

POLITÉCNICO DO PORTO  
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

---

## **LaRE - Laboratório Remoto Expansível**

---

**Eduardo Manuel Martins Ramalhadeiro**

Mestrado em Engenharia Electrotécnica e de Computadores  
Área de Especialização em Automação e Sistemas



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA  
Instituto Superior de Engenharia do Porto

Junho, 2025



*Esta dissertação satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores, Área de Especialização em Automação e Sistemas.*

**Candidato:** Eduardo Manuel Martins Ramalhadeiro, N° 1210171,  
1210171@isep.ipp.pt

**Orientação Científica:** André Vaz Fidalgo, anf@isep.ipp.pt



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA  
Instituto Superior de Engenharia do Porto  
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Junho, 2025



*Para ti, mãe*

*“( . . . ) às vezes ainda sou o menino  
que adormeceu nos teus olhos”*

*Eugénio de Andrade*



# Agradecimentos



# Resumo

Aqui deverá ser apresentado o resumo de todo o trabalho efetuado. Esta secção não deverá exceder uma página.

Deve contextualizar o problema que pretende resolver ou a hipótese que irá formular, procure evidenciar as vantagens e desvantagens (se as houver) da solução encontrada, como também a forma através da qual a solução/hipótese foi validada. Neste último ponto, deverá referir-se aos desenvolvimentos efetuados, e à forma como validou (conformidade) e avaliou (desempenho) a solução encontrada.

A dissertação deve conter sempre duas versões do resumo: uma primeira no idioma do texto principal e a segunda num outro idioma. Este *template* assume que os dois idiomas em consideração são sempre Português e Inglês, assim, a classe irá colocar os cabeçalhos respetivos de acordo com o idioma selecionado nas opções da classe no ficheiro `main.tex`.

**Palavras-Chave:** Lista, separada por vírgulas, de palavras, frases, ou acrónimos chave no âmbito do trabalho descrito neste texto.



# Abstract

The summary of all the developed work should be presented here. This section should not exceed one page.

Start the abstract with the contextualization of the problem you intend to solve or the hypothesis you will formulate. Try to highlight the advantages and disadvantages (if any) of the solution found, as well as the way in which the solution/hypothesis was validated. In this last point, you should refer to the developments made, and to the way you validated (compliance) and evaluated (performance) the solution found.

The dissertation must always contain two versions of the abstract: a first in the language of the main text and the second one in another language. This template assumes that the two languages are always Portuguese and English, therefore, the class will place the correct section headers according to the language selected in the class options in the `main.tex` file.

**Keywords:** Comma separated list of words, phrases, or key acronyms within the scope of your developed work.



# Índice

<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>Listagens</b>	<b>xv</b>
<b>Glossário</b>	<b>xvii</b>
<b>Lista de Acrónimos</b>	<b>xxi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento . . . . .	1
1.2 Objectivos . . . . .	2
1.3 Estrutura do documento . . . . .	3
<b>2 Estado da Arte</b>	<b>5</b>
2.1 Transformação digital na educação . . . . .	5
2.2 Educação Digital em Portugal: Avanços e Desafios . . . . .	9
2.3 eLearning e abordagem STEM . . . . .	11
2.4 Laboratórios na educação . . . . .	16
2.4.1 Tipos de laboratórios . . . . .	18
Uma questão de conceitos . . . . .	21
Laboratório real . . . . .	22
Laboratório virtual - Acesso local e/ou remoto . . . . .	25
Laboratório remoto . . . . .	30
VISIR . . . . .	39
PILAR . . . . .	44
2.5 COMENTÁRIO PROF . . . . .	45
<b>3 LaRE - Laboratório Remoto Expansível</b>	<b>47</b>
3.1 Contextualização . . . . .	47
3.2 Solução proposta . . . . .	50
3.3 Arquitectura . . . . .	51
3.4 Circuitos electrónicos - experiências? . . . . .	52
3.4.1 Lei de <i>Ohm</i> . . . . .	53

3.4.2	Rectificadores . . . . .	54
3.4.3	Filtros . . . . .	56
3.5	Hardware . . . . .	58
3.5.1	VirtualBench . . . . .	58
3.5.2	Matriz LaRE . . . . .	59
	Lei de <i>Ohm</i> . . . . .	59
	Rectificadores/Filtros . . . . .	60
	Fontes de alimentação . . . . .	61
3.5.3	Raspberry Pi . . . . .	61
3.5.4	Relés . . . . .	63
3.5.5	<i>Driver</i> de Relés . . . . .	64
3.5.6	Registo de deslocamento . . . . .	65
3.6	Software . . . . .	66
3.6.1	<i>Back-End</i> . . . . .	68
	<i>Python</i> . . . . .	68
	<i>Flask</i> . . . . .	68
	<i>Jinja2</i> . . . . .	69
3.6.2	Front-End . . . . .	70
	<i>Webpage</i> . . . . .	70
<b>4</b>	<b>Implementação</b>	<b>75</b>
4.1	Hardware . . . . .	76
4.1.1	Relés . . . . .	76
4.1.2	Driver de relés . . . . .	76
4.1.3	Registo de deslocamento . . . . .	77
4.1.4	Fontes de alimentação . . . . .	77
	Fonte de tensão 6 V - variável e Fonte de tensão 12 V - fixa .	79
	Fonte de tensão 5 V - fixa . . . . .	79
	Fontes de tensão 220 V/8 V CA . . . . .	81
4.1.5	Aparelhos de medida - voltímetro e amperímetro . . . . .	82
4.1.6	Experiências . . . . .	83
	Lei de Ohm . . . . .	84
	Rectificadores e filtros . . . . .	85
4.2	Software . . . . .	86
4.2.1	Servidor Flask . . . . .	87
	Estrutura base . . . . .	87
	Rotas . . . . .	89
	Blueprints . . . . .	90
	Pedidos . . . . .	91
	Base de dados . . . . .	91

Autenticação . . . . .	92
4.2.2 Arquitectura de Comunicação . . . . .	92
Comunicação entre ficheiros . . . . .	94
Comunicação com RaspberryPi . . . . .	96
VirtualBench - Configurações e medições . . . . .	97
Gráficos . . . . .	100
4.2.3 Interface WEB . . . . .	101
Lei de <i>Ohm</i> . . . . .	103
Rectificador de meia onda e onda completa . . . . .	105
Filtros . . . . .	106
<b>5 Resultados</b>	<b>109</b>
5.1 Lei de <i>Ohm</i> . . . . .	110
5.1.1 Resultados experimentais . . . . .	110
5.2 Rectificadores . . . . .	112
5.2.1 Resultados experimentais [meia onda] . . . . .	112
5.2.2 Resultados experimentais [onda completa] . . . . .	114
5.3 Filtros . . . . .	114
5.3.1 Resultados experimentais [filtro passa-alto] . . . . .	114
5.4 Limitações . . . . .	117
5.5 Melhoramentos . . . . .	117
<b>Referências</b>	<b>118</b>
<b>Anexo A Datasheet</b>	<b>135</b>



# Listas de Figuras

2.1	Tipos de laboratórios consoante a localização, segundo Zutin, <i>et al.</i> [57] . . . . .	19
2.2	Tipos de laboratórios consoante a localização, Heradio, de la Torre & Dormido, (2016) [54] . . . . .	20
2.3	LV <i>vs</i> simulador . . . . .	21
2.4	<i>Fritzing</i> (Exemplo) . . . . .	22
2.5	Tinkercad . . . . .	25
2.6	<i>Multisim</i> local . . . . .	26
2.7	<i>Multisim</i> remoto . . . . .	26
2.8	<i>Falstad</i> - amplificador inversor[87] . . . . .	30
2.9	Diferentes tipos de arquitectura (resumo) . . . . .	34
2.10	Implementação cliente-servidor MATLAB-EJS . . . . .	35
2.11	Arquitectura DIESEL (genérica)[107] . . . . .	36
2.12	Arquitectura ISA[112] . . . . .	37
2.13	Arquitectura <i>WebLab-Deusto</i> . . . . .	38
2.14	Arquitectura <i>gateway4labs</i> . . . . .	39
2.15	VISIR no ISEP . . . . .	39
2.16	Arquitectura VISIR[122] . . . . .	41
2.17	<i>Protoboard VISIR</i> . . . . .	41
2.18	<i>Chassis</i> PXI-1033 [123] . . . . .	41
2.19	Osciloscópio PXI-5114 [124] . . . . .	42
2.20	Gerador de sinal PXI-5402 [125] . . . . .	42
2.21	Multímetro Digital PXI-4072 [126] . . . . .	42
2.22	Fonte de tensão PXI-4110 [127] . . . . .	43
2.23	Matriz VISIR[128] . . . . .	43
3.1	<i>Arduino</i> Mega [136] . . . . .	48
3.2	<i>ESP32</i> [143] . . . . .	49
3.3	<i>RaspberryPI5</i> [144] . . . . .	49
3.4	Exemplo do VB usado como multímetro digital . . . . .	50
3.5	Arquitectura LaRE . . . . .	51
3.6	Fluxograma de uma chamada ao <i>socket</i> [147] . . . . .	52
3.7	Arquitectura LaRE - NOTE: Fig tem de ser revista . . . . .	53

3.8	Gráfico da Lei de <i>Ohm</i>	54
3.9	Experiência Lei de <i>Ohm</i>	54
3.10	Diagrama de blocos da rectificação [meia onda e onda completa]	55
3.11	Forma de onda à saída do Bloco 3 - Filtragem [meia onda] [148]	56
3.12	Forma de onda à saída do Bloco 3 - Filtragem [onda completa] [148]	56
3.13	Esquemas dos filtros [148]	57
3.14	Diagramas de <i>Bode</i> [ideal] [148]	57
3.15	Painel traseiro <i>VirtualBench VB-8012</i> [149]	58
3.16	Painel frontal <i>VirtualBench VB-8012</i> [149]	58
3.17	Matriz do LaRE	59
3.18	[Placa] Lei de <i>Ohm</i>	60
3.19	[Placa] rectificadores e filtros	60
3.20	[Placa] fontes de alimentação	61
3.21	<i>Raspberry PI5</i> com dissipador activo utilizado no LaRE	62
3.22	GPIOs <i>RaspberryPI</i> [151]	62
3.23	Representação alternativa dos GPIOs [151]	63
3.24	<i>Esquema simplificado de um relé</i> [153]	63
3.25	<i>Relés SPST e DPST</i> [153]	64
3.26	Diagrama de blocos <i>ULN2003A</i> [154]	65
3.27	Diagrama [em corte] de blocos <i>SN74HC595</i> [155]	67
3.28	Estrutura de uma página HTML [164]	70
4.1	Exemplo de utilização de relés SPST e DPST	76
4.2	Diagrama de blocos simplificado do <i>ULN2003A</i>	76
4.3	Exemplo de uso do <i>SN74HC595</i> e <i>ULN2003A</i>	77
4.4	Envio de <i>bits</i> para o registo de deslocamento	78
4.5	Fontes de tensão do VB	78
4.6	Implementação Fontes de 6 V e 12 V	79
4.7	Fonte de 5 V [ <i>LM317</i> ]	80
4.8	Problema de massa - onda completa	81
4.9	Problema de massa - onda completa [Resolvido]	82
4.10	Painel frontal Multímetro Digital	82
4.11	Esquema completo - Lei de <i>Ohm</i>	83
4.12	[Implementação] Lei de <i>Ohm</i>	84
4.13	Esquema completo (simplificado) - rectificadores e filtros	85
4.14	[Implementação] rectificadores e filtros	86
4.15	Funcionamento geral <i>Flask</i>	87
4.16	Estrutura de directórios - <i>Flask main.html está a mais</i>	88
4.17	Diagrama de comunicação simplificado [REVER]	93
4.18	Processo de leitura das medições de tensão e corrente	95

4.19	Erro de selecção . . . . .	102
4.20	Página de <i>login</i> . . . . .	102
4.21	Página inicial . . . . .	103
4.22	[Exemplo] Experiência Lei de <i>Ohm</i> . . . . .	103
4.23	Lei de <i>Ohm</i> - Selecção dos parâmetros . . . . .	104
4.24	Selecção de parâmetros . . . . .	105
4.25	Selecção de parâmetros . . . . .	106
5.1	Medição prática . . . . .	110
5.2	Gráfico da Lei de <i>Ohm</i> - 1 k $\Omega$ - LaRE . . . . .	111
5.3	Multisim - Lei de <i>Ohm</i> . . . . .	112
5.4	Tensão de <i>ripple</i> no LaRE- meia onda . . . . .	113
5.5	Valores experimentais e simulados - tensão de <i>ripple</i> . . . . .	113
5.6	Frequência de corte - filtro passa-alto - Diagrama de <i>Bode</i> . . . . .	115
5.7	Frequência de corte - <i>multisim</i> - Diagrama de <i>Bode</i> . . . . .	115
5.8	$v_{out}$ vs $v_{in}$ experimental LaRE - TEMPORÁRIO . . . . .	116
5.9	Valores prácticos e simulados - $v_{out}$ vs $v_{in}$ . . . . .	116
5.10	Frequência de corte - LaRE . . . . .	117
5.11	Valores experimentais reais e simulados - frequência de corte - $v_{out}$ vs $v_{in}$ . . . . .	117



# **Lista de Tabelas**

3.1	Raspberry PI4 <i>vs</i> Raspberry PI5 - <i>principais diferenças</i> [152]	62
3.2	Modos de funcionamento do SN74HC595 [155]	67
4.1	Exemplo funcionamento de medição da Lei de Ohm	83
4.2	Exemplo funcionamento do rectificador de meia onda	86
5.1	Valores prácticos - Lei de <i>Ohm</i>	111



# Listagens

3.1	Exemplo <i>Jinja</i> incluído na página <i>base.html</i> . . . . .	69
3.2	Exemplo CSS incluído na página HTML [167] . . . . .	71
3.3	Exemplo da página HTML com o CSS definida externamente [167] .	72
3.4	Exemplo do CSS definido externamente [167] . . . . .	72
3.5	Exemplo de <i>JavaScript</i> . . . . .	73
4.1	<i>Decorator route()</i> - <i>views.py</i> . . . . .	89
4.2	Construção de <i>urls</i> - <i>auth.py</i> . . . . .	90
4.3	<i>Blueprint views</i> - <i>views.py</i> . . . . .	90
4.4	<i>Blueprint auth</i> - <i>auth.py</i> . . . . .	90
4.5	Registo das <i>blueprints</i> - <i>__init__.py</i> . . . . .	91
4.6	Exemplo argumentos passados ao servidor - <i>ohm.html</i> . . . . .	91
4.7	Envio de parâmetros HTML → <i>views.py</i> . . . . .	94
4.8	Recepção de parâmetros . . . . .	94
4.9	Envio de parâmetros <i>views.py</i> → HTML . . . . .	94
4.10	Recepção de parâmetros . . . . .	94
4.11	Comunicação <i>python</i> - <i>python</i> . . . . .	95
4.12	<i>Block Mode [Sockets]</i> <i>configRelays.py</i> . . . . .	96
4.13	<i>And bit bit shift_register.py</i> . . . . .	97
4.14	Instanciação da classe <i>PyVirtualBench</i> . . . . .	98
4.15	Exemplo <i>ps_example.py</i> . . . . .	99
4.16	Cálculo do declive da recta . . . . .	100
4.17	Gráfico Corrente vs Tensão . . . . .	100
4.18	Filtros - Diagrama de <i>Bode</i> . . . . .	101
4.19	Gravação do gráfico . . . . .	101



# Glossário

## *Arduino*

O *Arduino* é uma placa de desenvolvimento de código aberto baseada em *hardware* e *software* fáceis de utilizar. As placas *Arduino* são capazes de ler entradas - luz num sensor, um dedo num botão ou uma mensagem do *Twitter* - e transformá-las numa saída - ativar um motor, ligar um *LED*, publicar algo *online* e utilizam a linguagem de programação *Arduino* (baseada em *Wiring*) e o *software Arduino* (IDE), baseado em *Processing* [1].

## *wrapper*

Um *wrapper* em Python, neste contexto, refere-se a uma camada de código desenvolvida para encapsular e simplificar a utilização da *interface* de programação do VirtualBench. Essencialmente, os *wrappers* funcionam como intermediários que permitem aos programadores interagir com a funcionalidade subjacente do VirtualBench através de comandos Python mais simples e diretos. Isso facilita a automação de tarefas e a integração do VirtualBench com outros componentes ou *scripts* desenvolvidos em *Python*, proporcionando uma maneira mais eficiente e acessível de controlar e operar os dispositivos e funcionalidades oferecidos.

## **DIAC**

O *Diode for Alternating Current* (DIAC) é um semicondutor que conduz a corrente elétrica num só sentido, tal como o díodo, mas de forma controlada, isto é, só após um disparo na sua porta. O SCR é constituído por quatro camadas PNPN ou NPNP, tendo numa extremidade um ânodo e um cátodo e, ainda, a porta. Ao aplicar uma tensão positiva entre o ânodo e cátodo, ele conduz, desde que se aplique um impulso positivo na porta, com um determinado valor mínimo. Tanto o ânodo como a porta devem ser polarizados directamente, senão, o SCR não conduz [2].

## **EEPROM**

EEPROM (*Electrically Erasable Programmable Read-Only Memory*) é um tipo de memória não volátil que pode ser escrita e apagada electricamente. É utilizada para armazenar pequenas quantidades de dados que devem ser preservados mesmo após desligar o sistema, como configurações, identificadores de

*hardware* ou parâmetros de funcionamento.

## **ESP32**

O *ESP32* é um microcontrolador de baixo custo e baixa potência com capacidade *Wi-Fi* e *Bluetooth*, desenvolvido pela *Espressif Systems*. É amplamente utilizado em projetos de IoT devido às suas capacidades de conectividade e processamento.

## **HAT**

O termo HAT (*Hardware Attached on Top*) refere-se a placas de expansão desenvolvidas segundo um padrão específico para o *Raspberry Pi*. Estas placas ligam-se diretamente ao *header GPIO* e incluem uma *EEPROM* que permite ao sistema detectar automaticamente o acessório e configurar os respectivos pinos e *interfaces* de forma transparente.

## **Industry 4.0**

A Indústria 4.0, uma iniciativa da Alemanha, converteu-se um termo adoptado globalmente na última década. Dez anos após a introdução da Indústria 4.0, a Comissão Europeia anunciou a Indústria 5.0. Considera-se que a Indústria 4.0 é orientada para a tecnologia, enquanto a Indústria 5.0 é orientada para o valor [3].

## **KiCad**

O *KiCad* é um conjunto de ferramentas de *software* de código aberto dedicada à Automatização do *design* electrónico (EDA - *Electronic Design Automation*). Esta ferramenta integra funcionalidades para a captura de esquemas elétricos e o desenho de placas de circuito impresso (*Printed Circuit Board (PCB)*), permitindo a exportação dos projetos nos formatos *Gerber* e *IPC-2581*. O *KiCad* é compatível com os sistemas operativos *Windows*, *Linux* e *macOS*, sendo distribuída sob a licença GNU General Public License, versão 3 (GPL v3)

## **PC/104**

As placas PC/104 são um padrão de computação embebida que se destaca pelo seu formato compacto e modula. Desenvolvidas pelo PC/104 Consortium, estas placas são amplamente utilizadas em aplicações industriais e de controle, oferecendo uma combinação de robustez e flexibilidade.

## **Python**

O *Python* é uma linguagem de programação de alto nível, orientada a objecto, ou seja, com sintaxe mais simplificada e próxima à linguagem humana

É utilizada em *web* ou servidores. Tem uma biblioteca padrão bastante completa e uma enorme comunidade que desenvolve ferramentas e bibliotecas (*frameworks*) adicionais. O código é aberto e gratuito [4]. O *Python* usa a indentação para delimitar blocos de código. Como curiosidade esta linguagem foi baptizada pelo seu criador, Guido van Rossum, como referência aos *Monty Python*.

### Raspberry Pi

Raspberry Pi é uma gama de pequenos computadores acessíveis e versáteis que podem ser utilizados para vários projectos e fins, desenvolvidos pela *Raspberry Pi Foundation*, uma organização educacional no Reino Unido. Esses dispositivos foram projetados para promover o ensino de ciências da computação básicas em escolas e países em desenvolvimento, além de serem populares entre entusiastas de eletrónica e *makers* devido à sua versatilidade e custo acessível. Desde o seu lançamento, os modelos de RaspberryPI evoluíram significativamente, oferecendo maiores capacidades de processamento, memória e conectividade, adequados a uma ampla gama de aplicações, desde simples projetos de bricolagem a sistemas embbebidos complexos e até mesmo servidores de baixo custo[5][6].

### SBC

SBC ou *Single-Board Computer* é um computador completo integrado numa única placa de circuito impresso, que inclui processador, memória, *interfaces* de entrada/saída e, muitas vezes, conectividade de rede. Os SBCs, como o *Raspberry Pi*, são amplamente utilizados em aplicações de ensino, automação, prototipagem e sistemas embarcados, devido ao seu baixo custo, tamanho reduzido e versatilidade.

### SCR

O *Silicon Control Rectifier* (SCR) é semicondutor que conduz a corrente elétrica nos dois sentidos, como se fossem dois diodos, em paralelo e em sentidos contrários; isto é, ora funciona um diodo, durante a alternância positiva, ora funciona o outro na alternância negativa. A alimentação da porta do TRIAC é, frequentemente, feita utilizando um DIAC [2].

### TRIAC

O *Triode for Alternating Current* (TRIAC) é um tiristor constituído por quatro camadas PNPN ou NPNP, com dois ânodos e uma porta. Contrariamente ao SCR, o TRIAC conduz nos dois sentidos da CA, desde que se aplique à porta um impulso suficiente, positivo ou negativo. O TRIAC é, geralmente, utilizado em CA, no comando e controlo de potência [2].

## **W3C**

O W3C é uma organização internacional que desenvolve padrões e diretrizes para a web visando garantir que esta permaneça aberta, acessível e interoperável para todos. Os principais objetivos são promover a compatibilidade entre diferentes sistemas e garantir que a web seja acessível a todos os utilizadores, independentemente das suas capacidades ou dispositivos. O W3C trabalha em várias áreas, incluindo HTML e CSS [7].

# Lista de Acrónimos

<b><math>\mu</math>C</b>	<i>microcontroladores</i>
<b><math>I^2C</math></b>	<i>Inter-Integrated Circuit</i>
<b>ADC</b>	<i>Analog-to-Digital Converter</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>ARM</b>	<i>Advanced RISC Machine</i>
<b>BTH</b>	<i>Blekinge Institute of Technology</i>
<b>CA</b>	Corrente Alternada
<b>CC</b>	Corrente Contínua
<b>COVID-19</b>	SARS-CoV-2
<b>CSS</b>	<i>Cascading Style Sheets</i>
<b>DAQ</b>	<i>Data Acquisition</i>
<b>DESI</b>	Digital Economy and Society Index
<b>DIAC</b>	<i>Diode for Alternating Current</i>
<b>DIESEL</b>	<i>Distance Internet-Based Embedded System Experimental Laboratory</i>
<b>DMM</b>	<i>Digital Multimeter</i>
<b>DPST</b>	<i>Double Pole Single Throw</i>
<b>EJS</b>	<i>Easy Java Simulations</i>
<b>EUA</b>	Estados Unidos da América
<b>FPGA</b>	<i>Field Programmable Gate Array</i>
<b>GPIO</b>	<i>General Purpose Input/Output</i>
<b>GPL</b>	<i>Gnu Public License</i>
<b>HTML</b>	<i>HyperText Markup Language</i>

<b>HTTP</b>	<i>Hypertext Transfer Protocol</i>
<b>IDC</b>	<i>Insulation Displacement Connector</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>IoT</b>	<i>Internet Of Things</i>
<b>ISA</b>	<i>iLab Shared Architecture</i>
<b>ISEP</b>	Instituto Superior de Engenharia do Porto
<b>JSON</b>	<i>JavaScript Object Notation</i>
<b>LabVIEW</b>	<i>Laboratory Virtual Instrument Engineering Workbench</i>
<b>LaRE</b>	Laboratório Remoto Expansível
<b>LMS</b>	<i>Learning Management System</i>
<b>LR</b>	Laboratório Remoto
<b>LV</b>	Laboratório Virtual
<b>MATLAB</b>	<i>MATrix LABoratory</i>
<b>MB</b>	Megabyte
<b>MCR</b>	Matriz de Comutação de Relés
<b>MHSA</b>	<i>Multipurpose Hardware and Software Architecture</i>
<b>MIT</b>	<i>Massachusetts Institute of Technology</i>
<b>NI</b>	<i>National Instruments</i>
<b>OCDE</b>	Organização para a Cooperação e Desenvolvimento Económico
<b>OE</b>	<i>Output Enable</i>
<b>PC</b>	<i>Personal Computer</i>
<b>PCB</b>	<i>Printed Circuit Board</i>
<b>PIC</b>	<i>Programmable Interrupt Controller</i>
<b>PILAR</b>	<i>Platform Integration of Laboratories based on the Architecture of vi-siR</i>
<b>PIP</b>	<i>Pip Installs Packages</i>
<b>PISA</b>	<i>Program for International Student Assessment</i>

<b>PLC</b>	<i>Programmable Logic Controller</i>
<b>PRR</b>	Plano de Recuperação e Resiliência
<b>PUC</b>	Pontifícia Universidade Católica
<b>PXI</b>	<i>PCI eXtensions for Instrumentation</i>
<b>RCLK</b>	<i>Register Clock</i>
<b>RLMS</b>	<i>Remote Laboratory Management System</i>
<b>SCADA</b>	<i>Supervisory Control and Data Acquisition</i>
<b>SCR</b>	<i>Silicon Control Rectifier</i>
<b>SER</b>	<i>Serial Input</i>
<b>SIG</b>	<i>Special Interest Groups</i>
<b>SIP0</b>	<i>Serial Input Parallel Output</i>
<b>SPICE</b>	<i>Simulation Program for Integrated Circuits Emphasis</i>
<b>SPST</b>	<i>Single Pole Single Throw</i>
<b>SRCLK</b>	<i>Shift Register Clock</i>
<b>SRCLR</b>	<i>Shift Register Clear</i>
<b>STEM</b>	<i>Science, Technology, Engineering and Mathematics</i>
<b>TIC</b>	Tecnologias da Informação e da Comunicação
<b>TRIAC</b>	<i>Triode for Alternating Current</i>
<b>ue</b>	União Europeia
<b>UNESCO</b>	<i>United Nations Educational, Scientific and Cultural Organization</i>
<b>URL</b>	<i>Uniform Resource Locator</i>
<b>USB</b>	<i>Universal Serial Bus</i>
<b>VB</b>	<i>VirtualBench</i>
<b>VISIR</b>	<i>Virtual Instrument Systems in Reality</i>
<b>WSGI</b>	<i>Web Server Gateway Interface</i>
<b>WWW</b>	<i>World Wide Web</i>



## Capítulo 1

# Introdução

“So it begins”

— King Theoden, *The Lord of the Rings: The Two Towers*

Esta dissertação insere-se no programa de Mestrado em Engenharia Eletrotécnica e de Computadores, Área de Especialização em Automação e Sistemas e tem como principal objectivo o desenvolvimento de um Laboratório Remoto (LR) aplicado ao ensino da eletrónica. Este capítulo aborda a motivação para a abordagem desta temática, nele é apresentada a contextualização, assim como os objectivos a serem atingidos e uma breve estruturação da dissertação.

### 1.1 Enquadramento

A educação em engenharia, especialmente em áreas como eletrónica, depende fortemente de actividades práticas. Os laboratórios são cruciais para que os estudantes possam aplicar os conceitos teóricos em cenários práticos, desenvolver habilidades técnicas e enfrentar desafios reais[8][9]. No entanto, a manutenção e o acesso a laboratórios físicos tradicionais apresentam várias dificuldades, incluindo altos custos, limitações de espaço e a necessidade de supervisão constante por parte de instrutores ou professores qualificados[10]. Além disso, o acesso a esses laboratórios pode ser restrito devido a factores como horários de funcionamento e a necessidade de presença física.

Nos últimos anos, os avanços na tecnologia, do *eLearning* e na implementação da abordagem *Science, Technology, Engineering and Mathematics* (STEM) abriram novas possibilidades para o ensino prático da eletrónica através de laboratórios remotos. Esses laboratórios permitem que os estudantes realizem experiências reais em equipamentos físicos controlados remotamente, independentemente da sua localização geográfica. A implementação de laboratórios remotos tem o potencial de democratizar o acesso ao ensino de qualidade, oferecendo oportunidades educacionais mais equitativas e inclusivas[11].

## 1.2 Objectivos

A génesis desta dissertação ocorreu há três anos e estava associada à ideia de desenvolver um Laboratório Remoto Expansível (LaRE) para o ensino da electrónica. Uma das soluções existentes no Instituto Superior de Engenharia do Porto (ISEP) é o *Virtual Instrument Systems in Reality* (VISIR)<sup>1</sup>. O VISIR é um projecto (LR) que pode ser aplicado no vasto campo da engenharia eletrotécnica e eletrónica e na área da teoria e prática de circuitos. Tem como objetivo definir, desenvolver e avaliar um conjunto de módulos educativos compostos por experiências práticas, virtuais e remotas [12]. No entanto, o desenvolvimento deste LR apresenta um grave problema, que corresponde ao facto de, por questões legais, não ser possível realizar qualquer tipo de modificação e/ou actualização ao *firmware* da matriz que controla os relés.

### **A REVER - Prof.**

Partindo, então, destes pressupostos definiram-se os seguintes objectivos:

- Contextualizar o uso dos laboratórios remotos no ensino;
- Estudar e analisar as alternativas existentes, nomeadamente, o VISIR;
- Identificar os requisitos de *hardware* e *software* necessários para implementar o LaRE como um projeto *open-source*, em conformidade com a licença *Gnu Public License* (GPL):
  - Pesquisar, analisar e avaliar a implementação de um servidor;
  - Pesquisar, analisar e avaliar as soluções de *software* existentes para a integração do *VirtualBench (VB)*<sup>2</sup> com o LaRE;
  - Definir quais, e quantos, circuitos integrar;
- Implementar e testar o LaRE;
- Identificar as vantagens e desvantagens do LaRE, assim como possíveis melhoramentos;

---

<sup>1</sup>Uma explicação mais pormenorizada VISIR será fornecida nos capítulos seguintes

<sup>2</sup>Uma explicação mais detalhada será dada nos capítulos seguintes

- Apresentar resultados e conclusões da implementação do LaRE.

### 1.3 Estrutura do documento

À data de 30/06 - Poderá haver modificações nos capítulos - **rever** Esta dissertação está organizada em quatro capítulos. Neste primeiro capítulo, foi feita uma breve introdução à implementação do LaRE e aos objetivos que a norteiam. O Capítulo 2 aborda o estado da arte no que diz respeito à Educação Digital e à integração dos laboratórios no ensino. O Capítulo 3 aborda detalhadamente os aspectos relacionados com o LaRE, a nível de *hardware* e *software*. Os resultados obtidos neste capítulo servem como base para, no Capítulo 4, se discutir as vantagens e desvantagens do LaRE e, tendo em conta as suas limitações, apontar melhoramentos que podem ser desenvolvidos no futuro. No Capítulo 5 serão apresentadas as conclusões da implementação do LaRE.



## Capítulo 2

# Estado da Arte

“‘You are entering a world of pain.’”

— Walter Sobchak, *The Big Lebowski*

Tendo em conta a nossa experiência como professor numa escola secundária, neste capítulo procuramos fazer o enquadramento das principais dificuldades que a Educação (digital) atravessa em Portugal e discutir em que medida a integração da tecnologia digital na educação é essencial para criar ambientes de aprendizagem mais eficazes e adaptados à evolução do panorama digital. Explora-se o potencial de tecnologias, como os laboratórios remotos, para o desenvolvimento de experiências de aprendizagem personalizadas e reflete-se sobre a forma como podem ajudar a suprir estas mesmas dificuldades. Face a isto, discute-se a ascensão do *eLearning*, particularmente no contexto da abordagem STEM, assim como a evolução da educação das salas de aula tradicionais para ambientes virtuais, enfatizando as oportunidades e possibilidades de aprendizagem alargadas oferecidas pelo *eLearning*.

### 2.1 Transformação digital na educação

**NOTA:** O prof sugeriu que os pontos 2.1 e 2.2 pudessem ficar num só ponto

no entanto, parece-me fazer sentido que fiquem separados, se calhar depois do prof ler possa ter outra percepção. Deixo para consideração depois de ler.

A educação está a sofrer grandes mudanças a nível tecnológico, social e humano. Estas foram impulsionadas pela pandemia SARS-CoV-2 (COVID-19), que expôs

as fragilidades em termos de gestão e capacitação dos recursos tecnológicos e do eLearning<sup>1</sup>. O sistema educativo português está longe da perfeição (ou, se nos atrevermos a dizê-lo, está longe de ser razoável) mas, durante a pandemia, devido a um grande esforço coletivo, respondeu imediatamente a este enorme obstáculo. Num curto espaço de tempo, o ensino presencial e o contacto humano foram substituídos pelo ensino à distância.

### **Estávamos pronto para isto?**

Antes da pandemia, no Ensino Básico, as tecnologias digitais eram mais frequentemente utilizadas como meio de comunicação do que como ferramenta pedagógica [14]. Nas últimas duas décadas aproximadamente, Portugal já introduziu programas para abordar a digitalização na educação. O programa de digitalização contemplado no Plano de Ação para a Transição Digital (Resolução do Conselho de Ministros n.º 30/2020, de 21 de abril) prevê, entre outras medidas, uma forte aposta na formação digital dos professores, no desenvolvimento digital das escolas e na disponibilização de recursos educativos digitais [15] [16].

No mundo de hoje, cada vez mais digital, 3.6 mil milhões de pessoas ainda não têm acesso à *Internet*[17]. Este problema reduz o acesso a um mundo de informações disponíveis *online* e limita o potencial de aprendizagem e crescimento, para não falar das competências digitais de que as pessoas necessitam para aprender e melhorar as suas vidas. A crise da COVID-19 mostrou-nos como a ligação à *Internet* é crucial para as actividades quotidianas, como o trabalho e a aprendizagem. Hoje, mais do que nunca, é necessário reforçar as infra-estruturas nacionais para garantir que a conetividade esteja mais amplamente disponível. Igualmente importante é reforçar os planos de conetividade das escolas e investir numa aprendizagem de qualidade, a fim de melhorar o acesso à educação, os resultados da aprendizagem e o potencial de ganhos dos jovens, bem como o desenvolvimento socioeconómico das suas comunidades e países [17]. Se adoptarmos uma abordagem geral da integração da tecnologia digital [18], Portugal ocupa o 19º lugar entre os 28 Estados-Membros da União Europeia (ue) no Digital Economy and Society Index (DESI) 2020. Apesar de alguns progressos na dimensão do capital humano, graças a uma melhoria do nível básico de competências digitais e a uma maior percentagem de diplomados em Tecnologias da Informação e da Comunicação (TIC), este aspeto continua a ser particularmente crítico para Portugal, dado o atual baixo nível de literacia digital da sua população.

Relativamente à dimensão capital humano, Portugal ocupa o 21º lugar entre 28 países da ue - uma melhoria de dois lugares em relação ao ano anterior, mas ainda abaixo da média da ue. Em 2019, a percentagem da população portuguesa sem, pelo menos, competências digitais básicas diminuiu de 50% para 48%. **REVER ESTA AFIRMAÇÃO** No entanto, cerca de 26% dos cidadãos portugueses não

<sup>1</sup>Jay Cross, é conhecido por ter “cunhado” pela primeira vez o termo *eLearning*[13]. Por isso, nesta tese, o termo será sempre referido desta forma, em vez de *e-Learning*

tinham quaisquer competências digitais. Isto deve-se principalmente ao facto de muitas pessoas nunca terem utilizado a Internet [18].

Podemos deduzir que esta realidade está relacionada com os baixos níveis de educação que ainda existem no país. No entanto, há outros factores que são visíveis para todos, como o elevado preço dos serviços de *Internet* ou o baixo número de licenciados em TIC: “A proporção de especialistas em TIC representa uma percentagem menor da força de trabalho em comparação com a média da ue (2,4% contra 3,9% na ue). Além disso, Portugal continua a ter uma das mais baixas percentagens de especialistas de TIC no emprego feminino total, representando metade da média da ue. A proporção de licenciados em TIC no total de licenciados melhorou, mas é ainda muito baixa para os padrões da ue (1,9% em comparação com 3,6% na ue)” [18]. Esta falta de competências básicas foi agravada pela pandemia. De acordo com o relatório “*Juventude e COVID 19: impactos no emprego, na educação, nos direitos e no bem-estar mental*” [19], 65% consideram que aprenderam menos desde o início da pandemia devido à transição das aulas presenciais para as aulas *online* durante o confinamento. Apesar dos seus esforços para continuar a estudar, metade destes jovens acredita que os seus estudos vão ficar para trás e 9% pensa que podem reprovar devido a estas dificuldades. A situação foi pior para os jovens que vivem em países de baixos rendimentos, por haver menos acesso à *Internet*, uma falta crítica de equipamentos, e às vezes nenhum lugar em casa para a aprendizagem adequada [19].

Com base nestes resultados, o estudo conclui que “os desafios colocados pela transição para o ensino fora da sala de aula e em casa” foram enormes. Mesmo quando as instituições conseguiram fazer a transição para a aprendizagem *online*, os professores, formadores e estudantes podem não ter tido a capacidade de “garantir a continuidade da aprendizagem”. Entre os factores que dificultam a eficácia do ensino *online* contam, como já foi referido: baixos níveis de acesso à *Internet*; lacunas nas competências digitais; incapacidade de ensinar e aprender à distância; falta de equipamento informático em casa; falta de espaço; falta de materiais preparados para o *eLearning*; e falta de trabalho de grupo e de contacto social - ambos componentes fundamentais do processo de aprendizagem.

Como professor do ensino secundário e director de turma, que tinha de lidar tanto com professores como com alunos, pudemos observar estes problemas do lado dos alunos, principalmente: não terem computador portátil ou pessoal, não terem acesso à *internet* ou esta ser lenta. Houve mesmo alguns casos de alunos que tiveram de assistir às aulas através do telemóvel. Do lado dos professores, esperava-se um enorme desafio, principalmente devido à falta de experiência na lecionação *online* e às dificuldades em encontrar e utilizar as várias ferramentas *online*. A “curva de aprendizagem” foi muito lenta.

Obviamente, não só em Portugal, mas também em todo o mundo, as escolas e

instituições encerraram em meados de março de 2020 e os alunos de todos os níveis de ensino assistiram às aulas e interagiram com os professores através de videoconferências e outras ferramentas *online*. A *United Nations Educational, Scientific and Cultural Organization* (UNESCO) apoiou a implementação de programas de ensino à distância em grande escala e recomendou aplicações e plataformas educativas gratuitas que as escolas e os professores poderiam utilizar para comunicar com os alunos à distância. A organização partilhou depois as melhores práticas para tirar partido de tecnologias móveis pouco dispendiosas para fins de ensino e aprendizagem, a fim de atenuar as perturbações no ensino [20].

O Plano de Recuperação e Resiliência (PRR) recomenda a entrega de computadores portáteis a todos os alunos e professores do ensino primário e secundário. Supostamente deveriam ter sido entregues 600 mil computadores (com acesso à Internet), mas ainda há um longo caminho a percorrer. Em agosto de 2022, apenas 70% desses computadores tinham chegado aos utilizadores. A principal razão? Os pais foram confrontados com o compromisso de reembolsar o dinheiro investido pelo Estado ,se não pudessem devolvê-los nas mesmas condições em que os tinham recebido[21].

Entretanto, **REFERIR AQUI CASO PESSOAL EM CONTEXTO**

**DE SALA DE AULA???** - **OPINIÃO PROF** as aulas de laboratório (electrónica) eram as que menos sentido faziam: por exemplo, o contacto com uma placa branca - também conhecida por *breadboard* ou placa de ensaio - e todos os componentes físicos foi substituído pelo recurso a dois simuladores online: **Multisim** e **Falstad**. Felizmente, estes já eram utilizados na sala de aula, embora como complemento da vertente mais prática do ensino.

Assim, a resposta à questão colocada anteriormente é: **não**.

Ninguém estava preparado para estas mudanças nem para o ensino *online* e o *eLearning*.

Como já foi referido, os desafios para alunos e professores foram enormes, a adaptação ao ensino/aprendizagem *online* teve de ser feita muito rapidamente e os recursos digitais tornaram-se a “tábua de salvação” da educação. No entanto, as oportunidades que as tecnologias digitais oferecem vão muito além do apoio de emergência utilizado durante a pandemia. Atualmente, está disponível um número infinito de possibilidades, novas respostas para o quê, como e quando as pessoas aprendem [22].

A forma de pensar dos estudantes de hoje é também muito diferente da de há trinta anos. De uma forma mais particular, o pensamento mudou muito desde que a *Internet* e os recursos digitais começaram a tornar-se massivos. É essencial que

os alunos possam aceder à tecnologia e utilizá-la para se desenvolverem ainda mais e a lição mais importante de todas é: **permitir-lhes compreender o que fazer com a informação e os recursos que estão à sua disposição de tantas formas diferentes.** Esta nova adaptação educativa (ou deveríamos chamar-lhe “evolução”?) “também adapta a aprendizagem aos estilos de aprendizagem pessoais com uma granularidade e precisão muito maiores do que qualquer ambiente de sala de aula tradicional pode fazer. Do mesmo modo, **laboratórios virtuais** dão aos alunos a oportunidade de conceber, realizar e aprender com as experiências, em vez de se limitarem a aprender sobre elas” [22].

Em agosto de 2020, numa citação incluída num artigo publicado no *Diário de Notícias*, um aluno afirmava:

**“As aulas de laboratório foram as que menos sentido fizeram para mim, pois fazer relatórios e cálculos sobre experiências que não fizemos, sem adquirir/desenvolver as competências e técnicas que é suposto esta componente da disciplina nos dar, é um pouco ridículo.”** [23].

Por outro lado, os professores devem desenvolver a sua literacia digital, para melhor dominarem estas novas ferramentas na sala de aula, de forma a ajudarem os alunos a construírem o seu próprio conhecimento.

Assim, é evidente que a Educação, como um todo, precisa de se inserir neste contexto que está cada vez mais presente no quotidiano de todos.

O mundo é cada vez mais dominado pela tecnologia (... e nestes tempos actuais começa o domínio da Inteligência Artificial), milhares de milhões de dispositivos físicos em todo o mundo estão agora ligados à *Internet*, todos recolhendo e partilhando dados. Atualmente, os *chips* de computador são muito baratos e a ubiquidade das redes sem fios permite ligar qualquer coisa, desde algo tão pequeno como um comprimido até algo tão grande como um avião. É o que se designa por *Internet Of Things* (IoT) [24]. Por conseguinte, a informação e os recursos estão “na ponta dos dedos de todos” e à distância de um clique.

*Chegou o momento de os países aproveitarem as lições da pandemia para reconfigurarem as pessoas, os espaços, o tempo e a tecnologia, de modo a criarem ambientes educativos mais eficazes e eficientes* [25].

## 2.2 Educação Digital em Portugal: Avanços e Desafios

Nos últimos anos, é evidente que estamos a enfrentar grandes mudanças tecnológicas. Os conhecimentos e as competências têm de acompanhar esta evolução. (Este facto ficou bem patente com a pandemia...).

Praticamente todas as aprendizagens e futuros empregos exigirão um certo nível de competências e aptidões digitais. A constante evolução tecnológica exige o

desenvolvimento de competências ao longo da vida[26]. No entanto, em média, dois em cada cinco europeus com idades compreendidas entre os 16 e os 74 anos ainda não possuem estas competências[18].

Se reflectirmos sobre esta ideia em contexto educativo e acrescentarmos o facto de o ensino da eletrónica em Portugal passar pelo ensino profissional **ACHO QUE NÃO REFERI!! - A REVER**, como já foi referido anteriormente, e se o objetivo é a evolução e inovação no processo de ensino/aprendizagem, é necessário proporcionar a todos os intervenientes as condições e ferramentas necessárias para que o trabalho possa ser desenvolvido e, consequentemente, os alunos realmente aprendam.

A evolução da educação digital em Portugal pode ser aferida pelas crescentes iniciativas governamentais e adopção de tecnologias digitais nas escolas e universidades. Nos últimos anos, Portugal tem investido significativamente na educação digital, reconhecendo a importância das competências digitais para o futuro dos seus cidadãos e a competitividade do país. Este esforço é materializado através de várias iniciativas e programas governamentais que visam integrar a tecnologia no sistema educacional, capacitar professores e preparar os estudantes para a era digital. As medidas de apoio aos objetivos digitais representam um montante que corresponde a 22% da dotação total do plano, ultrapassando o limiar de 20% definido pela regulamentação europeia [27]. ~~12 das 20 componentes do PRR têm contributo direto para a meta digital.~~

Lançado em 2017, o Plano Nacional de Competências Digitais (INCoDe.2030)[28] é uma iniciativa ambiciosa do governo português para aumentar a literacia digital de toda a população. Este plano está estruturado em cinco eixos principais: inclusão, educação, qualificação, especialização e investigação. Como complemento ao INCoDe.2030, o Programa Escola Digital é outra iniciativa crucial do Ministério da Educação. Este programa visa a modernização das escolas através da integração de tecnologias digitais, com objetivos claros de dotar as escolas de infra-estruturas adequadas, disponibilizar dispositivos e recursos digitais e capacitar os professores. [29] Os objectivos deste programa centram-se no melhoramento das infra-estruturas tecnológicas, na distribuição de computadores portáteis e tablets aos alunos e professores, no acesso a recursos educativos digitais de qualidade e uma aposta na capacitação e formação de professores.

No entanto, no meio destas oportunidades criadas, ainda há alguns desafios que precisam de ser ultrapassados. A desigualdade no acesso à tecnologia ainda é uma barreira significativa, especialmente em áreas rurais e entre famílias de baixos rendimentos. A formação contínua dos professores também necessita de maior suporte, assim como a atualização constante dos recursos digitais disponíveis.

**VERIFICAR O ENQUADRAMENTO DO PARÁGRAFO SEGUINTE**  
**Acho que não referi - A REVER UPDATE:** É referido mais à frente

Como foi referido no Capítulo 1 (**anterior - secção 1.1**) um dos pontos-chave do ensino profissional é também preparar os alunos para o mundo do trabalho. A ascensão da Industry 4.0 levou a que mais instituições de ensino adotassem os laboratórios remotos como uma alternativa contemporânea para desenvolver as competências técnicas e sociais necessárias aos estudantes e formadores de engenharia[30]. De facto, a transformação digital já alterou os postos de trabalho e, a nível mundial, a indústria, os produtos e as modalidades de ensino.

**complementar mais este parágrafo e ideia - OPINIÃO PROF???**

## 2.3 eLearning e abordagem STEM

Tem-se vindo a falar de ensino à distância e de ferramentas digitais e virtuais. Por isso, importa agora contextualizar o que foi escrito anteriormente em termos de *eLearning*, abordagem STEM, como estes dois conceitos se interligam e, mais especificamente, os conceitos de laboratórios remotos e virtuais e simuladores. É óbvio que o *eLearning* e o STEM não poderão ser dissociados, sendo que a relação entre os dois é mutuamente benéfica. **REFERÊNCIA** Além dos mais, as dificuldades na implementação destes dois conceitos em Portugal apresentam várias semelhanças. Ambas enfrentam desafios relacionados com as infra-estruturas tecnológicas, formação de professores, envolvimento dos alunos e métodos de avaliação. **ENUMERAR E REFERÊNCIAS??**

O COVID-19 impulsionou a utilização de recursos digitais e, consequentemente, o *eLearning*. Este tornou-se popular e até necessário e a educação evoluiu do tradicional contexto de sala de aula para um ambiente virtual. A utilização do *eLearning* não só cria uma forma mais abrangente de aprendizagem, como também envolve muitas possibilidades: cursos *online*, laboratórios virtuais e remotos, simuladores *online*, etc. No entanto, esta globalização do *eLearning* só foi possível graças à massificação da *Internet* e à evolução da tecnologia digital.

Em 2004, Jay Cross publicou um trabalho muito interessante e algo controverso intitulado “An informal history of eLearning” [13]. O conceito em si é controverso na medida em que existem múltiplas definições do que poderá ser o *eLearning*. Por exemplo, citado no mesmo artigo: “(...) Elliott Masie disse no final de 1997, “A aprendizagem *online* é a utilização da tecnologia de rede para conceber, fornecer, selecionar, administrar e alargar a aprendizagem.”” Jay Cross, por outro lado, define o *eLearning* como uma visão daquilo em que a formação empresarial se pode tornar.

No entanto, num estudo intitulado “O e-Learning no Ensino Superior: um caso de estudo” [31], os autores generalizam a definição dada por Maria João Gomes no seu artigo sobre reflexões em torno do *eLearning* [32], considerando que este “(…)

corresponderá a qualquer metodologia de ensino/aprendizagem que integre actividades, suportadas por TIC, essenciais para atingir os objectivos de aprendizagem definidos.”

No mesmo artigo, Maria João Gomes defende que “(...) o termo e-learning deve ser adotado como menos centrado nos aspectos tecnológicos e mais próximo das potencialidades pedagógicas decorrentes da utilização das “tecnologias de rede” na concepção de situações baseadas na interação e colaboração, no sentido da construção de aprendizagens significativas”.

Assim, é fácil perceber que não há grande consenso sobre o significado de *eLearning*,

Estas múltiplas definições do conceito “*eLearning*” têm a ver com as diferentes perspectivas, pontos de vista e concepções dos vários intervenientes que, de uma forma ou de outra, têm investido neste domínio.

Se se pretender uma definição que, de alguma forma, se aproxime de um cenário de ensino à distância baseado na comunicação e na colaboração[32], então a definição apresentada no “*Convite à apresentação de propostas DG EAC/46/02*” deve ser tida em conta[33]: “(...) a utilização das novas tecnologias multimédia e da *Internet* para melhorar a qualidade da aprendizagem, facilitando o acesso a recursos e serviços, bem como o intercâmbio/interação e a colaboração à distância.”

Uma das principais vantagens do *eLearning* é a flexibilidade que oferece. Os estudantes podem aprender ao seu próprio ritmo e horário, o que é particularmente útil para aqueles que têm outras responsabilidades, como o trabalho ou a assistência à família. Além disso, o *eLearning* também pode ocorrer a partir de qualquer lugar, desde que haja uma ligação à *Internet*, o que permite aos alunos estudar a partir de casa ou de qualquer outro lugar que lhes seja conveniente.

Outra vantagem do *eLearning* é que permite o acesso a uma grande variedade de recursos e materiais de aprendizagem, incluindo vídeos, textos, actividades interactivas, fóruns de discussão, laboratórios virtuais e remotos, entre outros. Estes recursos podem ser actualizados e adaptados rapidamente de acordo com as necessidades dos alunos e professores, garantindo que os conteúdos estão sempre actualizados e são relevantes.

O *eLearning* também pode ser uma forma eficaz de personalizar a aprendizagem, permitindo que os alunos se concentrem nas áreas em que precisam de mais ajuda e avancem rapidamente nas áreas em que já têm conhecimentos suficientes. Os alunos podem fazer avaliações *online* que ajudam a determinar as suas competências e conhecimentos, permitindo que o professor adapte os conteúdos e as actividades de aprendizagem às suas necessidades.

Para as instituições de ensino em Portugal, o *eLearning* pode ajudar a reduzir os custos, eliminando a necessidade de espaço físico e equipamento adicionais, assim

como, também pode ajudar a alcançar um público mais vasto de alunos, especialmente aqueles que não têm acesso ao ensino presencial devido a restrições geográficas, financeiras ou outras. O *eLearning* também permite que as instituições de ensino em Portugal ofereçam educação de qualidade a um custo mais baixo, permitindo que mais pessoas tenham acesso à educação.

Por último, o *eLearning* pode contribuir para melhorar a qualidade do ensino em Portugal, permitindo que os alunos aprendam de forma mais eficaz e eficiente. Os professores podem monitorizar os progressos dos alunos em tempo real, fornecendo *feedback* imediato e adaptando os conteúdos às suas necessidades. Além disso, o *eLearning* pode ajudar a promover a colaboração entre alunos e professores, permitindo-lhes trabalhar em conjunto em projectos e actividades de grupo, independentemente da sua localização física.

Embora existam desafios envolvidos na adoção do *eLearning*, os benefícios superam os problemas e é provável que o *eLearning* continue a ser uma parte importante do ensino no futuro.

Como já foi referido, STEM é acrónimo de Ciência, Tecnologia, Engenharia e Matemática e o conceito vai muito além de uma mera definição.

Se tomarmos como ponto de partida o endereço da UNESCO [34], é fácil compreender que as definições dão um sentido lato ao conceito. Citando Heather B. Gonzalez, Jeffrey J. Kuenzi, 2012 e o Australian Council of Learned Academies (ACOLA), 2014, verifica-se que consideram essencialmente a “ocupação”, embora em contextos diferentes. Enquanto a primeira publicação apresenta um diagnóstico de STEM nos Estados Unidos da América (EUA), a segunda publicação aborda uma forma de reduzir as lacunas nas competências “STEM”.

Jonathan Rothwel, no seu artigo [35], tenta perceber a ambiguidade de STEM definindo primeiro conhecimento STEM”, e assim “constata que uma grande parte dos empregos “STEM” são técnicos (“*blue-collar*”) e não “académicos””

Uma terceira definição - porque não é nosso objectivo principal dissecar as mais variadas definições de STEM - é a que eventualmente melhor se adequará a este trabalho: Mark Sanders no seu artigo “STEM, STEM Education, STEMmania” [36], centra-se no termo “STEM education”, mais precisamente, “faz a apologia do ensino “integrativo” STEM (significando aqui “ensino entre quaisquer duas ou mais áreas disciplinares STEM, ou entre uma disciplina STEM e outra) *versus* “educação STEM””. Tipicamente, a educação inclui actividades educativas em todos os níveis de ensino - desde a Educação Pré-escolar até ao pós-doutoramento - em contextos formais (por exemplo, salas de aula) e informais (por exemplo, programas pós-escolares). Nos últimos anos, surgiu uma nova definição baseada na ideia de acrescentar as artes ao currículo, recorrendo a princípios de raciocínio e conceção e incentivando soluções criativas. No entanto, não é objetivo do presente documento desenvolver ainda mais este conceito específico.

**NOTA: Colocar ou não este tipo de exemplo. OPINIÃO PROF**

No meu caso particular, os alunos estão envolvidos em vários projectos, tanto a nível nacional como internacional, aplicando o conceito STEM. O trabalho é desenvolvido em contexto de sala de aula e abarca vários tipos de trabalhos, desde enviar engenhos para a estratosfera até à construção de robôs.

Um dos principais desafios enfrentados em Portugal na implementação do STEM é a falta de recursos e infra-estruturas adequados. Como já foi referido no Capítulo 2 e é reforçado pelo *Program for International Student Assessment* (PISA) da Organização para a Cooperação e Desenvolvimento Económico (OCDE), muitas escolas não têm acesso a laboratórios bem equipados e à tecnologia mais recente para ensinar ciências e tecnologia de forma eficaz [37]. Extrapolando um pouco mais, a falta de investimento na educação em Portugal é evidente e limita a capacidade do país para inovar e criar novas tecnologias [38], [39], [40], [41], [42].

Os pontos fracos do sistema português - identificados no relatório da OCDE intitulado "Políticas Eficazes, Escolas de Sucesso" - são a falta de pessoal não docente, a taxa de retenções, a falta de equipamento informático, plataformas de ensino online e acesso rápido e eficaz à internet e a falta de equidade. - **A REVER**

Outro problema que afecta a implementação do STEM em Portugal é a escassez de profissionais qualificados nesta área [43] - "A procura de licenciados com este tipo de qualificação é baixa em relação à oferta (ou procura) (...)".

Por outro lado, muitos estudantes que se formam em ciência e tecnologia em Portugal procuram oportunidades de emprego no estrangeiro, o que pode levar à fuga de talentos e à perda de conhecimentos valiosos. **PROCURAR REFERÊNCIAS E CITAÇÕES - A REVER** Além disso, muitos estudantes não têm acesso a um ensino de qualidade STEM, o que pode limitar as suas futuras oportunidades de carreira - Estas dificuldades já foram mencionadas na Secção - **VERIFICAR SECÇÃO**.

Para além destes desafios estruturais, existe também uma falta de consciência da importância do STEM para o desenvolvimento económico e social do país. Muitos estudantes podem não compreender a relevância destas áreas de conhecimento para o seu quotidiano e para o futuro de Portugal. Consequentemente, podem não se sentir motivados para estudar ciência e tecnologia ou para seguir carreiras nestes domínios.

A pandemia COVID-19 também apresentou desafios adicionais para a implementação do STEM. Com muitas escolas fechadas e com aulas à distância, muitos alunos não tiveram acesso a laboratórios e equipamento especializado necessário para aprender ciência e tecnologia *hands-on*. Além disso, a pandemia aumentou a desigualdade na educação, com os alunos de famílias com baixos rendimentos a terem menos acesso à tecnologia e aos recursos necessários para acompanhar o ensino à distância [44][45].

Desde que a *World Wide Web* (WWW) se generalizou e se tornou um meio viável de aprendizagem, abriram-se novas oportunidades para um ensino mais flexível. No entanto, a educação em Portugal continua a ser condicionada por práticas instrucionais, administrativas e fortemente burocratizadas.

Este não é um problema exclusivamente português. De facto, “(...) os nossos sistemas de ensino a todos os níveis enfrentam desafios assustadores que são produtos da ineeficácia das várias componentes do sistema. Um rácio professor-aluno muito alto (particularmente nas escolas públicas), défice de infra-estruturas, instalações escolares inadequadas, indisponibilidade de recursos de aprendizagem e alunos pouco motivados são os mais óbvios dos nossos **desafios**[46].”

Esta afirmação descreve com exatidão o que está a acontecer nas escolas portuguesas em todos os níveis de ensino.

Ultrapassar estes desafios não é fácil.

#### **O que é que nós, professores, podemos fazer?**

Em primeiro lugar, procurar os recursos cuja utilização em contexto de sala de aula possa, pelo menos, permitir que o processo de ensino-aprendizagem evolua e que estes desafios sejam ultrapassados ou, pelo menos, atenuados: Os recursos (...) didácticos parecem estar no centro de tudo, os professores precisam deles para ajudar no seu ensino e os alunos precisam deles para os ajudar na sua aprendizagem[46].”

Por outro lado, num inquérito (link[pdf]) realizado em nome da Pearson<sup>2</sup> de 8 a 14 de junho de 2020 pela The Harris Poll<sup>3</sup>, envolvendo 7038 pessoas com idades compreendidas entre os 16 e os 70 anos em todo o mundo, verificou-se que 78% dos inquiridos acreditavam que a aprendizagem *online* daria às pessoas mais fácil acesso a uma educação de qualidade. **Talvez falte completar aqui mais um pouco OPINIÃO PROF**

Portanto, pelo que foi dito anteriormente e conhecendo bastante bem a realidade do ensino secundário por trabalhar com alunos que, na sua maioria não diretamente para o mundo do trabalho, pode-se concluir que a educação enfrenta vários desafios, nomeadamente a necessidade dos professores, em particular, se manterem actualizados em relação às tecnologias emergentes, bem como serem capazes de preparar os alunos para as exigências do mercado de trabalho em constante evolução. **Talvez precise de mais desenvolvimento OPINIÃO PROF**

No entanto, este não será o maior desafio. Existem vários e graves problemas a montante que tornam o processo de ensino/aprendizagem muito difícil. Este não é apenas um problema das escolas secundárias, também abrange o ensino universitário. De facto, podemos referir como um problema grave as dificuldades financeiras[47] [48] [49] das escolas e universidades, que limitam o acesso dos alunos a equipamentos

---

<sup>2</sup><https://plc.pearson.com/>

<sup>3</sup><https://theharrispoll.com/>

e materiais necessários essenciais nomeadamente para as aulas laboratoriais: computadores actualizados, software, etc. Uma das consequências desta limitação é a falta de ligação entre a teoria e a prática. Muitas vezes, os alunos são expostos a uma grande quantidade de informação teórica, mas não têm a oportunidade de aplicar esses conhecimentos em projectos práticos. Este facto pode levar a uma falta de compreensão sobre como as teorias se aplicam na vida real e pode dificultar a transferência de conhecimentos para a prática profissional.

Para ultrapassar os desafios da implementação do STEM em Portugal, é importante que o país continue a investir em recursos e infra-estruturas adequados ao ensino e aprendizagem das ciências e tecnologias. Além disso, é necessário sensibilizar para a importância do STEM para o futuro de Portugal e proporcionar oportunidades de carreira atractivas para os licenciados nestas áreas. Finalmente, é também importante que Portugal trabalhe para manter o seu talento científico e tecnológico no país, em vez de o perder no estrangeiro: “Portugal perde 4 mil cérebros por ano” [50].

Esta forma de ensino não só faz a integração das áreas do conhecimento, como também permite ao aluno usá-las para conexões na resolução de problemas diários. A aprendizagem é amplamente beneficiada com a inter-disciplinaridade. Além disso, nesta metedologia de ensino, os alunos aprendem a colaborar uns com os outros.

## 2.4 Laboratórios na educação

*“Teoria sem prática é utopia,  
prática sem teoria é rotina.”*

Anónimo<sup>4</sup>

Do que foi escrito no Capítulo1, pode-se concluir da importância dos laboratórios na educação ou, neste caso particular, do laboratório de eletrónica, principalmente em contexto de sala de aula.

A investigação sobre a eficácia do ensino e da aprendizagem num laboratório <sup>5</sup> remonta à década de 1960 e existem inúmeros trabalhos de investigação, ensaios e teses de doutoramento sobre esta matéria [8].

Algumas observações interessantes foram feitas nesta investigação[8]: **e estão de acordo com a secção anterior (**No capítulo/secção anterior**)**

- As actividades laboratoriais escolares têm **potencial especial como meios e estratégias para ensinar e aprender** que podem promover importantes resultados de aprendizagem das ciências para os alunos;

<sup>4</sup>Como curiosidade, esta frase estava num cartaz pendurado na sala de aula de eletrónica do 8º ano, no ano lectivo de 1985/86.

<sup>5</sup>A investigação feita por Hofstein [8] incide sobre laboratórios de ciências, nomeadamente de química. No entanto, as conclusões podem ser adaptadas ao caso particular dos laboratórios de eletrónica

(Este facto é comprovado nas aulas práticas dos meus alunos. Os alunos sentem-se muito mais motivados para aprender utilizando qualquer tipo de recurso, quer seja um laboratório real ou um simulador *online*).

- Os professores precisam de conhecimentos, competências e recursos que lhes permitam ensinar eficazmente em ambientes de aprendizagem práticos;

(Isto está de acordo com o que foi mencionado na secção anterior: Os professores devem desenvolver a sua própria literacia digital. Esta adaptação está a revelar-se um grande desafio).

- As percepções e os comportamentos dos alunos no laboratório de ciências são muito influenciados pelas expectativas e práticas de avaliação dos professores e pela orientação do guia de laboratório, das fichas de trabalho e dos meios electrónicos associados;

(Obviamente isto requer muito trabalho, empenho e dedicação, e os professores também precisam de motivação e das condições necessárias para poderem desenvolver este tipo de ensino. Em Portugal, sabe-se que vivemos tempos de incerteza relativamente a todas estas condições e situações.).

- Os professores precisam de formas de descobrir o que os seus alunos estão a pensar e a aprender no laboratório de ciências e na sala de aula.

(Precisam de fazer uma avaliação formativa, cujo objetivo não é atribuir classificações/dar notas aos alunos, mas determinar o que aprenderam e como chegaram lá, o que não aprenderam e porque é que isso aconteceu. Assim, poderão encontrar estratégias/recursos para ajudar os seus alunos a ultrapassar os obstáculos que vão surgindo ao longo do processo.).

Na mesma investigação é dito algo bastante interessante: “Se usado corretamente, o laboratório tem o potencial de ser um meio importante para introduzir os alunos nos conhecimentos e competências conceptuais e processuais centrais da ciência (Bybee, 2000) como citado em [8].”

Este é um ponto muito importante, uma vez que a investigação demonstrou que as experiências práticas em laboratórios desempenham um papel central (possivelmente *O* papel central) na educação científica, como referem, por exemplo, Hofstein e Lunetta, (2004), como citado em [9]. Isto deve-se em grande parte ao seu forte impacto nos resultados da aprendizagem e no desempenho dos estudantes e à sua eficácia na preparação dos estudantes para a sua futura profissão, como referem, por exemplo, Basey et al., (2008), citados [9].

Em Portugal, o estudo da eletrónica no ensino secundário é feito através do ensino profissional. Este tipo de ensino não só permite que os alunos entrem na universidade com os conhecimentos/competências laboratoriais adequados, como também lhes fornece as ferramentas necessárias para melhor enfrentarem o mercado de trabalho[51].

**Nota: *falar aqui nas competências dos alunos - A REVER! - OPINIÃO PROF***

É claro que esta ideia só é válida se houver um ensino eficaz e, subsequentemente, uma aprendizagem efectiva utilizando os laboratórios.

Na nossa carreira de mais 20 anos como professor de eletrónica, a leccionar a turmas entre o 10º e o 12º ano, temos utilizado de ferramentas digitais como o *multisim*<sup>6</sup> (tanto *online* como *stand-alone*) e o *falstad*<sup>7</sup> como complemento das práticas laboratoriais.

No entanto, devido a dificuldades de várias ordem (nomeadamente económicas) que as escolas têm enfrentado ao longo dos anos e que geram constrangimentos materiais, foi necessário desenvolver a pesquisa e utilização de novas ferramentas digitais, ou se assim lhe quisermos chamar, laboratórios virtuais e/ou simuladores. Este tipo de dificuldades aliado ao aparecimento do COVID-19, com a consequente transformação das aulas normais em aulas virtuais, obrigou ainda mais à procura de novas ferramentas digitais, para além das já referidas nos parágrafos anteriores.

É, pois, inquestionável a importância das práticas laboratoriais nas ciências - ou no caso particular desta dissertação - nas aulas práticas de eletrónica, tanto no Ensino Secundário como no Ensino Superior. A tentativa de mitigar as dificuldades já referidas neste capítulo levam à procura de soluções alternativas. Vários estudos sugerem que os laboratórios virtuais e remotos surgem como duas soluções que podem ajudar a ultrapassar a falta de laboratórios reais ou a falta de recursos nos laboratórios reais [52] [53] [54] [55].

#### 2.4.1 Tipos de laboratórios

No contexto educativo e científico, existem diversos tipos de laboratórios, cada um atendendo a diferentes necessidades e objetivos. Os laboratórios são espaços cruciais para a educação, pesquisa e inovação, proporcionando ambientes onde teorias podem ser testadas e conceitos podem ser explorados de maneira prática [56]. Os principais tipos incluem laboratórios físicos tradicionais, laboratórios virtuais e laboratórios remotos. Oferecem diferentes vantagens e desafios, sendo frequentemente utilizados de maneira complementar.

Os laboratórios físicos, virtuais e remotos oferecem diferentes vantagens e promovem diversos desafios, sendo frequentemente utilizados de maneira complementar. Os laboratórios físicos tradicionais proporcionam uma experiência prática essencial,

---

<sup>6</sup><https://www.multisim.com/>

<sup>7</sup><https://www.falstad.com/circuit/>

permitindo a manipulação direta dos materiais e equipamentos. Os laboratórios virtuais oferecem uma alternativa segura e económica, ideal para a simulação e modelagem de conceitos complexos. Já os laboratórios remotos combinam os benefícios dos dois, permitindo o controle de equipamentos reais à distância e facilitando a colaboração global. Sendo assim, cada tipo de laboratório desempenha um papel crucial na educação e na pesquisa e a escolha entre eles depende das necessidades específicas da disciplina, dos recursos disponíveis e dos objetivos pedagógicos. A integração desses diferentes tipos de laboratórios pode proporcionar uma experiência de aprendizagem rica e diversificada, preparando melhor os alunos para os desafios do mundo real e do ambiente académico [9] [52] [56].

Interessa nesta fase definir a classificação dos laboratórios em geral, classificação esta que, mesmo assim não é consensual, até porque não há uma normalização *standard* na forma como se definem estes conceitos.

Como exemplo desta não uniformização, uma forma possível de classificar os laboratórios está representada na Figura 2.1. Segundo Zutin, *et al.*,(2010), [57], citados por Zapata-Rivera e Larrondo, (2017), [58] os laboratórios podem ser divididos consoante a sua localização: local ou remota.

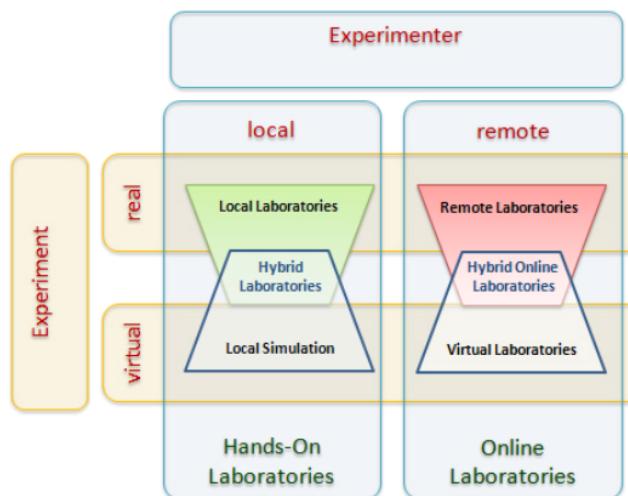


Figura 2.1: Tipos de laboratórios consoante a localização, segundo Zutin, *et al.* [57]

Também Heradio, de la Torre e Dormido(2016)[54] classificam os laboratórios de forma idêntica, tal como é representado na Figura 2.2:

As representações têm algumas diferenças ao nível da nomenclatura. No entanto, Zutin, *et al.*,(2010), [57], introduzem os conceitos de *Hybrid Laboratories* e *Hybrid Online Laboratories*. Não é propósito desta dissertação um estudo aprofundado destes conceitos, sendo que o enfoque está no estudo dos laboratórios remotos. Apresenta-se, de qualquer das formas, uma breve explicação dos laboratórios híbridos.



Figura 2.2: Tipos de laboratórios consoante a localização, Heradio, de la Torre & Dormido, (2016) [54]

Este tipo de laboratórios é criado através da combinação de laboratórios físicos e/ou laboratórios *online*/remotos [58]. Seguem-se alguns exemplos de configurações possíveis:

- Acesso a um laboratório real (no local) com acesso local a um Laboratório Virtual (LV) - experiência já realizada em contexto de sala de aula: os alunos testam o circuito no *multisim online*, antes de o testarem na placa branca;
- Acesso remoto a um laboratório real (no local) e acesso remoto a um LV - experiência a realizar com os alunos, por exemplo, acedendo ao VISIR e simulando o circuito no *multisim online*;
- Acesso remoto ao laboratório real (no local) e acesso local a um LV - experiência idêntica à referida no ponto anterior com o *multisim online* a ser substituído pela versão instalada no PC.

Portanto, a forma como são classificados os laboratórios, segundo Heradio, de la Torre & Dormido,(2016), [54], representados na Figura 2.2 - enquadra-se melhor nos conceitos e objectivos abordados neste capítulo:

- Acesso local;
  - Laboratório real
  - LV
- Acesso remoto.
  - Laboratório remoto
  - LV

### Uma questão de conceitos

Antes de mais importa clarificar os conceitos de “laboratório virtual” e “simulador” (válida para as versões *online* e locais), já que estas duas definições se sobrepõem na maioria dos casos. Alguma literatura, trabalho de investigação ou informação *online* usam já o termo “laboratório virtual” [9][46][59], visto que a montagem nesses sistemas é feita de forma a ter semelhança física com um laboratório real. No entanto, o propósito é o mesmo: o de **simular**.

#### Uso em sala de aula - REFERIR??

Nestes dois exemplos demonstrativos<sup>8</sup> - apresentados na Figura 2.3 - o *Tinkercad*<sup>9</sup> é um LV e um simulador. O *Multisim*<sup>10</sup>, por sua vez, é um simulador, já que a montagem dos circuitos não tem semelhança física com os circuitos montados num laboratório real.

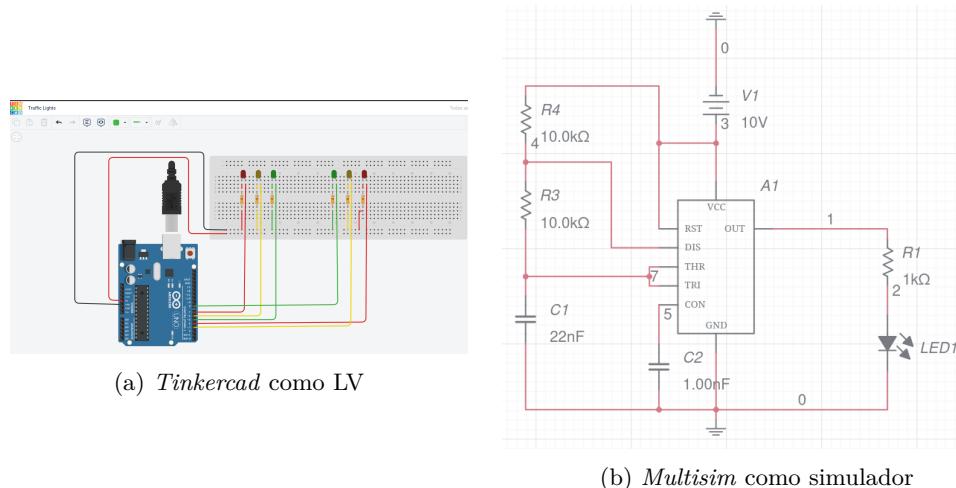


Figura 2.3: LV vs simulador

Na Figura 2.4, apresentamos outro exemplo de um LV recorrendo ao *Fritzing*, que também é um simulador: Tal como foi referido anteriormente, não há um consenso na forma como são denominadas estas ferramentas. Faiña, (2022), [60], refere-se ao *Tinkercad* como sendo um simulador, “Existem vários simuladores destinados a estudantes de eletrónica, como o *TinkerCAD*”. No entanto, no mesmo trabalho de investigação, refere que “O simulador apresentado neste artigo é implementado no *Fritzing*(...).” [60]. Já Knörig, Wettach e Cohen, (2009), no seu artigo *“Fritzing: a tool for advancing electronic prototyping for designers”* referem que “(...) decidiu permitir ao utilizador documentar o seu protótipo de trabalho baseado numa placa de ensaio com uma metáfora visual que imita a situação do utilizador no mundo real.” [61].

<sup>8</sup> *Multisim* disponível em link

<sup>9</sup> <https://www.tinkercad.com/>

<sup>10</sup> <https://www.multisim.com/>

Figura 2.4: *Fritzing* (Exemplo)

Além disso, “(...) um laboratório virtual pode ser definido como um ambiente no qual as experiências são conduzidas ou controladas parcial ou totalmente através de operação, simulação e/ou animação por computador, quer localmente, quer remotamente através da *Internet*.” [62]. Os laboratórios virtuais oferecem aos alunos um conjunto de oportunidades diferentes, não só como substituto, mas também como complemento dos laboratórios reais.

Assim sendo, e sem prejuízo para os assuntos abordados nesta dissertação, já que não existe uma uniformização dos critérios no que toca às definições e conceitos, é seguro considerar que todos os laboratórios virtuais são simuladores. Heying, Kejie e Jiang, (2010), [63] e Umenne, Hlalele, (2020), [64] consideram nos seus trabalhos de investigação o *Multisim*[65] como um LV. Este conceito é, pois, abrangente o suficiente para abranger o que se entende por simuladores. Doravante, neste documento, serão considerados e utilizados os termos “laboratório virtual” ou “laboratórios virtuais”.

### Laboratório real

**link para não esquecer** <https://digital.dge.mec.pt/laboratorios-de-educacao-digital>

Neste tipo de laboratórios, os alunos têm à sua disposição os recursos, o equipamento e os materiais físicos necessários para realizar experiências, analisar dados, desenvolver competências de trabalho em equipa e cultivar o interesse pela ciência,

por exemplo. Estas competências são cruciais para muitas carreiras e profissões, especialmente as relacionadas com a ciência, a tecnologia, a engenharia e a matemática - STEM.

Além disso, os laboratórios podem ajudar a melhorar a compreensão dos alunos sobre o processo científico e a importância da investigação científica.

O grande problema dos laboratórios reais é a falta de condições e recursos para os equipar de forma adequada de modo a que possam apoiar, por exemplo, uma turma de trinta alunos. Este não é um problema exclusivamente português, uma vez que, de um modo geral, “a quantidade de trabalho prático em laboratório no ensino da engenharia tem vindo a ser reduzida pouco a pouco durante as últimas décadas do século passado” [66].

No mesmo artigo [66] são referidas as principais razões para a diminuição dos laboratórios reais, o que curiosamente reflecte o que se passa atualmente no sistema educativo português:

- Número excessivo de alunos por turma [67] - embora o número tenha sido reduzido para entre 23-28, para aplicar o ensino laboratorial em contexto de sala de aula, os recursos materiais necessários seriam incomportáveis;
- Falta de investimento[39], [40], [41] - falando por experiência própria, em todas as escolas por onde passamos, a falta de verbas (ou a demora em recebê-las...)” foi constante; - **A REVER esta afirmação**
- Custos excessivos dos laboratórios - no contexto português e face a este número de alunos, a montagem e manutenção de um laboratório desta natureza seria demasiado dispendiosa, para além de que os laboratórios não estariam disponíveis durante o tempo que seria desejável;
- Falta de professores/Poucos professores - “Nos próximos sete anos, segundo números do *Conselho Nacional de Educação*, cerca de 20 mil professores do ensino público irão reformar-se. Atualmente, mais de metade dos professores tem 50 anos ou mais.” [68].

Sendo assim, que estratégias foram usadas para mitigar estes entraves?

Falando especificamente nos laboratórios do ensino secundário e por experiência própria, uma forma encontrada para equipar os laboratórios passa pela aquisição de *microcontroladores* ( $\mu$ C) - nomeadamente, *Arduinos*, *ESP32* ou *Raspberry Pi*. Todos estes dispositivos - em conjunto com a vasta variedade de sensores disponíveis - já são amplamente utilizados em ambientes educacionais e de pesquisa para a prototipagem rápida, experimentação e aprendizagem prática em engenharia e diversas disciplinas, como eletrónica ou física. Têm como principal vantagem o custo-eficácia. Estes dispositivos e os sensores são relativamente baratos em comparação com equipamentos de laboratório tradicionais, tornando-os acessíveis para

escolas e universidades com orçamento limitado. A comunidade de suporte e a documentação é vasta e isso facilita a aprendizagem e a implementação de projetos por estudantes e professores. Com uma enorme gama de sensores e módulos disponíveis no mercado, estes dispositivos podem ser utilizados para criar uma infinidade de experiências e projetos, desde simples medições de temperatura, passando pela robótica até complexos sistemas de automação e controlo. Em Portugal existem uma série de eventos e competições robóticas, que se destinam a promover o espírito inovador e empreendedor dos alunos, através de métodos activos de ensino, assim como a aquisição de competências transversais. Permite, ainda, por em prática as aprendizagens realizadas em contexto de sala de aula e em ambiente laboratorial [69][70][71]:

- *RoboParty*;
- Festival Nacional de Robotica;
- *Cansat*.

Yoder (2015) [72], desenvolveu uma alternativa ao laboratório tradicional recorrendo ao *Arduino*: “(...) a plataforma Arduino estimulou o interesse e o envolvimento dos alunos desde o início. Vários alunos começaram a alargar os projectos para além dos requisitos na terceira semana de aulas.”

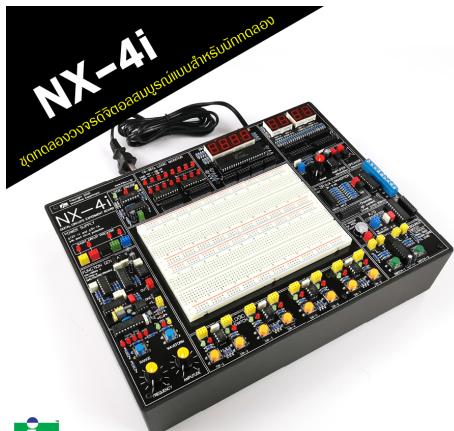
Graven, *et al.*,(2018), [73], citados por Plaza, *et al.*,(2018), [74] referem que “(...) No domínio da educação, o *Arduino* é também muito utilizado. Os estudantes de engenharia electrotécnica podem aceder a um laboratório compacto para trabalhar quando e onde quiserem.”. O mesmo autor refere ainda que “(...) Tal como foi demonstrado por muitas experiências e trabalhos, a robótica combinada com STEM constitui uma forma atractiva de transformar conceitos aborrecidos num processo de aprendizagem divertido.”

Marzoli *et al.*,(2021), [75], no trabalho de investigação “*Arduino: From Physics to Robotics*”, discutem vários pontos e questões, entre os quais se o *Arduino* pode melhorar a prática laboratorial no ensino secundário italiano e mudar as atitudes dos alunos em relação às disciplinas STEM e “Como melhorar as práticas laboratoriais no ensino secundário, tendo em conta os orçamentos e as instalações limitadas disponíveis?” As conclusões a que chegaram ajudam a revelar a potencialidade do *Arduino* em contexto laboratorial: “A interação profícua entre professores com diferentes formações foi fundamental para a concepção de novas soluções e a exploração de novas aplicações. (...) Assim, o *Arduino* pode ser considerado como uma espécie de micro-laboratório (...)”[75].

As possibilidades oferecidas por estes dispositivos - nomeadamente, o *Arduino* - em contexto de sala de aula e como parte de um laboratório, são imensas, permitindo a realização de uma vasta variedade de projetos inovadores. Esta flexibilidade faz

dele uma ferramenta valiosa para a educação e a investigação, facilitando a aprendizagem prática e o desenvolvimento de competências em eletrónica e programação.

#### Uso em sala de aula - REFERIR??



(a) Tinkercad como LV



(b) Tinkercad como um simulador *online*

Figura 2.5: Tinkercad

**Colocar fotos?**

**dar mais algum exemplo do que existe? PocketLAb?**

**osciloscópio portátil DSO NANO?**

**Projecto de uma caixa de sistemas digitais?**

**A REVER - opinião PROF**

#### Laboratório virtual - Acesso local e/ou remoto

Um LV é uma poderosa ferramenta que utiliza modelos matemáticos para emular dispositivos reais. Atualmente, os programas de simulação por computador são um padrão de desenvolvimento de produtos aceite pela indústria e são amplamente utilizados no ensino [54] [55].

No fundo, a diferença entre o acesso remoto e local prende-se com a forma de acesso ao LV. Dos programas usados em contexto de sala de aula, só um possui uma versão remota e local: *Multisim*[65]. As diferenças situam-se ao nível da interface gráfica e, sobretudo, na quantidade de recursos disponível, sendo que na versão local é obviamente muito mais extensa e completa, como se pode ver na Figura 2.6 e Figura 2.7.

O *Tinkercad*[76] só tem versão *online*, embora se possa instalar um *plug-in* no PC, que dispensa o uso do navegador. O *Fritzing* [77] é acedido localmente, não tendo versão *online*.

Estes programas de simulação oferecem aos alunos um conjunto de diferentes oportunidades, não só como substituto, mas também como complemento aos laboratórios reais[78]. Os alunos podem fazer experiências arriscadas sem se porem em



Figura 2.6: Multisim local



Figura 2.7: Multisim remoto

perigo. Considerando o nosso caso pessoal, numa turma com 20 ou mais alunos e numa primeira fase, é mais seguro simular um circuito de potência utilizando componentes como TRIAC, DIAC ou SCR. O complemento poderá vir com a montagem real.

Os laboratórios virtuais permitem a verificação e o ensaio de circuitos analógicos e digitais complexos. Têm a capacidade de reproduzir componentes reais com precisão suficiente. As faculdades e universidades de todo o mundo introduzem habitualmente este tipo de *software* de simulação nos cursos de engenharia. Uma vez que todos os circuitos são virtuais, os estudantes podem trabalhar e experimentar uma vasta gama de elementos possíveis sem o perigo de danificar ou destruir o equipamento.

**esta frase pode ficar mais à frente em jeito de conclusão** Já foi visto na secção 2.4.1 - **REVER A REFERÊNCIA** que os resultados não são prejudicados pela utilização dos LRs. **IMPORTANTE - REferências - há um trabalho para apresentar como referência** No caso de versão do LV ser remota, as vantagens são semelhantes às de trabalhar com qualquer programa na nuvem:

- Não há necessidade de instalar *software* adicional no computador (versão *online*);
- Desde que a velocidade da *Internet* seja estável, o simulador pode ser acedido em qualquer lugar;
- Os circuitos são guardados e armazenados na nuvem;
- É ideal para trabalhos colaborativos e partilha de recursos.

No entanto, se o LV for apresentado uma versão de acesso local, ou seja, que é instalada no PC, é óbvio que estas vantagens não se aplicam.

Mas as vantagens dos LVs vão muito para além das descritas em anteriormente [79][80][81][82]. Oferecem um ambiente seguro para a realização de experiências complexas e (potencialmente) perigosas, reduzindo os riscos para os estudantes e professores. Os estudantes podem progredir ao seu próprio ritmo e repetir as experiências as vezes que forem necessárias, estando os laboratórios acessíveis a qualquer altura e em qualquer lugar. Os laboratórios virtuais são, por inerência, mais baratos, de fácil instalação e manutenção e - a juntar a estes factos vantajosos - há ainda informação de ajuda e tutoriais *online*.

Mas os laboratórios virtuais também têm as suas desvantagens. A principal pretende-se com o desfasamento da realidade. Pegando no exemplo anterior - a simulação de circuitos de potência - o facto de nada de mau poder acontecer pode levar a uma atitude de irresponsabilização ou falta de cuidado, já que o estudante pode ser levado a pensar que nada de grave poderá acontecer [55]. Há sempre constrangimentos no que diz respeito à evolução tecnológica dos sistemas que suportam

o LV, assim como à disponibilidade de uma ligação à *Internet* adequada ao seu uso. Se não for usado em contexto de sala de aula, o LV reduz a interação direta entre os alunos e entre estes e os professores, uma vez que a comunicação é sobretudo feita *online*. Além do mais, exige do aluno mais iniciativa e autonomia. Há ainda a referir algumas vantagens que se podem tornar desvantagens. O facto de o aluno poder testar e simular vezes sem conta, acabará por lhe retirar sensibilidade quando utilizar os laboratórios e componentes reais; a imensa quantidade de informação disponível *online* terá que ser “filtrada” pelo aluno[55][82][83][84].

Um dos principais simuladores disponíveis gratuitamente, embora com limitações de funcionalidades, usado numa base regular em contexto de sala de aula, é o *Multisim*[65]. Em tempos de confinamento, este LV foi um “salva-vidas”. Existe também uma versão *premium* que oferece acesso a uma gama mais completa de recursos *online* e uma versão que pode ser instalada no PC. (O funcionamento destas duas versões é em tudo idêntico, portanto, quando nos referirmos ao *Multisim*[65], entende-se as duas versões, sem perda de generalidade. Nos casos em que houver necessidade de distinção, esse facto será devidamente referenciado). Este *software* integra a simulação virtual *Simulation Program for Integrated Circuits Emphasis* (SPICE) padrão da indústria com um ambiente esquemático interativo para visualizar e analisar instantaneamente o comportamento de circuitos electrónicos. É uma ferramenta poderosa que utiliza modelos matemáticos para simular componentes reais, dispositivos ou circuitos com uma precisão muito boa, através de um navegador *web*. Fornece uma plataforma para “colmatar” a lacuna entre a teoria dos manuais e os circuitos reais e também proporciona aos estudantes uma boa plataforma para experiências abrangentes e inovadoras[65].

Além disso, dispondo de poderosas funcionalidades de aprendizagem e integração de *hardware* de laboratório, o *Multisim* ensina aos alunos conceitos fundamentais de eletrónica analógica, digital e de potência presentes em todo o currículo de engenharia e ciências [85].

Citando Heying, Kejie e Li, (2010), “(...) podemos concluir que a criação de uma plataforma de simulação virtual através do *Multisim* num computador permite construir facilmente todo o tipo de circuitos. (...) Seguem-se os comentários de alguns alunos:”

- “O ensino experimental utilizando o *Multisim* é uma boa abordagem para me ajudar a compreender as matérias;”
- “Gostei de trabalhar na plataforma de simulação virtual *Multisim*;”
- “A simulação *Multisim* ajudou-me a compreender melhor as experiências.”

Os mesmos autores concluem que: “O laboratório virtual [*Multisim*] desempenha um papel muito importante na atualização do método de ensino experimental, na

melhoria da qualidade do ensino dos cursos em circuito e na otimização do efeito do ensino.”[86]

Outro laboratório bastante usado em contexto de sala de aula foi criado e desenvolvido por *Paul Falstad*, em 1985 e está disponível em Falstad [87]. Este LV é livre e de código aberto [88]. Funciona num navegador *web* como uma *applet JAVA*. É também importante por ser de fácil interacção e pela simplicidade na representação de circuitos eléctricos, aspectos muito importantes para os recursos multimédia nas áreas da electrónica e da electricidade<sup>11</sup>.

Além disso, o simulador *Falstad* tem sido desenvolvido como uma aplicação de *Internet* há muitos anos, segundo da Silva *et al.*, (2011), [90], citados por *Falstad*,(2024), [87].

O principal problema é que, devido à falta de parametrização avançada e de um modelo matemático, este LV não se destina a engenharia profissional ou real. No entanto, para fins didácticos, é muito eficaz. Por exemplo, é de referir que:

- Os pontos amarelos animados representam correntes e os seus movimentos demonstram um fluxo de cargas eléctricas em ”tempo real“;
- De acordo com a direção do fluxo da corrente, a cor do fio é representada por verde ou vermelho, tendo em conta a direcção do fluxo da corrente;
- Os componentes mostram diferentes tensões no circuito;
- As formas de onda também são representadas;
- Podem ser representados e controlados diferentes tipos de interruptores, por exemplo, para fazer variar frequência e/ou tensão, Figura 2.8, o que permite o utilizador observar os processos transitórios;
- A velocidade da simulação e a corrente também podem ser controladas.

A variedade de laboratórios virtuais no contexto da electrónica é muita, pelo que só referimos os mais usados em contexto de sala de aula. No entanto, outras soluções seriam de considerar, tais como:

- *Tinkercad* - é uma aplicação *online* gratuita para projetos 3D, de eletrónica e de programação;
- *Fritzing* - a interface é muito semelhante à do *TinkerCad*, com a excepção de não poder realizar simulações, no entanto, fornece uma variedade muito maior de componentes do que o *TinkerCad*;

<sup>11</sup>Para sermos mais rigorosos, este LV não se cinge somente aos circuitos electrónicos, ligando-se também a outros ramos da ciência como a termodinâmica, a mecânica quântica, o processamento de sinal, etc. [89]

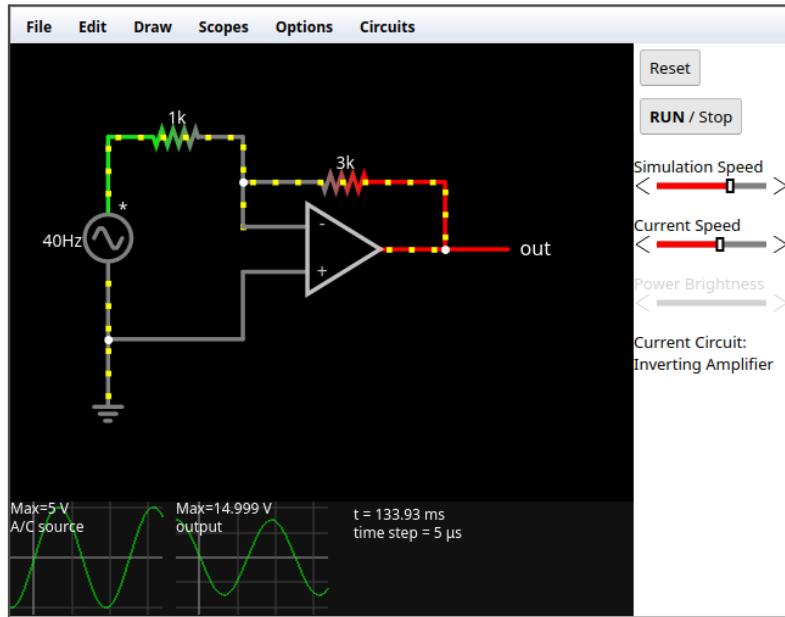


Figura 2.8: Falstad - amplificador inversor[87]

- *Wokwi*[91] - LV que usa as mais variadas placas que simulam o *Arduino*, *ESP32*, *STM32*, *RaspberryPi Pico* e uma grande variedade de sensores[91].

### FALTA AQUI UMC CONCLUZÃOZINHA

#### Laboratório remoto

Alguns dos problemas dos laboratórios reais já foram abordados ao longo destes dois capítulos. As vantagens são bastante evidentes e prendem-se, sobretudo, com a possibilidade de os alunos trabalharem utilizando recursos físicos. No entanto, no ensino secundário, nem sempre é possível ter acesso a esses recursos, quer por dificuldades em obter os fundos necessários para manter um laboratório, quer porque é simplesmente incomportável montar um laboratório onde vinte ou vinte e quatro alunos possam realizar experiências ao mesmo tempo. No ensino superior, em que, pelo menos teoricamente, os alunos são mais autónomos do que no ensino secundário, o principal problema é a disponibilidade de laboratórios fora das aulas normais.

Numa época em que a tecnologia é cada vez mais encarada como um facilitador no processo de ensino/aprendizagem, a utilização de laboratórios remotos é cada vez mais comum e generalizada. Além disso, os laboratórios remotos apresentam algumas vantagens em relação aos laboratórios reais e mesmo aos simuladores [92], tais como: flexibilidade, acessibilidade, disponibilidade e segurança.

**ATENÇÃO AOS ACRÓNIMOS DESTE PARÁGRAFO**Estes tipos de

laboratórios - que incluem, por exemplo, o VISIR, o *LabsLand*<sup>12</sup> ou o *WebLab-Deusto*<sup>13</sup> - permitem que professores/investigadores e alunos acedam a equipamentos e/ou computadores através da Internet para realizar experiências e tarefas laboratoriais sem estarem no espaço físico dos laboratórios [93].

Num LR, a interação tem lugar à distância com a ajuda da infraestrutura remota. Esta é uma nova camada que se situa entre o utilizador e o equipamento do laboratório. É responsável pela transmissão das ações do utilizador e pela receção da informação sensorial do equipamento. Várias investigações mostram que os estudantes podem efetivamente aprender com a utilização de laboratórios remotos e também de laboratórios virtuais, obviamente se estiverem empenhados no que estão a estudar [92].

De acordo com Viegas *et al.*, (2018)[52], citando Brinson, (2015)[9] e Corter *et al.*, (2011)[11], os resultados obtidos com simuladores e laboratórios remotos podem ser considerados semelhantes ou mesmo superiores aos dos laboratórios práticos reais. No entanto, uma abordagem à aprendizagem laboratorial que utilize uma combinação de laboratórios práticos, simulações e procedimentos de laboratórios remotos parece ser a forma mais eficaz de aprendizagem, tirando partido dos benefícios dos três [9].

Esta mesma conclusão foi obtida por Mel e Samaranayaka, (2017), no estudo intitulado “Extending the boundaries of remote laboratory by providing hands on experience”. Citando os autores: “(...) os estudantes estão disponíveis a usar o LR juntamente com o real (...)”, mas “(...) não aconselham a total substituição (...).” O estudo conclui, ainda, que não há grande diferença entre o laboratório real e remoto, sendo que os conhecimentos adquiridos são (praticamente) os mesmos[94].

Podem ser considerados outros estudos que, por exemplo, comparem laboratórios reais (práticos) e simulados, laboratórios reais e virtuais. De facto, Corter *et al.*, (2007) concluem que “(...) os laboratórios remotos e simulados podem ser pelo menos tão eficazes como os laboratórios práticos tradicionais no ensino de conceitos específicos da disciplina”[95]. Outro estudo feito por Kocijancic e O’Sullivan, (2004) conclui que “(...) não se trata de saber se é melhor utilizar experiências reais ou laboratórios virtuais no ensino das ciências, uma vez que ambas as abordagens, utilizadas de forma complementar, podem contribuir para uma aprendizagem ativa mais eficaz.” [96]. Tsihouridis *et al.*, (2019), concluem também que “não há um vencedor final nesta controvérsia intemporal entre os dois ambientes de laboratório experimental, de acordo com a nossa investigação.” [97].

Assim, através de estudos efectuados, será seguro afirmar que os laboratórios remotos são a melhor alternativa aos tradicionais em termos de baixo custo e ubiquidade, e algumas universidades já começaram a recorrer a laboratórios remotos

<sup>12</sup><https://labsland.com/en>

<sup>13</sup><https://weblab.deusto.es/website/>

nas suas sessões práticas[98]. **REVER esta frase em jeito de conclusão!**

De facto, de acordo com Nafalski *et al.*, (2010)[99], citando Auer e Gravier, (2009)[100] as razões para o crescimento dos laboratórios remotos em engenharia e ciência prendem-se com:

- A crescente complexidade das tarefas de engenharia;
- O equipamento cada vez mais especializado e dispendioso, as ferramentas de software e os simuladores necessários;
- A necessidade de utilizar equipamentos e ferramentas de *software*/simuladores dispendiosos em projectos com prazos curtos (como os apresentados em [93]);
- A aplicação de equipamentos de alta tecnologia necessários em pequenas e médias empresas;
- A necessidade de pessoal altamente qualificado para controlar os novos equipamentos;
- As exigências da globalização e da divisão do trabalho.

No entanto, se se pretende, segundo Fan, Evangelista e Indumathi, (2021), [30], citando Tawfik *et al.*(2016)[92], Chen *et al.*, (2010)[101] e Simão *et al.*, (2014), [102], incorporar os laboratórios remotos com Industry 4.0, já discutidas no Capítulo 1, tem vantagens óbvias:

- Os estudantes podem fazer os exercícios da disciplina ao seu próprio ritmo e de acordo com o seu nível de interesse;
- A quantidade de tempo (muitas vezes horas extraordinárias) que os professores têm de despender na preparação e ensino dos laboratórios pode ser reduzida;
- Podem ser efectuadas várias experiências utilizando a mesma configuração;
- Os laboratórios remotos permitem o acesso de um maior número de utilizadores, o que significa que a sua instalação acaba por ser mais barata do que a dos laboratórios físicos;
- Os laboratórios remotos estão acessíveis 24 horas por dia, 7 dias por semana;
- Os laboratórios remotos podem ajudar a reforçar o trabalho autónomo dos alunos;
- Os laboratórios remotos são mais seguros, tanto para o utilizador como para o equipamento ou software, uma vez que estão fisicamente separados e são orientados para a tecnologia.

Embora os laboratórios remotos ofereçam muitas vantagens, como as descritas em cima, é importante estar ciente das suas desvantagens (muitas delas comuns com as descritas para os laboratórios virtuais). A falta de interacção física directa, a dependência da ligação com a *Internet*, os desafios técnicos e de manutenção, a experiência de aprendizagem menos imersiva e a limitação da interacção com colegas e professores são pontos críticos que devem ser considerados ao implementar esses laboratórios. No entanto, a principal desvantagem dos laboratórios remotos, prende-se com a complexidade da sua implementação. Como os LRs seguem uma arquitectura cliente-servidor, além do servidor é ainda necessário ter em conta a questão do *hardware* e a sua integração com o *software* que pode, ou não, ser proprietário. Há ainda a questão essencial da forma como os utilizadores accedem ao laboratório. Isto levanta problemas de largura de banda, que restringe o número de utilizadores que podem aceder simultaneamente ao LR [54].

Sendo assim, os laboratórios remotos não são simplesmente uma versão reduzida de um laboratório real (prático), mas uma abordagem diferente com pontos fortes e fracos alternativos, oferecendo novas oportunidades que não são proporcionadas pelas abordagens tradicionais. No final, se forem combinados e complementados com laboratórios reais e simulações, é possível tirar partido das vantagens que estes três tipos de laboratórios proporcionam.

Existem muitos tipos de laboratórios nos mais variados campos da ciência (Física, Electrónica, Robótica ou Química) e com os mais variados tipos de arquitectura de *hardware* e *software*. Apesar de não estar no objectivos desta dissertação um estudo aprofundado desta matéria, ficam alguns exemplos que esclarecem e atestam o quanto vasto, complexo e variado é este assunto. vale a pena fotos? PROF?

Dentro dos laboratórios remotos que existem e actualmente estão activos, destacam - se os seguintes:

- VISIR - é um LR que interliga vários componentes reais, que podem ser ligados para realizar diferentes tarefas e para construir circuitos específicos concebidos pelo utilizador final, como se pode ver na Figura 2.15; permite a estudantes e professores fazerem medições reais em circuitos reais, algo que não é possível em simulações; será discutido com mais pormenor mais adiante;
- A Universidade de *Deusto*, em Bilbao tem disponível uma conjunto de LRs, que inclui o VISIR, o comando de um robô através de um *Arduino* ou a programação de uma *Field Programmable Gate Array* (FPGA);
- *iSES*<sup>14</sup> - tem várias experiências remotas que abrangem diversas áreas da ciências, como Física, Electrónica, Radioactividade, Electromagnetismo, etc.

<sup>14</sup><https://www.ises.info/index.php/en/systemises/sdkisesstudio>

- *OpenSTEM*<sup>15</sup> - estes laboratórios têm uma coleção de experiências abrangendo áreas científicas que vão desde a saúde, engenharia ou observatórios;
- *Remote-LAB GymKT*<sup>16</sup> - Várias experiências que vão desde o controlo de um braço robótico através de um *Arduino* até experiências no campo da física e da electrónica;

No entanto, a implementação dos laboratórios remotos assenta sobre as mais variadas, e complexas, soluções.

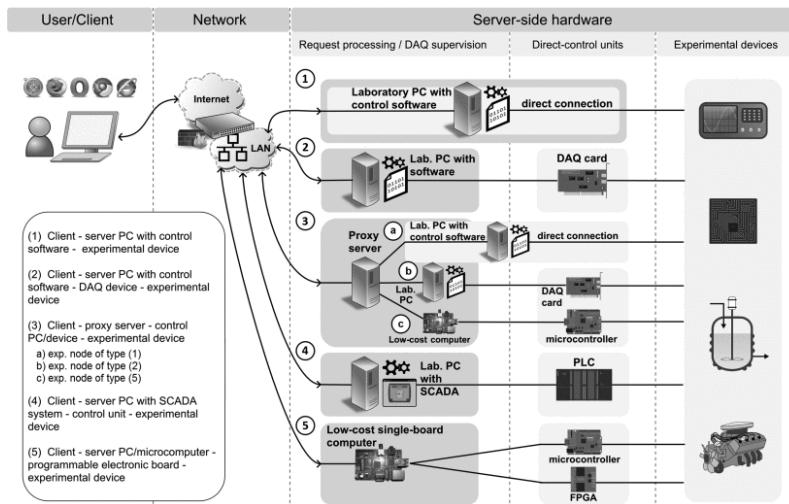


Figura 2.9: Diferentes tipos de arquitectura (resumo)

Segundo Kalúz *et al.*, (2015), Figura 2.9, os tipos mais frequentes de arquitecturas são[103]:

- Cliente-servidor com *software* de controlo - dispositivo experimental: este tipo de arquitetura muito comum utiliza uma ligação direta entre o PC(servidor) e o dispositivo controlado, por exemplo, através de uma porta série ou *Universal Serial Bus* (USB);
- Cliente-servidor com *software* de controlo - dispositivo de aquisição de dados (*Data Acquisition (DAQ)*);
- Configuração experimental remota com funcionalidade muito semelhante à anterior; a única diferença é que o dispositivo de aquisição de dados (normalmente uma placa DAQ) é uma parte separada da arquitetura;
- Nós cliente-servidor-*proxy-laboratório* (PC + unidade de controlo)-dispositivos experimentais: este tipo, também conhecido como arquitetura ramificada, oferece uma maior capacidade de possíveis ligações experimentais do que qualquer outra arquitetura;

<sup>15</sup><https://learn5.open.ac.uk/>

<sup>16</sup><http://remote-lab.fyzika.net/>

- Cliente-servidor com sistema de controlo de supervisão e aquisição de dados (*Supervisory Control and Data Acquisition (SCADA)*) - unidade de controlo - dispositivo experimental: a combinação do sistema SCADA com uma unidade de controlo (*Programmable Logic Controller (PLC)* ou outro controlador de *hardware*) é uma das abordagens industriais mais frequentemente utilizadas;
- Cliente - servidor/microcomputador - placa eletrónica programável - dispositivo experimental: as abordagens muito populares nos últimos anos, também frequentemente designadas na literatura como de baixo custo, baseiam-se em placas programáveis como as FPGA, microcontroladores *Arduino* e alternativas baratas aos computadores normais (por exemplo, computadores de placa única baseados na arquitetura *Advanced RISC Machine (ARM)*, como Raspberry Pi, entre outros);
- *Multipurpose Hardware and Software Architecture (MHSA)*: é composto por uma arquitetura de *hardware*, baseada em dispositivos industriais, que permite ao programador ligar vários tipos de experiências à sua configuração física, limitada apenas pelas capacidades de ligação dos nós experimentais e por um *software* baseado na *web* do lado do cliente, que proporciona uma grande versatilidade tanto para a implementação final como para a utilização; desde o início, o MHSA foi desenvolvido como uma arquitetura para a implementação de laboratórios de sistemas de controlo, sem ter em conta a utilização de dispositivos específicos, reduzindo o *back-end* a um conjunto de sinais.

Fabregas *et al.*, (2011), utilizam o *Simulink*, do lado do servidor, e o *Easy Java Simulations (EJS)*, do lado do cliente, para realizar a implementação do LR[104], Figura 2.10. Ainda no mesmo artigo - Lazar e Carari, (2008) - desenvolveram um

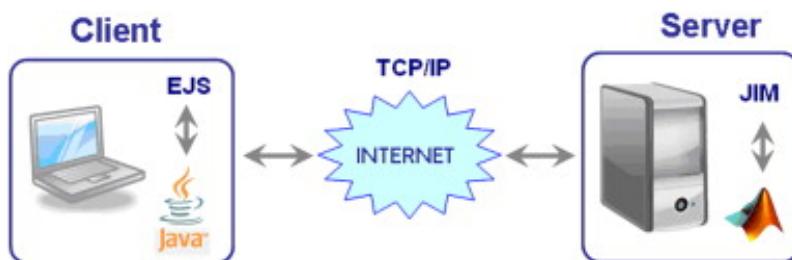


Figura 2.10: Implementação cliente-servidor MATLAB-EJS

LR baseado no *software LOOKOUT SCADA* da *National Instruments (NI)*, [105] citado por [104]. Por sua vez, Dormido *et al.*, (2008), desenvolveram um LR, em que o *Laboratory Virtual Instrument Engineering Workbench (LabVIEW)* estava implementado no servidor, [106] citado por [104]. Em ambas implementações foi utilizada uma arquitectura cliente-servidor.

A *Distance Internet-Based Embedded System Experimental Laboratory* (DIESEL) é outro tipo de arquitectura que é aplicada na implementação de laboratórios remotos. A abordagem cliente-servidor do DIESEL utiliza uma arquitetura de *software* distribuída, desenvolvida com a tecnologia *Microsoft .NET*. É constituída por uma aplicação cliente, um determinado número de estações de trabalho experimentais que alojam aplicações servidoras individuais, um servidor *Web* que aloja uma base de dados, um serviço *Web* e um sistema de reservas baseado na *Web* [107], como se pode ver na Figura 2.11.

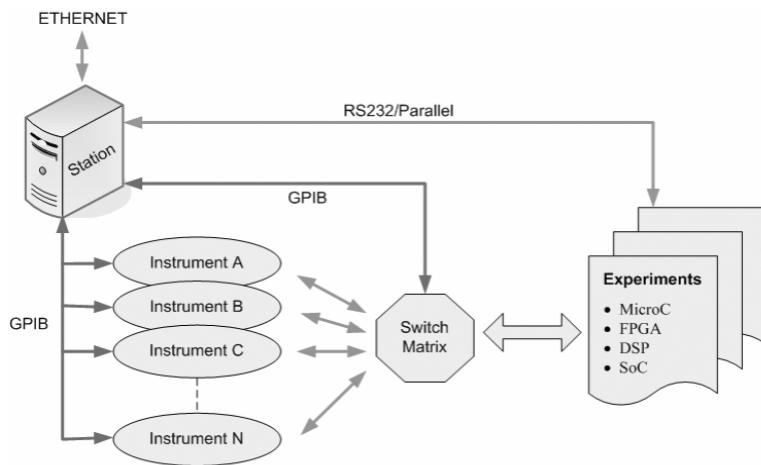


Figura 2.11: Arquitectura DIESEL (genérica)[107]

Jacko *et al.*, (2022) - no seu artigo “Remote IoT Education Laboratory for Microcontrollers Based on the STM32 Chips” - descrevem a implementação de um LR que utiliza o *Microsoft Visual Studio 2019* do lado do servidor [108].

Candelas *et al.*, (2003), implementaram um LR dedicado ao ensino da robótica em que a “ponte” entre a experiência e o utilizador é feita através de *Java Applet*.

No entanto, a implementação dos LRs vai muito para além dos conceitos abordados anteriormente, a **Interoperabilidade** torna possível a integração de diferentes LRs mas levanta outro tipo de problemas, como sendo a gestão dos LRs e funcionalidades, autenticação, autorização ou gestão de utilizadores. Este facto abriu espaços à criação de serviços que pudessem gerir estas funcionalidades, comumente designados por *Remote Laboratory Management System* (RLMS). Trata-se de *software* de gestão que pode suportar vários laboratórios remotos (por exemplo, um laboratório de robótica e um laboratório de eletrónica ao mesmo tempo) e fornece autenticação, autorização, gestão de utilizadores, monitorização de utilizadores e agendamento, bem como *Application Programming Interface* (API)s para desenvolvimento de novas funcionalidades e laboratórios [109].

A *iLab Shared Architecture* (ISA)<sup>17</sup>, Figura 2.12, é uma arquitetura de software, desenvolvida pelo *Massachusetts Institute of Technology* (MIT), que oferece aos criadores e utilizadores dos LRs uma estrutura comum para utilizar, partilhar e disponibilizar mecanismos para a partilha de LRs numa arquitetura distribuída baseada em serviços Web, gerindo o agendamento e a manutenção de sessões de laboratório durante a execução de uma experiência[110]. Os utilizadores acedem a estes LRs através de um início de sessão único e de uma *interface* administrativa normalizada simples. O projeto ISA demonstrou que a utilização de laboratórios pode ser alargada a milhares de estudantes dispersos globalmente[111][112]. Zutin, Auer e Gustavsson, (2011), apresentaram uma solução para integrar o VISIR no ISA[113].

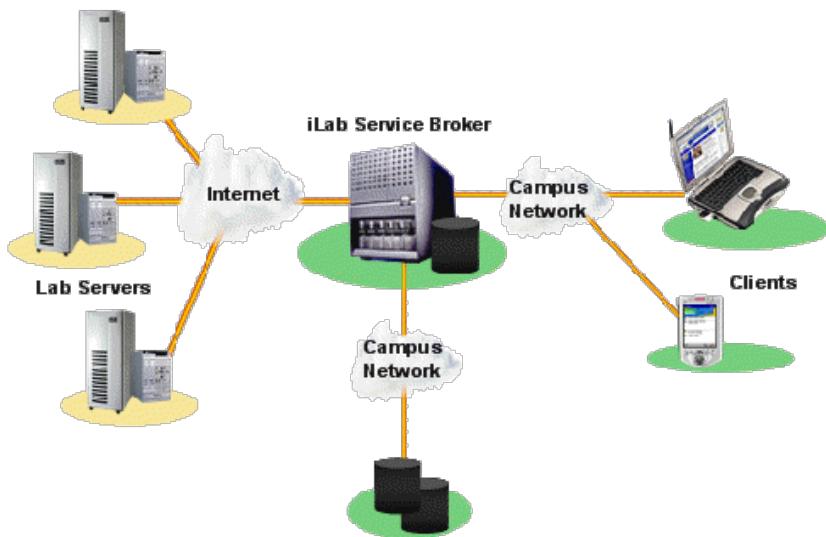


Figura 2.12: Arquitectura ISA[112]

O *WebLab-Deusto* é uma iniciativa desenvolvida pela Universidade de Deusto<sup>18</sup>, Bilbao, e consiste num sistema de gestão de LR de código de fonte aberto (*BSD 2-clause license*). A arquitectura do *WebLab-Deusto* é definida localmente e baseia-se na arquitetura distribuída cliente-servidor **MODULAR? REVER**, como podemos ver na Figura 2.13.

Nesta tipo de arquitectura, os utilizadores(clientes) ligam-se aos servidores principais e estes gerem a autenticação, a autorização, o controlo dos utilizadores, etc. Suporta nativamente a “Federação” de LRs, o que significa que se duas universidades instalarem o *WebLab-Deusto*, qualquer um dos sistemas poderá usar os laboratórios fornecidos pela outra universidade. Além disso, fornece bibliotecas tanto do lado do cliente, como do lado do servidor para lidar com as comunicações e também fornece componentes de *software* que estarão entre cliente e servidor [114].

<sup>17</sup><https://icampus.mit.edu/projects/ilabs/#architecture>

<sup>18</sup><https://www.deusto.es/es/inicio>



Figura 2.13: Arquitectura WebLab-Deusto

A gateways4labs<sup>19</sup> é, também, uma abordagem *open-source* que visa a integração de múltiplos laboratórios remotos em diferentes ambientes de aprendizagem digital (ferramentas digitais) através de uma *interface* única. A arquitectura do *gateway4labs* está representada na Figura 2.14. O lado esquerdo - denominado o “lado do cliente” - contém ferramentas de aprendizagem, como por exemplo o *moodle*. O lado direito - denominado o “lado do laboratório” - representa o RLMS, por exemplo, um dos dois casos abordados anteriormente: *WebLab-Deusto* ou *ISA*[115]. A parte central, *LabManager*, é responsável pela recepção dos vários pedidos de diferentes *Learning Management System* (LMS)s(que é o local onde os alunos e os professores se encontram e onde os alunos encontram outros recursos educativos) e compreenderá os protocolos de diferentes RLMSs, não tendo qualquer código dependente dos LMSs, mas dispondo de *plug-ins* dependentes de RLMSs[115][116].

Uma derivação do *WebLab-Deusto* é o *LabsLand*<sup>20</sup> que interliga escolas e universidades com laboratórios reais disponíveis noutro local<sup>21</sup>. Fornece um repositório de laboratórios remotos de diferentes instituições ligados entre si. Também presta serviços de consultoria e vende laboratórios remotos<sup>22</sup>.

Outra forma de integrar LRs - nomeadamente, o VISIR - é o sistema *Maxwell*<sup>23</sup>, desenvolvido e integrado na Pontifícia Universidade Católica (PUC) - Rio, Rio de Janeiro<sup>24</sup>. A PUC - Rio, possui um LMS integrado num repositório institucional e

<sup>19</sup><https://gateway4labs.readthedocs.io/en/latest/>

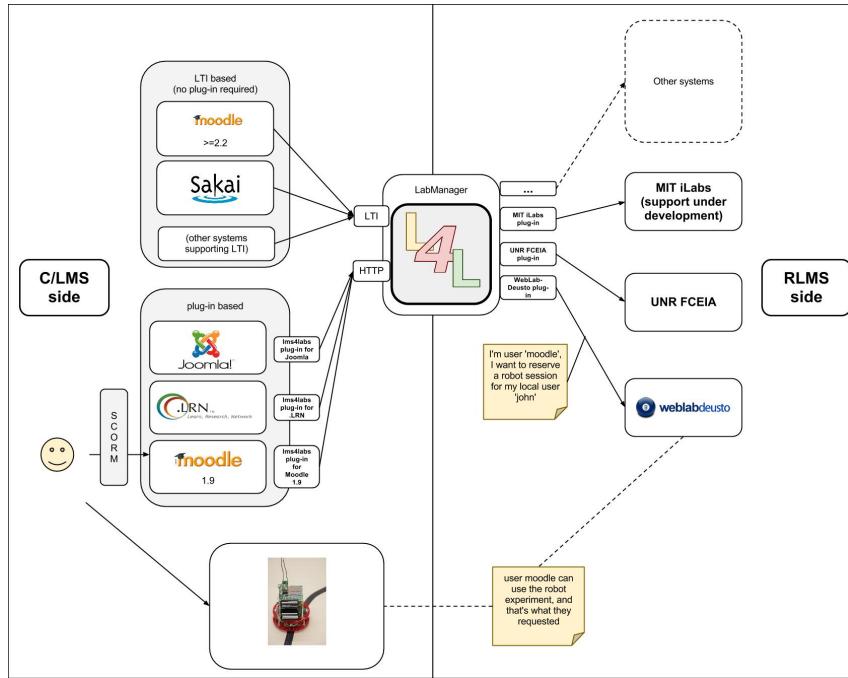
<sup>20</sup><https://labsland.com/>

<sup>21</sup><https://developers.labsland.com/weblablib/en/stable/>

<sup>22</sup><https://weblab.deusto.es/website/index.html>

<sup>23</sup><https://www.maxwell.vrac.puc-rio.br/>

<sup>24</sup><https://www.puc-rio.br/index.html>

Figura 2.14: Arquitectura *gateway4labs*

o VISIR, assim como o LabVIEW, podem ser utilizado a partir do LMS. Os alunos acedem ao LR através do *weblab*[117].

#### - Desenvolver um pouco mais o conceito de “FEDERAÇÃO” - PROF??

**VISIR** No contexto desta dissertação, importa abordar o caso mais particular do VISIR. De facto, este LR esteve na génese da criação do LaRE. Além do mais, está disponível no ISEP, Figura 2.15. Como já foi referido, o VISIR também se encontra implementado na Universidade de Deusto, mais concretamente, como parte integrante do *WebLab-Deusto*.



Figura 2.15: VISIR no ISEP

O VISIR é um LR para concepção, ligação e medição de circuitos electrónicos. O sistema VISIR foi desenvolvido no *Blekinge Institute of Technology* (BTH), *Karlskrona*, Suécia, em 1999, e as suas funcionalidades e comunidade de utilizadores têm vindo a crescer desde então [92]. O projecto foi lançado no final de 2006 em conjunto com a NI, nos EUA, (como fornecedor de instrumentos) e a *Axiom EduTECH* na Suécia (como fornecedor de educação, *software* técnico e serviços de engenharia para análise de ruído e vibrações). Foi apoiado financeiramente pela BTH e pela Agência Governamental Sueca para os Sistemas de Inovação (VINNOVA)[118].

O VISIR oferece aos estudantes a oportunidade de utilizarem recursos experimentais gratuitos 24 horas por dia, 7 dias por semana, sem um aumento significativo no custo por estudante. Com mais tempo dedicado a estas experiências, os estudantes tornam-se verdadeiros "experimentadores", capazes de criar bens e serviços que satisfaçam os requisitos de uma sociedade sustentável. Desta forma, o VISIR não só melhora a aprendizagem, mas também contribui para a formação de profissionais comprometidos com o desenvolvimento sustentável [119].

Em 2018, tal como referido em [120], o VISIR já tinha sido implementado em 8 Instituições de Ensino Superior diferentes, situados em 6 países, incluindo o ISEP (Portugal). No mesmo ano, segundo a mesma publicação, foram instalados novos sistemas VISIR em várias instituições da América do Sul. O software do VISIR é lançado sob uma licença GNU GPL.

O conceito subjacente a este consiste em acrescentar uma opção de operação remota aos laboratórios de ensino tradicionais para os tornar mais acessíveis, independentemente dos alunos estarem no *campus* ou principalmente fora do *campus*[121].

A arquitectura, do VISIR pode ser dividida em quatro partes, tal como referido em [122] e como se pode ver na Figura 2.16 esta arquitectura inclui:

- Servidor de equipamento - Compreende todo o equipamento do VISIR (módulos e *chassis*): as placas *PCI eXtensions for Instrumentation* (PXI) e instrumentação (que estão ligadas à matriz de relés) controlados através do LabVIEW;
- Servidor de medições - Trata-se de um servidor escrito em Visual C++ para a *Microsoft*. É programado por ficheiros "max list" que contêm os valores máximos dos componentes e ajustes dos instrumentos para cada experiência e servem para evitar a concepção de circuitos perigosos e proteger os instrumentos;
- Servidor *web* - Aloja a *interface Web* do VISIR e foi concebida em *Apache* com uma base de dados em *MySQL*;
- Servidor de *interface*, Figura 2.17 - É o *site* do VISIR e foi escrito em PHP, com o cliente de experiências integrado escrito em *Flash*.

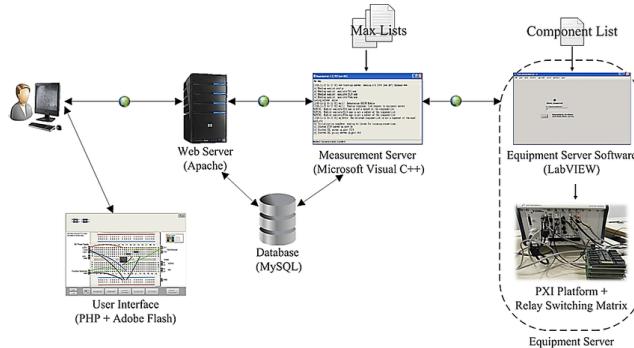


Figura 2.16: Arquitectura VISIR[122]

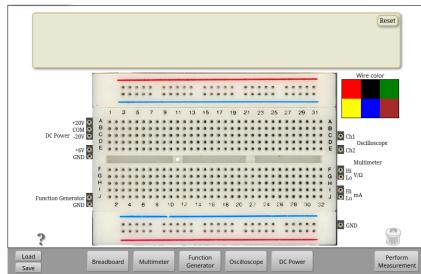


Figura 2.17: Protoboard VISIR

O *chassis* PXI-1033[123], Figura 2.18, é um controlador integrado com 5 *slots* que foi concebida para aplicações de controlo remoto. Obviamente que esta placa exige um *interface* para PC, que, neste caso, é feito através do LabVIEW.



Figura 2.18: Chassis PXI-1033 [123]

Os módulos integrados na PXI-1033 e que compõem o VISIR são:

- Osciloscópio, PXI-5114, 12 MHz, 250 MS/s, 8-Bit, Figura 2.19
- Gerador de sinal, PXI-5402, 10 MHz *Bandwidth*, 1-*Channel*, 14-Bit, Figura 2.20
- Multímetro Digital, 6<sup>1/2</sup>Digitos, ±300 V, *Onboard* 1.8 MS/s Digitalizador isolado, suporte para medições de indutâncias e capacitâncias, Figura 2.21;
- Fonte de tensão programável, 3 canais, corrente de saída máxima de 1 A, gama de tensão de saída analógica: -20 V a 20 V, Figura 2.22.



Figura 2.19: Osciloscópio PXI-5114 [124]



Figura 2.20: Gerador de sinal PXI-5402 [125]



Figura 2.21: Multímetro Digital PXI-4072 [126]



Figura 2.22: Fonte de tensão PXI-4110 [127]

O VISIR possuí uma Matriz de Comutação de Relés (MCR) concebida no BTH especialmente para o uso em experiências electrónicas em LRs, Figura 2.23. Esta matriz é constituída por quatro placas PC/104<sup>25</sup> (de baixo para cima): **NOTA: subentende-se que são placas de controlo - A REVER** fontes de tensão, multímetro digital, osciloscópio e a placa de topo permite configurar os componentes. A matriz é controlada por uma *Programmable Interrupt Controller* (PIC), PIC18F4550<sup>26</sup>, montada na placa de alimentação (fontes de tensão) e comunica com um PC, via USB, e com os controladores das outras placas através das PICs, PIC16F767<sup>27</sup>, via *Inter-Integrated Circuit* ( $I^2C$ ), montadas em cada placa[128].

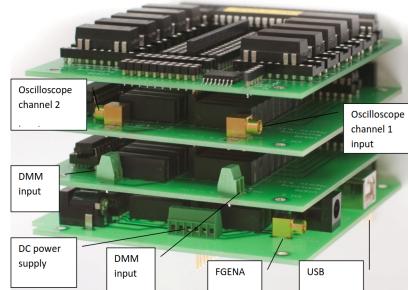


Figura 2.23: Matriz VISIR[128]

Segundo [122] e [129], em 2012, 6 Universidades tinham implementado o VISIR. No entanto, Pereira *et all.*, (2017), referem que o VISIR estava presente em 12 instituições e 7 países[130].

Levanta-se, então, uma questão: Como fazer a integração de todos os LRs de forma a aproveitar os recursos e potenciar o trabalho colaborativo?

<sup>25</sup><https://pc104.org/>

<sup>26</sup>Datasheet: <https://www.microchip.com/en-us/product/PIC18F4550>

<sup>27</sup>Datasheet: <https://www.microchip.com/en-us/product/PIC16F767>

**PILAR** O projecto *Platform Integration of Laboratories based on the Architecture of visiR (PILAR)* Erasmus+, foi iniciado em 2016 tendo terminado em 2019 e tinha como objectivo a criação de uma Federação de cinco nós VISIR existentes, partilhando experiências, capacidade e recursos entre as diversas instituições e permitir o acesso ao LR VISIR, através do consórcio PILAR, a estudantes de outras instituições de ensino[131].

O VISIR *Special Interest Groups* (SIG) é(**FOI - A REVER PROF**) organizado para pessoas interessadas em Engenharia *online* ou remota, especialmente na abertura de laboratórios universitários para acesso remoto 24/7. Este projeto foi lançado com o objetivo de divulgar métodos de abertura de laboratórios para acesso remoto, partilhar ideias, equipamento e material didático, bem como de discutir o desenvolvimento da plataforma VISIR. Outro dos objetivos é(**FOI - A REVER PROF**) a padronização de bancos de trabalho *online* localizados em universidades de todo o mundo, constituindo laboratórios de rede disponíveis para sessões de laboratório para estudantes dentro e fora do *campus*[132].

A missão da Federação VISIR é actualizar e alargar o VISIR SIG, integrando indivíduos e instituições. O objetivo é fornecer um sistema uniforme, em que os estudantes se possam registar e utilizar os laboratórios federados baseados no VISIR e os materiais de aprendizagem de diferentes instituições pertencentes à Federação. Através de um mecanismo comum partilhado, deverá ser possível aceder a cada experiência a partir de um único sistema de gestão da aprendizagem [133].

Os principais objectivos da Federação são[133]:

- Expandir o SIG a instituições;
- Conectar todos os nós VISIR;
- Alargar a gama de aplicações;
- Melhorar a utilização do VISIR;
- Promover o laboratório remoto VISIR;
- Partilhar materiais de aprendizagem;
- Trocar experiências entre os detentores e utilizadores do VISIR.

Como se viu anteriormente, o VISIR funciona numa base de cliente-servidor, o que significa que, se um determinado servidor não estiver disponível, também o laboratório não estará. Neste caso, a Federação permite o redirecccionamento dos pedidos dos clientes para um outro nó VISIR que esteja disponível[134].

## **2.5 COMENTÁRIO PROF**

**Desenvolver mais o PILAR e a Federação?**

**Desenvolver mais o VISIR? Vantagens? Desafios?**



## Capítulo 3

# LaRE - Laboratório Remoto Expansível

*“All we have to decide is what to do with the time that is given us.”*

— Gandalf, *The Lord of the Rings: The Fellowship of the Ring*

Neste capítulo descreve-se o processo que levou à criação do LaRE, a arquitectura de *hardware* e *software*, o motivo das escolhas e a implementação dos circuitos.

### 3.1 Contextualização

Como já foi referido na Secção 1.2 o principal objectivo desta dissertação passa pelo desenvolvimento de um LR para o ensino da electrónica, que visa colmatar (ou resolver) algumas das limitações que afectam o VISIR.

De acordo com a Secção 2.4.1, o *software* do VISIR foi disponibilizado sob licença GNU GPL. No entanto, o *software* de programação do *firmware* do controlador da matriz, fornecido com o próprio sistema, não é livre, o que significa que não pode ser modificado ou atualizado por terceiros. Acresce ainda que o equipamento associado aos módulos e placas de instrumentação é controlado através do LabVIEW, cujas licenças têm custos anuais que variam entre 523€ e 4300€, conforme a versão adquirida [135].

A criação do LR surge, assim, como resposta prática a estas limitações. Numa fase inicial do desenvolvimento, foram realizados testes preliminares de controlo de

relés utilizando um *Arduino* Mega, representado na Figura 3.1, em conjunto com um ambiente de programação simples implementado em LabVIEW.



Figura 3.1: *Arduino* Mega [136]

No entanto, de forma a ultrapassar o problema levantado pelo elevado preço do LabVIEW, surge um segundo objectivo que se prende com a substituição deste *software* por outro que fosse gratuito e *open source*. A eliminação do LabVIEW implicou a implementação de um servidor. Sendo assim, numa primeira abordagem, foram analisadas algumas opções, tais como: *FastApi*<sup>1</sup>, *Django*<sup>2</sup> e *Flask*<sup>3</sup>. Estas opções enquadram-se no que se pode chamar *frameworks* ou *micro-frameworks*. Neste caso são todas desenvolvidas para aplicação em Python.

Uma *micro-framework* é um tipo de *framework* minimalista, que fornece apenas as funcionalidades essenciais para o desenvolvimento de aplicações, sem incluir bibliotecas ou componentes adicionais que não os estritamente necessários. Isso permite a quem desenvolve adicionar apenas as funcionalidades específicas para cada projecto ou aplicação. Daqui resulta um ambiente de desenvolvimento mais leve e flexível [137]. Optou-se, então, pelo *Flask* e as razões da escolha, assim como uma explicação mais detalhada serão apresentadas na Secção 3.6.

É possível combinar o *Arduino* com o Python, mas isso implicaria uma mudança ao nível do *firmware*, uma tarefa que não é trivial [138]. A linguagem nativa do *Arduino* é similar ao *C++* e o *firmware* instalado foi projectado para interpretar e executar código escrito nesse tipo de linguagem. De forma mais rigorosa, o uso do *Python* no *Arduino* ou em qualquer outro microcontrolador, faz-se através de *MicroPython*, uma implementação simples e eficiente do Python que inclui um pequeno subconjunto da bibliotecas padrão e é optimizado para funcionar em microcontroladores e em ambientes limitados [139]. Quer isto dizer que todas as bibliotecas usadas na programação da aplicação ou projecto têm de ser carregadas para a memória dos microcontroladores. No caso do *Arduino* Mega, uma análise ao *datasheet* [140] revela que este tem 256 Kbytes reservados para o envio de programas e 8 Kbytes de memória *SRAM* reservado para variáveis temporárias. Além destes problemas de

<sup>1</sup><https://fastapi.tiangolo.com/>

<sup>2</sup><https://www.djangoproject.com/>

<sup>3</sup><https://flask.palletsprojects.com/en/3.0.x/>

memória e uma vez que é necessário que o *Arduino* funcione como servidor, a versão apresentada na Figura 3.1 não é adequada.

No mercado, existe o ESP32, uma alternativa mais poderosa que o *Arduino* e com placa de rede sem fios integrada, tal como é apresentado na Figura 3.2. No entanto, este microcontrolador sofre dos mesmos problemas de memória que o *Arduino* Mega e da utilização do *MicroPython*. Uma análise ao *datasheet* [141] revela que a memória *flash* varia entre os 4-16 Megabyte. No modelo "ESP32-DEVKITC-32E", que estava disponível para este projecto, o valor é de 4 MB [142]. Estas limitações não permitem a implementação de um servidor minimamente robusto usando o *Arduino* Mega ou o ESP32.

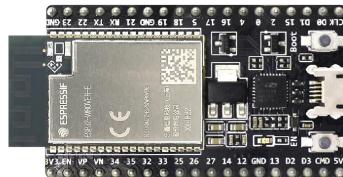


Figura 3.2: *ESP32* [143]

A escolha seguinte recaiu no Raspberry Pi, versão 5<sup>4</sup>, que à data da escrita desta dissertação é a versão mais actual, apresentada na Figura 3.3.



Figura 3.3: *RaspberryPI5* [144]

Considerou-se, portanto, que seria uma mais-valia desenvolver um LR com as seguintes características:

- Python como linguagem principal;
- Raspberry Pi como servidor *Flask*;
- *Interface* com o utilizador desenvolvido em *HyperText Markup Language* (HTML).

---

<sup>4</sup>Doravante, sempre que for referido *Raspberry Pi*, subentende-se a versão 5.

Estava dado o último passo na escolha do *hardware* e do *software* que viriam a compor o LaRE.

### 3.2 Solução proposta

Os objectivos principais foram apresentados na Secção 1.2 e as características gerais do projecto descritas na Secção 3.1. Pretende-se que o LaRE seja um LR capaz de controlar e comandar um conjunto de experiências electrónicas, assim como efectuar medições de várias grandezas eléctricas. Para que a solução proposta seja completa e definitiva, é necessário definir os instrumentos de medida e os circuitos que compõem as experiências.

No contexto desta dissertação, o instrumento de medida adoptado foi o *Virtual-Bench* (VB), modelo VB-8012 que pode ser controlado de duas formas: através do *software* fornecido pela NI ou através do *pyVirtualBench*<sup>5</sup> [145]. O *pyVirtualBench* é um *wrapper*<sup>6</sup> que permite controlar o VB através de uma aplicação Python [146]. No entanto, este *wrapper* não é compatível com *Linux*.

Perante este facto, decidiu-se integrar um PC no LaRE - pormenores da instalação do *driver* no *site* da NI [145] - sendo que todo o peso computacional passaria para o PC e o Raspberry Pi controlaria os relés. Esta decisão justifica-se pela possibilidade de, no futuro, o *pyVirtualBench* vir a ser compatível com sistemas *Linux*.

A outra forma de controlar o VB é usando a aplicação disponibilizada pela NI e compatível com *Windows*. Esta aplicação faz uma apresentação integrada dos cinco instrumentos apresentados no VB, como pode ser visto na Figura 3.4. A aplicação também inclui funcionalidades de fluxo de trabalho, como a importação e exportação de configurações de instrumentos e a captura de dados. Importa salientar que o VB não pode ser controlado simultaneamente por ambas as *interfaces*.

As características do LR são as seguintes:

- Python como linguagem principal;
- PC como servidor *Flask*;
- Raspberry Pi como controlador dos relés;
- *Interface* com o utilizador desenvolvida em HTML.

Definidas - de uma forma geral - as soluções de *hardware* e *software*, agora com a integração de um PC - passou-se ao estudo da arquitectura do LaRE.

---

<sup>5</sup>Apesar de não ser abordado no contexto desta dissertação, existe também a possibilidade de aceder ao VB através do LabVIEW.

<sup>6</sup>Este *wrapper* é um *software* de terceiros, suportado e mantido pela comunidade e não é diretamente suportado pela NI.

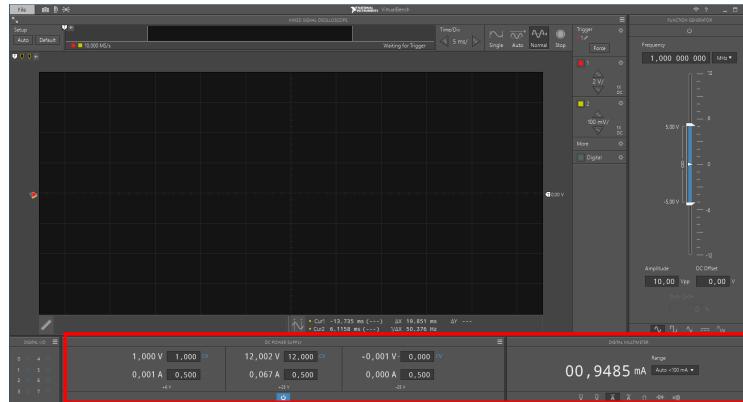


Figura 3.4: Exemplo do VB usado como multímetro digital

### 3.3 Arquitectura

Como já se viu na Secção 3.1, a arquitectura do LaRE proposta baseia-se numa estrutura cliente-servidor, suportada ao nível do *hardware* pelo VB, pelo Raspberry Pi e um PC como servidor. Uma representação geral do que será o LaRE pode ser vista na Figura 3.5.

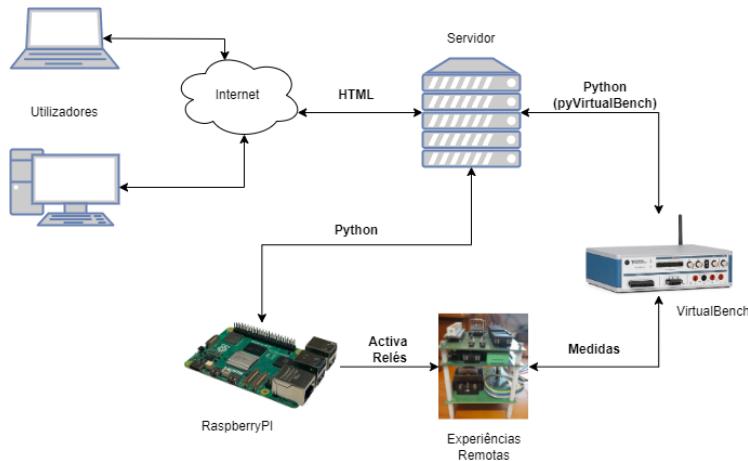


Figura 3.5: Arquitectura LaRE

Antes de mais, importa clarificar as formas de comunicação entre os diversos dispositivos que compõem o LaRE. Tanto o servidor, como o Raspberry Pi e o VB têm disponíveis interfaces de rede sem fios e com fios, assim como *interfaces USB*. A comunicação entre os diversos dispositivos é feita da seguinte forma:

- **Servidor - Raspberry Pi:** comunicação via rede sem fios. Em termos de simplicidade de programação, a comunicação entre o servidor e o Raspberry Pi é feita através da rede sem fios, utilizando *sockets* que, através da sua API, facilitam a comunicação entre processos em rede, independentemente da infra-estrutura utilizada — seja por ligação com fios, sem fios ou através

de redes locais, permitindo, assim, o envio e recepção de mensagens entre dispositivos ligados em rede [147]. O fluxograma de uma chamada ao *sockets* está representado na Figura 3.6;

- **Servidor - VB:** comunicação via rede sem fios ou USB. Entre estes dois dispositivos é, basicamente, indiferente a forma como é feita a comunicação. Isto porque esta parte é gerida pelos *drivers* da aplicação instalada no *Windows*. Independentemente de se utilizar o *pyVirtualBench* ou a aplicação;
- **Controlo de relés e medidas:** O controlo dos relés é feito através da ligação directa do Raspberry Pi ao *driver* dos relés. Por sua vez, as leituras e medições das grandezas eléctricas são efectuadas pelo VB, ligado directamente aos pontos específicos do circuito.

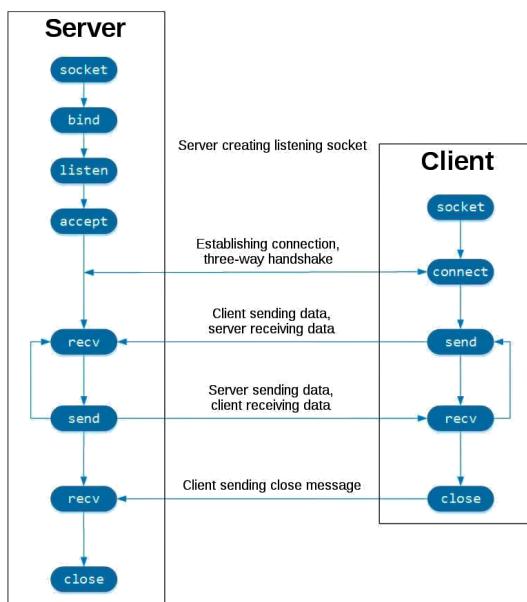


Figura 3.6: Fluxograma de uma chamada ao *socket* [147]

Ao nível do *software*, o servidor será implementado no PC com base no *Flask*, como já foi referido na Secção 3.1 e 3.2, e será descrito com mais pormenor na Secção 3.6. A comunicação entre o servidor e o VB, entre o servidor e o Raspberry Pi e o controlo dos relés é feita em *Python*. A *interface* com o utilizador é feita em *HTML*.

A Figura 3.7 apresenta com mais pormenor a solução implementada no LaRE. De uma forma geral, podemos ver como é realizada a comunicação e a troca de informação entre os diferentes dispositivos de *hardware*.

### 3.4 Circuitos electrónicos - experiências?

A escolha dos circuitos teve como base os que estão disponíveis no VISIR do ISEP e reduzindo os objectivos do LaRE à sua forma mais básica, pode afirmar-se que

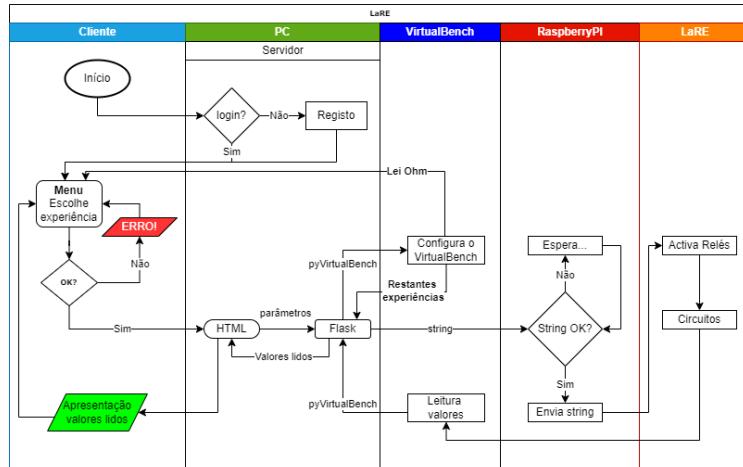


Figura 3.7: Arquitectura LaRE - NOTE: Fig tem de ser revista

se pretende “provar um conceito”. Escolhendo experiências com circuitos integrados ou transístores, poderia tornar a implementação desnecessariamente mais complexa e, por conseguinte, com menos experiências. Portanto, o objectivo passará por criar um LR com experiências típicas para a introdução à electrónica e que permitam uma aprendizagem gradual em contexto de sala de aula. E, assim, “provar o conceito”.

Sendo assim, os circuitos que compõem o LaRE e satisfazem os critérios definidos em cima são:

- Lei de Ohm;
- Rectificador de meia onda;
- Rectificador de onda completa;
- Filtro RC passa-baixo;
- Filtro RC passa-alto.

Doravante, considera-se que cada experiência é composta por dois circuitos distintos, com excepção da experiência dedicada à Lei de *Ohm*. Por exemplo, a experiência sobre rectificação inclui os circuitos de meia onda e de onda completa; já a experiência dos filtros abrange os circuitos passa-baixo e passa-alto.

### 3.4.1 Lei de *Ohm*

Na forma da Equação 3.1, a Lei de *Ohm* estabelece que a resistência eléctrica ( $R$ ) de um condutor é determinada pelo quociente entre a tensão elétrica ( $U$ ) aplicada aos seus terminais e a corrente eléctrica ( $I$ ) que o atravessa.

$$R = \frac{U}{I} \quad (3.1)$$

Isto significa que a Equação 3.1 define uma relação linear entre a tensão e a corrente para uma determinada resistência de valor fixo, o que implica que o gráfico obtido, representado na Figura 3.8, é uma linha recta que passa pela origem. O declive dessa recta representa o valor da resistência elétrica ( $R$ ).

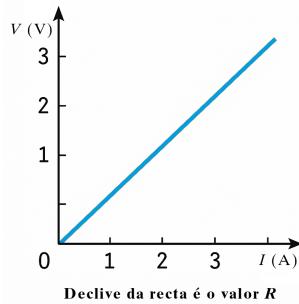


Figura 3.8: Gráfico da Lei de *Ohm*

Da forma como foi implementado, há duas possibilidades de realizar este estudo, exemplificado nas Figuras 3.9a e 3.9b. O tutor ou professor, poderá optar por apresentar o conceito de duas formas distintas, em ambos os casos são efectuadas cinco medições, construído o respectivo gráfico e calculado o valor do declive da recta analiticamente. A diferença está na forma como se confrontam os resultados práticos. No entanto, no contexto desta dissertação foi implementado o caso descrito na Figura 3.9a<sup>7</sup>.

**LaRE**  
LABORATÓRIO REMOTO EXPANSÍVEL

Menu Principal

- Lei de Ohm
- Rectificador de meia onda
- Rectificador de onda completa
- Filtro Passa-Alto

Escolha uma das três resistências

Clique OK para iniciar as medições, STOP para interromper

OK STOP/RESET

Selecionar Vcc:	Selecionar R:
Selecionar Vcc:	Selecionar R:
1	1
2	1.1
3	1.5
4	
5	

(a) Opção 1

Escolha uma das três resistências

Clique OK para iniciar as medições, STOP para interromper

OK STOP/RESET

Selecionar Vcc:	Selecionar R:
Selecionar Vcc:	Selecionar R:
1	Ra
2	Rb
3	Rc
4	
5	

(b) Opção 2

Figura 3.9: Experiência Lei de *Ohm*

No caso em que a resistência é dada, o declive pode ser calculado e confrontado com as soluções, sendo que no caso em que é desconhecida, o declive da recta é calculado e a partir deste o resultado infere-se o valor da resistência.

<sup>7</sup>Como trabalho futuro, o menu poderia ser desenvolvido para o utilizador, aluno ou professor escolher a opção pretendida.

### 3.4.2 Rectificadores

Nas experiências dos rectificadores pretende-se estudar e avaliar a diferença entre os dois tipos de rectificação, consoante as quatro combinações possíveis dos pares resistência/condensador e a influência que têm na variação da tensão de *ripple*. No caso da rectificação de meia onda, é ainda possível variar a frequência entre os 50 Hz e 2000 Hz - **NOTA: Estes limites podem ter que ser ajustados** mas no caso da rectificação de onda completa, a frequência é fixa ao valor da rede eléctrica - 60 Hz.

A Figura 3.10 apresenta o diagrama de blocos simplificado do processo de rectificação, que pode ser usado, por exemplo, para o projecto duma fonte de alimentação em contexto de sala de aula. Este é composto por quatro blocos principais:

- Bloco 1 - Transformador: reduz a tensão proveniente da rede eléctrica para valores mais baixos e adequada ao nível de tensão alternada que se deseja;
- Bloco 2 - Rectificação: realiza a conversão da tensão alternada CA em tensão contínua unidirecional, ainda com ondulação;
- Bloco 3 - Filtragem: responsável pela suavização da forma de onda, reduzindo a componente alternada (*ripple*);
- Bloco 4 - Estabilização: regula e estabiliza a tensão de saída, garantindo um valor contínuo e constante.

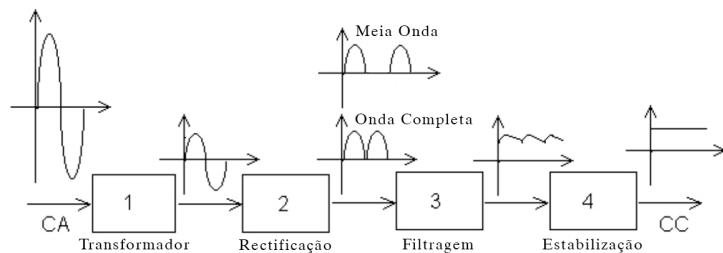


Figura 3.10: Diagrama de blocos da rectificação [meia onda e onda completa]

A tensão de *ripple* é uma componente variável no tempo que surge à saída da filtragem, Bloco 3, como representado na Figura 3.10. Esta componente deve ser minimizada para estabilizar a tensão de saída em corrente contínua, Bloco 4, representado na mesma figura [148]. A Figura 3.11 representa a forma de onda à saída do Bloco 3 - Filtragem - quando se utiliza a rectificação de meia onda.

A tensão de *ripple* pode ser calculada através da Equação 3.2:

$$U_r = U_P \left(1 - e^{-\frac{T}{RC}}\right) \quad (3.2)$$

Observa-se, a partir da Equação 3.2, que quando a constante de tempo  $RC \gg T$ , o cálculo da tensão de *ripple* pode ser reduzido à Equação 3.3.

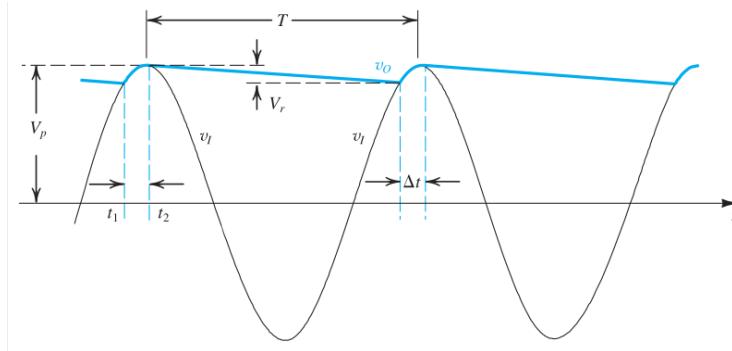


Figura 3.11: Forma de onda à saída do Bloco 3 - Filtragem [meia onda] [148]

$$U_r = \frac{U_P}{fRC} \quad (3.3)$$

Sendo que, neste caso, a tensão de *ripple* é baixa e  $v_o$ , ou seja, a tensão à saída do filtro é praticamente constante e dada pela Equação 3.4:

$$u_o = U_P - \frac{1}{2}U_r \quad (3.4)$$

No entanto, pode tornar-se este circuito mais eficiente se se usar a rectificação de onda completa e como pode ser visto na Figura 3.12, a frequência do *ripple* é o dobro da frequência da onda de entrada.

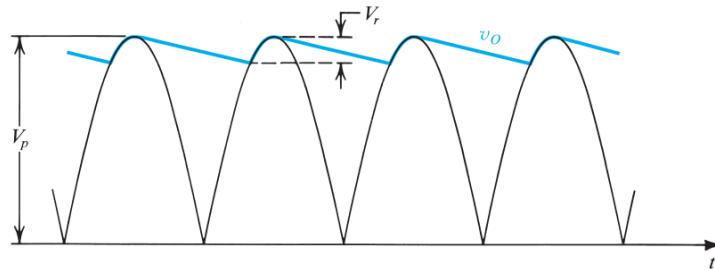


Figura 3.12: Forma de onda à saída do Bloco 3 - Filtragem [onda completa] [148]

Sendo assim, o valor da tensão de *ripple*, neste caso, será dado pela Equação 3.5:

$$U_r = \frac{U_P}{2fRC} \quad (3.5)$$

São estas formas de onda, representadas pelas Figuras 3.11 e 3.12, que se pretendem estudar e avaliar, assim como os valores dados pelas Equações 3.3 e 3.5. (A Equação 3.2 deve ser usada quando os valores dos condensadores e resistências são pequenos ou frequência é baixa, isto é, quando o *ripple* não é desprezável.)

### 3.4.3 Filtros

Representados na Figura 3.13 estão os filtros simplificados, leccionados em contexto de sala de aula no ensino secundário.

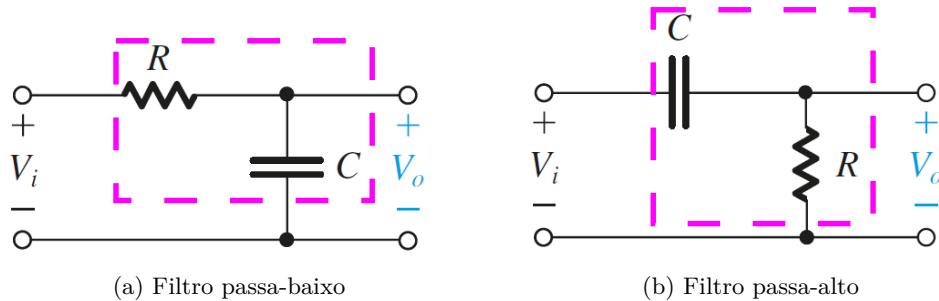


Figura 3.13: Esquemas dos filtros [148]

A possibilidade de variar a frequência do sinal de entrada permite que estas experiências sejam utilizadas para estudar a resposta em frequência dos filtros, analisar o Diagrama de *Bode*, determinar a frequência de corte dada pela Equação 3.6 e ainda relacionar, por exemplo, o valor da tensão de entrada com a tensão de saída.

$$f_c = \frac{1}{2\pi RC} \quad (3.6)$$

As Figuras 3.14a e 3.14b apresentam os Diagramas de *Bode* [ideais], que servem de referência para o estudo desta experiência.

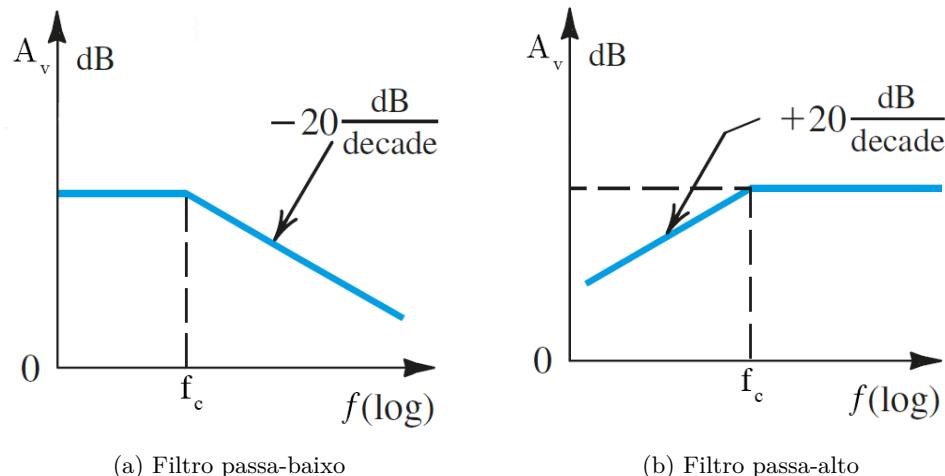


Figura 3.14: Diagramas de *Bode* [ideal] [148]

Para além da análise em frequência, um outro aspecto complementar a considerar é a relação entre as ondas de entrada e saída, em função da frequência. Nos filtros, o intervalo de frequências permitido é igual ao já referido para os rectificadores: entre 50 Hz a 80 000 Hz.

A frequência de corte de um filtro é definida como o ponto onde o ganho do filtro sofre uma atenuação de 3 dB em relação ao seu valor máximo. Para um filtro passa-baixo (passa-alto) ideal, o ganho em baixas (altas) frequências é unitário. Quando o sinal atinge a frequência de corte, a relação entre a tensão de saída e a tensão de entrada reduz-se para:

$$\frac{U_{out}}{U_{in}} = \frac{1}{\sqrt{2}} \approx 0.707 \quad (3.7)$$

... ou na escala logarítmica:

$$\frac{U_{out}}{U_{in}} = 20 \log_{10}(0.707) \approx -3 \text{ dB} \quad (3.8)$$

Assim, a frequência de corte de um filtro corresponde ao ponto em que a amplitude do sinal de saída é aproximadamente 70.7% da amplitude do sinal de entrada, o que corresponde a -3 dB.

## 3.5 Hardware

### 3.5.1 VirtualBench

Este dispositivo desenvolvido pela NI integra vários instrumentos e ferramentas de teste, tais como um osciloscópio digital com análise de protocolo, um gerador de formas de onda, um multímetro digital, uma fonte de alimentação Corrente Contínua (CC) programável e E/S digitais num único dispositivo que se liga a um PC, via USB ou rede sem fios, como se pode ver na Figura 3.15. As principais características deste modelo estão descritas na Figura 3.16 [149]. A forma como pode ser controlado e configurado já foi abordada na Secção 3.2.



Figura 3.15: Painel traseiro *VirtualBench* VB-8012 [149].

### 3.5.2 Matriz LaRE

Definidas as soluções de *hardware* e *software*, bem como as experiências a implementar, iniciou-se o projecto da matriz. Tal como anteriormente, e tendo o VISIR como referência, decidiu-se desenvolver uma matriz cujas placas respeitem as dimensões



Figura 3.16: Painel frontal *VirtualBench* VB-8012 [149].

definidas pelo consórcio PC/104 [150], o qual estabelece um padrão caracterizado pelo seu formato compacto e modular.

Os esquemas finais das experiências encontram-se no Anexo A, onde se pode verificar que o número de relés necessários varia entre 8 — no caso da Lei de *Ohm* — e 13 — nos casos dos rectificadores e dos filtros. Sendo assim, com o objectivo de facilitar a implementação, os testes, a detecção de erros e, mais tarde, a manutenção, optou-se por dividir o LaRE em três placas. O desenho dos esquemas e dos *Printed Circuit Board* (PCB) foram elaborados com recurso ao KiCad. A Figura 3.17 apresenta uma visão geral da matriz do LaRE.

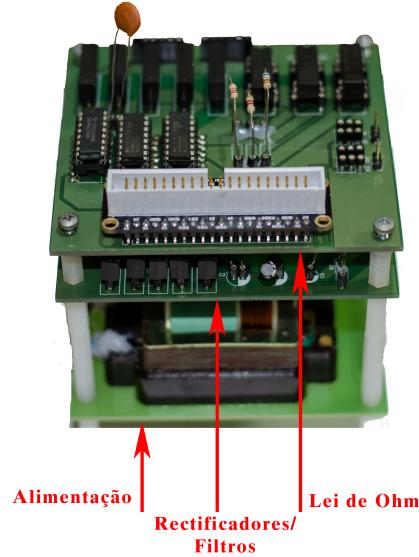


Figura 3.17: Matriz do LaRE

### **Lei de *Ohm***

A placa desenvolvida para esta experiência está representada na Figura 3.18:



Figura 3.18: [Placa] Lei de *Ohm*

1. Ligação ao Raspberry Pi através de um conector *Insulation Displacement Connector* (IDC) de 40 pinos (1);
2. Relés (2);
3. Ligações aos aparelhos de medida (3);
4. *Drivers* de relés e registo de deslocamento (4);
5. Resistências em estudo (5);
6. Ligação à placa seguinte através de conectores “Arduino stackable” (6).

### Rectificadores/Filtros

A Figura 3.19 apresenta a placa da experiência referente aos rectificadores/filtros:

1. Rectificação;
  - (a) Díodo rectificador - meia onda (1<sub>a</sub>) e Ponte rectificadora - onda completa (1<sub>b</sub>);
2. Relés (2);
3. Resistências - [componentes comuns a ambas as experiências](3);
4. *Drivers* de relés e registo de deslocamento (4);
5. Condensadores [componentes comuns a ambas as experiências](5);
6. Ligação à placa seguinte através de conectores “Arduino stackable” (6);
7. Resistência e condensador exclusivo dos filtros (7).

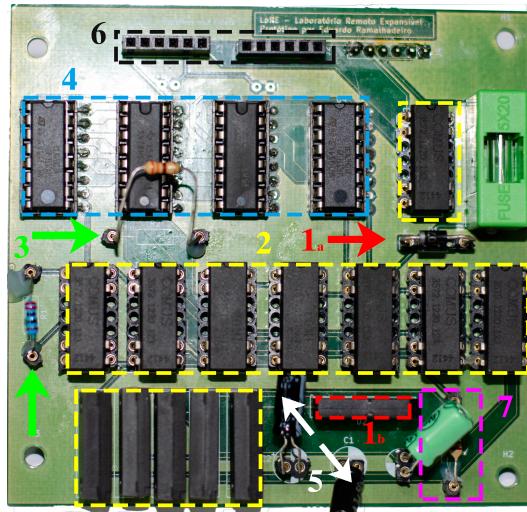


Figura 3.19: [Placa] rectificadores e filtros

### Fontes de alimentação

A placa referente às fontes de alimentação está representada na Figura 3.20. Esta placa é constituída por um transformador e uma fonte de 5 V. O esquema completo está disponível no Anexo A.



Figura 3.20: [Placa] fontes de alimentação

### 3.5.3 Raspberry Pi

Como já foi referido na Secção 3.1, a ideia inicial passava por utilizar o Raspberry Pi como servidor e como controlador dos relés, no fundo desempenhando as funções de RLMS. Este dispositivo insere-se numa gama de pequenos computadores de placa única SBC, acessíveis e versáteis, como o *Banana Pi*, *BeagleBone Black*, *Orange Pi* ou *Odroid*, amplamente utilizados em ensino, prototipagem e aplicações de IoT.

Em contexto de laboratório estavam disponíveis os modelos PI2 e PI3, algo desactualizados. O PI3 já se mostrou lento nos primeiros testes e, por isso, optou-se pela última versão do Raspberry Pi, que à data da escrita deste documento é a 5. Esta versão traz várias melhorias e novos recursos, sendo que as principais diferenças

de *hardware* para a versão 4B estão representadas na Tabela 3.1. No entanto, estas melhorias acarretam um maior consumo de energia e, por isso, é recomendado o uso de arrefecimento activo [151]. Na Figura 3.21 está representado o Raspberry Pi usado no LaRE.



Figura 3.21: *Raspberry PI5* com dissipador activo utilizado no LaRE

Tabela 3.1: *Raspberry PI4 vs Raspberry PI5 - principais diferenças*  
[152]

Raspberry PI4	Raspberry PI5
1.8 GHz	2.4 GHz
VideoCore VI @ 500 MHz, Vulkan 1.0	VideoCore VII @ 800 MHz, Vulkan 1.2
LPDDR4-3200 SDRAM até 8 GB	LPDDR4X-4267 SDRAM 4 GB/8 GHz
5 V/3 A via USB-C (15 W)	5 V/5 A via USB-C (27 W)

Comum às versões anteriores, o Raspberry Pi possui um conector de 40 pinos denominados *General Purpose Input/Output* (GPIO)s, como se pode ver na Figura 3.22. Os GPIOs são pinos versáteis e configuráveis e permitem que a placa interaja com uma variedade de componentes eletrónicos e outros dispositivos, como por exemplo, relés, sensores, motores, etc.. Os GPIOs são pinos digitais, isto é, o Raspberry Pi não possui um *Analog-to-Digital Converter* (ADC) interno para ler sinais analógicos. No entanto, pode usar-se um ADC externo, como por exemplo, o MCP3008 ou o módulo ADS1115. Quer isto dizer que só trabalham com duas tensões: 3.3 V ou 0 V, correspondendo aos valores lógicos “1” ou “0”, respectivamente. Dependendo da configuração dos GPIOs - entrada ou saída - estes podem ler ou fornecer as tensões correspondentes aos níveis lógicos desejados.

A corrente no Raspberry Pi pode ser configurada para valores entre 2 mA e 16 mA (em intervalos de 2 mA) por cada GPIO. No entanto, quando vários GPIOs são utilizados em simultâneo, o valor total combinado da corrente não deve exceder os 50 mA. Se for fornecida uma tensão superior a 3.3 V aos GPIOs, o Raspberry Pi corre o risco de se danificar irremediavelmente. [151]. Na Figura 3.23 está representada uma forma alternativa de visualizar os GPIOs, mais intuitiva e que permite perceber

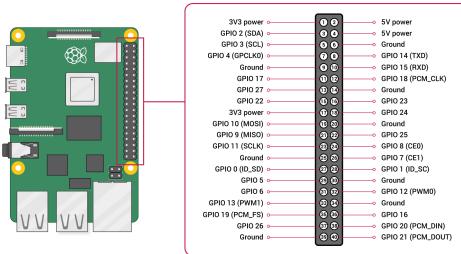


Figura 3.22: GPIOs RaspberryPI [151]

melhor a função dos pinos. Representados a cor amarela estão os GPIOs que podem ser configurados como entrada ou saída; a vermelho, laranja e preto as alimentações de 5 V, 3.3 V e *ground*, respectivamente. Por fim, os GPIOs representados pela cor branca, GPIO0 e GPIO1, correspondentes aos pinos físicos 27 e 28, respectivamente, estão reservados para utilização avançada, nomeadamente, a comunicação  $I^2C$  com a EEPROM de identificação presente em HATs compatíveis. Esta EEPROM permite a configuração automática do sistema aquando do arranque.

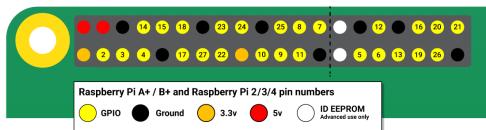


Figura 3.23: Representação alternativa dos GPIOs [151]

### 3.5.4 Relés

Como já foi referido na Capítulo 2, os circuitos que compõem o VISIR são controlados através de relés. Estes dispositivos funcionam como interruptores electromecânicos simples que utilizam um sinal elétrico para controlar um eletroíman. Os relés electromecânicos são constituídos por uma bobine e um contacto, que pode ser normalmente aberto ou fechado. A corrente eléctrica ao passar pela bobine gera um campo magnético que atrai o contacto, fechando-o ou abrindo-o, conforme o caso. A Figura 3.24 mostra um esquema simplificado de um relé electromecânico, onde se pode observar a inclusão do diodo de “roda livre”, comum em muitos circuitos de comando com relés. Este diodo é utilizado para proteger o circuito de comando de picos de tensão induzidos pela bobine do relé, quando a corrente é desligada, segundo a Lei de Faraday. O diodo permite que a corrente induzida pela bobine circule em circuito fechado, evitando danos nos componentes de comando.

Os relés utilizados no LaRE estavam disponíveis em contexto laboratorial e são em tudo idênticos aos que se encontram instalados no VISIR do ISEP. Foram usados



Figura 3.24: Esquema simplificado de um relé [153]

dois tipos de relés: simples (*Single Pole Single Throw* (SPST)) e duplos (*Double Pole Single Throw* (DPST)), como ilustrado na Figura 3.25. Ambos os modelos são da marca *Comus* - o relé SPST, ref. 3570-1331-123 e o relé DPST, ref. 3572-1220-123. As características principais encontram-se descritas no Anexo A e as completas nos respectivos *datasheet* [153].

Nos códigos de referência dos relés, os dois primeiros quartetos indicam a série, enquanto os três dígitos indicam a tensão nominal da bobine e a presença, ou não, do diodo de “roda livre”. No caso dos modelos utilizados, esta tensão é de 12 V e o diodo está ligado entre os pinos 2(+) e 6(-) [153]. Sempre que possível, tentou-se utilizar os relés SPST no comando das fontes e aparelhos de medida e os relés DPST no controlo dos componentes. Serão devidamente referidos os casos em que isso não foi possível, por indisponibilidade dos componentes.

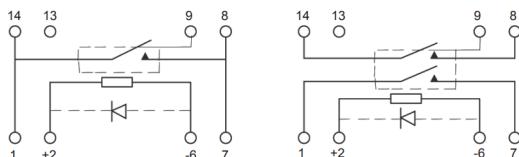


Figura 3.25: Relés SPST e DPST [153]

Como já foi referido na Secção 3.5.3, a tensão de funcionamento dos GPIOs, quer estejam configurados como entrada ou saída, é de 3.3 V, sendo que a máxima corrente por cada GPIO é de 16 mA. Com estes valores, o Raspberry Pi não tem capacidade para comandar os relés, já que funcionam a 12 V. Para activar os relés, é necessário o uso de *drivers*<sup>8</sup> que consigam fornecer a tensão e corrente necessárias para o efeito.

### 3.5.5 Driver de Relés

O circuito integrado ULN2003A, é um *driver* muito usado para controlar relés. Além disso estava disponível em contexto laboratorial. Tipicamente, é usado em conjunto

<sup>8</sup>Doravante, sempre que for referido o termo *driver* no contexto de relés, subentende-se o circuito integrado *ULN2003A*.

com microcontroladores ou Raspberry Pi<sup>9</sup> no comando de cargas indutivas, como motores, bobinas e relés. Estes *drivers* possuem sete pares de transístores NPN, em configuração *Darlington*, que apresentam saídas de alta tensão com díodos *clamp* de cátodo comum para comutação de cargas indutivas [154], como se representa esquematicamente na Figura 3.26.

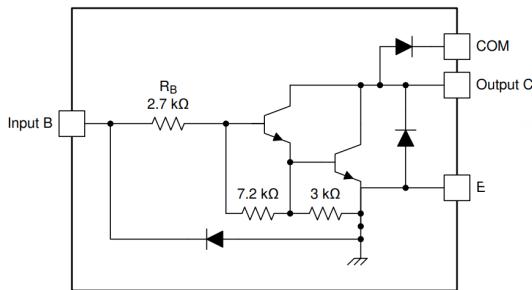


Figura 3.26: Diagrama de blocos ULN2003A [154]

A corrente de colector de saída (única) é de 500 mA e a corrente de entrada, para uma tensão de entrada de 3.85 V, é 0.93 mA [154].

Pela análise dos esquemas, representados no Anexo A e já referenciado na Secção 3.5.2, verifica-se que os GPIOs disponíveis no Raspberry Pi - 26 no total - não são suficientes para comandar os relés do LaRE. Ao todo são utilizados 21 relés, o que corresponderia a 21 saídas, faltando adicionar 10 GPIOs necessários para o controlo da transmissão das *strings* de *bits*. Sendo assim, houve a necessidade de criar uma solução que permitisse comandar os relés, já que o Raspberry Pi não possui GPIOs suficientes.

### 3.5.6 Registo de deslocamento

O uso de registos de deslocamento foi a solução encontrada de forma a ultrapassar o problema da falta de GPIOs. O *SN74HC595*<sup>10</sup> é um circuito integrado comum e bastante utilizado que contém um registo de deslocamento de 8 *bits* com saídas *3-State*, do tipo *Serial Input Parallel Output* (SIPO), que alimenta um registo de armazenamento do tipo D, também de 8 *bits* e com saídas paralelas *3-State*. O sinal de relógio para o registo de deslocamento e armazenamento é independente. A potência consumida é muito baixa, assim como a corrente de entrada [155].

Descrição dos pinos de controlo:

<sup>9</sup>Importa distinguir entre SBCs, como o Raspberry Pi e microcontroladores, como o Arduino. Enquanto os SBCs funcionam como computadores completos capazes de executar sistemas operativos e aplicações complexas, os microcontroladores são dispositivos mais simples, ideais para tarefas de controlo directo, operando sem sistema operativo e com recursos significativamente mais limitados.

<sup>10</sup>Doravante, sempre que for referido registo de deslocamento, subentende-se o circuito integrado *SN74HC595*.

- *Serial Input* (SER): é utilizado para enviar os *bits* para o registo de deslocamento, um *bit* de cada vez;
- *Shift Register Clock* (SRCLK): é o relógio do registo de deslocamento e é ativado no flanco ascendente, por cada impulso dado um *bit* é enviado para o registo de deslocamento;
- *Register Clock* (RCLK): é o relógio do registo de armazenamento e é activo no flanco ascendente. Quando activo, o conteúdo do registo de deslocamento é transferido para o registo de armazenamento, que eventualmente aparece na saída;
- *Shift Register Clear* (SRCLR): permite fazer o *clear* ou *reset* ao registo de deslocamento. Isto equivale a colocar todos os *bits* a zero. Este pino é activo baixo.
- *Output Enable* (OE): pino activo baixo que controla o estado da saída do registo: Se a zero as saídas estão activas, se a um as saídas estão desactivadas

Como referido na Secção 3.5.2, a experiência da Lei de *Ohm* requer 8 relés, enquanto as experiências de rectificadores e filtros necessitam de 13. Cada relé é acionado por um *bit* de um registo de deslocamento, sendo que cada registo permite o controlo de até 8 relés. Assim, é necessário um registo para a experiência da Lei de *Ohm* e dois registos para as experiências de rectificação e filtragem. Deste modo, é enviada uma *string* de 8 ou 13 *bits*, conforme a experiência a controlar - os *bits* são enviados um-a-um, armazenados no registo e depois enviados para a saída. Dado que o número de *bits* varia entre experiências e estas se encontram distribuídas por duas placas distintas (conforme descrito na Secção 3.5.2), optou-se por utilizar pinos diferentes do Raspberry Pi para o envio das *strings*. No Anexo A, Figura 11 e Figura 12, estão definidos os pinos usados na configuração e envio das *strings*. A Figura 3.27 apresenta o diagrama de blocos do registo de deslocamento, enquanto a Tabela 3.2 descreve os seus modos de funcionamento. Como se pode observar, o controlo do envio de uma *string* de dados requer 5 pinos GPIO configurados como saídas. Deste modo, para comandar os relés das experiências distribuídas pelas duas placas, são necessários 10 pinos GPIO no total.

Importa notar que seria tecnicamente possível reduzir este número para apenas 6 pinos, utilizando os mesmos sinais de controlo para ambos os registos de deslocamento e activando-os seletivamente através da linha *OE* - *Output Enable* de cada registo. No entanto, dado que a implementação e os testes dos circuitos foram realizados de forma faseada, optou-se por manter conjuntos de pinos independentes para cada registo de deslocamento, facilitando a deteção e resolução de eventuais erros. As implicações ao nível do *software* de programação são mínimas e ao nível do *hardware* estão dentro dos limites de número de GPIOs do Raspberry Pi.

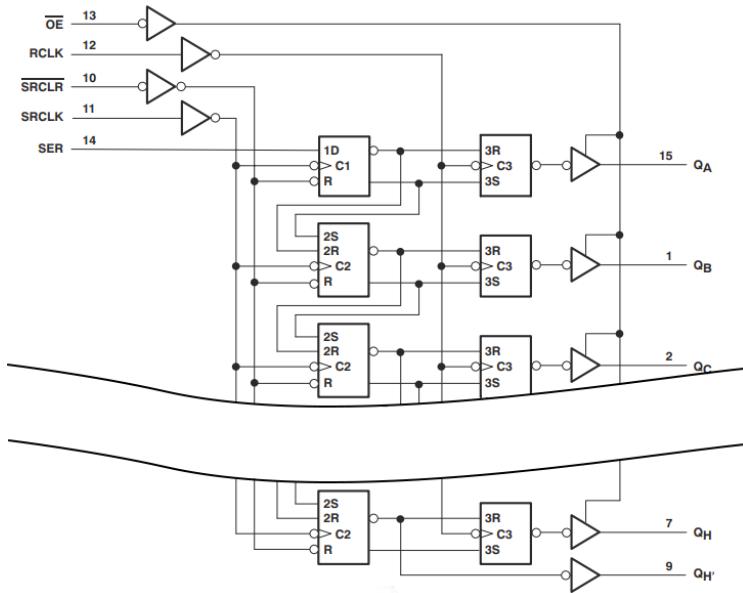


Figura 3.27: Diagrama [em corte] de blocos SN74HC595 [155]

Tabela 3.2: Modos de funcionamento do SN74HC595 [155]

Entradas					Função
SER	SRCLK	$\overline{SRCLR}$	RCLK	$\overline{OE}$	
X	X	X	X	H	Saídas $Q_A$ - $Q_H$ estão desabilitadas.
X	X	X	X	L	Saídas $Q_A$ - $Q_H$ estão habilitadas.
X	X	L	X	X	Registo de deslocamento é limpo.
L	↑	H	X	X	Primeiro passo do registo de deslocamento vai a "0". Passo seguinte armazena os dados do estado anterior, respectivamente.
H	↑	H	X	X	Primeiro passo do registo de deslocamento vai a "1". Passo seguinte armazena os dados do estado anterior, respectivamente.
X	X	X	↑	X	Os dados do registo de deslocamento são armazenados no registo.

## 3.6 Software

O *software* do LaRE foi dividido em duas grandes partes: *Back-end* e *Front-end*.

Por *Back-end* entende-se toda a programação e os processos que correm em segundo plano w que sustentam o funcionamento do LaRE, incluindo o servidor, as APIs ou base de dados [156]. Existem diversas linguagens de programação, *frameworks* e ferramentas para gerir esta parte do sistema. Na implementação do LaRE, foi utilizada a linguagem *Python*, em conjunto com a *Framework Flask*.

O *Front-end*, por sua vez, gera a *interface* do *site*, ou seja, a camada com que o utilizador interage directamente. As linguagens utilizadas na sua implementação foram o HTML, *Cascading Style Sheets* (CSS) e JavaScript. Enquanto o HTML constitui a estrutura base da página, o CSS define o estilo visual dos elementos o JavaScript permite a criação de comportamentos dinâmicos e interactivos [156].

De forma resumida, o desenvolvimento de *Front-end* corresponde ao lado do cliente (aspecto da página Web), enquanto o *Back-end* diz respeito ao lado do servidor (funcionamento da página Web).

### 3.6.1 Back-End

#### *Python*

O *Python* é uma linguagem poderosa e fácil de aprender. Possui estruturas de dados de alto nível eficientes e uma abordagem simples, mas eficaz à programação orientada para objectos [4]. Apresenta, também, uma série de vantagens que se enquadram nos objectivos do desenvolvimento e implementação do LaRE [157]:

- Curva suave de aprendizagem;
- Quantidade e variedade das bibliotecas;
- Portabilidade;
- Flexibilidade;
- Robustez;
- Suporte da comunidade.

Como desvantagens há a referir que, comparado com outras linguagens, o *Python* é mais lento em termos de execução, já que é um tipo de linguagem de alto-nível, pelo que não é adaptado para aplicações móveis e consome mais recursos [157] [158]. Ainda assim, segundo a *IEEE Spectrum* [159], a linguagem *Python* foi considerada a mais popular em 2023.

#### *Flask*

O *Flask* é uma *framework* leve e flexível para *Python* que segue a filosofia *UNIX* de “fazer uma coisa bem feita”. A escolha do *Flask* deveu-se, essencialmente, à facilidade de integração com o *Python*, sendo uma das *frameworks* mais populares em *Python* [137]. É uma *framework* de aplicações *Web Server Gateway Interface* (WSGI) que descreve a forma como um servidor *Web* comunica com aplicações *Web* e como essas aplicações podem ser encadeadas para processar um pedido [160]. Depende, ainda do *Jinja*, que é um motor de criação de modelos que permite escrever código semelhante à sintaxe do *Python*, de forma a renderizar o documento final [161].

Tal como foi referido na Secção 3.1, foram ainda analisadas várias opções, incluindo a *Django*, outra *framework* bastante popular. Em [162] e [163], por exemplo, é feita uma comparação exaustiva entre as duas *frameworks*, não havendo uma decisão final sobre qual a melhor, mas sim sobre qual a que melhor se adapta às necessidades de implementação e desenvolvimento dos projectos. No entanto, o *Flask* foi

concebido para tornar a iniciação rápida e fácil, com a capacidade de evoluir até aplicações mais complexas, sendo mais adaptado a pequenos projectos.

Pelas razões referidas anteriormente, e tendo em conta a análise feita aos prós e contras, considerou-se que tanto o *Python* como o *Flask* são as linguagens que melhor se enquadram e adaptam aos objectivos propostos para o LaRE.

### *Jinja2*

O *Jinja2*<sup>11</sup> é um motor de *templates* para aplicações *Web* escrito em *Python*, amplamente utilizado em conjunto com a *framework Flask*. O seu principal objetivo é permitir a separação entre a lógica da aplicação e a apresentação visual, promovendo um desenvolvimento mais organizado, modular e sustentável. Inspirado na sintaxe do próprio *Python*, o *Jinja* permite incorporar estruturas de controlo, como ciclos e condições, diretamente nos ficheiros HTML através de marcação específica, facilitando a criação dinâmica de conteúdo [161]. Na Listagem 3.1 pode ver-se um exemplo de código *Jinja* incluído na página *base.html*.

---

```

1   <title>{% block title %}Home{% endblock %}</title>
2   {% if user.is_authenticated %}
3       <a class="nav-item nav-link" id="home" href="/">
4           Home</a>
5   {% else %}
6       <a class="nav-item nav-link" id="login" href="/
7           login">Login</a>
8   {% endif %}
9
10  {% with messages = get_flashed_messages(with_categories
11      =true) %}
12      {% for category, message in messages %}
13          {{ message }}
14      {% endfor %}
15  {% endwith %}

```

---

Listagem 3.1: Exemplo *Jinja* incluído na página *base.html*

O mecanismo de *renderização* disponibilizado pelo *Flask*, através da função *render\_template()*, tira partido do *Jinja* para gerar páginas HTML com base em *templates* pré-definidos e dados fornecidos pela aplicação. Isto permite, por exemplo, apresentar resultados de medições, estado dos relés ou mensagens personalizadas, de forma automatizada e responsiva. O *Jinja2* permite gerar HTML dinâmico através da substituição de variáveis (`{{...}}`) e da execução de instruções lógicas (% ... %)

---

<sup>11</sup>O nome *Jinja2* refere-se à versão actual do motor de *templates* *Jinja*, resultante de uma reescrita significativa da sua versão original. Na documentação e na comunidade, é comum referir-se a este motor simplesmente como *Jinja* [161]. Assim, e sem perda de generalidade, será adoptada ao longo deste trabalho a designação *Jinja*.

no lado do servidor, antes de o conteúdo ser enviado ao navegador. Como resultado, o utilizador final visualiza apenas o HTML já processado, sem qualquer traço do código *Jinja*. Esta abordagem estabelece uma ligação simples e segura entre a lógica implementada em *Python* no *back-end* e a apresentação em HTML no *front-end* [161].

### 3.6.2 Front-End

#### *Webpage*

Como já foi referido na Secção 3.6, o *Front-end* diz respeito ao aspecto gráfico das páginas *Web* e pretende-se que a prioridade na construção da página seja a simplicidade. A escolha do desenvolvimento da página recaiu no HTML, CSS e *JavaScript*.

O HTML é a linguagem *standard*, usada para definir a estrutura do seu conteúdo e consiste numa série de elementos usados para delimitar ou agrupar diferentes partes do conteúdo. As *tags* ou etiquetas, podem transformar uma palavra ou imagem num *hyperlink*, pôr palavras em itálico, aumentar ou diminuir a fonte, etc. [164]

Existem 3 tipos de elementos [165]:

- **Normais:** são elementos que têm uma etiqueta de abertura e outra de fecho. Por exemplo, `<p></p>` define um parágrafo;
- **Texto:** são elementos que não têm uma etiqueta de fecho. Por exemplo, `<Title>` especifica o título da página em texto simples;
- **Vazios:** têm somente uma etiqueta de início na forma de `<tag>`. Por exemplo, a etiqueta `<img>` referencia um ficheiro externo, neste caso uma imagem ou `<br>` que força uma quebra de linha.

Na Figura 3.28 está representada a estrutura de uma página HTML. De referir os navegadores não exibem as etiquetas HTML. O seu propósito é ler os documentos HTML e apresentá-los corretamente.

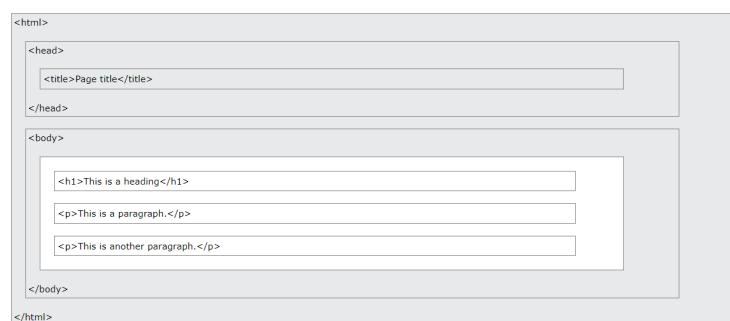


Figura 3.28: Estrutura de uma página HTML [164]

Segundo o consórcio W3C, as CSS são mecanismos fundamentais para a separação de preocupações no desenvolvimento *web*. Ao delegar a responsabilidade de formatação visual no CSS, o HTML concentra-se na estrutura semântica do conteúdo. Isso resulta num código mais limpo, de fácil manutenção e acessível. As regras de estilo, que especificam como os elementos HTML devem ser apresentados, podem ser aplicadas a elementos individuais, classes ou *IDs*, permitindo um alto grau de customização. Além disso, o CSS oferece a possibilidade de criar hierarquias de estilos, onde os mais específicos sobrepõem aos mais gerais. As folhas de estilo podem ser inseridas diretamente no documento HTML ou vinculadas a ele através de um arquivo externo, proporcionando maior organização e reutilização de estilos [166]. Um exemplo simples pode ser visto na Listagem 3.2.

---

```
1      <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
2      <html>
3          <head>
4              <title>My first styled page</title>
5              <style type="text/css">
6                  body {
7                      color: purple;
8                      background-color: #d8da3d }
9                  </style>
10             </head>
11
12         <body>
13             (...)
```

---

Listagem 3.2: Exemplo CSS incluído na página HTML [167]

As linhas 5 e 6 indicam que é uma folha de estilos escrita em CSS e que o estilo está aplicado ao elemento *body*. As linhas seguintes definem as cores do texto e fundo. No exemplo dado na Listagem 3.2, a folha de estilos está integrada directamente no ficheiro HTML. Mas à medida que a página cresce em complexidade, torna-se incomportável manter a folha de estilos no ficheiro (ou ficheiros) HTML. Para isso, cria-se a folha de estilos com a extensão *.css* e referencia-se este ficheiro em todas as páginas. Neste caso, o exemplo da página apresentado na Listagem 3.2 ficaria da forma apresentada na Listagem 3.3:

---

```

1      <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
2      <html>
3          <head>
4              <title>My first styled page</title>
5              <link rel="stylesheet" href="meuestilo.css">
6          </head>
7
8      <body>
9          (...)
```

---

Listagem 3.3: Exemplo da página HTML com o CSS definida externamente [167]

Por sua vez, o ficheiro ***meuestilo.css*** da forma apresentada na Listagem 3.4:

---

```

1      body {
2          color: purple;
3          background-color: #d8da3d }
4      </style>
5      (...)
```

---

Listagem 3.4: Exemplo do CSS definido externamente [167]

As folhas de estilo CSS e as páginas HTML podem ser validadas para garantir que estão de acordo com os padrões definidos pelo consórcio W3C [168, 169]. Esta validação é importante para assegurar a compatibilidade entre navegadores e dispositivos, bem como para melhorar a acessibilidade e a experiência do utilizador.

Já o *JavaScript* é uma linguagem de programação orientada para objectos e utilizada principalmente para *scripts* dinâmicos do lado do cliente, permitindo que uma aplicação coloque elementos num formulário HTML e responda a eventos do utilizador, como cliques do rato, preenchimento de formulários e navegação na página. Também pode ser utilizada no lado do servidor, permitindo que uma aplicação comunique com uma base de dados, mantenha a continuidade das informações entre diferentes chamadas da aplicação ou manipule ficheiro no servidor. Isso significa que, no navegador, o *JavaScript* pode alterar a aparência da página e, da mesma forma, o *Node.js* - versão mais avançada do *JavaScript* no servidor - pode responder a solicitações personalizadas enviadas pelo código executado no navegador. Os programas escritos nesta linguagem são designados por *scripts* e podem ser inseridos diretamente numa página HTML, sendo executados automaticamente aquando do seu carregamento. Por serem interpretados, estes scripts são escritos em texto simples e não requerem compilação prévia para serem executados. O *JavaScript* pode ser executado não apenas nos navegadores, mas também nos servidores, ou, basicamente, em qualquer dispositivo que tenha um *JavaScript Engine*, isto é, *software*

específico que execute o código *JavaScript* [170]. Na Listagem 3.5 está representado um exemplo de *JavaScript* que desabilita os seletores de um formulário quando o botão “OK” é clicado.

---

```
1 (...)  
2 <script>  
3 //Accionar a funcao habilitarSeletores quando o botao "OK"  
    for clicado  
4  
5 document  
6     .querySelector(".stop")  
7     .addEventListener("click", desabilitarSeletores);  
8 </script>  
9 (...)
```

---

Listagem 3.5: Exemplo de *JavaScript*

Estas linguagens - HTML, CSS e *JavaScript* - são interdependentes e frequentemente utilizadas em conjunto no desenvolvimento *web*, cada uma desempenhando um papel específico na construção e estilização de páginas *web* e na implementação de interatividade. Se o HTML define o conteúdo das páginas, o CSS define a sua aparência e o *JavaScript* define o comportamento [171].



## Capítulo 4

# Implementação

“Do or do not. There is no try.”

— Master Yoda, *Star Wars - The Empire Strikes Back*

Este capítulo aprofunda os aspectos técnicos do LaRE, tomando como referência a arquitetura apresentada na Figura 3.7. O desenvolvimento começou pela implementação e desenvolvimento do servidor *Flask* e página *web*, sendo que o primeiro circuito a ser implementado e testado foi a Lei de Ohm. O projecto foi evoluindo com a adição dos restantes circuitos que compõem o LaRE. Para o desenvolvimento do *software*, utilizou-se o IDE *Visual Studio Code*<sup>1</sup>, e o projeto encontra-se alojado no *GitHub*. O sistema operativo instalado no Raspberry Pi é uma versão ligeiramente modificada do *Arch Linux ARM*. Os circuitos que compõem o LaRE foram testados e validados individualmente em *breadboards*, antes de se proceder à construção da matriz de placas, como referido na Secção 3.5.2. No Anexo A encontram-se os esquemas completos de todas as experiências, bem como o desenho das placas que compõem a matriz do LaRE. Esta matriz foi organizada em três placas distintas: uma dedicada à experiência da Lei de *Ohm*, outra às experiências de rectificação e filtragem e uma terceira exclusivamente responsável pelas fontes de alimentação.

---

<sup>1</sup><https://code.visualstudio.com/>

## 4.1 Hardware

### 4.1.1 Relés

Como já foi referido na Secção 3.2, o LaRE é composto por cinco circuitos. A implementação começou com o desenho esquemático e, tal como referido na Secção 3.5.4, tentou-se, sempre que possível, utilizar os relés SPST no comando das fontes e aparelhos de medida e os relés DPST no controlo dos componentes, tal como se pode ver na Figura 4.1. O relé  $K_4$  - SPST - comanda a fonte de alimentação de 5 V e os relés  $K_1$  a  $K_3$  - DPST - comandam os componentes.

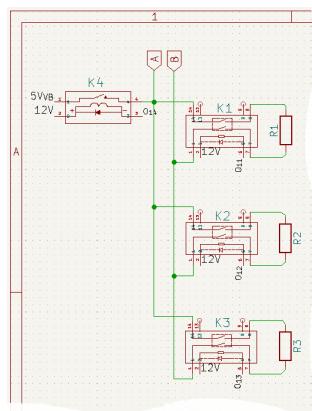


Figura 4.1: Exemplo de utilização de relés SPST e DPST

### 4.1.2 Driver de relés

Conforme referido na Secção 3.5.5, o Raspberry Pi não tem capacidade para comandar directamente os relés. Para esse efeito, é necessário utilizar um *driver* de relés. A Figura 4.2 apresenta o diagrama simplificado correspondente a um canal de entrada e saída do *driver* [154].

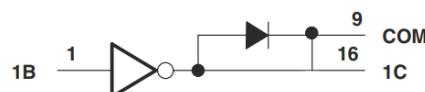
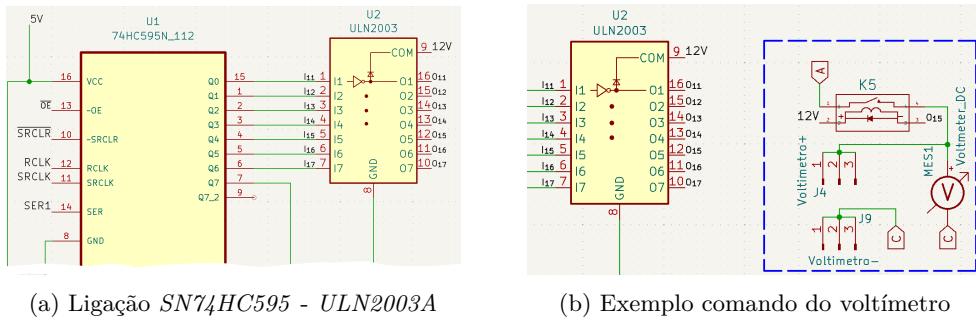


Figura 4.2: Diagrama de blocos simplificado do *ULN2003A*

O terminal 9, designado por *COM*, deve ser ligado à tensão de alimentação dos relés — 12 V no caso dos utilizados neste projecto — e destina-se aos díodos de roda livre”, conforme descrito na Secção 3.5.4. As Figuras 4.3a e 4.3b servem como exemplos práticos que ilustram o funcionamento do procedimento referido na Figura 4.4. Neste exemplo, foi utilizado um relé SPST. No entanto, o funcionamento é idêntico para os relés DPST, variando apenas a disposição dos pinos.

As entradas  $I_1$  a  $I_7$  recebem os *bits* enviados pelo Raspberry Pi através do registo de deslocamento, conforme apresentado na Figura 4.3a. As saídas  $O_1$  a  $O_7$  controlam

Figura 4.3: Exemplo de uso do *SN74HC595* e *ULN2003A*

os relés nos terminais “3”, enquanto os terminais “2” dos relés são ligados aos 12 V, como ilustrado na Figura 4.3b. No caso particular do exemplo descrito em cima, se  $I_5 = 0\text{V}$ , então, a saída  $O_5 = 12\text{V}$  e não há diferença nos terminais “2” e “3” na bobina do relé “K5”. O relé está desactivado. Por outro lado, se  $I_5 = 12\text{V}$ , então, a saída  $O_5 = 0\text{V}$ , há diferença de potencial aos terminais da bobina e o relé está activado. No caso em que a *string* a ser transmitida for maior que 8 bits, há a necessidade de usar mais do que um registo de deslocamento. Neste caso, o pino 9 -  $Q'_H$  - do primeiro registo é ligado ao pino SER do segundo registo e assim sucessivamente.

#### 4.1.3 Registo de deslocamento

O *SN74HC595* é um registo de deslocamento de 8 bits do tipo SIPO, cujo funcionamento já foi detalhado na Secção 3.5.6. Uma *string* de (até 8) bits é enviada pelo Raspberry Pi para o registo através do pino SER. Na Figura 4.4 pode ver-se o diagrama explicativo do processo de envio:

1. Pinos OE e SRCLR activados;
2. Um bit “0” ou “1” é enviado para o pino SER;
3.  $N$  impulsos ascendentes no pino SRCLK fazem com que os  $N$  bits sejam enviados para o registo de deslocamento ( $N$  bits  $\leq 8$ );
4. Um impulso ascendente no pino RCLK faz com que os  $N$  bits sejam enviados para o registo de memória e, por conseguinte, para os relés (através do *ULN2003A*).

#### 4.1.4 Fontes de alimentação

O LaRE utiliza várias fontes de alimentação, todas passíveis de serem controladas por software e neste aspecto tentou-se simplificar/limitar o uso de fontes externas, além das que poderiam ser fornecidas pelo VB.

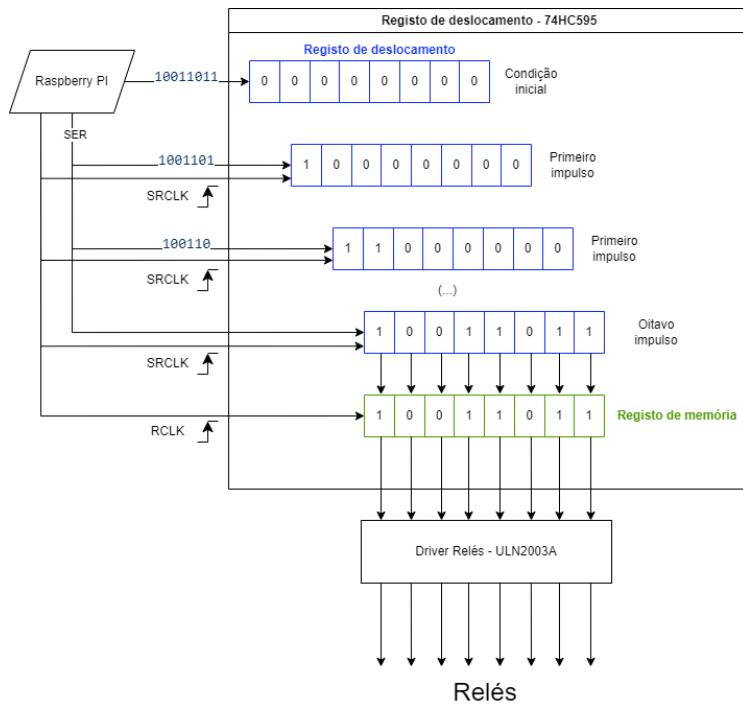


Figura 4.4: Envio de *bits* para o registo de deslocamento

Como pode ser visto na Figura 3.16 ou na Figura 4.5 com mais pormenor, o VB possuí três fontes de alimentação (CC variáveis): 6 V, 25 V e -25 V<sup>2</sup>.



Figura 4.5: Fontes de tensão do VB

Assim, referente ao VB e à Figura 4.5, o LaRE utiliza as seguintes fontes de alimentação:

- 6 V (CC), utilizada na experiência da Lei de Ohm;
- 25 V (CC), utilizada na alimentação dos relés e dos *drivers*.

No entanto, ainda são necessárias mais duas fontes de alimentação, nomeadamente:

<sup>2</sup>No contexto do LaRE a fonte de tensão negativa não é usada.

- 5 V (CC), fonte projectada com recurso ao *LM317* e com tensão de entrada obtida a partir da fonte de 25 V - utilizada na alimentação dos registos de deslocamento;
- Para o estudo dos circuitos rectificação e filtragem, foi utilizado um transformador 220 V/8 V CA, disponível em contexto laboratorial e que se mostrou adequado aos objectivos trabalho.

### Fonte de tensão 6 V - variável e Fonte de tensão 12 V - fixa

As fontes de 6 V e 25 V são configuradas por *software*. A primeira, utilizada na experiência da Lei de *Ohm*, varia entre 1 V e 6 V, enquanto a segunda é ajustada para fornecer 12 V. Mais detalhes sobre estas configurações são apresentados na Secção 4.2.2 e os esquemas consultados no Anexo A.

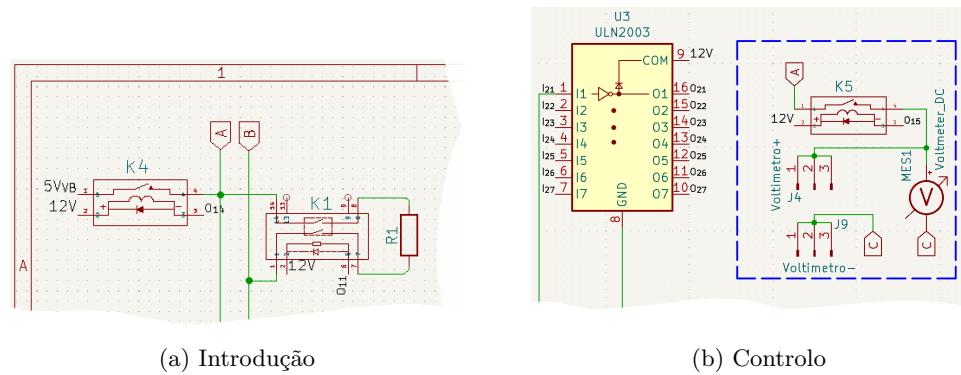


Figura 4.6: Implementação Fontes de 6 V e 12 V

As Figuras 4.6a e 4.6b ilustram os exemplos das ligações das respetivas fontes de tensão. Complementando com a Figura 4.5, o terminal “1” do relé *K<sub>4</sub>*, designado por *5V<sub>VB</sub>*, está ligado ao terminal de 6 V do VB, enquanto a alimentação dos relés - terminais “2” e a alimentação dos *drivers* - terminais “9”, estão ligados ao terminal de 25 V.

### Fonte de tensão 5 V - fixa

De forma a alimentar os circuitos com 5 V fixos foi projectada - Figura 4.7a e implementada - Figura 4.7b - uma fonte de alimentação [172].

Esta fonte teve como base o *LM317* e o esquema presente na página 11, Figura 9, do *datasheet* [172]. A utilização do *LM317* em detrimento, por exemplo, do *LM7805* prendeu-se com a disponibilidade do regulador em contexto laboratorial. Este regulador serve perfeitamente os propósitos deste projecto, já que a tensão de saída, regulável, varia entre os 1.25 V e os 37 V e a corrente de saída pode ser superior a 1.5 A.

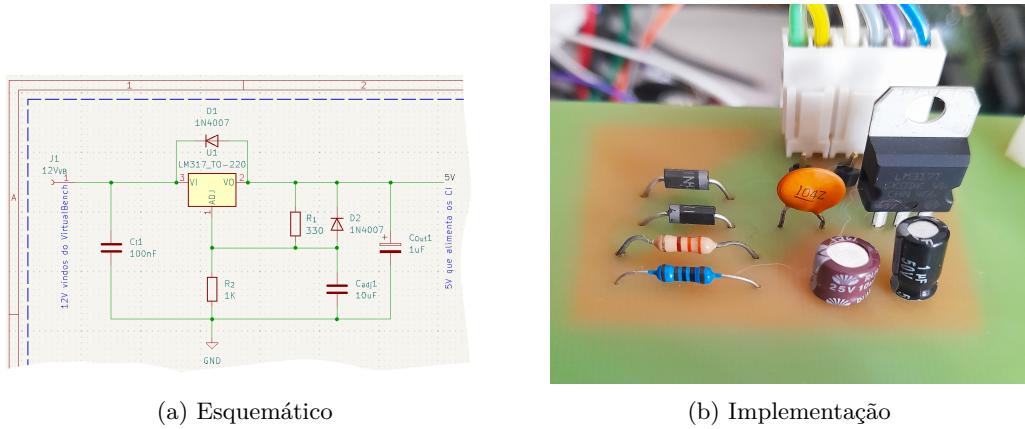


Figura 4.7: Fonte de 5 V [LM317]

Relativamente aos diodos de protecção,  $D_1$  e  $D_2$ , representados no esquema da Figura 4.7, em vez de se usarem os diodos 1N4002, por uma questão de disponibilidade dos componentes, usaram-se os diodos 1N4007. Estes diodos pertencem à mesma família de diodos retificadores da série 1N400x. Ambos têm especificações elétricas e mecânicas semelhantes, nomeadamente, a capacidade de suportar correntes directas até 1 A - valor suficiente para o nosso projecto que, relembrar-se, é limitado pelo VB a 0.5 A. No entanto, a principal diferença entre eles está na tensão inversa máxima que cada modelo é capaz de suportar [173]:

- 1N4002: Possui uma tensão inversa máxima de 100 V;
- 1N4007: Suporta uma tensão inversa máxima de 1000 V.

Estes valores são, pois, mais que suficientes para os níveis de tensão usados.

Sendo assim, para uma tensão de saída de 5 V, calcularam-se as resistências com base na expressão apresentada do *datasheet*, representada na Equação 4.1:

$$V_O = V_{REF}(1 + \frac{R_2}{R_1}) + (I_{ADJ} \times R_2) \quad (4.1)$$

De notar que, segundo o *datasheet* do LM317 [172],  $V_{REF} = 1.25$  V e o valor do termo  $I_{ADJ} \times R_2$  pode ser simplificado. De facto,  $I_{ADJ}$  é, tipicamente,  $50 \mu\text{A}$ , sendo que, para o nosso projecto, o valor escolhido para  $R_2$  foi de  $1\text{k}\Omega$ . Assim, segundo a Equação 4.2, tem-se que:

$$V_{ADJ} = I_{ADJ} \times R_2 = 0.05 \text{ V} \quad (4.2)$$

A Equação 4.1 fica então reduzida a:

$$V_O = V_{REF}(1 + \frac{R_2}{R_1}) \quad (4.3)$$

Considerando  $V_O = 5\text{ V}$ , e uma vez que a Equação 4.3 tem duas incógnitas,  $R_1$  e  $R_2$ , atribuiu-se a  $R_2$  o valor de  $1\text{ k}\Omega$ , tal como foi dito anteriormente. Desta forma, obteve-se  $R_1 = 333.33(3)\text{ }\Omega$ , tal como descrito na Equação 4.4:

$$5\text{ V} = 1.25 \times \left(1 + \frac{1000}{R_1}\right) \Leftrightarrow R_1 = 333.33(3)\text{ }\Omega \quad (4.4)$$

O valor da resistência normalizada mais próxima é de  $330\text{ }\Omega$ . Importa, pois, reverter a Equação 4.3 e confirmar o valor de  $V_O$  obtido:

$$V_O = 1.25 \times \left(1 + \frac{1000}{330}\right) \Leftrightarrow V_O = 5.037(87)\text{ V} \quad (4.5)$$

Portanto, este é um valor perfeitamente aceitável para alimentar os circuitos integrados.

### Fontes de tensão 220 V/8 V CA

As experiências respeitantes aos rectificadores e filtros utilizam tensão alternada sinusoidal. Tal como já foi referido anteriormente, tentou-se, sempre que possível, simplificar o uso de fontes externas.

Na prática e em contexto de sala de aula, a experiência de rectificação de onda completa levanta um problema de massas, se se pretender usar um gerador de sinal. Quer isto dizer que não é possível medir, simultaneamente, os sinais de entrada e saída. Como se pode ver pelo esquema representado na Figura 4.8, o diodo está curto-circuitado (linha magenta a tracejado, entre os pontos 2 e 3) devido ao facto das massas serem comuns.

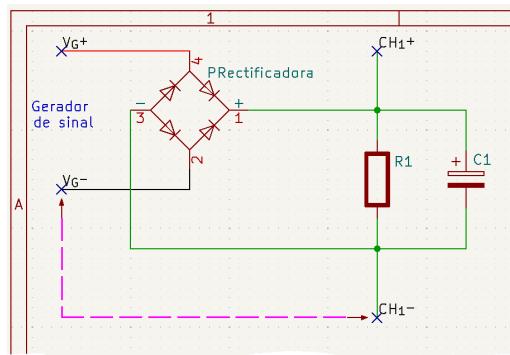


Figura 4.8: Problema de massa - onda completa

A forma de contornar este problema é usar um transformador e, assim, medir a onda de saída com o osciloscópio. Ainda assim, não é possível medir as ondas de entrada e saída **simultaneamente**, se for esse o objectivo.

No caso do LaRE, e com recurso ao uso de relés, é possível controlar a massa dos dois canais do osciloscópio por *software*, tal como representado no esquema simplificado da Figura 4.9 e representar as duas ondas (entrada e saída) na mesma

imagem. De referir que, desta forma, seria possível usar o gerador de sinal. No entanto, o uso do transformador justifica-se pelas seguintes razões:

- Facilidade e simplicidade em controlar as massas por *software* - três no caso do uso do gerador de sinal ( $CH_{1-}$ ,  $CH_{2-}$  e  $VG_-$ ) ou duas no caso de se usar um transformador ( $CH_{1-}$ ,  $CH_{2-}$ );
- Em contexto laboratorial e de sala de aula usa-se um transformador para rectificar a onda completa.

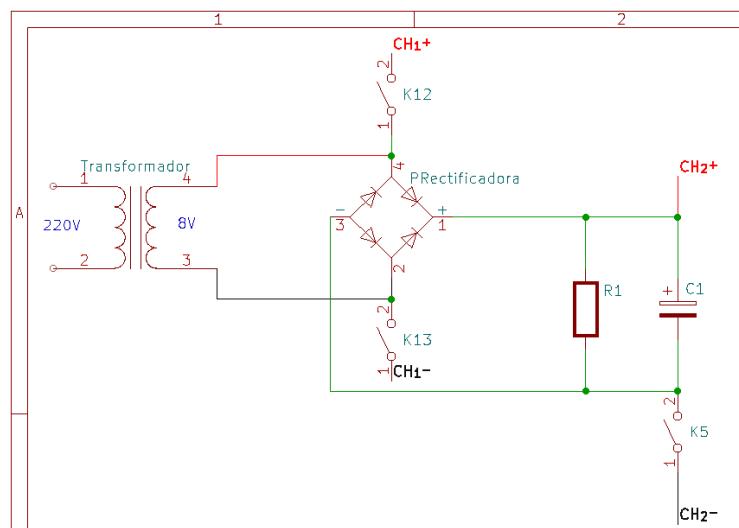


Figura 4.9: Problema de massa - onda completa [Resolvido]

O esquema completo da experiência de rectificação de onda completa pode ser consultado no Anexo A.

#### 4.1.5 Aparelhos de medida - voltímetro e amperímetro

A medição de tensão e corrente são feitas utilizando o multímetro digital - *Digital Multimeter* (DMM) - PXI-4072, referenciado na Secção 2.4.1, Figura 2.21 e cujo painel frontal está representado na Figura 4.10.



Figura 4.10: Painel frontal Multímetro Digital

Na Figura 4.11a está representado o esquema simplificado da Lei de *Ohm* que pode ilustrar o funcionamento dos aparelhos de medida. O ponto C é comum e é

controlado pelo relé  $K_7$ . O terminal positivo do voltímetro está ligado ao ponto A, controlado pelo relé  $K_5$  mas o amperímetro necessita de mais um relé, já que é inserido em série. Neste caso o relé  $K_8$  funciona como um *bypass* quando se pretende medir tensão e o relé  $K_6$  controla o terminal positivo do amperímetro.

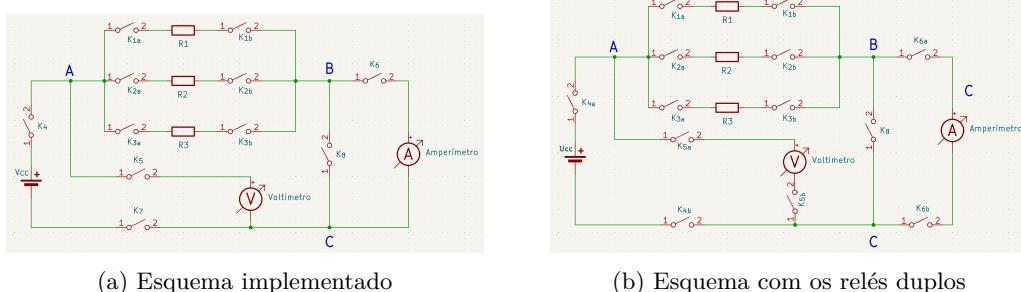


Figura 4.11: Esquema completo - Lei de *Ohm*

No entanto, faz-se notar o seguinte: Embora os aparelhos de medida sejam independentes, partilham a mesma massa. Esta escolha foi motivada pela indisponibilidade de relés duplos em contexto laboratorial, pelo que se optou por fazer a gestão da forma a que o propósito do projecto fosse atingido e o conceito provado. Na Figura 4.11b apresenta-se o esquema simplificado, com os relés duplos, adicionados aos aparelhos de medida.

De referir que, se se pretender expandir o LaRE com mais circuitos, é possível utilizar o voltímetro e o amperímetro da forma como estão implementados. A Tabela 4.1 apresenta o estado dos relés,  $K_5$  a  $K_8$  consoante a grandeza que se pretende medir.

Tabela 4.1: Exemplo funcionamento de medição da Lei de Ohm

Estado dos relés							
$K_1$	$K_2$	$K_3$	$K_4$	$K_5$	$K_6$	$K_7$	$K_8$
Medir Tensão	1	0	0	1	1	0	1
Medir Corrente	1	0	0	1	0	1	0

#### 4.1.6 Experiências

De forma a permitir um melhor desenvolvimento, foco, organização, detecção e correção de erros, as experiências foram implementadas de forma independente. As primeiras ideias passavam por, além da Lei de *Ohm*, implementar o estudo da rectificação de meia e onda completa. Contudo, verificou-se que seria possível incluir uma experiência adicional, dedicada ao estudo prático de filtros passa-alto e passa-baixo, já que estas duas experiências possuem componentes comuns, ampliando, assim, o alcance e a aplicabilidade do LaRE.

### Lei de Ohm

Na Figura 4.11a apresenta-se o esquema simplificado da implementação prática da Lei de *Ohm*. O esquema completo pode ser consultado no Anexo A. A Figura 4.12, anteriormente apresentada, é aqui repetida com o objetivo de ilustrar de forma mais clara a correspondência entre o circuito e a sua implementação física na placa, tal como representado na imagem.

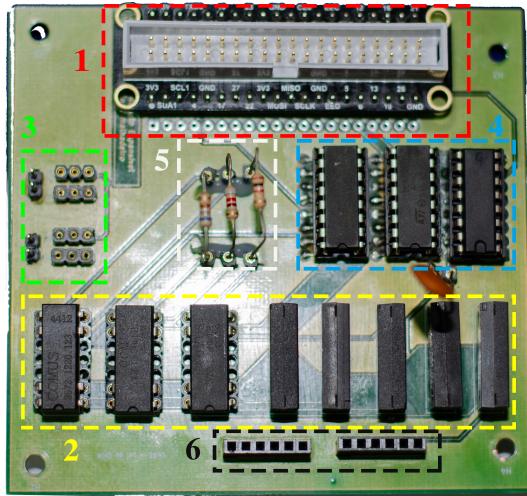


Figura 4.12: [Implementação] Lei de *Ohm*

A fonte de tensão e os aparelhos de medida foram projetados de forma a serem independentes, tal como referido na Secção 4.1.5, permitindo a sua utilização caso se pretenda expandir o LaRE com mais experiências. A fonte de tensão pode ser utilizada desativando  $K_5$  e  $K_7$ .

O funcionamento do circuito é relativamente simples, sendo que qualquer que seja a resistência a medir, é fechado o respectivo relé -  $K_1$  ou  $K_2$  ou  $K_3$ ; a fonte  $U_{CC}$  é ligada ao circuito através do relé  $K_4$  e  $K_7$ . A medição da tensão e da corrente faz-se da seguinte forma:

- Medição da tensão:
  - Os relés  $K_5$  e  $K_8$  são fechados e  $K_6$  é aberto (Amperímetro desactivado). Desta forma, o voltímetro fica em paralelo com a resistência a medir - entre os pontos A e B.
- Medição da corrente:
  - Os relés  $K_5$  e  $K_8$  são abertos (voltímetro desactivado) e  $K_6$  é fechado. Desta forma, o amperímetro fica em série com a resistência a medir - entre os pontos B e C.

Como exemplo, se se pretender estudar a Lei de Ohm para a resistência  $R_1$ , os relés seriam activados da forma representada na Tabela 4.1:

### Rectificadores e filtros

Uma vez que a implementação destas duas experiências foi realizada na mesma placa, além de que partilham componentes, optou-se por apresentá-las em conjunto. Na Figura 4.13 estão representados os esquemas simplificados dos circuitos. Nela se podem ver os componentes comuns - rectângulo tracejado laranja - assim como os rectificadores de meia e onda completa - rectângulo tracejado vermelho e verde, respectivamente. Exclusivo dos filtros estão, também, representados, o condensador e resistência, no rectângulo tracejado magenta. O esquema completo encontra-se representado no Anexo A.

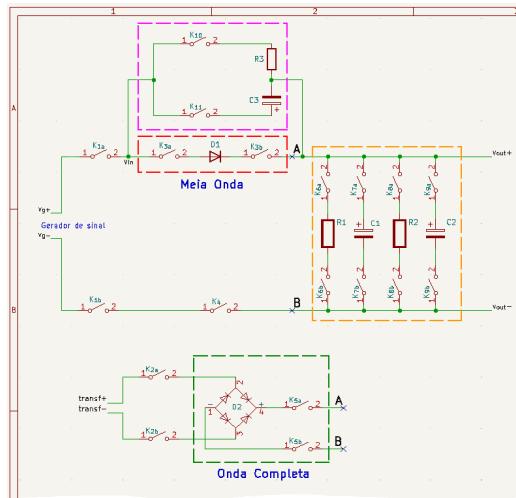


Figura 4.13: Esquema completo (simplificado) - rectificadores e filtros

A Figura 4.14, já anteriormente apresentada, é aqui retomada para ilustrar com maior clareza a correspondência entre o circuito e a sua implementação física na placa. Nesta implementação, foi necessário recorrer a dois tipos distintos de tensão de entrada. Como referido na Secção 4.1.4, o circuito de meia onda utiliza o gerador de sinal, mas, devido ao problema de massa já descrito nessa mesma secção, optou-se por utilizar um transformador no circuito de rectificação de onda completa.

Como referido na Secção 4.1.6, a implementação dos filtros foi realizada após a conclusão dos rectificadores. De forma a permitir a integração desta experiência no circuito já concluído, foi necessário realizar um *bypass* ao diodo  $D_1$  (assinalado pelo retângulo tracejado a vermelho na Figura 4.13). Esta alteração permitiu a introdução de uma resistência ( $R_3$ ) ou de um condensador ( $C_3$ ), consoante a configuração do filtro em estudo. Os esquemas teóricos dos filtros estão representados nas Figuras 3.13a e 3.13b, sendo que  $R$  e  $C$  correspondem a  $R_3$  e  $C_3$ .

Portanto, se se pretender estudar o filtro passa-baixo, é activado o relé  $K_{10}$  ( $R_3$ ) e o condensador de saída pode ser escolhido entre os  $C_1$  e  $C_2$ , consoante a configuração que se pretende estudar. O mesmo se aplica ao filtro passa-alto, onde o relé  $K_{11}$

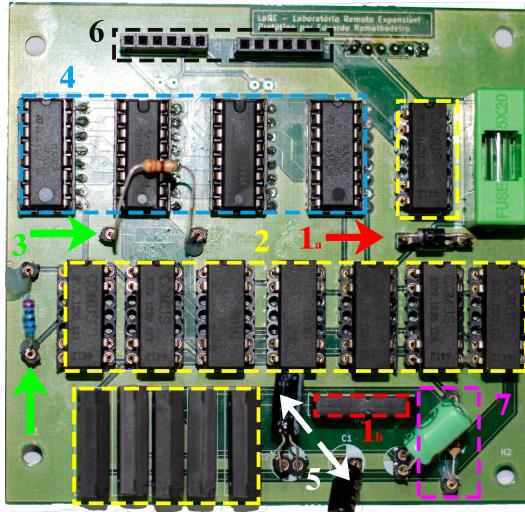


Figura 4.14: [Implementação] rectificadores e filtros

( $C_3$ ) é activado e a resistência de saída pode ser escolhida entre os  $R_1$  e  $R_2$ . Na Tabela 4.2 e com o auxílio da Figura 4.13 (ou do circuito completo apresentado no Anexo A), pode ver-se o estado dos relés, respeitante a uma combinação possível para cada circuito.

Tabela 4.2: Exemplo funcionamento do rectificador de meia onda

	Estado dos relés												
	$K_1$	$K_2$	$K_3$	$K_4$	$K_5$	$K_6$	$K_7$	$K_8$	$K_9$	$K_{10}$	$K_{11}$	$K_{12}$	$K_{13}$
Meia Onda	1	0	1	1	0	1	1	0	0	0	0	0	0
Onda Completa	0	1	0	0	1	0	0	1	1	0	0	1	1
Filtro Passa-Alto	1	0	0	1	0	0	1	0	0	1	0	1	1
Filtro Passa-Baixo	1	0	0	1	0	0	0	1	0	0	1	1	1

## 4.2 Software

Hoje em dia vive-se (n)uma Era em que toda a informação está disponível na “ponta dos dedos” e à distância de um *click*. A pesquisa, desenvolvimento e testes dos assuntos mais técnicos revelou-se um processo exigente e, por vezes, moroso. A abundância dos recursos disponíveis colocou o desafio adicional de tentar perceber a dicotomia certo/errado. No apoio a este processo, recorreu-se à Inteligência Artificial, documentação técnica *online*, fóruns de discussão, ajuda pessoal, tutoriais *online* e vídeos no YouTube. Inteligência Artificial e à consulta de documentação técnica especializada.

Este projeto de código fonte aberto está dividido em dois repositórios, disponíveis para consulta e modificação sob a licença GPL-3.0 (GNU General Public License,

versão 3) no GitHub: LaRE e PiCode. Assim, os ficheiros são referenciados relativamente ao respectivo repositório. Ao longo deste capítulo, sempre que se justificar e para fins de clareza, será apresentado o código necessário - em pequenos trechos - de forma a melhor ilustrar os conceitos discutidos. Não se pretende, contudo, abordar exaustivamente todas as instruções ou métodos de configuração do VB, uma vez que os ficheiros se encontram comentados, sendo possível obter informações adicionais através da sua consulta.

#### 4.2.1 Servidor Flask

Na Secção 3.6.1 foram já referidas as razões que levaram à escolha do *Flask* como servidor *web*. A base para a implementação do *Flask* teve como base e referência a documentação técnica disponível no *site* do *Flask* [137] e complementada com alguns tutoriais do *Youtube* [174, 175].

##### Estrutura base

O fluxograma apresentado na Figura 4.15 apresenta o funcionamento geral do servidor *Flask*.

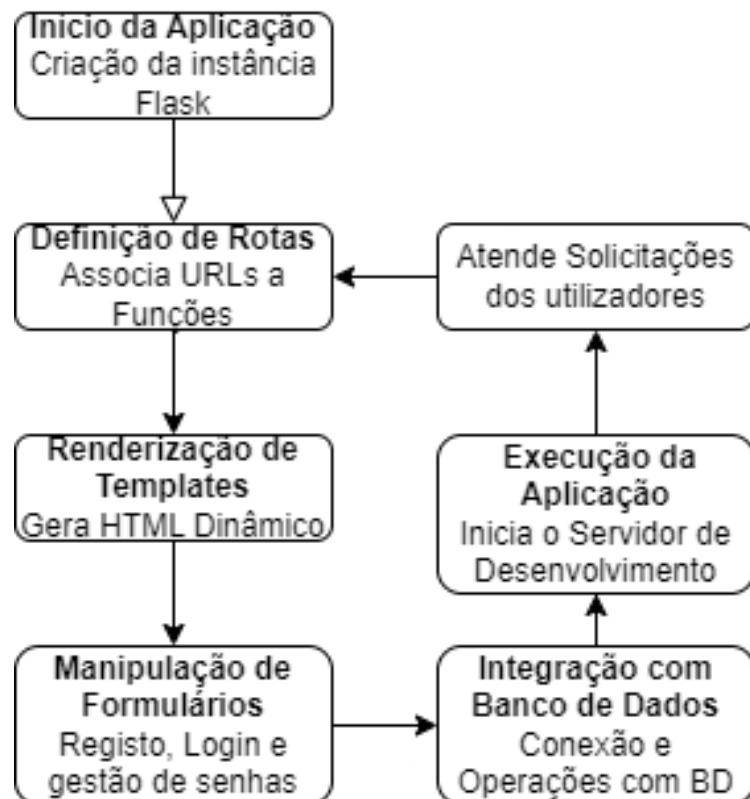


Figura 4.15: Funcionamento geral *Flask*

A estrutura de directórios do *Flask* tem uma base pré-definida que não é necessariamente rígida e pode ser adaptada consoante os requisitos do projecto [137]. No caso do LaRE a estrutura ficou organizada da forma como se mostra na Figura 4.16

```

webserver/
    ├── instance/
    │   └── database.db
    ├── website/
    │   ├── static/
    │   │   ├── images/
    │   │   ├── index.js
    │   │   └── style.css
    │   └── templates/
    │       ├── base.html
    │       ├── home.html
    │       ├── login.html
    │       ├── main.html
    │       └── sign_up.html
    ├── __init__.py
    ├── auth.py
    ├── models.py
    └── views.py
main.py

```

Figura 4.16: Estrutura de directórios - *Flask main.html está a mais*

A raiz do projecto é o directório “*webserver*” e nele estão contidos o ficheiro principal, assim como outros directórios e ficheiros de configuração essenciais:

- *main.py*: O ficheiro principal do aplicativo que inicializa e configura o *Flask*;
- *requirements.txt*: Uma lista de dependências do projeto que podem ser instaladas usando *Pip Installs Packages* (PIP);
- *static/*: Este directório contém os ficheiros estáticos como CSS, *JavaScript* e as imagens usadas em toda a aplicação;
- *templates/*: Contém os modelos HTML usado para renderizar as visualizações;
- *instance/*: Diretório para armazenar ficheiros de configuração ou ficheiros que mudam em tempo de execução, específicos da cada instância, por exemplo, ficheiros da base de dados.

Dentro da raiz, criou-se o directório *website* onde se incluiu os ficheiros respeitantes ao funcionamento do *site*. Nele constam os já descritos *templates*, *static* e ainda os seguintes ficheiros:

- *\_\_init\_\_.py*: Inicializa o *Flask* e define as configurações. Este ficheiro dentro da directória *website* faz com que esta seja tratada como um pacote *Python*.

Isto quer dizer que a directória pode ser importada e tudo o que estiver dentro é executado automaticamente;

- *views.py*: Contém as funções de visualização para o tratamento de pedidos *Hypertext Transfer Protocol* (HTTP);
- *models.py*: Define os modelos de dados para a aplicação;
- *auth.py*: Este ficheiro é responsável por lidar com a autenticação e autorização de utilizadores, pode incluir funções e rotas que permitem aos utilizadores registarem-se, fazer *login* e fazer *logout*.

## Rotas

As aplicações Web modernas utilizam URLs amigáveis para ajudar os utilizadores a memorizar e utilizar o nome para voltar a visitar directamente uma página.

No *Flask* utiliza-se o *decorator route()* para associar uma função a um URL, tal como pode ser visto na Listagem 4.1.

---

```
1 @views.route("/ohm", methods=['GET', 'POST'])
2 @login_required
3 def pagina_seguinte():
4     return render_template("ohm.html", user=current_user)
```

---

Listagem 4.1: *Decorator route() - views.py*

O *decorator route* define a rota correspondente à URL *ohm*, a qual aceita pedidos HTTP de métodos *GET* e *POST*. Adicionalmente, o *decorator login* assegura que apenas utilizadores autenticados podem aceder a esta rota; caso contrário, o utilizador será redirecionado para a página de autenticação. Sempre que esta rota é invocada, a função *página\_seguinte()* é executada. Por fim, a última linha da função invoca o template *home.html* e passa-lhe o objeto *current\_user*, permitindo personalizar a página com os dados do utilizador autenticado.

Para construir um URL para uma função específica, usa-se a função *url\_for()*. Esta função aceita o nome da função como seu primeiro argumento e qualquer número de argumentos de palavras-chave, cada um correspondendo a uma parte variável da regra de URL, tal como pode ser visto na Listagem 4.2

---

```

1  (...)

2  if user:
3      if check_password_hash(user.password, password):
4          flash('Logged in successfully!', category='success')
5      login_user(user, remember=True)
6      return redirect(url_for('views.home'))
7  (...)
```

---

Listagem 4.2: Construção de *urls - auth.py*

Isto quer dizer que o *Flask* gera a URL correspondente à função *home()* definida no ficheiro *views.py* dentro da Blueprint *views*. Quando usada dentro de *redirect*, redireciona o utilizador para essa URL. No *Flask*, os *decorators* são utilizados para associar funcionalidades adicionais a funções que representam rotas. Um *decorator* é uma construção da linguagem *Python* que permite modificar o comportamento de uma função sem alterar o seu conteúdo. Ao utilizar *@route*, *@login\_required* e outros *decorators*, o *Flask* cria uma camada adicional de lógica que é aplicada automaticamente antes (ou depois) da execução da função “decorada”.

## Blueprints

O *Flask* usa um conceito de *blueprints* para criar componentes de aplicações e suportar padrões comuns dentro de uma aplicação ou entre aplicações. As *blueprints* podem simplificar muito o funcionamento de grandes aplicações e fornecer um meio central para que as extensões do *Flask* registem operações em aplicações. As *blueprints* permitem organizar a aplicação em partes menores e de mais fácil gestão. O conceito básico das *blueprints* é que registam as operações a executar quando são integrados numa aplicação. O *Flask* associa funções de vista (*views*) a *blueprints* quando processa pedidos e gera URL entre diferentes pontos de acesso. Na Listagem 4.3 e Listagem 4.4 podem ver-se as duas *blueprints* definidas no caso da implementação do LaRE.

---

```

1  views = Blueprint('views', __name__)
```

---

Listagem 4.3: Blueprint *views - views.py*


---

```

1  auth = Blueprint('auth', __name__)
```

---

Listagem 4.4: Blueprint *auth - auth.py*

Ambas as *blueprints* são registadas no ficheiro `__init__.py`, como se pode verificar na Listagem 4.5

---

```
1 app.register_blueprint(views, url_prefix='/')
2 app.register_blueprint(auth, url_prefix='/')
```

---

Listagem 4.5: Registo das *blueprints* -  
`__init__.py`

Isto quer dizer que ao registar a *blueprint* com o prefixo “/”, os URLs serão acessíveis através da raiz da aplicação.

## Pedidos

A capacidade de uma aplicação *web* em responder a solicitações de dados do cliente é um requisito essencial. No *Flask* esta informação é fornecida pelo objeto global ***request***. Este objeto contém todos os detalhes da solicitação, como os métodos HTTP usados (*GET*, *POST*, etc.), os *headers* os *cookies* ou o corpo da requisição. O método de requisição pode ser determinado através do atributo ***method***. Para aceder aos dados de um formulário (transmitidos numa requisição *POST* ou *PUT*), utiliza-se o atributo ***form***, tal como descrito no ficheiro `auth.py`. Quando se adiciona um ponto de interrogação (?) seguido de pares chave-valor (key=value) no final de um URL, está a enviar-se dados adicionais ao servidor, esses dados são chamados de **parâmetros do URL**, como se pode ver na Listagem 4.6.

---

```
1 const url = '/config_VirtualBench?parameter=${parameter}&
Vcc=${Vcc}&R=${Resistance}';
```

---

Listagem 4.6: Exemplo argumentos passados ao servidor -  
`ohm.html`

Como referido na Secção 3.6.1, a renderização das páginas HTML é realizada através do motor *Jinja*. No caso específico do LaRE, o *Flask* procura os modelos a renderizar na pasta *templates*, conforme ilustrado na Figura 4.16. Na página oficial do *Flask* é recomendado que se acceda aos parâmetros do URL com ***get*** ou capturando o ***KeyError***, uma vez que os utilizadores podem alterar o URL e, nesse caso, apresentar uma página *400 bad request* não é de fácil interpretação.

## Base de dados

A aplicação desenvolvida usa uma base de dados e autenticação de utilizador, a ligação com a base de dados é aberta no início do pedido e é obtida a informação do utilizador. No final do pedido, a ligação com a base de dados é fechada.

No contexto da documentação do *Flask* e implementação de uma base de dados, são apresentados duas alternativas: *SQLite 3* e *SQLAlchemy*. Tal como sugerido na documentação, usou-se a extensão *Flask-SQLAlchemy*, descrita no ficheiro `__init__.py`.

### Autenticação

A segurança da aplicação é assegurada através das funcionalidades de *login* e *logout*, implementadas no ficheiro `auth.py`. Para esse fim, foi instalada a extensão *flask-login* que simplifica a gestão de sessões de utilizadores ao permitir manter o estado autenticado entre diferentes requisições. Esta ferramenta disponibiliza métodos como `login_user()`, `logout_user()` e `current_user`, que simplificam a verificação e controlo do estado de autenticação. O processo de registo envolve a recolha de credenciais, como nome de utilizador e senha, garantindo o armazenamento seguro da senha através de *hashing* com `generate_password_hash`. No *login*, a aplicação verifica as credenciais fornecidas, compara a senha inserida com a armazenada usando `check_password_hash` e, se for válida, inicia a sessão do utilizador. Além disso, certas rotas são protegidas para garantir que apenas utilizadores autenticados as possam aceder, utilizando o *decorator* `@login_required`, que redireciona utilizadores não autenticados para a página de *login*. A gestão de sessões é feita através de *cookies* seguros, permitindo ainda a funcionalidade de lembrar a sessão entre reinícios do navegador, caso seja activado o parâmetro `remember=True` no `login_user`.

Devido à complexidade da implementação de uma verificação por *email*, a aplicação restringe-se à autenticação baseada apenas no nome de utilizador e senha. A inclusão desse mecanismo exigiria um serviço de envio de *emails* e a gestão de *tokens* de confirmação.

#### 4.2.2 Arquitectura de Comunicação

Nesta secção, o foco será analisar a comunicação (e troca de parâmetros) entre os diversos tipos de ficheiros. Na Figura 4.17 é apresentado o diagrama simplificado que ilustra o funcionamento geral deste tipo de comunicação.

A arquitectura do sistema é composta por múltiplos ficheiros que comunicam entre si, englobando a troca de parâmetros. O critério utilizado foi o de manter o ficheiro `views.py` o mais leve e simples possível, mantendo-se o comando e controlo dos dispositivos em ficheiros separados.

Como mencionado na Secção 4.2.1, a estrutura de directórios do *Flask* possui uma base pré-definida, mas suficientemente flexível para ser adaptada conforme os requisitos específicos do projeto, como ilustrado na Figura 4.16. De uma forma geral, foram criados ficheiros específicos para cada experiência, adoptando o critério



Figura 4.17: Diagrama de comunicação simplificado [REVER]

de evitar uma excessiva modularização do código. Embora esta prática ofereça vantagens, também apresenta as suas desvantagens - devido à fragmentação excessiva - já que pode tornar a compreensão e manutenção do código mais complexa.

Os seguintes ficheiros foram criados e associados às respectivas experiências:

#### Específicos

- Lei de *Ohm* - *config.py*;
- Rectificadores/Filtros - *mixed\_signal\_oscilloscope.py*;
- Visualização - [ohm, meiaonda, ondacompleta, passaalto, passabaixo].html.

#### Comuns

- *views.py* - parte central da aplicação, pois nele estão definidas todas as rotas *Flask*. Quer isto dizer que, cada roda representa uma funcionalidade ou página *web* específica e qualquer troca de dados ou argumentos entre ficheiros e/ou *scripts*, passa obrigatoriamente por este ficheiro. Sendo assim, tentou-se, sempre que possível, reduzir o código ao estritamente necessário;
- *configRelays.py* - ficheiro responsável pelo envio da *string* para o *RaspberryPI*.
- [login, base, sign\_up].html - páginas de autenticação e registo de utilizadores.

## Comunicação entre ficheiros

A comunicação entre as páginas HTML e o ficheiro *python*<sup>3</sup> é feita nos dois sentidos - setas amarelas - já que os utilizadores enviam os parâmetros específicos de cada experiência e os resultados são apresentados nas páginas.

A Listagem 4.7 ilustra a forma como os parâmetros são enviados da página HTML para o ficheiro *python*, através de uma *query string*.

---

```
1 const url = '/config_VirtualBench?variavel_1=${parametro_1}&variavel_2=${parametro_2}&variavel_3=${parametro_3}';
```

---

Listagem 4.7: Envio de parâmetros HTML → *views.py*

Do lado do servidor, o *Flask* disponibiliza o objecto *request*, a partir do qual os parâmetros enviados por URL podem ser acedidos através da função *request.args.get()*, como representado na Listagem 4.8.

---

```
1 Vcc = request.args.get('parametro_1', 0, int)
```

---

Listagem 4.8: Recepção de parâmetros

O envio de parâmetros do ficheiro *python* para as páginas HTML é feito em *JavaScript Object Notation (JSON)*, usando a instrução definida na Listagem 4.9.

---

```
1 return jsonify({'measurement_result': measurement_result})
```

---

Listagem 4.9: Envio de parâmetros *views.py* → HTML

Do lado das páginas HTML a recepção do resultado é feita através do método *fetch()*, como se pode ver na Listagem 4.10.

---

```
1 fetch(url)
2   .then((response) => response.json())
3   // Aceder a variável measurement_result
4   .then((data) => {
5     document.getElementById("current-measure").innerHTML
6       =
7       data.measurement_result + " mA";
```

---

Listagem 4.10: Recepção de parâmetros

As setas verdes representam a comunicação entre dois ficheiros *python*. O processo é bastante simples, bastando para isso, importar o ficheiro que contém a função

<sup>3</sup>Pode-se, neste caso, reduzir a comunicação ao caso mais geral de troca de informação entre ficheiros HTML e *python*, já que, no fundo, o ficheiro *views.py* está escrito em *python*.

que se pretende chamar. A Lista 4.11 ilustra a forma como se importa um ficheiro *python* e a chamada da respectiva função com o parâmetro a passar.

---

```

1   from ctrl_hardware import configRelays, configVB,
2       mixed_signal_oscilloscope
configRelays.config_relays_ohm(Resistance,
    measure_parameter)

```

---

Listagem 4.11: Comunicação *python - python*

No entanto, o processo de medição na experiência da Lei de *Ohm* possui características específicas que justificam uma explicação detalhada. A Figura 4.18 ilustra o procedimento de leitura das medições de tensão e corrente.

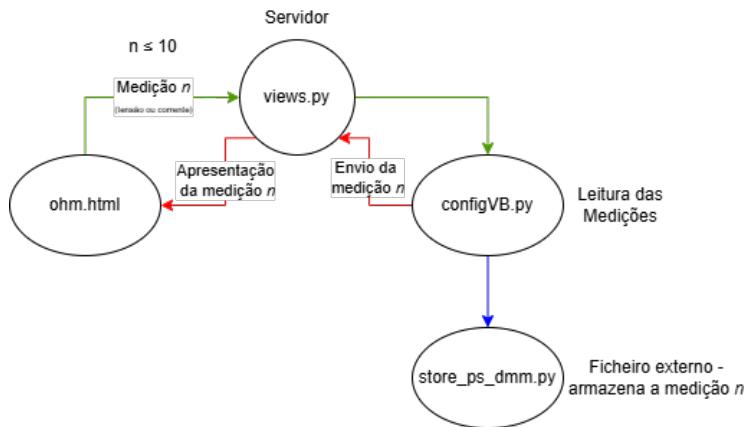


Figura 4.18: Processo de leitura das medições de tensão e corrente

As setas a verde e vermelho representam o fluxo do processo iniciado pelo utilizador ao solicitar uma medição. Este ciclo repete-se cinco vezes para cada grandeza (tensão e corrente). No entanto, como o ficheiro *configVB.py* é executado de forma independente em cada chamada, num total de dez execuções, os valores medidos não são retidos entre invocações sucessivas. Para contornar esta limitação e garantir a persistência das medições ao longo das múltiplas chamadas à API, sem recorrer a mecanismos clássicos de persistência, como ficheiros *.json*, *.csv* ou bases de dados locais — os quais implicariam operações adicionais de escrita e leitura no sistema de ficheiros — foi desenvolvido um módulo auxiliar em *Python*, implementado no ficheiro *store\_ps\_dmm.py*. Este módulo utiliza variáveis globais como mecanismo de armazenamento em memória. Os valores de tensão e corrente são armazenados num *array* através da função *np.append*, da biblioteca *NumPy* [176]. Dado que o servidor *Flask* se mantém activo durante todo o processo de medição, as variáveis globais permanecem acessíveis em memória entre chamadas à API. Esta abordagem simplifica a gestão do estado da aplicação numa fase inicial de desenvolvimento, sendo eficaz enquanto o processo da aplicação não for reiniciado.

## Comunicação com RaspberryPi

Do lado do *Raspberry Pi*, embora de uma forma mais simples, também se usou a premissa de modularização do código, tal como referido na Secção 4.2.2. Este ficou dividido em dois ficheiros:

- *receive.py* - recebe a *string*;
- *shift\_register.py* - responsável por activar os relés.

A comunicação com o *Raspberry Pi* é estabelecida através de sockets, um método já descrito na Secção 3.3. Após a configuração e teste dos parâmetros, uma *string* representando o número de relés a ativar - 8 *bits* ou 13 *bits*, dependendo dos circuitos - é enviada através da *interface* de rede. A implementação do mecanismo de comunicação encontra-se no ficheiro *configRelays.py*, na função *config\_Relays*.

Sendo assim, o ficheiro *receive.py* é executado inicialmente e aguarda pela envio da *string*. De referir que, embora o *socket* esteja configurado em *blocking mode* - quer isto dizer que o cliente ficará bloqueado na Linha 3, da Listagem 4.12, esperando uma resposta do servidor [147] (acontece enquanto a string não for enviada) - optou-se por enviar uma confirmação do servidor ao cliente. Não sendo estritamente necessário, foi adicionada como auxílio à depuração.

---

```

1      # Espera pela resposta do servidor
2      while True:
3          data = s.recv(1024)
4          if not data:
5              break
6          response = data.decode()
7          if response == 'True': # Espera por uma
8              confirmacao especifica do servidor
9              print("Confirmacao recebida do servidor:",
                   response)
10             break

```

---

Listagem 4.12: *Block Mode [Sockets]* *configRelays.py*

O passo final é a activação dos relés, realizada no ficheiro *shift\_register.py*, função *commandRelays*. O procedimento foi descrito na Secção 4.1.3, sendo que os modos de funcionamento estão representados na Tabela 3.2.

Para que as experiências funcionem correctamente, todos os relés devem ser activados em simultaneo. Para isso, a sequência de controlo dos relés é representada por uma *string* composta por 8 ou 13 dígitos binários, conforme as experiências, onde cada carácter indica o estado (ligado ou desligado) de um relé. Esta *string* é posteriormente convertida para um número inteiro, sobre o qual é realizada uma operação de *and bit a bit*. Cada um dos *bits* resultantes é, então, enviado sequencialmente para o registo de deslocamento, numa configuração SIPO, conforme ilustrado

na Listagem 4.13. Assim que os 8 ou 13 *bits* forem transferidos para o registo, a saída paralela será ativada, resultando na activação dos relés.

---

```

1   for i in range(n_bits):
2       binaryShift = binaryString & 1
3       if binaryShift == 1:
4           WriteReg (ON, SERCLK_pin_ctrl, SER_pin_ctrl,
5                     WaitTimeSR)
6       else:
7           WriteReg(OFF, SERCLK_pin_ctrl, SER_pin_ctrl,
8                     WaitTimeSR)
9       binaryString = binaryString >> 1
10      OutputReg(RCLK_pin_ctrl)
11  return True # Fim da transmissao da string de bits,
12      reles activados

```

---

Listagem 4.13: *And bit bit shift\_register.py*

### **VirtualBench - Configurações e medições**

O comando e controlo do VB (e dos respetivos instrumentos) — representado na Figura 4.17 pelas setas vermelhas — é realizado através da biblioteca *pyVirtualBench* [177], a qual, como referido na Secção 3.2, funciona como um *wrapper* que permite controlar o VB a partir de uma aplicação desenvolvida em *Python*. A documentação desta biblioteca, disponível no endereço indicado, é bastante completa e detalhada, sendo recomendada para aprofundamento técnico.

Além de instalar a biblioteca, os autores disponibilizam uma série de exemplos, sendo que, aqueles que mais se adequam aos objectivos deste projecto são:

- *dmm\_example.py*: exemplo que demonstra como efetuar medições utilizando o multímetro digital;
- *fgen\_example.py*: exemplo que demonstra como configurar e gerar uma onda através do gerador de sinal;
- *mso\_simple\_example.py*: exemplo que demonstra como configurar e adquirir dados do osciloscópio, utilizando a funcionalidade de configuração automática incorporada;
- *ps\_example.py*: exemplo que demonstra como efetuar medições utilizando a fonte de alimentação.

De forma a realizar as configurações e medições do VB, tomou-se como ponto de partida os exemplos já fornecidos com a referida biblioteca e adaptaram-se ao contexto do projecto.

Nesta secção, apresenta-se como exemplo a configuração da fonte de alimentação, por ser o único instrumento utilizado em todas as experiências realizadas. Através da análise dos exemplos disponibilizados para o VB (e que estão definidos no repositório já mencionado do *GitHub*), foi possível compreender a estrutura geral de configuração aplicada aos instrumentos e fontes de alimentação. Com base nessa análise, determinou-se a seguinte lógica de funcionamento:

1. Aquisição do *virtualbench*;
2. Aquisição dos instrumentos;
3. Configurações específicas de cada instrumento;
4. Desenvolvimento do programa;
5. Libertar os instrumentos;
6. Libertar o *virtualbench*.

A aquisição do VB, identificado como *VB8012-30A210F*, é feita através da criação de uma instância da classe *PyVirtualBench*, que é a classe principal da biblioteca *pyVirtualBench*, definida na Linha 1, da Listagem 4.14.

---

```

1     virtualbench = PyVirtualBench('VB8012-30A210F')
2     ps = virtualbench.acquire_power_supply()

```

---

Listagem 4.14: Instanciação da classe *PyVirtualBench*

A partir do objeto *virtualbench*, é possível aceder aos diferentes instrumentos, sendo que a aquisição é feita através do método *acquire\_<instrumento>()*. Essa instância é armazenada numa variável definida pelo utilizador, permitindo o acesso às funcionalidades de configuração, controlo e medição disponibilizadas por esse instrumento. Na Listagem 4.14, linha 2, pode ver-se, a título de exemplo, como é feita a aquisição da fonte de alimentação, que é armazenada na variável *ps*. Um exemplo da configuração da fonte está descrita na Listagem 4.15

---

```

1      try:
2          # Configuracao da fonte de alimentacao
3          channel = "ps/+25V"
4          voltage_level = 1.0
5          current_limit = 0.5
6          # Configuracao especifica
7          ps.configure_voltage_output(channel,
8              voltage_level, current_limit)
8          ps.enable_all_outputs(True)

```

---

Listagem 4.15: Exemplo *ps\_example.py*

- *channel* - linha 3: especifica o canal da fonte de tensão a utilizar, sendo que, neste exemplo, é selecionado o canal de 25 V, através do nome “ps/+25V” (podendo também ser utilizado “ps/+6V” para o canal de 6 V);
- *voltage\_level* - linha 4: define o valor da tensão de saída. No caso deste exemplo, 1.0 V. Pode ser, também, atribuída uma variável -  $V_{cc}$  - com base nos valores apresentados na Figura 4.22);
- *current\_limit* - linha 5: é configurado o limite de corrente para 0.5 A;
- *ps.configure\_voltage\_output(channel, voltage\_level, current\_limit)* - linha 7: configura a fonte de alimentação com os parâmetros definidos nas linhas 3 a 5;
- *ps.enable\_all\_outputs(True)* - linha 8: ativa todas as saídas da fonte de alimentação. No entanto, só a saída correspondente ao canal definido na linha 3 será ativada.

No final da execução, os instrumentos e o VB têm de ser libertados, por esta ordem. Isto é feito através do método *instrumento.release()*. Se tal não for feito, o VB e os instrumentos chamados não serão libertados, ficando inacessíveis para futuras chamadas.

De referir que vários instrumentos podem ser chamados simultaneamente e, para cada um deles, assim como para o VB, é criada uma instância para o respectivo objecto, cuja referência corresponde a um endereço de memória. O resultado das instruções na Linha 2 e Linha 1 da Listagem 4.14 é:

- (...)PyVirtualBench object at 0x000001E7DEECAD20.
- (...)PyVirtualBench.PowerSupply object at 0x000001E7DEECB2F0;

No entanto, se um ficheiro for chamado várias vezes (como no caso da experiência da Lei de *Ohm*), uma nova instância do instrumento será sempre criada a cada

chamada. Além disso, se houver várias funções no mesmo ficheiro, os objetos criados (por exemplo, 0x000001E7DEECAD20) devem ser passados entre funções ou, alternativamente, o instrumento deve ser encerrado e reinicializado entre chamadas de funções.

## Gráficos

Os gráficos apresentados são gerados com recurso às bibliotecas *NumPy* [176] e *Matplotlib* [178], amplamente utilizadas em *Python*. A *NumPy* é uma biblioteca fundamental para computação científica, permitindo a manipulação eficiente de *arrays* multidimensionais e fornecendo um vasto conjunto de funções matemáticas, estatísticas e de álgebra linear. No contexto da construção de gráficos, o *NumPy* é essencial na preparação e transformação dos dados a representar. O declive da recta obtido experimentalmente, no estudo da Lei de *Ohm*, representado na Listagem 4.16, foi calculado com recurso à biblioteca *SciPy*, que disponibiliza ferramentas avançadas de análise científica, incluindo ajustamento de curvas e complementa a *NumPy*, sobre a qual está construída.

---

```
1     slope, intercept, r_value, p_value, std_err = stats.
      linregress(voltage_measurements,
                  current_measurements)
```

---

Listagem 4.16: Cálculo do declive da recta

A visualização dos dados é realizada através da *Matplotlib*, uma biblioteca consolidada para criação de gráficos em *Python*. Esta permite gerar representações visuais, como gráficos de linhas, dispersão, histogramas, entre outros. Neste projeto, a *Matplotlib* é utilizada para representar de forma clara e interpretável os dados e resultados obtidos. A Listagem 4.17 ilustra a forma como se gera um gráfico experimental correspondente à Lei de *Ohm*.

---

```
1     plt.plot(voltage_measurements, current_measurements,
                  label='V Vs I', marker = 'o')
```

---

Listagem 4.17: Gráfico Corrente *vs* Tensão

Outro tipo de gráficos que são gerados são os diagramas de Bode, que representam a resposta em frequência dos filtros. A Listagem 4.18 ilustra a forma como são gerados os pontos das frequências para o gráfico de Bode. Neste caso, as frequências são geradas logaritmicamente, utilizando a função *logspace* da biblioteca *NumPy*.

---

```

1     num_points = int(np.log10(stop_freq / start_freq) *
2         points_per_decade)
3     frequencies = np.logspace(np.log10(start_freq), np.
4         log10(stop_freq), num=num_points)

```

---

Listagem 4.18: Filtros - Diagrama de *Bode*

As imagens geradas são gravadas na pasta “webserver/website/static/images/”, tal como indica o exemplo na Listagem 4.19.

---

```

1     plt.savefig("webserver/website/static/images/
2         ohm_graph.png")

```

---

Listagem 4.19: Gravação do gráfico

### 4.2.3 Interface WEB

Do ponto de vista do *software ou IDE*, e considerando a estrutura das experiências em termos de *design* e implementação, o estudo pode ser dividido em duas partes distintas: a primeira, que envolve a Lei de *Ohm*, apresenta um maior grau de complexidade, dado que há um número maior de parâmetros e o utilizador ou aluno tem um controle mais amplo sobre a experiência, já a segunda parte, que aborda os rectificadores e filtros, segue uma estrutura relativamente uniforme, aplicável a todas as experiências.

Assim, nesta secção, apresenta-se um estudo que permite, de forma geral, compreender a componente gráfica - IDE - e a interacção dos utilizadores no controlo das experiências. Sempre que se considerar pertinente, serão destacados aspectos específicos e detalhados.

Comum a todas as experiências é a configuração inicial do VB. Esta configuração é necessária, uma vez que, antes de se iniciar o processo de medições, optou-se por activar a fonte de alimentação que alimenta os relés e os *drivers* e a fonte de alimentação composta pelo LM317 que alimenta os registos de deslocamento, tal como descrito na Secção 4.1.4.

Desta forma, implementou-se nas páginas das experiências, um botão de “OK” - que habilita os selectores e configura a fonte do VB e “STOP/RESET” - que desabilita os selectores e desactiva as fontes enquanto a experiência está inactiva.

A verificação e detecção dos erros, Figura 4.19, acontece quando o utilizador tenta seleccionar um dos parâmetros antes de estes estarem habilitados ou, no caso dos rectificadores e filtros, colocar o valor da frequência fora do valor definido nas instruções. O alerta de erro é implementado nas páginas HTML das experiências, usando o método *alert()* do *JavaScript*.



Figura 4.19: Erro de selecção

Como mencionado na Secção 3.6.2, procurou-se manter a experiência de utilização e navegação o mais simples, prática e intuitiva possível. Além das páginas que compõem a base do site, representadas na Figura 4.16, há ainda a juntar as que permitem ao utilizador interagir com as experiências do LaRE. Especificamente, foram implementadas cinco páginas, correspondentes aos cinco circuitos definidos na Secção 3.2.

A navegação ficou dividida da seguinte forma:

- Autenticação;
  - Página inicial;
  - \* Página introdutória da experiência;
  - Página de controle e realização da experiência.

A página de autenticação, representada na Figura 4.20, está implementada no ficheiro `auth.py`, função `login()` e os formulários estão implementados na página `login.html`, sendo que os dados de `login` e registo são guardados na directória `instance`, ficheiro `database.db`.

Figura 4.20: Página de *login*

Após o registo e *login* bem sucedido, é apresentada a página inicial, onde é feito um pequeno resumo do que é o LaRE e apresentado o menu de escolha das experiências. O menu de separadores verticais foi retirado do site *W3Schools* e modificado de acordo com as necessidades do projecto, tal como pode ser visto na Figura 4.21.



Figura 4.21: Página inicial

O modelo adoptado é uniforme para todas as experiências, apresentando o respectivo esquema e contextualização da actividade. Após a selecção da experiência, tanto as páginas introdutórias como as de controlo seguem a mesma estrutura de menus, mantendo o mesmo padrão de organização e navegação. A título de exemplo, as Figuras 4.22a e 4.22b ilustram, respetivamente, as páginas de introdução e controlo relativas à Lei de Ohm.

(a) Introdução
(b) Controlo

Figura 4.22: [Exemplo] Experiência Lei de Ohm

### Lei de Ohm

Nesta experiência, optou-se por conceder liberdade de escolha e controlo aos utilizadores, o que representou um desafio à implementação. Em vez de permitir apenas a selecção do valor da resistência, com medições realizadas (automaticamente) por *software*, decidiu-se oferecer ao utilizador a possibilidade de escolher o valor da resistência e controlar as medições de tensão e corrente. Do ponto de vista pedagógico,

considerou-se mais benéfico que, neste caso, os alunos verifiquem os valores das grandezas medidas à medida que realizam as medições e avançam na experiência. Além disso, o resultado e o gráfico final serão independentes da ordem pela qual se efectuam as medições.

A Figura 4.23 ilustra a página de controlo da experiência da Lei de *Ohm*.



Figura 4.23: Lei de *Ohm* - Selecção dos parâmetros

A selecção dos valores da resistência e do  $U_{CC}$  é feita através de um formulário de selecção, feito em CSS e *JavaScript*, retirado do site *W3Schools* e modificado de acordo com as necessidades do projecto. Este formulário está implementado na página HTML da experiência. Como se pode observar pela Figura 4.23, a tensão medida no voltímetro é, efectivamente, a tensão aos terminais da fonte de tensão,  $U_{CC}$  e da resistência. Por razões pedagógicas e de coerência com o esquema, optou-se por medir e apresentar a tensão directamente nos terminais da resistência.

Como se observa na Figura 4.23, a tensão indicada pelo voltímetro corresponde à tensão presente entre os terminais da fonte de tensão  $U_{CC}$  e da resistência. Por razões pedagógicas e para manter a coerência com o esquema apresentado, optou-se por medir e representar a tensão diretamente nos terminais da resistência. As medições só serão efectivamente realizadas e os valores apresentados, quando a opção “Medir tensão” e/ou “Medir corrente” forem seleccionadas. O utilizador dispõe de alguma flexibilidade na forma como realiza as medições (cinco de tensão e cinco de corrente), desde que mantenha a mesma ordem. Por exemplo, pode optar por medir todos os valores de tensão - desde o mais baixo ao mais alto - e, de seguida proceder da mesma forma com as medições de corrente. Após a conclusão das dez medições, torna-se possível seleccionar a opção “Gráfico” para visualizar o gráfico da experiência. Caso esta opção seja seleccionada antes da conclusão de todas as medições, o gráfico será apresentado com as medições disponíveis até ao momento, permitindo, ainda assim, a continuação normal da experiência.

Sendo assim, o procedimento para a realização da experiência é o seguinte:

1. Selecionar “OK” para activar a fonte do VB e alimentar os integrados;

2. Escolher o valor da resistência;
3. Escolher o valor de  $U_{CC}$ ;
4. Efectuar cinco medições de tensão (“Medir tensão”) e cinco medições de corrente (“Medir corrente”);
5. Ver o gráfico resultante.
6. Seleccionar “STOP/RESET” para desactivar a fonte do VB.

### Rectificador de meia onda e onda completa

Nestas duas experiências, pretende-se estudar e avaliar a diferença entre estes dois tipos de rectificadores, tanto ao nível da rectificação como da filtragem, nomeadamente no estudo da variação da tensão de *ripple*. A principal diferença, como foi referido na Secção 4.1.6, reside no valor da frequência. Enquanto que no rectificador de meia onda, o sinal de entrada é alimentado pelo gerador de sinal do *virtualbench*, o rectificador de onda completa, devido ao problema de massa já referenciado na Secção 4.1.4, é alimentado através de um transformador 230 V/8 V, com uma frequência de 50 Hz.

A Figura 4.24a ilustra a página de controlo da experiência do rectificador de meia onda. Os utilizadores podem seleccionar quatro valores para o termo RC, sendo que os valores de C estão definidos para 1  $\mu\text{F}$  e 10  $\mu\text{F}$  e a frequência varia entre 5 Hz e 2000 Hz.

 LaRE



(a) Rectificador de meia onda

 LaRE



(b) Rectificador de onda completa

Figura 4.24: Selecção de parâmetros

As medições e construção do gráfico são realizadas por *software*, baseadas nos valores seleccionados pelos utilizadores. Os resultados obtidos podem, então, ser comparados com o gráfico teórico, definido na Figura 3.11 ou pela Equação 3.3. Para o mesmo par RC, os utilizadores podem, ainda, variar a frequência e comparar os resultados obtidos.

Relativamente ao rectificador de onda completa, a Figura 4.24b apresenta a página de controlo da experiência, que é praticamente idêntica à do rectificador de meia onda, excepto pelo facto de a frequência não poder ser ajustada pelo utilizador. Os valores dos condensadores e resistências são os mesmos para ambas as experiências.

O procedimento para a realização da experiência é o seguinte:

1. Seleccionar “OK” para activar a fonte do VB e alimentar os integrados;
2. Escolher o valor da resistência;
3. Escolher o valor do condensador;
4. Efectuar as medições;
5. Ver o gráfico resultante;
6. Seleccionar “STOP/RESET” para desactivar a fonte do VB.

## Filtros

As páginas dos filtros passa-baixo e passa-alto, são em tudo idênticas às dos rectificadores e estão representadas na Figura 4.25a e Figura 4.25b. Os objectivos foram já definidos na Secção ??.

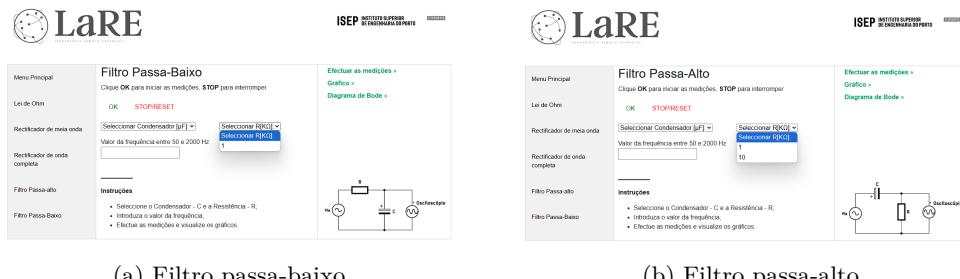


Figura 4.25: Selecção de parâmetros

Consoante a configuração do filtro, o utilizador escolhe entre dois condensadores ou duas resistências, representados no rectângulo tracejado cor-de-laranja, na Figura 4.13. Como já foi referido na Secção 4.1.6, os componentes dentro do rectângulo tracejado vermelho, representados na mesma figura, foram adicionados após a implementação dos rectificadores. Assim, quando se pretende estudar o filtro passa-baixo, o utilizador escolhe entre os dois condensadores (de saída) e se pretender estudar o passa-alto, escolhe entre as duas resistências (de saída). Eventualmente, poder-se-iam ter adicionado mais resistências/condensadores aos filtros, nomeadamente, no rectângulo tracejado vermelho, de forma a permitir uma maior versatilidade no estudo. No entanto, além de não existir o número suficiente de relés em contexto laboratorial, considerou-se que, desta forma, o estudo e os objectivos da experiência, ficariam provados. O intervalo de frequência mantém-se entre os 50 Hz e 2000 Hz. O procedimento para a realização da experiência é o seguinte:

1. Seleccionar “OK” para activar a fonte do VB e alimentar os integrados;

2. Escolher o valor do condensador - passa-baixo;
3. Escolher o valor da resistência - passa-alto;
4. Efectuar as medições;
5. Ver o gráfico resultante;
6. Ver o gráfico resultante da resposta em frequência;
7. Seleccionar “STOP/RESET” para desactivar a fonte do VB.



## Capítulo 5

# Resultados

“Newton’s third law. You’ve got to leave something behind”

— Cooper, *Interstellar*

Os testes foram realizados com o objetivo de validar a implementação do LaRE e verificar se os resultados obtidos correspondem aos esperados. Como forma de uniformizar conceitos, apesar de tanto as experiências desenvolvidas no LaRE como as realizadas na “placa branca” serem reais, designar-se-ão os testes efetuados na “placa branca” como experiências práticas ou valores práticos.

Sendo assim, a confrontação dos resultados obtidos no LaRE foi efetuada com recurso às experiências práticas, ao *MultisimLive* e aos valores teóricos previamente descritos na Secção 4.1.6. Para a análise dos resultados, serão apresentados um ou dois exemplos representativos por tipo de experiência, sendo que os restantes são obtidos de forma análoga. No caso específico da Lei de *Ohm*<sup>1</sup>, há ainda a possibilidade de verificar os valores com recurso a um multímetro de bancada. As experiências podem ser realizadas com diferentes combinações e configurações:

- **Lei de *Ohm*:** três opções de estudo de resistências;
- **Rectificadores:** quatro combinações possíveis de resistências e condensadores para cada rectificador;

---

<sup>1</sup>Uma vez que a versão grátsis *online* do *MultisimLive* não permite realizar a análise continua - *dc sweep* - esta análise fez-se utilizando o CircuitLab

- **Filtros:** duas combinações possíveis de resistências e condensadores para cada filtro.

## 5.1 Lei de Ohm

O objetivo desta experiência é determinar o valor de uma determinada resistência através do cálculo do declive da recta obtida no gráfico da tensão em função da corrente apresentado na Figura 3.8 e, assim, confirmar a Lei de *Ohm*.

### 5.1.1 Resultados experimentais

No LaRE o utilizador efectua cinco medições de cada grandeza (tensão e corrente), para uma das três resistências disponíveis. Na Figura 5.1 está representado o resultado de um par de medições para uma resistência de  $1\text{k}\Omega$ . As restantes medições, em intervalos de 1 V até 5 V, são obtidas de forma análoga.



Figura 5.1: Medição prática

O valor real da resistência, obtida por medição directa, é de  $998\Omega$ , estando dentro da tolerância que é de  $\pm 5\%$ . O erro associado aos instrumentos de medição pode ser considerado desprezável. De acordo com as especificações do modelo *VB-8012*[179], a precisão para medições de tensão em corrente contínua é:

- Escala de 1 V
  - Precisão garantida (1 ano):  $\pm(0.015\% \text{ da leitura} \pm 0.005\% \text{ da faixa})$

Estes valores indicam que, para uma leitura de 1 V, o erro máximo estimado devido à precisão do instrumento seria de, aproximadamente  $\pm 0.0002\text{V}$ , ou seja,  $\pm 0.2\text{mV}$ . Portanto, a diferença entre a tensão fornecida pela fonte — 1 V — e os valores medidos — 0.98 V e 0.97 mA — deve-se, principalmente, à tolerância da resistência utilizada. Assim, conclui-se que os valores medidos se encontram dentro dos limites expectáveis, sendo perfeitamente aceitáveis face às tolerâncias envolvidas.

O gráfico  $U$  vs  $I$ , obtido para esta resistência é apresentado na Figura 5.2. O declive da recta, que representa a resistência, foi calculado e comparado com o valor real da resistência.

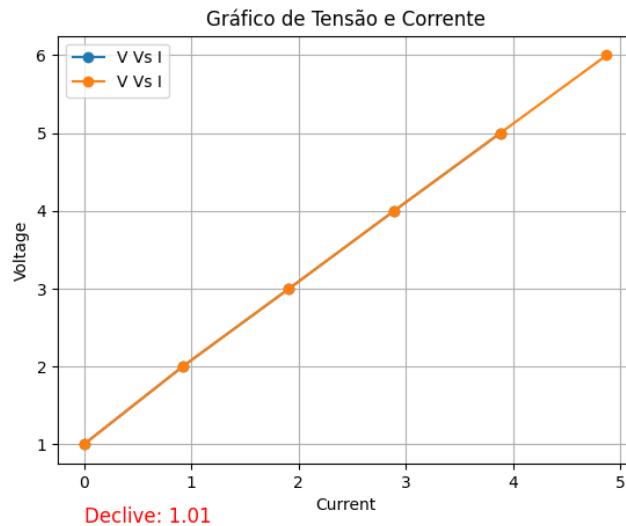


Figura 5.2: Gráfico da Lei de *Ohm* - 1 k $\Omega$  - LaRE

O erro relativo entre o valor teórico e o valor obtido no LaRE, pode ser calculado através da Equação 5.1:

$$\text{Erro Relativo} = \frac{|R_{real} - R_{obtido}|}{R_{real}} \times 100 \quad (5.1)$$

O erro relativo obtido é de, aproximadamente, 1.2%, sendo um valor perfeitamente aceitável.

Os resultados da experiência prática, efectuados com a mesma resistência utilizada no LaRE, estão representados na Tabela 5.1.

Tabela 5.1: Valores prácticos - Lei de *Ohm*

Tensão (V)	Corrente (mA)	Resistência (k $\Omega$ )
0.998	0.998	1
1.99	1.99	1
2.99	2.99	1
3.99	3.99	1
4.99	4.99	1

Estes valores foram obtidos com recurso ao VB e demonstram que o valor da resistência, que corresponde ao declive da recta é constante, o que confirma a Lei de *Ohm*. A partir da Equação 5.1, o valor do erro relativo é inferior a 1%.

Ainda assim e como forma de complementar a análise dos resultados, foram efectuados testes com o *Multisim*. A Figura 5.3 mostra o gráfico obtido.

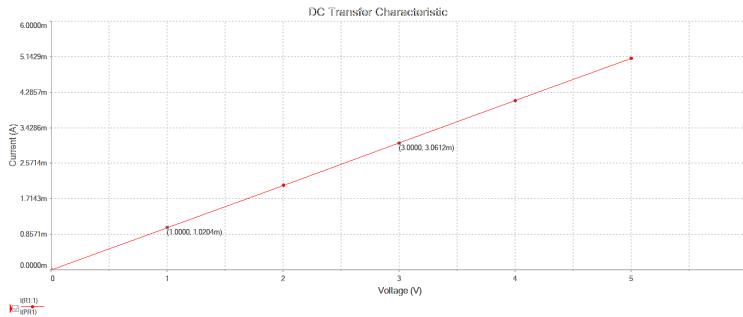


Figura 5.3: Multisim - Lei de *Ohm*

Analizando os valores é possível calcular o declive da recta, dado pela Equação 5.2:

$$m = \frac{y_b - y_a}{x_b - x_a} \quad (5.2)$$

O valor do declive, correspondente à resistência, obtido para estes simulador é de  $m_{\text{multisim}} \approx 0.980$ . Tendo em conta que a escala do eixo do  $y$  está representada em mA, os valores das resistências surgem expressos em  $k\Omega$ , isto é,  $R_{\text{multisim}} = 0.980 k\Omega$ . Assim, aplicando a Equação 5.1 obtém-se um erro relativo inferior a 1%.

Verifica-se, através da análise dos gráficos, que os dados obtidos com o LaRE estão em concordância com os resultados simulados e com os valores práticos.

## 5.2 Rectificadores

As duas experiências foram implementados com o objectivo de estudar e avaliar a tensão de *ripple* nos circuitos rectificadores de meia onda e onda completa, considerando as quatro combinações possíveis dos pares resistência/condensador. As formas de onda esperadas estão representadas na Figura 3.11 e Figura 3.12 sendo os valores teóricos da tensão de *ripple* determinados pela Equação 3.2 e Equação 3.5 - (ou Equação 3.3).

### 5.2.1 Resultados experimentais [meia onda]

Nestas experiências implementadas no LaRE os utilizadores podem escolher uma das quatro combinações de resistência e condensador, fazer variar a frequência entre 50 Hz e 2000 Hz e analisar a diferença de valores na tensão de *ripple*.

Sendo assim, o exemplo da análise será feita para o caso em que  $f = 200$  Hz,  $R = 2.2 k\Omega$  e  $C = 4.7 \mu F$ . Para se determinar qual expressão a usar para calcular a tensão de *ripple*, é necessário verificar a condição descrita na Equação 5.3:

$$2.2 \text{ k}\Omega * 4.7 \mu\text{F} >> \frac{1}{200} \quad (5.3)$$

Uma vez que esta condição não se verifica, a expressão a utilizar para calcular a tensão de *ripple* teórica é dada pela Equação 3.2. O resultado obtido está representado na Equação 5.4:

$$U_{ripple} = 5 \text{ V} \left(1 - e^{-\frac{0.005 \text{ s}}{2.2 \text{ k}\Omega * 4.7 \mu\text{F}}}\right) \approx 1.92 \text{ V} \quad (5.4)$$

A Figura 5.4 apresenta a forma de onda da tensão de saída e o valor da tensão de *ripple* obtido no LaRE, que foi de 1.34 V.

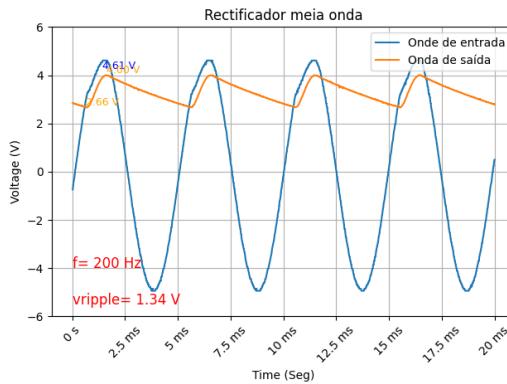


Figura 5.4: Tensão de *ripple* no LaRE- meia onda

As Figuras 5.5a e 5.5b representam, respectivamente, os resultados práticos e simulados. A tensão de *ripple* medida foi de 1.38 V e 1.51 V.

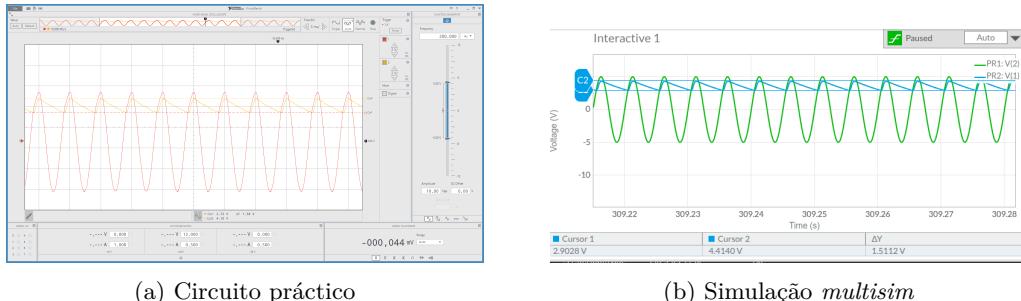


Figura 5.5: Valores experimentais e simulados - tensão de *ripple*

Os erros relativos obtidos foram de 30.21 % para o LaRE, 28.13 % e 21.35 %, para os valores medidos no circuito prácticos e simulados. Esta diferença em relação ao valor calculado teoricamente deve-se a diversos fatores, entre os quais se destaca a simplificação inherente à expressão analítica utilizada, que assume componentes ideais e condições perfeitas — como uma resistência constante e um condensador sem perdas [148]. A simulação, embora mais próxima da realidade, utiliza também componentes ideais por omissão, como é o caso do condensador sem resistência série

(ESR). Ainda assim, considera fenómenos como as quedas de tensão nos diodos e os efeitos transitórios do circuito. A maior discrepancia nos valores experimentais pode ser justificada por erros de leitura e pelas elevadas tolerâncias típicas dos condensadores electrolíticos. Assim, as diferenças observadas são compreensíveis e enquadram-se dentro do esperado.

### 5.2.2 Resultados experimentais [onda completa]

## 5.3 Filtros

Estas duas experiências foram realizadas com o objetivo de estudar a resposta em frequência dos filtros  $RC$ , nomeadamente nas configurações passa-alto e passa-baixo, resultantes das diferentes configurações entre as resistências e os condensadores. Os utilizadores podem variar a frequência entre 50 Hz e 80 000 Hz, analisando o comportamento da tensão de saída em função da frequência. A frequência de corte é calculada através da Equação 3.6, sendo posteriormente comparada com os valores obtidos experimentalmente no diagrama de *Bode*. Adicionalmente, a atenuação da tensão de saída pode ser determinada através da Equação 3.7, ou expressa em dB utilizando a Equação 3.8.

### 5.3.1 Resultados experimentais [filtro passa-alto]

O circuito analisado será o filtro passa-alto, representado na Figura 3.13b e configurado com  $R = 2.2\text{k}\Omega$  e  $C = 10\text{nF}$ . Os gráficos que se esperam obter estão representados na Figura 3.14b. Para os restantes valores e configurações, os resultados poderão ser obtidos de forma análoga.

Para estes valores, a frequência de corte teórica é dada pela Equação 3.6, sendo que o valor obtido é de 7234.32 Hz.

Os valores experimentais obtidos com o LaRE relativamente à análise do diagrama de *Bode*, estão representados na Figura 5.6, sendo que o valor aproximado da frequência de corte é de 7071.1 Hz. O erro relativo é, aproximadamente, 2.26 %. Este valor é perfeitamente aceitável, tendo em conta a tolerância dos componentes utilizados, os erros de medição e aproximações.

Os resultados complementares obtidos com base na simulação efectuada no *multisim*, já que não é possível gerar o diagrama de *Bode* do circuito práctico, representados na Figura 5.7, mostram que a frequência de corte é de 7282.4 Hz. O erro relativo é de 0.67 %, o que é prácticamente desprezável.

O valor da frequência de corte também pode ser obtido através da análise dos gráficos que representam a relação entre a tensão de saída e a tensão de entrada.

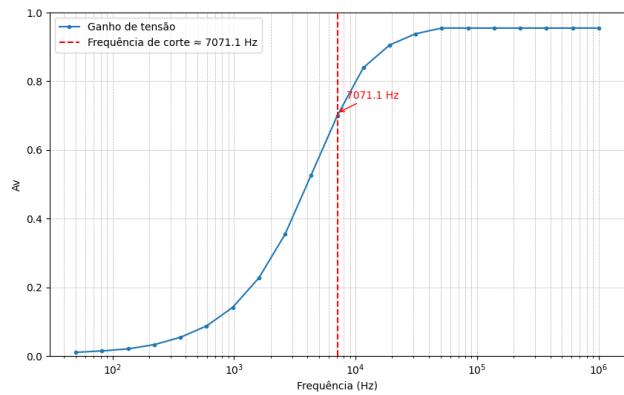


Figura 5.6: Frequência de corte - filtro passa-alto - Diagrama de *Bode*

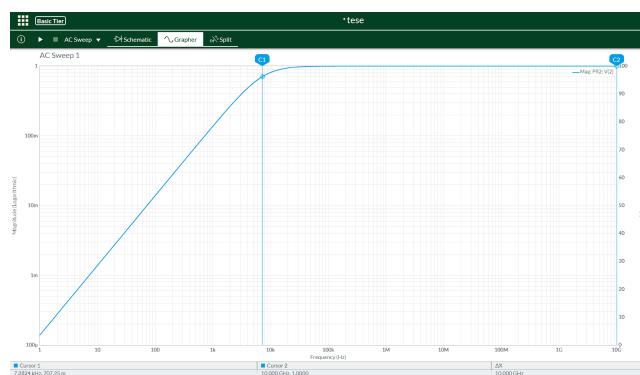


Figura 5.7: Frequência de corte - *multisim* - Diagrama de *Bode*

Como se pode ver pela análise da Figura 5.8, o ganho a altas frequências é praticamente unitário, estando em consonância com os valores obtidos no diagrama de *Bode*, representado na Figura 5.6.

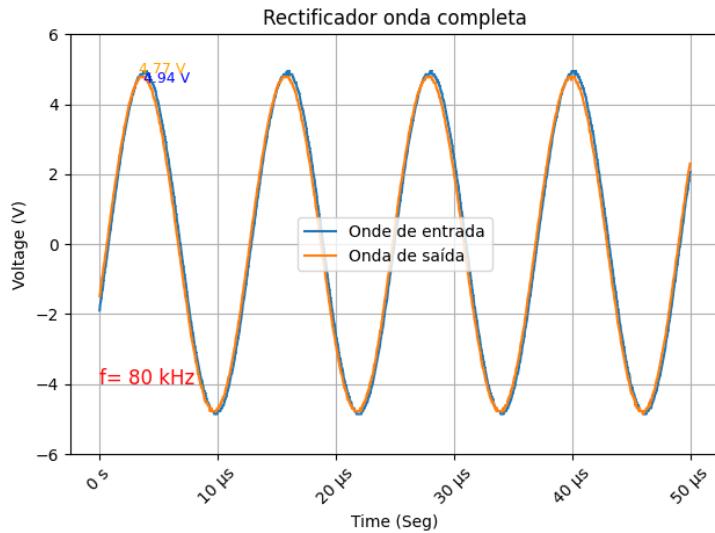


Figura 5.8:  $v_{out}$  vs  $v_{in}$  experimental LaRE - TEMPORÁRIO

Comparando estes valores com os obtidos através do circuito prático, representados na Figura 5.9a e Figura 5.9b do circuito simulado no *multisim* verifica-se que os valores obtidos são praticamente iguais, podendo ser considerado, com segurança, que em ambos os casos o ganho de tensão é unitário.

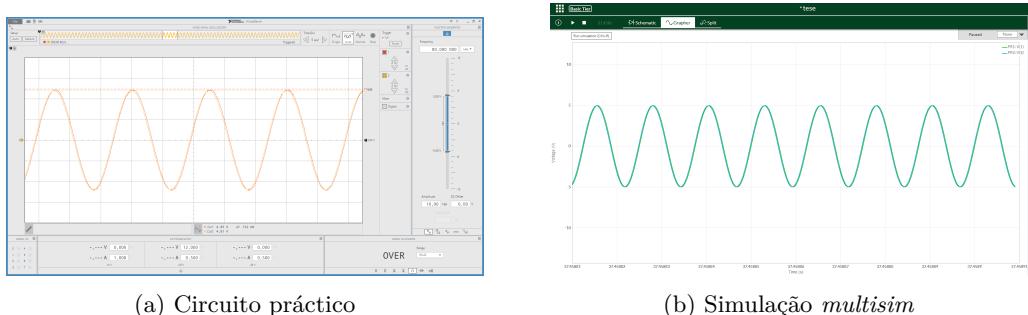


Figura 5.9: Valores práticos e simulados -  $v_{out}$  vs  $v_{in}$

Relativamente à análise da frequência de corte, representado na Figura 5.10, o valor obtido para o LaRE é de 7230 Hz, o que corresponde a um erro relativo menor que 1%.

O circuito prático apresenta uma frequência de corte de 7000 Hz, com um erro relativo de 3.24 %, enquanto a simulação no *Multisim* apresenta 7300 Hz, com um erro ainda mais reduzido de 0.9 %. Estes desvios podem ser considerados baixos e encontram-se dentro de margens aceitáveis, tendo em conta as limitações práticas do circuito real, como tolerâncias dos componentes, ruído, e eventuais imprecisões

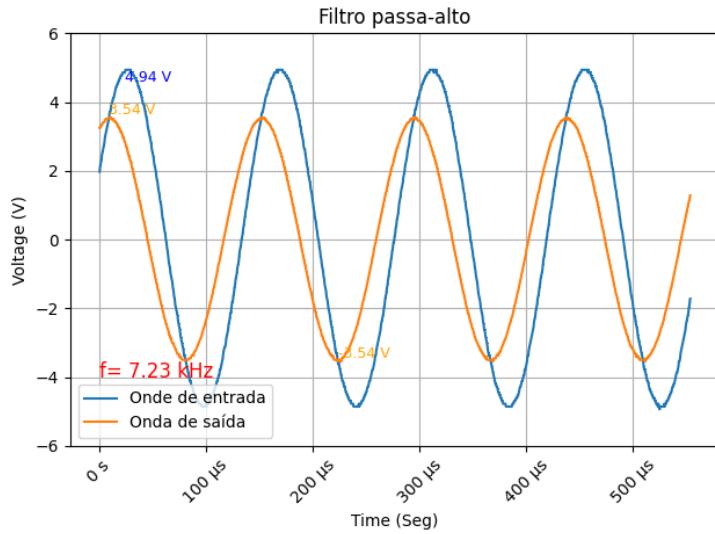
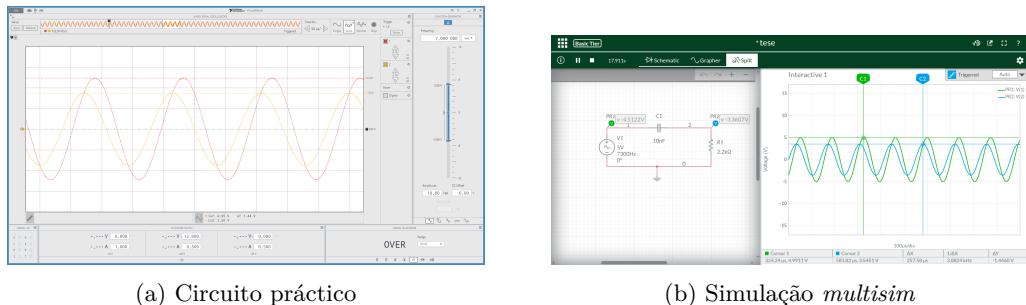


Figura 5.10: Frequência de corte - LaRE

nas medições. O valor obtido na simulação, por outro lado, está mais próximo do valor teórico, o que é expectável dado que os modelos usados assumem, geralmente, componentes ideais ou com parâmetros controlados.

Figura 5.11: Valores experimentais reais e simulados - frequência de corte -  $v_{out}$  vs  $v_{in}$ 

## 5.4 Limitações

## 5.5 Melhoramentos



# Referências

- [1] “What is arduino? | arduino.” <https://www.arduino.cc/en/Guide/Introduction>. (Accessed on 06/30/2024). [Citado na página xvii]
- [2] J. Matias, *Cursos Profissionais - Eletricidade e Eletrónica, Vol. 3.* Didáctica Editora, 2011. [Citado nas páginas xvii e xix]
- [3] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, “Industry 4.0 and industry 5.0— inception, conception and perception,” *Journal of Manufacturing Systems*, vol. 61, pp. 530–535, 2021. [Citado na página xviii]
- [4] “The python tutorial — python 3.12.4 documentation.” <https://docs.python.org/3/tutorial/index.html>. (Accessed on 07/26/2024). [Citado nas páginas xix e 68]
- [5] “Teach, learn, and make with the raspberry pi foundation.” <https://www.raspberrypi.org/>. (Accessed on 06/30/2024). [Citado na página xix]
- [6] “Raspberry pi.” <https://www.raspberrypi.com/>. (Accessed on 06/30/2024). [Citado na página xix]
- [7] “W3c.” <https://www.w3.org/>. (Accessed on 08/03/2024). [Citado na página xx]
- [8] A. Hofstein, *The Role of Laboratory in Science Teaching and Learning*, pp. 357–368. 01 2017. [Citado nas páginas 1, 16 e 17]
- [9] J. R. Brinson, “Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research,” *Computers & Education*, vol. 87, pp. 218–237, 2015. [Citado nas páginas 1, 17, 19, 21 e 31]
- [10] L. Feisel and A. Rosa, “The role of the laboratory in undergraduate engineering education,” *Journal of Engineering Education*, vol. 94, 01 2005. [Citado na página 1]
- [11] J. E. Corder, S. K. Esche, C. Chassapis, J. Ma, and J. V. Nickerson, “Process and learning outcomes from remotely-operated, simulated, and hands-on student laboratories,” *Computers & Education*, vol. 57, no. 3, pp. 2054–2067, 2011. [Citado nas páginas 2 e 31]

- [12] “Virtual instruments systems in reality - visir+ - home.” <https://www2.isep.ipp.pt/visir/>. (Accessed on 06/29/2024). [Citado na página 2]
- [13] J. Cross, “An informal history of elearning,” *on The Horizon*, vol. 12, pp. 103–110, 09 2004. [Citado nas páginas 6 e 11]
- [14] OECD, *Using Digital Technologies for Early Education during COVID-19*. OECD, 2021. [Citado na página 6]
- [15] “Resolução do Conselho de Ministros nº 30/2020.” [Citado na página 6]
- [16] “Capacitação digital das escolas.” <https://digital.dge.mec.pt/>. (Accessed on 02/26/2023). [Citado na página 6]
- [17] A. Sepúlveda, “The digital transformation of education: Connecting schools, empowering learners - broadband commission.” <https://broadbandcommission.org/publication/the-digital-transformation-of-education/>. (Accessed on 02/26/2023). [Citado na página 6]
- [18] “Digital Economy and Society Index (DESI) 2020 | Shaping Europe’s digital future,” June 2020. [Citado nas páginas 6, 7 e 10]
- [19] ILO, “Youth & COVID-19: Impacts on jobs, education, rights and mental well-being,” report, European Commission, Aug. 2020. [Citado na página 7]
- [20] “290 million students out of school due to COVID-19: UNESCO releases first global numbers and mobilizes response | UNESCO.” [www.unesco.org/en/articles/290-million-students-out-school-due-covid-19-unesco-releases-first-global-numbers-and-mobilizes](http://www.unesco.org/en/articles/290-million-students-out-school-due-covid-19-unesco-releases-first-global-numbers-and-mobilizes). [Citado na página 8]
- [21] SAPO, “Entrega de computadores a alunos e professores tem-se ‘revelado muito difícil de cumprir’.” . [Citado na página 8]
- [22] OECD, *The State of School Education*. OECD, 2021. [Citado nas páginas 8 e 9]
- [23] “Covid-19 teve impacto negativo na educação de 70% dos jovens.” [www.dn.pt/mundo/covid-19-teve-impacto-negativo-na-educacao-de-70-dos-jovens-12513822.html](http://www.dn.pt/mundo/covid-19-teve-impacto-negativo-na-educacao-de-70-dos-jovens-12513822.html), aug 2020. (Accessed on 03/08/2023). [Citado na página 9]
- [24] R. Steve, “What is iot ? everything you need to know about the internet of things right now.,” 2018. [Citado na página 9]
- [25] “The state of education – one year into covid - oecd education and skills today.” <https://oecdedutoday.com/state-of-education-one-year-into-covid/>. (Accessed on 02/26/2023). [Citado na página 9]

- [26] “Digital education initiatives | european education area.” <https://education.ec.europa.eu/focus-topics/digital-education/about-digital-education>. (Accessed on 03/04/2023). [Citado na página 10]
- [27] “Transição digital - prr - recuperar portugal.” <https://recuperarportugal.gov.pt/transicao-digital/>. (Accessed on 06/19/2024). [Citado na página 10]
- [28] “Home - incode 2030.” <https://www.incode2030.gov.pt/>. (Accessed on 06/19/2024). [Citado na página 10]
- [29] “Programa – escola digital.” <https://escoladigital.aerp.pt/programa/>. (Accessed on 06/19/2024). [Citado na página 10]
- [30] Y. Fan, A. Evangelista, and V. Indumathi, “Evaluation of remote or virtual laboratories in e-learning engineering courses,” in *2021 IEEE Global Engineering Education Conference (EDUCON)*, pp. 136–143, 2021. [Citado nas páginas 11 e 32]
- [31] J. Magano, A. V. Castro, and C. Vaz de Carvalho, “O e-learning no ensino superior: um caso de estudo,” *Educação, Formação & Tecnologias*, vol. 1, p. 79–92, Mai. 2008. [Citado na página 11]
- [32] M. J. Gomes, “E-learning: reflexões em torno do conceito,” in *E-learning: reflexões em torno do conceito*, Universidade do Minho. Centro de Competência do Projecto Nónio Século XXI, May 2005. Accepted: 2005-09-13T16:07:10Z. [Citado nas páginas 11 e 12]
- [33] “Convite à apresentação de propostas dg eac/46/02 — acções preparatórias e inovadoras 2002/b — e-learning,” Jul 2002. [Citado na página 12]
- [34] UNESCO, “Glossary article: Stem.” <https://unevoc.unesco.org/home/Glossary+article:+STEM>. (Accessed on 03/14/2023). [Citado na página 13]
- [35] J. Rothwell, “Thehiddenstemeconomy610.pdf.” <https://www.brookings.edu/wp-content/uploads/2016/06/TheHiddenSTEMEconomy610.pdf>. (Accessed on 03/14/2023). [Citado na página 13]
- [36] M. Sanders, “Ttt+stem+article\_1.pdf.” [http://esdstem.pbworks.com/f/TTT+STEM+Article\\_1.pdf](http://esdstem.pbworks.com/f/TTT+STEM+Article_1.pdf). (Accessed on 03/14/2023). [Citado na página 13]
- [37] OECD, *PISA 2018 Results (Volume V)*. 2020. [Citado na página 14]
- [38] “O que falta a portugal para ter um sistema de educação de sucesso?” <https://www.dn.pt/pais/o-que-falta-a-portugal-para-ter-um-sistema-de-educacao-de-sucesso-12777334.html>. (Accessed on 04/02/2023). [Citado na página 14]

- [39] “Falta de investimento na educação põe escola pública em risco: problemas rebentam um pouco por todo o lado e são cada vez mais graves! - fenprof.” <https://www.fenprof.pt/falta-de-investimento-na-educacao-poe-escola-publica-em-risco-problemas-rebentam-um-pouco-por-todo-o-lado-e-sao-cada-vez-mais-graves>. (Accessed on 04/02/2023). [Citado nas páginas 14 e 23]
- [40] “O (des)investimento na educação em portugal.” <https://www.msedu.pt/2020/12/o-desinvestimento-na-educacao-em.html>. (Accessed on 04/02/2023). [Citado nas páginas 14 e 23]
- [41] “Edustat - portugal gasta menos em educação do que há 20 anos.” <https://www.edustat.pt/detalhes-infostat?ID=14>. (Accessed on 04/02/2023). [Citado nas páginas 14 e 23]
- [42] “Portugal, o país menos educado da união europeia – observador.” <https://observador.pt/opiniao/portugal-o-pais-menos-educado-da-uniao-europeia/>. (Accessed on 04/02/2023). [Citado na página 14]
- [43] P. Ferreira, “Qualificação, steam, género.” <https://www.jn.pt/opiniao/convidados/qualificacao-steam-genero--14523190.html>. (Accessed on 04/04/2023). [Citado na página 14]
- [44] “Pandemia acentuou desigualdades na saúde, educação e no trabalho, confirma estudo,” July 2024. [Online; accessed 3. Jul. 2024]. [Citado na página 14]
- [45] R. B. Correia, “Ofap-efeitos-da-pandemia-digital-14.pdf.” <http://ofap.ics.ulisboa.pt/wp-content/uploads/2020/12/OFAP-Efeitos-da-pandemia-Digital-14.pdf>, 2020. (Accessed on 06/30/2024). [Citado na página 14]
- [46] A. Oladejo and A. Ebisin, “Virtual laboratory: An alternative laboratory for science teaching and learning,” *Journal of Pure & Applied Sciences*, vol. 3, pp. 82 – 91, 10 2021. [Citado nas páginas 15 e 21]
- [47] “Uma dificuldade nas nossas universidades é a falta de financiamento.” <https://www.dn.pt/vida-e-futuro/uma-dificuldade-nas-nossas-universidades-e-a-falta-de-financiamento-9574558.html>. (Accessed on 03/28/2023). [Citado na página 15]
- [48] “Financiamento do ensino profissional - fenprof.” <https://www.fenprof.pt/financiamento-do-ensino-profissional>. (Accessed on 03/28/2023). [Citado na página 15]

- [49] “Educação: financiar para uma melhor aprendizagem – observador.” <https://observador.pt/opiniao/educacao-financiar-para-uma-melhor-aprendizagem/>. (Accessed on 03/28/2023). [Citado na página 15]
- [50] L. Cerdeira, B. Cabrito, M. d. L. M. Taylor, and R. Gomes, “A fuga de cérebros em Portugal: hipóteses explicativas,” *Revista Brasileira de Política e Administração da Educação*, vol. 31, no. 2, pp. 409–418, 2020. [Citado na página 16]
- [51] ANQEP, “ANQEP - Cursos Profissionais,” July 2024. [Online; accessed 4. Jul. 2024]. [Citado na página 18]
- [52] C. Viegas, A. Pavani, N. Lima, A. Marques, I. Pozzo, E. Dobboletta, V. Atencia, D. Barreto, F. Calliari, A. Fidalgo, D. Lima, G. Temporão, and G. Alves, “Impact of a remote lab on teaching practices and student learning,” *Computers & Education*, vol. 126, pp. 201–216, 2018. [Citado nas páginas 18, 19 e 31]
- [53] L. F. Zapata-Rivera and M. Larrondo Petrie, “Models of collaborative remote laboratories and integration with learning environments,” *International Journal of Online Engineering (iJOE)*, vol. 12, pp. 14–21, 09 2016. [Citado na página 18]
- [54] R. Heradio, L. de la Torre, and S. Dormido, “Virtual and remote labs in control education: A survey,” *Annual Reviews in Control*, vol. 42, pp. 1–10, 2016. [Citado nas páginas ix, 18, 19, 20, 25 e 33]
- [55] V. Potkonjak, M. Gardner, V. Callaghan, P. Mattila, C. Guetl, V. M. Petrović, and K. Jovanović, “Virtual laboratories for education in science, technology, and engineering: A review,” *Computers & Education*, vol. 95, pp. 309–327, 2016. [Citado nas páginas 18, 25, 27 e 28]
- [56] A. Hofstein and V. Lunetta, “The laboratory in science education: Foundations for the twenty-first century,” *Science Education*, vol. 88, pp. 28 – 54, 01 2004. [Citado nas páginas 18 e 19]
- [57] D. Zutin, M. Auer, C. Maier, and M. Niederstatter, “Lab2go — a repository to locate educational online laboratories,” pp. 1741 – 1746, 05 2010. Citado Zapata-Rivera, Luis Felipe and Larrondo Petrie, María. [Citado nas páginas ix e 19]
- [58] L. F. Zapata-Rivera and M. Larrondo Petrie, “Models of collaborative remote laboratories and integration with learning environments,” *International Journal of Online Engineering (iJOE)*, vol. 12, pp. 14–21, 09 2016. [Citado nas páginas 19 e 20]

- [59] E. EMaster, “Por que os laboratórios virtuais são essenciais na educação moderna?,” *EMaster Cloud e Security*, May 2024. [Citado na página 21]
- [60] A. Faiña, “Learning hands-on electronics from home: A simulator for fritzing,” 06 2022. [Citado na página 21]
- [61] A. Knörig, R. Wettach, and J. Cohen, “Fritzing: a tool for advancing electronic prototyping for designers,” *ResearchGate*, pp. 351–358, Feb. 2009. [Citado na página 21]
- [62] C. Chan and W. Fok, “Evaluating learning experiences in virtual laboratory training through student perceptions: a case study in electrical and electronic engineering at the university of hong kong.” <https://hub.hku.hk/bitstream/10722/124702/1/content.pdf>. (Accessed on 04/11/2023). [Citado na página 22]
- [63] B. Heying, D. Kejie, and J. Li, “Application multisim to virtual laboratory for experiment teaching,” in *2010 International Conference on Computational Intelligence and Software Engineering*, pp. 1–4, Dec 2010. [Citado na página 22]
- [64] P. Umenne and T. Hlalele, “Evaluation of the effectiveness of virtual laboratory’s for electronics in the open distance learning context,” pp. 1–5, 08 2020. [Citado na página 22]
- [65] “Multisim Live Online Circuit Simulator,” July 2024. [Online; accessed 6. Jul. 2024]. [Citado nas páginas 22, 25 e 28]
- [66] I. Gustavsson, G. Alves, R. Costa, K. Nilsson, J. Zackrisson, U. Hernandez, and J. Garcia-Zubia, “The visir open lab platform 5.0 - an architecture for a federation of remote laboratories,” 07 2011. [Citado na página 23]
- [67] “Redução do número de alunos por turma nas escolas públicas - xxi governo - república portuguesa.” <https://www.portugal.gov.pt/pt/gc21/comunicacao/comunicado?i=reducao-do-numero-de-alunos-por-turma-nas-escolas-publicas>. (Accessed on 04/07/2023). [Citado na página 23]
- [68] “Falta de docentes: há três disciplinas em que a situação é mais preocupante - sic notícias.” <https://sicnoticias.pt/pais/2022-04-18-falta-de-docentes-ha-tres-disciplinas-em-que-a-situacao-e-mais-preocupante>. (Accessed on 04/07/2023). [Citado na página 23]
- [69] “RoboParty,” July 2024. [Online; accessed 5. Jul. 2024]. [Citado na página 24]
- [70] “Festival Nacional de Robótica,” July 2024. [Online; accessed 5. Jul. 2024]. [Citado na página 24]

- [71] “11.ª edição do CanSat Portugal,” July 2024. [Online; accessed 5. Jul. 2024]. [Citado na página 24]
- [72] R. Yoder, “An arduino-based alternative to the traditional electronics laboratory,” pp. 107–110, 11 2015. [Citado na página 24]
- [73] O. H. Graven and J. Bjørk, “The use of an arduino pocket lab to increase motivation in electrical engineering students for programming,” in *2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pp. 239–243, 2016. [Citado na página 24]
- [74] P. Plaza, E. Sancristobal, G. Carro, M. Blazquez, F. García-Loro, S. Martin, C. Perez, and M. Castro, “Arduino as an educational tool to introduce robotics,” in *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pp. 1–8, 2018. [Citado na página 24]
- [75] I. Marzoli, N. Rizza, A. Saltarelli, and E. Sampaolesi, *Arduino: From Physics to Robotics*. Jan. 2021. [Citado na página 24]
- [76] “Tinkercad - From mind to design in minutes,” July 2024. [Online; accessed 8. Jul. 2024]. [Citado na página 25]
- [77] Fritzing, “Download Fritzing,” July 2024. [Online; accessed 7. Jul. 2024]. [Citado na página 25]
- [78] I. Agafonov, “Web browser electric circuit simulators for education,” 05 2018. [Citado na página 25]
- [79] R. Scheckler, “Virtual labs: A substitute for traditional labs?,” *The International journal of developmental biology*, vol. 47, pp. 231–6, 02 2003. [Citado na página 27]
- [80] T. Lynch and I. Ghergulescu, “Review of virtual labs as the emerging technologies for teaching stem subjects,” pp. 6082–6091, 03 2017. [Citado na página 27]
- [81] “Blog emaster | por que os laboratórios virtuais são essenciais na educação.” <https://emaster.cloud/educacao/laboratorios-virtuais-na-educacao/>. (Accessed on 07/06/2024). [Citado na página 27]
- [82] “Advantages and Disadvantages of Virtual Laboratory,” June 2023. [Online; accessed 6. Jul. 2024]. [Citado nas páginas 27 e 28]
- [83] B. Gherasim, “Virtual lab – advantages and disadvantages – edict,” 12 2023. [Citado na página 28]

- [84] Dr. I. Ghergulescu, “Review Of Virtual Labs As The Emerging Technologies For Teaching Stem Subjects - Adaptemy,” *Adaptemy*, Feb. 2019. [Citado na página 28]
- [85] S. Mandal, “Some important simulation software tools for a student of electronics engineering,” *Global Journal on Advancement in Engineering and Science*, vol. 3, pp. 1–8, 01 2017. [Citado na página 28]
- [86] B. Heying, D. Kejie, and J. Li, “Application multisim to virtual laboratory for experiment teaching,” in *2010 International Conference on Computational Intelligence and Software Engineering*, pp. 1–4, Dec 2010. [Citado na página 29]
- [87] P. Falstad, “Circuit Simulator Applet,” Feb. 2024. [Online; accessed 7. Jul. 2024]. [Citado nas páginas ix, 29 e 30]
- [88] “Licensing,” Mar. 2011. [Online; accessed 7. Jul. 2024]. [Citado na página 29]
- [89] “Math, Physics, and Engineering Applets,” Feb. 2023. [Online; accessed 8. Jul. 2024]. [Citado na página 29]
- [90] J. B. da Silva, L. R. Machado, S. M. Sommer Bilessimo, and I. N. da Silva, “Remote teaching of electrical circuits: proposal for the use of online laboratories in secondary education,” in *2021 World Engineering Education Forum/Global Engineering Deans Council (WEEF/GEDC)*, pp. 594–600, Nov 2021. [Citado na página 29]
- [91] “Wokwi - Online ESP32, STM32, Arduino Simulator,” July 2024. [Online; accessed 8. Jul. 2024]. [Citado na página 30]
- [92] M. A. Marques, M. C. Viegas, M. C. Costa-Lobo, A. V. Fidalgo, G. R. Alves, J. S. Rocha, and I. Gustavsson, “How remote labs impact on course outcomes: Various practices using visir,” *IEEE Transactions on Education*, vol. 57, no. 3, pp. 151–159, 2014. [Citado nas páginas 30, 31, 32 e 40]
- [93] A. N. S. Karmakar, N. K. Roy and P. Kumbhakar, “A remotely operated high voltage laboratory for impulse voltage testing.” [http://wiete.com.au/journals/WTE&TE/Pages/Vol.7,%20No.1%20\(2009\)/3-Nafalski-17.pdf](http://wiete.com.au/journals/WTE&TE/Pages/Vol.7,%20No.1%20(2009)/3-Nafalski-17.pdf). (Accessed on 04/10/2023). [Citado nas páginas 31 e 32]
- [94] W. de Mel and S. T.G.A.A., “Extending the boundaries of remote laboratory by providing hands on experience,” 09 2017. [Citado na página 31]
- [95] J. E. Corder, J. V. Nickerson, S. K. Esche, C. Chassapis, S. Im, and J. Ma, “Constructing reality: A study of remote, hands-on, and simulated laboratories,” *ACM Trans. Comput.-Hum. Interact.*, vol. 14, p. 7–es, aug 2007. [Citado na página 31]

- [96] S. Kocijancic and C. O’Sullivan, “Real or virtual laboratories in science teaching - is this actually a dilemma?,” *Informatics in Education*, vol. 3, pp. 239–250, 10 2004. [Citado na página 31]
- [97] C. Tsipouridis, D. Vavougos, M. Batsila, and G. Ioannidis, *The Timeless Controversy Between Virtual and Real Laboratories in Science Education—“And the Winner Is...”*, pp. 620–631. 01 2019. [Citado na página 31]
- [98] M. Tawfik, E. San Cristobal, S. Martín, C. Gil, A. Pesquera, P. Losada, G. Díaz, J. Peire, M. Castro, J. García-Zubia, U. Hernández, P. Orduña, I. Angulo, M. C. Costa Lobo, M. A. Marques, M. C. Viegas, and G. R. Alves, “Visir deployment in undergraduate engineering practices,” in *2011 First Global Online Laboratory Consortium Remote Laboratories Workshop*, pp. 1–7, 2011. [Citado na página 32]
- [99] A. Nafalski, O. Göl, Z. Nedic, J. Machotka, J. M. M. Ferreira, and I. Gustavsson, “Experiences with remote laboratories,” 2010. [Citado na página 32]
- [100] M. E. Auer and C. Gravier, “Guest editorial: The many facets of remote laboratories in online engineering education,” *IEEE Transactions on Learning Technologies*, vol. 2, no. 4, pp. 260–262, 2009. [Citado na página 32]
- [101] X. Chen, Y. Zhang, L. Kehinde, and D. Olowokere, “Developing virtual and remote undergraduate laboratory for engineering technology,” 06 2010. [Citado na página 32]
- [102] J. Simão, J. de Lima, W. Rochadel, and J. B. da Silva, “Remote labs in developing countries an experience in brazilian public education,” in *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, pp. 99–105, 2014. [Citado na página 32]
- [103] M. Kalúz, J. García-Zubía, M. Fikar, and L. Čirka, “A flexible and configurable architecture for automatic control remote laboratories,” *IEEE Transactions on Learning Technologies*, vol. 8, pp. 299–310, July 2015. [Citado na página 34]
- [104] E. Fabregas, G. Farias, S. Dormido-Canto, S. Dormido, and F. Esquembre, “Developing a remote laboratory for engineering education,” *Computers & Education*, vol. 57, no. 2, pp. 1686–1697, 2011. [Citado na página 35]
- [105] C. Lazar and S. Carari, “A remote-control engineering laboratory,” *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 2368–2375, June 2008. [Citado na página 35]

- [106] R. Dormido, H. Vargas, N. Duro, J. Sánchez, S. Dormido-Canto, G. Farias, F. Esquembre, and S. Dormido, “Development of a web-based control laboratory for automation technicians: The three-tank system,” *IEEE Transactions on Education*, vol. 51, pp. 35–44, Feb 2008. [Citado na página 35]
- [107] “View of Paradigms in Remote Experimentation,” July 2024. [Online; accessed 10. Jul. 2024]. [Citado nas páginas ix e 36]
- [108] P. Jacko, B. Matej, I. Kováčová, J. Molnár, T. Vince, J. Dziak, B. Fecko, v. Gans, and D. Kováč, “Remote iot education laboratory for microcontrollers based on the stm32 chips,” *Sensors*, vol. 22, no. 4, 2022. [Citado na página 36]
- [109] “Vocabulary — gateway4labs 0.1 documentation.” <https://gateway4labs.readthedocs.io/en/latest/vocabulary.html>. (Accessed on 07/12/2024). [Citado na página 36]
- [110] D. G. Zutin and M. E. Auer, “Work in progress — integrating educational online lab platforms around the ilab shared architecture,” in *2011 Frontiers in Education Conference (FIE)*, pp. F1G–1–F1G–3, Oct 2011. [Citado na página 37]
- [111] V. J. Harward, J. A. del Alamo, S. R. Lerman, P. H. Bailey, J. Carpenter, K. DeLong, C. Felknor, J. Hardison, B. Harrison, I. Jabbour, P. D. Long, T. Mao, L. Naamani, J. Northridge, M. Schulz, D. Talavera, C. Varadharajan, S. Wang, K. Yehia, R. Zbib, and D. Zych, “The ilab shared architecture: A web services infrastructure to build communities of internet accessible laboratories,” *Proceedings of the IEEE*, vol. 96, pp. 931–950, June 2008. [Citado na página 37]
- [112] “Developers - ilabs dev - mit wiki service.” <https://wikis.mit.edu/confluence/display/ILAB2/Developers>. (Accessed on 07/12/2024). [Citado nas páginas ix e 37]
- [113] D. G. Zutin, M. E. Auer, and I. Gustavsson, “A visir lab server for the ilab shared architecture,” in *2011 IEEE Global Engineering Education Conference (EDUCON)*, pp. 30–33, April 2011. [Citado na página 37]
- [114] “Welcome to weblab-deusto’s documentation! — weblab-deusto 5.0 documentation.” <https://weblabdeusto.readthedocs.io/en/latest/index.html>. (Accessed on 07/12/2024). [Citado na página 37]
- [115] P. Orduña, D. G. Zutin, S. Govaerts, I. L. Zorrozua, P. H. Bailey, E. San-cristobal, C. Salzmann, L. Rodriguez-Gil, K. DeLong, D. Gillet, M. Castro, D. López-de Ipiña, and J. García-Zubia, “An extensible architecture for the

- integration of remote and virtual laboratories in public learning tools,” *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 10, pp. 223–233, Nov 2015. [Citado na página 38]
- [116] “Welcome to gateway4labs’s documentation! — gateway4labs 0.1 documentation.” <https://gateway4labs.readthedocs.io/en/latest/index.html>. (Accessed on 07/12/2024). [Citado na página 38]
- [117] G. R. Alves, J. B. Silva, A. Pavani, A. d. Costa, D. Cardoso, D. A. Lima, E. Takahashi, G. R. Roque, G. Temporão, H. Néri, J. Pereira, L. Mendes, R. Moura, R. Gedraite, S. Bilessimo, and T. Uhlmann, *Laboratórios Remotos no Ensino de Engenharia*. Brazil: ABENGE – Associação Brasileira de Educação em Engenharia, Sept. 2017. [Citado na página 39]
- [118] M. Tawfik, E. Sanchristobal, S. Martin, C. Gil, A. Pesquera, P. Losada, G. Diaz, J. Peire, M. Castro, J. García-Zubia, U. Hernández, P. Orduña, I. Angulo, M. C. Costa Lobo, M. Marques, M. C. Viegas, and G. R. Alves, “Visir: Experiences and challenges,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 8, p. pp. 25–32, Feb. 2012. [Citado na página 40]
- [119] “Openlabs - electrolab.” <http://openlabs.bth.se/index.php?page=ElectroLab>. (Accessed on 04/13/2023). [Citado na página 40]
- [120] F. Garcia-Loro, E. S. Cristobal, G. Diaz, A. Macho, P. Baizan, M. Blazquez, M. Castro, P. Plaza, P. Orduña, M. Auer, W. Kulesza, I. Gustavsson, K. Nilsson, A. Fidalgo, G. Alves, A. Marques, U. Hernandez-Jayo, J. Garcia-Zubia, C. Kreiter, A. Pester, C. Garcia-Hernandez, R. Tavio, K. Valtonen, and E. Lehtikangas, “Pilar: a federation of visir remote laboratory systems for educational open activities,” in *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pp. 134–141, 2018. [Citado na página 40]
- [121] I. Gustavsson, J. Zackrisson, L. Håkansson, I. Claesson, and T. Lagö, “The visir project – an open source software initiative for distributed online laboratories,” vol. 2, 01 2010. [Citado na página 40]
- [122] M. Tawfik, E. Sanchristobal, S. Martín, C. Gil, A. Pesquera, P. Losada, G. Diaz, J. Peire, M. Castro, J. Garcia-Zubia, U. Hernandez, P. Orduña, I. Angulo, C. Costa Lobo, M. Marques, C. Viegas, and G. Alves, “Visir: experiences and challenges,” *International Journal of Online Engineering (iJOE)*, vol. 8, pp. 25–32, 02 2012. [Citado nas páginas ix, 40, 41 e 43]
- [123] “PXI-1033.” <https://www.ni.com/pt-pt/support/model.pxi-1033.html>, Nov. 2016. [Online; accessed 13. Jul. 2024]. [Citado nas páginas ix e 41]

- [124] “PXI-5114.” <https://www.ni.com/pt-pt/support/model.pxi-5114.html>, Nov. 2016. [Online; accessed 13. Jul. 2024]. [Citado nas páginas ix e 42]
- [125] “PXI-5402.” <https://www.ni.com/pt-pt/support/model.pxi-5402.html>, Nov. 2016. [Online; accessed 13. Jul. 2024]. [Citado nas páginas ix e 42]
- [126] “National instrument pxi-4072 | leasametric.” <https://www.leasametric.com/produit/national-instrument-pxi-4072/>. (Accessed on 07/13/2024). [Citado nas páginas ix e 42]
- [127] “Pxi-4110 - ni.” <https://www.ni.com/pt-pt/shop/model/pxi-4110.html>. (Accessed on 07/13/2024). [Citado nas páginas ix e 43]
- [128] I. Gustavsson, K. Nilsson, W. Kulesza, F. Garcia, and M. Castro, “Evaluation of the pilar pilot phase v 2.1.” <https://ec.europa.eu/programmes/erasmus-plus/project-result-content/e8531ace-ded5-4103-bd7f-be12c7608f38/>. (Accessed on 07/14/2024). [Citado nas páginas ix e 43]
- [129] M. Tawfik, E. Sanchristobal, S. Martin, R. Gil, G. Diaz, A. Colmenar, J. Peire, M. Castro, K. Nilsson, J. Zackrisson, L. Håkansson, and I. Gustavsson, “Virtual instrument systems in reality (visir) for remote wiring and measurement of electronic circuits on breadboard,” *IEEE Transactions on Learning Technologies*, vol. 6, pp. 60–72, Jan 2013. [Citado na página 43]
- [130] J. Pereira, I. Nardi da Silva, J. Schardosim Simão, L. Mellos Carlos, J. Silva, S. Bilessimo, and J. Alves, “Modelo de repositório de práticas didáticas de circuitos elétricos e eletrônicos utilizando o laboratório remoto visir,” 09 2017. [Citado na página 43]
- [131] F. Garcia-Loro, A. Macho, E. S. Cristobal, G. Diaz, M. Castro, W. Kulesza, I. Gustavsson, K. Nilsson, A. Fidalgo, G. Alves, A. Marques, U. Hernandez-Jayo, J. Garcia-Zubia, C. Kreiter, R. Oros, A. Pester, D. Garbi-Zutin, M. Auer, C. Garcia-Hernandez, R. Tavio, K. Valtonen, and E. Lehtikangas, “Experimenting in pilar federation: A common path for the future,” in *2018 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1518–1523, 2018. [Citado na página 44]
- [132] “International association of online engineering (iaoe) & global online consortium (golc).” [https://online-engineering.org/SIG\\_visir.php](https://online-engineering.org/SIG_visir.php). (Accessed on 07/14/2024). [Citado na página 44]
- [133] “International association of online engineering (iaoe) & global online consortium (golc).” [https://online-engineering.org/VISIR-Federation\\_about.php](https://online-engineering.org/VISIR-Federation_about.php). (Accessed on 07/14/2024). [Citado na página 44]

- [134] C. Kreiter, R. Oros, A. Pester, I. Gustavsson, M. Castro, A. Fidalgo, and G. R. Alves, “Visir federation: Initial building steps: Pilar experience — work in progress,” in *2017 4th Experiment@International Conference (exp.at’17)*, pp. 24–27, June 2017. [Citado na página 44]
- [135] “labview test & measurement | farnell portugal.” <https://pt.farnell.com/c/test-measurement?st=labview>. (Accessed on 06/30/2024). [Citado na página 47]
- [136] “Arduino mega 2560 rev3 — arduino official store.” <https://store.arduino.cc/products/arduino-mega-2560-rev3>. (Accessed on 07/01/2024). [Citado nas páginas ix e 48]
- [137] “Flask | the pallets projects.” <https://palletsprojects.com/p/flask/>. (Accessed on 07/26/2024). [Citado nas páginas 48, 68, 87 e 88]
- [138] R. Candido, “Arduino with python: How to get started – real python.” <https://realpython.com/arduino-python/>. (Accessed on 08/01/2024). [Citado na página 48]
- [139] “Micropython - python for microcontrollers.” <https://micropython.org/>. (Accessed on 08/01/2024). [Citado na página 48]
- [140] “A000067-datasheet.pdf.” <https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf>. (Accessed on 08/01/2024). [Citado na página 48]
- [141] “esp32-wroom-32e\_esp32-wroom-32ue\_datasheet\_en.pdf.” [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e\\_esp32-wroom-32ue\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf). (Accessed on 06/30/2024). [Citado na página 49]
- [142] “Getting started with micropython on esp32 and esp8266 | random nerd tutorials.” <https://randomnerdtutorials.com/getting-started-micropython-esp32-esp8266/>. (Accessed on 07/01/2024). [Citado na página 49]
- [143] “Esp devkits | espressif systems.” <https://www.espressif.com/en/products/devkits>. (Accessed on 07/01/2024). [Citado nas páginas ix e 49]
- [144] “Introducing: Raspberry pi 5! - raspberry pi.” <https://www.raspberrypi.com/news/introducing-raspberry-pi-5/>. (Accessed on 07/01/2024). [Citado nas páginas ix e 49]
- [145] NI, “Automating ni virtualbench using python - ni.” <https://knowledge.ni.com/>. (Accessed on 08/01/2024). [Citado na página 50]

- [146] armstrap, “Github - armstrap/armstrap-pyvirtualbench: Python wrappers to control ni virtualbench.” <https://github.com/armstrap/armstrap-pyvirtualbench/>. (Accessed on 07/17/2024). [Citado na página 50]
- [147] N. Jennings, “Socket programming in python (guide) – real python.” <https://realpython.com/python-sockets/>. (Accessed on 08/01/2024). [Citado nas páginas ix, 51, 52 e 96]
- [148] K. C. S. Adel S. Sedra, *Microelectronic circuits*. The Oxford series in electrical and computer engineering, Oxford University Press, 2015. [Citado nas páginas x, 55, 56, 57 e 113]
- [149] N. Instruments, “Virtualbench all-in-one instrument product flyer - national instruments.” <https://www.ni.com/pdf/product-flyers/virtualbench-all-in-one-instrument.pdf>. (Accessed on 07/21/2024). [Citado nas páginas x e 58]
- [150] “Pc104 consortium home - pc/104 consortiumpc/104 consortium.” <https://pc104.org/>. (Accessed on 08/01/2024). [Citado na página 59]
- [151] “Raspberry pi hardware - raspberry pi documentation.” <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>. (Accessed on 07/24/2024). [Citado nas páginas x, 61, 62 e 63]
- [152] “Raspberry pi documentation.” <https://www.raspberrypi.com/documentation/>. (Accessed on 07/24/2024). [Citado nas páginas xiii e 62]
- [153] “Dry relays | relays | comus group.” <https://comus-intl.com/product-category/relays/dry-relays/>. (Accessed on 07/24/2024). [Citado nas páginas x, 63 e 64]
- [154] T. Instruments, “Uln2003 data sheet, product information and support | ti.com.” <https://www.ti.com/product/ULN2003A>. (Accessed on 11/13/2024). [Citado nas páginas x, 64, 65 e 76]
- [155] “Sn54hc595 data sheet, product information and support | ti.com.” <https://www.ti.com/product/SN54HC595>. (Accessed on 07/25/2024). [Citado nas páginas x, xiii, 65 e 67]
- [156] C. A. Team, “Front end vs. back end: Where should you start?” <https://www.codecademy.com/resources/blog/front-end-vs-back-end/>. (Accessed on 07/26/2024). [Citado na página 67]
- [157] “Top 10 linguagens de back-end para 2024 | qual é a melhor?” <https://blog.back4app.com/pt/linguagens-de-programacao-para-back-end/>. (Accessed on 07/26/2024). [Citado na página 68]

- [158] A. Joy, “5 main disadvantages of python programming language – pythonista planet.” <https://pythonistaplanet.com/disadvantages-of-python/>. (Accessed on 07/26/2024). [Citado na página 68]
- [159] “The top programming languages 2023 - ieee spectrum.” <https://spectrum.ieee.org/the-top-programming-languages-2023>. (Accessed on 07/27/2024). [Citado na página 68]
- [160] “What is wsgi? — wsgi.org.” <https://wsgi.readthedocs.io/en/latest/what.html>. (Accessed on 07/26/2024). [Citado na página 68]
- [161] P. Projects, “Jinja documentation (official),” 2024. Acedido em: 18 de maio de 2025. [Citado nas páginas 68, 69 e 70]
- [162] D. Mashutin, “Django vs flask: Which is the best python web framework? | the pycharm blog.” <https://blog.jetbrains.com/pycharm/2023/11/django-vs-flask-which-is-the-best-python-web-framework/>. (Accessed on 07/26/2024). [Citado na página 68]
- [163] R. Johns, “Flask vs django: Which python web framework to use in 2024?” <https://hackr.io/blog/flask-vs-django>. (Accessed on 07/26/2024). [Citado na página 68]
- [164] “Html basics - learn web development | mdn.” [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics). (Accessed on 07/26/2024). [Citado nas páginas x e 70]
- [165] “Html element - wikipedia.” [https://en.wikipedia.org/wiki/HTML\\_element](https://en.wikipedia.org/wiki/HTML_element). (Accessed on 08/03/2024). [Citado na página 70]
- [166] “Introduction to html 4.” <https://www.w3.org/TR/html40/intro/intro.html>. (Accessed on 08/03/2024). [Citado na página 71]
- [167] “Starting with html + css.” <https://www.w3.org/Style/Examples/011/firstcss>. (Accessed on 08/03/2024). [Citado nas páginas xv, 71 e 72]
- [168] “Serviço de validação de css do w3c.” [Online; accessed 2025-05-29]. [Citado na página 72]
- [169] “The w3c markup validation service.” [Online; accessed 2025-05-29]. [Citado na página 72]
- [170] “Javascript reference - javascript | mdn.” <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>. (Accessed on 08/03/2024). [Citado na página 73]

- [171] “Javascript tutorial.” <https://www.w3schools.com/js/default.asp>. (Accessed on 08/03/2024). [Citado na página 73]
- [172] T. Instruments, “Lm317 data sheet, product information and support | ti.com.” <https://www.ti.com/product/LM317>. (Accessed on 07/25/2024). [Citado nas páginas 79 e 80]
- [173] “1n400x - axial-lead glass passivated standard recovery rectifiers.” <https://cdn.sparkfun.com/assets/2/c/2/d/6/1N400x-D.PDF>. (Accessed on 11/14/2024). [Citado na página 80]
- [174] T. W. Tim, “(887) python website full tutorial - flask, authentication, databases & more - youtube.” [Online; accessed 2025-05-30]. [Citado na página 87]
- [175] DigitalOcean, “(887) getting started with flask | 1-hour tech talk - youtube.” [Online; accessed 2025-05-30]. [Citado na página 87]
- [176] “NumPy.” [Online; accessed 2025-04-15]. [Citado nas páginas 95 e 100]
- [177] N. Instruments, “Ni virtual bench — pymeasure 0.15.1.dev69+g6ab0fdb.d20250312 documentation.” [Online; accessed 2025-03-12]. [Citado na página 97]
- [178] “Matplotlib — visualization with python.” [Online; accessed 2025-04-30]. [Citado na página 100]
- [179] N. Instruments, “Ni virtual bench 8012 data sheet.” [Online; accessed 2025-04-25]. [Citado na página 110]

**Anexo A**

**Datasheet**



# Matriz de relés LaRE - Protótipo

## 1 Características

- Matriz com 3 placas baseadas nas dimensões PC/104;
- Fonte de alimentação-5 V/1 A, baseado no LM317<sup>a</sup>;
- Experiências:
  - Lei de Ohm;
  - Rectificador de meia onda e onda completa;
  - Filtros passa-alto e passa-baixo.



<sup>a</sup>Datasheet LM317

## 2 Descrição

A matriz de relés do LaRE é constituída por três placas: placa de alimentação, placa com o circuito da Lei de Ohm e a placa com os circuitos dos filtros e rectificadores. A matriz do LaRE é controlada pelo RaspberryPI. Há vários tipos de alimentação: 220 V AC que alimenta o rectificador de onda completa; o VirtualBench fornece alimentação de 12 V DC ao LM317 e aos registos de deslocamento, sn74hc595<sup>1</sup> e fornece tensão variável 0-5 V DC ao circuito da Lei de Ohm. Os ULN2003a<sup>2</sup>, são alimentados pela fonte de alimentação, baseada no LM317, 5 V. Todas as medições são feitas pelo VirtualBench.

## 3 Aplicações

- Laboratórios Remotos;
- Experiências remotas;
- Estudo da Lei de Ohm;;
- Estudo de rectificadores;
- Estudo de filtros.

<sup>1</sup>Datasheet SN74HC595

<sup>2</sup>Datasheet ULN2003a

## 4 Especificações técnicas

Tabela 1: Especificações técnicas genéricas - **Sujeito a mudança**

	Unidades	Valores
$I_{Max}$ do VirtualBench	A	0.5
$I_{Max}$ da fonte 5 V DC	A	1.5
Transformador	?	REVER
Dimensões placas	mm * mm	96x90

Tabela 2: Especificações técnicas genéricas dos relés - **Sujeito a mudança**

Relés	Bobine			Contactos		
	$V_{Nominal}$	$V_{Max}$	$P_{Max}$	$V_{Max}$	$I_{Max}$	$I_{Max}$
3570-1331-123	12 V	16 V	10 W	150 V <sub>DC</sub>	0.5 A	1 A
3572-1220-123						

Tabela 3: Ligações entre placas - **Sujeito a mudança**

Tipo de ligações	
RaspberryPI-LaRE	Conector IDC, fêmea, 40 pinos, 2.54 mm
Placa Lei de Ohm-Rectificadores/Filtros	Conector de 6 pinos Arduino Stackable
Placa Fontes de tensão-Rectificadores/Filtros	Conector KK, 2.54 mm, M/F

## 5 Conectores

Todas as ligações entre o VirtualBench e a matriz, incluindo alimentação DC, saída do gerador de sinal e canais do osciloscópio, bem como a alimentação do transformador, são efectuadas através de conectores tipo KK 2.54 mm, (macho/fêmea).

Para a interligação entre placas, foram utilizados conectores Arduino Stackable de 6 pinos e a ligação do RaspberryPI à placa é feita através de um conector IDC de 40 pinos. As Figuras 1, 2 e 3 apresentam os conectores referidos, respectivamente<sup>3</sup>

<sup>3</sup>Imagens retiradas do site <https://www.botnroll.com/pt/>.

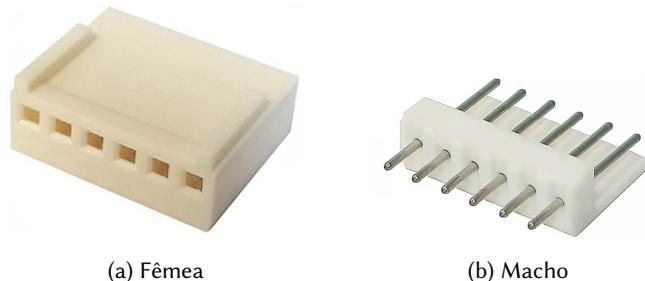
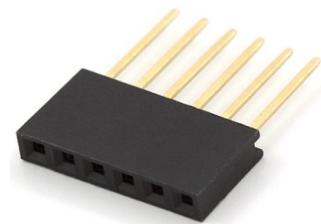
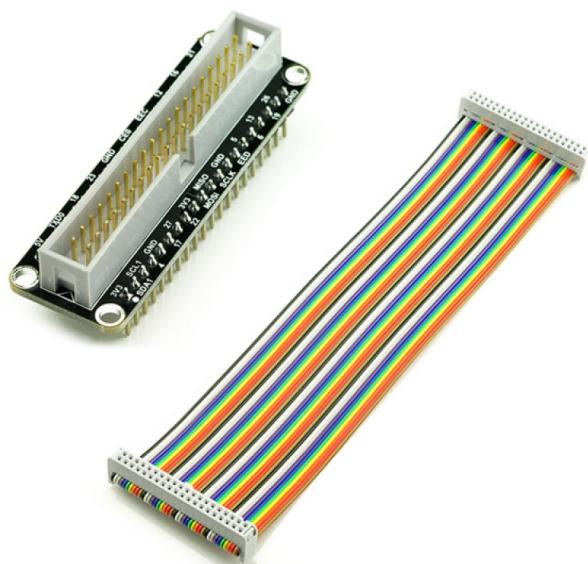
Figura 1: Conectores *KK*Figura 2: Conector *stackable* 6 pinos Arduino

Figura 3: Ligação IDC 40 pinos [Macho/Fêmea]

## Conteúdo

<b>1 Características</b>	<b>1</b>
<b>2 Descrição</b>	<b>1</b>
<b>3 Aplicações</b>	<b>1</b>
<b>4 Especificações técnicas</b>	<b>2</b>
<b>5 Conectores</b>	<b>2</b>
<b>6 Pinout</b>	<b>5</b>
<b>7 Dimensões mecânicas</b>	<b>8</b>
<b>8 Esquemas elétricos</b>	<b>10</b>
<b>9 Esquemas PCB</b>	<b>13</b>

## Listas de Tabelas

1 Especificações técnicas genéricas - <b>Sujeito a mudança</b> . . . . .	2
2 Especificações técnicas genéricas dos relés - <b>Sujeito a mudança</b> . . . . .	2
3 Ligações entre placas - <b>Sujeito a mudança</b> . . . . .	2

## 6 Pinout

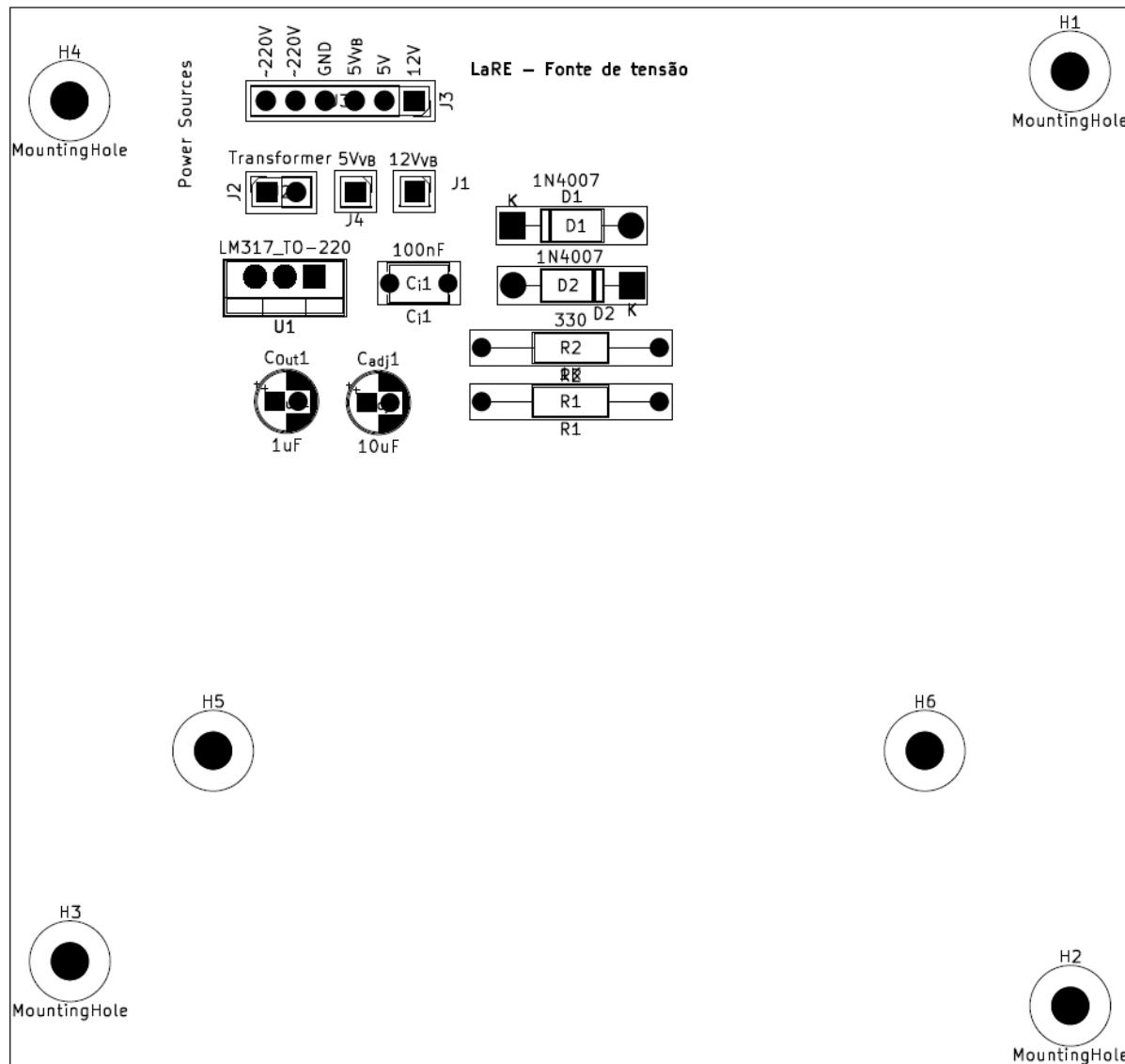


Figura 4: LaRE - Pinout [Fonte de tensão]

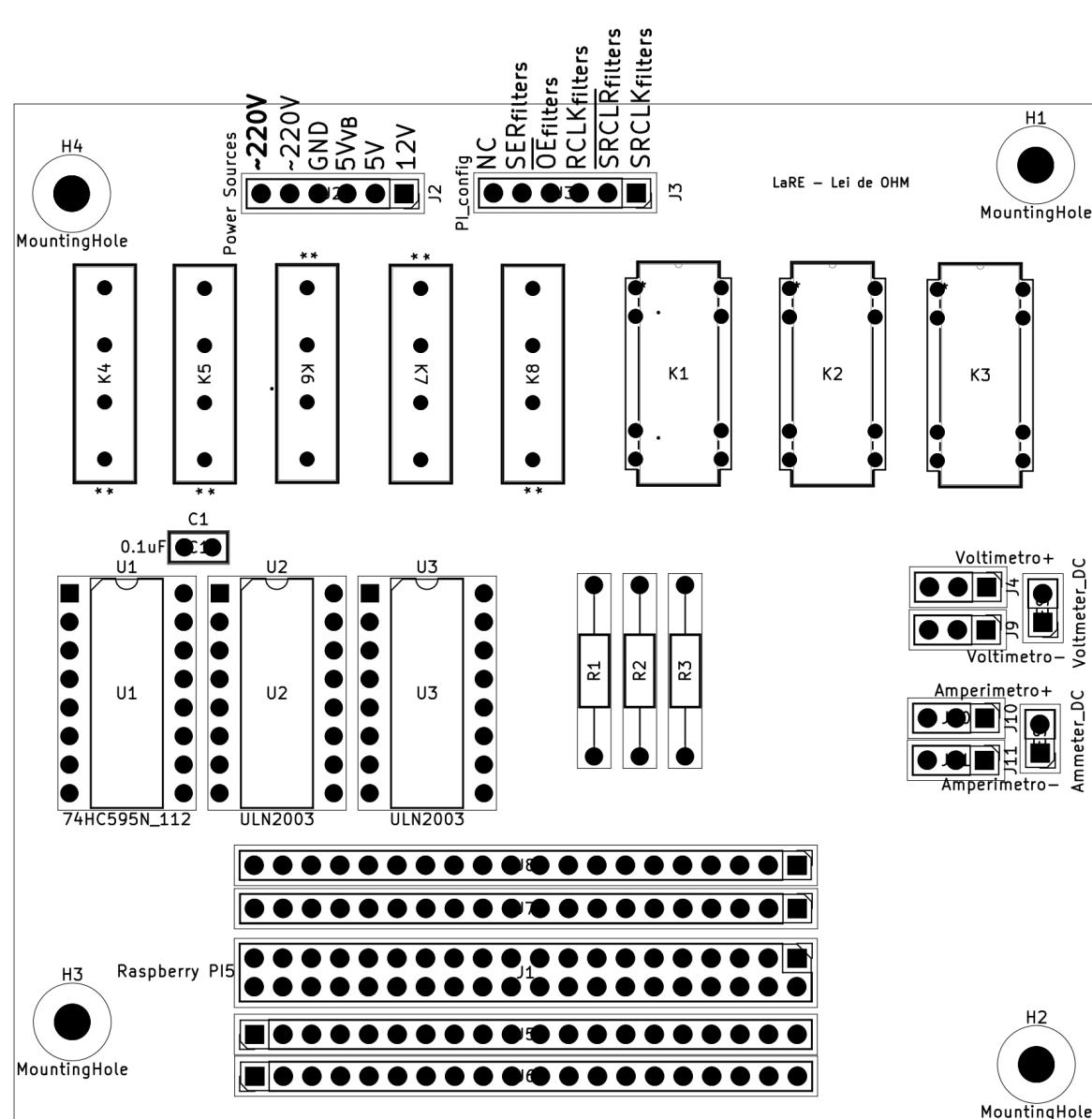


Figura 5: LaRE - Pinout [Lei de Ohm]  
Os símbolos \*\* indicam o pino 1 dos relés

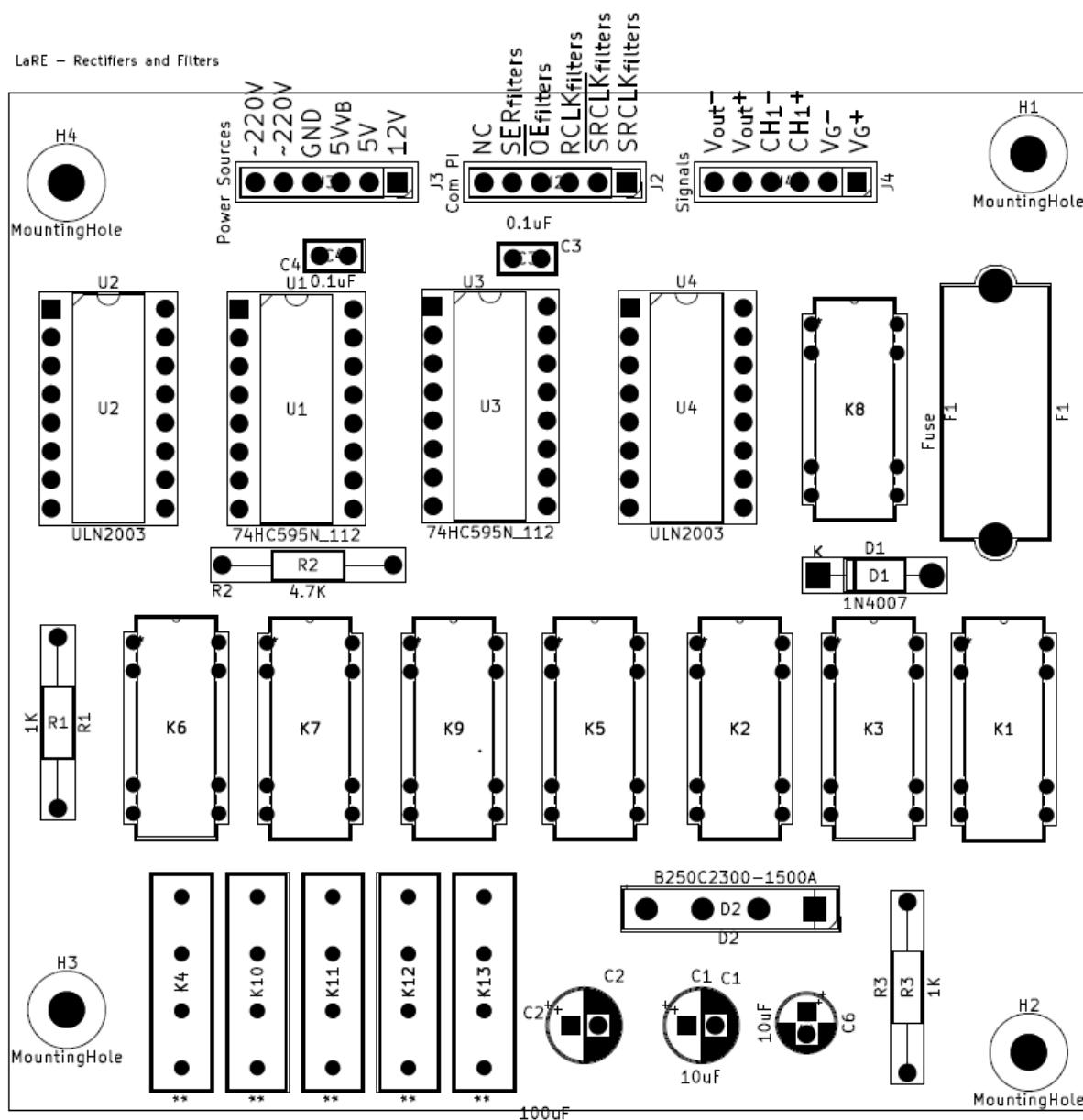
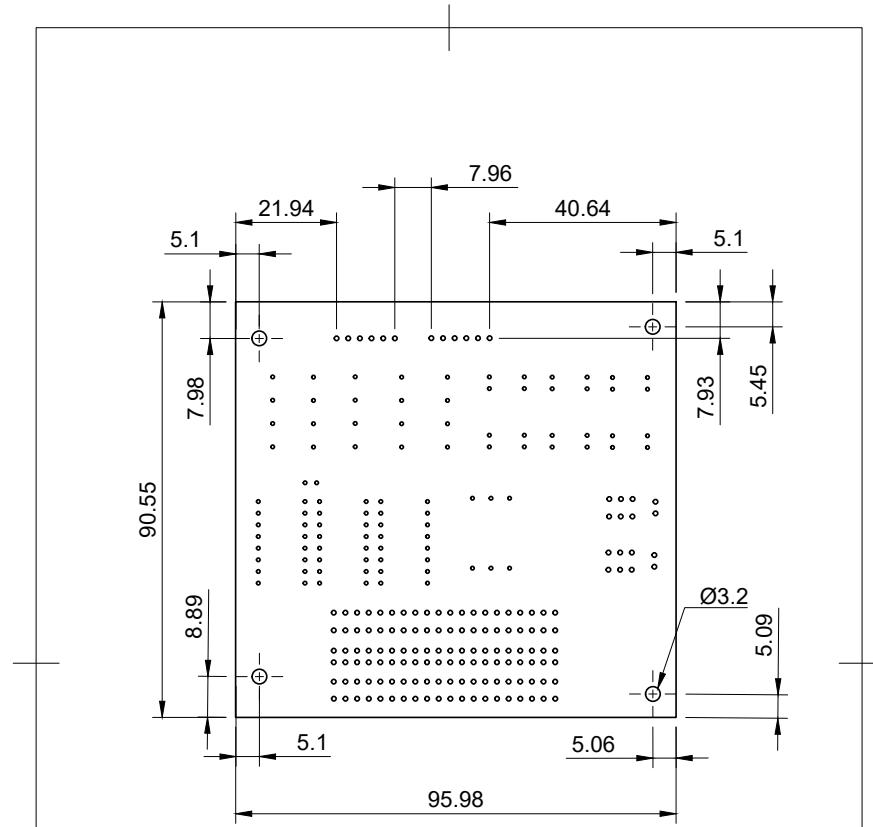


Figura 6: LaRE - Pinout [Rectificadores/Filtros]

## 7 Dimensões mecânicas



Dept. ISEP	Technical reference Dimensões	Created by Eduardo Ramalhadeiro 20/07/2024	Approved by
		Document type Datasheet Matriz	Document status Protótipo
	Title LaRE - Matriz	DWG No.	1
		Rev. 1	Date of issue 20/07/2024
			Sheet 1/1

Figura 7: Dimensões mecânicas PC/104 - LaRE

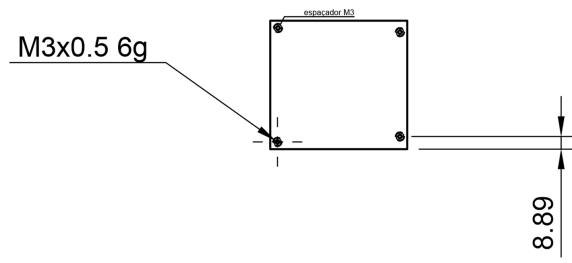
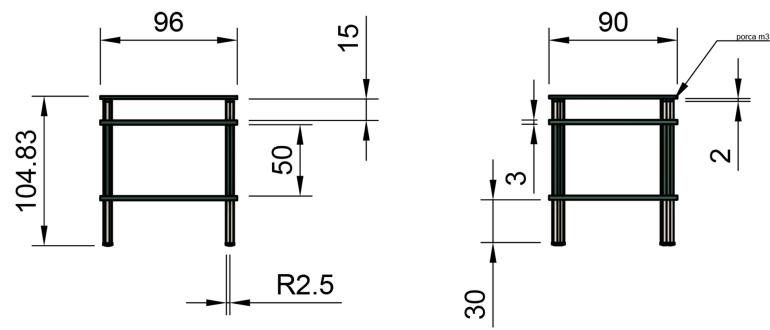


Figura 8: Dimensões mecânicas LaRE



Figura 9: Perspectiva LaRE

## 8 Esquemas elétricos

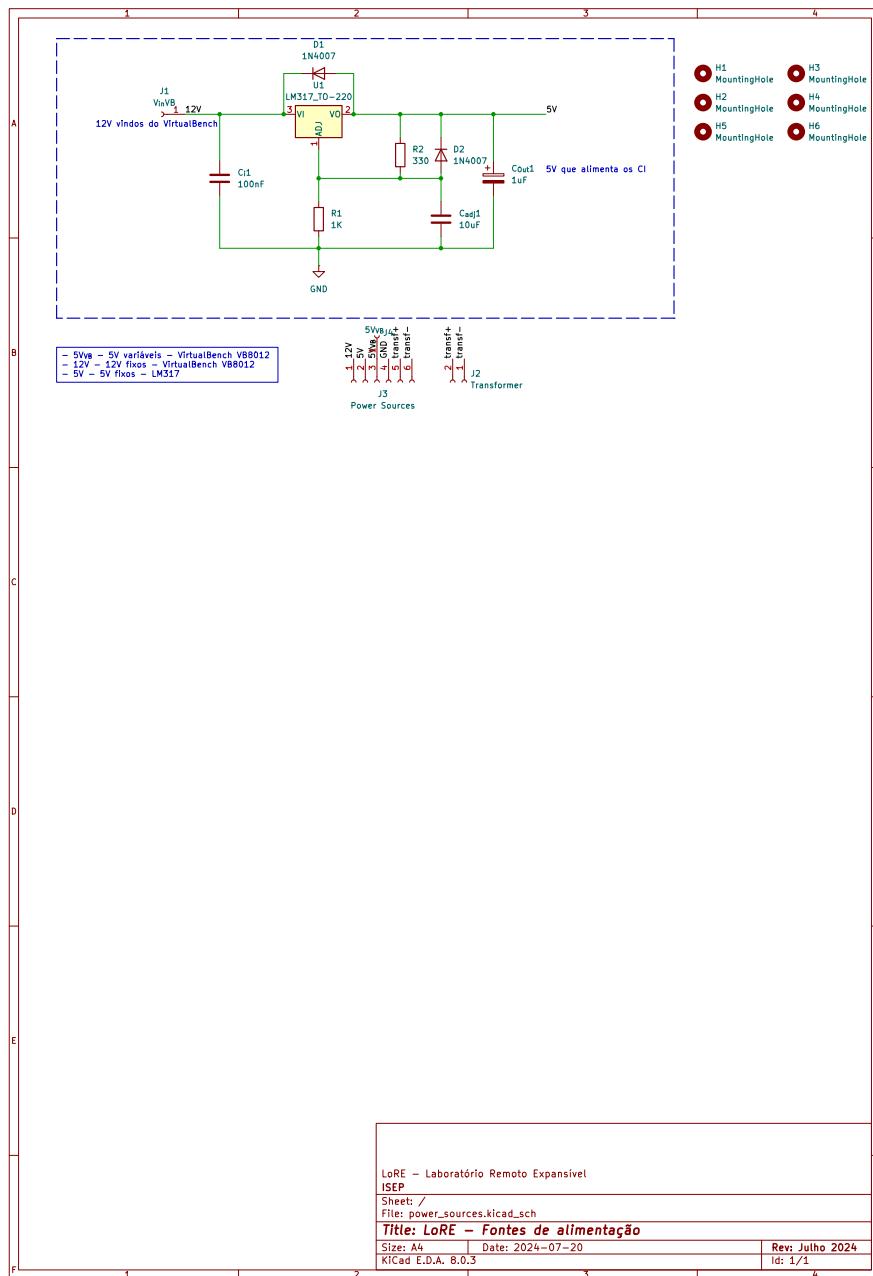


Figura 10: LaRE - Esquema [Fonte de tensão]

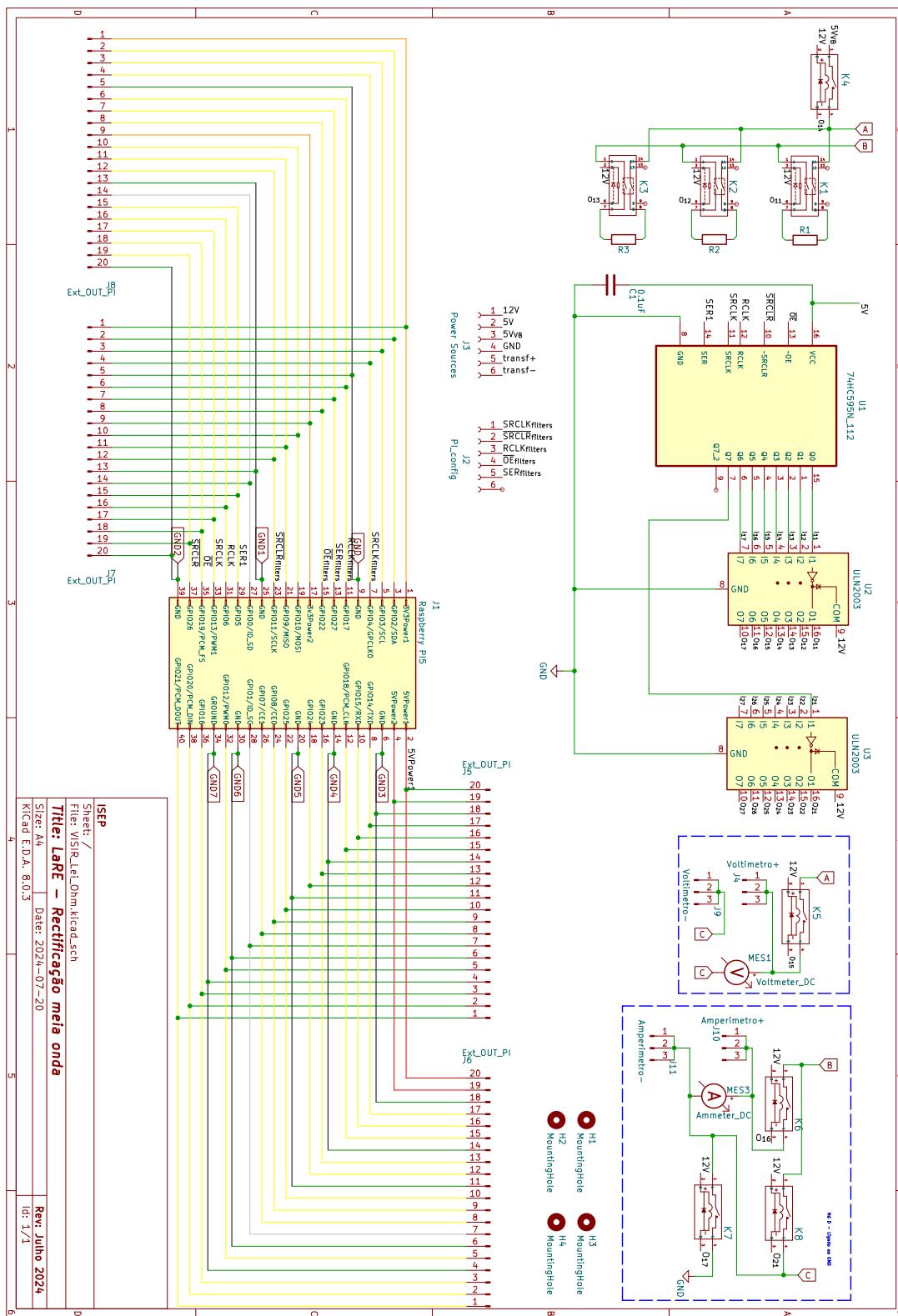


Figura 11: LaRE - Esquema [Lei de Ohm]

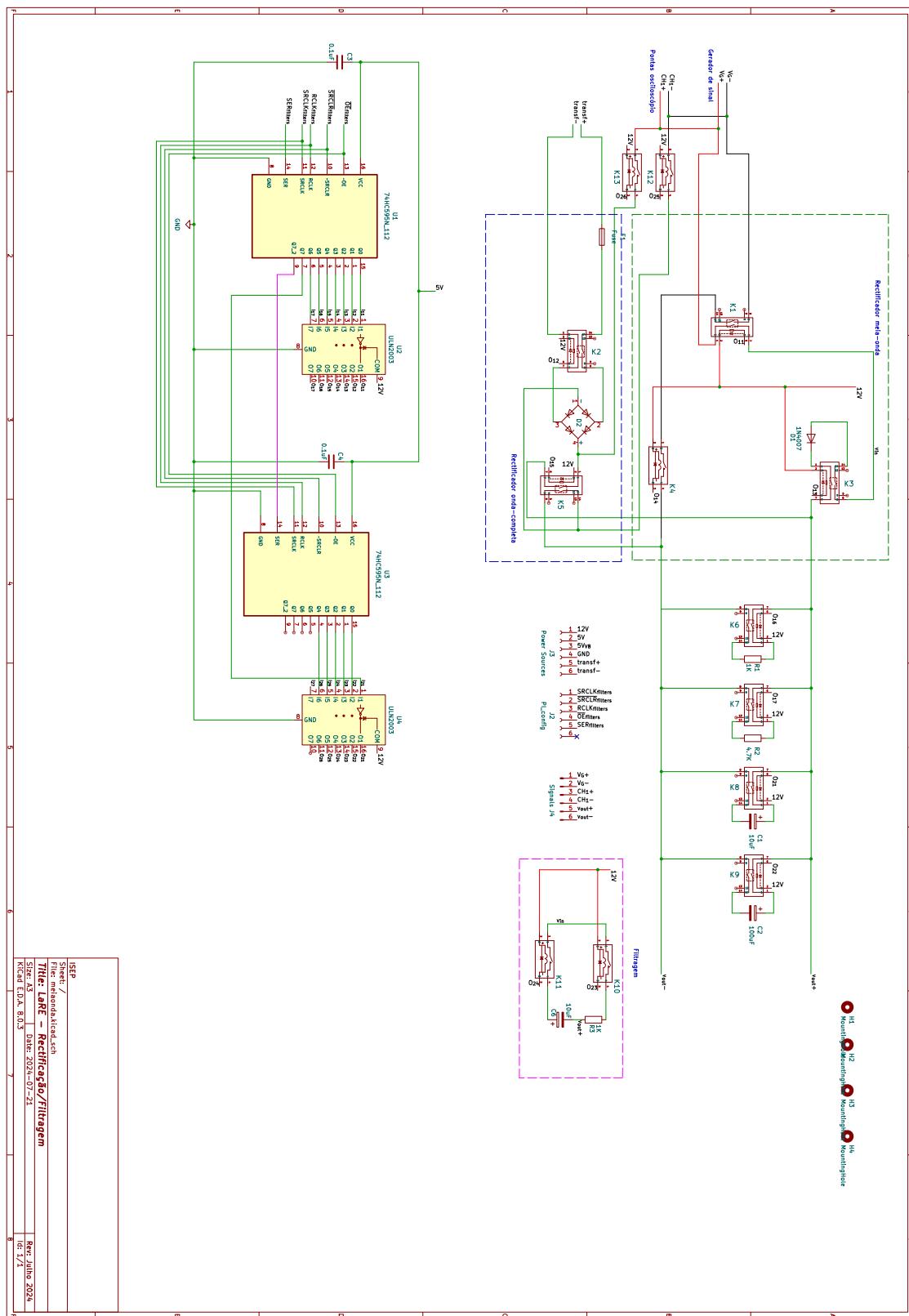


Figura 12: LaRE - Esquema [Recificadores/Filtros]

## 9 Esquemas PCB

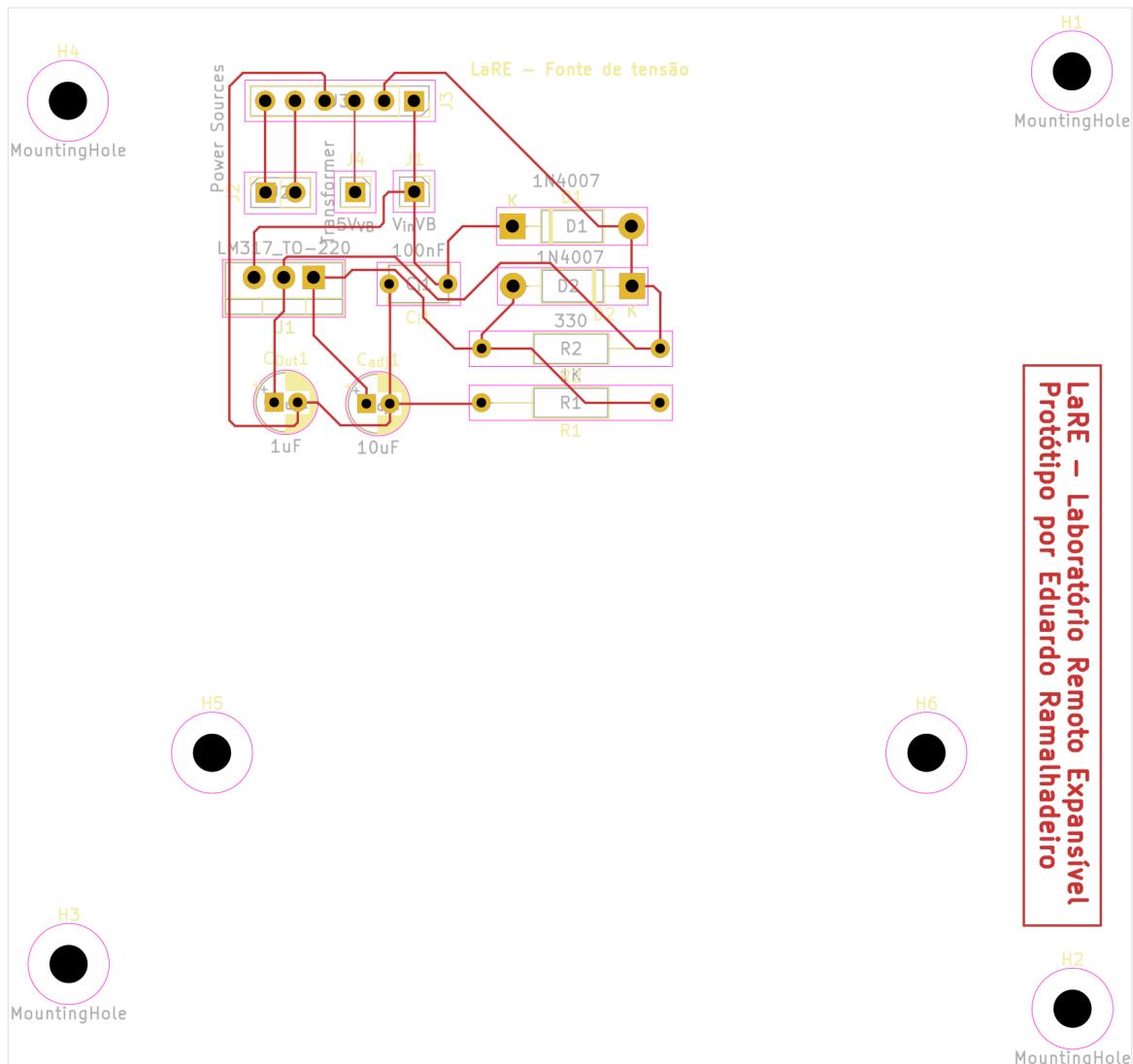


Figura 13: LaRE - PCB [Fonte de tensão]

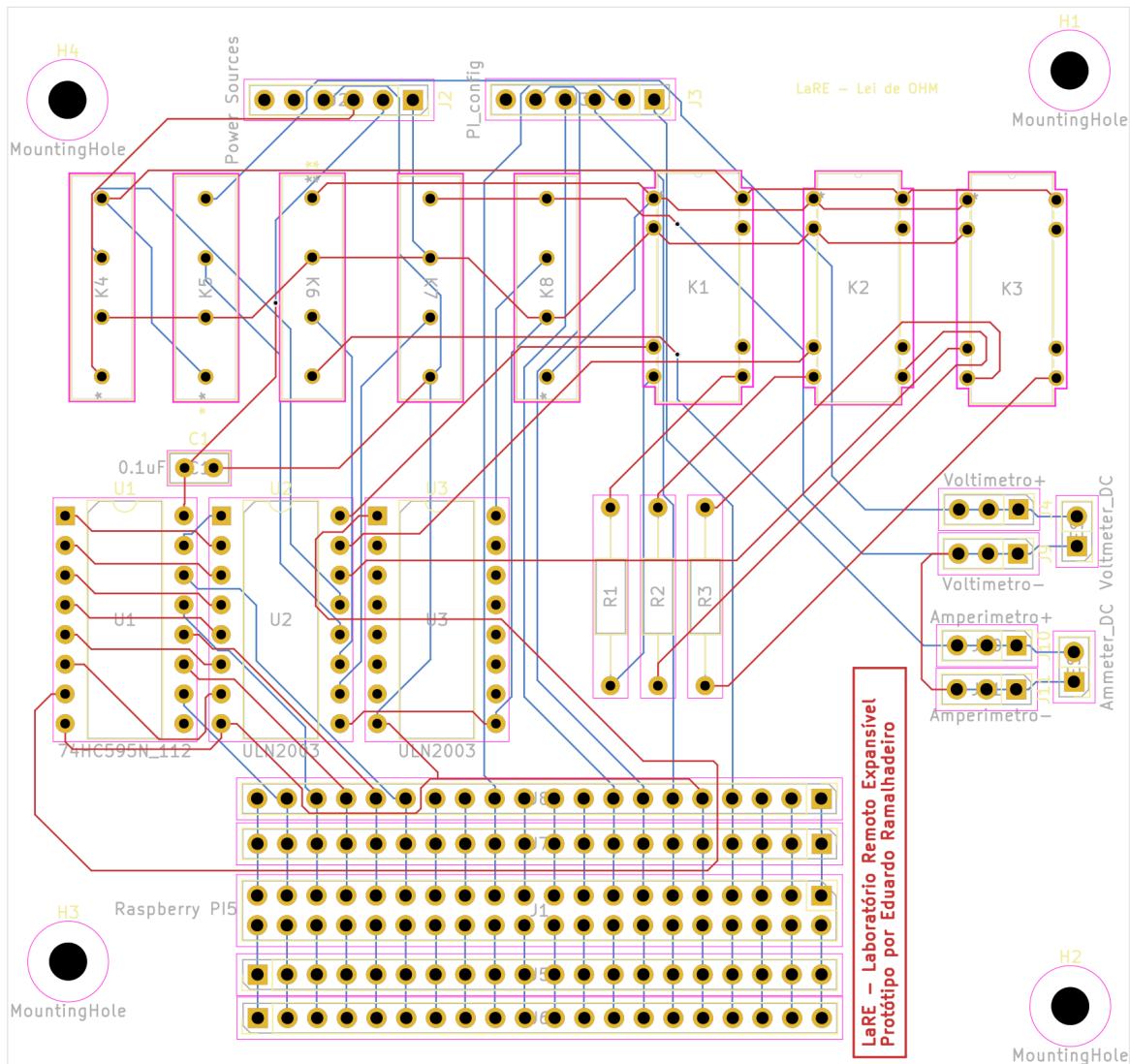


Figura 14: LaRE - PCB [Lei de Ohm]

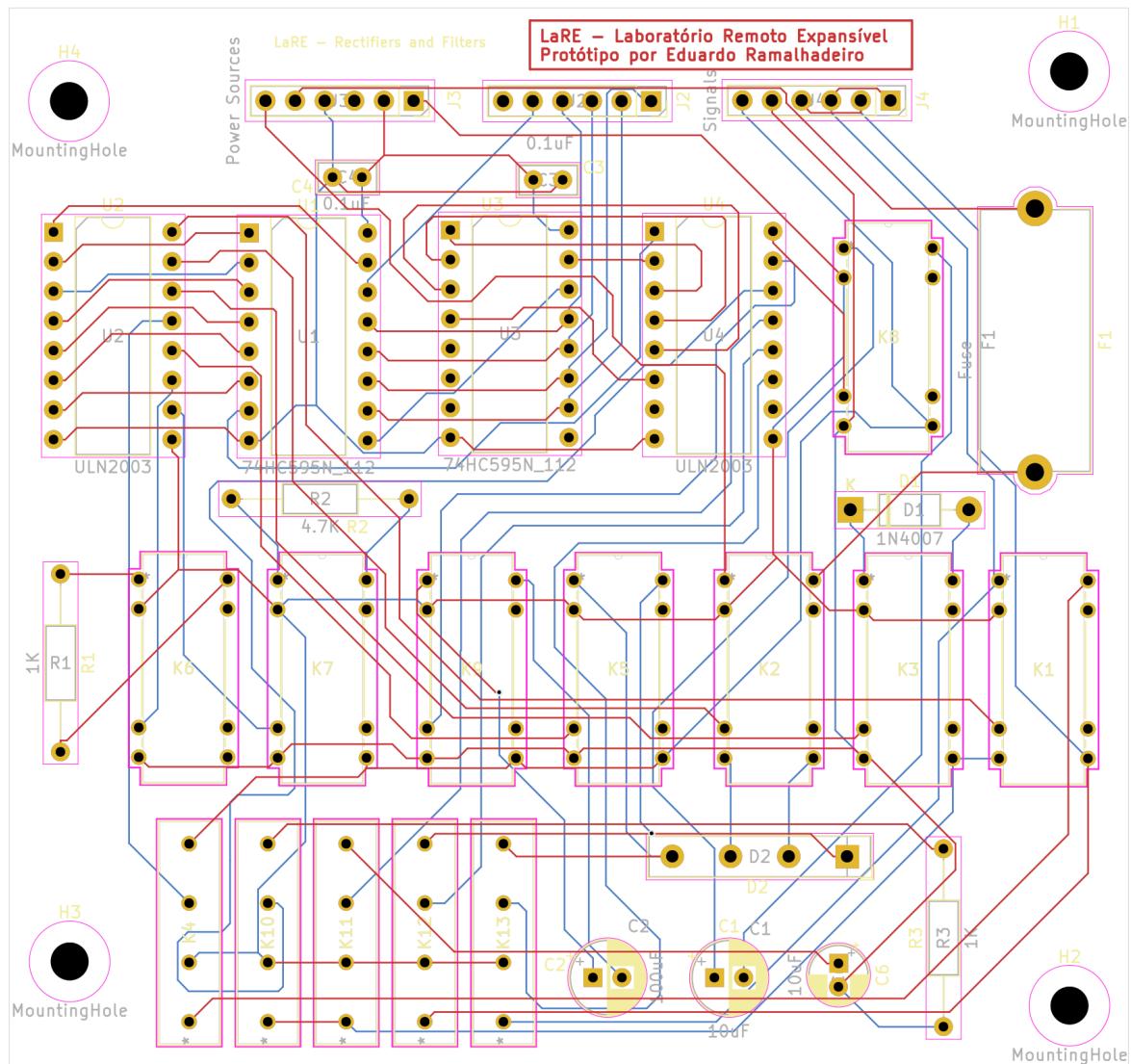


Figura 15: LaRE - PCB [Recificadores/Filtros]