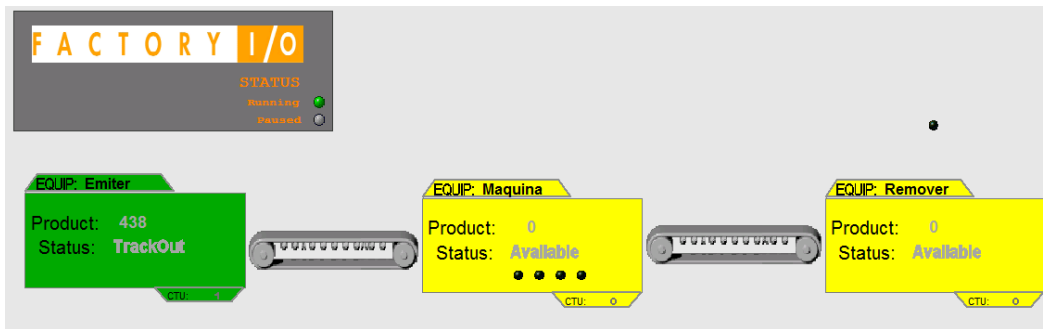


Figura 4.48 - HMI do SCADA usando Genies

4.5.1.1 Programação da comunicação com o cmNavego

A comunicação com o cmNavego foi criada a partir de uma ligação ODBC (*Open DataBase Connectivity*) no cliente Windows que define uma ligação, de acordo com a Figura 4.49, a uma base de dados e permite que as aplicações no computador tenham acesso livre a essa base de dados. Através destas ligações é lida a informação enviada pelo cmNavego e distribuída para as unidades presentes na célula que o SCADA controla. De seguida são expostos os vários códigos Cicode que estabelecem a ligação entre as *Tags* dos equipamentos e a informação do cmNavego.

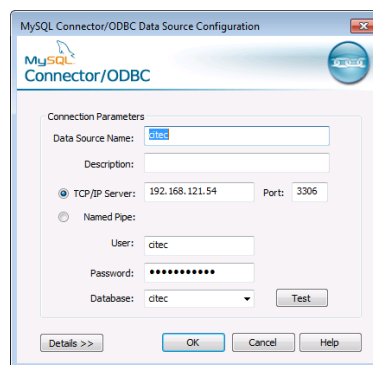


Figura 4.49 - Driver ODBC do MySQL

A comunicação do SCADA para o controlador dos equipamentos presente no CODESYS vai trocar valores de variáveis necessárias para o funcionamento dos equipamentos e respostas dos equipamentos através de uma comunicação Ethernet segundo o protocolo MODBUS TCP

4.5.1.2 Leitura da Receita

No SCADA existe então uma função Cicode, representada na Figura 4.50 que realiza a leitura da tabela *Recipe*. É realizada uma *query* que retorna as linhas da tabela receita com valor *TrackOut*=0. A leitura é realizada linha a linha e é lido o equipamento de cada receita sendo chamada a função *TaskNew*. A função *TaskNew* vai chamar uma outra função Cicode com o nome da máquina que transfere as variáveis da base de dados para as *Tags* do Equipamento. A função *TaskNew* tem vários modos e no modo “8” esta não volta a chamar funções que já estão a ser executadas.

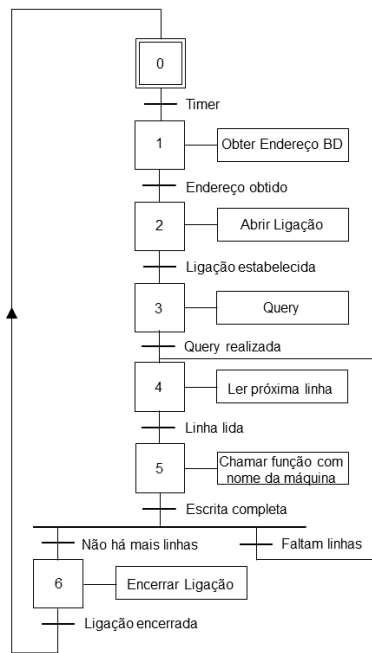


Figura 4.50 - Grafset comportamental da leitura da Receita

Mas se, por exemplo, um equipamento tiver capacidade para mais do que um material na função da máquina pode ler a tabela à procura de todas as receitas do equipamento. A limitação da capacidade do equipamento configura-se no MES restringido os materiais que podem estar *InProcess* no equipamento.

A função do equipamento, representada na Figura 4.51, vai ler o nome do material guardando a informação para permitir a monitorização do material e ler os parâmetros, transmitindo ao equipamento por *Tags*, que são transferidas por MODBUS, num ciclo While, pois a informação da receita é perdida na comunicação, se não for transmitida continuamente.

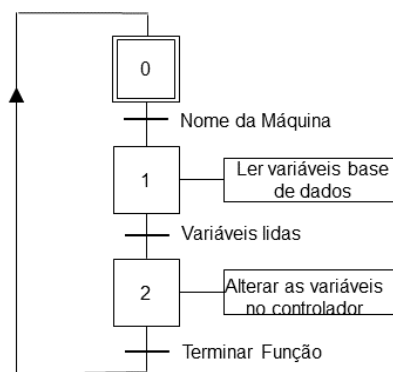


Figura 4.51 - Grafset comportamental das funções da máquina

4.5.1.3 Leitura do Dispatch

A função Dispatch, representada na Figura 4.52, lê a unidade de destino e o produto a ser transportado e é enviada para o equipamento de transporte, essa informação.

Como o nome do material circula pelo SCADA é possível saber a localização atual do material e a localização final do transporte.

Nesta aplicação consideramos que os equipamentos de transporte são entidades diferentes das unidades de produção e pretendemos programar os módulos de transporte separadamente. No entanto, seria possível ter considerado no CmNavigo que eram também equipamentos e que os transportes eram steps da produção e seriam criadas receitas.

Contudo, isto não é desejado, pois pretende-se uma estrutura modular. Qualquer alteração no chão de fábrica resultaria na necessidade de reprogramar a informação no cmNavigo, uma vez que a informação trocada no transporte é sempre a mesma (localização atual, material, localização final) ao se separar estes sistemas temos uma estrutura mais flexível a alterações. Esse controlo do transporte é então realizado ao nível da unidade ou da célula.

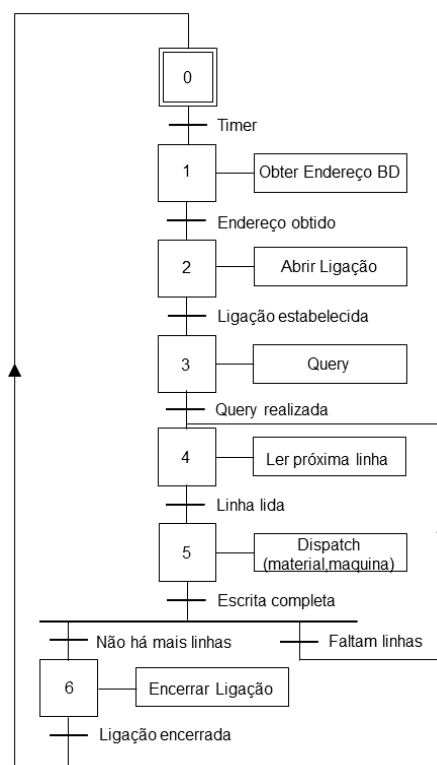


Figura 4.52 - Grafcet comportamental do Dispatch

4.5.1.4 Códigos de escrita do *TrackIn* e *TrackOut*

Depois existem os códigos de escrita do *TrackIn* e *TrackOut*, representados na Figura 4.53, onde o comando SQL vai alterar a variável *TrackOut* ou *TrackIn* para 1 da Tabela *Recipe* e Tabela *Dispatch* respetivamente. No *TrackOut* termina-se a função da máquina que estava a enviar a informação para o equipamento.

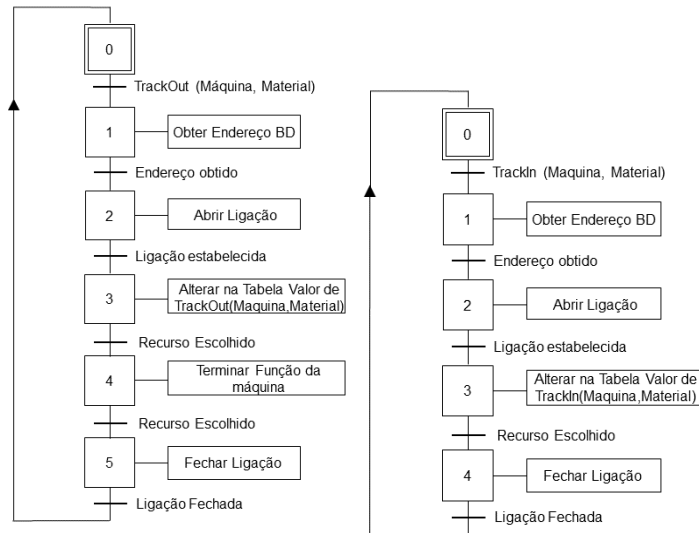


Figura 4.53 - *Grafcet* comportamental do *TrackOut* à esquerda e do *TrackIn* à direita

4.6 Resultados obtidos

Em jeito de conclusão o resultado final da integração, apresentado Figura 4.54, permite criar encomendas, através de uma *Web-Store* no *Opentaps*, das quais o *cmNavigo* recolhe a informação necessária para criar as ordens de produção, através da base de dados do *Opentaps*. Os produtos, das ordens, dão entrada no seu passo de produção inicial e aguardam, em lista de espera, pela sua produção.

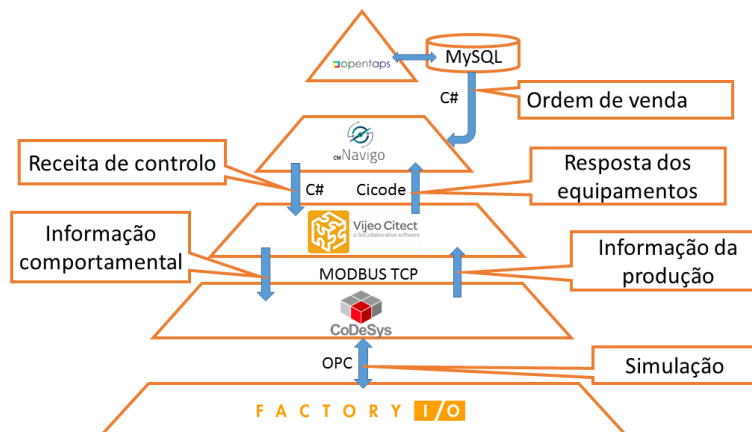


Figura 4.54 - Estrutura da Integração final

No momento em que se inicia a produção, o cmNavigo executa o controlo de receitas, em tempo real, enviando estas para a supervisão da célula. A supervisão analisa o destinatário e envia para o controlador da unidade de produção a correspondente informação comportamental que vai controlar as operações de produção. Este controlo altera as variáveis dos atuadores no simulador, realizando, assim, a ação de controlo sobre os atuadores no simulador. Ao correr a simulação os sensores existentes providenciam o feedback do estado de produção para o controlador. Concluída cada operação de produção o controlador envia a alteração do estado da unidade de produção para o supervisor, este atualiza o estado do material no cmNavigo, atualizando o estado da produção.

De modo a validar completamente a integração recorreu-se ao cenário no Factory I/O desenvolvido anteriormente, no qual se testaram todas as trocas de informação e atividades integradas em conjunto. Este teste validou a produção automática das encomendas dos clientes, de acordo com as receitas de produção de cada produto.

No entanto, é de referir um notório esforço no processamento de informação na aplicação cmNavigo, devido do facto de ser exigido, a esta aplicação, a execução de atividades complexas e uma resposta próxima de tempo real.

Em resultado, ao definir os ciclos de leitura do SCADA 3 em 3 segundos, o processamento da informação prolongava-se. Como cada leitura desencadeava de novo o processamento de informação, estes acumulavam-se, acabando por acontecer TimeOut das atividades de processamento. Foi necessário definir uns ciclos de leitura maiores de 15 segundos, para que o código de leitura não sobrecarrega-se os processos que decorriam no cmNavigo.

A principal solução a este problema seria o de utilizar um computador com maior capacidade de processamento, tendo a aplicação cmNavigo sobrecarregado o processador do computador da sala de aula, como demonstrado na Figura 4.55. Este computador apresentou um desempenho muito inferior, quando comparado com o desempenho da aplicação em computadores diferentes. Deste modo foi possível compreender as exigências tecnológicas da troca de informação em grandes quantidades e em tempo real.

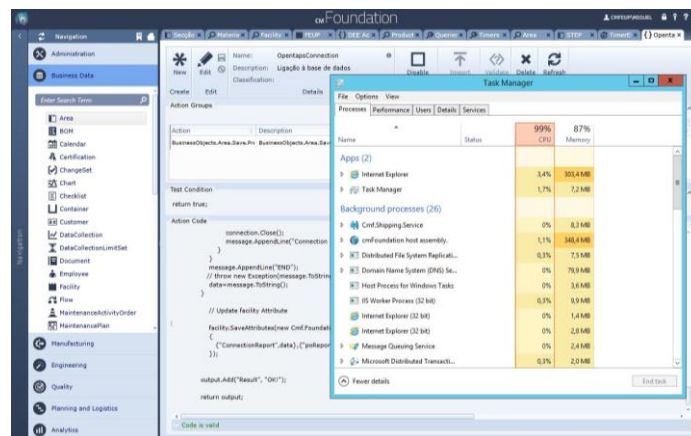


Figura 4.55 - Sobre carregamento do processador do computador

5 Conclusões

Esta tese teve como principal objetivo o de integrar, na sala de aula, um ambiente de produção e gestão da produção semelhante àquele que é esperado numa fábrica moderna, utilizando um ambiente virtual, para o ensino e instrução na área da automação industrial.

O desenvolvimento realizado no decorrer desta dissertação iniciou-se por um amplo estudo acerca de sistemas integrados de controlo com aplicações de gestão na indústria, analisando-se a estrutura destes sistemas, as normas aplicadas e as tecnologias atuais, tendo como resultado final a definição das características da integração a realizar na sala.

Após a definição da estrutura da integração foi realizada a seleção das aplicações a integrar, elegeu-se o ERP (Opentaps), o MOM (cmNavigo), o SCADA (Vijeo Citect), o SoftPLC da CODESYS e o simulador (FACTORY I/O). Com a aquisição das várias aplicações procedeu-se à integração destas e ao respetivo teste para validar o resultado obtido.

O resultado obtido foi positivo, uma vez que a integração desenvolvida na dissertação apresenta características que permitem, ao utilizador, desenvolver ambientes fabris complexos, com uma elevada variedade de produtos e apresentando-se como uma ferramenta didática na qual é possível desenvolver:

- Apetências de programação de PLCs, de uma forma modular e ajustada a integrações horizontais e verticais;
- Equipamentos com a finalidade de recolha de informação relevante do processo de produção, de modo a providenciar melhor informação às aplicações de gestão da empresa;
- Desenvolver HMIs para vários níveis da integração.

Permite também um primeiro contacto com aplicações MOM e ERP proporcionando aos alunos um conhecimento geral da hierarquia da automação.

No entanto, é importante referir a necessidade de instalar a aplicação cmNavigo num computador com maior capacidade de processamento, devido às exigências tecnológicas da troca de informação, em grandes quantidades e em tempo real.

Embora o resultado final seja positivo a falta de informação na integração do ERP revelou-se prejudicial, É ainda de salientar que a utilização das aplicações *Open-Source* têm aspetos menos positivos, especialmente em aplicações menos populares para comunidades de programadores, como os ERPs, pois a falta de documentação e apoio técnico requer aos utilizadores muita experiência na aplicação. Por estes motivos considero que a integração de

uma aplicação ERP da SAP, que é norma na indústria, seria algo interessante como ferramenta de formação.

Ao longo da realização do trabalho foi possível chegar a várias conclusões:

No decurso do extenso estudo realizado no início da dissertação, concluiu-se que existe um interesse muito grande, por parte da indústria, na integração de tecnologias de informação com a automação industrial, pelo que é relevante o ensino desta temática aos futuros engenheiros de automação.

O Factory I/O e a simulação virtual em geral constitui uma ferramenta com muito potencial, na área de ensino de sistemas integrados de controlo e gestão da produção, visto permitir trazer estes sistemas para dentro da sala de aula e apresentá-los, de forma interativa e didática.

Os conceitos da norma IEC 61131-3 aplicados durante a dissertação, revelaram-se valiosos para a realização das integrações entre equipamentos, sendo a aplicação da norma uma vantagem determinante para um programador de PLCs. Deste modo realça-se a necessidade do ensino da programação de autómatos de acordo com a norma IEC61131-3.

A integração vertical de várias aplicações constitui um problema de programação bastante complexo. A combinação de erros de programação, em aplicações diferentes, dificulta significativamente a sua correção. Deste modo é aconselhável o ensaio individual de cada código de comunicação, tendo em conta que o ensaio da comunicação em si, só se realize quando ambos os códigos já tenham sido validados.

5.1 Trabalhos Futuros

Esta dissertação realizou uma abordagem inovadora ao criar um sistema integrado de controlo e gestão da produção com o auxílio de simulação virtual para fins didáticos. O resultado da dissertação restringe-se a um fio condutor ao longo de toda a estrutura dos sistemas integrados, pelo que existe um enorme potencial ainda por explorar nesta área.

Tendo em conta que a aplicação cmNavigo apresenta funcionalidades que vão para além de um MES existem muitas possibilidades de desenvolvimento sobre a integração das áreas de controlo da qualidade e controlo da manutenção.

O mesmo ocorre na área de controlo das operações de qualidade, mais concretamente, na troca de informação, quando um material é rejeitado por falhar os requisitos de qualidade, como quando um equipamento automático deteta um problema no material e rejeita material, eliminando-o da simulação, implicando que esta alteração tenha de ser comunicada ao cmNavigo, de modo a criar uma nova instância de material e realizar uma análise estatística dos vários processos realizados.

Na área de controlo das operações de manutenção havendo a possibilidade de introduzir falhas nos dispositivos do Factory I/O, simulando avarias, que despoletam alarmes no controlo direto e de supervisão, existindo a necessidade de comunicar estas ocorrências ao cmNavigo sobre a forma de relatórios. A comunicação da alteração do estado da máquina deverá desencadear uma rota dos materiais, sendo os materiais da unidade redirecionados.

Foi possível concluir que o Factory I/O apresenta uma plataforma muito interessante para o ensino de sistemas integrados de controlo e gestão da produção. Propõe-se o desenvolvimento de um objeto adicional na aplicação que substitua as caixas de cartão, um vez que estas não se adequam visualmente às operações de produção. Este objeto poderá apresentar uma forma simples, como um retângulo, sendo necessário que se possam realizar operações de Assembly, utilizando o equipamento *Pick&Place* do Factory I/O e um objeto adicional semelhante ao *Emitter* e o *Remover* que altere as propriedades do objeto, como por exemplo a cor.

5.2 Balanço Pessoal

A nível pessoal, esta dissertação foi muito interessante e permitiu-me adquirir conhecimentos valiosos nas áreas de automação industrial, produção e programação. Mantive-me interessado e motivado por saber que o tema abordado é relevante na indústria.

A parceria com as empresas Real Games e a Critical Manufacturing foram também fatores de motivação e inspiração no decurso do trabalho realizado, tendo aprendido muito. Por outro lado a boa disposição, a entejuda e o profissionalismo ficarão sem dúvida marcados para sempre por esta experiência única.

O desenvolvimento desta tese conduziu a aprendizagem de importantes conceitos relacionados com a integração de aplicações, com automatismos industriais, com aplicações de gestão na indústria.

Para concluir é de destacar o desenvolvimento de outras capacidades pessoais como pensamento crítico ou a capacidade de resolução de problemas e muitas outras.

6 Referências Bibliográficas

1. Crowley, B., *MANUFACTURING EXECUTION SYSTEMS VS. ERP/MRP*. The Global Assembly Journal for SMT and Advanced Packaging Professionals, 2011. 11.
2. Institute, M.G., *THE INTERNET OF THINGS: MAPPING THE VALUE BEYOND THE HYPE*. June 2015, McKinsey&Company.
3. Reekmans, G.N.a.M. *OPEN INFORMATION EXCHANGE FOR FLEXIBLE MANUFACTURING*. 2013; Available from: http://ima-research.org/site/download/inexflex_scope_20130107.pdf.
4. Vasconcelos, D., *Emulação em tempo real de VFD (Variable-Frequency Drive), Células de Carga e Cortinas de Luz*, in *DEMec*. 2012, Universidade do Porto: FEUP.
5. Inc., A.I.S. <http://www.aispro.com/products/widescreen-touch-screen-panel-pc>. 2015.
6. Groover, M.P., *Automation, production systems, and computer-integrated manufacturing*. 3rd ed. 2008, Upper Saddle River, N.J.: Prentice Hall. xvi, 815 p.
7. Louis, P., *Manufacturing Execution Systems: Grundlagen und Auswahl*. 2009: Gabler Verlag.
8. Shell, R.L. and E.L. Hall, *Handbook of industrial automation*. 2000, New York: Marcel Dekker. xi, 887 p.
9. Gifford, C., *The Hitchhiker's Guide to Operations Management: ISA-95 Best Practices Book 1.0*. 2007: ISA.
10. *Applying Manufacturing Operations Models in a Discrete Hybrid Environment*. 2013: MESA International.
11. Scholten, B. *Integrating ISA-88 and ISA-95*. 2007.
12. Pereira, M., *Padrões de Desenho em Soluções Programadas de Automação Flexível*, in *DEMec*. 2011, FEUP.
13. Nelson, P.R. and R.S. Shull, *Organizing for an initial implementation of S88*. *ISA Transactions*, 1997. **36**(3): p. 189-195.
14. Mehta, B.R. and Y.J. Reddy, *Chapter 1 - Industrial automation*, in *Industrial Process Automation Systems*, B.R.M.J. Reddy, Editor. 2015, Butterworth-Heinemann: Oxford. p. 1-36.
15. MacDougall, W. *Industrie 4.0 Smart Manufacturing For The Future*. 2014.
16. Wang, L., M. Törngren, and M. Onori, *Current status and advancement of cyber-physical systems in manufacturing*. *Journal of Manufacturing Systems*.
17. Sousa, C.A.G.d., *Análise comparativa de tecnologias de baixo custo para reconhecimento de padrões em sistemas automáticos*. 2011, Universidade do Porto.
18. Monteiro, J.P.F., *Emulação em tempo real de Leitores de RFID e de Código de Barras*. 2012, Universidade do Porto.
19. Knapp, E., *Chapter 5 - How Industrial Networks Operate*, in *Industrial Network Security*, E. Knapp, Editor. 2011, Syngress: Boston. p. 89-110.
20. Mehta, B.R. and Y.J. Reddy, *Chapter 7 - SCADA systems*, in *Industrial Process Automation Systems*, B.R.M.J. Reddy, Editor. 2015, Butterworth-Heinemann: Oxford. p. 237-300.
21. Harrison, T.P., H.L. Lee, and J.J. Neale, *The Practice of Supply Chain Management: Where Theory and Application Converge*. 2003: Springer US.
22. Silva, L.M., *API de Serviços Web e Extensão para Software de Webanalytics/adversing*. 2009, Universidade do Porto.
23. Thuraisingham, W.S.a.B. *Security for Enterprise Resource Planning Systems*. 2007. DOI: 10.1080/10658980701401959.
24. Knapp, E., *Chapter 2 - About Industrial Networks*, in *Industrial Network Security*, E. Knapp, Editor. 2011, Syngress: Boston. p. 7-29.

25. Knapp, E.D. and J.T. Langill, *Chapter 4 - Introduction to Industrial Control Systems and Operations*, in *Industrial Network Security (Second Edition)*, E.D.K.T. Langill, Editor. 2015, Syngress: Boston. p. 59-84.
26. Magalhães, A. *FEATURES AND BENEFITS OF VIRTUAL SYSTEMS IN CONTROL EDUCATION*.
27. <http://www.realgames.pt/factory-io/>. [cited 2015].
28. Vogel-Heuser, B., et al., *Model-driven engineering of Manufacturing Automation Software Projects - A SysML-based approach*. *Mechatronics*, 2014. **24**(7): p. 883-897.
29. <http://www.wibu.com/codemeter-secures-codesys-applications2.html>. 2015.
30. Olson, D.L. and J. Staley, *Case study of open-source enterprise resource planning implementation in a small business*. *Enterprise Information Systems*, 2011. **6**(1): p. 79-94.
31. *MES Product Survey 2014*, C.G. INC, Editor. 2014.
32. *Top 10 Enterprise Resource Planning (ERP) Vendors*.
33. <http://www.opentaps.org/>. 2015.
34. http://www.opentaps.org/docs/index.php/Material_Resource_Planning. 2015.
35. Knapp, E., *Chapter 4 - Industrial Network Protocols*, in *Industrial Network Security*, E. Knapp, Editor. 2011, Syngress: Boston. p. 55-87.
36. <http://www.webopedia.com/TERM/Q/query.html>. 2015.

Anexos

Aplicações MES

Este anexo apresenta uma lista de todas as aplicações MES que cumprem a norma ISA-95 à data de publicação da tese, retirada do MES Product Survey 2014 realizado pela CGI. Para obter informações mais detalhas sobre cada aplicação poderá aceder ao *site* do vendedor.

No.	Product	Vendor
1	Aegis FactoryLogix	Aegis Software
2	A-MES	Advanced Manufacturing Solutions (AMS)
3	Ampla	Schneider Electric
4	APIS	Prediktor AS
5	AquiWEB	Astrée Software
6	aspenONE	Aspen Technology
7	ATS CM4D	ATS International B.V.
8	ATS Inspect	ATS International B.V.
9	ATS Intelligence	ATS International B.V.
10	Broner MES	Broner Metals Solutions
11	Camstar Manufacturing	Camstar Systems, Inc.
12	CAS-MES	Shenyang Institute of Automation (SIA)
13	Clinicopia	Catalent CTS Informatics, Inc
14	cmNavigo	Critical Manufacturing, S.A.
15	COOX / MESbox	ORDINAL Software
16	cpmPlus	ABB Ltd
17	D2000 Entis	IPESOFT s.r.o.
18	DABOM	ACS Co., Ltd.
19	Delfoi Planner	Delfoi Oy/Ltd
20	DELMIA Apriso	Dassault Systèmes
21	DIAMES	CSM Systems AG
22	edinn M2	edinn global, s.l.
23	Evolutio	Siemens Industry Software Ltda
24	Eyelit Manufacturing™	Eyelit Inc.
25	EZ-MES	EazyWorks LLC
26	Factory Framework	FORCAM GmbH
27	FactoryTalk	Rockwell Automation
28	Floormation	Infinite Functions, Inc.
29	Hydra MES	MPDV Mikrolab GmbH
30	IBS:prisma MES	IBS AG
31	Interax	CIMx Software
32	Intuition	Honeywell Process Solutions
33	iTAC.MES.Suite	iTAC Software AG
34	Krones MES	Krones AG
35	Legato	Gefasoft AG

36	LG CNS ezMES	LG CNS
37	Marco Trac-IT	Marco Limited
38	MEScontrol.net	BrightEye
39	MESplus	Miracom Inc Co., Ltd.
40	Mestec SmartBox	Mestec Limited
41	MOM4	NearSoft Europe GmbH
42	MV2	Paper-Less, LLC
43	nanoTrack	SAMSUNG SDS Co., Ltd
44	nMetric Suite	nMetric™, LLC
45	Nova-LIMS	Novatek International
46	Objective	Objective
47	Operator	Operator Systems ApS
48	PAS-X	Werum IT Solutions GmbH
49	PC-Factory MES	PPI-Multitask Sistemas e Automação Ltda
50	Performix xMES	Performix Inc.
51	PI System	OSIsoft, LLC
52	PILOT Suite	Felten GmbH
53	Plant iT	ProLeiT AG
54	PlantMaster	Belgian Monitoring Systems Bvba (BMSvision)
55	POMSnet	POMS Corporation
56	PRODAC	PRODAC Systems - ENDIMO ltd.
57	Proficy for Manufacturing	GE Intelligent Platforms
58	ProMANAGE	Doruk Automation and Software Inc.
59	PSI metals	PSI Metals GmbH
60	Quartis Optima	BODET-OSYS
61	Qubes	Creative IT
62	SAP ME	SAP AG
63	SAP MII	SAP AG
64	Shopfloor-Online	Lighthouse Systems Ltd
65	SIMATIC IT	Siemens AG
66	Solumina	iBASEt
67	SWD-PES	SW-Development Ltd
68	Syncade Suite	Emerson Process Management
69	Traceability Made Easy	MASS Group, Inc.
70	TrakSYS	Parsec Automation Corp.
71	Veri95	Inxites NV
72	Wonderware MES	Invensys
73	Yokogawa MIPP	Yokogawa Electric Corporation

Aplicações ERP *Open-Source*

Este anexo apresenta uma lista das aplicações ERP *Open-Source* analisadas no decorrer da dissertação, na qual consta alguma informação técnica acerca das aplicações e alguns comentários considerados oportunos. Para obter informações mais detalhadas sobre cada uma poderá aceder ao *site* de cada aplicação.

Nome			Comentários
Odoo	Linguagem	Python	Antigo OpenERP Bastante completo, embora seja Open-Source, apresenta um modelo comercial, a informação acerca da aplicação é reservada.
	Base de dados	PostgreSQL	
	Licença	AGPLv3	
	Lançamento	2014	
Opentaps	Linguagem	Java	Bastante completo e com Web-Store integrada. Apresenta casos de estudo com aplicação na Toyota e Honeywell. Apresentada uma Wiki com documentação acerca da aplicação e Forums.
	Base de dados	MySQL/PostgreSQL	
	Licença		
	Lançamento	2008	
Dolibarr	Linguagem	PHP	ERP e CRM. Simples, contêm uma aplicações para android, mas não possui módulo de produção.
	Base de dados	MySQL	
	Licença	GPLv3	
	Lançamento		
ERP5	Linguagem	Python	Bastante completo. Apresenta um modelo SaaS (Software as a Service) semelhante ao Odoo.
	Base de dados	MariaDB /MySQL	
	Licença	GPL	
	Lançamento	2002	
ERPNext	Linguagem	Python	Semelhante ao Odoo
	Base de dados	MySQL	
	Licença	GPLv2	
	Lançamento	2008	
inoERP	Linguagem	Java	Possui WebStore e Produção. Contêm pouca documentação.
	Base de dados	MySQL	
	Licença	MPL	
	Lançamento		
Adempiere	Linguagem	Java	Módulos ERP,SCM,CRM, Web Store integrada. Pouca documentação disponível.
	Base de dados		
	Licença	GPLv2	
	Lançamento	9/07/2006	
Idempiere	Linguagem	Java	Módulos ERP SCM, CRM
	Base de dados		
	Licença	GPLv2	
	Lançamento	31/09/2012	
WebERP	Linguagem	PHP	Não é instalado no PC, corre num Web-Browser. Produz Relatórios através de PDF. Pouca documentação.
	Base de dados	MySQL	
	Licença	GPLv2	
	Lançamento		