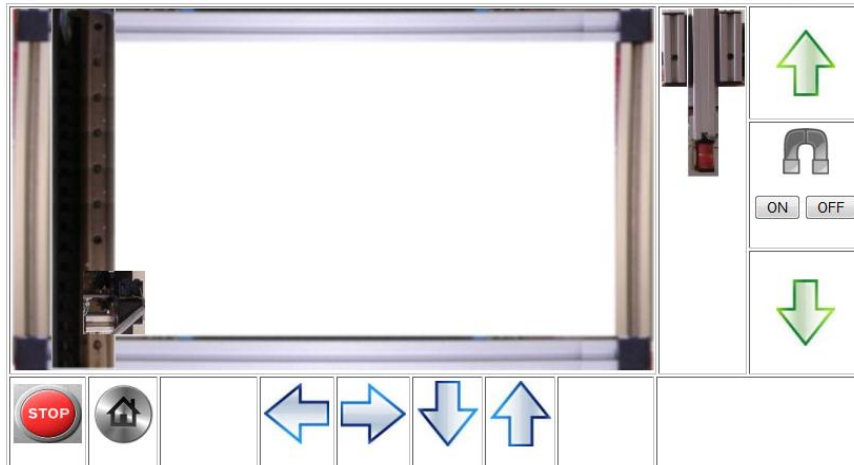




ISEL

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Área Departamental de Engenharia Mecânica



Monitorização via internet de Processos Industriais controlados com Autómatos Programáveis

RUI MIGUEL FERREIRA MATIAS
(Licenciado em Engenharia Mecânica)

Trabalho Final de Mestrado para obtenção do grau de Mestre
em Engenharia Mecânica

Orientador:

Doutor Mário José Gonçalves Cavaco Mendes

Júri:

Presidente: Doutor Rui Pedro Chedas Sampaio

Vogais:

Doutor João Carlos Prata dos Reis

Doutor Mário José Gonçalves Cavaco Mendes

Dezembro de 2014



ISEL

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Área Departamental de Engenharia Mecânica

Monitorização via internet de Processos Industriais controlados com Autómatos Programáveis

RUI MIGUEL FERREIRA MATIAS
(Licenciado em Engenharia Mecânica)

Trabalho Final de Mestrado para obtenção do grau de Mestre
em Engenharia Mecânica

Orientador:

Doutor Mário José Gonçalves Cavaco Mendes

Júri:

Presidente: Doutor Rui Pedro Chedas Sampaio

Vogais:

Doutor João Carlos Prata dos Reis

Doutor Mário José Gonçalves Cavaco Mendes

Dezembro de 2014

Agradecimentos

Em primeiro lugar gostaria de agradecer ao meu orientador Doutor Mário J. G. C. Mendes, pela sua disponibilidade, apoio e ajuda durante a realização deste trabalho e, em especial, pelo tempo despendido a rever este documento.

Gostaria também de agradecer à Siemens pelo material cedido à ADEM, sem o qual não me seria possível realizar este trabalho de Mestrado.

Por fim, mas não menos importante quero agradecer aos meus pais pelo incentivo que me deram ao longo da minha vida académica.

Resumo

A Automação Industrial é uma área muito importante na indústria portuguesa e mundial, e com a crescente utilização da internet como meio de comunicação e com a necessidade de supervisionar, controlar e manter os processos industriais, torna-se cada vez mais importante monitorizar remotamente cada processo industrial, com o objectivo de reduzir custos de manutenção e vigiar o controlo das máquinas e processos.

Por este motivo, o presente trabalho de Mestrado teve como objectivo a melhoria do Laboratório Remoto de Automação de Processos Industriais (LabAPI – ISEL), no sentido de incorporar mais experiências laboratoriais (assim como melhorar o que já estava feito) que possam ser utilizadas na leccionação destas matérias, podendo essas experiências ser acedidas e programadas remotamente pelos alunos de duas unidades curriculares distintas. As experiências que constituem o laboratório remoto são controladas por autómatos programáveis S7-1200 da Siemens, que possuem a funcionalidade de *webservice* (servidor) permitindo a monitorização e supervisão remota de cada experiência através de uma simples página *web* (do inglês, *webpage*). E com o intuito de controlar o acesso de alunos, devido ao número limitado de experiências e ao elevado número de alunos que as utilizarão, foi criado um sistema que permite apenas o acesso a um aluno de cada vez.

Este laboratório remoto foi elaborado com o auxílio de vários programas, onde o programa TIA Portal da Siemens foi o mais importante, e com o recurso a várias linguagens de programação, tais como, PHP, HTML e JavaScript, sendo assim possível a criação das referidas páginas *web*, alojadas num servidor que está localizado junto ao Laboratório Físico e onde também está a base de dados dos alunos que podem aceder ao laboratório.

Palavras-Chave

Laboratório Remoto; Automação Industrial; Autómato Programável.

Abstract

The Industrial Automation is a very important area in the portuguese industry and worldwide, and the increasing use of the internet as a communication medium and the need to supervise, control and maintain industrial processes, becomes increasingly important remotely monitor each industrial process in order to reduce maintenance costs and to monitor the control of machines and processes.

For this reason, the present work aimed to improve the Laboratório Remoto de Automação de Processos Industriais (LabAPI - ISEL), to incorporate more laboratory experiments (as well as improve what was already done) that can be used in teaching these subjects, these experiences can be remotely accessed and programmed by students in two different courses. The experiences that constitute the remote laboratory are controlled by Siemens PLCs S7-1200 that have the functionality of webserver allowing remote monitoring and supervision of each experience through a simple webpage. And in order to control the access of students, due to the limited number of experiments and the high number of students that will use it, was created a system that allows access to only one student at a time.

This remote laboratory was developed with the aid of several programs, where the program TIA Portal of Siemens was the most important, and the use of several programming languages such as PHP, HTML and JavaScript, therefore possible the creation of these web pages, hosted on a server that is located next to the Physical Laboratory and is also where the database of students who can access the laboratory is located.

Keywords

Remote Laboratory; Industrial Automation; Programmable Logic Controller

Glossário

CPU	Central Processing Unit (Unidade Central de Processamento)
FBD	Function Block Diagram (Diagrama de Funções)
HMI	Human-Machine Interface (Interface Homem-Máquina)
HTML	HyperText Markup Language (Linguagem de Marcação de Hipertexto)
IEC	International Electrotechnical Commission (Comissão Eletrotécnica Internacional)
IL	Instruction List (Lista de Instruções)
LD	Ladder Diagram (Diagrama de Contactos)
PHP	Hypertext Preprocessor
PLC	Programmable Logic Controller (Controlador Lógico Programável)
SFC	Sequential Function Chart (Gráfico Sequencial de Funções)
SL	Structural Text (Texto Estruturado)
SQL	Structured Query Language (Linguagem de Consulta Estruturada)
TIA Portal	Totally Integrated Automation Portal
VPN	Virtual Private Network (Rede Privada Virtual)
WYSIWYG	What You See Is What You Get (O Que Vê É O Que Tem)

Índice

Agradecimentos	I
Resumo	III
Palavras-Chave	III
Abstract.....	V
Keywords.....	V
Glossário.....	VII
Lista de Figuras	XIII
Lista de Tabelas	XVI
1. Introdução.....	1
1.1. Motivação	1
1.1.1. Automação Industrial	1
1.1.2. Laboratório Remoto.....	2
1.2. Objectivos	3
1.3. Estrutura do Trabalho	4
2. Estado da Arte	6
2.1. Automação Industrial.....	6
2.1.1. Controladores Lógicos Programáveis.....	7
2.1.2. Linguagens Normalizadas	8
2.2. Laboratórios Remotos.....	12
2.2.1. Vantagens e Desvantagens	13
2.2.2. Exemplos de Laboratórios Remotos.....	14
3. Laboratório Remoto de Automação de Processos Industriais do ISEL.....	23
3.1. PLC Siemens S7-1200	23
3.2. Programas Utilizados	25

3.2.1.	WampServer	25
3.2.2.	Notepad++	25
3.2.3.	Dreamweaver.....	26
3.2.4.	TIA Portal.....	27
3.2.5.	ThinVNC	28
3.2.6.	YAWCAM	29
3.3.	Linguagens de Programação	29
3.3.1.	HTML.....	29
3.3.2.	PHP.....	30
3.3.3.	JavaScript	31
3.3.4.	SQL.....	32
3.4.	Arquitectura do LabAPI – ISEL	33
3.5.	Limitar o acesso a um Aluno de cada vez.....	36
3.5.1.	Autentificação do Utilizador	37
3.5.2.	Menu da Experiência	42
3.5.3.	Logout.....	45
3.6.	Experiência: Pórtico.....	46
3.6.1.	Ligação dos PLCs e do Pórtico	47
3.6.2.	Comunicação entre os Dois PLCs	49
3.6.3.	Página <i>Web</i> da Experiência: Pórtico.....	53
3.7.	Outras Alterações.....	59
3.7.1.	Formulário de Inscrição.....	59
3.7.2.	Menu.....	60
3.7.3.	Expulsar o Utilizador.....	63
3.7.4.	Botão de Emergência.....	66
3.7.5.	Painel HMI	67

3.7.6. Câmara de Vídeo	68
4. Conclusão	70
4.1. Trabalho Final	70
4.2. Trabalho Futuro	71
Bibliografia.....	73
Anexo A.....	77
Anexo B.....	84
Anexo C.....	96

Lista de Figuras

Figura 1.1 – Painel usado no Controlo de um Processo Industrial.....	2
Figura 1.2 – Arquitectura do Laboratório Remoto da TAMU-Q	3
Figura 2.1 – PLC S7-1200 Siemens, 2014	8
Figura 2.2 – Exemplo de Lista de Instruções	9
Figura 2.3 – Exemplo de Texto Estruturado	10
Figura 2.4 – Exemplo de Diagrama de Funções.....	11
Figura 2.5 – Exemplo de Diagrama de Contactos	11
Figura 2.6 – Exemplo de Gráfico Sequencial de Funções.....	12
Figura 2.7 – Classificação dos vários tipos de Laboratórios	13
Figura 2.8 – Tabela de Agendamento de Sessões	15
Figura 2.9 – Interface dos 3 tanques.....	16
Figura 2.10 – Representação Virtual do Multímetro.....	17
Figura 2.11 – Escolha do Laboratório a usar.....	18
Figura 2.12 – Antes da alteração da Interface	19
Figura 2.13 – Após alteração da Interface	20
Figura 2.14 – Interface do Processo de Estampagem de Peças	21
Figura 2.15 – Montagem das tubagens virtuais.....	21
Figura 3.1 – Exemplo de PHP no programa Notepad++.....	26
Figura 3.2 – Exemplo de HTML no programa Dreamweaver	27
Figura 3.3 – Exemplo de <i>Ladder Diagram</i> no programa TIA Portal	28
Figura 3.4 – Menu de Acesso ao Computador Remoto.....	28
Figura 3.5 – Exemplo de Linguagem HTML.....	30
Figura 3.6 – Exemplo de Linguagem PHP	31
Figura 3.7 – Arquitectura da Rede do LabAPI – ISEL	34
Figura 3.8 – Estrutura do Site do LabAPI – ISEL ¹	35

Figura 3.9 – Base de Dados de Alunos do LabAPI – ISEL	37
Figura 3.10 – Fluxograma da Autenticação do Utilizador	41
Figura 3.11 – Mensagem de Alerta	43
Figura 3.12 – Fluxograma do Menu da Experiência	44
Figura 3.13 – Fluxograma do <i>Logout</i>	45
Figura 3.14 – Pórtico	46
Figura 3.15 – Botões do Pórtico	46
Figura 3.16 – Esquema de Ligação do PLC aos Sensores e Actuadores	48
Figura 3.17 – Exemplo Função GET.....	51
Figura 3.18 – Exemplo Função PUT	52
Figura 3.19 – Accionar o <i>Output</i>	52
Figura 3.20 – Programa referente ao Movimento para a Esquerda	53
Figura 3.21 – Activar <i>Web Server</i> no PLC	53
Figura 3.22 – Função WWW.....	53
Figura 3.23 – Página <i>Web</i> da Experiência: Pórtico	54
Figura 3.24 – Íman OFF/ Íman ON	57
Figura 3.25 – Formulário de Inscrição com Campo em Falta	60
Figura 3.26 – Indicação de Aluno <i>Online</i> debaixo da Experiência referente.....	61
Figura 3.27 – Indicação de todos os Alunos <i>Offline</i>	62
Figura 3.28 – Fluxograma do Menu do LabAPI	62
Figura 3.29 – Página <i>Web</i> expulsar.php	63
Figura 3.30 – Fluxograma de Expulsar pelo Supervisor	65
Figura 3.31 – Botão de Emergência não Activado.....	67
Figura 3.32 – Botão de Emergência Activado.....	67
Figura 3.33 – Painel HMI	68
Figura 3.34 – Transmissão da Experiência.....	68

Lista de Tabelas

Tabela 2.1 – Linguagens de Programação segundo IEC 61131-3.....	9
Tabela 3.1 – Características do PLC S7-1214C AC/DC/Rly	24
Tabela 3.2 – Utilização de <i>web browsers</i> em Fevereiro de 2014.....	36
Tabela 3.3 – Resoluções utilizadas em Janeiro de 2014.....	36
Tabela 3.4 – Identificação dos Fios	47
Tabela 3.5 – Número dos Cabos Ligados à Energia.....	47
Tabela 3.6 – Identificação das Conexões dos Sensores.....	48
Tabela 3.7 – Identificação das Conexões dos Actuadores.....	49
Tabela 3.8 – Frequência de cada <i>bit</i> do <i>Clock Memory</i>	50

1. Introdução

Actualmente, os laboratórios remotos fazem parte do percurso académico de qualquer aluno, permitindo observar, explorar aplicações reais de teorias fundamentais, desenvolver uma melhor compreensão das aulas teóricas e complementar as aulas práticas. Com a evolução da internet e das tecnologias de informação e comunicação abriram-se novos horizontes para o processo de ensino/aprendizagem (Jara et al., 2011; Restivo, 2011).

Com as ferramentas tecnológicas disponíveis, a necessidade de ensinar um maior número de alunos e estarmos actualmente a encarar a chamada Geração da Internet, que tem uma especial capacidade para entender gráficos, imagens e que possui uma fantástica literacia digital torna-se essencial a criação de laboratórios remotos (Elawady & Tolba, 2011; Restivo, 2011).

Com a criação de laboratórios controlados remotamente é possível, criar horários mais flexíveis e efectuar uma experiência sem qualquer deslocação geográfica. Mas, é preciso não esquecer que o laboratório remoto deverá ser um complemento à experiência física, pois sem ela não é possível interagir com os equipamentos e perde-se a troca de experiências com os colegas de grupo.

1.1. Motivação

1.1.1. Automação Industrial

Actualmente, a automação sem qualquer intervenção humana consegue atingir resultados que não seriam possíveis apenas com processos manuais e além disso permite melhorar a segurança dos processos industriais, pois minimiza o erro humano.

Além de melhorar a qualidade dos processos, também permite:

- O fabrico de bens em condições perigosas para o ser humano (ex: nuclear, tóxico);
- Reduz o tempo de procedimentos como o arranque de centrais eléctricas;

- Tomada de decisões em tarefas complexas (ex: controlo de tráfego aéreo) que exigem uma tomada de decisão num curto espaço de tempo.

A automação ainda não substituiu o ser humano em todos os processos industriais, devido a algumas limitações, sendo elas, determinação de informação sensorial (cheiros e sons) e planeamento estratégico de tarefas.

Na Figura 1.1 pode ver-se um exemplo de um processo industrial controlado por autómatos programáveis.



Figura 1.1 – Painel usado no Controlo de um Processo Industrial (El Kontrol LTD, 2014)

1.1.2. Laboratório Remoto

O ensino da automação é melhorado sempre que o ensino é sustentado por laboratórios adequados e experiências que seguem o paradigma de “aprender fazendo” (*“learning by doing”*), que permite os alunos terem um melhor conhecimento das lições teóricas. Contudo, equipamentos caros e tempo limitado previnem o professor de ter plataformas educacionais suficientes (Jara et al., 2011).

Mas, com o desenvolvimento das tecnologias torna-se possível a introdução de novas características no ensino à distância, transformando-o numa educação interactiva, que permite uma comunicação e interacção permanente entre professores, estudantes e materiais de aprendizagem permitindo o ensino das áreas referidas a um custo razoável (Tawfik et al., 2011).

Os laboratórios remotos (exemplo: Figura 1.2) permitem a experimentação e monitorização via internet através do controlo de instrumentos em tempo real, implementando conhecimentos práticos e teóricos, em qualquer altura e local geográfico.

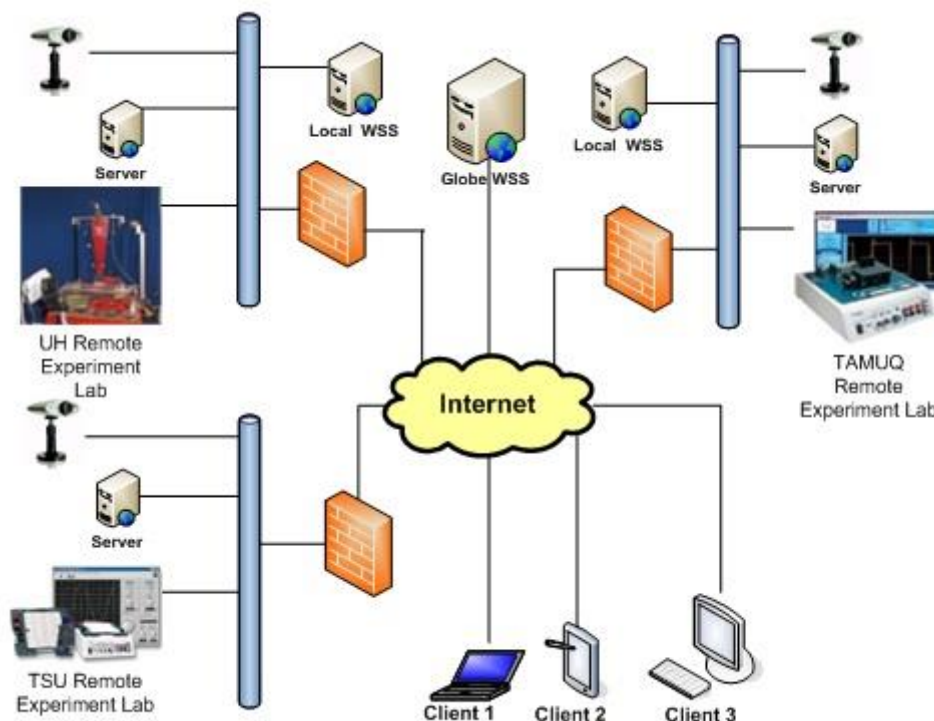


Figura 1.2 – Arquitectura do Laboratório Remoto da TAMU-Q (Remote Laboratory, 2011)

1.2. Objectivos

Os objectivos principais deste trabalho são os seguintes:

- Criação/melhoria de um laboratório remoto com várias experiências;
- Criação das páginas *web* relativas às experiências com o fim de as monitorizar, via internet;
- Criação de um sistema que permita apenas o acesso de um aluno de cada vez.

Ao longo deste trabalho foram propostos alguns objectivos secundários sendo eles, melhorar o *site* do Laboratório Remoto de Automação de Processos Industriais do ISEL (Instituto Superior de Engenharia de Lisboa), com o intuito de permitir a adição de novas experiências de uma forma mais simplificada, melhorar/alterar as experiências existentes e alteração do painel HMI (Human Machine Interface) da Siemens.

Só foi possível atingir estes objectivos, com a aprendizagem das linguagens de programação utilizadas em páginas *web*, como também, através do aprofundar dos conhecimentos em automação, principalmente as funções para comunicar entre os autómatos.

1.3. Estrutura do Trabalho

O trabalho é dividido em quatro capítulos, sendo o primeiro a Introdução, no qual se faz uma breve introdução, fala-se das motivações, objectivos do trabalho e a sua estrutura.

O Capítulo 2, o Estado da Arte, faz uma breve descrição da Automação Industrial, dos Controladores Lógicos Programáveis e das linguagens normalizadas por eles utilizados. Além disso, também descreve os vários tipos de laboratórios existentes, principalmente os laboratórios remotos, e são dados alguns exemplos dos mesmos.

O Capítulo 3 começa por dar uma breve introdução do autómato, dos programas e linguagens de programação utilizados, dando ao leitor os conhecimentos básicos para perceber o desenvolvimento do trabalho. Após essa introdução é descrito como se limitou o acesso a um aluno de cada vez, como foi criada a nova experiência e outras alterações que foram mencionadas nos objectivos.

No Capítulo 4 são tiradas as conclusões referentes ao presente trabalho e são indicadas algumas recomendações para futuros trabalhos.

2. Estado da Arte

Sendo as experiências laboratoriais importantes na consolidação dos conhecimentos teóricos, principalmente nas áreas da engenharia como a automação, e com o aumento de alunos por experiência torna-se importante a criação de alternativas aos laboratórios físicos, sendo uma delas os laboratórios remotos.

O presente capítulo pretende fazer uma descrição da automação industrial, focando-se nos equipamentos utilizados e nas linguagens de programação, como também descrever os laboratórios remotos, indicar as suas vantagens e desvantagens, e indicar alguns exemplos dos mesmos.

2.1. Automação Industrial

A Automação Industrial é a tecnologia relacionada com a aplicação de sistemas mecânicos, eléctricos e electrónicos, apoiados em meios computacionais, na operação e controlo dos sistemas de produção. Em termos gerais, os objectivos da automação são: melhorar as condições de trabalho e de segurança, através da diminuição da intervenção humana em tarefas perigosas e repetitivas; aumentar a produtividade e competitividade; e aumentar da qualidade dos produtos e a diminuição do seu custo. Além dos aspectos da produtividade e qualidade, um objectivo da automação é a realização de operações impossíveis de controlar manual ou intelectualmente, por exemplo, montagem de peças de reduzidas dimensões, operações muito rápidas e coordenações complexas (Pinto, 2004).

Entre os dispositivos electrónicos que podem ser aplicados estão os computadores ou outros dispositivos capazes de efectuar operações lógicas, como controladores lógicos programáveis, microcontroladores, entre outros. Estes equipamentos em alguns casos substituem tarefas humanas ou realizam outras que o ser humano não consegue realizar.

A automação industrial visa principalmente a produtividade, qualidade e segurança dos processos. Num sistema típico toda a informação dos sensores é concentrada num controlador programável o qual define o estado dos actuadores, de acordo com o programa em memória (Borracha, 2012).

2.1.1. Controladores Lógicos Programáveis

O Controlador Lógico Programável (PLC – Programmable Logic Controller) nasceu dentro da indústria automóvel americana, especificamente na Hydronic Division da General Motors, em 1968. Este aparecimento deveu-se à grande dificuldade de mudança dos painéis de comando a cada alteração efectuada na linha de montagem, uma vez que tais mudanças implicavam altos gastos de tempo e dinheiro.

Nascia assim, um equipamento bastante versátil e de fácil utilização, que tem vindo a ser progressivamente melhorado, diversificando cada vez mais os sectores industriais e as suas aplicações.

Desde o seu aparecimento até hoje, muita coisa evoluiu nos controladores lógicos, como a variedade de tipos de entradas e saídas, o aumento da velocidade de processamento, a inclusão de blocos lógicos complexos para tratamento das entradas e saídas, e principalmente o modo de programação e a interface com o utilizador.

PLC é o tipo de controlador de maior aplicação na indústria, possui elevada capacidade de processamento, funciona em tempo real e é projectado para controlar múltiplas entradas e saídas, pois suporta grandes variações de temperatura e elevada resistência a vibração e impacto (Gutierrez & Pan, 2008).

A arquitectura base de um PLC (Figura 2.1) é constituída por várias partes e cada uma delas com a sua função específica:

- Processador Central (CPU – *Central Processing Unit*) que executa as instruções residentes em memória;
- Entradas/Saídas (identificadas pelas letras b e c, respectivamente) fazem a ligação dos sensores e actuadores com o CPU;
- Fonte de Alimentação externa (Figura 2.1 a) que faz a alimentação do CPU, e dos sensores e actuadores (normalmente de 24V).

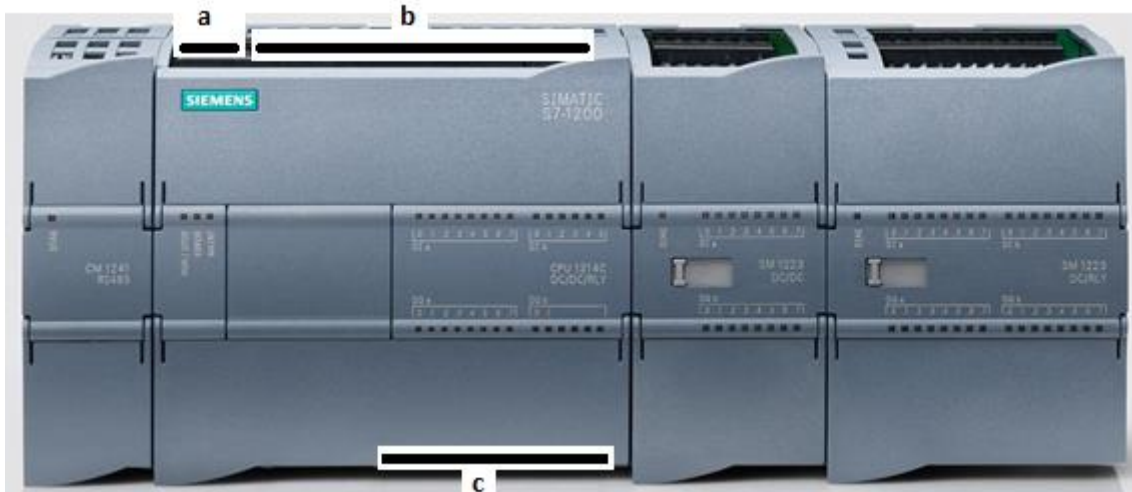


Figura 2.1 – PLC S7-1200 Siemens, 2014 (Siemens, 2014)

Os programas são geralmente construídos através de uma aplicação específica num computador e depois transferidos para o PLC via rede. Inicialmente, a linguagem de programação utilizada era proprietária, desenvolvida de forma isolada por cada fabricante, e não permitia interacção entre dispositivos de diferentes fabricantes (Gutierrez & Pan, 2008).

2.1.2. Linguagens Normalizadas

A crescente complexidade na programação dos PLCs motivou a necessidade da sua normalização. Sobre a direcção do IEC (*International Electrotechnical Commission*) foi publicada em Dezembro de 1993 (3ª edição publicada em Fevereiro de 2013), a norma IEC 61131-3 que define as linguagens de programação dos PLCs.

Com o objectivo de tornar a norma adequada a uma grande variedade de aplicações, foram definidas cinco linguagens de programação para PLCs (gráficas e textuais) (Tabela 2.1), constituindo uma forte base para uma boa programação dos PLCs (John & Tiegelkamp, 2007).

Tabela 2.1 – Linguagens de Programação segundo IEC 61131-3

Linguagens de Programação	
Linguagem Gráfica	Linguagem Textual
<i>Function Block Diagram (FBD)</i>	<i>Instruction List (IL)</i>
<i>Ladder Diagram (LD)</i>	<i>Structured Text (ST)</i>
<i>Sequential Function Chart (SFC)</i>	

Lista de Instruções – *Instruction List (IL)* – é uma linguagem de baixo nível composta por uma sequência de instruções, na qual se executa uma instrução por linha. A principal vantagem da IL advém do facto de dispor de um conjunto de funções básicas que podem ser utilizadas para construir uma aplicação complexa. Esta linguagem é muito útil na elaboração de pequenas aplicações onde a optimização do código é fundamental para garantir a performance da aplicação. Uma das desvantagens da IL prende-se com o tempo despendido na elaboração dos programas (ver Figura 2.2).

L I7 A I1 CU C1	L I7 A I5 A C1 K=3 S O2	L I7 A I2 R O2 R C1	L I7 A I2 CU C2	L I7 A I3 A C2 K=6 S O4	L I7 A I4 =T4 TB=2 K=2	L I7 A T4 R O4 R C2 EP
-----------------------	-------------------------------------	------------------------------	-----------------------	-------------------------------------	------------------------------------	------------------------------------

Figura 2.2 – Exemplo de Lista de Instruções

Texto Estruturado – *Structured Text (ST)* – é uma linguagem de alto nível, cuja sintaxe é parecida com a da linguagem Pascal. A linguagem ST pode ser empregue para realizar rapidamente aplicações que operam com uma grande diversidade de variáveis de diferentes tipos de dados, incluindo valores analógicos e digitais. A ST é uma linguagem adequada para a implementação de algoritmos matemáticos complexos. Deste modo, esta linguagem pode também ser utilizada para simplificar longos

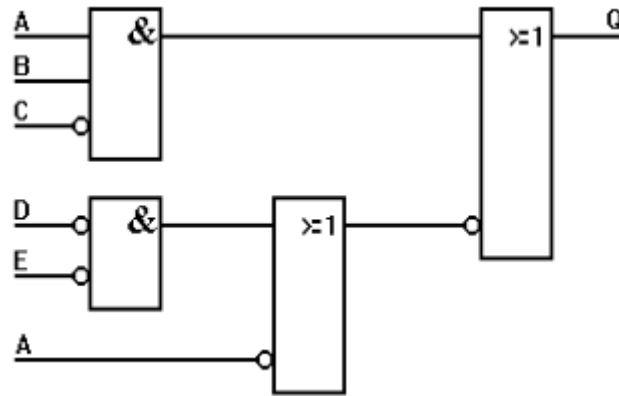


Figura 2.4 – Exemplo de Diagrama de Funções

Diagrama de Contactos – *Ladder Diagram (LD)* – é uma linguagem de programação gráfica que utiliza um conjunto de símbolos de programação normalizados. Os símbolos tradicionais do LD são as bobinas e os contactos. No entanto, o IEC permite a inserção de blocos de funções no programa, tornando a linguagem mais versátil. Esse passo tem em vista a migração progressiva dos tradicionais esquemas lógicos com relés para métodos mais avançados de diagramas de blocos de funções (ver Figura 2.5).

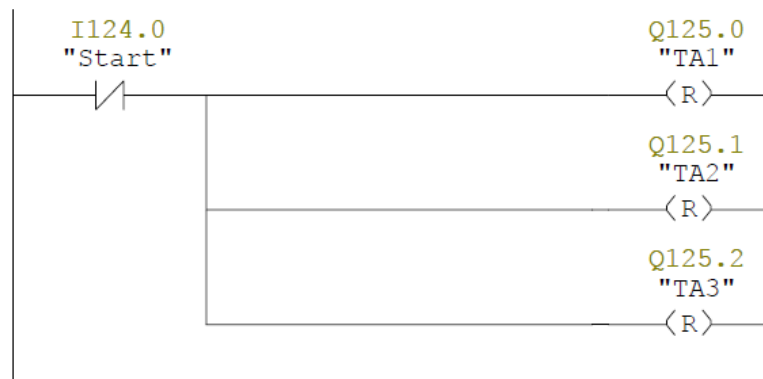


Figura 2.5 – Exemplo de Diagrama de Contactos

Gráfico Sequencial de Funções – *Sequential Function Chart (SFC)* – Antigamente chamada Grafcet (acrónimo do francês *Graphe Fonctionnel de Commande, Étapes Transitions*) é uma linguagem gráfica que proporciona uma representação do processo em forma de diagrama, particionando-se o programa em sequências lógicas. O gráfico é constituído por um conjunto de etapas e transições. As acções a executar são associadas às etapas e as condições a cumprir às transições. Como consequência das aplicações industriais funcionarem em forma de etapas, o SFC é a forma lógica de especificar e programar em mais alto nível um programa para PLCs (ver Figura 2.6).

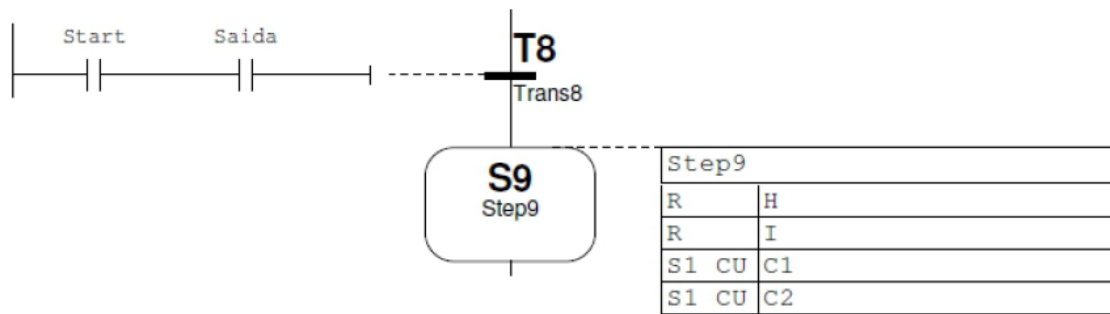


Figura 2.6 – Exemplo de Gráfico Sequencial de Funções

2.2. Laboratórios Remotos

Um estudo sobre a utilidade dos laboratórios remotos nas escolas secundárias fala do contexto que as experiências podem ter para influenciar os resultados da aprendizagem de um aluno. Os laboratórios remotos permitem que alunos e professores utilizem redes de alta velocidade, juntamente com câmaras, sensores e controladores, para realizarem experiências laboratoriais no laboratório físico, que está localizado remotamente. A pesquisa mostrou que, quando usado apropriadamente pode trazer uma série de benefícios, incluindo a partilha de recursos entre várias instituições e apoiar o acesso às instalações que seriam inacessíveis por razões de custo ou técnicas (Lowe, Newcombe, & Stumpers, 2013).

O laboratório remoto é um dos quatro tipos de laboratórios (Figura 2.7) existentes que podem ser classificados através dos seguintes critérios:

- Localização do Laboratório:
 - Igual à do Utilizador;
 - Diferente da do Utilizador;
- Tipo de Laboratório:
 - Físico;
 - Virtual.

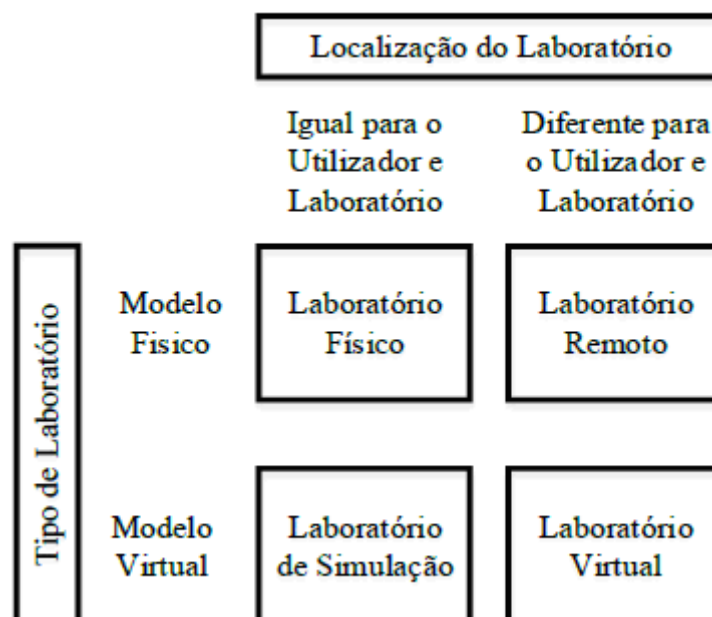


Figura 2.7 – Classificação dos vários tipos de Laboratórios (Gomes & Bogosyan, 2009)

O laboratório remoto é um laboratório real que é acedido remotamente, a localização física dos equipamentos a ensaiar estão deslocados do local onde o utilizador se encontra, ou seja, o local geográfico onde a experiência ocorre é diferente do local de quem realiza a mesma.

A arquitectura de um laboratório remoto conta com o paradigma cliente-servidor. Mesmo usando diferentes tecnologias, a maioria das soluções de laboratórios remotos contam com arquitecturas de programas simples, que permitem o controlo da experiência e a ligação com a rede, actuando como um servidor do laboratório (Gomes & Bogosyan, 2009). Na outra extremidade, o utilizador consegue a partir do próprio computador, recolher dados reais em tempo real, ou mesmo enviar instruções que são interpretadas e executadas no laboratório real (Leitão, 2006).

2.2.1. Vantagens e Desvantagens

Desde o início dos anos 90 que diversas universidades têm investido no desenvolvimento de laboratórios remotos de forma a expandirem os seus laboratórios tradicionais. Com o avanço das infra-estruturas tecnológicas, com o desenvolvimento da *web* e dos meios multimédia, cada vez mais se recorre a este tipo de laboratórios. As vantagens que oferecem são expostas de seguida (Rössler, Rausch, & Treytl, 2004):

- O aluno não necessita de estar fisicamente presente no laboratório para realizar a experiência;
- O aluno tem a possibilidade de aceder ao laboratório a qualquer hora do dia;
- O acesso a material dispendioso, a material que exige alto nível de segurança, ou a sistemas dedicados e específicos pode ser facilmente partilhado;
- O aluno pode facilmente repetir a experiência;
- Como em todas as disciplinas de carácter prático, a destreza de um aluno no laboratório deve ser uma das componentes da avaliação. Com os laboratórios de acesso remoto, os alunos podem melhorar o seu desempenho prático e preparar-se para testes práticos.

A substituição do laboratório tradicional por um laboratório remoto pode trazer algumas desvantagens, nomeadamente:

- O aluno não tem uma percepção real da experiência física;
- O aluno não interage tão facilmente com elementos de outros grupos;
- As experiências remotas estão já normalmente prontas para utilização imediata não incentivando o aluno a reflectir sobre os parâmetros de preparação – perda de sensibilidade de questões práticas.

Também como desvantagem, mas não menos relevante temos os orçamentos institucionais necessários para implementação dos sistemas. Embora os preços das infra-estruturas tenham vindo a baixar notavelmente, ainda se mantêm a níveis considerados elevados para algumas instituições. O esforço de programação e de manutenção dos sistemas é também ainda relativamente alto, mobilizando um grande número de recursos humanos.

2.2.2. Exemplos de Laboratórios Remotos

A seguir são indicados alguns exemplos de laboratórios remotos nas diferentes áreas da engenharia.

Faculdade de Engenharia da Universidade do Porto (FEUP):

Em 2010, foi proposto substituir a plataforma existente por uma baseada em LabVIEW, permitindo assim resolver os graves problemas de compatibilidades de sistemas informáticos, de gestão e de expansão.

Esse laboratório remoto foi desenvolvido utilizando a linguagem PHP para a aplicação alojada no servidor *Web* e JavaFX para os painéis de controlo das experiências. Usando o PostgreSQL como base de dados. Após essa renovação, a plataforma actual permite o agendamento de sessões (Figura 2.8), a criação, edição e remoção de experiências, utilizadores e grupos de utilizadores (Gonçalves, 2010).

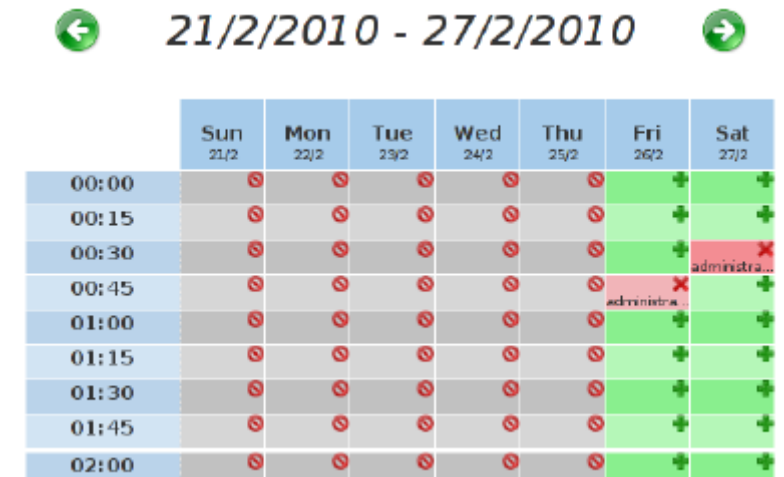


Figura 2.8 – Tabela de Agendamento de Sessões

Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCTUNL):

Devido à necessidade de criar um laboratório remoto capaz de representar e manipular 3 tanques e as válvulas instaladas no laboratório físico, recorreu-se ao Simulink e ao LabVIEW. Com o objectivo de facilitar a utilização por parte do utilizador, recorreu-se ao LabVIEW (Figura 2.9), mas por parte do supervisor foi necessário recorrer ao Simulink, que através do *Simulation Interface Toolkit* (SIT) permite fazer a ligação entre o Simulink e o LabVIEW (Silva, 2011).

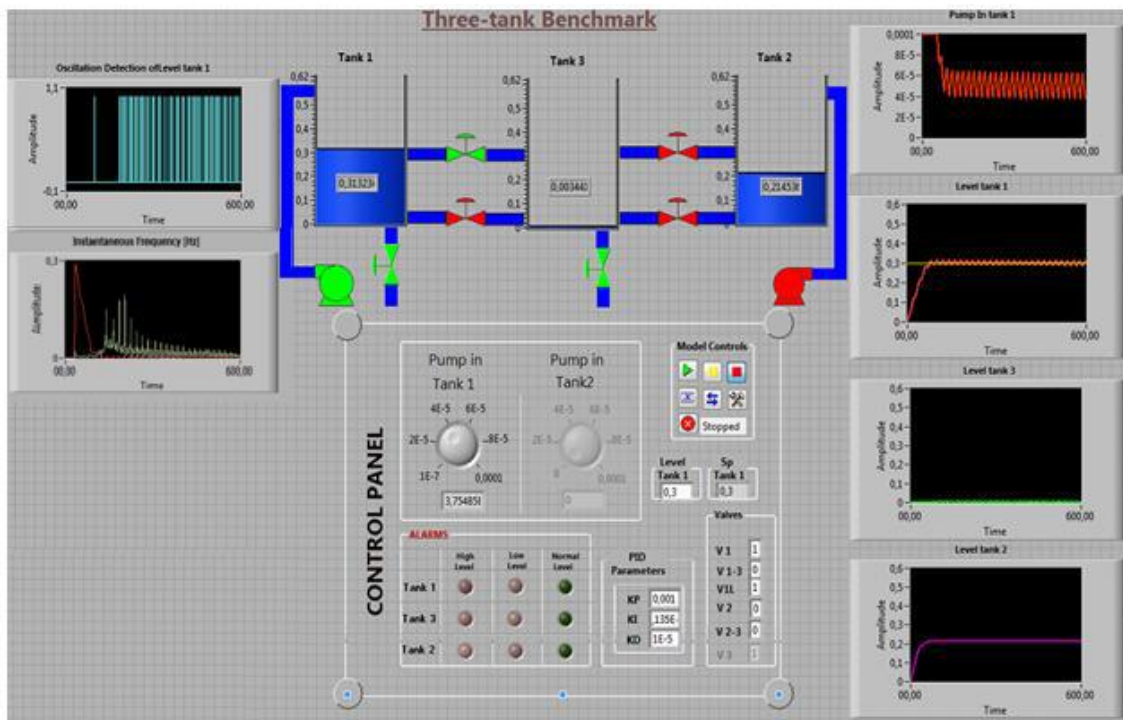


Figura 2.9 – Interface dos 3 tanques (Silva, 2011)

Instituto Superior de Engenharia do Porto (ISEP):

No ISEP, em vez de se criar um laboratório remoto comum, decidiu-se representar o equipamento que seria utilizado num ambiente imersivo (também chamado de ambiente 3D ou mundo virtual).

O ambiente 3D foi criado através do Open Wonderland, onde também foi criado um multímetro com um visor de seis dígitos igual ao que se encontrava instalado no laboratório (Figura 2.10), tendo como variáveis a tensão contínua e alternada, e a corrente contínua e alternada. E recorrendo ao Java Web Start por parte do utilizador, permitia aos alunos utilizarem o multímetro, através dos vários conectores destinados para ligações remotas existentes no aparelho (Costa, 2011).

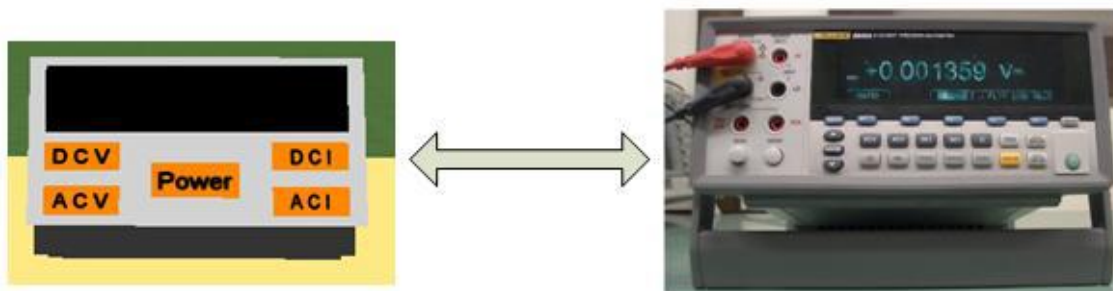


Figura 2.10 – Representação Virtual do Multímetro (Costa, 2011)

Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCTUNL):

Este laboratório ao contrário dos outros destina-se a controlar duas experiências, sendo elas:

- Um simulador de tráfego rodoviário
- Um monta-cargas industrial

Ambas as experiências usam PLCs da Schneider (Figura 2.11).

Este laboratório usa principalmente o WampServer, uma aplicação que fornece ferramentas que facilitam a criação de *sites* (conjunto de páginas *web*) dinâmicos. Recorrendo também ao MySQL um sistema de bases de dados e ao Apache, um servidor *Web* responsável por disponibilizar páginas, fotos ou qualquer outro tipo de objecto ao navegador de internet do utilizador.

Para a criação do *site* foi necessário utilizar a linguagem HTML, que define a forma na qual será apresentado o texto e outros elementos da página, mas além disso também se utilizou o JavaScript que torna possível incorporar com o HTML, alguma interactividade às páginas *web* criadas.



Na primeira experiência, como foi usada para controlar um simulador de tráfego rodoviário, foi instalada uma câmara, permitindo assim a visualização da experiência quase em tempo real. Na segunda experiência, que simula o monta-cargas industrial, foram usadas as seguintes variáveis: botão de subir, descer, stop, e fim de curso superior e inferior.

Também foi utilizado um gestor e servidor de conteúdos, que permite a partilha de ficheiros, criação de *blogs* e gestão de calendários. Este gestor e servidor de conteúdos é bastante semelhante ao Moodle, tendo a única vantagem de não estar dependente de serviços externos para funcionar (Borracha, 2012).

Welcome aaaa

[Logout](#)

Please Choose The LAB

Lab Name	Image	Description	Link
RemoteLab 1 TSX MICRO 3721		PLC - Modicon TSX Micro 3721 connected to a Traffic Light System Simulator with WebCam Visualization.	ENTER
RemoteLab 2 M340		PLC / PAC - M340 connected to a Elevator System Simulator with HMI.	ENTER

This is a password protected area only accessible to members.

Figura 2.11 – Escolha do Laboratório a usar (Borracha, 2012)

Instituto Superior Técnico (IST):

Neste laboratório não foi tanto substituir a plataforma existente, mas torná-la mais apelativa (Figura 2.12) para os alunos e, por isso, foi necessário torná-la mais fácil de utilizar, intuitiva e realista.

Sendo assim, partindo do sistema existente que utiliza o SAIA Project Manager, para entradas, saídas, *flags* e manipulação/leitura de registos, foi criada uma interface realista, através da utilização de fotos dos equipamentos instalados no laboratório (Figura 2.13) e através de algumas alterações à introdução de dados, tornou-se um laboratório remoto mais fácil de utilizar e intuitivo (Ribeiro, 2012).

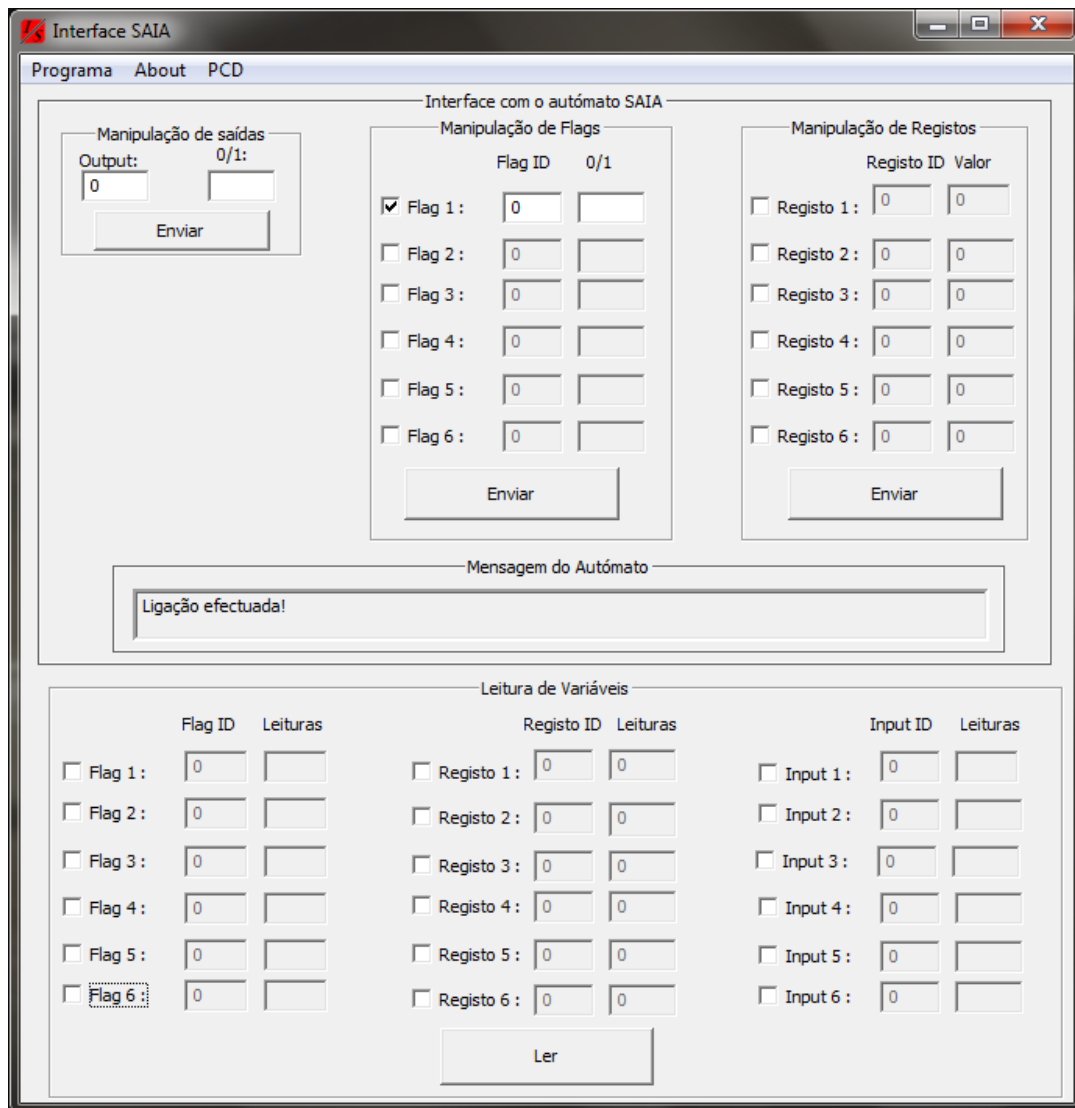


Figura 2.12 – Antes da alteração da Interface (Ribeiro, 2012)

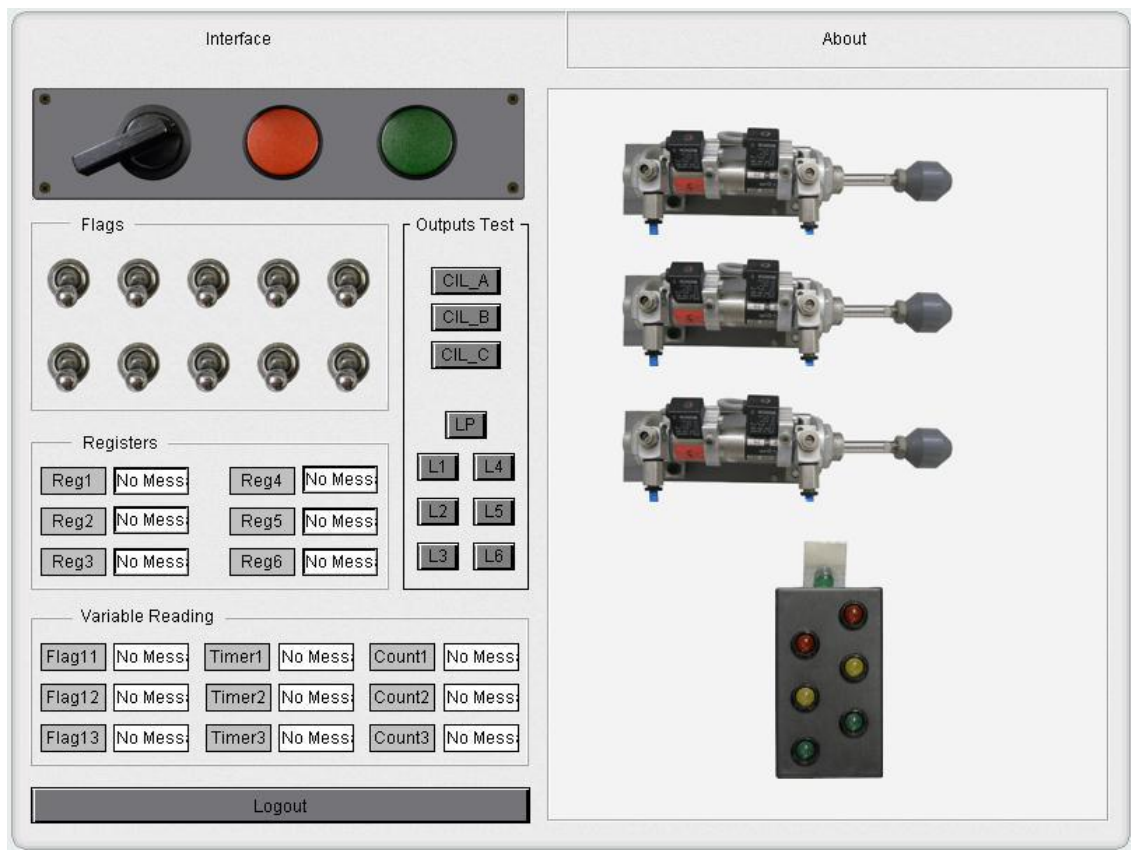


Figura 2.13 – Após alteração da Interface (Ribeiro, 2012)

Faculdade de Engenharia Mecânica e Electrotécnica da Universidade Veracruz (FIMEUV):

Recorrendo ao LabVIEW foi possível representar um processo industrial automatizado para a estampagem de peças (Figura 2.14).

Além de ter sido criado a representação do processo de estampagem, também foi criado um esquema dos cilindros e das válvulas. Tendo os alunos primeiro que fazer a “montagem” correcta das tubagens para poderem aceder ao processo (Figura 2.15). A montagem virtual das tubagens é feita através da indicação nos campos criados em LabVIEW, a que valor de entrada do cilindro deve corresponder o valor de saída da válvula e vice-versa.

Este processo é possível de aceder através de um *web browser* (navegador de internet), devido à *Web publishing tool* (ferramenta de publicação *web*), disponível no LabVIEW, que possibilita a criação de um ficheiro HTML que acede ao *front panel* criado (Villa-López et al., 2013).

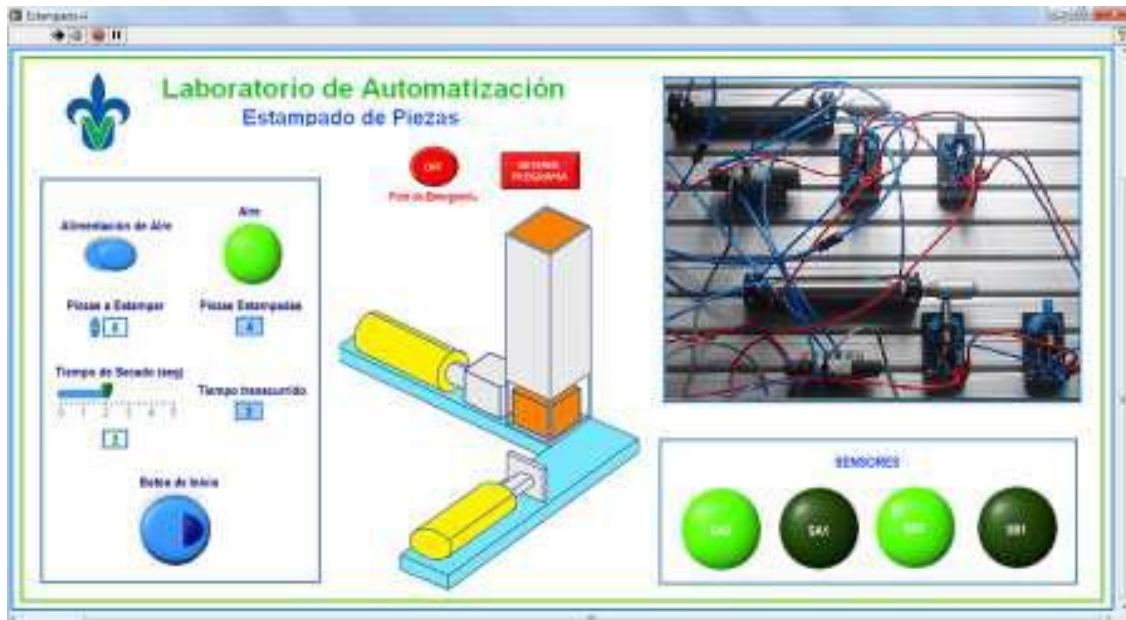


Figura 2.14 – Interface do Processo de Estampagem de Peças (Villa-López et al., 2013)

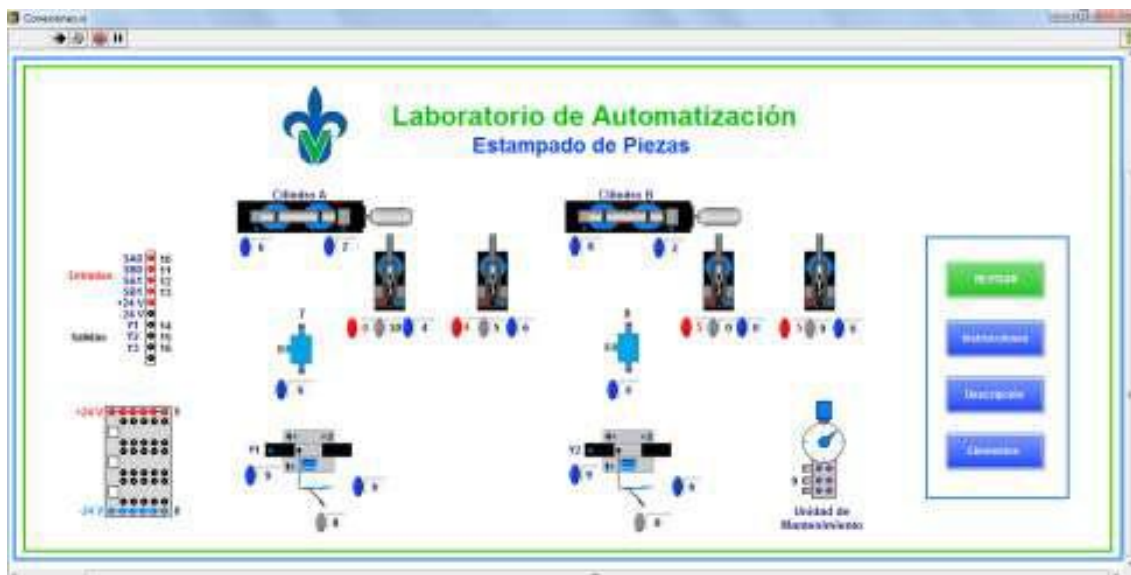


Figura 2.15 – Montagem das tubagens virtuais (Villa-López et al., 2013)

3. Laboratório Remoto de Automação de Processos Industriais do ISEL

Para a realização deste trabalho foram utilizadas várias ferramentas e, por isso, antes de expor o desenvolvimento do mesmo decidiu-se que seria essencial explicar qual o PLC utilizado, quais os programas necessários e quais as suas diferentes funções, como também dar uma breve introdução às linguagens de programação (HTML, PHP, JavaScript e SQL) utilizadas.

Após essas introduções é feita uma contextualização do trabalho, tanto em termos de arquitectura da rede (servidores, PLCs, *webcam*) como do *site* do laboratório remoto. Com todos estes conhecimentos expostos então é explicado todo o desenvolvimento do presente documento.

3.1. PLC Siemens S7-1200

O *design* compacto do PLC S7-1200, uma configuração flexível e um poderoso conjunto de instruções tornam-no numa solução perfeita para controlar uma ampla variedade de aplicações.

A CPU combina um microprocessador, uma fonte de alimentação integrada, circuitos de entrada e de saída, uma ligação PROFINET, e entradas analógicas, tudo numa caixa compacta. Permite monitorizar e controlar os dispositivos na sua aplicação de acordo com a lógica do programa, podendo este conter lógica booleana, contadores, temporizadores, operações matemáticas complexas e comunicações com outros dispositivos inteligentes (Siemens, 2012).

Na criação da experiência descrita neste trabalho foram utilizados dois PLCs S7-1214C AC/DC/Rly, 6ES7214-1BG31-0XB0, estando algumas características indicadas na seguinte tabela:

Tabela 3.1 – Características do PLC S7-1214C AC/DC/Rly

Fonte de Alimentação	120/230 V AC
Memória	
Memória de Trabalho Integrada	75 KB não expansível
Memória de Carregamento	4 MB
Memória Retentiva	10 KB
Tempo de Processamento da CPU	
1 bit	0.085 μ s / instrução
<i>Word</i> (16 bits)	1.7 μ s / instrução
Número de Blocos	Limitado pela RAM
Capacidade <i>Inputs/Outputs</i>	1024 <i>bytes</i> /1024 <i>bytes</i>
Entradas Digitais	14; 24 V DC
Saídas Digitais	10 por relé
Entradas Analógicas	2; 0 – 10V
Interface de Comunicação	PROFINET
Comunicação Ethernet	
TCP/IP	Sim
ISO-on-TCP	Sim
UDP	Sim
Web Server	
Suporta	Sim
<i>User-defined websites</i>	Sim

3.2. Programas Utilizados

3.2.1. WampServer

O WampServer é usado para instalar rapidamente o Apache HTTP Server, o phpMyAdmin e a base de dados MySQL. O Apache Server é usado para tornar o computador acessível via internet, um servidor, sendo compatível com o protocolo HTTP versão 1.1.

O programa phpMyAdmin é uma aplicação *web*, desenvolvida em PHP, para administração da base de dados MySQL. A partir deste sistema é possível criar e remover bases de dados; criar, remover e alterar tabelas; inserir, remover e editar campos. O MySQL é um sistema de gestão de base de dados, que utiliza a linguagem SQL (*Structured Query Language*) como interface (Bringing MySQL to the web, 2014).

3.2.2. Notepad++

Notepad++ é um editor de texto que permite editar mais de 50 tipos de linguagens usadas em programação, tais como (Notepad++, 2014):

- C, C++;
- Fortran;
- HTML;
- Java, Javascript;
- Matlab, MS-DOS;
- Pascal, Perl, PHP, Python;
- Shell, SQL;
- Visual Basic;
- XML.

Este programa foi utilizado principalmente na criação das páginas PHP, devido à sua simplicidade e porque as páginas em questão contêm poucos elementos visuais, não sendo necessário recorrer a programas mais avançados. Na Figura 3.1 está um exemplo de uma secção de linguagem PHP vista no Notepad++.

```

$session_life = time() - $_SESSION['timeout'];
if($session_life > $inactive) {
    header("Location: logout.php");
    exit;
} else if ($rowout ==1){
    header("Location: logout.php");
    exit;
}

```

Figura 3.1 – Exemplo de PHP no programa Notepad++

3.2.3. Dreamweaver

O Dreamweaver é um editor visual WYSIWYG (*What You See Is What You Get*), que tem características semelhantes ao Notepad++, mas está mais desenvolvido para a criação de páginas *web* com vários elementos visuais, tornando-se um programa bastante útil para páginas que contenham HTML e JavaScript.

As alterações feitas podem ser visualizadas e testadas directamente no próprio programa, permite a divisão da janela de visualização em duas, uma com as linguagens de programação e a outra com a página que será visualizada através do *web browser*.

As seguintes linhas de código em HTML dão origem à Figura 3.2 que é directamente visualizada no programa:

```

<table border="0">
  <tr>
    <td width="247"><center>
      <p></p>
      <p><a href="http://10.68.13.171/isel/login1.php"
target="_blank">CILINDROS</a></p>
    </center></td>
    <td width="247"><center>
      <p><a
href="http://10.68.13.171/isel/login2.php" target="_blank"></a>
      </p>
      <p>
      <a href="http://10.68.13.171/isel/login2.php" target="_blank">PÓRTICO</a>
      </center></td>
  </tr>
</table>

```

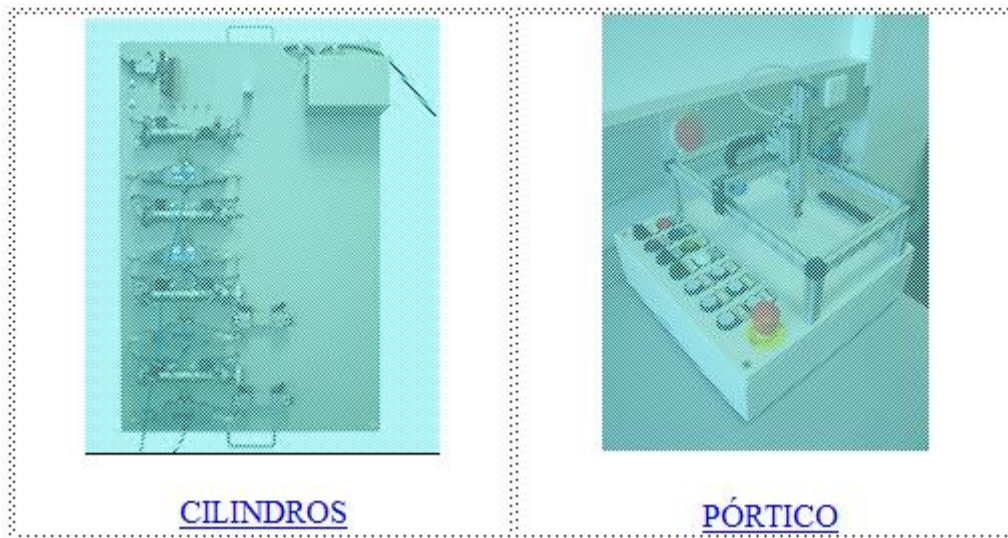


Figura 3.2 – Exemplo de HTML no programa Dreamweaver

3.2.4. TIA Portal

O TIA Portal (*Totally Integrated Automation Portal*) é o programa de programação utilizado para configurar o PLC Siemens S7-1200, fornecendo um ambiente amigável para desenvolver toda a programação do controlador, configurar a visualização do HMI e configuração da comunicação em rede.

O STEP 7 é o componente do programa de programação e configuração do TIA Portal para os diferentes PLCs da Siemens existentes na biblioteca. Além do STEP 7, este inclui também o WinCC para programação e configuração dos HMI, entre outros.

Projectos pré-existent, mesmo de versões anteriores do TIA Portal ou do STEP 7, podem ser facilmente integrados num mais actual. Nesta gama de PLCs as linguagens de programação utilizadas são o LD (*Ladder Diagram*), FBD (*Function Block Diagram*) e ST (*Structured Text*), sendo que nas gamas, dos S7-300, 400 ou 1500 também permitem IL (*Instruction List*) e SFC (*Sequential Function Chart*). Na Figura 3.3 está um exemplo da linguagem de programação LD usada no TIA Portal (Siemens, 2012).

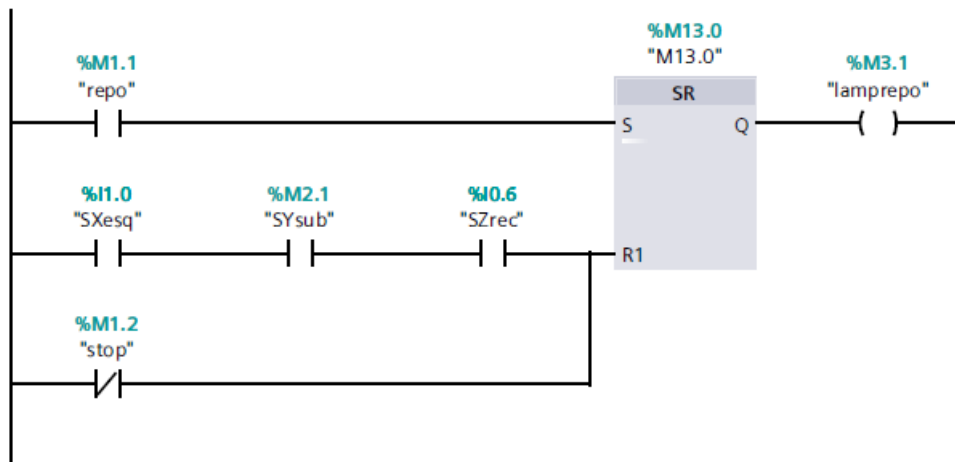


Figura 3.3 – Exemplo de *Ladder Diagram* no programa TIA Portal

3.2.5. ThinVNC

O ThinVNC permitir aos utilizadores aceder a computadores remotamente, partilhar o ambiente de trabalho ou assumir o controlo total do computador. Não requer Flash, Java, ActiveX, Silverlight ou qualquer outra configuração do lado do utilizador final e pode ser usado a partir de praticamente qualquer dispositivo com *web browser* (Cybele Software, 2014).

Na Figura 3.4 está representado o menu do programa ThinVNC que permite acesso remoto ao computador existente no LabAPI – ISEL.



Figura 3.4 – Menu de Acesso ao Computador Remoto

3.2.6. YAWCAM

O Yawcam é um programa de *webcam* para Windows. Que permite armazenar as imagens capturadas como ficheiros de imagem. Da mesma forma, podem ser enviados para um servidor HTTP. Permite também *streaming* de vídeo usando um endereço IP, podendo configurar-se e alterar-se a apresentação da página *web* correspondente (Yawcam, 2014).

3.3. Linguagens de Programação

3.3.1. HTML

HTML (*HyperText Markup Language*) é uma linguagem de programação utilizada para produzir páginas *web*. Esta linguagem caracteriza-se por ser simples e fácil de aprender. É na sua essência constituída por etiquetas (*tags*).

As *tags* não são mais do que palavras-chave delimitadas por parêntesis angulares como “<html>”. Estas vêm normalmente em pares como e </ b>, sendo que a primeira *tag* é a de início, e a segunda é a final ou de finalização do elemento HTML, também podem ser chamadas de *tags* de abertura e fecho.

Os nomes “*Tags HTML*” e “Elementos HTML” são frequentemente usados para descrever a mesma coisa. Mas, um elemento HTML é tudo entre a *tag* de início e final, incluindo as *tags*.

Os *web browsers* como o Google Chrome, Internet Explorer, Firefox, Safari lêem e descodificam os documentos HTML. O navegador não exhibe as *tags* HTML, mas utiliza-as para determinar como o conteúdo da página HTML deve ser apresentado.

Os elementos HTML formam os blocos de construção de todos os *sites*. O HTML permite que imagens e objectos possam ser inseridos e utilizados de forma interactiva e também permite inserir *scripts* escritos em JavaScript que irão afectar a interactividade das páginas *web*.

Exemplo:

Username:
Password:

Note: The characters in a password field are masked (shown as asterisks or circles).

Figura 3.5 – Exemplo de Linguagem HTML

Na Figura 3.5 podemos ver um exemplo de uma página *web*, que contém dois inputs e uma nota, tudo criado através dos seguintes elementos HTML:

```
<html>
<body>

<form action="">
Username: <input type="text" name="user"><br>
Password: <input type="password" name="password">
</form>

<p><b>Note:</b> The characters in a password field are masked (shown as asterisks or
circles).</p>

</body>
</html>
```

3.3.2. PHP

A linguagem PHP (*Hypertext Preprocessor*) é interpretada pelo servidor que cria a página *web* resultante, sendo que os comandos PHP podem ser inseridos directamente num documento HTML.

Os comandos só serão processados pelo servidor se estiverem delimitados pelos seguintes delimitadores: “<?php” e “?>”. Sendo que os comandos PHP serão inseridos entre os dois delimitadores.

A linguagem PHP, ao contrário da linguagem HTML, permite a criação de variáveis que são utilizadas em todo o tipo de situações, sendo identificadas através do seguinte símbolo: “\$”. Além disso, só é possível comunicar com a base de dados combinando os comandos PHP com a linguagem SQL.

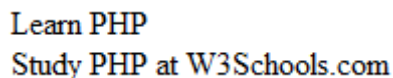
Exemplo:

Como se pode ver nas seguintes linhas de código os comandos PHP estão inseridos numa página HTML (“<html>” e “</html>”) e os próprios comandos estão delimitados (“<?php” e “?>”). Os seguintes comandos criam duas variáveis e representa-as na página *web*, como se pode ver na Figura 3.6:

```
<html>
<body>

<?php
$txt1="Learn PHP";
$txt2="W3Schools.com";
echo $txt1;
echo "Study PHP at $txt2";
?>

</body>
</html>
```



Learn PHP
Study PHP at W3Schools.com

Figura 3.6 – Exemplo de Linguagem PHP

3.3.3. JavaScript

O JavaScript é utilizado principalmente para interagir com os Objectos das páginas, sendo que em JavaScript quase tudo é um objecto, seja uma imagem, um botão, uma hiperligação, etc. Alguns exemplos simples da utilização de JavaScript são:

- Carregar novos conteúdos na página ou submeter informação ao servidor sem recarregar a página;
- Animação de elementos da página, desaparecer/aparecer, aumentar/diminuir, mover, etc.;
- Conteúdo interactivo, como por exemplo, vídeo e áudio.

Exemplo:

```
<html>
<head>
<script>
```

```

function open_win()
{
window.open("http://www.w3schools.com");
}
</script>
</head>

<body>
<form>
<input type="button" value="Open Window" onclick="open_win()">
</form>
</body>

</html>

```

Semelhante ao exemplo da linguagem PHP, os comandos JavaScript estão inseridos numa página HTML e neste caso delimitados pelos delimitadores do JavaScript (“<script>” e “</script>”). Este exemplo, cria um botão através da linguagem HTML e quando é clicado irá correr uma função criada pelo JavaScript, sendo que essa função abre um novo separador no *web browser* com o endereço indicado.

3.3.4. SQL

O SQL é uma linguagem de programação destinada unicamente para gerir informação contida numa base de dados. A informação da base de dados é gerida através de *queries* (pedidos), sendo alguns deles:

- SELECT – extrai informação da base de dados;
- UPDATE – actualiza a informação da base de dados;
- DELETE – elimina informação da base de dados.

Se a base de dados estivesse a ser gerida directamente bastaria o recurso aos *queries*, mas quando é utilizada numa página *web* é necessário recorrer ao PHP, inicialmente para fazer a ligação à base de dados e posteriormente para permitir uma interacção entre a página *web* e a base de dados.

O seguinte comando PHP liga a página *web* à base de dados, sendo que:

- host – o nome do anfitrião ou o endereço IP;
- username – o nome do utilizador;
- password – a palavra-passe do utilizador.

```
<?php
mysql_connect(host, username, password);
?>
```

Após feita a ligação basta inserir os delimitadores do PHP e entre os delimitadores escrever os comandos PHP e os *queries*, como se pode ver nas linhas seguintes:

```
<?php
$on = mysql_query("SELECT * FROM utilizadores WHERE online=1");
$rowon = mysql_num_rows($on);
$num = mysql_query("SELECT numero FROM utilizadores WHERE online = 1");
$numero = mysql_fetch_array($num);
?>
```

3.4. Arquitectura do LabAPI – ISEL

O LabAPI – ISEL é composto por duas componentes, uma componente física (com a arquitectura da rede) e uma virtual, o *site* do LabAPI – ISEL. A componente física é composta por servidores, computadores, PLCs e outros elementos periféricos. Enquanto a parte virtual são apenas páginas *web*, que permitem a comunicação entre os vários elementos da parte física.

A arquitectura da rede, como se pode ver na Figura 3.7, é constituída por um servidor, um computador (que serve de ambiente de trabalho remoto), três PLCs e uma *webcam*. Esta rede pode ser acedida por duas vias, interna ou externa, se for interna o utilizador irá aceder directamente ao servidor, caso contrário o utilizador terá de aceder através da VPN do ISEL (<http://www.isel.pt/pInst/eISEL/Portal/AUR.html>) e só depois poderá aceder ao servidor como um utilizador interno.

O servidor é onde se encontra o *site* do LabAPI – ISEL (<http://10.68.13.171/isel/menu.php>) e a base de dados onde será gerido o acesso dos utilizadores às páginas *web*. É a partir do *site* que os utilizadores têm acesso ao ambiente de trabalho remoto, à *webcam* e aos PLCs. A estrutura do *site* está representada na Figura 3.8, onde também se indica a localização das páginas *web* nos respectivos elementos físicos. As páginas indicadas na Figura 3.8 poderão ser visualizadas no Anexo A.

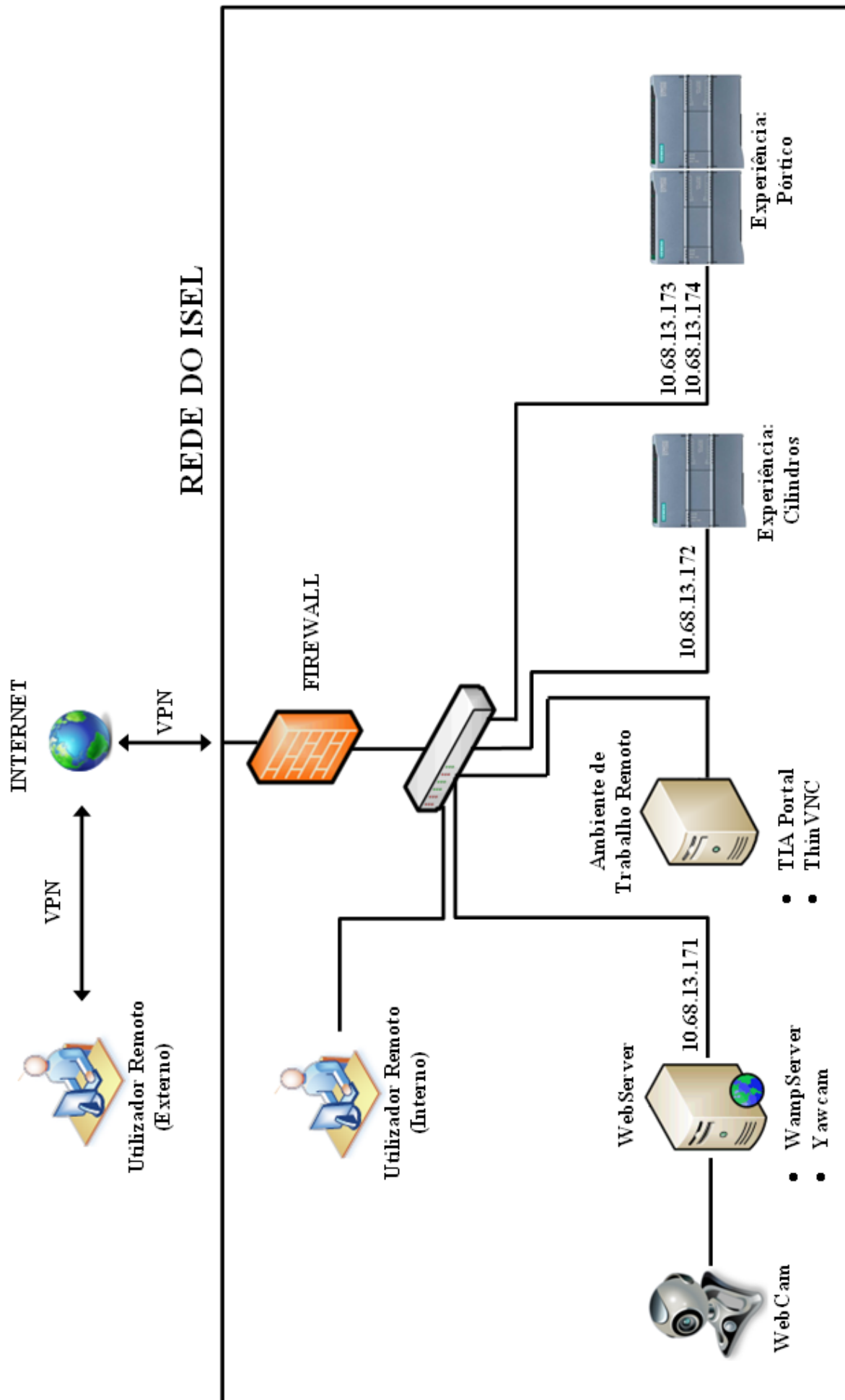


Figura 3.7 – Arquitectura da Rede do LabAPI – ISEL

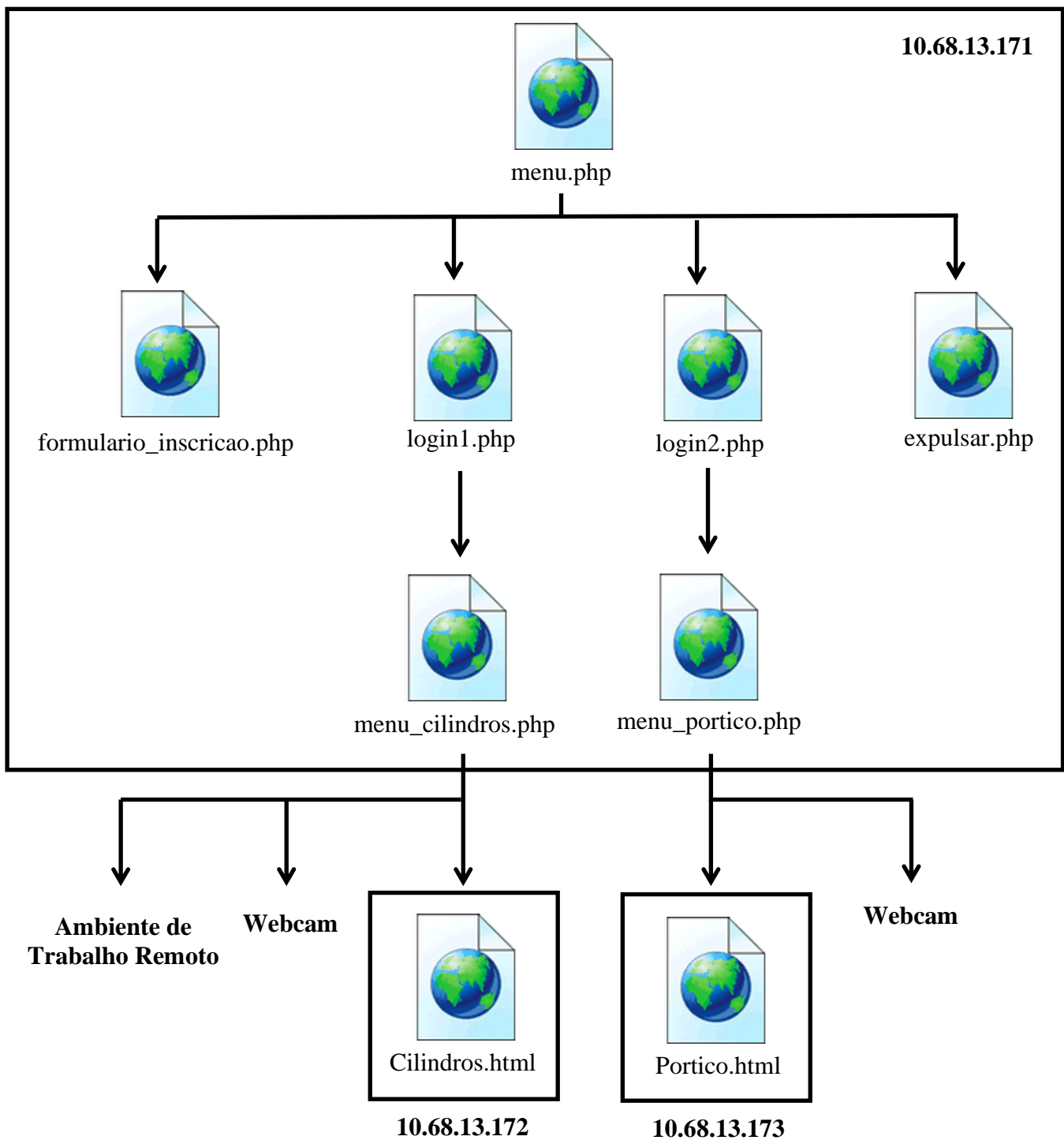


Figura 3.8 – Estrutura do Site do LabAPI – ISEL ¹

¹ **Nota:** Apenas estão representadas as páginas *web* e elementos físicos que o utilizador tem acesso, pois existem outras páginas que fazem operações a que o utilizador não tem acesso, tais como, comunicar com a base de dados ou o envio de formulários

Todas as páginas *web* neste trabalho foram criadas com o objectivo de funcionar nos três principais *web browsers*: Google Chrome, Mozilla Firefox e Internet Explorer, pois segundo dados de Fevereiro de 2014, estes *web browsers* são os mais utilizados mundialmente, representando um valor total de 92,6%, como se pode ver pela seguinte tabela (Browser Statistics, 2014).

Tabela 3.2 – Utilização de *web browsers* em Fevereiro de 2014

2014	Internet Explorer	Mozilla Firefox	Google Chrome	Safari	Opera
Fevereiro	9.8 %	26.4 %	56.4 %	4.0 %	1.9 %

Também se decidiu que todos os elementos da janela fossem possíveis de visualizar sem recorrer à *scrollbar*, por isso, foi preciso ter em conta as resoluções actualmente utilizadas nos monitores. Sendo assim, através da Tabela 3.3 (Browser Display Statistics, 2014) decidiu-se que as páginas seriam criadas para uma resolução de 1024x768, pois monitores que tenham uma resolução superior não terão necessidade de utilizar a *scrollbar* e os que têm uma resolução inferior representam apenas 1% dos utilizadores de internet. Mas, se por acaso o utilizador tiver uma resolução inferior à utilizada basta reduzir o tamanho da página no *web browser*.

Tabela 3.3 – Resoluções utilizadas em Janeiro de 2014

2014	Superior	1920x1080	1366x768	1280x1024
Janeiro	34%	13%	31%	8%
	1280x800	1024x768	800x600	Inferior
Janeiro	7%	6%	0.5%	0.5%

3.5. Limitar o acesso a um Aluno de cada vez

Para limitar o acesso a um aluno de cada vez é preciso adicionar uma coluna à base de dados, que neste caso foi chamada de ‘online’, que tem como valor predefinido ‘0’ (ver Figura 3.9). Na página em que são verificados os dados de acesso, irá alterar-se esse

valor para '1', no caso de os dados estarem correctos. E quando o aluno sair, a página de *logout* irá repor esse valor novamente a '0'. Além da coluna 'online' também foi criada a coluna 'exp' que tem '0' como valor predefinido e que após o *login* passará a ter um valor correspondente à experiência acedida (1 se for a primeira experiência, 2 se for a segunda, ...). Sendo que esse valor volta novamente a ser '0' após o *logout*. Os supervisores (ex: professor) têm a coluna 'online' e 'exp' sempre vazia.

O subcapítulo seguinte irá explicar mais em pormenor o referido atrás.

primeiro_nome	ultimo_nome	numero	turma	email	palavra_pass	id	online	exp
Mário	Mendes	1418		mmendes@dem.isel.pt		1		
Luis	Martins	32958		root		2		
Rui	Matias	35569		a35569@alunos.isel.pt		3		
A	A	1		matias.rui@sapo.pt	1	10	0	0
B	B	2		rui.miguel.matias@iol.pt	2	11	0	0

Figura 3.9 – Base de Dados de Alunos do LabAPI – ISEL

3.5.1. Autenticação do Utilizador

Devido à existência de várias experiências, determinou-se que se deveria identificar as experiências, e sendo assim a experiência já existente (Cilindros) seria identificada com o valor 1 e a experiência descrita no subcapítulo 3.6, com o valor 2.

A página *login2.php* funciona como um formulário em que são inseridos os dados de acesso, nº de aluno e password, que serão submetidos à *userauthentication2.php*, sendo nessa página que será alterado o valor da coluna 'online' da base de dados para 1 e da coluna 'exp' para 2.

Primeiro é preciso guardar os dados inseridos em variáveis reconhecidas em linguagem PHP ('\$numero' e '\$palavra_pass'):

```
<?php
$numero=$_POST['numero'];
$palavra_pass=$_POST['palavra_pass'];
```

Após guardados esses valores é preciso questionar a base de dados, através do comando *mysql_query*, várias situações com recurso à instrução *SELECT * FROM*.

Sendo que, no caso em particular deste trabalho foi necessário criar variáveis para cinco situações:

1. Os dados introduzidos estão correctos e o aluno está offline (online=0):

```
$sql = mysql_query("SELECT * FROM utilizadores WHERE numero = '$numero' and  
palavra_pass = '$palavra_pass' and online='0'");  
$row = mysql_num_rows($sql);
```

2. Os dados estão correctos:

```
$pass = mysql_query("SELECT * FROM utilizadores WHERE numero = '$numero'  
and palavra_pass = '$palavra_pass'");  
$rowpass = mysql_num_rows($pass);
```

3. Existe algum utilizador online (online=1), na experiência 2 (exp=2):

```
$son = mysql_query("SELECT * FROM utilizadores WHERE online=1 and exp=2");  
$rowon = mysql_num_rows($son);
```

4. Os dados introduzidos estão correctos e a coluna 'online' está vazia (online=' '):

```
$super = mysql_query("SELECT * FROM utilizadores WHERE numero = '$numero'  
and palavra_pass = '$palavra_pass' and online=' '");  
$rowsuper = mysql_num_rows($super);
```

5. O utilizador que está a aceder está na experiência 2 (exp=2):

```
$exp = mysql_query("SELECT * FROM utilizadores WHERE numero = '$numero' and  
exp=2");  
$rowexp = mysql_num_rows($exp);
```

Após guardadas as variáveis é preciso verificar em qual das situações se enquadra o utilizador que fez *login*. O aluno tem o número, password correctos, está *offline*, não existe nenhum aluno ligado então pode fazer *login* ('online' passa a ser 1 e 'exp' a 2).

Além disso, também é guardado o tempo em segundos desde da Era Unix (1-Janeiro-1970, 00:00:00 GMT) através da função `time()`, este valor será explicado quando se estiver a descrever as alterações feitas à página `menu_portico.php`:

```
if($row == 1 and $rowon == 0) {  
    session_start();  
    $_SESSION['numero']=$_POST['numero'];
```

```

$_SESSION['palavra_pass']=$_POST['palavra_pass'];
mysql_query("UPDATE utilizadores SET online='1' WHERE
numero='$numero'");
mysql_query("UPDATE utilizadores SET exp='2' WHERE
numero='$numero'");
$_SESSION['timeout']=time();
echo "<center>LOGIN EFETUADO COM SUCESSO!</center>";
echo "<script>loginsuccessfully()</script>";

```

O número, password estão correctos e a coluna ‘online’ está vazia então é um supervisor. O supervisor pode fazer *login* mesmo com um aluno *online*, e os alunos também podem fazer *login* após um supervisor ter feito *login*, pois como a valor da coluna ‘online’ do supervisor está vazio é considerado como *offline*:

```

} else if ($rowsuper == 1) {
    session_start();
    $_SESSION['numero']=$_POST['numero'];
    $_SESSION['palavra_pass']=$_POST['palavra_pass'];
    $_SESSION['timeout']=time();
    echo "<center>LOGIN EFETUADO COM SUCESSO!</center>";
    echo "<script>loginsuccessfully()</script>";

```

O aluno introduziu o número, password correctamente e está *offline*, mas já existe um aluno ligado nesta experiência. Neste caso irá aparecer uma mensagem que diz ‘JÁ EXISTE UM ALUNO LIGADO!’ e o aluno voltará novamente à página de *login*:

```

} else if ($row == 1 and $rowon == 1){
    echo "<center>JÁ EXISTE UM ALUNO LIGADO!</center>";
    echo "<script>loginfailed()</script>";

```

Para prevenir esta situação é indicado na página de menu se existe alguém ligado e em que experiência, mas como só indicar não impede o aluno de tentar aceder, foi necessário criar esta situação de acesso (a indicação na página de menu será explicada no subcapítulo 3.7.2).

Existe a situação em que, o aluno fechou o separador ou a janela, antes de fazer *logout* e quando tenta aceder novamente irá aparecer-lhe a mensagem que ‘JÁ ESTÁ LIGADO!’:

```

} else if ($row == 0 and $rowpass == 1 and $rowexp == 1){
    echo "<center>JÁ ESTÁ LIGADO!</center>";
    echo "<script>loginfailed()</script>";

```

Esta situação em particular é muito problemática, pois apesar de existirem dois eventos (onunload e onbeforeunload) que permitem detectar o fecho de um separador ou da janela, não funcionam correctamente nos três *web browsers*. E como foi referido no subcapítulo anterior todos os componentes das páginas teriam de funcionar em todos os *web browsers* e sendo assim optou-se por não se utilizar os eventos referidos.

Por isso, foi criada uma página, que permite o aluno expulsar-se a si próprio, que será nada mais que fazer o *logout* após ter fechado a página da experiência (a página em questão será explicada no subcapítulo 3.7.3).

Apesar das precauções feitas serem aceitáveis para este problema não impede que o aluno faça *logout* e permaneça com o separador aberto de uma das experiências, podendo continuar a controlar a experiência mesmo estando *offline*.

A situação seguinte foi criada com o objectivo de se poderem controlar as duas experiências paralelamente e por alunos diferentes. Por isso, é preciso determinar se o aluno introduziu os dados correctamente, se está *online* e não está ligado á experiência que tenta aceder:

```
} else if ($row == 0 and $rowpass == 1 and $rowexp == 0){  
    echo "<center>JÁ ESTÁ LIGADO NOOUTRA EXPERIÊNCIA!</center>";  
    echo "<script>loginfailed()</script>";
```

A última situação é o aluno ou o supervisor terem o número e/ou password incorrectos:

```
} elseif ($rowpass == 0){  
    echo "<center>PALAVRA PASSE INCORRETA OU Nº NÃO  
REGISTADO!</center>";  
    echo "<script>loginfailed()</script>";  
}  
?>
```

Na Figura 3.10 está representado um fluxograma do processo descrito ao longo deste subcapítulo.

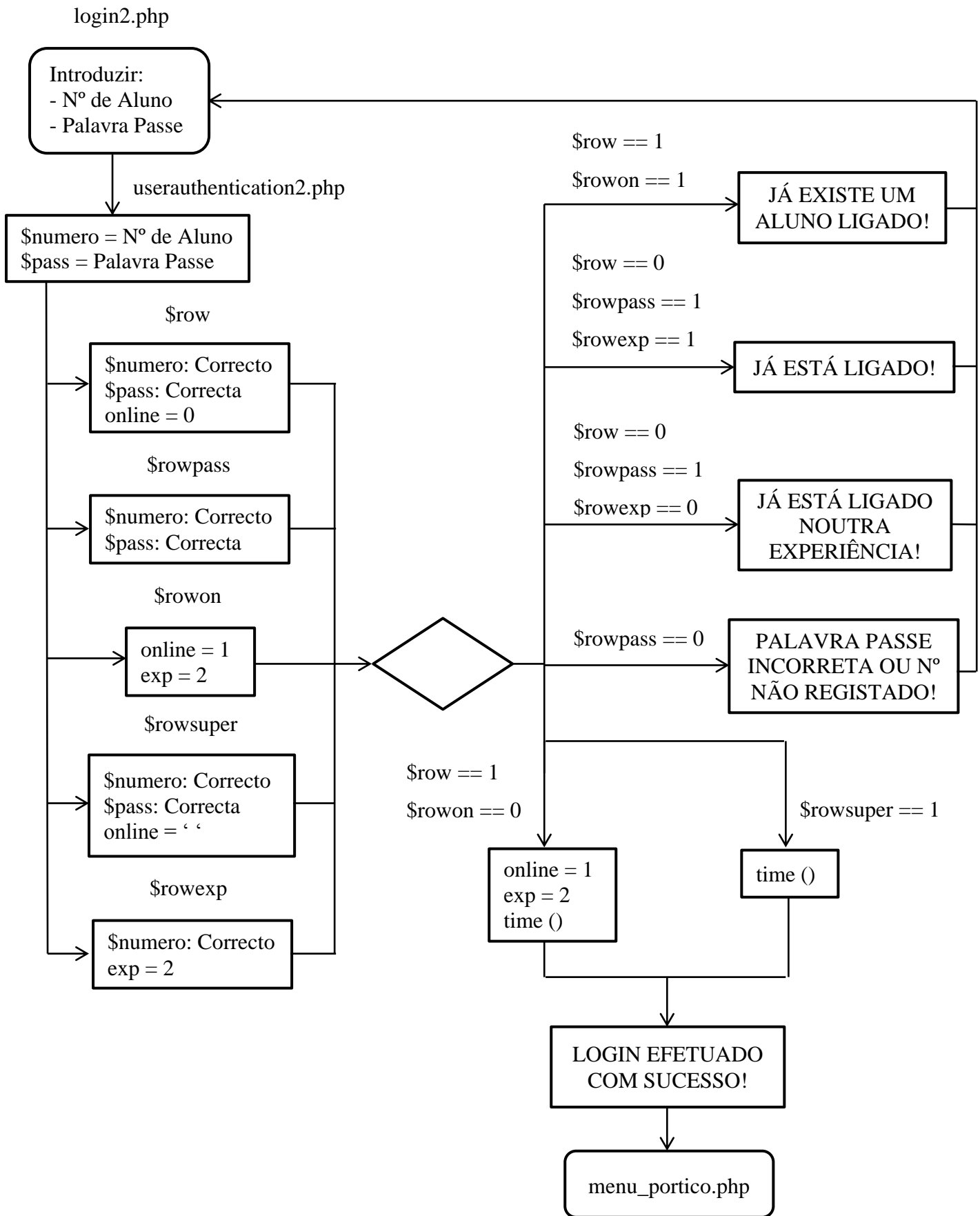


Figura 3.10 – Fluxograma da Autenticação do Utilizador

3.5.2. Menu da Experiência

Esta página primeiro verifica se está um supervisor *online* ou um aluno, se for um supervisor a variável '\$inactive' terá o valor 24h (24*60*60s) caso contrário terá 1h, estes tempos são usados para determinar o tempo da sessão:

```
<?php
$numero=$_SESSION['numero'];
$palavra_pass=$_SESSION['palavra_pass'];
$super = mysql_query("SELECT * FROM utilizadores WHERE numero = '$numero'
and palavra_pass = '$palavra_pass' and online=' '");
$rowsuper = mysql_num_rows($super);
if ($rowsuper == 1){
$inactive = 24*60*60; //ESTÁ EM SEGUNDOS
} else {
$inactive = 60*60; //ESTÁ EM SEGUNDOS
}
}
```

A página menu_portico sendo uma página PHP precisa de determinados comandos para que se actualize sem o auxílio do utilizador. Sendo assim, esta página irá actualizar-se de 5 em 5 min. no caso de estar um aluno ligado:

```
$page = $_SERVER['PHP_SELF'];
$sec = $inactive/12;
header("Refresh: $sec; url=$page");
```

Através da actualização da página estamos a obrigar a página a procurar na base de dados se o aluno foi expulso (foi feito o *logout* através da página expulsar, em vez da página menu_portico):

```
$out = mysql_query("SELECT * FROM utilizadores WHERE numero = '$numero' and
online='0'");
$rowout = mysql_num_rows($out);
```

A variável '\$_SESSION[timeout]' é a variável criada quando o utilizador fez *login*, marcando assim o início da sessão, que será deduzido à função *time()* contando, assim o tempo que passou após o *login*. O *logout* será feito automaticamente, em duas situações: o aluno foi expulso ou o tempo de sessão expirou ($\$session_life > \$inactive$). Estes comandos também só são possíveis de executar, pois a página actualiza-se automaticamente, porque em caso contrário só seriam executados quando a página é aberta, isto é, aos 0s após o início da sessão:

```

$session_life = time() - $_SESSION['timeout'];
if($session_life > $inactive) {
    header("Location: logout.php");
    exit;
} else if ($rowout ==1){
    header("Location: logout.php");
    exit;
}
?>

```

Para tornar a página mais interactiva e ser possível avisar o aluno de que a sua sessão vai expirar é necessário, recorrer à linguagem JavaScript. Como o JavaScript não reconhece as variáveis PHP e vice-versa, converteu-se a variável PHP ‘\$session_life’ na variável JavaScript ‘var php_session’.

Dentro da estrutura JavaScript criou-se uma mensagem de alerta (Figura 3.11) que irá aparecer na página menu_portico 5 min. antes da sessão expirar, para o aluno ter tempo para parar os equipamentos correctamente, fechar os separadores da experiência e fazer *logout* ou esperar que seja feito automaticamente:

```

<script>
var php_session = "<?php echo $session_life; ?>";
var php_inactive = "<?php echo $inactive; ?>";

if (php_session >= php_inactive - 300){
alert("A SUA SESSÃO VAI ENCERRAR DENTRO DE 5 MIN.!");
}
</script>

```

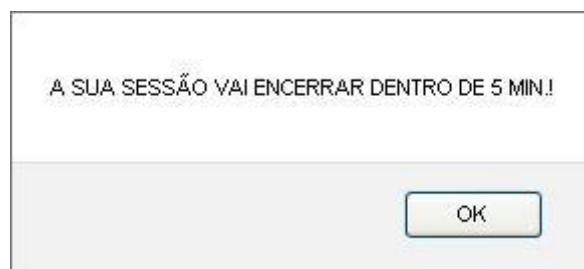


Figura 3.11 – Mensagem de Alerta

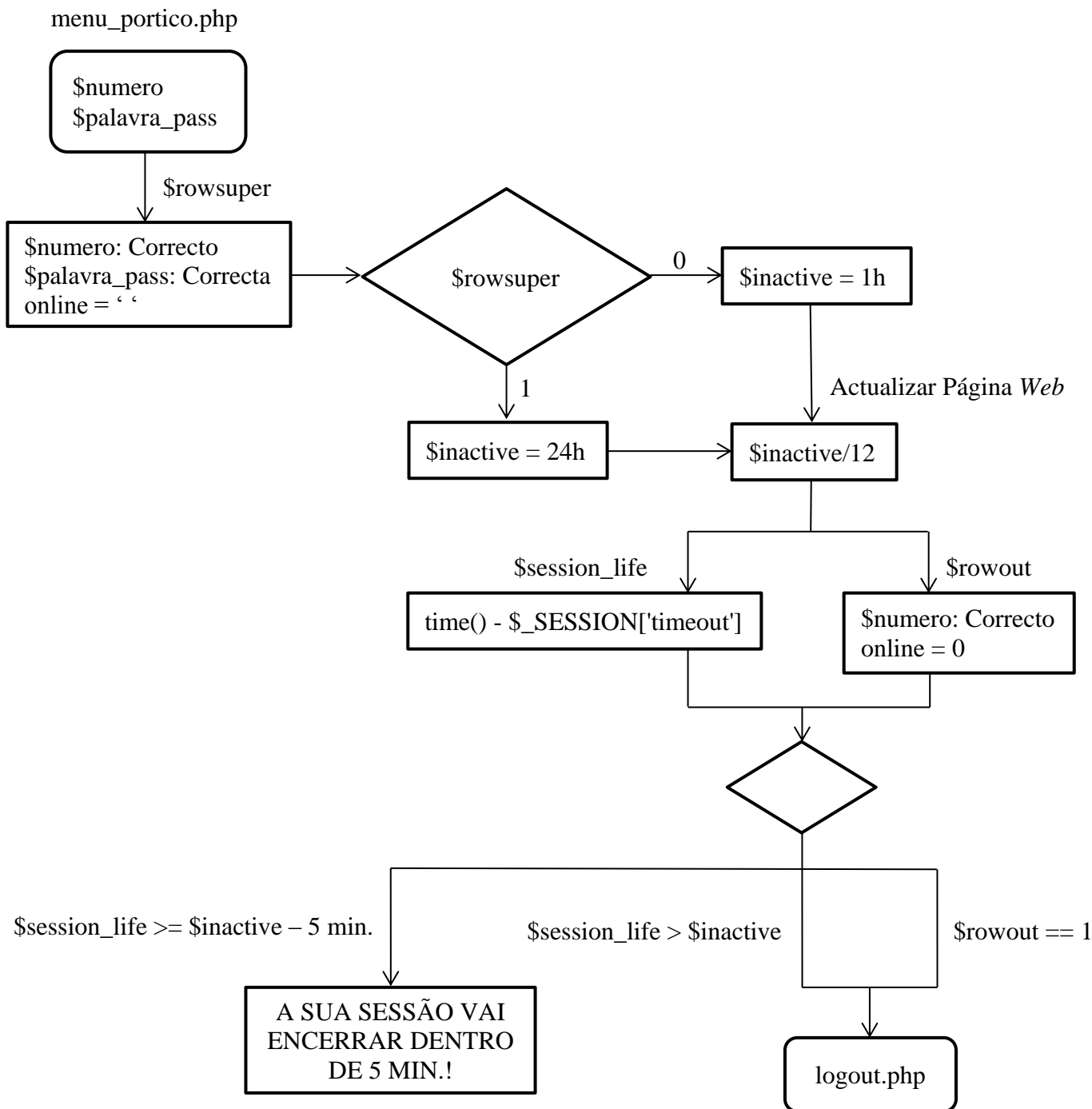


Figura 3.12 – Fluxograma do Menu da Experiência

O fluxograma da Figura 3.12 representa de uma forma sintetizada o funcionamento do menu de uma experiência.

3.5.3. Logout

Fazer *logout* ao aluno ou ao supervisor é bastante simples comparado com o *login*. Semelhante ao *login* procura-se na base de dados, a linha com o número em sessão e verifica-se se a coluna 'online' está vazia:

```
<?php
session_start();
session_destroy();
$numero=$_SESSION['numero'];

$out = mysql_query("SELECT * FROM utilizadores WHERE numero = '$numero' and
online='');
$rowout = mysql_num_rows($out);
```

No caso de ser falso ($\$rowout = 0$), quer dizer que é um aluno que está *online* então a coluna 'online' e a coluna 'exp' passarão a ter valor '0', independentemente da experiência. Mas, se for verdadeiro ($\$rowout = 1$) é um supervisor que está *online* e nesse caso irá para a página de menu:

```
if ($rowout != 1){
    mysql_query("UPDATE utilizadores SET online=0 WHERE
numero='$numero'");
    mysql_query("UPDATE utilizadores SET exp=0 WHERE numero='$numero'");
}
header("Location: menu.php");
?>
```

A Figura 3.13 apresentada a seguir demonstra o funcionamento da página *logout*.

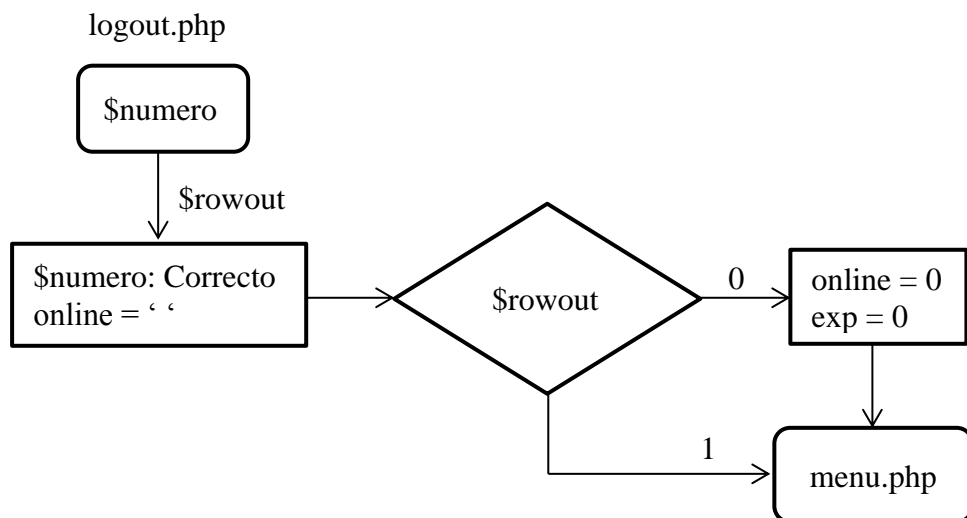


Figura 3.13 – Fluxograma do *Logout*

3.6. Experiência: Pórtico

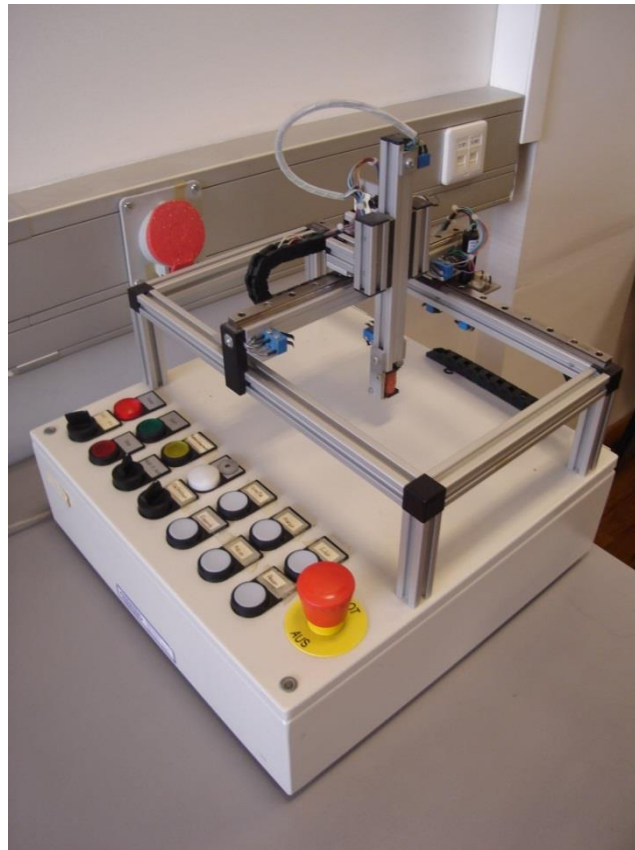


Figura 3.14 – Pórtico

A experiência descrita ao longo deste subcapítulo foi criada com o objectivo de controlar o pórtico, representado na Figura 3.14, que é constituído por 3 motores que irão controlar a posição do electroímã num eixo tridimensional, o electroímã, 6 sensores de fim de curso (2 para cada eixo), 13 botões (Figura 3.15) e 11 luzes.



Figura 3.15 – Botões do Pórtico

3.6.1. Ligação dos PLCs e do Pórtico

O pórtico está ligado a uma caixa através de um cabo com um ficha de 37 pins (37-pol Sub-D-Plug) nas duas extremidades, de onde saem os fios dos sensores e dos actuadores, estando ambos divididos entre duas filas (Tabela 3.4), e nas extremidades de cada fila encontram-se os fios que irão alimentá-los com 24V (Tabela 3.5).

Tabela 3.4 – Identificação dos Fios

Identificação dos Fios	
CIMA	20 21 22 23 ... 35 36 37
BAIXO	1 2 3 4 5 ... 16 17 18 19

Tabela 3.5 – Número dos Cabos Ligados à Energia

Ligação à Corrente		
Input/Output	19/37	24 V
	1/20	0 V

Devido ao número de sensores (19) e actuadores (16) ser elevado foi necessário recorrer a dois PLCs, pois cada um dos PLCs existentes tem um limite de 14 entradas e 10 saídas. Como os sensores e os actuadores estavam divididos em duas filas decidiu-se ligar os fios da fila de baixo ao PLC 1 e os da fila de cima ao PLC 2, ficando o PLC 1 ligado a 9 sensores e 8 actuadores, e o PLC 2 a 10 sensores e 8 actuadores.

A Figura 3.16 é uma representação esquemática das ligações entre o PLC e os respectivos fios dos sensores e actuadores, e as ligações internas do próprio PLC. A linha a vermelho (L+) de 24V será dividida em 4, duas irão alimentar as saídas do PLC (1L e 2L) e as outras duas irão alimentar os sensores e actuadores (fios 19 no caso do PLC 1 e 37 se for o PLC 2). A linha azul de 0V será dividida em apenas 3, uma ligada às entradas do PLC (1M) e as outras duas aos fios 1 (PLC 1) ou aos fios 20 (PLC 2). As linhas a preto serão ligadas aos fios dos sensores e dos actuadores.

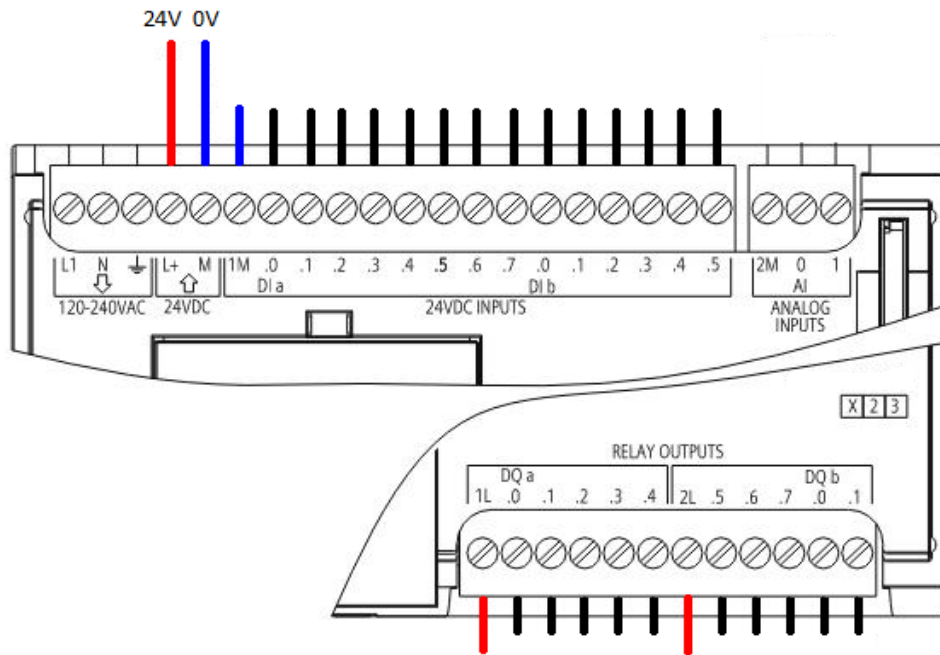


Figura 3.16 – Esquema de Ligação do PLC aos Sensores e Actuadores

Na Tabela 3.6 e 3.7 estão indicadas as correspondências entre os números dos fios dos sensores/actuadores, as entradas/saídas dos PLCs e as *tags* utilizadas no TIA Portal.

Tabela 3.6 – Identificação das Conexões dos Sensores

	Nome	37-pol Sub-D-Plug	S7-1200 Input	Tag
PLC 1	Auto	2	I 0.0 (M 6.0)	auto
	Magnet	3	I 0.1 (M 6.1)	electroiman
	Push Button Left	4	I 0.2 (M 6.2)	esq
	Push Button Right	5	I 0.3 (M 6.3)	dir
	Emergency-Stop	6	I 0.4 (M 6.4)	EM
	Z-Axis Forward	10	I 0.5	SZav
	Z-Axis Back	11	I 0.6	SZrec
	X-Axis Right	12	I 0.7	SXdir
	X-Axis Left	13	I 1.0	SXesq
PLC 2	Manual	21	I 0.0 (M 1.0)	manual
	Acknowledge	22	I 0.1 (M 1.1)	repo
	Stop	23	I 0.2 (M 1.2)	stop
	Start	24	I 0.3 (M 1.3)	st
	Push Button Up	25	I 0.4 (M 1.4)	sub
	Push Button Down	26	I 0.5 (M 1.5)	des
	Push Button Forward	27	I 0.6 (M 1.6)	av
	Push Button Back	28	I 0.7 (M 1.7)	rec
	Y-Axis Low	29	I 1.0 (M 2.0)	SYdes
	Y-Axis Top	30	I 1.1 (M 2.1)	SYsub

Tabela 3.7 – Identificação das Conexões dos Actuadores

	Nome	37-pol Sub-D-Plug	S7-1200 Output	Tag
PLC 1	Pilot Lamp Right	2	Q 0.0	lampdir
	Pilot Lamp Back	3	Q 0.1	lamprec
	Pilot Lamp Forward	4	Q 0.2	lampav
	Pilot Lamp Down	5	Q 0.3	lampdes
	Z-Axis Back	6	Q 0.4	MZrec
	Y-Axis Up	7	Q 0.5	MYsub
	Y-Axis Down	8	Q 0.6	MYdes
	Magnet on	9	Q 0.7	electroiman on
PLC 2	Pilot Lamp Left	21	Q 0.0 (M 3.0)	lampesq
	Pilot Lamp Acknowl.	22	Q 0.1 (M 3.1)	lamprepo
	Pilot Lamp Stop	23	Q 0.2 (M 3.2)	lampstop
	Pilot Lamp Start	24	Q 0.3 (M 3.3)	lampst
	Z-Axis Forward	25	Q 0.4 (M 3.4)	MZav
	X-Axis Left	26	Q 0.5 (M 3.5)	MXesq
	X-Axis Right	27	Q 0.6 (M 3.6)	MXdir
	Pilot Lamp Up	28	Q 0.7 (M 3.7)	lampsub

3.6.2. Comunicação entre os Dois PLCs

Como os PLCs independentemente não conseguem controlar o pórtilco será necessário uma forma de comunicarem entre si. O TIA Portal contém dois conjuntos de funções que podem ser utilizados para esse fim:

1. Permite a comunicação entre equipamentos Siemens que não sejam da mesma família e que não estejam ligados na mesma rede (TCON, TDISCON, TSEND e TREC);
2. A comunicação é feita unicamente entre equipamentos da família S7 e dentro da mesma rede (PUT e GET).

Tendo sido a segunda hipótese a escolhida para esta experiência.

A função PUT envia a informação do PLC que a contém para outro PLC, enquanto a função GET faz exactamente o oposto “retira” a informação de outro PLC para o PLC que contém a função. No caso de dois PLCs, basta um ter as duas funções e o programa

a ser executado, enquanto o outro apenas envia e recebe a informação pedida pelo primeiro.

As funções PUT/GET só são activadas se o valor de uma entrada, saída ou memória passar de 0 a 1, e como é preciso uma comunicação constante recorreu-se ao *clock memory* do PLC. O *clock memory* utiliza um *byte* (8 *bits*) de memória do PLC, sendo que cada *bit* corresponde a uma determinada frequência a que esse mesmo *bit* passa de 0 a 1 (Tabela 3.8).

Tabela 3.8 – Frequência de cada *bit* do *Clock Memory*

M 0.7	M 0.6	M 0.5	M 0.4	M 0.3	M 0.2	M 0.1	M 0.0
0,5 Hz	0,625 Hz	1 Hz	1,25Hz	2 Hz	2,5 Hz	5 Hz	10 Hz
2 s	1,6 s	1 s	0,8 s	0,5 s	0,4 s	0,2 s	0,1 s

A função GET (Figura 3.17) irá ler a entrada do PLC 2 (*ADDR_X*, *Address*) e irá guardar esse valor em *RD_X* (*Read*), que neste caso são memórias. Para cada entrada que o PLC 2 tem ligado foram criadas memórias no PLC 1. Como cada função só permite quatro ligações de cada vez foram necessárias três funções GET.

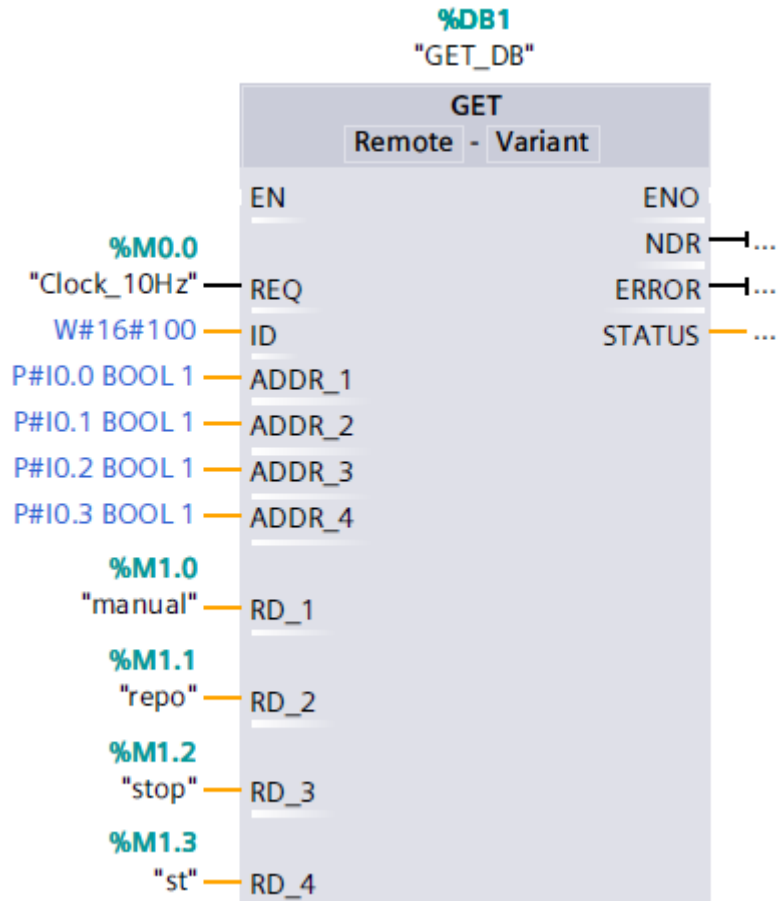


Figura 3.17 – Exemplo Função GET

A função PUT (Figura 3.18) é diferente da função GET em dois aspectos, o ADDR_X nesta função é o *bit* de destino e não o de origem, e em vez de RD_X teremos SD_X (*Send*).

Esta função pode enviar a informação logo directamente para as saídas do PLC de destino, mas isso iria criar um problema no caso de o actuador ser um motor, pois o motor iria estar num ciclo de pára-arranca. Para resolver esse problema basta enviar a informação para uma memória existente no PLC de destino e criar um programa igual ao da Figura 3.19, para cada saída que estiver a ser utilizada pelo PLC.

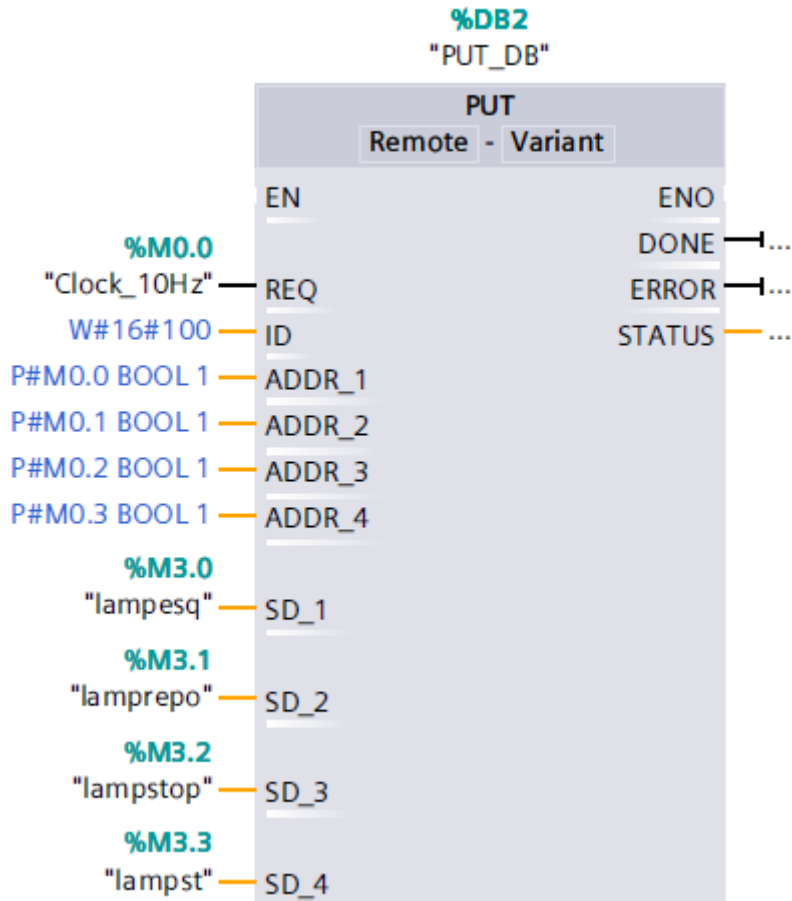


Figura 3.18 – Exemplo Função PUT

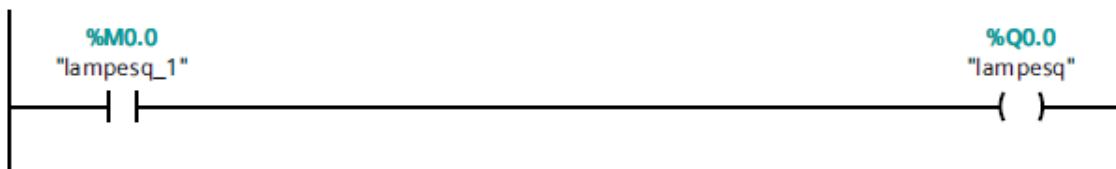


Figura 3.19 – Accionar o Output

Assim, com recurso a apenas duas funções simples, é possível dois PLCs independentes controlarem em conjunto uma experiência, através de uma ligação *ethernet* dentro da mesma rede.

Na Figura 3.20 está representada uma parte do programa que em conjunto com as funções atrás descritas permite controlar a experiência. No Anexo B poder-se-á visualizar todo o programa criado no TIA Portal, tanto do PLC 1 como do 2.

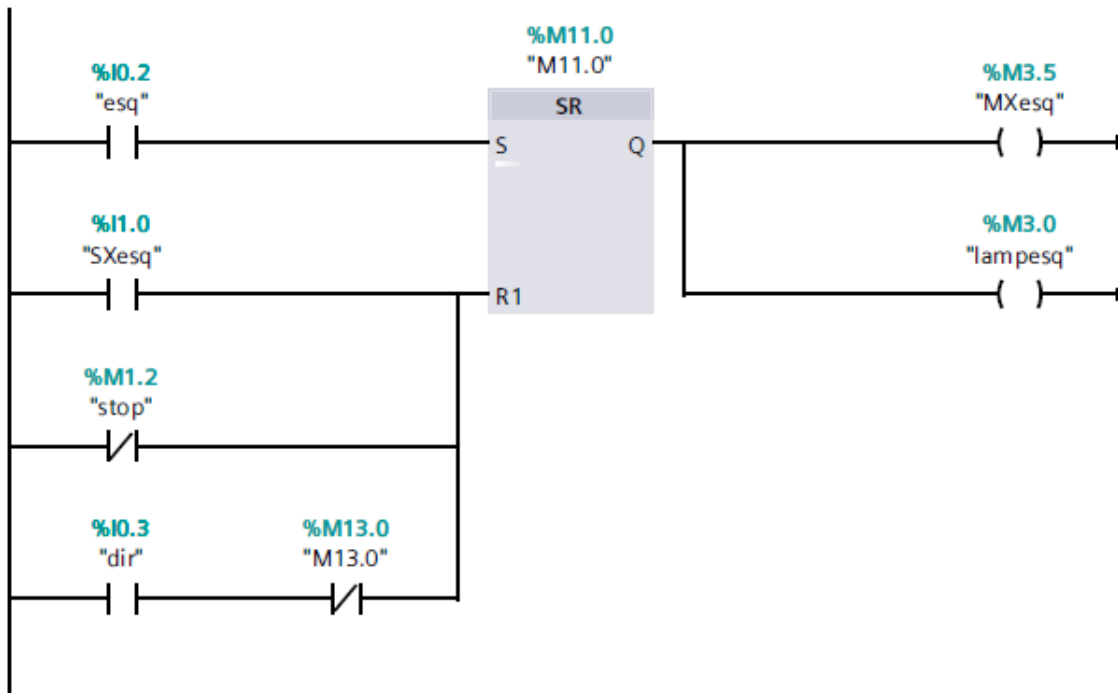


Figura 3.20 – Programa referente ao Movimento para a Esquerda

3.6.3. Página Web da Experiência: Pórtico

Para se poder utilizar o PLC como um servidor é necessário, antes de mais activar o servidor nas propriedades do mesmo (Figura 3.21). Após activado é necessário recorrer à função WWW (Figura 3.22) que permitirá enviar as páginas *web* criadas para o PLC e assim permitir controlar a experiência remotamente.

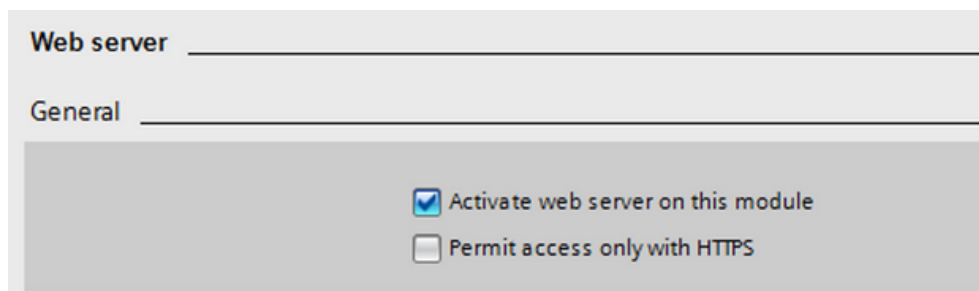


Figura 3.21– Activar Web Server no PLC

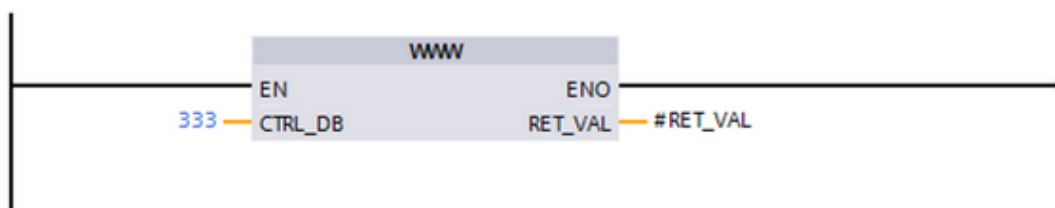


Figura 3.22 – Função WWW

Após esta breve introdução irá se explicar o funcionamento da página *web* referente à experiência do Pórtico (Figura 3.23).



Figura 3.23 – Página Web da Experiência: Pórtico

Como se pode ver existem vários botões direccionais, sendo que os azuis controlam os motores referentes à figura da esquerda (Horizontal: Eixo X; Vertical: Eixo Z) e os verdes o motor relativo ao subir e descer do electroíman (Eixo Y). Entre os restantes botões existe, o Stop que pára o movimento de todos os motores, o Home que irá comandar os motores para que o electroíman regresse à posição 0 (posição visualizada na figura) e o Íman que através dos botões ON e OFF irá activar e desactivar o electroíman.

Para o utilizador poder controlar a experiência através da página *web* é preciso, que cada botão tenha atribuído a *tag* da variável que está a ser utilizada pelo PLC. Sendo aconselhável que essas variáveis sejam memórias, pois se utilizarmos directamente a *tag* de uma entrada não será possível controlar a experiência directamente no laboratório.

Antes de se escrever qualquer linha de programação é necessário inserir o seguinte:

```
<!-- AWP_In_Variable Name="rec" -->  
<html>  
<head>
```

Sendo o rec uma *tag* utilizada pelo PLC. E para cada entrada que o utilizador queira controlar é necessário inserir o que foi descrito atrás, utilizando a *tag* da entrada em questão.

Para tornar o movimento do pórtico o mais real possível é necessário recorrer a diversas funções e para isso são necessárias variáveis. Portanto, antes do corpo da página são criadas as variáveis indicadas a seguir:

```
<head>
<script type="text/javascript">
var animate_right;
var animate_left;
var animate_forward;
var animate_back;
var animate_up;
var animate_down;
var pos_0 = 0;
var down = 0;
var right = 0;
var left = 0;
var frward = 0;
var bck = 0;
function init(){
  imgObj1 = document.getElementById('img1');
  imgObj1.style.position= 'relative';
  imgObj1.style.left = '0px';

  imgObj2 = document.getElementById('img2');
  imgObj2.style.position= 'relative';
  imgObj2.style.left = '0px';
  imgObj2.style.top = '0px';

  imgObj3 = document.getElementById('img3');
  imgObj3.style.position= 'relative';
  imgObj3.style.top = '0px';
}
window.onload = init;
</script>
</head>
```

As variáveis animate são necessárias para representar o movimento do pórtico na página, a pos_0 será utilizada em conjunto com o botão Home e, as down, right, left, frward e bck irão ser utilizadas com os botões ON/OFF do Íman. A função init() define as posições relativas dos diferentes objectos, sendo o objecto 1 a barra vertical que irá movimentar o electríman no eixo X, o objecto 2 o motor que irá movimentá-lo no eixo

Z e o objecto 3 a barra onde está o electroíman, que será movimentada no eixo Y. O evento window.onload irá colocar todos os objectos na sua posição 0 assim que a página for carregada.

Todos os botões de modo geral funcionam da mesma maneira, sendo assim, será só explicado o funcionamento do botão Stop. O botão funciona como um formulário que tem como input uma imagem (type="image"), sendo essa imagem a indicada pelo src (*source* - fonte), quando é clicado (onClick) irá accionar a função associada ao botão. A função que é accionada apenas controla o pórtico na página, por isso, é necessário outro input, um invisível (type="hidden") com um determinado valor (value) que será enviado para o PLC, após a submissão do formulário. O nome do input invisível tem de ser igual a '"stop"', onde o texto stop é o nome da *tag* utilizada, se fosse outro botão o texto seria o nome da *tag* respectiva:

```
<form method="post" action="move.html" target="_blank">
<input type="image" src="img\stop.png" name='&quot;stop&quot;'
onClick="moveStop()" alt="Submit">
<input type="hidden" name='&quot;stop&quot;' value="0">
</form>
```

A imagem do íman tem uma particularidade que está associada aos botões ON e OFF. O botão ON ao ser clicado irá accionar a função ImageOn() e o botão OFF a função ImageOff(), sendo que ambas as funções estão associadas às variáveis down, right, left, forward e bck, que indicam se o electroíman está em baixo (down == 1) e se os outros dois motores estão parados (right == 0, left == 0, forward == 0 e bck == 0):

```
<script>
function ImageOn(){
    if (down == 1 & right == 0 & left == 0 & forward == 0 & bck == 0){
document.getElementById("magnet").src =
"http://10.68.13.173/awp/P%C3%B3rtico/img%5Cmagnet1.png";
}
}
function ImageOff(){
    if (down == 1 & right == 0 & left == 0 & forward == 0 & bck == 0){
document.getElementById("magnet").src =
"http://10.68.13.173/awp/P%C3%B3rtico/img%5Cmagnet0.png";
}
}
</script>
```

Se o botão ON for clicado a imagem irá alterar-se da figura da esquerda para a da direita (Figura 3.24) e se for o botão OFF será o contrário, indicando assim se o electroímã foi accionado ou não.



Figura 3.24 – Ímã OFF/ Ímã ON

Como todos os botões direccionais têm funções semelhantes só será explicado as funções relativas a um par de botões, neste caso o par direita/esquerda. A função do botão da direita primeiro verifica se está na posição zero ($pos_0 == 0$), pois quando o botão Home é pressionado apenas permite ir para a esquerda, recuar e subir, sendo nesse caso que a variável pos_0 é 1.

Após verificado que é possível movimentar para a direita ($pos_0 == 0$), os objectos 1 e 2 irão movimentar-se para a direita a uma velocidade o mais próxima possível do real. Este movimento é conseguido através da variável $animate_right$ que estará ciclicamente a accionar a função ($moveRight()$). A função ao ser accionada irá parar o movimento para a esquerda através do $clearTimeout$ da variável $animate_left$. Quando os objectos atingem o extremo direito ($lim_XX \geq 446$) a animação irá parar:

```
<script type="text/javascript">
function moveRight(){
if (pos_0 == 0){
    imgObj1.style.left = parseInt(imgObj1.style.left) + 1 + 'px';
    imgObj2.style.left = parseInt(imgObj2.style.left) + 1 + 'px';
    animate_right = setTimeout(moveRight, 29.5);
    clearTimeout(animate_left);
    right = 1;
    left = 0;
    var lim_XX = parseInt(imgObj1.style.left);
if (lim_XX >= 446){
    clearTimeout(animate_right);
    right = 0;
}
}
}
}
```

A função que faz o movimento para a esquerda é semelhante ao que faz para a direita, diferenciado apenas que não precisa de verificar se foi pressionado o botão Home e quando os objectos atingem o extremo esquerdo será accionada a função Zero():

```
function moveLeft(){
  imgObj1.style.left = parseInt(imgObj1.style.left) - 1 + 'px';
  imgObj2.style.left = parseInt(imgObj2.style.left) - 1 + 'px';
  animate_left = setTimeout(moveLeft, 29.5);
  clearTimeout(animate_right);
  left = 1;
  reight = 0;
  var lim_XX = parseInt(imgObj1.style.left);
  if (lim_XX <= 0){
    clearTimeout(animate_left);
    Zero();
    left = 0;
  }
}
```

A função do botão Stop é uma das mais simples de todas, pois o que faz é simplesmente fazer *reset* ao temporizador de todas animações, parando qualquer movimento e colocando a variável pos_0 igual a zero, anulando o comando do botão Home:

```
function moveStop(){
  clearTimeout(animate_right);
  clearTimeout(animate_left);
  clearTimeout(animate_forward);
  clearTimeout(animate_back);
  clearTimeout(animate_up);
  clearTimeout(animate_down);
  right = 0;
  left = 0;
  frward = 0;
  bck = 0;
  pos_0 = 0;
}
```

A função associada ao botão Home, apenas acciona as funções para movimentar para a esquerda, recuar e subir, e coloca a variável pos_0 a 1, impedindo assim que o utilizador possa movimentar o electroímã para qualquer sentido que não seja o indicado na função:

```
function moveHome(){
  pos_0 = 1;
  moveLeft();
}
```

```
    moveBack();
    moveUp();
}
```

Quando o botão Home é clicado todos os objectos irão regressar à posição zero, mas como todos têm uma velocidade diferente e podem estar em qualquer posição é preciso recorrer à função Zero(), que irá indicar se todos os objectos estão na posição zero. Esta função é accionada sempre que um dos objectos atinge a posição zero, e só após todos os objectos estarem nessa posição é que será alterado o valor da variável pos_0 para 0, podendo as funções moveRight(), moveForward() e moveDown() serem accionadas:

```
function Zero(){
var lim_XX = parseInt(imgObj1.style.left);
var lim_ZZ = parseInt(imgObj2.style.top);
var lim_YY = parseInt(imgObj3.style.top);
if (lim_XX <= 0 & lim_ZZ >= 0 & lim_YY <= 0){
    pos_0 = 0;
}
}
</script>
```

As linhas de código que compõem a página *web* podem ser visualizadas no Anexo C.

3.7. Outras Alterações

3.7.1. Formulário de Inscrição

Para evitar que os alunos introduzam só alguns dados e submetam uma inscrição incompleta, adicionou-se aos inputs utilizados em PHP o atributo required, que apresenta uma mensagem de alerta junto do campo não preenchido (ver Figura 3.25), no caso de o aluno submeter a inscrição com esse campo em falta:

```
<input type="text" name="primeiro_nome" required="required" />
```

* Preenchimento Obrigatório

PRIMEIRO NOME: *

ULTIMO NOME: *

NÚMERO DE ALUNO: *

TURMA:

E-MAIL: *

PALAVRA PASS: Por favor preencha este campo.

[VOLTAR AO MENU](#)

Figura 3.25 – Formulário de Inscrição com Campo em Falta

Este atributo só está disponível devido ao HTML5, a quinta revisão da linguagem HTML, que começou a ser introduzida em 2008. E sendo assim, só a partir de determinadas versões (Features, 2014) é que o atributo atrás referido começou a ser suportado, sendo elas:

- Google Chrome 10 (Disponível: 2011-03-08);
- Mozilla Firefox 5 (Disponível: 2011-06-21);
- Internet Explorer 10 (Disponível: 2012-10-26).

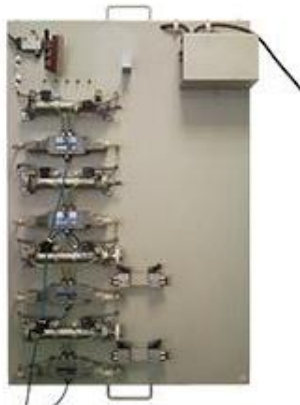
Como se pode ver qualquer *web browser* actualizado já suporta o atributo, mas existe uma situação em particular que não o suporta. Sendo essa situação a de o aluno ter como sistema operativo o Windows XP e utilizar o Internet Explorer, como o Windows XP apenas suporta até à versão 8 do Internet Explorer, impossibilita o suporte do referido atributo.

3.7.2. Menu

Para que um aluno saiba se já existe outro ligado, sem ter que introduzir os seus dados e esperar que apareça uma mensagem indicando isso mesmo, adicionou-se à página de menu uma mensagem que irá indicar se todos os utilizadores estão *offline* ou se existe algum aluno *online* e em que experiência.

Primeiro é necessário procurar na base de dados se existe algum aluno *online* e em que experiência, se existir é preciso saber o número desse aluno, através da instrução `SELECT`, e guardar numa variável através do comando `mysql_fetch_array`. Tendo o número do aluno será indicado debaixo da imagem da experiência o aluno que está *online* (ver Figura 3.26):

```
<?php
$on2 = mysql_query("SELECT * FROM utilizadores WHERE online=1 and exp=2");
$rowon2 = mysql_num_rows($on2);
if ($rowon2 == 1){
    $num = mysql_query("SELECT numero FROM utilizadores WHERE online=1
and exp=2");
    $numero = mysql_fetch_array($num);
    echo "<center>O ALUNO <font color=\"blue\">{$numero['numero']}</font>
ESTÁ <font color=\"green\">ONLINE </font>!</center>";
}
?>
```



CILINDROS



PÓRTICO

○ ALUNO 32958 ESTÁ ONLINE ! ○ ALUNO 35569 ESTÁ ONLINE !

EXPULSAR

Figura 3.26 – Indicação de Aluno *Online* debaixo da Experiência referente

Se existir pelo menos um aluno *online* irá aparecer a hiperligação para a página expulsar. Caso contrário irá aparecer a mensagem indicada na Figura 3.27:

```
<?php
$on1 = mysql_query("SELECT * FROM utilizadores WHERE online=1 and exp=1");
$rowon1 = mysql_num_rows($on1);
$on2 = mysql_query("SELECT * FROM utilizadores WHERE online=1 and exp=2");
```

```

$rowon2 = mysql_num_rows($on2);
if ($rowon1 == 1 | $rowon2 == 1){
    echo '<center><a href="expulsar.php">EXPULSAR</a></center>';
} else if ($rowon1 == 0 & $rowon2 == 0){
    echo "<center>TODOS OS UTILIZADORES ESTÃO <font
color=\"red\">OFFLINE</font>!</center>";
}
?>

```

TODOS OS UTILIZADORES ESTÃO OFFLINE!

Figura 3.27 – Indicação de todos os Alunos *Offline*

Apesar de o supervisor fazer *login* nunca irá aparecer se está *online* ou não, pois a mensagem foi criada para indicar os utilizadores com a coluna ‘online’ igual a 1, e como o supervisor tem essa coluna vazia será como se fosse invisível.

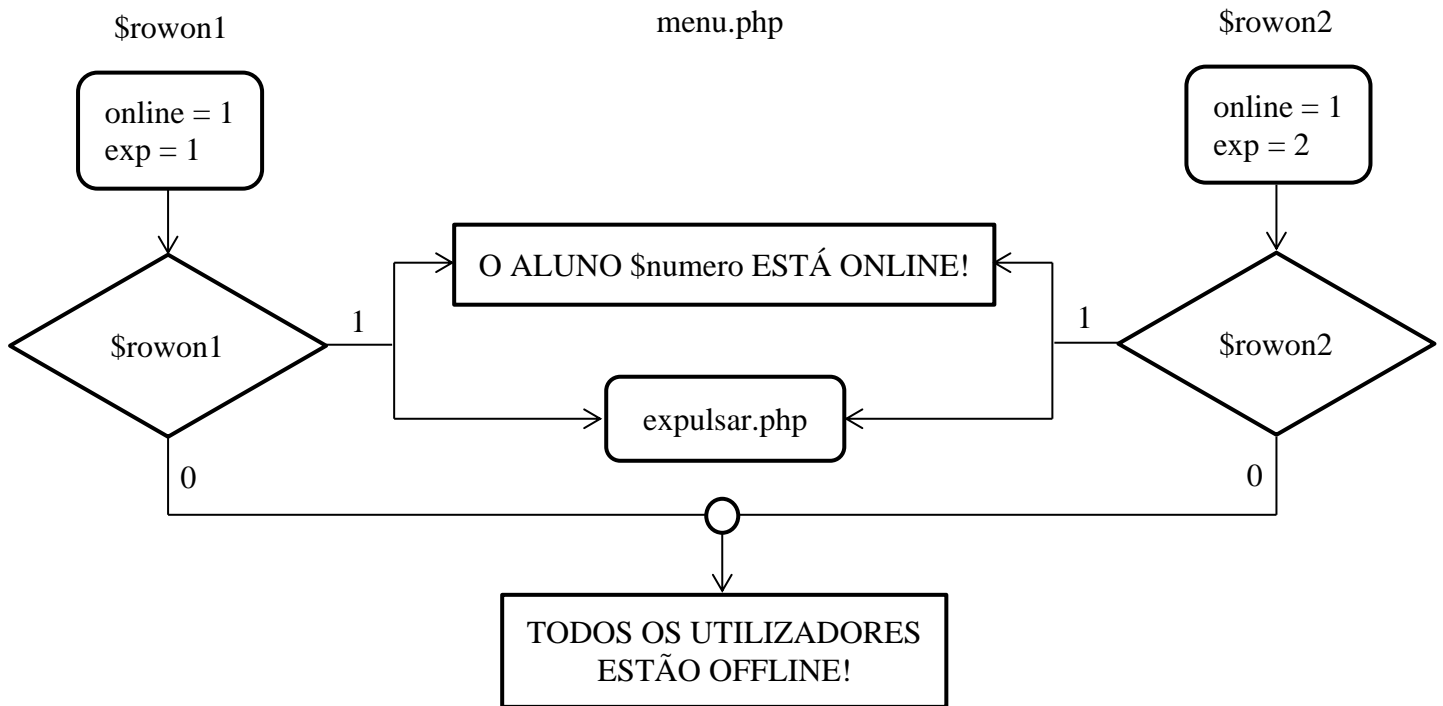


Figura 3.28 – Fluxograma do Menu do LabAPI

A Figura 3.28 descreve de uma forma simplificada como é feita a indicação dos alunos *online* e *offline* no menu do LabAPI.

3.7.3. Expulsar o Utilizador

Esta página é utilizada no caso de um aluno ter fechado o separador do menu da experiência ou a janela sem ter feito *logout*, levando a que na base de dados apareça como *online* apesar de não estar. Além da situação referente ao aluno expulsar-se a si próprio, também existe a situação de o supervisor querer expulsar o aluno, no caso de o aluno não estar a trabalhar correctamente com o equipamento.

A página expulsar funciona da mesma forma que a página *login* com um formulário em que são inseridos os dados de acesso, mas neste caso em vez de se ter um, tem se dois formulários, um para o supervisor e outro para o aluno (ver Figura 3.29). Sendo que cada um está ligado a uma diferente página que irá autenticar os dados inseridos.

<p>Nº DE SUPERVISOR: <input type="text"/></p> <p>PALAVRA PASSE: <input type="text"/></p> <p>Nº DO ALUNO: <input type="text"/></p> <p><input type="button" value="OK"/></p>	<p>Nº DE ALUNO: <input type="text"/></p> <p>PALAVRA PASSE: <input type="text"/></p> <p><input type="button" value="OK"/></p>
--	--

[VOLTAR AO MENU](#)

Figura 3.29 – Página Web expulsar.php

Expulsar pelo Supervisor

Esta página é muito semelhante à `userauthentication2.php`, pois verifica se os dados estão correctos e é um supervisor (`online=' '`), sendo que nesse caso irá expulsar o aluno indicado, e no caso de os dados estarem incorrectos ou não ser um supervisor apresentará uma mensagem a indicar que os dados estão incorrectos:

```
<?php
$numero=$_POST['numero'];
$palavra_pass=$_POST['palavra_pass'];
$num_user=$_POST['num_user'];

$pass = mysql_query("SELECT * FROM utilizadores WHERE numero = '$numero'
and palavra_pass = '$palavra_pass'");
$rowpass = mysql_num_rows($pass);

$super = mysql_query("SELECT * FROM utilizadores WHERE numero = '$numero'
and palavra_pass = '$palavra_pass' and online=' '");
$rowsuper = mysql_num_rows($super);

if($rowsuper == 1) {
    mysql_query("UPDATE utilizadores SET online=0 WHERE
numero='$num_user'");
    mysql_query("UPDATE utilizadores SET exp=0 WHERE
numero='$num_user'");
    echo "<center>UTILIZADOR EXPULSO!</center>";
    echo "<script>expulsarsuccessfully()</script>";
} elseif ($rowpass == 0 or $rowsuper == 0){
    echo "<center>PALAVRA PASSE INCORRETA OU N° NÃO
REGISTADO!</center>";
    echo "<script>expulsarfailed()</script>";
}
?>
```

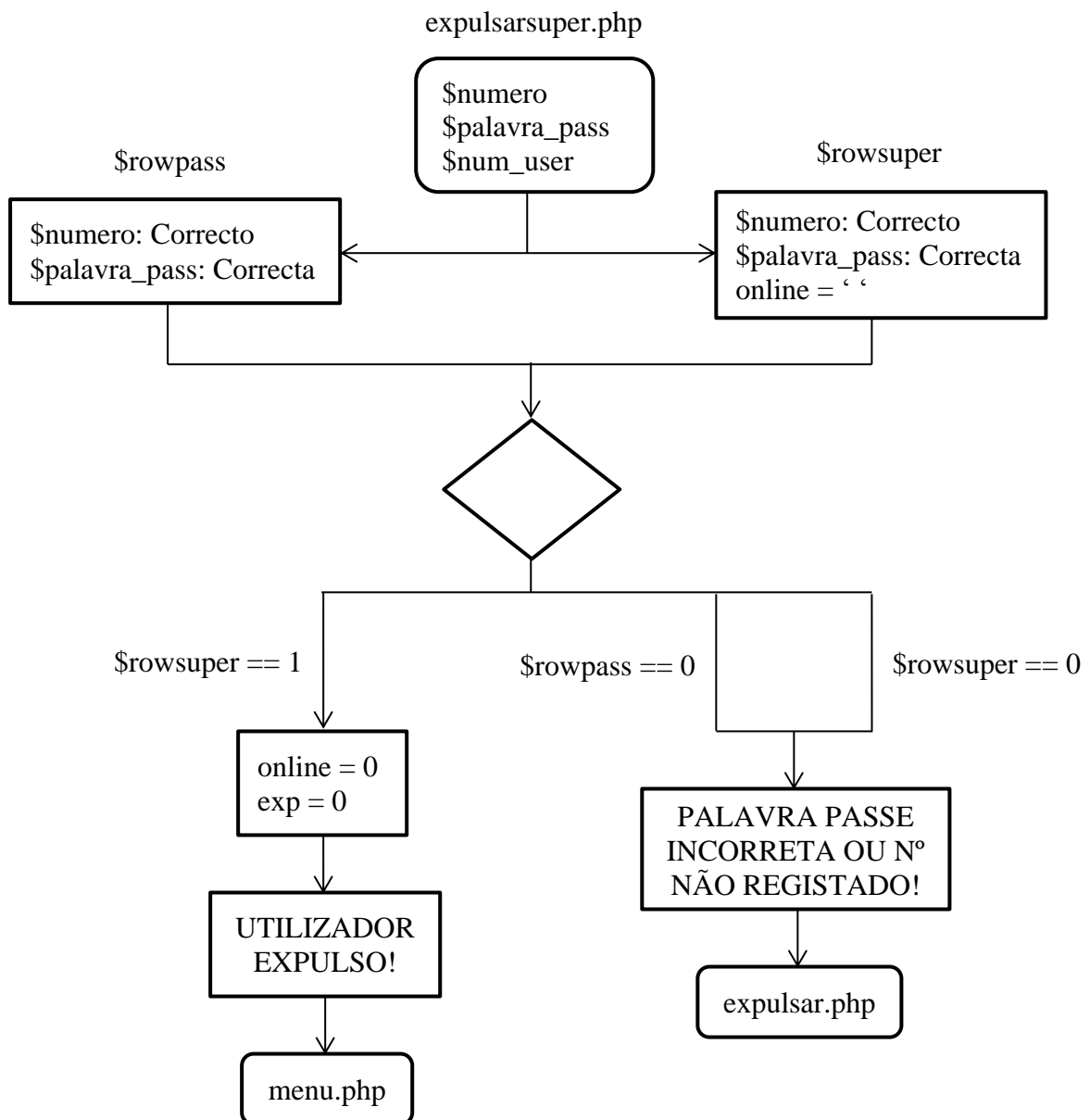


Figura 3.30 – Fluxograma de Expulsar pelo Supervisor

O fluxograma representado na Figura 3.30 pretende de uma forma resumida descrever o funcionamento da página expulsar por parte do supervisor.

Expulsar pelo Aluno

O expulsaruser.php é praticamente igual ao expulsarsuper.php só que utiliza os dados introduzidos no formulário referente ao aluno:

```
<?php
$num_user=$_POST['num_user'];
$palavra_pass=$_POST['palavra_pass'];

$pass = mysql_query("SELECT * FROM utilizadores WHERE numero = '$num_user'
and palavra_pass = '$palavra_pass' and online!=' '");
$rowpass = mysql_num_rows($pass);

if($rowpass == 1) {
    mysql_query("UPDATE utilizadores SET online=0 WHERE
numero='$num_user'");
    mysql_query("UPDATE utilizadores SET exp=0 WHERE
numero='$num_user'");
    echo "<center>UTILIZADOR EXPULSO!</center>";
    echo "<script>expulsarsuccessfully()</script>";
} else {
    echo "<center>PALAVRA PASSE INCORRETA OU N° NÃO
REGISTADO!</center>";
    echo "<script>expulsarfailed()</script>";
}
?>
```

3.7.4. Botão de Emergência

A experiência dos cilindros tinha em falta um botão de emergência para o caso de ser preciso e, por isso, criou-se um botão semelhante ao íman da experiência do pórtico. Mas neste caso foi possível utilizar a imagem do botão de emergência como input em vez de ser preciso utilizar os botões ON e OFF:

```
<form method="post">
<font size="3" color="red"><b>EMERGENCY<b></font>
<input type="image" src="img\red_button:= "stop":.png" name='&quot;stop&quot;'
alt="Submit" style="vertical-align: middle">
<script type="text/javascript">
if (:= "stop": == 0){
    document.write('<input type="hidden" name="&quot;stop&quot;"
value="1"/>');
} else {
    document.write('<input type="hidden" name="&quot;stop&quot;"
value="0"/>');
}
</script>
```

BUTTON
</form>

Da mesma forma que o íman tinha duas imagens para cada uma das situações, o botão de emergência também tem, sendo no caso de não estar activado a Figura 3.31 e se estiver activado a Figura 3.32.



Figura 3.31 – Botão de Emergência não Activado



Figura 3.32 – Botão de Emergência Activado

3.7.5. Painel HMI

A experiência dos cilindros estava ligada a um painel HMI que não tinha utilização, tornando unicamente possível controlar a referida experiência via *web*. Por isso, com recurso ao programa TIA Portal foi criado um ecrã para o painel que permite de uma forma intuitiva controlar os cilindros, ver Figura 3.33.



Figura 3.33 – Painel HMI

3.7.6. Câmara de Vídeo

A câmara inicialmente era para funcionar como alternativa à actual página *web* do pórtico, substituindo as imagens dos componentes do pórtico pela imagem em directo da câmara de vídeo. Mas, devido a uma das seguintes razões, o formato da imagem transmitido pela câmara não ser suportado pelo PLC ou o PLC não conseguir processar esses dados não foi possível efectuar o pretendido inicialmente.

Sendo que a única maneira de visualizar qualquer uma das experiências será direcciona-la fisicamente para a experiência pretendida. Na Figura 3.34 está representada uma imagem da transmissão da experiência dos cilindros.



Figura 3.34 – Transmissão da Experiência

4. Conclusão

O desenvolvimento do laboratório remoto descrito neste trabalho mostrou ser uma tarefa bastante abrangente, envolvendo conceitos fora da área da Engenharia Mecânica, principalmente conceitos informáticos. Devido a essa abrangência foram feitas várias pesquisas com o objectivo de entender o funcionamento de um laboratório remoto, as tecnologias disponíveis e as linguagens de programação necessárias para o seu funcionamento.

4.1. Trabalho Final

No início deste trabalho de mestrado foram indicados vários objectivos, sendo os principais criar novas experiências, melhorar a existente e criar um sistema que permitisse o acesso a um aluno de cada vez, em retrospectiva pode-se dizer que todos os objectivos foram cumpridos. De seguida será descrita como foram atingidos, tantos os objectivos principais como os secundários.

Relativamente ao acesso a um aluno de cada vez, foi necessário reestruturar o *site* do LabAPI – ISEL e recorrer às linguagens PHP e SQL, a fim de criar um sistema que permitisse indicar quais os alunos *online*. A existência de duas experiências levou à necessidade de condicionar o aluno a poder aceder a apenas uma experiência de cada vez. Além disso, também se criou a possibilidade de um supervisor (ex: professor) poder observar as experiências em uso.

A adição da experiência do pÓrtico foi o objectivo mais complexo por duas razões:

- Procurar e entender as funções do TIA Portal que permitissem a comunicação entre dois PLCs;
- Criação de uma página *web* que representa-se o mais fielmente a experiência física, recorrendo a linguagens como HTML e JavaScript.

A melhoria da experiência existente consistiu principalmente na alteração da página *web* para ter uma resolução igual às outras, introdução do botão de emergência na página e alteração do painel HMI da Siemens para permitir o controlo da experiência no laboratório físico.

Todas as melhorias e alterações atrás descritas permitirão dar apoio às aulas das Unidades Curriculares de Programação de Autómatos e Automação de Processos Industriais.

4.2. Trabalho Futuro

O actual laboratório tem algumas limitações principalmente de nível informático, que poderão ser corrigidas por alunos de Engenharia Mecânica, mas devido à sua complexidade seria recomendado alunos de informática, sendo essas limitações as seguintes:

- Poder aceder directamente aos PLCs sem fazer o *login*, se o aluno souber o IP do PLC;
- O aluno ao fechar a janela do menu da experiência sem fazer *logout* continuará a estar *online* na base de dados;
- Se o aluno fizer *logout* antes de fechar as outras janelas poderá continuar a controlá-las, pois as outras janelas não se fecham quando o aluno faz *logout*.

Recomendações que seriam exequíveis por parte de alunos de Engenharia Mecânica:

- Criação de experiências, que permitissem aos alunos da Unidade Curricular de Programação de Autómatos treinar o leccionado nas aulas;
- Criação de um sistema de agendamento das experiências, permitindo assim os alunos reservarem a experiência em questão para um dia e hora convenientes;
- Utilização do laboratório por parte dos alunos de Automação de Processos Industriais, Programação de Autómatos e de outras instituições do ensino superior;
- Realização de inquéritos e análise dos mesmos com o fim de melhorar o LabAPI – ISEL.

Bibliografia

- Borracha, A. (2012). *Laboratório Remoto de Automação Industrial (Lab-RAI)*. Dissertação, Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia, Lisboa.
- Bringing MySQL to the web*. (2014). Obtido em 13 de 6 de 2014, de phpMyAdmin: http://www.phpmyadmin.net/home_page/
- Browser Display Statistics*. (2014). Obtido em 21 de 4 de 2014, de w3schools.com: http://www.w3schools.com/browsers/browsers_display.asp
- Browser Statistics*. (2014). Obtido em 21 de 4 de 2014, de w3schools.com: http://www.w3schools.com/browsers/browsers_stats.asp
- Costa, D. (2011). *Acesso a Laboratórios Remotos via Ambientes Imersivos*. Dissertação, Instituto Superior de Engenharia do Porto, Porto.
- Cybele Software*. (2014). Obtido em 13 de 6 de 2014, de <http://www.cybelesoft.com/thinvnc/>
- El Kontrol LTD. (26 de 8 de 2014). *SIEMENS Demo - Truck Show*. Obtido em 16 de 9 de 2014, de http://www.elkontrol.com/en/each_new.aspx?which_new=20
- Elawady, Y. H., & Tolba, A. (1 de 2011). Analysis, Design and Implementation of a General. *International Journal of Computer Applications*, Vol. 14; N° 1, p. 1.
- Features*. (2014). Obtido em 21 de 4 de 2014, de HTML5test: <http://html5test.com/compare/feature/form-validation-required.html>
- Gomes, L., & Bogosyan, S. (2009). Current Trends in Remote Laboratories. *IEEE Transactions on Industrial Electronics*, 56, pp. 4744–56.
- Gonçalves, J. (2010). *Tecnologias Abertas de Suporte à Experimentação Remota via Web*. Dissertação, Universidade do Porto, Faculdade de Engenharia, Porto.
- Gutierrez, R., & Pan, S. (2008). *Complexo eletrônico: Automação do controle industrial*. Obtido em 5 de 8 de 2014, de BNDES: o banco nacional do

desenvolvimento:

http://www.bndes.gov.br/SiteBNDES/export/sites/default/bndes_pt/Galerias/Arquivos/conhecimento/bnset/set2807.pdf

How to: DIY Home Automation Tutorial. (2013). Obtido em 7 de 8 de 2014, de Hack n Mod: <http://hacknmod.com/hack/diy-home-automation-tutorial/>

Jara et al. (11 de 7 de 2011). Hands-on experiences of undergraduate students in Automatics and Robotics. *Computers & Education*, p. 1 e 5.

John, K., & Tiegelkamp, M. (2007). IEC 61131-3: Programming Industrial Automation Systems.

Leitão, S. M. (2006). *Um ambiente experimental à distância*. Faculdade de Engenharia da Universidade do Porto.

Lowe, D., Newcombe, P., & Stumpers, B. (2013). Evaluation of the Use of Remote Laboratories for Secondary School Science Education. *Research in Science Education*, 43, pp. 1197–1219.

Martins, L. (2013). *Projecto dum Laboratório Remoto para Automação de Processos Industriais*. Trabalho Final de Mestrado, Instituto Superior de Engenharia de Lisboa, Departamento de Engenharia Mecânica, Lisboa.

Notepad++. (2014). Obtido em 30 de 11 de 2014, de Notepad++: <http://notepad-plus-plus.org/>

Pinto, J. R. (2004). *Técnicas de Automação*. Lisboa, Portugal: Lidel - edições técnicas, lda.

Remote Laboratory. (2011). Obtido em 7 de 8 de 2014, de Remote Laboratory at TAMUQ: <http://remotelab.qatar.tamu.edu/>

Restivo, M. T. (2011). On the use and promotion of Remote Labs in Portugal: A Personal Commitment. In J. G. Zubía, & G. R. Alves, *Using Remote Labs* (p. 441). Bilbao: University of Deusto.

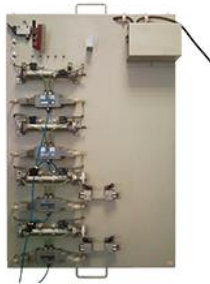
- Ribeiro, M. (2012). *Interface and Web Server Implementation for an Industrial Automation Remote Laboratory*. Dissertação, Instituto Superior Técnico, Lisboa.
- Rössler, P., Rausch, T., & Treytl, A. (2004). Analyzing three case studies of remote laboratories for education and training. *International Symposium Remote Engineering and Virtual Instrumentation*.
- Siemens. (2012). S7-1200 Programmable Controller System Manual. In *S7-1200 Programmable Controller System Manual*. Siemens.
- Siemens. (2014). *Siemens*. Obtido em 19 de 2 de 2014, de Siemens: http://support.automation.siemens.com/dnl/TU/TUyOTI4NwAA_57027057_Akt/S7-1200_Sys.jpg
- Silva, L. (2011). *Laboratório Remoto para Monitorização e Controlo baseado em MATLAB/Simulink e LabVIEW*. Dissertação, Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologias, Lisboa.
- Tawfik et al. (2011). Towards a Better Deployment of Remote Laboratories in Undergraduate Engineering Education. In J. G. Alves, *Using Remote Labs* (p. 387). Bilbao: University of Deusto.
- Villa-López et al. (2013). Electropneumatic system for industrial automation: a remote experiment within a web-based learning environment. *The 2013 Iberoamerican Conference on Electronics Engineering and Computer Science*, (pp. 198-207). México.
- Yawcam. (2014). Obtido em 13 de 6 de 2014, de <http://www.yawcam.com/>

Anexo A

Menu:



Laboratório Remoto de Automação de Processos Industriais



[CILINDROS](#)



[PÓRTICO](#)

TODOS OS UTILIZADORES ESTÃO **OFFLINE!**

[FAZER INSCRIÇÃO](#)

Formulário de Inscrição:



Laboratório Remoto de Automação de Processos Industriais

* Preenchimento Obrigatório

PRIMEIRO NOME:	<input type="text"/>	*
ULTIMO NOME:	<input type="text"/>	*
NÚMERO DE ALUNO:	<input type="text"/>	*
TURMA:	<input type="text"/>	
E-MAIL:	<input type="text"/>	*
PALAVRA PASSE:	<input type="text"/>	*
<input type="button" value="GUARDAR DADOS DE INSCRIÇÃO"/>		

[VOLTAR AO MENU](#)

Expulsar Utilizador:



Laboratório Remoto
de Automação de
Processos Industriais

Nº DE SUPERVISOR: <input type="text"/>	Nº DE ALUNO: <input type="text"/>
PALAVRA PASSE: <input type="text"/>	PALAVRA PASSE: <input type="text"/>
Nº DO ALUNO: <input type="text"/>	<input type="button" value="OK"/>
<input type="button" value="OK"/>	

[VOLTAR AO MENU](#)

Login da Experiência Cilindros:



Laboratório Remoto
de Automação de
Processos Industriais

Login Cilindros

Nº DE ALUNO: <input type="text"/>
PALAVRA PASSE: <input type="text"/>
<input type="button" value="ENTRAR"/>

Login da Experiência Pórtico:



Laboratório Remoto de Automação de Processos Industriais

Login Pórtico

Nº DE ALUNO: <input type="text"/>
PALAVRA PASSE: <input type="password"/>
<input type="button" value="ENTRAR"/>

Menu da Experiência Cilindros:



Laboratório Remoto de Automação de Processos Industriais

Experiência: Cilindros

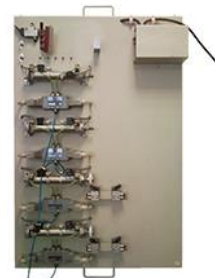
[SAIR](#)



[Ambiente de Trabalho Remoto](#)



[Web Server S7-1200](#)



[Webcam](#)

Menu da Experiência Pórtico:



Laboratório Remoto
de Automação de
Processos Industriais

Experiência: Pórtico

[SAIR](#)



[Web Server S7-1200](#)



[Webcam](#)

Experiência Cilindros:



Laboratório Remoto
de Automação de
Processos Industriais

Cilindros [Variáveis Video](#)

Cilindro A	Cilindro B	Cilindro C	Cilindro D
			
MODO AUTOMÁTICO 		MODO MANUAL 	
CILINDRO A	CILINDRO B	CILINDRO C	CILINDRO D
<input type="button" value="RECUAR"/> <input type="button" value="AVANÇAR"/>	<input type="button" value="RECUAR"/> <input type="button" value="AVANÇAR"/>	<input type="button" value="RECUAR"/> <input type="button" value="AVANÇAR"/>	<input type="button" value="RECUAR"/> <input type="button" value="AVANÇAR"/>
<input type="button" value="Ciclo 1"/>	<input type="button" value="Ciclo 2"/>	<input type="button" value="Ciclo 3"/>	

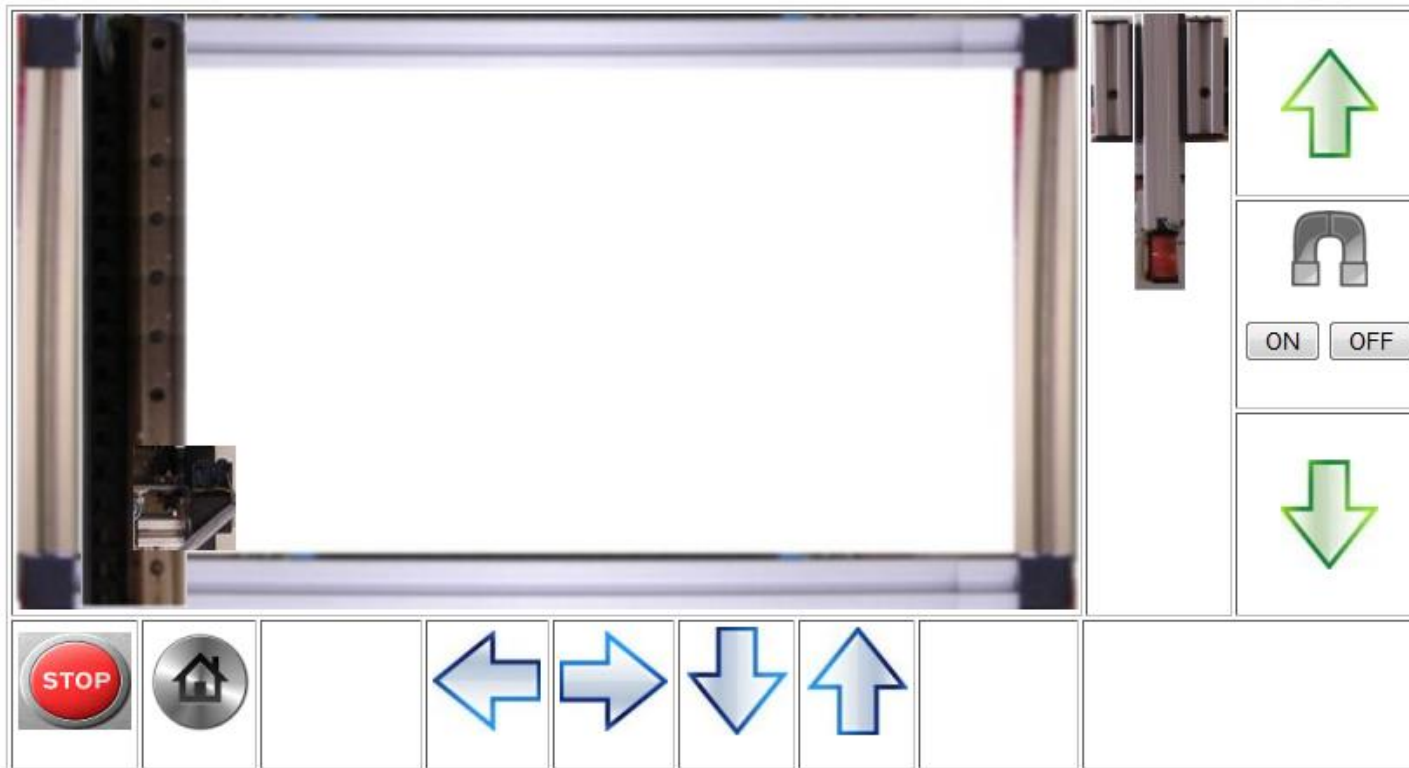


Experiência Pórtico:

ISEL

ÁREA DEPARTAMENTAL DE ENGENHARIA
MECÂNICA - ADEM

Laboratório Remoto de Automação de Processos Industriais



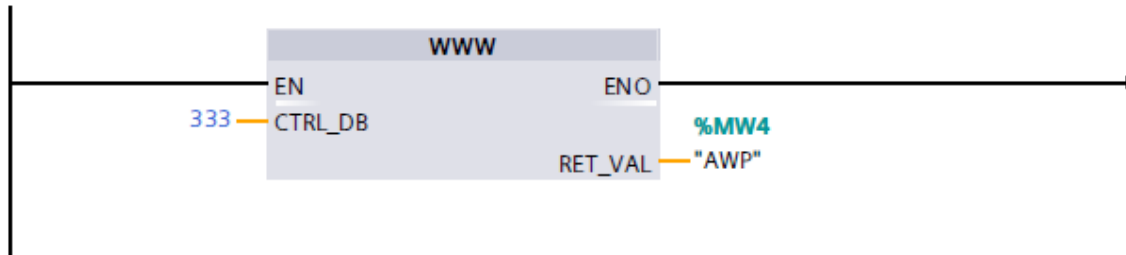
The interface consists of a central video window showing a gantry system. To the right of the video is a control panel with three sections: a green up arrow, a magnet icon with 'ON' and 'OFF' buttons, and a green down arrow. Below the video is a horizontal control bar with a red 'STOP' button, a home button, and four blue directional arrows (left, right, down, up).

Anexo B

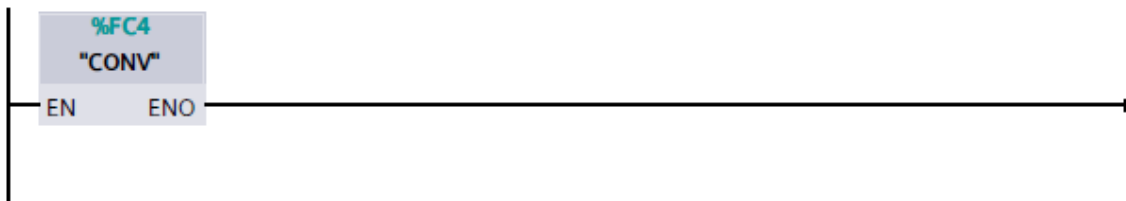
PLC 1:

➤ Main

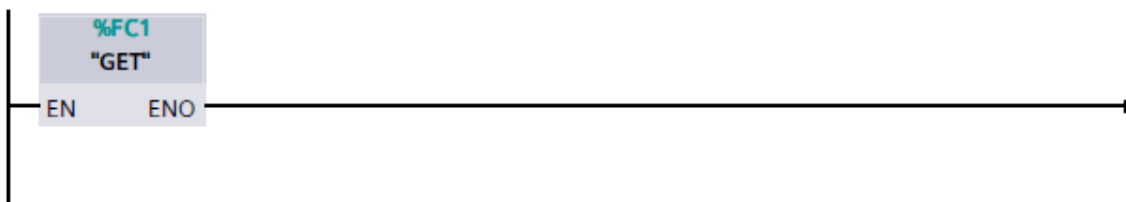
Network 1: Cria as Páginas *Web*



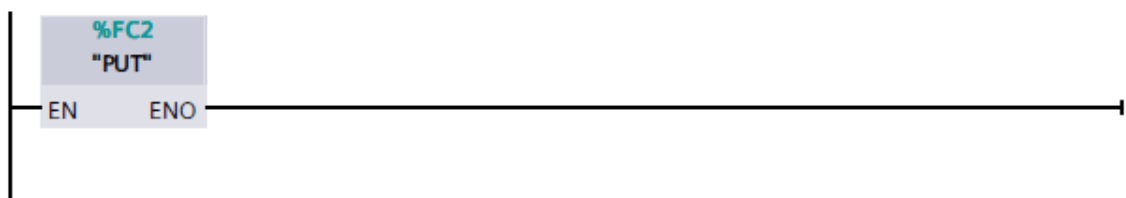
Network 2: Permite controlar os *inputs* (botões) do PLC_1 pelo *browser*



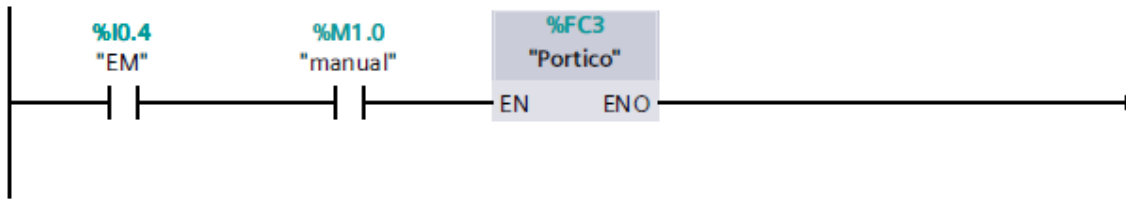
Network 3: Recebe os valores dos *inputs* do PLC_2



Network 4: Envia os valores dos *outputs* para o PLC_2

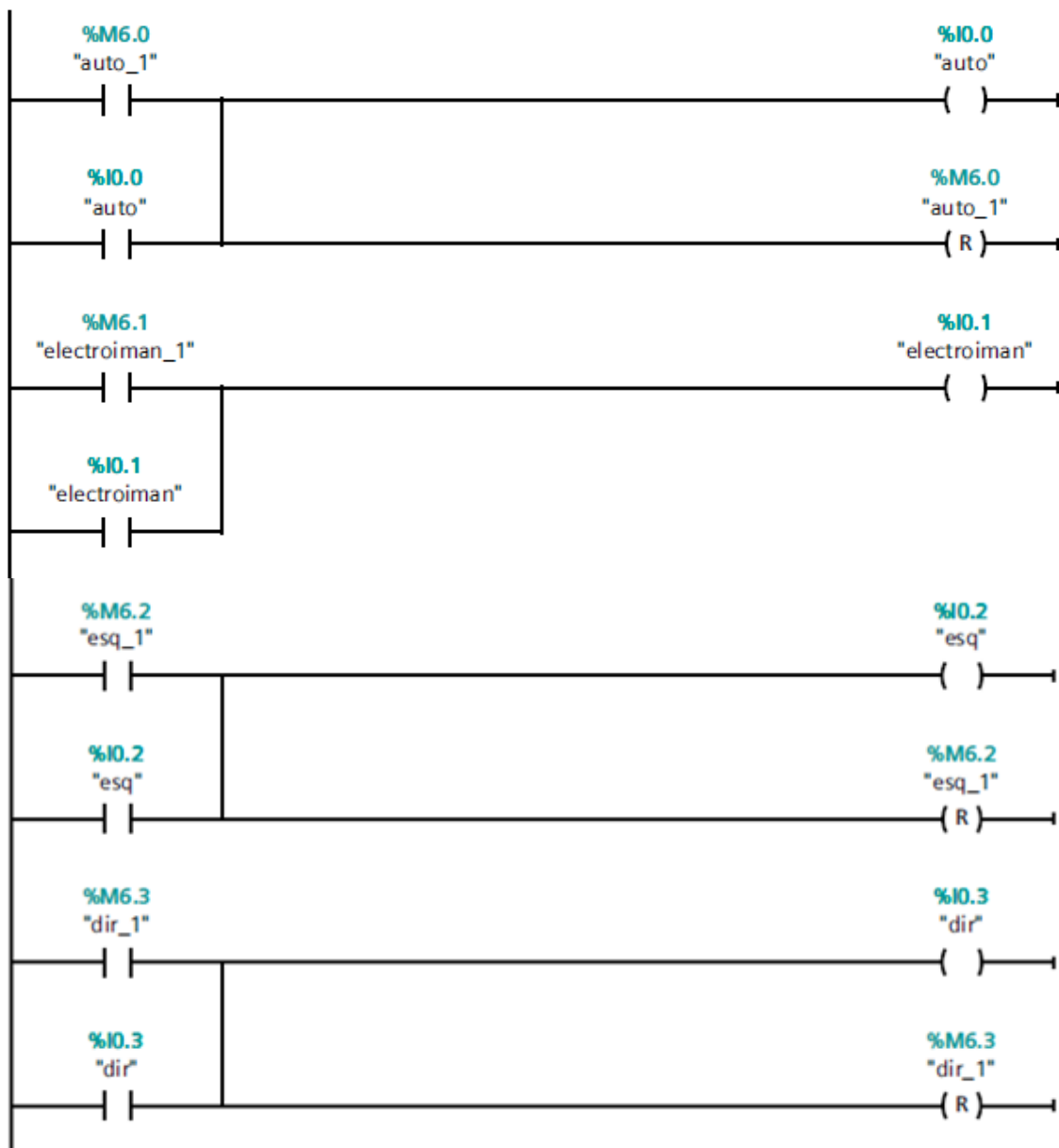


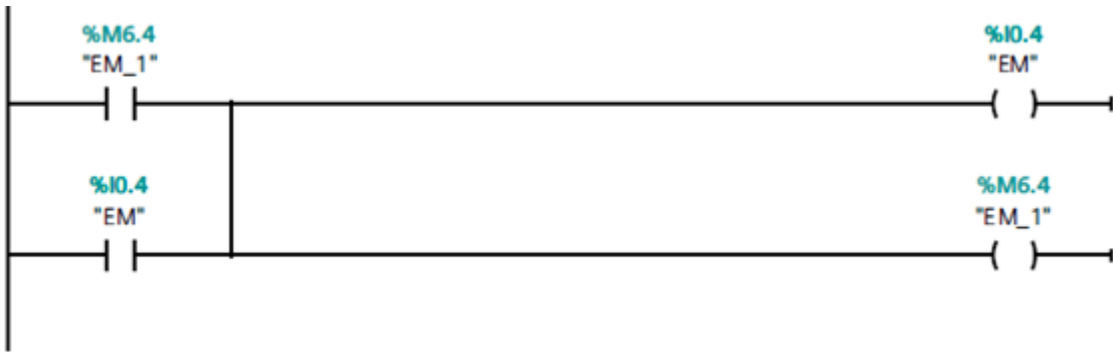
Network 5: Programa do Pórtico



➤ CONV

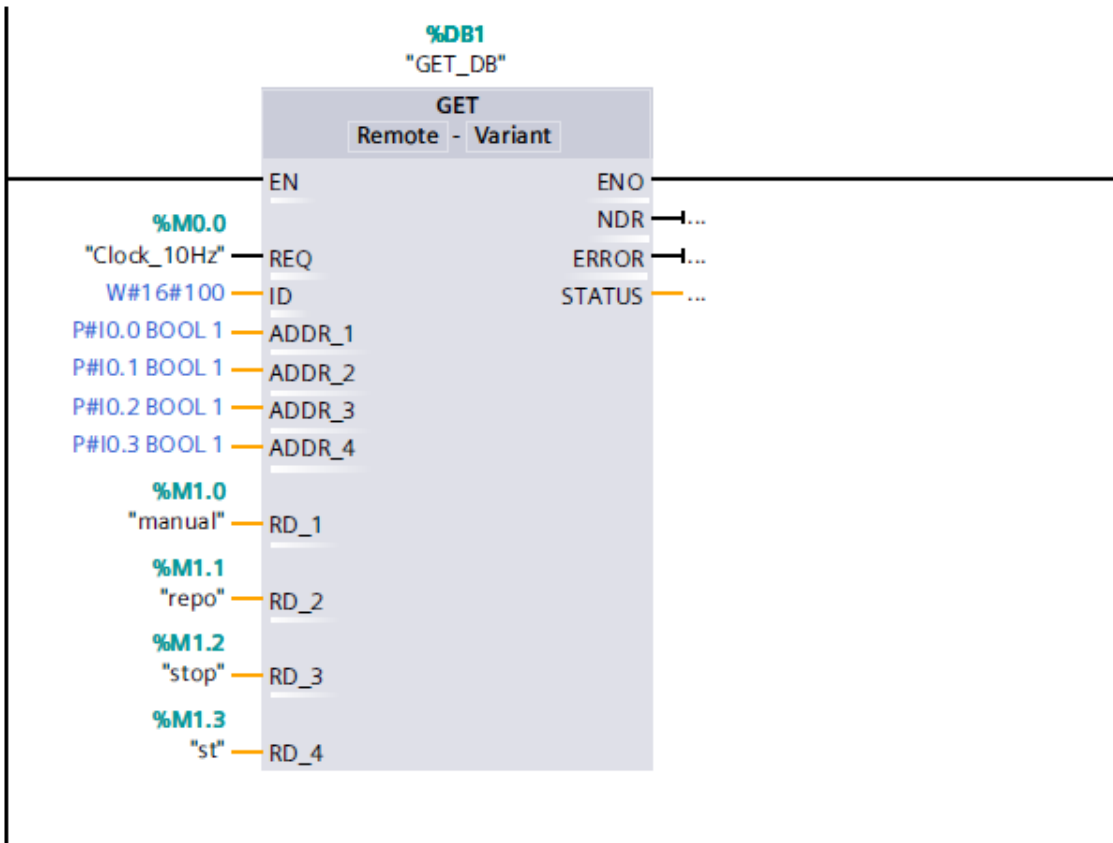
Network 1



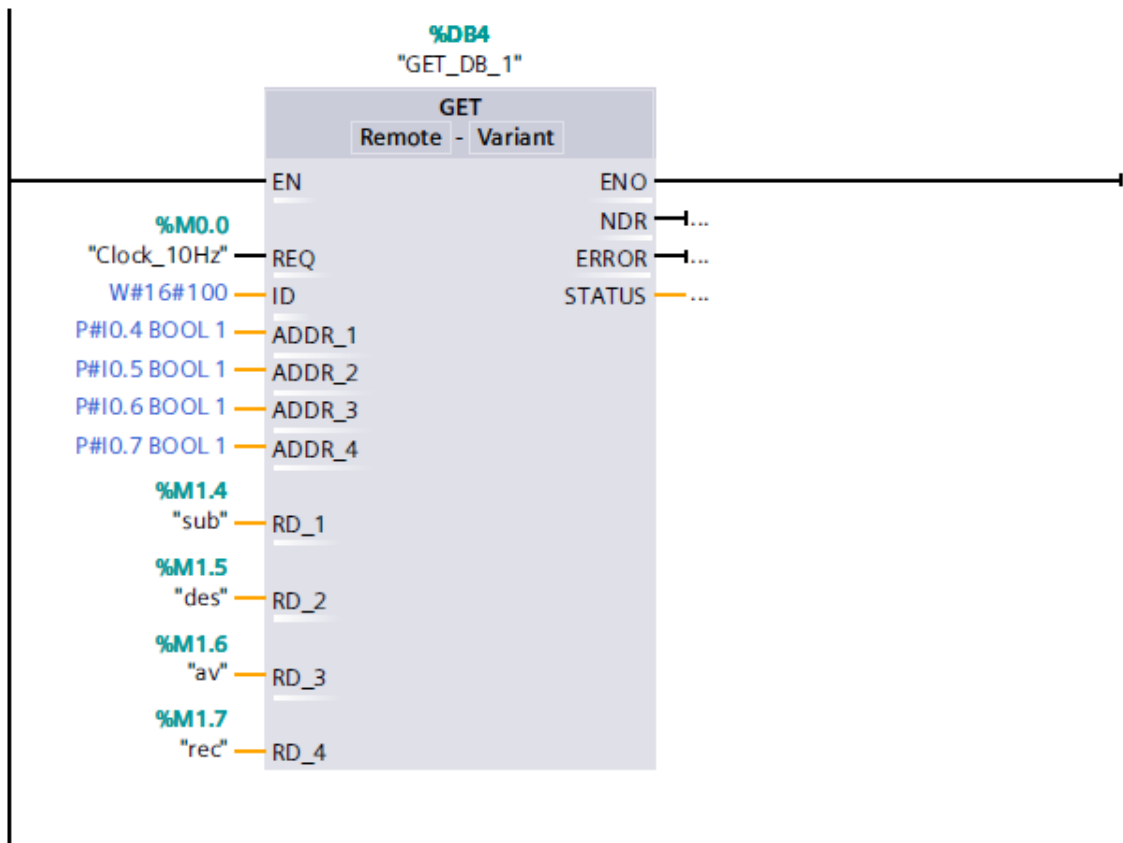


➤ **GET**

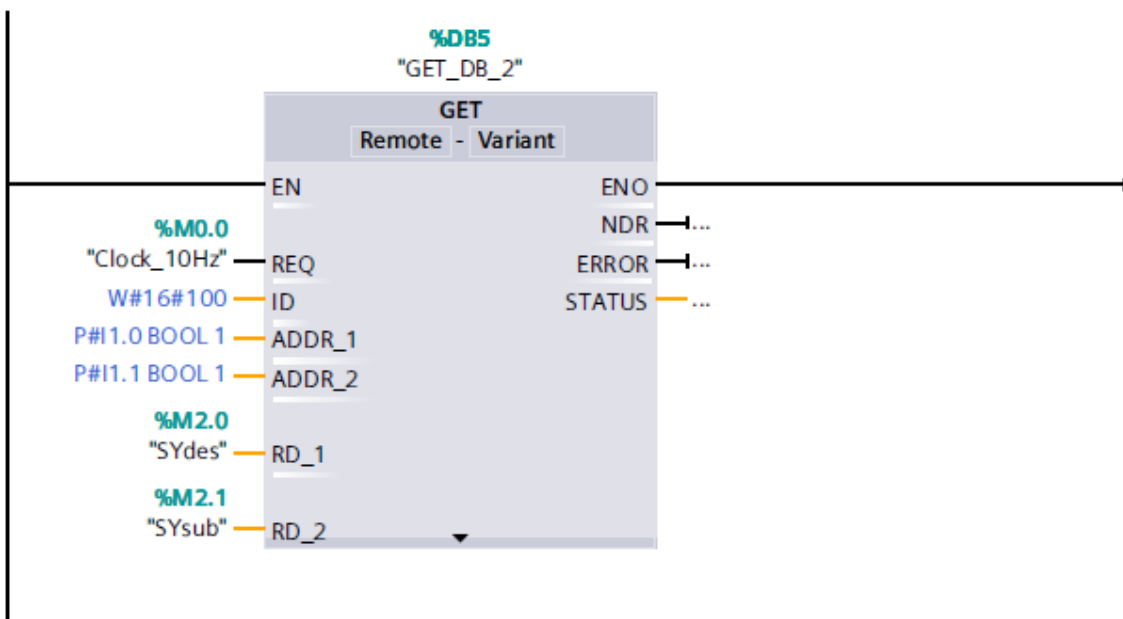
Network 1



Network 2

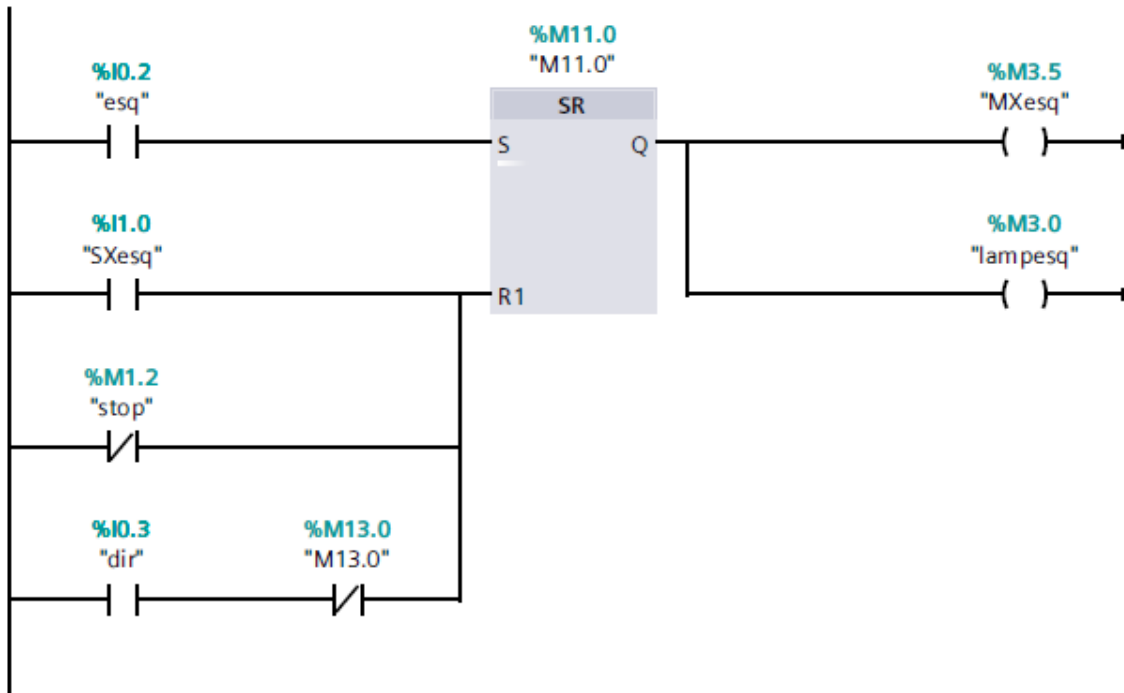


Network 3

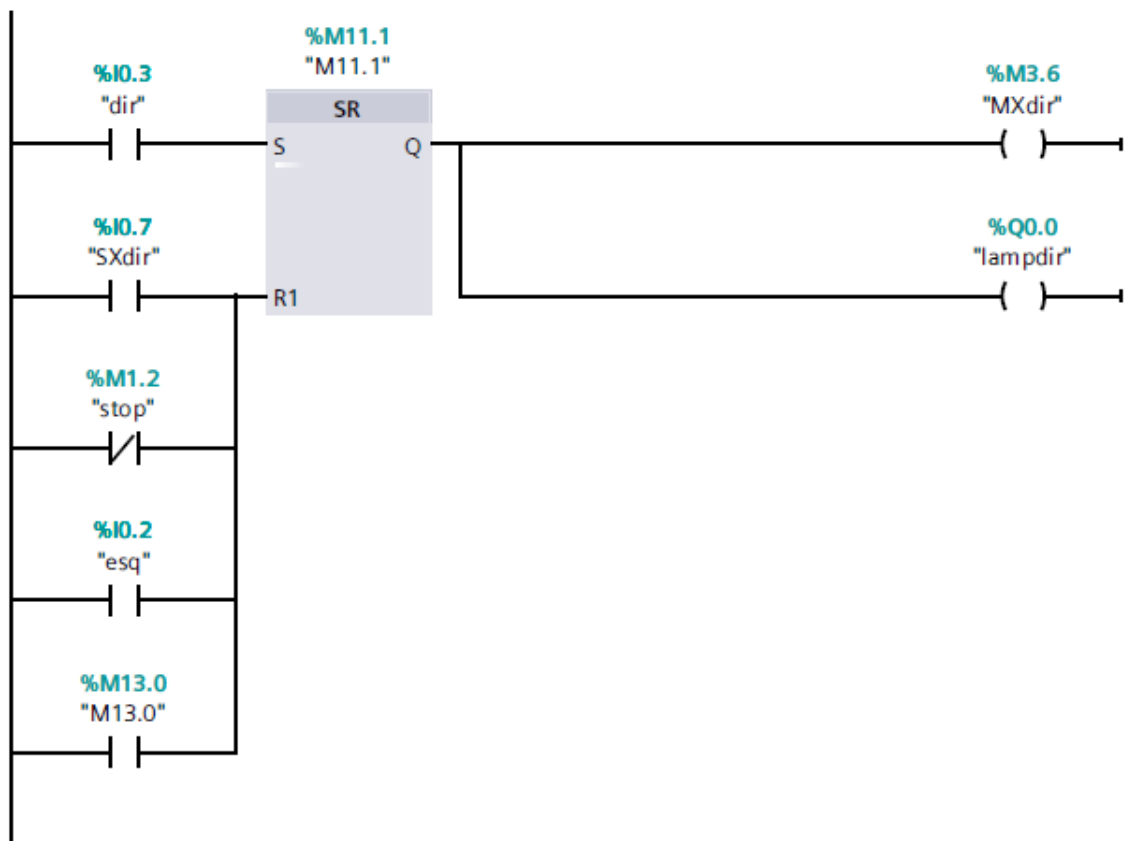


➤ **Portico**

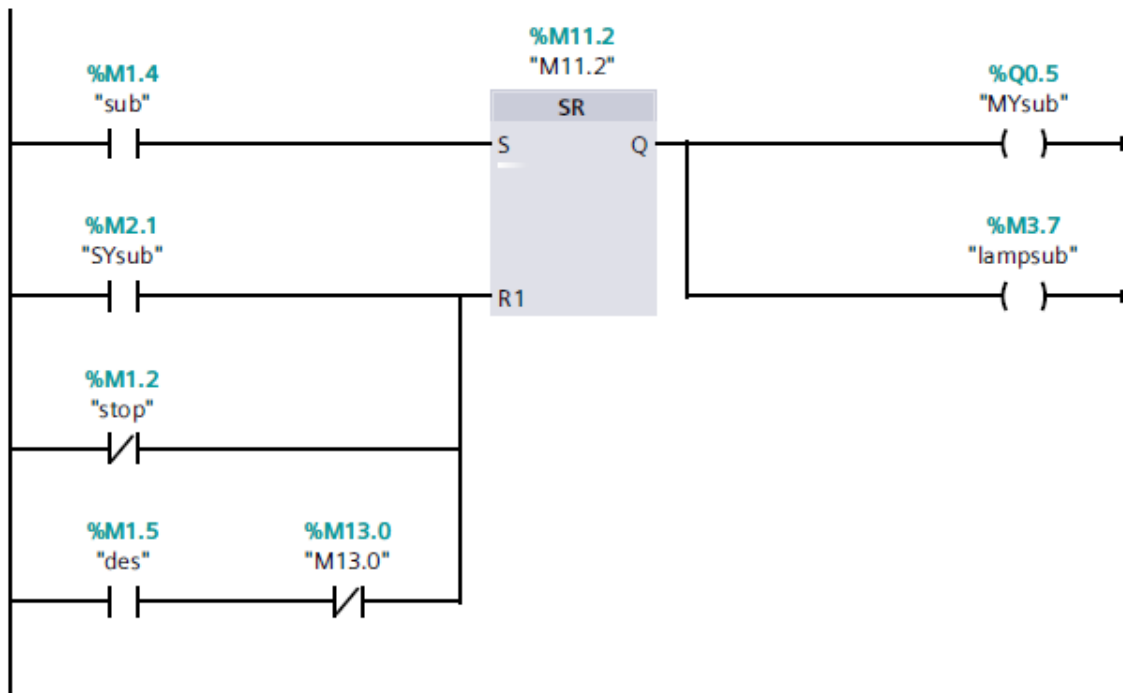
Network 1



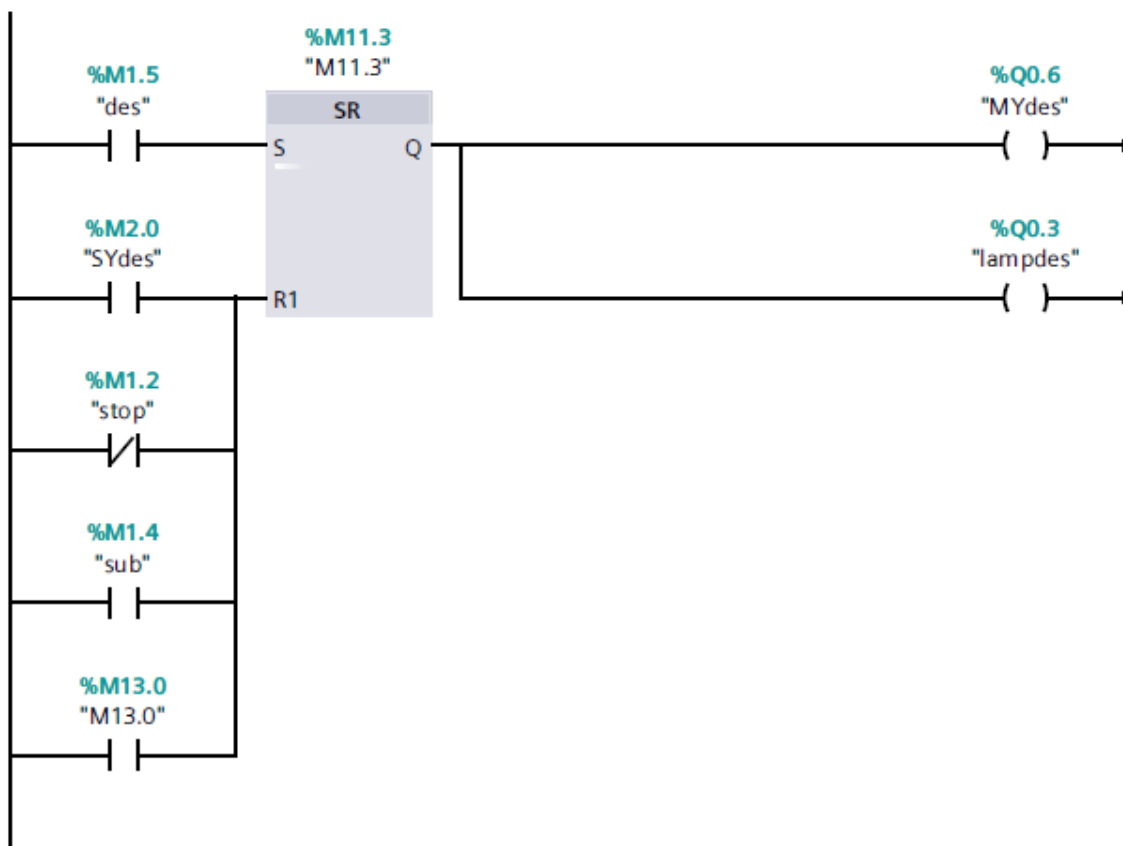
Network 2



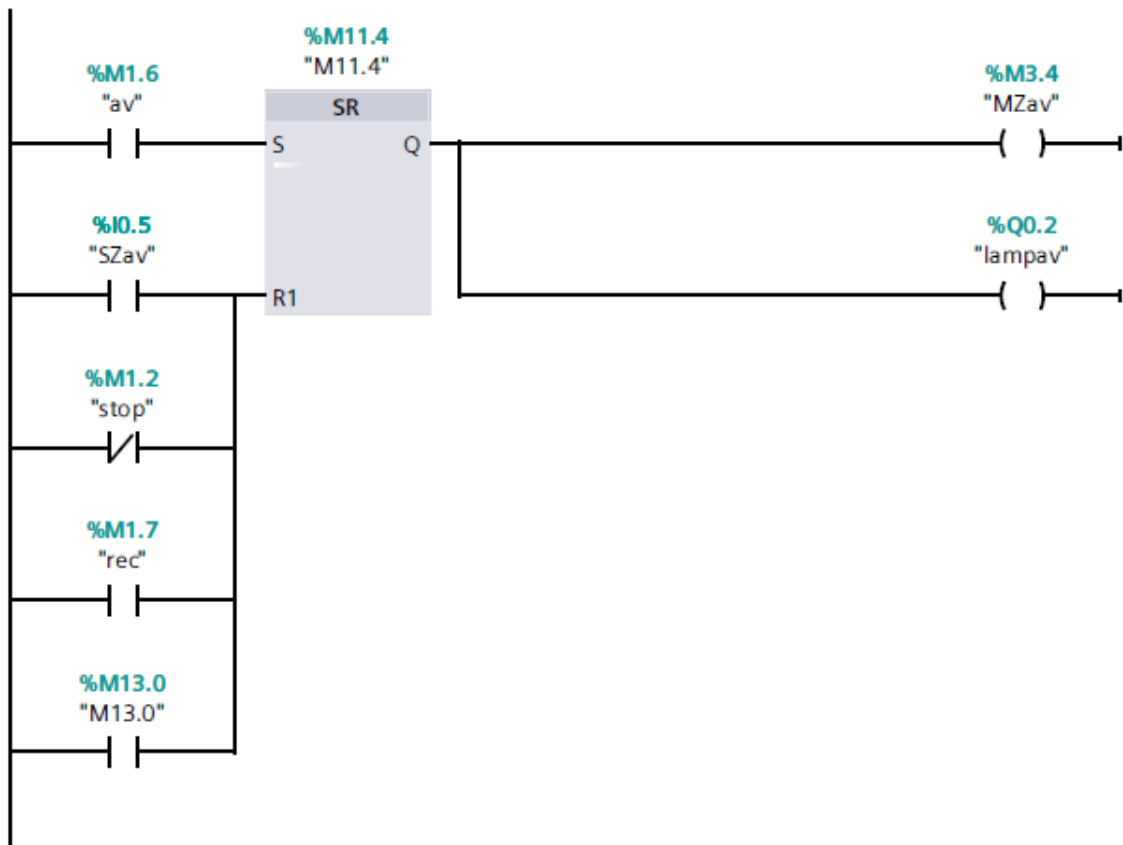
Network 3



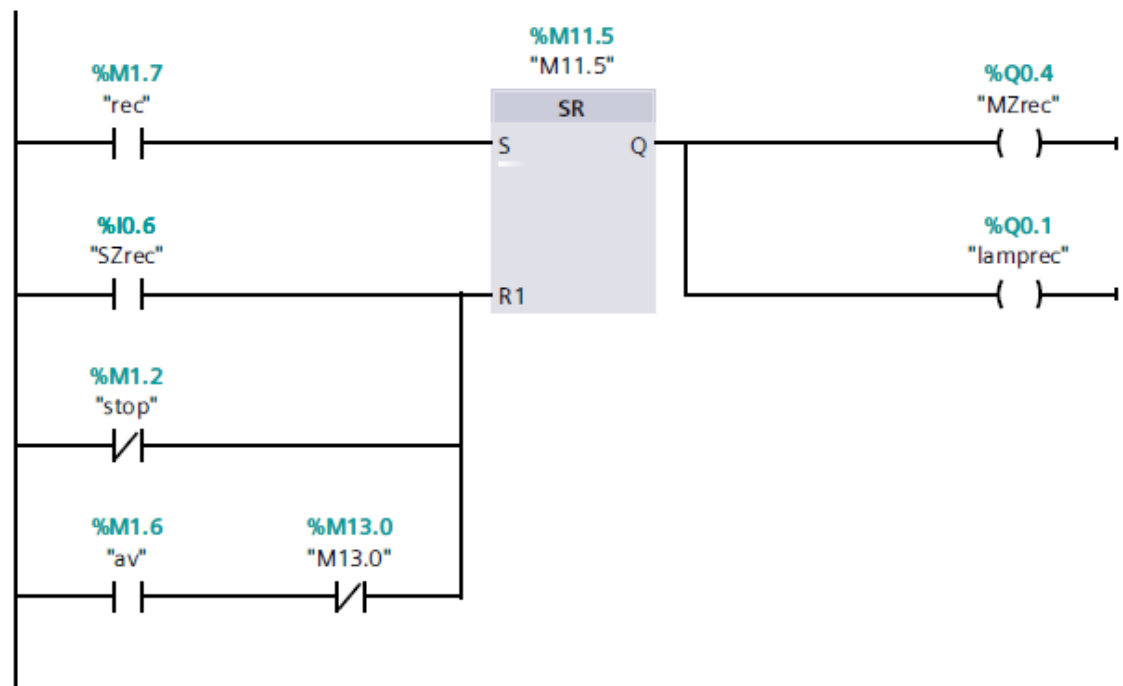
Network 4



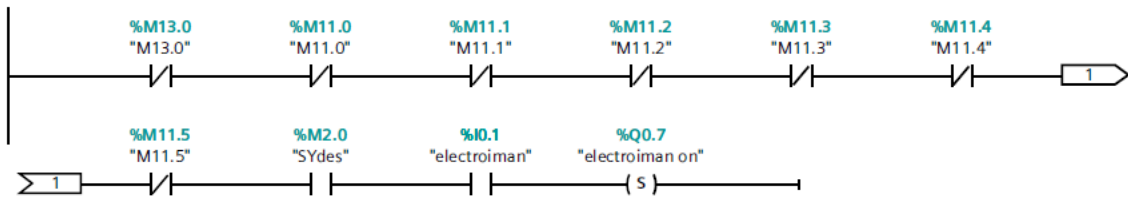
Network 5



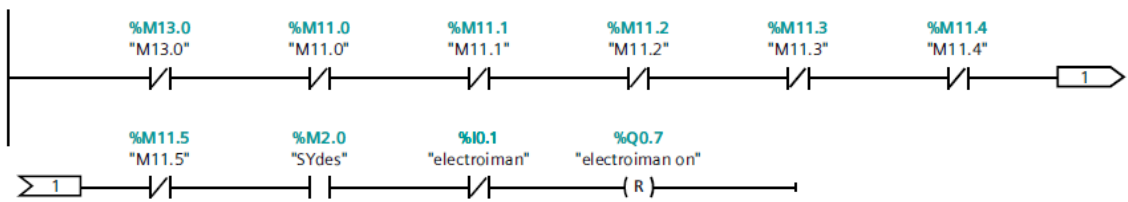
Network 6



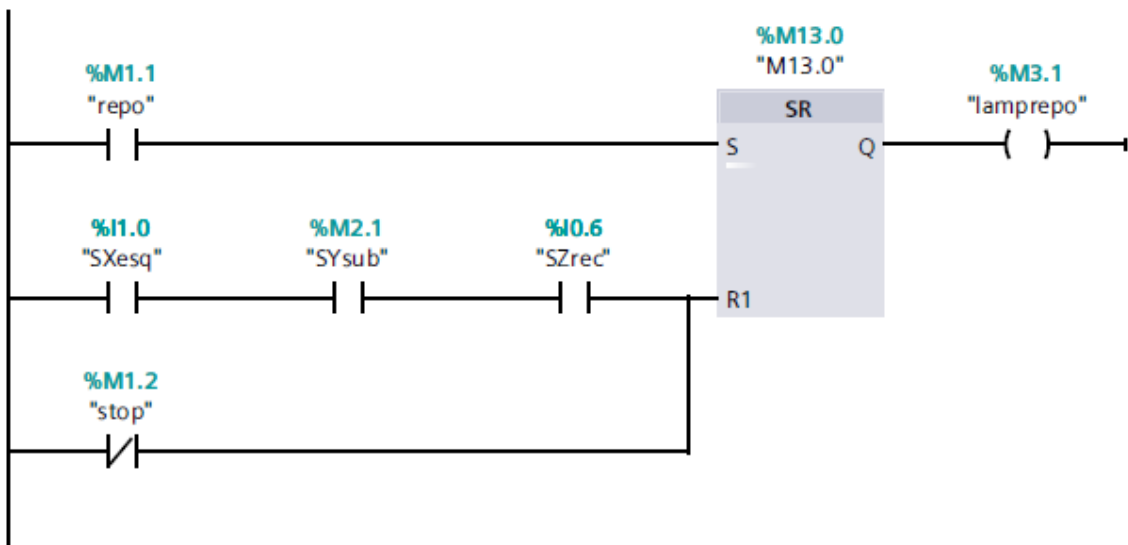
Network 7



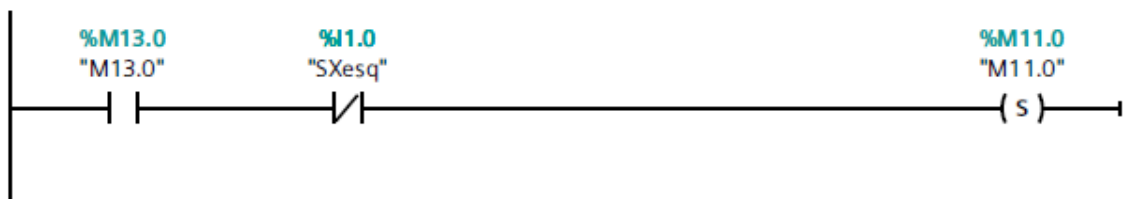
Network 8



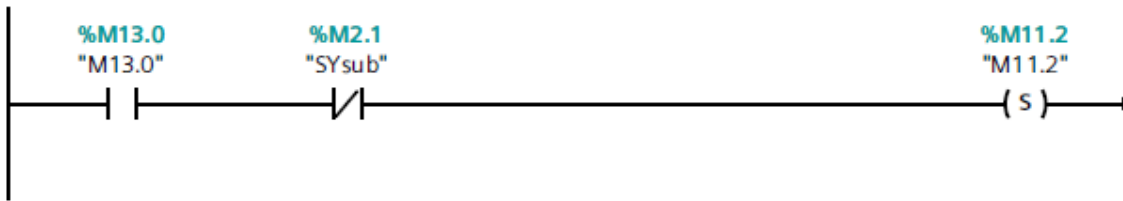
Network 9



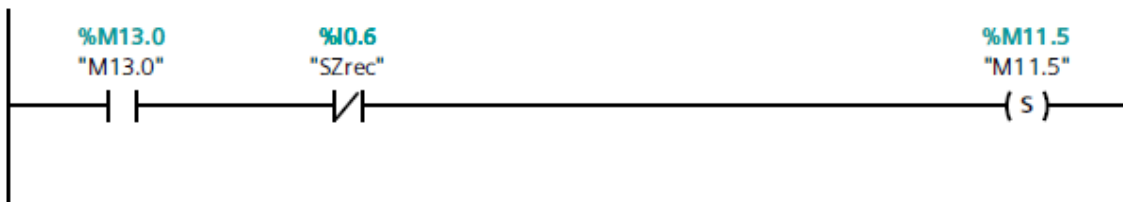
Network 10



Network 11

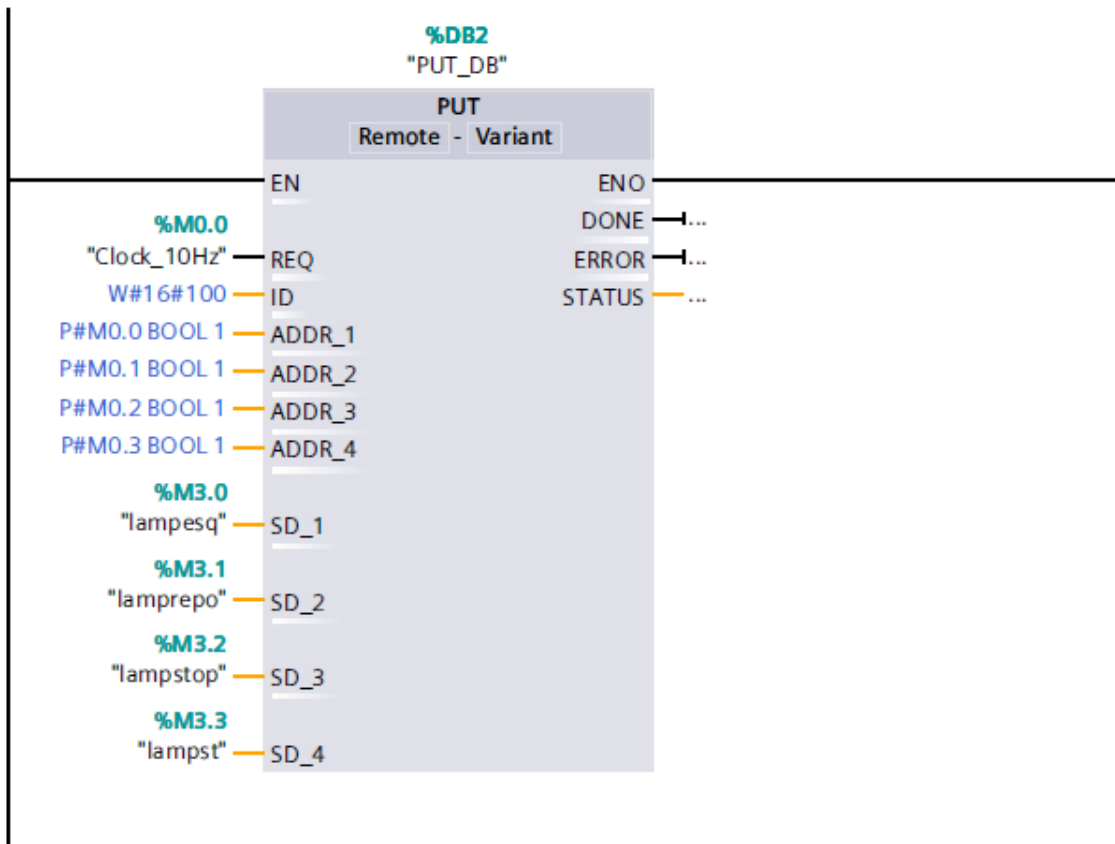


Network 12

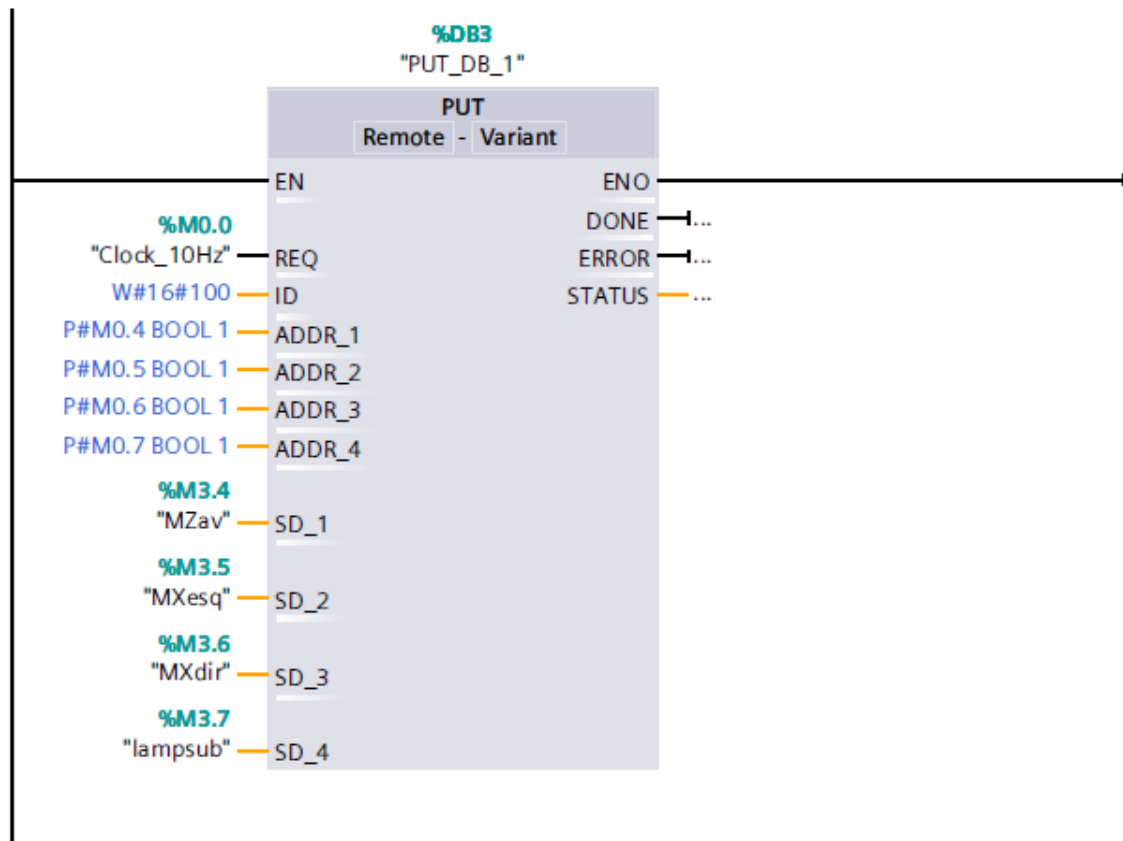


➤ **PUT**

Network 1



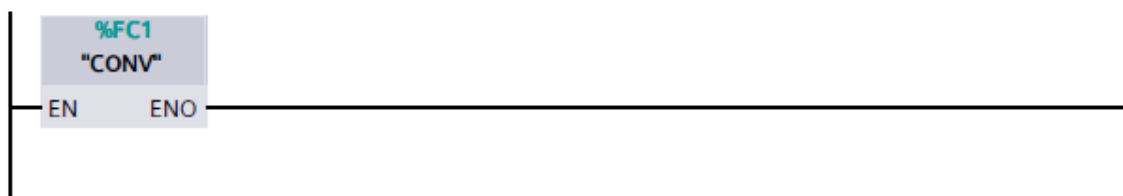
Network 2



PLC 2:

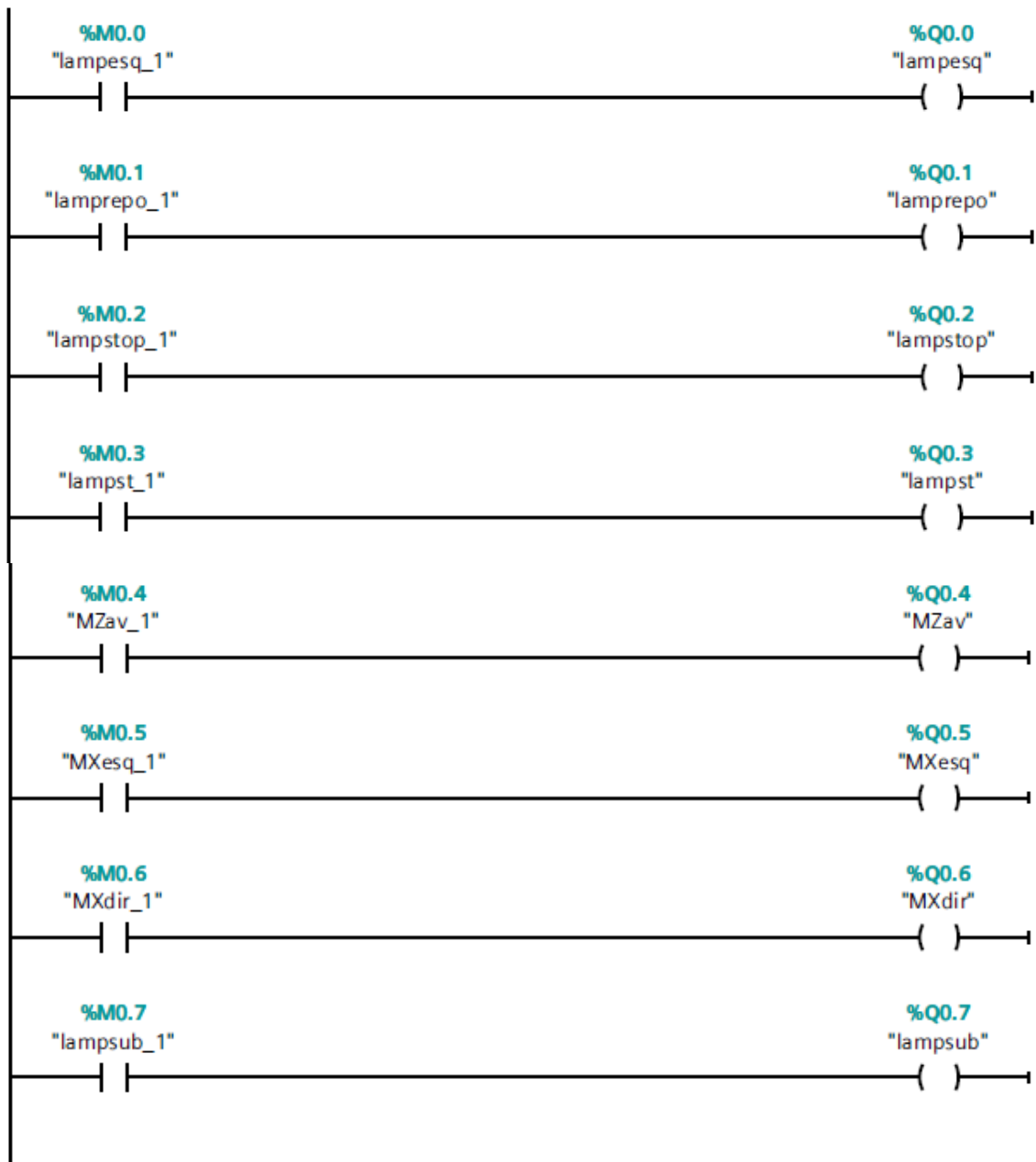
➤ Main

Network 1: O valor das memórias do PUT acciona os *outputs*



➤ CONV

Network 1



Anexo C

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<!-- AWP_In_Variable Name="auto_1" -->
<!-- AWP_In_Variable Name="electroiman_1" -->
<!-- AWP_In_Variable Name="esq_1" -->
<!-- AWP_In_Variable Name="dir_1" -->
<!-- AWP_In_Variable Name="EM_1" -->
<!-- AWP_In_Variable Name="manual" -->
<!-- AWP_In_Variable Name="repo" -->
<!-- AWP_In_Variable Name="stop" -->
<!-- AWP_In_Variable Name="st" -->
<!-- AWP_In_Variable Name="sub" -->
<!-- AWP_In_Variable Name="des" -->
<!-- AWP_In_Variable Name="av" -->
<!-- AWP_In_Variable Name="rec" -->
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<meta http-equiv="content-script-type" content="text/javascript">
<title>Pórtico</title>
<script type="text/javascript">
var animate_right;
var animate_left;
var animate_forward;
var animate_back;
var animate_up;
var animate_down;
var pos_0 = 0;
var down = 0;
var right = 0;
var left = 0;
var forward = 0;
var bck = 0;
function init(){
    imgObj1 = document.getElementById('img1');
    imgObj1.style.position= 'relative';
    imgObj1.style.left = '0px';

    imgObj2 = document.getElementById('img2');
    imgObj2.style.position= 'relative';
    imgObj2.style.left = '0px';
    imgObj2.style.top = '0px';

    imgObj3 = document.getElementById('img3');
    imgObj3.style.position= 'relative';
    imgObj3.style.top = '0px';
}
window.onload = init;
```

```

</script>
</head>

<body>
<div align="center">
</div>
<table width="801" height="442" border="1" align="center">
<tr>
<td height="105" colspan="9" rowspan="3" valign="top"> <div
style="width:611; height:342; table-layout:fixed; overflow:hidden">
<div style="position:relative; z-index:1"></div>

<div style="position:relative; left: 38; top:-342; z-index:3"></div>

<div style="position:relative; top:-435; right:-68; z-index:4"></div></div></td>
<td width="80" rowspan="3" align="center" valign="top"><div
style="width:80; height:226; table-layout:fixed; overflow:hidden"><div
style="position:relative; z-index:1;"></div>
<div style="position:relative; top: -140; z-index:2;"></div></div>
</td>
<td width="200" height="105" align="center">
<form method="post" action="move.html" target="_blank">
<p>
<input type="image" src="img\up.png" name='&quot;up&quot;'
onClick="moveUp()" alt="Submit">
</p>
<p>
<input type="hidden" name='&quot;sub&quot;' value="1">
</p>
</form></td>
</tr>
<tr>
<td height="119" align="center">
<table width="86" border="0">
<tr>
<td width="38" height="47" align="center"><form method="post"
action="move.html" target="_blank">
<input type="submit" name="&quot;electroiman_1&quot;" value="ON"
onClick="ImageOn()"/>
<input type="hidden" name="&quot;electroiman_1&quot;" value="1"/>
</form></td>
<td width="38" align="center"><form method="post" action="move.html"
target="_blank">
<input type="submit" name="&quot;electroiman_1&quot;" value="OFF"
onClick="ImageOff()"/>
<input type="hidden" name="&quot;electroiman_1&quot;" value="0"/>
</form></td>

```

```

        </tr>
    </table>
<script>
function ImageOn(){
    if (down == 1 & right == 0 & left == 0 & frward == 0 & bck == 0){
document.getElementById("magnet").src =
"http://10.68.13.173/awp/P%C3%B3rtico/img%5Cmagnet1.png";
    }
    }
function ImageOff(){
    if (down == 1 & right == 0 & left == 0 & frward == 0 & bck == 0){
document.getElementById("magnet").src =
"http://10.68.13.173/awp/P%C3%B3rtico/img%5Cmagnet0.png";
    }
    }
</script>
</td>
</tr>
<tr>
    <td height="115" align="center"><form method="post" action="move.html"
target="_blank">
        <p>
            <input type="image" src="img\down.png" name='&quot;down&quot;'
onClick="moveDown()" alt="Submit">
        </p>
        <p>
            <input type="hidden" name='&quot;des&quot;' value="1">
        </p>
    </form></td>
</tr>
<tr align="center">
    <td width="70" height="85"><form method="post" action="move.html"
target="_blank"><input type="image" src="img\stop.png" name='&quot;stop&quot;'
onClick="moveStop()" alt="Submit">
        <input type="hidden" name='&quot;stop&quot;' value="0">
    </form></td>
    <td width="62"><form method="post" action="move.html" target="_blank">
        <input type="image" src="img\home.png" name='&quot;home&quot;'
onClick="moveHome()" alt="Submit">
        <input type="hidden" name='&quot;repo&quot;' value="1">
    </form></td>
    <td width="89">&nbsp;</td>
    <td width="67">
<form method="post" action="move.html" target="_blank">
        <input type="image" src="img\left.png" name='&quot;left&quot;'
onClick="moveLeft()" alt="Submit">
        <input type="hidden" name='&quot;esq_1&quot;' value="1">
    </form>
</td>
    <td width="67">

```

```

<form method="post" action="move.html" target="_blank">
  <input type="image" src="img\right.png" name='&quot;right&quot;'
onClick="moveRight()" alt="Submit">
  <input type="hidden" name='&quot;dir_1&quot;' value="1">
</form>
</td>
  <td width="63">
<form method="post" action="move.html" target="_blank">
  <input type="image" src="img\back.png" name='&quot;back&quot;'
onClick="moveBack()" alt="Submit">
  <input type="hidden" name='&quot;rec&quot;' value="1">
</form></td>
  <td width="63"><form method="post" action="move.html" target="_blank">
  <input type="image" src="img\forward.png" name='&quot;forward&quot;'
onClick="moveForward()" alt="Submit">
  <input type="hidden" name='&quot;av&quot;' value="1">
</form></td>
  <td width="88">&nbsp;</td>
  <td colspan="3">&nbsp;</td>
</tr>
</table>
<script type="text/javascript">
function moveRight(){
if (pos_0 == 0){
  imgObj1.style.left = parseInt(imgObj1.style.left) + 1 + 'px';
  imgObj2.style.left = parseInt(imgObj2.style.left) + 1 + 'px';
  animate_right = setTimeout(moveRight, 29.5);
  clearTimeout(animate_left);
  right = 1;
  left = 0;
  var lim_XX = parseInt(imgObj1.style.left);
if (lim_XX >= 446){
  clearTimeout(animate_right);
  right = 0;
}
}
}
function moveLeft(){
  imgObj1.style.left = parseInt(imgObj1.style.left) - 1 + 'px';
  imgObj2.style.left = parseInt(imgObj2.style.left) - 1 + 'px';
  animate_left = setTimeout(moveLeft, 29.5);
  clearTimeout(animate_right);
  left = 1;
  reight = 0;
  var lim_XX = parseInt(imgObj1.style.left);
if (lim_XX <= 0){
  clearTimeout(animate_left);
  Zero();
  left = 0;
}
}

```

```

}
function moveForward(){
if (pos_0 == 0){
  imgObj2.style.top = parseInt(imgObj2.style.top) - 2 + 'px';
  animate_forward = setTimeout(moveForward, 67);
  clearTimeout(animate_back);
  forward = 1;
  bck = 0;
  var lim_ZZ = parseInt(imgObj2.style.top);
if (lim_ZZ <= -216){
  clearTimeout(animate_forward);
  forward = 0;
}
}
}
function moveBack(){
  imgObj2.style.top = parseInt(imgObj2.style.top) + 2 + 'px';
  animate_back = setTimeout(moveBack, 67);
  clearTimeout(animate_forward);
  bck = 1;
  forward = 0;
  var lim_ZZ = parseInt(imgObj2.style.top);
if (lim_ZZ >= 0){
  clearTimeout(animate_back);
  Zero();
  bck = 0;
}
}
function moveUp(){
  imgObj3.style.top = parseInt(imgObj3.style.top) - 1 + 'px';
  animate_up = setTimeout(moveUp, 49.5);
  clearTimeout(animate_down);
  var lim_YY = parseInt(imgObj3.style.top);
  down = 0;
if (lim_YY <= 0){
  clearTimeout(animate_up);
  Zero();
}
}
function moveDown(){
if (pos_0 == 0){
  imgObj3.style.top = parseInt(imgObj3.style.top) + 1 + 'px';
  animate_down = setTimeout(moveDown, 49.5);
  clearTimeout(animate_up);
  var lim_YY = parseInt(imgObj3.style.top);
if (lim_YY >= 60){
  clearTimeout(animate_down);
  down = 1;
}
}
}

```

```

}
function moveStop(){
  clearTimeout(animate_right);
  clearTimeout(animate_left);
  clearTimeout(animate_forward);
  clearTimeout(animate_back);
  clearTimeout(animate_up);
  clearTimeout(animate_down);
  right = 0;
  left = 0;
  forward = 0;
  bck = 0;
  pos_0 = 0;
}
function moveHome(){
  pos_0 = 1;
  moveLeft();
  moveBack();
  moveUp();
}
function Zero(){
  var lim_XX = parseInt(imgObj1.style.left);
  var lim_ZZ = parseInt(imgObj2.style.top);
  var lim_YY = parseInt(imgObj3.style.top);
  if (lim_XX <= 0 & lim_ZZ >= 0 & lim_YY <= 0){
    pos_0 = 0;
  }
}
</script>
</form>
</body>
</html>

```