

ScopeLab: Desenvolvimento de novas
funcionalidades e melhorias à
plataforma *Oscilloscope Interface*

Cláudia Alves Barbosa Coelho

LEEC
Licenciatura em Engenharia Eletrotécnica e de Computadores



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto

Setembro, 2026

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Projeto/Estágio, do 3º ano, da Licenciatura em Engenharia Eletrotécnica e de Computadores.

Candidato: Cláudia Alves Barbosa Coelho, N.º 1230755,
1230755@isep.ipp.pt

Orientação Científica: João Ferreira, jfa@isep.ipp.pt

Coorientação Científica: Mário Alves, mjf@isep.ipp.pt

ISEP INSTITUTO SUPERIOR
DE ENGENHARIA DO PORTO

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

Setembro, 2026

Agradecimentos

Primeiro queria começar por agradecer ao meu orientador, o professor João Ferreira, por todo o apoio ao longo deste projeto, tanto a nível mais pessoal quando a pressão era demasiada e a vontade de desistir grande, como a nível técnico, visto ser o desenvolvedor da versão original desta aplicação. Agradeço também o facto de confiar em mim para dar continuidade ao seu trabalho e acreditar em mim para o desenvolver, mesmo quando parecia impossível.

Agradeço também ao meu co-orientador, o professor Mário Alves, por toda a assistência que forneceu ao longo do projeto, que considero essencial para o seu sucesso. De igual forma agradeço ao professor Paulo Oliveira e ao professor André Rocha pela disponibilidade para a realização dos testes da aplicação, que permitiram a sua evolução e melhoria.

Por último, agradeço à minha família, sem eles nada disto seria possível, pois foram sempre o meu apoio ao longo de todo o meu percurso académico e que me permitiu chegar onde estou.

Resumo

Este documento tem como objetivo descrever o processo de aprimoramento e expansão de uma aplicação web preexistente, denominada Oscilloscope Interface. Atualmente, esta ferramenta já se encontra integrada na dinâmica letiva, sendo ativamente utilizada pelos alunos na preparação de guíões, como método de estudo e no decorrer das avaliações de laboratório. Esta plataforma permite manipular remotamente e em tempo real os osciloscópios GW-Instek GDS2062 instalados nas bancadas do Laboratório de Circuitos e Sinais Elétricos do Instituto Superior de Engenharia do Porto (ISEP), a partir de qualquer computador com acesso à internet. Em paralelo, o sistema atua como um simulador interativo, replicando o comportamento de um osciloscópio digital e de um gerador de sinais.

A arquitetura da plataforma assenta no modelo cliente-servidor e foi desenvolvida utilizando o ambiente de execução Node.js, a linguagem de programação *JavaScript* (JS), *Embedded JavaScript Templating* (EJS), e *Cascading Style Sheets* (CSS) para a estruturação visual. Durante o desenvolvimento deste projeto, a infraestrutura de software sofreu uma reestruturação, culminando na adoção de tecnologias de contenorização para assegurar a portabilidade e a integridade da aplicação em qualquer ambiente. A natureza web do sistema garante que os utilizadores possam usufruir de todas as capacidades do programa diretamente através do navegador, sem necessidade de efetuar instalações locais.

O grande destaque desta ferramenta reside na sua versatilidade em ambientes de ensino. O simulador foi enriquecido para permitir um estudo de sinais através de novas funcionalidades de análise e síntese, tais como o Modo XY e o Custom Wave. No modo de controlo e aquisição remoto, a aplicação evoluiu para suportar a seleção dinâmica de múltiplos osciloscópios ligados em simultâneo ao servidor. Para complementar a componente técnica, a introdução de uma secção de guíões (LabScripts) possibilita que a exploração dos equipamentos seja realizada de forma guiada e totalmente autónoma, consolidando a plataforma como um recurso vital para a aprendizagem à distância e a operação de laboratórios remotos.

Palavras-Chave: Simulador de Osciloscópio, GW-Instek GDS2062, Aprendizagem à Distância, Aplicação Web.

Abstract

This document aims to describe the process of improving and expanding a pre-existing web application called Oscilloscope Interface. Currently, this tool is already integrated into the teaching dynamics, being actively used by students in the preparation of scripts, as a study method, and during laboratory assessments. This platform allows remote and real-time manipulation of the GW-Instek GDS2062 oscilloscopes installed on the benches of the Electrical Circuits and Signals Laboratory at ISEP, from any computer with internet access. In parallel, the system acts as an interactive simulator, replicating the behavior of a digital oscilloscope and a signal generator.

The platform's architecture is based on the client-server model and was developed using the Node.js runtime environment, the .js and .ejs programming languages, and .css for visual structuring. During the development of this project, the software infrastructure underwent restructuring, culminating in the adoption of containerization technologies to ensure the portability and integrity of the application in any environment. The web-based nature of the system guarantees that users can enjoy all the program's features directly through their browser, without the need for local installations.

The great highlight of this tool is its versatility in educational environments. The simulator has been enriched to allow the study of signals through new analysis and synthesis features, such as XY Mode and Custom Wave. In remote control and acquisition mode, the application has evolved to support the dynamic selection of multiple oscilloscopes connected simultaneously to the server. To complement the technical component, the introduction of a scripting section (LabScripts) allows for guided and fully autonomous exploration of the equipment, consolidating the platform as a vital resource for distance learning and the operation of remote laboratories.

Keywords: Oscilloscope Simulator, GW-Instek GDS2062, Distance Learning, Web Application.

Índice

Lista de Figuras	vii
Lista de Tabelas	ix
Listagens	xi
Lista de Acrónimos	xiii
1 Introdução	1
1.1 Contextualização	1
1.2 Objetivos	2
1.3 Metodologia e Contribuições	3
1.4 Estrutura do Relatório	5
2 Contexto científico-tecnológico	7
2.1 O que é e para que serve um osciloscópio?	7
2.1.1 Funcionalidades de um osciloscópio	9
2.2 Aprendizagem Remota	10
2.2.1 Trabalhos anteriores	10
2.2.2 Importância do Ensino Remoto	11
2.3 Virtualização Laboratorial	12
2.3.1 Ecossistemas de Instrumentação Virtual na Engenharia Ele- trotécnica	12
2.4 Tecnologias e plataformas de apoio para o projeto de software	13
2.4.1 Desenvolvimento da Aplicação Web	13
2.4.2 Linguagens e Bibliotecas	13
2.4.3 Notion e Github	14
2.4.4 Plataformas de programação utilizadas	15
3 Abordagem de Implementação	17
3.1 Arquitetura de software e modos de utilização	17
3.1.1 Alterações - Modo simulação	19
3.1.2 Alterações - Modo aquisição e controlo	19
3.2 Aspetos de implementação	19

4	Guia de utilização da aplicação	25
4.1	Página Principal	25
4.2	Principais funcionalidades da aplicação	26
4.2.1	Guia de Resolução de Problemas	26
4.2.2	Trigger Level	27
4.2.3	Trigger Slope	28
4.2.4	Modo XY	31
4.2.5	Custom Wave - sinal personalizado	33
4.2.6	Página de Support e LabScripts	37
4.3	Modo de aquisição e controlo remoto	39
4.3.1	Seletor de Osciloscópio/Equipamentos	39
5	Avaliação e Teste de Resultados	43
5.1	Metodologia de Testes	43
5.2	Análise de Resultados	43
6	Conclusões	45
6.1	Aprendizagem	45
6.2	Resumo das Contribuições	46
6.3	Trabalho futuro	47
	Referências	49
Anexo A	Guiões Laboratoriais	51
A.1	Guião de Simulação - Atualizado com Trigger	51
A.2	Guião de Simulação - Modo XY	53
A.3	Guião de Simulação - Custom Wave	54
A.3.1	Acesso à Ferramenta	54
A.3.2	Utilização do Modo Simples	54
A.3.3	Utilização do Modo Avançado	55
A.4	Guião de Modo Aquisição e Controlo	55

Lista de Figuras

1.1	Cronograma do projeto	4
2.1	Componentes X, Y e Z de uma forma de onda [1].	8
2.2	Osciloscópio do Laboratório de Circuitos e Sinais Elétricos do ISEP	8
2.3	Display sem trigger [1]	9
2.4	Trigger Level e Slope [1]	9
2.5	Figuras de Lissajous [1]	10
3.1	Antiga estrutura do código	18
3.2	Nova estrutura	18
3.3	Atualização da frequência	20
3.4	Ficheiro CSV	21
3.5	Exemplo de utilização modo XY	22
3.6	Diagrama de sequência do seletor de osciloscópios	23
4.1	Interface	25
4.2	Exemplo das sugestões de diagnóstico apresentadas	27
4.3	Menu interativo de diagnóstico rápido de problemas na interface do simulador.	27
4.4	Efeito do Trigger Level em cada forma de onda	28
4.5	Trigger slope - rising	29
4.6	Trigger Slope - Falling	30
4.7	Comandos Slope	30
4.8	Comandos modo XY	31
4.9	Modo "normal"(main) - base de tempo ligada	32
4.10	Modo XY - CH1 no x e CH2 no y	33
4.11	Custom Wave - Janela configuração modo simples	34
4.12	Custom Wave - Janela configuração modo avançado	35
4.13	Custom Wave - Painel de ajuda	36
4.14	Custom Wave - Sinal cosseno - Modo simples	36
4.15	Custom Wave - Sinal seno - Modo avançado	37
4.16	Input CH1 - Custom Wave	37
4.17	Menus de Support e Labscripts	38
4.18	Página Labscripts	39

4.19	Seletor de osciloscópios	40
4.20	Menu de seleção do osciloscópio a controlar remotamente	40
4.21	Validação do modo de aquisição: correspondência entre o sinal físico e o sinal adquirido no simulador	41

Lista de Tabelas

4.1	Atualização das funcionalidades da aplicação.	26
6.1	Comparação de funcionalidades entre a versão original (V0.1.3) e a versão atual (V1.4.6).	47

Listagens

Lista de Acrónimos

CSS	<i>Cascading Style Sheets</i>
CSV	<i>Comma-Separated Values</i>
D3	<i>Data-Driven Documents</i>
EJS	<i>Embedded JavaScript Templating</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IA	<i>Inteligência Artificial</i>
IIFE	<i>Immediately Invoked Function Expression</i>
ISEP	Instituto Superior de Engenharia do Porto
JS	<i>JavaScript</i>
JSON	<i>JavaScript Object Notation</i>
LEEC	Licenciatura em Engenharia Eletrotécnica e de Computadores
PESTA	<i>Projeto / Estágio</i>
SVG	<i>Scalable Vector Graphics</i>
TCIRC	<i>Teoria dos Circuitos</i>
VISIR	<i>Virtual Instrument Systems in Reality</i>
VS Code	<i>Visual Studio Code</i>

Capítulo 1

Introdução

O presente relatório descreve o trabalho desenvolvido na unidade curricular de *Projeto / Estágio* (PESTA) da Licenciatura em Engenharia Eletrotécnica e de Computadores (LEEC) do Instituto Superior de Engenharia do Porto (ISEP).

O projeto consistiu na otimização e expansão da plataforma *Oscilloscope Interface*, uma aplicação web preexistente destinada à simulação e controlo remoto dos osciloscópios existentes no Laboratório de Circuitos e Sinais (I305) do ISEP. O trabalho centrou-se na melhoria contínua da ferramenta, com particular foco na correção de erros, e na implementação de novas funcionalidades.

1.1 Contextualização

O acesso regular a laboratórios é uma parte fundamental no ensino e na prática da engenharia eletrotécnica. No entanto, estes espaços apresentam algumas limitações, seja pelo custo elevado dos equipamentos, seja pelas restrições de horários e de espaço que muitas vezes impedem os estudantes de testarem conceitos ao seu próprio ritmo e fora do ambiente académico.

Para dar resposta a esta limitação, surge a necessidade crescente de criar alternativas digitais que facilitem o acesso a estes recursos. Plataformas web capazes de simular o comportamento de instrumentos reais e de permitir o seu controlo remoto tornam-se, assim, essenciais. Estas soluções não pretendem substituir o trabalho prático numa bancada física, mas sim complementar a aprendizagem, permitindo

que a análise de sinais e o teste de circuitos possam ser realizados de forma acessível, intuitiva e segura a partir de qualquer lugar. É com este propósito que a plataforma se estrutura em dois modos de operação fundamentais: o **Modo de Simulação**, focado na análise de formas de onda, e o **Modo de Aquisição e Controlo Remoto**, que permite a interação em tempo real com os osciloscópios físicos do laboratório.

Acompanhando a importância cada vez maior da instrumentação virtual, este projeto surge da necessidade direta de evoluir o trabalho desenvolvido numa versão anterior desta plataforma. Embora a iteração prévia cumprisse o seu propósito base, apresentava lacunas ao nível da estabilidade da arquitetura de software e carecia de ferramentas de análise mais avançadas, o que limitava o seu verdadeiro potencial pedagógico.

Desta forma, a principal motivação do presente projeto incidiu em colmatar essas limitações, tornando a aplicação mais versátil e rigorosa. O foco central passou por estabilizar o código, corrigir falhas em ambos os modos de operação e introduzir novas funcionalidades que aproximem a simulação da experiência real. Adicionalmente, para reforçar a componente didática e promover a autonomia dos estudantes, o projeto contemplou a elaboração de **guiões laboratoriais** estruturados, concebidos para guiar o utilizador na exploração de todas as capacidades da plataforma.

Ao longo dos próximos capítulos, procurarei detalhar os motivos destas melhorias e partilhar a visão que orientou o desenvolvimento prático. Serão exploradas as integrações das novas funcionalidades, a consolidação dos modos de operação e a utilidade dos guiões criados. O objetivo é demonstrar de que forma estas intervenções elevaram o projeto original, transformando-o numa solução muito mais completa, eficiente e adaptável às necessidades dos utilizadores.

1.2 Objetivos

O objetivo principal deste projeto consiste na evolução e otimização da aplicação *Oscilloscope Interface*, visando torná-la numa ferramenta mais robusta, completa e intuitiva.

Para alcançar esta meta global, o trabalho focou-se em objetivos específicos, começando pela retificação de erros detetados na versão original, de forma a assegurar uma simulação fiável. Em paralelo, estabeleceu-se como prioridade a adição de novas funcionalidades e a otimização da interface gráfica, garantindo um ambiente de trabalho mais fluido para o utilizador. Outro foco central consistiu na consolidação do Modo de Aquisição e Controlo Remoto, uma etapa essencial para assegurar uma comunicação estável, segura e sem falhas com os osciloscópios do laboratório.

Por fim, de modo a reforçar a vertente pedagógica do projeto, definiu-se a elaboração de guiões laboratoriais, criados com o intuito de facilitar a utilização da plataforma e promover o estudo e a exploração autónoma por parte dos estudantes.

1.3 Metodologia e Contribuições

A melhoria da plataforma seguiu uma abordagem incremental, estruturada em três fases principais: reestruturação da arquitetura, aprimoramento do modo de simulação e otimização do modo de aquisição e controlo remoto. O planeamento temporal e a execução destas tarefas encontram-se ilustrados no diagrama de Gantt da Figura 1.1.

Numa primeira etapa, eliminou-se a redundância de ficheiros através da centralização das dependências e introduziram-se mecanismos de contentorização via Docker, garantindo a total portabilidade do ambiente de execução. Estabilizada a infraestrutura, a segunda fase incidiu na correção de erros e na implementação de novas funcionalidades. Por fim, a terceira fase centrou-se na otimização da comunicação com o osciloscópio, corrigindo falhas e desenvolvendo uma camada de gestão para a ligação simultânea de múltiplos osciloscópios.

Reconhecendo que o ensino de conceitos complexos exige a conciliação entre a fundamentação teórica e a componente prática, o desenvolvimento orientou-se para a criação de um ambiente de experimentação seguro e interativo. Esta relevância prática já se verifica na unidade curricular de Teoria dos Circuitos (TCIRC) do ISEP, onde a aplicação atua como ferramenta de apoio à preparação laboratorial e elemento de avaliação.

O trabalho desenvolvido ao longo destas fases resultou num conjunto de melhorias que tornaram o *Oscilloscope Interface* mais robusto e intuitivo. Em síntese, as principais contribuições do projeto englobam:

- **Reestruturação e Aquisição Remota:** Consolidação da arquitetura de *software* num único diretório suportado por Docker e implementação de um seletor dinâmico que permite gerir e alternar remotamente entre múltiplos osciloscópios com *feedback* visual de estado.
- **Novas Funcionalidades de Simulação:** Correção das inconsistências no sistema de *Trigger* e introdução de novas funcionalidades, nomeadamente o Modo XY e *Custom Wave*.
- **Ferramentas de Apoio Pedagógico:** Criação de um sistema de diagnóstico rápido para mitigação de erros comuns e integração da secção *LabScripts*. Esta vertente didática foi materializada através do desenvolvimento de quatro guiões laboratoriais estruturados, focados na exploração do *Trigger*, Modo

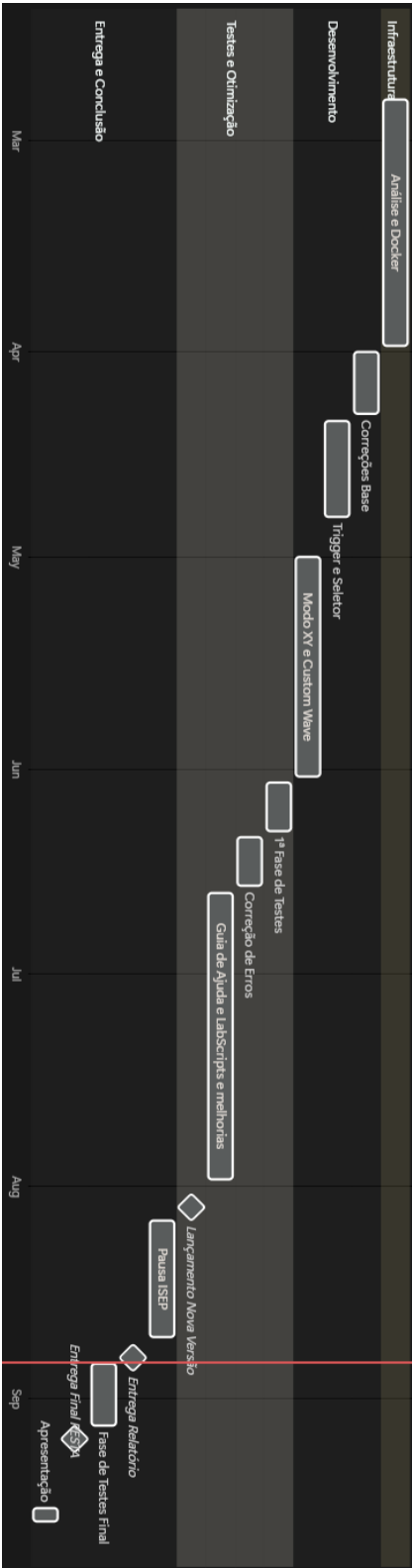


Figura 1.1: Cronograma do projeto

XY, *Custom Wave* e aquisição remota, permitindo guiar o utilizador na exploração integral da plataforma. O conteúdo integral destes guiões encontra-se disponível para consulta nos anexos do presente relatório

1.4 Estrutura do Relatório

No Capítulo 2, é descrito o contexto científico-tecnológico, abordando os conceitos fundamentais de instrumentação e circuitos elétricos, bem como a importância estratégica do ensino remoto e da virtualização laboratorial.

No Capítulo 3, é apresentada a abordagem de implementação, onde se detalham as escolhas tecnológicas, a arquitetura de software, as linguagens de programação e as ferramentas de suporte ao desenvolvimento.

No Capítulo 4, é detalhada a aplicação desenvolvida, onde se apresenta o guia de utilização da interface, demonstrando a intuitividade da ferramenta e a sua capacidade de simular o comportamento de um osciloscópio real.

No Capítulo 5, é realizada a avaliação e o teste dos resultados, descrevendo a metodologia de testes aplicada. Nesta secção, é feita uma análise crítica dos dados recolhidos, validando a funcionalidade e o impacto pedagógico do *Oscilloscope Interface*.

No Capítulo 6, são apresentadas as conclusões e o trabalho futuro, é feita a síntese das contribuições do projeto para o ISEP, refletindo sobre as limitações encontradas e delineando as diretrizes para a evolução do *Oscilloscope Interface* enquanto ferramenta.

Capítulo 2

Contexto científico-tecnológico

2.1 O que é e para que serve um osciloscópio?

Tal como é referido pela fabricante Tektronix, o osciloscópio atua como os "olhos do engenheiro", consistindo num instrumento de medição e diagnóstico cuja função principal é a aquisição e representação gráfica, em tempo real, de sinais elétricos que variam no tempo [1].

Conforme ilustrado na Figura 2.1, o equipamento exibe um gráfico, onde o eixo vertical (Y) quantifica a amplitude do sinal, expressa em tensão, o eixo horizontal (X) representa o tempo. Adicionalmente, é comum considerar-se uma terceira dimensão visual, o eixo Z, que corresponde à intensidade ou brilho do traço no ecrã [1].

O osciloscópio permite analisar diversas características de um sinal, nomeadamente [1]:

- A amplitude (tensão), bem como a medição direta do período e da frequência.
- O isolamento e avaliação da proporção exata entre a componente contínua (DC) e a componente alternada (AC).
- Detecção de anomalias operacionais através da identificação de distorções na forma de onda causadas por componentes eletrónicos defeituosos.
- A quantificação de ruído do sinal, monitorizando a evolução e o comportamento desse ruído ao longo do tempo.

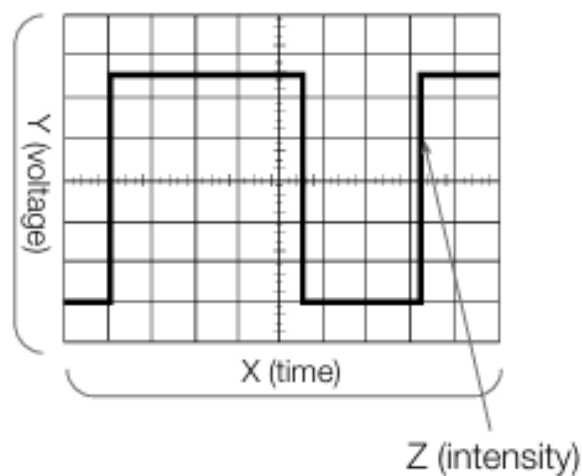


Figura 2.1: Componentes X, Y e Z de uma forma de onda [1].

Na Figura 2.2, é possível visualizar o osciloscópio presente nas bancadas do Laboratório de Circuitos e Sinais Elétricos:

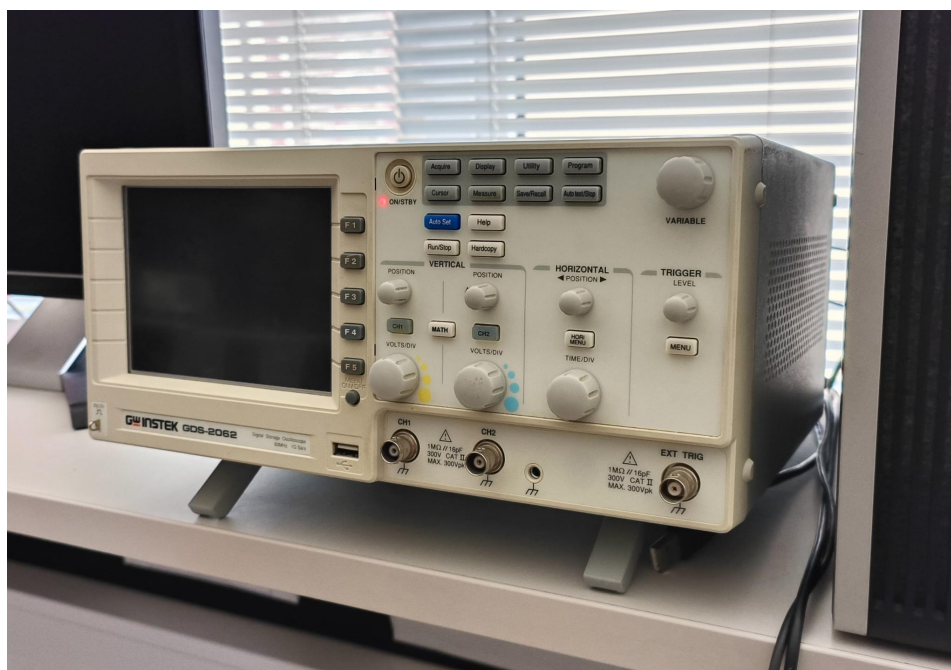


Figura 2.2: Osciloscópio do Laboratório de Circuitos e Sinais Elétricos do ISEP

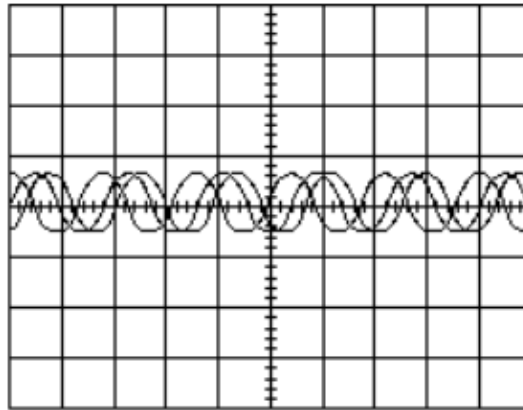


Figura 2.3: Display sem trigger [1]

2.1.1 Funcionalidades de um osciloscópio

Trigger

O sistema de trigger é um mecanismo essencial para estabilizar a aquisição de dados num osciloscópio, garantindo que as sucessivas amostras digitalizadas são temporalmente coordenadas na visualização da forma de onda [2]. Como representado na Figura 2.3, sem este alinhamento a um ponto de referência do sinal, a reconstrução gráfica da onda resultaria numa imagem instável e em aparente deslocamento horizontal [3].

Trigger Level e Slope

O *Trigger Level* define o valor de tensão que o sinal deve atingir para iniciar a captura. O *Trigger Slope* complementa esta definição ao especificar a polaridade da transição, *rising edge* ou *falling edge*, para efetuar o disparo [1]. Esta configuração assegura que a amostragem comece sempre na mesma fase do sinal de forma a que a onda visualizada fique estável (sincronizada) no ecrã [3]. Na Figura 2.4 é possível visualizar uma representação gráfica das funcionalidades.

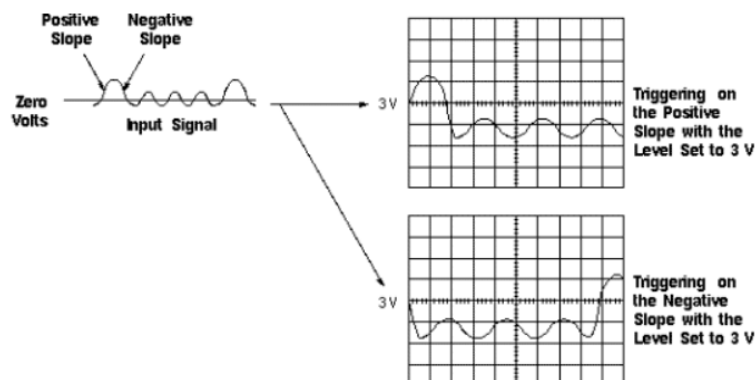


Figura 2.4: Trigger Level e Slope [1]

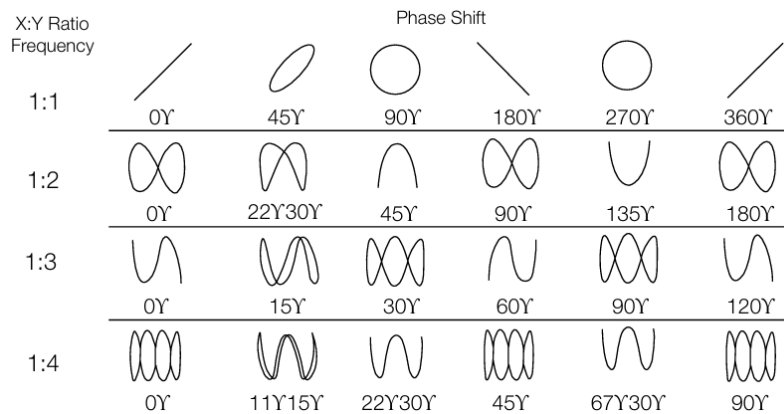


Figura 2.5: Figuras de Lissajous [1]

Modo XY

O modo XY distingue-se da representação convencional de um osciloscópio, na qual a amplitude de um sinal é traçada em função do tempo. Neste formato, a base de tempo interna é desativada, passando o ecrã a representar a relação direta entre as tensões de duas entradas distintas [1].

Esta funcionalidade pode ser utilizada para analisar o desfasamento e a relação de frequências entre dois sinais, ou o ciclo histórico de um circuito magnético. Como demonstrado na Figura 2.5, a composição de dois sinais sinusoidais em modo XY origina as figuras de Lissajous [1]. A análise destas figuras permite determinar com precisão a diferença de fase entre as ondas. Adicionalmente, este modo é uma ferramenta essencial para a análise de curvas características de componentes, como é o caso da análise da relação corrente-tensão num díodo.

2.2 Aprendizagem Remota

2.2.1 Trabalhos anteriores

O desenvolvimento de plataformas virtuais de apoio ao ensino laboratorial no ISEP tem sido alvo de uma evolução contínua, verificando-se já a existência de simuladores de osciloscópio anteriores, os trabalhos de Pedro Salgueiro [4] e João Pereira [5]. O presente projeto surge como uma continuação direta do trabalho desenvolvido por João Ferreira em 2023, intitulado "Projeto e desenvolvimento de aplicação web para simulação e interface de Osciloscópio" [6].

Nesse projeto, foi estabelecida a arquitetura base de uma aplicação web assente numa topologia cliente-servidor (utilizando Node.js, JavaScript, *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS)) capaz de simular e controlar remotamente o osciloscópio GW-Instek GDS2062. Apesar de ter sido estabelecida

uma base sólida de comunicação e simulação, o projeto anterior identificou várias limitações e perspectivas de trabalho futuro.

2.2.2 Importância do Ensino Remoto

A digitalização das metodologias de ensino superior tem sofrido uma aceleração sem precedentes, impulsionada pelas restrições de mobilidade impostas pela pandemia de COVID-19. Este período obrigou a uma transição súbita para o ensino à distância, expondo a necessidade crítica de ferramentas que assegurassem a continuidade pedagógica, nomeadamente em áreas intrinsecamente experimentais como a Engenharia Eletrotécnica [7]. No contexto específico do ISEP, cujo modelo pedagógico assenta fortemente numa matriz prática, esta imposição exigiu uma adaptação imediata. Para mitigar o impacto da suspensão das aulas presenciais, tornou-se imperativa a adoção e o desenvolvimento de laboratórios remotos e simuladores, ferramentas que garantiram que os estudantes mantivessem o contacto com a realidade prática da instrumentação e análise de circuitos, mesmo à distância.

No âmbito desta transição torna-se essencial distinguir duas abordagens complementares de ensino remoto: a componente síncrona e a assíncrona [8]. O ensino remoto síncrono caracteriza-se pela interação em tempo real, promovendo um acompanhamento imediato e feedback direto. Em contexto laboratorial, isto traduz-se, por exemplo, no controlo remoto de equipamentos físicos durante o período normal de aulas, com o apoio do docente. Em contrapartida, o ensino remoto assíncrono oferece total independência temporal, permitindo uma maior flexibilidade por parte do aluno [8]. Esta modalidade permite ao estudante aceder a simuladores e plataformas digitais fora do horário letivo, promovendo o estudo autónomo e ao seu próprio ritmo.

A conjugação destes paradigmas no ecossistema académico atual trouxe vantagens estruturais incontornáveis, destacando-se:

- **Flexibilidade de Acesso (Dinâmica Assíncrona):** O ensino remoto elimina barreiras geográficas e temporais. A possibilidade de aceder a recursos laboratoriais fora dos horários presenciais permite que o estudante consolide conceitos complexos de forma independente, mitigando as limitações físicas e os horários restritos dos laboratórios tradicionais [9].
- **Experimentação sem Risco:** A simulação permite um ciclo contínuo de tentativa e erro num ambiente virtual seguro. Esta característica encoraja a exploração de limites operacionais e a validação teórica sem o risco de danificar equipamentos físicos dispendiosos por má utilização de quem está a aprender [9].

- **Otimização do Tempo Presencial:** Estas ferramentas não visam substituir o trabalho físico de bancada, mas sim complementá-lo e otimizá-lo. Ao garantir que o aluno chegue ao laboratório já familiarizado com a interface e funcionamento do equipamento (através da preparação prévia assíncrona), o tempo presencial letivo torna-se significativamente mais produtivo, sendo inteiramente focado na montagem, resolução de problemas reais e análise crítica [9].

2.3 Virtualização Laboratorial

A virtualização laboratorial consolidou-se como uma solução estratégica para modernizar o ensino, superando barreiras como o elevado custo de equipamentos, riscos de segurança e restrições de acessibilidade física [9]. Esta tendência abrange domínios em áreas diversas, utilizando soluções digitais que tentam replicar o rigor científico da experimentação real.

2.3.1 Ecossistemas de Instrumentação Virtual na Engenharia Eletrotécnica

Virtual Instrument Systems in Reality (VISIR)

O sistema VISIR demarca-se por não ser um simulador tradicional, mas sim um laboratório remoto [10]. Este paradigma permite ao estudante aceder a uma matriz de comutação que controla componentes eletrónicos e instrumentos de medida reais à distância. A principal mais-valia reside na capacidade de observar ruído real, tolerâncias dos componentes e comportamentos físicos não ideais, oferecendo uma experiência de experimentação autêntica sem a necessidade de presença física no laboratório.

LabVIEW

O LabVIEW, desenvolvido pela National Instruments, representa um standard na indústria para a aquisição de dados e automação industrial [11]. Através de uma linguagem de programação gráfica, o aluno constrói "Instrumentos Virtuais" ligando blocos funcionais. Este ambiente permite ao estudante focar-se na lógica de controlo e no processamento de sinais, simulando a integração de sensores e atuadores antes da sua implementação em hardware físico.

NI Multisim

No contexto do projeto e análise rigorosa de circuitos elétricos e eletrónicos, o NI Multisim afirma-se como um paradigma de simulação baseado no standard industrial SPICE, com uma forte ênfase na emulação de bancada [12]. Ao contrário dos

simuladores puramente matemáticos ou das plataformas de prototipagem visual elementar, este software destaca-se por integrar modelos e painéis frontais fidedignos de instrumentos de medição reais (incluindo osciloscópios e geradores de sinais de fabricantes de referência. Esta abordagem permite ao estudante desenhar o esquema elétrico e interagir com os equipamentos virtuais através de controlos físicos emulados, reproduzindo a operação real dos comandos.

2.4 Tecnologias e plataformas de apoio para o projeto de software

2.4.1 Desenvolvimento da Aplicação Web

A arquitetura do *Oscilloscope Interface* assenta num modelo client-servidor. O funcionamento da plataforma exige que a interface no navegador do utilizador (o *frontend*) garanta uma resposta imediata e fluida à interação do utilizador com os painéis de controlo, assegurando o ajuste dos parâmetros do equipamento ocorra sem atrasos. Simultaneamente, esta interface precisa de comunicar de forma eficiente com o servidor (o *backend*) para gerir operações mais complexas, como a aquisição de dados reais vindos de portas série.

Como a plataforma é uma continuação do trabalho [6], a escolha destas tecnologias não partiu do zero. Todo o processo foi orientado por um princípio de continuidade. Assim, foi possível garantir que o projeto continuava a evoluir sobre uma fundação sólida e sustentada por ferramentas amplamente reconhecidas na área do desenvolvimento web, tal como o Node.js para a gestão do servidor, o *JavaScript* (JS) para assegurar toda a lógica e interatividade do simulador, e o *Embedded JavaScript Templating* (EJS) para a construção dinâmica das páginas.

2.4.2 Linguagens e Bibliotecas

O HTML e o CSS serviram de base para a estruturação e o design da interface da aplicação. O HTML definiu a organização dos painéis e visores no ecrã, enquanto o CSS garantiu toda a personalização visual, essencial para replicar a aparência de um instrumento real.

O JS assumiu o papel principal na lógica do simulador. Sendo uma linguagem amplamente utilizada na criação de aplicações web dinâmicas e interativas, o JS beneficia de um ecossistema muito rico, do qual se destaca o Node.js, o ambiente escolhido para executar o servidor neste projeto.

Para complementar estas tecnologias de base, o desenvolvimento recorreu a várias bibliotecas e *frameworks* que facilitaram a implementação das diferentes funcionalidades do simulador, cada uma com um contributo específico:

- **Express e Nodemon:** O Express [13] estruturou a gestão de rotas e pedidos no servidor, enquanto o Nodemon [14] agilizou o processo de programação ao reiniciar automaticamente a aplicação.
- **EJS:** Permitiu a construção dinâmica das páginas apresentadas ao utilizador [15].
- **jQuery e *Data-Driven Documents* (D3):** O jQuery [16] simplificou a manipulação dos botões e eventos da interface. Por sua vez, o D3 [17] assumiu a tarefa crucial de renderizar os gráficos dos sinais com fluidez.
- **Serialport:** Possibilitou a comunicação remota, permitindo ao servidor aceder às portas de comunicação físicas [18].
- **JavaScript Object Notation (JSON):** Foi adotado como o formato padrão para a troca de dados entre o navegador e o servidor [19].
- **Math.js:** Uma integração fundamental para a nova funcionalidade de geração de ondas (Custom Wave), permitindo avaliar e calcular as equações matemáticas introduzidas pelo utilizador [20].

2.4.3 Notion e Github

O desenvolvimento de um projeto de software, especialmente quando se trata de reestruturar e expandir uma aplicação já existente, exige uma organização rigorosa. Para garantir que o trabalho no *Oscilloscope Interface* decorresse de forma planeada, o processo baseou-se na utilização de duas ferramentas complementares: o Notion [21] e o GitHub [22].

Gestão de Projeto no Notion

O Notion [21] funcionou como o espaço central para a organização diária do projeto. Foi nesta plataforma que se estruturaram as tarefas a realizar, através de uma base de dados, permitindo um acompanhamento visual do progresso de cada etapa de desenvolvimento. O seu uso foi essencial para registar os erros encontrados ao longo do projeto, bem como planear as respetivas correções, detalhar os requisitos técnicos das novas funcionalidades e todas as reuniões realizadas ao longo do projeto. O Notion garantiu que os objetivos práticos não se perdiam ao longo do semestre.

Controlo de Versões no GitHub

Por sua vez, o [22] foi a ferramenta dedicada ao alojamento do código e ao controlo de versões. Sendo o *Oscilloscope Interface* um projeto em contínua evolução, o uso de um sistema de controlo de versões revelou-se de extrema importância. Esta plataforma permitiu guardar o histórico de todas as alterações efetuadas no código.

Serviu também como um porto seguro para testar diferentes abordagens de programação antes de as integrar na versão final da aplicação.

Resumindo, a conjugação destas duas plataformas foi fundamental para o sucesso do trabalho. Enquanto o Notion assegurou a clareza do planeamento e a documentação das ideias, o GitHub garantiu um ambiente de desenvolvimento seguro, organizado e preparado para futuras atualizações.

2.4.4 Plataformas de programação utilizadas

Ambiente de desenvolvimento

O processo de escrita de código e a gestão dos ficheiros do projeto foram centralizados no *Visual Studio Code* (VS Code) [23]. A escolha deste ambiente de desenvolvimento deveu-se à sua leveza, vasta capacidade de personalização através de extensões e excelente suporte nativo para o ecossistema JS e Node.js. O VS Code revelou-se fundamental não só para a edição de código, mas também para o processo de *debugging* e para a integração direta com o sistema de controlo de versões no GitHub, permitindo gerir os *commits* sem necessidade de alternar entre janelas.

Ferramentas de *Inteligência Artificial* (IA)

No panorama atual do desenvolvimento de software, a integração de ferramentas baseadas em IA tornou-se uma prática comum para aumentar a produtividade e auxiliar na resolução de obstáculos técnicos. Para apoiar o desenvolvimento contínuo do simulador e a elaboração da respetiva documentação, recorreu-se a duas ferramentas de IA.

GitHub Copilot

Integrado diretamente no VS Code, o Copilot [24] atuou como um assistente de programação em tempo real. A sua utilização focou-se principalmente na sugestão de estruturas sintáticas padronizadas. Outras funcionalidades exploradas incluíram a geração de código a partir de comentários de texto e a capacidade de autocompletar blocos de código recém-iniciados. Se no passado, perante a complexidade de um projeto deste tipo, estas tentativas de previsão podiam gerar sugestões pouco precisas e interromper o raciocínio, atualmente a realidade é outra. O modelo encontra-se muito mais desenvolvido e a sua capacidade de interpretar o contexto global da aplicação evoluiu significativamente. Desta forma, as sugestões tornaram-se muito mais assertivas e úteis, acelerando o desenvolvimento em vez de o atrapalhar, permitindo ao programador focar-se mais na lógica da aplicação e não tanto na estética da mesma.

Gemini

O Gemini [25] foi adotado como uma ferramenta interativa de suporte à pesquisa e estruturação de ideias. Auxiliou na clarificação de conceitos teóricos complexos (como as abordagens matemáticas necessárias para a renderização das formas de onda), no planejamento lógico da integração de novas bibliotecas e na revisão e estruturação formal dos textos redigidos para os guiões laboratoriais e para o relatório final do projeto.

Capítulo 3

Abordagem de Implementação

3.1 Arquitetura de software e modos de utilização

Como o desenvolvimento deste simulador consistiu no aprimoramento de um trabalho realizado anteriormente, uma das primeiras fases do projeto focou-se na análise e otimização da arquitetura base do código. A estrutura original da aplicação (Figura 3.1) apresentava uma divisão física rígida entre os diretórios de *backend* e *frontend*. Esta abordagem resultava numa redundância indesejada, evidenciada pela presença de vários ficheiros *package.json* e pastas *node_modules* espalhadas pelo projeto.

Para resolver estas repetições e facilitar a manutenção da aplicação, procedeu-se a uma reestruturação do código, resultando na arquitetura visível na Figura 3.2.

Principais alterações na estrutura

Nesta nova organização, a repetição foi eliminada através da adoção de um único ficheiro *package.json* e *package-lock.json* na raiz do projeto. Esta alteração centraliza a instalação de todos os pacotes Node.js, reduzindo drasticamente o tamanho global do projeto e evitando conflitos de versões entre as bibliotecas do servidor e da interface. Adicionalmente, todo o código fonte da aplicação foi consolidado dentro do diretório principal *app*. No seu interior, os ficheiros foram categorizados de acordo com a sua responsabilidade. As pastas *controllers* e *routes* separam as funções de controlo do encaminhamento dos URLs. Por sua vez, a pasta *views* agrupa todos os ficheiros EJS responsáveis pela renderização da interface gráfica do utilizador,

enquanto a diretoria *public* centraliza os recursos estáticos que são enviados diretamente para o *browser*, tais como o CSS, os scripts executados no cliente, as imagens e os ficheiros de configuração das formas de onda.

Para além da reorganização lógica, a nova estrutura introduz ficheiros fundamentais como o *Dockerfile*, *docker-compose.yml* e *.dockerignore* na raiz do projeto. Esta adição é um passo crucial, pois permite que o simulador esteja num contentor, garantindo que a aplicação corre de forma isolada e consistente em qualquer máquina, independentemente do sistema operativo subjacente. Por fim, foi ainda adicionado o ficheiro *.gitignore* para assegurar que ficheiros pesados gerados automaticamente ou dados sensíveis não são submetidos para o repositório no Github.

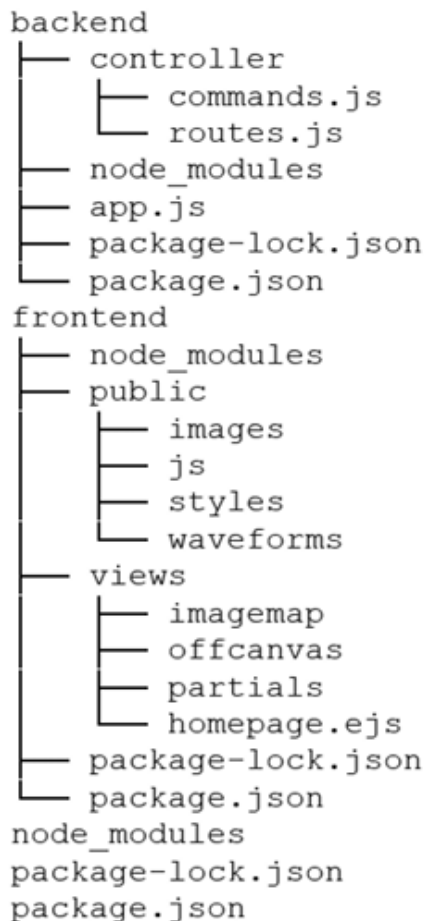


Figura 3.1: Antiga estrutura do código

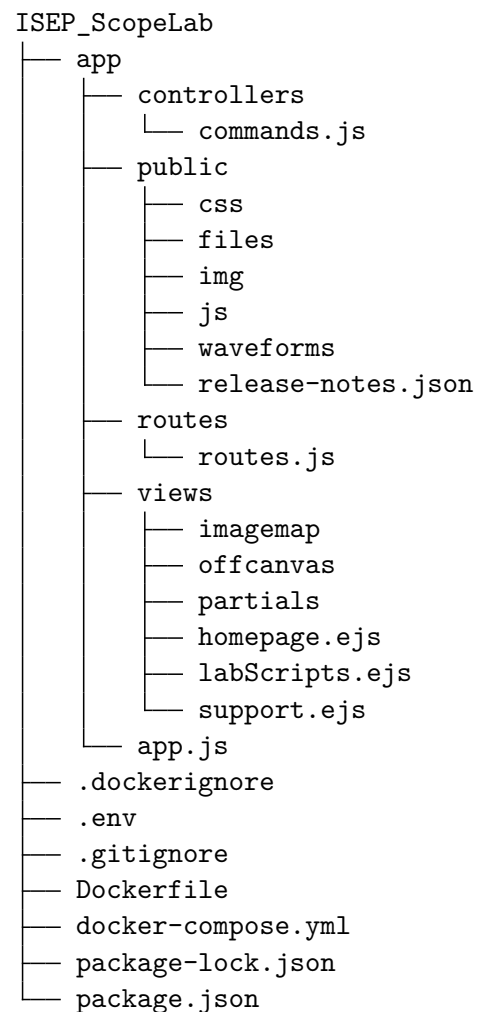


Figura 3.2: Nova estrutura

3.1.1 Alterações - Modo simulação

O Modo Simulação foi melhorado para corrigir erros da versão anterior e incluir novas ferramentas. Um dos focos principais foi o sistema de trigger, já que antes o ajuste do *slope* e do *trigger level* não funcionavam bem em ambos os canais. O código da aplicação foi ajustado para que a aplicação obedeça agora corretamente aos comandos da interface, permitindo estabilizar a imagem da onda sem falhas. Além desta correção, adicionou-se o Modo XY. Esta função altera a forma habitual de ver os gráficos, cruzando as tensões dos dois canais de forma direta para desenhar as Figuras de Lissajous [1], o que é essencial para analisar desfasamentos. Por último, o simulador ganhou a funcionalidade *Custom Wave*. Em vez de estar limitado a ondas mais básicas, tradicionalmente disponibilizadas nos geradores de sinais de baixa gama, o utilizador pode agora criar ondas totalmente personalizadas através de expressões matemáticas matemáticas, além da derfinição ponto a popnto, construindo manualmente o ficheiro Excel, que já existia na versão anterior, tornando a plataforma muito mais versátil para realizar diferentes experiências de laboratório.

3.1.2 Alterações - Modo aquisição e controlo

No Modo de Aquisição e Controlo, o trabalho inicial focou-se na resolução de *bugs* que impediam a visualização das ondas em ambos os canais, garantindo uma vizualização correta e estável dos sinais reais. No entanto, a principal inovação introduzida neste modo foi a criação de um seletor de osciloscópios. Esta nova funcionalidade tira partido da possibilidade de ligar vários osciloscópios físicos ao computador em simultâneo. Através da interface da aplicação, o utilizador pode agora abrir um menu e escolher exatamente qual dos equipamentos pretende controlar e visualizar no ecrã naquele momento. Esta melhoria torna a utilização do laboratório remoto muito mais prática e flexível, permitindo gerir e alternar entre múltiplos postos de trabalho a partir de uma única plataforma.

3.2 Aspetos de implementação

Em relação à implementação, o objetivo principal foi resolver falhas de funcionamento importantes e, ao mesmo tempo, adicionar novas capacidades de simulação. Houve sempre o cuidado de garantir que cada nova alteração ou adição de código não estragava o que já estava feito, mantendo a interface web rápida e a responder bem aos comandos do utilizador.

Trigger

Para resolver a instabilidade do trigger e um erro na apresentação do segundo canal. Percebemos que algumas ações no painel, como mudar o botão de declive (*slope*), só

estavam a atualizar a informação do primeiro canal. Isto causava uma leitura errada e fazia com que o sinal do segundo canal ficasse invertido de forma incorreta. A solução passou por reestruturar essa parte do código, garantindo que as atualizações de estado e de sincronismo são enviadas e aplicadas aos dois canais ao mesmo tempo, o que permitiu estabilizar as formas de onda.

Custom Wave

Para a implementação do modo de *Custom Wave* (onda personalizada), procedeu-se à criação de três novos ficheiros para manter o projeto bem organizado e estruturado: o *calculator.ejs* para a interface, o *calculator.css* para a componente visual e o *calculator.js* para a lógica de funcionamento.

Esta funcionalidade foi desenvolvida dentro de um módulo *JavaScript* isolado, utilizando um padrão conhecido como *Immediately Invoked Function Expression* (IIFE). Esta abordagem é bastante útil porque mantém todo o estado interno da ferramenta, como variáveis e dados, completamente separado, evitando que se misture ou cause conflitos com o resto do código do simulador. Este módulo guarda informações sobre o estado atual, como o modo escolhido (simples ou avançado) e os dados do último gráfico gerado. Estabeleceu-se também que qualquer onda gerada tem sempre uma resolução fixa de 251 amostras.

Para avaliar as equações matemáticas introduzidas pelo utilizador, recorreu-se à biblioteca *math.js*. Quando uma equação é submetida, o sistema verifica imediatamente se existem erros de sintaxe; caso existam, é apresentado um aviso ao utilizador. Sendo a equação válida, o módulo analisa o texto para extrair a frequência, como é possível ver na Figura 3.3.

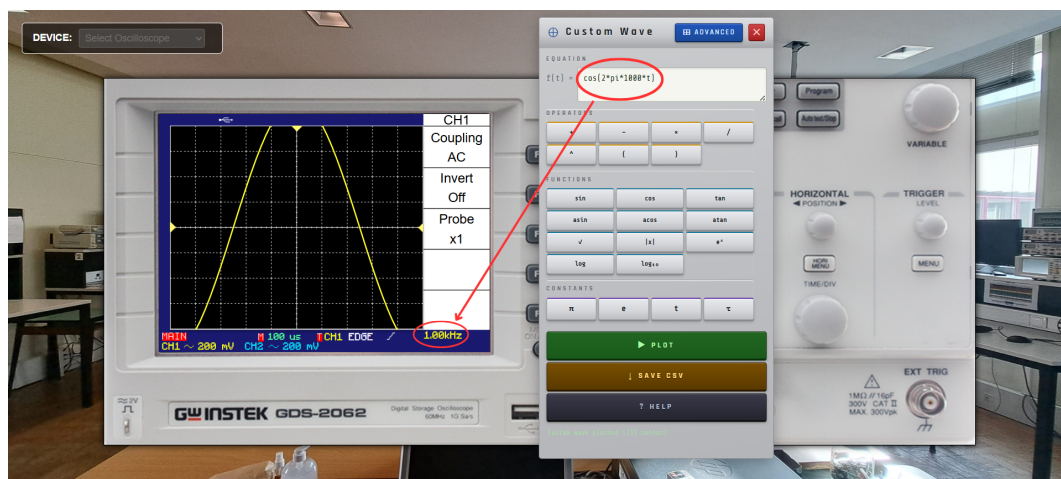


Figura 3.3: Atualização da frequência

Após o cálculo, os dados são injetados diretamente no canal principal do osciloscópio (CH1) e o gráfico é atualizado no ecrã. Em simultâneo, o sistema calcula e

exibe métricas úteis da onda, como a tensão pico-a-pico (V_{pp}), o valor eficaz (V_{rms}), os picos máximos e mínimos, bem como as respetivas escalas de tempo e tensão.

Para tornar a ferramenta mais completa, foi implementada a possibilidade de exportar os dados da onda para um ficheiro *Comma-Separated Values* (CSV). Este processo decorre inteiramente no browser, sem necessidade de comunicação com o servidor, gerando um ficheiro com o tempo e a amplitude, como na Figura 3.4.

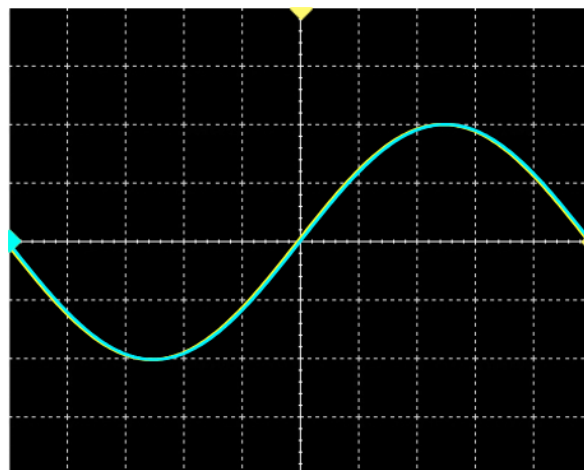
	A	B
1	Time,Amplitude	
2	0,1.0000	
3	0.000004,0.9997	
4	0.000008,0.9987	
5	0.000012,0.9972	
6	0.000016,0.9950	
7	0.00002,0.9921	
8	0.000024,0.9887	

Figura 3.4: Ficheiro CSV

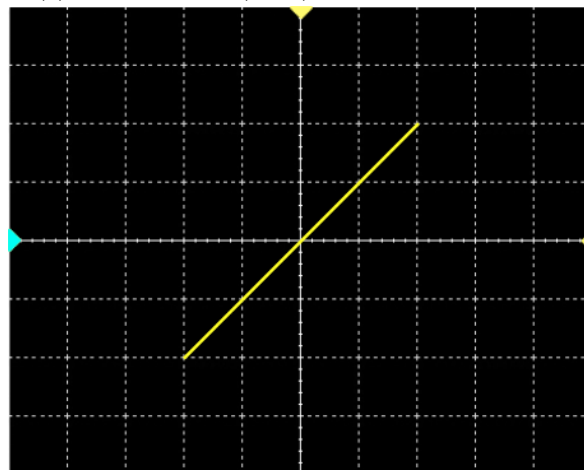
Modo XY

A implementação do modo XY diverge significativamente do funcionamento padrão do simulador, assentando a sua execução em duas funções principais: `updateChartXY()` e `showGraphXY()`. Ao contrário do modo *main* (ilustrado na Figura 3.5a), que representa a variação de tensão em função do tempo, esta lógica é substituída por uma análise bidimensional de sinais (visível na Figura 3.5b). Ou seja, passa a representar diretamente a tensão do primeiro canal (CH1) no eixo horizontal em função da tensão do segundo canal (CH2) no eixo vertical.

O processamento dos dados tem início na função `updateChartXY()`, que verifica obrigatoriamente se ambos os canais apresentam exatamente o mesmo número de amostras, garantindo assim a coerência na formação dos pares de coordenadas. Validada esta condição, é construída uma nova estrutura de dados onde cada elemento agrupa os valores de amplitude de ambos os sinais em simultâneo, descartando-se a informação do tempo original. Posteriormente, a função `showGraphXY()` utiliza este conjunto de dados para reconfigurar e redesenhar o gráfico *Scalable Vector Graphics* (SVG) do osciloscópio. Nesta etapa, são estabelecidos dois eixos de tensão distintos, os quais são dimensionados de forma independente em conformidade com as escalas verticais previamente definidas para cada canal. A representação final é então concretizada através da biblioteca D3.js, que processa a informação e traça a curva resultante a amarelo, completando a visualização no ecrã.



(a) Modo "normal"(main) - base de tempo ligada



(b) Modo XY - CH1 no x e CH2 no y

Figura 3.5: Exemplo de utilização modo XY

Seletor de Oscoscópios

A implementação do seletor assenta numa comunicação entre a interface gráfica e o servidor, cujo fluxo de execução se encontra ilustrado no diagrama da Figura 4.19. Na interface, o utilizador interage com um menu pendente, designado internamente por `scopeSelect`, que lista os equipamentos disponíveis para conexão. Ao selecionar uma das opções, é desencadeada a função `changeOscilloscope()`. Com o intuito de fornecer um retorno visual instantâneo ao utilizador, o sistema altera de imediato o indicador de estado para "Connecting..." (a cor laranja) enquanto a opção é validada. Simultaneamente, é enviado um pedido *Hypertext Transfer Protocol* (HTTP) POST para a rota `/change-port` no servidor, contendo o identificador único do osciloscópio escolhido.

Do lado do servidor, a rota `/change-port`, processa este pedido, convertendo o identificador recebido no caminho da porta de comunicação série correspondente, por

exemplo, mapeando o identificador 0 para a porta COM9, o 1 para COM10, e assim sucessivamente. Após este mapeamento, é invocada a rotina `switchSerialPort()`, que garante o encerramento de qualquer ligação pré-existente antes de inicializar a nova porta série com a taxa de transmissão previamente configurada.

O fluxo de comunicação termina com o tratamento da resposta do servidor na interface gráfica, operação gerida pela função `handleResponse()`. Esta lógica está preparada para lidar com três cenários de operação distintos. No caso de uma ligação bem-sucedida, o indicador visual transita para "Online" (a cor verde) e o seletor mantém-se ativo. Se a porta série solicitada não for encontrada no servidor, o estado é atualizado para "Offline" (a cor vermelho) e o elemento de seleção é desabilitado por precaução. Por fim, caso o sistema detete que o equipamento físico se encontra desligado ou inacessível, é acionado um alerta para o utilizador.

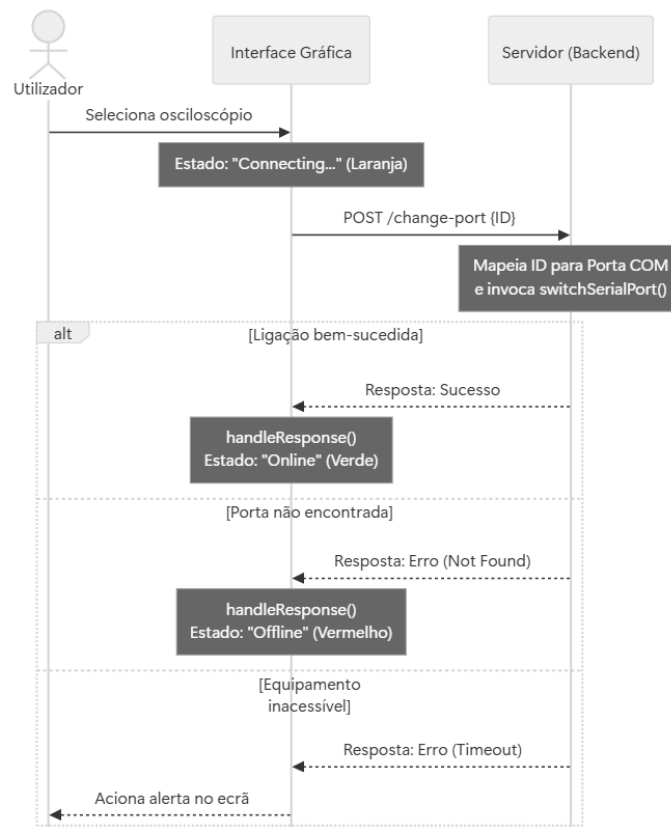


Figura 3.6: Diagrama de sequência do seletor de osciloscópios

Capítulo 4

Guia de utilização da aplicação

4.1 Página Principal

O acesso ao simulador é realizado de forma simples e direta, sem necessidade de instalar qualquer software adicional no computador. Para aceder à aplicação, basta abrir um navegador web comum e introduzir o endereço <http://osciloscopio.dee.isep.ipp.pt>. Ao abrir o link irá aparecer a página representada na Figura 4.1.



Figura 4.1: Interface

4.2 Principais funcionalidades da aplicação

Esta secção descreve as funcionalidades centrais do *Oscilloscope Interface*. O objetivo destas ferramentas é garantir que o simulador não só replica fielmente o comportamento de um instrumento físico, como também tira partido do ambiente virtual para oferecer uma experiência educativa completa e intuitiva.

Para facilitar a análise do estado atual da aplicação, encontra-se uma lista das alterações efetuadas na Tabela 4.1, que serão detalhadas nas subsecções seguintes.

Tabela 4.1: Atualização das funcionalidades da aplicação.

Funcionalidades	Estado
Menu <i>trigger</i>	✓
Nível do <i>trigger</i>	✓
Modo XY	✓
<i>Custom Wave</i>	✓
Seletor de dispositivo	✓
Página <i>Support</i>	✓
Página <i>LabScripts</i>	✓
<i>Real Oscilloscope</i>	✓
<i>Guia de Resolução de Problemas</i>	✓

4.2.1 Guia de Resolução de Problemas

A pensar nos utilizadores com pouca experiência na operação de osciloscópios de bancada, como é o caso dos alunos do primeiro ano da LEEC, a plataforma integra um sistema de ajuda contextual e interativo.

Acessível a qualquer momento através do ícone de ajuda no canto inferior esquerdo da interface, este menu apresenta um "Diagnóstico Rápido de Problemas" (Figura 4.3). O objetivo principal desta ferramenta é promover a autonomia dos estudantes, permitindo-lhes ultrapassar bloqueios comuns sem necessitarem de intervenção imediata do professor.

Para facilitar a compreensão, o guia lista os problemas sob a forma de sintomas descritos numa linguagem simples e direta (por exemplo, "Não aparece nada no ecrã" ou "A onda está instável"). Ao interagir com um destes sintomas, o menu expande-se para fornecer instruções passo a passo para a sua resolução (Figura 4.2).

Tendo como exemplo o cenário em que o utilizador seleciona a opção "A tela está reta (linha zero)", o sistema sugere de imediato verificações fundamentais de operação: instrui o aluno a confirmar se o acoplamento do canal não foi acidentalmente deixado na posição GND (Terra) e, em seguida, aconselha o ajuste da escala no seletor Volts/Div para tentar intercepar o sinal. Desta forma, o simulador atua também como uma ferramenta pedagógica, ensinando as boas práticas de despiste de falhas típicas na utilização de um osciloscópio.

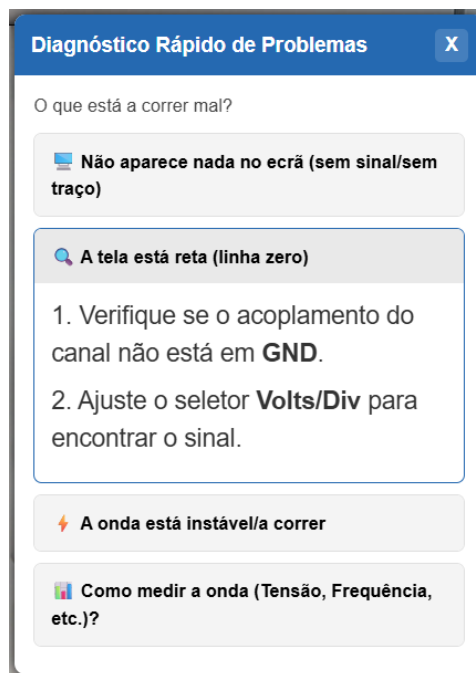


Figura 4.2: Exemplo das sugestões de diagnóstico apresentadas



Figura 4.3: Menu interativo de diagnóstico rápido de problemas na interface do simulador.

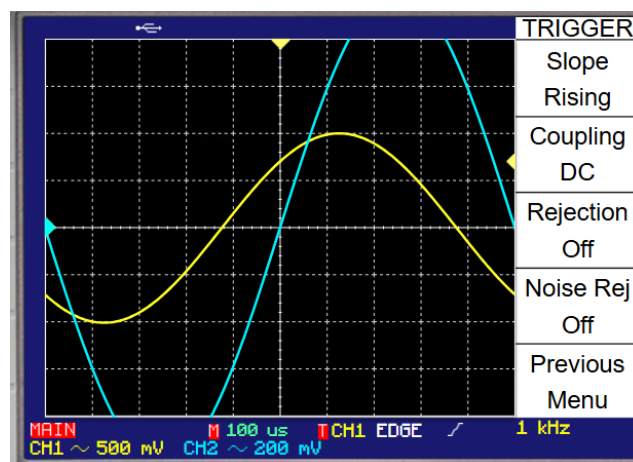
4.2.2 Trigger Level

Na interface do simulador, a interação com o painel de Trigger foi desenhada para replicar a experiência tátil de um equipamento físico, mas adaptada ao ambiente digital. Para ajustar o nível de disparo, o utilizador deve interagir com o botão rotativo abaixo de Trigger Level.

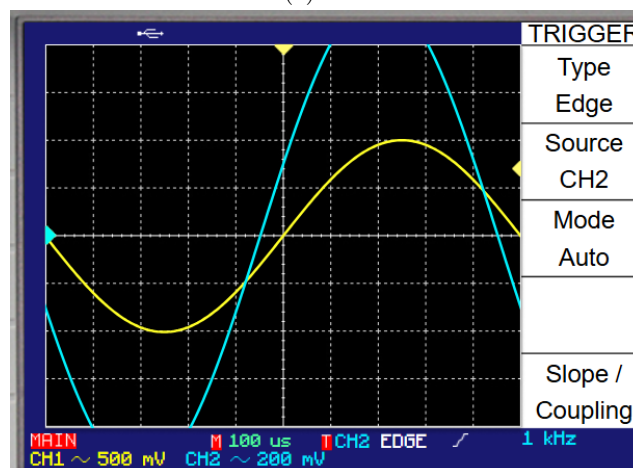
A operação realiza-se através de cliques com o rato diretamente sobre o botão:

- **Clique esquerdo:** Simula a rotação do comando num sentido, alterando o nível de tensão de referência passo a passo numa direção.
- **Clique direito:** Simula a rotação em sentido inverso, permitindo o ajuste do nível na direção oposta.

Como ilustrado nas imagens 4.4a e 4.4b, o impacto deste ajuste é imediatamente visível. Quando o nível de disparo se encontra devidamente alinhado com a amplitude do sinal capturado, seja através da fonte do Canal 1 (CH1) ou do Canal 2 (CH2), a onda estabiliza, permitindo a sua análise.



(a) CH1



(b) CH2

Figura 4.4: Efeito do Trigger Level em cada forma de onda

4.2.3 Trigger Slope

O parâmetro Trigger Slope define a direção em que o sinal tem de estar a cruzar essa mesma tensão para que a captura se inicie.

Na interface do simulador, o utilizador pode alternar entre as duas configurações principais de inclinação:

- **Rising (subida):** O disparo ocorre quando a tensão do sinal cruza o nível de referência de baixo para cima. Como é possível observar na Figura 4.5, o traçado da onda selecionada no ponto de disparo (identificado pelo cursor triangular no ecrã) inicia-se numa trajetória ascendente.
- **Falling (descida):** O disparo ocorre quando o sinal cruza o nível de referência de cima para baixo. A Figura 4.6 ilustra este comportamento com clareza; tanto na captura com base no Canal 1 (4.6a) como no Canal 2 (4.6b), verifica-se que no ponto exato de disparo a onda encontra-se numa trajetória de descida.

Para fazer essa alteração no simulador, o utilizador deve selecionar o botão "MENU" embaixo de Trigger Level, logo de seguida em F5 e F1, como está representado na Figura 4.7, tal como se faz no equipamento real.

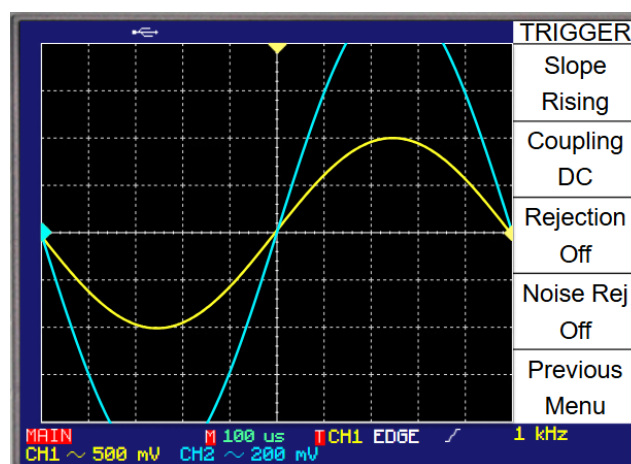
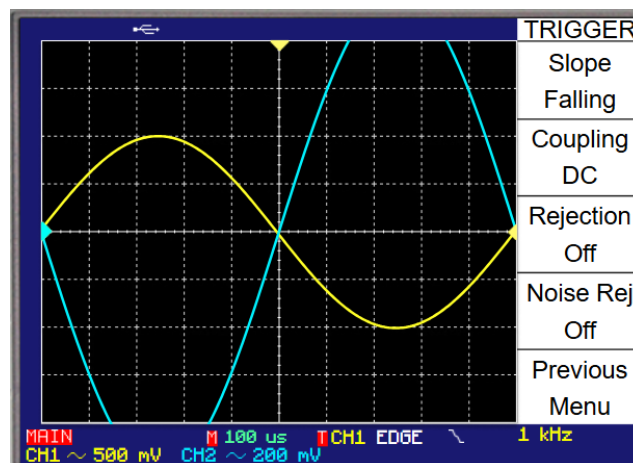
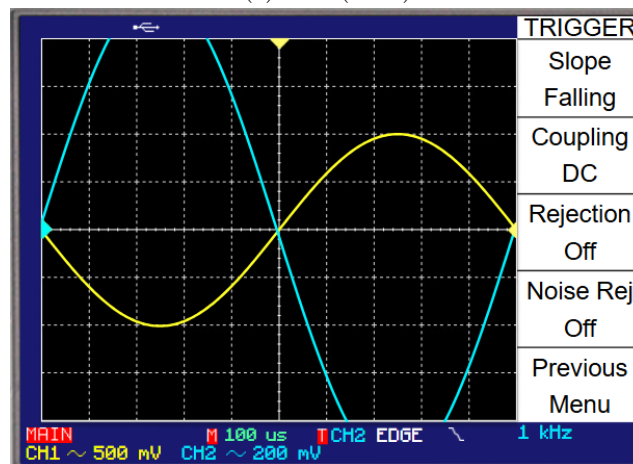


Figura 4.5: Trigger slope - rising



(a) CH1 (fonte)



(b) CH2 (fonte)

Figura 4.6: Trigger Slope - Falling

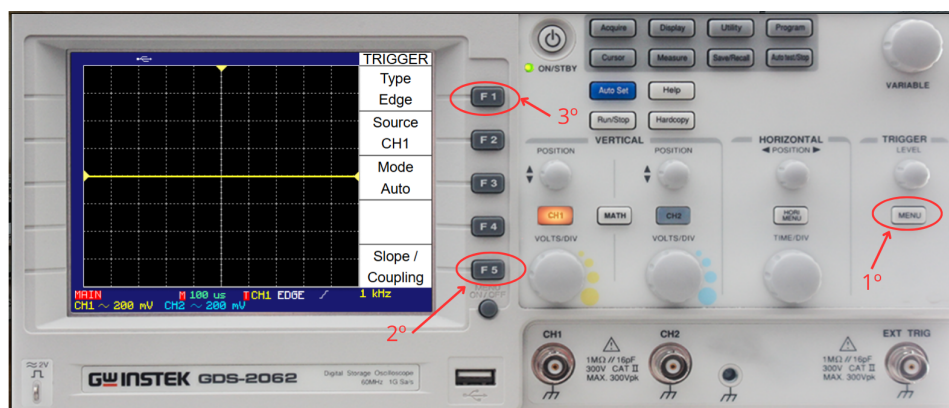


Figura 4.7: Comandos Slope

4.2.4 Modo XY

Para ativar o Modo XY na interface do simulador, o utilizador deve reproduzir as ações realizadas no equipamento físico. Primeiramente, é necessário selecionar o botão "HORI MENU", localizado na secção de controlo horizontal do painel. Esta ação faz surgir um menu de visualização na lateral direita do ecrã. De seguida, o utilizador deve pressionar o botão "F5", que corresponde diretamente à opção "XY" desse mesmo menu, conforme evidenciado na Figura 4.8.

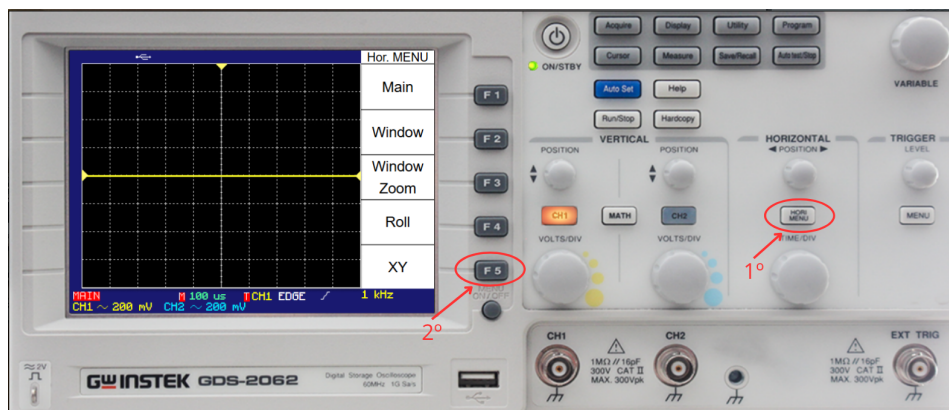
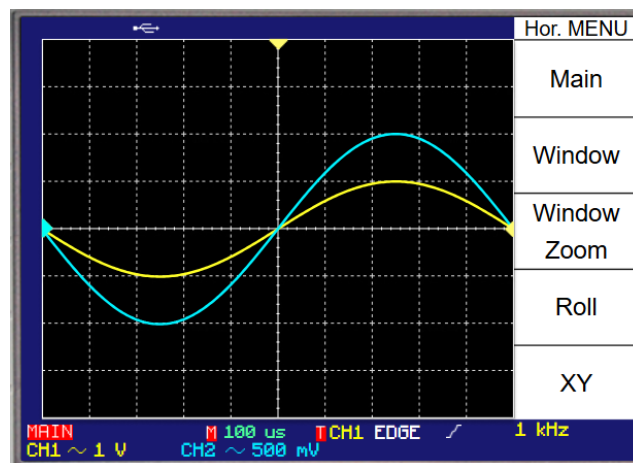


Figura 4.8: Comandos modo XY

Ao nível prático, a ativação deste modo altera imediatamente o comportamento do ecrã. A grelha deixa de desenhar as ondas da esquerda para a direita ao longo do tempo. Em vez disso, o eixo horizontal passa a ser diretamente controlado pela tensão do sinal introduzido no Canal 1, e o eixo vertical pela tensão do Canal 2, como pode ser verificado nas Figuras 4.9 e 4.10.



(a) Visualização de sinais sinuosoidais sincronizados

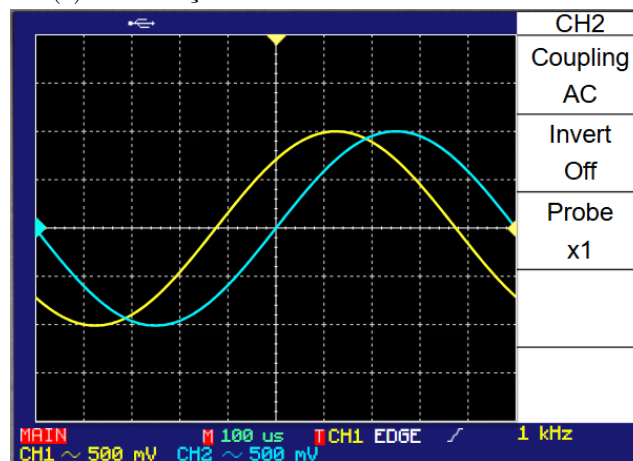
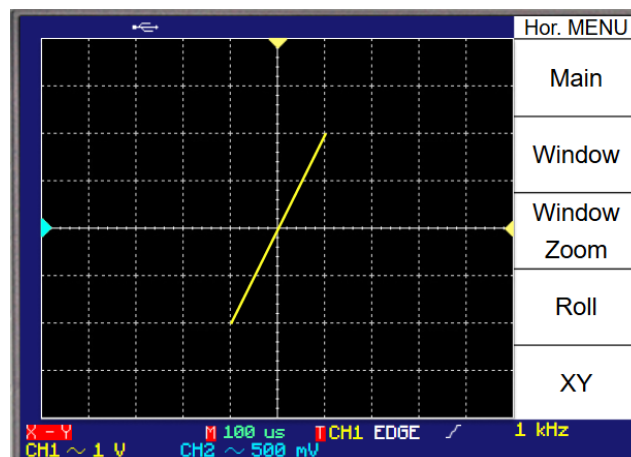
(b) Visualização de sinais sinuosoidais com desfaseamento (45°)

Figura 4.9: Modo "normal"(main) - base de tempo ligada



(a) Visualização de sinais sinusoidais sincronizados

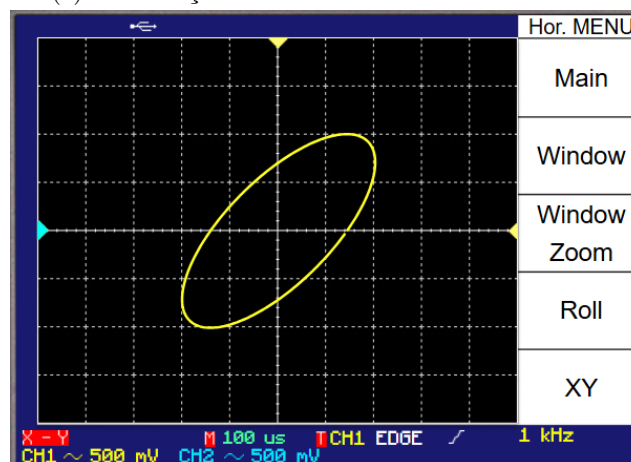
(b) Visualização de sinais sinusoidais com desfasamento (45°)

Figura 4.10: Modo XY - CH1 no x e CH2 no y

4.2.5 Custom Wave - sinal personalizado

A funcionalidade *Custom Wave* foi desenvolvida para ultrapassar as limitações dos geradores de sinais convencionais, permitindo ao utilizador criar formas de onda totalmente personalizadas através da introdução de expressões matemáticas. A interface desta ferramenta foi projetada para ser intuitiva e flexível, apresentando-se dividida em dois modos de operação principais: o *Modo Simples* (Figura 4.11) e o *Modo Avançado* (Figura 4.12).



Figura 4.11: Custom Wave - Janela configuração modo simples

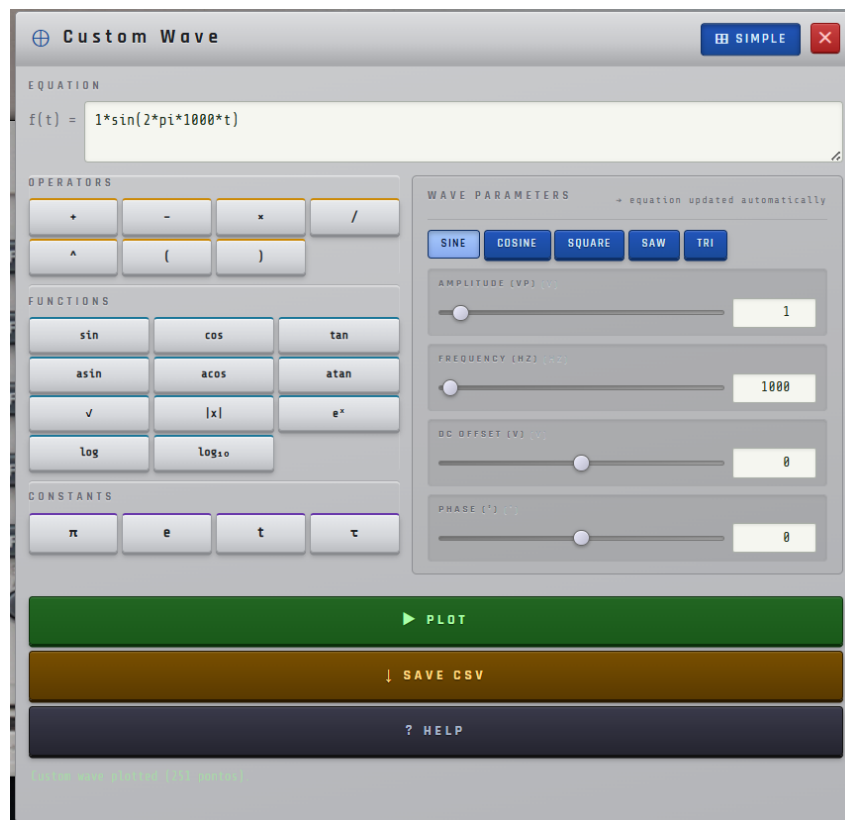


Figura 4.12: Custom Wave - Janela configuração modo avançado

No *Modo Simples*, o utilizador tem à sua disposição uma área de texto para a introdução direta da equação $f(t)$. Esta área é suportada por um painel interativo que inclui operadores aritméticos, funções trigonométricas e logarítmicas, bem como constantes matemáticas fundamentais, como o π e o e . Para auxiliar a formulação destas expressões, a ferramenta integra um Painel de Ajuda, representado na Figura 4.13, que detalha a sintaxe correta das variáveis e fornece exemplos práticos práticos de equações prontas a utilizar. A aplicação direta deste modo é exemplificada na Figura 4.14, onde a inserção de uma função matemática resulta na geração e visualização imediata de um sinal cosseno no ecrã do osciloscópio.



Figura 4.13: Custom Wave - Pannel de ajuda

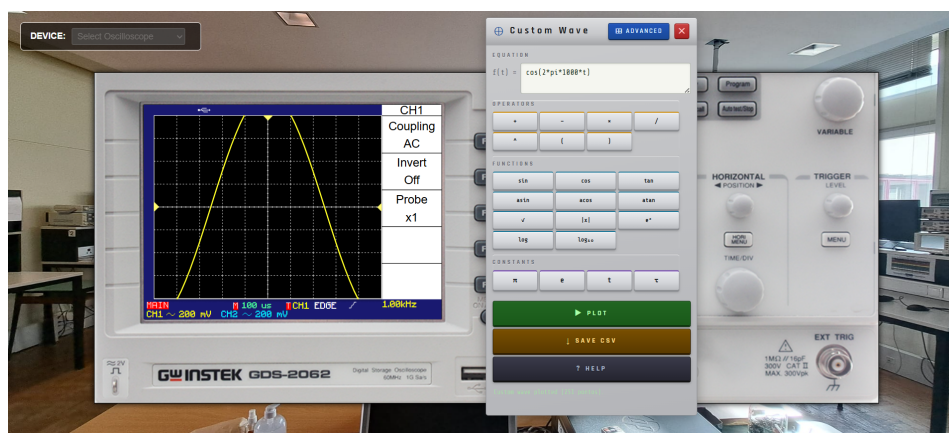


Figura 4.14: Custom Wave - Sinal cosseno - Modo simples

Por sua vez, o *Modo Avançado* expande as capacidades desta funcionalidade ao incorporar um painel lateral dedicado aos parâmetros da onda. Este modo revela-se particularmente útil para ajustar as características do sinal de forma dinâmica, sem que o utilizador necessite de reescrever a equação matemática manualmente. Através de controlos deslizantes, é possível manipular em tempo real a amplitude, a frequência, a componente contínua (DC offset) e a fase do sinal gerado. A Figura 4.15 exemplifica a utilização deste *modo avançado* na síntese de um sinal seno, evidenciando como os ajustes efetuados nos controlos deslizantes são automaticamente traduzidos para a equação e refletidos de imediato na forma de onda apresentada no simulador.

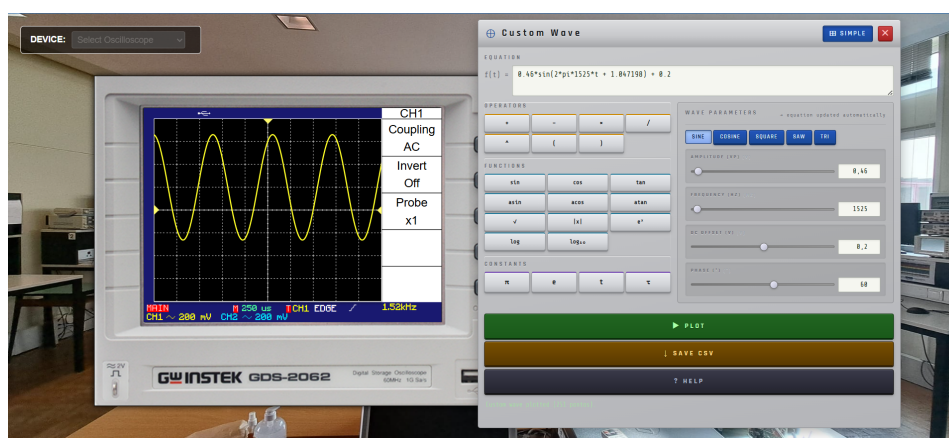


Figura 4.15: Custom Wave - Sinal seno - Modo avançado

Para poder utilizar este modo, o utilizador deve no canal de input do CH1, representado na Figura 4.16, seleccionar a opção "Custom Wave".

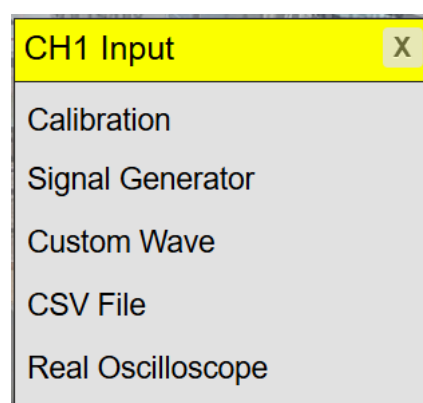


Figura 4.16: Input CH1 - Custom Wave

4.2.6 Página de Support e LabScripts

De forma a melhorar a experiência de navegação e a curva de aprendizagem da aplicação, a interface principal foi expandida com o acesso a duas secções: a página

de *Support* e a nova página de *LabScripts*, ambas acessíveis através da barra de navegação superior, conforme destacado na Figura 4.17.

A página de *Support* foi alvo de uma atualização. Nesta secção é possível encontrar os créditos da criação da aplicação, assim como uma pequena introdução e os objetivos da aplicação, uma tabela de versões e um formulário para partilha de erros.

A principal inovação neste campo foi, contudo, a introdução da página de *LabScripts*. Esta secção foi pensada de forma muito prática para disponibilizar guiões passo a passo de como usar o osciloscópio. Embora o simulador seja muito utilizado no ambiente académico, a ideia desta nova página não é ajudar apenas os alunos. O objetivo principal é oferecer um guia simples e direto para que qualquer pessoa (estudante ou professor) consiga explorar a plataforma. Desta forma, o utilizador consegue perceber rapidamente como tudo funciona e realizar as suas próprias experiências de forma totalmente autónoma.

Para suportar esta autonomia, a interface da página apresenta os guiões em formato de checklist interativa, onde cada instrução pode ser assinalada como concluída. As ações estão descritas com marcadores visuais que correspondem exatamente aos botões do simulador (como ON ou CH1), facilitando a navegação. Adicionalmente, um painel lateral organiza os diferentes módulos da simulação e exibe o progresso global do utilizador, garantindo um acompanhamento claro do seu ritmo de aprendizagem representado na Figura 4.18.

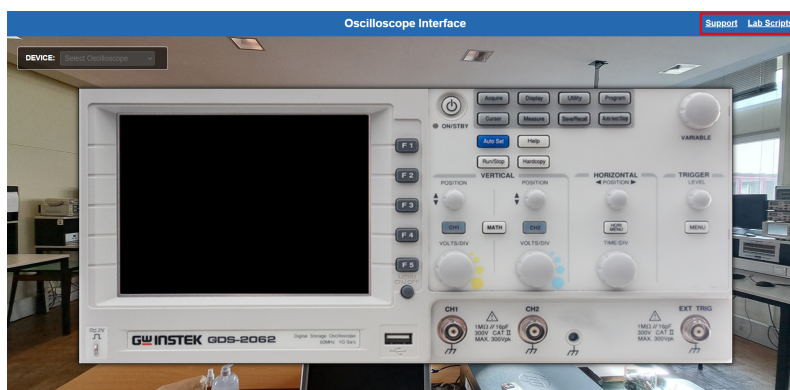


Figura 4.17: Menus de Support e Labscripts

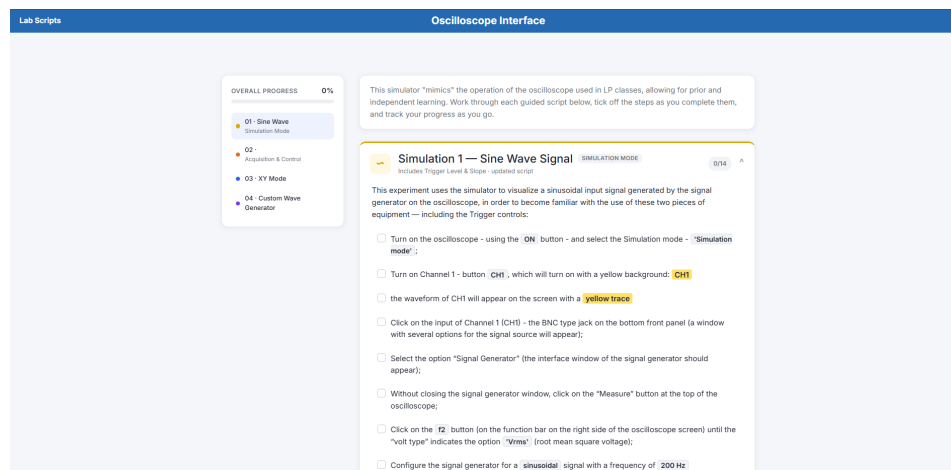


Figura 4.18: Página Labscripts

4.3 Modo de aquisição e controlo remoto

4.3.1 Seletor de Osciloscópio/Equipamentos

O seletor de osciloscópio é o componente da interface responsável por listar e gerir os equipamentos disponíveis para conexão.

Encontra-se visível na interface principal sob a forma de um menu pendente, respresentado na Figura 4.19, sendo o primeiro passo para estabelecer a comunicação com o laboratório. Para iniciar a comunicação, o utilizador deve abrir o menu e escolher o osciloscópio pretendido a partir da lista de equipamentos disponíveis (Figura 4.20). Assim que a seleção é feita, a aplicação fornece um retorno visual imediato para orientar o utilizador sobre o estado do processo. Durante a tentativa de comunicação, surge a indicação "Connecting..."(a cor laranja). Se o processo for concluído com sucesso, o estado altera-se para a indicação "Online"(a verde), confirmando que o equipamento está pronto a ser controlado remotamente e os dados físicos começarão a ser apresentados no ecrã da aplicação. Caso a porta de comunicação do equipamento não seja encontrada no servidor, o indicador passa a assumir o estado "Offline"(a vermelho), impedindo temporariamente novas seleções.



Figura 4.19: Seletor de osciloscópios

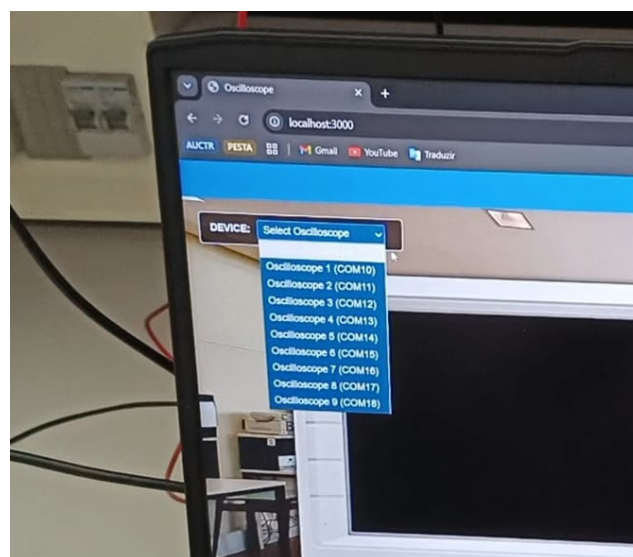
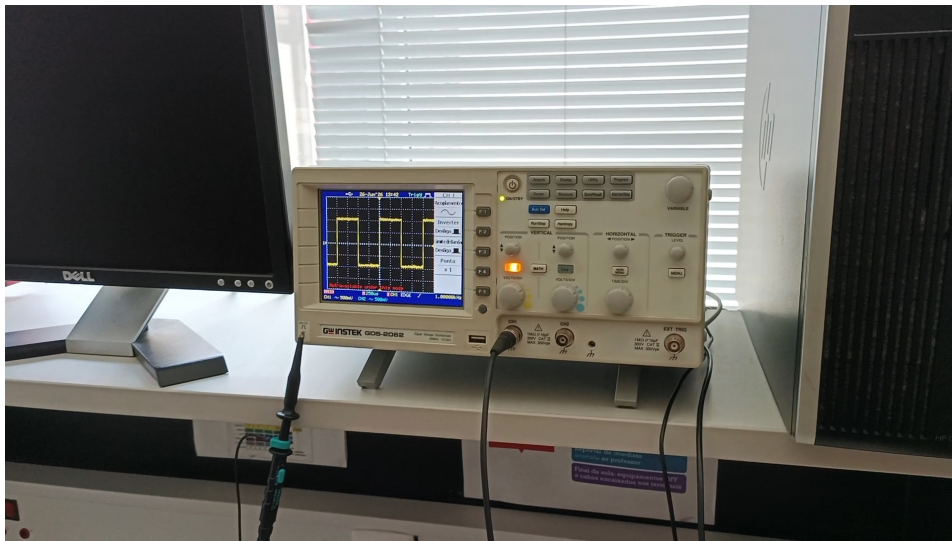


Figura 4.20: Menu de seleção do osciloscópio a controlar remotamente

Uma vez estabelecida a ligação com sucesso (estado "Online"), a aplicação passa a refletir em tempo real os dados capturados pelo osciloscópio. Para ilustrar esta capacidade de replicação, as figuras seguintes demonstram a aquisição do sinal de calibração no Canal 1 (CH1) do equipamento.

A Figura 4.21a exibe o sinal de onda quadrada lido diretamente no ecrã do osciloscópio presente no laboratório. Em simultâneo, a Figura 4.21b comprova a eficácia da comunicação remota, mostrando a reprodução exata e em tempo real desse mesmo sinal no simulador, onde também é possível observar o indicador verde confirmando o estado "Online" do equipamento selecionado.



(a) Visualização do sinal de calibração no osciloscópio



(b) Visualização do sinal de calibração no simulador

Figura 4.21: Validação do modo de aquisição: correspondência entre o sinal físico e o sinal adquirido no simulador

Capítulo 5

Avaliação e Teste de Resultados

5.1 Metodologia de Testes

A fase de testes constituiu uma etapa essencial para garantir o correto funcionamento e a fiabilidade da aplicação perante os utilizadores. Com esse objetivo, foi lançada uma versão de testes do dia 3 de junho, destinada especificamente para avaliação em ambiente real. O processo de teste foi conduzido de forma controlada, tendo o código da aplicação sido substituído temporariamente no computador da bancada 10 do Laboratório de circuitos e sinais (I305) do ISEP. Para assegurar a qualidade e o rigor da avaliação, os testes foram realizados por quatro docentes. O principal objetivo foi confirmar que as melhorias aplicadas no projeto original se integravam perfeitamente e não comprometiam o trabalho previamente consolidado. Durante esta fase, os professores exploraram várias partes da interface, prestando especial atenção a funcionalidades como a *Custom Wave*, o Modo XY e o comportamento do sistema de *Trigger*, para certificar que a apresentação visual dos sinais era exata e fluida. De forma a agilizar o processo, o reporte de erros e a partilha de feedback por parte dos docentes foram efetuados de forma direta através do *WhatsApp*.

5.2 Análise de Resultados

Os testes realizados pelos docentes revelaram-se fundamentais para afinar a aplicação, uma vez que a sua experiência permitiu detetar falhas que tinham escapado

inicialmente. Durante a avaliação da versão de testes do dia 3 de junho, os professores identificaram alguns problemas específicos no modo de *Custom Wave*, falhas no comportamento da função de *trigger* e no funcionamento do modo XY quando as frequências dos sinais eram diferentes. Todos estes erros já foram, entretanto, corrigidos. Com estas correções aplicadas, foi lançada uma nova versão da aplicação, no dia 4 de agosto. Este próximo passo permitirá realizar uma nova fase de testes, de forma a garantir que o sistema está agora totalmente estável e pronto para utilização.

Capítulo 6

Conclusões

6.1 Aprendizagem

A continuidade de um projeto de software previamente desenvolvido introduziu uma camada de complexidade distinta do paradigma habitual do percurso acadêmico, tipicamente centrado em projetos iniciados de raiz. Esta etapa exigiu uma análise cuidadosa para compreender a lógica arquitetural e as decisões de implementação do trabalho realizado anteriormente. Em diversas fases do projeto, o desafio não residiu apenas na criação de novas funcionalidades, mas na capacidade de interpretar o código existente e realizar os ajustes necessários de forma harmoniosa. Este processo de estruturação, focado em garantir a estabilidade da base antes da sua expansão, evidenciou que a prática de engenharia não se restringe à inovação tecnológica, englobando também a competência analítica para compreender, adaptar e melhorar sistemas preexistentes.

Considerando o estado atual da aplicação, os resultados obtidos refletem o sucesso do trabalho desenvolvido. Embora se reconheça que a ferramenta apresenta uma vasta margem para evolução futura, o presente relatório proporcionou não só a consolidação de conhecimentos técnicos e de desenvolvimento web, mas, fundamentalmente, uma preparação robusta para os desafios reais e práticos da profissão.

6.2 Resumo das Contribuições

O trabalho desenvolvido neste projeto centrou-se na otimização e expansão da plataforma *Oscilloscope Interface*, visando a consolidação da sua estabilidade e o enriquecimento com novas funcionalidades. As principais contribuições podem ser sintetizadas nos seguintes tópicos:

- **Reestruturação da Arquitetura e Contentorização:** A infraestrutura de software foi reorganizada de forma a evitar a duplicação de dependências entre o cliente e o servidor, centralizando a configuração do projeto num único ficheiro. Adicionalmente, implementou-se a contentorização da aplicação através do Docker, o que garante a total portabilidade e a integridade do ambiente de execução em qualquer máquina.
- **Aprimoramento e Expansão do Modo de Simulação:** Foram resolvidos comportamentos inconsistentes previamente detetados no sistema de *Trigger*, garantindo a estabilização correta das formas de onda em ambos os canais. Em paralelo, o simulador foi dotado de duas novas ferramentas de análise:
 - A implementação do **Modo XY**, permitindo o cruzamento de tensões entre os canais para o estudo de desfasamentos através de Figuras de Lissajous.
 - O desenvolvimento da ferramenta **Custom Wave**, que introduziu a capacidade de gerar sinais totalmente personalizados a partir de expressões matemáticas, com opções de ajuste dinâmico de parâmetros e exportação de dados analíticos.
- **Otimização do Modo de Aquisição e Controlo Remoto:** Para além da estabilização na transmissão dos sinais físicos, foi concebido e integrado um **seletor de osciloscópios**. Esta funcionalidade permite a gestão e alternância dinâmica entre múltiplos equipamentos ligados em simultâneo ao servidor, apresentando indicadores visuais em tempo real sobre o estado da comunicação com a porta série.
- **Desenvolvimento de Ferramentas de Apoio Pedagógico:** Focando na promoção da aprendizagem autónoma, a interface foi expandida com a nova secção de *LabScripts*, fornecendo práticas guiadas sob a forma de *checklists* interativas[cite: 1]. Complementarmente, foi desenhado um sistema de diagnóstico rápido de problemas, o qual atua como um guia contextual para auxiliar utilizadores a ultrapassar desafios iniciais de configuração.

Na Tabela 6.1 é possível observar uma comparação entre a versão original e a versão atual da plataforma, apresentando uma síntese clara das contribuições desenvolvidas.

Tabela 6.1: Comparação de funcionalidades entre a versão original (V0.1.3) e a versão atual (V1.4.6).

Característica	Versão Original (V0.1.3)	Versão Atual (V1.4.6)
Arquitetura de Software	Separação estrita <i>front-end/backend</i> (duplicação de dependências), sem conten-torização.	Estrutura centralizada num único diretório, ambiente iso-lado com integração Docker.
Sistema de <i>Trigger</i>	Apresentava inconsistências na aplicação do <i>slope</i> e na estabi-lização.	Comportamento estabilizado, respondendo corretamente aos comandos em ambos os canais.
Modo XY	Inexistente.	Implementado (cruzamento de tensões de canais e Figuras de Lissajous).
Geração de Sinais	Limitada a ondas padrão (ge-rador de sinais) ou construção manual ponto-a-ponto via fi-cheiro Excel.	Criação do <i>custom wave</i> .
Controlo Remoto e Aquisição	Ligação estática simples.	Ligação estabilizada e introdu-ção do seletor de osciloscópios para gestão de múltiplos equi-pamentos.
Página <i>LabScripts</i>	Inexistente.	Secção dedicada com <i>checklists</i> interativas focadas no estudo autónomo.
Apoio ao Utilizador	Inexistente.	Introdução do menu interativo de Diagnóstico Rápido de Pro-blemas.

6.3 Trabalho futuro

Como este projeto se focou na melhoria de uma aplicação já existente, ficou claro que este projeto permitiu abrir portas para futuros melhoramentos. Embora se tenha conseguido estabilizar o modo de aquisição e controlo remoto durante este trabalho, o sistema ainda precisa de ser aprimorado. Pensando no desenvolvimento contínuo da ferramenta e no seu potencial pedagógico, destacam-se os seguintes pontos para o trabalho futuro:

- **Aproximação ao Equipamento Real:** Continuar a adicionar as funciona-lidades práticas que ainda faltam na interface web, para que a experiência de utilização, tanto no controlo remoto como na simulação, se assemelhe cada vez mais à operação de um osciloscópio físico no laboratório.

- **Evolução do Guia de Ajuda para Assistente Virtual:** Tendo por base o atual sistema estático de "Diagnóstico Rápido de Problemas", seria de elevado valor pedagógico a sua transformação num *chatbot* interativo. A integração de um assistente virtual permitiria aos estudantes reportar problemas em linguagem natural (por exemplo, "A onda não para de piscar, o que faço?"), oferecendo um suporte muito mais dinâmico, personalizado e responsivo na resolução de bloqueios.
- **Integração com o editor de circuitos QUCS:** Tal como sugerido no desenvolvimento do projeto original, seria muito interessante criar uma ponte entre esta aplicação e o QUCS. Sendo um simulador amplamente utilizado pelos estudantes da LEEC na unidade curricular de *Teoria dos Circuitos* (TCIRC), esta ligação criaria um ecossistema de estudo muito mais rico e integrado.
- **Ensaios Remotos de Filtros:** Realizar todas as adaptações técnicas necessárias na interface e na comunicação para garantir que, quando existirem os equipamentos adequados, seja possível executar um guião de estudo de filtros de forma totalmente remota.
- **Integração com o Circuit Editor do U=RIolve Academy:** Explorar a interoperabilidade com esta plataforma, permitindo que os circuitos desenhados e parametrizados no editor possam ser diretamente analisados na interface do simulador.

Referências

- [1] Tektronix, Inc., *XYZs of Oscilloscopes: Primer*. Tektronix, 2014. [Citado nas páginas vii, 7, 8, 9, 10 e 19]
- [2] W. Hennig, H. Tan, M. Walby, P. Grudberg, A. Fallu-Labruyere, W. K. Warburton, C. Vaman, K. Starosta, and D. Miller, “Clock and trigger synchronization between several chassis of digital data acquisition modules,” *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 261, pp. 1000–1004, 2007. [Citado na página 9]
- [3] S. Bai and P. Yu, “High wideband digital oscilloscope design,” *International Journal of Computer Science and Information Technology*, vol. 3, no. 1, pp. 149–157, 2024. [Citado na página 9]
- [4] P. Salgueiro, “Relatório de projeto/estágio: Simulação de osciloscópio analógico,” relatório de projeto/estágio (pesta), Instituto Superior de Engenharia do Porto (ISEP), Porto, Portugal, 2005. [Citado na página 10]
- [5] J. Pereira, “Relatório de projeto/estágio: Simulação de osciloscópio digital,” relatório de projeto/estágio (pesta), Instituto Superior de Engenharia do Porto (ISEP), Porto, Portugal, 2015. [Citado na página 10]
- [6] J. P. C. Ferreira, “Projeto e desenvolvimento de aplicação web para simulação e interface de osciloscópio,” relatório de projeto/estágio (pesta), Instituto Superior de Engenharia do Porto (ISEP), Porto, Portugal, 2023. [Citado nas páginas 10 e 13]
- [7] M. A. Hodges *et al.*, “The impact of covid-19 on laboratory-based engineering education,” *IEEE Transactions on Education*, vol. 64, no. 4, pp. 386–392, 2021. [Citado na página 11]
- [8] S. Hrastinski, “Asynchronous and synchronous e-learning,” *Educause quarterly*, vol. 31, no. 4, pp. 51–55, 2008. [Citado na página 11]
- [9] D. May, G. R. Alves, A. A. Kist, and S. M. Zvacek, *Online Laboratories in Engineering Education: Research and Practice*. Springer International Publishing, 2022. [Citado nas páginas 11 e 12]

-
- [10] Blekinge Institute of Technology, “Visir - virtual instrument systems in reality,” 2026. Acedido a 11 jun. 2026. [Citado na página 12]
 - [11] National Instruments, *LabVIEW System Design Software*, 2026. Acedido a 11 jun. 2026. [Citado na página 12]
 - [12] National Instruments, *NI Multisim: Circuit Design and Simulation Software*, 2026. Acedido a 11 jun. 2026. [Citado na página 12]
 - [13] OpenJS Foundation, “Express - node.js web application framework.” <https://expressjs.com/>, 2024. [Citado na página 14]
 - [14] R. Sharp, “Nodemon.” <https://nodemon.io/>, 2024. [Citado na página 14]
 - [15] M. Eernisse, “EJS – embedded javascript templating.” <https://ejs.co/>, 2024. [Citado na página 14]
 - [16] OpenJS Foundation, “jQuery.” <https://jquery.com/>, 2024. [Citado na página 14]
 - [17] M. Bostock, V. Ogievetsky, and J. Heer, “D³: Data-driven documents,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, 2011. [Citado na página 14]
 - [18] C. Williams *et al.*, “Node serialport.” <https://serialport.io/>, 2024. [Citado na página 14]
 - [19] ECMA International, “The JSON data interchange syntax,” Standard ECMA-404, ECMA International, 2017. [Citado na página 14]
 - [20] J. de Jong, “Math.js: An extensive math library for javascript and node.js.” <https://mathjs.org/>, 2024. [Citado na página 14]
 - [21] Notion Labs, Inc., “Notion: The all-in-one workspace for your notes, tasks, wikis, and databases.” <https://www.notion.so/>, 2026. [Citado na página 14]
 - [22] GitHub, Inc., “Github: Let’s build from here.” <https://github.com/>, 2026. [Citado na página 14]
 - [23] Microsoft Corporation, “Visual studio code - code editing. redefined.” <https://code.visualstudio.com/>, 2026. [Citado na página 15]
 - [24] GitHub, Inc., “Github copilot: Your ai pair programmer.” <https://github.com/features/copilot>, 2026. [Citado na página 15]
 - [25] Google DeepMind, “Gemini: A family of highly capable multimodal models,” technical report, Google DeepMind, 2023. [Citado na página 16]

Anexo A

Guiões Laboratoriais

A.1 Guião de Simulação - Atualizado com Trigger

1. Ligue o osciloscópio - utilizando o botão **ON** - e selecione o modo de simulação - ‘Simulation mode’;
2. Ligue o Canal 1 - botão **CH1**, que acenderá com um fundo amarelo: **CH1**;
3. A forma de onda do **CH1** aparecerá no ecrã com um traço amarelo;
4. Clique na entrada do Canal 1 (**CH1**) - o conector tipo BNC no painel frontal inferior (surgirá uma janela com várias opções para a fonte de sinal);
5. Selecione a opção “Signal Generator” (a janela de interface do gerador de sinal deverá aparecer);
6. Sem fechar a janela do gerador de sinal, clique no botão “Measure” na parte superior do osciloscópio;
7. Clique no botão **F2** (na barra de funções do lado direito do ecrã do osciloscópio) até que o “volt type” (tipo de tensão) indique a opção ‘Vrms’ (tensão eficaz);
8. Configure o gerador de sinal para um sinal sinusoidal com uma frequência de 200 Hz:
 - O seletor ‘RANGE Hz’ (fator multiplicativo) deve estar na posição ‘1K’;

- Ajuste a frequência até que o indicador localizado no canto inferior direito do ecrã do osciloscópio mostre 200 Hz, da seguinte forma: coloque o cursor do rato sobre o botão ‘FREQUENCY’ (redondo), que permite o ajuste progressivo da frequência (à esquerda);
 - Clique com o botão direito do rato para diminuir o valor da frequência do sinal de entrada; (o botão esquerdo do rato permite aumentar a frequência);
9. Ajuste a tensão eficaz gerada para 3 V (aproximadamente). Para tal:
- Clique com o botão esquerdo/direito do rato no botão ‘AMPLITUDE’ (que permite ajustar a amplitude do sinal de entrada);
 - Guie-se pelo indicador V_{rms} no menu lateral do ecrã do osciloscópio;
10. Configure o comando de amplificação vertical (‘VOLTS/DIV’) do Canal 1 para 2 V/div;
11. Utilizando o comando ‘TIME/DIV’ (base de tempo - sistema horizontal), selecione a velocidade de varrimento de 250 μs /div.
12. Varie o *Trigger Level* do CH1:
- Carregue no botão **MENU** localizado na secção TRIGGER (por baixo do botão rotativo Level);
 - No menu lateral que surge no ecrã, verifique se em ‘Source’ aparece CH1, se não aparecer, carregue no botão **F2** até aparecer CH1;
 - Interaja com o botão rotativo **LEVEL** localizado na secção TRIGGER (à direita no painel);
 - Utilize o clique esquerdo do rato sobre o botão para simular a rotação num sentido, ou o clique direito para rodar no sentido inverso;
13. Altere o *Trigger Slope* para observar o seu impacto no ponto inicial da onda:
- Carregue no botão **MENU** localizado na secção TRIGGER (por baixo do botão rotativo Level);
 - No menu lateral que surge no ecrã, carregue no botão **F5** (correspondente a Slope / Coupling);
 - De seguida, carregue no botão **F1** (Slope) para alternar entre o flanco ‘Rising’ e ‘Falling’;
 - Deixe a configuração no modo ‘Rising’;
14. Utilizando o comando ‘TIME/DIV’ (base de tempo - sistema horizontal), selecione as seguintes velocidades de varrimento e, para cada uma delas, execute a seguinte instrução: 1 ms/div; 500 μs /div e 250 μs /div:

- Selecione o botão **Hardcopy** - surgirá uma janela com várias opções para guardar a imagem/forma de onda;
 - Escolha a opção 'All Screen Image', que guarda uma imagem de todo o ecrã (forma de onda, barra de menus lateral e barra inferior);
15. Após guardar as quatro imagens (cada uma como seu desfasamento), combine-as num único ficheiro de formato PDF (Portable Document Format) e anexe-o na caixa abaixo com o nome "simulacao1".

A.2 Guião de Simulação - Modo XY

1. Ligue o osciloscópio - utilizando o botão **ON** - e selecione o modo de simulação - 'Simulation mode';
2. Ligue o Canal 1 - botão **CH1**, que acenderá com um fundo amarelo: **CH1**;
3. A forma de onda do **CH1** aparecerá no ecrã com um traço amarelo;
4. Clique na entrada do Canal 1 (**CH1**) - o conector tipo BNC no painel frontal inferior (surgirá uma janela com várias opções para a fonte de sinal);
5. Selecione a opção "Custom Wave" (deverá aparecer uma interface de calculadora no ecrã);
 - Carregue no botão "Advanced", no canto superior direito, deverá aumentar a interface da calculadora;
 - Gere uma onda sinusoidal carregando no botão "Sine";
6. Ligue o Canal 2 - botão **CH2**, que acenderá com um fundo azul: **CH2**;
7. A forma de onda do **CH2** aparecerá no ecrã com um traço azul;
8. Clique na entrada do Canal 2 (**CH2**) e selecione a opção "Signal Generator" (a janela de interface do gerador de sinal deverá aparecer);
9. Altere a escala de ambos os canais para 500 mV no botão **VOLTS/DIV** respetivo a cada canal – garanta que elas são a mesma em ambos os canais sempre;
10. Utilizando o comando **TIME/DIV** (base de tempo - sistema horizontal), selecione a velocidade de varrimento de 250 $\mu\text{s}/\text{div}$;
11. Agora carregue no botão **HORI MENU** e de seguida no botão **XY (F5)**, deverá aparecer uma Figura de Lissajous no centro do ecrã (como uma elipse, círculo ou linha reta);

- Selecione o botão **Hardcopy** - surgirá uma janela com várias opções para guardar a imagem/forma de onda;
 - Escolha a opção ‘All Screen Image’, que guarda uma imagem de todo o ecrã (forma de onda, barra de menus lateral e barra inferior);
12. Repita o procedimento anterior para os valores de desfasamento: 45° , 90° e 180° ;
 13. Após guardar as quatro imagens (cada uma como seu desfasamento), combine-as num único ficheiro de formato PDF (Portable Document Format) e anexe-o na caixa abaixo com o nome “simulacaoXY”.

A.3 Guião de Simulação - Custom Wave

A.3.1 Acesso à Ferramenta

1. Ligue o osciloscópio (utilizando o botão **ON**) e selecione o modo de simulação (Simulation mode).
2. Ative o Canal 1 pressionando o botão **CH1**.
3. Clique na entrada analógica do Canal 1 (o conector BNC localizado no painel frontal inferior).
4. No menu **CH1 Input** que surge no ecrã, selecione a opção “Custom Wave”. A interface da calculadora surgirá imediatamente.

A.3.2 Utilização do Modo Simples

No Modo Simples, tem à sua disposição uma área de texto para a introdução direta da equação matemática, representada por $f(t)$. Como usar o modo simples:

- Utilize o painel interativo na interface para construir a equação. Este painel inclui operadores aritméticos (+, -, *, /), funções trigonométricas (sin, cos, tan) e logarítmicas, bem como constantes matemáticas (como o π , e , t e τ).
- Para ajuda na escrita da fórmula, clique no botão “? HELP” na parte inferior. A ferramenta integra um Painel de Ajuda que detalha a sintaxe correta das variáveis e fornece exemplos práticos de equações prontas a utilizar – basta carregar na equação pretendida do menu que ela aparecerá automaticamente na área de texto.
- Construa uma equação de teste. Por exemplo, tente criar um sinal cosseno - $\cos(2 \cdot \pi \cdot 1000 \cdot t)$.

- Após inserir a equação, pressione o botão verde “► PLOT”. A inserção da função matemática resulta na geração e visualização imediata do sinal no ecrã do osciloscópio.
- Ajuste a escala (**VOLTS/DIV**) e a base de tempo (**TIME/DIV**) no osciloscópio para visualizar corretamente a onda gerada.

A.3.3 Utilização do Modo Avançado

O Modo Avançado expande as capacidades da ferramenta ao incorporar um painel lateral dedicado aos parâmetros da onda. Este modo revela-se particularmente útil para ajustar as características do sinal de forma dinâmica, sem que o utilizador necessite de reescrever a equação matemática manualmente.

Passo a passo no Modo Avançado:

- Com a janela do Custom Wave aberta, clique no botão azul “ADVANCED” localizado no canto superior direito. A interface expandir-se-á para mostrar novos controlos.
- Na nova secção *Wave Parameters*, escolha uma das formas de onda base disponíveis (SINE, COSINE, SQUARE, SAW, TRI) para pré-carregar uma equação base.
- Experimente mover os controlos deslizantes. Através destes controlos, é possível manipular em tempo real os seguintes parâmetros:
 - Amplitude
 - Frequência (Frequency)
 - Componente Contínua (DC Offset)
 - Fase (Phase)
- Observe atentamente o ecrã do osciloscópio e a caixa de texto da equação: notará que os ajustes efetuados nos controlos deslizantes são automaticamente traduzidos para a equação e refletidos de imediato na forma de onda apresentada no ecrã do simulador.

A.4 Guião de Modo Aquisição e Controlo

1. Dirija-se à bancada de trabalho e ligue os dois osciloscópios utilizando os respetivos botões (**ON**).
2. Coloque a onda de calibração no Canal 1 (**CH1**) no primeiro osciloscópio e no Canal 2 (**CH2**) no segundo osciloscópio da bancada.

3. Ligue o osciloscópio - utilizando o botão **ON** - e selecione o modo de simulação - ‘Acquisition + Control Mode’;
4. Selecione qual o osciloscópio físico que pretende visualizar e controlar primeiro através do seletor que aparece na tela ao entrar nesse modo - **DEVICE**.

Nota: Caso não saiba qual a porta (Porta COM) de comunicação correspondente ao equipamento, abra o Gestor de Dispositivos (Gerenciador de Dispositivos) do computador para a identificar.

5. Ligue os canais pressionando os botões **CH1** e **CH2** na interface virtual.
6. Altere as escalas remotamente, ajustando os comandos **VOLTS/DIV** $\rightarrow$ 500 mV e **TIME/DIV** $\rightarrow$ 250 μ s/div no simulador para visualizar a onda.
 - Selecione o botão **Hardcopy** - surgirá uma janela com várias opções para guardar a imagem/forma de onda;
 - Escolha a opção ‘All Screen Image’, que guarda uma imagem de todo o ecrã (forma de onda, barra de menus lateral e barra inferior);
7. Repita o mesmo procedimento anterior para o segundo osciloscópio da bancada (selecionar a nova porta, ativar canais e ajustar escalas).
8. Após guardar as duas imagens, combine-as num único ficheiro de formato PDF (Portable Document Format) e anexe-o na caixa abaixo com o nome “acquisicao1”.