



**JOÃO JOSÉ VEIGA
VIVEIROS**

CIP – CONTROLO INTEGRADO DE PRODUÇÃO



**JOÃO JOSÉ VEIGA
VIVEIROS**

CIP – CONTROLO INTEGRADO DE PRODUÇÃO

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica realizada sob a orientação científica do Doutor José Paulo Santos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Dedico este trabalho à minha esposa e família pelo incansável apoio.

o júri

Presidente

Prof. Doutor António Gil D'Orey de Andrade Campos
Professor auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

Prof. Doutor Joaquim José Borges Gouveia
Professor Catedrático, Convidado Aposentado da Universidade de Aveiro

Prof. Doutor José Paulo Oliveira Santos
Professor auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

agradecimentos

Deixo aqui os meus agradecimentos ao meu orientador, Prof. Dr. José Paulo Santos pela sua participação activa no desenvolvimento do projecto relacionado com esta dissertação.

Aos meus pais, esposa e amigos por me terem apoiado ao longo de todos estes anos.

palavras-chave

Controlo integrado, monitorização de sistemas, ambientes industriais, ambiente de comunicação, integração de sistemas em bases de dados, sistemas *SCADA*, *Modbus*, *TCP/IP*.

resumo

O presente trabalho procura o desenvolvimento de um sistema de comunicação entre equipamentos industriais recorrendo a protocolos de comunicação *Modbus* e *TCP/IP*. Procura-se criar aplicações *SCADA* para controlo, monitorização e armazenamento de informação através de uma base de dados. Pretende-se monitorizar e armazenar o histórico do estado de variáveis de natureza eléctrica. A partir destas variáveis podemos realizar estudos para redução do consumo energético de equipamentos.

keywords

Integrated control, monitoring systems, industrial environments, communication environments, database integration systems, SCADA systems, *Modbus*, *TCP/IP*.

abstract

This work seeks to develop a communication system between industrial devices using *Modbus* and *TCP/IP* protocols. The aim is to create *SCADA* applications for control and monitoring using a database to record the history of data. It is intended to monitor and store the history of variables of electrical nature. From this variables we can perform studies to reduce equipment energy consumption.

ÍNDICE

Capítulo 1 – Introdução	1
1.1 Enquadramento global	1
1.2 Identificação do problema	2
1.3 Objectivos principais	3
1.4 Estrutura da Dissertação	4
Capítulo 2 – Sistemas integrados e tecnologias de apoio	7
2.1 Conceitos e enquadramento histórico	7
2.2 Protocolos de comunicação mais utilizados	8
2.2.1 <i>TCP/IP</i>	8
2.2.2 <i>HTTP</i>	9
2.2.3 <i>Web Services</i>	10
2.3.4 <i>Modbus</i>	10
2.3.5 <i>RS-485</i>	12
2.3 Trabalhos na área de integração de sistemas	13
2.3.1 Serviços Web: Supervisão e controlo de sistemas de produção	13
2.3.2 Integração de sistemas de produção	15
2.3.3 Sistema de gestão de produção para ambientes industriais	19
2.3.4 “ <i>SISAL – Um sistema supervisorio para elevação artificial de petróleo</i> ”	21
2.3.5 “ <i>Sistema Supervisorio de Aquisição de Dados em um Protótipo de Estação de Tratamento de Efluentes</i> ”	22
2.3.6 Integração da <i>WEB 2.0</i> no Desenvolvimento de Aplicações em Ambiente Industrial	24
2.3.7 Artigo “ <i>Utilities Control</i> ”	26
2.3.8 Artigo “ <i>SUBSEA MONITORING - Monitoring in a harsh environment</i> ”	28
2.3.9 Artigo “ <i>Subsea Control</i> ”	30
2.3.10 <i>Open-Source Implementation of Monitoring and Controlling Services for EMS/SCADA Systems by Means of Web Services – IEC 61850 and IEC 61970 Standards</i>	33
2.3.11 <i>Web services-based automation for the control and monitoring of production systems</i>	36
2.3.12 Planeamento e controlo de produção na <i>Euronete, S.A.</i>	39
2.4 Soluções <i>SCADA/HMI</i>	44
2.4.1 <i>WinCC</i>	45
2.4.2 <i>Lookout</i>	46

2.4.3	<i>Movicon</i>	47
2.4.4	Movicon Web Client (iOS)	48
2.5	Análise e escolha de soluções	49
Capítulo 3 - Proposta de uma nova solução		51
3.1	Arquitetura da solução proposta	51
Capítulo 4 - Implementação da solução proposta.....		55
4.1	<i>Hardware</i> envolvido na implementação	55
4.1.1	<i>Moxa Nport 5630</i>	55
4.1.2	<i>WM22-DIN (Three-Phase Power Analyser)</i>	56
4.1.3	Controlador Lógico Programável (<i>PLC FBS-20MC</i>).....	58
4.1.4	Variador de Frequência <i>D720-S-070SC</i>	59
4.1.5	Motor Monofásico	61
4.1.6	Compressor 1.5 <i>HP</i>	62
4.1.7	Transmissor de pressão <i>Jumo MIDAS</i>	63
4.1.8	Caudalímetro	64
4.1.9	Outro <i>hardware</i>	66
4.2	Sistema em estudo	67
4.2.1	Configuração e teste do <i>WM22-DIN</i>	67
4.2.2	Configuração e teste do <i>Moxa Nport 5630</i>	70
4.2.3	Aplicação em <i>Visual Studio 2012</i>	71
4.2.4	Base de dados <i>Oracle 11g</i>	74
4.2.4.1	Como aceder	75
4.2.4.2	<i>Oracle SQL Developer</i>	76
4.2.4.3	<i>Oracle Enterprise Manager</i>	77
4.2.5	Aplicação <i>Movicon</i>	79
4.2.5.1	Configuração <i>Nport - Movicon</i>	82
4.3	Análise de resultados	85
4.3.1	Método <i>ON-OFF</i>	87
4.3.2	Método do Intervalo de Frequências	89
Capítulo 5 - Conclusões e trabalho futuro		93
5.1	Conclusões	93
5.2	Trabalho futuro	94
Bibliografia		96

ÍNDICE DE FIGURAS

Figura 1.1 – Ilustração simbólica do problema em questão.	3
Figura 2.1 - Composição de uma mensagem tipo do protocolo <i>Modbus</i> . [5].....	12
Figura 2.2 - Arquitectura esquemática da solução proposta. [7].....	14
Figura 2.3 - Recurso a controlar. [7]	15
Figura 2.4– Interface de controlo da aplicação <i>SCADA</i> . [7].....	15
Figura 2.5 - Solução proposta para o caso de estudo 1. [4]	16
Figura 2.6 - Implementação da solução em página <i>Web</i> , <i>Windows</i> e <i>PDA</i> . [4]	17
Figura 2.7 - Arquitectura da solução proposta para o caso de estudo 2. [4].....	17
Figura 2.8 - Aplicação implementada no caso de estudo 2. [4]	17
Figura 2.9 - Esquema da solução proposta para o caso de estudo 3. [4]	18
Figura 2.10 - Aplicação <i>CIP</i> para implementação da solução do caso de estudo 3. [4].....	18
Figura 2.11 - Diagrama de implementação. [8].....	20
Figura 2.12 - Interface do menu de documentação. [8]	20
Figura 2.13 - Arquitectura do <i>SISAL</i> . [9]	21
Figura 2.14 - Ambiente de automação com o <i>SISAL</i> . [9]	22
Figura 2.15 - Interface de supervisão e controlo. [10]	23
Figura 2.16 - Protótipo da <i>ETAR</i> a monitorizar. [10]	24
Figura 2.17 - Arquitectura do sistema. [11].....	25
Figura 2.18 - Interface <i>HMI</i> desenvolvida no sistema <i>GERINF</i> . [11].....	26
Figura 2.19 - Sistema <i>SCADA</i> desenvolvido. [12]	27
Figura 2.20 - Rede <i>TCP/IP</i> do sistema e equipamentos intervenientes. [12].....	28
Figura 2.21 - Montagem típica no oceano. Esquema de comunicação entre <i>CPS</i> 's engatados ao longo dos vãos do gasoduto e da torre <i>MSP</i> . [13].....	29
Figura 2.22 - Montagem de <i>CSP</i> 's no gasoduto pelo <i>ROV</i> . [13]	30
Figura 2.23 - Esquema integrado de detecção e diagnóstico de falhas.	32
Figura 2.24 - Esquema de detecção de falhas. [14]	32
Figura 2.25 - Sistema inteligente de supervisão. [14].....	33
Figura 2.26 - Proposta baseada em serviços <i>WEB</i> e nos padrões <i>IEC (International Electrotechnical Commission)</i>	34
Figura 2.27 - Modo de interacção. [15].....	35
Figura 2.28 - Integração da estrutura básica dos serviços <i>WEB</i> – <i>SOA</i> [16].....	37

Figura 2.29 - Arquitectura do software de serviços <i>WEB</i> do componente. [16].....	38
Figura 2.30 - Aplicação do mecanismo de monitorização e controlo baseada em HMI.	39
Figura 2.31 - Esquematização do sistema <i>MRP II</i> implementado. [17].....	41
Figura 2.32 - Exemplo de uma consulta na interface gráfica da ferramenta. [17].....	43
Figura 2.33 - Diagrama de decisão com árvore de probabilidade da ferramenta. [17].....	43
Figura 2.34 - Interface gráfica em <i>WinCC</i> . [20]	45
Figura 2.35 - Interface gráfica construída com o <i>Lookout</i> . [22].....	46
Figura 2.36 - Interface gráfica criada no <i>Movicon</i> . [24].....	47
Figura 2.37 - Painel de configuração do <i>Movicon Web Client</i> para <i>iOS</i>	48
Figura 2.38 – Menus: inicial, gestão de energia, alarmes e controlo de processo para <i>iOS</i>	49
Figura 3.1 - Arquitectura global da solução proposta.	52
Figura 4.1 – Legenda do <i>Nport5630</i> . [30]	56
Figura 4.2 – <i>WM22-DIN</i> . [32]	57
Figura 4.3 – Legenda de entradas e saídas do analisador de energia <i>WM22-DIN</i> . [32].....	57
Figura 4.4 – <i>PLC – FATEK</i> modelo <i>FBs-20MC</i>	58
Figura 4.5 - Variador de Frequência <i>Mitsubishi</i> da série <i>D700-SC</i>	60
Figura 4.6 – Motor eléctrico monofásico do compressor utilizado.	61
Figura 4.7 – Modelo do compressor utilizado na implementação. [43]	62
Figura 4.8 – a) Sensor de pressão utilizado; b) Ligação eléctrica do Sensor.	63
Figura 4.9 – Adaptação do caudalímetro ao sistema de ar comprimido.	64
Figura 4.10 – Caudalímetro e esquema eléctrico do circuito.	65
Figura 4.11 - Outros componentes eléctricos utilizados.....	66
Figura 4.12 – Ligação <i>RS-485</i> por 2 fios. [31].....	67
Figura 4.13 – Exemplificação da mensagem enviada para o <i>WM22</i>	68
Figura 4.14 – Aplicação <i>VB</i> de testes para a comunicação <i>Modbus</i> com o conversor <i>RS-485</i>	69
Figura 4.15 – Mensagem <i>Modbus</i> enviada e respectiva resposta observada no <i>Picoscope</i>	69
Figura 4.16 – Modo de comunicação Servidor <i>TCP</i> . [30]	70
Figura 4.17 – Troca de mensagens entre aplicação <i>VB</i> e <i>WM22</i>	72
Figura 4.18 – Janela principal da aplicação em <i>VB</i> para a comunicação <i>TCP/IP</i>	73
Figura 4.19 – Aplicação em <i>Visual Studio 2012</i> – Versão melhorada.....	74
Figura 4.20 – Verificação dos dados através do <i>Oracle SQL Developer</i>	77
Figura 4.21 - Verificação das trocas de informação no sistema através do <i>Oracle Enterprise Manager</i>	78

Figura 4.22 – Aplicação em <i>Movicon</i> para controlo e monitorização do sistema de ar comprimido.....	80
Figura 4.23 – Página de leitura de variáveis eléctricas do WM22.....	80
Figura 4.24 – Tabela de valores estudados no <i>Movicon</i>	81
Figura 4.25 – Gráfico obtido no <i>Movicon</i>	81
Figura 4.26 – Mensagem <i>Modbus TCP</i> enviada para o WM22.....	83
Figura 4.27 – Mensagem <i>Modbus RTU</i> enviada para o WM22.....	83
Figura 4.28 – Painel de configuração do <i>Nport</i> e parâmetros utilizados.....	84
Figura 4.29 – Configuração de estação e de tarefa no <i>Movicon</i>	84
Figura 4.30 – Circuito eléctrico desenvolvido.....	85
Figura 4.31 – Informação da base de dados obtida na aplicação <i>VB</i>	86
Figura 4.32 – Gráfico do consumo global/médio do compressor e respectiva pressão de ar no sistema de ar comprimido – Método <i>ON-OFF</i>	88
Figura 4.33 - Gráfico do consumo global/médio do compressor e respectiva pressão de ar no sistema de ar comprimido – Método do Intervalo de Frequências.....	90

Capítulo 1 – Introdução

O mercado actual é cada vez mais competitivo, de modo que é necessário atingir determinados objectivos de produção mais rápido, no menor prazo e com o menor número de encargos possível, seja de mão-de-obra, seja de recursos energéticos.

O aparecimento de novas tecnologias torna possível que os processos produtivos sejam mais rápidos e eficientes recorrendo à integração de sistemas e à automação.

Torna-se essencial dominar as novas tecnologias através da centralização de informação e da redução do seu consumo. Deste modo consegue-se rapidez e baixo custo de produção, tornando as empresas mais competitivas.

Existem um leque de ferramentas que deve ser estudado para alcançar esta eficiência produtiva, tais como *softwares SCADA/HMI*, bases de dados, protocolos de comunicação entre equipamentos, etc.

1.1 Enquadramento global

A automação é importante para qualquer sistema industrial, pois é necessário que haja um controlo do ambiente fabril no que diz respeito a contagem de peças e/ou recursos, alerta de problemas em diversos postos de trabalho, monitorizar e identificar falhas na produção ou ainda para otimizar consumos de energia. Esta necessidade obriga a estudar detalhadamente as infra-estruturas de comunicação e as mensagens que os novos recursos fabris a integrar reconhecem. Actualmente é essencial conhecer os equipamentos de determinado ambiente industrial, não só a nível de funcionalidades, escalonamento de produção, optimização de tempos de processo, mas também ao nível de consumos para se otimizar os gastos de determinada empresa. O mercado actual cada vez se encontra mais competitivo não só no que diz respeito à procura de um produto,

mas também no que diz respeito aos gastos energéticos no final do mês. Numa grande empresa, esta variante é responsável pelo gasto de alguns milhares de euros que com o devido estudo pode ser reduzido.

Neste trabalho, o recurso principal a integrar terá a função de indicar variáveis de natureza eléctrica de determinados equipamentos que a ele se ligam para realizar um registo de consumos acessível através de um computador para facilitar a monitorização. A integração de dispositivos de leitura, monitorização e controlo de variáveis desta natureza pode trazer grandes benefícios económicos.

1.2 Identificação do problema

Esta dissertação procura desenvolver um conjunto de aplicações em conjunto com a utilização de alguns *softwares* (a seleccionar) para solucionar problemas no âmbito da integração de sistemas, armazenamento de informação em bases de dados e estudo de poupança energética num ambiente industrial.

Este estudo foca-se, mais concretamente nos compressores do sistema de ar comprimido. Este estudo de consumo de energia é importante, pois pode-se estar a utilizar energia que não está a ser aproveitada. É necessário avaliar um conjunto de soluções de integração de sistemas, estudo de interfaces *SCADA/HMI* e protocolos de comunicação entre equipamentos utilizados na actualidade. O estudo destas ferramentas é essencial para se simular um ambiente industrial. A **Figura 1.1** ilustra alguns dos componentes que precisamos de integrar para realizar este estudo.



Figura 1.1 – Ilustração simbólica do problema em questão.

A secção seguinte explica em maior detalhe o que se pretende solucionar para este problema.

1.3 Objectivos principais

Neste contexto, pretende-se abordar a integração de sistemas, realizar um levantamento do estado da arte, realizar uma análise comparativa de várias tecnologias comuns e emergentes, propor uma solução a partir dela, e por fim implementar soluções particulares para sistemas integrados de produção.

É essencial simular uma pequena rede industrial, recorrendo à escolha de alguns equipamentos disponíveis para registo de variáveis de natureza eléctrica, com o intuito de se fazer a leitura das mesmas de forma automática e contínua. Procura-se centralizar estes dados numa unidade principal e manter o seu histórico numa base de dados. A facilidade de implementação será fundamental para reduzir custos, assim como a escolha de *softwares* que permitam a integração de sistemas de comunicação padrão para facilitar a elaboração da solução.

O objectivo principal será a redução dos consumos de energia dos compressores do sistema de ar comprimido em ambiente industrial, conseguindo que a pressão e caudal se mantenham nos níveis óptimos de funcionamento.

1.4 Estrutura da Dissertação

Esta dissertação está dividida 5 capítulos que fundamentam o tema em estudo. Nesta secção destaca-se o conteúdo dos capítulos deste documento.

Cap. 1 – Introdução

Descreve o enquadramento da dissertação na actualidade. É ainda identificado o problema em estudo, principais objectivos e a estrutura da dissertação.

Cap. 2 – Estado da arte

Este capítulo trata a problemática dos sistemas integrados de produção no que diz respeito a soluções existentes, propostas e implementações na actualidade por outros autores de forma a se ter uma ideia do panorama geral em que se encontra a indústria no sector. Fala-se ainda sobre algumas tecnologias utilizadas neste âmbito e em alguns *softwares* que as podem auxiliar e servir de suporte ao trabalho proposto. Finaliza-se este capítulo focando as escolhas de *software*, *hardware* e meios de comunicação.

Cap. 3 – Proposta de uma nova solução

Descreve a arquitectura da solução proposta para a integração dos equipamentos necessários à leitura de variáveis de natureza eléctrica intrínsecas ao estudo da gestão dos consumos dos compressores do sistema de ar comprimido em ambiente industrial.

Cap. 4 – Implementação da solução

Descreve em detalhe a implementação da solução proposta e todos os aspectos relevantes dos *softwares* escolhidos, assim como dos dispositivos físicos e aplicações desenvolvidas no trabalho. Neste capítulo aborda-se de forma detalhada as configurações de *hardware*, *software*, comunicações e base de dados.

Cap. 5 – Conclusões e trabalho futuro

Conclusões tiradas a partir da implementação do sistema e sobre a globalidade da dissertação. Trata ainda de assuntos relacionados com perspectivas de trabalho futuro sobre o tema desenvolvido.

Capítulo 2 – Sistemas integrados e tecnologias de apoio

2.1 Conceitos e enquadramento histórico

O controlo integrado da produção (*CIP*) baseia-se em sistemas de informação que integram todos os dados e processos de determinada empresa em um único sistema. Este sistema consiste num *software* desenvolvido para integrar os diversos departamentos, processos de fabrico e/ou equipamentos físicos num determinado ambiente fabril recorrendo à automação e armazenamento de todas as informações e dados envolventes.

Na década de 70, com a evolução económica e computacional, surgiu os *MRP* (Planeamento da requisição de materiais) que possibilitavam o planeamento de bens, serviços e a administração dos mais diversos processos produtivos.

As redes de computador ligadas a servidores surgiram na década de 80 o que permitiu a transformação do *MRP* em *MRP II* (planeamento de recursos de manufactura) que possibilita também o controlo de mão-de-obra e do conjunto de máquinas envolvidas no processo de fabrico. Mais tarde nesta década o *MRP II* passou a ser denominado de *ERP* (*Enterprise Resource Planning*) devido a sua abrangência de controlo e gestão. A agilização de processos e comunicação entre departamentos de índole administrativa e de apoio à produção entraram na era da automação.

O *ERP* evoluiu bastante na década de 90 devido a um grande desenvolvimento tecnológico nas redes de comunicação entre computadores e na arquitectura cliente/servidor feito agora por microcomputadores ligados a servidores com preços muito mais competitivos que os antigos “*Mainframes*” (primeiros computadores de grande porte com grande capacidade de processamento). [\[1\]](#)

Actualmente o controlo de consumos também se encontra num patamar de grande importância. Com a diminuição de recursos naturais no nosso planeta, a subida

de preços das fontes de energia tem-se acentuado, e com isso a gestão dos consumos energéticos em qualquer empresa começa a ser cada vez mais importante.

A necessidade desta centralização de informação inerente à evolução tecnológica implica o estudo de diversas formas de comunicação informática. Dada a necessidade deste estudo, a secção seguinte aborda alguns protocolos de comunicação existentes com o intuito de os aplicar na solução do problema exposto no capítulo anterior.

2.2 Protocolos de comunicação mais utilizados

Nesta secção são abordadas algumas das tecnologias mais usuais e/ou relevantes para o desenvolvimento da solução proposta.

2.2.1 *TCP/IP*

O conjunto de protocolos denominado por *TCP/IP* é responsável pela comunicação entre computadores e equipamentos em rede. O seu nome provém de dois protocolos, o *TCP* (Protocolo de Controlo de Transmissão) e o *IP* (Protocolo de Interconexão). Consiste num modelo de camadas (Modelo *OSI*), cada uma responsável por um conjunto de tarefas, fornecendo um conjunto de serviços à camada superior.

As camadas superiores ou camadas de aplicação estão logicamente mais perto do utilizador lidando com dados mais abstractos, enquanto as camadas inferiores lidam com tarefas de baixo nível de abstracção.

Considerado um protocolo pesado em termos de memória e *hardware*, o protocolo *TCP/IP* tornou-se indispensável com a evolução dos computadores pois é muito robusto, isto é, escalável, multiplataforma, com estrutura para sistemas operacionais do tipo cliente/servidor entre dois pontos distantes com grau elevado de

confiabilidade. Um dos grandes benefícios, é a sua padronização, isto é, todos os sistemas operacionais actualmente oferecem o suporte para o *TCP/IP* e a maioria das grandes redes se baseia nele para a maioria do tráfego de dados. Esta tecnologia permite interconectividade entre sistemas não similares para aceder e transferir dados entre eles, tais como, o *FTP (File Transfer Protocol)* e *Telnet (Terminal Emulation Protocol)*. Em relação ao roteamento, permite que ambas as tecnologias antigas e modernas se conectem à internet e trabalhem com as interfaces mais utilizadas pelos sistemas operacionais como *Windows Sockets* e *NetBIOS*. [2]

2.2.2 HTTP

O *HTTP* (Protocolo de Transferência de Hipertexto, é o protocolo de comunicação de dados base da *World Wide Web*. Ao texto estruturado que utiliza ligações lógicas (*hyperlinks* - hiperligações) entre “nós” contendo texto dá-se o nome de Hipertexto. Uma hiperligação consiste numa referência a um documento na *web*. Estes documentos encontram-se num sítio com endereço de página na internet que para acesso é necessário digitar o *URI* (Identificador Universal de Recurso) num browser (Navegador *Web*) através do sistema operativo de um computador, por exemplo.

Num modelo cliente/servidor, utiliza-se este protocolo pois funciona através de requisição resposta, isto é, o cliente (*Browser*) envia uma mensagem de requisição *HTTP* ao servidor e este vai fornecer recursos como arquivos *HTML* (Linguagem de Marcação de Hipertexto – Linguagem utilizada para produzir páginas *Web*) e outros como mensagem de resposta ao cliente. Existem mais agentes cliente que podemos utilizar além de um *browser*, tais como aplicações móveis, navegadores vocais, etc. Hoje em dia, qualquer dispositivo capaz de exibir conteúdo *Web* pode agir como agente utilizador.

O *HTTP* é projectado para melhorar ou habilitar a comunicação entre clientes e servidores. Os navegadores *Web* armazenam em cachê alguns dos recursos previamente acedidos e reutilizam-nos quando requisitados novamente para reduzir o tráfego de rede. [3]

2.2.3 *Web Services*

Os *web services* são muito utilizados na integração de sistemas porque permitem a comunicação entre aplicações com linguagens diferentes e permitem que estas enviem e recebam dados em formato *XML* (*Extensible Markup Language* - linguagem universal) através da *Internet*. Esta normalização de recursos da aplicação disponível na rede, faz com que determinado processo seja mais ágil e eficiente. Outras tecnologias também o conseguem fazer, como os navegadores *web* que recorrem ao *HTTP* e *HTML*, porém estas não são tão bem sucedidas na comunicação e integração de aplicações como os serviços *web*. Esta capacidade de operar com outras aplicações e/ou sistemas diferentes com linguagens diferentes permite que os seus recursos estejam disponíveis para qualquer aplicação cliente que necessite de extrair recursos fornecidos pelo *Web Service*. O facto deste modelo se basear em tecnologias normalizadas e abertas como o *XML*, *HTTP*, *SOAP* (*Simple Object Access Protocol*), *WSDL* (*Web Services Description Language*) e *UDDI* (*Universal Description, Discovery and Integration*), faz com que ele seja muito utilizado.

Recorre-se aos *web services* para facilitar a integração das aplicações de determinada empresa, como por exemplo a comunicação electrónica entre clientes e fornecedores. Têm também a vantagem de serem acessíveis como componentes em qualquer lugar da internet, entre outras. [\[4\]](#)

2.3.4 *Modbus*

O *Modbus* é um protocolo de comunicação de dados criado pela *Modicon* e é muito utilizado em sistemas de automação na indústria. Este sistema é dos mais

antigos e utilizados da actualidade em redes de *PLC's* (Controladores Lógicos Programáveis) para aquisição de sinais provenientes de equipamentos e comandar actuadores. Este protocolo é uma das soluções mais baratas actualmente em automação industrial pela sua facilidade de adaptação a diversos meios físicos e por ser livre de taxas de licenciamento.

Quando existem vários equipamentos num determinado barramento e não se pode transmitir dados em simultâneo é necessário definir um conjunto de regras para determinar qual equipamento e quando ele pode transmitir dados. Para este fim, o protocolo *Modbus* pode ser utilizado com protocolos como o *RS-232*, *RS-485*, *Ethernet* e outros, para controlar o meio de comunicação e endereçar todos os equipamentos e suas entradas/saídas. O *Modbus* define várias mensagens para activar estas saídas, sendo elas enviadas *byte a byte*, por exemplo pela comunicação série (*RS-485*).

O modelo de comunicação do protocolo *Modbus* é do tipo mestre-escravo, isto é, um equipamento escravo não age enquanto não for solicitado pelo equipamento mestre. Por outras palavras, a estação mestre (*PLC* – geralmente) envia mensagens *Modbus* a pedir dados lidos na instrumentação pelos escravos, ou envia sinais a serem escritos nas saídas de actuadores.

O equipamento mestre é apenas um, e pode enviar uma mensagem para apenas um equipamento (*unicast*) ou para todos os equipamentos ligados no sistema (*broadcast*).

As mensagens *Modbus* são compostas por uma sequência de quatro campos: endereço do equipamento remoto a que se destina a mensagem, função (ler entrada digital, ler saída digital, ler posição de memória – por exemplo), dados e controlo de erros de transmissão (bytes *CRC* ou *LRC*). Este controlo de erros de transmissão consiste no envio desses dois *bytes* adicionais no final de cada mensagem com um valor de controlo que é calculado em função dos outros *bytes* enviados. Quando um equipamento envia uma mensagem são calculados estes dois valores, e quando o equipamento de destino recebe a mensagem também calcula esses valores em função da mensagem recebida. Se os valores calculados no envio e na recepção forem iguais, então a mensagem foi transmitida correctamente. [5] A **Figura 2.1** ilustra a composição típica de uma mensagem *Modbus* com comunicação série.

<i>1 byte</i>	<i>1 byte</i>	<i>Comprimento variável: 0 a 252</i>	<i>2 bytes</i>
Endereço	Função	Dados	CRC/LRC

Figura 2.1 - Composição de uma mensagem tipo do protocolo *Modbus*. [5]

2.3.5 RS-485

O protocolo *RS-485* permite interligar vários equipamentos, com boa imunidade a ruído electromagnético e com uma maior distância de comunicação ao contrário do protocolo *RS-232* que possibilitava a comunicação entre dois equipamentos entre si.

No *RS-485* continua a ser possível enviar palavras série entre equipamentos. As palavras série consistem numa sequência de *bits*, isto é, um *start bit*, vários *bits* de dados (5, 7 e 8), 1 *bit* de paridade e *stop bits* (1, 1.5 ou 2).

Para realizar a comunicação por *RS-485* utilizam-se 2 ou 4 condutores de cobre. A grande diferença entre o protocolo *RS-232* e o *RS-485* é o facto deste último utilizar um par de fios condutores para enviar dados e outro par de fios para receber dados, podendo ainda haver um 5º condutor a ligar a terra dos dois equipamentos. Os dados enviados são bits com valor lógico “0” ou “1” que são identificados pela diferença de potencial entre esses condutores. Valores de tensão muito baixos (<200mv) serão ignorados pelo receptor.

O tipo de cabo e a velocidade de transmissão, são responsáveis pela atenuação do sinal eléctrico, o que influencia a distância máxima entre equipamentos. O equipamento receptor pode ter dificuldade em identificar um valor lógico (“0” ou “1”) quando as frequências de funcionamento forem elevadas.

No caso de ser usado um barramento com um par de fios condutores, é possível que vários equipamentos estejam ligados ao mesmo par de fios, mas não podem ser aplicadas tensões em simultâneo, ou seja, quando um dos equipamentos aplicar tensões, os outros têm que actuar como se não estivessem ligados electricamente a esse par de condutores (estado de alta impedância). Este requisito físico permite apenas um

diálogo do tipo *half-duplex*, pois enquanto é aplicada uma tensão num par de fios condutor, os outros têm de ficar a escuta. Se for utilizado um barramento com dois pares de condutores, torna-se possível que um equipamento possa enviar e receber dados ao mesmo tempo, tratando-se de um diálogo do tipo *full-duplex*. Os equipamentos que possibilitam esta capacidade de diálogo, são designados de *Masters*, pois podem enviar e receber dados de todos os outros equipamentos que são designados por *Slaves*. Os *Slaves* não podem aplicar tensões em simultâneo para evitar a corrupção dos sinais que se pretende que enviem. [6]

2.3 Trabalhos na área de integração de sistemas

Nesta secção podemos contemplar alguns resumos de outros estudos feitos com o intuito de se perceber como abordar o tema da presente dissertação e se conhecer algumas das possíveis soluções que poderemos seleccionar e adaptar.

2.3.1 Serviços Web: Supervisão e controlo de sistemas de produção

Este tema foi desenvolvido com o objectivo de aperfeiçoar um sistema industrial para se conseguir um controlo do número de peças produzidas num determinado período de tempo, por determinado recurso e ainda verificar o estado dos outros recursos industriais identificando as falhas e/ou problemas na produção.

A necessidade crescente deste controlo ser o mais eficaz possível, levou a que este pudesse ser centralizado em um só lugar e que não fosse dependente do lugar físico onde o problema ocorre, recorrendo-se assim ao controlo remoto dos recursos.

A solução proposta neste tema foi a de aliar os benefícios das aplicações SCADA e dos serviços *Web* para permitir o controlo e monitorização de recursos industriais, independentemente da sua localização, interface de comunicação, plataforma informática e linguagem de programação. [7] Na **Figura 2.2** está representado a arquitectura de funcionamento proposta neste tema.

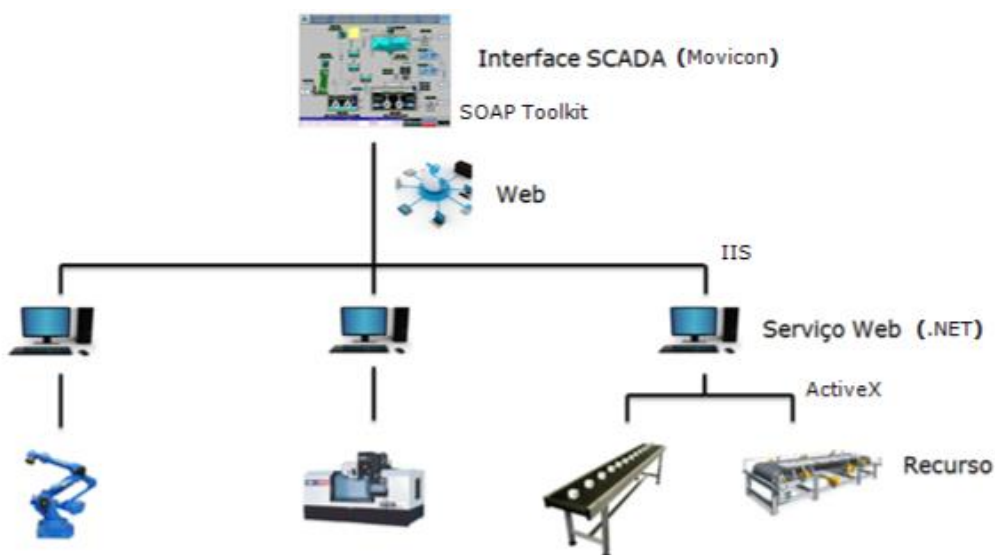


Figura 2.2 - Arquitectura esquemática da solução proposta. [7]

Para implementação desta solução usou-se um tapete de transporte automatizado que simboliza um recurso num panorama fabril. Para controlar este recurso foi utilizado um programa em *Ladder* e efectuou-se a comunicação entre o *PLC* do recurso e o computador onde está implementado o serviço *Web*. Desenvolveu-se ainda uma aplicação cliente no *Movicon* (Interface *SCADA*) integrando o serviço *Web* no mesmo. Na **Figura 2.3** pode-se observar o recurso a controlar e monitorizar. Na **Figura 2.4** observa-se a interface *SCADA* desenvolvida neste trabalho. [7]

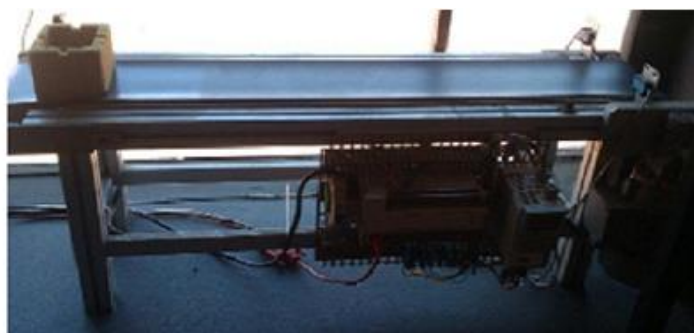


Figura 2.3 - Recurso a controlar. [7]

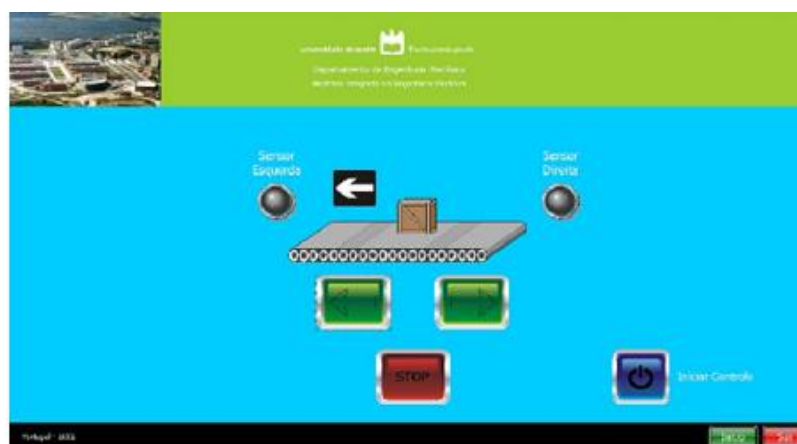


Figura 2.4– Interface de controlo da aplicação SCADA. [7]

2.3.2 Integração de sistemas de produção

A crescente necessidade das empresas se adaptarem à evolução do mercado fez com que a automação e a produção integrada por computador se tornassem importantes para o desenvolvimento das empresas. O aparecimento das ferramentas *Web* veio servir de apoio a esta necessidade, facilitando a implementação das mesmas.

Este trabalho pretendia propor uma arquitectura orientada a serviços (*Service Oriented Architecture - SOA*), como infra-estrutura de suporte à integração de sistemas de produção distribuídos e descentralizados. Recorreu-se à arquitectura orientada a serviços como solução genérica para resolver problemas de integração nos sistemas de produção. Neste âmbito propôs-se ainda, desenvolver um protótipo que permitisse o

controlo e monitorização de um recurso industrial, com acesso remoto em qualquer hora e lugar, que pudesse ser implementado em qualquer plataforma informática e em qualquer linguagem de programação que suporte as mais recentes especificações *Web*. Apresentou-se como solução uma infra-estrutura informática com recurso a *Web Services* para maquinagem e simulação gráfica *3D* remota para um centro de maquinagem *CNC*. [4]

O desenvolvimento de um protótipo para o controlo integrado do sistema flexível de produção (*FMS*) do *DEM* da Universidade de Aveiro também foi abordado neste tema.

Para implementação da solução dividiu-se o problema em 3 casos de estudo simbólicos de ambientes industriais. No caso de estudo 1 foi abordado um recurso de transporte com controlo por *PLC*; no caso de estudo 2 foi abordado um recurso de produção (centro de maquinagem) com controlador *CNC*; no caso de estudo 3 foi abordado o controlo integrado da produção (*CIM*) de um sistema flexível de produção (*FMS*), com vários recursos de transporte e de produção, diversos tipos de controlo e diferentes protocolos de comunicação. A **Figura 2.5** e **Figura 2.6** representam o esquema da solução e implementação, respectivamente, para o caso de estudo 1. [4]

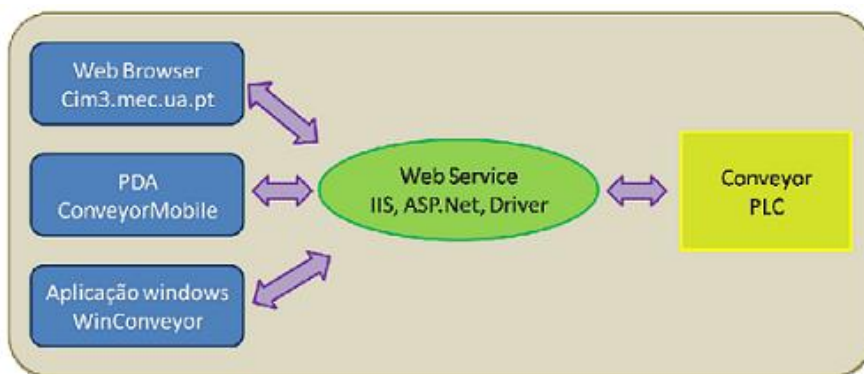


Figura 2.5 - Solução proposta para o caso de estudo 1. [4]

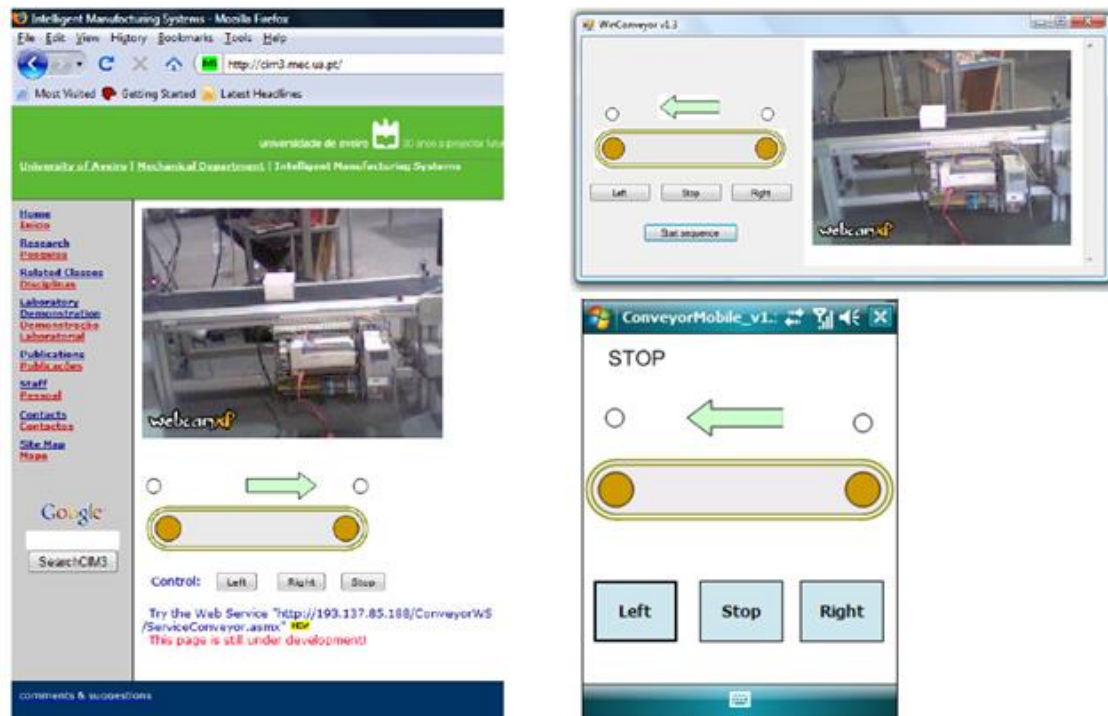


Figura 2.6 - Implementação da solução em página Web, Windows e PDA. [4]

Na Figura 2.7 temos a arquitectura proposta para o problema no caso de estudo 2, e na Figura 2.8 apresenta-se a interface gráfica de monitorização e controlo do CNC. [4]

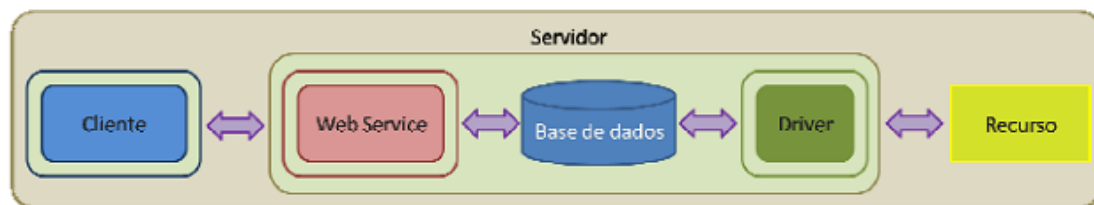


Figura 2.7 - Arquitectura da solução proposta para o caso de estudo 2. [4]



Figura 2.8 - Aplicação implementada no caso de estudo 2. [4]

A **Figura 2.9** representa a arquitectura utilizada para solução do problema do caso de estudo 3 e a **Figura 2.10** apresenta a aplicação de controlo integrado de produção. Nesta aplicação pretendia-se que os diversos trabalhos a executar por determinado recurso fossem apresentados como um cronograma de tarefas, isto é, a tarefa seguinte só ocorria quando a tarefa anterior era concluída. [4]

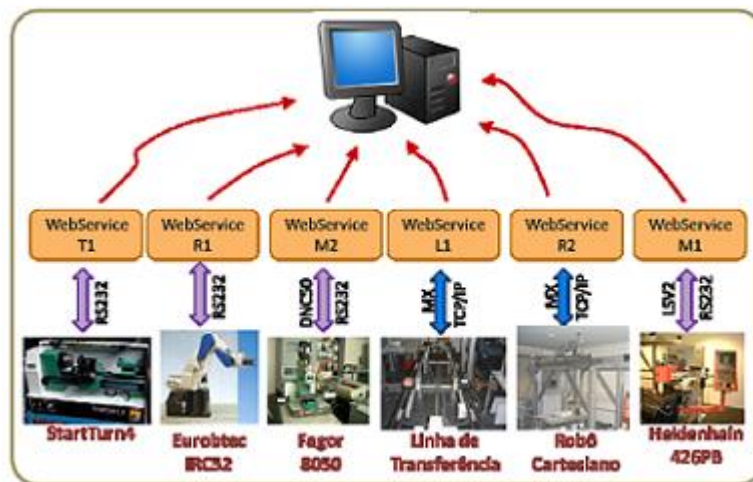


Figura 2.9 - Esquema da solução proposta para o caso de estudo 3. [4]

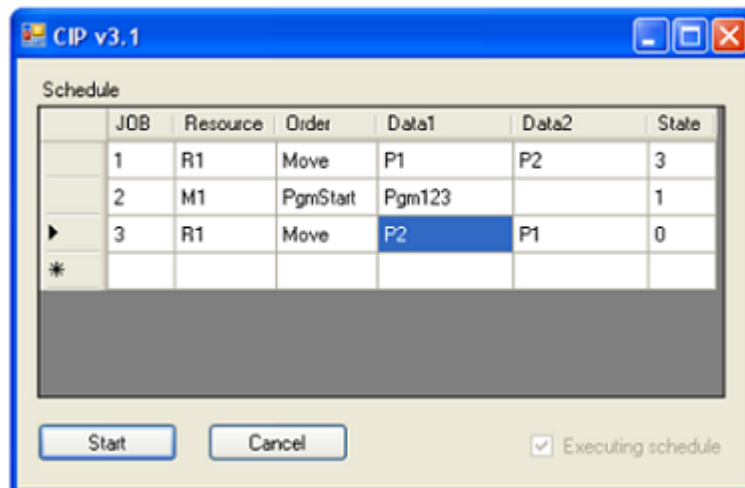


Figura 2.10 - Aplicação CIP para implementação da solução do caso de estudo 3. [4]

2.3.3 Sistema de gestão de produção para ambientes industriais

No contexto económico actual de mercado cada vez mais competitivo há necessidade das empresas alterarem os métodos tradicionais de organização. A globalização dos mercados devido ao aparecimento de novas tecnologias de informação e comunicação leva as empresas à sua incorporação, pois estas desempenham um papel fundamental.

Neste âmbito é descrito o processo de desenvolvimento de um sistema de gestão e controlo de processos produtivos em ambientes industriais. Foi desenvolvido e configurado um sistema protótipo para ser adaptado a actividades associadas à indústria dos moldes. [8]

O protótipo de gestão de produção e controlo de processos produtivos desenvolvido neste trabalho englobou a modulação, concepção, implementação e teste de um módulo de “Gestão Documental” e de “Aprovisionamento de Matérias-Primas e Componentes”.

Para a implementação do sistema usou-se o recurso *ASP.NET 2.0* que suporta *Common Language Runtime (CLR)*, que permite a escolha da linguagem de preferência no desenvolvimento da aplicação. Utilizou-se a tecnologia *Web* pelas facilidades que esta oferece aos utilizadores, desde o acesso remoto à informação do sistema, até ao facto de se poder utilizar apenas um *browser Web* para aceder ao sistema. A camada de apresentação desenvolveu-se em *HTML*, *JavaScript* e *Ajax*, enquanto a camada de negócio foi desenvolvida em *C#*. Estas linguagens foram escolhidas pois a sua conjugação permite construir uma boa solução de *software*.

Pretende-se que o sistema seja acessível a partir de qualquer computador na *intranet* dos departamentos responsáveis pelo planeamento dos moldes. Usou-se o *SQL Server 2005* para a base de dados das informações relevantes.

Na **Figura 2.11** podemos ver o diagrama da arquitectura da *intranet* onde estão ligados os pontos de acesso pelos funcionários da empresa e o servidor Web.[8]

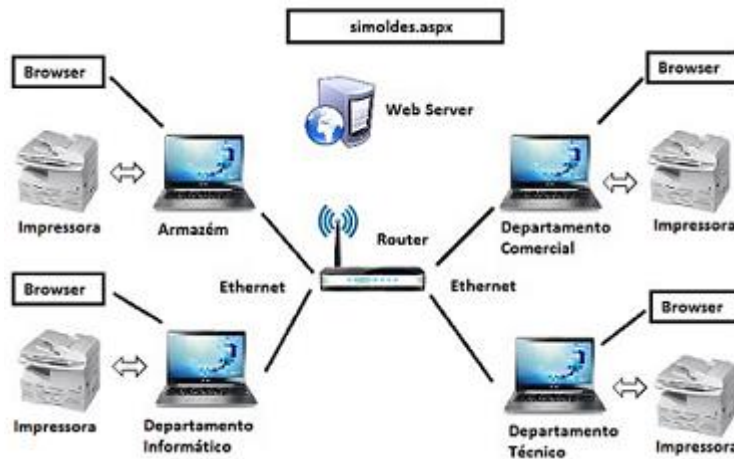


Figura 2.11 - Diagrama de implementação. [8]

Na Figura 2.12 apresenta-se a *layout* da página Web inicial para se ter uma ideia do que foi implementado neste trabalho. A partir desta página podemos aceder aos diversos menus que constituem as várias tarefas a executar neste ambiente *online*, tais como, *upload* de ficha técnica, consulta de documentos na base de dados, registo de elementos recebidos, formulários para preenchimento, pedidos de material, aprovisionamentos e envio de correio electrónico. [8]



Figura 2.12 - Interface do menu de documentação. [8]

2.3.4 “SISAL – Um sistema supervisorio para elevação artificial de petróleo”

A necessidade de gerir a elevação artificial de petróleo torna essencial a obtenção de informações sobre o processo. Para centralizar estas informações considerou-se que cada método de elevação tem as suas próprias variáveis a monitorizar. Neste sentido, várias empresas de automação têm desenvolvido controladores específicos com protocolos de comunicação diferenciados.

A solução proposta referia-se a um sistema único de supervisão e controlo (SCADA) que guarda todos os dados e consegue manipulá-los independentemente do método de elevação e dos controladores utilizados. O *hardware* de controlo do processo é composto por sensores, actuadores e controladores, enquanto o *software* de supervisão acede aos vários dispositivos para obter dados do processo e tratá-los de modo que se tornem informações úteis. A comunicação *hardware-software* fez-se através de uma rede de comunicação que geralmente utiliza uma arquitectura mestre/escravo, enquanto as redes locais utilizam a arquitectura cliente/servidor. Os protocolos de comunicação *Modbus* e *Profibus* são utilizados para tentar normalizar a comunicação na rede. No caso em questão, tem-se dois poços automatizados com dois processos diferentes de automatização: *Gas-lift* e Bombeio Mecânico. São necessários dois *softwares* para supervisão dos poços. [9]

O sistema de monitorização utilizado tem o nome de *SISAL* (desenvolvido em parceria com a Petrobrás e *UFRN* – Universidade Federal do Rio Grande do Norte). A arquitectura do sistema pode ser vista na **Figura 2.13**.

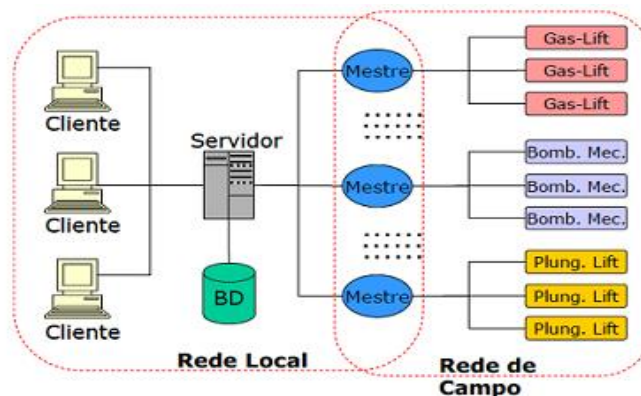


Figura 2.13 - Arquitectura do SISAL. [9]

A arquitectura do *SISAL* permite monitorizar poços com diferentes métodos de elevação utilizando um único *software* de supervisão. Pretende-se que os utilizadores do *SISAL* interajam com a aplicação cliente, que por sua vez monitoriza os poços e indica os detalhes e o ponto de situação de cada poço através do histórico, listas, bases de dados e alarmes. A **Figura 2.14** representa de forma simbólica a interacção dos utilizadores com o processo. [9]

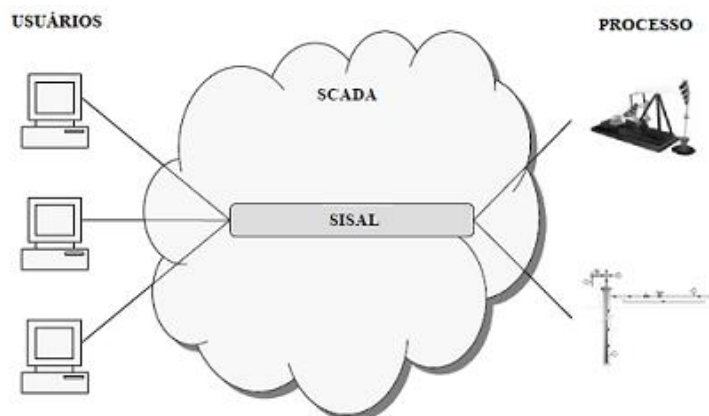


Figura 2.14 - Ambiente de automação com o *SISAL*. [9]

2.3.5 “Sistema Supervisório de Aquisição de Dados em um Protótipo de Estação de Tratamento de Efluentes”

No âmbito deste artigo, tem-se como problema principal, a necessidade de substituir os métodos de controlo manuais pelos automáticos, de forma a obter um maior rendimento na produção e aumentar o seu grau de credibilidade através do registo de dados de todo processo. Muitas vezes se dá o problema dos equipamentos, cujos dados se quer obter, estarem em ambientes industriais complexos ou até muito dispersos geograficamente. Por este motivo se implementaram vias de acesso remoto a estes dados. Actualmente usa-se os sistemas *SCADA* para este tipo de problemática, pois permitem a recolha de dados e a manipulação dos mesmos pelo utilizador do *software*.

Foi proposto o desenvolvimento de um protótipo de supervisão para uma estação de efluentes, utilizando o *hardware* indicado (na **Figura 2.16**) para verificar esta necessidade.

Usa-se o *software Laquis 3.7*, que é uma ferramenta de linguagem de controlo e aquisição de dados, monitorização de processos e geração de relatórios. Também se usam actuadores, sensores e um *PLC* para a comunicação em rede através do padrão *RS-485*. O computador comunica através de uma porta serie padrão *RS-232*, logo utiliza-se um conversor de padrões *RS-232* para *RS-485*. [10]

A programação do *PLC* é feita em *Ladder*. A **Figura 2.15** contempla o sistema de monitorização no *software Laquis 3.7*.

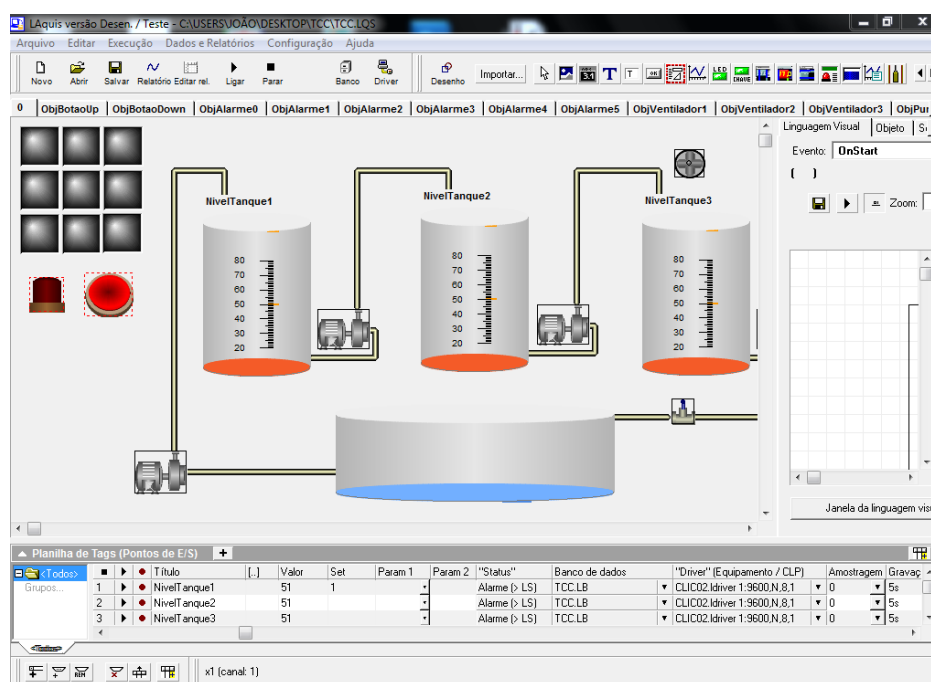


Figura 2.15 - Interface de supervisão e controlo. [10]

Na fase de implementação desenvolveu-se um sistema protótipo de simulação das etapas do processo de uma *ETAR*. Utiliza-se dois tanques de decantação, um de aração e um reservatório para representar o local de despejo do efluente para o meio ambiente. Cada tanque tem um sensor para controlo do nível de líquido. Quatro bombas hidráulicas são responsáveis pela drenagem do fluido entre tanques. No final

do tratamento uma válvula liberta o líquido. [10] A **Figura 2.16** representa o sistema protótipo de simulação.



Figura 2.16 - Protótipo da ETAR a monitorizar. [10]

2.3.6 Integração da WEB 2.0 no Desenvolvimento de Aplicações em Ambiente Industrial

No actual mundo globalizado, as empresas com processos de automação precisam de se tornar dinâmicas e flexíveis através do uso de modelos estratégicos. A produção integrada por computador (*CIM*) tem que ser utilizada pelas empresas de modo a torná-las mais competitivas. Além de ser necessário a informatização, tem que se integrar os computadores através do uso de novas tecnologias que sirvam o propósito pretendido.

Neste contexto, em um ambiente industrial torna-se imperativo criar uma estação global que permita a um engenheiro visualizar todos os dados e históricos de cada equipamento. Para satisfazer estas necessidades, apresenta-se um sistema que recorre à tecnologia *WEB 2.0* criando uma ferramenta (interface gráfica) que permite gerir e transportar dados de produção a todos os sectores da empresa de forma rápida e eficiente. Esta tecnologia proporciona a interacção do utilizador através da união de várias tecnologias como *Web Service*, *XML*, *Javascript*, entre outros. A ferramenta

apresentada, proporciona a aquisição e monitorização das diversas interfaces de comunicação existentes num processo de automação. A infra-estrutura de comunicação entre máquinas, aplicações clientes e servidoras recorre ao *framework Java Web Start*. [11] Na **Figura 2.17** podemos ter uma ideia da arquitectura do sistema implementado.



Figura 2.17 - Arquitectura do sistema. [11]

O servidor de dados é responsável por adquirir e armazenar dados do processo. O sistema *GERINF* comunica com os servidores de dados através de protocolos, como o *OPC (Ole for Process Control)*.

As aplicações cliente são manipuladas pelos utilizadores para interagir com o sistema. O servidor *GERINF* contém as informações relativas ao ambiente em estudo, que são recuperadas através do *login* de determinado utilizador. Para o acesso ao sistema utiliza-se a tecnologia *Java Web Start* e para isso o utilizador precisa de um *browser* para o acesso a uma página *Web* armazenada no servidor *GERINF*. Nessa página existe um *link* que quando é utilizado faz o *download* da aplicação para a máquina do utilizador para esta ser executada localmente. O *software* instalado, permite a cada utilizador, em cada posto, criar o seu próprio ambiente de supervisão utilizando as ferramentas disponibilizadas e gerir os dados externos como entender. [11] Na **Figura 2.18** podemos ter uma ideia da aplicação cliente.

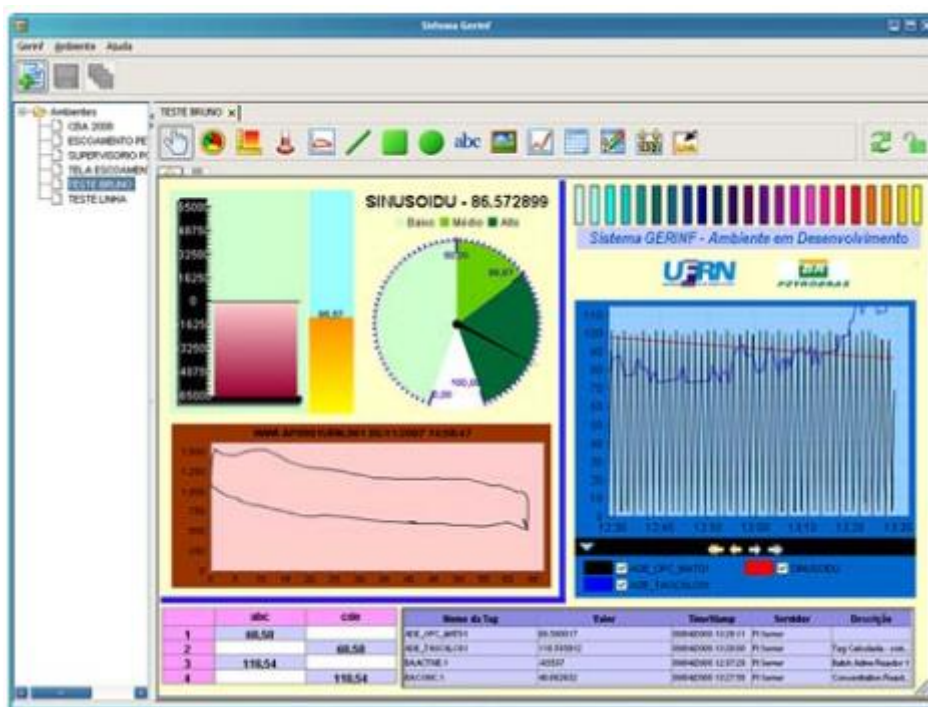


Figura 2.18 - Interface HMI desenvolvida no sistema GERINF. [11]

2.3.7 Artigo "Utilities Control"

A Corporação Geral de Água e Electricidade do Qatar é responsável pela operação e manutenção do sistema de distribuição de água do país. A gestão da rede de abastecimento e de distribuição de água é uma actividade feita, na grande maioria, à base do papel (manualmente) com o apoio de equipamento de registo de dados e comunicação verbal por telefone para se obter o estado mais recente da rede. Este sistema manual é obviamente limitado no que diz respeito a aquisição e monitorização de dados quando comparado com os sistemas SCADA actuais, por isso foi substituído para possibilitar a expansão da rede de água.

O projecto requer unidades de terminal remoto (RTU) nas principais estações de fornecimento de água e na rede de distribuição. A rede de fornecimento de água consiste principalmente em reservatórios e instalações de dessalinização, estações de bombeamento e tratamento por osmose invertida. [12]

Propõe-se que as estações *RTU* comuniquem com o centro de controlo através de uma rede de fibra óptica. As estações na rede de distribuição de água consistem em pequenas cabines montadas no exterior, ligadas a medidores de fluxo e a transmissores de pressão que comunicam com o centro de controlo através de um modem *GPRS*.

O sistema *SCADA* desenvolvido chama-se *Televant's SCADA OASyS DNA 7.4* e tem um conjunto de aplicações importantes, nomeadamente, gestão da pressão nos reservatórios, sistema de detecção de fugas através da pressão e equilíbrio da água na rede de distribuição e monitorização da qualidade da água nas áreas de fornecimento. Isto torna-se possível através das várias informações recolhidas pelas diferentes estações remotas. A **Figura 2.19** ilustra o sistema *SCADA* desenvolvido. [12]



Figura 2.19 - Sistema *SCADA* desenvolvido. [12]

Implementou-se dois tipos diferentes de *PLC's*, uns com maior capacidade de *inputs/outputs* e memória nas estações da rede de fornecimento de água, e *PLC's* de menor capacidade para estações na rede de distribuição. Os sistemas de comunicação para as estações primárias recorrem à fibra óptica, e nas estações secundárias usa-se a comunicação por *GPRS*. A comunicação *GPRS* é feita por rede *VLAN (Virtual Lan)*, pois permite a criação de uma rede fechada por *TCP/IP* assegurando a confiabilidade da informação transmitida. [12] Na **Figura 2.20** pode-se observar o esquema da rede *TCP/IP*.

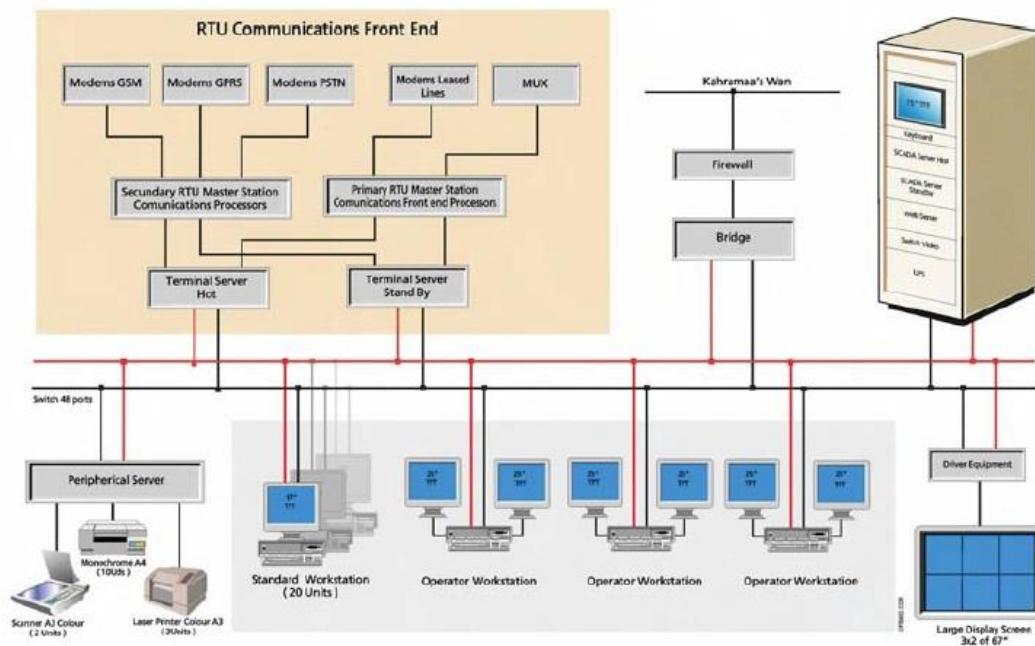


Figura 2.20 - Rede TCP/IP do sistema e equipamentos intervenientes. [12]

As RTU's das estações primárias permitem o controlo da pressão e o controlo remoto do novo centro de controlo. As RTU's das estações secundárias estão configuradas para transmitir dados relativos a pressão e fluxo em determinados intervalos de tempo. O sistema SCADA usa a informação recebida dos aparelhos de medição de qualidade de água para oferecer uma análise gráfica e estatística detalhada. [12]

2.3.8 Artigo "SUBSEA MONITORING - Monitoring in a harsh environment".

A extração de gás natural em *Ormen Lange* na Noruega necessitava de um sistema de monitorização da condição do enorme gasoduto devido às condições adversas do mar. O gás era transportado em 120km de gasoduto, através de terrenos irregulares, a profundidades de 850 metros no mar até às instalações de produção em terra através da pressão dos poços. Este terreno irregular fazia com que vários

segmentos do gasoduto não estivessem em contacto com o solo oceânico e as grandes correntes marítimas podiam induzir vibrações nestes vãos livres. Para solucionar este problema desenvolveu-se um sistema em *LabView* para monitorização destas vibrações. [13]

Como solução criou-se um sistema de monitorização para os vãos livres do gasoduto, que consiste numa rede de instrumentação que inclui vários pacotes de sensores indutivos autónomos e sincronizados (*CSP*), e um pacote de sensores mestre (*MSP*) que monitoriza as vibrações nos vãos mais longos.

Este projecto permite que vários “nós” iniciem e parem o registo com uma precisão de 2.5 *ms* a uma distância de pelo menos 100 metros, assim como a amostragem de três canais analógicos em paralelo com quatro *IO*'s em série. Todos os “nós” operam por *wireless* usando o som para transferência de dados e sinalização. Desenvolveu-se uma bateria com 6 meses de duração através de um esquema de gestão de energia inteligente. [13]

A solução consiste em quatro sistemas que possibilitam a medição multi-nós sincronizada de vibrações nos vãos longos, movimento estático e dinâmico dos vãos livres do gasoduto, medição da corrente marítima, pressão e temperatura.

Estes sistemas também permitem o armazenamento e análise matemática das medições até 6 meses, comunicação *wireless* nó-a-nó, manejo e recuperação de erros nos nós para uma operação sem falhas e interface acústica no *ROV*, (Veículos de Operação Remota) para recuperação de dados e reconfiguração de sistema. [13] Na **Figura 2.21** podemos observar a montagem do equipamento na conduta e a torre *MSP*.



Figura 2.21 - Montagem típica no oceano. Esquema de comunicação entre *CPS*'s engatados ao longo dos vãos do gasoduto e da torre *MSP*. [13]

Os *CSP's* têm como objectivo principal gravar as vibrações em todas as direcções dos 3 eixos e são montados em intervalos regulares pelo gasoduto. Foram desenhados para ser engatados no gasoduto por um *ROV*. Acoplou-se no topo um *pod* electrónico para medir as vibrações usando um sistema de ligação que pode ser facilmente removido mais tarde, tal como uma substituição de baterias. Estes *CSP's* são controlados e sincronizados por uma torre *MSP* inercial instalada no solo oceânico por uma grua e um *ROV*. Este *MSP* também regista as correntes marítimas, salinidade, temperatura e pressão.

O sistema também necessita de uma bateria de baixa potência mas confiável com um sistema de comunicação acústica incorporado. A ligação entre as unidades *CSP* e *MSP* são feitas por *wireless* através de modems acústicos. [13] Na **Figura 2.22** observa-se o método de montagem dos *CSP's* no gasoduto pelo *ROV*.

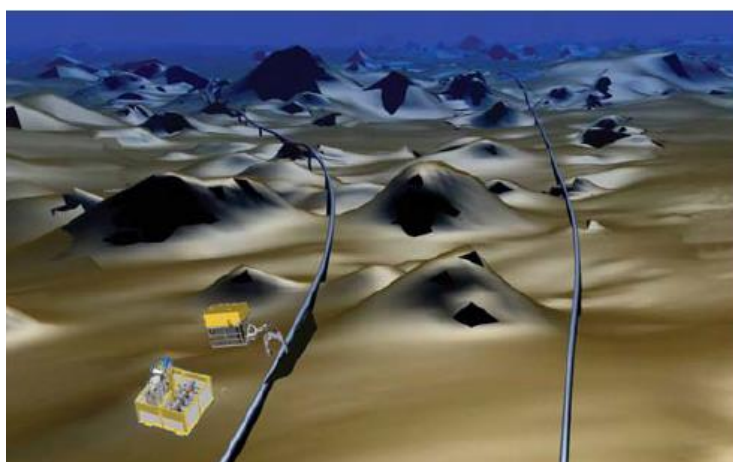


Figura 2.22 - Montagem de *CSP's* no gasoduto pelo *ROV*. [13]

2.3.9 Artigo “*Subsea Control*”

A necessidade de um maior controlo da operação de uma instalação é suportada pela disponibilidade de hardware e software moderno para o processo industrial. A monitorização torna-se cada vez mais importante para aumentar a confiabilidade e segurança no sector industrial. A detecção de falhas com antecedência pode ser conseguida através do levantamento de informação por via de vários factores

quantificáveis para utilização de modelos de processamento matemáticos. É importante a informação heurística, assim como as observações do operador ou a experiência em determinado processo. Um diagnóstico é mais completo tendo em conta ambos os sintomas matemáticos e heurísticos. Actualmente não existem maneiras de detectar a falha com antecedência na área dos sistemas de controlo submarinos. Normalmente, as falhas são detectadas quando estas já estão a afectar o sistema. [14]

Os sistemas submarinos das plataformas petrolíferas normalmente são constituídos por poços, válvulas e condutas, sistemas de condutas e elementos estruturais, entre outros. Em grande parte dos casos os poços têm que ser controlados de um único lugar. Os sistemas de controlo permitem a operação das válvulas, bobinas, colectores e condutas. Também têm que permitir o encerramento seguro em caso de falha nos equipamentos ou perdas hidráulicas ou eléctricas na plataforma petrolífera à superfície.

Neste estudo, o uso dos conhecimentos analíticos do processo ajuda a quantificar a informação que é usada para gerar valores característicos através da verificação do valor limite dos sinais que podem ser medidos directamente. Estes sinais característicos (variâncias, amplitudes, frequências ou parâmetros de modelo) são analisados por funções de correlação, espectro de frequências, etc. Em alguns casos as características físicas podem ser deduzidas dos valores característicos, que posteriormente são comparados com as características normais de um processo sem falhas. Além destes sintomas quantificáveis, há os sintomas heurísticos que são registados como manutenções feitas, reparos, falhas anteriores, tempo de vida, etc. O diagnóstico de falhas, consiste portanto em determinar o tipo, magnitude e localização da falha, assim como o tempo da sua detecção baseada em sintomas heurísticos e analíticos e desta forma encontrar estratégias para combater as falhas. Hierarquizou-se os sintomas encontrados escolhendo os mais apropriados para tornar o processo robusto e desta forma obter um sistema de diagnóstico hierárquico por grupos como se ilustra na **Figura 2.23**. [14]

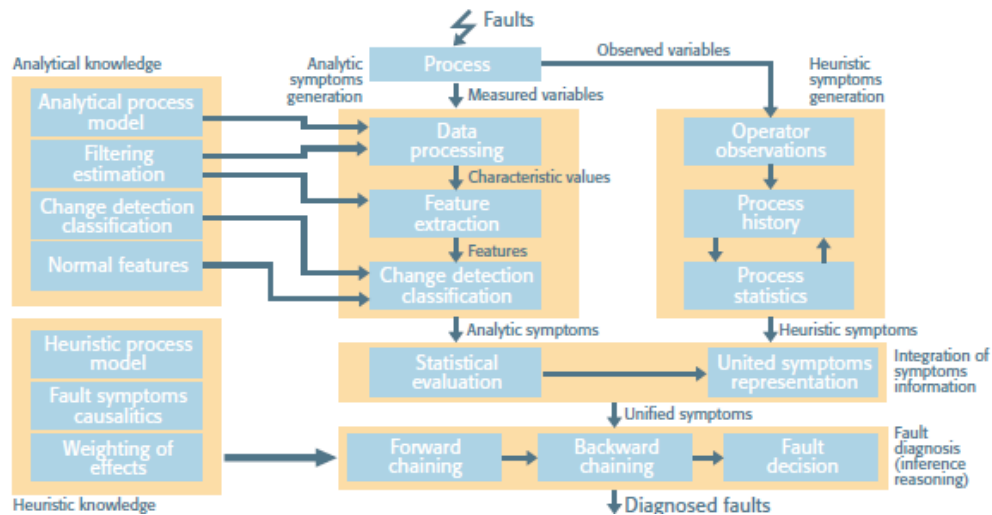


Figura 2.23 - Esquema integrado de detecção e diagnóstico de falhas.

O sistema de controlo de produção foi dividido em vários subsistemas nomeadamente, o hidráulico, o de comunicação, o eléctrico, o electrónico e o de instrumentação. [14] Na Figura 2.24 observa-se o esquema para detecção de falhas.

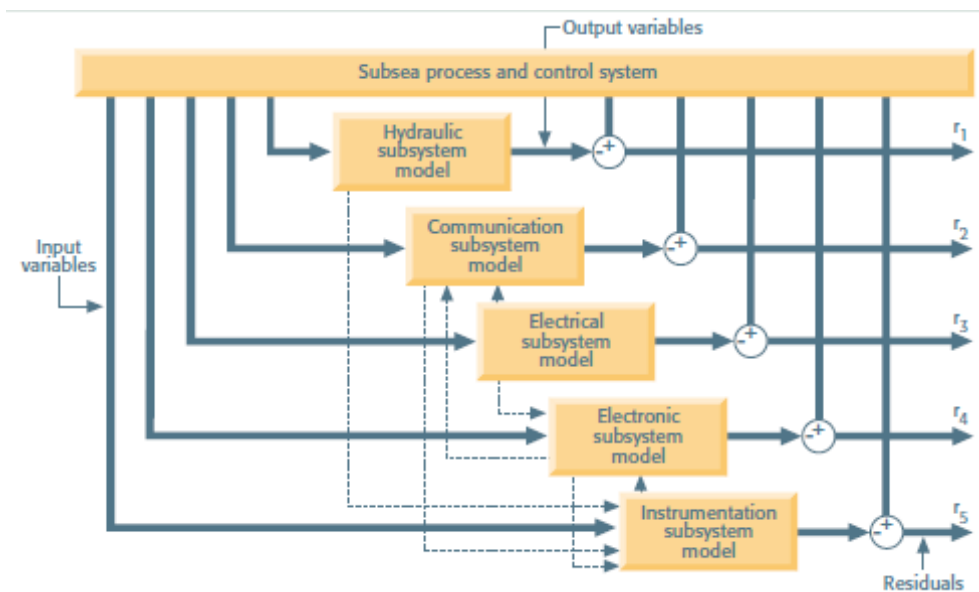


Figura 2.24 - Esquema de detecção de falhas. [14]

Os modelos de diagnósticos correspondentes às condições normais de funcionamento de cada subsistema têm de ser arquivados para se gerar os sintomas analíticos. Com base nesta filosofia, o esquema de supervisão proposto dá apoio à

tomada de decisão, causa da falha e mensagens de aviso para que se possa encontra-la e decidir a tarefa de manutenção a realizar. [14] A **Figura 2.25** ilustra o esquema de supervisão inteligente.

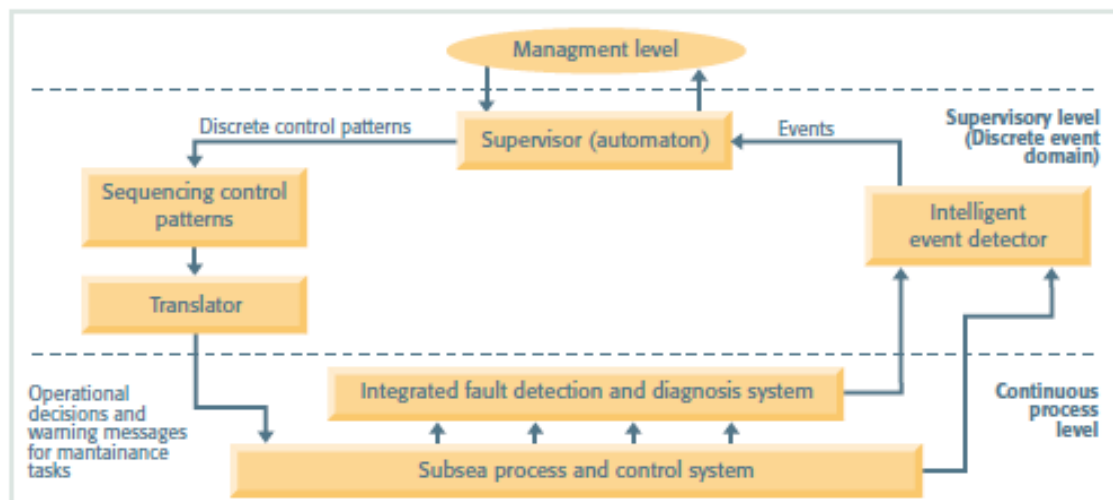


Figura 2.25 - Sistema inteligente de supervisão. [14]

2.3.10 *Open-Source Implementation of Monitoring and Controlling Services for EMS/SCADA Systems by Means of Web Services – IEC 61850 and IEC 61970 Standards*

Neste projecto pretende-se desenvolver um sistema de monitorização e controlo para sistemas de gestão de energia (*EMS/SCADA*). Estes sistemas permitem aos operadores de empresas de energia eléctrica otimizar o desempenho e/ou o sistema de energia. Propôs-se a implementação destes sistemas de forma *open-source* para reduzir custos de urbanização e facilitar o processo de aquisição de novas aplicações. A troca de informações é baseada em modelos de dados padronizados através do uso de serviços *WEB*, para se otimizar as operações em rede como se resume na **Figura 2.26.** [15]

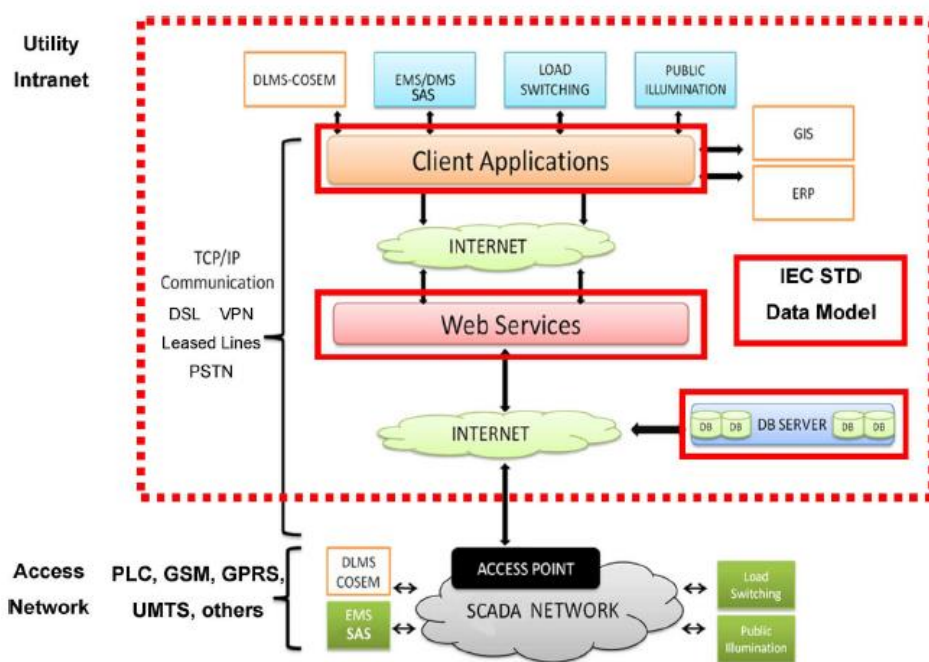


Figura 2.26 - Proposta baseada em serviços WEB e nos padrões IEC (International Electrotechnical Commission).

Utilizou-se o conceito de interface genérica (*GID*) para se obter interfaces padronizadas e utilizou-se também o modelo de informação comum (*CIM*) para tratar os modelos de dados padronizados.

As *GID* permitem criar uma estrutura consistente e fácil de usar, facilitam a integração e fornecem um modelo de programação unificado para integração de aplicativos. A combinação entre *CIM* e *GID* ajuda a interoperabilidade de componentes, reduz custos para integração de aplicações e permite a ligação dos componentes para trocar informações. Além disto são necessários os modelos de relacionamento de entidade, que são utilizados no projecto do banco de dados para representar dispositivos físicos tais como geradores, medidores e disjuntores para se testar o servidor *WEB* desenvolvido. [15]

É necessário uma interface de acesso a dados de alta velocidade (*HSDA*) para que a transferência de dados em ambientes distribuídos seja eficiente, e ainda uma interface para transferência eficiente de dados de séries temporais (*TSDA*). O modelo de dados relacionado com a *HSDA* e *TSDA* é concebido a partir da representação orientada a objectos fornecida pelo modelo de informação comum (*CIM*). O *CIM*

consiste num modelo abstracto que representa todos os principais objectos de um *EMS* (*Energy Management System*) como classes e baseia-se em linguagem de modelagem (*UML*) para definir o nome das classes, atributos e relações com outras classes. A *HSDA* é adoptada em termos de serviços para o acesso directo aos dispositivos representados no banco de dados. Pretende-se que estes serviços interajam directamente com os dispositivos simulados em termos de leitura e definição de parâmetros controláveis, o que é muito útil quando se pretende ler parâmetros de dispositivos ou agir sobre os seus modos de operação. Ao incluir os dados binários em elementos *XML* para aumentar a eficiência, torna-se mais óbvia a escolha dos serviços *WEB* para monitorização e controlo de serviços para sistemas *EMS/SCADA*.

É usada uma base de dados *open-source* na implementação. Para implementação dos serviços *WEB* usa-se os serviços *Java Apache Axis2*. Usa-se o protocolo *HTTP* como camada de transporte, e são efectuadas chamadas síncronas e assíncronas. As interfaces *standard IEC* necessitam da implementação da gestão de eventos.

A implementação de serviços *WEB* para monitorização e controlo de serviços *EMS/SCADA* é baseada em padrões de implementação específicos chamados de “Serviços *proxy* padrão” que derivam de “domínios padrão” e que garantem o desacoplamento das aplicações. Isto é feito graças a um pequeno número de módulos *proxy* interpostos entre clientes e serviços. [15] A Figura 2.27 representa um exemplo do uso de um serviço *proxy* em interacção com a interface gráfica do utilizador (*GUI*), serviços *web* e equipamento físico.

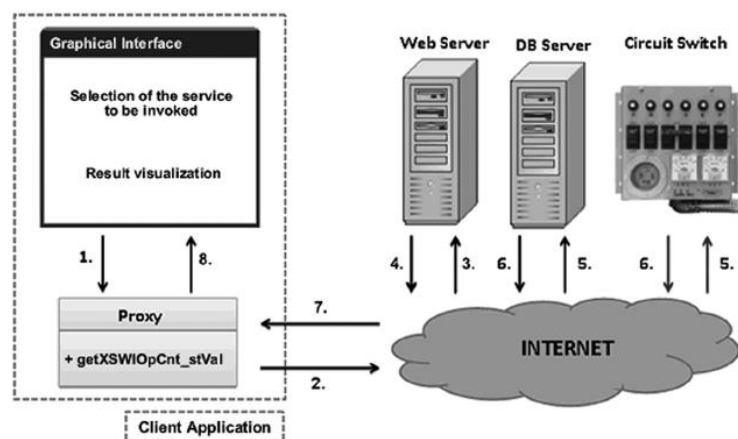


Figura 2.27 - Modo de interacção. [15]

A invocação do serviço *getXSWIOpCnt_stVal* retorna um valor do atributo relativo ao interruptor de circuito (*Circuit Switch*) e é dividido em alguns passos. A *GUI/Cliente* é responsável de fazer o pedido de um serviço específico e este pedido é gerido pelo módulo de serviço *proxy* que é local à aplicação cliente. Este serviço *proxy* é responsável pela invocação dos serviços *WEB*. O serviço de implementação localizado nos serviços *WEB*, permite a leitura da base de dados que contém a informação responsável pela interacção com os equipamentos físicos. De seguida a informação é obtida pela *GUI/Cliente* onde se pode obter os resultados para visualização ou para processamento. [15]

2.3.11 *Web services-based automation for the control and monitoring of production systems*

A automação fabril é essencial na redução de custos de produção, tornando mais fácil a reconfiguração das linhas de produção. As funcionalidades inteligentes do processo de monitorização, diagnósticos rápidos e eficientes, facilidade de uso e manutenção são mais-valias nestes processos industriais.

Neste contexto recorreu-se aos serviços *WEB* para facilitar a reconfiguração e integração dos equipamentos de controlo através de aplicações industriais e de negócios. O uso de arquitecturas orientadas a serviços (*SOA*) implementadas através de tecnologias de serviços *WEB* unifica as tecnologias a todos os níveis dentro de dada empresa. O facto de se basear em padrões comuns de comunicação como *HTTP*, *XML* e *SOAP*, torna as ligações homogéneas tanto ao nível da produção como ao nível de gestão empresarial. [16] Na **Figura 2.28** ilustra-se a solução para o problema em questão.

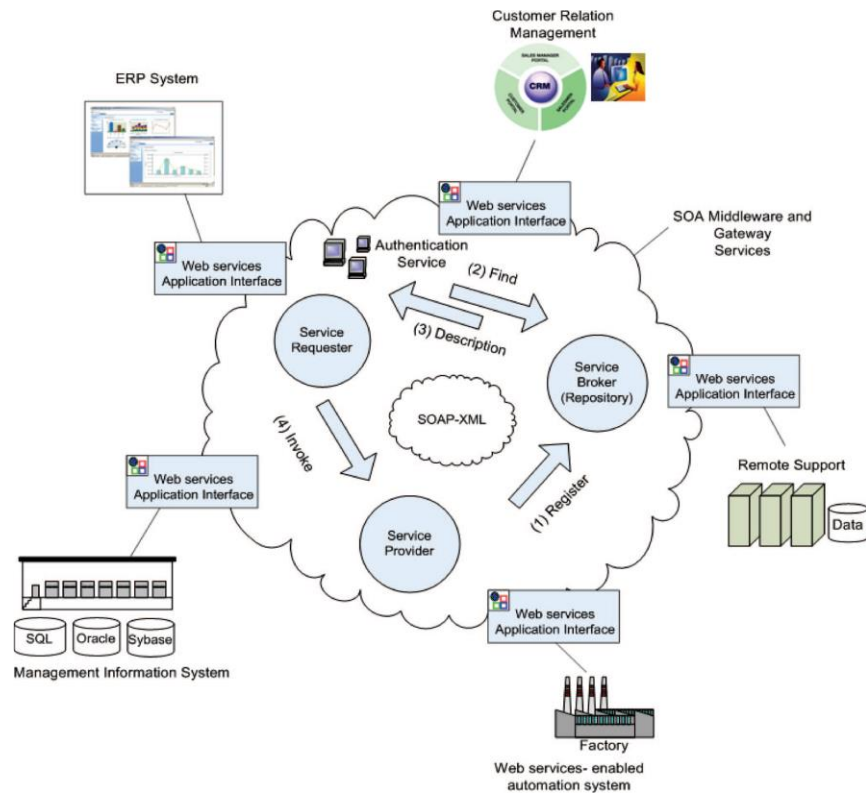


Figura 2.28 - Integração da estrutura básica dos serviços WEB – SOA [16]

Na implementação de serviços WEB para automação, o aparelho de controlo FTB (*field terminal block*) oferece capacidade de processamento, I/O local e uma conexão Ethernet padrão. A estrutura do *software* dos dispositivos de controlo ou componentes foi construído com uma camada com aplicação de serviços WEB ao topo como ilustrado Figura 2.29.

Para lidar com as comunicações TCP/IP é necessário um RTOS (responsável pela calendarização multitarefa em tempo real) e o IP stack abaixo para lidar com a tarefa DPWS (perfil de dispositivo de serviço web). Com isto conseguimos realizar tarefas industriais em tempo real. [16]

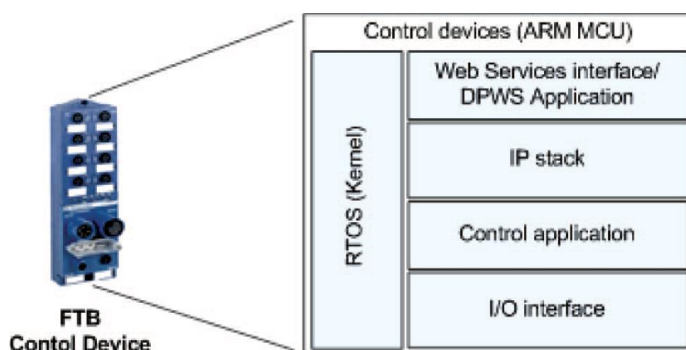


Figura 2.29 - Arquitectura do software de serviços *WEB* do componente. [16]

O motor de instrumentação de serviços que funciona no componente com a transição de estado lógico, é implementada em linguagem *C* e em *Java* para explorar a interoperabilidade da plataforma de controlo *DPWS*. O motor é utilizado para manipular a acção nos controladores embutidos (*FTBs*) enviando um comando de invocação *DPWS* por mensagem *SOAP* através da *Ethernet* para o destino usando linguagem *XML*. O sistema de automação baseado em serviços *WEB* foi implementado com *DPWS* nos dispositivos *FTB* distribuídos que são executados pelos clientes *DPWS* distribuídos na instrumentação (por *PC*) em cada estação. Para garantir a robustez no envio das mensagens *SOAP* e para garantir que todas as mensagens *DPWS* são recebidas com sucesso a nível das aplicações, é implementado um recibo de reconhecimento da operação *DPWS* além da sincronização de packet's no *TCP*. Foi também definida uma variável *ID* para a ordem incremental definida na aplicação *DPWS* para identificar e acompanhar cada mensagem recebida pelos componentes. A tarefa do temporizador *RTOS* no cliente (que envia a mensagem) é iniciada e respondida num tempo limite pelo receptor. Quando o cliente recebe esta mensagem, notifica quem a enviou com uma mensagem *SOAP* (i.e *ACK*). Se a mensagem de confirmação for perdida, é originado um *timeout* com uma mensagem de erro apropriada para cada situação.

Além da visualização e controlo do processo, o sistema *HMI*, fornece um ambiente que todos os utilizadores possam comparar dados e enviar comandos para os diversos componentes do sistema distribuído através de uma interface de *browser* simples. Na Figura 2.30 observa-se a arquitectura da solução. [16]

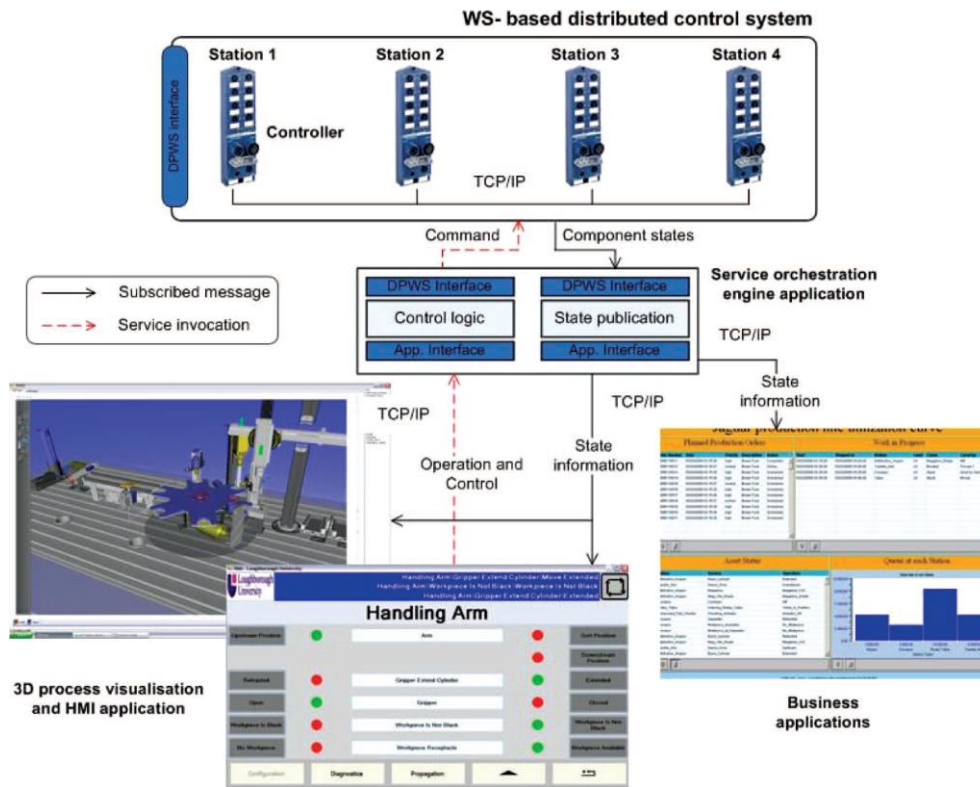


Figura 2.30 - Aplicação do mecanismo de monitorização e controlo baseada em HMI.

2.3.12 Planeamento e controlo de produção na Euronete, S.A.

O principal objectivo deste projecto é optimizar o fluxo de informação no sector produtivo para o cliente da empresa *Euronete, S.A.*, e para tal há a necessidade de adoptar um sistema de planeamento para o controlo de produção. Este aperfeiçoamento estende-se aos sectores da produção de fios e de extrusão de monofilamentos, pois é necessária a integração dos mesmos. Também é necessário desenvolver uma ferramenta de cálculo de peso de redes para melhorar a rapidez e qualidade de comunicação com o cliente.

No modelo de planeamento e controlo da produção, realiza-se a correcção dos roteiros das redes e tempos de fabrico. A elaboração de relatórios, construção de pequenas ferramentas como geração de planos de embarque, criação de *packing lists*, actualização de tabelas de preços e registo de horas extraordinárias na fábrica são

algumas das medidas tomadas para auxiliar e automatizar as actividades dos diversos sectores da fábrica. [17]

Como solução para redução de *stocks*, tempos e custos de produção criaram-se ferramentas com recurso a programação em *VBA* para se determinar as necessidades de materiais e máquinas em termos de quantidade e tempo de produção.

Uma das ferramentas desenvolvidas fornece um apoio ao planeador para tomadas de decisão de dimensionamento e escalonamento de forma a ter noção do impacto dos planos em tempo real por simulação (na interface gráfica). Além disso há a possibilidade do controlo de produção e do que se passa dentro da fábrica a cada instante, o que permite que haja revisões e alterações periódicas e regulares no momento certo.

A aplicação tem como função extrair o *ERP* de todos os dados necessários (*MPS*, inventário mestre, *BOM*, roteiros dos produtos e capacidade produtiva das máquinas) para realizar o *MRP* e *CRP*. A importação periódica do *MPS* determina todas as necessidades de fios e as suas datas. De seguida é apresentado ao utilizador um relatório dos resultados. Qualquer mudança de dados que influencie o processo gera um aviso para a revisão do plano que o utilizador está a consultar. [17]

Em cada folha o planeador tem uma ferramenta para gerar ou actualizar um horizonte de planeamento automaticamente, sendo apenas necessário especificar as horas de trabalho envolvidas no processo. Há ainda um campo reservado para o utilizador determinar a ordem de sequenciamento da produção, o que minimiza tempos de *setup* e as prioridades de produção para tornar o seu dimensionamento e escalonamento mais evidentes. Quando todas as decisões estão tomadas, é feito um cálculo dos dias de produção de cada ordem e confronta-se os resultados com os pedidos de determinado cliente. Há outra folha que permite calcular as necessidades totais de fio para cada ordem mostrando a data e a quantidade em que o fio é necessário para a produção da rede. Após este processo, quando há redes marcadas como produzidas, são gerados relatórios e gráficos semanalmente para a administração. [17]

Ao concluir o plano relativo à produção de redes, o planeador da fábrica de fio em Boticas, tem que importar a lista de necessidades apresentada anteriormente pelo

planeador da fábrica de redes na Maia. A ferramenta de controlo usada na fábrica de fios é semelhante à da fábrica de redes em termos de funcionalidades, mas difere nos grupos de máquinas a utilizar. Nesta ferramenta de planeamento dos fios existe uma folha responsável pelas necessidades de filamentos e corantes com as respectivas datas e quantidades. Após a conclusão do processo do planeamento dos fios, os *output* de dados daí resultantes servem de *input* ao processo que diz respeito a extrusão de monofilamentos. Não é feito um plano de produção como nos dois processos anteriores pois os tempos de *setup* são referentes à mudança de corante que não implica a paragem do processo. Dadas as características contínuas do processo de extrusão, a ferramenta reservada a este sector, é muito mais simples. [17]

A **Figura 2.31** ilustra um esquema do sistema *MRP II* implementado, no que diz respeito a integração do planeamento dos três sectores.

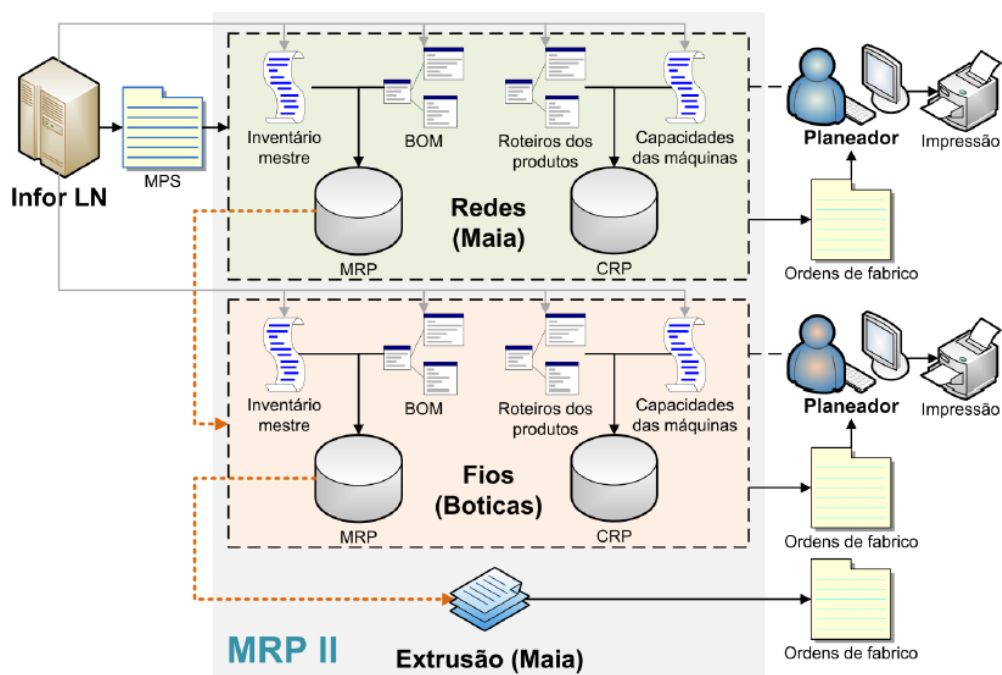


Figura 2.31 - Esquematização do sistema *MRP II* implementado. [17]

Como falado anteriormente, é necessário elaborar uma ferramenta eficaz que indique em tempo útil uma previsão do peso final da rede. É importante saber o peso

da rede antes de esta estar concluída para se poder dar um orçamento fiável ao cliente, o que não acontecia inicialmente, e o que implicou perdas no volume de vendas da empresa. Era fundamental melhorar este procedimento pois o peso estimado das redes tem por vezes um desvio de 40% em relação ao peso real, o que implicava custos acrescidos que poderia levar a perda do interesse por parte do cliente. Então realizou-se uma aplicação em *VBA* para realizar esta tarefa em tempo útil dada a considerável capacidade de processamento que esta tarefa exigia. Esta nova abordagem baseia-se no peso por malha, mas esta tem variações de rede para rede e a variação é tanto maior quanto a dimensão da amostra. A escolha do peso por malha tem de ser feito por uma média mas acrescenta-se ainda um campo de amplitude que nos dá a indicação útil dos desvios dos pesos dados pela ferramenta. Este desenvolvimento é um grande progresso mas baseava-se apenas em redes já produzidas e é necessário antecipar sobre o peso de malhas que nunca foram fabricadas. Nesta análise faz-se variar a dimensão da malha, mantendo o resto constante, e verifica-se que o peso da rede aumenta linearmente num dos componentes (malha), mas mantém o peso constante no outro componente (nó), qualquer que fosse a malha. Ao parametrizar estes dois conjuntos de dados que obedecem a uma lei linear, o indicador relevante é o coeficiente de correlação desses dados. O critério arbitrado define como aceitável $R \geq 0.8$, e apenas um número muito pequeno de agrupamentos são rejeitados, mas manteve-se os agrupamentos para posteriores consultas e comparações. [17]

Recolhida toda a informação necessária, realizou-se uma interface gráfica em modelo escada que permite ao utilizador consultar as características da rede pretendida de uma lista de todos os tamanhos de malha já produzidos em determinado agrupamento ou introduzir manualmente um novo valor para a malha e também escolher o seu tipo. [17]

Na **Figura 2.32** podemos ver o exemplo de uma consulta na interface de utilizador. O passo seguinte consiste em sistematizar os processos de importação, estruturação e filtragem de futuros dados vindos do *Infor LN* (versão do *ERP* da empresa). É essencial que o processo seja automatizado para haver a menor intervenção possível por parte do utilizador de modo a garantir a ausência de erros que influenciem os dados com o passar do tempo.

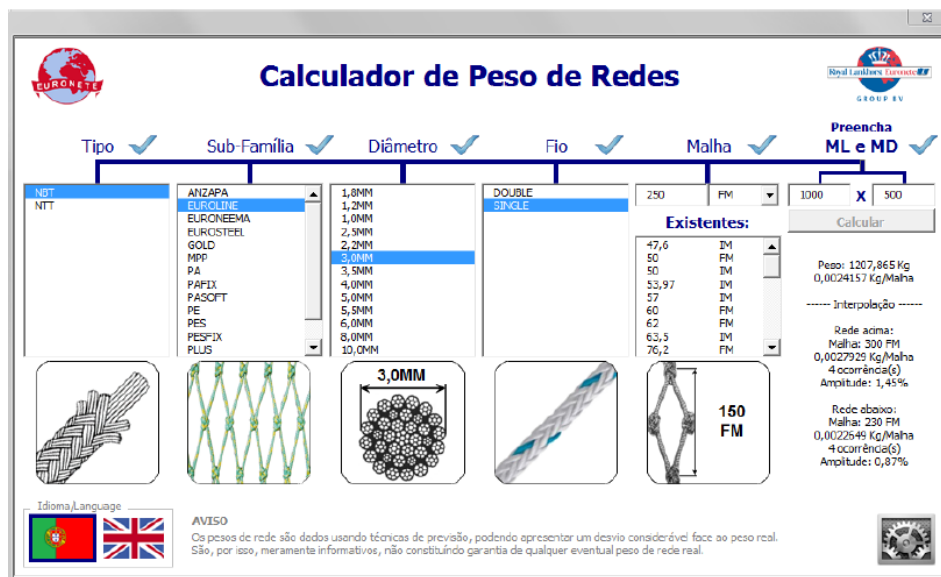


Figura 2.32 - Exemplo de uma consulta na interface gráfica da ferramenta. [17]

A Figura 2.33 pormenoriza os *outputs* da ferramenta e revela o seu encadeamento lógico. A árvore dos possíveis *outputs* apresentada mostra, para uma consulta de peso aleatória, a probabilidade de acontecer cada um dos desfechos possíveis.

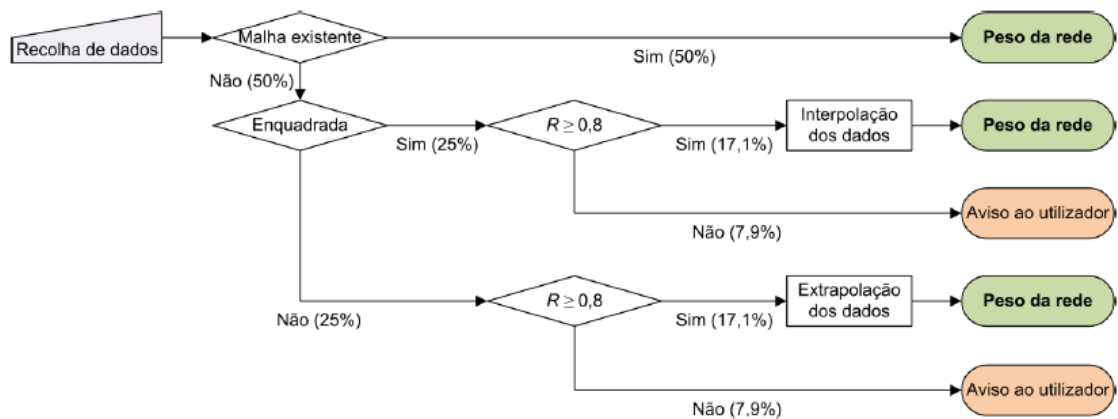


Figura 2.33 - Diagrama de decisão com árvore de probabilidade da ferramenta. [17]

Criou-se uma aplicação secundária para melhorar os planos de embarque pois estes são vitais para o armazém coordenar o envio de encomendas para o cliente, juntando o que foi produzido com os artigos em *stock*, para agrupar artigos por destino e ordenar por prioridade, para se poder criar o espaço ideal e necessário para acondicionamento, assim como para realizar a atempada reserva dos transportes. Criou-se uma aplicação em *VBA* responsável por importar os ficheiros escolhidos pelo utilizador, e que ao detectar um formato correcto faz a filtragem de dados, formatação, complemento de informação e agrupamento por cliente e data. Nesta ferramenta há ainda um módulo para a manter actualizada e com todos os dados necessários do *ERP* para complementar o plano de embarques. [17]

Foi criada outra aplicação secundária em *VBA*, para automatizar as *packing lists* que consiste numa relação de referências, descrição, quantidade e peso das redes nos fardos prontos para envio ao cliente. Serve para identificar de forma rápida e precisa o interior de cada fardo para manter o controlo da armazenagem e embarque. A aplicação permite importar dados do *ERP* e instantaneamente preencher alguns dados que não precisassem da introdução manual pelos funcionários da empresa.

Por fim, uma ferramenta para registo de horas extraordinárias foi criada de forma a informatizar um processo manuscrito e com vários passos até chegar ao *RH*. O programa precisava apenas da introdução do número de trabalhador e respectivas datas e horas para proceder ao preenchimento automático do nome do trabalhador e cálculo das horas para ser enviado semanalmente para o *RH*. [17]

2.4 Soluções *SCADA/HMI*

Os sistemas *SCADA/HMI* estão cada vez mais difundidos, existindo na actualidade diversas soluções para as empresas existentes e emergentes que querem ocupar a sua posição no mercado cada vez mais competitivo e dinâmico. Algumas das soluções mais utilizadas são dadas a conhecer nas subsecções seguintes.

2.4.1 WinCC

WinCC é um sistema HMI (Human Machine Interface), ou seja a interface entre homem e máquina. Este programa, usado na actualidade, é um sistema SCADA que pode ser usado em plataformas informáticas que usem como sistema operativo o Microsoft Windows XP, Microsoft Windows Vista, Microsoft Windows Server 2003, entre outros. Este tipo de programa serve para monitorização, controlo de processos em ambiente industrial e aquisição de dados. Permite ao utilizador configurar uma interface gráfica para observar o processo de produção de determinada empresa. [18]

O WinCC é um sistema da Siemens e foi criado principalmente para os sistemas operativos Windows e usa o Microsoft SQL Server. Vem com VBScript e aplicação de interface de programação ANSI C. [19]

É possível, por exemplo, criar um botão de abertura de válvula que através de um sistema de automação (PLC – por exemplo) vai desencadear a ação em tempo real no mecanismo físico. Podemos também criar alarmes para certos acontecimentos críticos que possam eventualmente acontecer numa linha de produção, entre outros. Na Figura 2.34 pode-se observar o tipo de interface que se pode construir com o WinCC para controlo de válvulas e motores num determinado processo industrial.

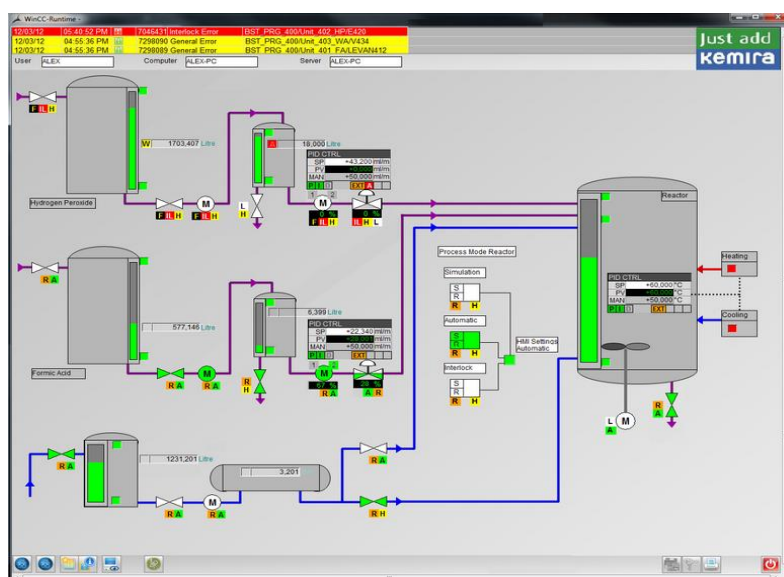


Figura 2.34 - Interface gráfica em WinCC. [20]

2.4.2 Lookout

Lookout da *National Instruments* é outra solução de sistemas *HMI/SCADA* utilizada actualmente. É um *software* de grande facilidade de manipulação e utilizado para controlo da produção e de processos de grande exigência. A sua facilidade e rapidez de utilização permitem grandes reduções nos custos gerais do projecto envolvido.

A aquisição de dados com o *Lookout* é feita em tempo real e proporciona o acesso apenas a utilizadores autorizados. Permite seleccionar qualquer computador numa rede para se ligar e utiliza a tecnologia de rede *TCP/IP*. Tem a vantagem de se poder monitorar e controlar o processo usando um navegador *Web* sem precisar de experiência de programação em *HTML*, *XML*, etc.

É utilizado pelos engenheiros numa grande variedade de aplicações industriais tais como, indústria química e petroquímica, gasodutos e oleodutos, águas e águas residuais, fábricas de papel, energia, sistemas de gestão e automação de edifícios comerciais, entre outros. Permite ainda, grande facilidade de ligação com diversos ambientes de programação, tais como, *Microsoft Visual Basic*, *Java*, *Java-enabled tools*, *Borland Delphi*, *C*, e *C++*. [21] Na **Figura 2.35** pode-se ter uma noção da aparência de uma interface criada com o *Lookout*.

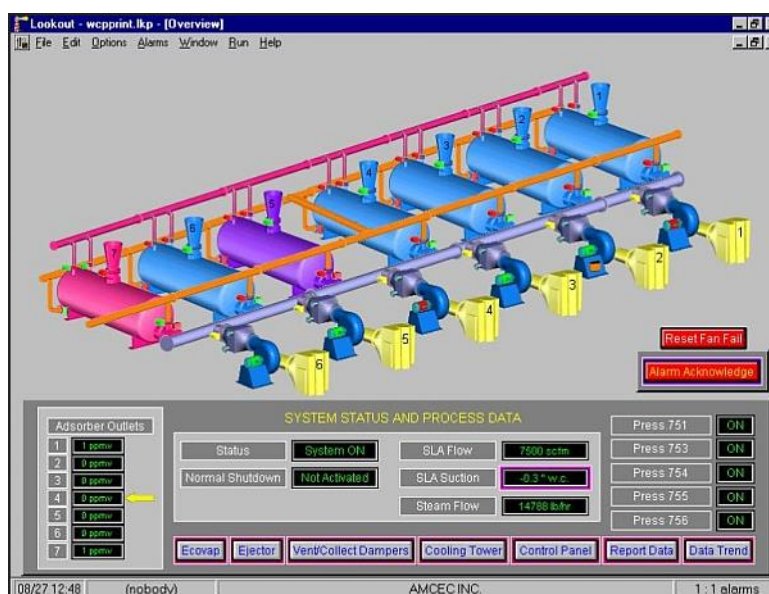


Figura 2.35 - Interface gráfica construída com o Lookout. [22]

2.4.3 *Movicon*

É uma plataforma de controlo e supervisão industrial utilizada numa vasta gama de aplicações industriais. Permite o escalonamento da produção de forma simples. É o *software* proposto no mercado como plataforma padrão para as empresas de automação, controlo remoto e automação predial. Integra tecnologia *SoftPLC* que une os benefícios do *SCADA/HMI* com os ambientes de programação *PLC* de acordo com a norma *IEC-61131*.

O *Movicon* tem uma grande flexibilidade a nível de instalação em sistemas operativos, desde *Windows CE* ao *Windows 8* com acesso a *Web* em *smarthphones* e *Tablets* (*iOS* e *Android*). [23]

Na **Figura 2.36** temos um exemplo de interface gráfica para a área da automação predial.

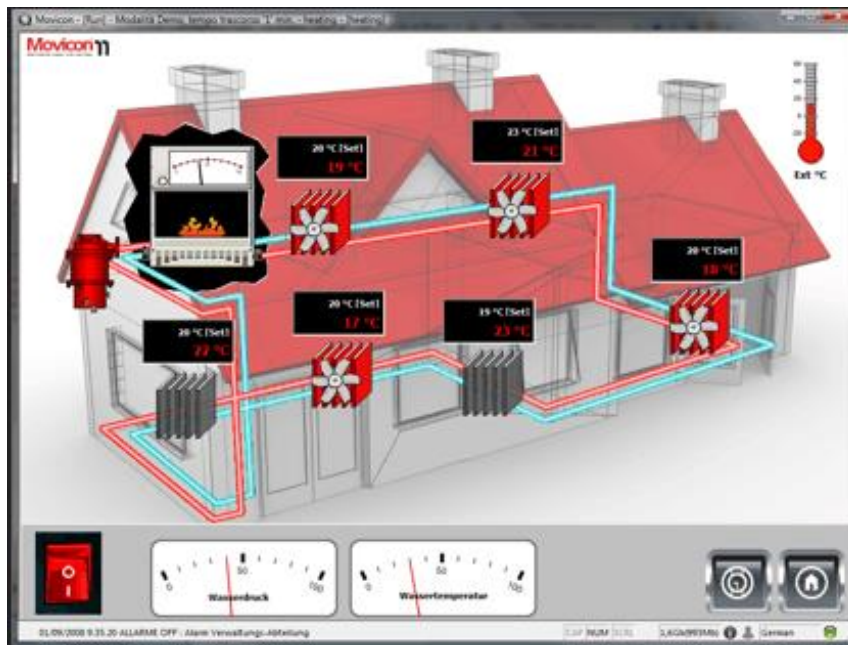


Figura 2.36 - Interface gráfica criada no *Movicon*. [24]

2.4.4 Movicon Web Client (iOS)

Com o aparecimento de novas tecnologias como os *smartphones*, hoje em dia é possível monitorizar e controlar um dado processo industrial através do telemóvel, bastando que este tenha alguma capacidade de processamento e se possa ligar a qualquer rede *Wifi* no planeta. É possível também proceder à comunicação através da rede móvel 3G ou 4G.

O *Movicon Web Client* para o *iOS*, criado pela Progea, é uma aplicação para *iphone* que permite comunicar com a aplicação servidora no *Movicon*. [25] É possível controlar em tempo real o que se passa em ambiente industrial através de um sistema do tipo cliente-servidor, como é usual actualmente, mas em vez da necessidade de um computador, é possível fazê-lo com um equipamento mais leve e que cabe no bolso.

Torna-se mais simples, prático e económico de certa forma, e eleva-se o controlo e monitorização de processos em tempo real a outro nível de eficiência, pois a qualquer momento pode-se receber um aviso no *smartphone* pela aplicação cliente e proceder a resolução de alguns problemas imediatamente sem haver a necessidade de se deslocar a outro sector onde está o computador, reduzindo o tempo de resposta a algumas eventualidades que possam ocorrer. [25]

Na **Figura 2.37** podemos ver um *print screen* do ecrã de configuração da aplicação cliente num *smartphone*.



Figura 2.37 - Painel de configuração do *Movicon Web Client* para *iOS*.

Estas são as opções de configuração da aplicação cliente. Neste caso o endereço do servidor e a porta foram escolhidos de forma a se poder comunicar com a aplicação servidora de demonstração no *site* da *Progea*, e se conseguir ter uma noção do potencial do *software*. Na **Figura 2.38** podemos ver os menus de demonstração que podemos visualizar ao comunicar com o servidor fornecido.



Figura 2.38 – Menus: inicial, gestão de energia, alarmes e controlo de processo para iOS.

Podemos observar que as funcionalidades de monitorização e controlo são semelhantes às do *Movicon 11* para computador. As funcionalidades de aviso e alarme para eventos, registo em tempo real em gráficos, botões de simulação, entre outros, estão presentes nesta aplicação, o que torna o *software* para *smartphone* uma ferramenta com grande potencial.

2.5 Análise e escolha de soluções

Nesta secção podemos encontrar as soluções escolhidas. Estas soluções foram seleccionadas pelo facto de serem facilmente normalizáveis para vários tipos de dispositivos, *softwares* de controlo integrado e ambientes de comunicação industrial.

Após análise das soluções apresentadas por outros autores, notou-se que nenhuma delas utiliza a base de dados *Oracle*. Esta base de dados é uma ferramenta poderosa para os sistemas integrados.

Ao usar a base de dados *Oracle* temos um leque maior de sistemas operativos onde podemos trabalhar, tais como, *Windows, Linux, Solaris, HP-UX, OS X, z/Os* e *AIX*. Podemos trabalhar com as linguagens de suporte mais utilizadas como *C, C#, Java, Ruby* e *Objective C*. Não é *Open-Source* mas é a única paga que trabalha com grande número de linguagens de suporte e compatível com mais sistemas operativos. Além disso é a única base de dados paga que usa interface *SQL* e interface gráfica de utilizador com todas as vantagens anteriormente referidas. [26]

A maioria dos processos industriais necessitam de uma interface com o utilizador, por isso existem diversas soluções *SCADA* actualmente. Dos *softwares* analisados, o *Movicon* é dos mais baratos que podemos encontrar no mercado. Podemos utilizar também o *Visual Studio 2012* por um custo relativamente reduzido e com um grande potencial no que diz respeito a *drivers* de comunicação, linguagens de programação, criação de aplicações para diversas plataformas. [27] Outros *SCADA* como *Lookout* [28] e *WinCC* [29] ficariam mais caros.

Além dos *softwares*, escolheram-se também mecanismos de comunicação entre equipamentos. Foi escolhido o protocolo *TCP/IP* e *Modbus* com *RS485*, pois estes são os mais utilizados actualmente, devido à sua estabilidade, rapidez e preço. Como foi referido no Capítulo 2, o protocolo *Modbus* é dos mais utilizados por ser de fácil adaptação a diversos meios industriais e mais barato por ser livre de taxas de licenciamento. O protocolo *RS-485* foi escolhido por ter especificações úteis, como a distância admissível entre equipamentos e número de equipamentos que podem estar ligados ao barramento, como foi referido anteriormente. O protocolo *TCP/IP* é dos mais fiáveis em termos de transmissão completa de pacotes de dados e rápido na sua transferência, além de ser de fácil ligação com computadores.

Em relação ao *hardware*, utilizaram-se os equipamentos disponíveis no laboratório do Departamento de Engenharia Mecânica da Universidade de Aveiro.

No Capítulo 4 aborda-se de forma detalhada os *softwares* e *hardwares* escolhidos, suas configurações e aplicações desenvolvidas.

Capítulo 3 - Proposta de uma nova solução

Na subsecção seguinte aborda-se a arquitectura da solução proposta para o trabalho desenvolvido nesta dissertação.

3.1 Arquitectura da solução proposta

Propõe-se criar uma aplicação cliente em *Visual Studio 2012*, que possa ser utilizada em qualquer computador para efectuar trocas de mensagens com alguns dos dispositivos a estudar e com a base de dados. Numa fase inicial é didáctico trabalhar com esta aplicação para visualização de mensagens *Modbus* em caixas de texto. Além disso, no *Visual Studio* podemos ter uma noção sobre a linguagem *SQL* e sobre programação que está por trás das comunicações *TCP/IP* e série, assim como perceber o funcionamento e conceito de *sockets* e *threads* do *Windows*. Após este estudo propõe-se utilizar o *Movicon* para criar uma aplicação universal que seja responsável por toda a troca de mensagens entre equipamentos, controlo do *hardware* envolvido e ainda documentar toda a informação na base de dados.

Temos disponível um *Router (Nport 5630)* para a troca de dados com o computador por comunicação *TCP/IP* (portas *Ethernet*). Este *Router* permite ligar até 16 dispositivos ligados por *RS-485*, embora no nosso caso só tenha sido utilizado um para exemplificar o processo em pequena escala. O analisador de energia (*WM22-DIN*) é ligado a uma das portas do *Nport* para se efectuar a leitura das variáveis de natureza eléctrica do compressor ligado ao sistema de ar comprimido. Esta leitura é feita por troca de mensagens em formato *Modbus* com a aplicação *VB* (inicialmente) e com a aplicação universal no *Movicon*.

Após chegar a esta solução é necessário organizar os dados lidos na instrumentação por data e hora de leitura, de forma a se ter um histórico da variação dos mesmos. Para tal propôs-se a utilização da base de dados *Oracle 11g*.

Seguidamente adapta-se a aplicação no *Movicon* para comunicar com um *PLC*. Esta aplicação é responsável pela saída analógica programável (de 0 a 10v), que faz variar a frequência de saída do variador. O valor desta frequência irá influenciar a rotação (r.p.m.) do motor monofásico do compressor. Havendo o controlo desta frequência podemos diminuir a energia fornecida ao compressor, de forma que a pressão e caudal de ar se mantenham ao nível necessário para o bom funcionamento do mesmo durante determinada operação.

Estando concluída esta planificação, reunimos todas as condições para formular um esquema da solução proposta para este estudo como ilustrado na **Figura 3.1**. ilustra-se a solução que é implementada e explicada no Capítulo 4.

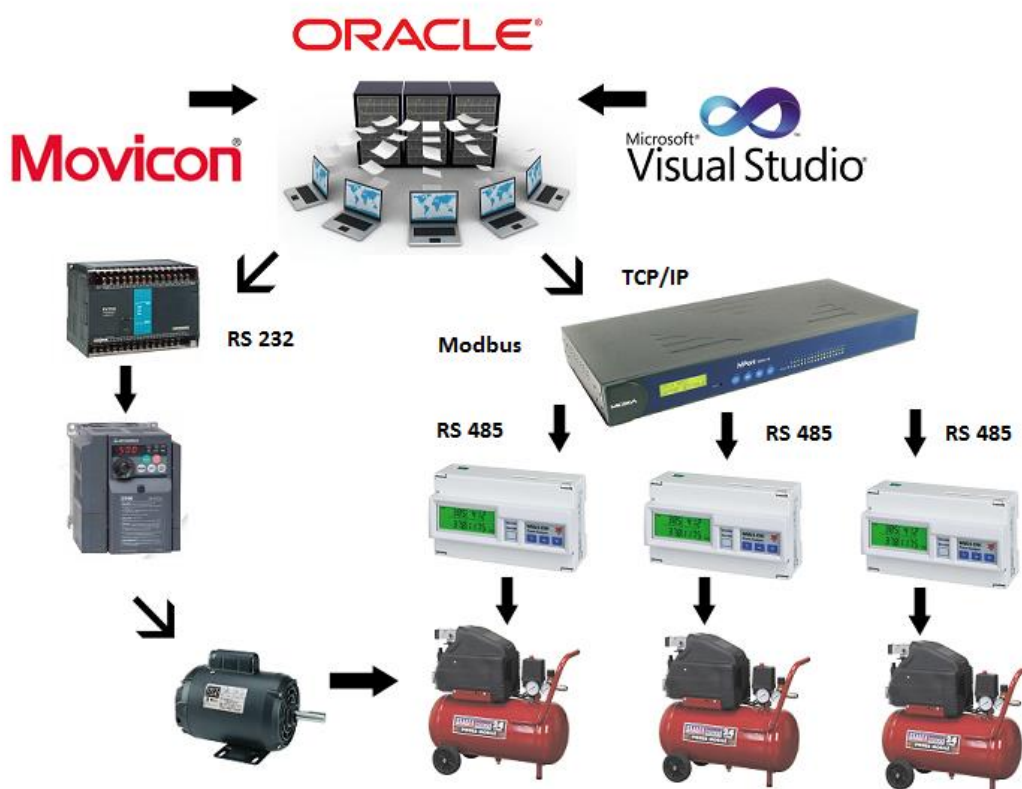


Figura 3.1 - Arquitectura global da solução proposta.

Resumindo, a solução consiste numa aplicação em *Visual Studio 2012* ou *Movicon* em ambiente *Windows*, que gerencia toda a troca de mensagens que existe com o analisador de energia, através do *Nport*. Os *slaves* (vários analisadores de energia)

estarão eventualmente ligados a motores trifásicos ou monofásicos, responsáveis pelo funcionamento dos compressores, para fazer a leitura das variáveis eléctricas inerentes aos mesmos. O *Master (Nport)* solicita por mensagem *Modbus* (aos *Slaves*) a leitura de determinada variável (kWh,kVarh,W,PF,A,etc), e estes retornam uma mensagem no mesmo formato para o *Master* com o seu valor. Por sua vez a aplicação no computador interpreta esses dados e armazena-os na base de dados *Oracle*. Outros utilizadores que pretendam aceder a essa informação poderão fazê-lo a partir de outra plataforma ou da mesma com as suas próprias credências de utilizador. Basta aceder à página *Web* do *OEM (Oracle Enterprise Manager)* ou através do seu próprio *Oracle SQL Developer*. Após a leitura e armazenamento das variáveis eléctricas do motor monofásico, recorre-se à aplicação em *Movicon* para fazer de interface gráfica entre utilizador e *PLC*. Esta aplicação também tem a função, variar a tensão de saída do autómato. Fazendo variar esta tensão de saída, também varia a frequência de saída do variador de frequência, que por sua vez fará variar a velocidade de rotação do motor. Ao possibilitar o controlo desta variação de velocidade do motor, podemos definir um *setpoint* de pressão de funcionamento do compressor de acordo com as suas necessidades de funcionamento. Este *setpoint* é definido para se ter um controlo do sistema de ar comprimido, isto é, quando se ultrapassa um determinado valor de pressão no sistema que se considere excessivo para determinada operação de consumo de ar, reduz-se as r.p.m. do motor do compressor para evitar excesso de pressão que não trás benefício. Desta forma podemos poupar no consumo energético.

No capítulo seguinte será discutido de forma mais detalhada os protocolos de comunicação, aplicações, *hardware* e *softwares* envolvidos nesta proposta.

Capítulo 4 - Implementação da solução proposta

4.1 *Hardware* envolvido na implementação

Nos subcapítulos seguintes fala-se sobre os dispositivos físicos utilizados no âmbito deste trabalho e algumas das suas especificações técnicas, modos de operação e configurações utilizadas na implementação.

4.1.1 *Moxa Nport 5630*

O aparelho *Nport* usado faz parte da serie 5600 de servidores para aparelhos que comunicam por serie (*RS-422/485*) como *PLC's*, medidores e sensores. Foi criado para tornar a ligação entre estes aparelhos industriais a uma *Ethernet LAN* baseada em *IP* de forma mais fácil e rápida. Desta forma torna-se possível para o nosso software aceder aos dispositivos através de uma rede local (*LAN*) ou por Internet. O *Nport 5630* garante a compatibilidade com *software* com padrões de rede *API* (*Winsockets* e *BSD Sockets - Berkeley Sockets*) o que permite que se use os modos de servidor *TCP*, cliente *TCP* e modo *UDP*.

O dispositivo permite-nos a configuração automática de protocolos *IP* (*DHCP - Dynamic Host Configuration Protocol*, *BOOTP - Bootstrap Protocol*) e também a configuração manual directamente no dispositivo (*Nport 5630*) ou via navegador *web*.

É possível exportar as configurações feitas para o ambiente de trabalho em formato *“.txt”* para o caso de se precisar de usar outros dispositivos *Nport*, na rede, com as mesmas configurações, importando esse ficheiro para os mesmos. [30] Na **Figura 4.1** ilustra-se a legenda dos botões e *slots* de entrada do *Nport*.

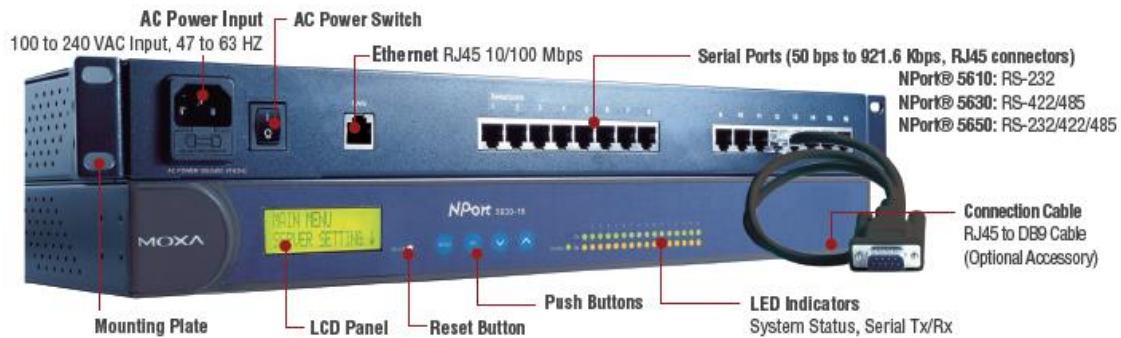


Figura 4.1 – Legenda do Nport5630. [30]

4.1.2 WM22-DIN (Three-Phase Power Analyser)

O WM22-DIN é um dispositivo analisador de energia que permite monitorizar determinados parâmetros eléctricos de uma linha ou carga com o intuito de se melhorar o consumo de energia. A medição até 100A não obriga a utilização de transformador externo. Permite também a visualização de quatro variáveis em estudo e é de fácil interface com aparelhos externos.

As vantagens principais deste dispositivo são:

- Análise harmónica da corrente e da voltagem avisando sempre que houver falhas de carga potenciais.
- Controlo de assimetria de fase, avisando as falhas na linha.
- Saída de pulso duplo, saída analógica, RS485 ou porta Dupline (Sistema remoto I/O preparados para PLC's e PC's com software SCADA).
- Comunicação em série e avisos de estado para conexões mal feitas tornam o sistema mais eficiente e menos passível de transmitir erros.
- Bateria interna que em caso de falha na linha, continua a medir todas as variáveis. [31]

Na Figura 4.2 podemos ver o aspecto do dispositivo usado no presente trabalho.



Figura 4.2 – WM22-DIN. [32]

Ao remover a cobertura frontal do dispositivo podemos observar as diversas entradas e saídas, como podemos ver na **Figura 4.3**. As saídas são constituídas pelos módulos A e B, sendo A o módulo digital, e B o módulo analógico. A fonte de energia auxiliar está no topo do equipamento à esquerda como se pode ver na legenda.

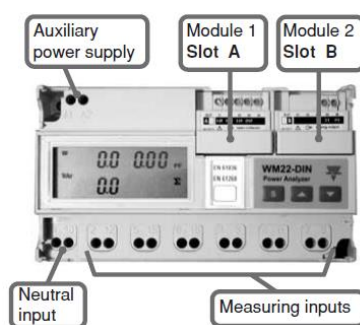


Figura 4.3 – Legenda de entradas e saídas do analisador de energia WM22-DIN. [32]

A ligação pelo modo analógico pode ser feita por 4 fios ou por 2 fios, e neste caso a ligação é feita por 2 fios para verificar a boa comunicação entre o analisador de energia e o computador utilizando um conversor *USB* para *RS485*. No modelo 5A, a alimentação auxiliar do analisador tem que ser feita, ou seja, tem que se fornecer os 220V através dos *bornes* A1 e A2, sendo um deles o neutro e o outro a fase. Segundo o diagrama eléctrico do manual deve-se ligar outro neutro ao *borne* 11 e outra fase ao *borne* 2. Os *bornes* 11 e 3 devem estar ligados ao neutro e à fase (respectivamente) do equipamento que se quer estudar. Os *bornes* 2 e 3 têm que estar em paralelo, portanto liga-se o borne 1 ao 12. [32]

4.1.3 Controlador Lógico Programável (PLC FBS-20MC)

O PLC é um dispositivo baseado num microprocessador que desempenha funções de controlo de entradas e saídas (analógicas e/ou digitais) através de *softwares* desenvolvidos pelo utilizador. Existem vários *softwares* para programação de PLC's, nomeadamente *WinProLadder*, *Mitsubishi Electric*, *Twido Suite*, entre outros. O *software* utilizado nesta implementação foi o *WinProLadder* dada a familiarização com este. [33]

O programa criado em *Ladder* é transferido para o PLC pela sua porta 0 (cabo preto). A ideia é programar a saída analógica (0 a 10V) para se ter o controlo da saída do variador de frequência (secção 4.1.4). A esta saída está ligado o *borne* verde da direita, em que o fio azul faz ligação com o Vo0+ da carta, e o outro fio faz ligação com o *GND* (*Ground*). Esta frequência de saída do variador irá gerar uma velocidade de rotação no motor monofásico (secção 4.1.5) do compressor. Na **Figura 4.4** pode-se observar o PLC utilizado no trabalho e suas ligações.

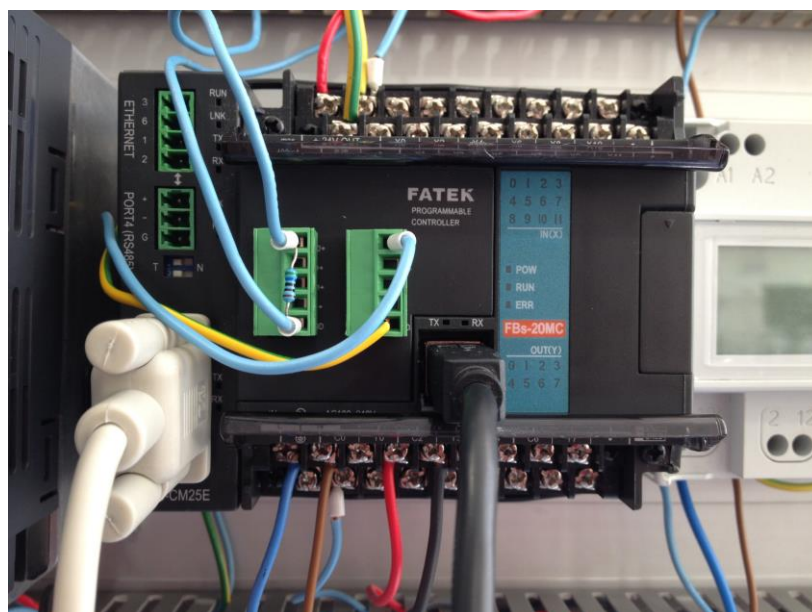


Figura 4.4 – PLC – FATEK modelo FBS-20MC.

Nas entradas do autómato (*bornes* de cima) tem-se o fio vermelho e azul que ligam aos 24V e 0V, respectivamente, para alimentar o sensor de pressão como explicado mais a frente na secção 4.1.7. Nos *bornes* de baixo, os fios castanho e azul são responsáveis pela alimentação de 220V do autómato, enquanto as saídas Y1, Y2 e C0, que estão ligadas aos *bornes* STR, STF e PC do variador que controlam o sentido de rotação do motor. Neste trabalho o sentido de rotação é feito apenas para teste das saídas e não é importante para as conclusões.

O PLC comunica por *Modbus RTU* (pela porta 3 – cabo cinzento) com a aplicação *Movicon* no computador. Esta ligação é feita por *RS232*, assim como a comunicação entre o programa em *WinProladder* com o PLC. Os parâmetros de comunicação têm de ser os mesmos no *Movicon*, no *WinProladder* e nas configurações de dispositivos do *Windows* relativamente às portas de comunicação (*COM ports*).

Na **Figura 4.4**, pode-se observar também o *borne* verde da esquerda que serve para fazer a leitura de pressão como explicado na secção 4.1.7.

4.1.4 Variador de Frequência D720-S-070SC

O Variador de Frequência ou Inversor de Corrente (AC) gera sinais de potência, com frequências e correntes essenciais à alimentação dos motores de indução. A sua funcionalidade é variar a velocidade de rotação (*r.p.m.*) desses motores.

O objectivo da utilização do variador é o de alterar a velocidade do motor ligado ao compressor. Reduzindo esta velocidade de rotação, é necessária uma menor solicitação de energia para o funcionamento do compressor até um limite que permita que ele trabalhe em condições óptimas. A escolha do modelo do variador indicado é feita de acordo com as necessidades do motor do compressor. Na **Figura 4.5** mostra-se o variador de frequência da *Mitsubishi* utilizado e suas ligações eléctricas.

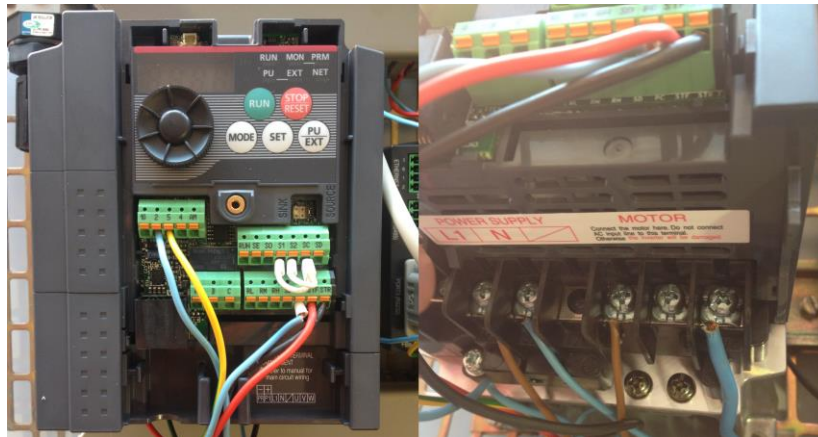


Figura 4.5 - Variador de Frequência *Mitsubishi* da série *D700-SC*.

Liga-se os bornes *Vo0+* e *GND* (*Ground*) da carta analógica do *PLC* às entradas 2 e 5 do variador (**Figura 4.5**), respectivamente. Atribuindo valores ao registo *D4076* do *PLC* via *WinProLadder* (que posteriormente é feito pela aplicação *Movicon*), varia-se uma tensão contínua, regulável, de 0 a 10V. Esta tensão é responsável pela frequência gerada à saída do variador para a entrada do motor monofásico do compressor. Nos bornes *L1* e *N* à direita, estão os dois fios responsáveis pela alimentação do variador. O neutro e a fase do motor monofásico ligam-se a dois dos três bornes (*U,V,W*) do variador, não importando quais, desde que dois distintos. Neste caso ligaram-se as fases do motor ao *U* e *W*, mas poderia ter sido *U* e *V* por exemplo.

Importante referir que alguns parâmetros do inversor são configurados de modo a se operar o motor monofásico do compressor sem ocorrência de problemas. O tempo de aceleração (*Pr.7*) e de desaceleração (*Pr.8*) são reduzidos de 5s (padrão) para 3s, pois observa-se problemas no arranque quando este demora a atingir a frequência de funcionamento. O *Pr.9* (relativo à corrente em Amperes) já se encontra em 7A por definição. O compressor também é de 7A, logo usa-se este valor padrão para o *Pr.9*. Outro problema que se verifica, é o alarme de aviso de perda de fase (*Pr.251*). Estes inversores estão pré-configurados para trabalhar com motores trifásicos, e quando se aplica apenas as duas fases do motor monofásico utilizado, um alarme de segurança faz com que o inversor se desligue automaticamente para prevenir danos ao equipamento. Por este motivo configura-se o *Pr.251=0* para que este alarme fique

desligado. É natural que haja perda de fase pois só se está a utilizar duas em vez de três.

4.1.5 Motor Monofásico

A função dos motores monofásicos é a de converter energia eléctrica, absorvida de uma rede monofásica, em energia mecânica. [35]

No sistema implementado neste trabalho, este motor é responsável pelo funcionamento do compressor ligado ao sistema de ar comprimido. Ao ser alimentado por uma determinada frequência (como referido na subsecção anterior e está explicado mais a frente), atinge a r.p.m necessária para o funcionamento do compressor. A **Figura 4.6** apresenta o aspecto do motor monofásico.



Figura 4.6 – Motor eléctrico monofásico do compressor utilizado.

É necessário ligar a fase e o neutro do motor aos respectivos *bornes* no variador de frequência como foi referido na subsecção anterior.

4.1.6 Compressor 1.5 HP

Um compressor de ar serve para converter energia eléctrica em energia cinética através do uso de ar comprimido. Quando o compressor liberta um jacto de ar, origina uma quantidade de energia cinética que pode ser aproveitada para a activação de sistemas pneumáticos ou enchimento de pneus, por exemplo.

Os compressores de ar funcionam em dois modos de operação diferentes: operação de compressão de ar e operação de libertação do mesmo. Quanto maior a potência do compressor, maior o caudal e pressão.

Neste trabalho o modelo do compressor utilizado pode ser visto na **Figura 4.7**.



Figura 4.7 – Modelo do compressor utilizado na implementação. [43]

Este compressor é eléctrico e efectua os dois modos de operação em simultâneo, isto é, o enchimento do tanque e o disparo do ar ocorrem em simultâneo após ligar o equipamento à corrente, o que o torna ideal para este estudo. Portanto pode-se manter a pressão e o caudal do ar numa gama óptima de funcionamento para dada operação através do controlo do seu motor.

4.1.7 Transmissor de pressão *Jumo MIDAS*

Este transmissor de pressão serve para medição de pressão em sistemas de ar comprimido. Na implementação deste trabalho é usado para medir a pressão à saída do compressor utilizado e enviar para o *PLC* o seu valor. Este valor é interpretado pela aplicação *Movicon* desenvolvida e lido no indicador de pressão da aplicação. Se a pressão lida for maior ou menor que um determinado valor definido como limite (*setpoint*), a aplicação envia essa informação ao *PLC* que irá regular a frequência do variador de acordo com as necessidades. Na **Figura 4.8** pode-se observar o sensor de pressão utilizado e sua ligação eléctrica.

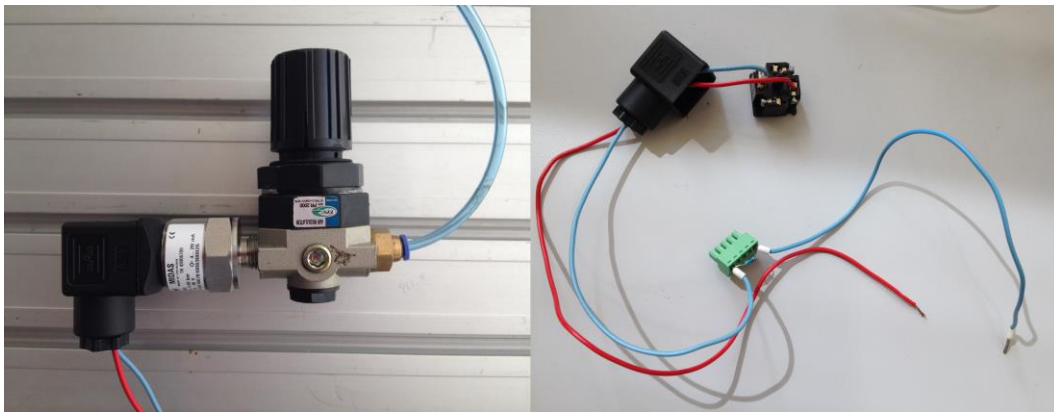


Figura 4.8 – a) Sensor de pressão utilizado; b) Ligação eléctrica do Sensor.

Na extremidade direita do sensor (**Figura 4.8 a**) liga-se o devido adaptador à tubagem que está sobre a pressão criada pelo compressor. A ligação eléctrica do equipamento é apresentada à direita (**Figura 4.8 b**). Na extremidade inferior pode-se observar os fios utilizados na ligação. Existe duas pontas soltas que servem para ligar aos *bornes* 0V (fio azul) e 24V (fio vermelho) do *PLC* para alimentação do sensor de pressão. As duas pontas que estão ligadas ao *borne* verde estão em série com uma resistência de 500 Ω (aproximadamente). Estas pontas ligam-se ao *Vi0+* e *GND* da carta do *PLC*. É necessário que se faça esta ligação para se poder ler uma tensão de 0 a 10V vinda do sensor. Como a sua saída é de 0 a 20mA, usa-se uma resistência de 500 Ω para termos um sinal de 0 a 10V no *PLC*.

4.1.8 Caudalímetro

Um caudalímetro serve para medir o caudal de determinado escoamento de ar. Neste trabalho é usado para se medir os mL/min (velocidade do fluido) de ar que é injectado no nosso sistema de ar comprimido. Sabendo a velocidade do fluido e o diâmetro da tubagem em estudo obtém-se o caudal.

Desenvolveu-se um caudalímetro a partir de um sensor de fluxo térmico *FS5A* que se baseia no princípio do fio quente. Consiste num elemento rectangular com duas resistências de platina. Através de um circuito electrónico em placa branca é possível aquecer a resistência menor, enquanto a resistência maior é usada para medir a temperatura do fluido em estudo. Estes sensores baseiam-se no coeficiente de transferência de calor, que é função da velocidade do fluxo. Esta provoca uma variação da energia térmica perdida na resistência aquecida, ou seja, quando a velocidade do fluxo de ar (no sistema de ar comprimido) aumentar tem-se um maior arrefecimento, o que provoca a variação do coeficiente de transferência térmica. A energia eléctrica fornecida ao circuito para se gerar esta diferença de temperatura é função da velocidade do fluido. A energia eléctrica é convertida num sinal de saída através do circuito e pode ser facilmente obtido pelo *PLC* e lido num *software SCADA*. [36] Na **Figura 4.9** pode-se observar a adaptação do sensor ao sistema de ar comprimido.



Figura 4.9 – Adaptação do caudalímetro ao sistema de ar comprimido.

Legenda:

- 1 – Entrada do fluido no sistema.
- 2 – Saída de ar para o sensor de pressão.
- 3 – Ligação para a válvula mecânica de consumo de ar.
- 4 – Entrada do caudalímetro isolada com o soldador de tubeiras.
- 5 – Ligações para o circuito eléctrico.

O sensor é adaptado ao sistema de ar comprimido como se pode observar na **Figura 4.10** recorrendo a um adesivo soldador para tubeiras em PVC. Pode-se ainda observar em detalhe o circuito eléctrico responsável pelo condicionamento do sinal. A alimentação do circuito é feita pelo *PLC* e o sinal de *output* (da **Figura 4.10**) em voltagem é enviado para o *Vi1+* do *PLC*. Este sinal, após ser amostrado e quantificado é alocado no endereço de memória D4073 para estudo no *PC* via *Movicon*.

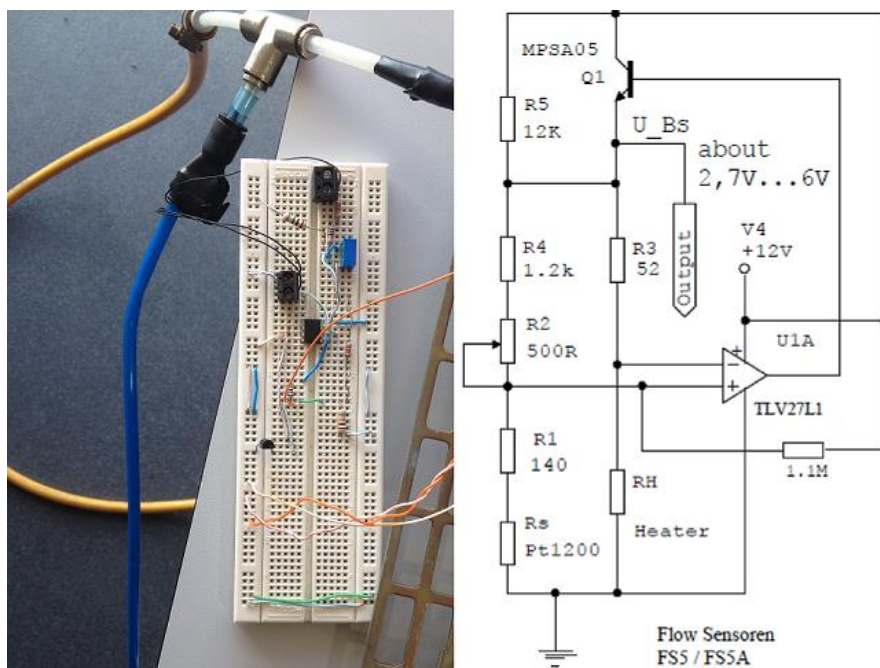


Figura 4.10 – Caudalímetro e esquema eléctrico do circuito.

4.1.9 Outro *hardware*

Para elaboração do circuito eléctrico que envolve todo o equipamento referido nas subsecções anteriores, utilizaram-se ainda dois disjuntores (Merlin Gerin – Multi9 – DPN-N), uma tomada (Legrand - 16A – 250V) e seis *bornes* de ligação. Na **Figura 4.11** pode-se observar este equipamento montado no circuito.

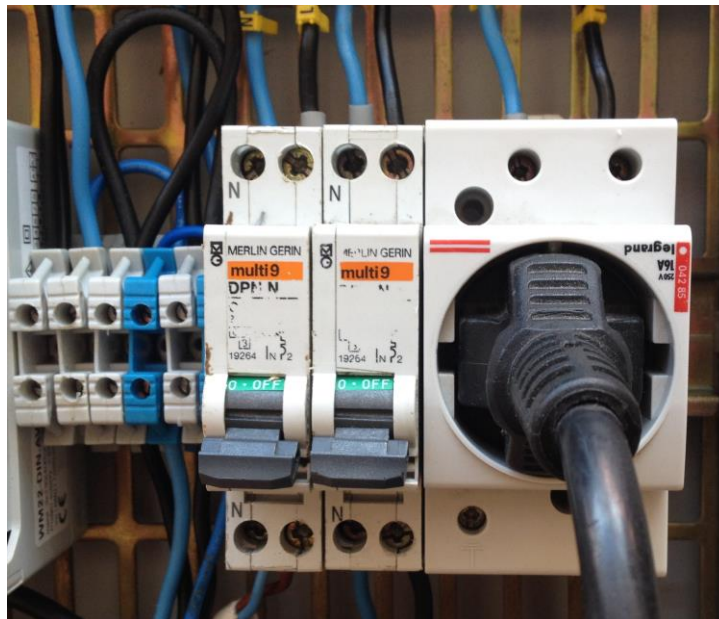


Figura 4.11 - Outros componentes eléctricos utilizados.

O disjuntor da esquerda fornece energia eléctrica (220V) ao circuito quando está ligado. A fase e o neutro da corrente são partilhados pelos *bornes* à esquerda. O primeiro par de *bornes* (a contar da esquerda) é responsável por conduzir a corrente eléctrica à fonte auxiliar do analisador de energia; o segundo par é responsável por fornecer corrente ao *PLC*; o terceiro par fornece corrente aos respectivos *bornes* do analisador de energia, segundo o diagrama eléctrico do manual do equipamento para a classe 5A como referido anteriormente na subsecção do *WM22*. O disjuntor da direita é responsável por fornecer energia ao equipamento em estudo, neste caso o compressor ligado à tomada. Nas ligações da tomada, a fase e o neutro estão ligadas aos *bornes* de duas das três fases do inversor como explicado na subsecção do variador.

4.2 Sistema em estudo

Nesta subsecção encontram-se as configurações do *hardware* utilizado na implementação. Trata ainda sobre os testes efectuados, com *Visual Studio 2012*, para confirmação da ligação e troca de mensagens *Modbus* com *RS-485* e *TCP/IP*. Aborda-se ainda, de forma detalhada, outras aplicações desenvolvidas na implementação para desenvolver o sistema proposto.

4.2.1 Configuração e teste do *WM22-DIN*

Inicialmente foi criada uma aplicação em *Visual Studio 2012* para verificar que era possível comunicar por porta série com o analisador de energia *WM22-DIN*. O protocolo de comunicação serie deste dispositivo é o *MODBUS-RTU*. [37]

Utilizou-se um conversor *USB-RS485* para ligar o computador ao dispositivo em estudo através de uma ligação por 2 fios. O formato dos dados de comunicação com o analisador têm que ter formato predefinido pelo fabricante: *1 start bit, 8 data bit, 1 stop bit, 9600 baudrate*, sem controlo de paridade. [37] Na **Figura 4.12** ilustra-se a ligação por 2 fios entre componentes.

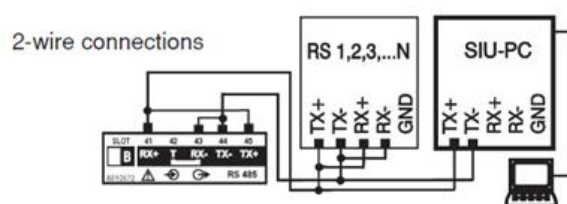


Figura 4.12 – Ligação *RS-485* por 2 fios. [31]

Após o estudo do manual do analisador, procedeu-se à ligação dos fios. Observa-se ainda que o analisador aceita mensagens com determinado formato padrão (formato *Modbus* sobre *RS-485*). Portanto, na aplicação em *VB*, criou-se um algoritmo

para enviar um array de bytes, em que cada posição nesse array fosse um bloco/palavra com a informação necessária para aceder ao lugar na memória do *WM22-DIN* correspondente à variável cujo estado se queria visualizar.

O array a enviar, pela aplicação *VB*, para a porta série tem que conter a seguinte combinação ordenada de elementos: endereço do analisador de energia (valor *01* – valor escolhido no menu de programação do dispositivo de *1* a *255*), código da função que define se o tipo de comando a enviar é de leitura ou escrita (*04* - leitura; ou *06* - escrita), endereço de memória que define os parâmetros que queremos obter (*02 4B* – leitura dos contadores de energia activa e reactiva totais – por exemplo), número de bytes necessários para invocar essa leitura (*00 04* – 4 bytes, nesse caso) e o valor do *CRC16* para a combinação *[01][04][02][0B][00][04]*(nesse caso – *80 67*). [37] O cálculo de *CRC16* pode ser feito no programa *WinProladder* ou em qualquer ferramenta disponibilizada *online*. Na **Figura 4.13** ilustra-se o procedimento de envio de mensagem do *PC* para o *WM22*.

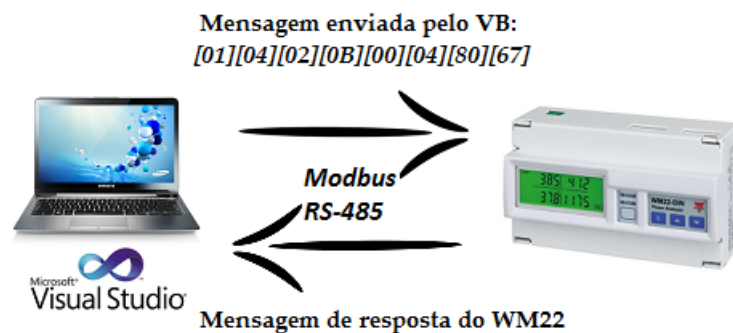


Figura 4.13 – Exemplificação da mensagem enviada para o *WM22*.

Recorrendo à aplicação no *PC* para envio dessa mensagem, obtém-se a resposta para a leitura dos contadores. Todos estes parâmetros para construção da mensagem estão no manual do protocolo do *WM22-DIN* e são inseridos na aplicação, um a um, de modo que só seja necessário um botão para cada ordem de leitura de determinada variável. Na **Figura 4.14** está ilustrada a versão de testes para confirmação de comunicação com o *WM22-DIN*.

O seu princípio de funcionamento, como se pode observar, é bastante básico, pois serve apenas para verificar se após envio de determinada mensagem, se obtém a resposta pretendida.

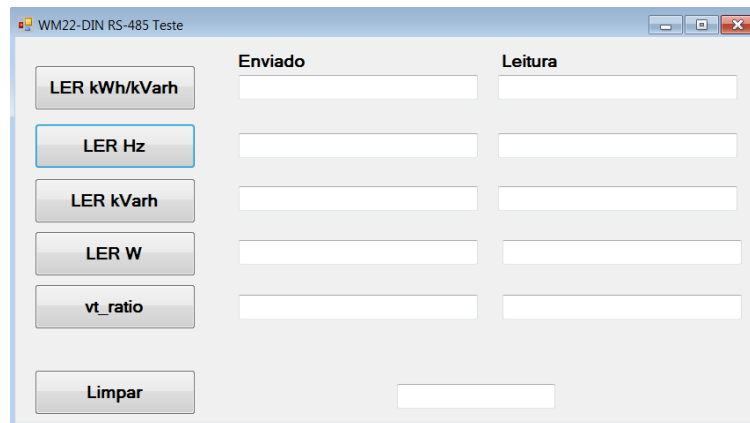


Figura 4.14 – Aplicação VB de testes para a comunicação Modbus com o conversor RS-485.

Este primeiro passo serve para verificar se o aparelho responde e se as ligações físicas ao aparelho estão correctas, isto é, se houver uma resposta com os valores semelhantes aos do visor de leitura do WM22-DIN, a leitura está correcta. Ainda para confirmar se a resposta recebida no computador é igual à resposta dada pelo analisador, usa-se um osciloscópio e seu software de apoio (*Picoscope 6*). Na Figura 4.15 pode-se ver a ilustração gráfica dessa troca de mensagens quando analisada com o *Picoscope*.

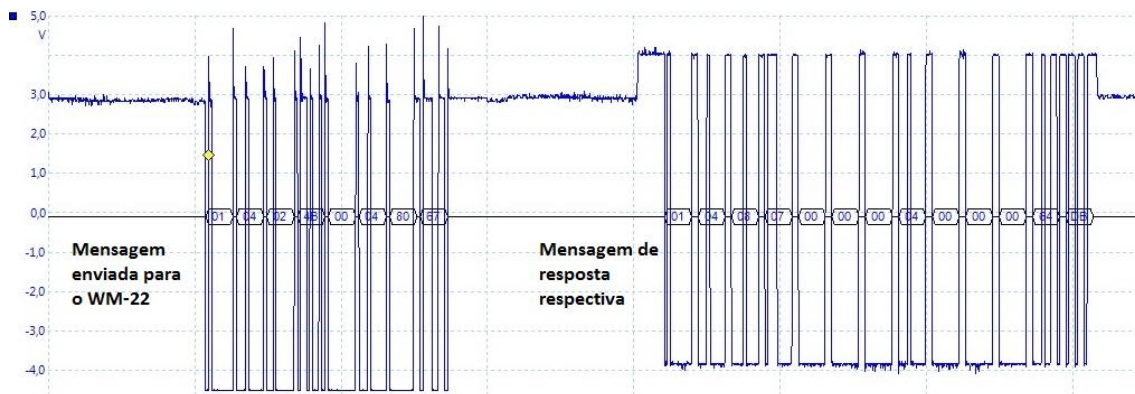


Figura 4.15 – Mensagem Modbus enviada e respectiva resposta observada no Picoscope.

4.2.2 Configuração e teste do Moxa Nport 5630

Após esta fase, ligou-se o *PC* ao *Nport 5630* em modo de Servidor *TCP* e o *WM22-DIN* ao *Nport* (como se ilustra na **Figura 4.16**) de modo que se pudesse enviar mensagens *Modbus* (sobre *RS-485*) do *PC* para o *WM22*, utilizando o *Nport* como intermediário. Em modo Servidor *TCP*, o *Nport* oferece um endereço *IP:port* único numa rede *TCP/IP*. Ele fica à espera de ser contactado pelo servidor para estabelecer uma ligação e para fazer a leitura de informação de um dispositivo em série ligado a uma das portas do *Nport*. É possível ligar 4 servidores em simultâneo para a transmissão de dados. [30] A **Figura 4.16** ilustra o modo de transmissão de dados neste modo de comunicação.

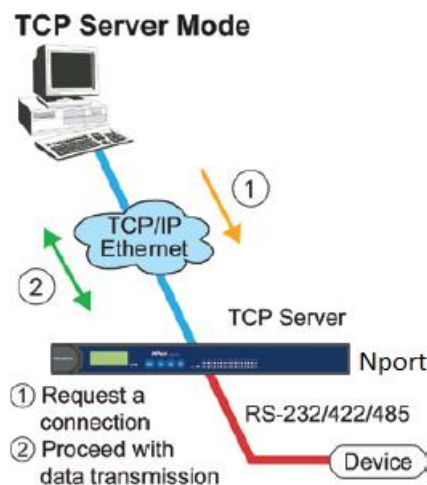


Figura 4.16 – Modo de comunicação Servidor TCP. [30]

Uma vez estabelecida a ligação, os dados podem ser transmitidos do *PC* para o *Nport* e vice-versa. [30]

No modo seleccionado, o *Nport* necessita de ser ligado por um cabo *Ethernet*, da sua porta *LAN* à do computador para a utilização do protocolo *TCP/IP*. De seguida procede-se à ligação da porta 1 com o *WM22* por *RS-485* (ligação por 2 fios como referido anteriormente na subsecção 4.2.1).

Estando a ligação física concluída, procede-se à configuração do *Nport*, começando pela escolha do *IP* que pode ser feita directamente no seu painel frontal ou na página *Web* (introduzindo o seu *IP* de origem no *browser* podemos aceder às definições e configurações do aparelho). Tanto no painel como na sua página *Web* é possível configurar os parâmetros do servidor, da rede, da comunicação série e o modo de operação. No painel frontal também se pode encontrar a opção de activação da consola *Web* ou *Telnet* (este ultimo serve para configuração através da linha de comandos no *Windows*) e fazer o *ping* ao *IP* do servidor para saber se a ligação está bem configurada.

O *software* de administrador do *Nport*, também possibilita configurações de rede e permite ficar à escuta se algum dos equipamentos ligados a qualquer uma das suas portas de saída se desligou por qualquer razão e executar alarmes sonoros no computador para aviso desses eventos. Se o equipamento voltar a ficar activo, outros avisos de notificação serão executados.

4.2.3 Aplicação em *Visual Studio 2012*

O *Microsoft Visual Studio 2012* é uma solução integrada que permite o desenvolvimento de todo o tipo de aplicações em várias linguagens de programação. É possível criar aplicações para *Windows*, para dispositivos móveis, páginas *web*, *scripts*, integração com diversas bases de dados entre muitas outras. [38] Para o trabalho desenvolvido é uma ferramenta bastante útil, pois permite um leque ilimitado de recursos para a integração de sistemas, e ainda deixa em aberto a possibilidade de adicionar outros módulos para outros dispositivos de controlo. Os módulos de comunicação série ou *TCP/IP (sockets)*, entre outros, são fundamentais para o controlo integrado de sistemas e são facilmente importados para a área de programação.

Este *software* é muito usado na indústria visto que é uma ferramenta poderosa para diversos sectores da engenharia. Tem uma interface *user friendly*, sendo fácil criar

objectos visuais como botões, caixas de texto, tabelas, gráficos, etc, e associar-lhes código de programação.

Neste trabalho, após se concluir os passos de configuração, criou-se uma nova aplicação, também em *Visual Studio* mas para comunicação *TCP/IP*. Esta aplicação tem um princípio de funcionamento semelhante à anterior, no que diz respeito à troca de mensagens entre equipamentos. Após verificação de uma ligação com o *IP* do *Nport* é possível a troca de mensagens entre o *PC* e o analisador de energia.

As mensagens são enviadas pelo *PC* por *TCP/IP* para a porta 1 (até 16 portas possíveis) do *Nport*, que por sua vez envia a informação por *Modbus* (sobre *RS-485*) ao *Slave WM22*. De seguida as mensagens são interpretadas pelo analisador de energia que gera uma resposta com os valores da variável que se pretende, também em formato *Modbus* (sobre *RS-485*). Esta mensagem é enviada de volta para o *Nport* que a direcciona para o *PC*. Sempre que as mensagens passam pelo *Nport*, os seus *leds* de confirmação acendem. Os dados chegam ao *PC* como um conjunto de valores todos seguidos que precisam de ser interpretados pela aplicação *VB*. Na **Figura 4.17** ilustra-se a troca de mensagens entre *PC* e *WM22*.



Figura 4.17 – Troca de mensagens entre aplicação VB e WM22.

A leitura por temporizadores no *VB* possibilita a leitura contínua da totalidade das variáveis em estudo. Esta leitura contínua de valores é importante para a criação de uma base de dados.

Na **Figura 4.18** pode-se observar o aspecto da nova aplicação VB. Na janela principal desta versão inicial, são facilmente perceptíveis as funcionalidades da aplicação. Cada botão é responsável pelo envio do pedido do estado de determinada variável e envia uma mensagem em formato *Modbus* para esse efeito. Na barra de estado indica-se a hora, o estado da ligação, a que *IP* esta ligação é pedida e a porta onde está ligado o *WM22*. Ao ligar o cabo *RS-485* á porta 1 do *Nport*, é necessário seleccionar a porta 4001 no VB. Ao optar pela porta 2, seleccionamos a porta 4002 (se temos 16 portas a configuração da porta vai da 4001 á 4016).

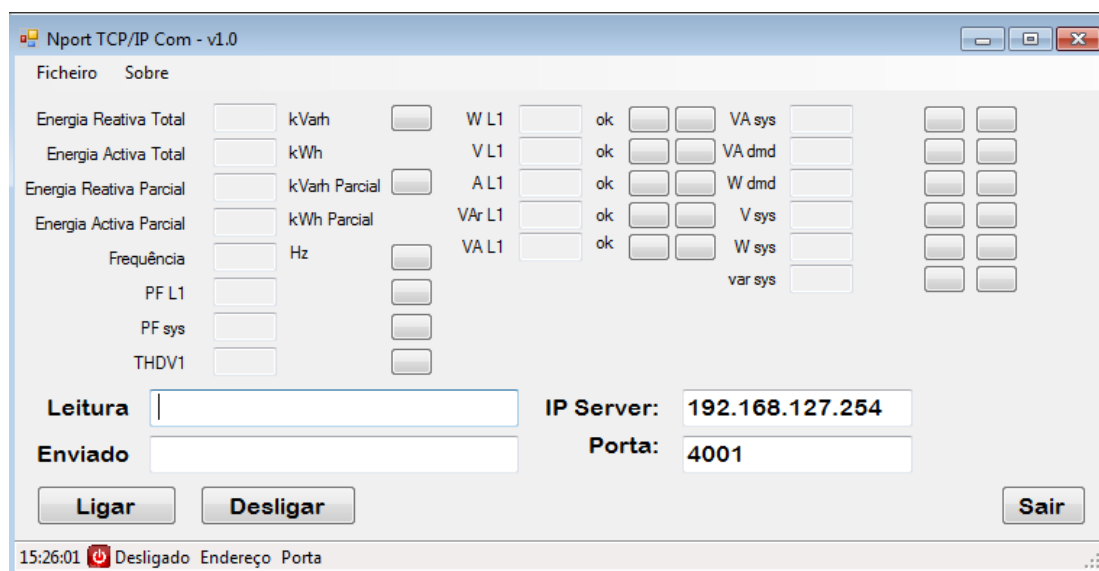


Figura 4.18 – Janela principal da aplicação em VB para a comunicação TCP/IP.

Concluída a fase de testes de troca de mensagens, procede-se ao aprimoramento da aplicação em *Visual Studio*. Adicionou-se as funcionalidades de leitura do estado actual da base de dados (janela que abre através do botão “*Data Base*”), botão de ligação à base de dados, botão de estado (abre o painel direito com informação sobre as mensagens trocadas com o *WM22* e os parâmetros de comunicação com o *Nport*), menu “*Help*” e “*File*”. No menu de ajuda encontra-se informações sobre o trabalho e o autor. A **Figura 4.19** mostra a nova versão da aplicação VB. A redução do número de variáveis de leitura é pertinente, pois nem todas são necessárias para o estudo do consumo de energia que se pretende.

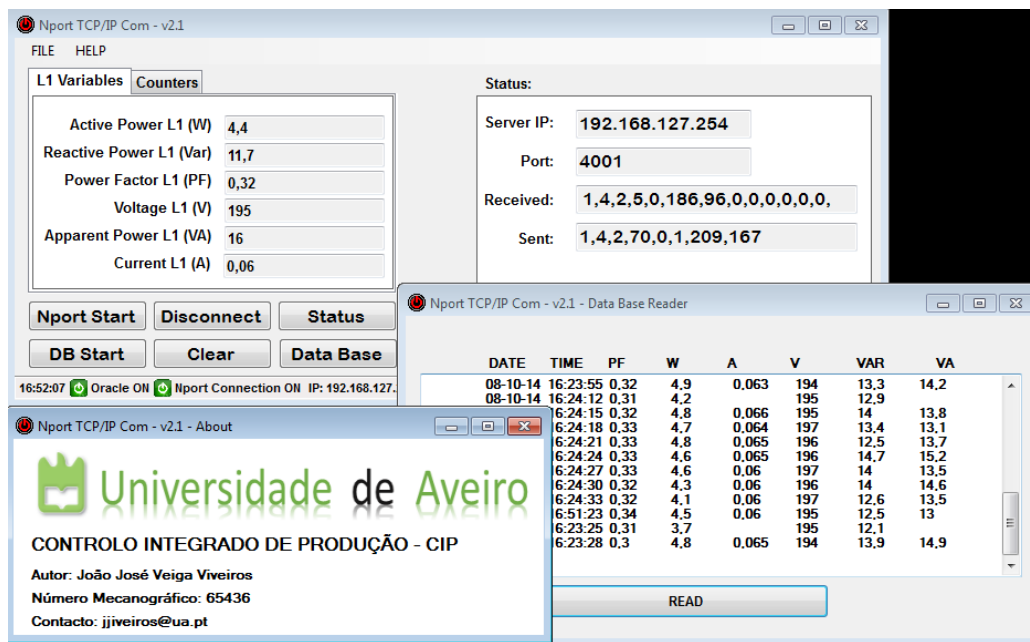


Figura 4.19 – Aplicação em Visual Studio 2012 – Versão melhorada.

4.2.4 Base de dados Oracle 11g

A base de dados *Oracle* é utilizada como ferramenta de estudo para o desenvolvimento deste trabalho. É uma ferramenta poderosa para armazenamento de informação.

A base de dados *Oracle* permite a criação de utilizadores para diversos departamentos e cada um com determinados privilégios de acesso definidos pelo administrador da base de dados. A criação de tabelas pode ser feita manualmente ou por programas externos em tempo real.

A base de dados *Oracle* é uma aplicação que pode ser usada em processos com arquitectura cliente/servidor. O cliente não necessita de grande capacidade de disco pois muitas vezes existem vários clientes em computadores diferentes que acedam à base de dados no servidor. O servidor é responsável gerir e armazenar a informação partilhada na rede. [39]

A manipulação de tabelas pode ser realizada de duas formas principais: pelo *Oracle SQL Developer* ou pelo *Oracle Enterprise Manager*. Estes softwares de gestão serão abordados nos subcapítulos seguintes.

4.2.4.1 Como aceder

Após instalação da base de dados no computador convém verificar várias coisas importantes para se conseguir uma ligação bem sucedida. Para garantir que está tudo funcional utiliza-se a linha de comandos do *Windows* para definir variáveis e activar serviços de forma manual:

- Verificação do estado do “*Listener*” (responsável pela ligação à base de dados) através do comando “*lsnrctl status*”. No caso de estar desligado, utiliza-se o comando “*lsnrctl start*”.
- Se solicitado define-se as variáveis de ambiente “*ORACLE_HOSTNAME*” e “*ORACLE_UNQNAME*” através dos comandos “*set ORACLE_HOSTNAME=localhost*” e “*set ORACLE_UNQNAME=orcl*”.
- Verificar se o serviço “*dbconsole*” está activo através do comando “*emctl status dbconsole*”. Se estiver desactivo usa-se o comando “*emctl start dbconsole*” para iniciar a instância da base de dados.

É aconselhável deixar os serviços do Oracle desligados por definição, e activá-los manualmente sempre que quisermos fazer testes, caso contrário o computador demora bastante tempo a iniciar. Se os serviços estiverem ligados por definição, é desnecessário este processo manual.

Concluídos os passos, pode-se entrar no OEM (*Oracle Enterprise Manager*) através da introdução do link <https://localhost:1158/em/> num *browser*. Nesta página Web, é possível gerir e monitorizar todos os utilizadores, conferir privilégios para aceder a determinadas opções, criar tabelas com determinados tipos de informação, ver o estado

da ligação, definir horários de manutenção, ligar/desligar a base de dados, etc. É necessário um “login” de administrador para se ter acesso total ao sistema. O administrador tem todos os privilégios e pode concedê-los a outros utilizadores da base de dados. [40]

4.2.4.2 Oracle SQL Developer

O software Oracle 11g permite a gestão de tabelas através do *browser* ou do *Oracle SQL Developer* (programa em ambiente *Windows*).

É possível aceder à base de dados via *browser* ou via aplicação *Windows* fazendo um pedido de ligação ao servidor OEM (*Oracle Enterprise Manager*) com os dados de utilizador. Após o acesso do utilizador ao servidor pela aplicação *SQL*, é possível a alteração de dados intrínsecos à tabela, criar novas tabelas, excluir tabelas, etc. Esta manipulação de dados gera códigos *SQL* para implementação em outros softwares, correcção de erros, adicionar novas tabelas ou adicionar dados às tabelas existentes. Tudo o que é manipulável por objectos pode ser convertido para código *SQL* ou então escrito em *SQL* para originar novos objectos. [41]

No presente trabalho a tabela criada foi nomeada de “*ENERGY_ANALYZER_TABLE*”. Esta tabela pode ser criada por *Visual Basic*, mas visto ser um banco de dados existente, procedeu-se a sua criação manual através do gerente *SQL* para ser manipulada posteriormente pela aplicação.

Nesta fase, a integração do WM22 com a base de dados está concluída, pois é possível monitorizar e gerir um histórico de dados, a partir de uma plataforma comum. Após algumas leituras com a aplicação *VB* criada, verifica-se o estado dos dados armazenados no gerente *SQL*. A figura **Figura 4.20** ilustra algumas dessas leituras no referido *software*.

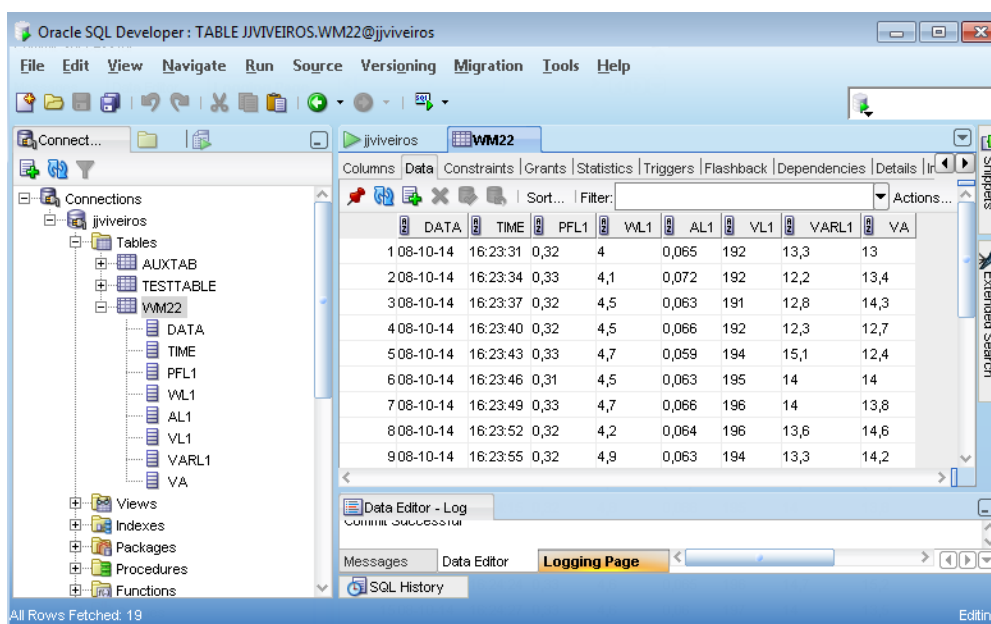


Figura 4.20 – Verificação dos dados através do Oracle SQL Developer.

4.2.4.3 Oracle Enterprise Manager

O *Oracle Enterprise Manager* combina a consola de gestão no sistema operativo, serviços comuns e plataforma integrada de ferramentas gráficas. Funciona como um conjunto de aplicações integradas para a gestão da base de dados e simplifica as tarefas diárias dos administradores. Funciona como um servidor que armazena os dados provenientes de outras aplicações clientes, ou até introduzidas manualmente no nesse servidor por cada utilizador, seja o administrador ou não. Cada utilizador com os devidos privilégios pode aceder à sua área de gestão, manipular as suas tabelas e visualizar se os dados foram escritos correctamente caso tenham sido enviados por aplicações cliente externas. O administrador do servidor atribui os privilégios aos utilizadores, gere, cria, altera e determina espaços para as tabelas, etc.

O *OEM* permite administrar, diagnosticar, gerir servidores *Web*, gerir aplicações e serviços, calendarizar tarefas em diferentes sistemas em intervalos de tempo variáveis, monitorizar as condições da base de dados através da rede, partilhar tarefas com outros administradores, gerir relatórios em *HTML*, entre outros. [42]

Outro modo de gerir os dados contidos nas tabelas, como referido anteriormente, seria através do *Oracle SQL Developer*. Na **Figura 4.21** ilustra-se a tabela da base de dados com a informação proveniente da aplicação em *Visual Studio*, mas agora acedida através do *browser*.

ORACLE Enterprise Manager 11g
Database Control

Database Instance: orcl168.59.213 > Tables >
View Data for Table: JJVIVEIROS.WM22

Query: `SELECT "DATA", "TIME", "PFL1", "WL1", "AL1", "VL1", "VARL1", "VA" FROM "JJVIVEIROS"."WM22"`

DATA	TIME	PFL1	WL1	AL1	VL1	VARL1	VA
08-10-14	16:23:31	0.32	4	0.065	192	13,3	13
08-10-14	16:23:34	0.33	4,1	0.072	192	12,2	13,4
08-10-14	16:23:37	0.32	4,5	0.063	191	12,8	14,3
08-10-14	16:23:40	0.32	4,5	0.066	192	12,3	12,7
08-10-14	16:23:43	0.33	4,7	0.059	194	15,1	12,4
08-10-14	16:23:46	0.31	4,5	0.063	195	14	14
08-10-14	16:23:49	0.33	4,7	0.066	196	14	13,8
08-10-14	16:23:52	0.32	4,2	0.064	196	13,6	14,6
08-10-14	16:23:55	0.32	4,9	0.063	194	13,3	14,2

Figura 4.21 - Verificação das trocas de informação no sistema através do Oracle Enterprise Manager

O código *SQL* para as tabelas também é gerado e pode ser posteriormente utilizado para construção por *SQL* no *Oracle SQL Developer* ou noutros *softwares* de programação.

Concluída esta etapa de integração com o analisador de energia, desenvolveu-se uma aplicação *Movicon* para os efeitos explicados na secção seguinte.

4.2.5 Aplicação *Movicon*

Após a integração dos componentes descritos nas subsecções anteriores criou-se uma aplicação *Movicon* para controlar a saída analógica (0 a 10v) do *PLC*, que serve para regulação da frequência de saída do variador com a intenção de se controlar a velocidade de saída do motor monofásico e se conseguir um novo gasto energético para o compressor (como referido anteriormente).

O objectivo principal desta aplicação consiste em controlar o motor do compressor de forma manual ou automática e ainda monitorizar a pressão e o caudal do sistema de ar comprimido. Além disso, tal como na aplicação em *VB*, é possível monitorizar o estado das variáveis lidas no analisador de energia através da leitura directa dos respectivos endereços de memória em caixas de texto, gráficos ou *Data Loggers*.

Na página inicial da aplicação *Movicon* existe um botão *Start* para ligar o motor e permitir o controlo do painel virtual; permite definir um *setpoint* máximo e mínimo (pressão) de funcionamento para determinada operação através das barras deslizantes respectivas; permite o controlo manual da velocidade do motor monofásico do compressor ajustando uma dada frequência ao variador através da barra deslizante respectiva. Quando a pressão lida no sensor ultrapassar um determinado valor seleccionado na barra deslizante do *setpoint* máximo, a aplicação faz com que o motor acelere ou desacelere para manter essa pressão na gama de valores desejada. O valor de pressão lido no sensor pode ser monitorizado na aplicação *Movicon* (no indicador de pressão que vai de 0 a 10 bar). O indicador de caudal indica o valor lido pelo caudalímetro no escoamento de ar. Na **Figura 4.22** ilustra-se o painel de controlo da aplicação desenvolvida. De notar que os restantes objectos (*PLC*, Variador, Motor monofásico, Sensor de pressão, Caudalímetro) servem apenas de ilustração do que se pretende, não tendo qualquer funcionalidade na aplicação.

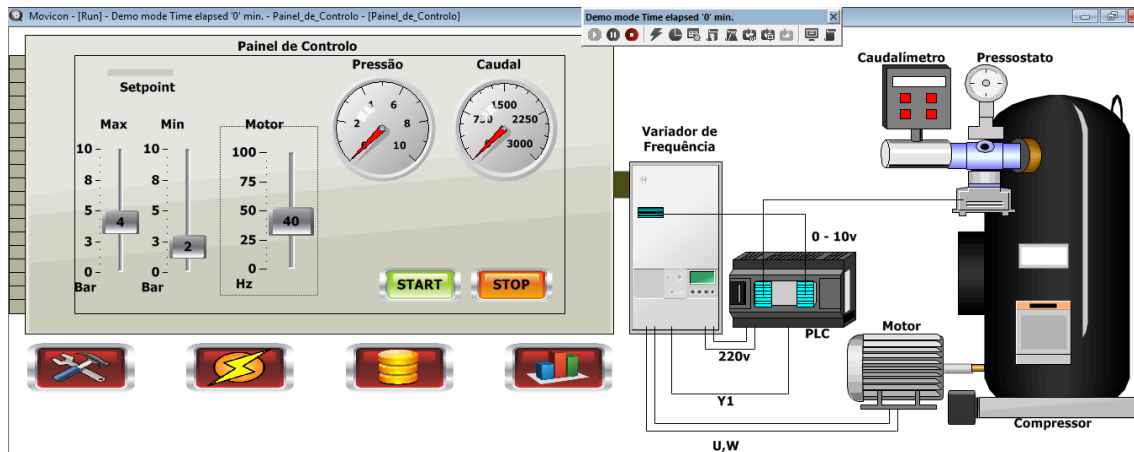


Figura 4.22 – Aplicação em *Movicon* para controlo e monitorização do sistema de ar comprimido.

Na página seguinte da aplicação, observa-se alguns valores das variáveis do analisador de energia em caixas de texto (Figura 4.23). Note-se que os valores indicados na figura são obtidos com o compressor em repouso.

Para leitura destas variáveis, encontraram-se algumas dificuldades, entre elas a configuração do modo de funcionamento do *Nport* e a configuração compatível no *Movicon*, que está explicado em detalhe na subsecção seguinte.

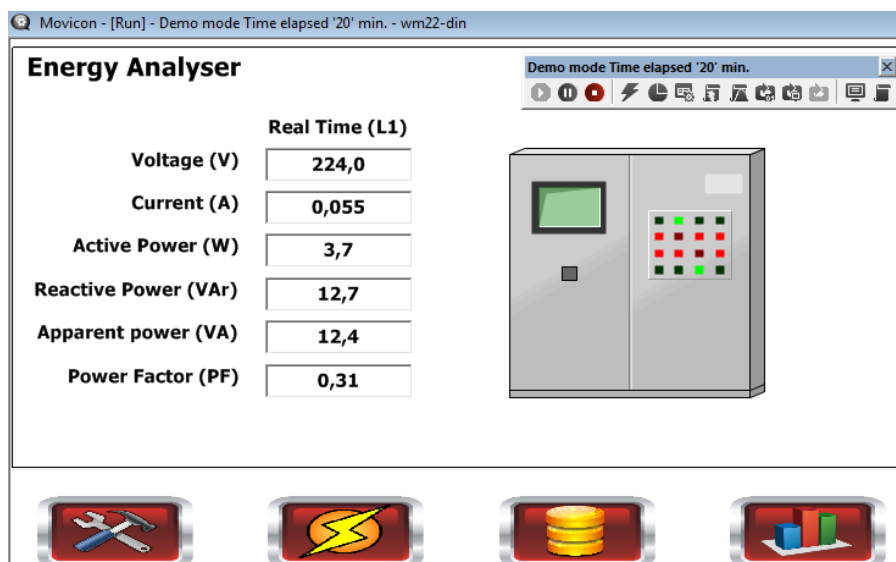


Figura 4.23 – Página de leitura de variáveis eléctricas do *WM22*.

Na terceira página tem-se a leitura dos valores numa tabela associada a um *Data Logger* que é responsável pela leitura e escrita dos dados estudados na base de dados *Oracle*. Por fim, na quarta página, obtém-se um gráfico de leitura dos valores pontuais e médio do consumo energético, em *Watts (W)*, ao longo do tempo. Na **Figura 4.24** e **Figura 4.25** observa-se a tabela e o gráfico em questão. Os dados apresentados pertencem ao método do Intervalo de Frequências explicado na subsecção 4.3.2. Os valores da tabela devem ser interpretados com uma casa decimal, excepto para as variáveis *A* e *PF* (*Current*: 3 casas decimais; *Power Factor*: 2 casas decimais; respectivamente).

Event Time	User	V	A	W	VAr	VA	PF
2014-10-16 12:...		2230	5728	8479	9555	12775	190
2014-10-16 12:...		2223	6556	9502	11058	14580	191
2014-10-16 12:...		2225	5769	6547	11043	14247	205
2014-10-16 12:...		2225	5556	9852	10818	14632	189
2014-10-16 12:...		2226	6184	9257	10125	13691	189
2014-10-16 12:...		2227	5960	8877	9871	13276	189
2014-10-16 12:...		2232	5708	8474	9514	12741	189
2014-10-16 12:...		2223	6512	9849	10611	14477	188
2014-10-16 12:...		2225	6049	6747	11649	13462	206
2014-10-16 12:...		2236	5156	6669	9404	11529	198
2014-10-16 12:...		2228	6242	9323	10355	13934	189
2014-10-16 12:...		2227	6195	9048	10431	13786	190
2014-10-16 12:...		2229	5798	8486	9906	13106	190

Figura 4.24 – Tabela de valores estudados no *Movicon*.

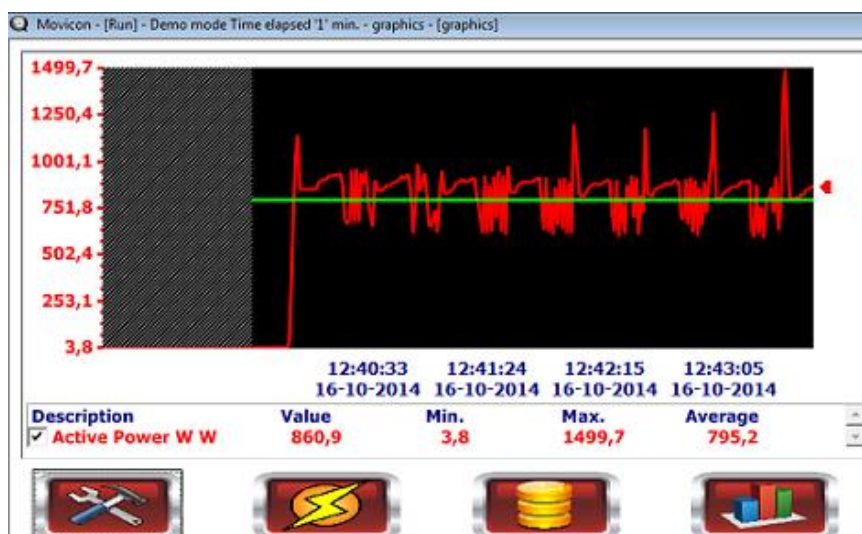


Figura 4.25 – Gráfico obtido no *Movicon*.

4.2.5.1 Configuração *Nport* - *Movicon*

Para haver comunicação entre o *Movicon* e o analisador de energia por intermédio do *Nport*, é necessário ter em conta algumas configurações específicas. Anteriormente referiu-se que na aplicação em *Visual Studio*, o *Nport* foi configurado para funcionar em modo servidor *TCP*. Este modo de operação recorre ao protocolo *Modbus TCP/IP*. Para este modo tem que se utilizar um driver de comunicação no *Movicon* do tipo *Modbus TCP*. Ao enviar uma mensagem deste tipo, da aplicação *Movicon* para o *Nport*, esta é recebida com sucesso e retransmitida para o analisador de energia. O analisador recebe a mensagem mas não transmite resposta para o *Nport*, pois só se consegue fazer um pedido de leitura de determinada variável, se este for feito no formato *Modbus RTU*. Sabendo que a ligação do *PC* para o *Nport* é feita por porta *Ethernet*, o tipo de mensagem trocado será sempre do tipo *TCP* e não se consegue utilizar o protocolo adequado para comunicar com o analisador sem se ter em conta outro factor. Para resolver este problema, o *Nport* tem que ser configurado para operar em modo *Real COM*. Neste modo, se o analisador estiver ligado à porta 4001 do *Nport*, o *Movicon* pode utilizar um driver *Modbus RTU* configurando os parâmetros da comunicação série adequados para operar na *COM1*. Desta forma a porta *Ethernet* do *PC* será considerada a *COM1* para a comunicação série. Note-se que ao ligar o analisador à porta 4002, os parâmetros série no *Movicon* têm de ser configurados para operar na *COM2*. O formato da mensagem enviada, neste modo de operação, é do tipo *Modbus TCP*, mas o *Nport* fica responsável por transformá-la no tipo *Modbus RTU*, de forma a se conseguir pedir informação ao analisador. Na **Figura 4.26** à esquerda, ilustra-se uma mensagem típica *Modbus TCP*, onde os primeiros 6 bytes são indicativos deste tipo de comunicação, e os 6 bytes seguintes são iguais à mensagem série que se pretende enviar. Na **Figura 4.27** tem-se um exemplo de uma mensagem do tipo *Modbus RTU* (enviada para o analisador em modo *Real COM*) seguida da resposta respectiva.

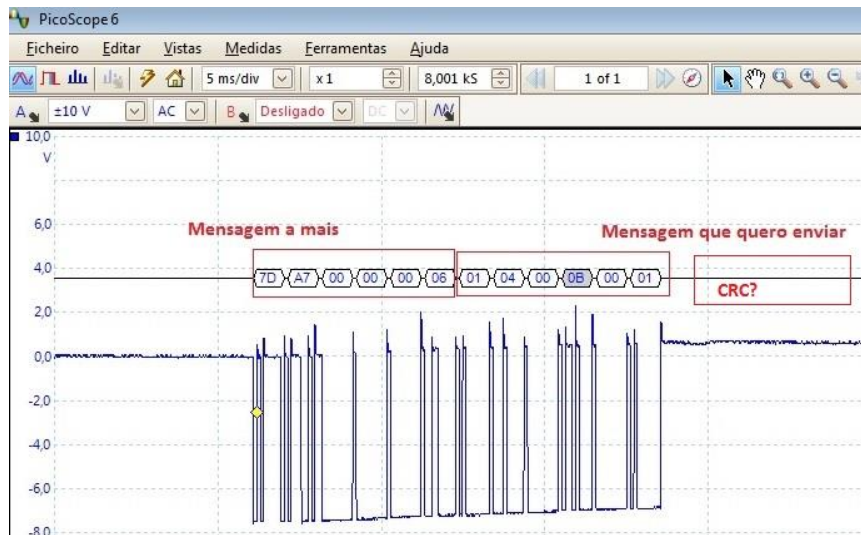


Figura 4.26 – Mensagem Modbus TCP enviada para o WM22.

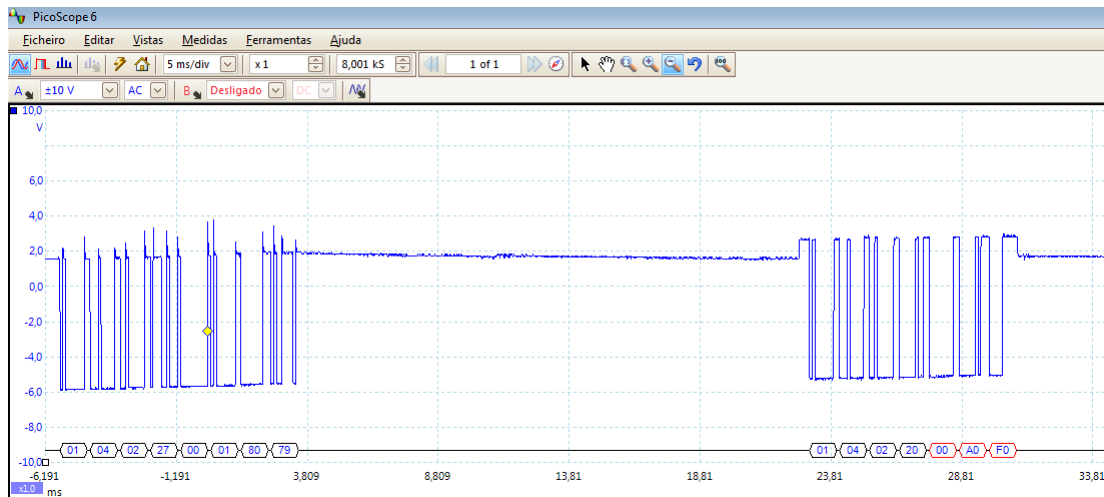


Figura 4.27 – Mensagem Modbus RTU enviada para o WM22.

Para se conseguir uma troca de mensagens bem sucedida entre o PC e o WM22, observa-se na Figura 4.28 e Figura 4.29 as configurações efectuadas.

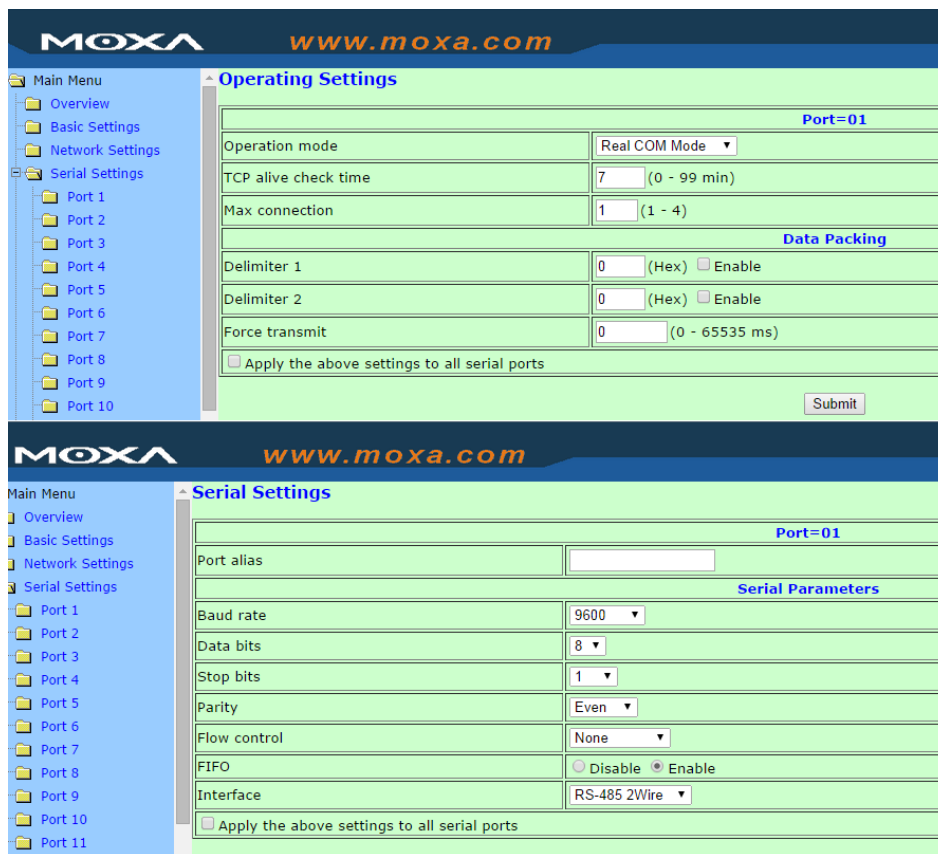


Figura 4.28 – Painel de configuração do *Nport* e parâmetros utilizados.

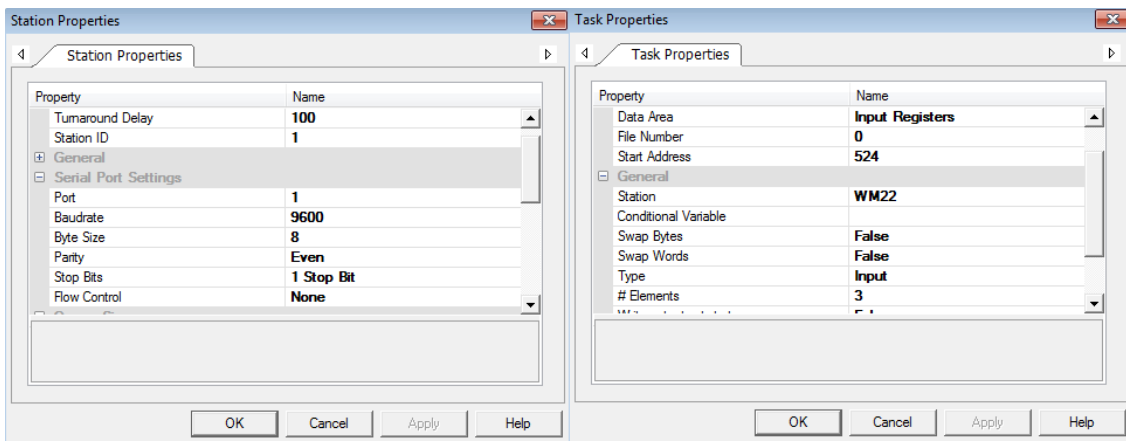


Figura 4.29 – Configuração de estação e de tarefa no *Movicon*.

Na Figura 4.29 à esquerda observa-se os parâmetros série escolhidos, nas propriedades de estação, para actuar na *COM1* (*port 1*) do *PC*. À direita observa-se a configuração de determinada mensagem a trocar com o analisador. Escolheu-se *Input*

Registers na Data Area, isto é, escolheu-se a função “04” para leitura da variável no endereço $020C(\text{Hex}) = 524(\text{Dec})$, que corresponde ao consumo em *W* [37]. A *Station* criada no *Movicon* com estes parâmetros de comunicação, é a *WM22*. A opção *Swap Bytes* e *Swap Words* estão em *False* para não se trocar a ordem da informação lida no programa. O tipo é *Input* (referente à função 04 do protocolo *Modbus*), e o *#Elements* indica o número de *bytes* que é necessário requisitar para leitura desta variável [37].

Concluída esta etapa, fecha-se a malha do sistema no que diz respeito à implementação. Na secção seguinte aborda-se o sistema global implementado e a análise de resultados da experiência.

4.3 Análise de resultados

Nesta secção ilustra-se o sistema implementado na sua globalidade e confrontam-se os valores dos consumos obtidos, com o analisador de energia, a partir de diferentes métodos para controlo do compressor.

Na **Figura 4.30** observa-se a montagem final dos componentes físicos envolvidos na implementação e a legenda simplificada.

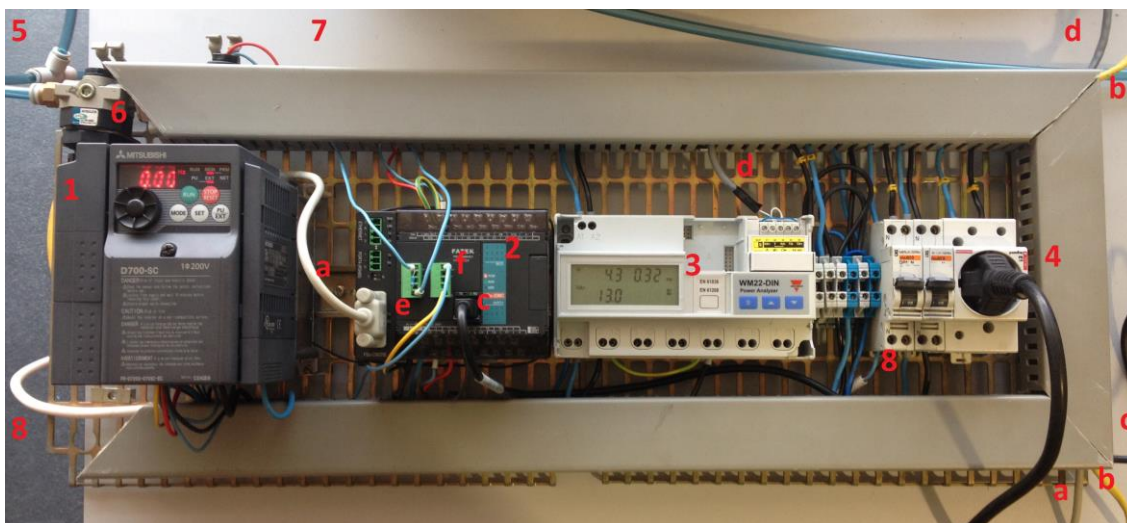


Figura 4.30 – Circuito eléctrico desenvolvido.

Legenda:

- | | |
|-------------------------------|--|
| 1- Variador de Frequência; | a- Modbus RTU (RS232) -> Movicon |
| 2- PLC; | b- Ethernet -> Nport |
| 3- WM22-DIN; | c- RS232 -> Ladder |
| 4- Alimentação do compressor; | d- Modbus (RS485) -> Nport |
| 5- Sistema de ar comprimido; | e- Monitorização do Sensor de Pressão. |
| 6- Regulador de pressão; | f- Controlo do Variador de Frequência. |
| 7- Sensor de Pressão; | |

Na **Figura 4.31** apresentam-se algumas linhas da tabela da base de dados relativas ao funcionamento normal do compressor, sem aplicação de qualquer método para redução de consumo de energia.

DATE	TIME	PF	W	A	V	VAR
08-10-14	15:14:13	0	0	0	200	0
08-10-14	15:14:16	0,74	1478,5	0	135	987,2
08-10-14	15:14:19	0,75	1103,7	6,784	133	986,6
08-10-14	15:14:22	0,75	1106,2	6,786	135	982,7
08-10-14	15:14:25	0,75	1104,4	6,774	133	973,8
08-10-14	15:14:28	0,76	1116,5	6,794	133	966,8
08-10-14	15:14:31	0,76	1117,6	6,79	133	966,4
08-10-14	15:14:34	0,75	1117,3	6,776	134	968,1
08-10-14	15:14:37	0,75	1117,7	6,783	136	972,7
08-10-14	15:14:40	0,75	1116,9	6,78	136	974,8
08-10-14	15:14:43	0,75	1112,7	6,773	137	974,5
08-10-14	15:14:46	0,75	1109,5	6,768	136	976,3
08-10-14	15:14:49	0,75	1103,2	6,737	135	976

Figura 4.31 – Informação da base de dados obtida na aplicação VB.

Observa-se que à frequência normal de funcionamento (50 Hz) o compressor tem uma potência activa de 1478.5 W e potência reactiva de 987.2 VAR no arranque. Em média o compressor consome 1139 W e 975 Var neste pequeno intervalo de tempo. A potência activa tende a estabilizar perto dos 1,1 kW como é de esperar, pois é coerente

com a informação da placa do compressor. Nota-se ainda que o factor de potência médio é de 0,75 e a corrente é de 6.8 A.

De forma a reduzir a energia gasta pelo compressor, são realizados experimentos com métodos diferentes que respeitem um *setpoint* de 2 a 4 bar (por exemplo). Este intervalo de pressão pode facilmente ser mantido com uma torneira reguladora de pressão, mas dessa forma o compressor gasta energia que não está a ser aproveitada.

Nas subsecções seguintes, são explicados os métodos aplicados ao sistema para se conseguir uma redução do consumo de energético, seleccionando, controlando de forma automática e em tempo real os *setpoints* de pressão convenientes.

4.3.1 Método ON-OFF

O método *ON-OFF* é o mais simples dos métodos utilizados. Ao atingir a pressão máxima estipulada, durante um funcionamento a 50 Hz, desliga-se o compressor e continua-se a consumir o ar para determinada operação. Ao consumir o ar por uma válvula mecânica aplicada na tubagem do sistema de ar comprimido, a pressão diminui até chegar ao *setpoint* mínimo definido, e o motor volta ao funcionamento normal para repor a pressão perdida. Sem o *setpoint*, a pressão continuaria a subir normalmente.

Observa-se que este é um método eficaz, pois consome menos energia que numa situação de funcionamento normal (sem *setpoints*), mas a longo prazo, os picos de arranque podem reduzir a vida útil do equipamento.

Na **Figura 4.32**, pode-se observar o gráfico da aplicação *Movicon* com os dados obtidos durante a aplicação do método *ON-OFF*.

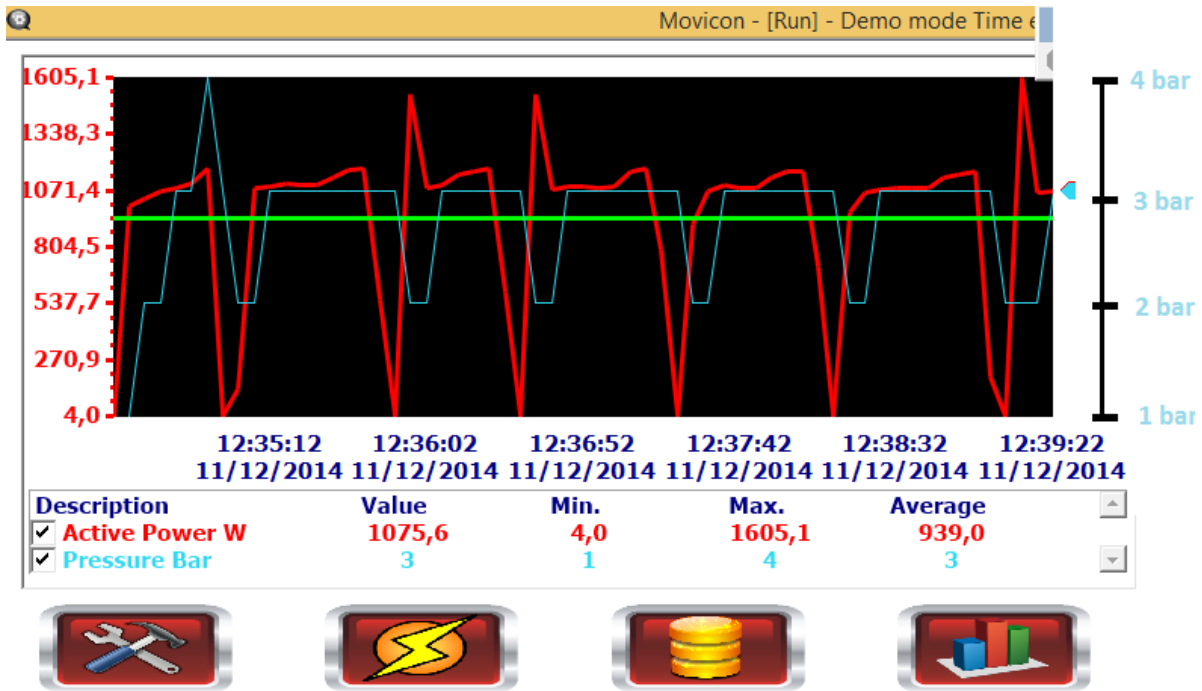


Figura 4.32 – Gráfico do consumo global/médio do compressor e respectiva pressão de ar no sistema de ar comprimido – Método *ON-OFF*.

Nesta figura procura-se recolher informação sobre um intervalo de valores quando o compressor está ligado e quando ele é forçado a desligar por atingir o *setpoint* máximo. Agora, pode-se observar que a potência activa (*W*) atinge 1600 *W* de pico de arranque e em determinados períodos (quando está desligado) o compressor não consome praticamente energia nenhuma. A média de consumo energético neste intervalo de tempo é de 939 *W*. Ao utilizar este método obtém-se, em média, um consumo de 0,94 *kW*.

Note-se que para esta experiência usou-se um intervalo de *setpoints* baixo de 2 a 4 *bar*. Normalmente não se irá obter uma redução de consumo tão acentuada se a operação de consumo de ar necessitar de um intervalo de 6 a 8 *bar*, por exemplo. Note-se ainda que o diâmetro de secção da tubagem da instalação de ar comprimido experimental é de 8 *mm* em material polimérico, e portanto, justifica-se a escolha dos limites de pressão atribuídos. Desta forma não se corre o risco de sobrecarregar o sistema ou haver colapso das tubagens.

De facto consome-se menos energia pois só se utiliza energia eléctrica quando a pressão não é suficiente para a tarefa em execução, mas temos um desgaste maior do equipamento devido aos picos de arranque ao longo do tempo. Verifica-se ainda que temos uma gama de variação de pressão de 2 *bar*.

Podemos reduzir os picos de arranque e a gama de valores de pressão aplicando outros métodos, nomeadamente o da subsecção seguinte.

4.3.2 Método do Intervalo de Frequências

Neste método a ideia dos *setpoints* mantém-se, mas agora não se desliga o compressor quando este atinge a pressão do *setpoint* máximo. Em vez de se desligar o compressor a dada pressão, trabalha-se com ele ligado a uma frequência suficiente para manter o intervalo de pressões desejado. Neste caso, escolheu-se 40 *Hz* para atingir essa pressão. Ao atingir esse *setpoint* máximo, a frequência de funcionamento aplicada ao motor do compressor irá ser reduzida, por exemplo, para 25 *Hz*. Quando se reduz esta frequência, o motor do compressor reduz as *r.p.m.*, e não consegue ultrapassar essa pressão máxima estipulada. Se a válvula mecânica de libertação de ar estiver aberta para a execução de determinada tarefa, a pressão tende a descer até ao *setpoint* mínimo, e a aplicação volta a aumentar a frequência para os 40 *Hz* para se estabilizar a pressão. O intervalo de pressões que se pretende trabalhar é escolhido pelo utilizador do *software SCADA*. Na **Figura 4.33** pode-se observar alguns dados relativos a este método de funcionamento.

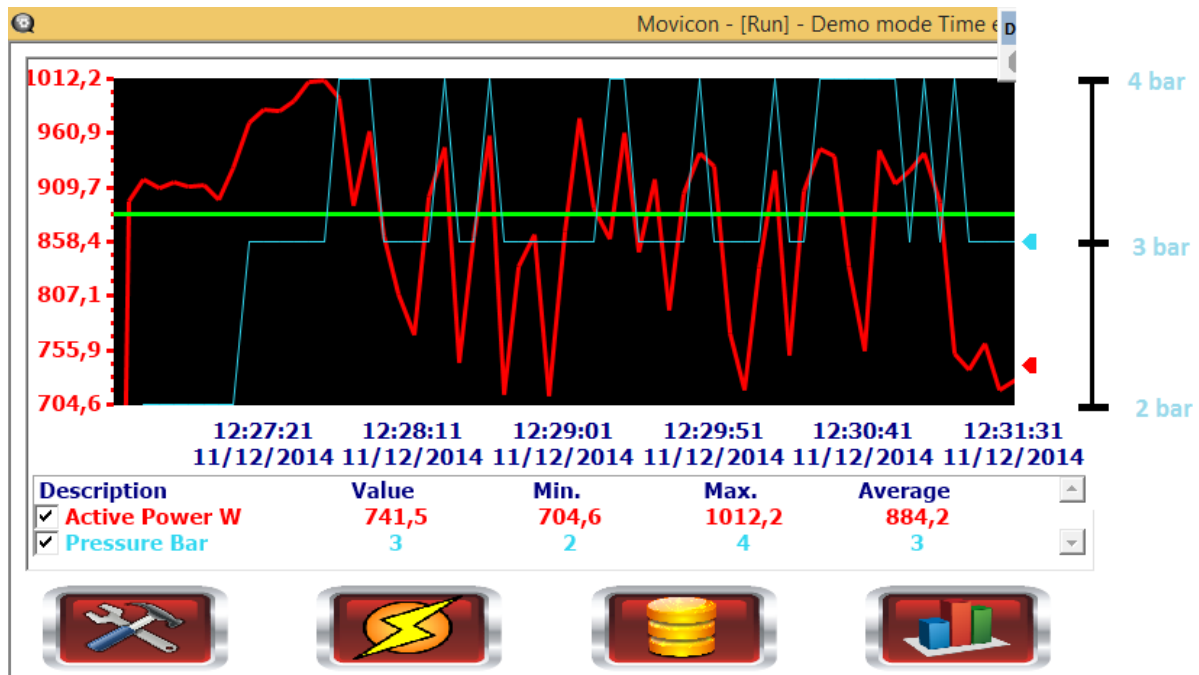


Figura 4.33 - Gráfico do consumo global/médio do compressor e respectiva pressão de ar no sistema de ar comprimido – Método do Intervalo de Frequências.

De acordo com os dados obtidos, comprova-se a poupança energética, e desta vez com picos de arranque muito mais suaves sempre que se voltar ao *setpoint* mínimo, pois agora o motor não desliga e dispensa o esforço adicional que tem ao arrancar do zero. Por este método, consegue-se uma poupança energética significativa sem a deterioração do equipamento com os constantes arranques iniciais.

Analisando os dados recolhidos durante a aplicação deste método, consegue-se um valor médio de 884,2 W (0,88 kWh) de consumo energético. Temos uma diferença significativa em relação há situação inicial (sem método de poupança aplicado ao sistema) e ainda uma diferença de consumo em relação ao Método *On-Off*. Esta diferença torna-se mais significativa à escala industrial, onde se pode verificar um maior número de compressores de maior potência e a funcionar mais horas diárias. Observa-se ainda que os picos de arranque atingem os 1012 W (em média) por este método, enquanto no método *ON-OFF* atingem os 1600 W. Esta “suavização” de arranque do compressor pode fazer com que ele tenha um maior tempo de vida útil.

No Método *On-Off* obtém-se um consumo energético 5% superior ao Método do Intervalo de Frequências e ainda se verifica a desvantagem de ter uma variação de pressão maior que pode não atender às necessidades de determinada tarefa nos momentos que atingimos o *setpoint* mínimo. No Método do Intervalo de Frequências, podemos admitir *setpoints* fixos ou mais reduzidos, isto é, com menos de 2 *bar* de diferença entre o *setpoint* máximo e mínimo, trabalhando com diferentes intervalos de frequência e funcionamento.

Capítulo 5 - Conclusões e trabalho futuro

Para cumprimento dos objectivos propostos e para realização deste documento procurou-se dividir o trabalho descrito em quatro fases. A primeira fase consiste em angariar, organizar e seleccionar informação a partir de várias soluções académicas propostas por outros, informação documentada sobre tecnologias de comunicação existentes e ainda informação sobre *softwares* utilizados na área do controlo integrado. Concluída esta fase, procedeu-se a fase seguinte que consiste em propor uma solução para o problema em mãos, com base em toda a informação recolhida e sintetizada. Na terceira fase, propõe-se uma solução para por em prática toda a teoria consolidada. Na fase final, que diz respeito ao presente capítulo da dissertação tiram-se as conclusões sobre a globalidade do trabalho desenvolvido.

5.1 Conclusões

Podemos concluir que um problema desta natureza, facilmente se torna muito complexo, pois num ambiente industrial o número de equipamentos e de dados que circulam em cada sector de produção pode ser elevado. Ao aplicar soluções de integração de sistemas podemos transformar o “caos industrial” num arquivo de fácil consulta, realizando o tratamento de dados e a reprogramação dos equipamentos numa unidade central, substituindo a necessidade de deslocação a cada sector da linha de produção. Trabalhos neste sector tornam-se fundamentais nos dias de hoje para se reduzir custos, tanto de mão-de-obra como de maquinaria e/ou consumos energéticos.

No caso abordado neste documento procurou-se reduzir o custo de energia através do estudo do seu consumo pelo compressor, utilizando diferentes métodos de funcionamento. Uma das mais valias importantes a ter em conta sobre o trabalho desenvolvido é o facto de ambos os métodos serem facilmente aplicáveis aos

compressores de uma instalação fabril sem haver troca do equipamento já existente, evitando um custo acrescido à sua implementação.

Conclui-se de facto, que o sistema proposto é eficiente pois através da leitura das variáveis de natureza eléctrica dos equipamentos e aplicando os métodos propostos, se pode reduzir o consumo energético do compressor. Confrontando a informação recolhida no analisador de energia acoplado ao motor em funcionamento normal, com a informação obtida utilizando os diferentes métodos de poupança energética abordados, verifica-se o bom funcionamento do compressor, mas agora dependendo menos energia para determinada tarefa.

Outra conclusão que se pode tirar é que, ao automatizar o processo para manter a pressão do compressor a um *setpoint* desejado, pode-se reduzir os consumos de energia sem danificar o equipamento, evitando picos de arranque desnecessários.

O trabalho desenvolvido satisfaz o objectivo inicial desde o estudo e integração de sistemas até à redução dos consumos de energia.

5.2 Trabalho futuro

O sistema proposto nesta dissertação encontra-se em estado funcional para o efeito pretendido, embora se possa melhorar a nível da aplicação *VB* e *Movicon* para se tornar mais rápido e perfeito.

Numa perspectiva de trabalho futuro pode-se aplicar este sistema num ambiente em escala maior, isto é, com mais equipamentos de modo a tornar mais significativas as reduções de consumo e por consequência, a redução de custo de energia. O facto de a implementação ter utilizado sistemas de comunicação padrão e *software* compatível com diferentes plataformas e linguagens de programação, torna-o prático no que diz respeito ao acréscimo de outros módulos de integração. Outros equipamentos são facilmente integráveis neste sistema para a gestão de informação através de bases de dados.

Em ambiente industrial, existe a necessidade de conjugação do consumo de energia com a linha de produção e com os tarifários aplicáveis através do estudo dos dados no histórico de informação recolhida na base de dados.

Nos métodos de estudo mencionados no capítulo anterior, explorou-se uma variação de frequência de rotação do motor entre um intervalo de valores fixos (máximo e mínimo), deixando em aberto a possibilidade de utilização de outros algoritmos para o seu funcionamento.

Bibliografia

- [1] “*Sistema integrado de gestão empresarial*”.
<http://pt.wikipedia.org/wiki/Sistema_integrado_de_gest%C3%A3o_empresarial> Data de acesso: 02/06/2014
- [2] “*TCP/IP*”. <<http://pt.wikipedia.org/wiki/TCP/IP>>
Data de acesso: 24/05/2014
- [3] “*Hypertext Transfer Protocol*”.
<http://pt.wikipedia.org/wiki/Hypertext_Transfer_Protocol> Data de acesso: 24/05/2014
- [4] Quintã, André Figueiredo – *Integração de Sistemas de Produção* – Aveiro: Universidade de Aveiro, 2008. Tese de Mestrado em Engenharia Mecânica apresentada à Universidade de Aveiro.
- [5] Apontamentos de Informática Industrial; “*Protocolo de comunicação Modbus*” – Aveiro: Departamento de Engenharia Mecânica, 2011.
- [6] Apontamentos de Informática Industrial; “*Protocolo de comunicação série EIA 485*” – Aveiro: Departamento de Engenharia Mecânica, 2011.
- [7] Figueiredo, Luís Carlos da Costa – *Serviços Web – Supervisão e Controlo de Sistemas de Produção*. Aveiro: Universidade de Aveiro, 2012. Tese de Mestrado em Engenharia Mecânica apresentada à Universidade de Aveiro.
- [8] Aires, Afonso Costa – *Sistema de Gestão da Produção para Ambientes Industriais* – Aveiro: Universidade de Aveiro, 2008. Tese de Mestrado em Engenharia Electrónica e Telecomunicações apresentada à Universidade de Aveiro.

- [9] Rodrigo B. Souza, Adelardo A. D. Medeiros, João M. A. Nascimento, André L. Maitelli, Heitor P. Gomes; *“SISAL – Um Sistema Supervisório para Elevação Artificial de Petróleo”*; Rio Oil & Gás Expo and Conference, 2006.
- [10] João C. A. Bonifácio, Letícia D. Siqueira, João M. M. Freitas; *“Sistema Supervisório de Aquisição de Dados em um Protótipo de Estação de Tratamento de Efluente”*; Minas Gerais: Centro Universitário do Sul de Minas – UNIS-MG, 2011; Trabalho de Ciências da Computação apresentado à UNIS-MG.
- [11] Luiz A. Guedes, Alessandro J. De Souza, Clauber Bezerra, Rafael Feijó, Bruno Costa, Emília Eidelwein, Dannilo Cunha; *“Integração da WEB 2.0 no Desenvolvimento de Aplicações em Ambiente Industrial”*; Instituto Brasileiro de Petróleo, Gás e Biocombustíveis – IBP, 2009.
- [12] *“Utilities Control”*; Inglaterra, Wales e Escócia: Institution of Engineering and Technology – IET, 2007; Control & Automation.
- [13] Harald Manum, Marco Schmid; *“SUBSEA MONITORING - Monitoring in a harsh environment”*; Inglaterra, Wales e Escócia: Institution of Engineering and Technology – IET, 2007; Control & Automation.
- [14] E. Altamiranda, E. Colina; *“Subsea Control”*; Inglaterra, Wales e Escócia: Institution of Engineering and Technology – IET, 2007; Control & Automation.
- [15] Andrea Mercurio, Alessandro Di Giorgio, and Pierfrancesco Cioci; *“Open-Source Implementation of Monitoring and Controlling Services for EMS/SCADA Systems by Means of Web Services – IEC 61850 and IEC 61970 Standards”*; IEEE Transactions on Power Delivery, Vol. 24, Nº 3, Julho 2009.

- [16] Punnuluk Phaithoonbuathong; Robert Harrison; Andrew West; Radmehr Monfared; Thomas Kirkham; *“Web services-based automation for the control and monitoring of production systems”*; International Journal of Computer Integrated Manufacturing, 2010.
- [17] Guedes, Jorge Ricardo Teixeira Dias Couto - *“Planeamento e controlo de produção na Euronete, S.A.”* - Porto: Faculdade de Engenharia da Universidade do Porto, 2011. Tese de Mestrado em Engenharia Industrial e Gestão apresentada à Universidade do Porto.
- [18] *“Siemens Simatic WinCC”*.
<<http://www.ebah.com.br/content/ABAAAfZ9gAF/wincc-apostila>> Data de acesso: 24/05/2014
- [19] *“WinCC”*. <<http://en.wikipedia.org/wiki/WinCC>> Data de acesso: 11/04/2014
- [20] *“S7-WinCC supplement BST blocks”*. <<http://alexsentcha.wordpress.com/s7-wincc-supplement-bst-blocks/>> Data de acesso: 11/04/2014
- [21] *National Instruments – “Lookout Web-Enabled HMI/SCADA Software”*. – *Measurement and Automation Software*.
- [22] *“Software provides insight into solvent recovery operation”*.
<<http://www.amcec.com/case5.html>> Data de acesso: 11/04/2014
- [23] *“The solution that stands out among Scada/HMI for excellent flexibility, power, openness and simplicity”*. <<http://www.progea.com/en-us/products/scadahmimovicon11.aspx>> Data de acesso: 11/04/2014
- [24] <<http://www.dago.it/immagini/domotica/bahouse>> Data de acesso: 11/04/2014

- [25] “Web Client Technology All Movicon™ projects can be run on Win64, Win32 and WinCE using powerful Web Client solutions”. <<http://www.progea.com/en-us/products/scadahmimovicon11/webclient.aspx>> Data de acesso: 11/04/2014
- [26] “Oracle vs. MySQL vs. SQL Server: A Comparison of Popular RDBMS”. <<https://www.udemy.com/blog/oracle-vs-mysql-vs-sql-server/>> Data de acesso: 11/04/2014
- [27] Ward, Keith; “Microsoft Announces Visual Studio 2013 Pricing” – Visual Studio Magazine, 2013.
- [28] National Instruments; “NI Lookout Development/Run-Time Server” – [National Instruments](#), 2014
- [29] “Automation Sales”. <<http://www.automationsales.com.au/category.aspx?catID=42>> Data de acesso: 11/04/2014
- [30] Moxa Inc. – “NPort 5600 Series User’s Manual” - 14ª Edição, Fevereiro 2014.
- [31] Gavazzi, Carlo; “Innovation for Energy Management” - Lainate Italy: Carlo Gavazzi Automation Components, 2011.
- [32] Gavazzi, Carlo; “WM22-DIN: three-phase power analyzer” - Lainate Italy: Carlo Gavazzi Automation Components.
- [33] “Controlador lógico programável”. <http://pt.wikipedia.org/wiki/Controlador_l%C3%B3gico_program%C3%A1vel> Data de acesso: 27/05/2014

- [34] Apontamentos de Informática Industrial; “*Apêndice PLC-FATEK*” – Aveiro: Departamento de Engenharia Mecânica, 2011.
- [35] “*Motor monofásico*”. <http://pt.wikipedia.org/wiki/Motor_monof%C3%A1sico>
Data de acesso: 03/05/2014
- [36] Innovative Sensor Technology; “*Flow Sens FS5/FS5A – Thermal Mass Flow Sensor for all-purpose use in Gases*” – Switzerland: IST AG
- [37] Gavazzi, Carlo; “*Serial communication Protocol: WM22-DIN/EM4-DIN V1 R0*” – Lainate Italy: Carlo Gavazzi Controls, 2003.
- [38] Qbs Software; “*Visual Studio 2012 by Microsoft*” - QBS Software Ltd, 2014.
- [39] “*Oracle Database*”. <http://en.wikipedia.org/wiki/Oracle_Database> Data de acesso: 04/06/2014
- [40] “*What’s ORACLE_UNQNAME?*”
<http://blog.mclaughlinsoftware.com/2012/08/23/whats-oracle_unqname/>
Data de acesso: 04/06/2014
- [41] “*Oracle SQL Developer 4.0*”.<<http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>> Data de acesso: 04/06/2014
- [42] Oracle Corporation; “*Oracle Enterprise Manager Concepts Guide Release 9.2.0*” – Oracle, 2002
- [43] “*Rapid Electronics*”.
<<http://www.rapidonline.com/catalogueimages/Module/M178100P01WL.jpg>> Data de acesso: 04/06/2014