

Universidade de Évora



Plataforma interactiva para edifícios inteligentes com controlo preditivo

Dissertação para obtenção do Grau de Mestre em Engenharia
Mecatrónica

Elaborada por: Eduardo Manuel Ferreira Henriques

Orientada por: Professor Doutor João Manuel Gouveia Figueiredo

Outubro de 2009

Esta dissertação não inclui as críticas e sugestões feitas pelo júri.

Universidade de Évora

Plataforma interactiva para edifícios inteligentes com controlo preditivo

**Dissertação para obtenção do Grau de Mestre em Engenharia
Mecatrónica**

Elaborada por: Eduardo Manuel Ferreira Henriques

Orientada por: Professor Doutor João Manuel Gouveia Figueiredo



Outubro de 2009

Esta dissertação não inclui as críticas e sugestões feitas pelo júri.

Resumo

Esta dissertação desenvolve uma plataforma de controlo interactiva para edifícios inteligentes através de um sistema SCADA (Supervisory Control And Data Acquisition). Este sistema SCADA integra diferentes tipos de informações provenientes das várias tecnologias presentes em edifícios modernos (controlo da ventilação, temperatura, iluminação, etc.).

A estratégia de controlo desenvolvida implementa um controlador em cascada hierárquica onde os “loops” interiores são executados pelos PLC's locais (Programmable Logic Controller), e o “loop” exterior é gerido pelo sistema SCADA centralizado, que interage com a rede local de PLC's.

Nesta dissertação é implementado um controlador preditivo na plataforma SCADA centralizada. São apresentados testes efectuados para o controlo da temperatura e luminosidade de salas com uma grande área. O controlador preditivo desenvolvido tenta otimizar a satisfação dos utilizadores, com base nas preferências introduzidas em várias interfaces distribuídas, sujeito às restrições de minimização do desperdício de energia.

De forma a executar o controlador preditivo na plataforma SCADA foi desenvolvido um canal de comunicação para permitir a comunicação entre a aplicação SCADA e a aplicação MATLAB, onde o controlador preditivo é executado.

Palavras-chave: edifícios inteligentes, SCADA, protocolo OPC, controlo preditivo, constrangimentos, problemas de optimização.

Interactive platform for intelligent buildings with predictive control

Abstract

This dissertation develops an operational control platform for intelligent buildings using a SCADA system (Supervisory Control And Data Acquisition). This SCADA system integrates different types of information coming from the several technologies present in modern buildings (control of ventilation, temperature, illumination, etc.).

The developed control strategy implements a hierarchical cascade controller where inner loops are performed by local PLCs (Programmable Logic Controller), and the outer loop is managed by the centralized SCADA system, which interacts with the entire local PLC network.

In this dissertation a Predictive Controller is implemented at the centralized SCADA platform. Tests applied to the control of temperature and luminosity in huge-area rooms are presented. The developed Predictive Controller tries to optimize the satisfaction of user explicit preferences coming from several distributed user-interfaces, subjected to the constraints of energy waste minimization.

In order to run the Predictive Controller at the SCADA platform a communication channel was developed to allow communication between the SCADA application and the MATLAB application where the Predictive Controller runs.

Keywords: intelligent buildings, SCADA, OPC protocol, predictive control, constrains, optimization problem.

Agradecimentos

Ao professor João Figueiredo, meu orientador, pelas sugestões apresentadas e pelo apoio prestado na execução deste trabalho.

Aos meus colegas de mestrado, dos quais não realçarei nenhum nome em particular uma vez que todos eles, de uma forma directa ou indirecta, tiveram importância para que este trabalho fosse levado até ao fim com afinco e boa disposição.

Aos meus pais, por todos os sacrifícios que fizeram ao longo do tempo que estive em Évora e que me possibilitou não só a execução deste trabalho mas de toda a minha formação no ensino superior. Também ao meu irmão pelas nossas “guerras” que me permitiam ao fim-de-semana descomprimir do ambiente universitário.

Índice geral

Índice de figuras	VI
Índice de tabelas.....	IX
Lista de siglas	X
Capítulo 1 – Introdução	1
1.1 Conceitos chave.....	1
1.2 Estrutura da dissertação	5
Capítulo 2 – Edifício Inteligente.....	6
2.1 Apresentação do edifício	6
2.2 Equipamento de controlo local.....	7
2.3 Outros Equipamentos.....	8
Capítulo 3 – Modelação.....	9
3.1 Introdução	9
3.2 Modelo matemático	9
3.2.1 Temperatura	9
3.2.2 Luminosidade	10
3.3 Modelação em espaço de estados.....	11
Capítulo 4 – Controlo preditivo	12
4.1 Introdução	12
4.2 Função de custo.....	14
4.3 Pesos	15
4.4 Restrições	16
4.5 Predição.....	16
4.6 Problema não restringido.....	19
4.7 Problema QP	22
4.7.1 Formulação.....	22
4.7.2 Relaxamento das restrições.....	24
Capítulo 5 – Implementação.....	26
5.1 Introdução	26
5.2 Protocolo OPC.....	26
5.2.1 “Station configuration editor”	27

5.2.3 “OPC Scout”	28
5.3 Controlo local (PLC)	32
5.3.1 Principais características.....	32
5.3.2 Configuração dos PLC's.....	32
5.3.3 Programação dos PLC's.....	40
5.2.4 Configuração das interfaces HMI.....	41
5.4 Controlo remoto e supervisão (SCADA)	54
5.4.1 Principais características.....	54
5.4.2 Configuração do sistema SCADA.....	55
5.5 Matlab	71
5.5.1 Principais características.....	71
5.5.2 Configuração do controlo preditivo (Matlab).....	72
Capítulo 6 – Resultados	76
6.1 Introdução	76
6.2 Simulação Matlab.....	76
6.3 Implementação experimental condicionada	80
Capítulo 7 – Conclusões e trabalhos futuros	85
7.1 Conclusões	85
7.2 Trabalhos futuros	86
Referências bibliográficas	87
Anexo A – OPC Scout	89
Anexo B – OB1 conference room	91
Anexo C – Símbolos conference room	102
Anexo D – OB1 unidade fornecimento.....	105
Anexo E – Símbolos unidade fornecimento	107
Anexo F – ProTool conference room	109
Anexo G - WinCC	124
Anexo H - NetPro.....	129
Anexo I - Matlab.....	133

Índice de figuras

Figura 1: Sistema SCADA como integrador de tecnologias	2
Figura 2: Vista de perfil de uma porção do vidro considerado	6
Figura 3: Ligação entre a rede de autómatos e o sistema SCADA.....	7
Figura 4: Controlo preditivo - ideia fundamental.....	14
Figura 5: Station Configuration Editor	28
Figura 6: Conexão do servidor	28
Figura 7: Criação do grupo	29
Figura 8: Adicionar itens.....	29
Figura 9: Adicionar tags à comunicação.....	30
Figura 10: Adicionar tags ao grupo criado.....	31
Figura 11: Resultado final do processo "OPC Scout".....	31
Figura 12: Criação de projecto	33
Figura 13: Inserir autómato.....	34
Figura 14: Janela de configuração do hardware	34
Figura 15: Criação da rack.....	35
Figura 16: Aspecto da rack.....	35
Figura 17: Adição do módulo na rack do autómato	37
Figura 18: Configuração da rede Ethernet	38
Figura 19: Criação de conexão entre PC Station e autómato.....	38
Figura 20: Informação S7 connection_1	39
Figura 21: Informação S7 connection_2.....	39
Figura 22: Inserção de uma interface HMI ao projecto	41
Figura 23: Escolha da interface.....	41
Figura 24: Designação do autómato e tipo de comunicação	42
Figura 25: Parâmetros de ligação.....	42
Figura 26: Menu inicial do novo projecto	43
Figura 27: Criação de uma "tag"	43
Figura 28: Configuração das "tags"	44
Figura 29: Criação de um ecrã.....	45
Figura 30: Introdução de um campo de texto	45
Figura 31: Configuração do botão de estado "Entrar"	46
Figura 32: Configuração do botão de estado "Configuracao sistema"	46
Figura 33: Configuração do botão de estado "Preferências utilizador"	47
Figura 34: Configuração do botão de estado "Sair"	47
Figura 35: Configuração do campo de saída "Temp_actual"	48
Figura 36: Configuração do campo de saída "Lum_actual"	48
Figura 37: Configuração da área de mensagens	49
Figura 38: Configuração do botão de estado "Voltar"	49

Figura 39: Configuração do campo de entrada "Ref_Temp".....	50
Figura 40: Configuração do campo de entrada "Ref_Lum"	50
Figura 41: Criação de "Event Messages"	51
Figura 42: Configuração da "Event Message" criada	51
Figura 43: Texto da "Event Message"	52
Figura 44: Ecrã "Início" (interface HMI).....	52
Figura 45: Ecrã "Informação" (interface HMI)	53
Figura 46: Ecrã "Preferências" (interface HMI).....	53
Figura 47: Selecção do tipo de projecto	55
Figura 48: Criação do projecto	56
Figura 49: Adição do "channel"	56
Figura 50: Adição de uma "Driver Connection"	57
Figura 51: Adicionar tags (Passo 1)	57
Figura 52: Adicionar tags (Passo 2)	58
Figura 53: Adicionar tags (Passo 3)	58
Figura 54: Tags adicionadas no WinCC	59
Figura 55: Configuração da janela de figuras	60
Figura 56: Configuração do botão "Menu inicial"	60
Figura 57: Configuração do botão "Input Conf. room".....	61
Figura 58: Configuração do botão "Output Uni. fornecimento"	61
Figura 59: Configuração do botão "Condições actuais"	62
Figura 60: Configuração do botão "ligar sistema" (alteração de tag).....	62
Figura 61: Configuração do botão "ligar sistema" (alteração da cor).....	63
Figura 62: Configuração do botão "desligar sistema" (alteração de tag)	63
Figura 63: Configuração do botão "desligar sistema" (alteração da cor)	64
Figura 64: Configuração de um campo de saída	64
Figura 65: Configuração do campo de saída "Luz artificial"	65
Figura 66: Configuração do campo de saída "Abertura dos estores"	65
Figura 67: Configuração do campo de saída "Temperatura AVAC"	66
Figura 68: Configuração do campo de saída "Nº utilizadores"	66
Figura 69: Configuração do campo de saída "Temperatura actual"	67
Figura 70: Configuração do campo de saída "Med_Ref_Temp"	67
Figura 71: Configuração do campo de saída "Luminosidade actual"	68
Figura 72: Configuração do campo de saída "Med_Ref_Lum"	68
Figura 73: Ecrã "Menu inicial" (SCADA).....	69
Figura 74: Ecrã "Input conference room" (SCADA)	69
Figura 75: Ecrã "Output Uni. fornecimento" (SCADA).....	70
Figura 76: Ecrã "Condições actuais" (SCADA)	70
Figura 77: Toolbox MPC.....	72
Figura 78: Estrutura geral do controlador	73
Figura 79: Horizontes do controlador	73

Figura 80: Constrangimentos do controlador.....	74
Figura 81: Pesos do controlador.....	74
Figura 82: Bloco "OPC Read".....	75
Figura 83: Bloco "OPC Write".....	75
Figura 84: Temperatura e referência conference room (simulação Matlab).....	77
Figura 85: Luminosidade e referência conference room (simulação Matlab).....	77
Figura 86: Fluxo de calor gerado pelo AVAC (simulação Matlab).....	78
Figura 87: Percentagem de luz natural (simulação Matlab).....	78
Figura 88: Percentagem de luz artificial (simulação Matlab).....	79
Figura 89: Temperatura de funcionamento do AVAC (simulação Matlab).....	79
Figura 90: Temperatura e referência conference room (implementação).....	82
Figura 91: Luminosidade e referência conference room (implementação).....	82
Figura 92: Fluxo de calor gerado pelo AVAC (implementação).....	83
Figura 93: Percentagem de luz natural (implementação).....	83
Figura 94: Percentagem de luz artificial (implementação).....	84
Figura 95: Temperatura de funcionamento do AVAC (implementação).....	84

Índice de tabelas

Tabela 1: Dimensão de matrizes e vectores.....	21
Tabela 2: Módulos do autómato "conference room".....	36
Tabela 3: Módulos do autómato "unidade fornecimento"	36
Tabela 4: Módulos da "PC Station"	36
Tabela 5: Endereço Ethernet dos autómatos do sistema	37
Tabela 6: Referências introduzidas no minuto zero.....	80
Tabela 7: Referências introduzidas no minuto cinco	80
Tabela 8: Referências alteradas no minuto quinze.....	81
Tabela 9: Referências alteradas no minuto vinte e cinco	81
Tabela 10: Referências alteradas no minuto quarenta.....	81
Tabela 11: Referências alteradas no minuto quarenta e cinco.....	81
Tabela 12: Referências alteradas no minuto cinquenta.....	81

Lista de siglas

AVAC	Aquecimento, Ventilação e Ar Condicionado
DMC	Dinamic Matrix Control
EPSAC	Extended Prediction Self-Adaptive Control
GPC	Generalized Predictive Control
HMI	Human-Machine Interface
MAC	Model Algorithmic Control
MIMO	Multiple-input, multiple-output
MPC	Model-Based Predictive Control
PFC	Predictive Functional Control
PID	Proporcional integrador derivador
PLC	Programmable Logic Controller
QDMC	Quadratic Dynamic Matrix Control
SCADA	Supervisory Control And Data Acquisition
SISO	Single-input, single-output
SOLO	Sequential Open Loop Optimization
QP	Quadratic Programming
STL	Statement List
OPC	Object Linking and Embedding (OLE) for Process Control
COM	Component Object Model
DCOM	Distributed Component Object Model
OLE	Object Linking and Embedding

Capítulo 1 – Introdução

1.1 Conceitos chave

Em meados dos anos 70 surgiu o conceito de *sistemas de controlo distribuído*. Este conceito era aplicado exclusivamente no controlo de processos industriais tais como refinação de petróleo, produção de cimento, geração de energia em centrais eléctricas, fabrico de papel, entre outros.

Com o desenvolvimento das tecnologias associadas à automação e comunicações este conceito expandiu-se para outras áreas: fabricação de produtos, gestão de energia, agricultura, automação de edifícios, etc. [Figueiredo and Sá Costa, 2007], [Christensen, 2002], [Figueiredo and Sá Costa, 2008a, 2008b], [Silva et al., 2007]. Os grandes melhoramentos dos dispositivos electrónicos e das comunicações à distância permitiram a miniaturização de componentes, com a correspondente redução de custos. Estas vantagens são os principais factores para que seja cada vez mais real o conceito de *edifício inteligente*, que tem assumido uma enorme relevância ao longo das últimas décadas.

Desta forma surgiu nos anos 80 os primeiros *edifícios inteligentes*, nos quais se pretendia controlar a iluminação, as condições climáticas, a segurança e a interligação destes três elementos.

No entanto, hoje em dia, é possível controlar um maior número de elementos num sistema automatizado aplicado a um edifício. Estas tecnologias permitem que seja cada vez mais simples satisfazer as suas necessidades das pessoas que utilizam o edifício, tanto ao nível do conforto como de comodidade. Tarefas como regar um jardim, aspirar a casa, controlar a luminosidade ou a temperatura de toda a casa, tornam-se algo completamente banal e nada aborrecidas. As preocupações com a segurança também foram evoluindo e é possível actualmente ter sistemas de detecção de fugas de gás, inundações e incêndio, que ligados ao sistema fornecem a informação necessária para evitar uma catástrofe. É igualmente possível através de uma rede de comunicação (Internet, telemóvel) controlar câmaras de segurança que em tempo real mostram o que se passa na habitação. Ainda ao nível da segurança o controlo do acesso ao edifício pode ser feito por dispositivos de leitura de impressões digitais, da retina ou mesmo do padrão de voz de uma pessoa. Além do acesso ao edifício este tipo de identificação pessoal pode permitir a personalização do edifício mediante as preferências da pessoa.

Um factor que surgiu recentemente e promoveu uma aceleração no desenvolvimento da automação de edifícios está relacionado com as questões energéticas. A elevada frequência de crises energéticas e a correspondente pressão

sobre os preços da energia motivaram o desenvolvimento de sistemas inteligentes de gestão de energia. [Figueiredo and Sá da Costa, 2008a], [Khan et al., 2007]

As melhorias no sistema de comunicações têm estimulado a implementação de sistemas distribuídos. Estes sistemas são então distribuídos normalmente gerido por uma plataforma centralizada de supervisão, conhecida como um sistema SCADA (Supervisory Control And Data Acquisition) [SIEMENS, 2008].

Uma plataforma interactiva para *edifícios inteligentes* é composta por três níveis interligados: o *nível interactivo*, no qual os habitantes podem comunicar ao edifício as suas preferências em relação a determinadas variáveis controláveis; o *nível operacional*, onde os equipamentos são controlados; e o nível de controlo elevado, *overall building well-being model*, que gere todo o edifício, otimizando as preferências dos habitantes, limitado pelos recursos disponíveis.

O nível de controlo elevado, *overall building well-being model*, tem de resolver conflitos entre os dados fornecidos pelos utilizadores e deve ter a capacidade para se adaptar ao comportamento e mudanças de preferências dos habitantes. A *Multi-Agent System theory – MAS* – [Silva et al., 2005] pode ser usada nesta abordagem, onde cada interface do habitante é modelada como um agente singular. No futuro, quando existirem centenas destas interfaces distribuídas ao longo de edifícios de grandes dimensões (hotéis, hospitais, edifícios empresariais, etc.), estas passam a fazer parte de um problema de optimização distribuída. Na verdade, o conjunto de meios disponíveis no edifício tem de ser distribuído de uma forma óptima, em tempo real, de forma a atingir as expectativas do utilizador, mas ao mesmo tempo minimizando o consumo de energia (abastecimento de água, energia eléctrica, fluxo de ar, etc.).

Nesta dissertação é apresentado e testado o *nível operacional*, através de uma plataforma de interactiva de controlo para edifícios inteligentes que foi desenvolvida usando um sistema SCADA. Este sistema SCADA integra diferentes tipos de informação provenientes de várias tecnologias presentes em edifícios modernos – sistemas de controlo de ventilação e temperatura, redes de computadores, sistemas de controlo de iluminação, etc.) – como se pode observar na figura 1.

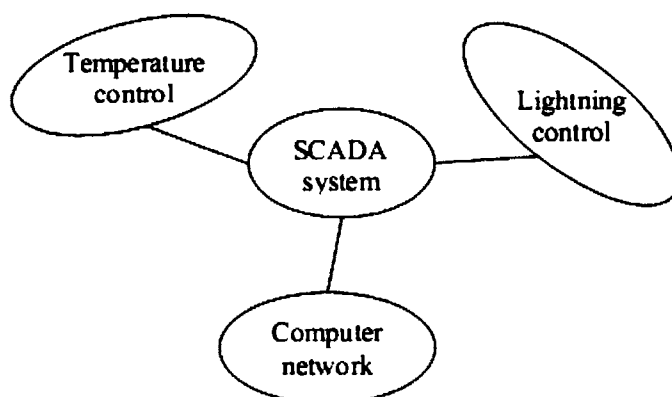


Figura 1: Sistema SCADA como integrador de tecnologias

A estratégia de controlo desenvolvida implementa um controlador em cascada hierárquica onde os “loops” interiores são executados pelos PLC’s locais (Programmable Logic Controller), e o “loop” exterior é gerido pelo sistema SCADA centralizado, que interage com a rede local de PLC’s. Uma primeira abordagem a esta estratégia pode ser encontrada em [Figueiredo and Sá da Costa, 2008b].

Na presente dissertação um controlador preditivo é implementado na plataforma centralizada SCADA. Os resultados do controlo da temperatura e luminosidade em salas com uma grande área são apresentados. O controlador preditivo desenvolvido tenta otimizar a satisfação dos utilizadores, com base nas preferências introduzidas em várias interfaces distribuídas, sujeito às restrições de minimização do desperdício de energia.

O SCADA é um sistema computadorizado para supervisão e controlo de processos. Embora seja considerado um sistema de controlo, não possui capacidade de efectuar cálculo numérico tendo de recorrer a um segundo programa, de computação matemática. O programa de computação matemática efectua os cálculos necessários ao controlo e comunica os resultados ao SCADA. No que respeita à função de supervisão o sistema é extremamente eficaz, o que proporciona uma melhoria na qualidade dos processos supervisionados, uma redução dos custos operacionais e um melhor desempenho dos processos. Todas as características anteriormente referidas demonstram como a utilização de um sistema SCADA na gestão de um edifício inteligente é uma decisão acertada quando se pretende supervisionar e controlar o referido edifício.

Como foi referido anteriormente o sistema SCADA não tem capacidade para efectuar cálculo numérico, sendo necessário recorrer a um programa de computação matemática. Uma das opções existentes no mercado e que foi a escolhida para esta tese é o programa Matlab [MathWorks, 2008].

Um PLC é um computador especializado, baseado num microprocessador que desempenha funções de controlo de diversos tipos e níveis de complexidade. Utiliza uma memória programável para armazenar internamente instruções e para implementar funções específicas, como por exemplo funções lógicas, temporização, contagem e aritmética. Estas funções possibilitam o controlo de sistemas e processos através de módulos de entradas e saídas existentes no PLC. O facto de um PLC poder comunicar com outros PLC’s, com sistemas SCADA, com interfaces HMI e ainda actuar sobre uma vasta gama de dispositivos tornam este dispositivo ideal para ser implementado no *nível operacional* de uma plataforma interactiva para edifícios inteligentes.

No *nível interactivo* da plataforma é comum encontrar interfaces HMI, estas podem ser de vários tipos. No caso de um edifício inteligente o tipo de HMI mais usado é o painel táctil, que permite ao utilizador de uma forma fácil introduzir as suas preferências em relação às variáveis que estão disponíveis para ser controladas e

permite ao mesmo tempo visualizar a reacção do controlo às preferências que o utilizador definiu previamente.

A presente dissertação tem os seguintes objectivos:

- Desenvolver uma aplicação com base em autómatos Siemens S7-300 para estabelecer uma plataforma interactiva para edifícios inteligentes.
- Desenvolver uma aplicação SCADA para monitorização e controlo remoto do sistema automático desenvolvido. A aplicação deverá ser acessível por internet (remote desktop).
- Desenvolver uma aplicação em Matlab para implementação de um controlador preditivo para gestão do edifício inteligente, que comunicará com o software SCADA.

Esta dissertação foi objecto de um artigo científico (Henriques and Figueiredo, 2009) submetido a aprovação à conferência "IFAC Intl. Conference on Control Methodologies and Technology for Energy Efficiency, March 29-31, 2010, Vilamoura, Portugal".

1.2 Estrutura da dissertação

A presente dissertação está dividida em sete capítulos.

O primeiro capítulo, no qual nos encontramos, proporciona uma introdução ao tema sobre o qual a dissertação se debruça. É feita uma pequena descrição da evolução temporal do conceito de edifício inteligente, assim como de algumas das tecnologias que serão focadas no desenvolvimento da dissertação.

No segundo capítulo faz-se uma apresentação do edifício inteligente estudado e para o qual foi desenvolvido o controlo preditivo. São analisados os equipamentos de controlo local e os equipamentos a controlar.

No terceiro capítulo aborda-se a modelação do edifício tendo em conta as variáveis a controlar. Um modelo matemático é escolhido e é estudada a sua utilização através da sua forma em espaço de estados.

O quarto capítulo contém a informação necessária ao dimensionamento e projecto de controlador preditivo.

No quinto capítulo é abordada a temática da implementação do sistema de controlo e supervisão do edifício. É focada a elaboração do controlo local, o sistema SCADA, o desenvolvimento do controlador no Matlab e também a ligação entre todos os componentes do sistema.

No sexto capítulo são apresentados os resultados obtidos das simulações efectuadas ao controlador desenvolvido e da posterior implementação.

No sétimo e último capítulo são apresentadas as conclusões e uma lista de possíveis trabalhos futuros.

Capítulo 2 – Edifício Inteligente

Os pontos seguintes têm por objectivo apresentar ao leitor o edifício inteligente estudado de forma a facilitar a compreensão do mesmo. Proceder-se-á à apresentação dos sensores e actuadores presentes no edifício. Serão apresentadas informações relevantes para a modelação e projecto do controlador nos capítulos seguintes.

2.1 Apresentação do edifício

Na presente dissertação é considerada a existência de um edifício inteligente, que pode ser um qualquer edifício de uma empresa. No entanto a aplicação a desenvolver será particularizada para uma conference room e uma unidade de fornecimento (AVAC e energia eléctrica). A conference room terá a capacidade de interagir com os utilizadores através de 10 interfaces distribuídas existentes na sala. Os utilizadores fornecerão as suas preferências no que diz respeito à temperatura e luminosidade da sala em que se encontram.

Para a modelação do sistema foi considerado que a conference room possui as seguintes dimensões:

- 12 metros de comprimento;
- 16 metros de largura;
- 2,5 metros de altura.

Foi ainda definido que apenas uma das paredes de 16 metros estará em contacto com o exterior e será constituída por um vidro duplo, cujas dimensões correspondem à totalidade da parede (16m x 2,5m). O vidro é constituído por três camadas: a primeira de vidro com 4 milímetros de espessura, a segunda é uma caixa-de-ar com 10 milímetros de espessura e a terceira novamente vidro com 8 milímetros de espessura.

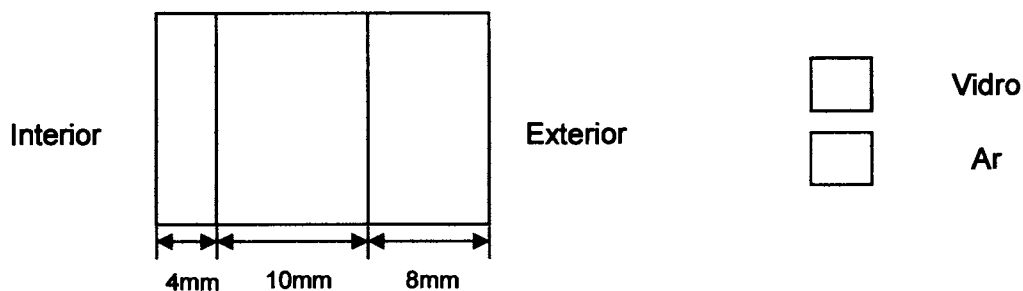


Figura 2: Vista de perfil de uma porção do vidro considerado

2.2 Equipamento de controlo local

O equipamento de controlo local engloba dois autómatos Siemens S7-300 e 10 interfaces HMI.

Será instalado um autómato na conference room, sendo que o segundo fará o controlo da unidade de fornecimento. Os autómatos comunicarão entre si e com o sistema SCADA [SIEMENS, 2008] por Ethernet [SIEMENS, 2003], como está ilustrado na figura 3. A ligação por Ethernet possibilita que todo o sistema seja acedido por um qualquer software de controlo remoto de computadores (Remote desktop software).

No mesmo computador onde se encontra o sistema SCADA é implementado o controlador preditivo criado no software Matlab [MathWorks, 2008] que comunica com o sistema SCADA através do protocolo OPC.

As interfaces HMI serão 10 painéis tácteis, instalados na conference room e comunicando através de Profibus-DP [SIEMENS, 2001] com o autómato respectivo.

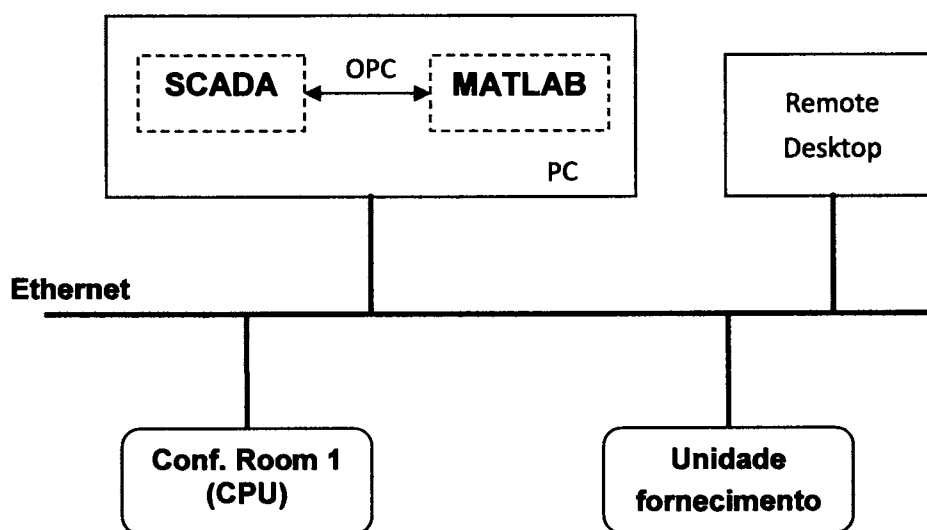


Figura 3: Ligação entre a rede de autómatos e o sistema SCADA

O conceito apresentado, para a automação de edifícios, é um passo rumo ao futuro que permitirá que vários utilizadores interajam com a unidade de controlo do edifício, que possui um controlador inteligente capaz:

- a) de satisfazer em tempo real as preferências dos utilizadores;
- b) de garantir a minimização do consumo de energia do edifício.

2.3 Outros Equipamentos

Para além dos equipamentos anteriormente referidos, a plataforma idealizada é composta por outros equipamentos. Tais como:

- Um sistema AVAC;
- Sensores de temperatura e luminosidade (interior e exterior);
- Um motor de passo para abertura e fecho dos estores;
- Um dimmer para variação da luminosidade da iluminação artificial.

Capítulo 3 – Modelação

3.1 Introdução

De seguida estudar-se-á um modelo que permita o controlo da temperatura e luminosidade de uma conference room. O estudo do modelo a implementar representa um passo importante na fase de projecto do controlador, devendo este modelo ser o mais exacto possível.

Um controlador é desenvolvido em várias etapas, sendo uma delas a simulação do sistema a controlar. A simulação toma uma importância ainda maior quando, por uma qualquer razão, não seja possível realizar testes no sistema real. Revela-se óbvio que o modelo de simulação deve ser o mais semelhante possível ao sistema real, de forma a facultar resultados precisos.

3.2 Modelo matemático

Uma vez que se pretende controlar duas variáveis, temperatura e luminosidade, serão considerados numa primeira fase os modelos de controlo em separado.

3.2.1 Temperatura

No que se refere à temperatura será considerado um modelo com base num sistema térmico com uma fonte de calor, uma capacitância térmica e uma resistência térmica [Close et al., 2002]. O modelo considerado tem a seguinte equação:

$$T_{int}^{(t+1)} = \frac{1}{C} \left[q^{(t)} - \frac{1}{R} (T_{int}^{(t)} - T_{ext}^{(t)}) \right]$$

Equação 1

Onde:

- T_{int} é a temperatura da sala (°C);
- C é a capacitância térmica da sala, depende da geometria da sala e da pressão atmosférica (J/°C);
- R é a resistência térmica da sala, depende das propriedades da parede (°C/W);
- T_{ext} é a temperatura exterior (°C);
- q é um fluxo de calor, neste sistema proveniente do AVAC (W).

Das variáveis anteriormente apresentadas existem duas que têm de ser calculadas através de fórmulas próprias antes de serem aplicadas no sistema. As duas variáveis são a capacitância e a resistência térmica da sala. São de seguida apresentadas as fórmulas de cálculo das variáveis referidas.

$$C = C_p * m$$

Equação 2

$$m = \rho * V$$

Equação 3

Nas fórmulas apresentadas anteriormente temos as seguintes variáveis:

- C_p – calor específico do ar (J/Kg*°C);
- ρ – densidade do ar (Kg/m³);
- V – volume da sala (m³).

$$R = \frac{d}{A * k}$$

Equação 4

Onde:

- d é a espessura da parede (m);
- A é a área transversal da parede (m²);
- k é a condutividade térmica do material constituinte da parede (W/m*°C).

3.2.2 Luminosidade

Para a luminosidade será considerado um modelo que relaciona a iluminação artificial instalada na sala com a percentagem de abertura dos estores que permitem a entrada de luz natural proveniente da rua. O modelo considerado tem a seguinte equação:

$$L_{int}^{(t+1)} = E^{(t)} + I^{(t)}$$

Equação 5

Sendo:

- E é a abertura dos estores da sala (%);
- I é a iluminação artificial instalada na sala (%).

3.3 Modelação em espaço de estados

A representação em espaço de estados é o modelo matemático de um sistema físico composto por entradas, saídas e variáveis de estado que são relacionadas por equações diferenciais de primeira ordem. De forma a determinar o número de entradas, saídas e estados, as variáveis são expressas na forma de vectores e as equações algébricas e diferenciais são escritas na forma de matrizes [Ogata, 2001].

$$x^{(k+1)} = A * x^{(k)} + B * u^{(k)}$$

$$y^{(k)} = C * x^{(k)}$$

Equação 6

Sendo:

- $x^{(k)}$ – vector de estado;
- $y^{(k)}$ – vector de saída;
- $u^{(k)}$ – vector de entrada (ou controlo);
- A – matriz de estado;
- B – matriz de entrada;
- C – matriz de saída;
- k – tempo de amostragem.

Tendo sido obtidas na secção anterior as formulações de forma independente da temperatura e luminosidade, nesta secção proceder-se-á à passagem dessas formulações para uma representação em espaço de estados.

Sendo o sistema do tipo MIMO (multiple-input, multiple-output) a representação em espaço de estados proporciona uma formulação compacta, de integração simples no Matlab e capaz de produzir resultados com elevada precisão.

Em baixo encontra-se o modelo do sistema de controlo em espaço de estados desenvolvido para a realização desta dissertação e os valores das constantes R_1 e C_1 .

$$\begin{Bmatrix} T_{int1}^{(t+1)} \\ L_{int1}^{(t+1)} \end{Bmatrix} = \begin{bmatrix} -\frac{1}{R_1 * C_1} & 0 \\ 0 & 0 \end{bmatrix} * \begin{Bmatrix} T_{int1}^{(t)} \\ L_{int1}^{(t)} \end{Bmatrix} + \begin{bmatrix} -\frac{1}{C_1} & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} * \begin{Bmatrix} q_1^{(t)} \\ E_1^{(t)} \\ I_1^{(t)} \\ T_{ext1}^{(t)} \end{Bmatrix}$$

$$\begin{Bmatrix} T_{int1}^{(t)} \\ L_{int1}^{(t)} \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{Bmatrix} T_{int1}^{(t)} \\ L_{int1}^{(t)} \end{Bmatrix}$$

Equação 7

$$R_1 = 0.01 \text{ } ^\circ\text{C/W}$$

$$C_1 = 581292 \text{ J/}^\circ\text{C}$$

Capítulo 4 – Controlo preditivo

Neste capítulo é apresentado o controlo preditivo desenvolvido para a realização desta dissertação. A escolha deste tipo de controlo baseou-se no facto de ser a única técnica de controlo, mais avançada que o controlo PID, a ter impacto significativo no controlo de processos industriais.

4.1 Introdução

Não existe a certeza de quando se deu o aparecimento do controlo preditivo. As primeiras formulações surgem na indústria, em 1978, quando Richalet propõem um modelo que dá pelo nome de “Model Predictive Heuristic Control” e mais tarde, em 1980, através de Cutler e Ramaker que propõem outro modelo chamado “Dynamic Matrix Control” (DMC). Depois do aparecimento destas formulações, e acima de tudo, após a aceitação deste tipo de controlo pela comunidade académica foram surgindo outras formulações, tais como:

- Extended Prediction Self-Adaptive Control (EPSAC);
- Generalized Predictive Control (GPC);
- Model Algorithmic Control (MAC);
- Predictive Functional Control (PFC);
- Quadratic Dynamic Matrix Control (QDMC);
- Sequential Open Loop Optimization (SOLO).

Embora estas formulações tenham sido desenvolvidas de forma independente, por diferentes pessoas em épocas diferentes, todas elas partilham os pontos que são apresentados de seguida:

- Existência de um modelo interno para predição, no qual as saídas em instantes futuros determinados previamente são calculados até ao horizonte de predição.
- Execução da minimização, normalmente restringida, de uma função de custo. Uma sequência de acções de controlo resulta do problema de optimização.
- Conceito de horizonte recidivo. O processo de optimização, em cada instante de amostragem, é reiniciado com a informação actualizada do sistema.

Os principais conceitos envolvidos no controlo preditivo e que são ao mesmo tempo as principais razões para o sucesso do mesmo são:

- A capacidade para lidar com problemas de controlo multivariável de forma natural.
- O facto de lidar com as limitações dos actuadores.
- O modelo interno tem a capacidade de simular o comportamento do sistema mais rapidamente que o tempo real.
- A definição de uma trajectória de referência que define o comportamento desejado num ciclo de controlo fechado.
- A caracterização da trajectória futura é efectuada com base num número finito de movimentos.
- Possibilidade de operar perto dos constrangimentos, o que produz frequentemente uma operação mais rentável (comparando com as técnicas de controlo convencionais).
- Os tempos de actualização do controlo são relativamente baixos, de modo que existe bastante tempo para a necessária computação online.
- Existência de optimização online, possivelmente limitada, para determinação da estratégia de controlo futura.

Hoje em dia a formulação mais comum na industria dá pelo nome de “Model-Based Predictive Control”, MPC [Maciejowski, 2002]. A justificação para o nome prende-se com a utilização de um modelo interno capaz de prever o comportamento do sistema.

De uma forma simplificada, a estratégia de controlo preditivo compara os estados previstos do sistema com os objectivos pretendidos e ajusta os actuadores de modo a se atingirem os objectivos de seguimento ou regulação pretendidos obedecendo às restrições impostas, de uma forma óptima. As restrições podem dever-se aos limites físicos do sistema, a imposições de segurança ou a limites de qualidade, entre outras hipóteses.

O controlador preditivo lida bem tanto com sistemas SISO, como com sistemas MIMO. Junta-se ainda o facto que uma representação em espaço de estados potencia o carácter multivariável deste tipo de controlo. Além disso, uma das grandes vantagens advém da possibilidade de explorar melhores soluções se forem conhecidas de antemão as referências, assim como as perturbações medidas.

Aplicações do controlador MPC a outros sistemas para além dos problemas de controlo de processos foram surgindo ao longo das últimas duas décadas. Em [Kolokotsa et al., 2009] o controlador MPC foi aplicado na gestão da qualidade do ambiente interior de edifícios.

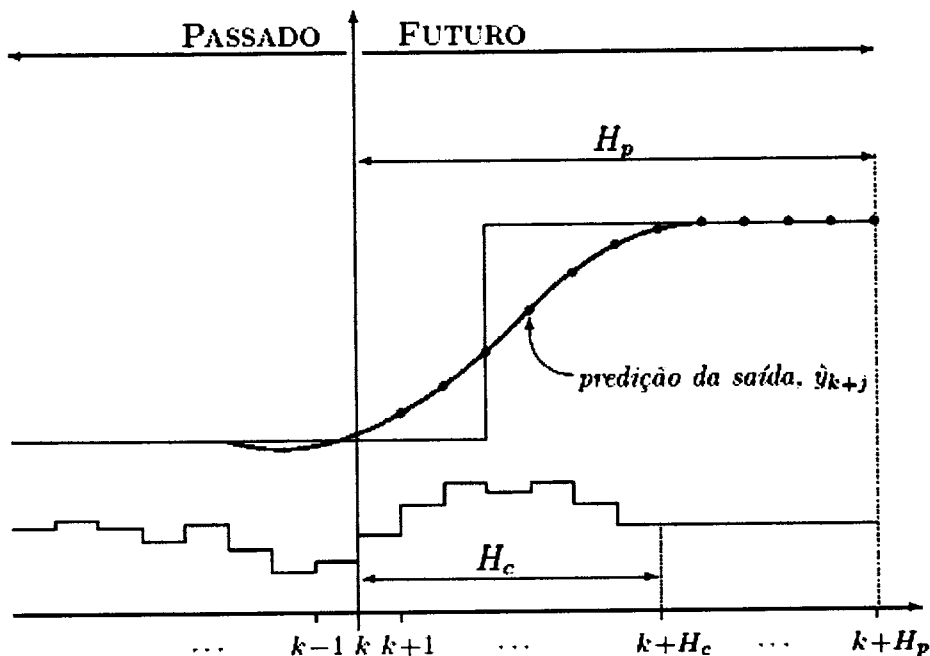


Figura 4: Controlo preditivo - ideia fundamental

Na figura anterior [Silva P., 2007] temos:

- k – tempo presente;
- H_c – horizonte de controlo;
- H_p – horizonte de predição.
- Normalmente $H_c \leq H_p$ e $H_p \geq 1$.
- Como se pode ver pela figura $\Delta u_c^{k+1} = 0$, $i \in [H_c; H_p-1]$.

Assume-se $H_c \leq H_p$ porque o peso computacional envolvido no cálculo das acções óptimas a aplicar ao sistema é bastante elevado e também para garantir a estabilidade da solução. Além disso, prever saídas futuras é bastante mais rápido do que determinar por optimização as acções de controlo e um horizonte de predição elevado possibilita antever com maior antecedência quando o sistema se encontra perto de atingir uma restrição. A predição das saídas é feita com base no modelo interno do controlador preditivo.

4.2 Função de custo

A formulação de um controlo preditivo envolve uma função de custo, geralmente quadrática. A optimização das acções de controlo passa pela minimização da função de custo e o facto de a função ser quadrática simplifica bastante o problema da optimização, porque torna-o convexo para modelos de predição lineares.

Uma função de custo V penaliza desvios das saídas $\hat{y}^{(k+i|k)}$ em relação aos valores da trajectória de referência $r^{(k+i|k)}$, desvios nas acções de controlo $\hat{u}_c^{(k|k)}$ em relação ao valor ideal $r_{u_c}^{(k)}$, e ainda as variações das acções de controlo $\Delta\hat{u}^{(k+i|k)}$.

$$V = \begin{pmatrix} \hat{y}^{(k+H_w|k)} \\ \vdots \\ \hat{y}^{(k+H_p|k)} \end{pmatrix} - \begin{pmatrix} r^{(k+H_w)} \\ \vdots \\ r^{(k+H_p)} \end{pmatrix} \cdot Q \cdot \begin{pmatrix} \hat{y}^{(k+H_w|k)} \\ \vdots \\ \hat{y}^{(k+H_p|k)} \end{pmatrix} - \begin{pmatrix} r^{(k+H_w)} \\ \vdots \\ r^{(k+H_p)} \end{pmatrix} \\ + \begin{pmatrix} \hat{u}_c^{(k|k)} \\ \vdots \\ \hat{u}_c^{(k+H_p-1|k)} \end{pmatrix} - \begin{pmatrix} r_{u_c}^{(k)} \\ \vdots \\ r_{u_c}^{(k+H_p-1)} \end{pmatrix} \cdot W \cdot \begin{pmatrix} \hat{u}_c^{(k|k)} \\ \vdots \\ \hat{u}_c^{(k+H_p-1|k)} \end{pmatrix} - \begin{pmatrix} r_{u_c}^{(k)} \\ \vdots \\ r_{u_c}^{(k+H_p-1)} \end{pmatrix} \\ + \begin{pmatrix} \Delta\hat{u}^{(k|k)} \\ \vdots \\ \Delta\hat{u}^{(k+H_u-1|k)} \end{pmatrix} \cdot R \cdot \begin{pmatrix} \Delta\hat{u}^{(k|k)} \\ \vdots \\ \Delta\hat{u}^{(k+H_u-1|k)} \end{pmatrix}$$

Equação 8

4.3 Pesos

$Q^{(j)}$, $W^{(j)}$ e $R^{(j)}$ presentes na equação anterior são matrizes de peso que devem influenciar o resultado da optimização. Na forma matricial as matrizes tem a seguinte forma:

$$\begin{bmatrix} Q^{H_w} & 0 & \dots & 0 \\ 0 & Q^{H_w+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q^{H_p} \end{bmatrix}$$

Equação 9

$$\begin{bmatrix} W^0 & 0 & \dots & 0 \\ 0 & W^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & W^{H_p-1} \end{bmatrix}$$

Equação 10

$$\begin{bmatrix} R^0 & 0 & \dots & 0 \\ 0 & R^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R^{H_u-1} \end{bmatrix}$$

Equação 11

Sendo:

H_w = Parâmetro de “janela”;

H_p = Horizonte de predição;

H_u = Horizonte de controlo.

Estas matrizes normalmente são diagonais, não sendo no entanto obrigatória verificar-se essa condição.

As matrizes $Q^{(j)}$, $W^{(j)}$ e $R^{(j)}$ deverão ser positivas, definidas de modo a garantir que o problema de optimização se mantenha convexo. No caso de se verificar a condição referida no parágrafo anterior, $Q^{(j)}$, $W^{(j)}$ e $R^{(j)}$ serem matrizes diagonais, basta que todos os elementos da diagonal não sejam negativos.

Desta forma é perceptível que os pesos condicionam a resposta do sistema e que utilizando valores diferentes nas matrizes de pesos $Q^{(j)}$, $W^{(j)}$ e $R^{(j)}$ podem-se obter objectivos distintos. É possível através da variação dos valores nas matrizes atribuir maior importância a determinada variável, em detrimento de outras. No caso do controlo da luminosidade da conference room, através de uma determinada configuração das matrizes é possível privilegiar a abertura dos estores (luz natural) em detrimento da utilização da luz artificial (lâmpadas).

4.4 Restrições

O controlo preditivo lida de forma óptima com a existência de restrições, como foi referido no início do capítulo. As restrições ao sistema podem ser efectuadas na amplitude dos actuadores ou nas suas variações, bem como na amplitude de determinadas saídas. Alguns exemplos de restrições:

- Amplitude dos actuadores – luz artificial (lâmpadas acesas a 100% de luminosidade);
- Variação da amplitude dos actuadores – variação da luz natural (velocidade do motor que abre ou fecha os estores);
- Amplitude das saídas – temperatura da conference room (limite mínimo e máximo permitido ao sistema).

Por norma as restrições traduzem-se por inequações lineares. Na presente dissertação, com a utilização do Matlab não houve preocupação com este pormenor uma vez que o software resolve o problema automaticamente.

4.5 Predição

De forma a resolver um problema de controlo preditivo temos de obter uma maneira de calcular os valores previstos das variáveis controladas, $\hat{y}^{(k+i|k)}$, a partir da estimativa do estado presente, $\hat{x}^{(k|k)}$, de entradas futuras ou da última entrada $u^{(k-1)}$, e ainda das variações de entradas futuras, $\Delta \hat{u}^{(k+i|k)}$. Esta é realmente uma parte da

solução, mas não de todo o problema de formulação, que será solucionado nas próximas secções. No entanto revela como a forma que é feita a previsão influencia o desempenho do sistema em ciclo fechado do controlo preditivo.

Mas na verdade fazer uma predição pode ser bastante complicado. Portanto vamos assumir, para começar, a solução mais simples em que $\hat{x}^{(k|k)} = x^{(k)}$. Assim temos que:

$$\begin{aligned}\hat{x}^{(k+1|k)} &= A * x^{(k)} + B * \hat{u}^{(k|k)} \\ \hat{x}^{(k+2|k)} &= A * \hat{x}^{(k+1|k)} + B * \hat{u}^{(k+1|k)} \\ &= A^2 * x^{(k)} + A * B * \hat{u}^{(k|k)} + B * \hat{u}^{(k+1|k)} \\ &\vdots \\ \hat{x}^{(k+H_p|k)} &= A * \hat{x}^{(k+H_p-1|k)} + B * \hat{u}^{(k+H_p-1|k)} \\ &= A^{H_p} * x^{(k)} + A^{H_p-1} * B * \hat{u}^{(k|k)} + \dots + B * \hat{u}^{(k+H_p-1|k)}\end{aligned}$$

Equação 12

Na primeira linha usou-se $\hat{u}^{(k|k)}$ em vez de $u^{(k)}$ porque quando se proceder ao primeiro cálculo o valor $u^{(k)}$ ainda não se encontra disponível.

Agora, recorde-se que $\Delta u^{(k+i|k)} = 0; i \in [H_u; H_p - 1]$

As expressões seguintes correspondem às acções de controlo, \hat{u} , em função de $\Delta \hat{u}$:

$$\begin{aligned}\hat{u}^{(k|k)} &= \Delta \hat{u}^{(k|k)} + u^{(k-1)} \\ \hat{u}^{(k+1|k)} &= \Delta \hat{u}^{(k+1|k)} + \Delta \hat{u}^{(k|k)} + u^{(k-1)} \\ &\vdots \\ \hat{u}^{(k+H_u-1|k)} &= \Delta \hat{u}^{(k+H_u-1|k)} + \dots + \Delta \hat{u}^{(k|k)} + u^{(k-1)}\end{aligned}$$

Equação 13

Considerando as expressões da equação 12 e substituindo-as na equação 11 obtemos:

$$\begin{aligned}\hat{x}^{(k+1|k)} &= A * x^{(k)} + B * [\Delta \hat{u}^{(k|k)} + u^{(k-1)}] \\ \hat{x}^{(k+2|k)} &= A^2 * x^{(k)} + A * B * [\Delta \hat{u}^{(k|k)} + u^{(k-1)}] + B * [\Delta \hat{u}^{(k+1|k)} + \Delta \hat{u}^{(k|k)} + u^{(k-1)}] \\ &= A^2 * x^{(k)} + (A + I) * B * \Delta \hat{u}^{(k|k)} + B * \Delta \hat{u}^{(k+1|k)} + (A + I) * B * u^{(k-1)} \\ &\vdots \\ \hat{x}^{(k+H_u|k)} &= A^{H_u} * x^{(k)} + (A^{H_u-1} + \dots + A + I) * B * \Delta \hat{u}^{(k|k)} + \dots + B * \Delta \hat{u}^{(k+H_u-1|k)} + \\ &\quad + (A^{H_u-1} + \dots + A + I) * B * u^{(k-1)}\end{aligned}$$

A partir deste ponto temos:

$$\begin{aligned} \hat{x}^{(k+H_u+1|k)} &= A^{H_u+1} * x^{(k)} + (A^{H_u} + \dots + A + I) * B * \Delta\hat{u}^{(k|k)} + \dots + \\ &\quad + (A + I) * B * \Delta\hat{u}^{(k+H_u-1|k)} + (A^{H_u} + \dots + A + I) * B * u^{(k-1)} \\ &\quad \vdots \\ \hat{x}^{(k+H_p|k)} &= A^{H_p} * x^{(k)} + (A^{H_p-1} + \dots + A + I) * B * \Delta\hat{u}^{(k|k)} + \dots + \\ &\quad + (A^{H_p-H_u} + \dots + A + I) * B * \Delta\hat{u}^{(k+H_u-1|k)} + \\ &\quad + (A^{H_p-1} + \dots + A + I) * B * u^{(k-1)} \end{aligned}$$

Equação 14

Por fim escrevemos as equações anteriores na forma matricial:

$$\begin{aligned} \begin{bmatrix} \hat{x}^{(k+1|k)} \\ \vdots \\ \hat{x}^{(k+H_u|k)} \\ \hat{x}^{(k+H_u+1|k)} \\ \vdots \\ \hat{x}^{(k+H_p|k)} \end{bmatrix} &= \begin{bmatrix} A \\ \vdots \\ A^{H_u} \\ A^{H_u+1} \\ \vdots \\ A^{H_p} \end{bmatrix} * x^{(k)} + \begin{bmatrix} B \\ \vdots \\ \sum_{i=0}^{H_u-1} A^i * B \\ \sum_{i=0}^{H_u} A^i * B \\ \vdots \\ \sum_{i=0}^{H_p-1} A^i * B \end{bmatrix} * u^{(k-1)} + \\ &+ \begin{bmatrix} B & \dots & 0 \\ A * B + B & \dots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_u-1} A^i * B & \dots & B \\ \sum_{i=0}^{H_u} A^i * B & \dots & A * B + B \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_p-1} A^i * B & \dots & \sum_{i=0}^{H_p-H_u} A^i * B \end{bmatrix} * \begin{bmatrix} \Delta\hat{u}^{(k|k)} \\ \vdots \\ \Delta\hat{u}^{(k+H_u-1|k)} \end{bmatrix} \end{aligned}$$

Equação 15

As predições de y são agora obtidas através de:

$$\begin{aligned} \hat{y}^{(k+1|k)} &= C * \hat{x}^{(k+1|k)} \\ \hat{y}^{(k+2|k)} &= C * \hat{x}^{(k+2|k)} \\ &\quad \vdots \\ \hat{y}^{(k+H_p|k)} &= C * \hat{x}^{(k+H_p|k)} \end{aligned}$$

Ou na forma matricial:

$$\begin{bmatrix} \hat{y}^{(k+1|k)} \\ \hat{y}^{(k+2|k)} \\ \vdots \\ \hat{y}^{(k+H_p|k)} \end{bmatrix} = \begin{bmatrix} C^1 & 0 & \dots & 0 \\ 0 & C^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & C^{H_p} \end{bmatrix} * \begin{bmatrix} \hat{x}^{(k+1|k)} \\ \hat{x}^{(k+2|k)} \\ \vdots \\ \hat{x}^{(k+H_p|k)} \end{bmatrix}$$

Equação 16

Manipulando a equação 15, através da substituição do vector $\hat{x}^{(k)}$ pela equação 14, obtém-se a relação entre $\hat{y}^{(k)}$ e $\Delta\hat{u}^{(k)}$:

$$\hat{y}^{(k)} = \Psi * x^{(k)} + Y * u^{(k-1)} + \theta * \Delta\hat{u}^{(k)}$$

Equação 17

4.6 Problema não restringido

Tendo sido encontrada na secção anterior a relação entre $\hat{y}^{(k)}$ e $\Delta\hat{u}^{(k)}$ (equação 16), é agora possível reescrever a função de custo (equação 8) de outra forma:

$$V^{(k)} = \|y^{(k)} - \mathcal{J}^{(k)}\|_Q^2 + \|\Delta u^{(k)}\|_R^2$$

Equação 18

Onde:

$$y^{(k)} = \begin{bmatrix} \hat{y}^{(k+H_w|k)} \\ \vdots \\ \hat{y}^{(k+H_p|k)} \end{bmatrix} \quad \mathcal{J}^{(k)} = \begin{bmatrix} \hat{f}^{(k+H_w|k)} \\ \vdots \\ \hat{f}^{(k+H_p|k)} \end{bmatrix} \quad \Delta u^{(k)} = \begin{bmatrix} \Delta\hat{u}^{(k|k)} \\ \vdots \\ \Delta\hat{u}^{(k+H_u-1|k)} \end{bmatrix}$$

Equação 19

$$\begin{bmatrix} Q^{H_w} & 0 & \dots & 0 \\ 0 & Q^{H_w+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q^{H_p} \end{bmatrix} \quad \begin{bmatrix} R^0 & 0 & \dots & 0 \\ 0 & R^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R^{H_u-1} \end{bmatrix}$$

Equação 20

Defina-se agora uma nova variável, $\varepsilon^{(k)}$, que corresponde ao erro de seguimento:

$$\varepsilon^{(k)} = \mathcal{J}^{(k)} - \Psi * x^{(k)} + Y * u^{(k-1)}$$

Equação 21

O erro de seguimento é a diferença entre o objectivo da trajectória futura e a 'resposta livre' do sistema, ou seja, a resposta que deve ocorrer sobre o horizonte de predição se não forem efectuadas alterações nas entradas.

Agora é possível voltar a escrever a função de custo de uma nova forma:

$$\begin{aligned} V^{(k)} &= \|\theta * \Delta U^{(k)} - \varepsilon^{(k)}\|_Q^2 + \|\Delta U^{(k)}\|_R^2 \\ &= [\Delta U^{(k)T} * \theta^T - \varepsilon^{(k)T}] * Q * [\theta * \Delta U^{(k)} - \varepsilon^{(k)}] + \Delta U^{(k)T} * R * \Delta U^{(k)} \\ &= \varepsilon^{(k)T} * Q * \varepsilon^{(k)} - 2 * \Delta U^{(k)T} * \theta^T * Q * \varepsilon^{(k)} + \Delta U^{(k)T} * [\theta^T * Q * \theta * R] * \Delta U^{(k)} \end{aligned}$$

Equação 22

A equação anterior pode ser escrita de uma forma simplificada:

$$V(k) = const. - \Delta U^{(k)T} * \mathcal{G} + \Delta U^{(k)T} * \mathcal{H} * \Delta U^{(k)}$$

Equação 23

Onde:

$$\mathcal{G} = 2 * \theta^T * Q * \varepsilon^{(k)} \text{ e } \mathcal{H} = \theta^T * Q * \theta + R$$

Equação 24

Observa-se que nem \mathcal{G} , nem \mathcal{H} dependem de $\Delta U^{(k)}$.

De forma a se achar o $\Delta U^{(k)}$ óptimo calcula-se o gradiente de $V(k)$ e iguala-se a zero.

$$\begin{aligned} \nabla_{\Delta U^{(k)}} V &= -\mathcal{G} + 2 * \mathcal{H} * \Delta U^{(k)} \\ \Delta U^{(k)}_{opt} &= \frac{1}{2} * \mathcal{H}^{-1} * \mathcal{G} \end{aligned}$$

Equação 25

É certo que $\Delta U^{(k)}_{opt}$ é um ponto estacionário, mas ainda não se está em condições de afirmar que se trata de um mínimo. Para se poder garantir que $\Delta U^{(k)}_{opt}$ fornece um mínimo da função de custo, diferencia-se o gradiente $\nabla_{\Delta U^{(k)}} V$ em relação a $\Delta U^{(k)}$

$$\frac{\delta^2 V}{\delta \Delta U^{(k)2}} = 2 * \mathcal{H} = 2 * (\theta^T * Q * \theta + R)$$

Equação 26

Foi assumido que $Q^{(i)} \geq 0$ para todos os i e isso assegura que $\theta^T * Q * \theta \geq 0$. Portanto se $R > 0$ então $\frac{\delta^2 V}{\delta \Delta U^{(k)2}}$ é seguramente positiva, o que garante que estamos na presença de um mínimo.

Caso se verifique $R \geq 0$, para o caso em que $R = 0$ necessitamos que $\theta^T * Q * \theta > 0$ para assegurar a existência de um mínimo.

De seguida é apresentada a informação relativa às dimensões das matrizes e vectores envolvidos na computação das entradas optimizadas (com l entradas, n estados e m saídas controladas).

Q	$m * (H_p - H_w + 1) \times m * (H_p - H_w + 1)$
R	$l * H_u \times l * H_u$
Ψ	$m * (H_p - H_w + 1) \times n$
Y	$m * (H_p - H_w + 1) \times l$
θ	$m * (H_p - H_w + 1) \times l * H_u$
ε	$m * (H_p - H_w + 1) \times 1$
\mathcal{G}	$l * H_u \times 1$
\mathcal{H}	$l * H_u \times l * H_u$

Tabela 1: Dimensão de matrizes e vectores

Como foi referido anteriormente, a solução de um problema não restringido pode ser obtida através de $\Delta \mathcal{U}^{(k)}_{opt} = \frac{1}{2} * \mathcal{H}^{-1} * \mathcal{G}$. No entanto a solução óptima deste tipo de problema nunca deve ser obtida calculando a inversa de \mathcal{H} , devido à dificuldade no cálculo do inverso de uma matriz.

A melhor forma de encontrar a solução óptima é resolver o problema como um problema de “mínimos quadrados”.

Desde que $Q \geq 0$ e $R \geq 0$, é possível encontrar as matrizes S_Q e S_R que são as suas “raízes quadradas”:

$$S_Q^T * S_Q = Q \quad S_R^T * S_R = R$$

Equação 27

Se Q e R são diagonais é trivial fazer esta operação, basta fazer a raiz quadrada de cada elemento da diagonal. Se as matrizes não forem quadradas podem ser obtidas usando o algoritmo de Cholesky.

Com base nestas matrizes é possível escrever a função de custo da seguinte forma:

$$V(k) = \left\| \left\| \begin{matrix} S_Q * \{\theta * \Delta \mathcal{U}^{(k)} - \varepsilon^{(k)}\} \\ S_R * \Delta \mathcal{U}^{(k)} \end{matrix} \right\| \right\|^2$$

Equação 28

A solução óptima pode ser obtida através da resolução de:

$$\begin{bmatrix} S_Q * \{\theta * \Delta U^{(k)} - \varepsilon^{(k)}\} \\ S_R * \Delta U^{(k)} \end{bmatrix} = 0 \Leftrightarrow$$

$$\Leftrightarrow \begin{bmatrix} S_Q * \theta \\ S_R \end{bmatrix} * \Delta U^{(k)} = \begin{bmatrix} S_Q * \varepsilon^{(k)} \\ 0 \end{bmatrix}$$

Equação 29

4.7 Problema QP

4.7.1 Formulação

No que se refere à temperatura será considerado um modelo com base num

Num problema *QP* (*Quadratic Programming*) as restrições já estão presentes. Consideremos as restrições presentes nas seguintes formas:

$$E * \begin{bmatrix} \Delta U^{(k)} \\ 1 \end{bmatrix} \leq 0; \quad F * \begin{bmatrix} U^{(k)} \\ 1 \end{bmatrix} \leq 0; \quad G * \begin{bmatrix} y^{(k)} \\ 1 \end{bmatrix} \leq 0$$

Equação 30

Em que $U^{(k)} = [\hat{u}^{(k|k)T}, \dots, \hat{u}^{(k+H_u-1|k)T}]^T$ e $\Delta U^{(k)}$ é definida de forma análoga.

Supondo que F apresenta a forma $F = [F_1, F_2, \dots, F_{H_u}, f]$. Sendo que cada F_i tem a dimensão $q \times m$, enquanto f tem a dimensão $q \times 1$ surge a seguinte expressão:

$$\sum_{i=1}^{H_u} F_i * \hat{u}^{(k+i-1|k)} + f \leq 0$$

Equação 31

Dado que $\hat{u}^{(k+i-1|k)} = u^{(k-1)} + \sum_{j=0}^{i-1} \Delta \hat{u}^{(k+j|k)}$ pode então escrever-se:

$$\sum_{j=1}^{H_u} F_j * \Delta \hat{u}^{(k|k)} + \sum_{j=2}^{H_u} F_j * \Delta \hat{u}^{(k+1|k)} + \dots + F_{H_u} * \Delta \hat{u}^{(k+H_u-1|k)} + \sum_{j=1}^{H_u} F_j * u^{(k-1)} + f \leq 0$$

Equação 32

Agora definindo $\mathcal{F} = \sum_{j=i}^{H_u} F_j$ e $\mathcal{F} = [\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_{H_u}]$, pode-se escrever:

$$\mathcal{F} * \Delta U^{(k)} \leq -\mathcal{F}_1 * u^{(k-1)} - f$$

Equação 33

Passando agora para as restrições nas saídas, temos:

$$G * \left[\begin{array}{c} \Psi * x^{(k)} + \Upsilon * u^{(k-1)} + \theta * \Delta U^{(k)} \\ 1 \end{array} \right] \leq 0$$

Equação 34

Definindo $G = [\Gamma, g]$, em que g é a última coluna de G , ficando a restrição como:

$$\begin{aligned} \Gamma * [\Psi * x^{(k)} + \Upsilon * u^{(k-1)}] + \Gamma * \theta * \Delta U^{(k)} + g &\leq 0 \Leftrightarrow \\ \Leftrightarrow \Gamma * \theta * \Delta U^{(k)} &\leq -\Gamma * [\Psi * x^{(k)} + \Upsilon * u^{(k-1)}] - g \end{aligned}$$

Equação 35

Resta reescrever as restrições nas variáveis das entradas, bastando para tal definir $E = [W, w]$. Deste modo as restrições podem ser escritas da seguinte forma:

$$W * \Delta U^{(k)} \leq w$$

Equação 36

Resumindo estas restrições numa forma matricial temos:

$$\left[\begin{array}{c} \mathcal{F} \\ \Gamma * \theta \\ W \end{array} \right] * \Delta U^{(k)} \leq \left[\begin{array}{c} -\mathcal{F}_1 * u^{(k-1)} - f \\ -\Gamma * [\Psi * x^{(k)} + \Upsilon * u^{(k-1)}] - g \\ w \end{array} \right]$$

Equação 37

A função de custo $V(k)$ a minimizar continua a ser a mesma do problema não restringido. É portanto necessário resolver o seguinte problema de optimização constringido:

$$\text{minimizar } \Delta U^{(k)T} * \mathcal{H} * \Delta U^{(k)} - \Delta U^{(k)T} * \mathcal{G}$$

Equação 38

A inequação das restrições apresenta agora a seguinte forma:

$$\min_{\theta} \frac{1}{2} * \theta^T * \Phi * \theta + \phi^T * \theta$$

Equação 39

sujeito a

$$\Omega * \theta \leq w$$

Equação 40

que é um problema de optimização conhecido como *Quadratic Programming problem* e para o qual são conhecidos algoritmos comuns que fornecem solução.

De forma similar à resolução do problema não restringido, é preferível passar o problema QP para um algoritmo de solução de “raízes quadradas”, como o seguinte:

$$\min_{\Delta U^{(k)}} \left\| \begin{bmatrix} S_Q * \{\theta * \Delta U^{(k)} - \varepsilon^{(k)}\} \\ S_R * \Delta U^{(k)} \end{bmatrix} \right\|^2$$

Equação 41

Desde que $\mathcal{H} \geq 0$, o problema QP a resolver torna-se convexo. Devido à sua convexidade podemos garantir a resolução do problema de optimização.

4.7.2 Relaxamento das restrições

Um problema grave que pode ocorrer com o controlo preditivo que foi formulado é o facto da optimização se deparar com um problema impossível de resolver. Tal situação pode acontecer devido a ocorrência de grandes perturbações, sendo que nessas alturas o sistema não se consegue manter dentro das restrições especificadas. É ainda possível que o modelo real seja diferente do modelo interno. Nesse caso o controlador preditivo pode confundir a diferença entre o modelo real e o modelo interno como sendo uma grande perturbação, e se esta diferença continuar a aumentar o controlador pode decidir que não tem condições de manter o sistema dentro dos constrangimentos. Existe muitas formas de um problema de controlo preditivo se tornar impossível de resolver e a maioria delas são difíceis de antecipar.

Devido a este facto é essencial que haja uma estratégia para lidar com a possibilidade da não resolução do problema. Os algoritmos convencionais para resolução de problemas QP apenas fornecem uma mensagem do estilo “problema impossível de resolver”, logicamente isto não é ideal para um controlo on-line.

Uma das estratégias para lidar com este tipo de problema passa pelo relaxamento das restrições. Isto é, ao invés de assumir as restrições como limites rígidos que nunca podem ser cruzados, passam a ser assumidas restrições que podem ser cruzadas ocasionalmente, mas somente se realmente necessário.

É importante fazer a distinção entre as restrições nas entradas e as restrições nas saídas. Normalmente as restrições nas entradas são realmente restrições rígidas e não existe forma de serem relaxadas. Exemplos que sustentam a afirmação anterior são válvulas, superfícies de controlo e outros actuadores com limites de acção ou na variação das acções. Uma vez atingido limite não existe forma de o exceder, excepto através da utilização de actuadores com maior margem de acção. Portanto as restrições nas entradas, geralmente, não são relaxadas.

Uma forma simples de obter o relaxamento das restrições nas saídas é a introdução de novas variáveis, chamadas “variáveis de relaxamento”, que são definidas de forma a serem diferentes de zero apenas se as restrições forem violadas. Estes valores diferentes de zero são severamente penalizados na função de custo, desta forma o otimizador tem um forte incentivo para manter as variáveis de relaxamento com o valor zero se possível.

Uma forma de efectuar o que foi referido no parágrafo anterior é adicionar uma penalização quadrática para violações de restrições. O problema de optimização que havia sido definido pelas equações 38 e 39 é modificado para a seguinte formulação:

$$\min_{\theta, \epsilon} \frac{1}{2} * \theta^T * \Phi * \theta + \phi^T * \theta + \rho * \|\epsilon\|^2$$

Equação 42

sujeito a

$$\begin{cases} \Omega * \theta \leq w + \epsilon \\ \epsilon \geq 0 \end{cases}$$

Equação 43

onde ϵ é um vector não negativo com a mesma dimensão de w , e ρ é um escalar também não negativo. Este ainda é um problema *QP*, embora com um número maior de variáveis.

Capítulo 5 – Implementação

5.1 Introdução

O sistema a implementar terá uma componente de controlo local que se baseia num determinado número de PLC's. Num patamar superior ao do controlo local encontramos o nível de controlo remoto e supervisão, no qual se encontra um sistema SCADA [SIEMENS, 2008]. Tanto os PLC's como o sistema SCADA permitem a implementação de determinados controladores, tais como os controladores P, PI, PD e PID. No entanto o controlador que foi desenvolvido no capítulo anterior é mais avançado, envolvendo a minimização de uma função de custo e outras operações matriciais que se revelam bastante elaboradas. Devido a estes condicionalismos não é possível implementar o controlador preditivo nos PLC's ou no sistema SCADA, sendo necessário recorrer a outras soluções que serão posteriormente interligadas com o sistema SCADA.

Neste capítulo é explicado o desenvolvimento do controlo local, é apresentado o sistema SCADA utilizado na monitorização e controlo remoto do sistema. É também exemplificado o desenvolvimento do controlador no Matlab [MathWorks, 2008] e a sua ligação ao sistema SCADA.

5.2 Protocolo OPC

OPC é um protocolo de conectividade aberta para automação industrial e sistemas industriais de empresas. A interoperabilidade do protocolo é assegurada pela criação e manutenção de especificações de padrões abertos. Existem actualmente sete especificações padrão concluídas ou em desenvolvimento.

Baseado em padrões fundamentais e tecnologia do mercado de computação em geral, para o protocolo OPC são adaptadas e criadas especificações que satisfazem as necessidades específicas da indústria.

A especificação OPC foi baseada nas tecnologias Microsoft *OLE* (Object Linking and Embedding), *COM* (Component Object Model) e *DCOM* (Distributed Component Object Model), a especificação definia um conjunto padrão de objectos, interfaces e métodos para uso em controlo de processos de forma a facilitar a interoperabilidade. As tecnologias COM/DCOM forneciam a base para o desenvolvimento dos softwares.

O primeiro padrão desenvolvido, originalmente chamado de *OPC specification* e hoje em dia conhecido como *Data Access specification*, resultou da colaboração entre um grupo constituído pelos principais fornecedores de automação de todo o mundo e a Microsoft [<http://www.opcfoundation.org/>]. O *OPC Data Access* é usado para a leitura e escrita de dados em tempo real. O protocolo OPC foi desenvolvido para fornecer a ponte entre as aplicações informáticas que corriam sobre o sistema operativo Windows e o hardware que efectuava o controlo de processos. Existem actualmente centenas de servidores e clientes OPC Data Access.

Os servidores OPC fornecem um método para que vários pacotes de softwares consigam aceder a dados provenientes de aparelhos de controlo de processos, como por exemplo PLC's. Normalmente, em qualquer altura que um software precisava de aceder aos dados fornecidos por um aparelho, uma interface tinha de ser escrita para tornar esta operação possível. O propósito do protocolo OPC é definir uma interface comum, que uma vez escrita pode ser reutilizada por qualquer empresa quer para implementação de um sistema SCADA, uma interface HMI ou qualquer outro software.

Quando um servidor OPC é configurado para um dispositivo em particular, ele pode ser reutilizado por qualquer aplicação que seja capaz de agir como um cliente OPC.

É de realçar que o protocolo OPC é uma especificação publicada. Por conseguinte, nenhuma empresa "possui" o protocolo OPC e qualquer um pode desenvolver um servidor OPC, independentemente de serem ou não um membro da Fundação OPC. Além disso, qualquer pessoa pode integrar produtos OPC, e não há nenhum pré-requisito que obrigue o integrador de pertencer a qualquer organização. Assim, cabe a cada empresa que necessita de produtos OPC garantir que os seus produtos são certificados e que os integradores de seu sistema possuem a formação necessária.

Exemplos de alguns trabalhos desenvolvidos sobre a aplicação do protocolo de comunicação OPC a diversas áreas podem ser encontrados em [Sahin and Bolat, 2009], [Santos et al., 2005], [Wang et al., 2007].

5.2.1 “Station configuration editor”

Para que o computador seja reconhecido pelo **Siemens Simatic STEP 7** como uma “Simatic PC Station” é necessário configurar um software chamado **Station configuration editor**. Neste software é adicionada a configuração que mais à frente é também adicionada ao pelo **Siemens Simatic STEP 7** e que consiste num “OPC Server” e num “IE General”, como se pode observar na figura seguinte.

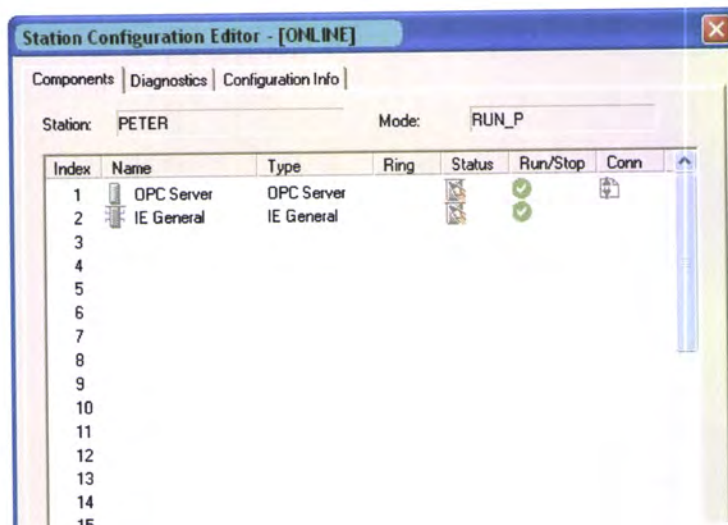


Figura 5: Station Configuration Editor

5.2.3 “OPC Scout”

Através do software “OPC Scout” [SIEMENS, 2007] são configuradas as “tags” de modo a que estas estejam disponíveis para todos os componentes da plataforma criada. Desta forma qualquer dos componentes tem possibilidade de ler ou escrever valores nas “tags” configuradas. Embora esta secção surja neste local, devido à sua componente OPC, a configuração apresentada de seguida apenas pode ser feita depois da configuração dos PLC’s.

1. Primeiro é necessário conectar-se ao servidor. Na presente dissertação o servidor usado foi o “OPC.SimaticNET”.

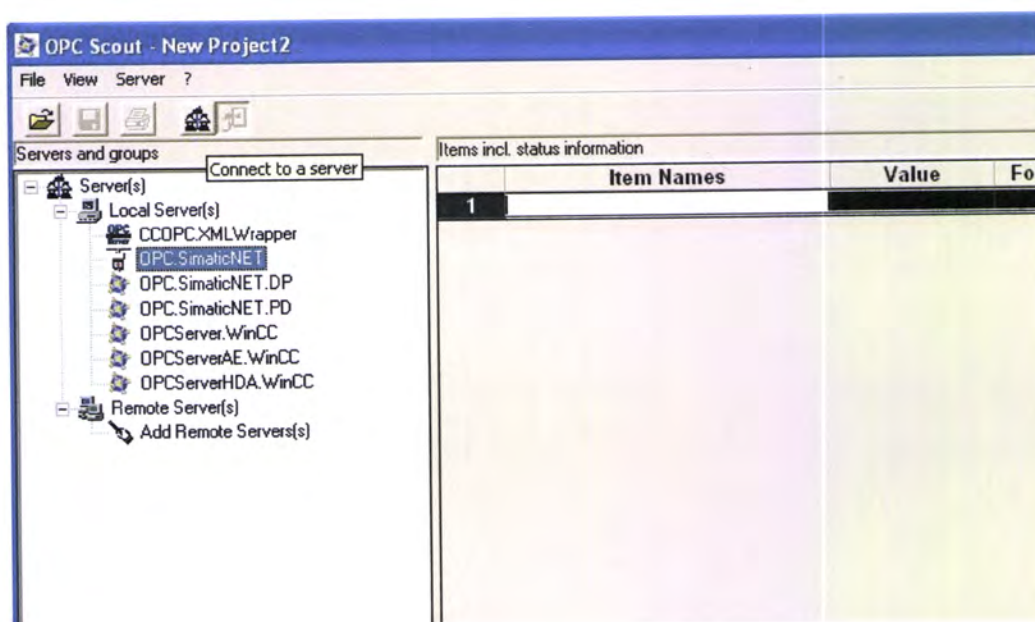


Figura 6: Conexão do servidor

2. De seguida o software pede para que se crie um grupo.

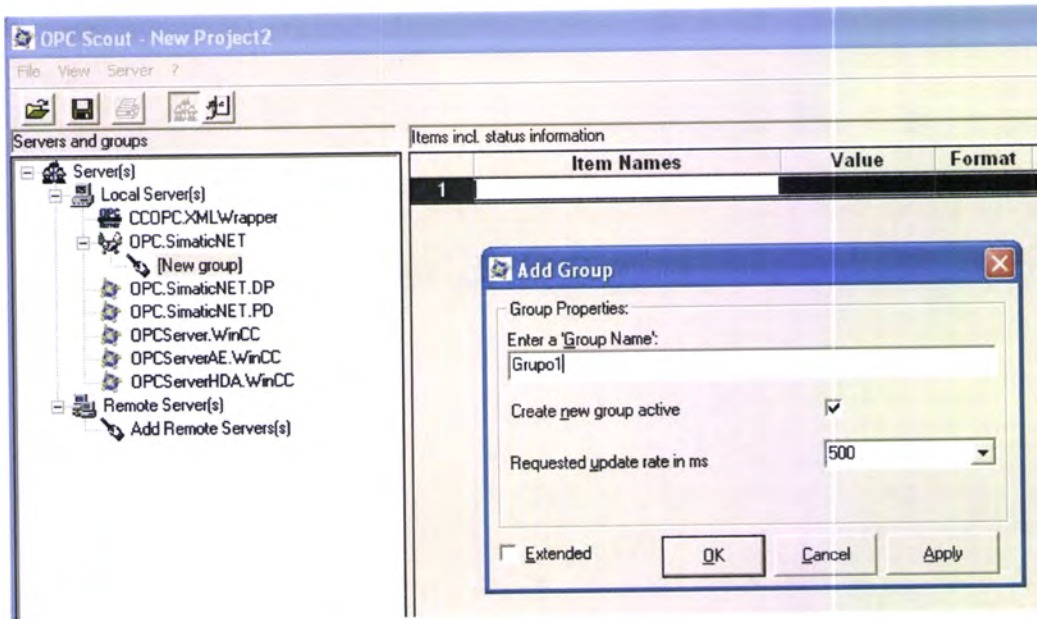


Figura 7: Criação do grupo

3. Após a criação do grupo faz-se duplo clique com o rato no nome do grupo criado e surge uma nova janela. No painel branco do lado direito da janela clica-se com o botão direito do rato e selecciona-se “Add item”.

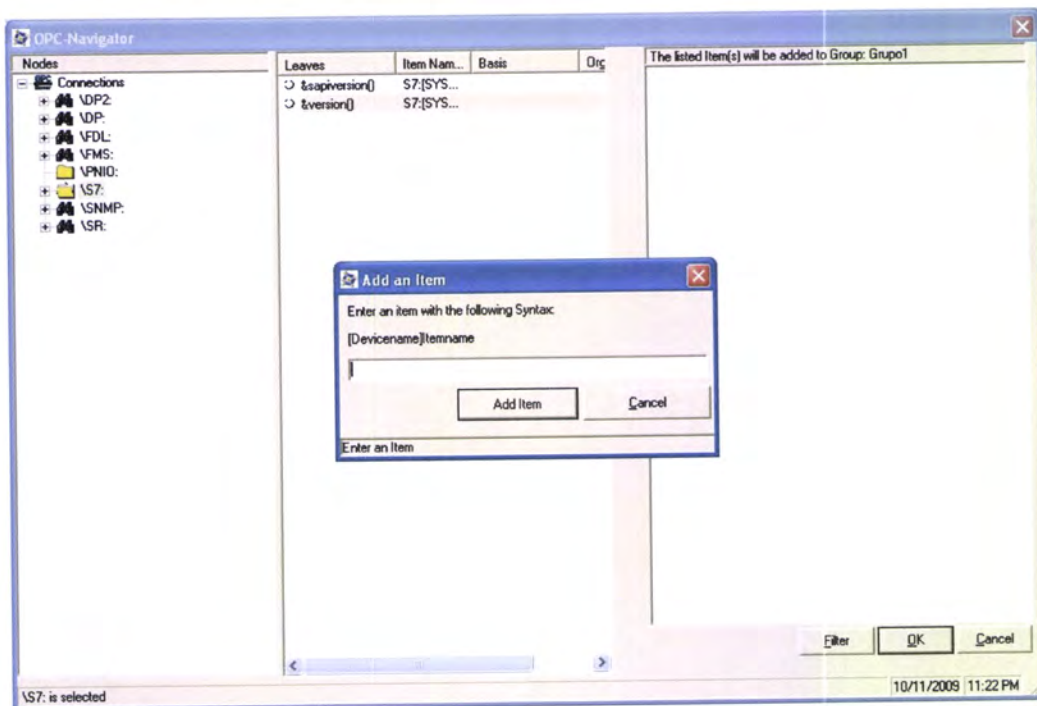


Figura 8: Adicionar itens

Na janela anterior é necessário introduzir os seguintes itens:

- S7:[S7 connection_1|OPC_Server|S7ONLINE|10.11,192.168.0.4,03.02,1]MBO
- S7:[S7 connection_2|OPC_Server|S7ONLINE|11.11,192.168.0.1,03.02,1]MBO

A informação necessária para a introdução destes itens é consultada aquando da criação das comunicações entre os autómatos e a PC Station, através do software **NetPro**, incorporado no software **Siemens Simatic STEP 7**. Estes itens permitem que na “árvore” existente do lado esquerdo da janela sejam criadas as comunicações entre os autómatos e a PC Station.

4. Expandindo a “árvore” e chegando à comunicação criada é agora necessário adicionar as “tags” que foram criadas nessa comunicação. Define-se o tipo de “tag” e introduz-se o endereço da mesma.

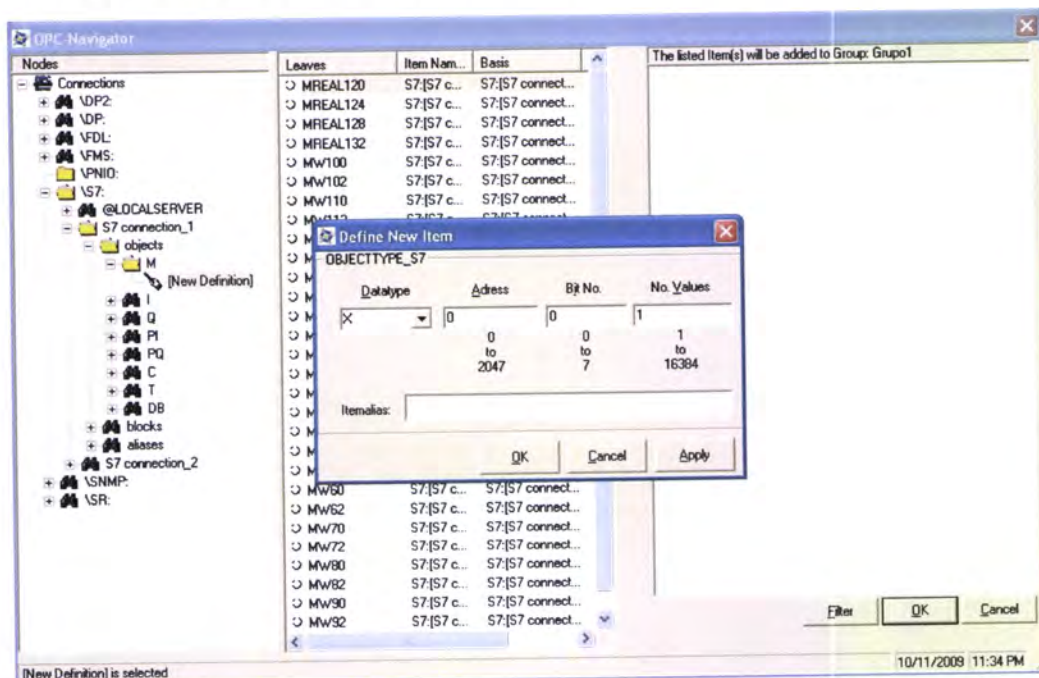


Figura 9: Adicionar tags à comunicação

5. Para finalizar o processo é necessário adicionar as tags ao grupo.

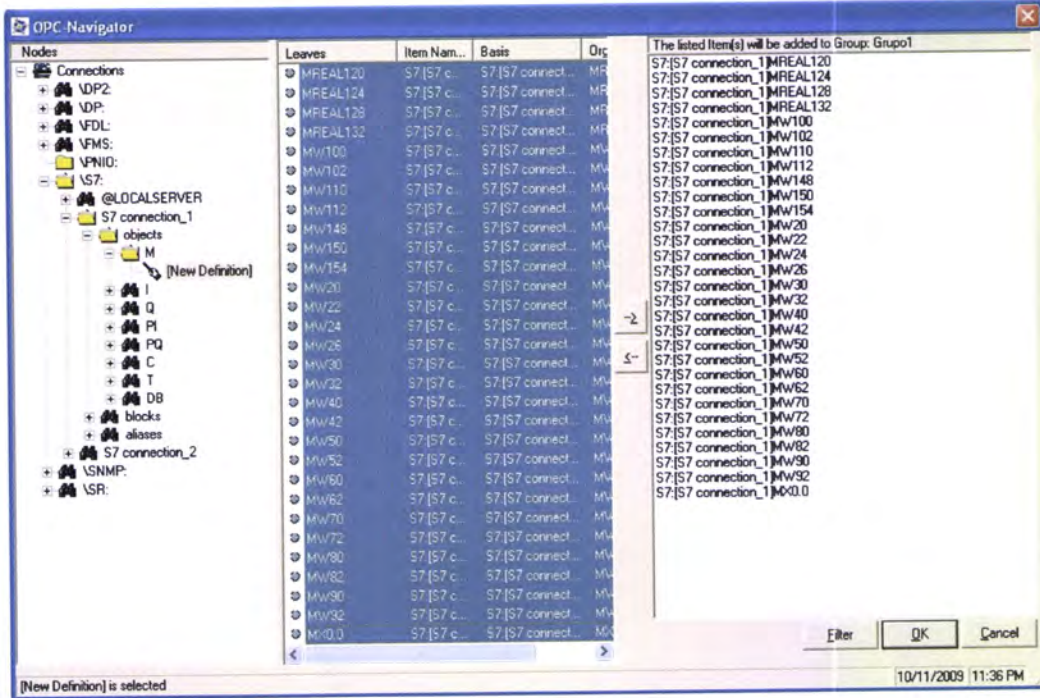


Figura 10: Adicionar tags ao grupo criado

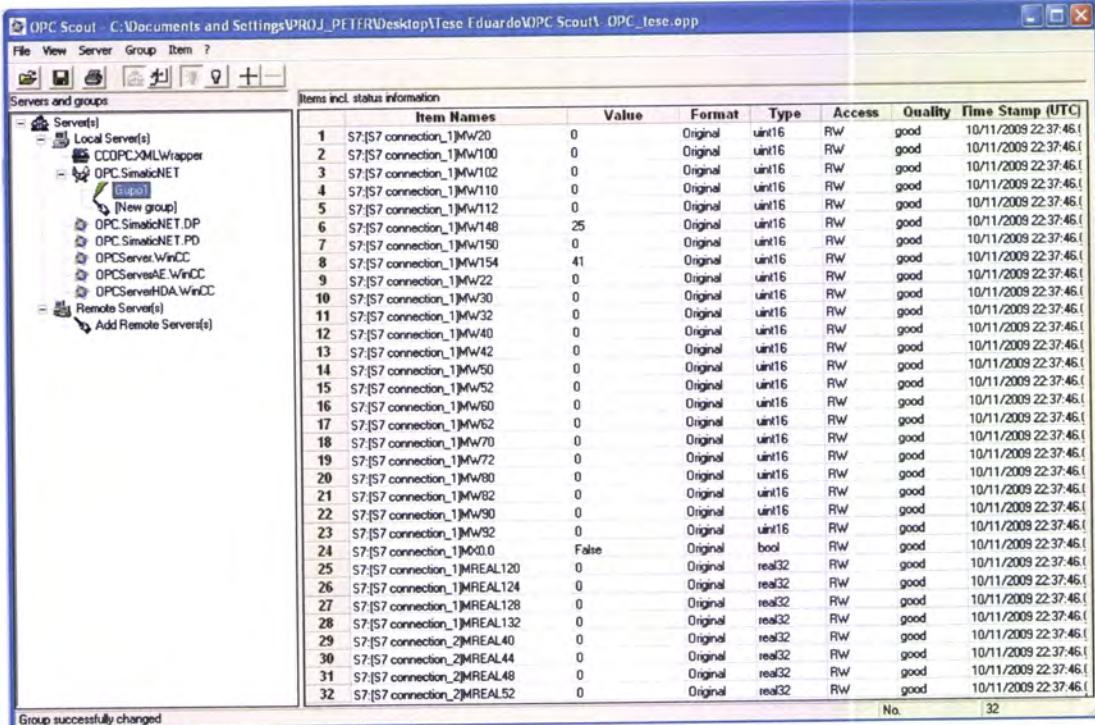


Figura 11: Resultado final do processo "OPC Scout"

A lista de todas as "tags" utilizadas encontra-se no anexo A.

5.3 Controlo local (PLC)

5.3.1 Principais características

As principais características de um PLC são:

- Existência de várias linguagens de programação que tornam o processo fácil;
- Facilidade de manutenção;
- Alta fiabilidade;
- Envio de dados para processamento centralizado do sistema;
- Possibilidade de expansão através de módulos.

O nível correspondente à plataforma interactiva estudada nesta dissertação é composto por dois PLC's. O primeiro dos PLC's terá como função o controlo da conference room do edifício e o segundo controlará a unidade de fornecimento (AVAC, estores e iluminação artificial).

Tomemos como exemplo o PLC da conference room (código disponível no anexo B) para explicar o seu desenvolvimento. O PLC comunica com os demais constituintes do sistema através de tags, variáveis que podem ser de diversos tipos dependendo do que se pretende comunicar.

Neste caso particular o PLC vai:

- Receber os valores de referência introduzidos por cada utilizador nas interfaces distribuídas existentes na conference room e fornecer os valores ao sistema SCADA;
- Receber os valores provenientes dos sensores de temperatura e luminosidade instalados na conference room e fornecer esses valores às interfaces distribuídas e ao SCADA.

Passo agora a explicar o funcionamento do PLC que controla a unidade de fornecimento (código disponível no anexo D). Este irá receber os valores de funcionamento provenientes do SCADA e actuará sobre o AVAC, estores e iluminação artificial da conference room.

5.3.2 Configuração dos PLC's

O software **Siemens Simatic Manager STEP 7** [SIEMENS, 2006a] por defeito vem configurado para que a comunicação com o autómato seja feita através de MPI. Na plataforma desenvolvida a comunicação é feita por Ethernet, para tal é necessário modificar a interface de comunicação.

Este procedimento é efectuado acedendo ao **painel de controlo** do Windows e de seguida abrindo a aplicação **Set PG/PC Interface**. Aberta a aplicação, no campo **Access Point of the Application** altera-se a configuração existente para “S7ONLINE (STEP7) → TCP/IP (nome da placa de rede do computador).

A programação dos PLC's que constituem o sistema de controlo local da plataforma interactiva é efectuada através do software **Siemens Simatic Manager STEP7**. De seguida serão apresentados os passos necessários à programação da plataforma.

1. Abrir o software **Siemens Simatic Manager STEP 7** e de seguida carregar no ícone de criação de um novo projecto.
2. Definir o nome do projecto e o local de gravação do projecto. Ao projecto realizado foi atribuído o nome “plataforma_interactiva”.

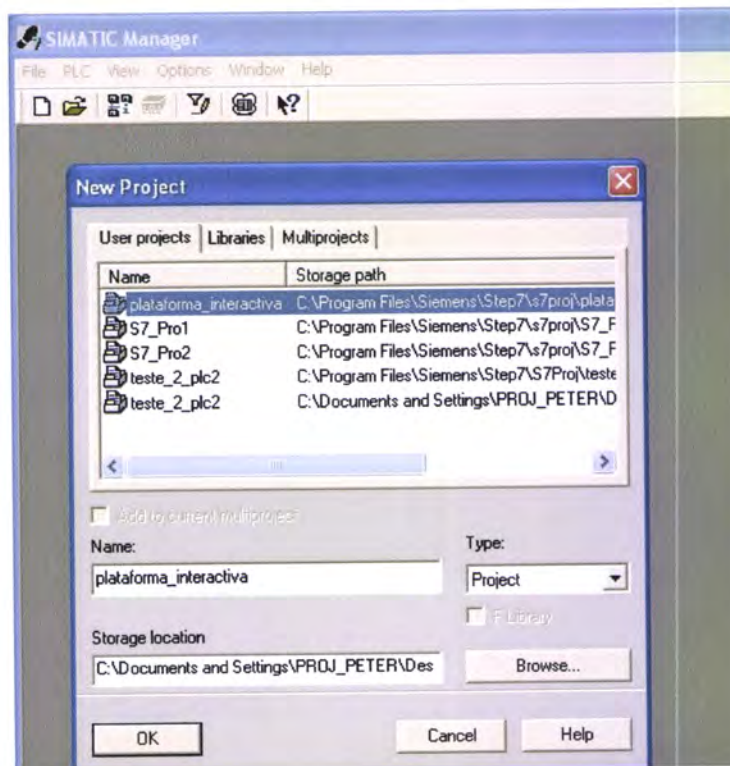


Figura 12: Criação de projecto

3. Após ser criado o projecto é necessário adicionar os autómatos que compõem o projecto desenvolvido. Para se adicionar os autómatos selecciona-se o projecto criado e de seguida carrega-se na barra de ferramentas em **“Insert → Station → 2 SIMATIC 300 Station”**. Este procedimento é feito duas vezes, dado que existem dois autómatos. Os autómatos podem ser renomeados, no projecto desenvolvido o primeiro passou a chamar-se “conference room” e o segundo “unidade fornecimento”. É também necessário proceder à introdução de uma SIMATIC PC Station (**4 SIMATIC PC Station**) para permitir a comunicação entre os vários componentes do sistema.

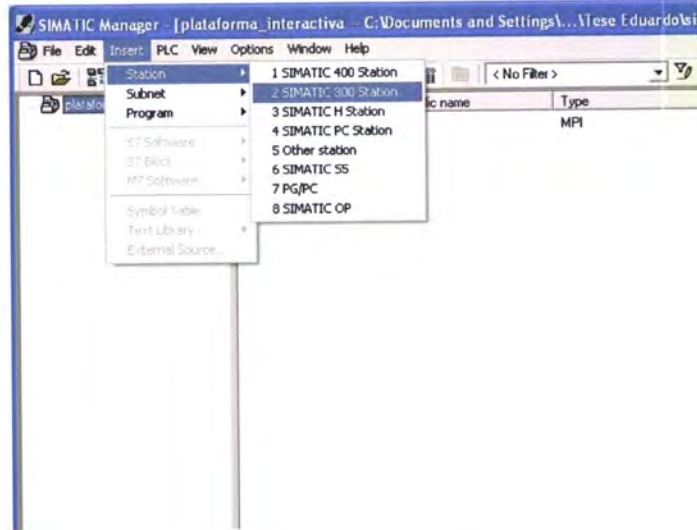


Figura 13: Inserir autómato

- Depois de adicionados, os autómato têm de ser configurados. Selecciona-se o autómato pretendido na “árvore” do projecto (divisória do lado esquerdo da janela) e na tela do lado direito da janela surge o ícone “**Hardware**”. Ao clicar no ícone “**Hardware**” irá surgir uma nova janela com o nome “HW Config”.

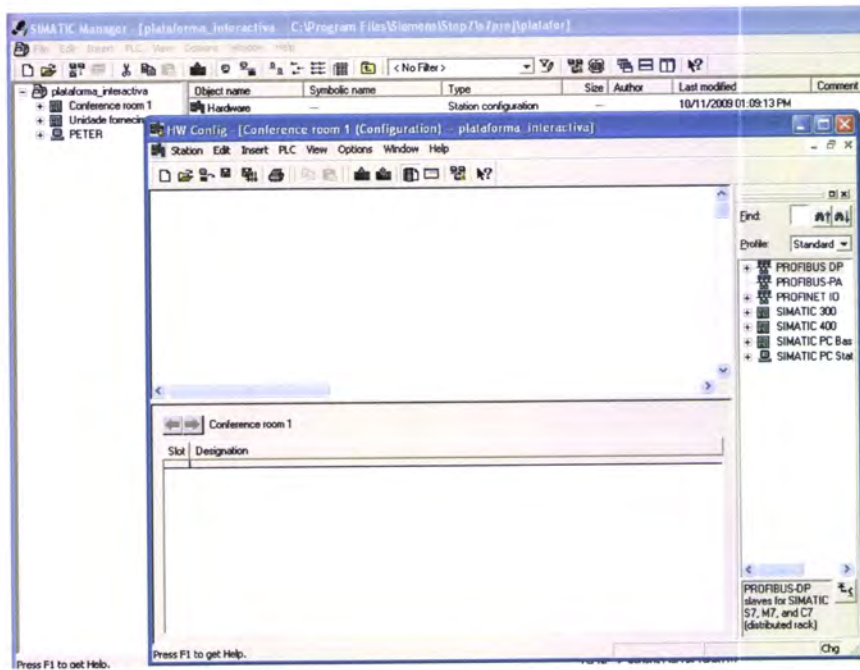


Figura 14: Janela de configuração do hardware

- Nesta nova janela a primeira coisa a ser efectuada é a introdução de uma “rack” onde o autómato será configurado. A adição da “rack” é feita carregando com o botão direito do rato sobre a janela em branco, aparece um quadro de opções e nesse quadro carrega-se em “**SIMATIC 300** → **RACK-300** → **Rail**”.

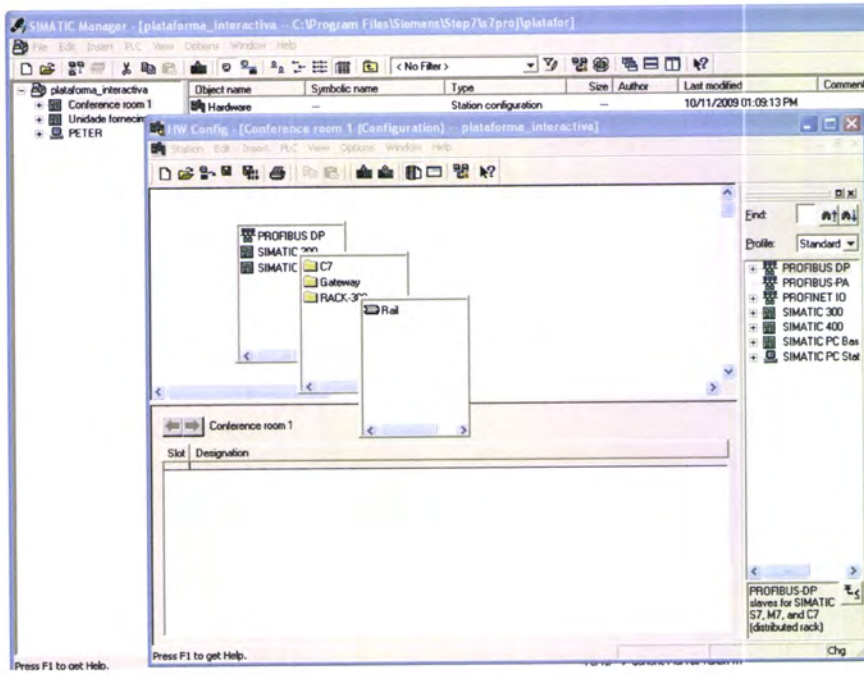


Figura 15: Criação da rack

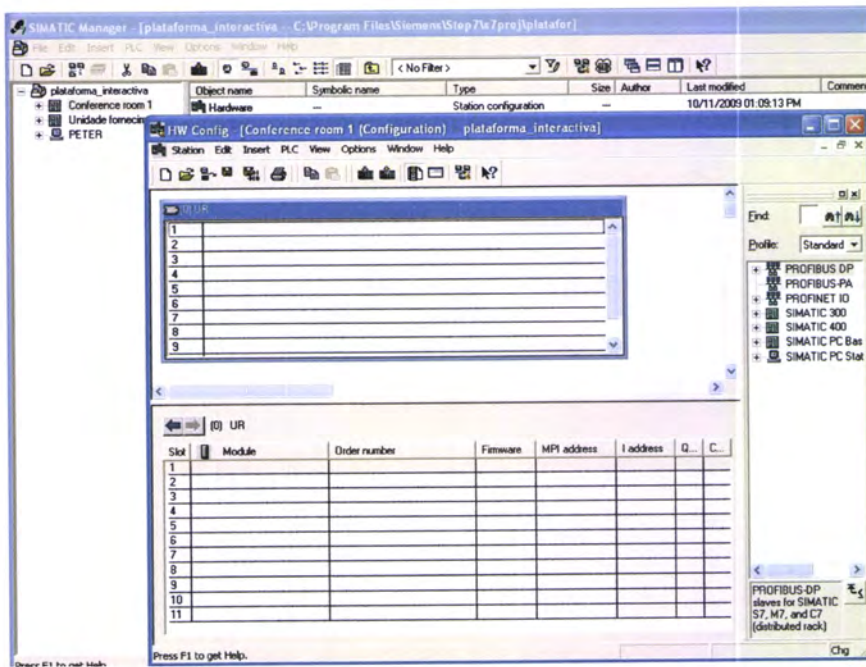


Figura 16: Aspecto da rack

6. A configuração da “rack” depende do CPU que está a ser usado e dos componentes a este ligado na mesma “rack”. As configurações das “racks” dos autómatos e Simatic PC Station usados neste projecto são as seguintes:

- Autómato “conference room 1”

Módulo	Slot	Número de ordem
PS 507 5A	1	6ES7 307-2EA00-0AA0
CPU 315-2 DP	2	6ES7 315-2AG10-0AB0
CP 343-1 IT	4	6GK7 343-1GX20-0XE0
DI8/DO8 x DC24V/0,5A	5	6ES7 323-1BH01-0AA0
DI8/DO8 x DC24V/0,5A	6	6ES7 323-1BH01-0AA0
AI8 x 12Bit	7	6ES7 331-7KF02-0AB0
AO4 x 12Bit	8	6ES7 332-5HD01-0AB0

Tabela 2: Módulos do autómato “conference room”

- Autómato “unidade fornecimento”

Módulo	Slot	Número de ordem
	1	
CPU 315-2 DP	2	6ES7 315-2AG10-0AB0
CP 342-5	4	6GK7 342-5DA02-0XE0
DI8/DO8 x DC24V/0,5A	5	6ES7 323-1BH01-0AA0
AI8 x 12Bit	6	6ES7 331-7KF02-0AB0
AO4 x 12Bit	7	6ES7 332-5HD01-0AB0
FM 350 Counter	8	6ES7 350-1AH03-0AE0
FM 350 Counter	9	6ES7 350-1AH03-0AE0
FM 353 Stepper	10	6ES7 353-1AH01-0AE0
FM 353 Stepper	11	6ES7 353-1AH01-0AE0

Tabela 3: Módulos do autómato “unidade fornecimento”

- Simatic PC Station “PETER”

Módulo	Slot	Número de ordem
OPC Server	1	
IE General	2	IE_CP

Tabela 4: Módulos da “PC Station”

A adição dos módulos é bastante simples, bastando para tal seleccionar a linha (corresponde à slot indicada na tabela) da “rack”, clicar com o botão direito do rato e então escolher o que se pretende adicionar da lista que surge ao dispor do utilizador. Na figura seguinte é exemplificada a adição do módulo DI8/DO8 x DC24V/0,5A na linha 5 do autómato “conference room 1”.

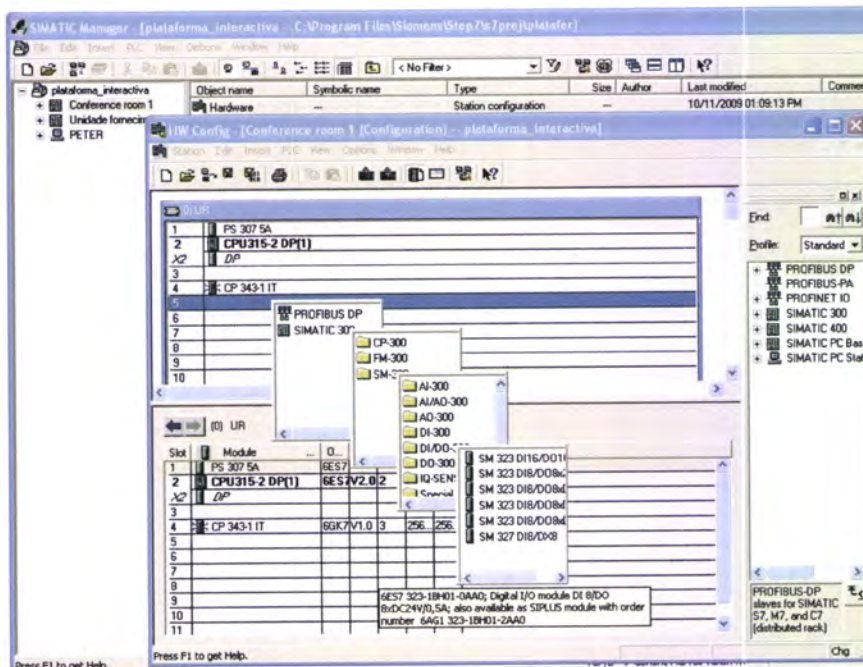


Figura 17: Adição do módulo na rack do autómato

Ao autómato “conference room 1” deveriam ser ligadas as interfaces distribuídas através de uma rede PROFIBUS, no entanto como o projecto apenas foi implementado em parte para permitir a simulação, a ligação da interface usada foi feita através de MPI.

- Por fim é necessário efectuar a configuração da rede Ethernet que vai permitir a ligação entre os autómatos e o sistema SCADA.

Autómato	Endereço Ethernet
Conference room	192.168.0.4
Unidade fornecimento	192.168.0.1
OPC Server	192.168.0.10

Tabela 5: Endereço Ethernet dos autómatos do sistema

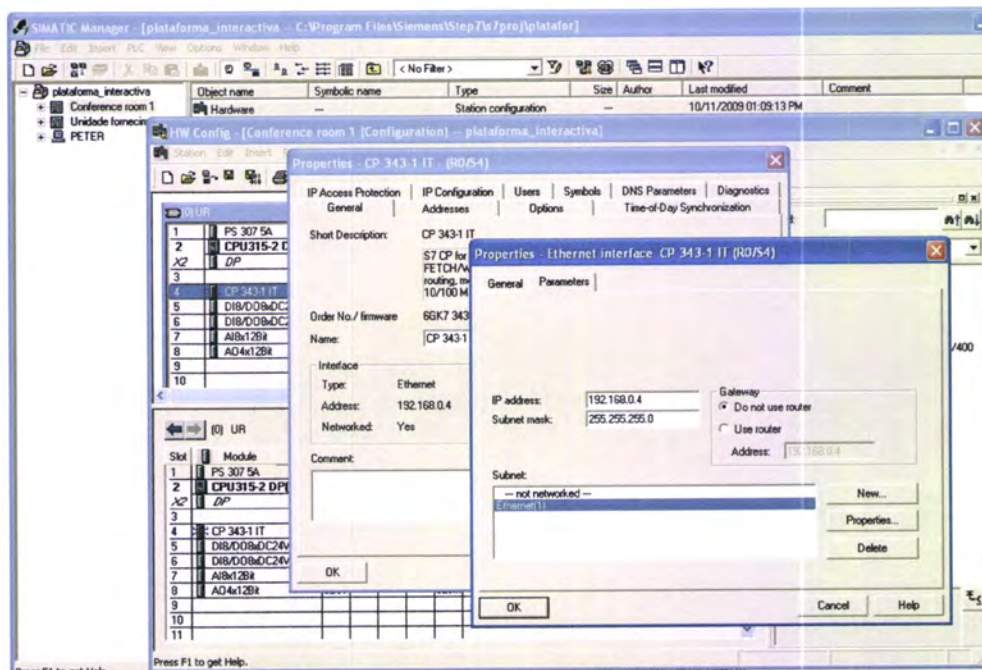


Figura 18: Configuração da rede Ethernet

8. Tendo os hardwares configurados é necessário abrir o software **NetPro** incorporado no **Siemens Simatic STEP 7**. De seguida selecciona-se o OPC Server, na janela “Local ID” clica-se com o botão direito do rato e escolhe-se “Insert New Connection”. Surge uma janela onde se indica o autómato com o qual a PC Station vai comunicar.

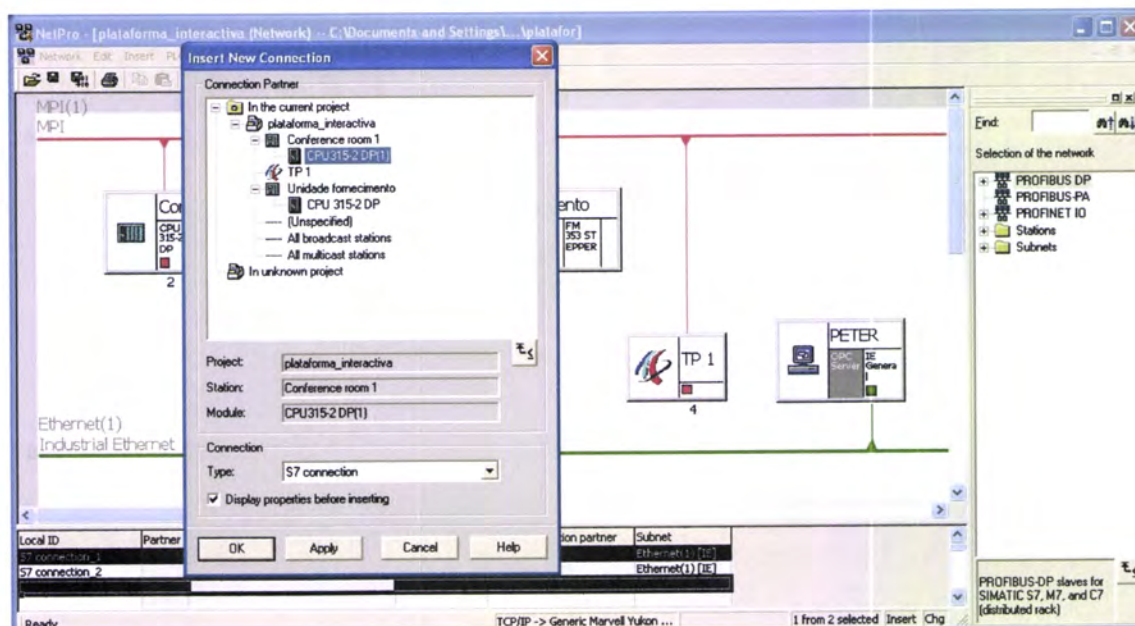


Figura 19: Criação de conexão entre PC Station e autómato

Após a criação da conexão surge uma janela com a informação da conexão, que é necessária para a configuração do “OPC Scout” (secção 5.2.3).

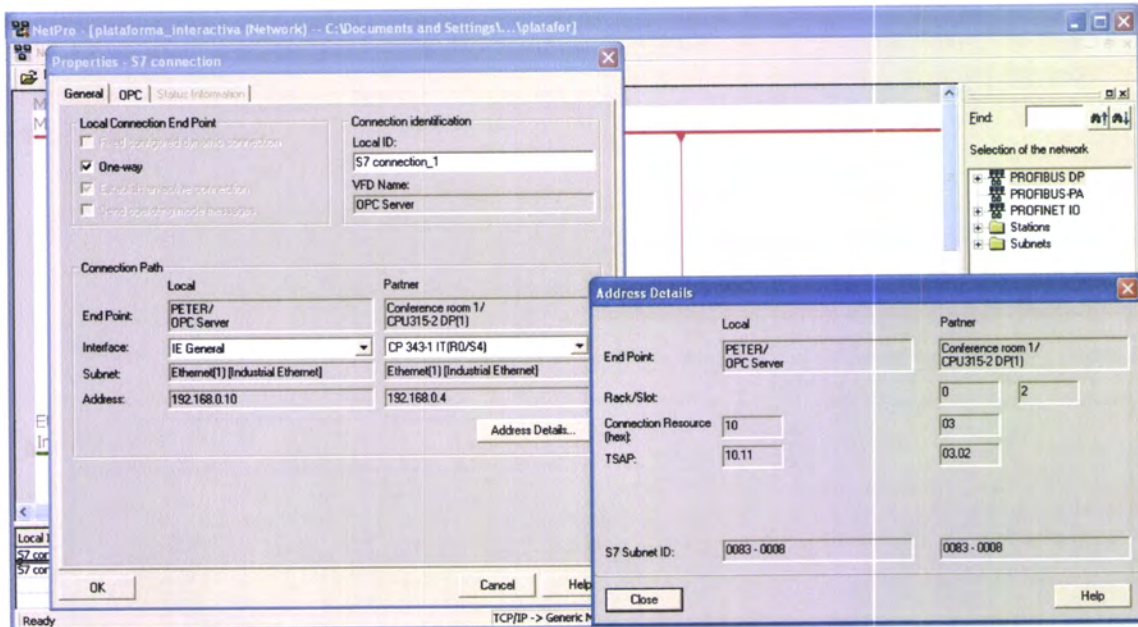


Figura 20: Informação S7 connection_1

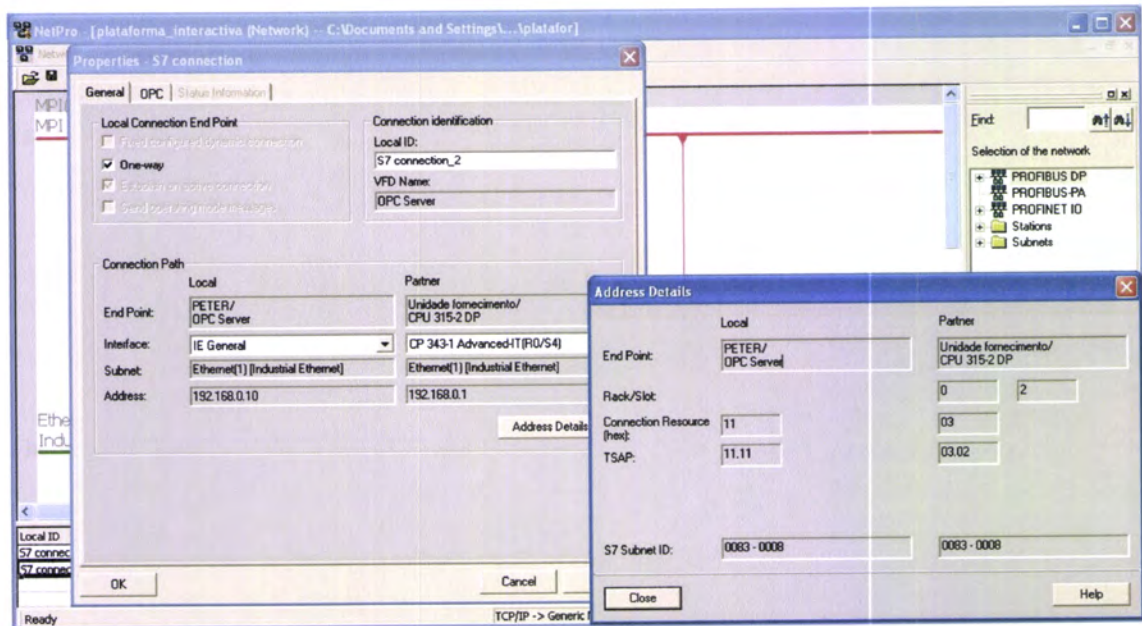


Figura 21: Informação S7 connection_2

O aspecto final do projecto desenvolvido através do NetPro encontra-se no anexo H.

5.3.3 Programação dos PLC's

A programação dos PLC's foi feita através da linguagem de programação STL (Statement List), que consiste num conjunto de simples instruções. Embora não seja uma programação fácil no início da sua utilização, quando o programador ganha uma certa rotina encontra com mais facilidade erros que tenha efectuado, comparando com a programação de outras linguagens de programação.

De seguida são apresentadas as instruções [Siemens, 2006b] utilizadas na programação dos autómatos presentes no sistema:

Instruções lógicas

- **A** (And);
- **AN** (And Not);
- **O** (Or);
- **A(** (And with Nesting Open);
- **O(** (Or with Nesting Open);
- **)** (Nesting Closed);
- **=** (Assign);

Instruções de comparação

- **<>** (Compare Integer (16-Bit));

Instruções de controlo lógico

- **JNB** (Jump if RLO=0 with BR);

Instruções matemáticas (números inteiros)

- **+I** (Add ACCU1 and ACCU2 as integer (16-Bit));

Instruções de leitura e transferência

- **L** (Load);
- **T** (Transfer);

Instruções de acumulação

- **NOP 0** (Null Instruction);

5.2.4 Configuração das interfaces HMI

1. Na janela principal do **Simatic Manager STEP 7** clica-se na janela do projecto criado e de seguida **“Insert New Object → SIMATIC OP”**.

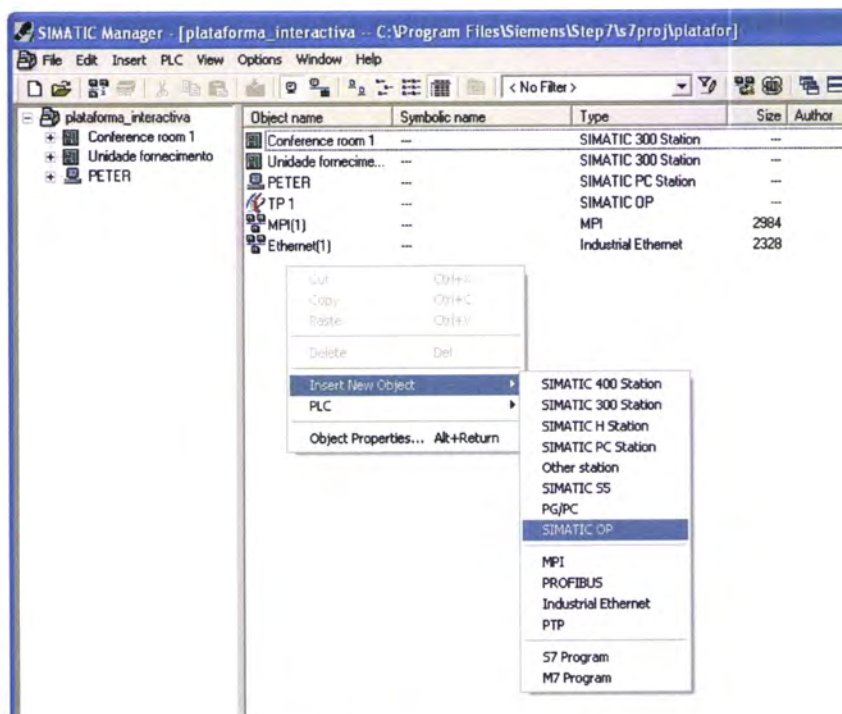


Figura 22: Inserção de uma interface HMI ao projecto

2. Surgirá uma nova janela chamada **“Project Wizard – Device Selection”** onde é escolhido o modelo da interface. No presente projecto escolher **“Windows-based systems → Panels → TP170A”** e clicar em seguinte.

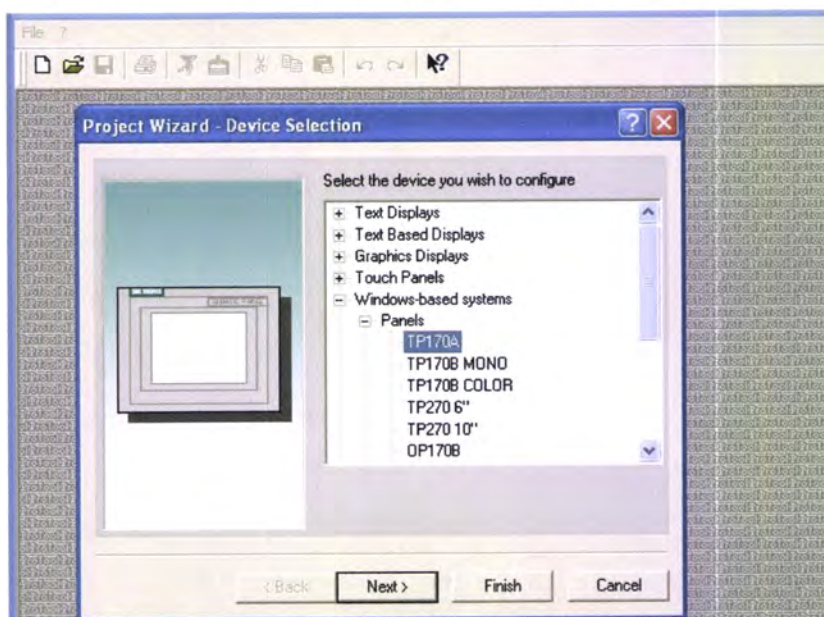


Figura 23: Escolha da interface

- De seguida atribui-se o nome ao autómato que será objecto da comunicação com o painel a ser configurado e selecciona-se o protocolo de comunicação.

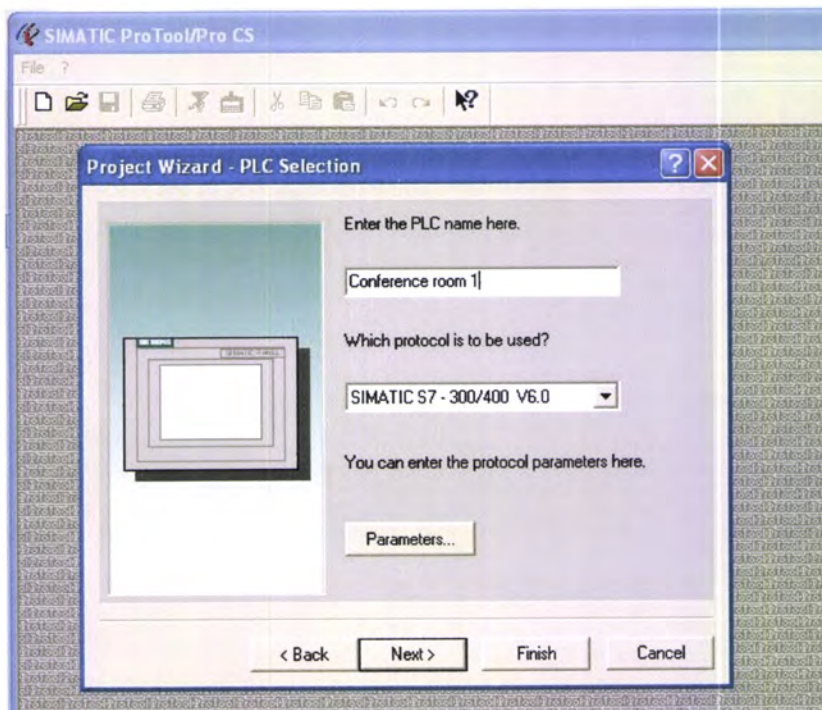


Figura 24: Designação do autómato e tipo de comunicação

- É necessário configurar os parâmetros de ligação da interface ao autómato. Clica-se em **“Parameters”**, aparece uma nova janela na qual se deve escolher o **“profile”** MPI e o **“Address”** 4.

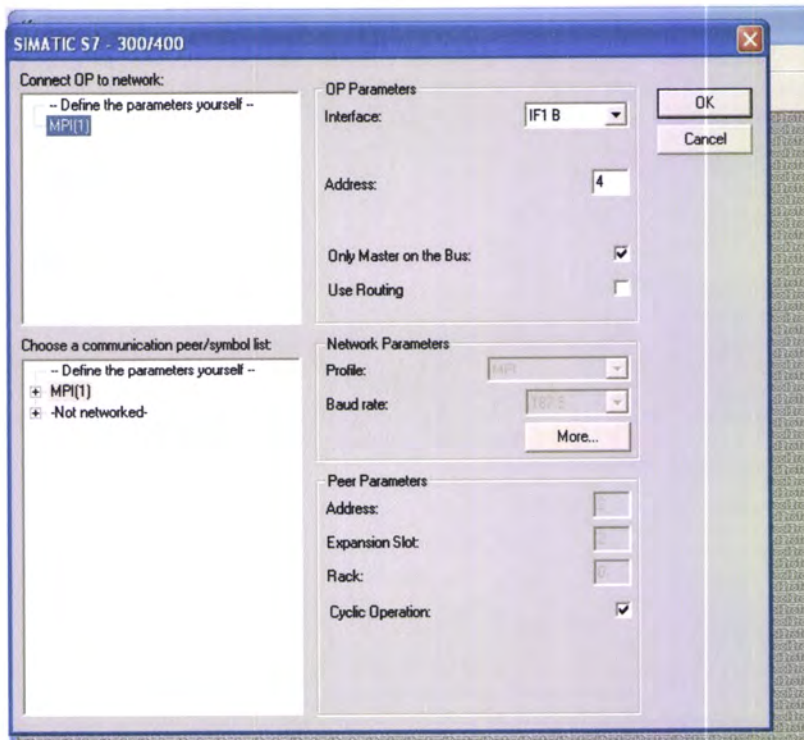


Figura 25: Parâmetros de ligação

5. Após as configurações anteriores carrega-se em concluir e surge a janela do projecto.

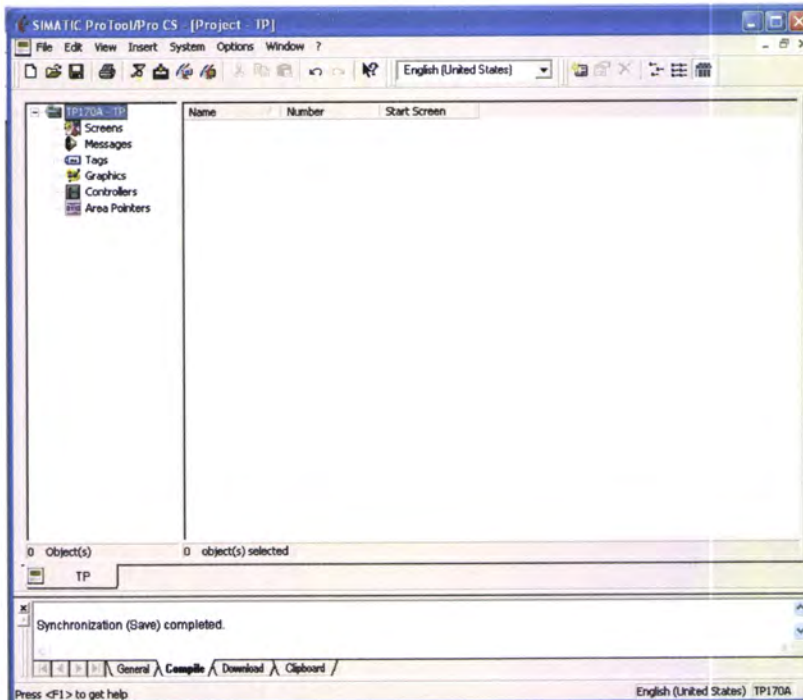


Figura 26: Menu inicial do novo projecto

6. Na janela *Simatic ProTool* [SIEMENS, 2004] começa-se por configurar as “tags” de forma a existir comunicação entre o autómato e a interface. Para criar uma “tag”, selecciona-se na árvore do projecto o menu “tags” e clica-se com o botão direito do rato na tela do lado direito e escolher a opção “*Tag Insert*”.

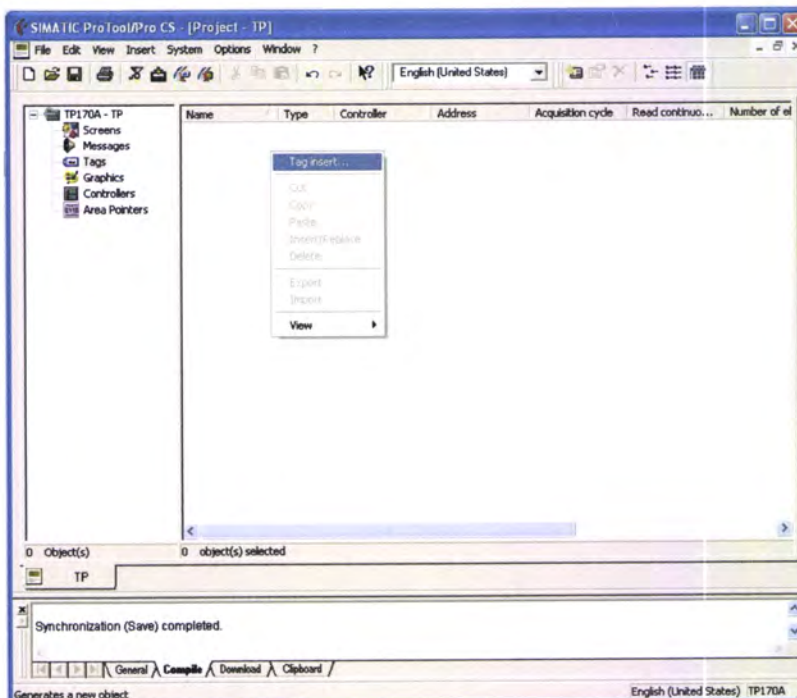


Figura 27: Criação de uma “tag”

- Na figura seguinte apresenta-se a configuração da tag “Temp_actual” e na janela em segundo plano a lista completa das “tags” usadas no projecto.

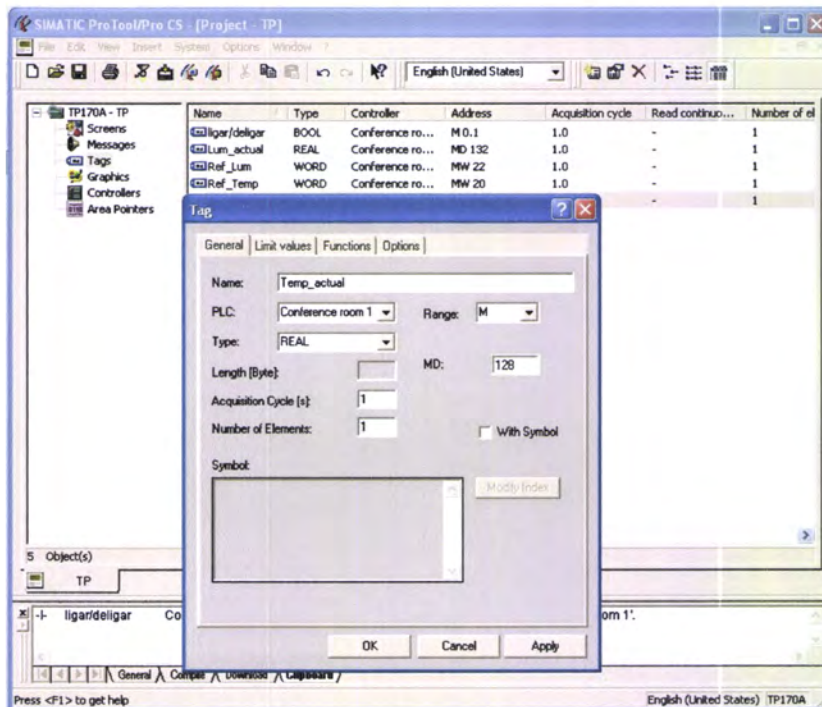


Figura 28: Configuração das “tags”

- O passo seguinte é a criação dos ecrãs que permitem fornecer as preferências do utilizador ao sistema e visualizar as condições actuais de temperatura e luminosidade da conference room. Os ecrãs podem ser criados clicando 2 vezes no menu “Screens” na árvore do projecto, aparecendo uma nova janela onde o ecrã é configurado.

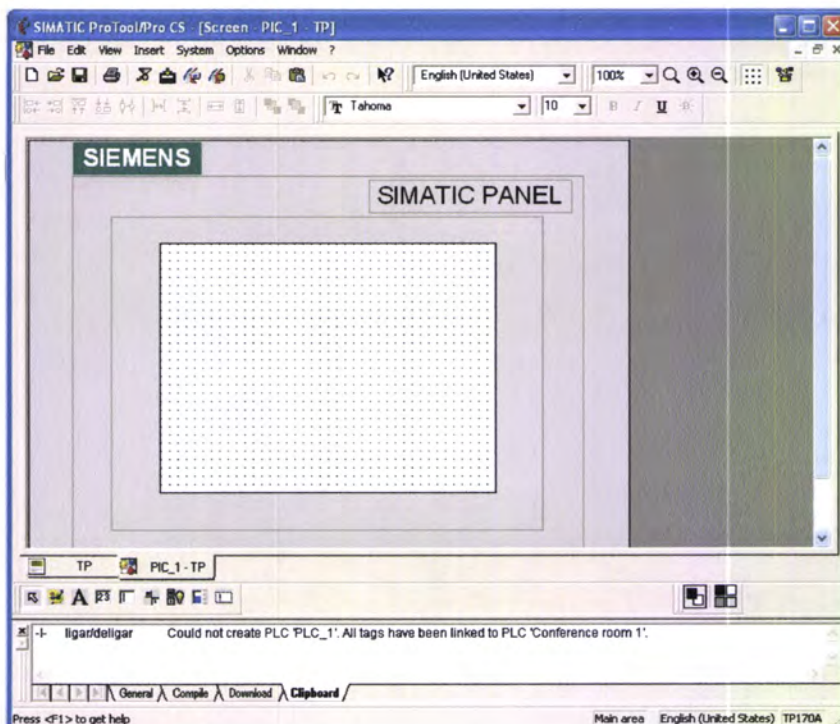


Figura 29: Criação de um ecrã

Um novo ecrã é configurado através da introdução de vários campos disponíveis ao utilizador. Existem:

- **Campos de texto** (*Text Field*) onde são visíveis mensagens de texto fixas, podendo estas serem alteradas apenas através da programação do ecrã.
- **Botões de estado** (*State Button*) que executam tarefas como mudança de ecrã, alteração de valores de “tags” ou ainda funções especiais. Um exemplo de uma função especial é o “Exit_Runtime” que permite ao operador sair do programa para aceder ao menu raiz da interface HMI.
- **Campos de entradas/saídas** (*Input/Output field*) onde se podem visualizar os valores das “tags”. Um campo de saída apenas fornece o valor da “tag”, enquanto um campo de entrada permite a alteração do valor da “tag” associada.
- **Mensagens** (*Event messages*) que disponibilizam informações sobre o funcionamento do sistema.

No primeiro ecrã criado (*Inicio*) foram introduzidos dois campos de texto e dois botões de estado. A figura 30 mostra a introdução de um campo de texto, as figuras 31 e 32 mostram a configuração dos botões de estado do ecrã.

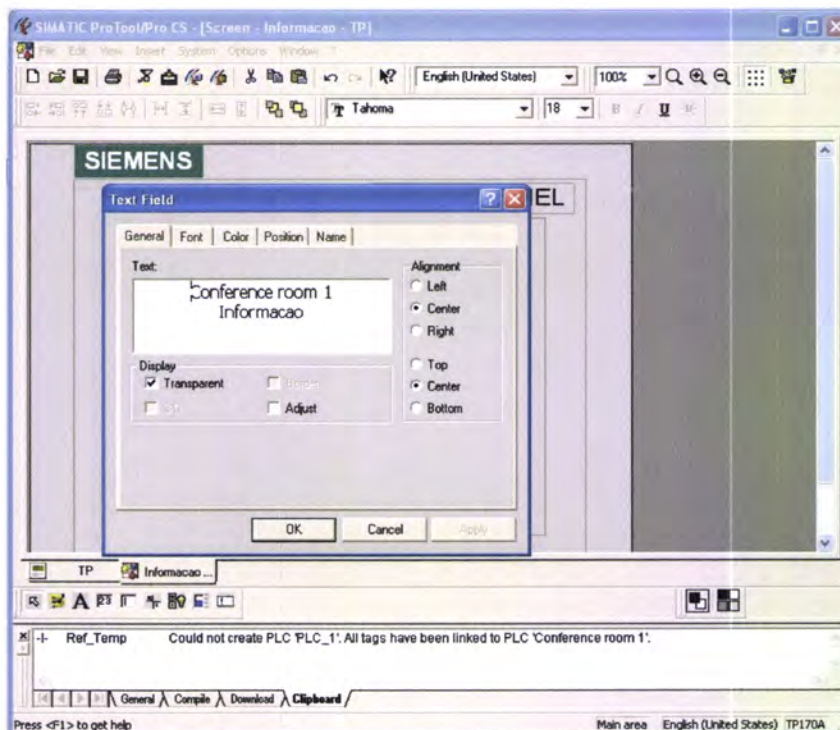


Figura 30: Introdução de um campo de texto

Na figura 31, quando se configura o botão “Entrar” para efectuar a mudança de ecrã, tem de se seleccionar em **Screen Name** o nome do ecrã para o qual se quer mudar. O mesmo botão tem outra função, quando este é pressionado o valor de uma “tag” (*ligar/desligar*) é alterado, de zero para um, através de uma função (*set_bit*).

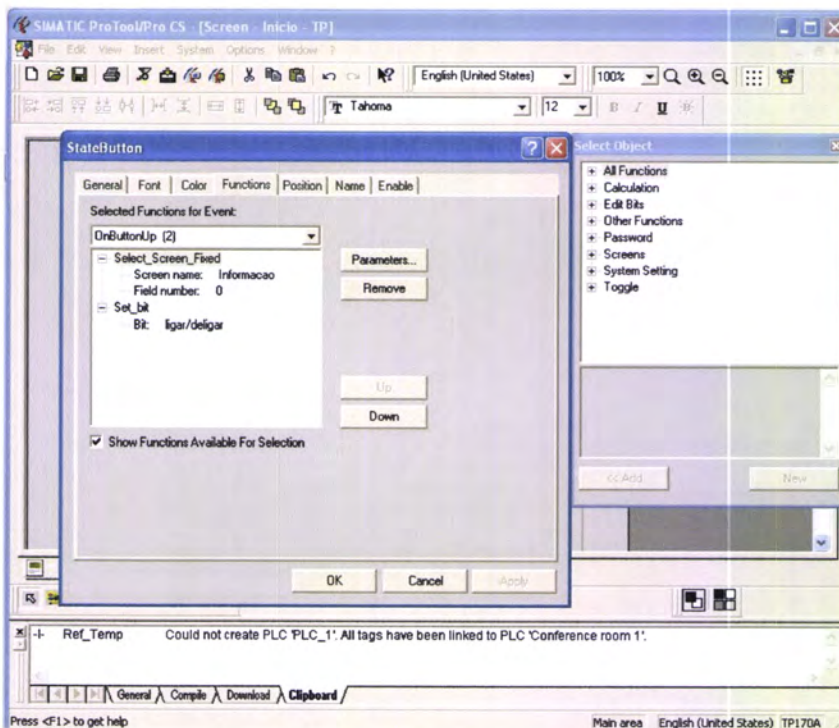


Figura 31: Configuração do botão de estado "Entrar"

O botão "Configuracao sistema" efectua a função especial "Exit_Runtime".

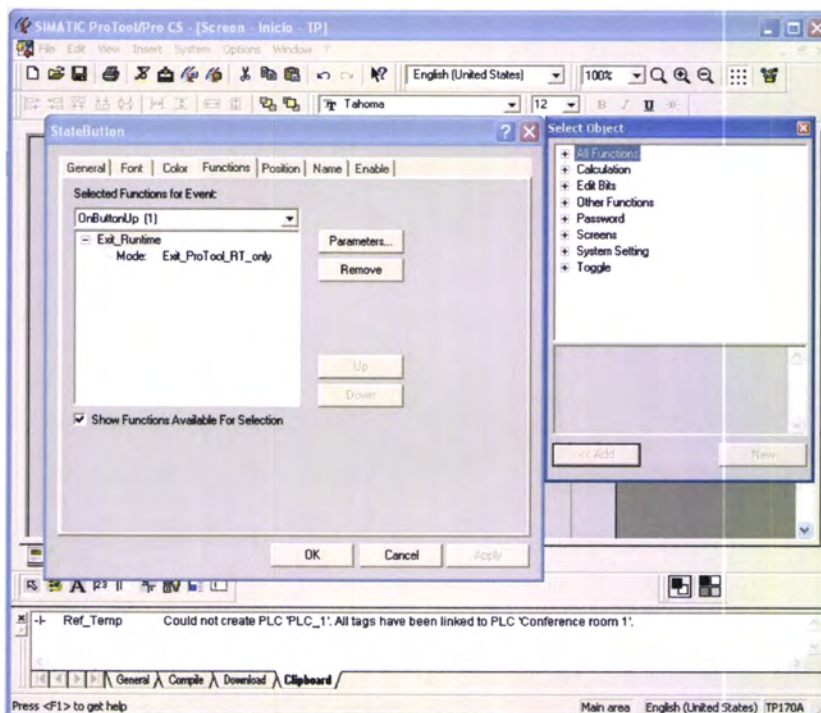


Figura 32: Configuração do botão de estado "Configuracao sistema"

No segundo ecrã criado (*Informacao*) foram introduzidos três campos de texto, dois botões de estado, dois campos de saída e uma área de mensagem. Os dois

botões de estados terão, tal como no primeira ecrã, a função de efectuar a mudança de ecrã (figuras 33 e 34).

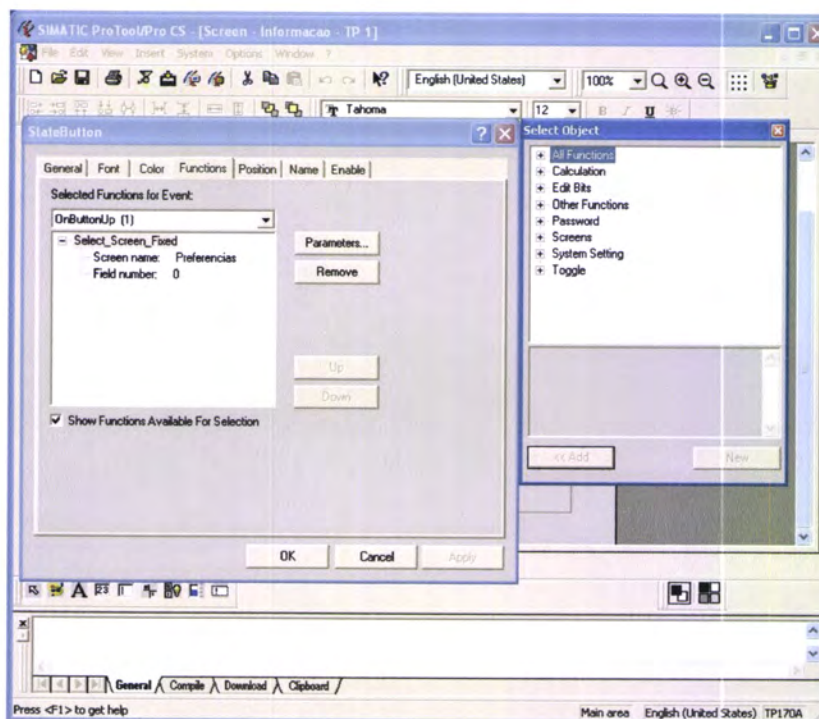


Figura 33: Configuração do botão de estado "Preferências utilizador"

O botão “Sair” tem ainda a função de mudar o valor da “tag” *ligar/desligar*, de um para zero, através de uma função (*reset_bit*).

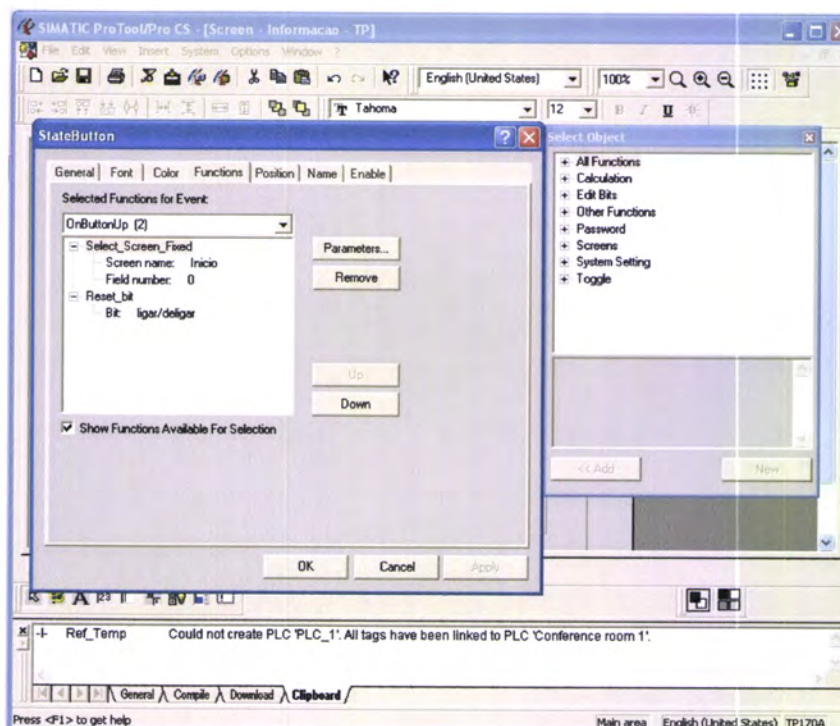


Figura 34: Configuração do botão de estado "Sair"

Os campos de saída fornecem ao utilizador a informação de duas "tags" *Temp_actual* e *Lum_actual*, figuras 35 e 36 respectivamente.

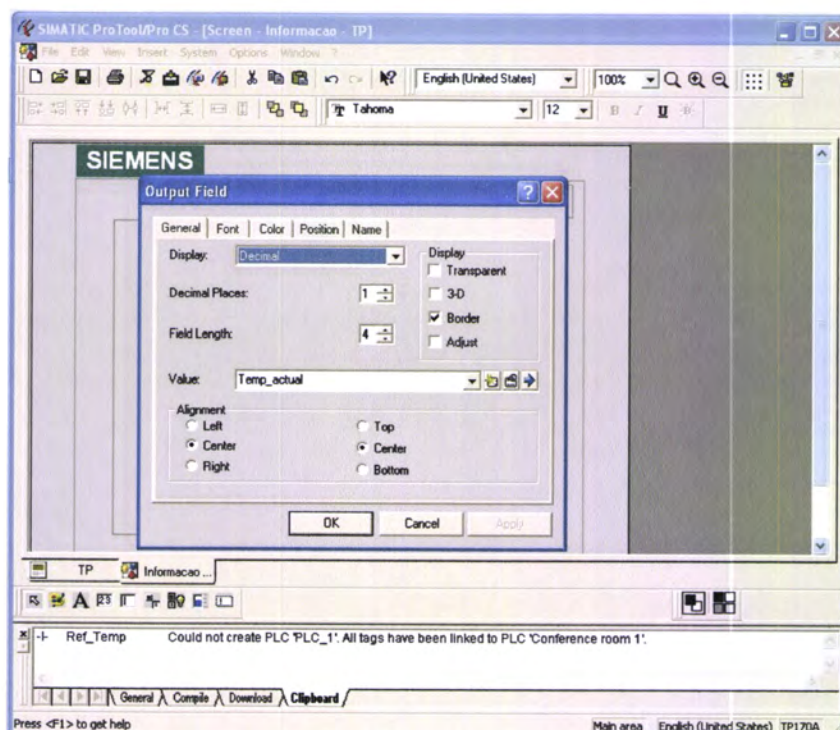


Figura 35: Configuração do campo de saída "Temp_actual"

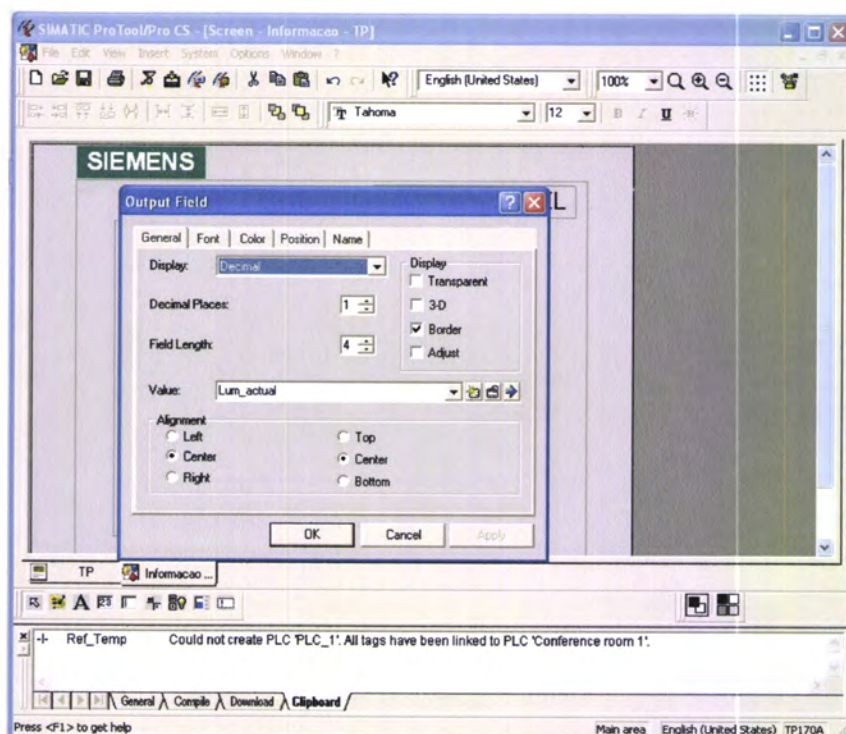


Figura 36: Configuração do campo de saída "Lum_actual"

A área de mensagem vai dar informações sobre a actividade do sistema e a hora a que surgiu a mensagem na interface, figura 37.

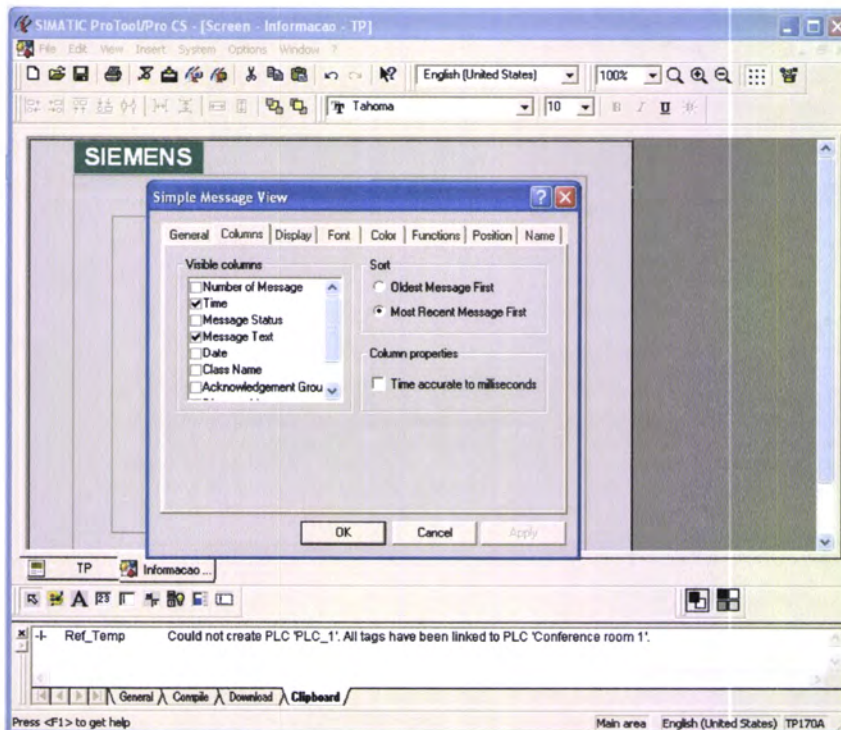


Figura 37: Configuração da área de mensagens

No terceiro ecrã criado (*Preferencias*) foram introduzidos três campos de texto, um botão de estado e dois campos de entrada. O botão de estados, tal como nos ecrãs anteriores, terá a função de efectuar a mudança de ecrã (figura 38).

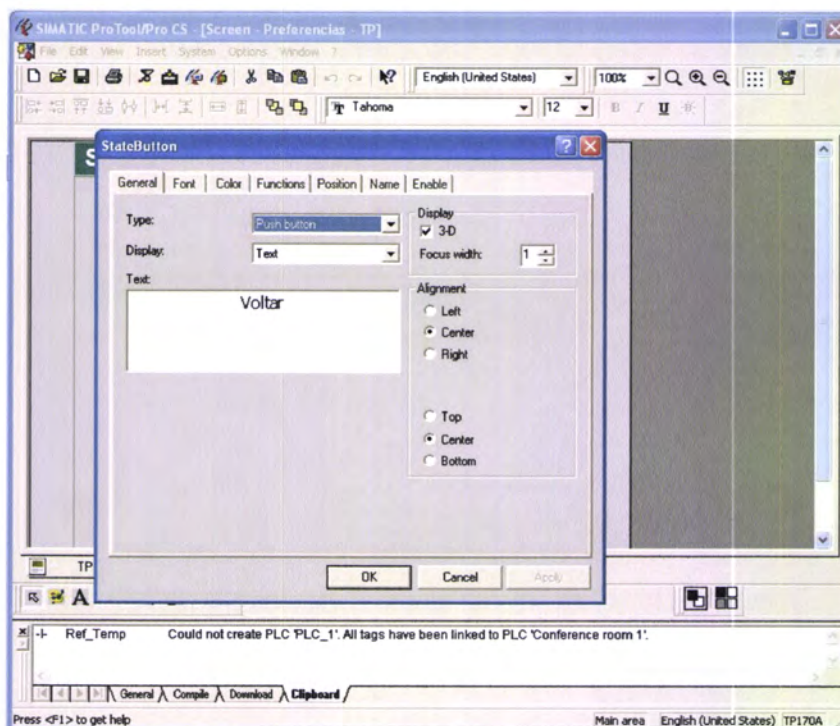


Figura 38: Configuração do botão de estado "Voltar"

Os dois campos de entrada permitem ao utilizador alterar os valores de duas “tags” *Ref_Temp* e *Ref_Lum*, figuras 39 e 40.

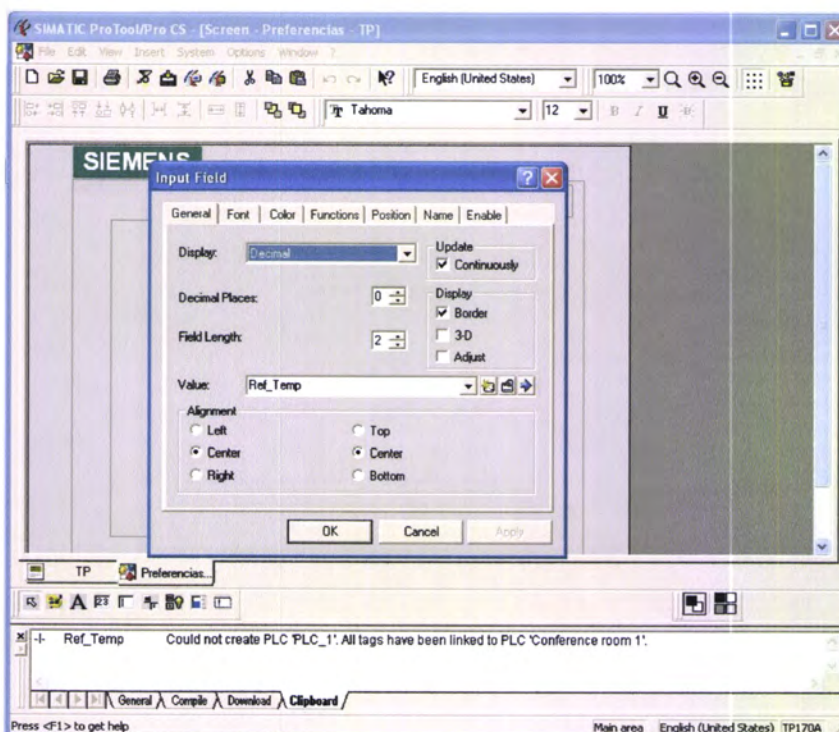


Figura 39: Configuração do campo de entrada "Ref_Temp"

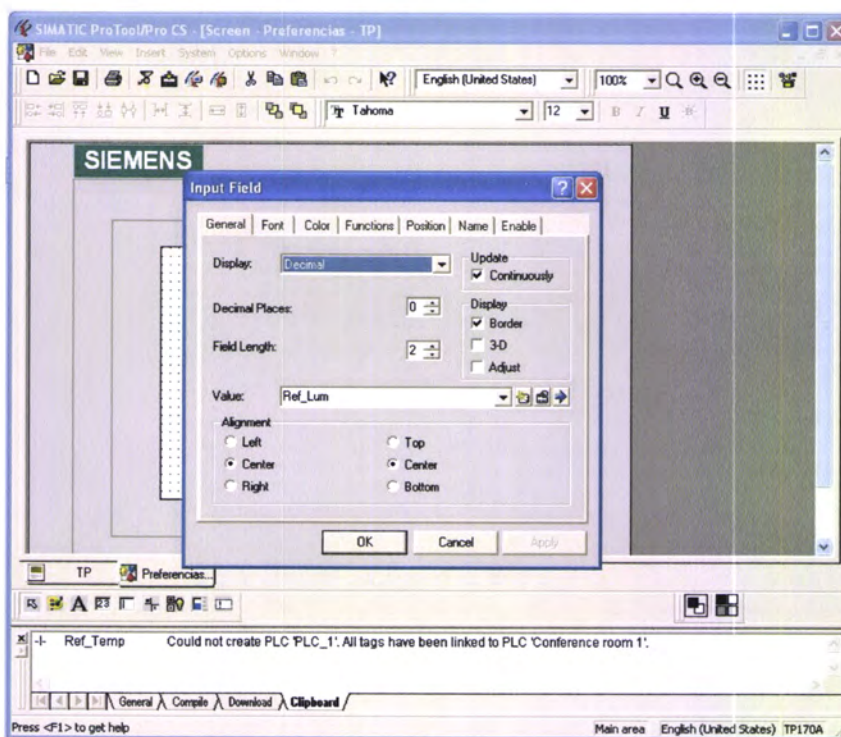


Figura 40: Configuração do campo de entrada "Ref_Lum"

- Falta referir de que modo é configurado as mensagens para que surjam na área de mensagens criada no ecrã *Informacao*. Para tal é necessário ir ao menu **“Area Pointers”** na árvore do projecto e de seguida clicar em **“Event Messages”** na janela que se abre, conforme é ilustrado na figura 41. Após essa selecção é necessário configurar a *Event Message*, figura 42.

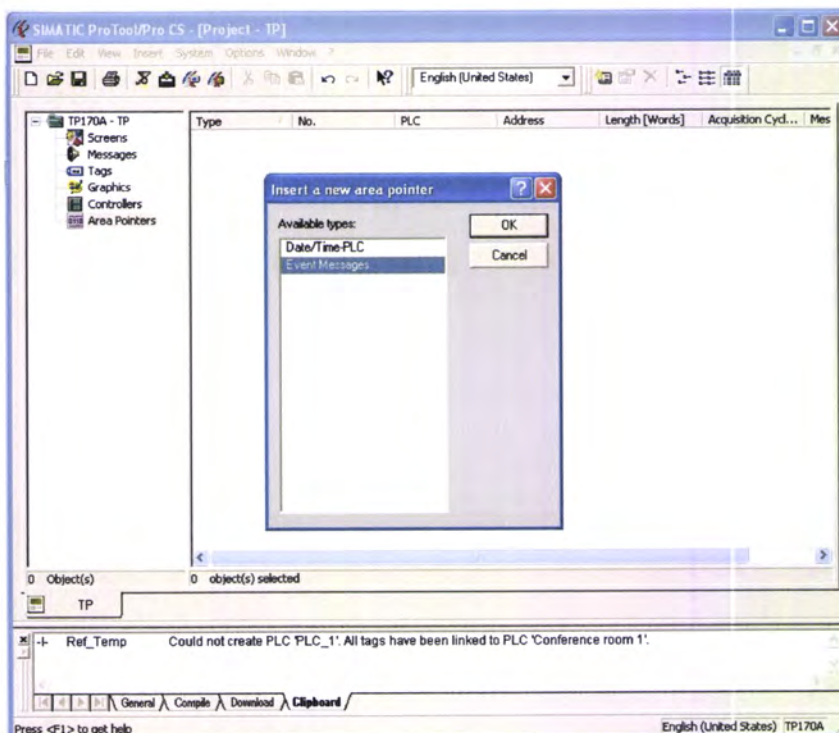


Figura 41: Criação de "Event Messages"

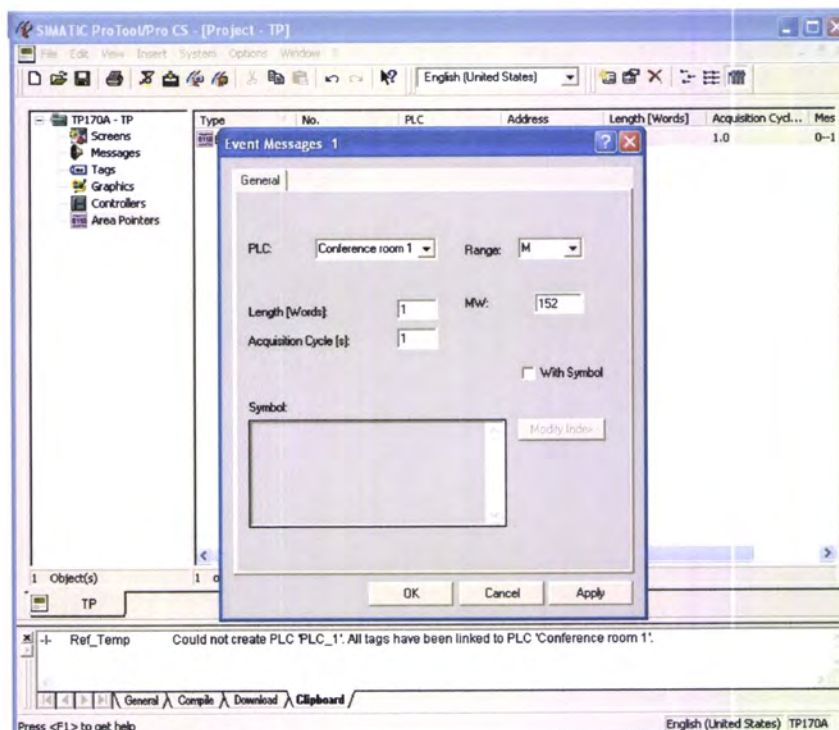


Figura 42: Configuração da "Event Message" criada

Por fim, para inserir o texto das mensagens é necessário ir ao menu **“Messages”** na árvore do projecto e depois clicar em **“Event Messages”**, como se pode ver na figura 43.

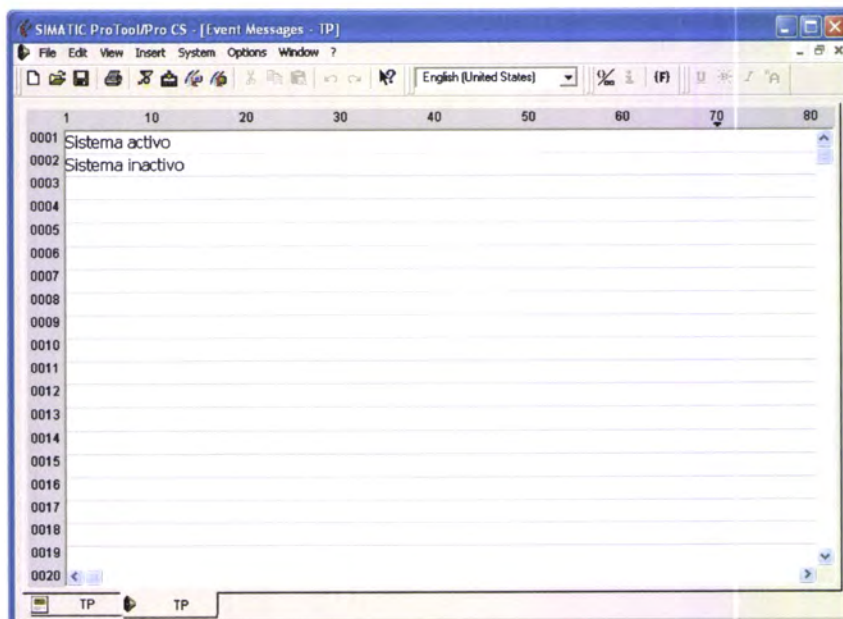


Figura 43: Texto da "Event Message"

O resultado da criação dos ecrãs das interfaces HMI pode ser observado nas figuras seguintes.

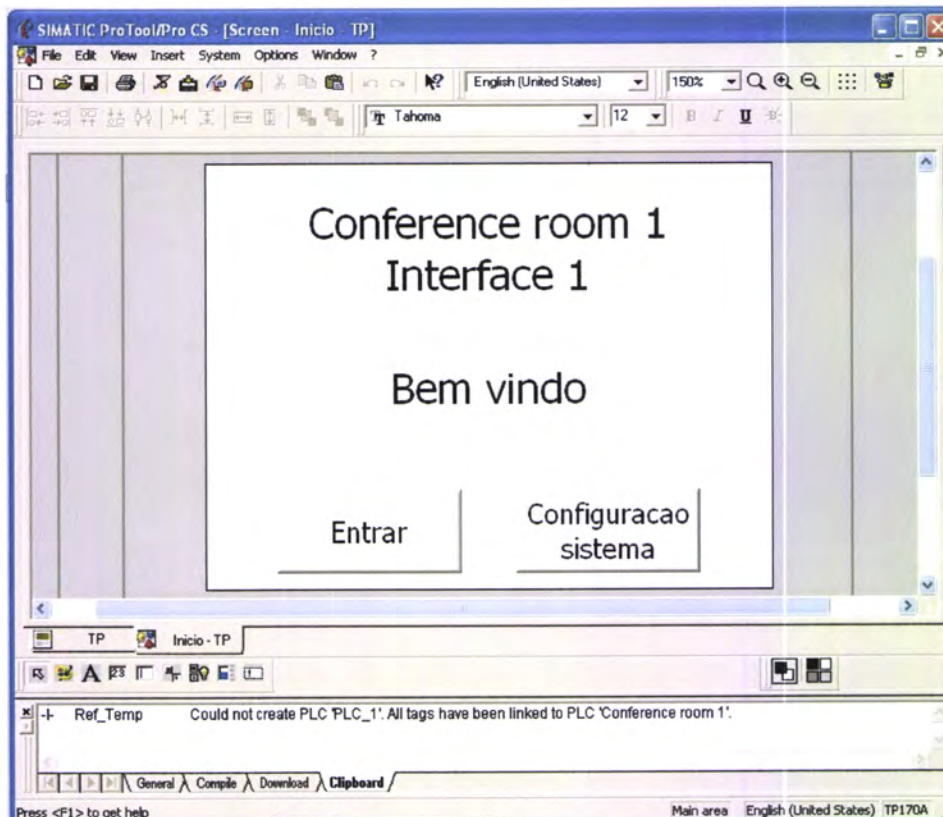


Figura 44: Ecrã "Inicio" (interface HMI)

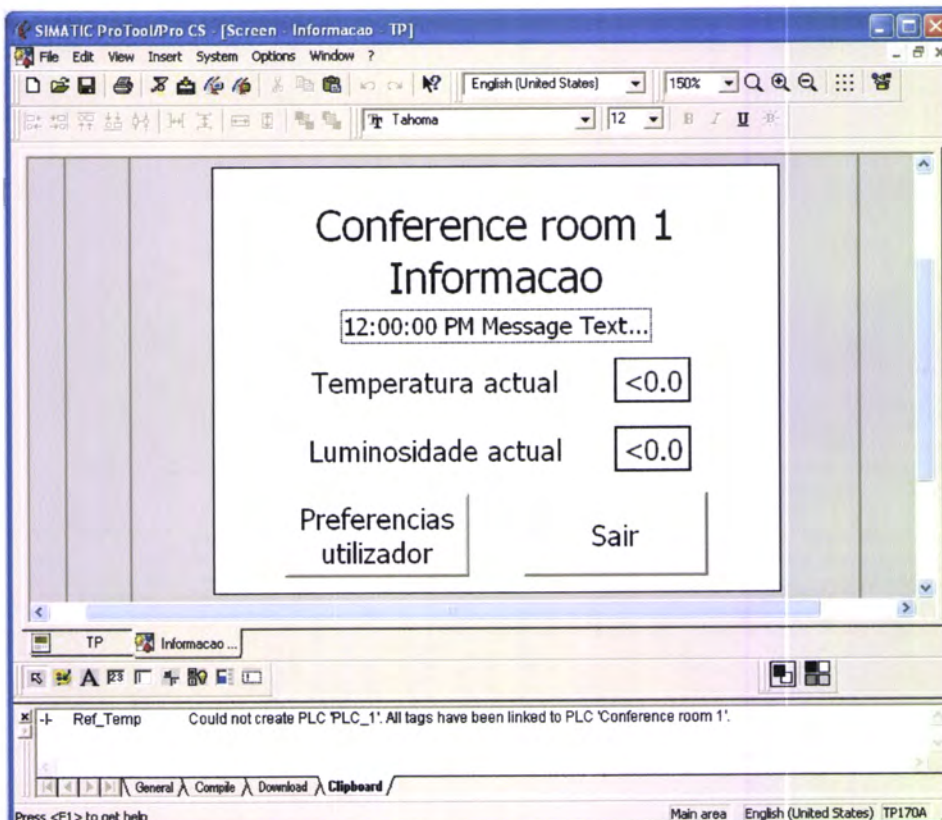


Figura 45: Ecrã "Informação" (interface HMI)

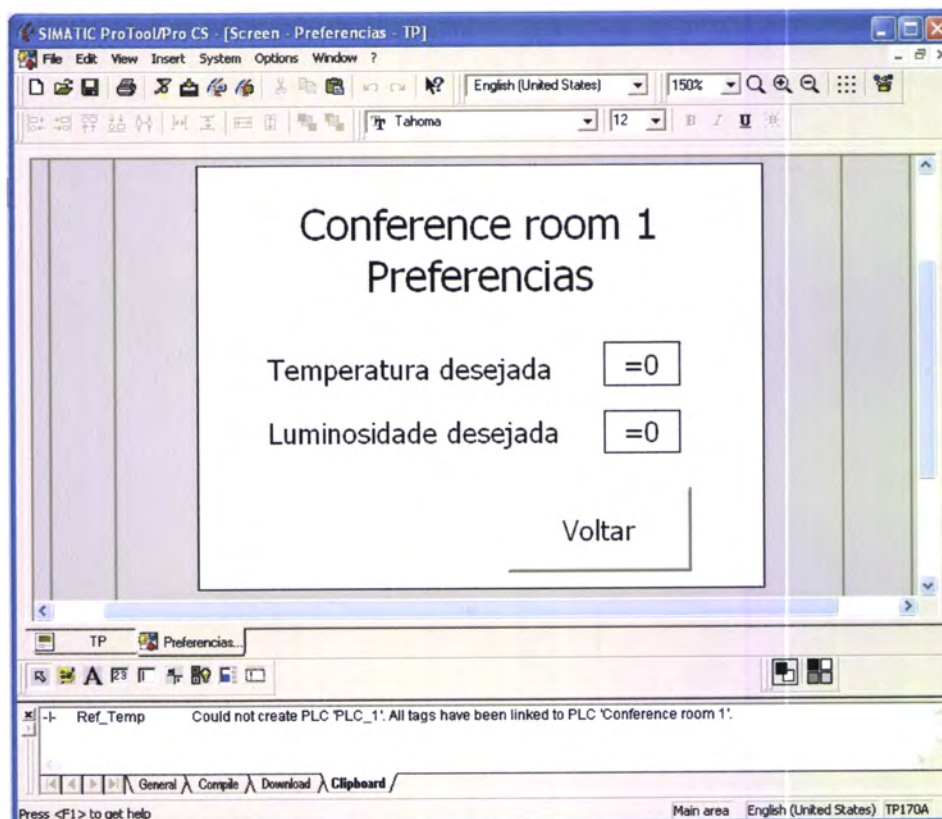


Figura 46: Ecrã "Preferências" (interface HMI)

A informação do projecto desenvolvido no ProTool encontra-se no anexo F.

5.4 Controlo remoto e supervisão (SCADA)

5.4.1 Principais características

Num nível superior ao do controlo local temos o controlo remoto e supervisão do sistema. As duas acções são levadas a cabo por um sistema SCADA [SIEMENS, 2008], no entanto devido à limitação de controlo do SCADA é utilizado um software de computação matemática que ligado ao SCADA fornece o controlo pretendido. Conclui-se portanto que o SCADA na realidade irá apenas supervisionar toda a plataforma interactiva.

As principais características de um sistema SCADA são:

- Interface gráfica amigável;
- Supervisão automatizada de PLC's;
- Visualização de eventos e alarmes;
- Possibilidade de interligação com vários tipos de software: logísticos, controladores, análise de dados;
- Grau de segurança elevado;
- Fácil manutenção;
- Fiabilidade reconhecida.

Em sistemas com vários PLC's a informação fornecida por cada um é apresentada de uma forma desagradável para a visualização pelo utilizador, deste facto surgiu a necessidade de se criar um interface gráfica que torna-se agradável a visualização da informação. De uma forma mais pormenorizada, a interface gráfica de um sistema SCADA consiste numa série de "telas" que estão disponíveis ao utilizador de modo que este visualize o estado do sistema completo. Estas "telas" fornecem de uma forma agradável à vista toda a informação que chega ao SCADA, proveniente dos PLC's ou de outros equipamentos.

A ocorrência de uma falha num sistema de controlo pode provocar custos elevados e em certos casos até morte de pessoas. É por estes motivos que a supervisão automatizada, a visualização de eventos, os alarmes disponíveis e a elevada segurança dos sistemas SCADA são de uma importância extrema. Derivado da necessidade de tais níveis de segurança, os sistemas SCADA foram desenvolvidos para suportarem ambientes agressivos, onde podem estar sujeitos a condições extremas ao nível de temperatura, tensão de alimentação e vibração. Outra forma de garantir segurança ao sistema parte do facto de existir uma redundância de hardware e dos canais de comunicação. Assim, no caso de uma estação SCADA falhar, essa falha é rapidamente detectada e a estação substituída imediatamente por uma estação de backup. Outra vantagem é que todo este processo de substituição de uma estação ocorre sem que o funcionamento do sistema tenha de ser interrompido.

A fiabilidade destes sistemas é de tal forma elevada que alguns fabricantes estimam o tempo médio entre falhas na ordem das centenas de anos.

O sistema SCADA que foi implementado nesta dissertação contém aspectos comuns aos que foram referidos anteriormente. Existe uma interface gráfica amigável que consiste em várias telas que fornecem a informação das condições actuais, das preferências dos utilizadores que se encontram na sala, assim como das ordens de funcionamento dadas pelo controlo preditivo à unidade de fornecimento.

Por se tratar de um projecto com uma implementação laboratorial de dimensões reduzidas, as necessidades de segurança da plataforma interactiva não foram consideradas de uma importância extrema, portanto não se verifica a existência de estações de backup ou redundância de canais de comunicação. Caso a plataforma venha a ser aplicada num edifício as redundâncias referidas devem ser aplicadas.

A comunicação entre os PLC's e o sistema SCADA é efectuada através do protocolo de comunicação OPC

Todas as “tags” utilizadas no sistema SCADA estão presentes no anexo G.

5.4.2 Configuração do sistema SCADA

1. Depois de iniciar o software **WinCC** carregar no ícone de criação de um novo projecto.
2. Definir o tipo de projecto, de seguida o nome do projecto e o local de gravação. O projecto criado foi do tipo “single-user project” e o nome atribuído “proj2_v1”.

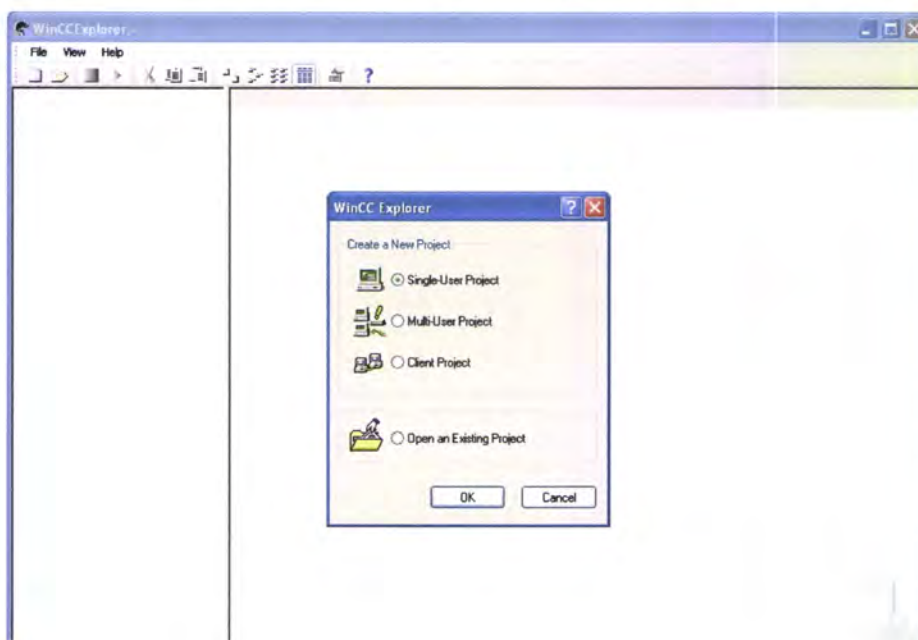


Figura 47: Selecção do tipo de projecto

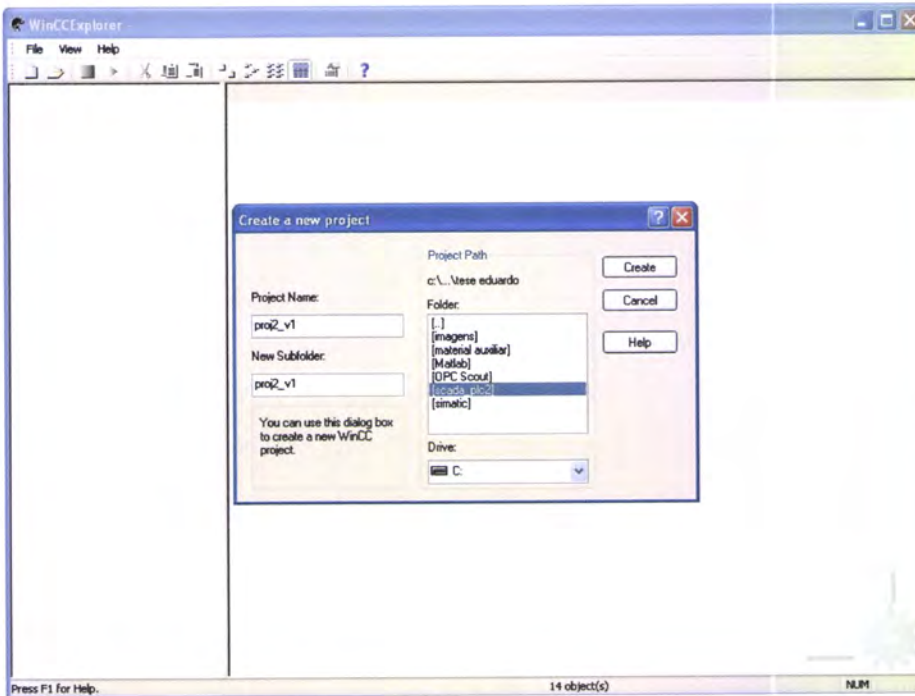


Figura 48: Criação do projecto

3. Adicionar “tags” no WinCC é mais complicado do que no ProTool porque a comunicação do WinCC com o sistema é feita através do protocolo OPC. O primeiro passo é adicionar o “channel” OPC, para tal clica-se com o botão direito do rato em “Tag Management” e depois “Add new driver”.

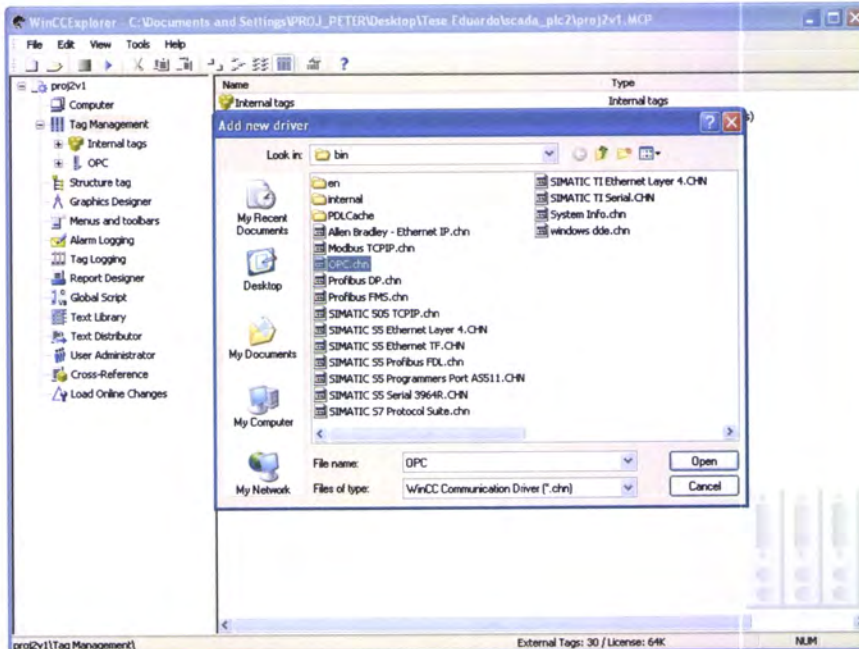


Figura 49: Adição do “channel”

De seguida é necessário adicionar uma “New Driver Connection”, clicando com o botão direito do rato em “OPC Groups”. Atribui-se o nome, selecciona-se o servidor e clica-se em propriedades. Nas propriedades introduz-se o “OPC Server Name” e a indicação para que o servidor corra noutro computador.

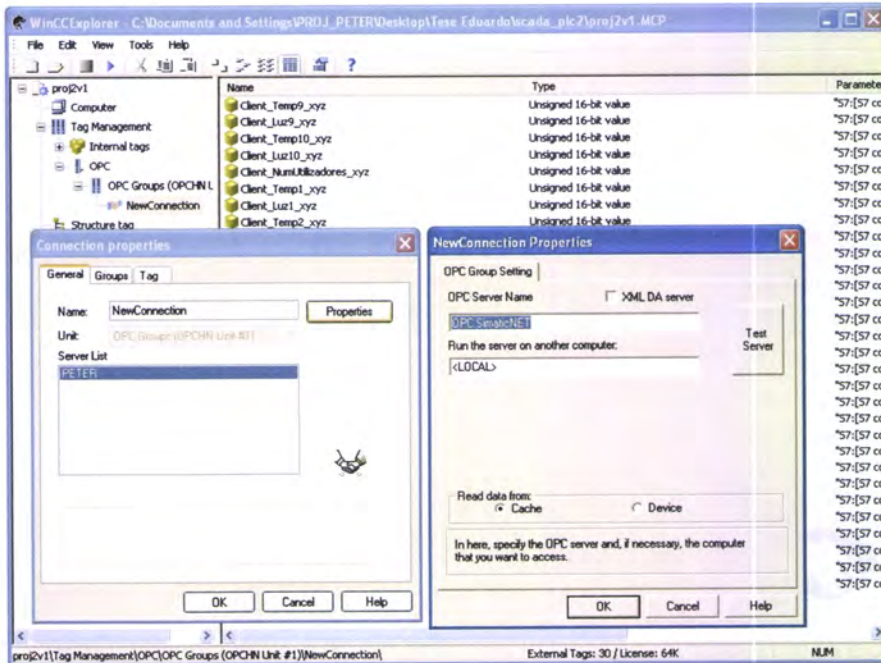


Figura 50: Adição de uma "Driver Connection"

Carrega-se novamente com o botão direito do rato em “OPC Groups”, mas desta vez selecciona-se “System Parameter”. Surge uma janela “OPC Item Manager”, expandindo o item “\\<LOCAL>” selecciona-se “OPC.SimaticNET” e clica-se em “Browse Server”.

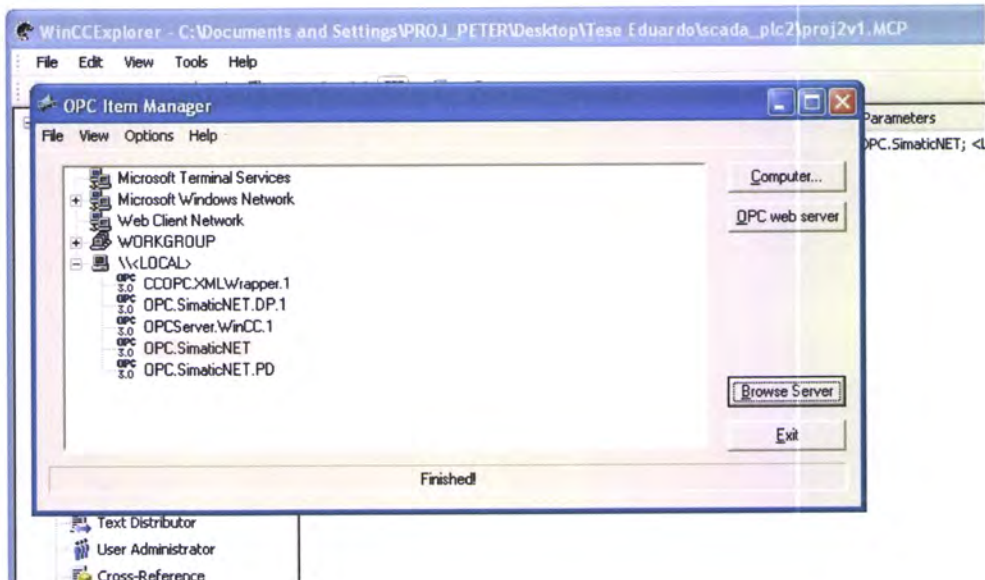


Figura 51: Adicionar tags (Passo 1)

Aparece uma pequena janela onde se clica “Next”, que fará surgir uma nova janela. Nesta janela tem de se expandir o item como se pode observar na figura 52, surgem todas as “tags” existentes na conexão seleccionada. Para adicionar as “tags” à “Driver Connection” criada é necessário selecciona-las e carregar em “Add Items”.

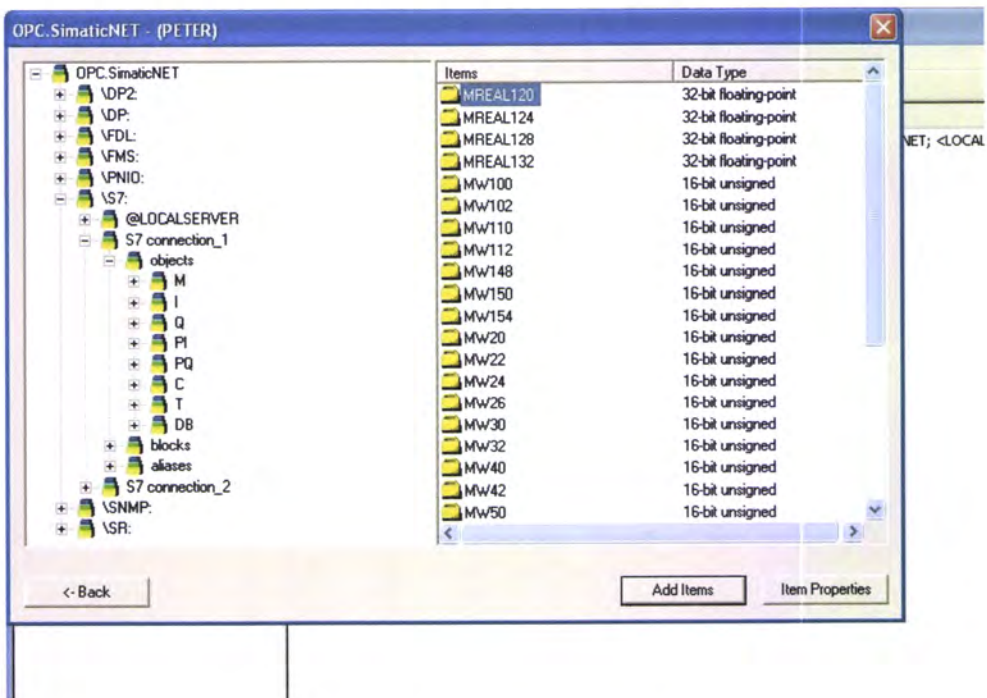


Figura 52: Adicionar tags (Passo 2)

Abre-se nova janela onde é preciso indicar o “prefix” e “suffix” da tag e a “Driver Connection” onde a tag será adicionada.

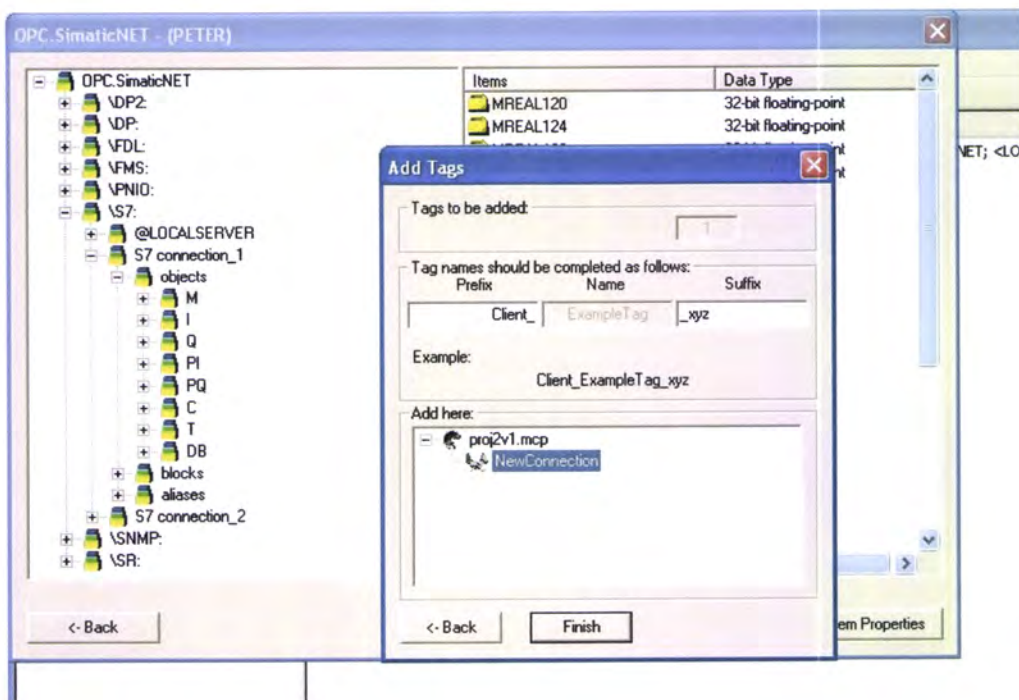


Figura 53: Adicionar tags (Passo 3)

Name	Type	Parameter
Client_Temp9_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Luz9_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Temp10_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Luz10_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_NumUtilizadores_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Temp1_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Luz1_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Temp2_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Luz2_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Temp3_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Luz3_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Temp4_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Luz4_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Temp5_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Luz5_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Temp6_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Luz6_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Temp7_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Luz7_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Temp8_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Luz8_xyz	Unsigned 16-bit value	"S7:(S7 coi
Client_Start_xyz	Binary Tag	"S7:(S7 coi
Client_Med_Ref_Temp_xyz	Floating-point number 32-bit IEEE 754	"S7:(S7 coi
Client_Med_Ref_Lum_xyz	Floating-point number 32-bit IEEE 754	"S7:(S7 coi
Client_Temp_Act_xyz	Floating-point number 32-bit IEEE 754	"S7:(S7 coi
Client_Lum_Act_xyz	Floating-point number 32-bit IEEE 754	"S7:(S7 coi
Client_TempAVAC_xyz	Floating-point number 32-bit IEEE 754	"S7:(S7 coi
Client_LuzARTF_xyz	Floating-point number 32-bit IEEE 754	"S7:(S7 coi
Client_AberEST_xyz	Floating-point number 32-bit IEEE 754	"S7:(S7 coi
Client_PercAberEST_xyz	Floating-point number 32-bit IEEE 754	"S7:(S7 coi

Figura 54: Tags adicionadas no WinCC

- O próximo passo é a criação dos ecrãs que permitem ligar e desligar o sistema, visualizar as preferências dos utilizadores presentes na conference room, a informação de funcionamento do sistema e as condições actuais de temperatura e luminosidade.

Num novo ecrã é possível introduzir vários campos que se encontram disponíveis ao utilizador. Tais como:

- **Campos de texto** (*Text Field*) onde são visíveis mensagens de texto fixas, podendo estas serem alteradas apenas através da programação do ecrã.
- **Botões** (*Button*) que executam tarefas como mudança de ecrã, alteração de valores de “tags” ou ainda funções específicas. Um exemplo de uma dessas funções é a mudança de cor do próprio botão quando este é pressionado.
- **Campos de entradas/saídas** (*Input/Output field*) onde se podem visualizar os valores das “tags”. Um campo de saída apenas fornece o valor da “tag”, enquanto um campo de entrada permite a alteração do valor da “tag” associada.
- **Janelas de figuras** (*Picture Windows*) nas quais é possível alternarem vários ecrãs consoante o pedido do utilizador.

O primeiro ecrã criado é composto por uma janela de figuras e quatro botões. A figura 55 corresponde à configuração da janela de figuras, tendo sido definido o *Menu inicial* como o ecrã que surge ao arrancar o sistema.

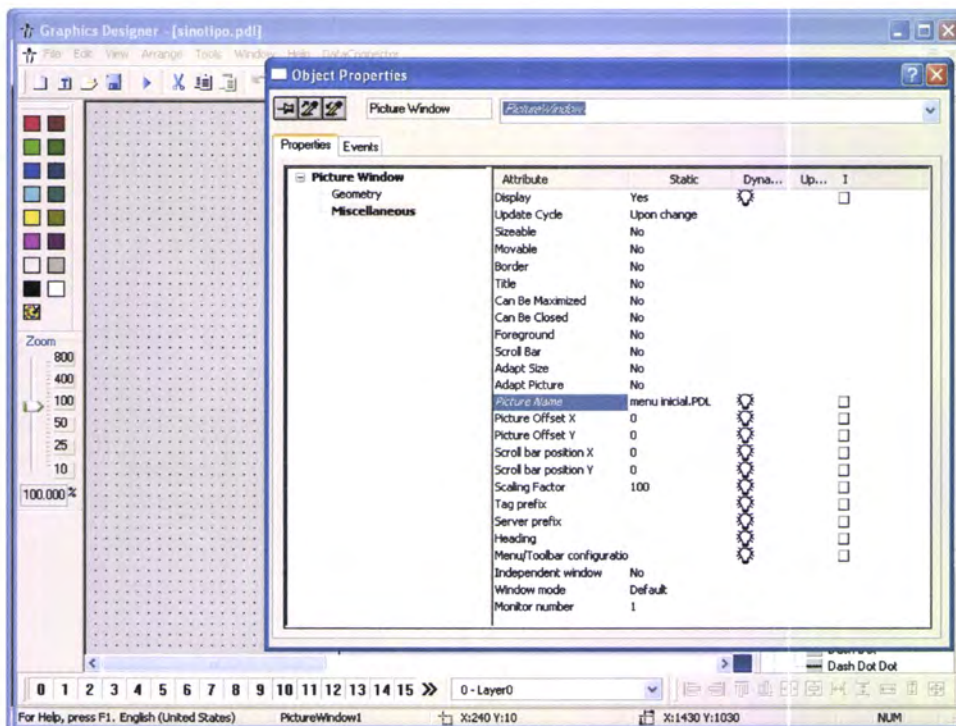


Figura 55: Configuração da janela de figuras

Os quatro botões permitem alterar o ecrã que surge na janela de figuras, mediante o que foi configurado no botão e que se pode observar nas figuras 56 a 59.

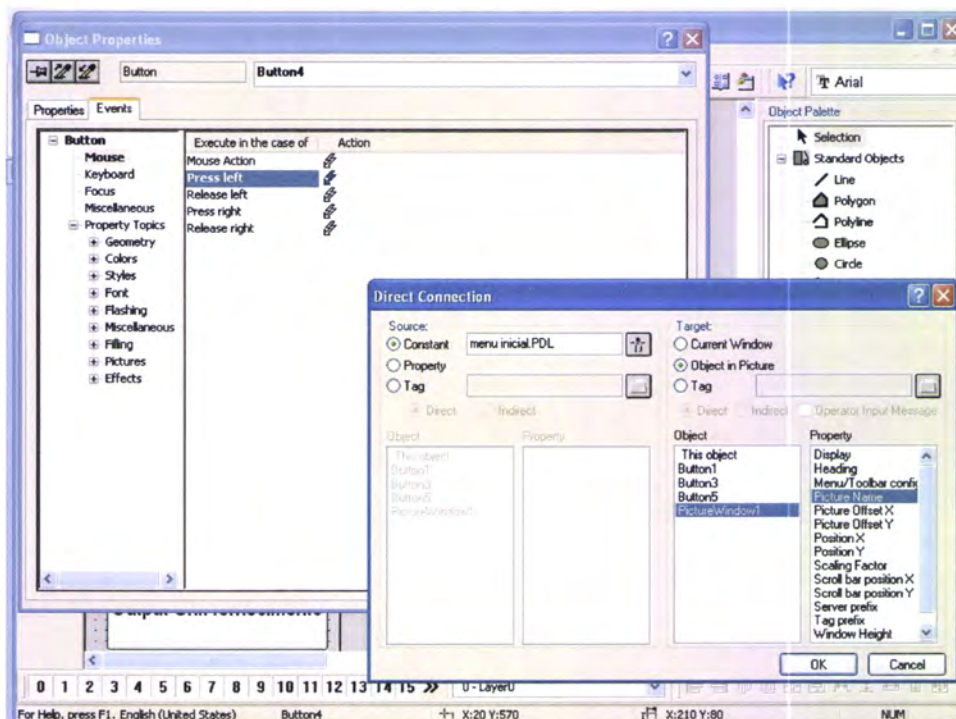


Figura 56: Configuração do botão "Menu inicial"

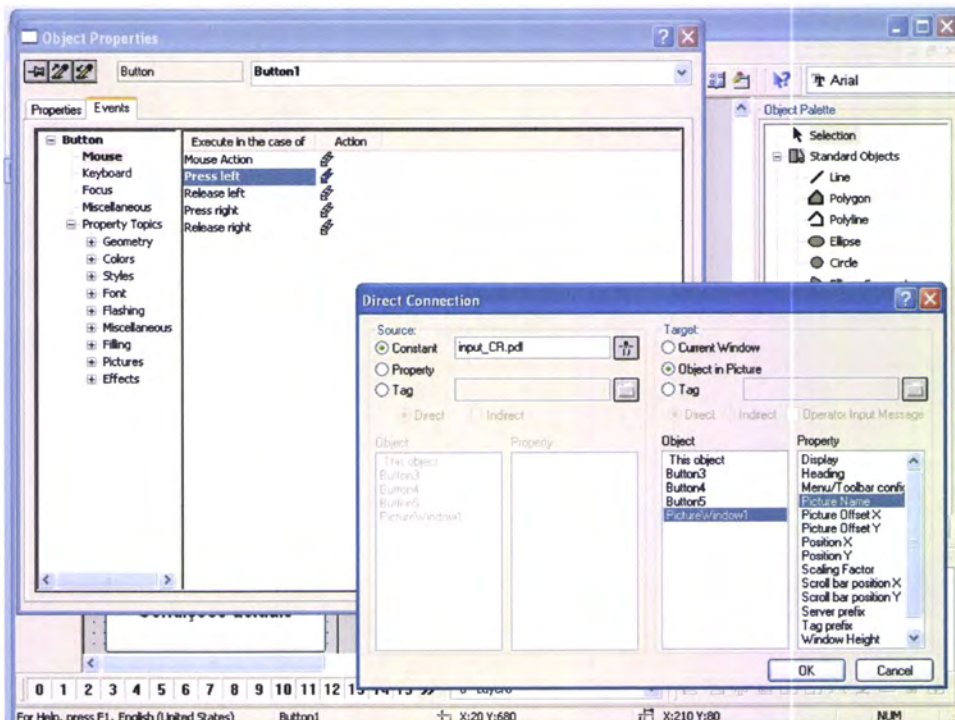


Figura 57: Configuração do botão "Input Conf. room"

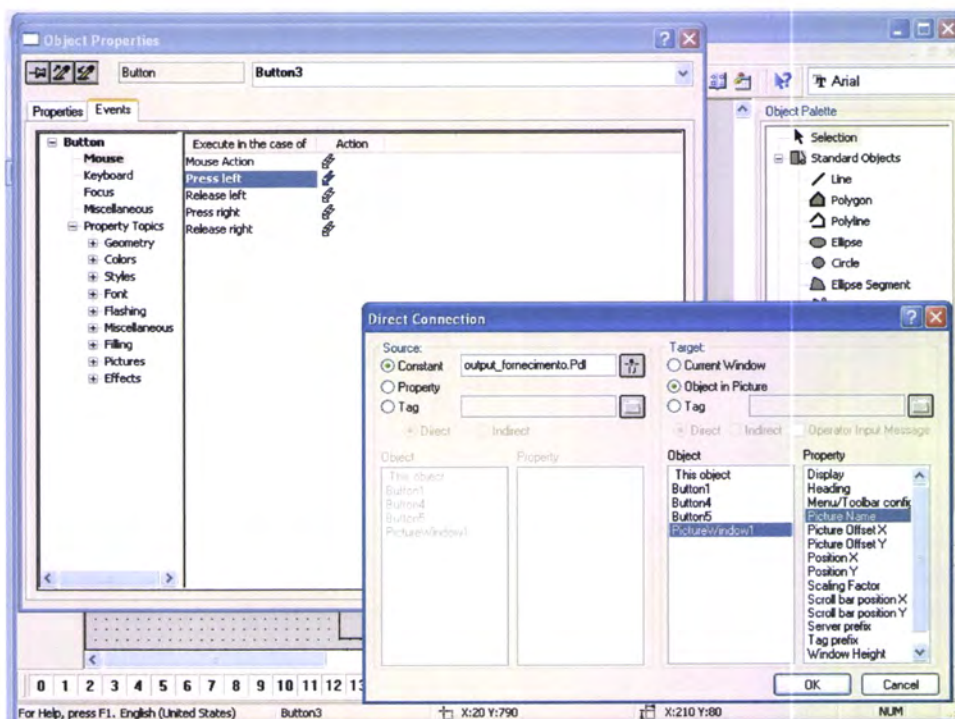


Figura 58: Configuração do botão "Output Uni. fornecimento"

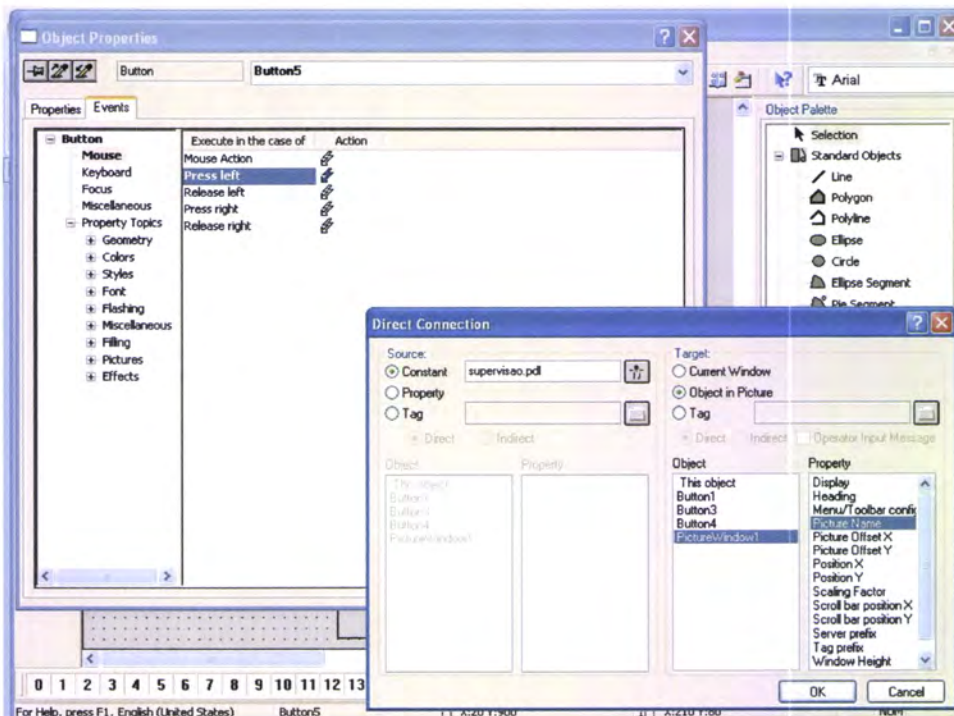


Figura 59: Configuração do botão "Condições actuais"

O ecrã *Menu inicial* é composto por um campo de texto e dois botões. Das figuras 60 à 63 é exemplificado a configuração dos dois botões. O botão *ligar sistema* tem como função mudar o valor, de zero para um, de uma "tag Client_Start_xyz". O botão encontra-se ainda configurado para ficar verde quando esta alteração na "tag" é efectuada, sempre que a "tag" volta a zero o botão volta à cor cinza.

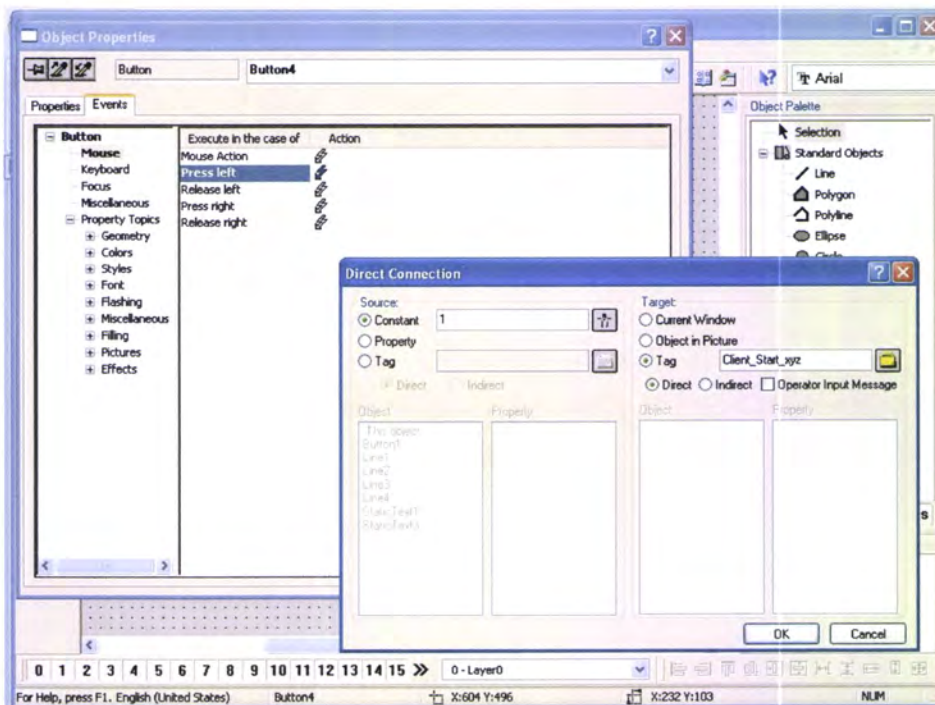


Figura 60: Configuração do botão "ligar sistema" (alteração de tag)

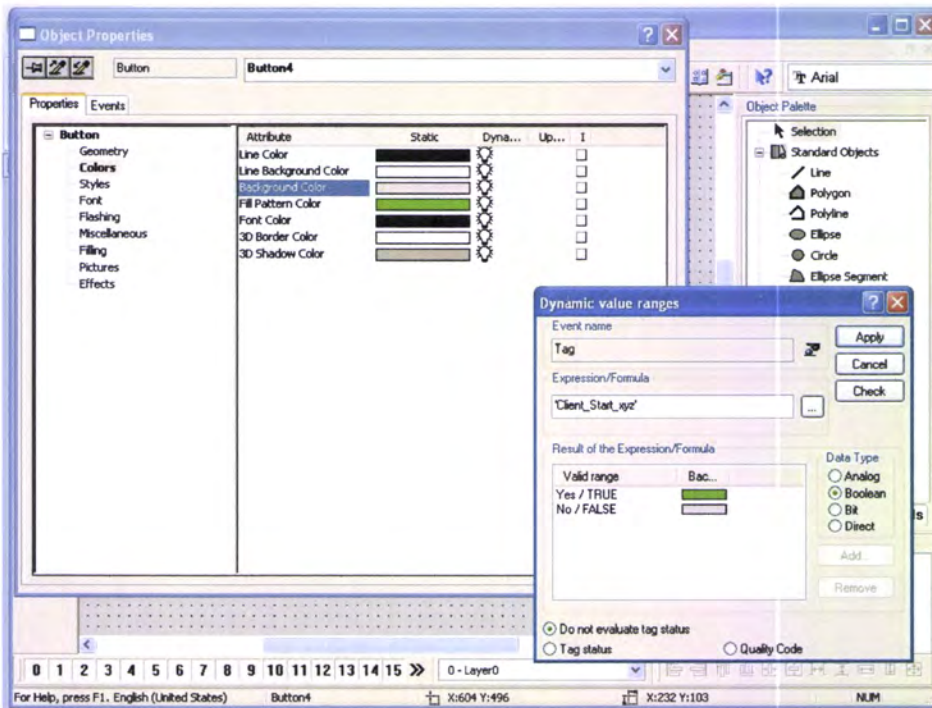


Figura 61: Configuração do botão "ligar sistema" (alteração da cor)

O botão *desligar sistema* tem a função contrária do botão anterior, muda o valor da referida "tag" de um para zero. Em relação à cor este permanece vermelho sempre que a "tag" é zero e muda para cinza se a "tag" passar a valer um.

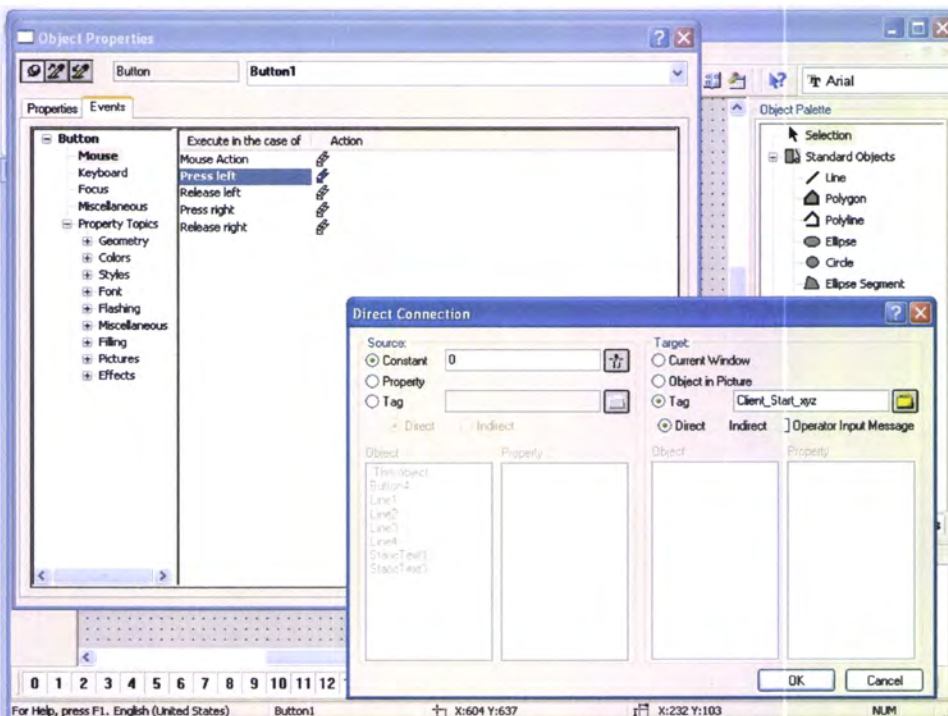


Figura 62: Configuração do botão "desligar sistema" (alteração de tag)

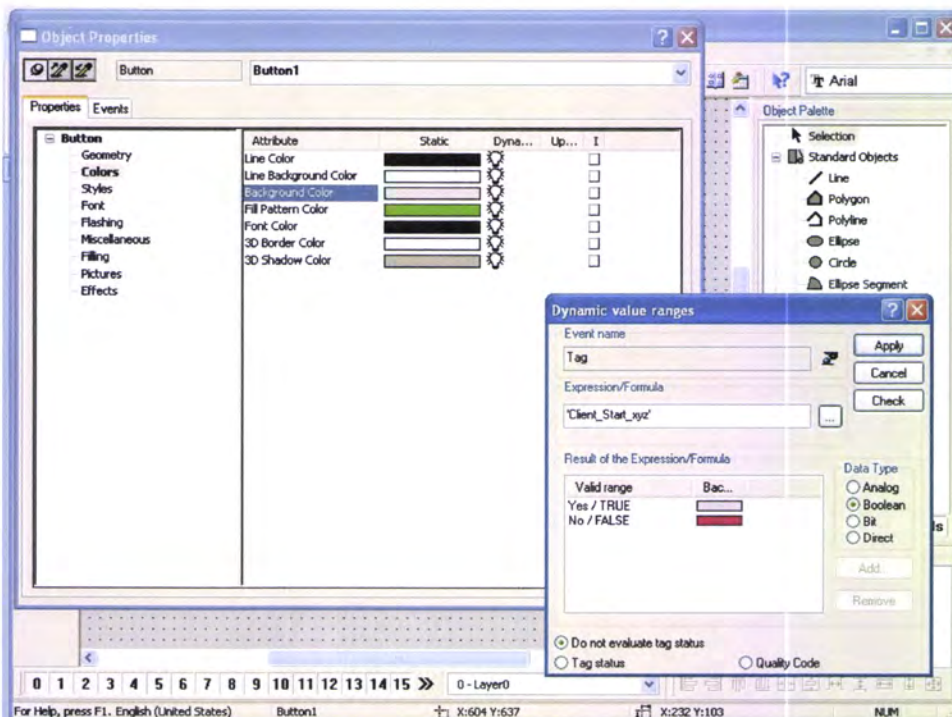


Figura 63: Configuração do botão "desligar sistema" (alteração da cor)

O menu *Input Conference room* é constituído por treze campos de texto e vinte campos que deveriam ser somente de saída, no entanto para facilitar a simulação do sistema os campos são ao mesmo tempo de entrada e saída. Os campos fornecem ao utilizador os valores de referência que cada ocupante da conference room definiu como sua preferência para a temperatura e luminosidade.

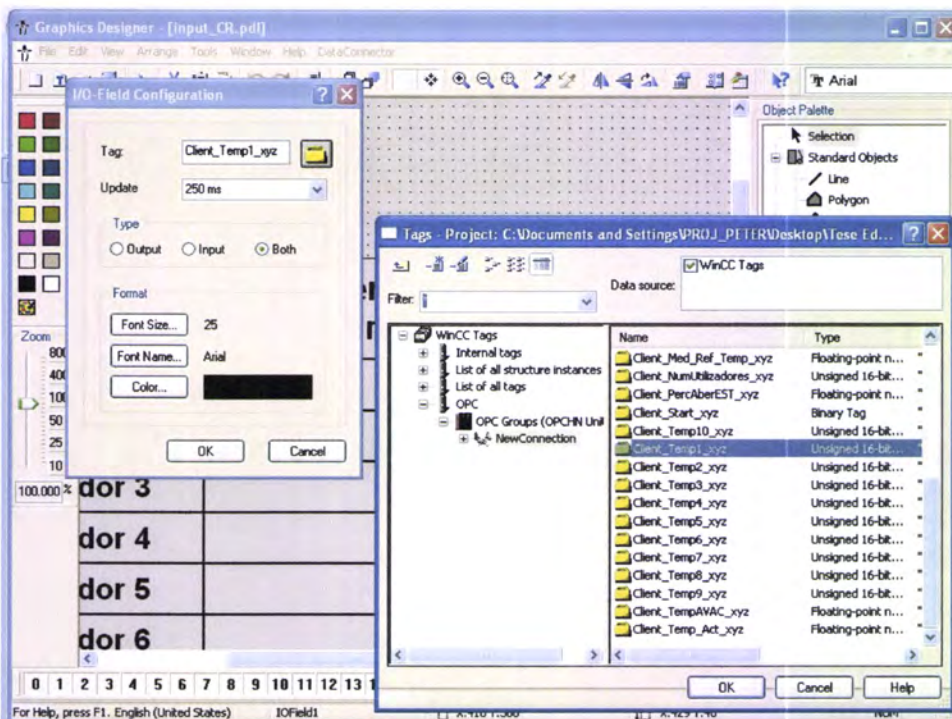


Figura 64: Configuração de um campo de saída

No menu *Output Unidade fornecimento* foram utilizados quatro campos de texto e três campos de saída. Os campos de saída indicam quais os valores que o controlo preditivo se encontra a transmitir referentes à abertura dos estores, à iluminação artificial e à temperatura do AVAC.

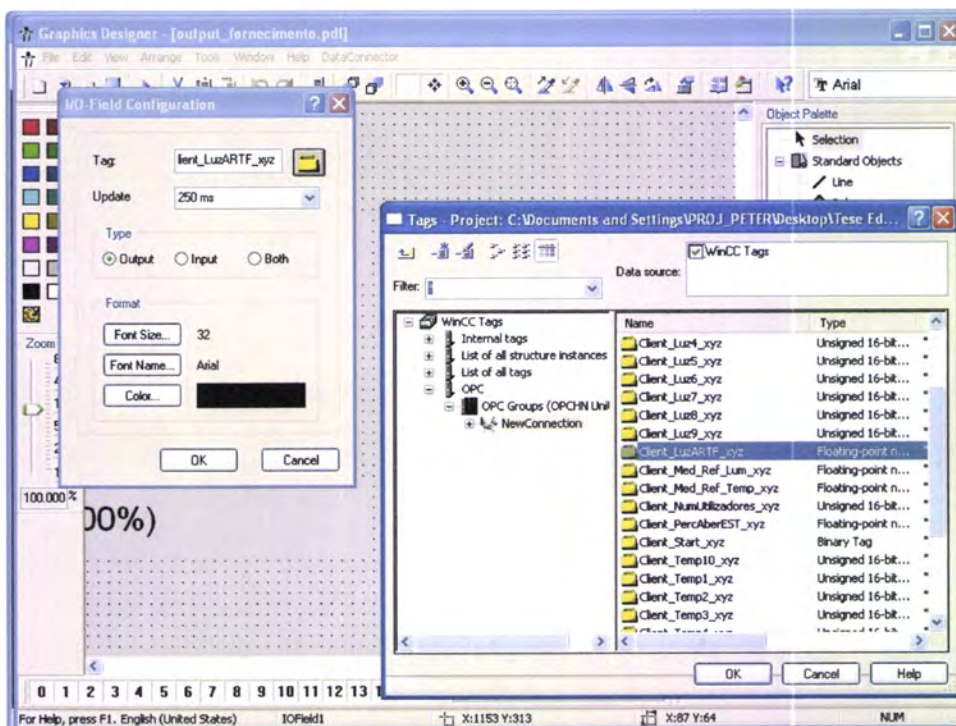


Figura 65: Configuração do campo de saída "Luz artificial"

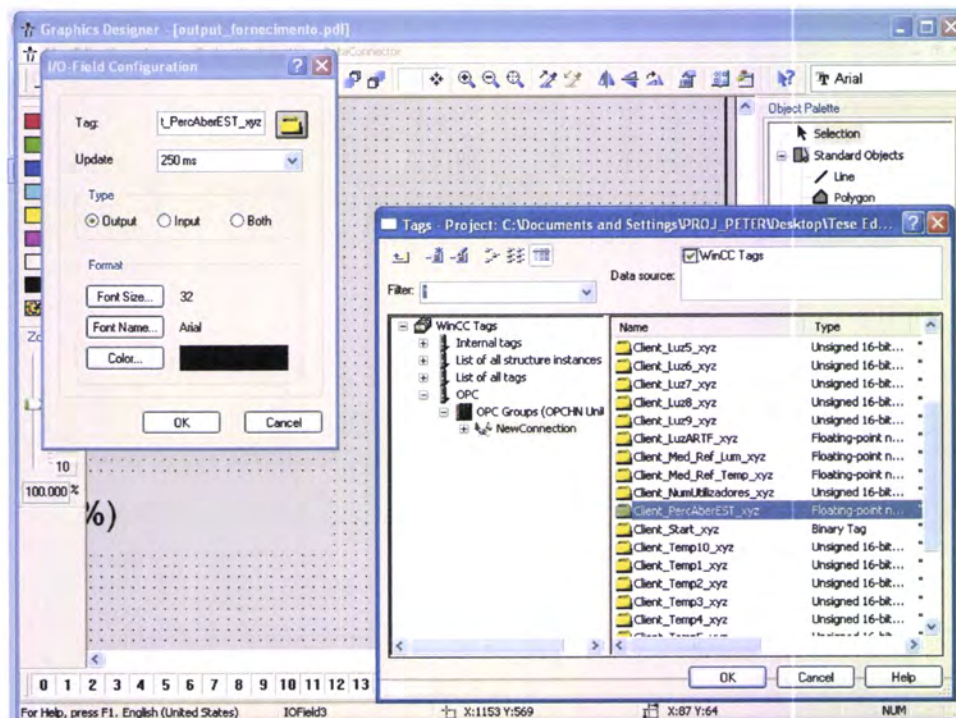


Figura 66: Configuração do campo de saída "Abertura dos estores"

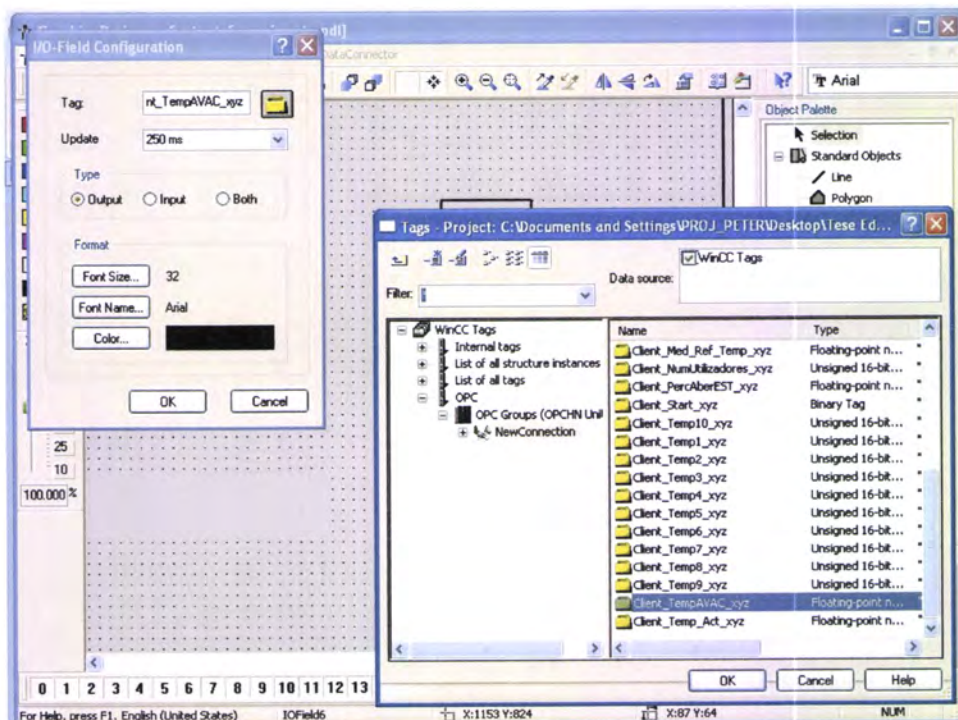


Figura 67: Configuração do campo de saída "Temperatura AVAC"

O menu *Condições actuais* é composto por seis campos de texto e cinco campos de saída. Os campos de saída indicam as condições actuais, temperatura e luminosidade, indicam também o número de utilizadores presentes na conference room e a média das referências fornecidas pelos utilizadores.

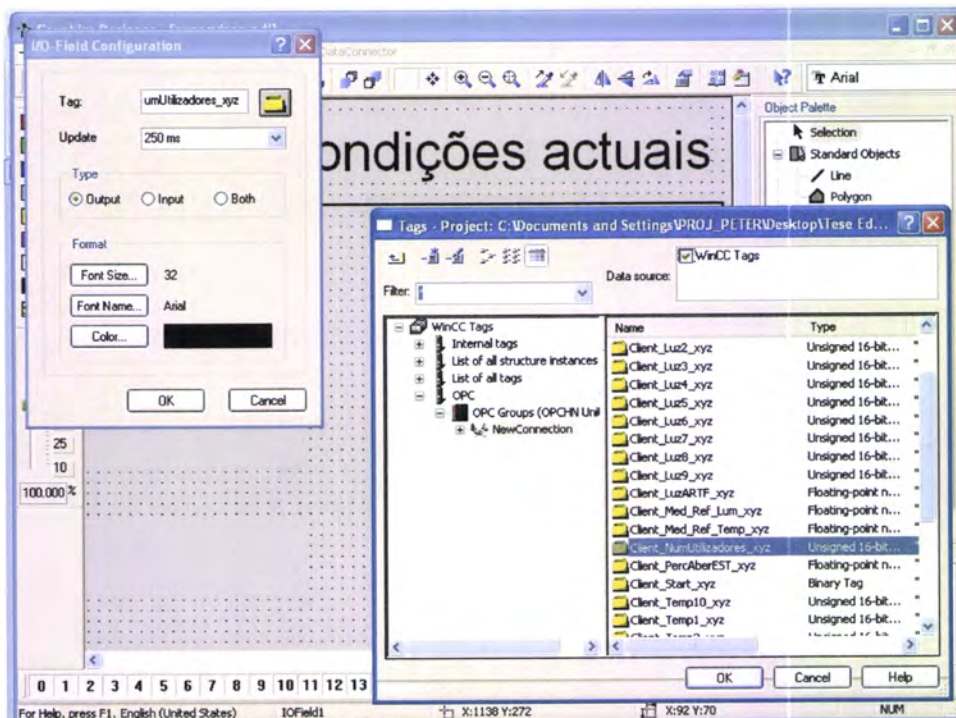


Figura 68: Configuração do campo de saída "Nº utilizadores"

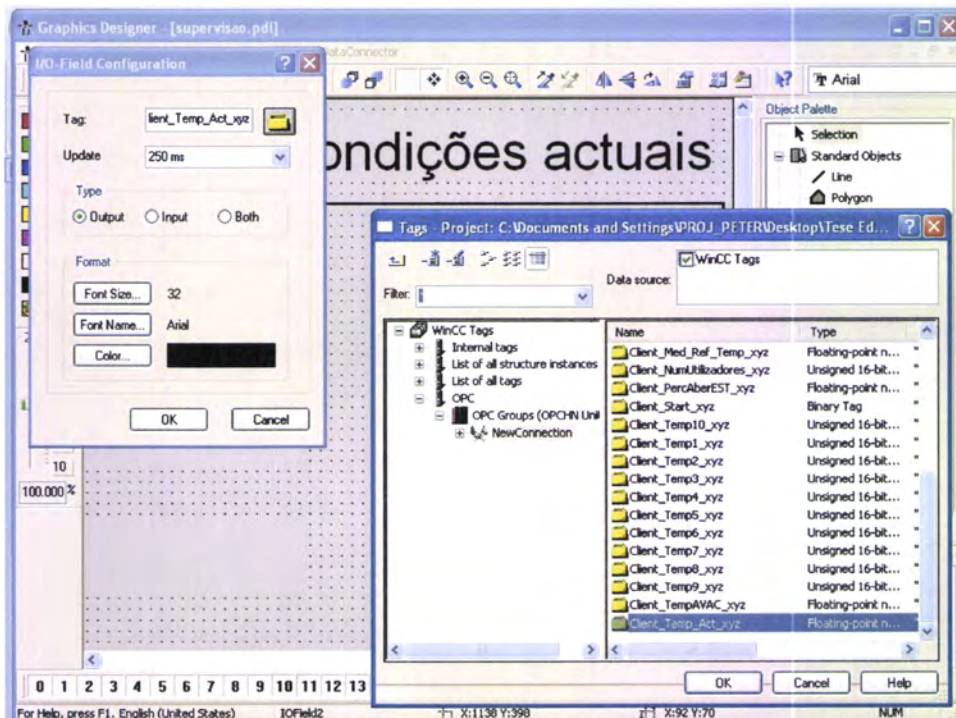


Figura 69: Configuração do campo de saída "Temperatura actual"

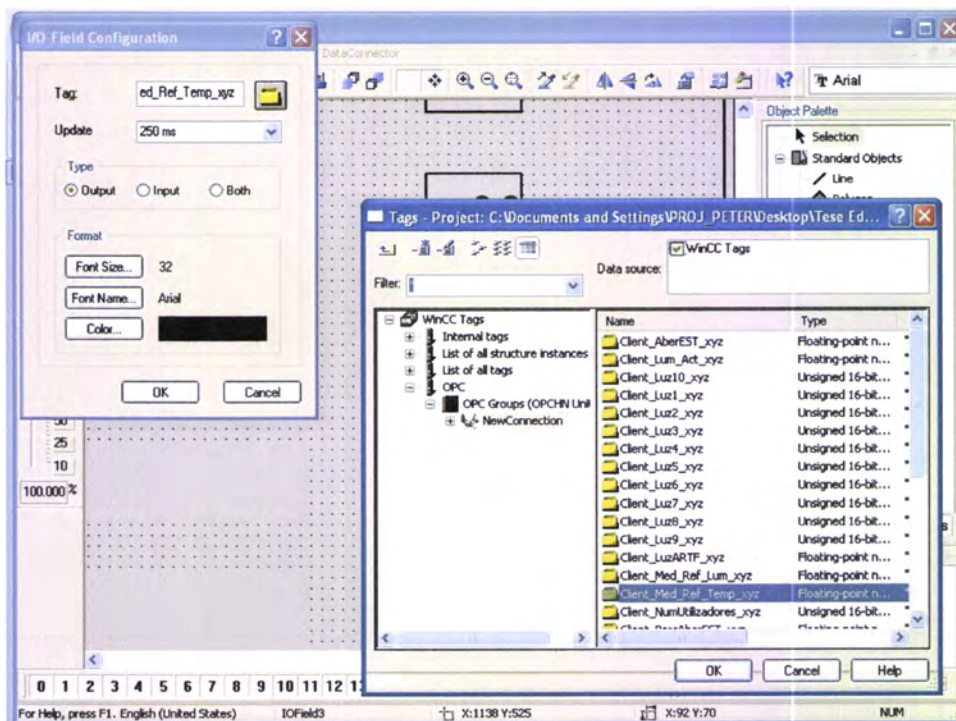


Figura 70: Configuração do campo de saída "Med_Ref_Temp"

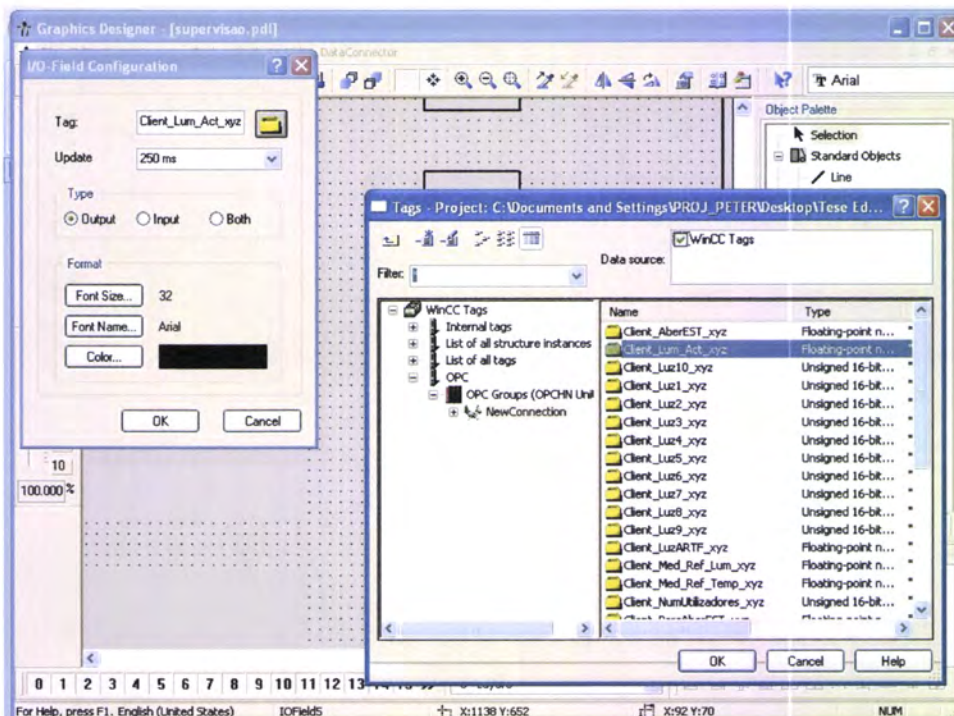


Figura 71: Configuração do campo de saída "Luminosidade actual"

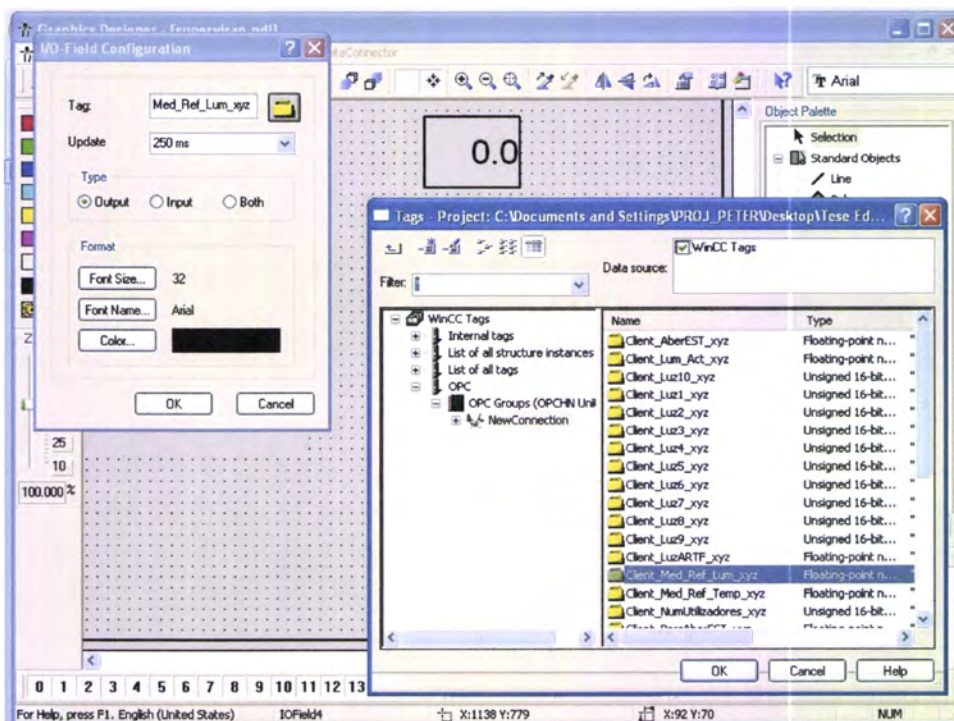


Figura 72: Configuração do campo de saída "Med_Ref_Lum"

As figuras seguintes correspondem aos ecrãs finais criados no sistema SCADA.

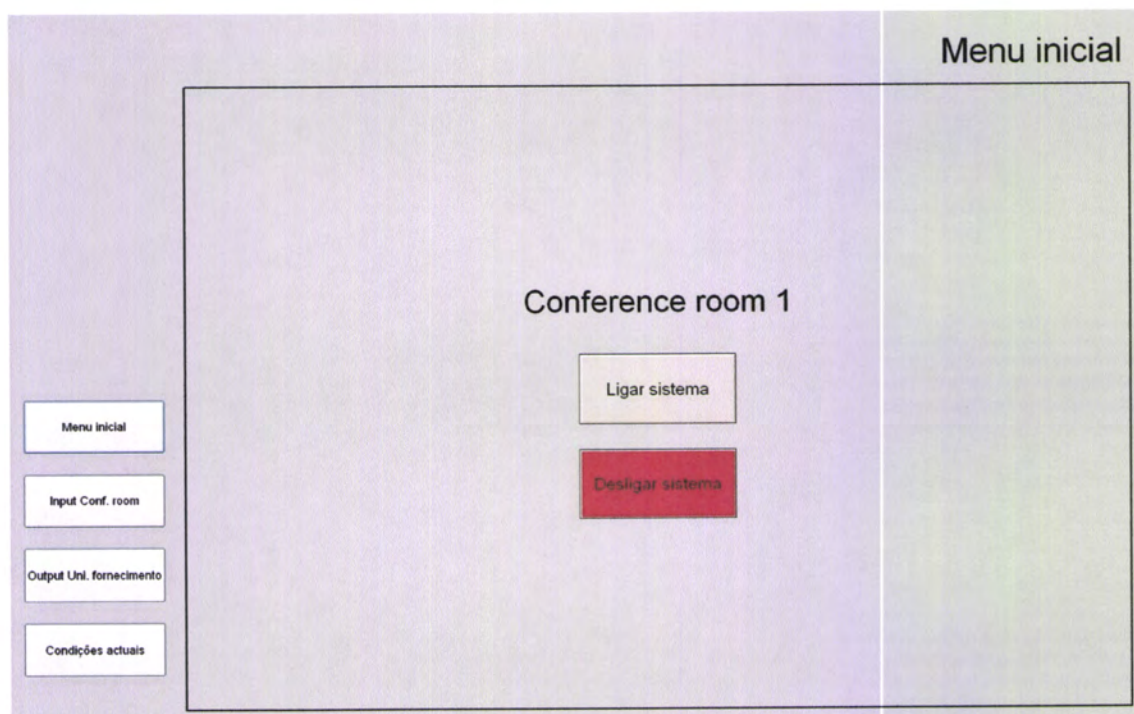


Figura 73: Ecrã "Menu inicial" (SCADA)

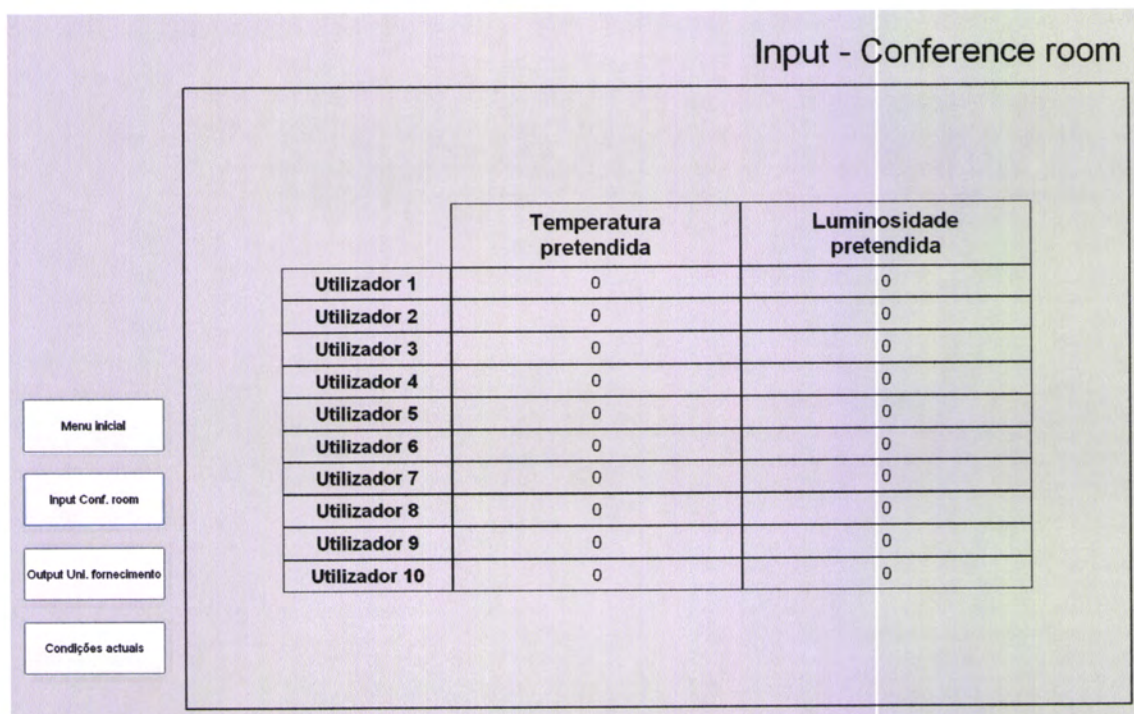


Figura 74: Ecrã "Input conference room" (SCADA)

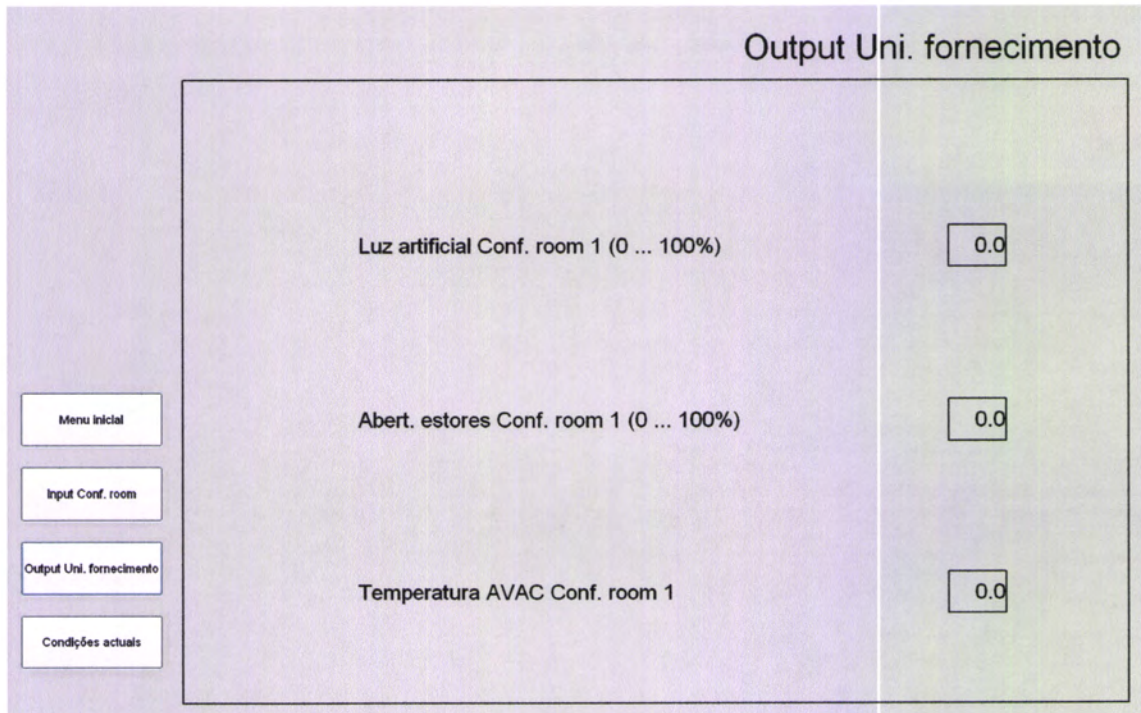


Figura 75: Ecrã "Output Uni. fornecimento" (SCADA)

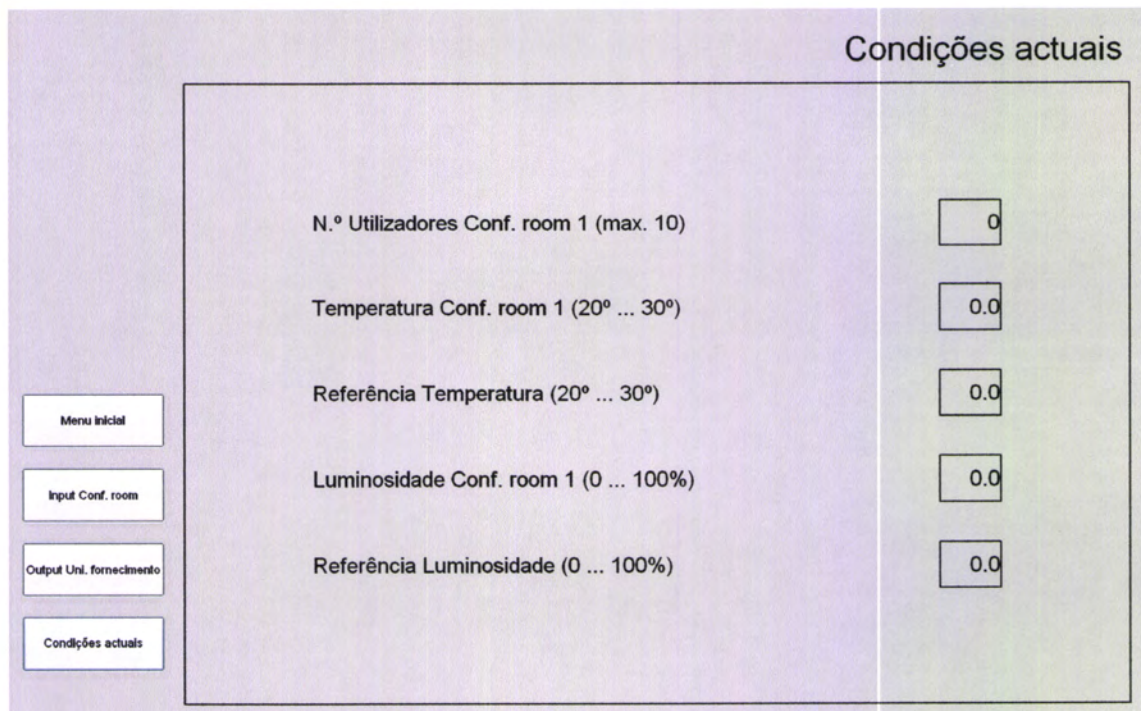


Figura 76: Ecrã "Condições actuais" (SCADA)

5.5 Matlab

Como foi referido anteriormente, os PLC's e o sistema SCADA presentes no sistema não possuem capacidade para implementar um controlo tão complexo como o controlo preditivo. Também já foi referido que para a implementação do controlo preditivo foi escolhido o software Matlab [MathWorks, 2008], que será interligado com o sistema SCADA através do protocolo de comunicação OPC (Object Linking and Embedding (OLE) for Process Control).

5.5.1 Principais características

O Matlab é um software que utiliza uma linguagem de alto nível técnico de computação e um ambiente interactivo para o desenvolvimento de algoritmos, visualização de dados, análise de dados e cálculo numérico [<http://www.mathworks.com/>]. Usando o MATLAB, pode-se resolver problemas técnicos de computação mais rapidamente do que com outras linguagens de programação tradicionais, como C, C++ e Fortran.

O Matlab possui uma ampla gama de aplicações, incluindo processamento de sinal e imagem, comunicação, teste e medição, biologia computacional, entre muitas outras áreas. Conjuntos especiais de funções (toolboxes) permitem ampliar o ambiente Matlab, permitindo a resolução de determinados conjuntos de problemas em áreas de aplicação específicas.

O código Matlab pode ser criado através de sequências de comandos que podem ser guardadas num arquivo de texto criado através do Matlab Editor. Outra hipótese é a utilização do Simulink, uma ferramenta que permite a modelação, simulação e análise de sistemas dinâmicos. O Simulink consiste na criação de sistemas por diagramas de blocos, sendo amplamente usado em sistemas de controlo e processamento digital de sinais.

As principais características do Matlab são:

- Linguagem de alto nível para computação técnica;
- Ambiente de desenvolvimento para gestão de código, arquivos e dados;
- Ferramentas interactivas para exploração iterativa, design e resolução de problemas;
- Funções matemáticas de álgebra linear, estatística, análise de Fourier, optimização e integração numérica;
- Funções gráficas a 2-D e 3-D para visualização de dados;
- Funções para integração de algoritmos Matlab em aplicações externas e linguagens, tais como C, C++, Fortran, Java, COM e Microsoft Excel.

O controlo preditivo presente nesta dissertação foi desenvolvido através da ferramenta Simulink, utilizando a toolbox MPC (Model Predictive Control), a toolbox OPC, e outras funções mais gerais presentes no software. No anexo I encontra-se o ambiente gráfico do controlo preditivo criado no simulink e ainda o ficheiro “.m” que cria as matrizes do sistema em espaço de estados, carrega o controlador desenvolvido e inicia o ficheiro simulink do controlo preditivo.

5.5.2 Configuração do controlo preditivo (Matlab)

Numa primeira fase do projecto equacionou-se o desenvolvimento da plataforma interactiva com duas conference rooms, devido à inexistência de autómatos suficientes no laboratório para a implementação abandonou-se a ideia. No entanto o controlador desenvolvido encontra-se dimensionado de forma a controlar uma segunda conference room, caso um dia venha a ser implementada.

Na imagem apresentada em baixo é possível observar a toolbox MPC e o bloco de parametrização onde se indica o nome do controlador desenvolvido, MPC1.

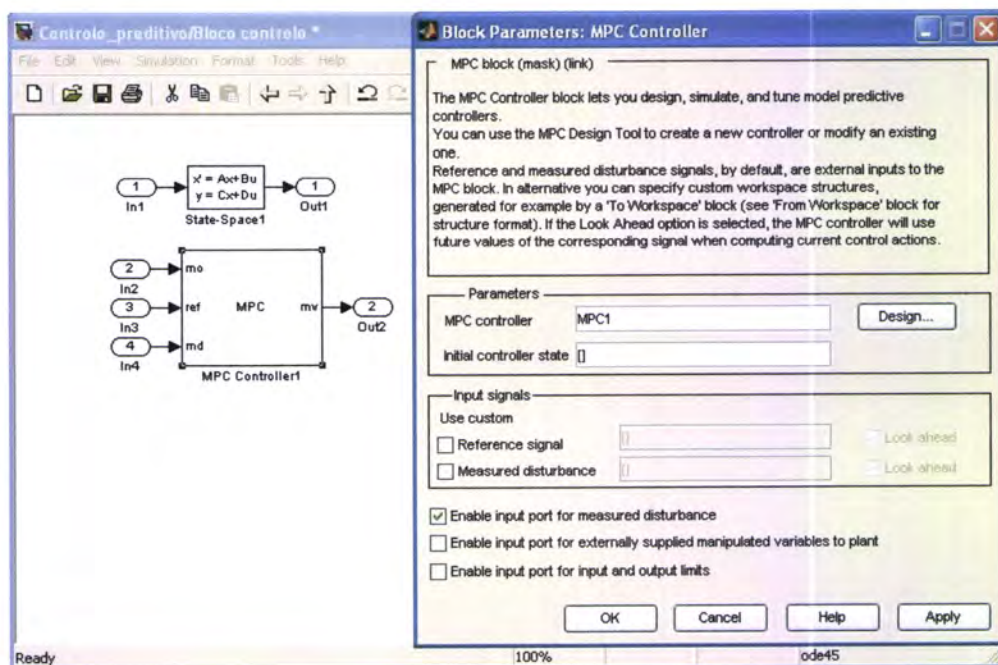


Figura 77: Toolbox MPC

A configuração do controlador foi efectuada através do processo de “tentativa e erro” durante o seu desenvolvimento, até se encontrar o ponto que forneceu os melhores resultados.

A imagem seguinte apresenta a estrutura geral do controlador usado.

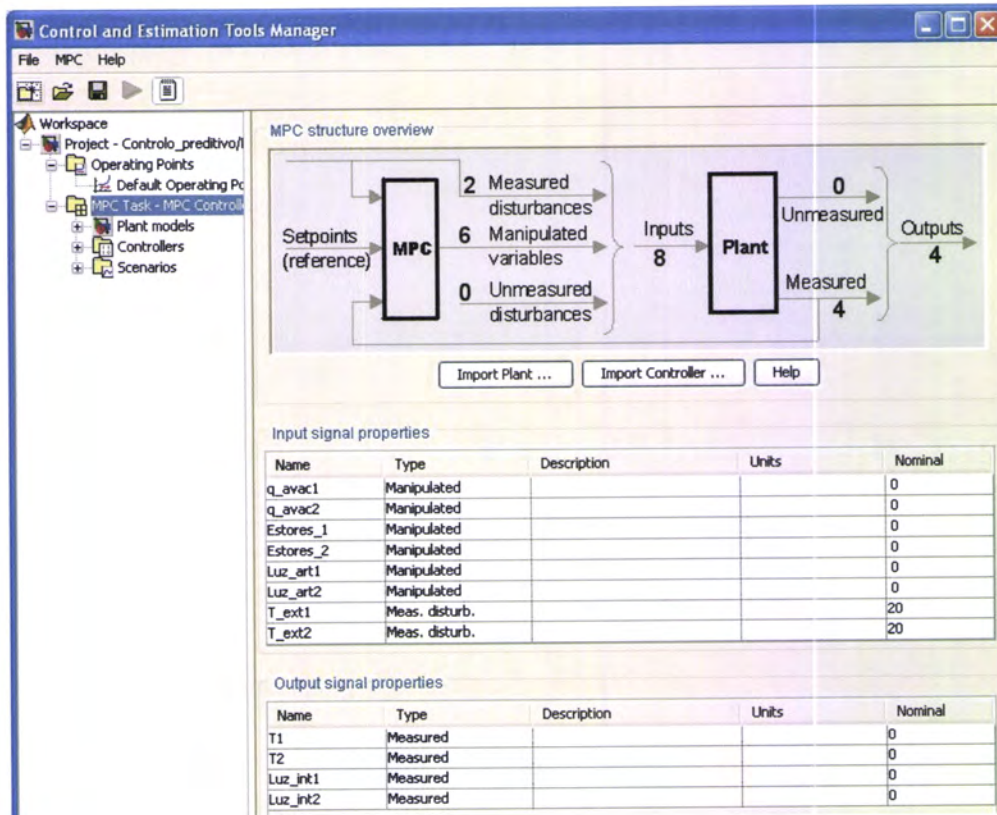


Figura 78: Estrutura geral do controlador

Nas três imagens que são apresentadas de seguida observa-se os horizontes usados no controlador, os constringimentos e os pesos.

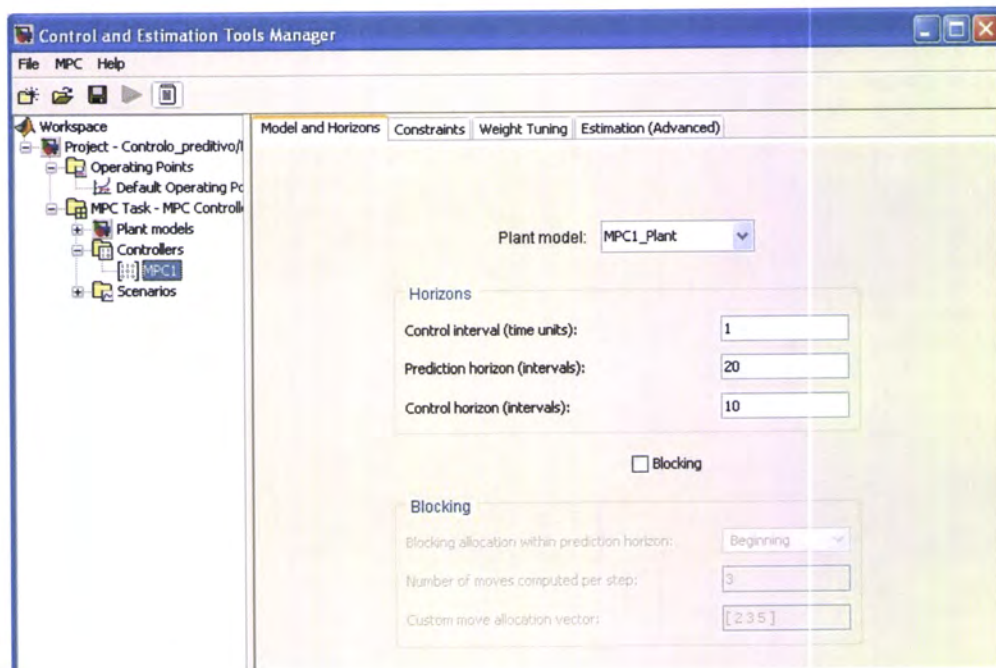


Figura 79: Horizontes do controlador

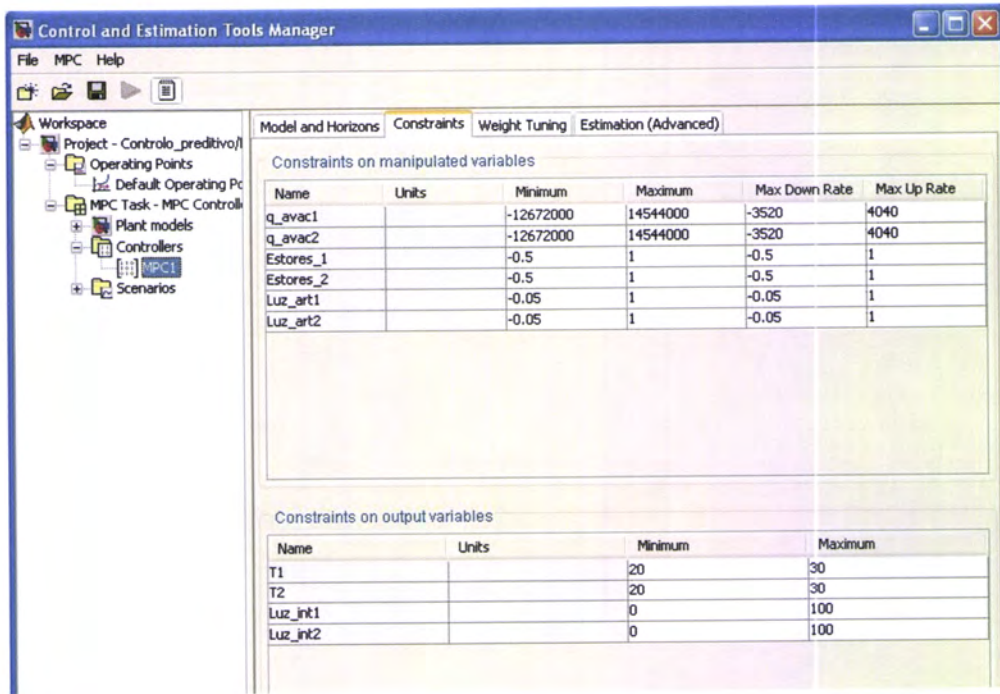


Figura 80: Constrainmentos do controlador

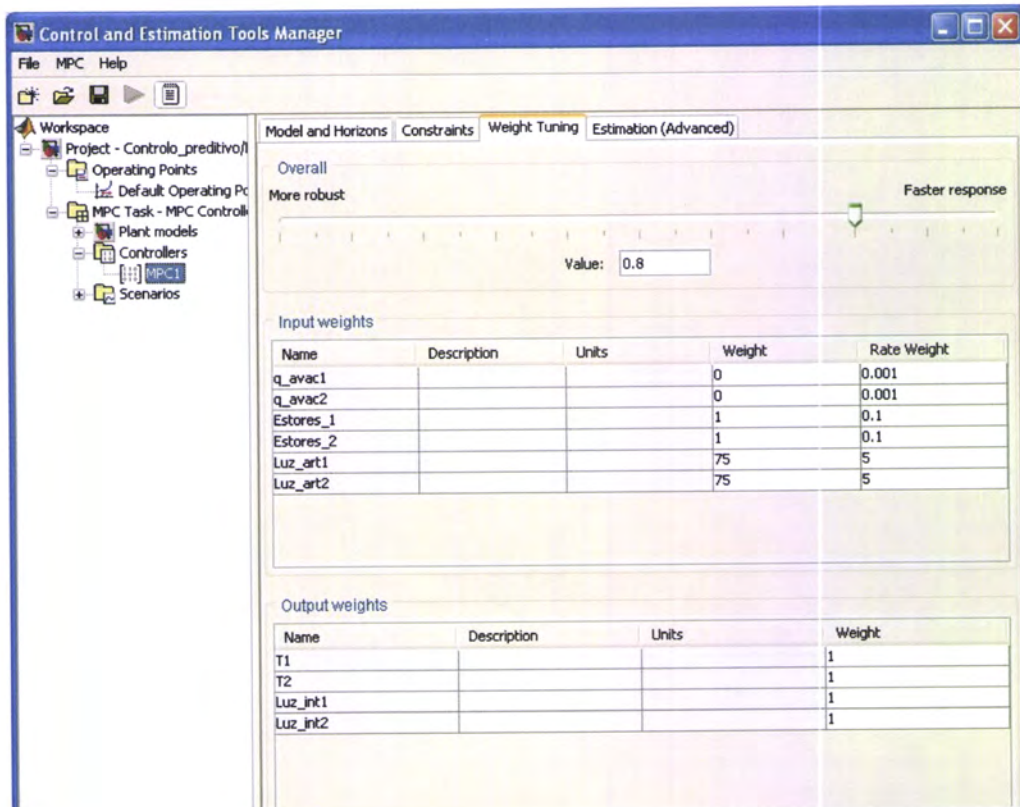


Figura 81: Pesos do controlador

Para que o controlador funcione é necessário que receba informação vinda do sistema, para esse efeito é utilizado o bloco “OPC Read”. Na figura seguinte é apresentado o bloco e a sua configuração.

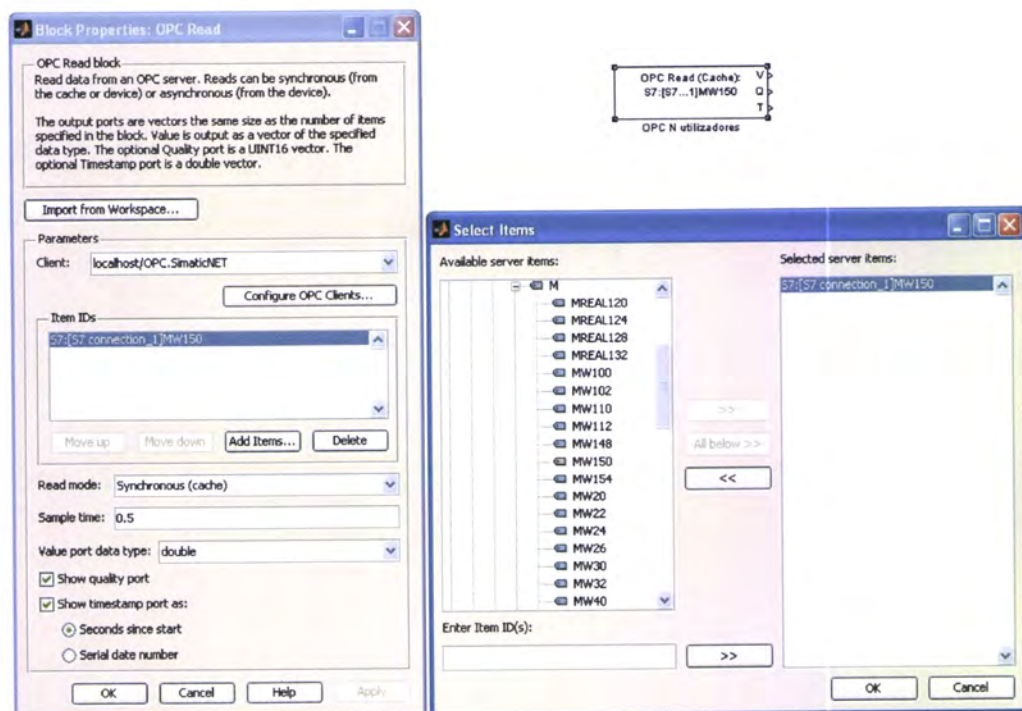


Figura 82: Bloco "OPC Read"

Em sentido inverso seguem as acções de controlo que são comunicadas ao sistema através do bloco “OPC Write” que se pode visualizar na figura seguinte, bem como a sua configuração.

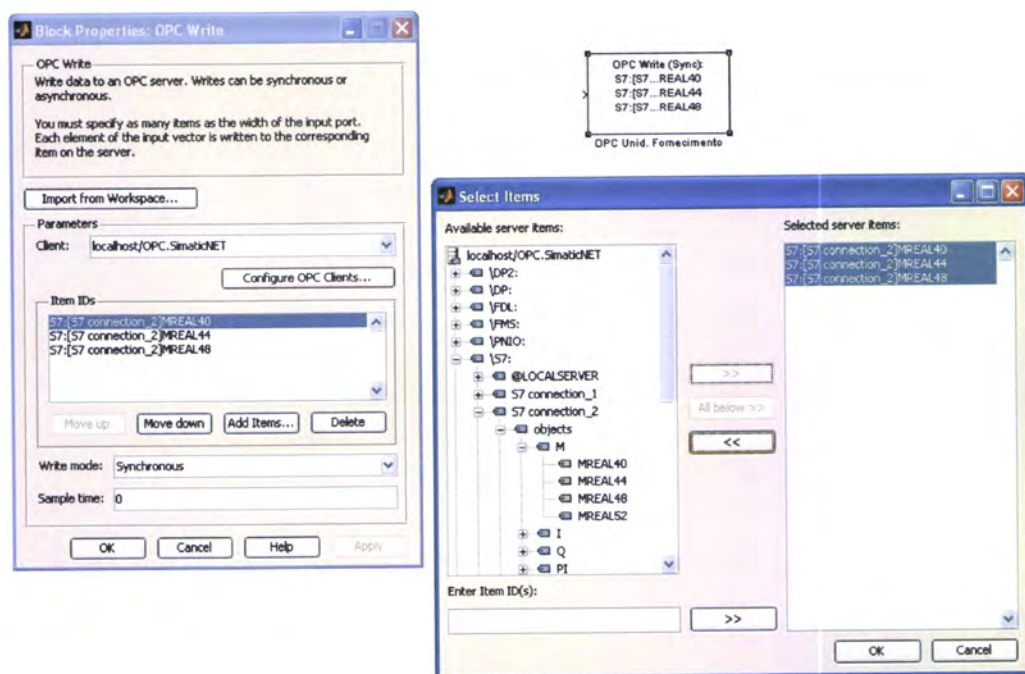


Figura 83: Bloco "OPC Write"

Capítulo 6 – Resultados

6.1 Introdução

Neste capítulo serão apresentados os resultados obtidos através da simulação do sistema desenvolvido, mediante vários cenários possíveis de ocorrer na realidade. O facto de apenas existirem resultados de simulação deveu-se à inexistência de material suficiente para a implementação do sistema completo, mais concretamente por falta de um AVAC e do sistema de controlo da luz artificial e estores.

6.2 Simulação Matlab

Durante o processo de desenvolvimento do controlador foram sendo efectuadas simulações para aferir o desempenho do mesmo.

Os gráficos seguintes correspondem à última simulação do controlador através do Matlab, mediante as seguintes condições:

- Período de tempo de simulação 60 minutos, sendo que o efectua o processo em menos tempo.
- As referências de temperatura eram geradas aleatoriamente, com o valor a variar entre 20°C e 30°C, em intervalos de 10 minutos.
- As referências de luminosidade eram geradas aleatoriamente, com o valor a variar entre 0% e 100%, em intervalos de 10 minutos.
- A temperatura exterior encontrava-se estável nos 24°C.
- A luminosidade exterior encontrava-se estável a 100% (dia de céu limpo).

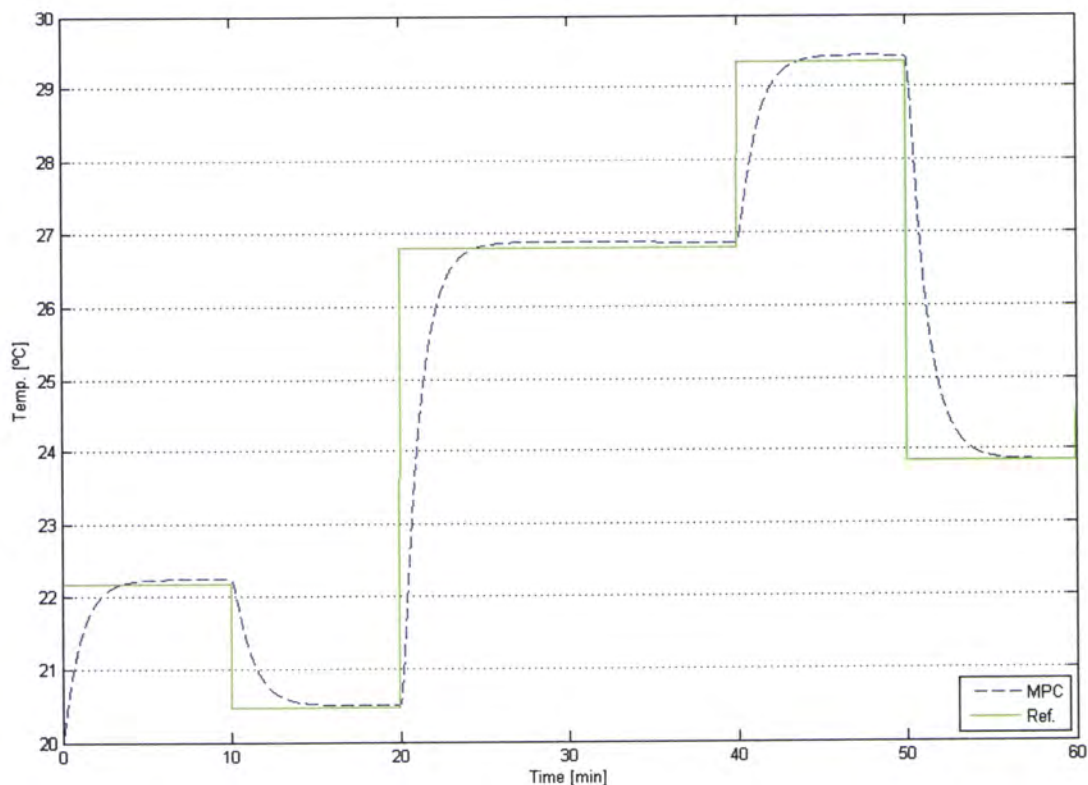


Figura 84: Temperatura e referência conference room (simulação Matlab)

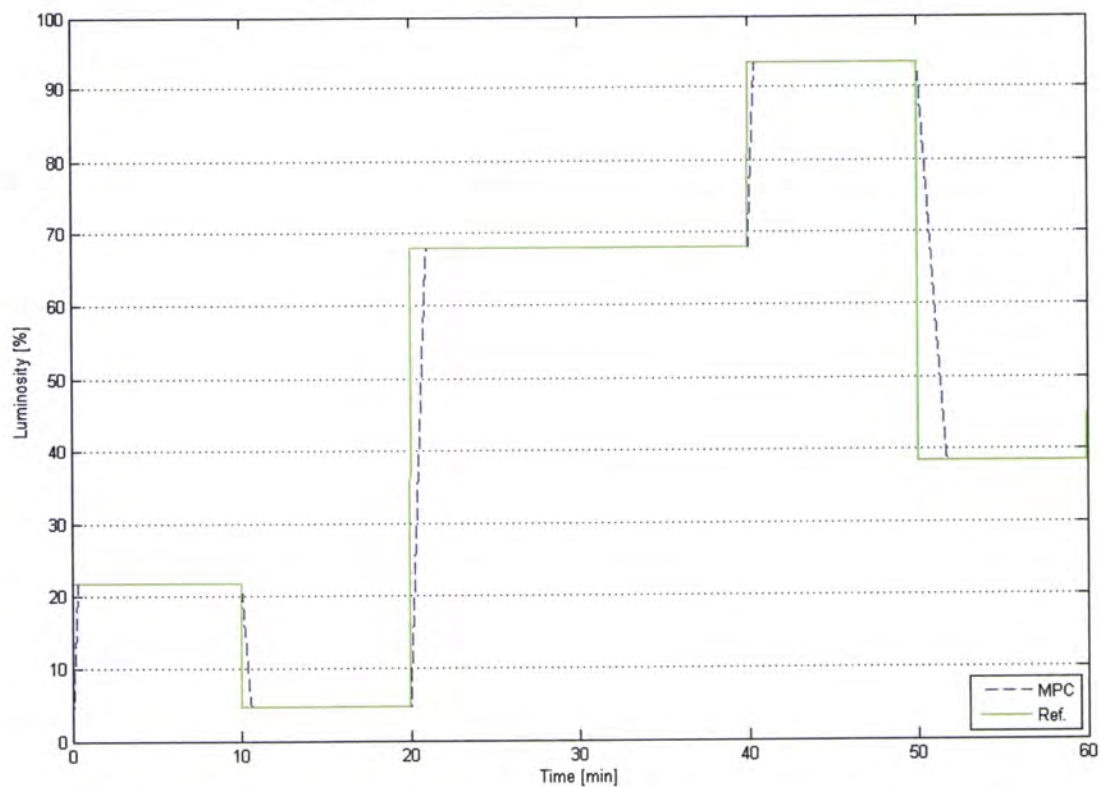


Figura 85: Luminosidade e referência conference room (simulação Matlab)

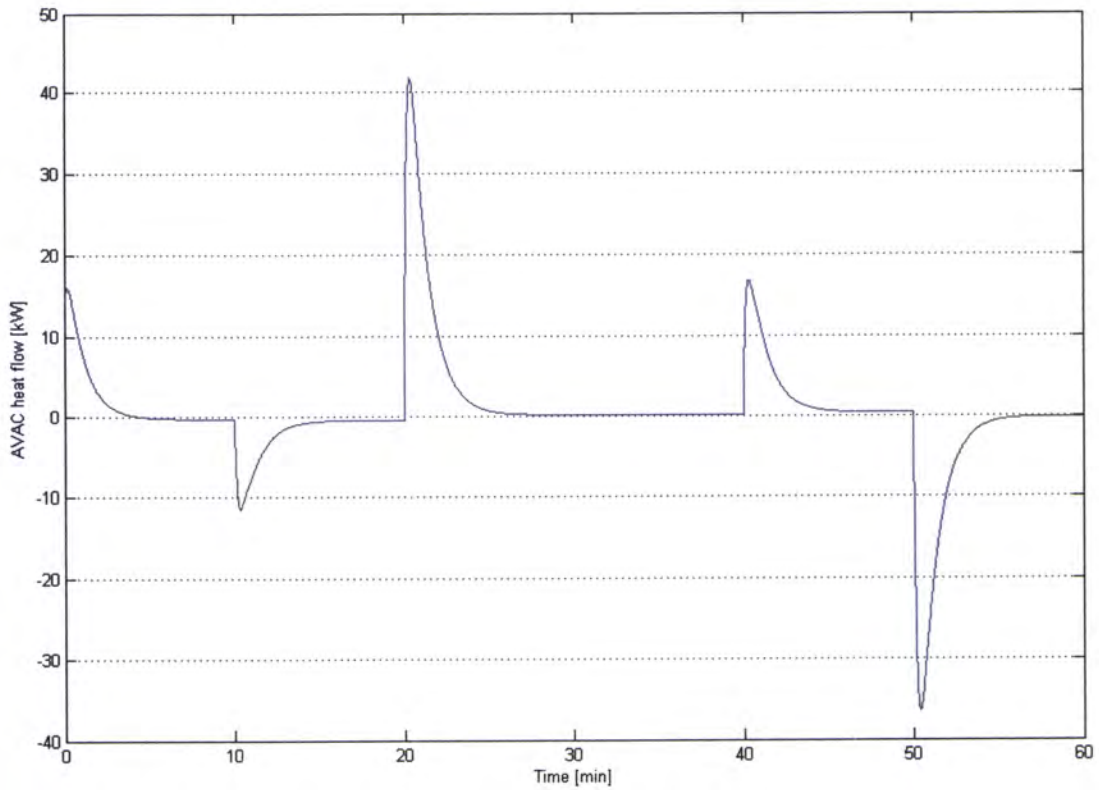


Figura 86: Fluxo de calor gerado pelo AVAC (simulação Matlab)

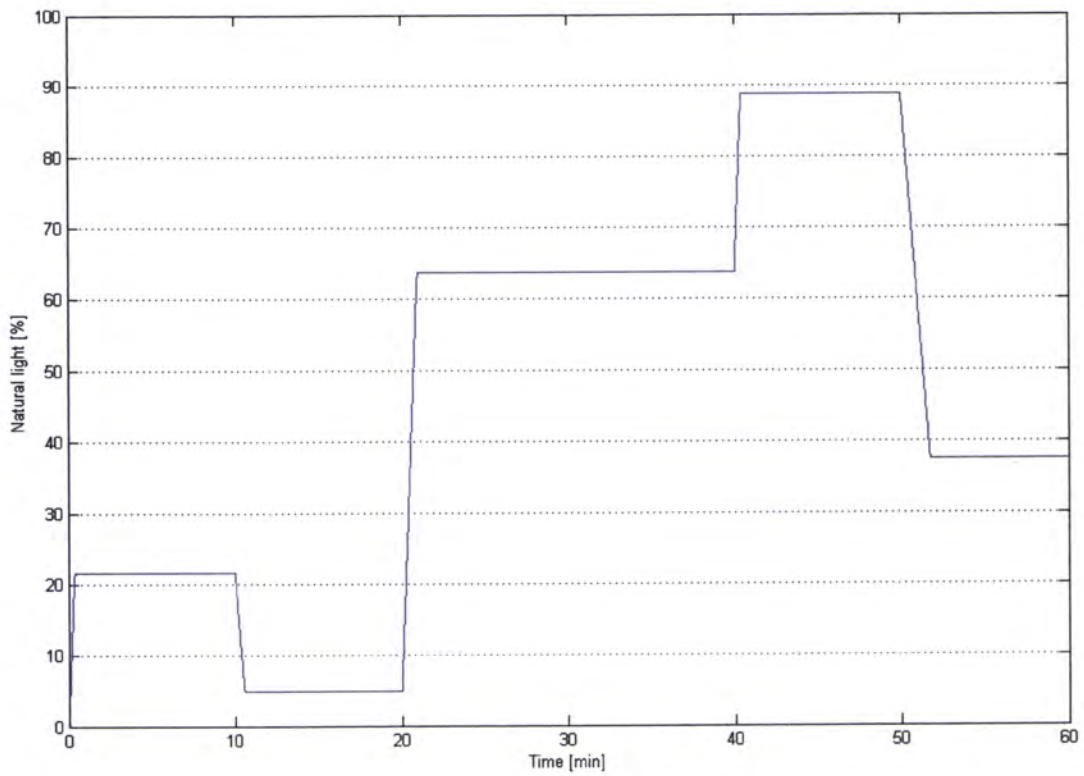


Figura 87: Percentagem de luz natural (simulação Matlab)

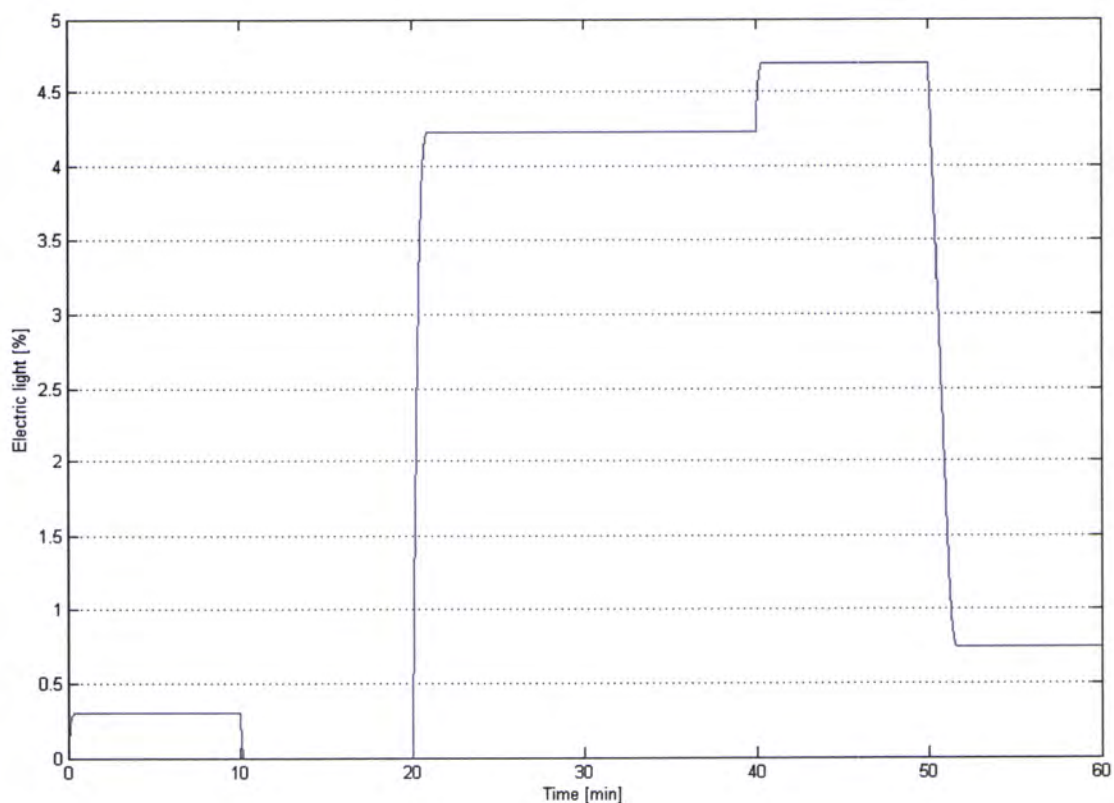


Figura 88: Percentagem de luz artificial (simulação Matlab)

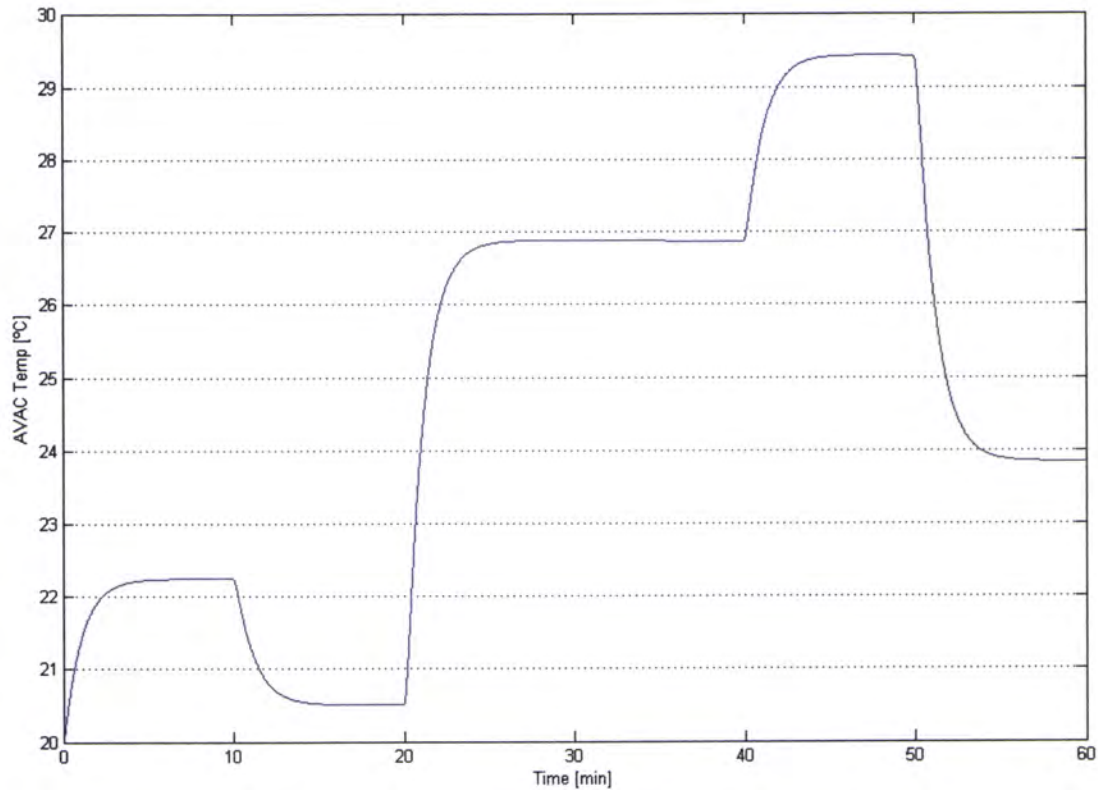


Figura 89: Temperatura de funcionamento do AVAC (simulação Matlab)

6.3 Implementação experimental condicionada

Após o processo de desenvolvimento do controlador procedeu-se à implementação do sistema no laboratório da Universidade de Évora.

No entanto o laboratório não corresponde à conference room idealizada para a aplicação do sistema. Mas mesmo que corresponde-se a plataforma continuava sem poder ser testada a 100% devido à falta de um AVAC, de um sistema de estores controlados por um motor, e do controlo da iluminação artificial variável. Devido a todos estes condicionalismos considero a implementação experimental condicionada.

Os gráficos seguintes correspondem ao teste da plataforma implementada, mediante as seguintes condições:

- Período de tempo de teste 60 minutos reais.
- As interfaces gráficas eram ligadas através de uma tabela de variáveis definida no Simatic STEP 7
- As referências de temperatura e luminosidade foram introduzidas através dos campos de entrada/saída do WinCC, seguindo um plano definido previamente e que apresento de seguida.
- A temperatura exterior encontrava-se estável nos 21°C.
- A luminosidade exterior encontrava-se estável a 100% (dia de céu limpo).

O plano de teste usado no decorrer da simulação consiste nas seguintes acções:

- No minuto zero entram na sala nove utilizadores que indicam as suas preferências.

Utilizador	Referência temperatura	Referência luminosidade
1	22	45
2	23	25
3	26	60
4	23	50
5	23	48
7	24	55
8	22	53
9	22	47
10	25	56

Tabela 6: Referências introduzidas no minuto zero

- Ao fim de cinco minutos chega o utilizador que ainda não se encontrava na sala e introduz as suas preferências na interface.

Utilizador	Referência temperatura	Referência luminosidade
6	24	43

Tabela 7: Referências introduzidas no minuto cinco

- Chegando ao décimo quinto minuto de teste dois dos utilizadores decidem mudar as preferências que tinham introduzido anteriormente.

Utilizador	Referência temperatura	Referência luminosidade
1	21	54
10	27	52

Tabela 8: Referências alteradas no minuto quinze

- Passados mais dez minutos, ao vigésimo quinto minuto de teste outros quatro utilizadores alteraram as suas preferências.

Utilizador	Referência temperatura	Referência luminosidade
3	28	60
4	23	60
7	24	64
9	22	60

Tabela 9: Referências alteradas no minuto vinte e cinco

- Aproximadamente ao quadragésimo minuto de teste um utilizador abandona a sala e quatro utilizadores que permanecem decidem alterar as suas preferências.

Utilizador	Referência temperatura	Referência luminosidade
1	27	75
2	Abandonou a sala	Abandonou a sala
4	23	70
6	28	43
8	22	80
9	28	60

Tabela 10: Referências alteradas no minuto quarenta

- Ao minuto quarenta e cinco outro utilizador abandona a sala e dois dos utilizadores que permanecem ajustam as suas preferências.

Utilizador	Referência temperatura	Referência luminosidade
4	20	80
5	21	65
10	Abandonou a sala	Abandonou a sala

Tabela 11: Referências alteradas no minuto quarenta e cinco

- Passados cinco minutos, no minuto cinquenta três utilizadores abandonam a sala e dois utilizadores que permanecem modificam as suas preferências.

Utilizador	Referência temperatura	Referência luminosidade
1	Abandonou a sala	Abandonou a sala
4	Abandonou a sala	Abandonou a sala
5	Abandonou a sala	Abandonou a sala
6	26	15
7	24	50

Tabela 12: Referências alteradas no minuto cinquenta

- Os utilizadores que ainda se encontravam na sala, abandonam-na ao passar do quinquagésimo minuto de teste.

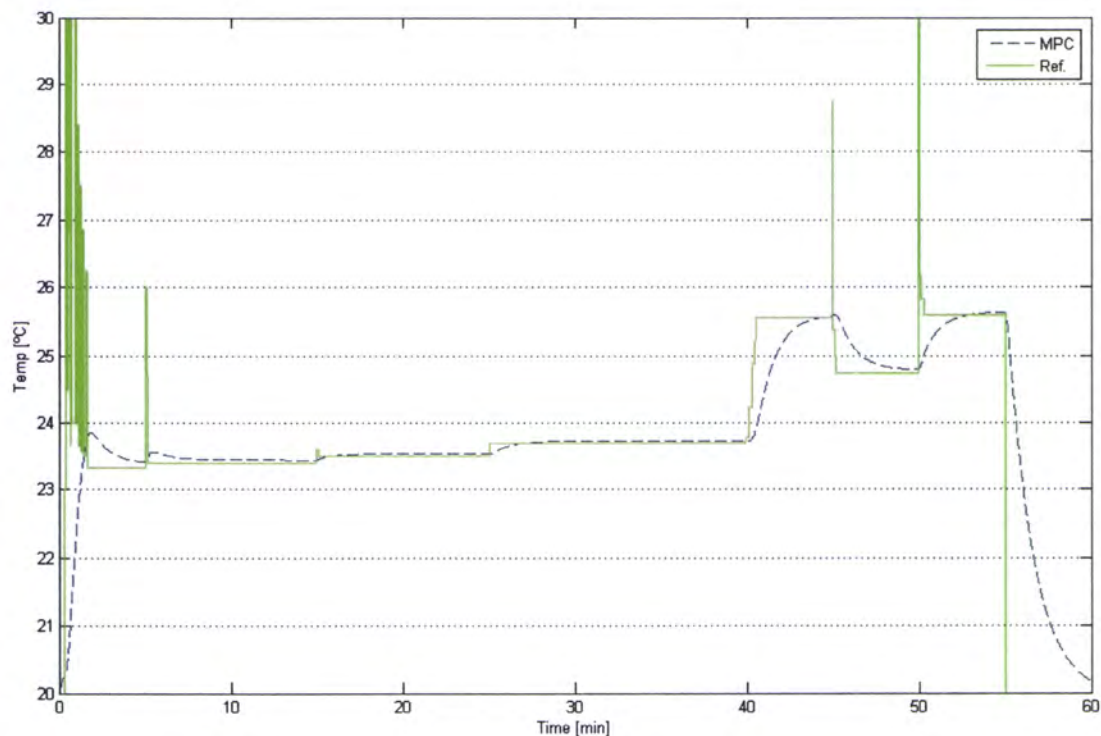


Figura 90: Temperatura e referência conference room (implementação)

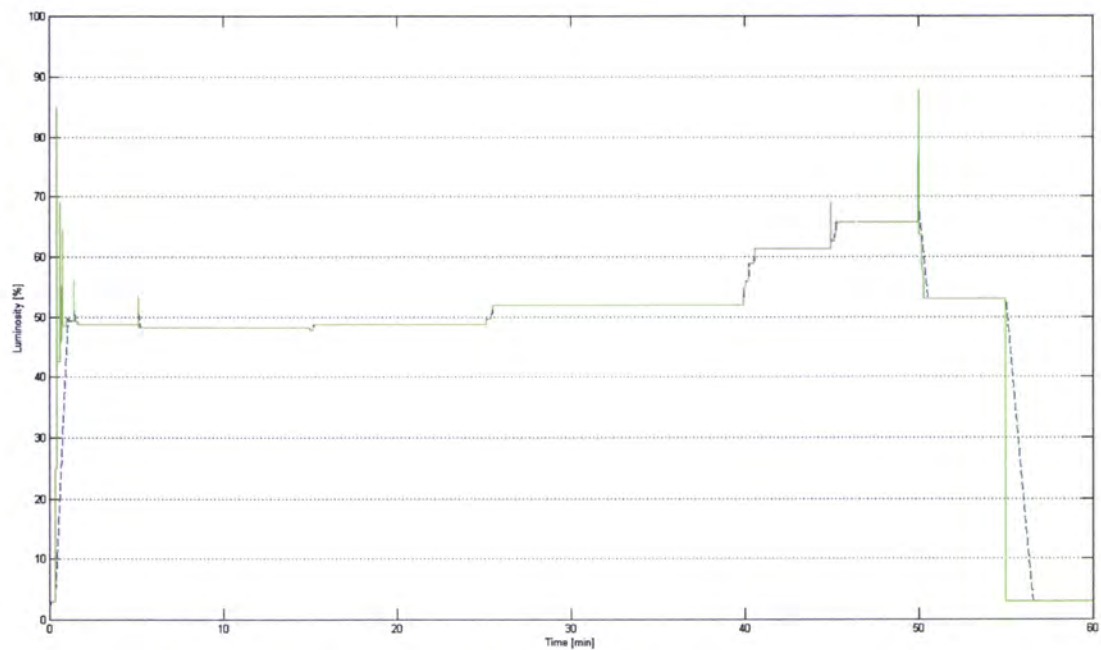


Figura 91: Luminosidade e referência conference room (implementação)

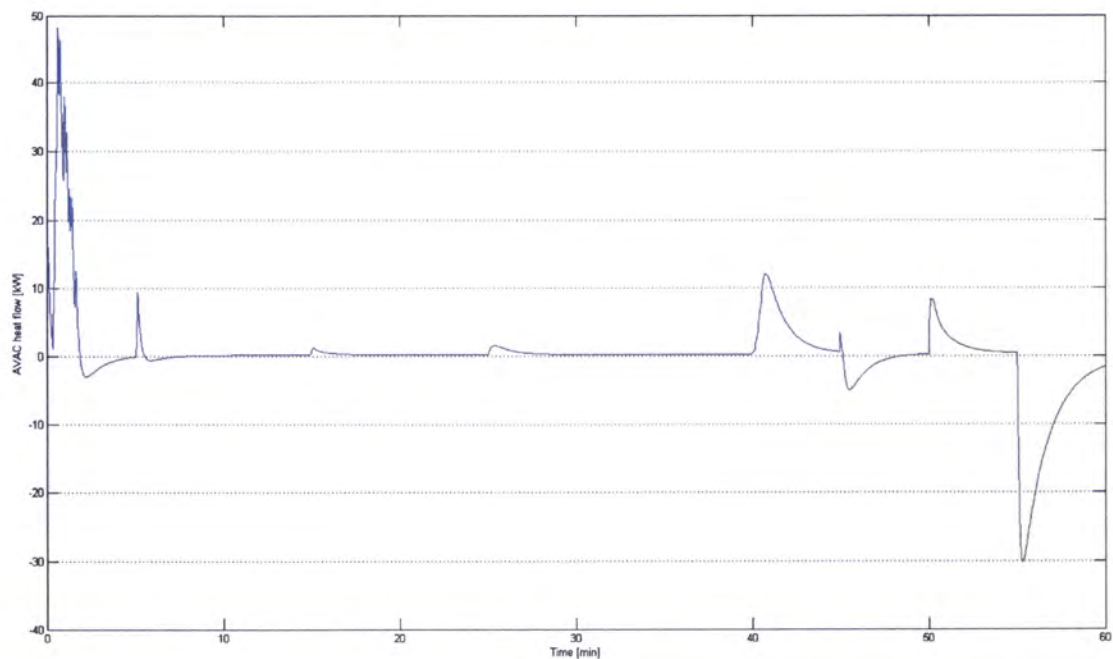


Figura 92: Fluxo de calor gerado pelo AVAC (implementação)

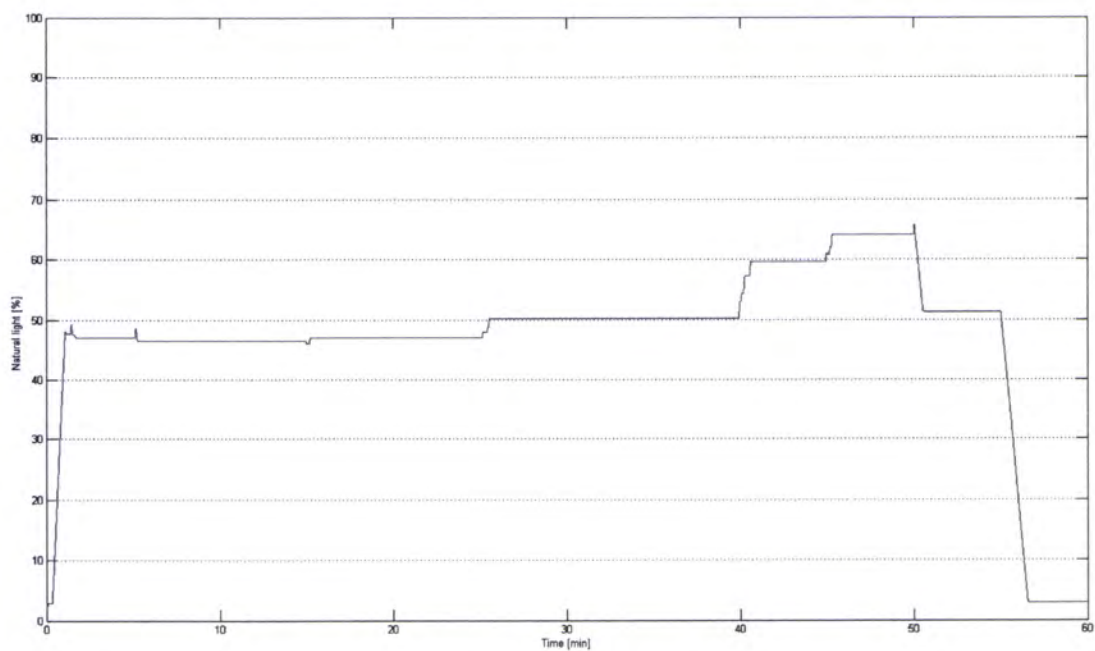


Figura 93: Percentagem de luz natural (implementação)

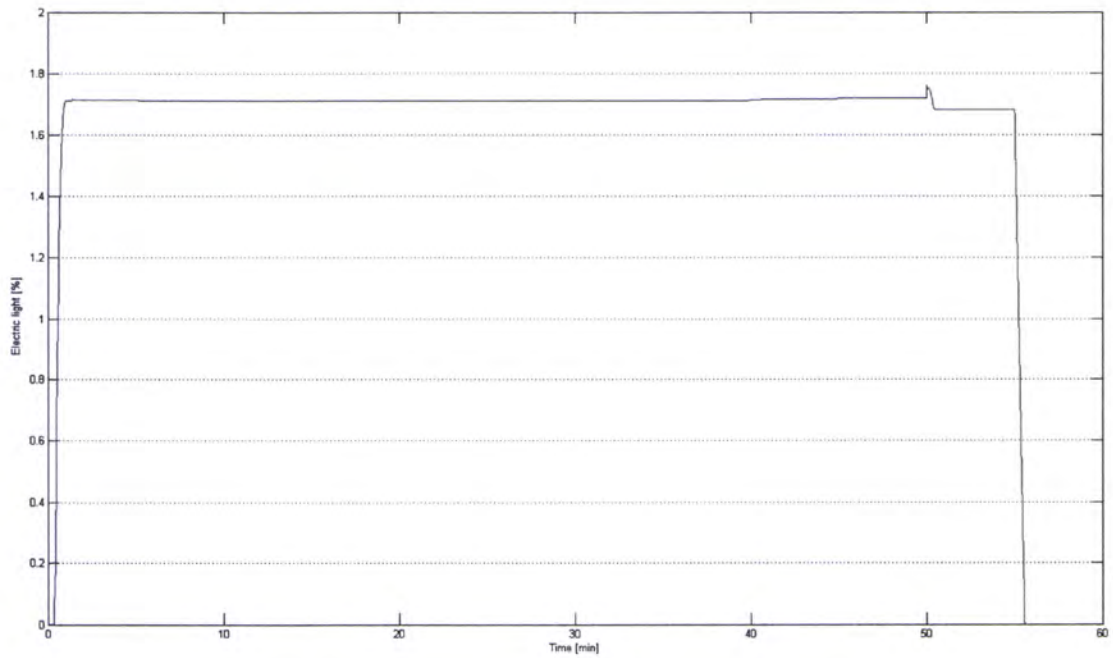


Figura 94: Percentagem de luz artificial (implementação)

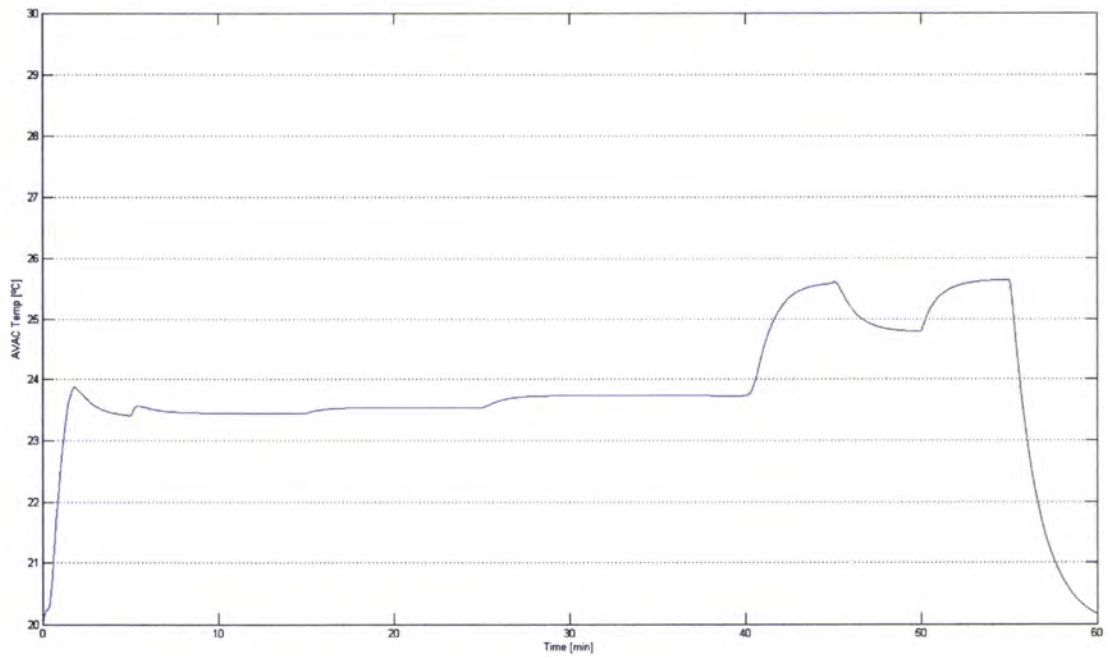


Figura 95: Temperatura de funcionamento do AVAC (implementação)

Capítulo 7 – Conclusões e trabalhos futuros

7.1 Conclusões

Nesta dissertação desenvolveu-se e posteriormente implementou-se uma plataforma interactiva para edifícios inteligentes. Para possibilitar a implementação da plataforma foi desenvolvida uma aplicação com base em autómatos Siemens S7-300 para controlo local do sistema, foi também desenvolvida uma aplicação SCADA que proporciona monitorização e controlo remoto do sistema, e por fim foi desenvolvido um controlo preditivo para gestão do edifício inteligente.

Para o desenvolvimento de todo este sistema foi necessário definir o ambiente em que este seria implementado, de forma a satisfazer as necessidades exigidas e tornar o sistema o mais eficaz possível.

A plataforma foi desenvolvida de modo a ajustar a temperatura e luminosidade de uma conference room às referências fornecidas pelos utilizadores, através do controlo do funcionamento de um AVAC, da abertura ou fecho de estores e controlo da iluminação artificial instalada na sala.

Devido à natureza multivariável do modelo que representa o sistema foi desenvolvido um controlador MPC para controlo do sistema. Este tipo de controlador parece adequar-se bastante bem ao sistema em questão. Os resultados demonstram o que acaba de ser afirmado. Ou seja, este controlador é eficaz a lidar com as restrições naturais do sistema e ajusta o sistema às referências pretendidas de uma forma natural.

Através da observação dos gráficos presentes no capítulo anterior é visível a forma como o controlo gere o sistema, ajustando-o ao longo de um determinado espaço de tempo, numa forma suave e natural no que diz respeito à temperatura e de uma forma mais célere para a luminosidade. A forma mais célere com que o controlo ajusta a luminosidade do sistema não é nem preocupante ou de estranhar, uma vez que abrir estores e ajustar luzes é mais rápido do que aquecer ou arrefecer uma sala de média/grande dimensão. Observa-se ainda pelos gráficos do capítulo anterior que é dada preferência à abertura de estores e consequente entrada de luz natural em detrimento da luz artificial, demonstrando assim o esforço em proporcionar uma maior eficiência energética do edifício.

Convém no entanto fazer uma chamada de atenção para um pormenor nos gráficos do capítulo anterior. Na simulação através do Matlab os gráficos das referências apresentam patamares constantes enquanto não existe variação na referência, o mesmo não se verifica nos gráficos da implementação experimental. Nestes gráficos são visíveis uns picos incomuns, que ocorrem quando mais do que uma referência na sala são alteradas num curtíssimo espaço de tempo (cerca de um ou dois segundos no máximo), portanto num caso de quase simultaneidade na alteração das referências.

A ligação dos diferentes componentes do sistema através do protocolo OPC revelou-se uma escolha certa, em grande parte devido à elevada velocidade de comunicação que este protocolo proporciona e ao facto de ser um protocolo aberto a diferentes fabricantes de hardware e software.

Os objectivos propostos no princípio da dissertação foram cumpridos quase na totalidade, ficando apenas a faltar a conclusão da implementação da plataforma no edifício inteligente e a acessibilidade por remote desktop.

7.2 Trabalhos futuros

A conclusão da implementação da plataforma no edifício inteligente é um dos trabalhos futuros que se pode idealizar para aferir a potencialidade real do sistema desenvolvido.

A acessibilidade à plataforma por remote desktop, através da internet pode ser equacionada como um trabalho futuro. Desta forma a monitorização do sistema chegará ao patamar máximo das comunicações.

O englobamento de uma segunda conference room é também um trabalho futuro e que não necessita de um esforço adicional de grande dimensão uma vez que o controlador MPC ficou preparado para a implementação da segunda sala no sistema. Apenas pequenas alterações ao nível das entradas e saídas do controlador seriam necessárias, seria necessária a configuração de um novo PLC para a sala (em tudo idêntico ao da sala existente), e seria também necessária a criação de uma nova tela e campos de entradas e saídas no WinCC.

Referências bibliográficas

- Christensen, H. (2002). Domestic Robots. 10th IEEE *Mediterranean Conf. on Control and Automation – MED2002*, Lisbon, 2002.
- Close, C. M., Frederick, D. K., Newell, J. C. (2002). *Modelling and Analysis of Dynamic Systems, 3rd edition*. John Wiley & Sons, Inc.
- Figueiredo, J. and Sá da Costa, J. (2007). A Concept for an Operational Management System for Industrial Purposes. IEEE *Intl. Symposium on Intelligent Signal Processing*, Madrid 2007.
- Figueiredo, J. and Sá da Costa, J. (2008a). An Efficient System to Monitor and Control the Energy Production and Consumption. 5th IEEE *Intl. Conf. On European Electricity Market*, Lisbon 2008.
- Figueiredo, J. and Sá da Costa, J. (2008b). Operative platform applied to building automation. *Computer-Aided Civil and Infrastructure Engineering*, 23, 639-653.
- Henriques, E. and Figueiredo, J. (2009). Energy Management in Intelligent Buildings using Predictive Control. *IFAC Intl. Conference on Control Methodologies and Technology for Energy Efficiency, March 29-31, 2010, Vilamoura, Portugal* (paper submitted).
- <http://www.mathworks.com/>
- <http://www.opcfoundation.org/>
- Khan, N., Mariun, Z., Saleem, N., Abas, N. (2007). Fossil Fuels, New Energy Sources and the Great Energy Crisis. *Renewable and Sustainable Energy Rev* (2007), doi:10.1016/j.rser.2007.11.011.
- Kolokotsa, D., Pouliezos, A., Stavrakakis, G. and Lazos, C (2009). Predictive control techniques for energy and indoor environmental quality management in buildings. *Building and Environment*, 44, 1850-1863.
- Maciejowski, J.M. (2002). *Predictive Control with Constrains, 2nd edition*. Prentice-Hall.
- MathWorks (2008). *Matlab 7.6.0.324 (R 2008a)*, The MathWorks Inc. 2008.
- Ogata, K. (2001). *Modern Control Engineering, 4th edition*, Prentice-Hall.
- Sahin, C., Bolat, E. D. (2009). Development of remote control and monitoring of web-based distributed OPC system, *Computer Standards & Interfaces* 31, 984-993.

- Santos, R. A., Normey-Rico, J. E., Gómez, A. M., Arconada, L. F. A., Moraga, C. de P. (2005). OPC based distributed real time simulation of complex continuous processes, *Simulation Modelling Practice and Theory* 13, 525-549
- SIEMENS (2001), *Simatic Net – NCM S7 for Profibus/FMS*, SIEMENS, 12/2001.
- SIEMENS (2003), *Simatic Net – NCM for Industrial Ethernet - Manual*, SIEMENS, 06/2003.
- SIEMENS (2004), *Simatic ProTool/Pro CS V6.0 SP3*, SIEMENS, 2004.
- SIEMENS (2006a), *Simatic Manager Step 7 V5.4+SP1*, SIEMENS, 2006.
- SIEMENS (2006b), *Statement List (STL) for S7-300 and S7-400 Programming Reference Manual*, SIEMENS, Edition 03/2006.
- SIEMENS (2007), *OPC Scout V2.5*, SIEMENS, 2007.
- SIEMENS (2008). *Simatic WinCC V7.0*, SIEMENS, 2008.
- Silva, C., Sousa, J.M., Runkler, T., Palm, R. (2005), Soft Computing Optimization Methods applied to Logistic Processes. *Intl. Journal of Approximate Reasoning*, 40, (2005), 280-301.
- Silva, P., Botto, M., Figueiredo, J. and Rijo, M. (2007). Model Predictive Control of an Experimental Water Canal. *IFAC/IEEE European Control Conference, ECC'07*, pp. 2977-2984, Greece 2007.
- Silva, P. (2007), *Controlo Preditivo de um Canal de Rega Experimental*, Tese de Mestrado, IST, Janeiro 2007.
- Wang, S., Xu, Z., Cao, J., Zhang, J. (2007). A middleware for web service-enabled integration and interoperation of intelligent building systems, *Automation in Construction* 16, 112-121.

Anexo A – OPC Scout

Item Names	Value	Format	Type	Access	Quality	Time Stamp (UTC)
S7:IS7 connection_1JMW20	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW100	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW102	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW110	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW112	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW148	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW150	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW154	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW22	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW30	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW32	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW40	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW42	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW50	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW52	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW60	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW62	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW70	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW72	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW80	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW82	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW90	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMW92	0	Original	uint16	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JMX0.0	False	Original	bool	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JREAL120	0	Original	real32	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JREAL124	0	Original	real32	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JREAL128	0	Original	real32	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_1JREAL132	0	Original	real32	RW	good	10/13/2009 20:19:56.984
S7:IS7 connection_2JREAL40	0	Original	real32	RW	good	10/13/2009 20:19:52.062
S7:IS7 connection_2JREAL44	0	Original	real32	RW	good	10/13/2009 20:19:52.062
S7:IS7 connection_2JREAL48	0	Original	real32	RW	good	10/13/2009 20:19:52.062
S7:IS7 connection_2JREAL52	0	Original	real32	RW	good	10/13/2009 20:19:52.062

Anexo B – OB1 conference room

OB1 - <offline>

"Cycle Execution"

Name: **Family:**
Author: **Version:** 0.1
 Block version: 2
Time stamp Code: 10/12/2009 03:44:28 PM
 Interface: 02/15/1996 04:51:12 PM
Lengths (block/logic/data): 01340 01164 00020

OB1	OB1	OB1	OB1
		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Block: OB1 "Main Program Sweep (Cycle)"

Network: 1

//Estado inicial do sistema

```

A      "Start_Scada"      //Com o Start_Scada ligado
JNB   _004
L      "Med_Ref_Temp"     //Carrega a média das referências de temperatura para a tag Med_Ref_Temp
T      "Med_Ref_Temp"
L      "Med_Ref_Lum"     //Carrega a média das referências de luminosidade para a tag Med_Ref_Lum
T      "Med_Ref_Lum"
L      "Temp_actual"     //Carrega a temperatura actual para a tag Temp_actual
T      "Temp_actual"
L      "Lum_actual"     //Carrega a luminosidade actual para a tag Lum_actual
T      "Lum_actual"
L      "Temp_ext"       //Carrega a temperatura exterior para a tag Temp_ext
T      "Temp_ext"
L      "Lum_ext"       //Carrega a luminosidade exterior para a tag Lum_ext
T      "Lum_ext"
L      1                //Carrega a informação de sistema activo

```



```

T      "Mensagem"
_004: NOP 0

AN     "Start_Scada" //Com o Start_Scada desligado
JNB    _005
L      2 //Carrega a informação de sistema inactivo
T      "Mensagem"
_005: NOP 0

// Interface 1

O(
A      "Start_Scada" // Se
AN     "Start_TP_1" // Start_Scada ligado
O      // E Start_TP_1 desligado
O      // Ou
AN     "Start_Scada" // Start_Scada desligado
A      "Start_TP_1" // E Start_TP_1 ligado
O      // Ou
AN     "Start_Scada" // Start_Scada desligado
AN     "Start_TP_1" // Start_TP_1 desligado
)
JNB    _001
L      0 // Carrega 0 para as tags Ref_Temp_1, Ref_Lum_1 e TP_1 on_off
T      "Ref_Temp_1"
T      "Ref_Lum_1"
T      "TP_1 on_off"
_001: NOP 0

A      "Start_Scada" // Com Start_Scada ligado
A      "Start_TP_1" // Com Start_TP_1 ligado
=      L 10.0
A      L 10.0
JNB    _002
L      "Ref_Temp_1" // Carrega a tag Ref_Temp_1
T      "Ref_Temp_1"
L      "Ref_Lum_1" // Carrega a tag Ref_Lum_1
T      "Ref_Lum_1"
_002: NOP 0

A      L 10.0
A(
L      "Ref_Temp_1"
L      0
<>I // Caso a tag Ref_Temp_1 seja diferente de 0
)
A(
L      "Ref_Lum_1"
L      0
<>I // E a tag Ref_Lum_1 seja diferente de 0
)
JNB    _003
L      1 // Carrega a informação de que existe utilizador presente TP 1
T      "TP_1 on_off"
_003: NOP 0

// Interface 2

O(
A      "Start_Scada" // Se
AN     "Start_TP_2" // Start_Scada ligado
O      // E Start_TP_2 desligado
O      // Ou
AN     "Start_Scada" // Start_Scada desligado
A      "Start_TP_2" // E Start_TP_2 ligado
O      // Ou
```

```
AN "Start_Scada" // Start_Scada desligado
AN "Start_TP_2" // Start_TP_2 desligado
)
JNB _006
L 0 // Carrega 0 para as tags Ref_Temp_2, Ref_Lum_2 e TP_2 on_off
T "Ref_Temp_2"
T "Ref_Lum_2"
T "TP_2 on_off"
_006: NOP 0

A "Start_Scada" // Com Start_Scada ligado
A "Start_TP_2" // Com Start_TP_2 ligado
= L 10.1
A L 10.1
JNB _007
L "Ref_Temp_2" // Carrega a tag Ref_Temp_2
T "Ref_Temp_2"
L "Ref_Lum_2" // Carrega a tag Ref_Lum_2
T "Ref_Lum_2"
_007: NOP 0

A L 10.1
A(
L "Ref_Temp_2" // Caso a tag Ref_Temp_2 seja diferente de 0
L 0
<>I
)
A(
L "Ref_Lum_2" // E a tag Ref_Lum_2 seja diferente de 0
L 0
<>I
)
JNB _008
L 1
T "TP_2 on_off" // Carrega a informação de que existe utilizador presente TP 2
_008: NOP 0

// Interface 3

O(
A "Start_Scada" // Se
AN "Start_TP_3" // Start_Scada ligado
O // E Start_TP_3 desligado
// Ou
AN "Start_Scada" // Start_Scada desligado
A "Start_TP_3" // E Start_TP_3 ligado
O // Ou
AN "Start_Scada" // Start_Scada desligado
AN "Start_TP_3" // Start_TP_3 desligado
)
JNB _009
L 0 // Carrega 0 para as tags Ref_Temp_3, Ref_Lum_3 e TP_3 on_off
T "Ref_Temp_3"
T "Ref_Lum_3"
T "TP_3 on_off"
_009: NOP 0

A "Start_Scada" // Com Start_Scada ligado
A "Start_TP_3" // Com Start_TP_3 ligado
= L 10.2
A L 10.2
JNB _010
L "Ref_Temp_3" // Carrega a tag Ref_Temp_3
T "Ref_Temp_3"
L "Ref_Lum_3" // Carrega a tag Ref_Lum_3
T "Ref_Lum_3"
```

```
_010: NOP    0
      A      L      10.2
      A(
      L      "Ref_Temp_3"      // Caso a tag Ref_Temp_3 seja diferente de 0
      L      0
      <>I
      )
      A(
      L      "Ref_Lum_3"      // E a tag Ref_Lum_3 seja diferente de 0
      L      0
      <>I
      )
      JNB    _011
      L      1
      T      "TP_3 on_off"      // Carrega a informação de que existe utilizador presente TP 3
_011: NOP    0

// Interface 4

      O(
      A      "Start_Scada"      // Se
      AN     "Start_TP_4"      // Start_Scada ligado
      O      // E Start_TP_4 desligado
      AN     "Start_Scada"      // Ou
      A      "Start_TP_4"      // Start_Scada desligado
      O      // E Start_TP_4 ligado
      AN     "Start_Scada"      // Ou
      AN     "Start_TP_4"      // Start_Scada desligado
      )
      JNB    _012
      L      0
      T      "Ref_Temp_4"      // Carrega 0 para as tags Ref_Temp_4, Ref_Lum_4 e TP_4 on_off
      T      "Ref_Lum_4"
      T      "TP_4 on_off"
_012: NOP    0

      A      "Start_Scada"      // Com Start_Scada ligado
      A      "Start_TP_4"      // Com Start_TP_4 ligado
      =      L      10.3
      A      L      10.3
      JNB    _013
      L      "Ref_Temp_4"      // Carrega a tag Ref_Temp_4
      T      "Ref_Temp_4"
      L      "Ref_Lum_4"      // Carrega a tag Ref_Lum_4
      T      "Ref_Lum_4"
_013: NOP    0

      A      L      10.3
      A(
      L      "Ref_Temp_4"
      L      0
      <>I
      )
      A(
      L      "Ref_Lum_4"
      L      0
      <>I
      )
      JNB    _014
      L      1
      T      "TP_4 on_off"      // Carrega a informação de que existe utilizador presente TP 1
_014: NOP    0
```

// Interface 5

```

O(
A      "Start_Scada" // Start_Scada ligado
AN     "Start_TP_5"  // E Start_TP_5 desligado
O      // Ou
AN     "Start_Scada" // Start_Scada desligado
A      "Start_TP_5"  // E Start_TP_5 ligado
O      // Ou
AN     "Start_Scada" // Start_Scada desligado
AN     "Start_TP_5"  // Start_TP_5 desligado
)

```

JNB _015

L 0 // Carrega 0 para as tags Ref_Temp_5, Ref_Lum_5 e TP_5 on_off

T "Ref_Temp_5"

T "Ref_Lum_5"

T "TP_5 on_off"

_015: NOP 0

A "Start_Scada" // Com Start_Scada ligado

A "Start_TP_5" // Com Start_TP_5 ligado

= L 10.4

A L 10.4

JNB _016

L "Ref_Temp_5" // Carrega a tag Ref_Temp_5

T "Ref_Temp_5"

L "Ref_Lum_5" // Carrega a tag Ref_Lum_5

T "Ref_Lum_5"

_016: NOP 0

A L 10.4

A(

L "Ref_Temp_5"

L 0

<>I

// Caso a tag Ref_Temp_5 seja diferente de 0

)

A(

L "Ref_Lum_5"

L 0

<>I

// E a tag Ref_Lum_5 seja diferente de 0

)

JNB _017

L 1 // Carrega a informação de que existe utilizador presente TP 1

T "TP_5 on_off"

_017: NOP 0

// Interface 6

```

O(
A      "Start_Scada" // Start_Scada ligado
AN     "Start_TP_6"  // E Start_TP_6 desligado
O      // Ou
AN     "Start_Scada" // Start_Scada desligado
A      "Start_TP_6"  // E Start_TP_6 ligado
O      // Ou
AN     "Start_Scada" // Start_Scada desligado
AN     "Start_TP_6"  // Start_TP_6 desligado
)

```

JNB _018

L 0 // Carrega 0 para as tags Ref_Temp_6, Ref_Lum_6 e TP_6 on_off

T "Ref_Temp_6"

T "Ref_Lum_6"

T "TP_6 on_off"

_018: NOP 0


```

A      "Start_Scada"      // Com Start_Scada ligado
A      "Start_TP_6"       // Com Start_TP_6 ligado
=      L      10.5
A      L      10.5
JNB    _019
L      "Ref_Temp_6"      // Carrega a tag Ref_Temp_6
T      "Ref_Temp_6"
L      "Ref_Lum_6"      // Carrega a tag Ref_Lum_6
T      "Ref_Lum_6"
_019: NOP 0

A      L      10.5
A(
L      "Ref_Temp_6"
L      0
<>I      // Caso a tag Ref_Temp_6 seja diferente de 0
)
A(
L      "Ref_Lum_6"
L      0
<>I      // E a tag Ref_Lum_6 seja diferente de 0
)
JNB    _020
L      1
T      "TP_6on_off"      // Carrega a informação de que existe utilizador presente TP 1
_020: NOP 0

// Interface 7

O(
A      "Start_Scada"      // Se
AN     "Start_TP_7"      // Start_Scada ligado
O      // E Start_TP_7 desligado
O      // Ou
AN     "Start_Scada"      // Start_Scada desligado
A      "Start_TP_7"      // E Start_TP_7 ligado
O      // Ou
AN     "Start_Scada"      // Start_Scada desligado
AN     "Start_TP_7"      // Start_TP_7 desligado
)
JNB    _021
L      0
T      "Ref_Temp_7"      // Carrega 0 para as tags Ref_Temp_7, Ref_Lum_7 e TP_7 on_off
T      "Ref_Lum_7"
T      "TP_7 on_off"
_021: NOP 0

A      "Start_Scada"      // Com Start_Scada ligado
A      "Start_TP_7"       // Com Start_TP_7 ligado
=      L      10.6
A      L      10.6
JNB    _022
L      "Ref_Temp_7"      // Carrega a tag Ref_Temp_7
T      "Ref_Temp_7"
L      "Ref_Lum_7"      // Carrega a tag Ref_Lum_7
T      "Ref_Lum_7"
_022: NOP 0

A      L      10.6
A(
L      "Ref_Temp_7"
L      0
<>I      // Caso a tag Ref_Temp_7 seja diferente de 0
)
A(
L      "Ref_Lum_7"

```

```
L      0
<>I           // E a tag Ref_Lum_7 seja diferente de 0
)
JNB     _023
L       1           // Carrega a informação de que existe utilizador presente TP 1
T      "TP_7 on_off"
_023: NOP      0

// Interface 8

O(           // Se
A      "Start_Scada" // Start_Scada ligado
AN     "Start_TP_8"  // E Start_TP_8 desligado
O           // Ou
AN     "Start_Scada" // Start_Scada desligado
A      "Start_TP_8"  // E Start_TP_8 ligado
O           // Ou
AN     "Start_Scada" // Start_Scada desligado
AN     "Start_TP_8"  // Start_TP_8 desligado
)
JNB     _024
L       0           // Carrega 0 para as tags Ref_Temp_8, Ref_Lum_8 e TP_8 on_off
T      "Ref_Temp_8"
T      "Ref_Lum_8"
T      "TP_8 on_off"
_024: NOP      0

A      "Start_Scada" // Com Start_Scada ligado
A      "Start_TP_8"  // Com Start_TP_8 ligado
=      L      10.7
A      L      10.7
JNB     _025
L      "Ref_Temp_8" // Carrega a tag Ref_Temp_8
T      "Ref_Temp_8"
L      "Ref_Lum_8" // Carrega a tag Ref_Lum_8
T      "Ref_Lum_8"
_025: NOP      0

A      L      10.7
A(
L      "Ref_Temp_8"
L      0
<>I           // Caso a tag Ref_Temp_8 seja diferente de 0
)
A(
L      "Ref_Lum_8"
L      0
<>I           // E a tag Ref_Lum_8 seja diferente de 0
)
JNB     _026
L       1           // Carrega a informação de que existe utilizador presente TP 1
T      "TP_8 on_off"
_026: NOP      0

// Interface 9

O(           // Se
A      "Start_Scada" // Start_Scada ligado
AN     "Start_TP_9"  // E Start_TP_9 desligado
O           // Ou
AN     "Start_Scada" // Start_Scada desligado
A      "Start_TP_9"  // E Start_TP_9 ligado
O           // Ou
AN     "Start_Scada" // Start_Scada desligado
```

```
AN      "Start_TP_9"      // Start_TP_9 desligado
)
JNB     _027
L       0                  // Carrega 0 para as tags Ref_Temp_9, Ref_Lum_9 e TP_9 on_off
T       "Ref_Temp_9"
T       "Ref_Lum_9"
T       "TP_9 on_off"
_027: NOP 0

A       "Start_Scada"    // Com Start_Scada ligado
A       "Start_TP_9"     // Com Start_TP_9 ligado
=       L      11.0
A       L      11.0
JNB     _028
L       "Ref_Temp_9"     // Carrega a tag Ref_Temp_9
T       "Ref_Temp_9"
L       "Ref_Lum_9"     // Carrega a tag Ref_Lum_9
T       "Ref_Lum_9"
_028: NOP 0

A       L      11.0
A(
L       "Ref_Temp_9"
L       0
<>I    // Caso a tag Ref_Temp_9 seja diferente de 0
)
A(
L       "Ref_Lum_9"
L       0
<>I    // E a tag Ref_Lum_9 seja diferente de 0
)
JNB     _029
L       1
T       "TP_9 on_off"
_029: NOP 0

// Interface 10

O(
A       "Start_Scada"    // Se
AN      "Start_TP_10"    // Start_Scada ligado
O       // E Start_TP_10 desligado
O       // Ou
AN      "Start_Scada"    // Start_Scada desligado
A       "Start_TP_10"    // E Start_TP_10 ligado
O       // Ou
AN      "Start_Scada"    // Start_Scada desligado
AN      "Start_TP_10"    // Start_TP_10 desligado
)
JNB     _030
L       0                  // Carrega 0 para as tags Ref_Temp_10, Ref_Lum_10 e TP_10 on_off
T       "Ref_Temp_10"
T       "Ref_Lum_10"
T       "TP_10 on_off"
_030: NOP 0

A       "Start_Scada"    // Com Start_Scada ligado
A       "Start_TP_10"    // Com Start_TP_10 ligado
=       L      11.1
A       L      11.1
JNB     _031
L       "Ref_Temp_10"    // Carrega a tag Ref_Temp_10
T       "Ref_Temp_10"
L       "Ref_Lum_10"    // Carrega a tag Ref_Lum_10
T       "Ref_Lum_10"
_031: NOP 0
```

```
A      L      11.1
A(
L      "Ref_Temp_10"
L      0
<>I          // Caso a tag Ref_Temp_10 seja diferente de 0
)
A(
L      "Ref_Lum_10"
L      0
<>I          // E a tag Ref_Lum_10 seja diferente de 0
)
JNB    _032
L      1
T      "TP_10 on_off" // Carrega a informação de que existe utilizador presente TP 1
NOP    0

L      "TP_1 on_off" // Presença de utilizador TP 1
L      "TP_2 on_off" // Presença de utilizador TP 2
+I
T      "N_utilizadores" // Soma dos 2 parametros anteriores
NOP    0 // Numero total de utilizadores na sala

L      "N_utilizadores" // Numero total de utilizadores na sala
L      "TP_3 on_off" // Presença de utilizador TP 3
+I
T      "N_utilizadores" // Soma dos 2 parametros anteriores
NOP    0 // Numero total de utilizadores na sala

L      "N_utilizadores" // Numero total de utilizadores na sala
L      "TP_4 on_off" // Presença de utilizador TP 4
+I
T      "N_utilizadores" // Soma dos 2 parametros anteriores
NOP    0 // Numero total de utilizadores na sala

L      "N_utilizadores" // Numero total de utilizadores na sala
L      "TP_5 on_off" // Presença de utilizador TP 5
+I
T      "N_utilizadores" // Soma dos 2 parametros anteriores
NOP    0 // Numero total de utilizadores na sala

L      "N_utilizadores" // Numero total de utilizadores na sala
L      "TP_6 on_off" // Presença de utilizador TP 6
+I
T      "N_utilizadores" // Soma dos 2 parametros anteriores
NOP    0 // Numero total de utilizadores na sala

L      "N_utilizadores" // Numero total de utilizadores na sala
L      "TP_7 on_off" // Presença de utilizador TP 7
+I
T      "N_utilizadores" // Soma dos 2 parametros anteriores
NOP    0 // Numero total de utilizadores na sala

L      "N_utilizadores" // Numero total de utilizadores na sala
L      "TP_8 on_off" // Presença de utilizador TP 8
+I
T      "N_utilizadores" // Soma dos 2 parametros anteriores
NOP    0 // Numero total de utilizadores na sala

L      "N_utilizadores" // Numero total de utilizadores na sala
L      "TP_9 on_off" // Presença de utilizador TP 9
+I
T      "N_utilizadores" // Soma dos 2 parametros anteriores
NOP    0 // Numero total de utilizadores na sala
```



```
L   "N_utilizadores" // Numero total de utilizadores na sala
L   "TP_10 on_off"   // Presença de utilizador TP 10
+I                                     // Soma dos 2 parametros anteriores
T   "N_utilizadores" // Numero total de utilizadores na sala
NOP  0
```




Anexo C – Símbolos conference room

Properties of symbol table

Name: Symbols
 Author:
 Comment:
 Created on: 10/11/2009 03:01:46 PM
 Last modified on: 10/10/2009 06:49:52 PM
 Last filter criterion: All Symbols
 Number of symbols: 51/51
 Last Sorting: Symbol Ascending

Status	Symbol	Address	Data type	Comment
	Cycle Execution	OB 1	OB 1	
	Lum_actual	MD 132	REAL	
	Lum_ext	MW 154	WORD	
	Med_Ref_Lum	MD 124	REAL	
	Med_Ref_Temp	MD 120	REAL	
	Mensagem	MW 152	WORD	
	N_utilizadores	MW 150	WORD	
	Ref_Lum_1	MW 22	WORD	
	Ref_Lum_10	MW 112	WORD	
	Ref_Lum_2	MW 32	WORD	
	Ref_Lum_3	MW 42	WORD	
	Ref_Lum_4	MW 52	WORD	
	Ref_Lum_5	MW 62	WORD	
	Ref_Lum_6	MW 72	WORD	
	Ref_Lum_7	MW 82	WORD	
	Ref_Lum_8	MW 92	WORD	
	Ref_Lum_9	MW 102	WORD	
	Ref_Temp_1	MW 20	WORD	
	Ref_Temp_10	MW 110	WORD	
	Ref_Temp_2	MW 30	WORD	
	Ref_Temp_3	MW 40	WORD	
	Ref_Temp_4	MW 50	WORD	
	Ref_Temp_5	MW 60	WORD	
	Ref_Temp_6	MW 70	WORD	
	Ref_Temp_7	MW 80	WORD	
	Ref_Temp_8	MW 90	WORD	
	Ref_Temp_9	MW 100	WORD	
	Start_Scada	M 0.0	BOOL	
	Start_TP_1	M 0.1	BOOL	
	Start_TP_10	M 1.2	BOOL	
	Start_TP_2	M 0.2	BOOL	
	Start_TP_3	M 0.3	BOOL	
	Start_TP_4	M 0.4	BOOL	
	Start_TP_5	M 0.5	BOOL	
	Start_TP_6	M 0.6	BOOL	
	Start_TP_7	M 0.7	BOOL	
	Start_TP_8	M 1.0	BOOL	
	Start_TP_9	M 1.1	BOOL	
	Temp_actual	MD 128	REAL	
	Temp_ext	MW 148	WORD	
	TP_1 on_off	MW 28	WORD	
	TP_10 on_off	MW 118	WORD	

Status	Symbol	Address	Data type	Comment
	TP_2 on_off	MW 38	WORD	
	TP_3 on_off	MW 48	WORD	
	TP_4 on_off	MW 58	WORD	
	TP_5 on_off	MW 68	WORD	
	TP_6 on_off	MW 78	WORD	
	TP_7 on_off	MW 88	WORD	
	TP_8 on_off	MW 98	WORD	
	TP_9 on_off	MW 108	WORD	
	VAT_1	VAT 1		



Anexo D – OB1 unidade fornecimento

OB1 - <offline>

"Cycle Execution"

Name: **Family:**
Author: **Version:** 0.1
 Block version: 2
Time stamp Code: 10/13/2009 09:43:45 PM
 Interface: 02/15/1996 04:51:12 PM
Lengths (block/logic/data): 00152 00018 00020

Param	Format	Start	Description
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Block: OB1 "Main Program Sweep (Cycle)"

Network: 1

// A simplicidade deste programa deve-se à não implementação do AVAC, controlo de abertura de estores e controlo das lampadas. De futuro, caso o sistema venha a ser implementado é necessário completar o programa de forma a controlar os referidos dispositivos.

```

L   "Temp_avac"           // Transferencia do valor da temperatura de funcionamento do AVA
C
T   "Temp_avac"

L   "Luz_artificial"     // Transferencia da percentagem da luminosidade artificial
T   "Luz_artificial"

L   "Abert_estores"      // Transferencia do valor de numero de passos para o motor que a
T   "Abert_estores"      bre os estores

L   "Perc_Abert_estores" // Transferencia da percentagem correspondente à abertura dos es
T   "Perc_Abert_estores" tores

```



Anexo E – Símbolos unidade fornecimento

Properties of symbol table

Name: Symbols
Author:
Comment:
Created on: 10/11/2009 03:01:46 PM
Last modified on: 10/13/2009 09:45:06 PM
Last filter criterion: All Symbols
Number of symbols: 5/5
Last Sorting: Symbol Ascending

Status	Symbol	Address	Data type	Comment
	Abert_estores	MD 48	REAL	
	Cycle Execution	OB 1	OB 1	
	Luz_artificial	MD 44	REAL	
	Perc_Abert_estores	MD 52	REAL	
	Temp_avac	MD 40	REAL	

Anexo F – ProTool conference room

SIEMENS

SIMATIC PANEL

TP 1

TP170A

CHAPTER SUMMARY

- **Project informations**
- **Screens**
- **Event Messages**
- **Tags**
- **Area pointer**

Project informations

Device : TP170A
Project name : TP 1
Storage Location : C:\DOCUMENTS AND SETTINGS\PROJ_PETER\DESKTOP\TESE EDUARDO\SIMATIC\PLATAFOR\TDOP\PRO.PDB
Created on : 7/25/2009 7:32:25
Modified on : 10/11/2009 1:19:30
Generated on : 10/11/2009 1:21:55
Downloaded on : 10/11/2009 1:22:11
Edited with : V 6.0
Memory Requirement - After Download:
Configuration : 108 KByte

CONTENT

Informacao	(Screen number: 2)	2
Inicio	(Screen number: 1)	4
Preferencias	(Screen number: 3)	6

Conference room 1

Informacao

12:00:00 PM Message Text

Temperatura actual

<0.0

Luminosidade actual

<0.0

Preferencias
utilizador

Sair

Simple Message View : **OBJECT 10**
Tab-Order : 3

StateButton : **OBJECT 7**
Tab-Order : 1
Functions : to Event 'OnButtonUp'
- Select_Screen_Fixed(Screen name:=Preferencias,Field number:=0)

StateButton : **OBJECT 8**
Tab-Order : 2
Functions : to Event 'OnButtonUp'
- Select_Screen_Fixed(Screen name:=Inicio,Field number:=0)
- Reset_bit(Bit:=ligar/deligar)

Output Field : **OBJECT 3**
Value : Tag Temp_actual

Output Field : **OBJECT 6**
Value : Tag Lum_actual

Conference room 1

Interface 1

Bem vindo

Entrar

Configuracao
sistema

StateButton

: OBJECT 3

Tab-Order : 1
Functions : to Event 'OnButtonUp'
 - Select_Screen_Fixed(Screen name:=Informacao,Field number:=0)
 - Set_bit(Bit:=ligar/deligar)

StateButton

: OBJECT 4

Tab-Order : 2
Functions : to Event 'OnButtonUp'
 - Exit_Runtime(Mode:=Exit_ProTool_RT_only)

Conference room 1

Preferencias

Temperatura desejada

=0

Luminosidade desejada

=0

Voltar

StateButton : OBJECT 8
Tab-Order : 3
Functions : to Event 'OnButtonUp'
 - Select_Screen_Fixed(Screen name:=Informacao,Field number:=0)

Input Field : OBJECT 10
Tab-Order : 1
Value : Tag Ref_Temp

Input Field : OBJECT 11
Tab-Order : 2
Value : Tag Ref_Lum

Event Messages

0001 Sistema activo

0002 Sistema inactivo

CONTENT

ligar/deligar

Tag : ligar/deligar
Format : BOOL
Acquisition cycle : 1.0 sec.
Address : M 0.1
PLC : PLC_1
Read : x
Read continuously : -

Tag : Lum actual
Format : REAL
Acquisition cycle : 1.0 sec.
Address : MD 132
PLC : PLC_1
Read : x
Read continuously : -

Tag : Ref Lum
Format : WORD
Acquisition cycle : 1.0 sec.
Address : MW 22
PLC : PLC_1
Read : x
Read continuously : -
Limit values : Upper limit : 100
Lower limit : 0

Tag : Ref Temp
Format : WORD
Acquisition cycle : 1.0 sec.
Address : MW 20
PLC : PLC_1
Read : x
Read continuously : -
Limit values : Upper limit : 30
Lower limit : 20

Tag : Temp actual
Format : REAL
Acquisition cycle : 1.0 sec.
Address : MD 128
PLC : PLC_1
Read : x
Read continuously : -

Area pointer

Event Messages 1

Acquisition cycle : 1.0 sec.
Messages : 1-16
Address : MW 152
PLC : PLC_1

Anexo G - WinCC

WinCC™ Control Center - CS Copyright © 1994-2003 by SIEMENS AG \\PETER\WinCC_Project_proj2v1\proj2v1.mcp	
---	--

Computer	Computer Type	Server
PETER		

Tag Management		
@SCRIPT_COUNT_TAGS	Data Type	32-Bit unsigned
	Group	Script
	Parameters	internal tag
@SCRIPT_COUNT_REQUESTS_IN_QUEUE	Data Type	32-Bit unsigned
	Group	Script
	Parameters	internal tag
@SCRIPT_COUNT_ACTIONS_IN_QUEUES	Data Type	32-Bit unsigned
	Group	Script
	Parameters	internal tag
@TLGRT_SIZEOF_NOTIFY_QUEUE	Data Type	64-Bit IEEE 754
	Group	TagLoggingRt
	Parameters	internal tag
@TLGRT_SIZEOF_NULL_INPUT_QUEUE	Data Type	64-Bit IEEE 754
	Group	TagLoggingRt
	Parameters	internal tag
@TLGRT_TAGS_PER_SECOND	Data Type	64-Bit IEEE 754
	Group	TagLoggingRt
	Parameters	internal tag
@TLGRT_AVERAGE_TAGS_PER_SECOND	Data Type	64-Bit IEEE 754
	Group	TagLoggingRt
	Parameters	internal tag
@CurrentUser	Data Type	Text 8-Bit
	Parameters	internal tag
@DeltaLoaded	Data Type	32-Bit unsigned
	Parameters	internal tag
@LocalMachineName	Data Type	Text 8-Bit
	Parameters	internal tag
@ConnectedRTClients	Data Type	16-Bit unsigned
	Parameters	internal tag
@RedundantServerState	Data Type	16-Bit unsigned
	Parameters	internal tag
@DataSourceNameRT	Data Type	Text 16-Bit
	Parameters	internal tag
@ServerName	Data Type	Text 16-Bit
	Parameters	internal tag
@CurrentUserName	Data Type	Text 16-Bit
	Parameters	internal tag
Client_Temp9_xyz	Data Type	16-Bit unsigned
	Parameters	"S7:S7 connection_1\MW100". "" . 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Luz9_xyz	Data Type	16-Bit unsigned
	Parameters	"S7:S7 connection_1\MW102". "" . 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Temp10_xyz	Data Type	16-Bit unsigned
	Parameters	"S7:S7 connection_1\MW110". "" . 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Luz10_xyz	Data Type	16-Bit unsigned
	Parameters	"S7:S7 connection_1\MW112". "" . 18
	Connection	NewConnection



Tag Management		
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_NumUtilizadores_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW150". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Temp1_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW20". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Luz1_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW22". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Temp2_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW30". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Luz2_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW32". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Temp3_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW40". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Luz3_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW42". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Temp4_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW50". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Luz4_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW52". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Temp5_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW60". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Luz5_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW62". "", 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Temp6_xyz	Data Type	16-Bit unsigned
	Parameters	"S7;S7 connection_1 MW70". "", 18



Tag Management		
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Luz6_xyz	Data Type	16-Bit unsigned
	Parameters	"S7{S7 connection_1}MW72". "" . 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Temp7_xyz	Data Type	16-Bit unsigned
	Parameters	"S7{S7 connection_1}MW80". "" . 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Luz7_xyz	Data Type	16-Bit unsigned
	Parameters	"S7{S7 connection_1}MW82". "" . 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Temp8_xyz	Data Type	16-Bit unsigned
	Parameters	"S7{S7 connection_1}MW90". "" . 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Luz8_xyz	Data Type	16-Bit unsigned
	Parameters	"S7{S7 connection_1}MW92". "" . 18
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Start_xyz	Data Type	Bit value
	Parameters	"S7{S7 connection_1}MX0.0". "" . 11
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Med_Ref_Temp_xyz	Data Type	32-Bit IEEE 754
	Parameters	"S7{S7 connection_1}MREAL120". "" . 4
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Med_Ref_Lum_xyz	Data Type	32-Bit IEEE 754
	Parameters	"S7{S7 connection_1}MREAL124". "" . 4
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Temp_Act_xyz	Data Type	32-Bit IEEE 754
	Parameters	"S7{S7 connection_1}MREAL128". "" . 4
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_Lum_Act_xyz	Data Type	32-Bit IEEE 754
	Parameters	"S7{S7 connection_1}MREAL132". "" . 4
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_TempAVAC_xyz	Data Type	32-Bit IEEE 754
	Parameters	"S7{S7 connection_2}MREAL40". "" . 4
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_LuzARTF_xyz	Channel unit	OPC Groups (OPCHN Unit #1)
	Data Type	32-Bit IEEE 754



Tag Management

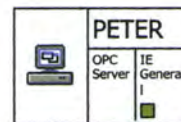
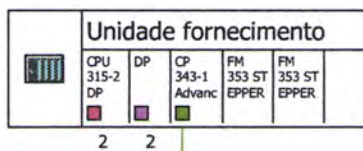
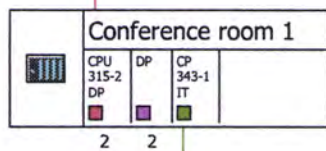
	Parameters	"S7:S7 connection_2\MREAL44". "" . 4
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_AberEST_xyz	Data Type	32-Bit IEEE 754
	Parameters	"S7:S7 connection_2\MREAL48". "" . 4
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)
Client_PercAberEST_xyz	Data Type	32-Bit IEEE 754
	Parameters	"S7:S7 connection_2\MREAL52". "" . 4
	Connection	NewConnection
	Channel	OPC
	Channel unit	OPC Groups (OPCHN Unit #1)

Connections

NewConnection	Unit	OPC Groups (OPCHN Unit #1)
	Parameters	OPC.SimaticNET: <LOCAL>; 0.00; 0; 0; 1

Anexo H - NetPro

MPI(1)
MPI



Ethernet(1)
Industrial Ethernet

Industrial Ethernet: Ethernet(1) (S7 subnet ID: 0083 - 0008) contains the following network connections:

Station address:	Station:	Module:	R/S:
192.168.0.1	Unidade fornecimento	CP 343-1 Advanced-IT	0/4
192.168.0.4	Conference room 1	CP 343-1 IT	0/4
192.168.0.10	PETER	IE General	-

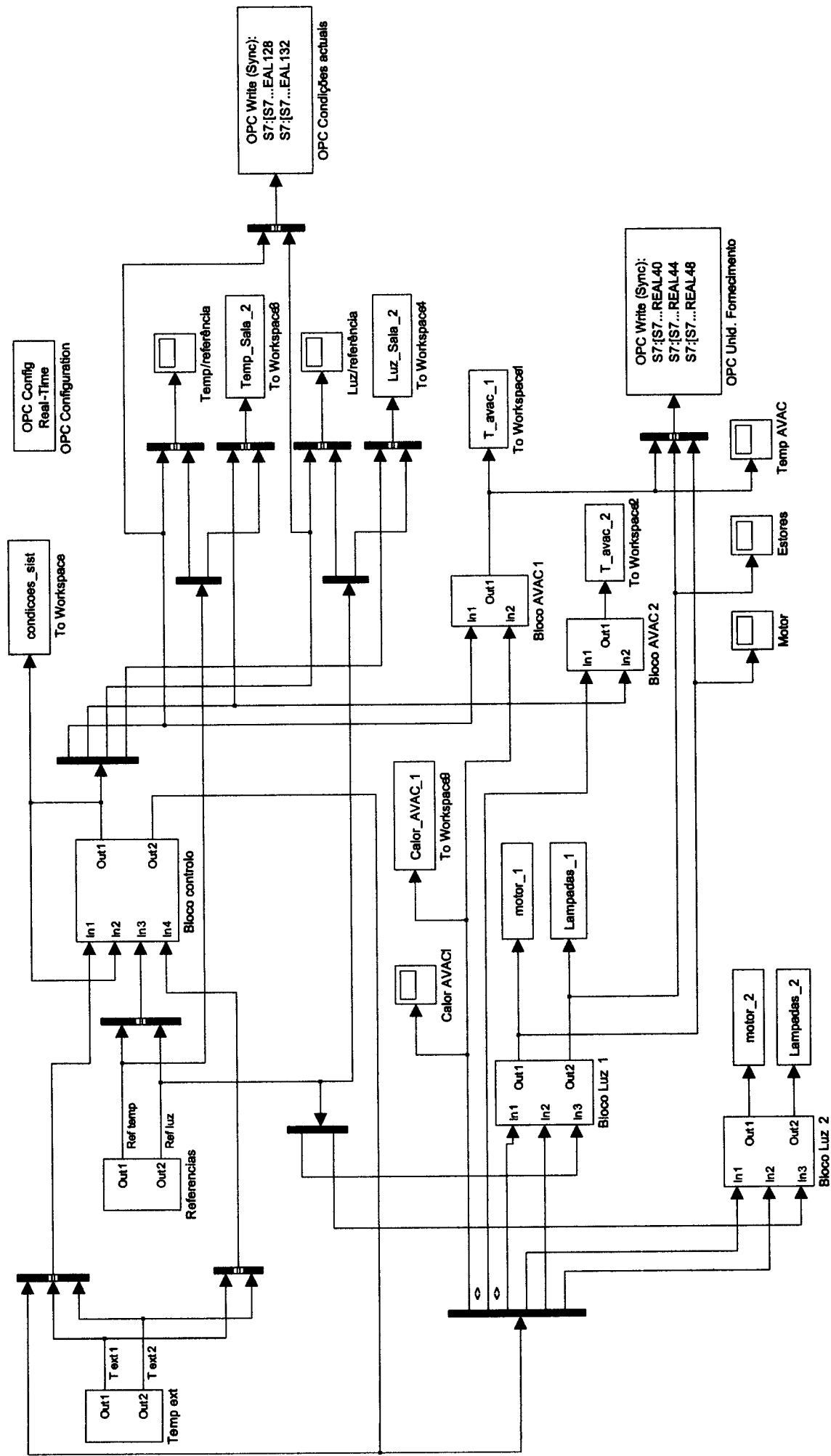
API: MPI(1) (S7 subnet ID: 0083 - 0001) contains the following network connections:

Station address:	Station:	Module:	R/S:
2	Conference room 1	CPU 315-2 DP	0/2
3	Conference room 1	CP 343-1 IT	0/4
4	TP 1	-	-

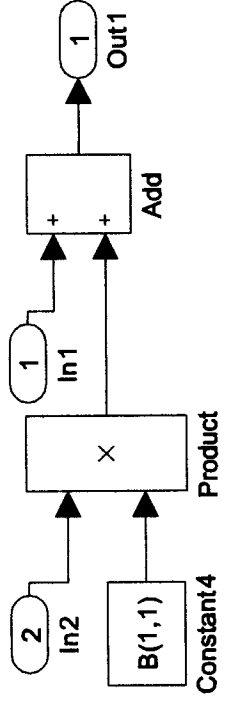
plataforma_interactiva (C:\Documents and Settings\PROJ_PETER\Desktop\Tese Eduardo\simatic\platafor)

the project contains no cross-project subnets

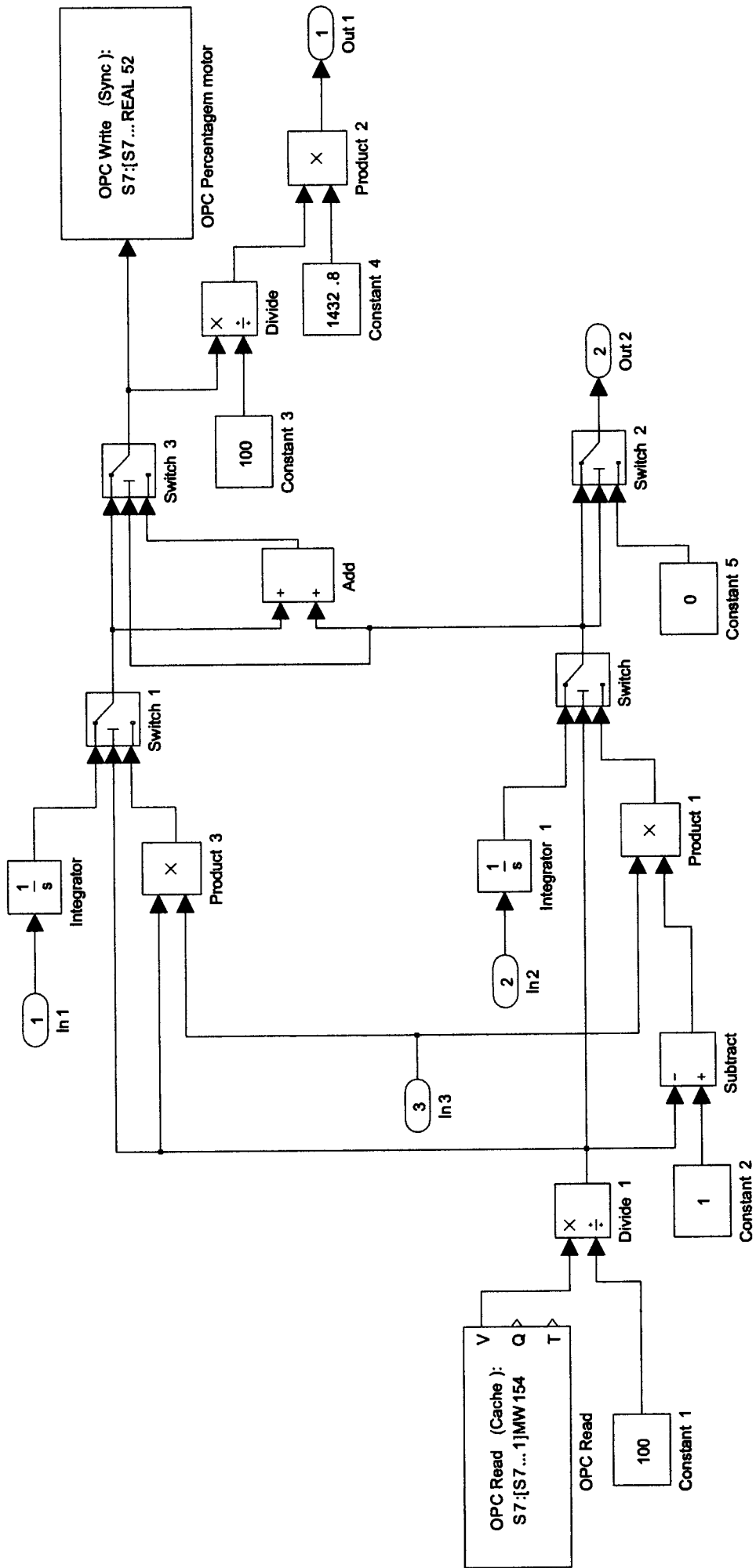
Anexo I - Matlab



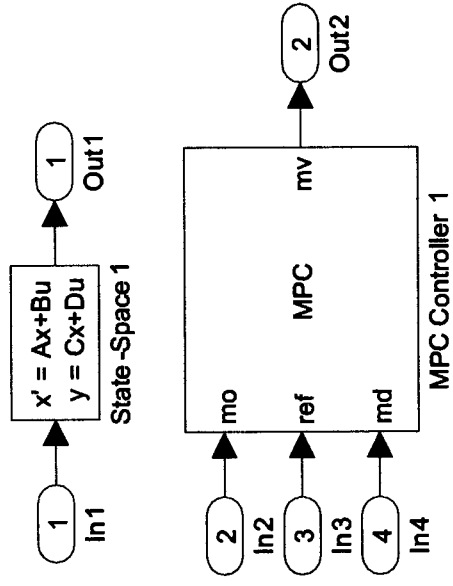
Bloco AVAC 1



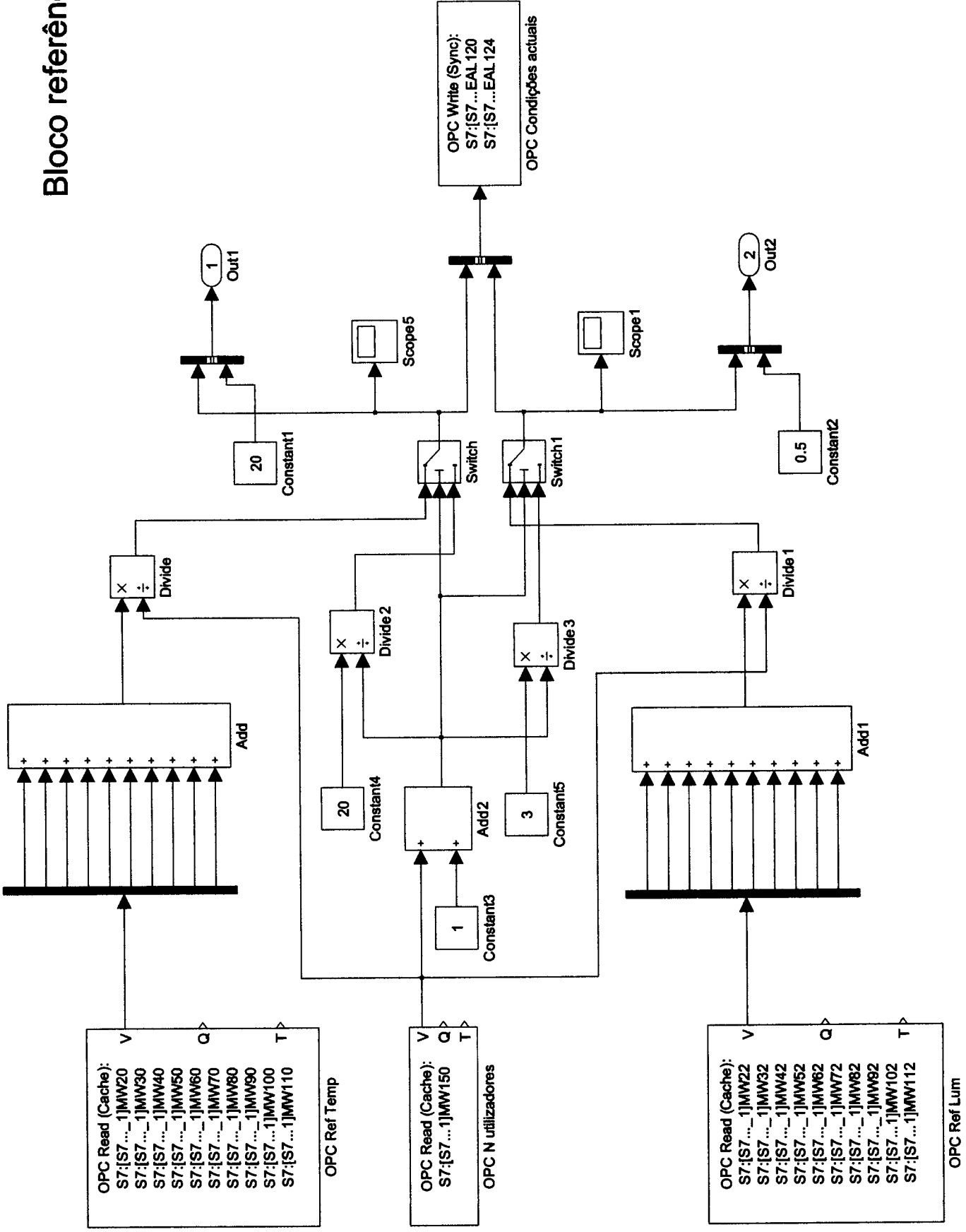
Bloco Luz 1



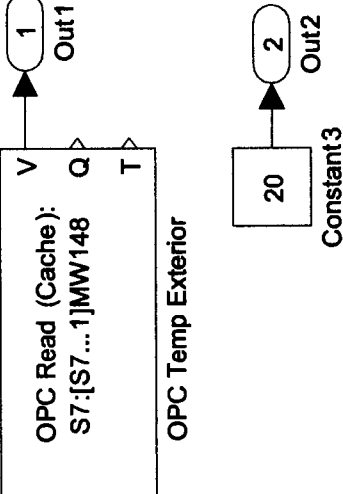
Bloco controle



Bloco referências



Bloco Temp. ext



```
clear
clc

% Criação das matrizes do sistema em espaço de estados
A=[-1.72e-4    0    0 0;
    0    -1.72e-4    0 0;
    0    0    0 0;
    0    0    0 0];

B=[1.72e-6    0    0 0 0 0 1.72e-4    0;
    0    1.72e-6    0 0 0 0    0    1.72e-4;
    0    0    1 0 1 0    0    0;
    0    0    0 1 0 1    0    0];

C=[1 0 0 0;
    0 1 0 0;
    0 0 1 0;
    0 0 0 1];

D=[0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0];

condicoes_ini=[20;
               20;
               0;
               0];

% Criação do modelo do controlador
sys=ss(A,B,C,D)

% Atribuição das características do controlador
sys.InputName = {'q_avac1', 'q_avac2', 'Estores_1', 'Estores_2', 'Luz_art1', 'Luz_art2',
'T_ext1', 'T_ext2'};
sys.OutputName = {'T1', 'T2', 'Luz_int1', 'Luz_int2'};
sys.StateName = {'T_int1', 'T_int2', 'Luz_sala1', 'Luz_sala2'};
sys=setmpcsignals(sys, 'MD', 7:8);
sys

% Carregamento do controlador criado
load MPC1

% Arranque do ficheiro Simulink
Controlo_preditivo
```