

Marisa de Matos Oliveira

Comando à distância de sistema doméstico



Marisa de Matos Oliveira

Comando à distância de sistema doméstico

Tese de Mestrado

Engenharia Eletrotécnica - Energia e Automação Industrial

Professor Doutor Miguel Francisco Martins de Lima



“Success never comes around to look for you while you wait around. You’ve got to get up and work at it to make your dreams come true.”

Poh Yu Khing

RESUMO

O ritmo de evolução que a sociedade atual atravessa, tanto a nível científico como tecnológico, conduz o homem a aumentar, cada vez mais, as suas exigências no seio do seu ambiente doméstico, possibilitando que as tarefas mais rotineiras e maçadoras sejam realizadas automaticamente. A domótica surge assim como uma aliada do homem, não só porque lhe proporciona o bem-estar tão desejado como também lhe proporciona uma maior segurança, gestão e consciencialização sustentável em termos de despesas com a sua habitação.

São já vários os parâmetros de controlo possíveis numa habitação. O melhor é que a localização já não é limite. Com o acesso à internet, o controlo da habitação e de todas as suas componentes, passa a poder ser realizado em qualquer parte do mundo.

Neste trabalho desenvolveu-se um sistema capaz de comandar à distância um sistema doméstico. Implementou-se num PLC S7-1200 um conjunto de funcionalidades domésticas, podendo estas serem avaliadas através de registos históricos e geridas à distância (web, wifi, gsm) recorrendo a tecnologias como página web, sms, app e e-mail.

ABSTRACT

The pace of evolution that the present society is going through, both at scientific and technological levels, leads man to increase, even more, his requirements within his home environment, enabling that the most routine and boring tasks, should be done automatically. Home automation comes as an ally of man, not only because it provides the so desired well-being, but also because it provides high security, management and sustainable awareness in terms of expenses with your housing.

Nowadays, there are already several possible control parameters in a house. The best is that location is no longer the limit. With internet access, the control of a house and all its components, can now be done from anywhere in the world.

In this project, we have developed a system capable of control, at long distance, a home automation system. It was implemented in a PLC S7-1200 a set of home automation features, that can be evaluated through historical records and remotely managed (web, WiFi, GSM) using technologies such as a website, SMS, an app and email.

PALAVRAS CHAVE

Domótica
Casa inteligente
PLC
Comunicações

KEY WORDS

Home automation
Smart home
PLC
Communications

AGRADECIMENTOS

Eu gostaria de agradecer:

Aos meus pais, por todos os sacrifícios, pelo apoio incondicional que me prestaram neste meu percurso, por acreditarem sempre em mim e naquilo que faço e por todos os ensinamentos de vida, a eles, um bem-haja. Ficar-lhes-ei grata pelo resto da minha vida.

À minha irmã pelas palavras repletas de conforto e amizade nos momentos mais propícios, pela transmissão de confiança e apoio nas minhas escolhas e projetos.

Aos meus sobrinhos, pela alegria e afeto. A eles peço desculpa pela atenção que não lhes pude proporcionar.

Ao Tiago, por estar sempre ao meu lado a lutar pelos meus objetivos. Pela paciência nos dias mais difíceis, pelo carinho e amor, pelas ideias prestadas e acima de tudo pela força e conselhos durante as minhas crises e dificuldades. A ele o meu sincero agradecimento.

Ao meu orientador, prof. doutor Miguel Francisco Martins de Lima, pela receptividade, pelas sugestões, pelo incentivo e auxílio. A ele um enorme obrigado.

Ao técnico João Ôlas pelas ideias e disponibilidade de material.

Aos meus colegas Daniel, Óscar, Miguel pelo companheirismo e amizade.

Ao Departamento de Engenharia Eletrotécnica da Escola Superior de Tecnologia e Gestão de Viseu.

Àqueles que, não estando aqui referidos, mas que contribuíram para a realização desta dissertação de mestrado.

ÍNDICE GERAL

ÍNDICE GERAL	xiii
ÍNDICE DE FIGURAS	xvii
ÍNDICE DE SECÇÃO DE CÓDIGO.....	xxi
ÍNDICE DE TABELAS	xxiii
ABREVIATURAS E SIGLAS	xxv
1. Introdução.....	1
1.1 Contexto.....	1
1.2 Objetivos.....	2
1.3 Estrutura da Dissertação	3
2. Domótica	5
2.1 Definição.....	5
2.2 Funcionalidades	6
2.2.1 Conforto.....	6
2.2.2 Eficiência energética.....	7
2.2.3 Segurança.....	7
2.2.4 Comunicação	8
2.3 Elementos constituintes de um Sistema Domótico.....	8
2.4 Arquitetura dos Sistemas Domóticos.....	9
2.4.1 Arquitetura centralizada	9
2.4.2 Arquitetura descentralizada	10
2.4.3 Arquitetura distribuída.....	10
2.4.4 Arquitetura Híbrida/Mista	11
2.5 Meios de transmissão de dados.....	12
2.6 Protocolos de comunicação.....	13
2.6.1 Método de comunicação para evitar colisões de mensagens.....	15
2.6.2 Protocolo X-10	15
2.6.3 Protocolo KNX.....	22

ÍNDICE GERAL

2.6.4	Protocolo LonWorks	31
2.6.5	Tecnologia baseada em PLC	37
2.7	Comparação entre protocolos	39
3.	Desenvolvimento e comando do sistema domótico	41
3.1	PLC S7-1200	41
3.1.1	Aspetos do <i>Hardware</i> de um PLC	42
3.1.2	Aspetos do <i>Software</i> de um PLC	44
3.1.3	Ciclo de processamento	44
3.2	Aplicação de desenvolvimento Tia Portal	46
3.2.1	Sinais Analógicos	50
3.2.2	Leitura de sinais analógicos	51
3.2.3	Funcionalidades implementadas	52
3.3	Aplicação Android	74
3.3.1	Android Studio	75
3.3.2	Estrutura do <i>Android Studio</i>	75
3.3.3	Estrutura de um Projeto <i>Android</i>	77
3.3.4	Comunicação <i>Android</i> - PLC S7-1200	79
3.3.5	Aplicação <i>Android</i>	82
3.4	Página WEB	87
3.4.1	Noção do documento HTML	87
3.4.2	Servidor Web	89
3.4.3	Programação	90
3.4.4	Acesso às páginas criadas	95
3.5	Comunicação GSM	96
3.5.1	Set/Get	100
3.6	Maquete desenvolvida	102
3.6.1	Alimentação elétrica	103
3.6.2	Massas	104
3.6.3	Funcionalidades e Equipamentos	104
4.	Conclusões e trabalhos futuros	115
	REFERÊNCIAS	117

ANEXO 1	121
ANEXO 2	122
ANEXO 3	124
ANEXO 4	125

ÍNDICE DE FIGURAS

Figura 1- 1: Tecnologias de gestão do sistema domótico.....	2
Figura 2-1: Funcionalidades domóticas.....	6
Figura 2-2: Arquitetura centralizada.	9
Figura 2-3: Arquitetura descentralizada.	10
Figura 2-4: Arquitetura distribuída.....	11
Figura 2-5: Arquitetura hibrida/mista.....	11
Figura 2-6: Meios de transmissão.....	12
Figura 2-7: Camadas do modelo OSI.	13
Figura 2-8: CSMA.....	15
Figura 2-9: Exemplo do protocolo X-10.	16
Figura 2-10: Dispositivos X-10.....	16
Figura 2-11: Recetores X-10.	17
Figura 2-12: Exemplos de dispositivos X-10.	18
Figura 2-13: Disposição dos dispositivos X-10.....	19
Figura 2-14: Injeção de sinal na rede elétrica.....	19
Figura 2-15: Injeção de sinais na rede elétrica.	20
Figura 2-16: Envio do sinal binário 1 e 0.	20
Figura 2-17: Códigos X-10.....	21
Figura 2-18: Sistema KNX.....	23
Figura 2-19: Estrutura de um sistema KNX.	23
Figura 2-20: Exemplo de arquitetura centralizada.	24
Figura 2-21: Topologias KNX.....	24
Figura 2-22: Dispositivos KNX.....	25
Figura 2-23: Configuração dos dispositivos no software ETS.....	26
Figura 2-24: Síntese dos modos de configuração.....	27
Figura 2-25: Transmissão de sinais no sistema KNX.....	28
Figura 2-26: Sequência da transmissão de sinais	28
Figura 2-27: Endereço físico.	29
Figura 2-28: Endereço de grupo de 2 e 3 níveis.....	30
Figura 2-29: Comunicação.	30
Figura 2-30: Sistema <i>LonWorks</i>	32
Figura 2-31: Dispositivo LON.....	33
Figura 2-32: <i>Neuron Chips</i>	33
Figura 2-33: Constituintes de uma rede LON.	34
Figura 2-34: Ligação entre nós.....	35
Figura 2-35: Ligação entre variáveis na rede.	35

ÍNDICE DE FIGURAS

Figura 2-36: Troca de informação num sistema <i>LonWorks</i>	36
Figura 2-37: Comparação entre um sistema convencional e o sistema <i>LonWorks</i>	36
Figura 2-38: Ciclo de funcionamento de um PLC.....	37
Figura 2-39: Resultados do inquérito realizado acerca dos protocolos de comunicação.....	39
Figura 3-1: PLC S7-1200 1214C AC/DC/RLY.....	42
Figura 3-2: Hardware de um PLC.....	42
Figura 3-3: Tipos de memória num PLC.....	43
Figura 3-4: Ciclo de processamento de um PLC.....	45
Figura 3-5: Estrutura do TIA Portal.....	46
Figura 3-6: Ambiente de programação do TIA Portal.....	47
Figura 3-7: Constituintes do TIA Portal.....	47
Figura 3-8: Blocos do Tia Portal.....	49
Figura 3-9: Conversão de variáveis.....	50
Figura 3-10: Conversão digital de variáveis analógicas.....	50
Figura 3-11: Normalização de variáveis analógicas.....	51
Figura 3-12: Mapeamento de valores.....	51
Figura 3-13: Sensor de Temperatura LM35.....	52
Figura 3-14: Resultado da temperatura com a média deslizante.....	54
Figura 3-15: Tarefas agendadas, função do PC.....	56
Figura 3-16: Interface <i>Web Server</i> do PLC.....	57
Figura 3-17: Painel solar com estore.....	64
Figura 3-18: Sensor de humidade do solo.....	66
Figura 3-19: Sensor magnético.....	69
Figura 3-20: Blocos gerados.....	73
Figura 3-21: <i>Ping</i> na linha de comandos.....	74
Figura 3-22: Propriedades do PLC.....	74
Figura 3-23: Estrutura do <i>Android Studio</i>	75
Figura 3-24: Estrutura do projeto.....	76
Figura 3-25: Estrutura do <i>layout</i> do projeto.....	76
Figura 3-26: <i>Java Class</i>	77
Figura 3-27: Conteúdo da pasta <i>res</i>	78
Figura 3-28: <i>Android Manifest</i>	78
Figura 3-29: <i>Intents</i>	79
Figura 3-30: Conteúdo Moka 7.....	79
Figura 3-31: <i>Layout</i> da primeira página da aplicação desenvolvida.....	80
Figura 3-32: Serviço <i>dynIP</i>	81
Figura 3-33: Endereço do PLC na DMZ.....	81
Figura 3-34: Propriedades do <i>Data Block</i>	82
Figura 3-35: <i>Widgets</i>	82

Figura 3-36: Identificação do PLC.	83
Figura 3-37: <i>Layout</i> e os seus constituintes.	85
Figura 3-38: Propriedades do Servidor <i>Web</i> do PLC.	89
Figura 3-39: Pasta com as várias páginas criadas.	89
Figura 3-40: <i>Refresh</i> da página não recomendado.	92
Figura 3-41: Blocos gerados a partir do código HTML desenvolvido.	94
Figura 3-42: Descodificação do código.	95
Figura 3-43: Interface servidor <i>Web</i> do PLC.	95
Figura 3-44: PLC e o módulo GSM.	96
Figura 3-45: Propriedades do módulo GSM.	97
Figura 3-46: Formato das SMS e os comandos respetivos.	100
Figura 3-47: Maquete.	102
Figura 3-48: Equipamentos, circuitos e componentes.	102
Figura 3-49: Transformador (230V AC – 12V DC).	103
Figura 3-50: DC-DC <i>buck converter</i>	103
Figura 3-51: Sensor de temperatura e ventoinhas referentes à climatização.	104
Figura 3-52: Ligações do relé.	105
Figura 3-53: Ligação da fita led.	105
Figura 3-54: Iluminação.	105
Figura 3-55: Circuito referente à variação da intensidade luminosa.	106
Figura 3-56: Fita led sujeita à variação da intensidade luminosa.	106
Figura 3-57: Circuito referente à rotação do motor em dois sentidos.	107
Figura 3-58: Circuito referente à rotação de dois motores em dois sentidos.	108
Figura 3-59: Circuito referente ao PWM.	108
Figura 3-60: Circuito capaz de controlar velocidades e sentidos de rotação dos motores.	109
Figura 3-61: Ligação do relé.	109
Figura 3-62: Protótipo de um estore.	110
Figura 3-63: Protótipo de um portão automático.	110
Figura 3-64: Sensor de humidade do solo e seu módulo controlador.	111
Figura 3-65: Sensor instalado no jardim da maquete.	111
Figura 3-66: Sensor magnético instalado na porta de entrada da maquete.	112
Figura 3-67: Besouro.	112
Figura 3-68: Circuito referente à intrusão.	112
Figura 3-69: Caixa de correio.	113

ÍNDICE DE SECÇÃO DE CÓDIGO

Secção de código 3-1: Leitura do sensor de temperatura.	53
Secção de código 3-2: Média deslizante.....	53
Secção de código 3-3: Modos de comando da variável correspondente ao aquecimento.	55
Secção de código 3-4: Bloco de função “Toggle_Preset”.....	57
Secção de código 3-5: Obtenção da saída analógica.	59
Secção de código 3-6: Programação horária.	59
Secção de código 3-7: Descida automática do estore.....	60
Secção de código 3-8: Inibição de recorrência de uma descida	61
Secção de código 3-9: Espera entre operações.....	61
Secção de código 3-10: Conversão da variável em impulsos.....	62
Secção de código 3-11: Contador.....	62
Secção de código 3-12: Conversão da variável em impulsos.....	62
Secção de código 3-13: Ativação e comparação para posições do estore.	63
Secção de código 3-14: Barreira de raios infravermelhos.....	65
Secção de código 3-15: Luz intermitente.	66
Secção de código 3-16: Humidade e consequente ativação de variáveis.....	67
Secção de código 3-17: Programação da rega automática.	68
Secção de código 3-18: Modos de comando da rega.....	68
Secção de código 3-19: Modo automático do sistema de intrusão.....	69
Secção de código 3-20: Sinalização de intrusão.....	70
Secção de código 3-21: Bloco de função “ClockAlarmWithOnTimeDuration”.....	71
Secção de código 3-22: Aviso de existência de correspondência.	72
Secção de código 3- 23: Instrução “TM_MAIL”.....	72
Secção de código 3-24: Parâmetros imprescindíveis no envio de e-mails.....	73
Secção de código 3-25: Permissão de internet.	80
Secção de código 3-26: Importação de bibliotecas, <i>widjets</i> , entre outros.	83
Secção de código 3-27: Leitura da temperatura.	84
Secção de código 3-28: Resposta para variáveis booleanas.....	85
Secção de código 3-29: Escrita de variáveis.	85
Secção de código 3-30: Leitura e escrita de variáveis.....	86
Secção de código 3-31: Verificação automática do estado para agir em conformidade.	86
Secção de código 3-32: Estrutura do documento HTML.....	90
Secção de código 3-33: Inserção da variável do PLC no documento.	91
Secção de código 3-34: Menu.	91
Secção de código 3-35: Leitura da variável.....	92
Secção de código 3- 36: <i>Gifs</i>	92
Secção de código 3-37: Escrever valores nas variáveis reais do PLC.....	92
Secção de código 3-38: Atualização de variáveis.	93

ÍNDICE DE SECÇÃO DE CÓDIGO

Secção de código 3-39: Modo de alterar valores das variáveis booleanas.	93
Secção de código 3-40: <i>Script</i> com variável “op”	94
Secção de código 3-41: Instrução WWW do TIA Portal.....	94
Secção de código 3-42: <i>SMS_Send</i>	98
Secção de código 3-43: <i>SMS_Recv</i>	98
Secção de código 3-44: <i>Escalation</i>	99
Secção de código 3-45: Descodificação de mensagens recebidas pelo PLC.....	101
Secção de código 3-46: Descodificação de mensagens recebidas pelo PLC.....	101
Secção de código 3-47: Descodificação de mensagens recebidas pelo PLC.....	101

ÍNDICE DE TABELAS

Tabela 2-1: Comparação entre o modelo OSI e o sistema KNX.....	29
Tabela 3-1: Tipos de dados do TIA Portal	49
Tabela 3-2: Parte dos registos da temperatura.....	56
Tabela 3-3: Propriedades CSS.....	88
Tabela 3-4: Tags.....	88

ABREVIATURAS E SIGLAS

A	Ampere
APN	Access Point Name
AVAC	Aquecimento, Ventilação e Ar Condicionado
CSMA	Carrier Sense Multiple Access
°C	Grau Celsius
FBD	Diagrama de Blocos Funcionais
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IV	Infravermelhos
LED	Light Emitting Diode
LDR	Light Dependent Resistor
M2M	Machine to Machine
NEMA	National Electrical Manufacturers Association
NTP	Network Time Protocol
PIN	Personal Identification Number
PLC	Controladores Lógicos Programáveis
PV	Painéis Fotovoltaicos
SCL	Structured Control Language
SMS	Short Message Service
SMSC	Short Message Service Center
SMTP	Simple Mail Transfer Protocol
SNVT	Standard Network Variable Type
TCP	Transmission Control Protocol
V	Volt
VPN	Virtual private network
WWW	World Wide Web

1. Introdução

1.1 Contexto

Nos dias de hoje, vivemos uma verdadeira revolução tecnológica. Constantemente são anunciados novos conceitos, novos desenvolvimentos tecnológicos, novas alusões futuristas, entre outros. A par do momento que se vive, regista-se cada vez mais um aumento na procura de formas de proporcionar comodidade, conforto e segurança, não só num contexto industrial como também num contexto doméstico.

A Automação Residencial ou simplesmente Domótica segue essa linha de raciocínio e, ao longo dos últimos anos tem vindo a expandir-se de forma a propiciar uma melhoria na vida dos seus utilizadores, satisfazendo as suas necessidades de conforto, comunicação, segurança e eficiência energética.

Graças aos bons resultados dos sistemas instalados e à procura extasiante de sistemas domóticos cada vez mais extensíveis, surgiu a motivação pelo projeto, pois possibilita que o utilizador tenha um maior controlo sobre a sua habitação, podendo geri-la de forma rentável quer localmente como remotamente. Tudo isto se torna possível aliando conceitos modernos de arquitetura e de construção, que possibilitam o uso mais eficiente de fontes naturais de energia, reduzindo o consumo dos equipamentos.

1.2 Objetivos

O principal objetivo deste trabalho consiste no desenvolvimento de comandos à distância para um Sistema Domótico.

Numa fase inicial, pretende-se implementar um conjunto básico de funcionalidades domóticas em PLC, que podem incluir a gestão de um sistema de climatização, iluminação, estores, rega, intrusão, simulação de presença, entre outros.

Para se poderem avaliar algumas das funcionalidades referidas pode realizar-se um registo histórico das atividades respetivas (*logging*). Em seguida pretende-se que as funcionalidades possam ser geridas à distância (web, wifi, gsm) recorrendo a tecnologias como página web, sms, app e e-mail, como se mostra na Figura 1-1.

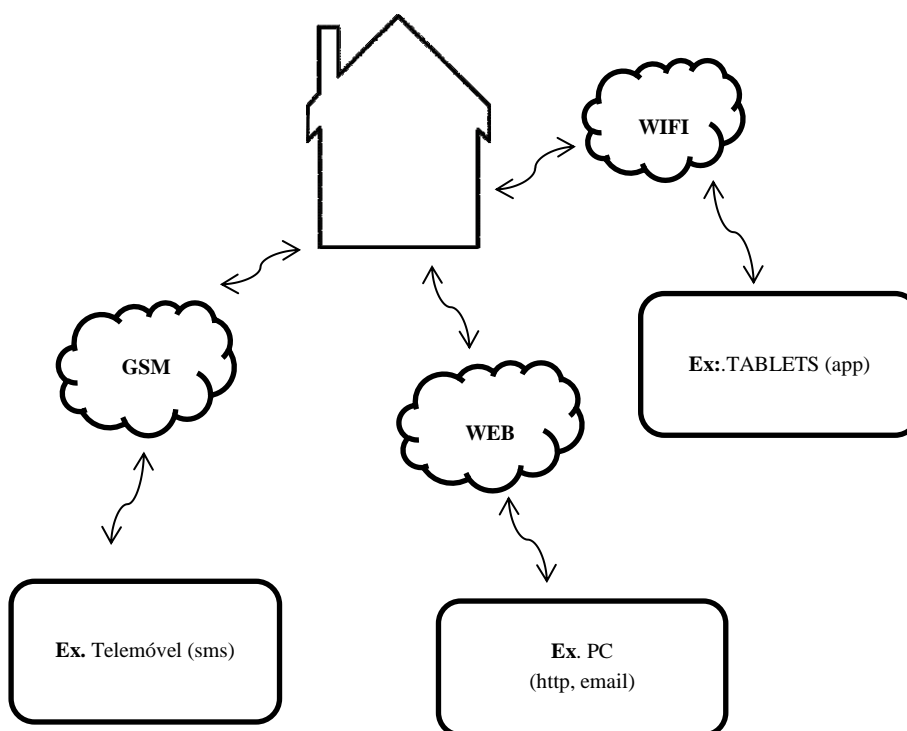


Figura 1- 1: Tecnologias de gestão do sistema domótico.

A nível de programação, o trabalho incide sobre a programação *Ladder/SCL* (S7-1200), criação de páginas Web e desenvolvimento/configuração de app em *android*. Com o intuito de simular todo o sistema, embora não esteja nos objetivos inicialmente propostos, construiu-se uma maquete onde se colocaram os vários componentes e dispositivos a serem comandados.

No entanto existem outros objetivos com a realização deste trabalho que não se prendem apenas na obtenção pessoal de experiência no desenvolvimento de *software*, de *hardware*, na sua integração e no desenvolvimento de *interfaces*.

Assim pretende-se também em desmistificar de um modo geral o caráter futurista da automação residencial, na medida em que é possível construir sistemas rápidos, únicos, específicos e inteligentes sem grandes necessidades protocolares, apenas recorrendo à automação e à eletrónica. De notar que num passado relativamente próximo a domótica era vista como um luxo. Hoje, cada vez mais se torna uma necessidade, simplificando diariamente a vida das pessoas, nomeadamente as que são portadoras de determinadas deficiências, em que um simples gesto, como fechar um estore, exige muito esforço. Assim, o manuseamento de um sistema domótico pode ser executado em conformidade com as necessidades de cada um. Em sistemas passivos o elemento reage apenas quando lhe é transmitida uma ordem, dada diretamente pelo utilizador ou por um comando. Nos sistemas automáticos e inteligentes, o sistema não só interpreta parâmetros, como também reage às circunstâncias.

Nesta dissertação irão ser apresentadas propostas para a implementação de várias funcionalidades habitacionais, assim como diversos meios de as comandar à distância.

Para tal, serão estudadas as tecnologias que permitem suportar esta implementação e, com base neste estudo, serão apresentadas soluções.

1.3 Estrutura da Dissertação

O presente trabalho encontra-se organizado em quatro capítulos e dois anexos. Em seguida serão descritos sucintamente os conteúdos presentes em cada capítulo e anexos. No primeiro capítulo de introdução apresenta-se o contexto, a motivação, a necessidade e/ou pertinência e os objetivos do projeto. O capítulo 2 apresenta algumas particularidades da domótica e/ou sistemas domóticos, funcionalidades, elementos constituintes, arquiteturas existentes, meios de transmissão de dados, protocolos e vantagens e desvantagens de uns em relação aos outros. No capítulo 3 propõe-se uma solução para o tema em causa. Por fim, no capítulo 4 apresentam-se as conclusões que resultaram do estudo e elaboração do projeto, referindo-se ainda aspetos de evolução futura. Para além dos capítulos referidos existem quatro anexos. No anexo 1 apresentam-se os *layouts* da aplicação *android*. No anexo 2 mostra-se alguns exemplos de páginas Web desenvolvidas. No anexo 3 apresentam-se, como exemplo, alguns comandos desenvolvidos no TIA Portal relativos à leitura e à escrita de variáveis do PLC através de SMS. Por fim, no anexo 4 apresentam-se duas tabelas com as entradas e saídas utilizadas no TIA Portal e a respetiva função.

2. Domótica

Neste capítulo será abordado o tema domótica e serão apresentadas algumas particularidades nomeadamente, as funcionalidades, os elementos constituintes, as arquiteturas existentes, os meios de transmissão de dados e os principais protocolos existentes.

2.1 Definição

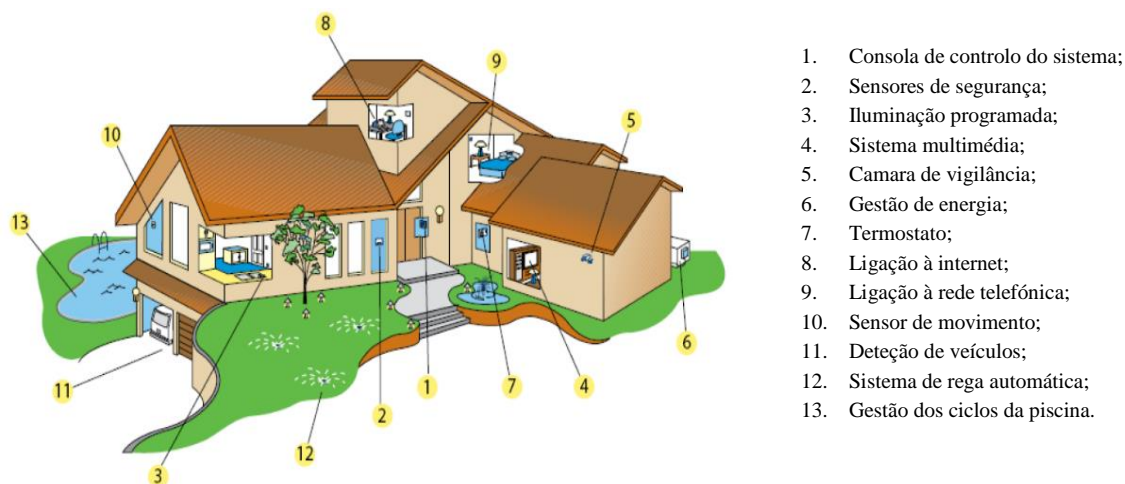
O termo “Domótica” resulta da junção da palavra “*Domus*” (casa) com a palavra “Robótica” (controlo automatizado de algo) e é essencialmente este último conceito que rentabiliza todo o sistema, simplificando a vida diária das pessoas, fazendo com que as tarefas mais rotineiras e maçadoras sejam executadas de forma automática e com ausência de preocupação.

A domótica surgiu em meados dos anos 80. Nos últimos anos a domótica tem sido alvo de investigação em áreas tão diversas como a iluminação [1], a eficiência energética [2], estudo de novos materiais para a construção, entre outras. A investigação aplicada que se faz no âmbito da domótica recorre a diferentes abordagens, onde se inclui a inteligência artificial (ex: lógica *fuzzy* e redes neuronais) [3], o controlo ótimo e técnicas de otimização [4] [5].

Inicialmente a ideia da domótica baseava-se sobretudo no controlo de iluminação, de condições térmicas e a interligação entre os vários elementos. Hoje, fruto da investigação e desenvolvimento entretanto ocorridos, a domótica vem proporcionar ainda mais benefícios. Pois para além de apresentar circuitos que controlam, verificam e comparam as mais diversas funções, todos com o objetivo de automatizar o ambiente doméstico, ainda permite ter acesso a equipamentos através da Internet, bem como verificar o seu estado. Assim, o controlo remoto deixa de ser uma utopia para ser uma realidade dos nossos dias [6].

2.2 Funcionalidades

Quando se pretende instalar um sistema domótico, uma das primeiras coisas a serem consideradas são as funcionalidades que se pretendem implementar, normalmente correspondentes às necessidades de cada utilizador e/ou habitação (Figura 2-1). Essas funcionalidades podem ser divididas em quatro áreas: Conforto, Segurança, Eficiência Energética e Comunicação [7] [8] [9]. São estas funcionalidades que vão contribuir para gerir e otimizar os recursos.



1. Consola de controlo do sistema;
2. Sensores de segurança;
3. Iluminação programada;
4. Sistema multimédia;
5. Camara de vigilância;
6. Gestão de energia;
7. Termostato;
8. Ligação à internet;
9. Ligação à rede telefónica;
10. Sensor de movimento;
11. Detecção de veículos;
12. Sistema de rega automática;
13. Gestão dos ciclos da piscina.

Figura 2-1: Funcionalidades domóticas [9].

2.2.1 Conforto

Esta funcionalidade tem como principal função proporcionar conforto e está maioritariamente associado às instalações AVAC(aquecimento, ventilação e ar condicionado), embora também se possam incluir nesta área todos os outros sistemas que possam contribuir para a comodidade e bem-estar dos utilizadores. As funções aplicáveis no domínio do conforto são, por exemplo [7] [8],

- Controlo da climatização;
- Controlo de iluminação;
- Automatização da irrigação de jardins;
- Abertura e fecho automático de portas;
- Controlo de *SmartGlass*¹;
- Criação de cenários;
- Acionamento automático de vários sistemas com base em dados do meio ambiente, como por exemplo o fecho de estores e janelas, em caso de tempestade ou vento forte;
- Auxiliar de memória (lembrar a toma de medicamentos, etc.).
- Apoio a pessoas idosas, doentes ou com deficiências (permitir maior autonomia e telemedicina);

¹ SmartGlass - Vidros com características controláveis eletricamente

- Comando de aparelhos de entretenimento (vídeo, áudio e multimédia);

2.2.2 Eficiência energética

A eficiência e/ou otimização dos consumos energéticos é dos conceitos mais importantes da instalação de um sistema domótico. É uma realidade que não implica a ausência do consumo, mas sim a sua gestão e conseqüente racionalização (por exemplo, uso de uma UPS em caso de falha de energia). O objetivo passa por satisfazer as necessidades domésticas com o mais baixo custo possível. As ações aplicáveis no domínio da eficiência energética são, por exemplo [7] [8],

- O aproveitamento das tarifas reduzidas;
- Detecção de fontes de perdas;
- Atuação, por exemplo, sobre as persianas de modo a que seja aproveitada a luz solar;
- Monitorização e registo de consumos, de forma a poder otimizar a eficiência.

2.2.3 Segurança

A segurança constitui uma preocupação crescente e atualmente é cada vez maior o número de interessados que a colocam no topo das suas preferências. As tarefas aplicáveis no domínio da segurança são, por exemplo [7] [8],

- Iluminação de zonas de risco;
- Videovigilância do exterior e interior da habitação;
- Gerar alarmes técnicos em situações de emergência ou anómalas em caso de:
 - Inundação;
 - Fuga de gás e água;
 - Falta de energia;
 - Detecção de fumo e fogo;
 - Forno, luzes, esquentador ou fogão estarem ligados;
 - Portas ou janelas abertas;
 - Alarme médico (monitorização e diagnóstico remoto de sinais vitais).
- Corte automático da água e gás em caso de ocorrência de fuga;
- Acionamento automático dos serviços de segurança:
 - Alerta a moradores (SMS, e-mail, chamada telefónica);
 - Telefonar para bombeiros e/ou polícia.

2.2.4 Comunicação

Atualmente, as comunicações são um outro alvo de crescente investimento. De entre as várias possibilidades abrangidas nesta área, são de particular interesse as iniciativas de supervisão, controlo e monitorização das instalações à distância como, por exemplo [7] [8]:

- Controlo/Comando da habitação através de mensagens SMS;
- Controlo/Comando da habitação através de uma aplicação *Android*;
- Controlo/Comando da habitação via TCP/IP usando um *browser* e tecnologias Web.

2.3 Elementos constituintes de um Sistema Domótico

A amplitude de serviços que um sistema domótico é capaz de fornecer pode variar de um único elemento, que executa uma única ação, para grandes sistemas compostos por vários elementos que controlam praticamente todas as instalações dentro da habitação [10] [11]. Esses elementos podem classificar-se como:

- **Sensores** – Os sensores são os olhos e ouvidos da rede doméstica. São responsáveis por captar valores e informações do local onde se encontram inseridos, como por exemplo, a temperatura, humidade, luminosidade, intrusão, presença de pessoas, fugas de água ou gás, incêndio, entre outros.
- **Atuadores** – Os atuadores são as mãos da rede doméstica. Eles são os meios de como a rede inteligente pode realmente fazer as coisas no mundo real, são responsáveis por realizar o controlo de determinados elementos, onde se incluem motores, electroválvulas, ventilação, sirenes de alarme.
- **Controladores** – Os controladores são dispositivos responsáveis por receber a informação proveniente dos sensores, processá-la e posteriormente comandar os atuadores.
- **Interfaces** – São dispositivos que recebem e transmitem informação do utilizador. Toma-se como exemplos, o teclado, os telemóveis, a televisão, os terminais LCD, entre outros.
- **Dispositivos específicos** - São dispositivos cruciais no funcionamento de um sistema, pois permitem o envio de informação entre vários dispositivos e permitem também efetuar o controlo de um sistema, tanto remotamente como localmente. Toma-se como exemplo, os Modems, Routers, Telefones, entre outros.

2.4 Arquitetura dos Sistemas Domóticos

Num sistema domótico, como em qualquer outro sistema de controlo, existem diferentes formas dos elementos interagirem e se localizarem. Existem essencialmente quatro arquiteturas básicas, a arquitetura centralizada, a arquitetura descentralizada, a arquitetura distribuída e a arquitetura híbrida/mista.

2.4.1 Arquitetura centralizada

A arquitetura centralizada caracteriza-se por possuir uma unidade central, o controlador, responsável por processar toda a informação e agir em conformidade [12] [13] (Figura 2-2).

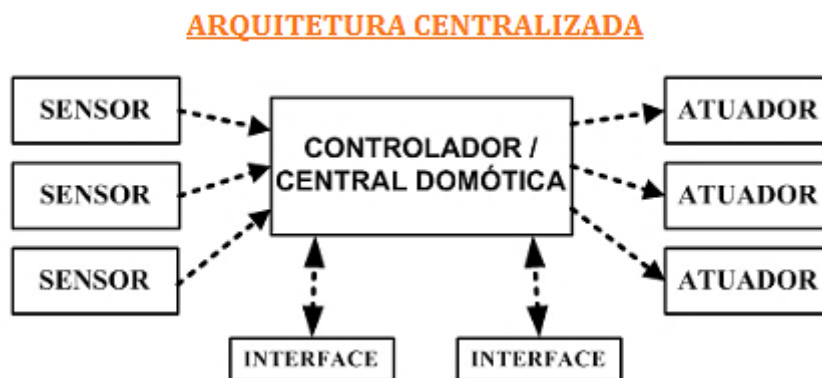


Figura 2-2: Arquitetura centralizada (adaptada de [14]).

O controlador central é o “cérebro” de todo o sistema. Recebe informações de vários sensores e interfaces e, uma vez processadas, gera os comandos apropriados para os atuadores e as suas interfaces. É um elemento fundamental, no entanto, pode falhar, comprometendo todo o sistema. Esta característica é, sem dúvida, a sua principal desvantagem.

Os sistemas que apresentam uma arquitetura centralizada são sistemas modulares em que se lhes pode agregar unidades para expandir a sua capacidade. É o tipo de arquitetura típica dos sistemas proprietários².

Um exemplo desta arquitetura pode ser a implementação da domótica com um PLC.

² Sistemas proprietários - São desenvolvidos por uma única empresa e apenas essa empresa, ou outras devidamente licenciadas, fabricam produtos capazes de comunicar entre si.

2.4.2 Arquitetura descentralizada

A arquitetura descentralizada, é constituída por vários controladores repartidos e interligados entre si através de um barramento de dados para trocarem informação [12] [13] [14] (Figura 2-3).

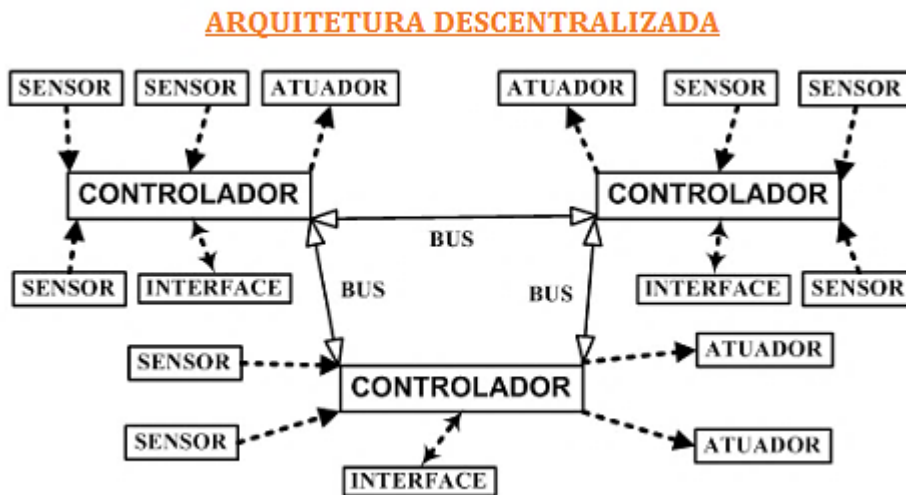


Figura 2-3: Arquitetura descentralizada (adaptada de [14]).

Os sistemas com este tipo de arquitetura são sistemas com grande facilidade de uso, são flexíveis e apresentam uma menor probabilidade de ocorrência de falhas (robustos). Caso ocorram falhas num controlador, apenas compromete o funcionamento desse elemento ou conjunto localizado de elementos, não interferindo com os restantes. No entanto, a sua grande desvantagem incide muitas vezes no seu elevado custo de integração em função das tecnologias empregues.

Um exemplo desta arquitetura pode ser a implementação da domótica com diversos PLC que comunicam entre si. Neste caso, a integração de mais que um PLC é relativamente simples, pois a maioria dos PLC atuais possui mecanismos fáceis de comunicação entre si.

2.4.3 Arquitetura distribuída

A arquitetura distribuída, tal como a arquitetura descentralizada não utiliza um único elemento central, tornando-se mais flexível e imune a falhas, já que a falha de um dos elementos apenas compromete o funcionamento desse mesmo elemento, não pondo em causa o funcionamento dos restantes (Figura 2-4). Na arquitetura distribuída todos os nós presentes na rede têm a capacidade de processar a informação e reagir [12] [13].

É o tipo de arquitetura adotada pelos principais protocolos de domótica disponíveis, nomeadamente X-10 e KNX.

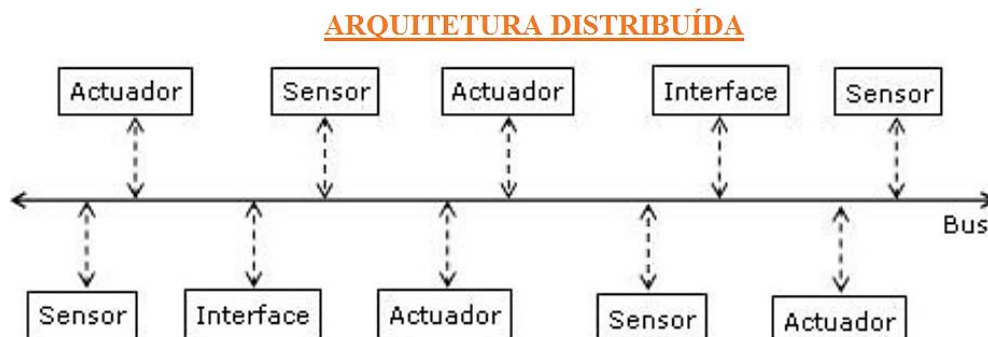


Figura 2-4: Arquitetura distribuída (adaptada de [14]).

2.4.4 Arquitetura Híbrida/Mista

A arquitetura híbrida/mista, como o próprio nome indica, é uma associação das arquiteturas anteriormente citadas. Assim, existem nós capazes de adquirir informação dos sensores, processá-la, reagir e depois enviá-la para os outros nós (Figura 2-5) [13].

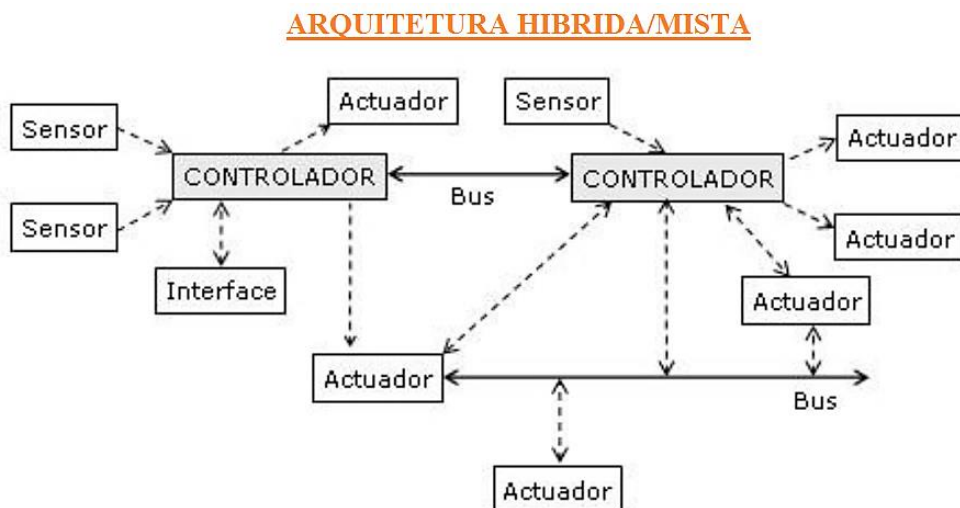


Figura 2-5: Arquitetura híbrida/mista (adaptada de [13]).

2.5 Meios de transmissão de dados

Os meios de transmissão de dados são o suporte físico onde ocorre o fluxo de informação entre os diversos dispositivos de uma rede domótica. Podem ser, a rede elétrica, a cablagem coaxial, a cablagem de baixa tensão, a radiofrequência e raios infravermelhos (Figura 2-6) [10]. Um novo meio começa a conquistar o mercado. A fibra óptica, pois apresenta um elevado rendimento, fiabilidade e imunidade a interferências eletromagnéticas.

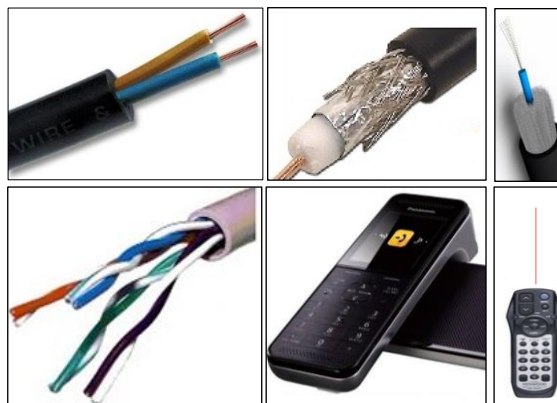


Figura 2-6: Meios de transmissão.

- A rede elétrica (*power line*) é um meio de transmissão viável e que implica menor investimento, pelo simples facto de que se adapta à rede existente. Através da inserção de altas frequências conseguem-se modular sinais na rede elétrica sem interferir com os habituais aparelhos elétricos. Os inconvenientes deste meio de comunicação são as interferências a que esta poderá ser alvo e a baixa velocidade de comunicação comparativamente a outros meios de transmissão.
- A cablagem coaxial continua a ser o meio de eleição, por ser pouco dispendioso e facilmente manipulável (peso, flexibilidade, etc.). Os sistemas que adotam este meio de transmissão pretendem fundamentalmente obter sinais de fraca potência, alta velocidade e alta frequência. Geralmente é utilizada para transmitir sinais de áudio, vídeo ou dados.
- A cablagem de baixa tensão funciona normalmente entre os 5V e os 24V e encontram-se essencialmente nos sistemas de segurança, intercomunicadores, telefones, autômatos industriais, que impliquem baixa potência e baixa frequência. Caracterizam-se por ser de fácil manuseio, instalação e por apresentarem um custo relativamente baixo quando comparado, por exemplo, com a fibra óptica ou a cablagem coaxial. No entanto, este meio de transmissão é suscetível a ruídos e interferências eletromagnéticas e apresenta limitações de distâncias e largura de banda.

- A radiofrequência é um bom meio para transmitir sinais de alta frequência e apresenta uma maior flexibilidade no controlo à distância podendo o sinal ultrapassar paredes e outros obstáculos. Este tipo de transmissão não requer qualquer modificação das infraestruturas nem qualquer instalação especial, no entanto, é bastante sensível às interferências eletromagnéticas, sendo normalmente usada em controlos remotos e em sinais de áudio/vídeo.
- Os raios infravermelhos não necessitam de um meio físico específico (fio) para a transmissão de dados. Permitem que vários dispositivos móveis ou não, possam comunicar com maior facilidade sem a necessidade de uma infraestrutura pré-montada. Oferecem velocidade, segurança (não sofrem interferências de outras ondas) e sigilo, no entanto, estes não possuem a capacidade de atravessar paredes ou objetos.

2.6 Protocolos de comunicação

Para se estabelecer qualquer tipo de comunicação é necessário obrigatoriamente a existência de um emissor, de uma mensagem e de um recetor. Um protocolo de comunicação descreve de forma rigorosa o formato da mensagem a ser transmitida e as respetivas regras de transmissão relativamente à sintaxe, semântica e sincronização da comunicação.

O modelo OSI (*Open Systems Interconnection*), constituído por sete camadas, como se observa na Figura 2-7, descreve as várias fases pelas quais a informação passa até chegar ao destino. No entanto, nem todos os protocolos usados em domótica seguem rigorosamente o modelo, alguns possuem apenas a camada física, lógica e de rede, passando de imediato para a camada de aplicação [15].

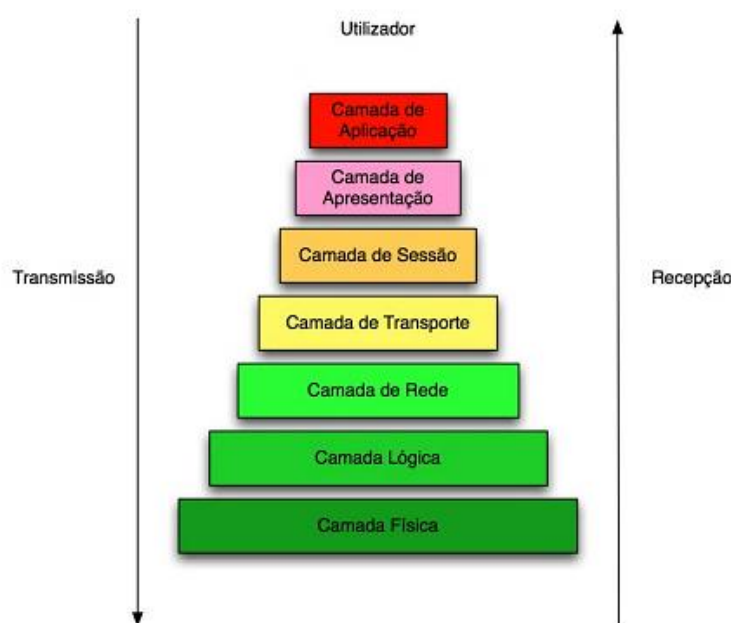


Figura 2-7: Camadas do modelo OSI.

Atualmente o mercado dispõe de dois grandes tipos de protocolos de comunicação que se subdividem em [10]:

- Protocolos proprietários
 - VIS;
 - X2D;
 - CAD;
 - entre outros.

- Protocolos normalizados ou abertos

Na Europa

- BATIBUS;
- EIBUS;
- JBUS;
- MODBUS;
- D2B;
- KNX;
- PROFINET.

Nos Estados Unidos

- CEBUS;
- X-10;
- *SMART HOUSE*;
- *LONWoks*.

No Japão

- HBS.

Relativamente aos protocolos abertos, estes são sobretudo utilizados nas zonas geográficas mencionadas. Todavia, neste trabalho serão apenas descritos alguns dos protocolos.

O protocolo X-10, que em ambiente doméstico é sem dúvida a tecnologia de comunicação mais antiga, mais económica e de mais fácil instalação. O protocolo KNX que é o mais utilizado, fiável e robusto. O Protocolo *LonWoks* que se assemelha ao KNX, sendo-lhe habitualmente comparado e os sistemas que utilizam a tecnologia baseada em PLC.

2.6.1 Método de comunicação para evitar colisões de mensagens

O método de comunicação para evitar colisões de mensagens adotado por alguns sistemas domóticos é o método *Carrier Sense Multiple Access* (CSMA). O CSMA funciona segundo o princípio de que, em cada instante apenas um dispositivo pode transmitir dados, evitando uma colisão e consequentemente perda de informação. Inicialmente quando o dispositivo pretende enviar dados, ele deve escutar quaisquer outras transmissões que possam estar em curso antes de transmitir. Se se detetar uma transmissão, o dispositivo irá esperar e estando aquela concluída, o dispositivo poderá iniciar a transferência de dados [16] [17] (Figura 2-8).

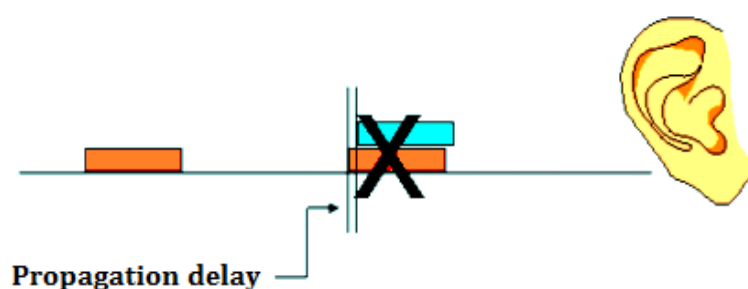


Figura 2-8: CSMA [17].

2.6.2 Protocolo X-10

O protocolo X-10 foi desenvolvido entre 1976 e 1978 pela empresa *Pico Electronics Ltd*, em Glenrother, Escócia e é o protocolo mais antigo e com maior número de módulos vendidos. O seu nome deve-se ao facto de este ser o décimo projeto dos seus autores.

O X10 define a rede elétrica (230V com uma velocidade de comunicação de 50 bits/s em Portugal) como principal meio físico para a transmissão de dados. No entanto, esta exige por parte dos sistemas, filtros e codificadores de modo a não influenciar outros sistemas existentes na vizinhança. Atualmente, o sistema X10 também suporta a radiofrequência (310 MHz nos EUA ou a 443 MHz na Europa), em que os módulos comunicam entre si via ondas rádio (Figura 2-9). Na comunicação não existe deteção de colisões e caso se verifiquem, podem vir a gerar comportamentos imprevistos nos dispositivos [18] [19] [20] [21].

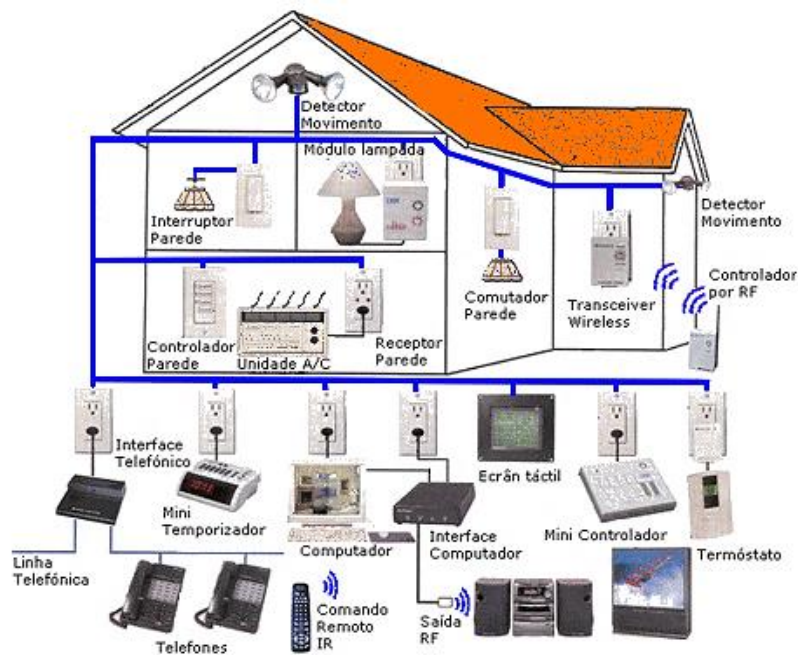


Figura 2-9: Exemplo do protocolo X-10 [21].

2.6.2.1 Dispositivos X-10

O X-10 foi projetado inicialmente para comunicação apenas entre dispositivos X-10. Usualmente estes dispositivos são colocados no lugar de tomadas e interruptores, embora alguns necessitem de caixas elétricas especiais. Existem essencialmente três tipos de dispositivos básicos X-10, como se observa na Figura 2-10. Os dispositivos que só podem transmitir ordens (transmissores), os dispositivos que só podem receber ordens (receptores) e os dispositivos que podem receber e enviar ordens (dispositivos bidirecionais) [7] [22].



Figura 2-10: Dispositivos X-10 [22].

Os transmissores são dispositivos que transmitem ordens e podem endereçar univocamente 256 receptores. Os receptores são adaptadores que se instalam normalmente entre a fonte de corrente elétrica e o dispositivo que se pretende controlar. Os dispositivos apresentam dois comutadores giratórios, um com 16 letras (A a P) e o outro com 16 números (1 a 16), que permitem identificar um endereço dos 256 possíveis (Figura 2-11) [23].

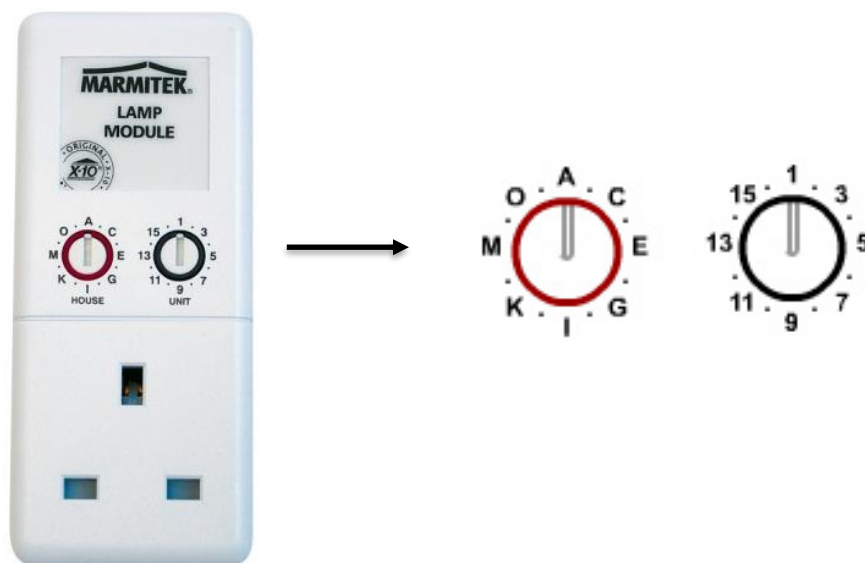


Figura 2-11: Recetores X-10.

A atribuição de endereços aos vários dispositivos é feita manualmente nos próprios dispositivos e cabe ao utilizador assegurar que não existem dispositivos com endereços repetidos. Qualquer recetor pode receber ordens de diferentes transmissores.

Os dispositivos bidirecionais têm a aptidão de responder e confirmar a execução correta de uma ordem (*feedback*).

Uma vez, instalados os dispositivos, também podem ser comandados a partir de qualquer lugar da casa, com controladores à distância, como, por exemplo, por telefone, por mensagem e até com os controladores IP através da internet. Na Figura 2-12 apresentam-se alguns exemplos.

Módulos para carril DIN		Módulos Plug&Play domótica fácil		
Módulo de iluminación con dimmer carril Din	Módulo de potencia carril Din	Módulo de iluminación P&P con dimmer	Módulo de potencia P&P	Casquillo iluminación X-10
CONTROLADORES				
Receptor de RF/MA 5A	Mando universal y multimedia 8x1 RF-IR	Mando RF 16		
Programadores				
Maxicontrolador telefónico con LCD	Mini programador	Programador PC USB + nuevo Software ActiveHome Pro		
Control Remoto SMS				
Control de calefacción y Remoto por SMS de Móvil	DoMobile SMS + apertura de puerta automática sin coste alguno + sensor externo de temperatura			
Controladores IP e Imagen				
Controlador IP 55X: 5 entradas / 5 salidas, con software de Visualización de Cámaras IP	Controlador IP 55S 5 entradas / 5 salidas	Controlador IP X10, con software de Visualización de Cámaras IP		

Figura 2-12: Ejemplos de dispositivos X-10.

A alimentação dos dispositivos X10 está assegurada pela rede elétrica. Os dispositivos contêm normalmente um transformador 110/220 V AC para 12V AC , posteriormente o sinal passa por um retificador de onda completa e com a apoio de um regulador de tensão, mantêm-na a 5V DC para alimentar o dispositivo [24].

Um exemplo da forma como os dispositivos X-10 se dispõem numa instalação e comunicam entre si, mostra-se na Figura 2-13.

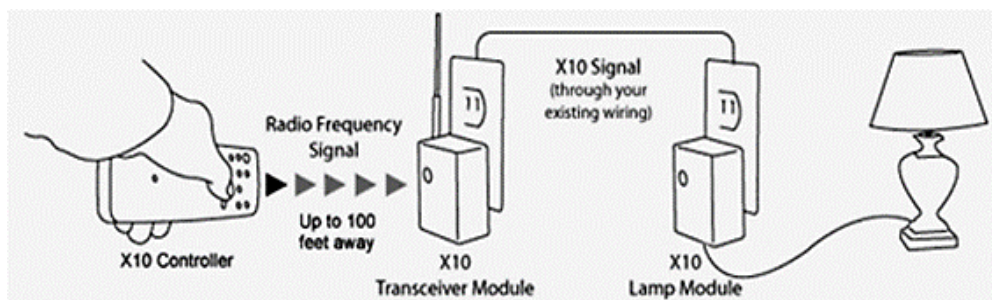


Figura 2-13: Disposição dos dispositivos X-10.

Nos últimos anos têm surgido dispositivos X-10 que permitem a integração deste protocolo com as novas tecnologias, sobretudo associadas à internet.

2.6.2.2 Comunicação

A comunicação baseia-se na “injeção” de sinais de frequências elevadas (120 KHz) sobre a rede de 230V AC, 50 Hz, como se observa na Figura 2-14, representando sinais binários (1 ou 0). O sinal é inserido logo a seguir à passagem pela origem da onda sinusoidal de 50Hz, com um atraso de alguns microssegundos. Deste modo, os recetores têm tempo suficiente para se preparar para a receção dos sinais [25].

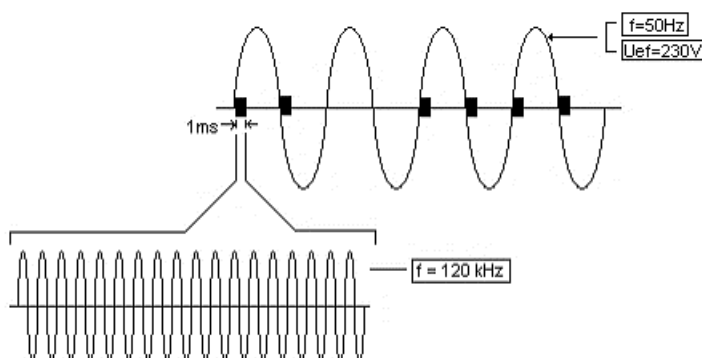


Figura 2-14: Injeção de sinal na rede elétrica.

Em instalações elétricas trifásicas os sinais de 120kHz são emitidos três vezes em cada ciclo, em instantes que coincidem com a passagem por zero da tensão de cada uma das fases, tal como se observa na Figura 2-15. Assim, e com o auxílio de acopladores próprios, torna-se possível comunicar com qualquer dispositivo, independentemente da fase em que esteja instalado.

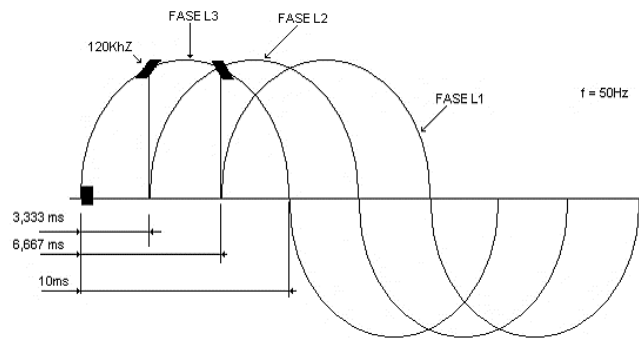


Figura 2-15: Injeção de sinais na rede elétrica.

Cada bit "1" numa transmissão X-10 é um *burst* (sequências curtas) de 120KHz no sinal AC, e cada bit "0" é a ausência desse *burst* (Figura 2-16). A duração de cada *burst* é de 1ms. Cada bit é sempre representado pelo seu valor lógico seguido do seu complemento, evitando assim, que se confundam sinais com ruídos inerentes à rede elétrica.

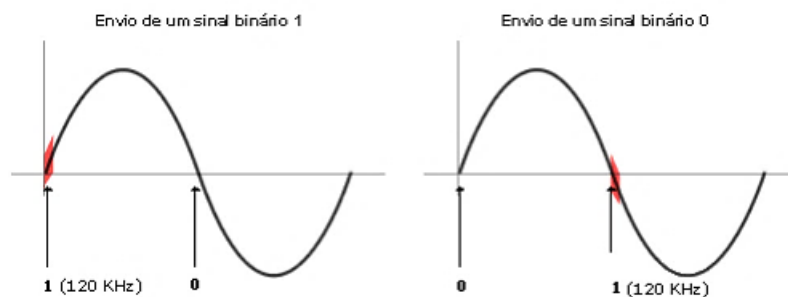


Figura 2-16: Envio do sinal binário 1 e 0 [24].

A transmissão completa engloba a transmissão de quatro campos, que ocupam, onze ciclos da onda elétrica. O primeiro campo, composto pelos dois primeiros ciclos, representam o Código de Início (*Start Code* = 1110), o campo seguinte é composto pelos quatro ciclos seguintes que representam o Código de Casa (letras A-P) e os seus complementos, os quatro bits seguintes, que ocupam quatro ciclos, representam o Código Numérico (1-16), ou o Código de Função (acender luz, desligar luz, variar intensidade). Para distinguir este último campo é enviado um

último bit (e o respetivo complemento) que identifica se o campo anterior se refere ao número de um dispositivo (*bit* = 0) ou ao código de uma função (*bit* = 1).

Para aumentar a fiabilidade do sistema, esta trama (Código de início, Código de casa e Código de Função ou numérico) é sempre transmitida duas vezes, separadas por três ciclos completos da onda sinusoidal da corrente. Existe uma exceção a este procedimento típico, nomeadamente em funções de regulação de intensidade, onde são transmitidos continuamente apenas os códigos numéricos (pelo menos duas vezes) (Figura 2-17) [26].



Figura 2-17: Códigos X-10.

2.6.2.3 Vantagens

Seguidamente referem-se algumas vantagens que explicam a grande utilização deste protocolo, sobretudo no mercado americano.

- Fácil manuseio;
- Acessível;
- Universal;
- Transportável;

- Possibilidade de ser aplicado a qualquer momento, tanto na altura da construção como posteriormente, não necessitando da aquisição de uma nova rede nem da sua alteração;
- Visão de futuro.

2.6.2.4 Desvantagens

As desvantagens do protocolo X-10 que se referem em seguida refletem a “antiguidade” do protocolo.

- Fraca robustez;
- Natureza rudimentar;
- Suscetibilidade a interferências;
- Número limitado de endereços para os dispositivos (máximo 256).

Em suma, o protocolo X-10 caracteriza-se por ser um protocolo de comunicação aberto, normalmente com uma arquitetura distribuída, permitindo que qualquer fabricante desenvolva a comunicação dos seus módulos em X10. Do ponto de vista prático caracteriza-se por ser simples de aplicar e acessível do ponto de vista económico.

2.6.3 Protocolo KNX

O protocolo KNX foi criado em maio de 1999 numa parceria entre a *European Installation Bus Association* (EIBA), *European Home Systems Association* (EHSA) e *Batibus Club International* (BCI). O KNX foi projetado de forma a poder ser instalado em edifícios de grande dimensão, nomeadamente, hospitais, escolas, fabricas, entre outros. Oferece especificações para a automatização de equipamentos de instalação elétrica e soluções para aplicações AVAC. A utilização de sistemas KNX pressupõe a utilização de duas redes distintas, a rede de potência, para a distribuição de energia e alimentação dos dispositivos elétricos e a rede de comando, para a transmissão da informação de comando, como se observa na Figura 2-18. A existência das duas redes torna o sistema mais rentável e flexível, possibilitando, a qualquer altura expandir a instalação (sistema reconfigurável) [26] [27] [28].

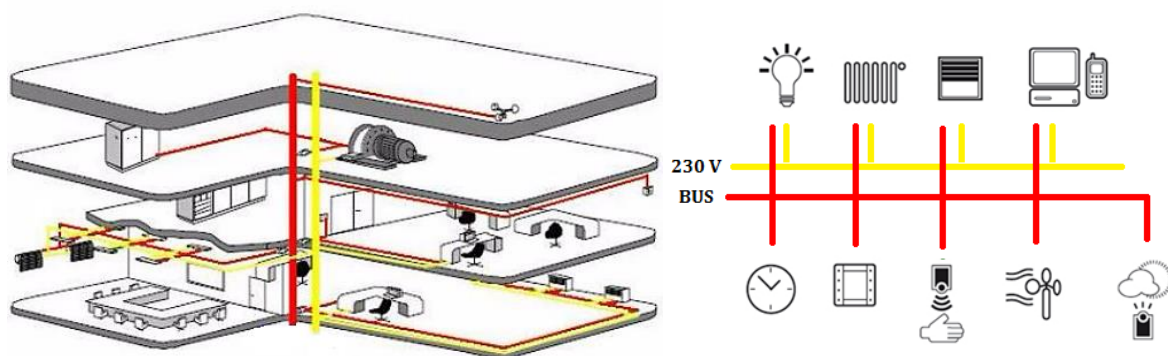


Figura 2-18: Sistema KNX [26].

É um protocolo aberto, normalmente com uma arquitetura distribuída. Os dispositivos comunicam diretamente entre si sem necessidade de um único elemento central, podendo ser recetores e emissores, oferecendo assim maior robustez e flexibilidade ao sistema.

Esta arquitetura distribuída pode abarcar até $65.536(2^{16})$ dispositivos, correspondendo a um espaço de endereço individual de 16bits. A rede está dividida em segmentos onde cada segmento pode conter até um máximo de 64 dispositivos ligados. Quando os segmentos se interligam através de um repetidor, formam, uma linha, que pode conter no máximo quatro segmentos, perfazendo um total de 256 dispositivos. As várias linhas existentes podem ser interligadas numa linha principal através de acopladores de linha, dando assim origem a uma área. Podem existir no máximo 15 áreas, que são conectadas entre si através de uma linha de área. A ligação da linha principal à linha de área é feita através de um acoplador de área [26]. Na Figura 2-19 apresenta-se uma estrutura do sistema KNX.

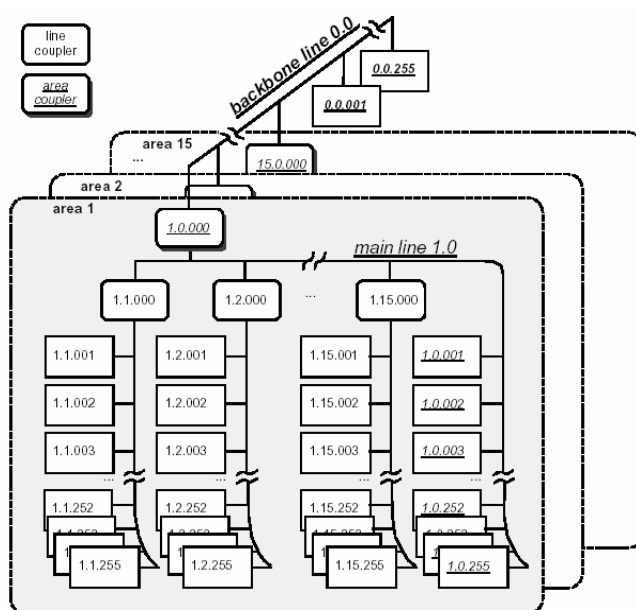


Figura 2-19: Estrutura de um sistema KNX.

No entanto, sempre que necessário, é possível implementar arquiteturas centralizadas para gestão do sistema, por exemplo, ligação à rede de uma aplicação de controlo. A aplicação corre normalmente num computador que pode ser ligado à rede a partir de qualquer localização (Figura 2-20) [23].

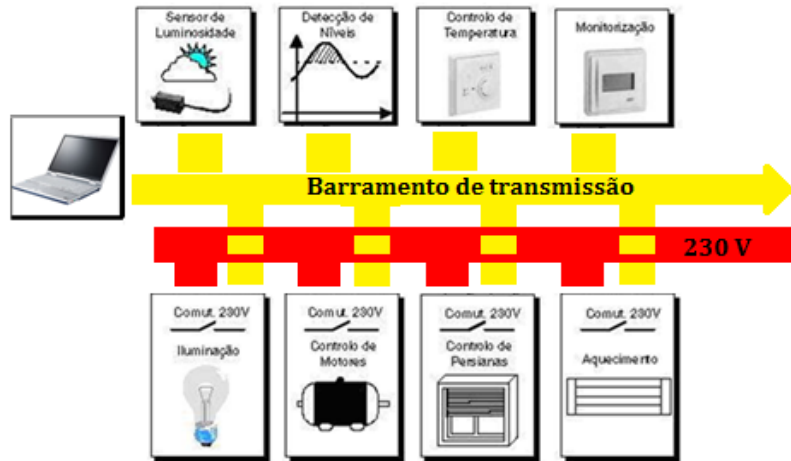


Figura 2-20: Exemplo de arquitetura centralizada.

O KNX não possui uma topologia física definida para sua implementação, podendo existir, topologias lineares ou barramento, em estrela, em anel, em árvore, ou combinações destas, desde que, cumpram os requisitos elétricos (Figura 2-21) [29].

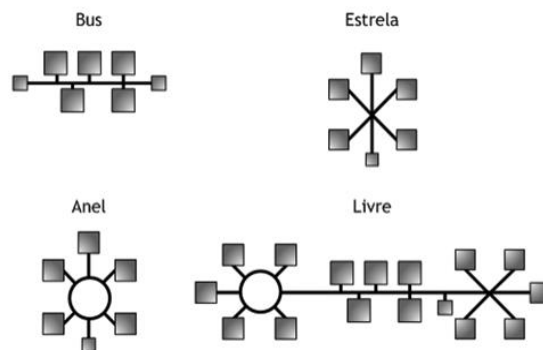


Figura 2-21: Topologias KNX.

O KNX admite diferentes meios físicos de comunicação como cabo coaxial, par de cobre entrançado, fibra ótica, radiofrequência e raios infravermelhos. No entanto, é o par de cobre entrançado o meio mais usual. Juntamente com estes meios, existe ainda a possibilidade de

fazer uma interligação com outro tipo de dispositivos, como é o caso, por exemplo, de um modem GSM, para a transmissão de alertas de situações de emergência para o telemóvel. Esta interligação de dois meios diferentes é possibilitada através de uma unidade conversora, o *gateway*.

2.6.3.1 Dispositivos

Os dispositivos KNX dividem-se em três grandes tipos:

- Componentes básicos - componentes que não têm parte ativa no sistema.
 - fontes de alimentação;
 - filtros de sinal.

- Componentes de sistema – componentes que permitem a criação de uma rede.
 - Acopladores de BUS (BCU – Bus Coupling Unit);
 - Acopladores de Linha (LC – Line Coupler);
 - Acopladores de Fase;
 - Repetidores.

- Componentes orientados para aplicações - componentes que entram diretamente no funcionamento do sistema.
 - Sensores;
 - Atuadores;
 - Painéis de comando.

Na Figura 2-22 apresentam-se alguns exemplos [30].



Figura 2-22: Dispositivos KNX.

2.6.3.2 Configuração dos dispositivos

Os dispositivos KNX foram concebidos para serem configurados de diversas formas com o intuito de satisfazer determinadas funcionalidade e necessidades.

As configurações deveram ser modificadas com o auxílio do Software próprio o ETS (*Engineering Tool Software*) da KNX, embora existam outros no mercado de outras marcas. Estas ferramentas acedem às características e configurações de cada dispositivo e configuram-no conforme a utilidade e funcionalidade pretendida. De seguida descrevem-se resumidamente os vários modos de configuração suportados pelo KNX [7] [19] [28] [31].

- **E-mode (Easy mode)** – Trata-se de uma configuração dirigida essencialmente a instaladores com formação básica em KNX e a instalações de médias dimensões. Os dispositivos já estão pré-programados e carregados com um conjunto de parâmetros predefinidos. No entanto, existe a possibilidade da sua reconfiguração parcial. A configuração é feita com o auxílio de um controlador central, botões ou rodas de código, dispensando a ajuda de um computador com a aplicação ETS.
- **S-mode (System mode)** – É uma configuração com um grau elevado de flexibilidade e dirigida essencialmente a instaladores formados, permitindo realizar funções de domótica mais sofisticadas. O planeamento da instalação e configuração é feito através de um PC com o software ETS instalado, onde os dados dos dispositivos dos diferentes fabricantes estão contidos na base de dados ETS (Figura 2-23). O modo S destina-se a instalações de grandes dimensões.

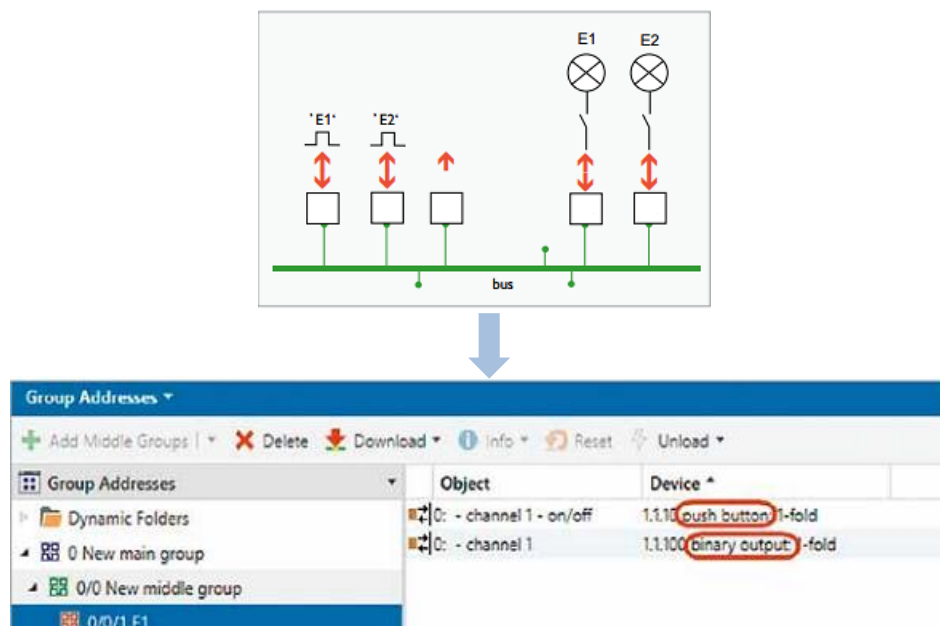


Figura 2-23: Configuração dos dispositivos no software ETS (adaptada de [31]).

- **A-mode (Automatic mode)** - É um modo de configuração mais simples, pois é automático, porém bastante limitado. Segue uma filosofia *Plug&Play*, nem o instalador nem o utilizador final têm de configurar os dispositivos. Este modo de configuração é indicado para os eletrodomésticos e equipamentos de entretenimento (consolas, boxes, áudio e vídeo, entre outros).

Os vários modos de configuração possibilitam que os fabricantes tenham mais liberdade para criar diferentes dispositivos, garantindo a interfuncionalidade entre eles (Figura 2-24).

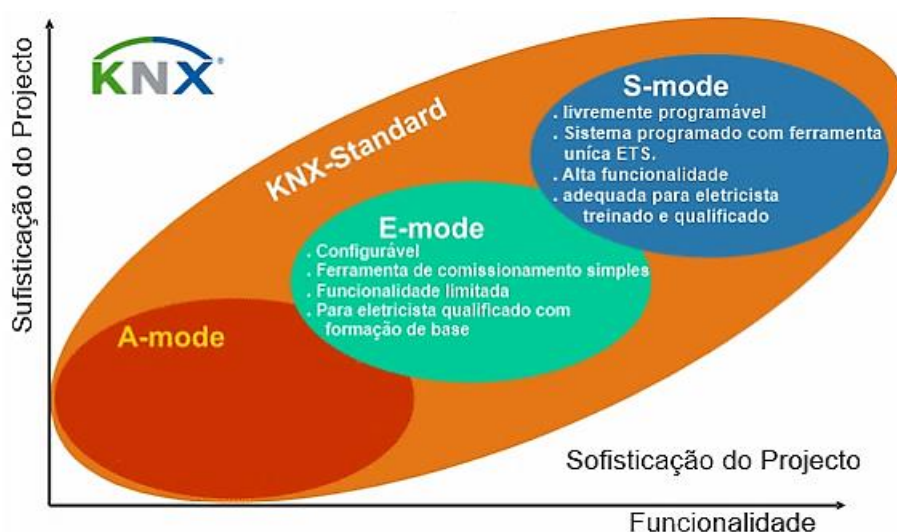


Figura 2-24: Síntese dos modos de configuração [7].

2.6.3.3 Transmissão de sinais

A transmissão de sinais no par de cobre entrançado é realizada por meio da diferença de tensão entre os dois condutores. As mensagens que percorrem o cabo são denominadas de telegramas, e estes são constituídos por sequências binárias. Cada telegrama enviado por um determinado dispositivo será codificado em binário sendo posteriormente convertido em sinais elétricos [28]. O sinal elétrico é transmitido de modo simétrico no *Bus*, como se observa na Figura 2-25. Caso ocorra uma interferência no *Bus*, normalmente irá afetar igualmente os dois condutores do *Bus* (facto assinalado a amarelo na Figura 2-25). Como o sinal é detetado pela diferença de tensões entre os dois condutores, a interferência terá um efeito reduzido.

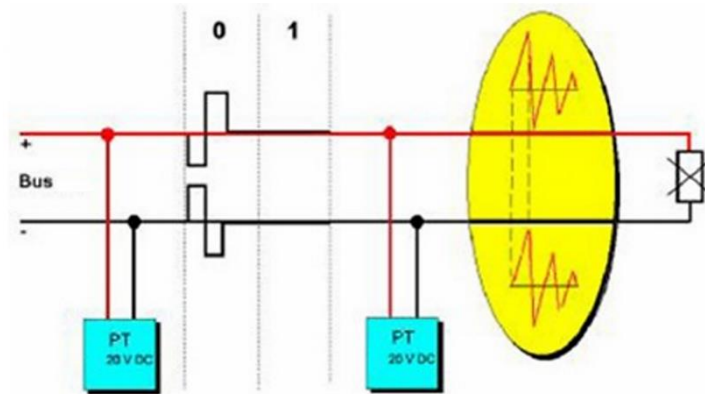


Figura 2-25: Transmissão de sinais no sistema KNX.

Os dispositivos quando pretendem realizar uma transmissão, fazem-no quando o barramento estiver livre, contudo, mesmo durante a transmissão, estes “escutam” o barramento por forma a detetarem qualquer colisão. Quando um dispositivo tenta estabelecer o estado lógico “1” (menos prioritário) e deteta o estado lógico “0”, este pára de transmitir para que outro dispositivo o continue a fazer e, só por fim, é que prossegue com uma nova tentativa. A sequência da transmissão de sinais é apresentada na Figura 2-26.

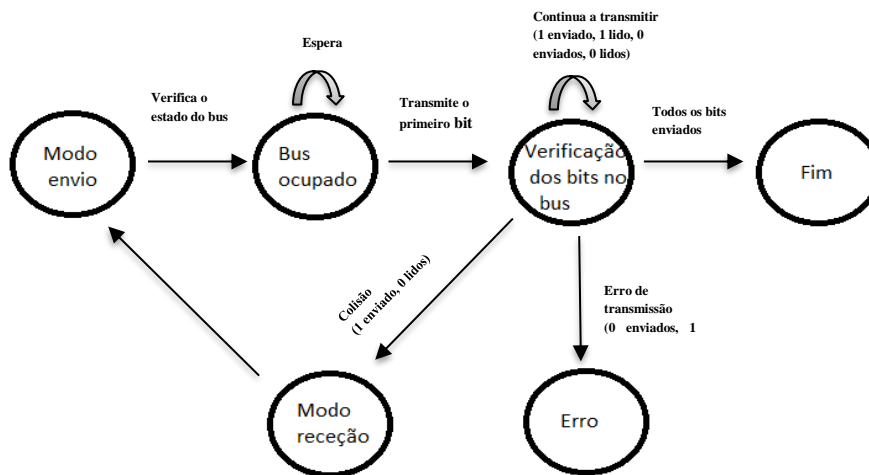


Figura 2-26: Sequência da transmissão de sinais

2.6.3.4 Comunicação

O sistema de comunicação do KNX é constituído por uma pilha protocolar estruturada num conjunto de camadas semelhante ao modelo OSI. No caso particular do protocolo KNX, apenas são implementadas cinco camadas: Física, Ligação de dados, Rede, Transporte e por último, a de Aplicação, como se observa na Tabela 2-1.

Tabela 2-1: Comparação entre o modelo OSI e o sistema KNX.

CAMADA	MODELO OSI	KNX
7	Camada de aplicação	Camada de aplicação
6	Camada de apresentação	
5	Camada de sessão	
4	Camada de transporte	Camada de transporte
3	Camada de rede	Camada de rede
2	Camada de ligação de dados	Camada de ligação de dados
1	Camada física	Camada física

Cada dispositivo inserido no sistema terá uma designação única, denominada de endereço físico. O endereço físico de 16 bit é constituído por um número de zona (4bits), por um número de linha (4bits) e por um número de participante (8bits), que descreve a sua localização e define univocamente cada dispositivo na rede (Figura 2-27). Este tipo de endereçamento é apenas usado como endereço para a inicialização, programação e para as operações de diagnóstico. O endereço físico poderá ser, por exemplo, 3.11.52. Isto significa que será o 52º participante instalado, na linha 11 da zona ou área 3.

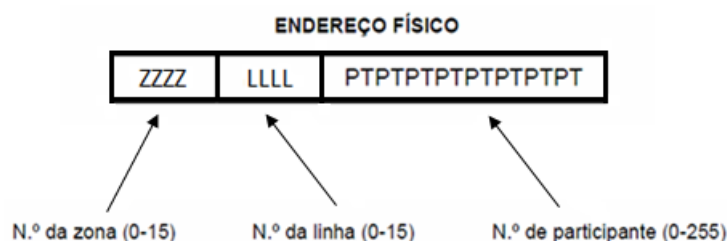


Figura 2-27: Endereço físico [28].

Existe um outro tipo de endereço para além do endereço físico conhecido como endereço de grupo. O endereço de grupo pode ser de dois níveis (grupo principal e grupo secundário) ou três níveis (grupo principal, grupo médio e grupo secundário), consoante as necessidades na hierarquização das funções do sistema (Figura 2-28). Estes endereços estipulam quais os dispositivos ligados ao barramento que vão atuar em sintonia entre si. Pode-se afirmar que as ligações físicas das instalações convencionais (cabladas) são realizadas no KNX através dos endereços de grupo. Por exemplo, quando se pretende “ligar” dois dispositivos dá-se-lhes o mesmo endereço de grupo. O endereço de grupo tem uma sintaxe diferente do endereço físico (exemplo: 3/4/5).

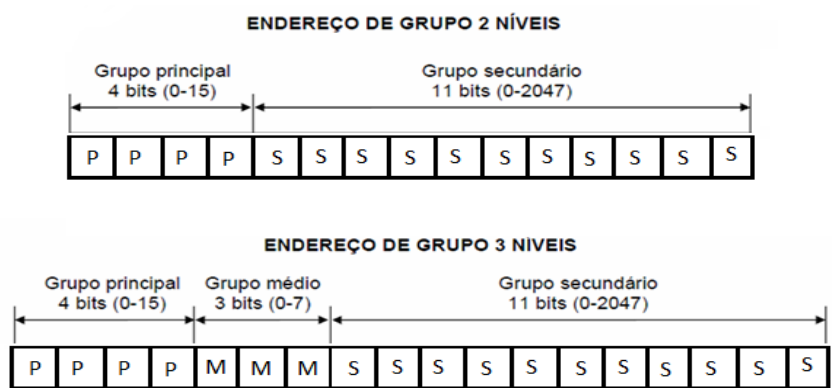


Figura 2-28: Endereço de grupo de 2 e 3 níveis [28].

Assim, quando um dispositivo, por exemplo um interruptor com um determinado endereço físico pretende comunicar com um atuador, fá-lo através do correspondente endereço de grupo (Figura 2-29).

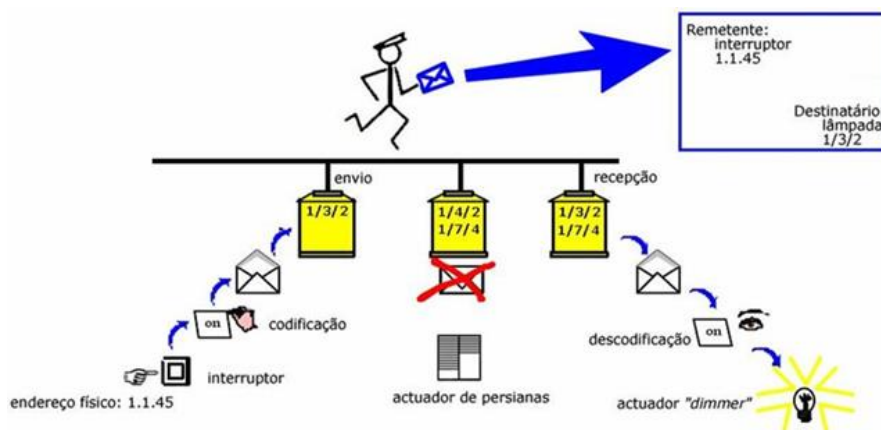


Figura 2-29: Comunicação [28].

2.6.3.5 Vantagens

Grande parte das vantagens do KNX advém do facto de ser um protocolo relativamente recente e, por isso, contempla grande parte dos desenvolvimentos entretanto conseguidos. Referem-se em seguida algumas das vantagens mais relevantes.

- Interoperabilidade e conjugação de produtos;
- Qualidade do produto;
- Versatilidade;
- Suporta vários tipos de modos de configuração;

- Suporta vários meios físicos de comunicação;
- Possibilidade de expansão, mudanças e atualizações;
- Ferramenta de configuração independente do fabricante - A ferramenta de software ETS, ferramenta de configuração independente do fabricante, permite o planeamento, engenharia e a configuração de todos os produtos KNX certificados.
- Poupança energética;
- Visão de futuro;
- Pode ser acoplado a outros sistemas.

2.6.3.6 Desvantagens

Apesar das diversas vantagens referidas, existem algumas desvantagens que advêm da grande potencialidade do sistema e da experiência prática entretanto obtida.

- Complexidade das instalações;
- Custo também relativamente elevado do equipamento certificado;
- Segurança;
- Fonte de alimentação dos segmentos representa um ponto único de falha.

A prática levou à constatação de algumas limitações dos sistemas, nomeadamente em termos de segurança, onde são conhecidos alguns casos de falta de segurança. Pense-se, por exemplo, no caso de um hotel que usa o sistema KNX. Através do Bus KNX existente num quarto de hotel é possível aceder a toda a rede KNX. Por outro lado a fonte de alimentação dos segmentos constitui um ponto único de falha, que se traduz num problema de fiabilidade do sistema.

Em suma, o protocolo KNX é um protocolo muito flexível e com uma ampla gama de benefícios tanto para os instaladores como para os utilizadores, especialmente em tempos de redes inteligentes (*Smart Grids*), onde a gestão habitacional e o uso otimizado de energia renovável são um fator determinante.

2.6.4 Protocolo LonWorks

A empresa *Echelon Corporation* (EUA) apresentou, em 1990, a tecnologia *LonWorks* e o conceito de rede LON (*Local Operating Network*). O grande objetivo da empresa com a apresentação da rede LON, foi a conceção de um microprocessador com uma interface de comunicação *standard*, onde cada aparelho possuísse a capacidade de “comunicar e trabalhar” com qualquer outro, independentemente do fabricante sem perder a inteligência e eficiência dentro de uma rede. Estima-se que aproximadamente 4000 fabricantes mundiais desenvolveram

aparelhos e sistemas para tecnologia *LonWorks™* (*LonWorks™* é a descrição do sistema para toda a tecnologia LON).

Embora tenha sido projetada para atender também aos requisitos da maioria das aplicações domóticas, a sua implementação foi bem sucedida em edifícios de escritórios, hotéis e indústrias (Figura 2-30) [20] [27] [32] [33].

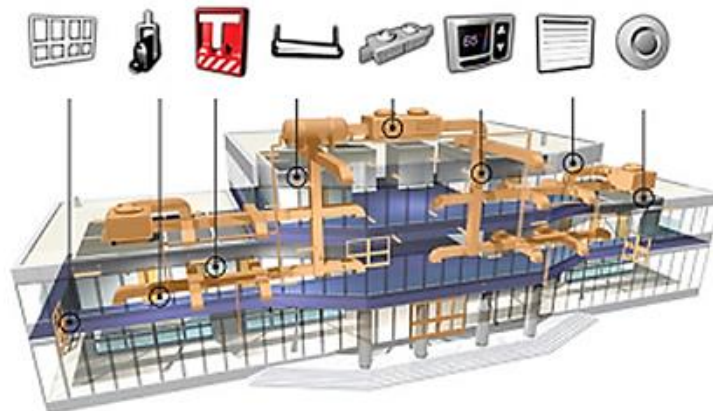


Figura 2-30: Sistema *LonWorks* [33].

A rede LON não possui uma topologia física definida para sua implementação, podendo existir, topologias lineares ou barramento, em estrela, em anel, em árvore, ou combinações destas, desde que, cumpram os requisitos elétricos.

Admite também diferentes meios físicos de transmissão de dados, cabo coaxial, par de cobre entrançado, fibra ótica, radiofrequência, infravermelho, rede elétrica. No entanto, é o par de cobre entrançado o meio mais usual. Além desta facilidade é possível ligar uma rede LON à Internet.

2.6.4.1 Constituição de uma rede LON

Uma rede LON é constituída por diversos dispositivos de controlo inteligente.

- Nós ou nodos;
- *Neuron Chips*;
- Variáveis de rede;
- Ferramentas de desenvolvimento, monitorização e teste.

Os Nós ou Nodos (sensores inteligentes de temperatura, de pressão, entre outros, ou atuadores ou interfaces para o operador) possuem uma interface física (Transdutor) que permite a sua ligação ao meio físico de comunicação (Figura 2-31).

Os Nós ou Nodos servem-se de um protocolo comum, o *LonTalk*, que implementa as sete camadas do modelo OSI e introduz mecanismos de proteção que impedem as alterações intencionais ou imprevistas. Além disso, este protocolo tem a capacidade de autenticação do remetente, funções de reconhecimento, prioridade na transmissão, deteção de mensagens repetidas e correção de erros.

A inteligência dos Nós é fornecida por microcontroladores especiais chamados *Neuron Chips*. Estes microcontroladores permitem a implementação do protocolo *LonTalk* e a execução de todas as funções de controlo [32] [23].

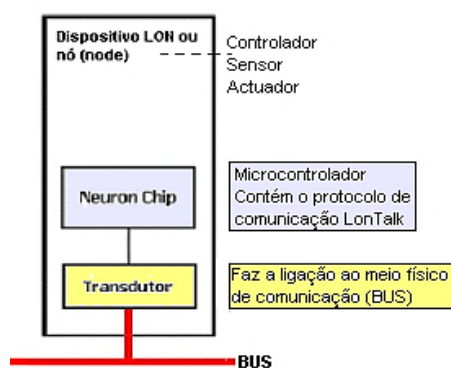


Figura 2-31: Dispositivo LON [23].

Os *Neuron Chips* (Figura 2-32). possuem um *Neuron ID*, identificador único, que permite direccionar qualquer Nó de forma unívoca dentro de uma rede e possuem também um modelo de comunicação que é independente do meio físico sobre o qual funcionam.

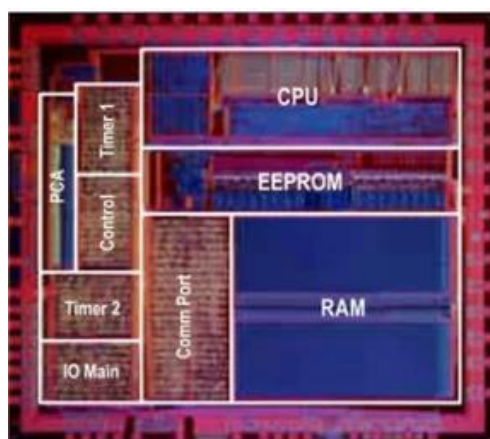


Figura 2-32: *Neuron Chips* [23].

2.6.4.2 Estrutura de uma rede LON

Uma rede LON é organizada em Domínio, Subrede e Nodes ou Nós. O Domínio representa uma área onde estão localizadas um máximo de 255 Subredes, uma Subrede poderá abranger um máximo de 127 Nós e um Domínio um máximo de 32385 Nós. Ainda existe a possibilidade de ligar diferentes Domínios entre si (máximo de 248).

Apenas os Nós inseridos no mesmo Domínio podem comunicar diretamente entre si e possuem um endereço único na rede e o seu endereço segue a seguinte hierarquia:

Domínio – Subrede – Nó

Na construção e ampliação de uma rede LON são utilizados routers (têm o objetivo de ligar duas Subrede entre si), pontes (ligação entre dois domínios) e repetidores (amplificadores físicos), como se observa na Figura 2-33 [32].

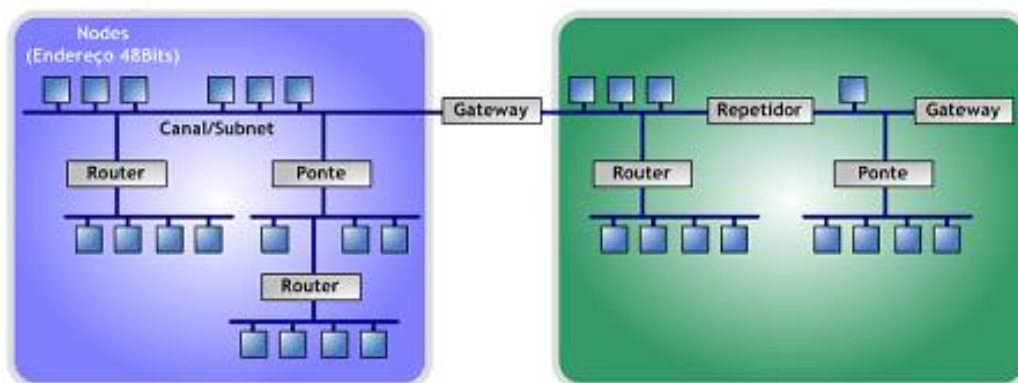


Figura 2-33: Constituintes de uma rede LON.

2.6.4.3 Processamento da informação

Uma rede LON é constituída por diversos Nós, que se encontram ligados fisicamente pelos respetivos transmissores ao *Bus* de comunicação. Trocam informações entre si, enviando telegramas que contêm a direção do destinatário, informação para o *routing*, dados de controlo assim como os dados da aplicação do utilizador e um *checklist* como código detetor de erros (Figura 2-34).

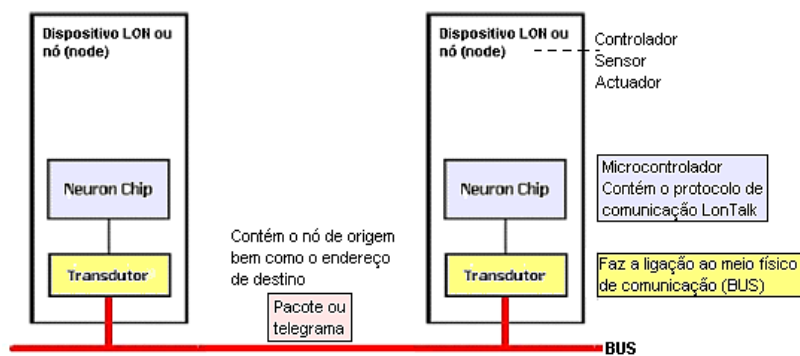


Figura 2-34: Ligação entre nós [23].

Existem casos em que se torna necessário criar ligações lógicas entre Nós. Um exemplo disso é utilizar um Nó como interruptor da iluminação e transferir informações para atuar a iluminação de uma determinada área, como se observa na Figura 2-35.

A informação a transferir pelo Nó interruptor é enviada via variável de rede, SNVT (*Standard Network Variable Type*), criando assim uma interface lógica entre Nós.

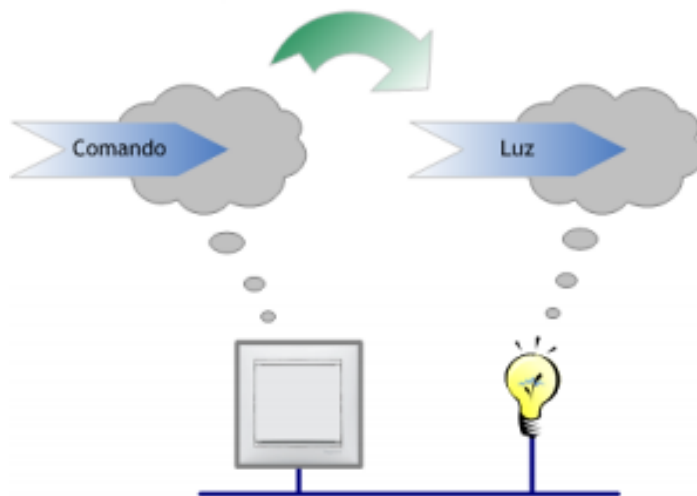


Figura 2-35: Ligação entre variáveis na rede [32].

As variáveis SNVT encontram-se estabelecidas pela entidade LONMARK (compatibilidade do funcionamento dos equipamentos) e estão todas referenciadas numa *Master List* que está disponível a qualquer criador de equipamentos LON (Figura 2-36) [34].

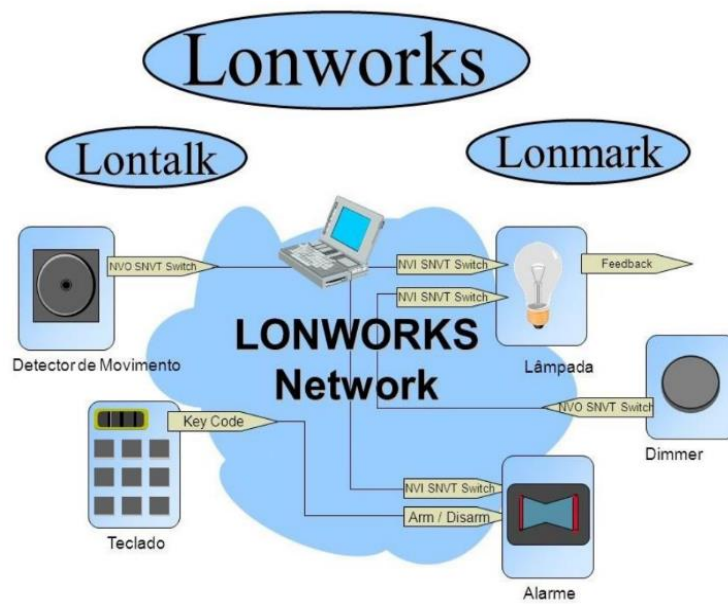


Figura 2-36: Troca de informação num sistema *LonWorks* [34].

Deste modo para os dispositivos comunicarem entre si, é estabelecida via software uma ligação virtual entre Nós, podendo esta ser alterada no dispositivo e também facilmente apagada, modificada ou adicionada qualquer tipo de ligação.

Na Figura 2-37 pode-se observar que no caso de aplicações em que estejam em causa a troca de um grande número de variáveis entre dispositivos, um sistema baseado no *LonWoks* pode ser vantajoso.

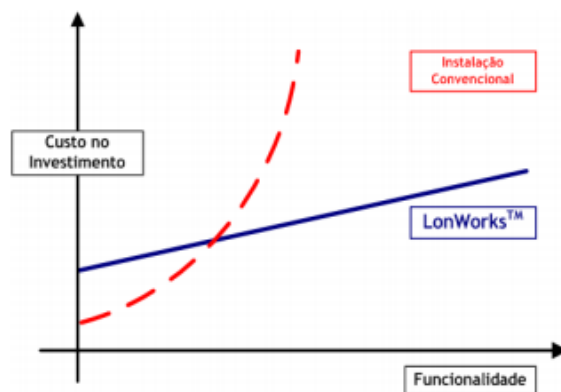


Figura 2-37: Comparação entre um sistema convencional e o sistema *LonWorks* [32].

2.6.4.4 Vantagens

Seguidamente referem-se algumas vantagens referentes ao protocolo *LonWoks*.

- Robustez;
- Fiabilidade;
- Ideal para ambientes industriais;
- Grande variedade de meios de transmissão de dados;
- Viasão de futuro.

2.6.4.5 Desvantagens

As desvantagens do protocolo *LonWorks* que se enunciam advêm da instalação e utilização do mesmo.

- Problemas com comunicação wireless;
- Pouco estandardizado;
- Elevado custo inicial.

2.6.5 Tecnologia baseada em PLC

Segundo a NEMA, um controlador lógico programável é um aparelho eletrónico digital que utiliza uma memória programável para o armazenamento interno de instruções para implementações específicas, tais como lógica, seqüências, temporização, contagem e aritmética, para controlar, através de módulos de entradas e saídas, vários tipos de máquinas e/ou processos (Figura 2-38) [35].

Os PLC surgiram na década de 60, com o objetivo de reduzir o enorme custo que a substituição e manutenção de sistemas de controlo baseados em painéis de relés na indústria automóvel exigia. Primeiramente foram concebidos para trabalhar apenas com variáveis digitais (0 e 1), posteriormente e após diversos desenvolvimentos e testes conseguiu-se o processamento de variáveis analógicas.

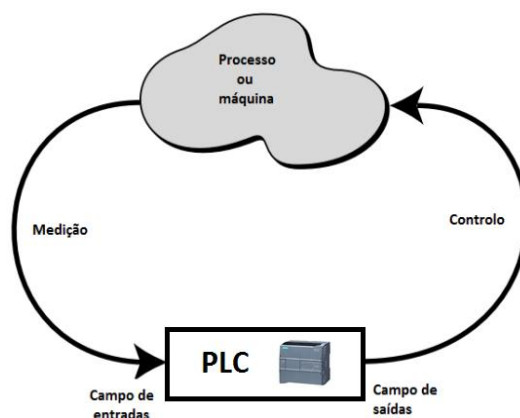


Figura 2-38: Ciclo de funcionamento de um PLC.

Assim, tornou-se ao longo das últimas décadas num equipamento com aplicações generalizadas na indústria.

Uma outra abordagem possível, mas menos frequente é a utilização da tecnologia baseada em PLC para implementar a totalidade ou parte das funcionalidades domóticas. Assim a habitação moderna passa a ser vista como um sistema de automação (“home automation”).

As “home automation” fazem parte do presente e é preciso tornar a domótica um serviço simples e agradável de se utilizar.

2.6.5.1 Vantagens

A utilização dos PLC na implementação da domótica apresenta diversas vantagens, das quais se destacam:

- Programas únicos e específicos;
- Alteração da programação e dos componentes de acordo com o arbítrio do utilizador ou necessidade da habitação;
- Controladores expansíveis, escalonáveis e robustos;
- Grande variedade de componentes no mercado, mesmo em segunda mão, para se poderem interligar com os controladores;
- Manutenção fácil e rápida;
- Alta flexibilidade e soluções inteligentes através de *hardware* escalonável;
- Interface de comunicação com outros controladores, HMI, aparelhos eletrónicos e computadores de controlo;

2.6.5.2 Desvantagens

No entanto, para aplicações domóticas os PLC por terem sido concebidos sobretudo para a indústria, apresentam algumas desvantagens, tais como:

- Sistemas maioritariamente centralizados, caso falhe o controlador todo o sistema é posto em causa;
- Programação e montagem um pouco trabalhosa, uma vez que normalmente não existe programação pré-definida para domótica.

2.7 Comparação entre protocolos

O X-10 é o protocolo mais utilizado no mundo, contudo, não preenche todos os requisitos e possui algumas limitações funcionais quando comparado com o KNX e o *LonWorks*.

O KNX apresenta uma elevada capacidade e fiabilidade podendo ser instalado tanto em pequenos como em grandes edifícios sem necessidades adicionais, embora acarrete custos elevados e complexidade na instalação.

O *LonWorks* é o protocolo que mais se assemelha ao KNX, sendo-lhe habitualmente comparado, contudo está claramente adaptado à automação de grandes edifícios.

Por forma a subscrever tudo o que foi citado anteriormente realizou-se uma pesquisa acerca da opinião das pessoas sobre quais os protocolos que conhecem, os que mais utilizam e a sua preferência. A pesquisa recaiu num relatório/inquérito realizado numa feira anual em 2008 por uma Universidade de Ciências Aplicadas do Sul da Alemanha, Westphalia.

Pela Figura 2-39 observa-se claramente que 69% de todos os entrevistados disseram que usavam o protocolo KNX com mais frequência, por ser na sua opinião o que preenche todos os requisitos [36]. Embora o protocolo X-10 não se encontre enunciado na figura, facilmente se depreende que jamais apresentaria resultados superiores ao KNX, não só por ser mais antigo, mas por ter limitações no que respeita à associação de novas tecnologias. Por outro lado, o protocolo X-10 tem uma expressão diminuta na Europa.

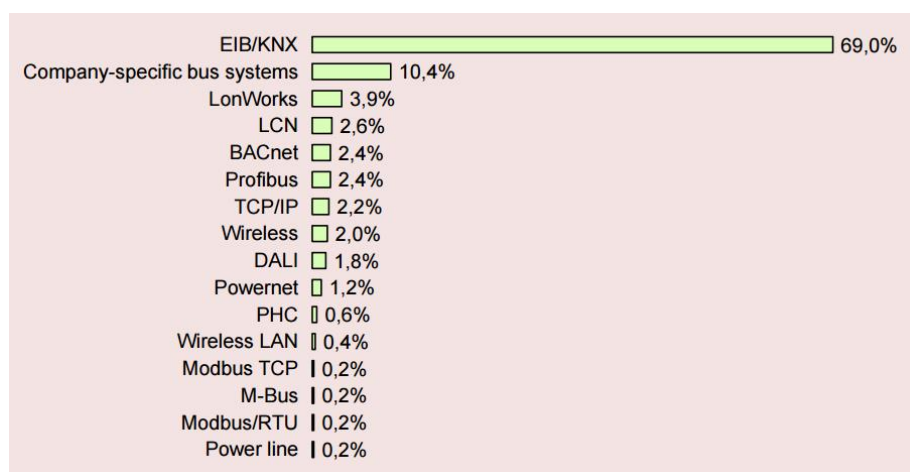


Figura 2-39: Resultados do inquérito realizado acerca dos protocolos de comunicação.

Relativamente aos sistemas que utilizam o PLC na implementação da maioria das funcionalidades estes caracterizam-se pela sua flexibilidade e baixo custo comparativamente aos protocolos de comunicação. Apresentam-se como uma solução acessível a um maior número de pessoas.

3. Desenvolvimento e comando do sistema domótico

Neste capítulo apresenta-se o desenvolvimento de grande parte do projeto. Após reunida toda a informação foram implementadas inúmeras funcionalidades em um PLC, para posteriormente poderem ser geridas à distância (web, wifi, gsm), recorrendo a tecnologias como página web, sms, app e e-mail.

Em termos de *hardware* o equipamento base para implementação das funcionalidades domóticas foi o PLC S7-1200, acompanhado de um módulo de entradas e saídas digitais, de um módulo com uma saída analógica, de um módulo GSM CP 1242-7 e dos mais diversos componentes e dispositivos elétricos e eletrônicos.

Em termos de software na elaboração do projeto utilizou-se o *software* da Siemens STEP 7 Professional (Tia Portal), o *Android Studio* e um editor HTML.

3.1 PLC S7-1200

O PLC S7-1200 é um controlador modular versátil, seguro e que se enquadra perfeitamente numa ampla variedade de aplicações, através de módulos de sinal conectáveis e módulos de comunicação. Apresenta um design escalonável e flexível para soluções compactas e inteligentes, interface PROFINET integrado para programação, diagnóstico, ligação ao HMI e comunicação CPU-a-CPU.

A gama S7-1200 incorpora 4 CPU diferentes e cada CPU está disponível com uma escolha de tensão de alimentação AC ou DC, e com saídas a relé ou estado sólido.

No desenvolvimento do projeto optou-se pelo PLC 1214C AC/DC/RLY (tensão de alimentação 230V AC, entradas DC e saídas a relé com a referência do fabricante 6ES7 214-1BE30-0XB0) por ser o mais utilizado em ambiente académico (Figura 3-1).[37]



Figura 3-1: PLC S7-1200 1214C AC/DC/RLY.

3.1.1 Aspetos do *Hardware* de um PLC

Geralmente constituído por uma unidade central ou de processamento, por uma fonte de alimentação, um dispositivo de programação, uma memória, uma secção de interface entradas/saídas e uma interface de comunicação, como se observa na Figura 3-2 [38].

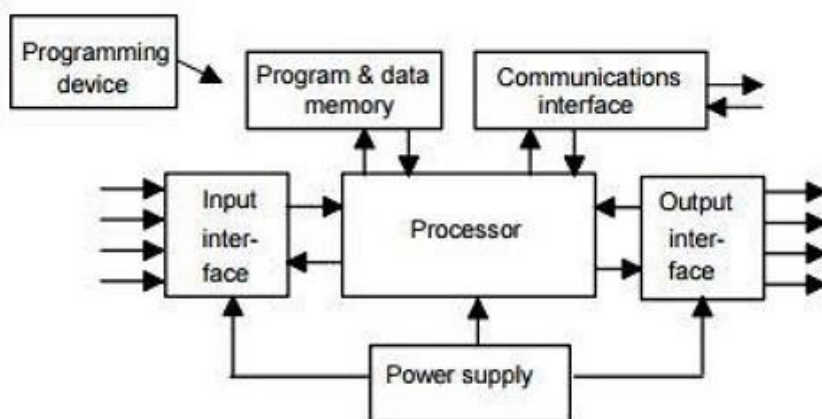


Figura 3-2: Hardware de um PLC.

- A unidade central ou de processamento (CPU) é a unidade que contém o microprocessador que é responsável por interpretar todos os sinais de entrada e dar início às ações de controlo, de acordo com o programa armazenado em memória, atribuindo os sinais de ação para as saídas.
- A fonte de alimentação fornece energia aos elementos eletrónicos internos do controlador, converte a tensão de entrada numa tensão adequada à alimentação do PLC e protege os componentes do mesmo contra eventuais picos de tensão.
- A memória é dos elementos mais importantes de um PLC quando se pretende alterar o programa de controlo de forma rápida e simples. O sistema de memória de um PLC é composto por três áreas de memória onde os dados do programa do utilizador e a configuração são armazenados. Pode-se falar de três tipos de memória: a memória de programa, memória de trabalho e a memória retentiva. A memória de programa é diferente, dependendo do CPU escolhido e permite armazenar de forma não volátil o programa do utilizador, os dados e a configuração. O programa do utilizador é carregado pela primeira vez para esta área. Por outro lado, esta memória pode ser substituída por um cartão de memória (pré-configurado para esta finalidade). A memória de trabalho também é diferente dependendo do CPU escolhido e oferece um armazenamento volátil, sempre que a energia é desligada esta área é perdida. A memória retentiva permite armazenar dados de forma não-volátil da memória de trabalho e o número de dados é limitado. Quando ocorre uma falha de energia, o CPU irá restaurar os valores ali guardados para recomeçar (Figura 3-3).

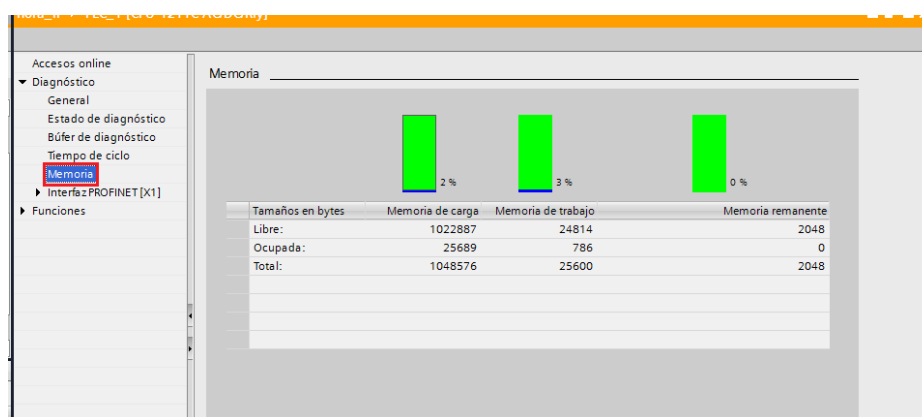


Figura 3-3: Tipos de memória num PLC.

- Nas interfaces de entrada e saída o processador recebe e posteriormente escreve a informação proveniente dos dispositivos externos. Os sensores e interruptores são exemplos de entradas, os motores e as válvulas, são exemplos de saídas. Os dispositivos

de entrada e saída podem ser classificados como sinais digitais (0/1) ou analógicos (tensão ou corrente proporcional a uma variável).

- A interface de comunicação recebe e transmite a informação em redes de dados.

3.1.2 Aspectos do *Software* de um PLC

São inúmeras as funções que estes equipamentos podem suportar, de entre uma ampla gama podem-se citar, a lógica AND, OR, XOR, o SET e RESET, os temporizadores e contadores, os comparadores, os cálculos com aritmética básica, os conversores, registos e saltos no programa. Aquando do aumento da capacidade de processamento, surge a necessidade de funções mais avançadas, nomeadamente, cálculos trigonométricos e integrais, controlo PID, leitura de sinais analógicos, linearização de sinais analógicos, contagem rápida, medição, lógica difusa, funções, entre outros.

3.1.3 Ciclo de processamento

No momento em que o PLC é ligado executa uma série de operações pré-programadas (inicialização) em que:

- Verifica o funcionamento eletrónico do C.P.U., memórias e circuitos auxiliares;
- Verifica a configuração interna e compara com os circuitos instalados;
- Verifica os estados principais (RUN / STOP, PROG, etc.);
- Desativa todas as saídas;
- Verifica a existência de um programa de utilizador;
- Emite um aviso de erro, caso algum dos itens acima descritos falhe.

A função de um PLC passa por executar um programa, que por sua vez executa o processamento de sinais de entrada provenientes de diferentes tipos de sensores e outros dispositivos de campo, e fornece sinais de saída compatíveis com o acionamento de diversos componentes como atuadores, motores, válvulas, etc. A Figura 3-4 ilustra um esquema do ciclo de funcionamento

do PLC, resumindo a sua função em quatro etapas em que três delas são executadas ciclicamente [38].

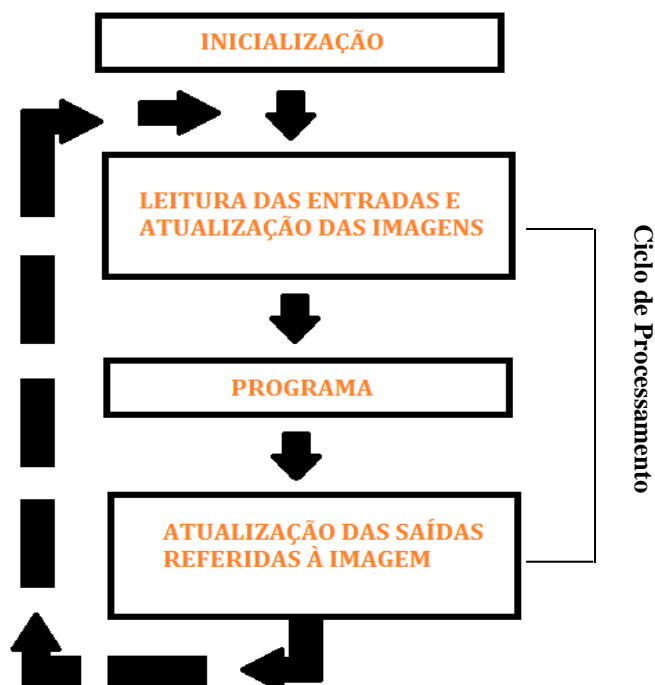


Figura 3-4: Ciclo de processamento de um PLC

Leitura de entradas – Lê os valores físicos dos módulos de entradas e copia-os para a memória imagem;

Execução do programa – Baseando-se nos valores na memória imagem das entradas, executa o programa atualizando a memória imagem com os valores das saídas;

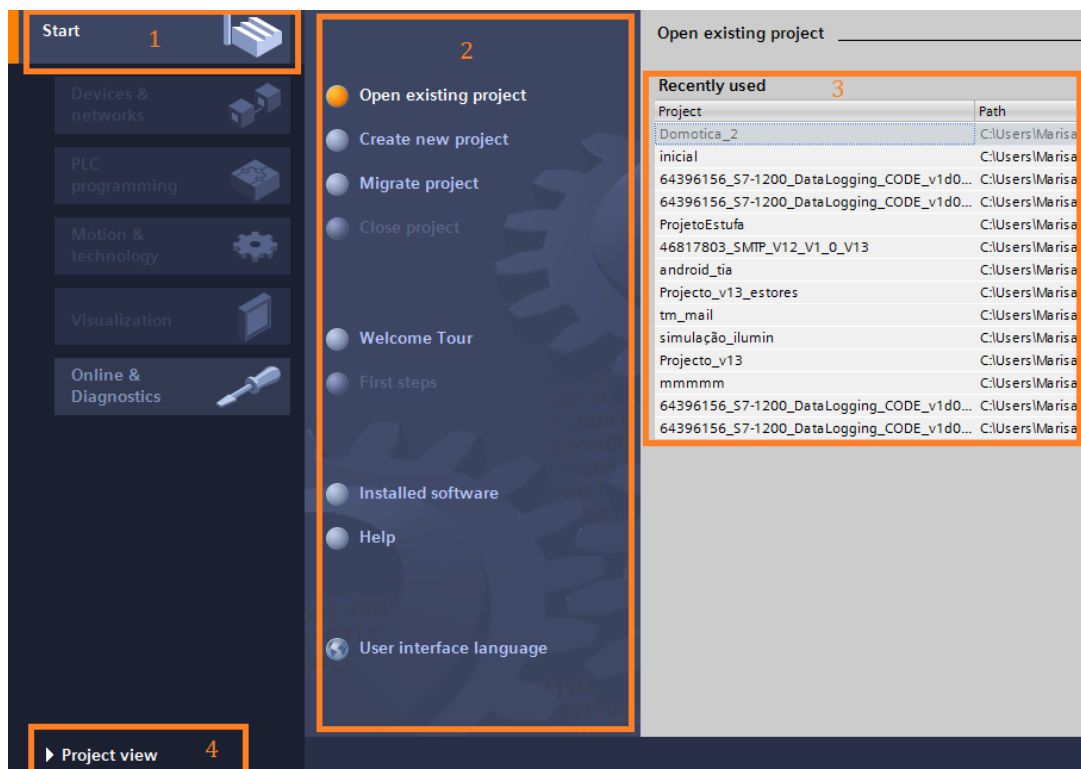
Atualização das saídas – Copia os valores das saídas em memória imagem para os módulos de saídas.

É de salientar que na memória de trabalho (memória volátil) existe uma zona conhecida por memória imagem das entradas e das saídas, que reflete o estado das entradas e saídas físicas e o programa do utilizador processa a informação contida nesta zona do programa.

3.2 Aplicação de desenvolvimento Tia Portal

O STEP 7 Professional (TIA Portal) foi um dos *softwares* utilizados na elaboração do projeto. Esta ferramenta de desenvolvimento caracteriza-se por ser um *software* para configuração, programação, teste e diagnóstico dos diversos controladores do fabricante Siemens.

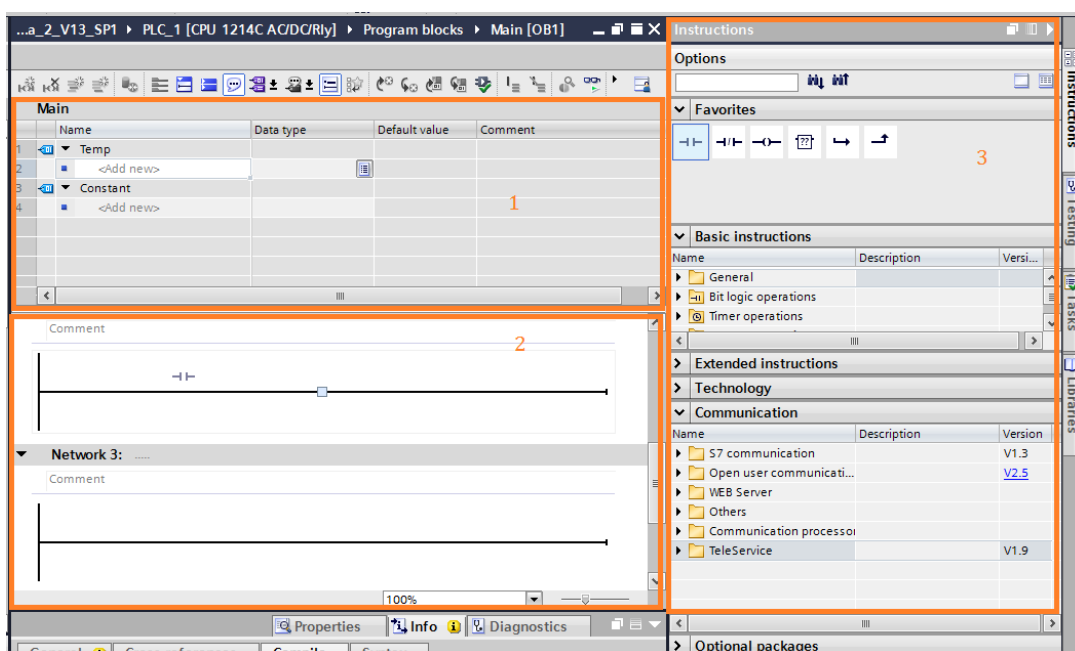
Dotado de uma variedade de funções fáceis de utilizar, o *software* STEP 7 garante assim economia de custo significativa no desenvolvimento de tarefas de automação. Para além de permitir a criação de projetos, também permite converter projetos criados em programas de versões anteriores e efetuar migrações ao mesmo tempo, permite a total configuração, programação, controlo e monitorização de vários PLC quando ligados em rede (Figura 3-5) [39]. A aplicação permite ainda programar uma variedade alargada de consolas de interface Homem-máquina (HMI).



1- Portal - Start; 2-Tarefas para o portal selecionado; 3- Pannel de seleção; 4-Visualizar o projeto.

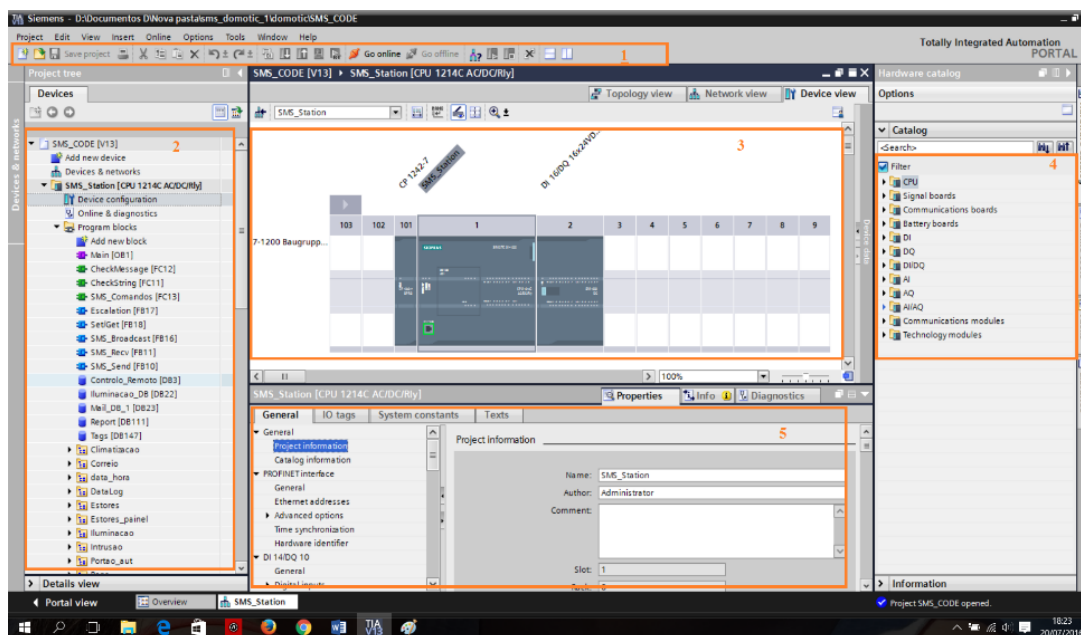
Figura 3-5: Estrutura do TIA Portal.

Apresenta um ambiente gráfico bastante intuitivo e funções para os mais variados fins. Todas as funções existentes são auxiliadas por uma ajuda detalhada (Figura 3-6 e Figura 3-7).



1-Nome e tipo de variáveis a utilizar; 2- Área de trabalho; 3-Instruções.

Figura 3-6: Ambiente de programação do TIA Portal.



1-Menus e barras de ferramentas; 2-Navegador do projeto; 3-Device configuration; 4-Hardware; 5-Propriedades.

Figura 3-7: Constituintes do TIA Portal.

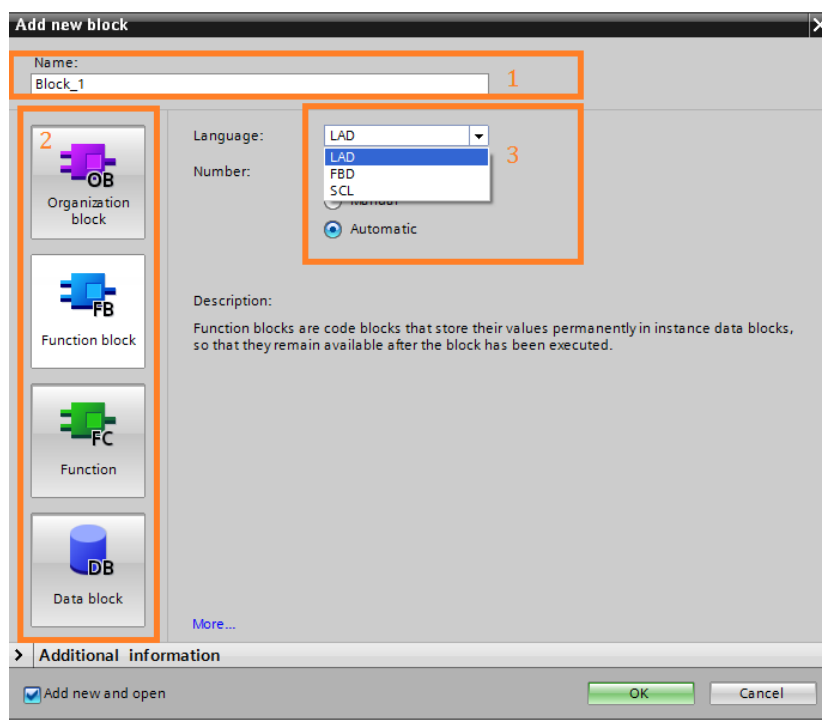
O TIA Portal possui a vantagem de permitir a programação e visualização de programas em 3 tipos distintos de linguagem [39]:

- LAD (*Ladder*): Linguagem gráfica baseada em símbolos. É a mais utilizada, estando presente em praticamente todos os PLC.
- FBD (Diagrama de Blocos Funcionais): Linguagem gráfica muito utilizada na representação de sistemas de controlo industriais, adotando um conjunto de símbolos e convenções.
- SCL (*Structured Control Language*): Linguagem de alto nível (Pascal).

Adicionalmente para alguns PLC mais evoluídos permite a programação em *Grafcet*.

Possui também uma programação estruturada, ou seja, o programa é subdividido em blocos, geridos e ordenados conforme as funções e posteriormente chamados no bloco principal, *Main*. A Figura 3-8 ilustra os tipos de blocos existentes e que são:

- OB (bloco de organização): Um OB é chamado ciclicamente pelo sistema operativo formando uma interface entre o programa do utilizador e o sistema operativo. Neste OB, a unidade de controlo do PLC é informada sobre os comandos que deverão ser processados pelos blocos do programa.
- FC (função): Uma FC não possui nenhum espaço de memória atribuído. Os dados locais de uma função perdem-se após o processamento da função. Numa função poderão ser chamados outros FB e FC.
- FB (bloco de função): O FB necessita de um espaço de memória atribuído para cada chamada (instância). Quando um FB é chamado, por exemplo, um bloco de dados (DB) pode ser atribuído como DB de instância. Os dados neste DB de instância são, então, acedidos através das variáveis do FB. Quando este for chamado várias vezes, diferentes espaços de memória deverão ser atribuídos a um FB. Num bloco de função poderão ser chamados outros FB e FC. Pode-se dizer que um FB é uma função com dados próprios.
- DB (bloco de dados): Os DB são usados para disponibilizar espaço de memória para as variáveis de dados. Existem dois tipos de blocos de dados. DB globais, onde todos os OB, FB e FC podem ler os dados armazenados ou gravar dados no DB e DB de instância, que são atribuídos a um determinado FB. Pode-se dizer que os DB são estruturas de dados.



1-Nome do bloco; 2-Blocos; 3-Tipos de linguagem e número do bloco.

Figura 3-8: Blocos do Tia Portal.

Para além do referido anteriormente, o TIA Portal tem a capacidade de trabalhar com uma grande variedade de tipos de dados, por meio dos quais é possível representar diferentes formatos de números. Na Tabela 3-1 apresenta-se uma lista com os vários tipos de dados [37].

Tabela 3-1: Tipos de dados do TIA Portal

Tipo de dados	Tamanho (bit)	Gama	Exemplo
Bool	1	0 a 1	TRUE, FALSE, 0, 1
Byte	8	16#00 a 16#FF	16#12, 16#AB
Word	16	16#0000 a 16#FFFF	16#ABCD, 16#0001
DWord	32	16#00000000 a 16#FFFFFFFF	16#02468ACE
Char	8	16#00 a 16#FF	'A', 'r', '@'
Sint	8	-128 a 127	123, -123
Int	16	-32.768 a 32.767	123, -123
Dint	32	-2.147.483.648 a 2.147.483.647	123, -123
USint	8	0 a 255	123
Uint	16	0 a 65.535	123
UDint	32	0 a 4.294.967.295	123
Real	32	+/-1,18 x 10 ⁻³⁸ a +/-3,40 x 10 ³⁸	123,456, -3,4, -1,2E+12, 3,4E-3
LReal	64	+/-2,23 x 10 ⁻³⁰⁸ a +/-1,79 x 10 ³⁰⁸	12345,123456789 -1,2E+40
Time	32	T#-24d_20h_31m_23s_648ms a T#24d_20h_31m_23s_647ms Salvo como: -2,147,483,648 ms a +2,147,483,647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String	Variável	0 a 254 caracteres em tamanho de byte	'ABC'
DTL	96	Min.: DTL#1970-01-01-00:00:00.0 Max.: DTL#2262-04-11-23:47:16.854775807	DTL#2008-12-16-20:30:20.250

3.2.1 Sinais Analógicos

Os sinais digitais não variam continuamente ao longo do tempo, apenas podem assumir dois estados lógicos, o estado 0 e o estado 1.

Ao contrário de um sinal digital, os sinais analógicos podem registar qualquer valor dentro de uma determinada gama. Exemplos de variáveis analógicas são:

- Temperatura;
- Luminosidade;
- Humidade;
- Rotação;
- entre outras.

Estas variáveis são convertidas em tensões ou correntes elétricas com a ajuda de um transdutor. Quando se deseja registar, por exemplo, a rotação, a gama de rotação de 500 a 1500 rpm pode ser convertida numa gama de tensão de 0 a +10 V (Figura 3-9). Estas tensões ou correntes elétricas são conectadas a um módulo analógico que digitaliza este sinal [37].

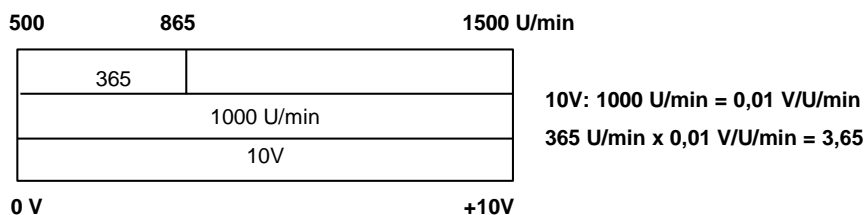


Figura 3-9: Conversão de variáveis.

Ao processar variáveis analógicas no PLC, o valor lido de tensão ou corrente deve ser convertido em informação digital (Figura 3-10). Esta conversão é denominada como conversão analógica-digital (Conversão A/D). Isto significa que, por exemplo, o valor de tensão de 3,65 V é armazenado como informação na forma de uma série de dígitos binários. Quanto mais dígitos binários forem usados para a representação digital, maior será a resolução [37].

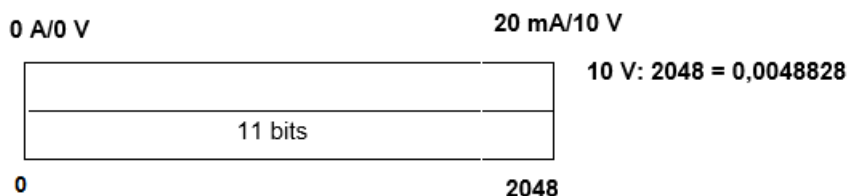


Figura 3-10: Conversão digital de variáveis analógicas.

3.2.2 Leitura de sinais analógicos

A leitura de sinais digitais é facilmente executada, uma vez que, apenas se tem de detetar a transição do estado lógico 0 (1 bit) para 1 ou vice-versa.

Os sinais analógicos são lidos no formato “*Word*”. Em seguida apresentam-se alguns exemplos de endereçamento de sinais analógicos:

- %IW64 Palavra de entrada analógica 64
- %IW66 Palavra de entrada analógica 66
- %QW80 Palavra de saída analógica 80

Quando um valor de entrada ou saída analógica estiver disponível precisa de ser normalizado para que os valores numéricos correspondam às variáveis físicas do processo (Figura 3-11) [37].

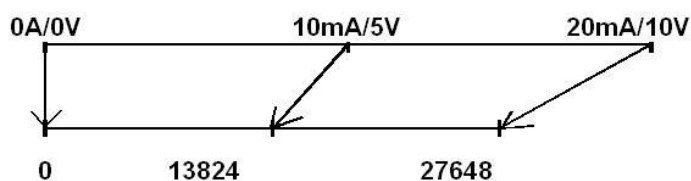


Figura 3-11: Normalização de variáveis analógicas.

Por fim, é necessário mapear o valor para um intervalo de valores específico (Figura 3-12). Quando a instrução é executada, o valor é dimensionado para a gama de valores que foi definido pelos parâmetros MIN e MAX.

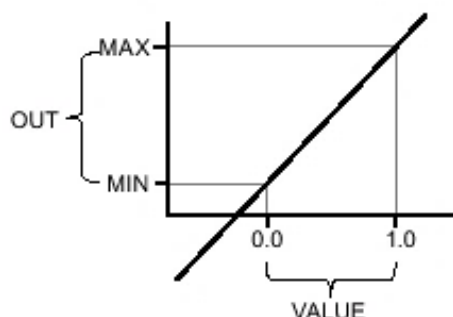


Figura 3-12: Mapeamento de valores.

3.2.3 Funcionalidades implementadas

Após reunida toda a informação foram implementadas inúmeras funcionalidades em um PLC, nomeadamente a gestão e o controlo de um sistema de climatização, iluminação, estores, rega, intrusão, simulação de presença, entre outras. As funcionalidades serão posteriormente geridas remotamente. Para a sua implementação foi necessário definir determinadas propriedades no CPU e associar módulos.

Apesar do tema do projeto se direcionar para o comando à distância, as funcionalidades implementadas permitem o comando local e remoto.

3.2.3.1 Climatização

A climatização é das funcionalidades que mais conforto proporciona numa habitação, tornando o ambiente bastante acolhedor e agradável.

A climatização foi a primeira funcionalidade implementada no TIA Portal. A referência é a temperatura desejada e em conformidade com o seu valor parte do sistema reage. A temperatura é medida com um sensor de temperatura LM35 (Figura 3-13) [40]. Trata-se de um sensor com uma precisão adequada para o efeito, que apresenta uma saída de tensão linear relativa à temperatura em que ele se encontrar no momento. Alimentado por uma tensão de 4-20V DC e GND, apresenta na saída um sinal de 10mV para cada Grau Celsius de temperatura. Assim, apresenta uma vantagem em relação aos demais sensores de temperatura calibrados em “Kelvin”, não necessitando de nenhuma subtração de variáveis para que se obtenha uma escala de temperatura em Graus Celsius.

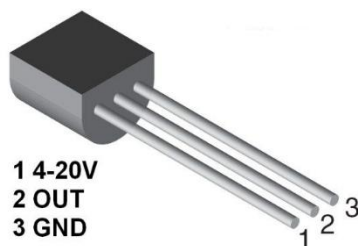


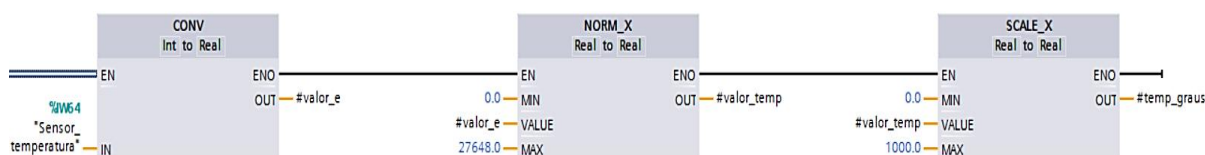
Figura 3-13: Sensor de Temperatura LM35.

Relativamente ao programa elaborado no Tia Portal para a climatização é composto por vários blocos programados em LAD e SCL,

- Climatização(FB)(DB);
- Média Deslizante(FB)(DB);
- data_hora(FB)(DB);
- DataLog(FB)(DB);

- Toggle_Preset(FB)(DB);
- data_hora_atual(DB);
- Climatização(FC);
- Outras funções auxiliares.

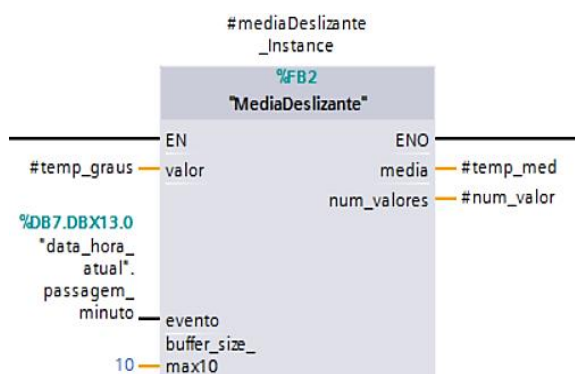
Inicialmente ligou-se à entrada analógica IW64 do PLC o sensor de temperatura, para deste modo se conseguir obter uma leitura de valores. À medida que o sensor capta mais ou menos “tensão”, a entrada do PLC deteta a variação e conseqüentemente, converte-a em °C, através das funções CONV (realiza uma conversão), NORM_X (normaliza valores 0-27648) e SCALE_X (realiza um mapeamento de valores num determinado intervalo), para se tornar perceptível ao utilizador (Secção de código 3-1).



Secção de código 3-1: Leitura do sensor de temperatura.

Como o sensor de temperatura é muito sensível à variação de temperatura foi necessário criar em linguagem SCL uma média deslizante (por ser mais realista), como se apresenta na Secção de código 3-2. Esta média deslizante lê o valor instantâneo da temperatura e realiza a média das últimas dez temperaturas registadas num periodo de tempo e é esse valor que vai servir de base para as instruções seguintes.

Para evitar a utilização intensiva de uma função do sistema que faz a leitura do tempo de relógio em tempo real (RTC) do PLC, pois é necessário saber o instante de tempo atual em várias outras funcionalidades implementadas, optou-se por colocar os valores lidos do RTC numa posição de memória. Assim foi necessário programar uma leitura da hora atual para ativar variáveis em diferentes períodos de tempo (passagem num minuto, em dez minutos, numa hora, numa semana, num mês) e guardá-las num bloco de dados para serem utilizadas também pela média deslizante. Após alguns testes realizados, optou-se pelo periodo de “passagem num minuto” para a média deslizante, uma vez que é a solução mais realista e que estabiliza mais a temperatura (Figura 3-14).



Secção de código 3-2: Média deslizante.

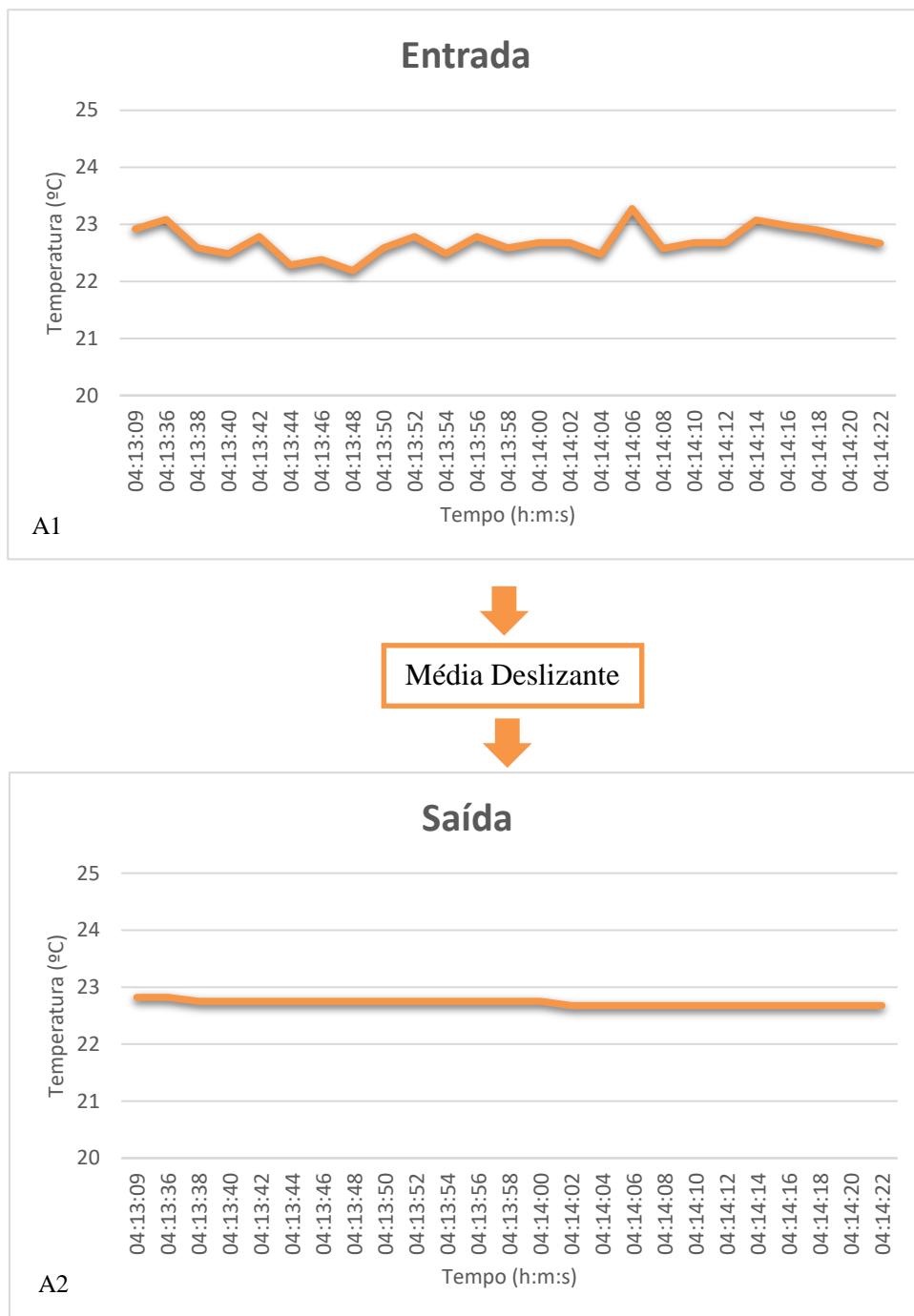
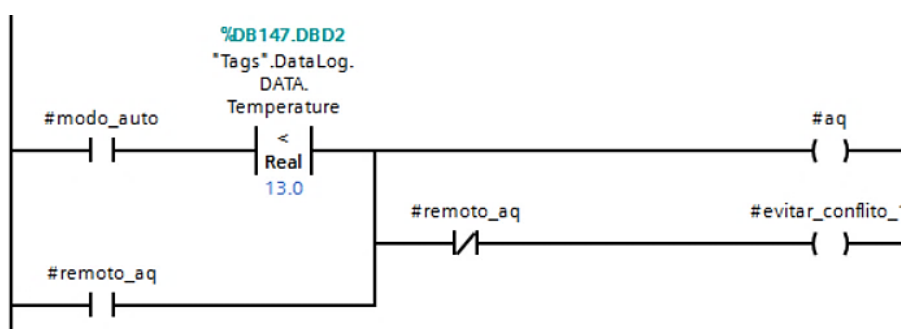


Figura 3-14: Resultado da temperatura com a média deslizante (A1: Temperatura a variar na entrada; A2: Temperatura na saída com a média deslizante).

Existem dois modos de comando, o modo automático, sem necessidade de intervenção instensiva do utilizador e o controlo remoto (controlo aquecimento e controlo AC), como se ilustra na Secção de código 3-3.

- O modo automático pode ser ativado localmente e remotamente. Caracteriza-se por ser uma gestão interna, ou seja, caso o valor obtido pela média deslizante seja inferior a 13°C, a variável aquecimento é ativada, caso o valor obtido seja superior a 27°C, a variável AC é ativada. Quando a temperatura se encontra entre os valores anteriormente mencionados é realizado um reset às variáveis aquecimento e AC.
- O controlo remoto, como o próprio nome indica, apenas pode ser ativado remotamente. Independentemente do valor de temperatura lido no momento, o utilizador, sempre que desejar pode ativar a variável aquecimento ou a variável AC.

Sempre que estes dois modos de operação forem ativados são ativadas variáveis (que se encontram num DB específico) de leitura remota do estado. Os dois modos têm de funcionar separadamente para evitar eventuais conflitos.



Secção de código 3-3: Modos de comando da variável correspondente ao aquecimento.

Sempre que a temperatura for excessivamente elevada (>35°C) e o modo automático estiver desligado é enviada uma mensagem ao utilizador a alertá-lo da situação. O mesmo acontece quando a temperatura é excessivamente baixa (<10°C). Ou seja, sempre que se verificarem situações anormais recorre-se ao bloco de função relativo ao envio das mensagens que faz uso das instruções TC_CON, TC_SEND e TC_DISCON do TIA Portal.

Também foi implementada uma função que permite o registo e armazenamento de dados relativos ao sistema, nomeadamente a temperatura registada ao longo do tempo, como se observa na Tabela 3-2. Para complementar a informação, cada registo tem associado o nº de registo, a data e a hora a que foi efetuado (*timestamp*).

Tabela 3-2: Parte dos registos da temperatura

Record	Date	Time	Temperature
12001	19/07/2016	03:45:13	28,4
12002	19/07/2016	03:45:15	28,5
12003	19/07/2016	03:45:17	28,5
12004	19/07/2016	03:45:19	28,5
12005	19/07/2016	03:45:21	28,5
12006	19/07/2016	03:45:23	28,5
12007	19/07/2016	03:45:25	28,6

Neste caso o utilizador devia apenas ter o cuidado, por exemplo, de mês a mês descarregar e apagar os *DataLogs* da memória interna do PLC, pois o PLC possui memória finita e quando chega ao fim de um certo tempo de armazenamento de dados ou quando a sua memória interna fica cheia, começa a sobrepor os valores registados nos *DataLogs* anteriores. Por forma a solucionar o problema de falta de memória e não se perder informação relevante procedeu-se ao *download* automático de arquivos *DataLog* através de uma função do computador denominada de "tarefas agendadas", onde é colocado o endereço do argumento e agendada a hora de descarregamento (Figura 3-15). Esta solução requer o uso de um navegador de Internet específico, o *Mozilla Firefox*. Cada vez que um ficheiro é descarregado terá o nome idêntico ao nome do arquivo descarregado/utilizado anteriormente.

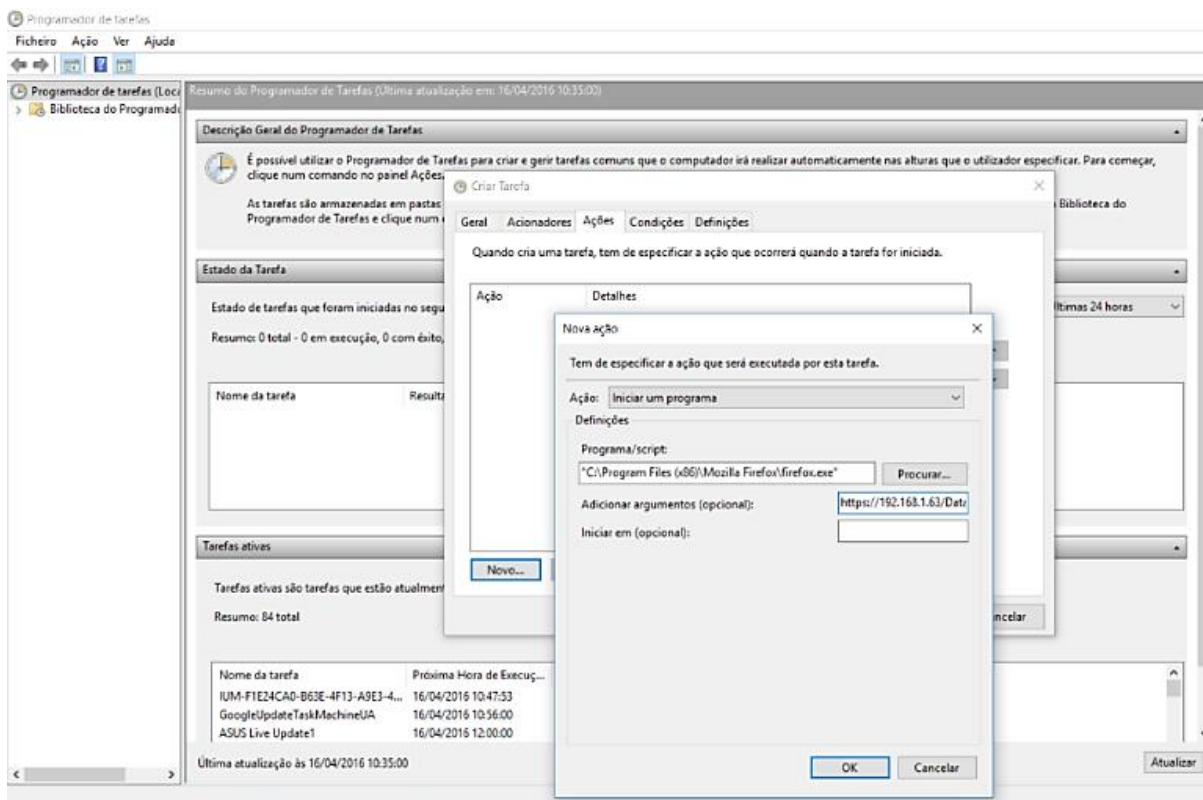


Figura 3-15: Tarefas agendadas, função do PC.

O modo de consultar, aceder e apagar os *DataLogs* referidos é realizado através da interface *Web Server* do PLC, Figura 3-16.

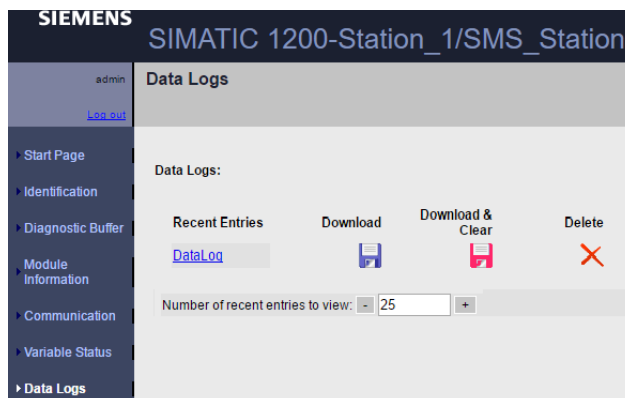
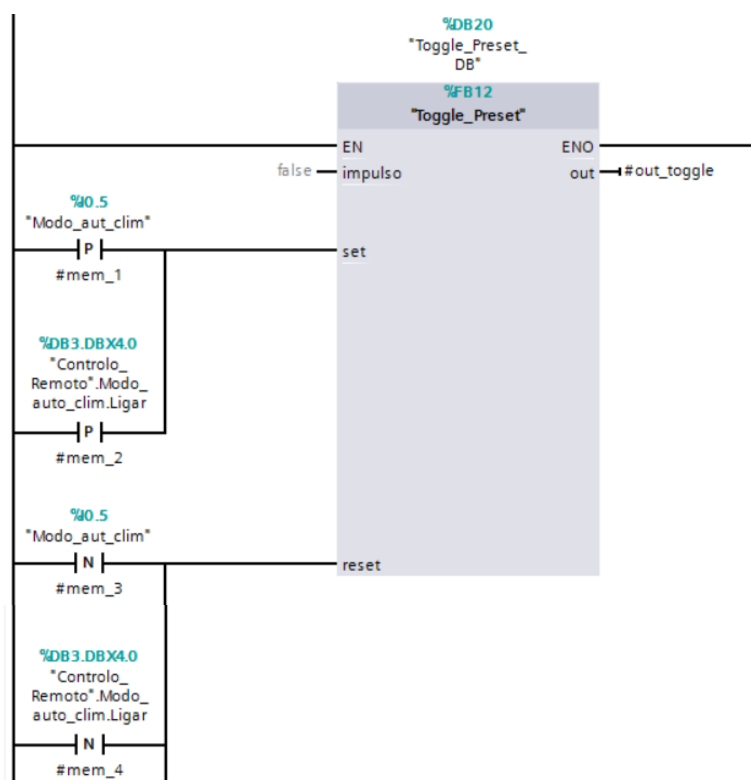


Figura 3-16: Interface *Web Server* do PLC

Na função “Climatização” concentram-se todos os blocos de funções anteriormente citados e um outro designado de “Toggle_Preset” que reage aos impulsos realizados pelo utilizador para que, deste modo, as variáveis possam ser manipuladas, sem interferirem umas com as outras (Secção de código 3-4). Assim torna-se possível, por exemplo, ligar o modo automático localmente e desligá-lo remotamente, sem que a primeira opção prevaleça em relação às restantes.



Secção de código 3-4: Bloco de função “Toggle_Preset”.

3.2.3.2 Iluminação

A iluminação foi outra das funcionalidades implementadas no TIA Portal. É uma funcionalidade de grande importância não só pela sua imprescindibilidade numa habitação como, por contrapartida, é responsável por cerca de 40% do consumo de energia numa habitação. Torna-se assim necessário utilizá-la de uma forma racional.

Relativamente ao programa elaborado no Tia Portal ele é composto por vários blocos programados em LAD,

- Iluminação(FB)(DB);
- Iluminação_Analógica(FB)(DB);
- Iluminação_Presença(FB)(DB);
- Iluminação_P (FC);
- Outras funções auxiliares.

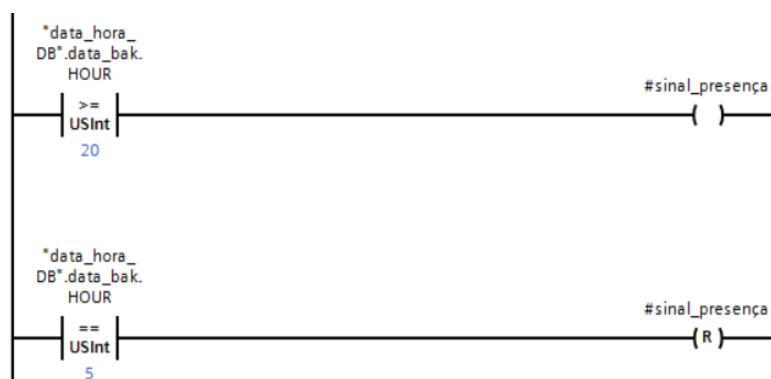
A iluminação é uma funcionalidade muito abrangente, podendo ser tratada de diversas maneiras. O programa elaborado possui três maneiras diferentes de funcionamento, a iluminação usual das habitações (ON/OFF), a iluminação através da criação de ambiente (regulação da intensidade luminosa) e a iluminação programada para um determinado horário.

- A iluminação usual recorre ao controlo local e remoto. Localmente faz uso da manipulação de variáveis de entradas para assim ativar/desativar a respectiva variável de saída e sempre que for necessário ativar/desativar a variável de saída também é possível fazê-lo remotamente, com ajuda, por exemplo, de um *smartphone*. Assim o utilizador não precisa de se preocupar em encontrar o interruptor do quarto às escuras, ou mesmo quando se esquecer de desligar uma lâmpada localmente, pode fazê-lo remotamente, evitando assim desperdícios de energia. Sempre que a variável de saída é ativada é também ativada em simultâneo uma outra variável que serve de leitura remota do estado.
- A regulação da intensidade luminosa permite ao utilizador criar ambientes próprios para determinadas ações que pretenda realizar. Por exemplo, se o utilizador desejar, pode regular a iluminação para a ver um filme, ler, entre outras coisas. A regulação será apenas conseguida com o auxílio, por exemplo, de um *smartphone* ou de um PC, que manipula uma variável que se encontra num DB relativo apenas a potencialidades remotas. Assim, torna-se possível criar o tão desejado ambiente, sem o utilizador ter de se ausentar do seu conforto. Na programação da regulação da intensidade luminosa manipula-se a gosto a intensidade pretendida (compreendida entre 0 e 10), fazendo variar a saída analógica, QW80, do módulo associado (Secção de código 3-5).



Secção de código 3-5: Obtenção da saída analógica.

- A iluminação programada para um determinado horário ativa a variável para o horário definido pelo utilizador. (Secção de código 3-6).



Secção de código 3-6: Programação horária.

3.2.3.3 Estores elétricos

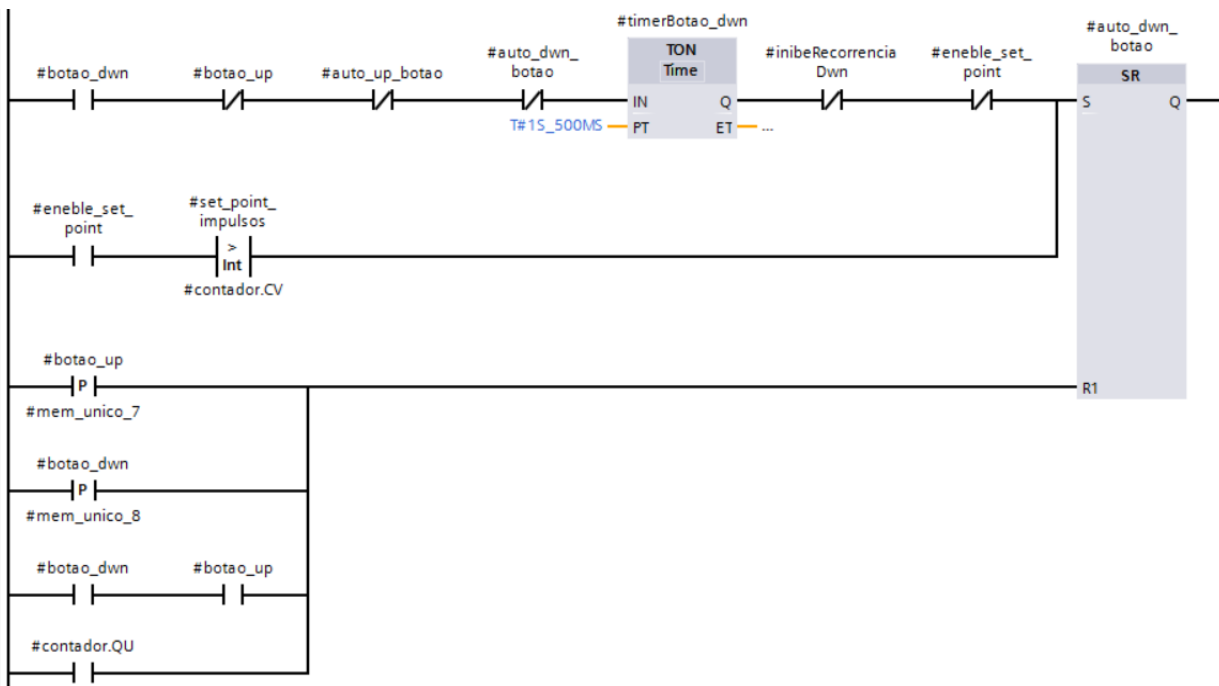
Os estores elétricos foi outra funcionalidade implementada no PLC. Nos dias de hoje, é das funcionalidades mais adotadas e quando acompanhada de uma boa gestão contribui muito na melhoria do ambiente da habitação. Por exemplo, em dias de inverno com algum sol, se o utilizador elevar os estores com o intuito de aproveitar durante o dia a incidência da luz solar, para além de prescindir durante algum tempo da iluminação artificial e do aquecimento, diminui a fatura de eletricidade. Assim como no verão, se se baixarem os estores durante os picos de calor contribui para manter o ambiente mais fresco e conseqüente poupança de energia.

Relativamente ao programa elaborado no Tia Portal, é composto essencialmente por dois blocos programados em LAD,

- EstoreUpDwn (FB)(DB);
- Estore(FC);

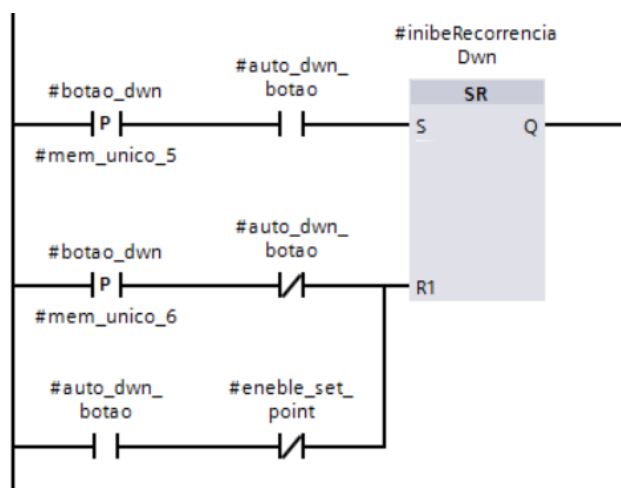
Inicialmente a ideia seria controlar os estores elétricos com o auxílio de um sensor de luminosidade, por exemplo, um LDR. No entanto, devido às limitações de entradas analógicas do PLC o controlo dos estores alterou-se.

O comando dos estores baseou-se essencialmente na manipulação de variáveis de entrada para ativar variáveis de saída. Existem essencialmente dois modos e duas formas de comandar os estores, localmente e remotamente, por pressão imediata ou de forma prolongada respetivamente. Sempre que se desejar subir ou descer os estores por etapas recorre-se a uma pressão momentanea no botão, se se desejar uma subida ou descida na totalidade (Secção de código 3-7, exemplo), recorre-se a uma pressão prolongada no botão. Programou-se a pressão prolongada num periodo igual ou superior a um segundo e meio por forma a evitar que o utilizador permaneça tanto tempo a pressionar os botões. Assim sempre que esse tempo for atingido, os estores sobem e descem automaticamente na sua totalidade.

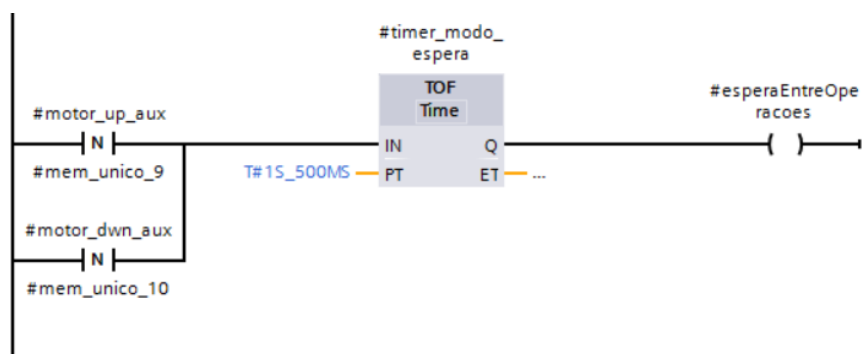


Secção de código 3-7: Descida automática do estore.

Por forma a evitar eventuais problemas realizou-se uma inibição de recorrência de descida (Secção de código 3-8) e subida consecutiva dos estores e definiu-se um tempo de espera entre operações (Secção de código 3-9).

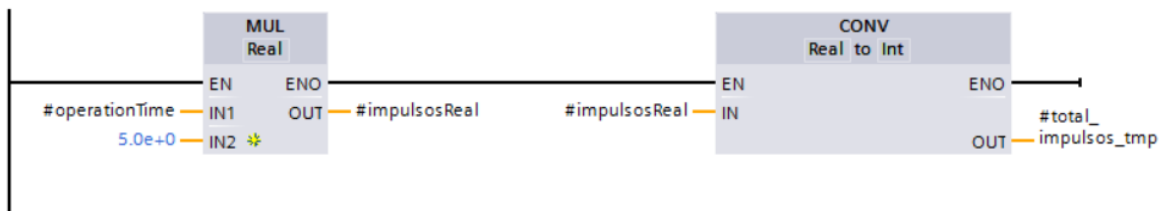


Secção de código 3-8: Inibição de recorrência de uma descida

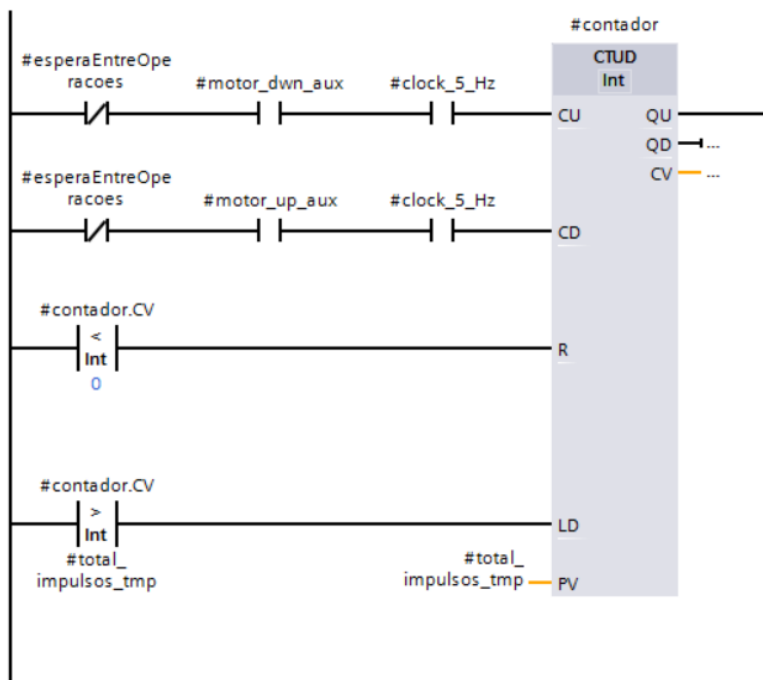


Secção de código 3-9: Espera entre operações.

Existem várias formas de definir a duração e os limites de subida e descida dos estores, no entanto, a escolha recaiu num contador. Primeiramente definiu-se uma variável “#operationTime” à qual se atribuiu um valor real (duração máxima de trajeto) que a qualquer altura pode ser modificado. Posteriormente esse valor é convertido em impulsos (Secção de código 3-10). Esses impulsos vão entrar nos limites máximos do contador e quando alcançados traduzem-se numa paragem do contador e consequentemente numa paragem de movimento. O contador, Secção de código 3-11, realiza a contagem das várias posições que os estores podem ocupar, por forma a acompanhá-los passo a passo, podendo ser observada em qualquer momento a posição a que se encontram. Quando o estore se encontra descido na totalidade toma o valor máximo do contador, quando se encontra subido na totalidade toma o valor 0 do contador.

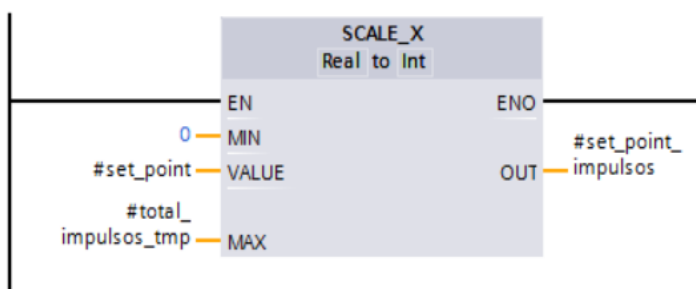


Secção de código 3-10: Conversão da variável em impulsos.



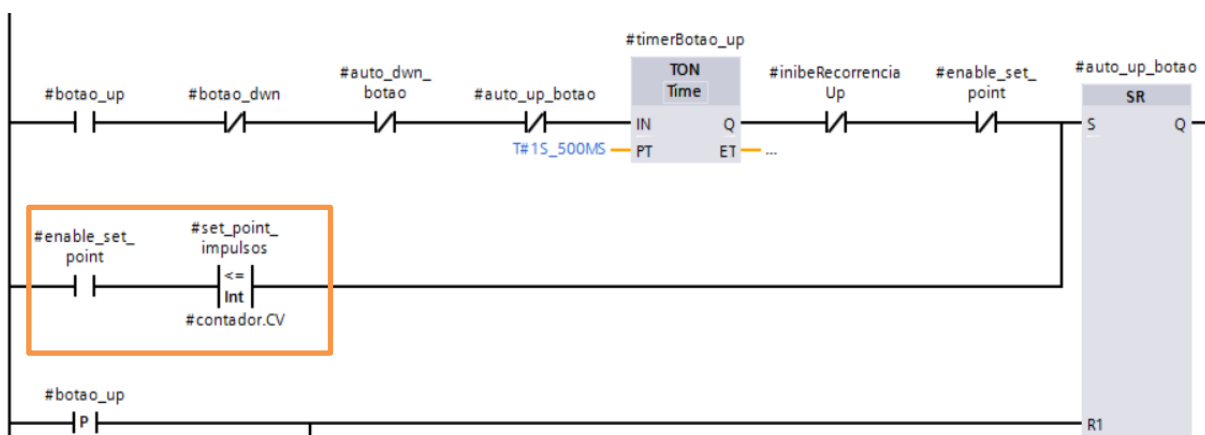
Secção de código 3-11: Contador.

Uma particularidade do programa, para além da possibilidade de remotamente se poder realizar a subida e descida dos estores na totalidade, é a capacidade de definir posições a que se deseja que eles se posicionem, atribuindo valores à variável “#set_point” para serem convertidos em impulsos, como se observa na Secção de código 3-12.



Secção de código 3-12: Conversão da variável em impulsos.

Para isso é necessário ativar a variável “#enable_set_point” e sempre que esta variável for ativada é impossível realizar a subida e a descida dos estores na totalidade e vice-versa. A colocação dos estores numa determinada posição é realizada por comparação, por exemplo, na Secção de código 3-13. Ou seja, sempre que a variável “#enable_set_point” estiver ativa e por exemplo, o valor da variável “#set_point_impulsos” for menor ou igual que o valor do contador é realizada uma subida automática dos estores até a posição pretendida, caso seja maior que o valor do contador é realizada uma descida automática dos estores até a posição pretendida.



Secção de código 3-13: Ativação e comparação para posições do estore.

3.2.3.4 Estore do painel solar

Um dos grandes problemas dos painéis solares é o seu sobreaquecimento e uma das formas de contornar o problema e, talvez a menos conhecida, mas não deixa de ser uma solução eficiente, é a colocação de estores nos painéis solares, como se ilustra na Figura 3-17.

Para além da colocação dos estores (sombreamento) existe ainda a possibilidade de colocação de dissipadores de calor, no entanto, é uma solução mais cara e mais trabalhosa.

O sobreaquecimento dos painéis solares é facilmente identificável através das altas temperaturas no circuito primário do sistema solar térmico, entrando muitas vezes o sistema em estagnação. Este problema deve-se ter em conta logo na fase do dimensionamento [41].



Figura 3-17: Painel solar com estore.

Relativamente ao programa elaborado no Tia Portal ele é composto essencialmente por dois blocos programados em LAD,

- Painel (FB)(DB);
- Painel_estore(FC);

Devido às limitações de entradas analógicas do PLC o controlo dos estores do painel foi realizado com o auxílio do sensor de temperatura LM35, o sensor de referência da climatização. Inicialmente definiu-se um modo automático no programa e sempre que este modo for ativado (localmente e remotamente) e o valores de temperatura registada for excessivamente alta em relação ao limite imposto, procede-se imediatamente à descida do estore por forma a evitar o sobreaquecimento. De notar que quando o modo automático for ativado é ativada uma variável de leitura remota do estado.

Sempre que o modo automático não estiver ativo e se verificarem valores elevados de temperatura é enviada uma mensagem ao utilizador a alertá-lo.

De notar que este sistema implementado não é eficiente na sua totalidade muito devido às limitações do número de entradas analógicas disponíveis e do sensor utilizado que mede temperaturas numa gama entre -55°C a 150°C . Os painéis solares no modo de estagnação podem atingir temperaturas da ordem dos 200°C .

3.2.3.5 Portão Automático

Outra das funcionalidades implementadas no PLC foi o controlo do portão automático. O portão automático é muito usado em habitações, pois além de reforçar a segurança do património, poupa o tempo gasto para abrir um portão manual que muitas vezes é pesado.

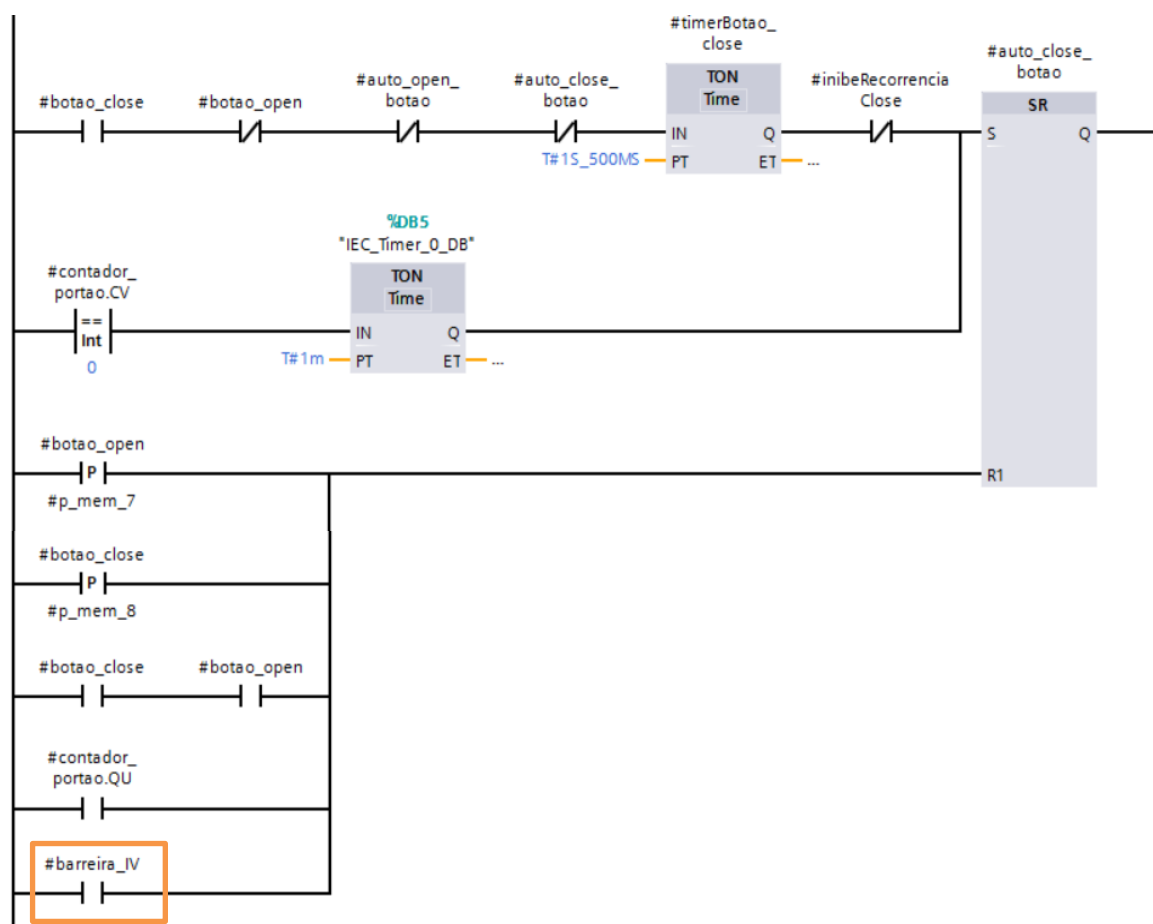
O portão comum requer que a pessoa saia do carro para poder abri-lo e muitas das vezes torna-se um incómodo, nomeadamente quando se está atrasado ou até mesmo quando chove. O portão automático é a melhor das soluções.

Relativamente ao programa elaborado no Tia Portal ele é composto principalmente por dois blocos programados em LAD,

- PortaoOpenClose (FB)(DB);
- Portão(FC);

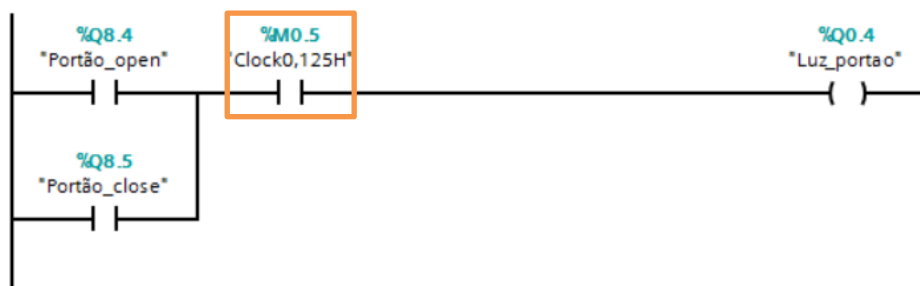
O programa implementado do portao automático é muito semelhante ao comando dos estores eletricos à exceção da capacidade de remotamente definir posições, uma vez que, não é muito relevante no caso.

O que se torna imprescindível num portão automático e que foi implementado é claramente a existência de uma barreira de raios infravermelhos que cada vez que é accionada faz um *reset* ao fecho automático do portão por forma a zelar pela seguranças de pessoas e veículos (Secção de código 3-14).



Secção de código 3-14: Barreira de raios infravermelhos.

Outra característica do programa é a iluminação intermitente durante a abertura e o fecho do portão (Secção de código 3-15).



Secção de código 3-15: Luz intermitente.

3.2.3.6 Sistema de Rega

O grande objetivo do sistema de rega automática é o fornecimento eficiente de água para garantir que a vegetação se mantenha viva e em bom estado de conservação, para que o espaço tenha a função e a estética desejada.

O comando do sistema de rega foi outras das funcionalidades implementadas no TIA Portal. A referência de todo o sistema é a humidade, uma vez que, é em conformidade com o seu valor que o sistema reage. O comando do sistema de rega realizou-se com o auxílio de um sensor de humidade do solo que realiza a leitura da quantidade de humidade presente no solo (Figura 3-18).

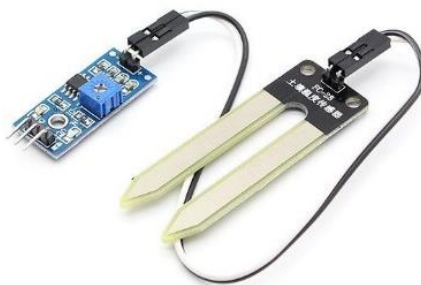


Figura 3-18: Sensor de humidade do solo.

É um sensor ideal para monitorar um jardim ou plantas de estimação que utiliza dois eléctrodos para passar corrente pelo solo e assim ler o nível de humidade por comparação com a resistência. Num solo húmido maior é a condução e conseqüentemente menor a resistência, num solo seco existe uma maior dificuldade de condução.

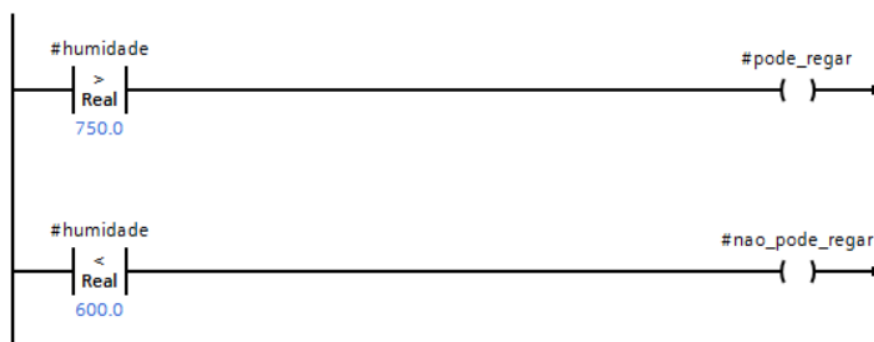
Relativamente ao programa elaborado no Tia Portal ele é composto por vários blocos programados em LAD:

- Rega(FB)(DB);

- Aux_rega(DB);
- Rega(FC);
- Outras funções auxiliares.

Inicialmente ligou-se à entrada analógica IW66 do PLC o sensor de humidade do solo (5V). De notar que o fabricante do sensor não indica qual a unidade de grandeza que utiliza. No entanto para se conseguir obter uma leitura da variável primária e convertê-la supostamente em humidade, utilizou-se as funções CONV (realiza uma conversão), NORM_X (normaliza valores 0-13824 por ser alimentado a 5V) e SCALE_X (realiza um mapeamento de valores num determinado intervalo).

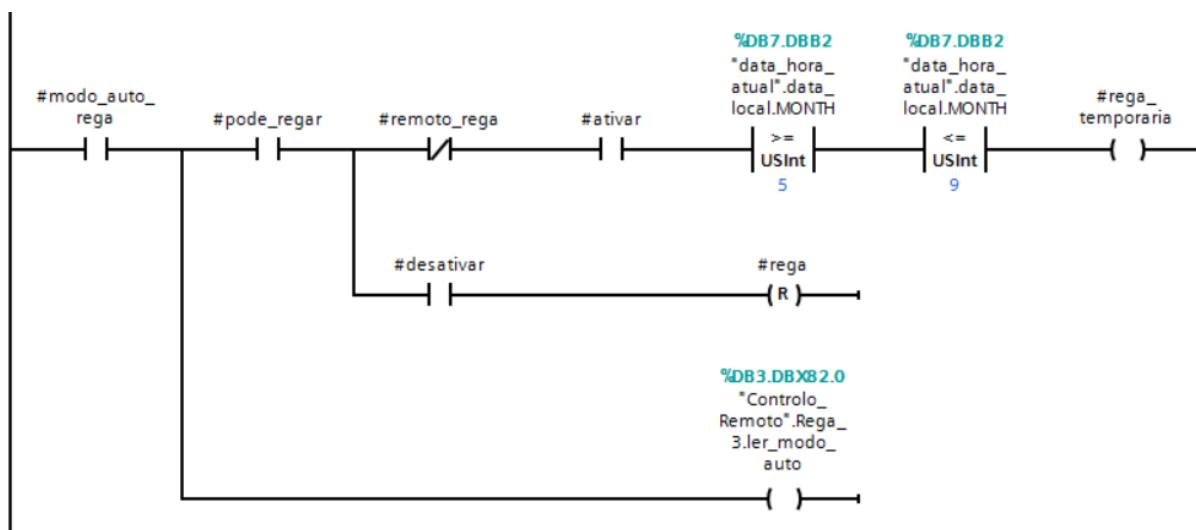
Após diversos testes, verificou-se que o sensor apresentava valores que variavam de 0 a 1000, com a sua introdução em diferentes tipos de solos. No entanto, verificou-se que os resultados eram diferentes aos valores esperados, ou seja, em solo seco apresentava valores superiores a 750 e em solo húmido apresentava valores inferiores a 600 (Secção de código 3-16).



Secção de código 3-16: Humidade e consequente ativação de variáveis.

No programa existem dois modos de comando, o modo automático, sem necessidade de intervenção direta do utilizador, e o controlo remoto.

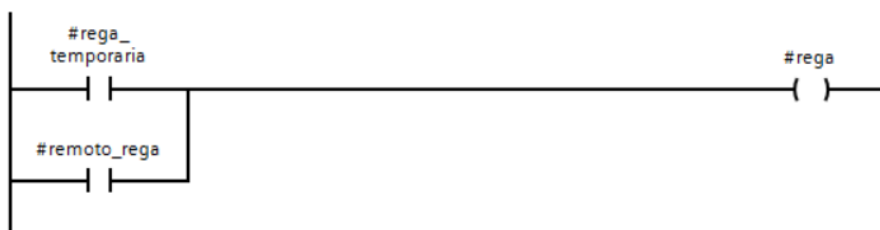
- No modo automático (pode ser ativado localmente e remotamente) caso o valor de humidade obtido seja superior a 750 e se verificarem igualdades nas comparações das horas (comparação da hora do PLC com a hora pré-definido num DB) e dos meses programados (comparação do mês do PLC com o mês pré-definida num DB) é ativada uma variavel de saída correspondente à rega e uma variável referente à leitura remota do estado. Quando a hora do PLC coincidir com a hora programada de fim de rega é realizado um reset à variavel correspondente à rega e à variavel referente à leitura remota do estado (Secção de código 3-17).



Secção de código 3-17: Programação da rega automática.

- O controlo remoto é independente do valor da humidade obtido e é ativado sempre que o utilizador assim o pretenda, podendo-se verificar o seu estado em qualquer momento.

Deve-se porém evitar que os dois modos de comando não funcionem em simultâneo (Secção de código 3-18). Para isso é necessário uma leitura previa do estado das variáveis.



Secção de código 3-18: Modos de comando da rega.

3.2.3.7 Intrusão

O sistema de intrusão é projetado e usado para sinalizar situações de perigo, particularmente no caso de intrusão ou roubo e foi outra das funcionalidades implementadas no TIA Portal. Realizou-se com o auxílio de um sensor magnético (Figura 3-19), colocado num local estratégico com o objetivo de assegurar a proteção de determinados espaços consoante o grau de segurança pretendido.

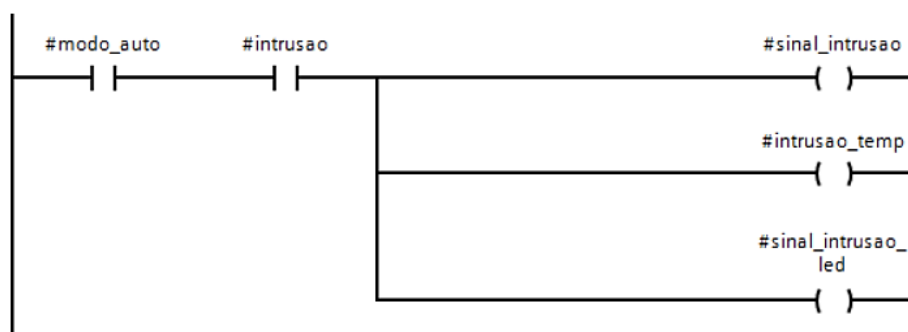


Figura 3-19: Sensor magnético.

Relativamente ao programa elaborado no Tia Portal ele é composto por vários blocos programados em LAD,

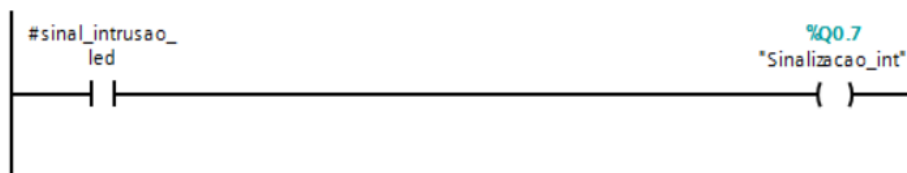
- Intrusao(FB)(DB);
- Intrusao(FC);
- Outras funções auxiliares.

Primeiramente definiu-se um modo automático de comando (Secção de código 3-19) para sempre que possível ativar e desativar o sistema de intrusão. Assim, sempre que o modo automático for ativado juntamente com o sensor magnético é ativada uma variável de saída que corresponderá a um aviso sonoro e ao envio de mensagens a determinados habitantes da casa em intervalos de um minuto.



Secção de código 3-19: Modo automático do sistema de intrusão.

Sempre que a sirene é ativada é também ativada uma variável correspondente a uma sinalização de perigo, como se verifica na Secção de código 3-20.



Secção de código 3-20: Sinalização de intrusão.

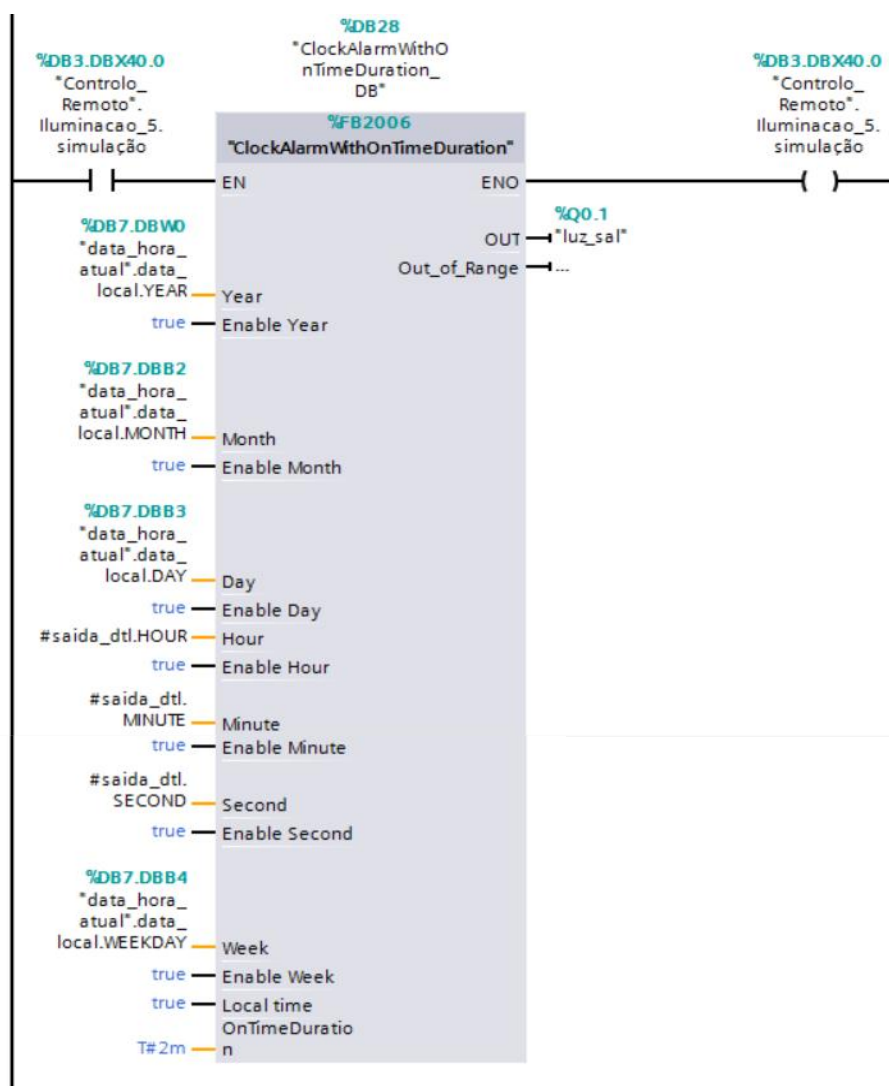
3.2.3.8 Simulação de presença

A simulação de presença foi outra das funcionalidades implementadas no TIA Portal. A simulação de presença corresponde a táticas realizadas em situações de ausência de pessoas numa casa (férias ou períodos mais alargados). É possível simular a atividade de pessoas mediante a activação da iluminação, do sistema de som, subida ou descida de estores, ou qualquer outro equipamento doméstico. No entanto, a escolha recaiu na ativação aleatória da iluminação.

Relativamente ao programa elaborado no Tia Portal ele é composto por vários blocos programados em LAD,

- Simulacao_rand_ml(FC);
- Simulacao_randTimeAround(FC);
- ClockAlarmWithOnTimeDuration(FB)(DB);
- Simulação_iluminação(FB)(DB);
- Outras funções auxiliares.

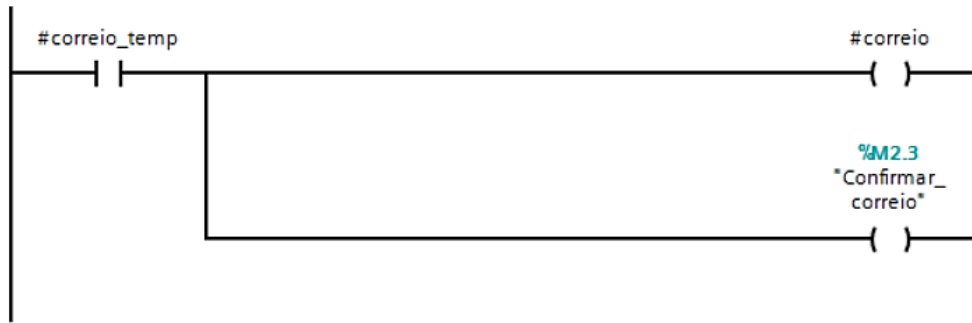
No programa, o modo simulação de presença permite que a habitação tenha uma aparência de estar habitada (quando não se encontra ninguém em casa). Para isso, programaram-se as luzes a acenderem alternadamente nos diversos compartimentos da habitação num período definido (2min), dissuadindo assim possíveis assaltantes. Pormenorizadamente o modo de simulação é processado com o auxílio da função criada denominada de “Simulacao_Rand”. Esta função lê a data e hora atual, e como o valor dos nanosegundos é aquele que modifica com mais rapidez, é ele que vai servir de base para a criação de *random values*. De seguida na função “Simulacao_RandTime” programou-se uma hora de referência e um tempo limite para gerar tempos aleatórios. Posteriormente esses tempos aleatórios converteram-se em variáveis DTL. Com a ajuda da função de blocos “ClockAlarmWithOnTimeDuration” sempre que as variáveis DTL coincidirem com a hora e data atual será ativada durante um determinado período (para criar mais credibilidade) uma saída que corresponderá à iluminação, como se ilustra na Secção de código 3-21.



Secção de código 3-21: Bloco de função “ClockAlarmWithOnTimeDuration”

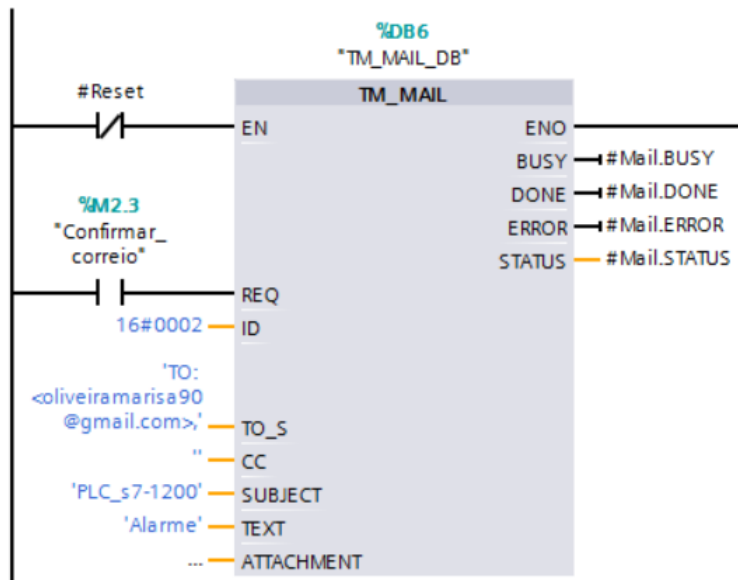
3.2.3.9 Correio

O correio foi outra das funcionalidades implementadas no TIA Portal e realizou-se com o auxílio de um sensor magnético, colocado dentro da caixa do correio. Assim, sempre que o carteiro colocar correspondência dentro da caixa de correio, o sensor é ativado e consequentemente é ativada uma variável de saída e uma posição de memória (Secção de código 3-22).



Secção de código 3-22: Aviso de existência de correspondência.

Essa memória vai servir de base para o envio de e-mails sempre que haja entrega de correspondência. No envio de e-mails é necessário “chamar” a instrução do TIA Portal designada de “TM_MAIL” (Secção de código 3-23) e sempre que esta instrução é “arrastada”, é criado automaticamente um bloco de funções e um DB, como se verifica na Figura 3-20. A instrução “TM_MAIL” usa o SMTP (*Simple Mail Transfer Protocol*) via porta 25 para enviar um e-mail via TCP/IP através da conexão Ethernet. A instrução é executada de forma assíncrona e o trabalho de transmissão envolve várias chamadas de “TM_MAIL”.



Secção de código 3- 23: Instrução “TM_MAIL”.

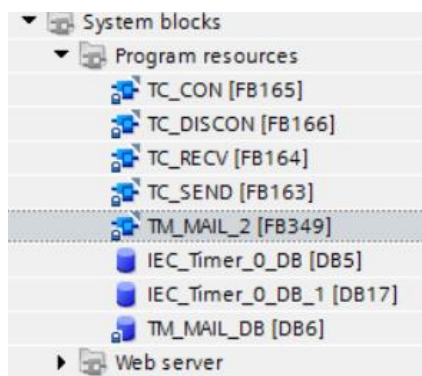
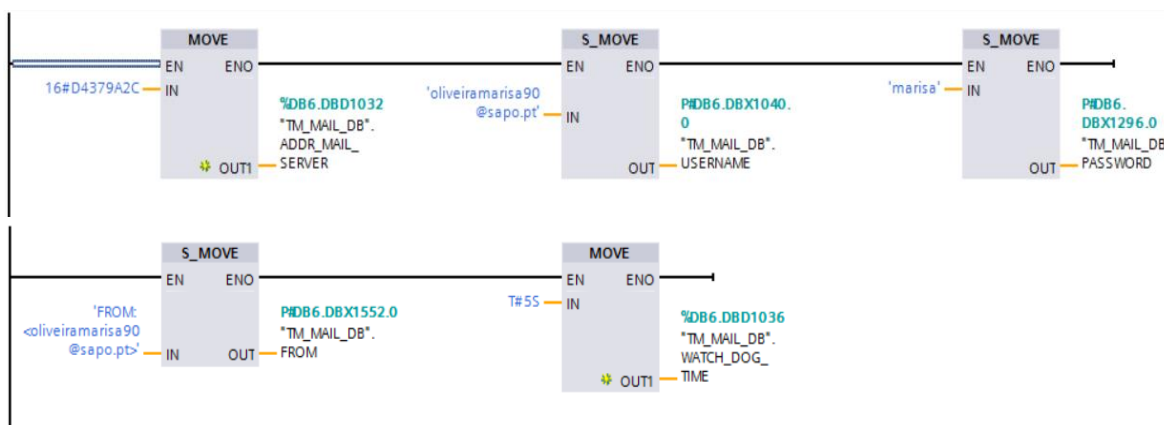


Figura 3-20: Blocos gerados.

Foi necessário definir determinados parâmetros para o efeito. Nomeadamente, o “ADDR_MAIL_SERVER” (endereço IP do servidor de e-mail), o mail, a password, o destinatário e o “WATCH_DOG_TIME” e atribuí-los ao DB criado (Secção de código 3-24).



Secção de código 3-24: Parâmetros imprescindíveis no envio de e-mails.

O mail, a password, o destinatário e o “WATCH_DOG_TIME” são parâmetros facilmente preenchidos, no entanto, para o preenchimento do “ADDR_MAIL_SERVER” foi necessário efetuar um *ping* na linha de comandos (Figura 3-21):

Nome: smtp.sapo.pt

Endereço IP: 212.55.154.44

Formato HEX: DW 16#D4379A2C

```
C:\Users\maris>ping smtp.sapo.pt

Pinging smtp.sapo.pt [212.55.154.44] with 32 bytes of data:
Request timed out.
Reply from 194.65.162.162: Destination net unreachable.
Request timed out.
Request timed out.

Ping statistics for 212.55.154.44:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),

C:\Users\maris>
```

Figura 3-21: Ping na linha de comandos.

No envio de e-mail é imprescindível colocar o IP do router nas propriedades do PLC (Figura 3-22).

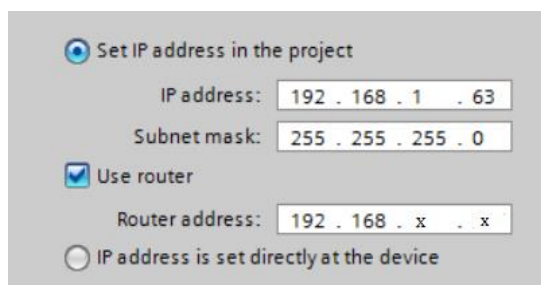


Figura 3-22: Propriedades do PLC.

Por fim, criou-se a função “Correio_FC” que contém os blocos de funções necessários para o envio de correio eletrónico.

3.2.3.10 Integração das funcionalidades

Para que todas as funcionalidades atrás referidas sejam processadas no PLC é necessário que sejam chamadas no programa principal (Main). Deste modo, no programa principal são chamadas as funções correspondentes a todas as funcionalidades implementadas.

3.3 Aplicação Android

De forma a complementar todo o sistema, foi criada uma outra potencialidade de comunicação e monitorização, a aplicação *android*. A aplicação *android* foi desenvolvida no *Android Studio*. Desta forma torna-se importante explicar o *Android Studio*, as suas potencialidades, estrutura e constituintes. Posteriormente descreve-se o desenvolvimento da aplicação.

3.3.1 Android Studio

Lançado pela *Google*, o *Android Studio* é o IDE (Ambiente de Desenvolvimento Integrado) oficial para o desenvolvimento de aplicações baseado na *IntelliJ IDEA*³. Partilha o mesmo objetivo do Eclipse + ADT (*Android Developer Tools*) (IDE concorrente), fornecendo aos seus utilizadores um ambiente de desenvolvimento interativo, um editor de código inteligente, modelos de códigos, desenvolvimento de aplicações em multitelas (*smartphones, tablets, Android Wear, Android TV, Android Auto e Google Glass*), dispositivos virtuais para todas as formas e tamanhos, entre outras funcionalidades. Segundo a própria *Google*, com o *Android Studio* a programação para *Android* é mais simples e rápida. O *Android Studio* está disponível para *Windows, MacOS e Linux* [42].

3.3.2 Estrutura do *Android Studio*

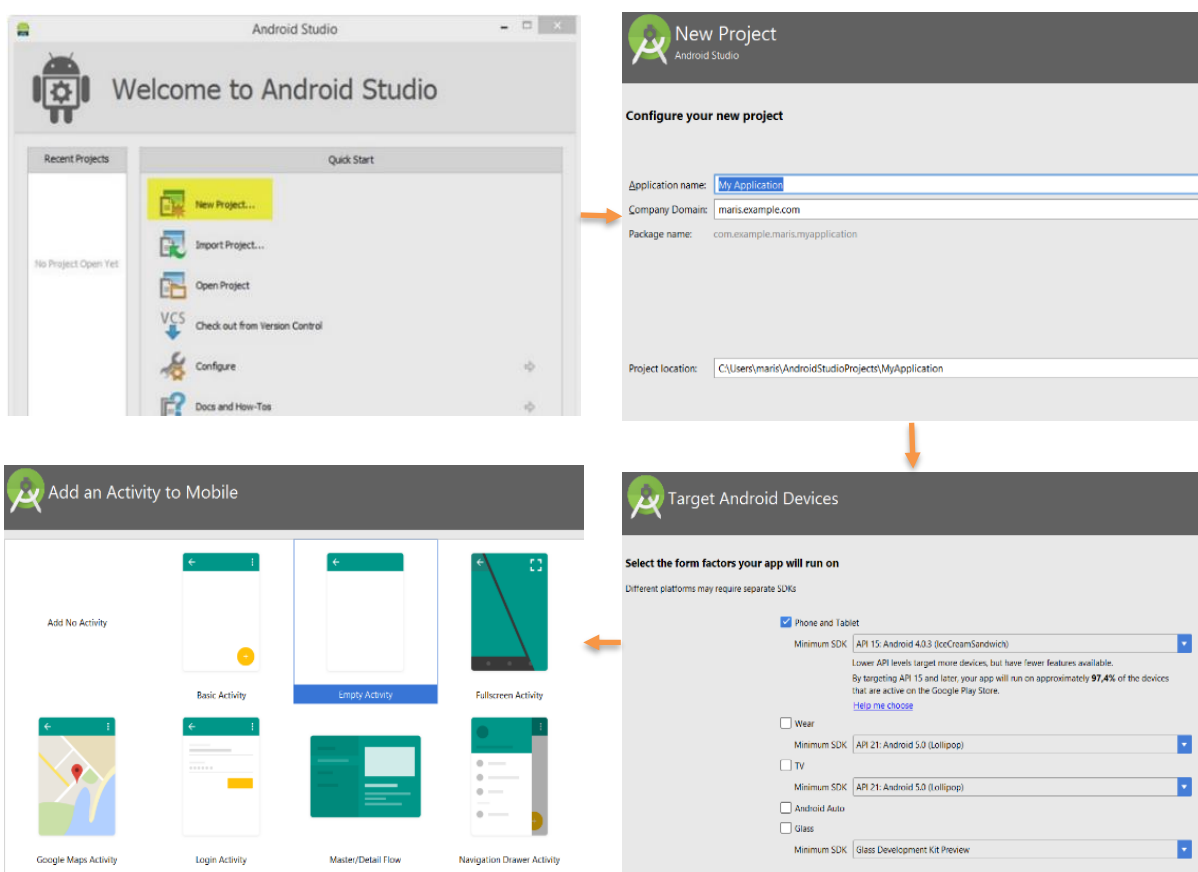
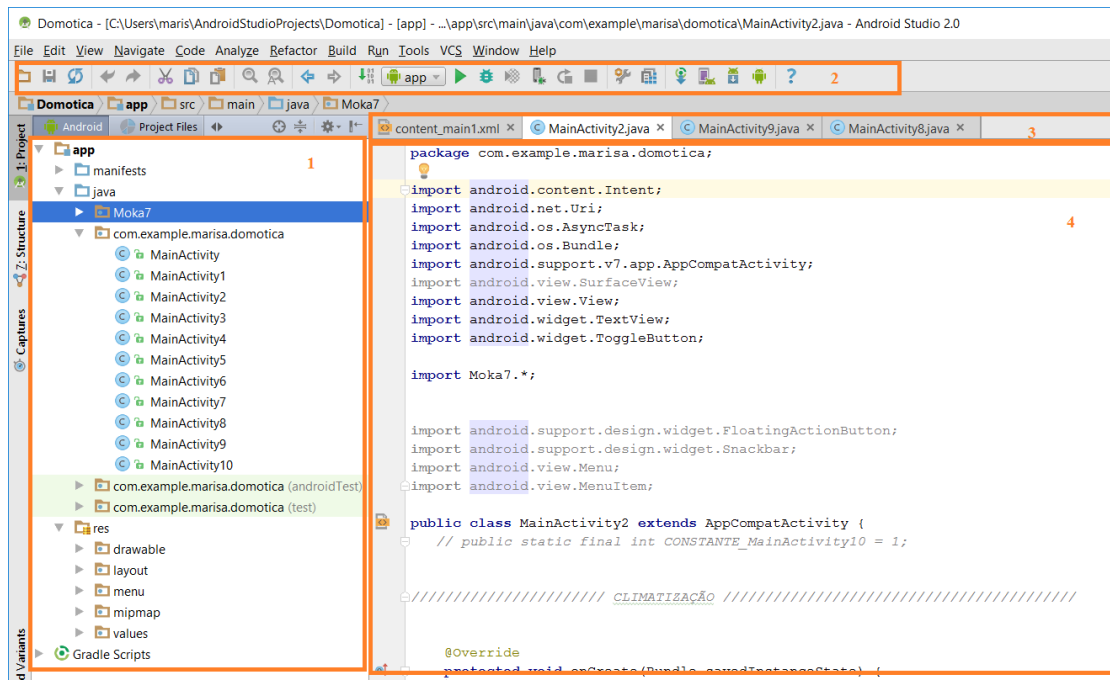


Figura 3-23: Estrutura do *Android Studio*.

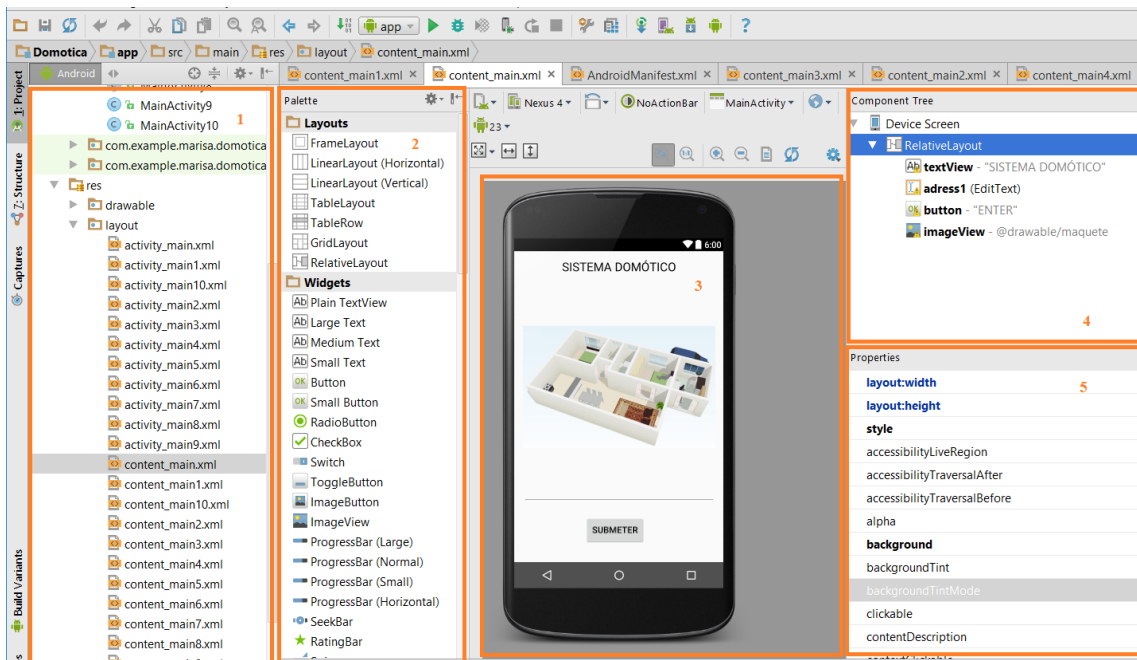
³ IntelliJ IDEA - ambiente Java de desenvolvimento integrado (IDE).

3 - Desenvolvimento e comando do sistema domótico



1-Estrutura do projeto; 2-Menus e barras de ferramentas; 3-Janelas utilizadas; 4-Área de trabalho.

Figura 3-24: Estrutura do projeto.



1-Estrutura do layout; 2-Componentes; 3-Layout; 4-Componentes utilizados; 5-Propriedades.

Figura 3-25: Estrutura do layout do projeto.

3.3.3 Estrutura de um Projeto *Android*

Um projeto *android* tem elementos-chave essenciais para a criação de qualquer projeto e/ou aplicação e são eles,

- Java;
- *Resources*;
- *AndroidManifest.xml*;
- *Intent*.

Java

No elemento Java encontram-se as classes programadas pelo utilizador ou classes que se podem importar para complementar o projeto (Figura 3-26). As classes são escritas e/ou programadas em Java⁴.

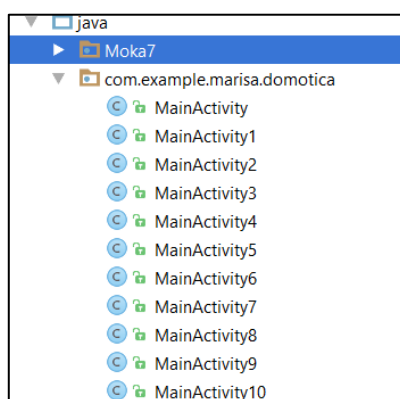


Figura 3-26: *Java Class*.

Resources

Resources são os recursos utilizados na aplicação, exceto o código Java, ou seja, assume-se como um complemento do código Java. Na plataforma *Android*, existem muitos tipos de *resources*, como, texto, cor, layout, dimensão, entre outros. A pasta raiz de *resources* é a pasta *res* e esta pasta contém muitas sub-pastas para cada tipo de *resource*, como se verifica na Figura 3-27.

⁴ Java - Linguagem de programação interpretada e orientada a objetos.

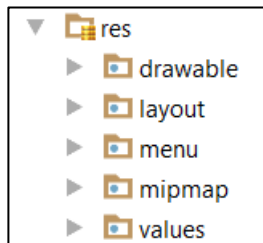
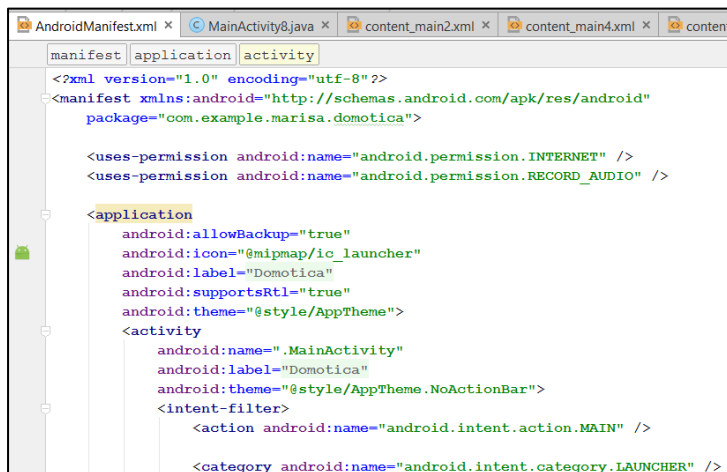


Figura 3-27: Conteúdo da pasta res.

- *Drawable* – pasta que contém as imagens do projeto;
- *Layout* – pasta que contém os *layouts*.
- *Menu* – pasta que contém os itens que serão exibidos na barra de ação.
- *Values* – pasta com dimensões em XML e onde se definem os estilos, as cores, entre outras características.
-

AndroidManifest.xml

AndroidManifest é um arquivo poderoso e essencial num projeto, pois é nele que se encontra o nome da aplicação, o ícone principal, o nome do pacote, as permissões (internet, som, entre outros), as atividades, a lista de serviços e outras informações relevantes (Figura 3-28).

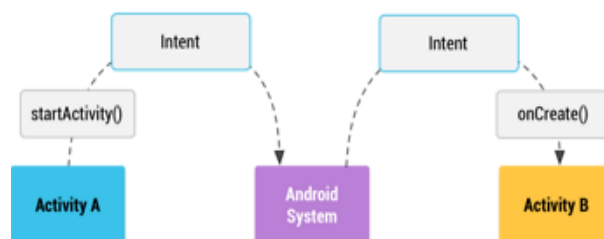


```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.marisa.domotica">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Domotica"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="Domotica"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Figura 3-28: *Android Manifest*.

Intent

Uma *intent* é um objeto de mensagens que se pode usar para solicitar uma ação de outro componente da aplicação (Figura 3-29).

Figura 3-29: *Intents*.

Facilitam a comunicação entre os componentes de várias maneiras, e existem três casos de uso fundamentais, para iniciar uma atividade, um serviço e para entregar uma transmissão.

3.3.4 Comunicação *Android* - PLC S7-1200

Inicialmente para se poder estabelecer a comunicação entre o *Android* e o PLC S7-1200 realizou-se uma pesquisa pormenorizada de bibliotecas (*LibNoDave*, *Prodave*, *Snap7*, *PLCcom*, *OPC Server*) ou conjunto de classes (classes Java do Protocolo S7) que permitissem o estabelecimento da comunicação. De entre as várias existentes, a escolha recaiu na *Snap7*, especificamente *Moka7* (*Java port of Snap7*), que apesar de não ser a mais utilizada em programação, é a que oferece a melhor documentação [43].

Moka7 é *Java Snap7* Cliente, ou seja, java puro do Protocolo de S7. *Moka7* é implementado como um conjunto de classes de código fonte que se podem importar para comunicar com PLCs S7, no entanto nem todas as funções são suportadas. É projetada para trabalhar com pequeno hardware baseado em Java, telefones *android* ou até mesmo para grandes projetos que não necessitem de realizar funções de controlo prolongado [44].

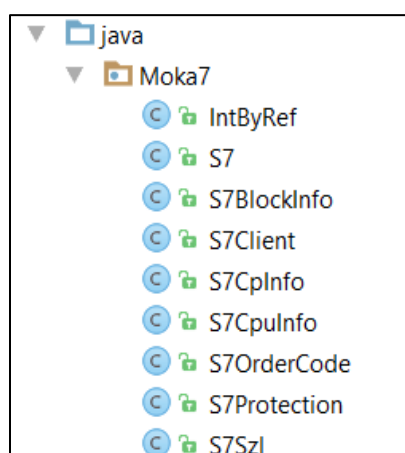


Figura 3-30: Conteúdo Moka 7.

3.3.4.1 Comunicação interna e externa

Para se estabelecer uma comunicação foi necessário realizar um conjunto de passos, ou seja, para além da importação da biblioteca referida anteriormente no *Android Studio* foi necessária uma permissão de internet no *AndroidManifest* (Secção de código 3-25). Posteriormente foi necessário conectar o PLC a um router para este lhe atribuir um IP, sendo este IP colocado nas propriedades do CPU no TIA Portal.

```
<uses-permission android:name="android.permission.INTERNET" />
```

Secção de código 3-25: Permissão de internet.

Desde cedo, um dos pontos cruciais no desenvolvimento da aplicação *android* foi a possibilidade de introdução de qualquer endereço IP e a possibilidade de comunicação a partir de qualquer parte do mundo (Figura 3-31).



Figura 3-31: *Layout* da primeira página da aplicação desenvolvida.

Um endereço IP é um valor numérico exclusivo que é utilizado para identificar um equipamento numa rede. Existem dois tipos de endereços IP: privados e públicos.

Os endereços IP privados são endereços reservados para redes privadas, não acessíveis na Internet. Geralmente o IP da placa de rede de um computador, uma impressora e o endereço IP

público é um endereço encaminhável na Internet, permitindo assim que as nossas redes (ou o nosso router que faz a fronteira entre a nossa rede e a Internet) estejam acessíveis publicamente através da Internet. No entanto, tem de se ter em conta que o endereço IP público por questões de segurança muda de X em X tempo e para resolver este problema a solução passa por se optar pelo serviço *dynIP* (Figura 3-32), que é semelhante ao popular *no-ip*, que permite mapear o endereço público num nome, ou seja, é um serviço que permite traduzir o endereço IP público (por exemplo: 82.111.181.25) por um nome do tipo “name.dynip.sapo.pt”. Este nome por sua vez nunca muda.



Figura 3-32: Serviço *dynIP*.

Tecnicamente, o objetivo era poder aceder a um equipamento, que neste caso é o PLC a partir da rede interna e a partir da Internet. Em relação à rede interna não existiu qualquer consternação, uma vez que, quando o PLC se conecta ao router, este atribui-lhe um IP interno e facilmente se estabelece a comunicação. Aceder à rede interna a partir da Internet foi um pouco mais trabalhoso, na medida em que, o IP público dá acesso ao router e não ao equipamento em questão, assim tornou-se necessário redirecionar os pedidos que chegam ao router para a porta 80 (HTTP), onde se colocou virtualmente o PLC. Ao processo em causa dá-se o nome de *port forwarding*. No entanto, cedo se descobriu que a porta 80 era utilizada pelos serviços da operadora. A alternativa ao problema passou pela colocação do IP privado do PLC na DMZ (permite mapear todos os serviços de um dos dispositivos de rede local para o IP público) (Figura 3-33).



Figura 3-33: Endereço do PLC na DMZ.

Contudo existiam outras formas de aceder ao equipamento pela Internet, nomeadamente através de VPNs ou DynDNS.com⁵.

3.3.5 Aplicação *Android*

Antes mesmo de se iniciar a explicação mais detalhada da aplicação desenvolvida, torna-se pertinente mencionar que a comunicação é realizada através de DB. Como anteriormente citado, criou-se um *Data Block* no TIA Portal intitulado de “Controlo_Remoto” com as várias variáveis a manipular (são utilizadas pelas outras potencialidades do sistema por forma a que estas entrem em concordância) e alteraram-se as suas propriedades, ou seja, retiraram-se os atributos seleccionados, de modo a tornarem-se acessíveis remotamente (Figura 3-34).

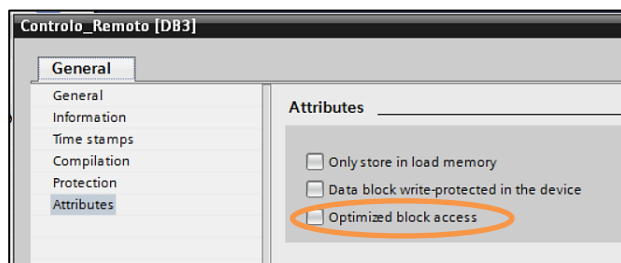


Figura 3-34: Propriedades do *Data Block*.

Na programação da aplicação *android* umas das primeiras coisas a ter em conta é a existência de uma ligação lógica entre as funções Java criadas e os *layouts* correspondentes (anexo 1). Especificamente, criam-se funções Java para depois serem “chamadas” no respetivo *layout* por *widgets* (Figura 3-35) ou por imagens do *drawable*. Ou de outra forma, introduz-se informação no *layout* para ser posteriormente processada.

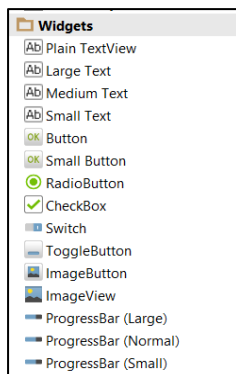


Figura 3-35: *Widgets*.

⁵ DynDNS.com - produto comercial.

3.3.5.1 Programação

As funções são programadas na pasta Java e especificamente em ficheiros Java Class. Normalmente quando se inicia a programação, existe um ficheiro Java Class criado por defeito, *Main Activity*. Como é evidente foram criados vários ficheiros Java Class, devido às várias funcionalidades implementadas. A primeira acção realizada em ficheiros Java Class foi a importação de tudo o que se utiliza, nomeadamente bibliotecas, *widgets*, entre outros, como se verifica na Secção de código 3-26.

```
package com.example.marisa.domotica;

import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.SurfaceView;
import android.view.View;
import android.widget.TextView;
import android.widget.ToggleButton;

import Moka7.*;
```

Secção de código 3-26: Importação de bibliotecas, *widgets*, entre outros.

Seguidamente e das coisas mais importantes e específicas foi a colocação das coordenadas para a leitura do PLC.

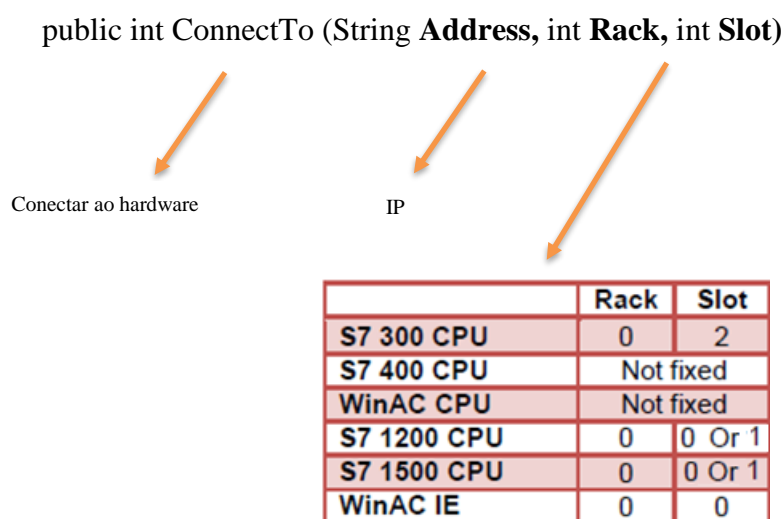


Figura 3-36: Identificação do PLC.

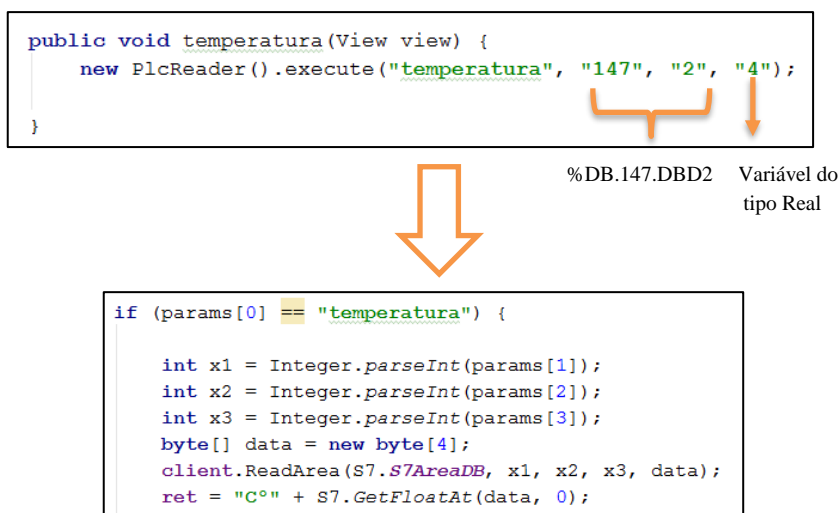
A *String Address* (endereço IP) é colocada manualmente pelo utilizador na primeira página da aplicação, deste modo é necessário programar e especificar o envio dessa informação pelas várias *Java Class* criadas.

Os problemas que possam surgir na aplicação em relação às coordenadas atribuídas ou até mesmo por outra razão qualquer, são codificados numericamente e apresentadas no topo de cada página. O número do erro é decifrado numa tabela existente no manual da biblioteca importada.

3.3.5.1.1 Leitura e Escrita de variáveis

Para a leitura de variáveis foi necessário criar funções e introduzir o endereço correspondente no DB [45].

- As variáveis do tipo real diferenciam-se de todas as outras por lhes ser atribuído o valor numérico 4⁶. No projeto, a leitura da temperatura, é exemplo de uma variável real, sendo assim, são verificadas as coordenadas estabelecidas. Se estas coordenadas estão corretas, realiza-se uma leitura e visualiza-se uma resposta ao pedido.



Secção de código 3-27: Leitura da temperatura.

- As variáveis do tipo booleana são diferenciáveis por lhes ser atribuído o valor 1 e pela resposta ao pedido, ser referente a uma variável booleana, *S7.GetBitAt* (Secção de código 3-28) [44] [45].

⁶ Descrito no manual “Moka 7” como o número atribuído a variáveis do tipo real.

```

ret = "Aquecimento: " + S7.GetBitAt(data, 0, 0);
}

```

Secção de código 3-28: Resposta para variáveis booleanas.

Para a escrita de variáveis foi necessário realizar umas pequenas alterações, tal como se verifica na Secção de código 3-29 [44] [45].

```

byte[] dataWrite = new byte[1];
S7.SetBitAt(dataWrite, 0, 0, x);
client.WriteArea(S7.S7AreaDB, x1, x2, x3, dataWrite);
}

```

Secção de código 3-29: Escrita de variáveis.

Após a escrita de funções para a realização da leitura e escrita, o passo seguinte diz respeito ao *layout*. Assim, colocaram-se imagens exteriores e *widgets* intuitivos e associou-se nas propriedades de cada função criada. Assim sempre que se executa um “onClick” é processada a função, como se observa no exemplo da Figura 3-37. As respostas aos pedidos de leitura são visualizadas no topo da página.

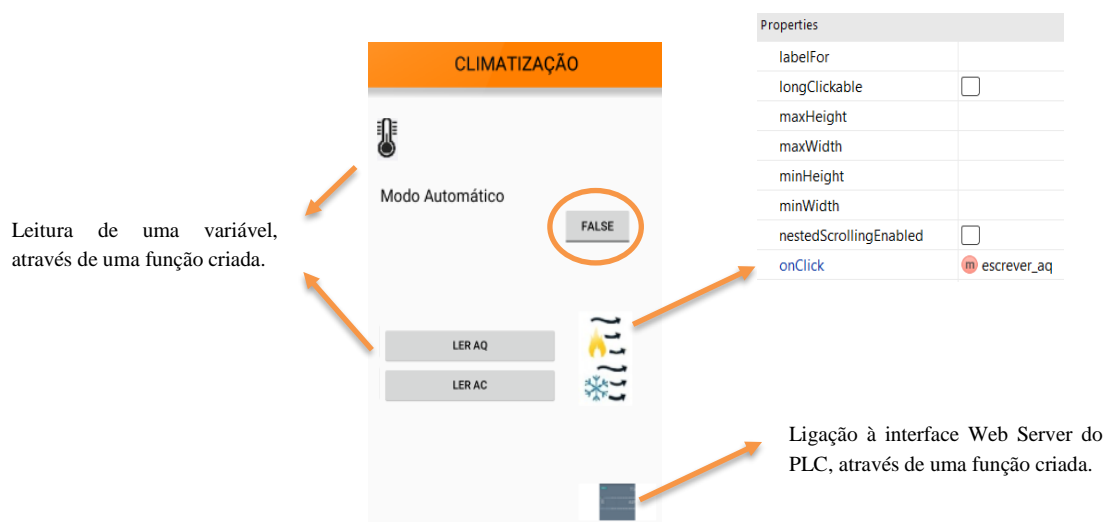


Figura 3-37: Layout e os seus constituintes.

A função na Figura 3-37 “escrever_aq” permite escrever dados, contudo, também é realizada uma leitura. Assim se a variável em causa for *true*, escreve *false* e vice-versa (Secção de código 3-30).

```
if (params[0] == "escrever_aq") {
    byte[] data = new byte[1];
    int x1 = Integer.parseInt(params[1]);
    int x2 = Integer.parseInt(params[2]);
    int x3 = Integer.parseInt(params[3]);
    client.ReadArea(S7.S7AreaDB, x1, x2, x3, data);
    boolean var = S7.GetBitAt(data, 0, 0);

    boolean x;

    if (var == true)
        x = false;

    else
        x = true;

    byte[] dataWrite = new byte[1];
    S7.SetBitAt(dataWrite, 0, 0, x);
    client.WriteArea(S7.S7AreaDB, x1, x2, x3, dataWrite);
}
```

Secção de código 3-30: Leitura e escrita de variáveis.

Ainda na Figura 3-37 é possível verificar um *toggle button*, referente ao modo automático. A função que lhe é associada é chamada e verificada automaticamente cada vez que se acede à página. O *toggle button* muda consoante o seu estado, sempre que a variável for *true* ele será seleccionado, e vice-versa (Secção de código 3-31).

```
//////////////////////////////////LER E ESCRIVER Modo Auto//////////////////////////////////
if (params[0] == "modo_aut_ler") {

    int x1 = Integer.parseInt(params[1]);
    int x2 = Integer.parseInt(params[2]);
    int x3 = Integer.parseInt(params[3]);

    byte[] data = new byte[1];
    client.ReadArea(S7.S7AreaDB, x1, x2, x3, data);

    ////////////////////////////////////Toggle Button (estado)//////////////////////////////////

    if (S7.GetBitAt(data, 0, 0) == true) {
        result_modos_auto = "Modo auto: true";
    } else
        result_modos_auto = "Modo auto: false";
}

if (params[0] == "modo_aut_l") {

    byte[] data = new byte[1];
    int x1 = Integer.parseInt(params[1]);
    int x2 = Integer.parseInt(params[2]);
    int x3 = Integer.parseInt(params[3]);
    client.ReadArea(S7.S7AreaDB, x1, x2, x3, data);
    boolean var = S7.GetBitAt(data, 0, 0);

    boolean x;

    if (var == true) {
        x = false;
        result_modos_auto = "Modo auto: false";
    } else {
        x = true;
        result_modos_auto = "Modo auto: true";
    }

    byte[] dataWrite = new byte[1];
    S7.SetBitAt(dataWrite, 0, 0, x);
    client.WriteArea(S7.S7AreaDB, x1, x2, x3, dataWrite);
}
```

Secção de código 3-31: Verificação automática do estado para agir em conformidade.

Após todo o desenvolvimento da aplicação, o passo seguinte consistiu em conectar via USB o telemóvel para descarregar a aplicação. Na maioria das vezes é necessário realizar o *download* do *driver* referente à marca do telemóvel.

Por fim, a potencialidade anteriormente descrita é muito interessante, muito requisitada e com possibilidade de constantes mudanças e inovações.

3.4 Página WEB

Uma outra potencialidade de comunicação e monitorização criada foi a página web interna do PLC (anexo 2). Previamente torna-se importante salientar determinados aspetos e conceitos imprescindíveis de um documento ou página HTML (formato utilizado na criação das páginas web deste sistema).

3.4.1 Noção do documento HTML

Uma página HTML (*Hyper Text Markup Language*) é um simples ficheiro de texto que por convenção, tem uma extensão .htm ou .html, mas pode potencialmente ter uma qualquer extensão [12] [46].

Uma página web pode ser construída a partir dos editores de texto básicos, por exemplo, blocos-notas, *notepad++*, entre outros. Contudo existem editores sofisticados para o efeito, como, por exemplo:

- Dreamweaver;
- Macromedia;
- HomeSite 5.5 – incluído no Dreamweaver;
- Microsoft Expression Web;
- Microsoft Frontpage;
- HTML-kit – free.

Tipicamente existem três linguagens básicas para criar uma página web: HTML, CSS e *JavaScript*. O HTML (linguagem de marcação) é a linguagem que exhibe a informação e lhe dá significado [47]. O CSS é a linguagem que vai melhorar a apresentação da informação, tipo de letra, tamanho, cores, margens, altura, largura, entre outros aspetos. Existem várias maneiras de a integrar num ficheiro HTML [47]:

- dentro de um elemento HTML;
- entre o `<script>` e `</script>`;
- num ficheiro CSS externo.

Tabela 3-3: Propriedades CSS.

CSS property	Function	Examples for values	CSS property	Function	Examples for values
position	Position type	static, relative, absolute, fixed	color	Text color	rgb(51,102,170), #FFFFFF
top	Start position from top	10px, 2%	vertical-align	Vertical alignment	top, middle, bottom
left	Start position from left		text-align	Horizontal alignment	left, center, right, justify
bottom	Start position from bottom		margin	Margin/Distance general Margin/distance top Margin/distance right Margin/distance bottom Margin/distance left	10px, 5%
right	Start position from right		margin-top		
width	Width	margin-right			
height	Height	margin-bottom			
direction	Direction	ltr, rtl	margin-left	Margin/distance left	
z-index	Layer position for overlapping	1, 2	padding	Padding general	10px, 5%
font-family	Font	Arial, Helvetica	padding-top	Padding top	
font-style	Font style	italic, oblique, normal	padding-right	Padding right	
font-size	Font size	20px, 100%, small, medium, large	padding-bottom	Padding bottom	
font-weight	Font weight	bold, normal, bolder, lighter, 100 to 900	padding-left	Padding left	
text-decoration	Text decoration	underline, blink, none	border[-top, -right, -bottom, -left]	Border general	2px solid white
text-transform	Text transformation	uppercase, lowercase	border[-top, -right, -bottom, -left]	Border thickness	2px, 1%, thin, medium, thick
			border[-top, -right, -bottom, -left]	Border color	#FFFFFF00, white
			border[-top, -right, -bottom, -left]	Border type	none, hidden, dotted, solid, dashed, double
			border-collapse	Border model	separate, collapse

O *JavaScript* é a linguagem que vai fazer com que a informação receba alguns comportamentos, como, por exemplo, criar um submenu ou controlar algo que aparece e desaparece na página. Existem várias maneiras de integrar comandos *JavaScript* num ficheiro HTML:

- entre o `<script>` e `</script>`;
- por referências;
- como parâmetro de uma *tag* HTML;
- num ficheiro JS externo.

Num documento HTML, para uma melhor perceção e interpretação do seu conteúdo é importante distinguir dois conceitos, *tag* e elemento. As *tags* são o conjunto de caracteres que identificam um elemento, ou seja, quando nos referenciamos à *tag* "p" estamos falando em: `<p>`. Os elementos são formados a partir de *tags* e é entre as *tags* que se encontra o conteúdo do elemento.

Tabela 3-4: Tags.

Tag	Descrição
<code><html></code>	Define um documento HTML
<code><body></code>	Define o corpo do documento
<code><h1></code> a <code><h6></code>	Define cabeçalho 1 ao 6
<code><p></code>	Define um parágrafo
<code>
</code>	Insera uma quebra de linha
<code><hr></code>	Define uma linha horizontal
<code><!--></code>	Define um comentário
<code></code>	Define a formatação de um texto

3.4.2 Servidor Web

Para a criação das várias páginas web, foi necessário ativar a função servidor Web (*Web server*) do PLC que se encontra na página de configuração de dispositivos no TIA Portal, como ilustra a Figura 3-38 [48].

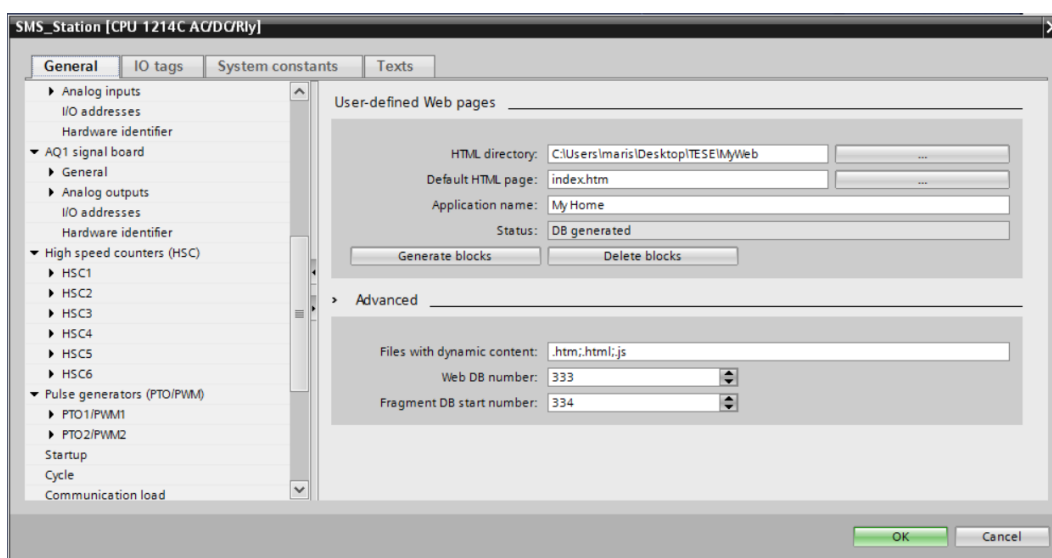


Figura 3-38: Propriedades do Servidor *Web* do PLC.

Após a compilação do programa, o PLC cria automaticamente o ficheiro referente à página inicial do utilizador e atribuiu-lhe o nome do projeto. No entanto, para uma melhor perceção alterou-se o nome para “index”, uma vez que, é a página padrão nos servidores de websites e é nela que se concentra muita informação relevante.

De seguida criaram-se diversas páginas Web de monitorização e controlo do sistema utilizando o editor *notepad++*. Este programa possante e gratuito admite vários tipos de formatos e linguagens de programação, incluindo HTML. No entanto, é necessário que todas as páginas produzidas sejam gravadas numa mesma pasta (Figura 3-39).

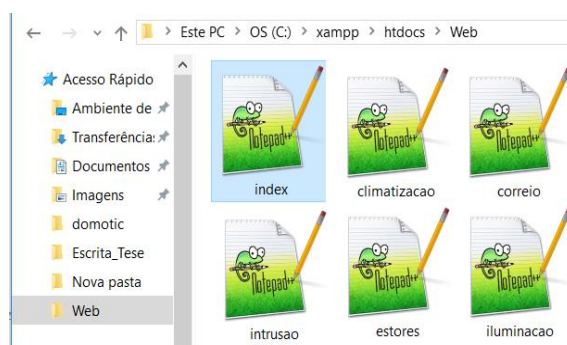


Figura 3-39: Pasta com as várias páginas criadas.

3.4.3 Programação

Na programação de cada página é fundamental que estas sigam uma determinada estrutura composta basicamente por três instruções ou *tags*. Estas devem estar sempre presentes e devem seguir exatamente a seguinte sequência (Secção de código 3-32) [49]:

- **Definição do documento HTML (*doctype*):** Elemento indispensável e deve ser sempre a primeira linha do documento, com o seguinte código: `<!DOCTYPE html>`.
O *doctype* não é uma *tag* HTML, mas uma instrução e é utilizada pelo navegador para identificar que o documento em questão se trata de um HTML e não de texto puro. É uma declaração do tipo de documento, permitindo desde logo, perceber do que se trata. De seguida devem existir as *tags* `<html>` e `</html>` e estas devem conter todas as outras *tags*, as informações e todo o conteúdo.
- **Head:** É uma *tag* utilizada para inserir outras *tags*. Entre `<head>` e `</head>` encontram-se os dados e informações que não são exibidos na página do navegador, mas utilizados por este para os mais diversos fins, como dados implícitos, de uso e controlo do documento, vinculação com outros ficheiros, aplicação de lógica de programação de *scripts*, etc.
- **Body:** É uma *tag* representativa do corpo do documento. Entre `<body>` e `</body>` está o que vai, efetivamente, ser exibido na página do navegador. Dentro do *body* encontram-se os elementos semânticos da construção de uma página.

```
<!DOCTYPE html>
<html lang="esp">
  <head>
    <title> DOM&Oacute;TICA </title>
    <meta charset="utf-8">
  </head>
  <body>
    C&Oacute;DIGO HTML
  </body>
</html>
```

Secção de código 3-32: Estrutura do documento HTML.

Inicialmente na página index.htm incluíram-se os ficheiros estáveis *jQuery* e bibliotecas *JavaScript*. Normalmente realiza-se o *download* dos ficheiros estáveis⁷, no entanto, por falta de memória do PLC, incluiu-se apenas o *link*.

Foi também necessário fazer a interligação entre as variáveis do PLC e as variáveis utilizadas na leitura e escrita de valores através das páginas Web com recurso à seguinte Secção de código[50]:

```
<!-- AWP_In_Variable Name= '"Controlo_Remoto".Aquecimento.Ligar_aq'-->
```

Nome da DB

Nome da variável

Secção de código 3-33: Inserção da variável do PLC no documento.

Apesar de estar em comentário o servidor Web do PLC consegue decifrar e processar essas variáveis.

Todas as páginas criadas são carregadas na página index.htm e a interligação entre elas realiza-se com recurso a um menu (Secção de código 3-34). Primeiramente é carregada e ativada a página climatizacao.htm e quando se navega para outras páginas essa ativação vai sendo alterada com o auxílio do *JavaScript*. Assim, é acompanhado todo o trajeto de navegação do utilizador. Um ponto interessante foi a utilização de um cursor por forma a estruturar e tornar perceptível todo o trajeto.

```
<div class='menu' style="cursor:pointer;">
  <ul>
    <li><a id='climatizacao' class="active" src='climatizacao.htm' >Climatiza&ccedil;&atildeo </a></li>
    <li><a id="iluminacao" src='iluminacao.htm' > Ilumina&ccedil;&atildeo</a></li>
    <li><a id="estores" src='estores.htm' >Estores </a></li>
    <li><a id="estores" src='estorespainel.htm' >EstoresPainel </a></li>
    <li><a id="rega" src='rega.htm' >Sistema de rega </a></li>
    <li><a id="portao" src='portao.htm' >Port&atildeo autom&aacutetico </a></li>
    <li><a id="intrusao" src='intrusao.htm' >Intrus&atildeo </a></li>
    <li><a id="presenca" src='simulacao.htm' >Simula&ccedil;&atildeo de presen&ccedila </a></li>
    <li><a id="correio" src='correio.htm' >Correio </a></li>
    <li><a id="sobre" src='sobre.htm' >Sobre </a></li>
  </ul>
</div>
```

Secção de código 3-34: Menu.

Na implementação das várias páginas recorreu-se às mais diversas instruções e comandos HTML. No entanto, existem instruções e comandos próprios à interligação com o PLC, como,

⁷ Ficheiros estáveis – São ficheiros do programa correspondentes a funções desenvolvidas e testadas, normalmente por outras entidades.

por exemplo, a Secção de código 3-35, que permite ler o valor da variável referente à temperatura.

```
<p><b> Temperatura atual ::= "Tags".DataLog.DATA.Temperature: </p></b>
```

Secção de código 3-35: Leitura da variável.

A Secção de código 3-36 permite alterar a imagem de acordo com o valor da variável. São guardadas (na mesma pasta das páginas web) duas imagens, por exemplo, uma lâmpada ligada com o nome de “lamp01” e uma lâmpada desligada com o nome de “lamp00”. Assim sempre que a variável toma o valor=1 surge a imagem da lâmpada ligada. Quando toma o valor=0 surge a imagem da lâmpada desligada.

```

```

Secção de código 3- 36: *Gifs*.

A Secção de código 3-37 explica a escrita de valores em variáveis do PLC. Corresponde a um conjunto de *radio buttons* que, quando ativados, vão escrever na variável do tipo real os valores correspondentes.

```
<form>  
<input name="Controlo_Remoto".Estores_3.setPoint' type="radio" value="0.25"> 25%<br>  
<input name="Controlo_Remoto".Estores_3.setPoint' type="radio" value="0.5"> 50%<br>  
<input name="Controlo_Remoto".Estores_3.setPoint' type="radio" value="0.75"> 75%<br>  
<p>  
<button type="submit">OK</button>  
</p>  
</form>
```

Secção de código 3-37: Escrever valores nas variáveis reais do PLC.

Importa salientar que, quando se navega na página web, é necessário que haja uma atualização automática. Contudo essa atualização não diz respeito à totalidade da página em si, mas sim apenas às variáveis, que na realidade é o seu valor que se vai alterando. Se a atualização fosse à página (Figura 3-40), esta estava constantemente a aparecer e desaparecer o que não é muito favorável.

```
<meta http-equiv="refresh" content="30">
```

Figura 3-40: *Refresh* da página não recomendado.

Assim, criou-se um *script* que lê as variáveis que se encontram num ficheiro denominado de “*outputs*” seguido do nome da funcionalidade e atribui-lhe o estado ON ou OFF consoante o seu valor 1 ou 0 (Secção de código 3-38). No modo automático foi criado um *toggle button* e sempre que a variável se encontra a 1, o botão toma o nome “Desligar”, quando a variável se encontra a 0 o botão toma o nome “Ligar”.

```

<script type="text/javascript">
$(document).ready(function(){
$.ajaxSetup({ cache: false });
setInterval(function() {
$.getJSON("outputsClimatizacao.htm", function(json){
// Variables
document.getElementById("p1").innerHTML = stringNaoSim(json.Aquecimento);
document.getElementById("p2").innerHTML = stringNaoSim(json.AC);
document.getElementById("p3").innerHTML = stringNaoSim(json.ModoAuto);
document.getElementById("btn").value = stringBotao(json.ModoAuto);
});
},2000);
});

function stringBotao(valor_bool) {
if (valor_bool ==0)
return "Ligar";
else
return "Desligar";
}

function stringNaoSim(valor_bool) {
if (valor_bool ==0)
return "OFF";
else
return "ON";
}

```

Secção de código 3-38: Atualização de variáveis.

Para alterar o valor das variáveis booleanas recorreu-se às instruções da Secção de código 3-39.

```

<br>
<p><b> Aquecimento: <label id="p1"> </label></b></p>
<form action="index.htm?op=climatizacao" method="post">
<p>
<input type="submit" value="On">
<input type="hidden" name="Controlo_Remoto.Aquecimento.Ligar_aq" value="1" >
</p>
</form>

<form action="index.htm?op=climatizacao" method="post">
<p>
<input type="submit" value="Off">
<input type="hidden" name="Controlo_Remoto.Aquecimento.Ligar_aq" value="0" >
</p>
</form>
<br>

```

Secção de código 3-39: Modo de alterar valores das variáveis booleanas.

Contudo sempre que se executava um *submit* de uma variável, por exemplo na página climatização, não se mantinha na mesma página, voltava sempre para o *index.htm*. A solução passou pelo uso de um *script* no *index.htm*, onde se definiu uma variável “*op*” que vai ser igual

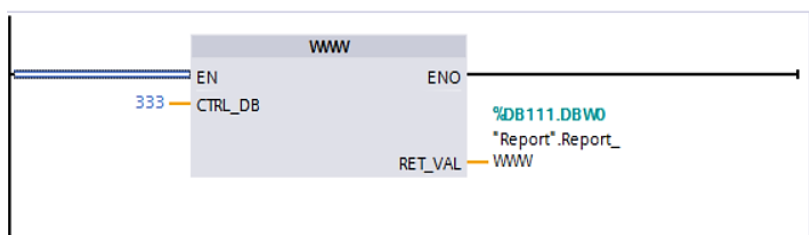
à página ativa ou carregada no momento, independentemente das ações realizadas (Secção de código 3-40).

```
var pag= getUrlParameter('op');  
if(pag==null)  
    pag='climatizacao';  
$("#section").load($("#'+pag).attr('src'));  
$(".active").removeClass("active");  
$("#'+pag).addClass('active');
```

Secção de código 3-40: Script com variável “op”

Por fim, criou-se ainda uma página “style.css” onde se encontram discriminados todos os aspetos da informação a mostrar.

Para que haja uma atualização permanente entre as páginas Web e as variáveis do PLC é necessário introduzir a instrução da Secção de código 3-41 no ciclo de processamento do PLC.



Secção de código 3-41: Instrução WWW do TIA Portal.

Na página de configuração de dispositivos no TIA Portal, após o carregamento da pasta que contém as páginas criadas escolheu-se a opção "Generate blocks" e foram gerados dois blocos (DB333 e DB334), como se observa na Figura 3-41.

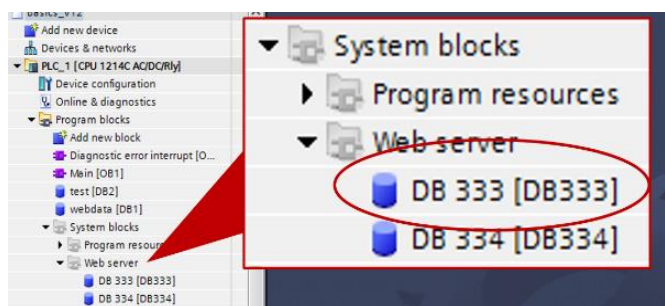


Figura 3-41: Blocos gerados a partir do código HTML desenvolvido.

Ao escolher a opção "Generate blocks", Figura 3-38, o compilador junta os ficheiros e cópias de cada byte numa matriz (Figura 3-42). A cada página HTML é atribuída uma matriz no DB334 [48].

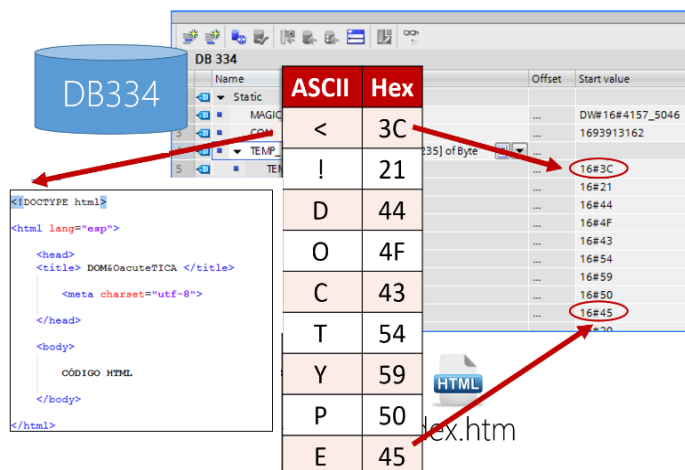


Figura 3-42: Descodificação do código.

Por razões de falta de memória no PLC, optou-se por desenvolver páginas Web relativamente simples. Caso se desenvolvessem páginas Web mais elaboradas, para além do DB 334 geravam-se outros DB (DB 335, DB 336,...) que necessariamente iriam ocupar mais memória no PLC. De uma forma resumida, pode-se dizer que o DB 333 contém um índice para as várias páginas Web. O DB 334 (e seguintes, caso necessário) armazena o conteúdo das páginas Web.

3.4.4 Acesso às páginas criadas

O acesso às páginas criadas requer que o utilizador esteja identificado como administrador para conseguir alterar qualquer parâmetro do sistema (Figura 3-43). Caso contrário, é-lhe permitido apenas visualizar, os parâmetros [48].

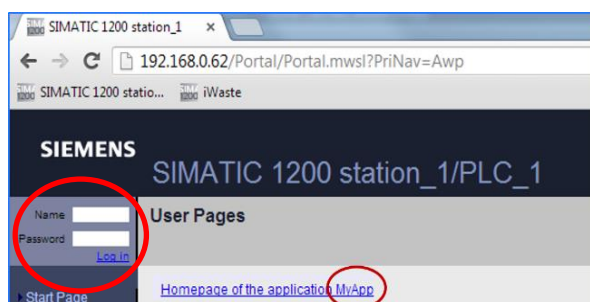


Figura 3-43: Interface servidor Web do PLC.

O utilizador ao identificar-se como administrador possibilita-lhe inúmeras vantagens, nomeadamente, o controlo e monitorização de todo o sistema à distância e possibilidade de descarregar e eliminar remotamente os *Data Logs* da memória do PLC.

3.5 Comunicação GSM

Uma outra vertente do projeto consistia em comandar à distância, através de SMS, as funcionalidades implementadas no TIA Portal. Para o efeito foi necessário conectar (*hardware*) e associar (*software*) ao PLC S7-1200 um módulo GSM CP 1242-7 (Figura 3-44).

O Módulo GSM CP1242-7 é um processador de comunicações para a transmissão de dados através de GPRS e destina-se essencialmente ao uso em ambientes industriais (exigentes) e apresenta as seguintes características [51]:

- Pequeno e compacto;
- Suporta cartões de diversos operadores;
- Apresenta diversos LED de sinalização;
- Possibilita um rápido comissionamento;
- Proporciona a troca de dados sem fio entre S7-1200 e/ou centros de controlo com ligação à Internet;
- Apresenta comunicação com base no GPRS (*General Packet Radio Service*);
- Envia e recebe mensagens de texto (SMS);
- GPRS operação com endereços IP fixos com contrato de M2M, bem como endereços IP dinâmicos (contrato de telefone móvel);
- Sincronização tempo real com base em NTP (*Network Time Protocol*);
- Funcionalidade *wake-up* por chamada ou SMS para ativar o módulo de comunicação se estiver em "modo de espera".

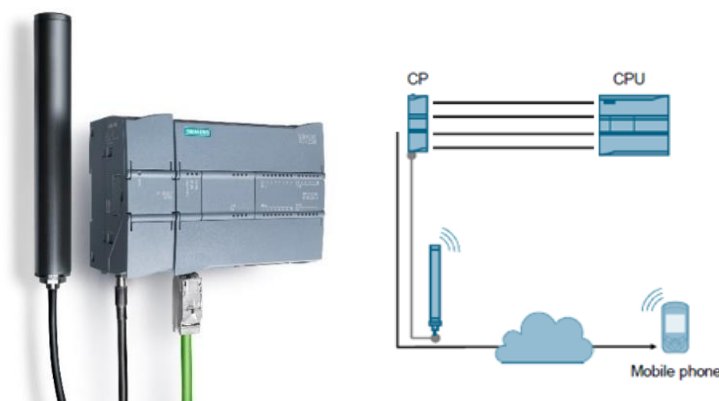


Figura 3-44: PLC e o módulo GSM.

No TIA Portal e mais precisamente no módulo associado ao PLC é necessário definir determinadas propriedades (Figura 3-45) que são cruciais no estabelecimento da comunicação, tais como:

- Modo de operação: *Telecontrol* (troca de dados com um servidor sendo necessária uma ligação à internet) ou *GPRS direct* (comunicação direta entre as estações remotas via rede GSM. Nenhum servidor de telecontrol é necessário);
- Número do cartão SIM inserido, o PIN e SMSC;
- Números autorizados;
- APN.

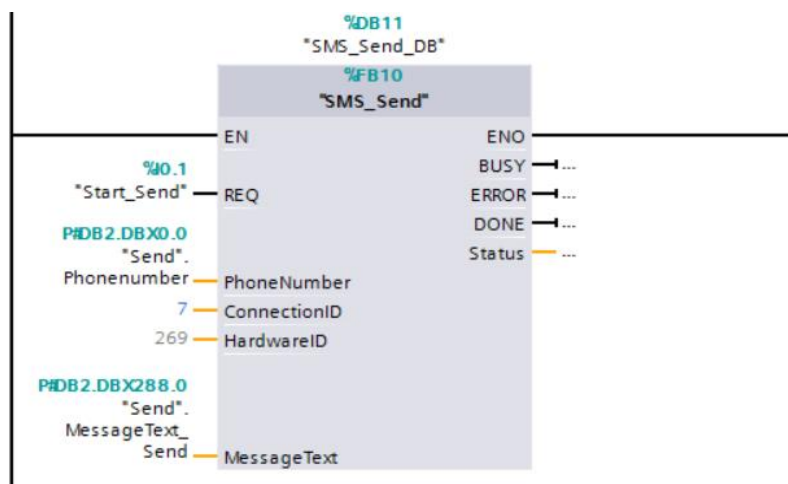
Figura 3-45: Propriedades do módulo GSM.

Estando todas as propriedades definidas, o próximo passo implica programação no TIA Portal. Para a comunicação na rede GSM existem determinados blocos no TIA Portal, que formam a base para a comunicação [51] ,

- TC_CON - Conectar;
- TC_SEND - Enviar dados;
- TC_RECV - Receber dados;
- TC_DISCON – Desconectar/Desligar.
-

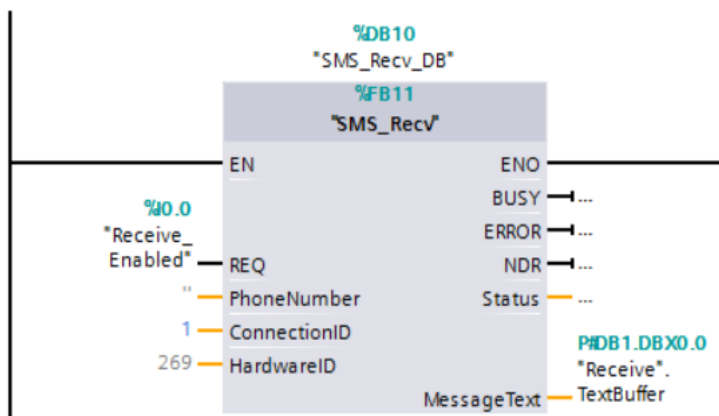
Os blocos TC acima listados são chamados por dois blocos criados,

- SMS_Send - responsável por enviar mensagens texto em *loop* para um determinado número de telemóvel (Secção de código 3-42).



Secção de código 3-42: *SMS_Send*

- SMS_Recv - responsável por receber as mensagens armazenando-as numa área de dados (Secção de código 3-43).



Secção de código 3-43: *SMS_Recv*

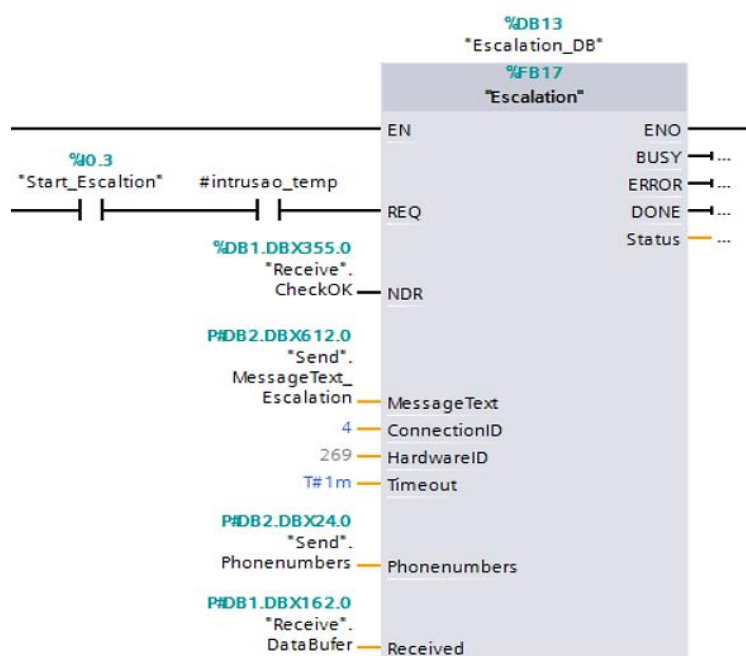
Criaram-se outros blocos mais específicos que chamam as instâncias de blocos anteriormente citados e acedem aos seus *buffers*, como é o caso dos seguintes,

- SMS_Broadcast - responsável por enviar as mensagens de texto (uma SMS em cada caso, evitando o envio em *loop*) para qualquer número de telemóvel existente no *buffer*. Este bloco chama o bloco TC “SMS_Send” e é muitas vezes chamado para enviar as mensagens descritas em cada funcionalidade [51].

- Escalation - responsável por enviar as mensagens de texto para vários números de telemóvel que se encontram no *buffer*. No entanto, o envio é realizado em tempos de processamento diferentes, após o envio para o primeiro número e após um determinado tempo, o próximo número é contactado se, e só se, assim o justifique [51]. O envio deste tipo de mensagens/alarmes por parte do PLC segue um determinado formato com campos separados pelo carácter “#”, como por exemplo:

#Q#+351961234567#pass#Alarm: INTRUSÃO

Este bloco é chamado para enviar as mensagens descritas na funcionalidade sistema de intrusão (Secção de código 3-44).



Secção de código 3-44: *Escalation*

Também foram criadas funções em SCL para a verificação de cada SMS e/ou *string* recebida no PLC, assim como para definir especificações de cada SMS:

- CheckString – Verifica os caracteres válidos do número de telemóvel;
- CheckMessage – Verifica o formato específicos de cada SMS e retorna comandos (Figura 3-46);

#	Command	#	Phone number	#	Password	#	Data
	Char		String[22]		String[16]		String

- 0 –SMS ok;
- 1 – Número errado de #;
- 2 – Nenhum comando entre o primeiro e o segundo #;
- 3 – O número de telemóvel contém caracteres errados;
- 4 – Senha errada.

Figura 3-46: Formato das SMS e os comandos respetivos.

- Set / Get Comands – Esta função contém todos os comandos configurados livremente por cada utilizador. Quando é enviada uma SMS, é procurada a parte da mensagem com o comando executado-o, assim como um texto de resposta consoante o resultado (anexo 3).

3.5.1 Set/Get

As SMS enviadas pelo utilizador e recebidas pelo PLC apresentam um formato específico para o processamento, sobretudo, o tipo de comando S(*Set*) ou G(*Get*), o número de telemóvel de quem envia, a *pass* e a instrução.

No entanto, para o utilizador torna-se muitas vezes, fastidioso escrever todo o formato ou até pode mesmo esquecer-se. Como alternativa, pode-se gravar parte do formato no modo mensagens rápidas do telemóvel ou utilizar aplicações *android* próprias para o envio de SMS. No entanto, caso o utilizador se esqueça de algum comando, pode enviar uma mensagem de ajuda.

- *#G#+351961234567#pass#Help?* – Formato de mensagem a enviar para o PLC com o intuito de relembrar todos os comandos criados.
- *#G#+351961234567#pass#AQ?* – Formato de mensagem a enviar para o PLC a questionar o estado do Aquecimento. Se a variável “Aquecimento” estiver ativa ele envia a uma resposta “AQ: TRUE”, caso contrário envia “AQ: FALSE” (Secção de código 3-45).

```

// Exemplo para um valor BOOL - Aquecimento (CLIMATIZACAO)
IF FIND(IN1 := #Data, IN2 := 'AQ?') <> 0 THEN
  #Found := True;
  #Msg := CONCAT(IN1 := #Msg, IN2 := 'AQ:');

  IF ("Aquecimento") THEN
    #Msg := CONCAT(IN1 := #Msg, IN2 := 'TRUE');
  ELSE
    #Msg := CONCAT(IN1 := #Msg, IN2 := 'FALSE');
  END_IF;
END_IF;

```

Secção de código 3-45: Descodificação de mensagens recebidas pelo PLC.

- *#G#+351961234567#pass#Temp?* – Formato de mensagem a enviar para o PLC a questionar a temperatura atual na habitação, e o PLC responde, por exemplo, “Temp: 25°C” (Secção do código 3-46).

```

// Example for Real Value - Temperatura
IF FIND(IN1 := #Data, IN2 := 'Temp?') <> 0 THEN
  #Found := True;
  #Msg := CONCAT(IN1 := #Msg, IN2 := 'Temp:');
  #Msg := CONCAT(IN1 := #Msg, IN2 := REAL_TO_STRING("Tags".DataLog.DATA.Temperature));
END_IF;

```

Secção de código 3-46: Descodificação de mensagens recebidas pelo PLC.

- *#S#+351961234567#pass#AQ=ON/OFF* – Liga ou desliga o aquecimento e retorna de seguida uma mensagem de confirmação de estado (Secção de código 3-47).

```

// Exemplo para um valor BOOL - Aquecimento (CLIMATIZACAO)
IF FIND(IN1 := #Data, IN2 := 'AQ=ON') <> 0 THEN
  #Found := True;
  "Controlo_Remoto".Aquecimento.Ligar_aq := TRUE;
  #Msg := CONCAT(IN1 := #Msg, IN2 := 'AQ:ON');

ELSIF FIND(IN1 := #Data, IN2 := 'AQ=OFF') <> 0 THEN
  #Found := True;
  "Controlo_Remoto".Aquecimento.Ligar_aq := FALSE;
  #Msg := CONCAT(IN1 := #Msg, IN2 := 'AQ:OFF');
END_IF;

```

Secção de código 3-47: Descodificação de mensagens recebidas pelo PLC.

De notar que o envio de SMS pelo utilizador é bastante vantajoso, na medida em que não prende o utilizador a uma ligação à internet, contudo é pouco intuitiva, ao contrário da página Web.

3.6 Maquete desenvolvida

Com o intuito de simular todo o sistema e de o tornar mais realista foi desenvolvida uma maquete, onde se dispuseram pelas várias divisões os equipamentos a comandar, como se pode observar na Figura 3-47.



Figura 3-47: Maquete.

O controlador S7-1200 constitui a base de todo o sistema, acompanhado pelo módulo GSM e um módulo de entradas e saídas digitais. Para uma melhor estruturação de todas as ligações efetuadas na maquete recorreu-se a ligadores e caixas de vários tamanhos.



Figura 3-48: Equipamentos, circuitos e componentes.

Seguidamente descrevem-se as ligações físicas do PLC relativas a diversas funcionalidades desenvolvidas, exceto as ligações relativas ao estore do painel solar, uma vez que se torna uma repetição do estore e portão, embora comandado de uma forma diferente.

3.6.1 Alimentação elétrica

Embora o PLC utilizado na maquete seja alimentado a 230V AC da rede elétrica, ele dispõe de uma tensão de 24V DC para os diversos fins. Inicialmente, aquando da construção da maquete, o primeiro aspeto a ter em conta foram as tensões de alimentação dos equipamentos. Deste modo, chegou-se à conclusão que se iriam utilizar-se tensões de 5V DC, 12V DC e 24V DC. Em relação aos 24V DC não haveria quaisquer problemas pois eram fornecidos pelo PLC. Os 12V DC foram obtidos com o auxílio de um transformador (230V AC – 12V DC) (Figura 3-49) e os 5V DC foram conseguidos com o auxílio de um componente eletrónico denominado de DC-DC *buck converter* (*step down*). Este componente transforma a tensão de entrada, que neste caso em particular são os 12V DC vindos do transformador para uma tensão mais baixa, 5V DC. Os 5V DC foram regulados pelo regulador azul que se encontra representado na Figura 3-50.



Figura 3-49: Transformador (230V AC – 12V DC)



Figura 3-50: DC-DC *buck converter*

3.6.2 Massas

Relativamente às massas existentes, estas foram separadas pelas tensões correspondentes e posteriormente ligadas entre si.

3.6.3 Funcionalidades e Equipamentos

Seguidamente descrevem-se as funcionalidades implementadas em termos do *hardware* utilizado. No anexo 4 apresenta-se uma tabela indicando as entradas e saídas do PLC utilizadas e a respetiva função.

3.6.3.1 Climatização

O sistema de climatização (Figura 3-51) foi simulado com um sensor de temperatura LM35, com duas ventoinhas que simulam a circulação de ar quente e de ar frio e um interruptor. As ventoinhas foram alimentadas pela tensão disponibilizada pelo transformador (12V DC). O circuito integrado LM35 e o interruptor foram ligados diretamente aos 24V DC disponibilizados pelo PLC.



Figura 3-51: Sensor de temperatura e ventoinhas referentes à climatização.

O modo automático é simulado com um interruptor ligado à entrada I0.5 do PLC. Assim quando é ativado e a média da temperatura desce abaixo do valor imposto pelo utilizador (13°C), a ventoinha do ar quente é ativada. Quando a média da temperatura sobe acima do valor imposto pelo utilizador (27°C), a ventoinha de ar frio é ativada. Quando a temperatura se encontra entre os valores mencionados, nenhuma ventoinha é ativada. As ventoinhas de ar quente e frio ligaram-se por intermédio de dois relés de 24V (Figura 3-52), às saídas Q8.0 e Q8.1 do PLC, respetivamente.

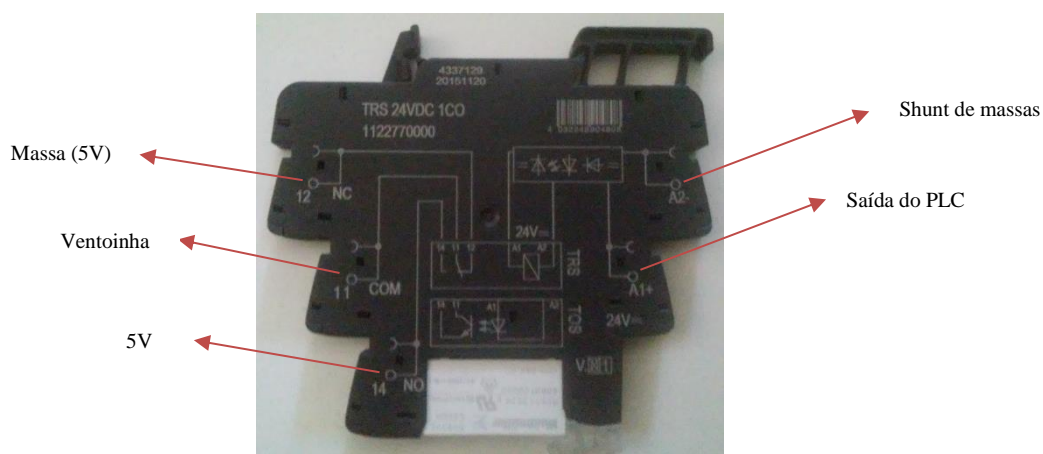


Figura 3-52: Ligações do relé.

3.6.3.2 Iluminação

A iluminação foi simulada com o auxílio de fitas led e interruptores. A escolha pela fita led deve-se ao facto de esta ter um consumo muito reduzido em relação, por exemplo, a uma lâmpada de halogéneo. A fita led é alimentada a 12V DC, no entanto as fitas foram ligadas aos 24V DC por intermédio de uma resistência de 2.2K Ω (Figura 3-53), porque no momento da sua instalação o transformador ainda não estava disponível.

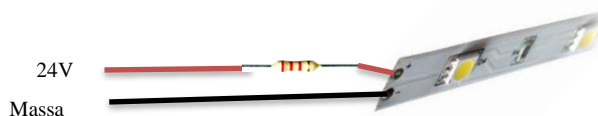


Figura 3-53: Ligação da fita led.

Por fim, dispersaram-se as fitas led e os interruptores pelas várias divisões da maquete (Figura 3-54) e realizou-se a interligação com o PLC.



Figura 3-54: Iluminação.

O interruptor correspondente à luz do wc ligou-se à entrada I1.3 e a fita led do wc ligou-se à saída Q0.3. O interruptor correspondente à luz do quarto ligou-se à entrada I1.2 e a fita led do quarto ligou-se à saída Q0.2. O interruptor correspondente à luz da sala ligou-se à entrada I1.1 e a fita led da sala ligou-se à saída Q0.1. O interruptor correspondente à luz da cozinha ligou-se à entrada I1.0 e a fita led da cozinha ligou-se à saída Q0.0.

A simulação da intensidade luminosa está disponível apenas num compartimento da maquete, sendo possível criar determinados ambientes consoante o fim a que se destinam. Para isso, foi necessário dimensionar um circuito eletrónico que permitisse variar a intensidade da fita led sem a danificar. O circuito é composto por um transístor NPN (BD647), uma resistência de $2.2k\Omega$ e, como é evidente, pela fita led. Como se observa na Figura 3-55, a base de transístor ligou-se à saída analógica do PLC (A0), conseguindo-se assim variar a intensidade luminosa sempre que se deseja (Figura 3-56).

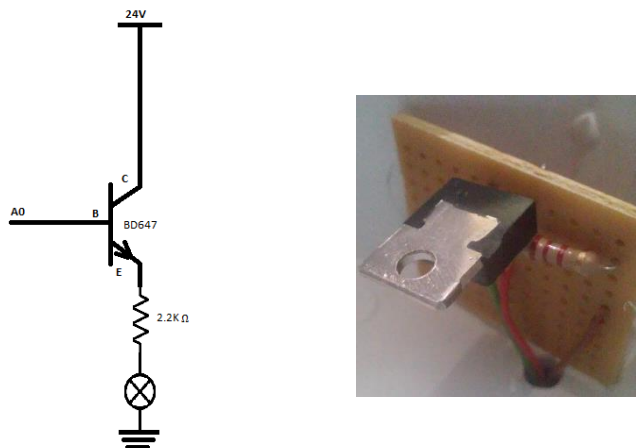


Figura 3-55: Circuito referente à variação da intensidade luminosa.



Figura 3-56: Fita led sujeita à variação da intensidade luminosa.

A iluminação com horário programado está disponível na parte exterior da maquete, mais precisamente no jardim. Como a iluminação foi simulada por uma fita led, esta ligou-se à saída Q1.0 do PLC.

3.6.3.3 Estore e Portão Automático

A simulação do estore realizou-se com o auxílio de um motor, alimentado a 5V DC. Para realizar o movimento e com o auxílio de botões de pressão, ligados aos 24V DC do PLC, para comandar os movimentos. Para se conseguir rodar o motor nos dois sentidos, subida e descida, dimensionou-se um circuito eletrónico (Figura 3-57) constituído essencialmente por um circuito integrado L293D que consiste basicamente numa ponte-H que suporta uma corrente constante de 600mA e uma corrente de pico de 1.2A [52].

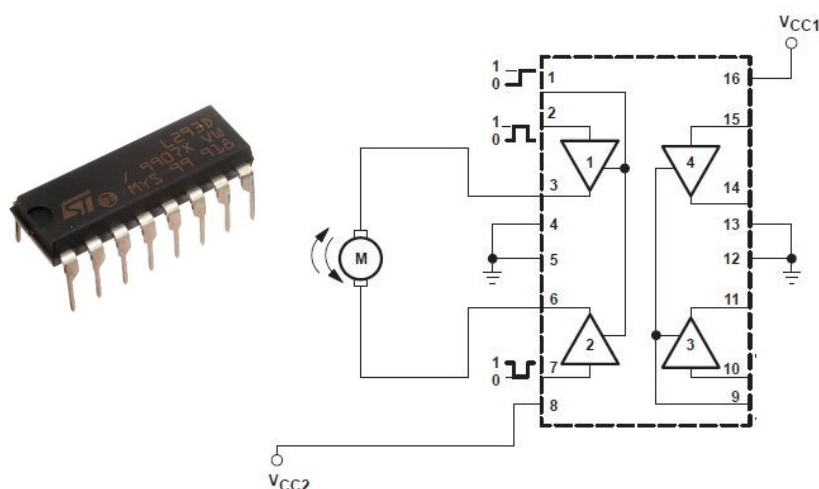


Figura 3-57: Circuito referente à rotação do motor em dois sentidos.

De acordo com a Figura 3-57, o contato 2 e 7 vão determinar o sentido de rotação do motor. Para o motor girar é necessário um pino estar em nível alto e outro em nível baixo, se o 2 tiver em nível alto e o 7 em nível baixo o circuito conduzirá corrente do contato 3 para o 6, fazendo o motor girar. Se o pino 2 tiver em nível baixo e o pino 7 em nível alto conduzirá corrente do pino 6 ao 3 fazendo o motor girar no outro sentido. O pino 1 corresponde ao controlo da intensidade de corrente que passa pelo motor, usando o PWM (*Pulse-Width Modulation*). Todo o circuito é alimentado a 5V DC [52].

Verificou-se que o circuito implementado teria capacidade para ligar outro motor. Deste modo, ligou-se o motor encarregue de realizar o movimento do portão automático (Figura 3-58) [53]

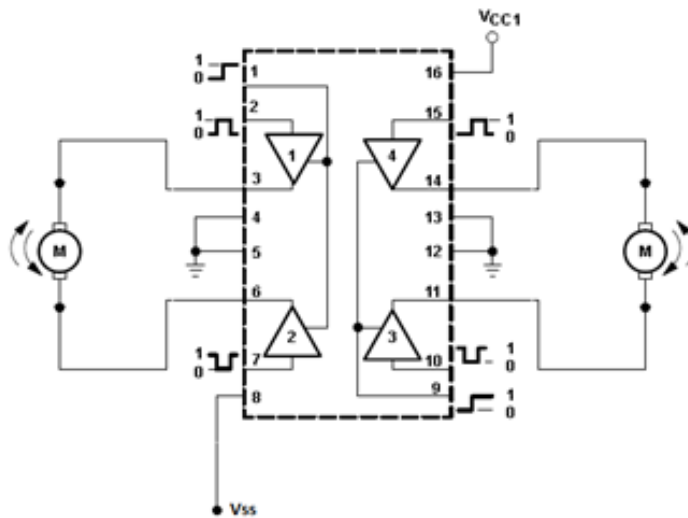


Figura 3-58: Circuito referente à rotação de dois motores em dois sentidos.

Como a velocidade dos motores era muito elevada para o efeito desejado, após algumas pesquisas ponderou-se utilizar as saídas DC do módulo associado ao PLC para realizar o PWM. No entanto, optou-se por dimensionar um circuito capaz de realizar o mesmo efeito (Figura 3-59). O circuito é composto essencialmente por um NE555 (*chip* utilizado numa variedade de aplicações como temporizador ou oscilador) [54].

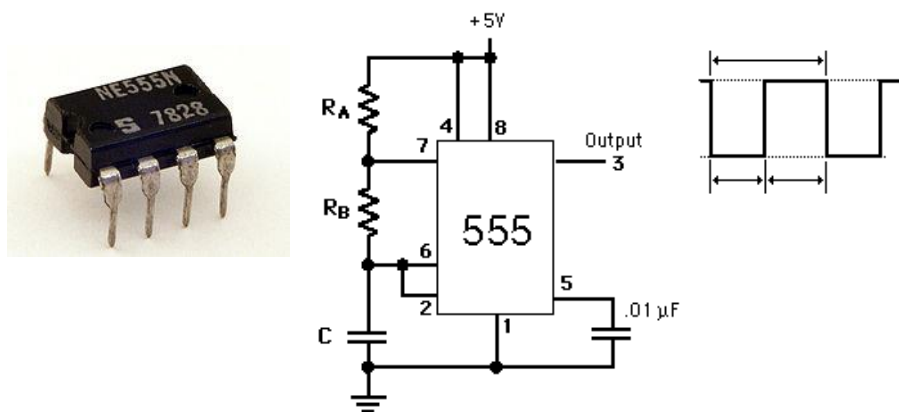


Figura 3-59: Circuito referente ao PWM.

O pino 3 da Figura 3-59 ligou-se ao pino 1 do circuito da Figura 3-58. Assim resultou um circuito capaz de controlar velocidades e sentidos de rotação (Figura 3-60).

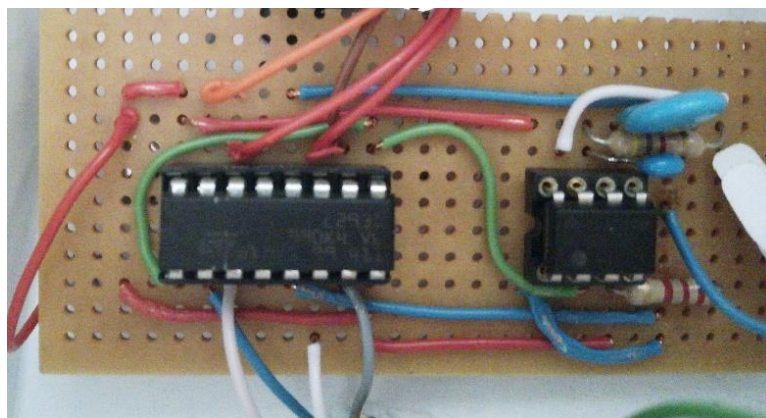


Figura 3-60: Circuito capaz de controlar velocidades e sentidos de rotação dos motores.

De seguida, realizou-se a interligação com o PLC. O botão de pressão correspondente à subida do estore ligou-se à entrada I1.4 e o botão de pressão correspondente à descida do estore ligou-se à entrada I1.5.

O botão de pressão correspondente à abertura do portão ligou-se à entrada I8.0 e o botão de pressão correspondente ao fecho do portão ligou-se à entrada I8.1.

O pino 2 e 7 e o pino 10 e 15 do circuito L293D ligaram-se a quatro relés (Figura 3-61) que por sua vez se ligaram às saídas do PLC, Q8.2, Q8.3, Q8.4 e Q8.5.

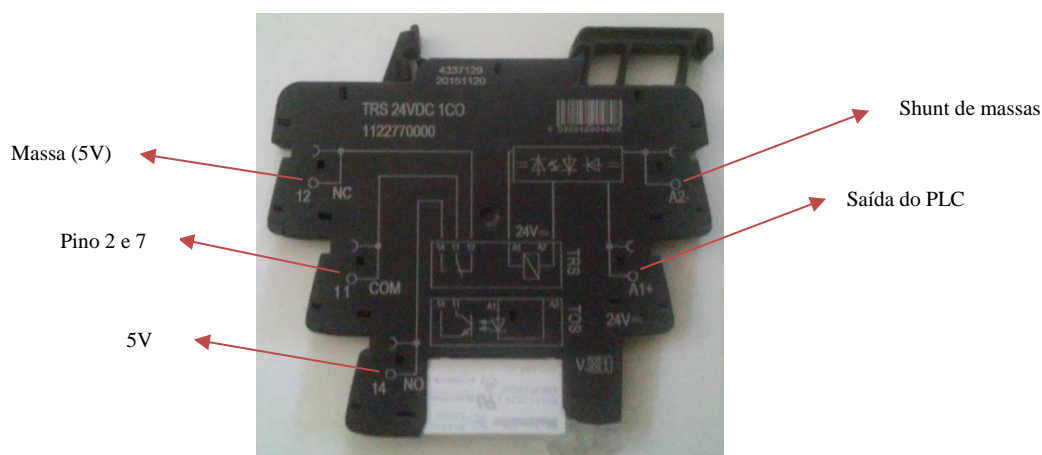


Figura 3-61: Ligação do relé.

Por fim, montou-se na maquete um protótipo de um estore, Figura 3-62 e de um portão, Figura 3-63. O eixo de rotação do motor faz rodar uma roda dentada que, por sua vez, roda sobre uma régua dentada solidária ao estore (e ao portão) causando movimento. Na maquete o estore

encontra-se na janela do quarto e o portão automático no jardim. Este, por sua vez, quando realiza o fecho ou abertura é ativada uma luz intermitente (M0.5 - Clock0,125H). Ainda no portão automático existe uma barreira de raios infravermelhos (simulada por um interruptor) que se encontra ligada à entrada I8.2 do PLC que, quando é ativada, o fecho automático é imediatamente cancelado.



Figura 3-62: Protótipo de um estore.

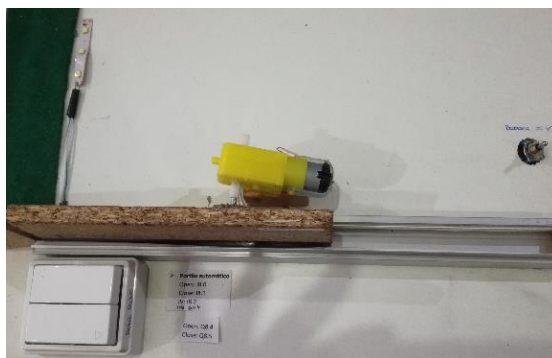


Figura 3-63: Protótipo de um portão automático.

3.6.3.4 Sistema de Rega

A simulação do sistema de rega implementou-se com o auxílio de um sensor de humidade do solo, alimentado a 5V DC, juntamente com um módulo controlador (Figura 3-64), um interruptor e um led de sinalização de ocorrência de rega. O led de sinalização foi alimentado a 5V DC por intermédio de uma resistência.

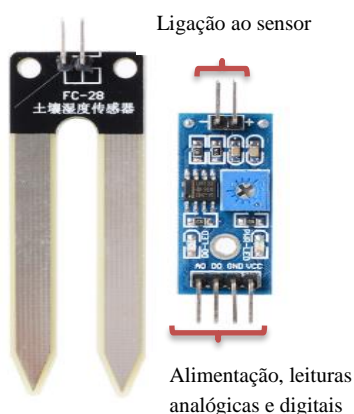


Figura 3-64: Sensor de humidade do solo e seu módulo controlador.

O modo automático do sistema de rega é simulado por um interruptor ligado à entrada I0.6 do PLC e o sinal de ocorrência de rega está associado à saída Q0.5.



Figura 3-65: Sensor instalado no jardim da maquete.

3.6.3.5 Sistema de Intrusão

A simulação do sistema de intrusão realizou-se com o auxílio de um sensor magnético, um interruptor para accionar o modo automático e por um bezouro. O sensor magnético foi colocado na porta que dá acesso ao interior da maquete (Figura 3-66). Sempre que este sensor for ativado o bezouro (Figura 3-67) emite um som, desde que o modo automático esteja ativo. O modo automático encontra-se simulado por um interruptor ligado à entrada I8.4 do PLC.



Figura 3-66: Sensor magnético instalado na porta de entrada da maquete.



Figura 3-67: Besouro.

O sensor magnético foi alimentado a 24V DC por intermédio de um paralelo de duas resistências de 10K Ω (Figura 3-68) e ligado à entrada I8.3. O besouro foi ligado diretamente à saída Q0.6 do PLC.

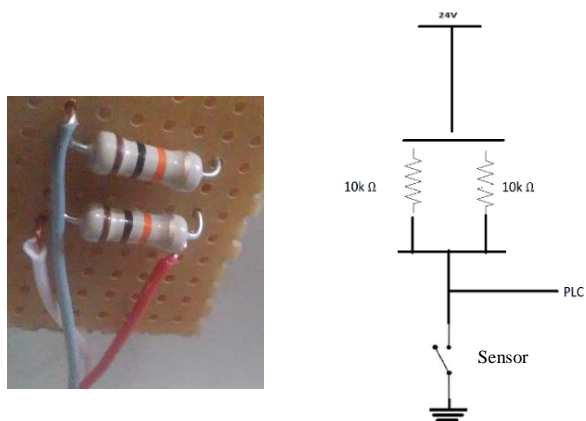


Figura 3-68: Circuito referente à intrusão.

3.6.3.6 Simulação de presença

A simulação de presença utiliza as ligações da iluminação já descritas anteriormente.

3.6.3.7 Correio Postal

A simulação de chegada de correio postal realizou-se com o auxílio de um sensor magnético que se encontra incorporado na caixa de correio (Figura 3-69). O sensor magnético foi alimentado a 24V DC por intermédio de um paralelo de duas resistências de 10K Ω e ligado à entrada I8.6 do PLC. O sinal de aviso de chegada de correio postal corresponde à saída Q1.1 do PLC.



Figura 3-69: Caixa de correio.

3.6.3.8 Outros

As entradas I0.0, I0.1, I0.2, I0.3 e I0.4 utilizaram-se para ativar o envio e receção das mensagens e assim testar algumas potencialidades do sistema implementado.

4. Conclusões e trabalhos futuros

Com a realização deste projeto, pode-se afirmar que os principais objetivos foram cumpridos. Contudo ao longo do desenvolvimento foram surgindo novas ideias e que foram implementadas. Nomeadamente construiu-se uma maquete com o intuito de simular todo o sistema, não só remotamente, como inicialmente proposto, através de SMS, de uma aplicação *android* e de uma página Web, como também localmente, com recurso a diversos equipamentos e componentes. Adicionalmente o sistema implementado é capaz de enviar e-mails de alerta e de armazenar dados e valores referentes ao sistema. Este tipo de projeto apresenta grandes vantagens aos potenciais utilizadores e aos locais onde possam ser instalados, uma vez que, permitem o controlo e a monitorização pretendidos.

Ao longo da implementação do projeto surgiram várias dificuldades, nomeadamente problemas com o PLC, devido à sua memória reduzida e escassez de entradas e saídas digitais e analógicas. Durante o desenvolvimento da aplicação *android* surgiram dificuldades em conseguir comunicar a partir do IP público. Na criação da página Web surgiu a necessidade de comprimir tudo e não inovar em demasia, devido às limitações do PLC. Em relação ao envio das mensagens a maior dificuldade residiu na criação dos comandos.

As dificuldades referidas contribuíram para a aprendizagem em saber lidar com as limitações não pondo em causa o objetivo inicial.

O sistema está também preparado para eventuais expansões e futuras melhorias, como, por exemplo, a ampliação de funcionalidade com recurso a outros PLC, um sistema de controlo e monitorização *ZigBee*, entre outros. Estas futuras melhorias podem ser consideradas como uma mais-valia a implementar com o intuito de preencher algumas lacunas.

Com a realização deste projeto posso concluir que enriqueci muito os meus conhecimentos em áreas diversificadas e adquiri bases necessárias para continuar a trabalhar e desenvolver projetos nesta área.

REFERÊNCIAS

- [1] C. Seongju, A. Mahdavi, A Hybrid system for daylight - responsive lighting, School of Architecture Carnegie Mellon University Pittsburgh PA 15213 USA. (2001).
- [2] V. Badescu, B. Sicre, Renewable energy for passive house heating II. Model, Faculty of Mechanical Engineering, Candida Oancea Institute of Solar Energy, Polytechnic University of Bucharest, Spl. Independentei 313, Bucharest 79590, Romania, Computational Physics, Institute of Physics, Technical University of Chemnitz, D-09107 Chemnitz, Germany, Elsevier. (2003).
- [3] F. Calvino, M. Gennusa, G. Rizzo, G. Scaccianoce, The control of indoor thermal comfort conditions: introducing a fuzzy adaptive controller, Department of Energy and Environmental Researches (D.R.E.A.M.), Università Di Palermo, Viale Delle Scienze, Palermo 90128, Italy, Elsevier. (2004).
- [4] M. Mozer, The Neural Network House: An Environment hat Adapts to its Inhabitants, Department of Computer Science and Institute of Cognitive Science University of Colorado Boulder, CO 80309-0430, AAI Technical Report SS-98-02. Compilation Copyright © 1998. (n.d.).
- [5] R. Keyser, C. Ionescu, Modelling and simulation of a lighting control system, EeSA – Department of Electrical Energy, Systems & Automation, Ghent University, Technologiepark 913, B-9052 Gent, Belgium, Elsevier. (2010).
- [6] Domótica, Domótica. (2016). <https://pt.wikipedia.org/wiki/Dom%C3%B3tica> (accessed March 28, 2016).
- [7] L. Sousa, Sistema modular de comunicação e controlo de dispositivos sensores/atuadores, Faculdade do Porto, 2012. <http://paginas.fe.up.pt/~ee04228/actas/tese2.pdf> (accessed May 24, 2016).
- [8] J. Gil, Instalaciones automatizadas en viviendas y edificios, Paraninfo, 2001.
- [9] La domótica o sistema de automatización de vivienda, Tecnologías Residenciales. (2012). <https://aicointegracion.wordpress.com/tag/tecnologia-residencial/>.
- [10] L. Araújo, Domótica, (2000). <http://slideplayer.com.br/slide/281409/> (accessed February 6, 2016).
- [11] C. Garcia, Instalaciones domoticas, S.L. EDICIONES CEYSA. CANO PINA, 2011.
- [12] J. Ferreira, Interface homem-máquina para domótica baseado em tecnologias Web, 2008. https://web.fe.up.pt/~ee01117/rel_6JUL.pdf (accessed March 29, 2016).
- [13] S. Hernaski, Introdução, (TCC). <http://www.eletrica.ufpr.br/ufpr2/tccs/145.pdf> (accessed April 23, 2016).
- [14] P. Rea, Domótica en la vivienda, (Estudiante Universidad Politécnica Salesiana). <http://www.monografias.com/trabajos93/domotica-vivienda/domotica-vivienda.shtml> (accessed April 28, 2016).
- [15] F. Pinto, Desenvolvimento de um Protótipo de um Sistema Domótico, Instituto Superior Técnico, 2010. <https://fenix.tecnico.ulisboa.pt/downloadFile/395142228713/dissertacao.pdf> (accessed February 5, 2016).
- [16] D. Santos, Telegrama, (2010). <http://ave.dee.isep.ipp.pt/~dss/Disciplinas/DOMOT/Teorica/Telegrama.pdf> (accessed March 17, 2016).
- [17] Y. Amir, Carrier Sense Multiple Access (CMSA), (2000). <http://www.cs.jhu.edu/~yairamir/cs437/week2/sld023.htm> (accessed February 4, 2016).

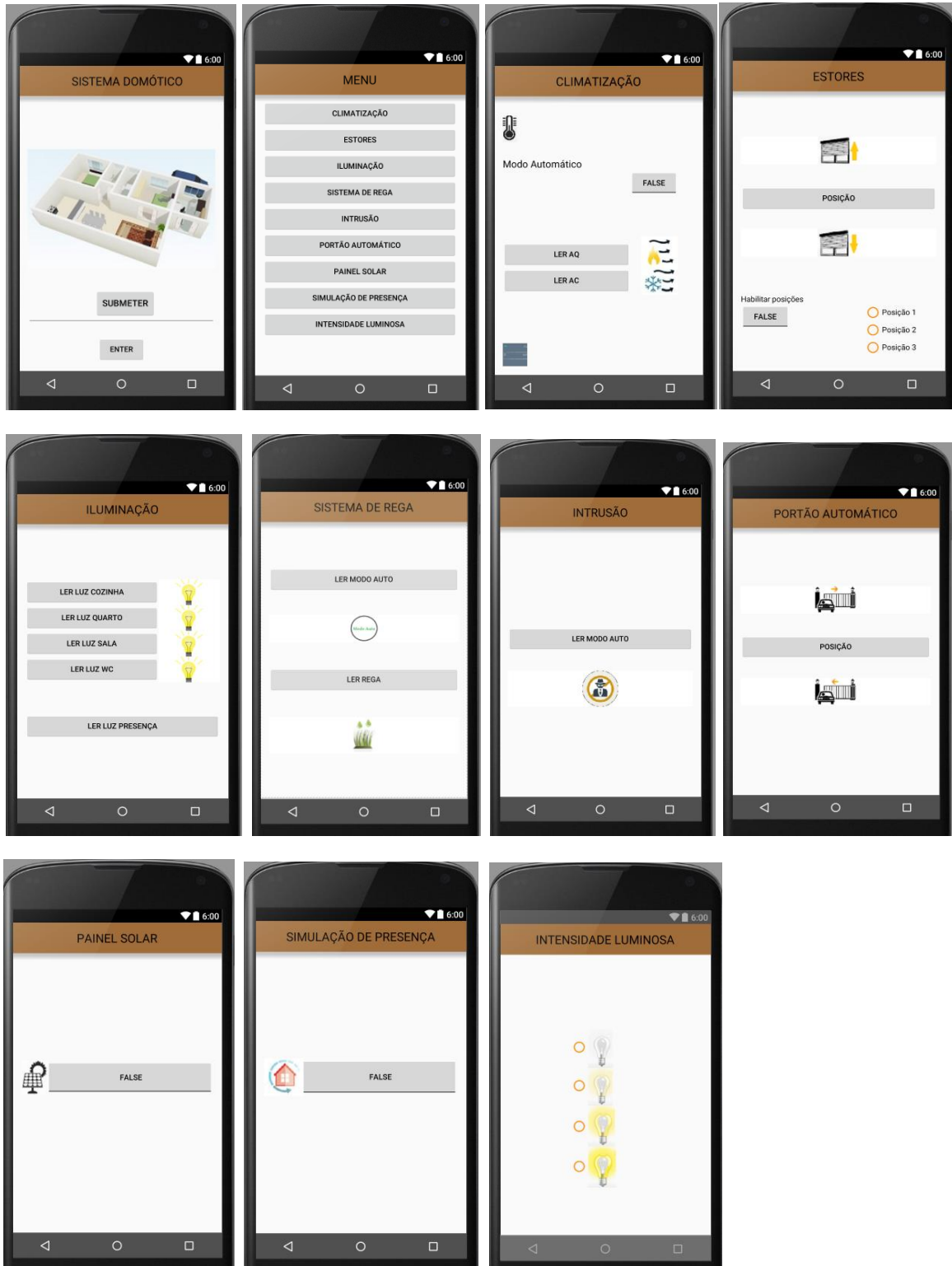
REFERÊNCIAS

- [18] S. Wang, Intelligent Buildings and Building Automation, Taylor & Francis e-Library, 2010.
- [19] C. Dias, Domótica - Aplicabilidade e Sistemas de Automação Residencial, (2004). <http://essentiaeditora.iff.edu.br/index.php/vertices/article/viewFile/1809-2667.20040015/86> (accessed April 14, 2016).
- [20] J.M. Maestre, Domótica para ingenieros, Paraninfo, 2015.
- [21] X10 - Casa Inteligente - Home Automation, (n.d.). <http://www.electronica-pt.com/domotica/x10> (accessed March 24, 2016).
- [22] K. Betnacourt, Tipos de dispositivos X-10, Domótica - Sistema X-10. (Universidad Politécnica Salesiana, Sede Cuenca). <http://www.monografias.com/trabajos104/domotica-sistemax-10/domotica-sistemax-10.shtml> (accessed February 4, 2016).
- [23] A. Gamboa, Eletricista de Instalações, (2013). https://cld.pt/dl/download/33da0b52-c122-4175-82bd-eef95e2f91b2/Dom%C3%B3tica_EI.pdf (accessed March 27, 2016).
- [24] Domótica, Protocolo de comunicação X-10, (2000). www.prof2000.pt/users/lpa/x10.ppt (accessed April 14, 2016).
- [25] D. Brewer, Home Automation, Made Easy, QUE, United States of America, 2014.
- [26] A. Silva, Transdutores domóticos inteligentes, Faculdade do Porto, 2004. <http://paginas.fe.up.pt/~ee99075/projecto/relatorio.pdf> (accessed February 4, 2016).
- [27] P. Matutino, Conceção e Desenvolvimento de uma Rede Domótica, Instituto Superior Técnico, 2001. <https://pt.scribd.com/document/26604774/Master-Thesis> (accessed March 22, 2016).
- [28] Domótica, Protocolo de comunicação EIB-KNX, (2000). www.prof2000.pt/users/lpa/eib_knx.ppt (accessed April 18, 2016).
- [29] L. González, Instalaciones domóticas, Librería Central, SA, Madrid, 2010.
- [30] S. Electric, KNX Sistema de Controlo Inteligente de Edifícios, (n.d.). <http://www.schneiderelectric.pt/sites/portugal/pt/produtos-servicos/product-launch/knx/how-does-knx-works.page> (accessed May 13, 2016).
- [31] KNX, KNX Principios de funcionamiento, (n.d.). https://www.knx.org/media/docs/downloads/Marketing/Flyers/KNX-System-Principles/KNX-System-Principles_es.pdf (accessed April 22, 2016).
- [32] Integrador Oficial LonWorks, Sistema LON (Conceito), (2007). <http://www.contimetra.com/paginas/arcondicionado/segcontraincendios/documentacaocontimetra/LON-conceito.pdf> (accessed May 28, 2016).
- [33] Echelon Corporation, LCT06 LonWorks Integration, (2015). <https://vimeo.com/groups/lct/videos/122066771> (accessed May 14, 2016).
- [34] Á. Tiribicá, Automação predial, (2014). <http://slideplayer.com.br/slide/290476/> (accessed April 26, 2016).
- [35] Wikipédia, Controlador lógico programável, (2016). https://pt.wikipedia.org/wiki/Controlador_l%C3%B3gico_program%C3%A1vel (accessed March 21, 2016).
- [36] South Westphalia University of Applied Sciences, Market survey, Busbased installation technology in residential and commercial building, (2009). http://www.knx.org/fileadmin/news/12287318302050466721Survey_English.pdf (accessed June 9, 2016).
- [37] Siemens, SCE training documentation for universal automation solution Totally Integrated Automation (TIA), (2013). <https://w3.siemens.com/mcms/sce/de/fortbildungen/ausbildungsunterlagen/tia->

- portal_module/tabcardseiten/Documents/SCE_PT_010-050_R1209_S7-1200_Analogwerte.pdf (accessed March 27, 2016).
- [38] B. Mendes, Desenvolvimento de aplicação para a otimização de processo produtivo, isep, n.d. file:///D:/Downloads/DM_BrunoMendes_2015_MEEC%20(3).pdf (accessed February 4, 2016).
- [39] L. Martins, Projecto dum Laboratório Remoto para Automação de Processos Industriais, ISEL, 2013. <http://repositorio.ipl.pt/bitstream/10400.21/3181/1/Disserta%C3%A7%C3%A3o.pdf> (accessed April 18, 2016).
- [40] LM35 Temperature Sensor, (n.d.). <http://www.instructables.com/id/LM35-Temperature-Sensor/> (accessed April 26, 2016).
- [41] J. Silva, Soluções para evitar o sobreaquecimento dos coletores solares térmicos, para painel plano e tubos de vácuo, (2012). <http://dicasesquemas.blogspot.pt/2012/09/solucao-para-evitar-o-sobreaquecimento.html> (accessed April 15, 2016).
- [42] Android Studio, (n.d.). <https://developer.android.com/studio/index.html> (accessed March 1, 2016).
- [43] Snap7, (2014). <http://snap7.sourceforge.net/> (accessed February 3, 2016).
- [44] D. Nardella, Moka7, Reference Manual, (2014).
- [45] Communication between Android and Siemens PLC, (2015). <https://www.youtube.com/watch?v=HrctU8MasE0> (accessed February 15, 2016).
- [46] (n.d.). <http://www.w3schools.com/> (accessed February 4, 2016).
- [47] Introdução a HTML e CSS, (n.d.). <https://www.caelum.com.br/apostila-html-css-javascript/introducao-a-html-e-css/#2-8-primeira-pagina> (accessed April 17, 2016).
- [48] Jager, Siemens S7-1200 Web server Tutorial - From Getting started to HTML5 User Defined Pages, (2013). <https://www.dmcinfo.com/latest-thinking/blog/id/8567/siemens-s7-1200-web-server-tutorial--from-getting-started-to-html5-user-defined-pages> (accessed April 25, 2016).
- [49] W. Windorpski, Tutorial: how to make basic html web page for your Siemens S7 PLC webserver, (2013). <https://www.youtube.com/watch?v=u9RgUeyjaAM> (accessed April 14, 2016).
- [50] Siemens, Simatic S7-1200 Programmable controller, (2013).
- [51] Siemens, Industrial Remote Communication: Data Communication via GPRS with S7-1200 and CP 1242-7, (2013).
- [52] R. Theis, Motor CC com driver de potência, (2012). <http://arduinoifsul.blogspot.pt/2012/10/motor-cc-com-driver-de-potencia.html?view=timeslide> (accessed May 14, 2016).
- [53] A. Martínez, Práctica de manejo de motores CC con arduino, (n.d.). http://platea.pntic.mec.es/~mhidalgo/documentos/10_manejo_de_motores_CC_con_arduino.pdf (accessed May 6, 2016).
- [54] The 555 Timer, (n.d.). <http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/t555.html#c2> (accessed February 5, 2016).

ANEXO 1

Diferentes *layouts* da aplicação *android* desenvolvida.



ANEXO 2

Alguns exemplos de páginas Web desenvolvidas.



Comando da Climatização

Temperatura atual:22.26201

Aquecimento: OFF

On

Off

AC: OFF

On

Off

Modo Auto: OFF

Ligar

Marisa Matos Oliveira
ESTGV-MEE



Comando dos Estores

Subir Estore

Subir

Descer Estore

Descer

Habilitar Posições: OFF

Ligar

25%

50%

75%

OK

Posição Estore: Totalmente subido

Marisa Matos Oliveira
ESTGV-MEE





Comando da Iluminação

Luz cozinha: OFF

Ligar

Luz quarto: OFF

On

Off

Luz sala: OFF

On

Off

Luz wc: OFF

On

Off

Luz presença: OFF



Intensidade Luminosa: Sem intensidade

- sem intensidade
- mínima
- média
- máxima

OK



ANEXO 3

Comandos de leitura e escrita no PLC relativo ao envio e receção de mensagens.

```
#Found := False;

#Msg := '';

// Help? Mostra os Comandos
IF FIND(IN1 := #Data, IN2 := 'Help?') <> 0 THEN
  #Found := True;
  #Msg := CONCAT(IN1 := #Msg, IN2 := 'Temp?, Aquecimento?, AC?, MA?, MS?, SP?, P1?, P2?, SR?, Rega?, Int?, Posicao=, Aquecimer
END_IF;

// Ler valores do PLC
IF #Cmd = 'G' THEN

  // Exemplo para um valor REAL - Temperatura (CLIMATIZACAO)
  IF FIND(IN1 := #Data, IN2 := 'Temp?') <> 0 THEN
    #Found := True;
    #Msg := CONCAT(IN1 := #Msg, IN2 := 'Temp:');
    #Msg := CONCAT(IN1 := #Msg, IN2 := REAL_TO_STRING("Tags".DataLog.DATA.Temperature));
  END_IF;

  // Exemplo para um valor BOOL - Aquecimento (CLIMATIZACAO)
  IF FIND(IN1 := #Data, IN2 := 'Aquecimento?') <> 0 THEN
    #Found := True;
    #Msg := CONCAT(IN1 := #Msg, IN2 := 'Aq:');

    IF ("Aquecimento") THEN
      #Msg := CONCAT(IN1 := #Msg, IN2 := 'TRUE');
    ELSE
      #Msg := CONCAT(IN1 := #Msg, IN2 := 'FALSE');
    END_IF;
  END_IF;

  // Exemplo para um valor BOOL - AC (CLIMATIZACAO)
  IF FIND(IN1 := #Data, IN2 := 'AC?') <> 0 THEN
    #Found := True;
    #Msg := CONCAT(IN1 := #Msg, IN2 := 'AC:');

    IF ("AC") THEN
      #Msg := CONCAT(IN1 := #Msg, IN2 := 'TRUE');
    ELSE
      #Msg := CONCAT(IN1 := #Msg, IN2 := 'FALSE');
    END_IF;
  END_IF;

  // Escrever valores no PLC
ELSIF #Cmd = 'S' THEN

  // Exemplo para um valor BOOL - Aquecimento (CLIMATIZACAO)
  IF FIND(IN1 := #Data, IN2 := 'Aquecimento=ON') <> 0 THEN
    #Found := True;
    "Controlo_Remoto".Aquecimento.Ligar_aq := TRUE;
    #Msg := CONCAT(IN1 := #Msg, IN2 := 'Aq:ON');

  ELSIF FIND(IN1 := #Data, IN2 := 'Aquecimento=OFF') <> 0 THEN
    #Found := True;
    "Controlo_Remoto".Aquecimento.Ligar_aq := FALSE;
    #Msg := CONCAT(IN1 := #Msg, IN2 := 'Aq:OFF');
  END_IF;

  // Exemplo para um valor BOOL - AC (CLIMATIZACAO)
  IF FIND(IN1 := #Data, IN2 := 'AC=ON') <> 0 THEN
    #Found := True;
    "Controlo_Remoto".AC.Ligar_ac := TRUE;
    #Msg := CONCAT(IN1 := #Msg, IN2 := 'AC:ON');

  ELSIF FIND(IN1 := #Data, IN2 := 'AC=OFF') <> 0 THEN
    #Found := True;
    "Controlo_Remoto".AC.Ligar_ac := FALSE;
    #Msg := CONCAT(IN1 := #Msg, IN2 := 'AC:OFF');
  END_IF;
```

ANEXO 4

Entradas digitais e analógicas	Descrição
I0.0	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> Receber Enviar Broadcast Escalation S/G </div> <div style="font-size: 2em;">}</div> <div style="margin-left: 10px;">SMS</div> </div>
I0.1	
I0.2	
I0.3	
I0.4	
I0.5	Modo automático da climatização
I0.6	Modo automático da rega
I1.0	Interruptor luz da cozinha
I1.1	Interruptor luz da sala
I1.2	Interruptor luz do quarto
I1.3	Interruptor luz do WC
I1.4	Botão subir estore
I1.5	Botão descer estore
I8.0	Botão abrir portão
I8.1	Botão fechar portão
I8.2	IV do portão
I8.3	Sensor intrusão
I8.4	Modo automático da intrusão
I8.6	Sensor do correio
I8.7	Modo automático do estore do painel solar
IW64	Sensor temperatura
IW66	Sensor de humidade do solo
Saídas digitais e analógicas	Descrição
Q0.0	Luz da cozinha
Q0.1	Luz da sala
Q0.2	Luz do quarto
Q0.3	Luz do WC
Q0.4	Luz do portão
Q0.5	Rega
Q0.6	Intrusão
Q1.0	Luz presença
Q1.1	Correio
Q8.0	Aquecimento
Q8.1	AC
Q8.2	Subida do estore
Q8.3	Descida do estore
Q8.4	Abertura do portão
Q8.5	Fecho do portão
Q8.6	Subida do estore do painel solar
Q8.7	Descida do estore do painel solar
QW80	Variação da Intensidade luminosa