

POLITÉCNICO DO PORTO  
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

---

# Projeto e desenvolvimento de aplicação web para simulação e interface de Osciloscópio

João Pedro Cardoso Ferreira

---

LEEC

Licenciatura em Engenharia Eletrotécnica e de Computadores



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA  
Instituto Superior de Engenharia do Porto

Setembro, 2023



*Este relatório foi realizado no âmbito dos requisitos que constam da Ficha de Unidade Curricular de Projeto/Estágio, do 3º ano, da Licenciatura em Engenharia Eletrotécnica e de Computadores.*

**Candidato:** João Pedro Cardoso Ferreira, Nº 1190729, 1190729@isep.ipp.pt

**Orientação Científica:** Mário Alves, mjf@isep.ipp.pt

**Coorientação Científica:** André Rocha, anr@isep.ipp.pt



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

Setembro, 2023



# Agradecimentos

Gostaria de expressar a minha sincera gratidão às pessoas que tornaram possível a realização deste projeto.

Primeiramente, agradeço ao meu orientador, Mário Alves, por me ter proporcionado a oportunidade de desenvolver este projeto, que enriqueceu significativamente o meu *know-how*. Além disso, quero destacar o apoio ao longo do processo, que incluíram orientações, reuniões e assistência na elaboração deste relatório.

Também sou grato ao meu co-orientador, André Rocha, que não apenas me guiou nos primeiros passos da abordagem às tecnologias envolvidas, como também ofereceu apoio em questões técnicas no decorrer do projeto.

Não posso deixar de agradecer ao Engenheiro Vítor Cerqueira pela sua contribuição que permitiu a comunicação do servidor com o exterior, o que foi sem dúvida essencial para o sucesso deste projeto.

Por último, mas não menos importante, desejo expressar a minha profunda gratidão à minha família. O apoio emocional e a orientação que me proporcionaram para lidar com a ansiedade e o *stress* ao longo de todo o desenvolvimento do projeto foram essenciais. Além disso, agradeço também pelas conversas sobre as reflexões e desabafos do trabalho, as quais foram igualmente significativas e contribuíram para o meu crescimento.

Cada um de vocês desempenhou um papel vital na minha formação e na conclusão deste trabalho, e por isso, sou vos profundamente grato.



# Resumo

Este documento tem como objetivo descrever o projeto e implementação de uma aplicação *web*, capaz de manipular em tempo real o osciloscópio *GW-Instek GDS2062* existente no *Laboratório de Circuitos e Sinais Elétricos* do Instituto Superior de Engenharia do Porto (ISEP), a partir de um computador com acesso à *internet*; e é também capaz de simular o comportamento de um osciloscópio digital e de um gerador de sinais, virtualmente.

Para a criação da aplicação foi utilizada a tecnologia *Node.js* com a utilização da linguagem de programação *JavaScript* (JS) e linguagens de desenvolvimento *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS). Foi também necessário aplicar bibliotecas de JS que se orientam ao apoio da criação de aplicações cliente-servidor e conceitos de manipulação de imagens.

Esta aplicação funciona através de uma ligação cliente-servidor, o que permite ao utilizador usufruir da aplicação sem a necessidade de a instalar localmente no seu computador.

Destaca-se ainda esta aplicação, pela sua versatilidade quanto à sua utilização, pois, pode utilizada para estudar sinais a partir do simulador; pode ser utilizada para manipular o osciloscópio à distância; permite partilhar o sinal recebido do osciloscópio por todos os utilizadores que estejam a utilizar a aplicação e desta forma possibilita que cada utilizador possa analisar o sinal conforme mais lhe convenha.

**Palavras-Chave:** Osciloscópio, *GW-Instek GDS2062*, gerador de sinais, aquisição, simulador, ficheiros *Comma-separated values* (CSV), *Node.js*, JS, HTML, CSS, laboratórios remotos, aprendizagem à distância, aplicação *web*, cliente-servidor, *frontend*, *backend*





# Abstract

*This document aims to describe the project and implementation of a web application capable of real-time manipulation of the GW-Instek GDS2062 oscilloscope located in the Laboratory of Electrical Circuits and Signals at the ISEP, from a computer with internet access and it is also capable of simulating the behavior of a digital oscilloscope and a signal generator virtually.*

*To create the application, the Node.js technology was used, along with the JS programming language, and web development languages HTML and CSS. It was also necessary to apply JS libraries geared towards supporting the creation of client-server applications and image manipulation concepts.*

*This application operates through a client-server connection, allowing users to use the application without the need to install it locally on their computer.*

*This application is notable for its versatility in terms of usage because it can be used to study signals from the simulator, manipulate the oscilloscope remotely, share the received signal from the oscilloscope with all users using the application, thereby enabling each user to analyze the signal as they need.*

**Keywords:** Oscilloscope, GW-Instek GDS2062, signal generator, acquisition, simulator, CSV file, Node.js, JavaScript, HTML, CSS, remote labs, e-learning, web application, client-server, frontend, backend.



# Índice

<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>Lista de Acrónimos</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização e identificação do problema . . . . .	1
1.2 Objetivos . . . . .	2
1.3 Metodologia/Abordagem de IDT . . . . .	3
1.4 Contribuições para utilização do osciloscópio . . . . .	3
1.5 Estrutura do Relatório . . . . .	4
<b>2 Contexto científico-tecnológico</b>	<b>5</b>
2.1 Fundamentos da Instrumentação: Osciloscópio/Gerador de Sinais . .	5
2.1.1 O que é e para que serve um osciloscópio? . . . . .	5
2.1.2 O que é um gerador de sinais? . . . . .	6
2.1.3 O que é um osciloscópio virtual? . . . . .	6
2.1.4 Vantagens de usar osciloscópios virtuais ou reais . . . . .	6
2.1.5 Osciloscópio virtual com capacidade de controlar remotamente um osciloscópio real . . . . .	8
2.2 Análise comparativa de simuladores e interfaces de aquisição/controlo remoto . . . . .	8
2.3 Enquadramento pedagógico . . . . .	10
2.4 Tecnologias e plataformas de apoio para o projeto de software . . . .	11
2.4.1 Desenvolvimento da Aplicação <i>Web</i> : Escolhas Tecnológicas e Plataformas Utilizadas . . . . .	11
2.4.2 Linguagens e Bibliotecas . . . . .	12
2.4.3 <i>Trello</i> e <i>Github</i> . . . . .	12
<b>3 Abordagem de Implementação</b>	<b>13</b>
3.1 Plataformas e linguagens de programação utilizadas . . . . .	13
3.1.1 Ambiente de desenvolvimento integrado utilizado . . . . .	13
3.1.2 Ferramentas de IA . . . . .	14

3.1.3	Páginas e aplicações utilizadas . . . . .	15
3.2	Arquitetura de <i>software</i> e modos de utilização da aplicação desenvolvida . . . . .	15
3.2.1	Modo de simulação . . . . .	17
3.2.2	Modo de aquisição e controlo remoto . . . . .	18
3.3	Aspetos de implementação . . . . .	18
3.3.1	Ligação cliente-servidor . . . . .	18
3.3.2	Ligação servidor-osciloscópio . . . . .	18
	Escrever comandos . . . . .	19
	Leitura e análise das formas de onda . . . . .	21
3.3.3	Representação das formas de onda . . . . .	22
<b>4</b>	<b>Guia de utilização da aplicação</b> . . . . .	<b>25</b>
4.1	Página principal . . . . .	25
4.2	Principais funcionalidades da aplicação . . . . .	26
4.2.1	Menus CH1 e CH2 . . . . .	26
4.2.2	Escalas verticais e horizontais . . . . .	27
4.2.3	Posições verticais e horizontais . . . . .	28
4.2.4	<i>Auto set</i> . . . . .	29
4.2.5	Modos de gravação . . . . .	30
4.2.6	<i>Measure</i> . . . . .	31
4.2.7	<i>Run/Stop</i> . . . . .	32
4.2.8	Mensagens de alerta . . . . .	32
4.3	Modo de Simulação . . . . .	33
4.3.1	<i>Inputs</i> . . . . .	33
	Calibração . . . . .	33
	Gerador de Sinais . . . . .	34
	Ficheiro CSV . . . . .	35
	Sinal real proveniente do osciloscópio . . . . .	37
4.4	Modo de aquisição e controlo remoto . . . . .	38
4.4.1	Funcionalidades que não estão implementadas face ao modo simulação . . . . .	38
	<i>Inputs</i> . . . . .	38
4.5	Limitações do osciloscópio . . . . .	39
<b>5</b>	<b>Teste, depuração e melhoramentos</b> . . . . .	<b>41</b>
5.1	Identificação de erros/lacunas . . . . .	41
5.1.1	Fase preliminar de desenvolvimento/testes por parte da equipa envolvida . . . . .	41

<b>6</b>	<b>Considerações Finais</b>	<b>43</b>
6.1	Conclusões sobre o Desenvolvimento do Projeto . . . . .	43
6.1.1	Aprendizagem e evolução . . . . .	43
6.1.2	Melhores práticas para futuros projetos . . . . .	44
	Planificação prévia flexível . . . . .	44
	Definição de regras de comunicação e organização . . . . .	44
6.1.3	Avaliação das conquistas . . . . .	44
6.2	Perspetivas de trabalho futuro . . . . .	45
	<b>Referências</b>	<b>46</b>
	<b>Anexo A Adaptação da preparação do guião de Teoria dos Circuitos</b>	
	<b>(TCIRC)</b>	<b>49</b>
A.1	Novo Exercício 13 . . . . .	49
A.1.1	Simulação 1 . . . . .	49
A.1.2	Simulação 2 . . . . .	51



# Lista de Figuras

1	Osciloscópio do <i>Laboratório de Circuitos e Sinais Elétricos</i> do Instituto Superior de Engenharia do Porto (ISEP) . . . . .	6
2	Gerador de sinais do <i>Laboratório de Circuitos e Sinais Elétricos</i> do ISEP . . . . .	7
3	Estrutura do código da aplicação . . . . .	16
4	Comando que modifica a escala temporal para 5ms . . . . .	19
5	Cabeçalhos de alguns comandos do ficheiro <i>commands.js</i> . . . . .	19
6	Exemplos de comandos utilizados . . . . .	20
7	Comando que pede as configurações atuais do osciloscópio . . . . .	20
8	Resposta ao comando: <i>*LRN?</i> . . . . .	20
9	Comando que pede as informações das formas de onda do canal 1 e 2. . . . .	21
10	Cabeçalho da mensagem. . . . .	21
11	Corpo da mensagem. . . . .	21
12	Primeiros valores do cabeçalho da imagem . . . . .	22
13	Mensagem hexadecimal que indica o tempo entre dois pontos. . . . .	22
14	Função que traduz a mensagem hexadecimal para um valor do tipo float . . . . .	23
15	Informação sobre um ponto . . . . .	23
16	Sinal que está representado por 100 períodos . . . . .	23
17	Interface de Osciloscópio. . . . .	25
18	Escolha do modo de funcionamento . . . . .	26
19	Botões dos menus dos canais 1 e 2 . . . . .	26
20	Canal 1 ligado . . . . .	28
21	Canal 1 ligado com outro menu lateral selecionado . . . . .	28
22	Grandezas da escala vertical . . . . .	28
23	Grandezas da escala horizontal . . . . .	29
24	Cursor da posição vertical do canal 1 . . . . .	29
25	Sinal mostrado no canal 1 não otimizado . . . . .	29
26	Sinal ajustado com o <i>auto set</i> . . . . .	29
27	Sinais do CH1 e CH2 ajustados com <i>auto set</i> . . . . .	30
28	Escolha do modo de gravação . . . . .	30

29	Exemplo da captura do ecrã completo . . . . .	31
30	Exemplo da captura da forma de onda . . . . .	31
31	Exemplo da captura dos valores da forma de onda num ficheiro <i>Comma-separated values</i> (CSV) . . . . .	31
32	Menu <i>Measure</i> . . . . .	32
33	Menu <i>Display All</i> . . . . .	32
34	Botão <i>Run/Stop</i> . . . . .	33
35	Mensagem de alerta que indica a posição vertical do canal 1 . . . . .	33
36	Botões de <i>input</i> . . . . .	34
37	Seleção do sinal de entrada. . . . .	34
38	Calibração do osciloscópio . . . . .	35
39	Interface do gerador de sinais . . . . .	35
40	Forma de onda sinusoidal do gerador de sinais . . . . .	36
41	Forma de onda quadrada do gerador de sinais . . . . .	36
42	Forma de onda triangular do gerador de sinais . . . . .	36
43	Manípulo da frequência . . . . .	37
44	Ficheiro CSV da forma de onda sinusoidal . . . . .	37
45	Forma de onda real e simulada apresentadas simultaneamente . . . . .	38
46	Exercício 13 da preparação do guião 2 . . . . .	50



# Lista de Tabelas

4.1	Funcionalidades da aplicação . . . . .	27
-----	--	----



# Lista de Acrónimos

**AC** *Alternating Current*

**AJAX** *Asynchronous Javascript and XML*

**ASCII** *American Standard Code for Information Interchange*

**CA** *Corrente Alternada*

**CSS** *Cascading Style Sheets*

**CSV** *Comma-separated values*

**D3** *Data-Driven Documents*

**DC** *Direct Current*

**DOM** *Document Object Model*

**EJS** *Embedded JavaScript Templating*

**FEELE** *Fundamentos da Engenharia Eletrotécnica*

**GND** *Ground*

**HTML** *HyperText Markup Language*

**HTTP** *Hypertext Transfer Protocol*

**IDE** *Integrated Development Environment*

**ISEP** *Instituto Superior de Engenharia do Porto*

**JS** *JavaScript*

**JSON** *JavaScript Object Notation*

**LEEC** *Licenciatura em Engenharia Eletrotécnica e de Computadores*

**PDF** *Portable Document Format*

**PESTA** *Projeto / Estágio*

**TCIRC** *Teoria dos Circuitos*

**USB** *Universal Serial Bus*

**VSC** *Visual Studio Code*

# Capítulo 1

## Introdução

O trabalho descrito no presente relatório foi realizado no âmbito da unidade curricular de Projeto / Estágio (PESTA) da Licenciatura em Engenharia Eletrotécnica e de Computadores (LEEC) do ISEP.

Nele, é descrito o desenvolvimento de uma aplicação *web*, concebida para manipular e também servir de simulador do osciloscópio de amostragem presente nas bancadas dos laboratórios do ISEP, bem como, servir de simulador do gerador de sinais, também existente nos laboratórios do ISEP.

### 1.1 Contextualização e identificação do problema

Num cenário de constante evolução tecnológica, ao qual a área da eletrotecnia e da eletrónica não escapam, os instrumentos de medida são uma ferramenta essencial para os engenheiros, os cientistas ou até mesmo os entusiastas por este tema explorarem e compreenderem os domínios dos sinais elétricos.

Ser capaz de visualizar graficamente a forma de onda de um sinal ao longo do tempo, torna possível determinar frequências, amplitudes e outras características cruciais necessárias para identificar anomalias e solucionar problemas, tornando assim, o osciloscópio uma ferramenta muito útil no campo da eletrotecnia e da eletrónica.

A transição, cada vez mais presente, do mundo físico para o mundo digital tem redefinido os padrões e modificado a nossa forma de interagir com o meio. As

ferramentas de teste e os instrumentos de diagnóstico não escapam a esta transição tecnológica.

Fazendo uma comparação entre a instrumentação real e a virtual, podemos concluir que não só cada uma tem as suas vantagens como também respondem a desafios específicos.

Enquanto os osciloscópios físicos oferecem interações táteis e a sensação palpável de manusear um equipamento, os osciloscópios virtuais introduzem flexibilidade, escalabilidade e a capacidade de operar remotamente. Esta mudança de paradigma, redefine a forma como os eletrotécnicos e eletrónicos abordam as suas tarefas diárias ao oferecer soluções eficientes e adaptáveis.

Este projeto visa desenvolver uma plataforma de simulação *web* que não reproduza apenas as funcionalidades de um osciloscópio de amostragem, mas que permita também efetuar a comunicação remota com o osciloscópio.

Em anos anteriores, no âmbito desta mesma unidade curricular, foram já desenvolvidos dois simuladores de osciloscópio.

Ao longo deste relatório procurarei enumerar os motivos subjacentes que levaram à criação deste novo simulador, e partilhar a visão e os objetivos que motivaram a sua criação.

## 1.2 Objetivos

A aplicação *web* do simulador e interface do osciloscópio, tem como propósito servir uma ampla gama de indivíduos, desde estudantes, professores, engenheiros ou entusiastas e técnicos de eletrónica.

O uso da aplicação abrange desde a compreensão fundamental da teoria dos circuitos até à validação de eventuais circuitos elétricos.

Os objetivos a atingir com o desenvolvimento desta aplicação foram criar uma aplicação que permita a receção e manipulação das configurações do osciloscópio remotamente; que permita a análise de formas de onda provenientes do osciloscópio ou de um computador; e que permita a captura de imagens e a gravação das formas de onda estudadas.

Num contexto académico, pretendeu-se que a aplicação desenvolvida fosse assente numa tecnologia *web* que lhe desse uma maior vida útil e com uma estrutura de programação facilmente percetível e de fácil interação que permita por um lado facilitar a evolução da aplicação por futuros alunos e cujo interesse na aplicação se mantenha por um período de tempo suficientemente longo que justifique continuar a desenvolver a aplicação.

### 1.3 Metodologia/Abordagem de IDT

O osciloscópio de amostragem abordado neste relatório é o osciloscópio *GW-Instek GDS2062* presente nas bancadas do *Laboratório de Circuitos e Sinais Elétricos* do ISEP.

Este osciloscópio permite a comunicação com um dispositivo exterior via *Universal Serial Bus* (USB).

Para além da comunicação e transmissão da forma de onda, o osciloscópio também permite a modificação das configurações remotamente, sem necessitar de entrar em contacto físico direto com o osciloscópio.

No desenvolvimento desta aplicação esteve subjacente a utilização da tipologia de comunicação cliente-servidor. Dentro desta linha de pensamento, esta aplicação foi desenvolvida com base em duas áreas de programação, o *frontend* e o *backend*, sendo que, o primeiro conterá o código que irá correr no lado do cliente (computador do utilizador) e o último, conterá o código que irá correr no lado do servidor (computador do ISEP).

Por outro lado, ao nível das funcionalidades da aplicação, foi necessário desenvolver a aplicação tendo sempre presente que a mesma poderia ser utilizada em dois modos de funcionamento distintos: o modo de simulação e o modo de aquisição e controlo remoto. A implementação destas funcionalidades obrigou e distinguir e tratar de forma personalizada cada um destes modos com relação á sua interligação e utilização da tipologia de comunicação cliente-servidor.

De forma resumida, pode-se dizer que no modo de simulação o utilizador intervem essencialmente ao nível do *frontend*, por outro lado, no modo de aquisição e controlo remoto o utilizador terá a possibilidade de manipular as configurações do osciloscópio, e deste modo intervem essencialmente ao nível do *backend*. Em modo de aquisição e controlo remoto, só é permitido o acesso a um utilizador de cada vez.

### 1.4 Contribuições para utilização do osciloscópio

No contexto da aprendizagem da utilização do osciloscópio de bancada, poderá ser útil introduzir a utilização desta aplicação *web*, adaptando para tal o guião "Funcionamento do Osciloscópio e do Gerador de Sinais", já existente, de modo a que este guião tenha em consideração a utilização desta aplicação. Deste modo o aluno poderá ter uma ferramenta que lhe permita aprender o funcionamento do osciloscópio e do gerador de sinais sem estar dependente dos equipamentos físicos.

Esta aplicação permitirá o estudo de um sinal simulado num dos canais e no outro canal o estudo adquirido do osciloscópio de bancada. O que permitirá ao utilizador compreender se os sinais visualizados estão de acordo com o esperado ou não.

Pode também servir para fazer a captura de ecrã da interface do osciloscópio real com o sinal e respetivos valores gerados que são necessários para dar as respostas aos guiões que possam vir a ser solicitados aos alunos.

Num contexto de contigência, como o que foi vivido no ano de 2020, em que não era possível a presença dos alunos no laboratório; esta aplicação pode ser utilizada pelo docente que no laboratório pode partilhar o sinal do osciloscópio com os alunos que estejam a assistir via *web*.

## 1.5 Estrutura do Relatório

No capítulo 2, é feita uma contextualização do osciloscópio de bancada e virtual com o objetivo de introduzir os conceitos base necessários para a compreensão deste relatório; são também referidas as tecnologias utilizadas no desenvolvimento da aplicação.

No capítulo 3, é descrita a abordagem tida ao longo do desenvolvimento da aplicação com vista à definição das linguagens de programação e demais ferramentas a ser utilizadas com vista à construção da aplicação. São também referidas alguns dos pressupostos que serviram de referência para a construção da aplicação.

O capítulo 4 é dedicado principalmente ao utilizador e nele é descrita como funciona a aplicação. No decorrer deste capítulo são apresentadas diversas imagens com o objetivo de facilitar o entendimento das funcionalidades que estão a ser abordadas em cada ponto.

No capítulo 5 serão apresentados os resultados dos testes, validações e correções que foram efetuadas após a finalização da primeira versão desta aplicação.

Por último, no capítulo 6 serão apresentadas as conclusões retiradas do desenvolvimento deste projeto tanto ao nível do desenvolvimento académico/pessoal como ao nível da aplicação desenvolvida e as perspetivas de evolução futura da aplicação.

Toda a documentação que é relevante para este relatório que não está presente no decorrer destes capítulos, será apresentada nos anexos.



## Capítulo 2

# Contexto científico-tecnológico

### 2.1 Fundamentos da Instrumentação: Osciloscópio/Gerador de Sinais

#### 2.1.1 O que é e para que serve um osciloscópio?

Um osciloscópio é um instrumento utilizado para medir sinais elétricos que são representados graficamente por duas dimensões. No modo principal, o eixo vertical representa a amplitude do sinal (tensão), enquanto que o eixo horizontal representa o tempo. Na figura 1 é possível visualizar o osciloscópio que está presente na primeira bancada do *Laboratório de Circuitos e Sinais Elétricos* do ISEP

Este modelo de representação permite analisar diversas características de um sinal, nomeadamente:

- Amplitude (de tensão): valores máximo, mínimo, pico-a-pico, eficaz, diferenciais de amplitude e componentes contínua e alternada.
- Tempo: período, frequência, diferenciais de tempo num sinal e entre dois sinais, atrasos, desfasamento entre dois sinais e tempos de subida.
- Existência de interferências (ruído) continuadas, perturbações transitórias.
- Comparação entre entrada e saída de sistemas, nomeadamente para analisar ganhos, desfasamentos, filtragens, rectificações, permitindo projectar e depurar os mesmos sistemas.



Figura 1: Osciloscópio do *Laboratório de Circuitos e Sinais Elétricos* do ISEP

Se num osciloscópio estiverem incluídos mais canais de entrada é também possível analisar mais do que um sinal em simultâneo.

### 2.1.2 O que é um gerador de sinais?

Um gerador de sinais permite criar sinais elétricos de diferentes formatos, frequências ou amplitudes. As formas de onda geradas pelo gerador de sinais podem ser sinusoidais, triangulares, quadradas, dente-de-serra, entre outras, dependendo do que cada gerador de sinais disponibiliza.

No caso do gerador de sinais que está presente nas bancadas do *Laboratório de Circuitos e Sinais Elétricos* do ISEP, ilustrado na Figura 2, é possível gerar formas de onda, sinusoidais, triangulares e quadradas. Também é possível modificar a frequência do sinal desde os 0,2 Hz até aos 2MHz.

### 2.1.3 O que é um osciloscópio virtual?

Os osciloscópios virtuais reúnem os blocos da cadeia de instrumentação e controlo num ambiente computacional. Esta abordagem oferece uma alternativa aos osciloscópios convencionais que proporcionam resultados comparáveis, com a vantagem adicional de eliminar a necessidade de um equipamento físico dedicado.

### 2.1.4 Vantagens de usar osciloscópios virtuais ou reais

Ambas as tecnologias tem as suas vantagens e desvantagens, nomeadamente em questões como por exemplo:



Figura 2: Gerador de sinais do *Laboratório de Circuitos e Sinais Elétricos* do ISEP

- **Custo:** O osciloscópio virtual tende a ser mais acessível em comparação com o osciloscópio real. O *software* apesar de ser algo complexo e que exige trabalho, pode eventualmente ser código aberto e sem custo, fazendo com que o único custo que o utilizador precise de se preocupar seja com uma placa de som ou outros métodos necessários para fazer a captação do sinal de entrada.
- **Portabilidade e flexibilidade:** No caso dos osciloscópios virtuais o código está presente num computador, que caso este seja portátil, permitirá que o utilizador possa ter bastante flexibilidade sobre onde poder usufruir da aplicação.
- **Análise e armazenamento dos dados:** Apesar de vários osciloscópios reais de amostragem serem capazes de transferir os seus dados via USB para um dispositivo externo, o osciloscópio virtual conseguirá fazer o mesmo sem a necessidade de se conectar a um dispositivo externo pois neste caso o local onde se pretende armazenar os dados ou outras informações relevantes está no mesmo ambiente que o osciloscópio virtual.
- **Resposta temporal:** Com a evolução da tecnologia a diferença da velocidade de cálculo, resposta ou até de amostragem entre osciloscópios virtuais e osciloscópio reais tende cada vez mais para se equipararem, fazendo com que a opção de utilizar um osciloscópio virtual não esteja propriamente atrás.
- **Confiabilidade em ambientes críticos:** Em ambientes industriais ou situações críticas, os osciloscópios reais e de alta qualidade podem ser preferíveis

devido à sua confiabilidade, apesar de poderem vir a ser bem mais dispendiosos.

### 2.1.5 Osciloscópio virtual com capacidade de controlar remotamente um osciloscópio real

Alguns osciloscópios de amostragem incluem *software* que permitem comunicar com um dispositivo externo o que o faz capaz de tanto receber como transmitir informações.

Esta capacidade pode ser explorada permitindo que: seja possível estudar simultaneamente uma onda simulada com uma real no mesmo ambiente; o sinal que esteja no osciloscópio possa ser partilhado em tempo real com outros utilizadores; e também se possa estudar o sinal com outras métricas que não estão disponíveis no osciloscópio real.

## 2.2 Análise comparativa de simuladores e interfaces de aquisição/controlo remoto

No ISEP, dois alunos já desenvolveram projetos sobre simuladores de osciloscópio, nomeadamente o ex-aluno Pedro Salgueiro [1] que desenvolveu um simulador de osciloscópio analógico e o ex-aluno João Pereira [2] que desenvolveu um simulador de osciloscópio de amostragem muito semelhante ao desenvolvido neste projeto, pois utiliza o mesmo modelo de osciloscópio.

Ambos os projetos têm aspetos que são muito importantes e que serviram de referência para o meu projeto.

Começando por abordar o projeto realizado para o simulador de osciloscópio analógico e sendo que o osciloscópio analógico difere muito do osciloscópio de amostragem, podemos ser levados a pensar numa primeira abordagem que os projetos são muito diferentes um do outro, contudo, apesar das diferenças de interface utilizadas em cada um dos projetos e o ambiente de desenvolvimento serem diferentes, os princípios de funcionamento da aplicação são muito semelhantes.

O Pedro Salgueiro desenvolveu funcionalidades como no caso o *trigger* que foi implementado no projeto dele e que é muito semelhante ao do osciloscópio real, bem como o modo X-Y.

Já a aplicação desenvolvida pelo João Pereira, simulador de osciloscópio de amostragem, por ser do mesmo modelo de osciloscópio que foi utilizado neste projeto, permite uma melhor comparação entre as duas aplicações e das funcionalidades que cada uma contém.

O simulador de osciloscópio de amostragem permite o controlo do osciloscópio real através de comunicação USB, sendo que a aplicação é também capaz de receber

e representar a forma de onda proveniente do osciloscópio no computador, o que permite a utilização da aplicação não só como simulador mas também como uma substituição da interface do osciloscópio real.

Nessa aplicação, é também possível alterar valores do osciloscópio e também é possível receber outros valores como por exemplo informações de medida da forma de onda.

Como o projeto desenvolvido no âmbito deste relatório tem como base o mesmo objetivo de simular o osciloscópio de amostragem, em baixo estão apresentadas algumas lacunas que foram identificadas numa fase de testes à aplicação já desenvolvida e que serviram de inspiração para o desenvolvimento da nova aplicação.

Aspetos do projeto do simulador de osciloscópio de amostragem:

- Menu *math*:

O sinal do *math* segue a posição vertical do canal 1.

A escala do menu *math* não é possível ser modificada.

Informações do *measure* sobre o sinal *math* em falta.

Ao sair do menu *math* automaticamente oculta a forma de onda do *math*.

Não permite realizar a operação "ch2-ch1", somente "ch1-ch2".

- Menu horizontal:

Não existe por isso não permite estudar o modo X-Y.

- Menus CH1 e CH2:

Os menus CH1 e CH2 funcionam de forma diferente dos restantes menus do osciloscópio, sendo que para passar por exemplo do modo *measure* para o menu do ch1, deve-se carregar no *measure* para o desligar primeiro.

Ao estar num canal e trocar para outro, desliga automaticamente o sinal, o que não permite modificar o sinal, (não acontece no osciloscópio real).

Operação *invert* inverte os 2 sinais ao invés de inverter só o que está em estudo, o mesmo para a ponta de prova e acoplamento.

Os símbolos dos acoplamentos não coincidem.

- *Auto set*

Ao focar apenas no CH1 coloca a escala vertical do ch2 a 2mv sem motivo, e vice versa

- Ficheiros CSV

Ao guardar a forma de onda CSV não adiciona o cabeçalho, o que impossibilita carregar diretamente este ficheiro que acabou de ser guardado, novamente para a aplicação.

- **Cursors**

Quando mostra a diferença entre os tempos apenas mostra arredondado às unidades, o que faz com que em alguns casos não consiga mostrar o valor com precisão

Não indica de forma nenhuma qual o cursor que se está a usar (No osciloscópio real indica com o tracejado da linha).

Ao sair do menu cursor deveria tirar os cursores pois eles não interessam fora do menu.

- **Measure**

Apenas mostra as informações de um canal de cada vez.

Apenas mostra um valor de amplitude e um valor de tempo, ou seja não é possível visualizar o valor eficaz e o valor pico a pico de uma onda em simultâneo.

- **Gerador de sinais**

Apenas contém uma pequena gama de valores de frequência em comparação com o gerador de sinais de bancada.

Não ajusta a forma de onda em tempo real, é necessária uma confirmação a cada vez que se fizer uma alteração.

Não é possível navegar na aplicação enquanto o gerador de sinais está aberto.

- **Captura de ecrã**

Por vezes a captura de ecrã não captura o que é pretendido.

Alguns dos problemas referidos em cima já estão corrigidos na atual aplicação, no entanto, existem ainda algumas correções a serem implementadas.

## 2.3 Enquadramento pedagógico

A aplicação de um simulador de osciloscópio e gerador de sinais pode ser uma boa ferramenta de apoio ao ensino e aprendizagem na disciplina de Teoria dos Circuitos (TCIRC).

Uma das principais vantagens é possibilitar a aprendizagem autónoma aos estudantes, permitindo com que eles consigam praticar e adquirir conhecimentos em casa, sem depender do acesso ao laboratório no ISEP. O modo de simulação da aplicação oferece aos estudantes a liberdade de experimentar e aprender por conta

própria, sem necessidade de instalar *software* próprio, enquanto que o modo de aquisição e controlo permite aos professores expandirem os seus métodos de ensino seja em regime presencial ou não presencial.

A aplicação pode ser utilizada pelos professores para preparar *slides*, criar problemas para os cadernos de exercícios ou exames, realizar testes do *moodle* e até melhorar as preparações e execuções dos guiões. Neste último caso os estudantes poderiam utilizar o simulador para gravar sinais e formas de onda, que poderiam depois ser inseridos nas preparações ou execuções dos guiões do *moodle*, permitindo assim uma melhor e mais correta avaliação.

Além disso, há também a possibilidade de no futuro utilizar a aplicação em outras experiências e de uso com controlo remoto. A capacidade de adquirir e controlar sinais, bem como realizar análises temporais de circuitos Corrente Alternada (CA), abre portas para explorar novas possibilidades educacionais.

Sendo esta aplicação desenvolvida para poder ser utilizada via *web*, permite que no futuro possa ser adicionada à plataforma U=RI solve, permitindo que os estudantes possam ter todos os instrumentos necessários para aprenderem os conceitos de Fundamentos da Engenharia Eletrotécnica (FEELE) e TCIRC numa só aplicação.

Resumindo, no contexto pedagógico, a utilização de um simulador de osciloscópio e gerador de sinais na cadeira de TCIRC vantagens significativas para o ensino e aprendizagem, desde logo permite a aprendizagem autónoma, a preparação de materiais didáticos, contribuindo para que os estudantes possam melhorar os seus conhecimentos, autonomamente, mais facilmente.

De referir ainda que esta aplicação pode ser facilmente aprimorada com o intuito de não só ajudar os estudantes a aprender sobre este tema como também ajudar investigadores que pretendam estudar formas de onda mais eficientemente ou detalhadamente.

## 2.4 Tecnologias e plataformas de apoio para o projeto de software

### 2.4.1 Desenvolvimento da Aplicação *Web*: Escolhas Tecnológicas e Plataformas Utilizadas

As principais linguagens de programação e plataformas de desenvolvimento utilizadas para desenvolver a aplicação *web* foram *JavaScript* (JS), *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS).

A escolha destas linguagens foi motivada pela ampla adoção e suporte que elas possuem na comunidade de desenvolvedores de aplicações *web* bem como por interesse pessoal.

### 2.4.2 Linguagens e Bibliotecas

O JS é uma linguagem amplamente utilizada para o desenvolvimento de aplicações interativas e dinâmicas, possui também um vasto ecossistema de bibliotecas e *frameworks* tal como o *Node.js* utilizado neste projeto.

O HTML e o CSS foram utilizados para a estruturação e criação dos estilos da interface da aplicação. O HTML permitiu definir a estrutura dos elementos e a organização do conteúdo na página, enquanto que o CSS permitiu personalizar a página.

Além da linguagem de programação e das plataformas de desenvolvimento referidas em cima, foram utilizadas algumas bibliotecas e *frameworks* que facilitaram o desenvolvimento da aplicação. Por exemplo, bibliotecas tais como: *jQuery*, *Data-Driven Documents (D3)*, *Embedded JavaScript Templating (EJS)*, *express*, *nodemon*, *JavaScript Object Notation (JSON)*, *serialport*, e mais que foram utilizados no decorrer do projeto cada uma com a sua contribuição.

### 2.4.3 Trello e Github

O *Trello* e o *Github* são as duas plataformas que foram utilizadas para a interação com os orientadores e a gestão do projeto.

O *Trello* é uma plataforma de gestão de projetos que permite várias equipas possam organizar tarefas e acompanhar o progresso das mesmas. Esta ferramenta é de grande utilidade quando se está a desenvolver um projeto de grande dimensão, pois a melhor forma de se conseguir ter sucesso na resolução de um projeto extenso, é dividindo o projeto em partes mais simples e organiza-las de forma a ser possível nunca se sentir perdido.

O *Github* é uma plataforma de hospedagem que permite organizar o código da aplicação em desenvolvimento. Os colaboradores do projeto podem acompanhar, analisar e trabalhar no código em diferentes partes sem modificá-lo para todos, o que é muito útil no desenvolvimento de uma aplicação.



## Capítulo 3

# Abordagem de Implementação

### 3.1 Plataformas e linguagens de programação utilizadas

#### 3.1.1 Ambiente de desenvolvimento integrado utilizado

Foi escolhido o *Visual Studio Code* (VSC) como *Integrated Development Environment* (IDE) pois contém uma interface amigável, suporta as linguagens aplicadas no trabalho, bem como contém muitas extensões que ajudam a organizar ou até simplificar o código. Para além disso permite também conseguir navegar pelas pastas e ter múltiplos ficheiros abertos em simultâneo o que simplifica o trabalho em projetos complexos.

De forma a conseguir executar o código do JS do lado do servidor e também para conseguir utilizar muitas bibliotecas e dependências importantes, foi utilizada a plataforma de desenvolvimento *Node.js*.

Algumas das dependências que foram utilizadas para complementar o código foram:

- *Express*: Uma *framework* que simplifica a criação de rotas, manipulação de solicitações e respostas *http*.
- *serialport*: Permite com que o *Node.js* se comunique através da porta série, necessário para comunicar com o osciloscópio de bancada.

- D3: Foi utilizado com dois propósitos neste projeto, um deles para a representação da forma de onda graficamente e outro para a leitura dos ficheiros CSV.
- EJS: Apesar de o EJS ter muitas funcionalidades, nomeadamente a possibilidade de incorporar código JS em ficheiros HTML, a utilização do EJS neste projeto teve o propósito de poder simplificar os ficheiros HTML de forma a que estes pudessem ser repartidos por outros ficheiros mais pequenos para facilitar a organização e visualização do código.
- *jQuery*: Simplifica a manipulação do *Document Object Model* (DOM) e interações com *Asynchronous Javascript and XML* (AJAX). Com o *jQuery* é mais simples selecionar e manipular os elementos HTML e é também possível fazer solicitações assíncronas ao servidor, permitindo que o código fique mais simples e direto.
- *request*: Utilizada para realizar solicitações *Hypertext Transfer Protocol* (HTTP) de forma simples. A utilização desta biblioteca permite também enviar informações para o servidor. Esta funcionalidade é necessária para aplicar o método *POST* do protocolo de comunicação HTTP.

### 3.1.2 Ferramentas de IA

Neste projeto foram utilizados o *Github Copilot* como extensão no próprio VSC e o *ChatGPT*.

O *Github Copilot* foi utilizado para simplificar a escrita do código e é muito útil quando se pretende escrever funções que tenham funcionalidades semelhantes contudo variáveis diferentes. Nesta situação, o que o *github copilot* faz é sugerir o código que se deve escrever com base no que já está escrito.

Outras funcionalidades são a criação de código através de comentários, ou a tentativa de continuar o código que se está a começar a escrever, contudo, talvez devido à complexidade deste projeto, a maior parte das vezes esta funcionalidade apenas atrapalhava pois recomendava código que não era nem parecido com o que se pretendia.

O *ChatGPT* foi usado para entender conceitos, explicar código e tirar dúvidas de alguns comandos, apesar de que muitas das vezes algumas das explicações tinham erros e lacunas que caso não tivesse sido crítico o suficiente teria aprendido e aplicado mal alguns conceitos.

Também foi usado o *ChatGPT* para pedir código quando em dúvida de como começar a escrever devido ao desconhecimento da sintaxe da linguagem ou até devido à falta de raciocínio sobre como abordar a situação, e nestes casos é preciso ter

especial cuidado devido ao código dado pelo *ChatGPT* normalmente não representar exatamente o que se quer, apesar de fazer o mesmo efeito.

Apesar das diversas limitações e incertezas que estas tecnologias apresentam, sem dúvida que me ajudaram a estudar os conceitos que precisava de dominar para poder criar uma aplicação mais completa.

Estas aplicações ainda não contém capacidades suficiente para substituir um programador contudo quando utilizadas com sentido crítico e monitorizadas por quem saiba o que está a fazer, podem ajudar muito no desenvolvimento de um projeto.

### 3.1.3 Páginas e aplicações utilizadas

No desenvolvimento deste projeto foi necessário lidar com alguns aspetos que envolviam questões fora da programação, nomeadamente a edição e a manipulação de imagem e também a organização.

De forma a conseguir simular os botões do osciloscópio e do gerador de sinais foi preciso utilizar um conceito chamado *ImageMap* [3] que resumidamente cria uma área através das coordenadas da imagem que depois pode ser manipulada com JS.

Utilizei também o *Adobe Photoshop* para criar uma interface gráfica semelhante à do osciloscópio e também para manipular a imagem do osciloscópio por exemplo para criar a iluminação dos botões dos menus CH1 e CH2 e também para editar as restantes imagens que são utilizadas pela aplicação.

Uma página *web* que me ajudou a entender muitas questões de sintaxe de HTML, CSS e JS foi o *w3schools* [4], como não tinha conhecimentos prévios destas linguagens, este site, foi bastante consultado.

No final do projeto comecei também a utilizar uma plataforma chamada *Notion* para poder organizar o que tinha de fazer bem como organizar alguns documentos importantes para o estudo ou até informações de sintaxe. Este tipo de aplicações se forem bem utilizadas desde o início do desenvolvimento de um projeto podem ser uma mais valia para a organização do projeto.

## 3.2 Arquitetura de *software* e modos de utilização da aplicação desenvolvida

Como esta aplicação permite efetuar ligações cliente-servidor decidi separar a construção e desenvolvimento do código em duas áreas, *backend* e *frontend*.

O *backend* contém o código que irá correr no servidor, aqui contém, por exemplo, as receções das solicitações dos pedidos HTTP. Tem também o código que escreve os comandos que irão ser enviados para o osciloscópio bem como o código para receber e analisar as respostas do servidor.

O *frontend* contém o código que irá correr no cliente e constitui a grande parte do código da aplicação. Dentro do *frontend* existem duas pastas importantes a pasta *public* e a pasta *views*. A pasta *views* contém todos os ficheiros HTML, na pasta *public*, está tudo o resto, nomeadamente, as imagens usadas na aplicação, o código JS que corre no lado do cliente, os estilos usados e também as formas de onda do gerador de sinais e outras no formato CSV.

A estrutura do código representada em árvore pode ser visualizada na imagem: Figura 3.

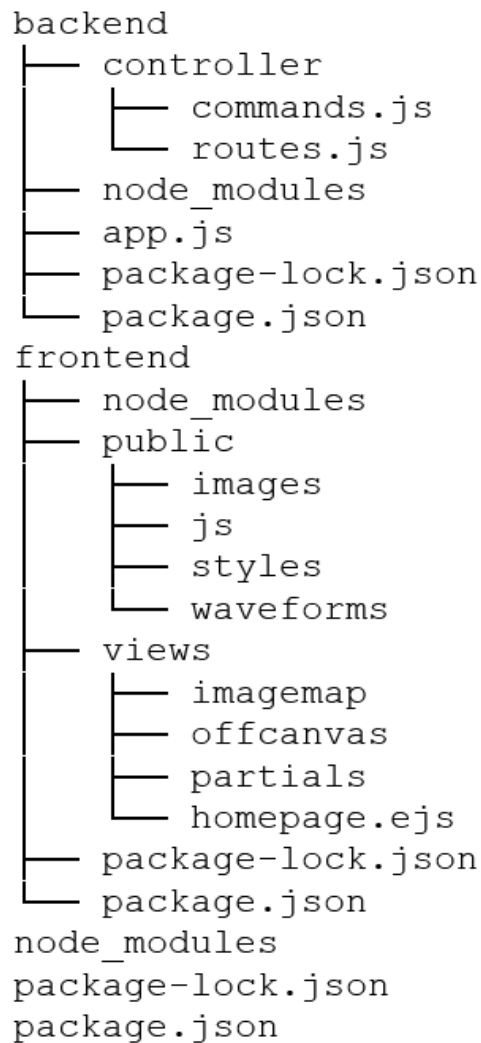


Figura 3: Estrutura do código da aplicação

O ficheiro *app.js* que é usado para iniciar o servidor, contém os caminhos para as pastas *public* e *views* bem como a inicialização do servidor na porta 3000.

O ficheiro *routes.js* contém o código que irá correr na parte do servidor e está dividido em duas grandes partes. Uma delas contém as informações que recebe da porta série que provém do osciloscópio e também as solicitações do cliente, que

apenas são utilizadas quando se pretende comunicar com o osciloscópio. Deste modo, o servidor é o intermediário que recebe os pedidos do utilizador, gera os comandos e envia-os para o osciloscópio, depois, é também o responsável por receber de volta a resposta do osciloscópio, analisá-la e enviá-la para o utilizador.

No *commands.js* existe uma lista de todos os comandos relevantes para a aplicação que são aceites pelo osciloscópio.

Passando para o *frontend*, e começando pelo código JS que corre no cliente, este está dividido em 5 ficheiros com o objetivo de ajudar o programador a organizar-se. O primeiro que tem o nome *events.js* e contém o código que espera pelas interações do utilizador com a página. Neste caso, estas interações podem ser cliques com o botão direito ou esquerdo, ou ainda com o *scroll* do rato para cima ou para baixo.

Quase todos estes eventos estão associados a uma função que irá realizar alguma ação, sendo que estas funções estão divididas em 3 ficheiros um que representa as funções utilizadas globalmente, outro no modo em que se controla o osciloscópio e o terceiro no modo em que está como simulador.

Os ficheiros do formato CSS (estilos) da aplicação, estão divididos por categorias e também estão a usar variáveis para permitir alterações mais rápidas e simples.

As formas de onda que foram criadas como *template* estão no formato CSV, a primeira coluna representa o tempo em segundos e a segunda a respetiva amplitude em volts. Nos ficheiros que foram criados como *template*, existem formas de onda sinusoidais, quadradas, triangulares e dentes de serra.

Passando agora para a pasta *views* onde se encontra o código EJS da aplicação, este contém a página principal, que é a página apresentada aos clientes, e nesta página existem diversos *includes*, estes *includes* fazem parte da biblioteca *ejs* que permite adicionar outros ficheiro *ejs* dentro deste. Esta possibilidade permite criar uma maior organização no código e também possibilitar no caso de uma aplicação com múltiplas páginas de conseguir aproveitar o mesmo código para diferentes páginas.

### 3.2.1 Modo de simulação

O modo de simulação tem como objetivo o utilizador da aplicação conseguir estudar virtualmente o funcionamento do osciloscópio de bancada. Neste modo é possível utilizar formas de onda simuladas pelo gerador de sinais virtual, formas de onda de ficheiros CSV externos, ou até estudar a forma de onda proveniente do osciloscópio real, esta última utilização tem como nome "modo de aquisição".

Neste "modo de aquisição" é possível o cliente utilizar todas as funcionalidades normais existentes no modo de simulação, com a particularidade que a forma de onda é o sinal proveniente do osciloscópio real. A receção deste sinal pode ser interrompida através do clique no botão *run/stop* que também contém um nome chamado "modo *Hold*". Estes dois submodos, dentro do modo simulação, fazem com que seja possível

estudar em simultâneo uma forma de onda simulada num canal com uma forma de onda real noutra canal.

### 3.2.2 Modo de aquisição e controlo remoto

Neste modo o utilizador controla remotamente o osciloscópio, ou seja caso seja modificada a escala vertical no *browser* da aplicação, também será modificada no osciloscópio real, o mesmo acontece para os restantes comandos que estejam disponíveis no osciloscópio. É também importante realçar que no modo de aquisição e controlo remoto, o osciloscópio está em completo sincronismo com a aplicação sendo que se for realizada uma alteração no osciloscópio de bancada, a alteração pode ser visualizada, pelo utilizador, no *browser* da aplicação.

## 3.3 Aspetos de implementação

Para desenvolver esta aplicação, foi necessário adquirir e aplicar conhecimentos sobre: ligações cliente-servidor, comandos e comunicações entre o osciloscópio *GW-Instek GDS2062* e o computador via USB e as funcionalidades do osciloscópio real que foram replicadas no modo de simulação.

### 3.3.1 Ligação cliente-servidor

De forma a que vários utilizadores consigam comunicar com o osciloscópio e também de forma a ser possível utilizar esta aplicação sem a necessidade de a instalar no computador do utilizador, foi preciso primeiro pensar em como desenvolver uma aplicação capaz de comunicar entre cliente e servidor.

Por este motivo, esta aplicação está dividida em *backend* e *frontend*. O cliente e o servidor comunicam-se tendo por base os métodos HTTP, principalmente os métodos *POST* e *GET* que são os mais relevantes nesta aplicação. O método *POST* apenas é utilizado no modo de aquisição e controlo-remoto devido à necessidade de modificar os valores do osciloscópio real. Já o método *GET*, é usado em ambos os modos pois em ambos os modos é possível pedir informações sobre os valores e a forma de onda que estão no osciloscópio.

### 3.3.2 Ligação servidor-osciloscópio

O osciloscópio *GW-Instek GDS2062* dispõe de uma porta para comunicar com um dispositivo externo por via USB. O manual *GDS-2000 Series Programming Manual* [5], distribuído pela *gwinstek*, contém informações sobre como enviar comandos e receber valores.

### Escrever comandos

Para poder receber ou modificar algo do osciloscópio, é sempre necessário enviar o comando pretendido, sendo que cada um funciona da sua forma. A sintaxe dos comandos é dividida por três partes, o cabeçalho, o parâmetro e o delimitador, tal como mostra a Figura 4, que tem como exemplo o comando que modifica a escala temporal para 5ms. O cabeçalho é o que define o que se pretende alterar, sendo que

:TIM:SCAL 5.000E-3 \n  
└──────────┘      └──────────┘ └──┘  
Cabeçalho      Parâmetro Delimitador

Figura 4: Comando que modifica a escala temporal para 5ms

no ficheiro *commands.js* da aplicação, está contida uma lista de todos os cabeçalhos que são utilizados na aplicação, na Figura 5 está apresentada parte da organização dos comandos do ficheiro *commands.js*.

```

110 this.commands.timCommands=[]; // List of Time Related commands
111 this.commands.timCommands.push(':TIM:DEL'); // Horizontal delay time [0]
112 this.commands.timCommands.push(':TIM:SCAL'); // Horizontal scale [1]
113 this.commands.timCommands.push(':TIM:SWE'); // Horizontal sweep mode [2]
114 this.commands.timCommands.push(':TIM:WIND:DEL'); // Delay time of the window zoomed timebase [3]
115 this.commands.timCommands.push(':TIM:WIND:SCAL'); // Scale of the window zoomed timebase [4]
116
117 this.commands.trigCommands=[]; // List of Trigger Related commands
118 this.commands.trigCommands.push(':TRIG:COUP'); // Coupling [0]
119 this.commands.trigCommands.push(':TRIG:DEL:TIM'); // Delay time [1]
120 this.commands.trigCommands.push(':TRIG:DEL:EVEN'); // Delay trigger events [2]
121 this.commands.trigCommands.push(':TRIG:DEL:LEV'); // Delay trigger level [3]
122 this.commands.trigCommands.push(':TRIG:DEL:MOD'); // Delay trigger mode [4]
123 this.commands.trigCommands.push(':TRIG:DEL:TYP'); // Delay trigger type [5]
124 this.commands.trigCommands.push(':TRIG:LEV'); // Trigger level [6]
125 this.commands.trigCommands.push(':TRIG:MOD'); // Trigger mode [7]
126 this.commands.trigCommands.push(':TRIG:NREJ'); // Noise rejection [8]
127 this.commands.trigCommands.push(':TRIG:PULS:MOD'); // Pulse trigger mode [9]
128 this.commands.trigCommands.push(':TRIG:PULS:TIM'); // Pulse trigger time [10]
129 this.commands.trigCommands.push(':TRIG:REJ'); // Trigger rejection [11]
130 this.commands.trigCommands.push(':TRIG:SLOP'); // Trigger slope [12]
131 this.commands.trigCommands.push(':TRIG:SOUR'); // Trigger source [13]
132 this.commands.trigCommands.push(':TRIG:TYP'); // Trigger type [14]

```

Figura 5: Cabeçalhos de alguns comandos do ficheiro *commands.js*

O que define se o comando é um pedido para receber valores ou um pedido para modificar valores, é o parâmetro. Nos comandos em que é possível modificar valores deve-se substituir o parâmetro pelo valor ou flag em questão, tal como mostram os exemplos na Figura 6.

O comando da esquerda que já foi mostrado, aceita como parâmetro os valores usados na escala temporal, já o comando da direita aceita como parâmetro o valor 0, 1 e 2, sendo que estes valores representam os diferentes acoplamentos; 0 o acoplamento *Alternating Current* (AC); 1 o acoplamento *Direct Current* (DC) e 2 o acoplamento *Ground* (GND).

Todos os significados de cabeçalhos e parâmetros possíveis encontram-se no manual de programação oficial.

```
:TIM:SCAL 5.000E-3\n           :CHAN1:COUP 2\n
                    5ms           CH1  Acoplamento ground
```

Figura 6: Exemplos de comandos utilizados

Para pedir valores deve-se utilizar o sinal "?" no lugar do parâmetro. Existem alguns valores que apenas podem ser pedidos e não modificados. Os comandos cuja utilização apenas sirvam para pedir valores, necessitam na mesma do parâmetro "?".

Para além dos comandos que modificam ou pedem um valor específico, existe um comando que tem como função pedir informação sobre diversos valores.

O comando ilustrado na Figura 7 quando enviado para o osciloscópio, retorna a informação mostrada na Figura 8.

```
*LRN?
```

Figura 7: Comando que pede as configurações atuais do osciloscópio

```

:SYSTem:TIME 17 59 57;DATE 18 10 2006;:TRIG
ger:TYpe 0;COUple 1;LEVel 940mV ;MODE 1;NREJ 0;REJect 0;SLOP 0;SOURce 0;ADVance:
DELay 20.0ns;EVENT 3;LEVel 1.48V ;MODE 0;TYpe 0;PULSe:TIME 0.000E-9 ;MODE: 0;TV:
FIEld 1;LINE 1;POLarity 1;TYpe 1;:ACQuire:AVERage 1;LENGth 500;MODE 0;POINT;:DIS
Play:WAVEform 0;DISPCONTrast -10;GRATICule 0;:CURSor:SOURce 1;X1Position
;X2Position ;Y1Position ;Y2Position ;XDELTA
;YDELTA ;XDISPlay 2;YDISPlay 2;:CHANnel1:BWLimit 0;COUpling 0;DISPlay
1;INVert 0;MATH 0;OFFSet 4.700e-01;PROBe 0;SCALe 5.000e-01;:CHANnel2:BWLimit 0;
COUpling 0;DISPlay 0;INVert 0;MATH 0;OFFSet 0.000e+00;PROBe 0;SCALe 5.000e-01;:M
EASure:SOURce 1;FALL ?;FREQuency ?;NWIth ?;PDUTy ?;PERiod ?;PWIDth ?;RISe 7.918
us;VAMPLitude ?;VAVERage 941mV ;VHI ?;VLO ?;VMAX 1.94V ;VMIN -60.0mV ;VPP ?;V
RMS 1.35V ;ROVShoot ?;FOVShoot ?;RPReshoot ?;FPReshoot ?;:MEASure:SOURce 2;FALL
?;FREQuency ?;NWIth ?;PDUTy ?;PERiod ?;PWIDth ?;RISe chan off;VAMPLitude ?;VAV
erage chan off;VHI ?;VLO ?;VMAX chan off;VMIN chan off;VPP ?;VRMS chan off;ROVSh
oot ?;FOVShoot ?;RPReshoot ?;FPReshoot ?;DELAY1 1;DELAY2 2;FRRDelay ?;FRFDelay ?
;FFRDelay ?;FFFDelay ?;LRRDelay ?;LRFDelay ?;LFRDelay ?;LFFDelay ?;:TIMEbase:DEL
ay 0.000e+00;SCALe 2.500e-04;SWEp 0;:AUToset;:PRINt;:REFresh;:RUN;:STOP

```

Figura 8: Resposta ao comando: \*LRN?

Devido à grande variedade de valores que este comando retorna, optou-se por utilizar somente este comando para poder receber as informações de todos os valores do osciloscópio.

Juntamente com este comando, é também necessário outro comando cuja funcionalidade também é só de resposta de valores que são os comandos ilustrados na



Figura 9 que enviam as informações sobre as formas de onda do canal 1 e canal 2 respetivamente.

```

:ACQ1:MEM?
CH1

:ACQ2:MEM?
CH2

```

Figura 9: Comando que pede as informações das formas de onda do canal 1 e 2.

### Leitura e análise das formas de onda

Tal como já foi mencionado, o primeiro comando da Figura 9, retorna a informação da forma de onda do canal 1, sendo que esta informação é dividida em duas partes, o cabeçalho, que contém as informações que estão na Figura 10 e os dados que são mostrados como se mostra na Figura 11.

```
<Buffer 23 34 31 30 30 38 37 a7 c5 ac 01 00 03 e8>
```

Figura 10: Cabeçalho da mensagem.

```
<Buffer 00 00 00 63 00 62 00 62 00 63 00 60 00 65 00 64 00 62 00 61 00 63 00 62 00 62 00 64 00 62 00 61 00 62 00 63 00 66 00 62 00 63 00 63 00 62 00 62 ... 950 mo
re bytes>
```

Figura 11: Corpo da mensagem.

O cabeçalho começa com o caractere # ou 23 em hexadecimal.

Depois, são ilustrados o número de *bytes* em que o tamanho dos dados estão representados, podendo esta indicação ser "4" ou "5", seguido do valor que indica o tamanho dos dados. Todos estes valores estão em hexadecimal e quando traduzidos com a tabela *American Standard Code for Information Interchange* (ASCII), ilustram os caracteres. Na Figura 12 encontram-se estes primeiros três valores e os seus respetivos caracteres segundo a tabela ASCII.

De seguida são representados 4 *bytes* que quando são convertidos do formato hexadecimal para ponto flutuante, indicam o tempo entre dois pontos que estão seguidos. Na Figura 13 verifica-se a mensagem hexadecimal e na Figura 14 a função que foi criada com o objetivo de traduzir a mensagem para ponto flutuante.

Por fim é mostrado um *byte* que indica em hexadecimal o valor do canal do qual está a ser enviada a informação, sendo que "01" indica que é do canal 1 e "02" do canal 2, juntamente com outros três *bytes* que são reservados e não são utilizados nesta aplicação.

Após o cabeçalho, é então lido o corpo da mensagem, que contém o número de *bytes* que foi indicado no cabeçalho da mensagem.

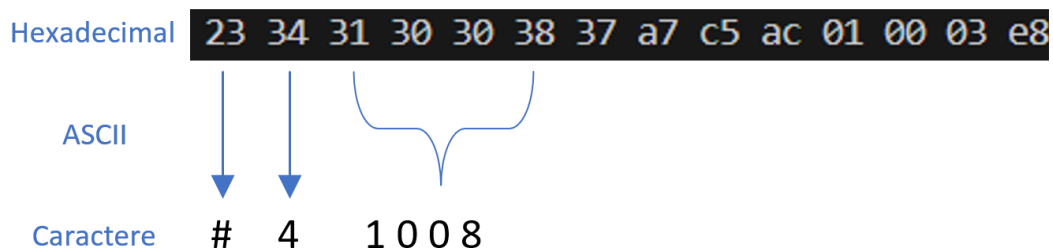


Figura 12: Primeiros valores do cabeçalho da imagem

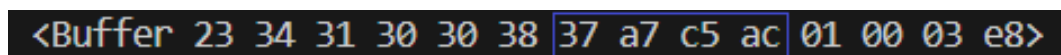


Figura 13: Mensagem hexadecimal que indica o tempo entre dois pontos.

A informação sobre cada ponto está dividido por dois *bytes*, tendo como exemplo a Figura 15, o primeiro *byte* indica sempre a direção do ponto, sendo que este ou está localizado no eixo positivo das ordenadas representado por "FF" ou no eixo negativo das ordenadas representado por "00", sendo que neste caso o ponto encontra-se na parte superior do eixo.

O segundo *byte* representa numa escala de 0 a 127, o quão afastado este ponto está do eixo do x, sendo que o valor 0 indica que o ponto estaria localizado sobre o eixo do x, e o valor 127 indica que o ponto está localizado 5 divisões verticais acima ou abaixo, sendo que os restantes 126 valores encontram-se igualmente espaçados no decorrer do eixo das ordenadas.

### 3.3.3 Representação das formas de onda

A forma de onda é representada com o uso da biblioteca D3, para tal foi criada uma função que cria as escalas verticais e horizontais dependendo da escala vertical ou horizontal que está selecionada. Escreve os pontos da forma de onda independentemente do número de pontos que tenha, o que dá a liberdade para o ficheiro a ser lido ter mais ou menos informação. Quando a escala horizontal é muito superior ao necessário para representar um período do sinal, este período é repetido até num máximo de 100 vezes, sendo que não se encontra necessidade de se representar mais vezes pois o sinal não será visível.

Na figura 16, encontra-se representado um gráfico cujo sinal está representado por 100 períodos.

Como é possível observar na imagem, caso se opta-se por disponibilizar mais períodos, a conclusão que se tiraria do sinal seria a mesma, portanto, a fim de poupar recursos computacionais, são representados, no máximo, 100 períodos.

```
function hexToValue(hexString) {  
  let buffer = new ArrayBuffer(4);  
  let dataView = new DataView(buffer);  
  
  dataView.setUint8(0, hexString[3]);  
  dataView.setUint8(1, hexString[2]);  
  dataView.setUint8(2, hexString[1]);  
  dataView.setUint8(3, hexString[0]);  
  
  let floatValue = dataView.getFloat32(0, true); // The second parameter (true) is for little endian  
  return floatValue.toExponential(3);  
}
```

Figura 14: Função que traduz a mensagem hexadecimal para um valor do tipo float

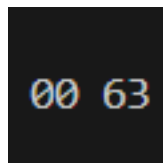


Figura 15: Informação sobre um ponto

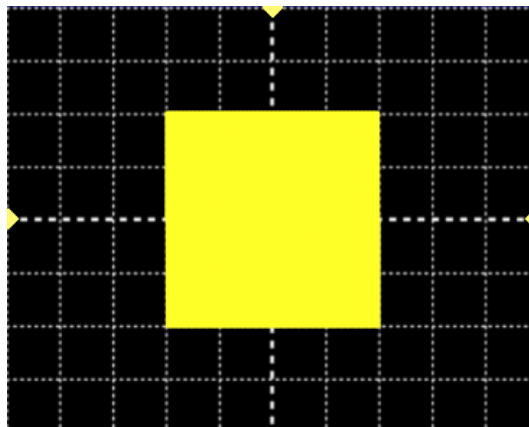


Figura 16: Sinal que está representado por 100 períodos



## Capítulo 4

# Guia de utilização da aplicação

### 4.1 Página principal

Para aceder à aplicação é necessário introduzir o link "[osciloscopio.dee.isep.ipp.pt](http://osciloscopio.dee.isep.ipp.pt)" num browser. Depois de aceder à página, será apresentada a página principal da aplicação mostrada na Figura 17.

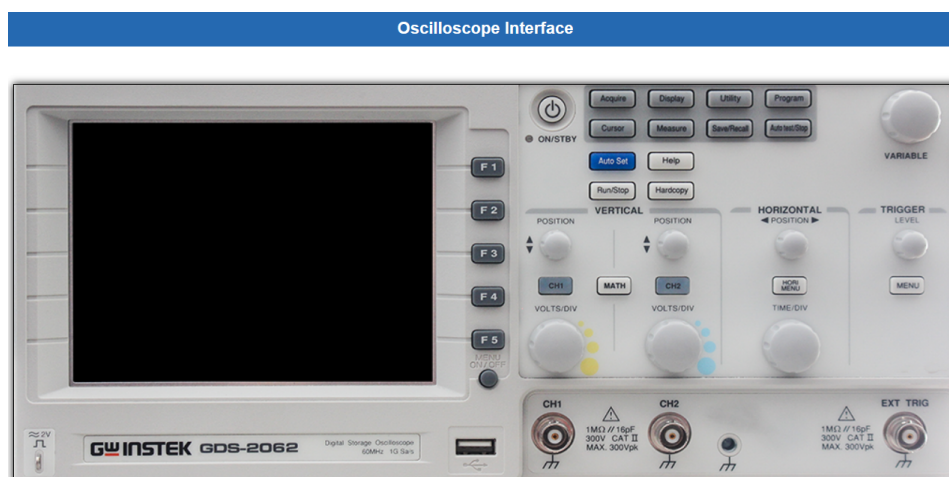


Figura 17: Interface de Osciloscópio.

Ao pressionar o botão que liga o osciloscópio, aparece um menu que permite escolher o modo de utilização, tal como mostra na Figura 18.

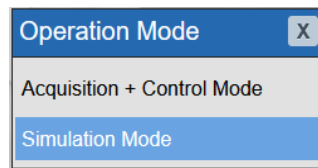


Figura 18: Escolha do modo de funcionamento

Nos próximos capítulos vão se abordar as funcionalidades e aspetos sobre cada modo de funcionamento.

## 4.2 Principais funcionalidades da aplicação

Na Tabela 4.1 está apresentada uma síntese de todas as funcionalidades da aplicação, juntamente com o estado da tecnologia no momento da submissão (segunda coluna) e atual (terceira coluna).

As funcionalidades que estão a verde significam que já foram implementadas e estão funcionais, a amarelo indicam que estão incompletas e que ainda estão em desenvolvimento e a vermelho estão as funcionalidades que de momento não estão implementadas.

Abaixo serão enunciadas e explicadas as principais funcionalidades que são similares aos dois modos de funcionamento da aplicação.

### 4.2.1 Menus CH1 e CH2

O botão do CH1 e CH2 destacados na Figura 19, quando clicados, acionam as respetivas funcionalidades do CH1 e CH2. No caso de o canal estar previamente desligado, com o clique esquerdo do rato o canal é ligado e é apresentado o menu lateral com as informações do canal.

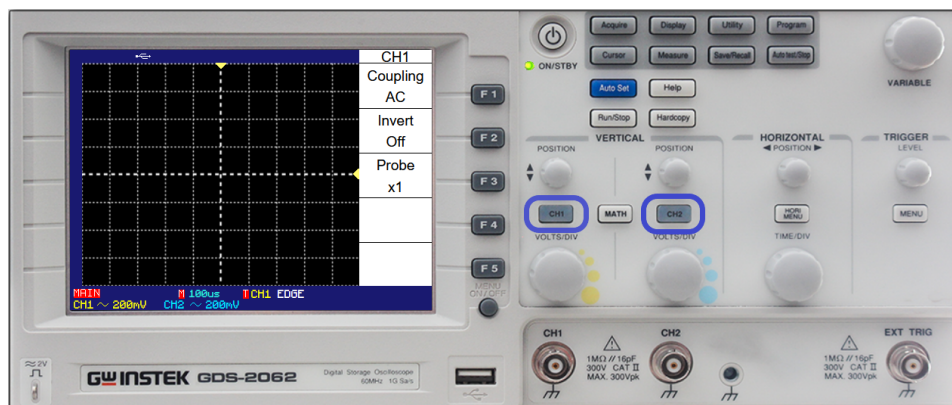


Figura 19: Botões dos menus dos canais 1 e 2

Quando o menu está ligado, é possível visualizar o sinal do CH1 e também visualizar o menu lateral. Se não existir nenhum sinal de entrada, o sinal que é

Tabela 4.1: Funcionalidades da aplicação

Funcionalidades	Estado
Menus do CH1 e CH2	✓
Representação das formas de onda	✓
Modo XY	✚
Acoplamentos DC, AC e GND	✓
Inversão da forma de onda	✓
Atenuação das pontas de prova	✓
Escalas verticais e horizontais	✓
Posições verticais e horizontais	✓
Menu <i>measure</i>	✓
<i>Display all</i>	✓
Menu <i>trigger</i>	✓
Nível do <i>trigger</i>	✓
Gerador de sinais	✓
Leitura de ficheiros externos CSV	✓
Aquisição da forma de onda do osciloscópio real	✓
<i>Run stop</i>	✓
<i>Auto set</i>	✓
Gravação da imagem do ecrã	✓
Gravação da forma de onda no formato CSV	✓
Gravação da imagem do <i>display all</i>	✓
Operações matemáticas da soma e subtração	✓
Comunicação com o osciloscópio remotamente	✓
Menu de suporte	✗
Cursores	✗

apresentado é um sinal contínuo de 0 volts tal como é possível visualizar na Figura 20.

No caso de o canal estar ligado contudo, o menu lateral não ser o do canal 1, como mostrado na Figura 21, ao clicar no botão do CH1, altera o menu lateral para o menu do CH1 igual ao mostrado na Figura 20.

Para poder desligar o canal, é necessário estar a mostrar o menu lateral do canal 1 e só depois clicar no botão para desligar.

#### 4.2.2 Escalas verticais e horizontais

Clicar com o botão esquerdo do rato no botão na escala vertical do CH1 irá aumentar uma grandeza na escala que está representada na Figura 22. Com o clique do botão direito do rato a escala irá diminuir. Quando o rato está sobre o botão, caso o *scroll* seja deslizado para cima, a escala aumenta e caso seja deslizado para baixo, a escala diminui. Este mesmo raciocínio aplica-se para a escala horizontal, sendo que desta

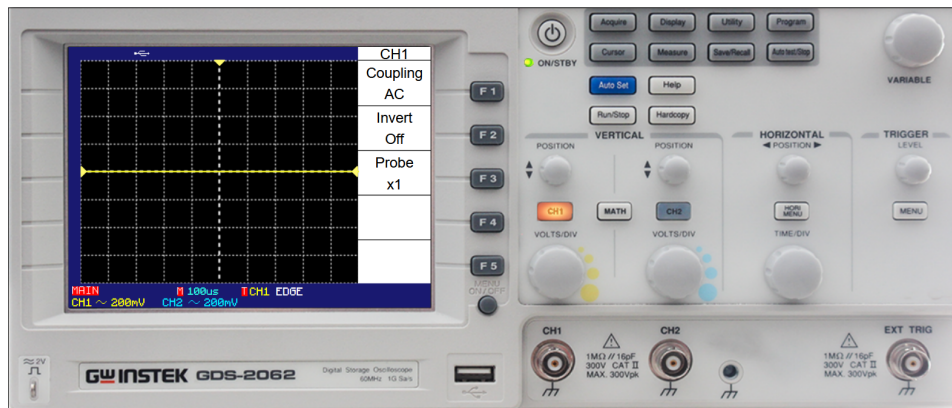


Figura 20: Canal 1 ligado

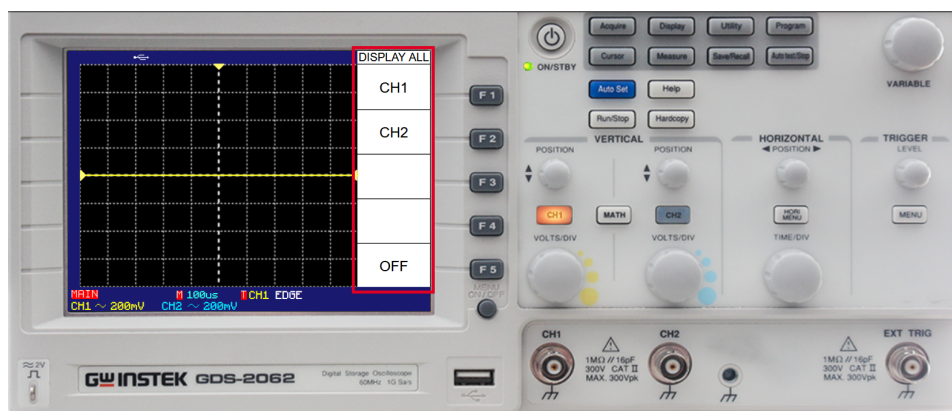


Figura 21: Canal 1 ligado com outro menu lateral selecionado

vez a escala é diferente, como é possível verificar na Figura 23.

### 4.2.3 Posições verticais e horizontais

Ao clicar com o botão direito do rato no manípulo das posições verticais ou horizontais, é decrementado o valor da posição em 0,01, sendo que este valor está contido no intervalo de  $[-1;1]$ . Ao clicar com o botão esquerdo, o valor é incrementado em 0,01.

Se quando o rato estiver sobre o manípulo, for deslizado o *scroll* para cima, o valor é incrementado em 0,05. Caso seja deslizado para baixo é decrementado 0,05.

Este valor serve para depois, quando multiplicado pelo valor que representa quatro escalas verticais, ou cinco escalas horizontais no caso de ser a posição horizontal, apresentar o valor em volts ou segundos, respetivamente, que o sinal foi deslocado.

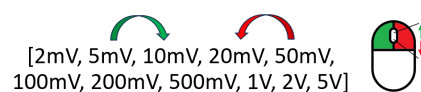


Figura 22: Grandezas da escala vertical



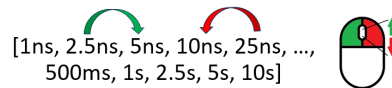


Figura 23: Grandezas da escala horizontal

Para além de calcular o valor da posição em volts ou segundos, este valor também indica a posição em que o cursor representado na Figura 24 precisa de se deslocar de forma a poder indicar o deslocamento da onda.



Figura 24: Cursor da posição vertical do canal 1

#### 4.2.4 *Auto set*

Caso apenas um sinal esteja representado, como no caso da Figura 25, ao carregar na *auto set*, o sinal irá auto-ajustar-se de forma a que a posição vertical passe para zero, a escala vertical fique adequada de forma a poder ser possível visualizar toda a forma de onda e também a escala horizontal é atualizada de forma a poder se visualizar pelo menos 1 período. A Figura 26 mostra o mesmo sinal após o *auto set*.

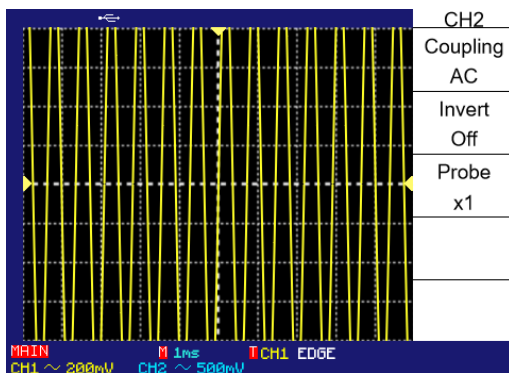
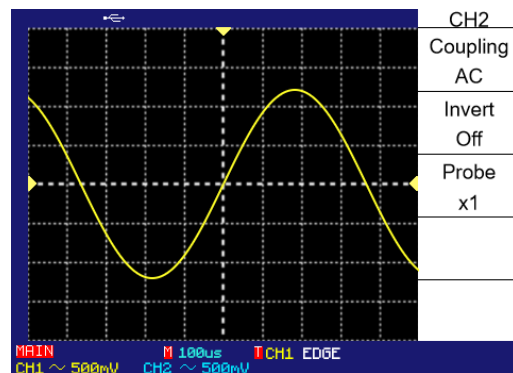


Figura 25: Sinal mostrado no canal 1 não otimizado

Figura 26: Sinal ajustado com o *auto set*

Se ambos os sinais estiverem ativos, o CH1 será redirecionado para ocupar a metade superior do gráfico, ajustado com a posição vertical, e o CH2 para a metade inferior do gráfico, sendo que desta vez a escala vertical irá ser ajustada tendo em conta o novo limite que é a metade que em que estão localizados. O *auto set* irá adaptar a escala horizontal para o canal 2. Na Figura 27 é possível visualizar o resultado de como ficaria o *auto set* de dois sinais ligados em simultâneo.

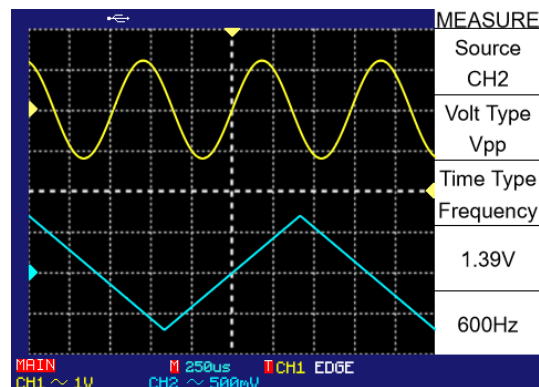


Figura 27: Sinais do CH1 e CH2 ajustados com *auto set*

#### 4.2.5 Modos de gravação

Ao seleccionar o botão *hardcopy* do osciloscópio aparece um menu *offcanvas* que mostra ao utilizador as diversas formas de gravar os dados, sejam estes armazenados através de imagens ou num ficheiro CSV, tal como mostra a Figura 28.

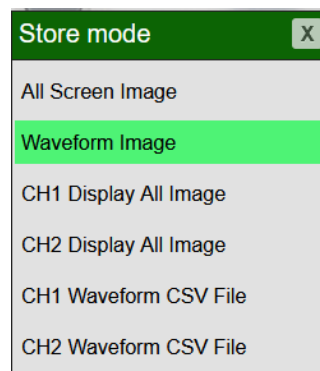


Figura 28: Escolha do modo de gravação

Quanto às capturas de imagens, foram criados 2 modos que guardam diferentes informações, o primeiro permite capturar o ecrã completo e a segunda apenas guarda a forma de onda.

É também possível armazenar os valores temporais e de tensão da forma de onda no formato CSV. Este modo de gravação permite guardar qualquer tipo de forma de onda seja ela gerada pelo simulador ou pelo osciloscópio real, sendo que o que será registado é somente a informação de um período.

Na Figura 31, é possível visualizar as informações sobre os primeiros 10 pontos da forma de onda. O primeiro valor é o tempo em que está localizado o ponto em segundos, e o segundo valor é a tensão em volts.

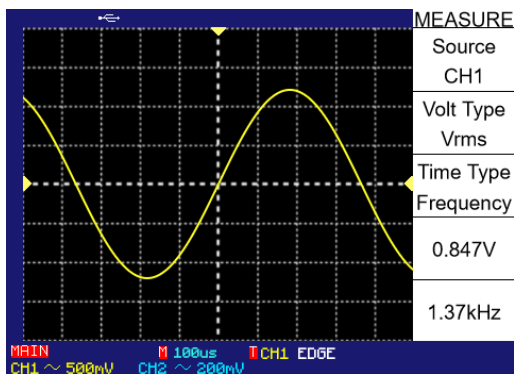


Figura 29: Exemplo da captura do ecrã completo

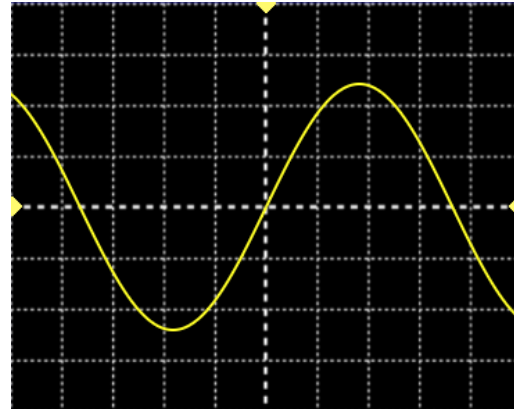


Figura 30: Exemplo da captura da forma de onda

```
Time,Amplitude
0,0
0.0000029197080291970804,0.030120000000000004
0.000005839416058394161,0.060240000000000001
0.000008759124087591241,0.090360000000000002
0.000011678832116788322,0.120480000000000002
0.000014598540145985402,0.150360000000000002
0.000017518248175182482,0.180240000000000004
0.000020437956204379563,0.210000000000000002
0.000023357664233576643,0.239640000000000002
0.000026277372262773727,0.269160000000000007
0.000029197080291970804,0.298440000000000004
0.00003211678832116788,0.327600000000000006
```

Figura 31: Exemplo da captura dos valores da forma de onda num ficheiro CSV

#### 4.2.6 Measure

Ao seleccionar o botão *measure*, é mostrado o menu lateral que contém informações sobre os valores medidos da forma de onda.

Na Figura 32, está apresentado o menu lateral do *measure*.

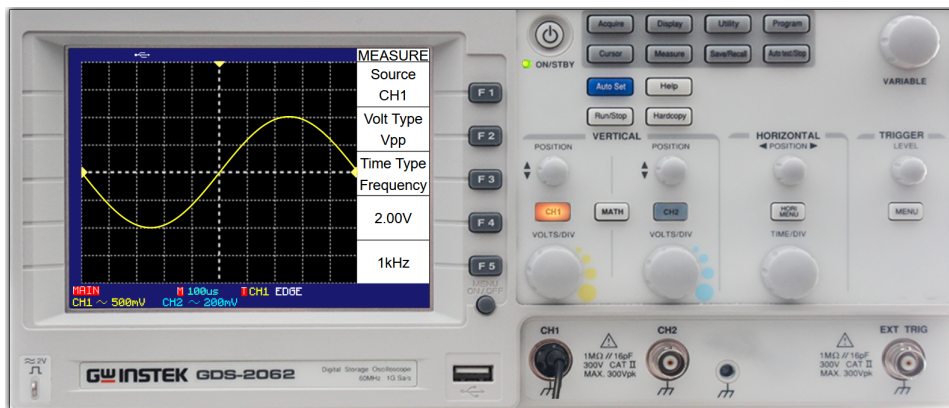
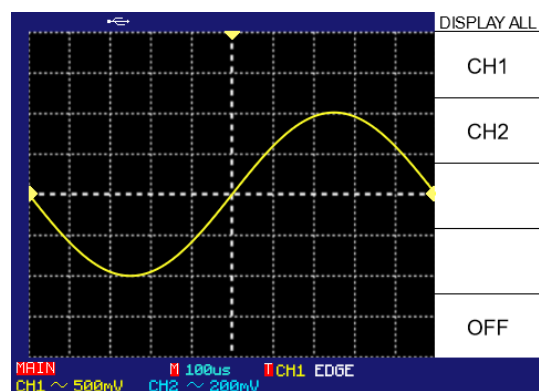
Ao pressionar o botão F1, é possível escolher o sinal que se pretende avaliar.

O botão F2 e F3 contém as diferentes possíveis medidas que são calculadas e apresentadas no quarto e quinto espaço.

No segundo espaço existe uma lista das diferentes grandezas de tensão: [Vpp, Vmax, Vmin, Vrms, Vavg].

No terceiro espaço, a lista contém grandezas temporais: [Frequency, Period].

Com o menu *measure* seleccionado, ao voltar a premir o botão do *measure*, é mostrado o menu lateral do *display all*, tal como é possível verificar na Figura 33.

Figura 32: Menu *Measure*Figura 33: Menu *Display All*

Ao pressionar o botão F1 ou F2, irá aparecer um *offcanvas* no ecrã que contém todas as medidas de cada canal, este *offcanvas* consegue ser arrastado e minimizado.

#### 4.2.7 *Run/Stop*

O botão *run/stop* destacado na Figura 34 apenas está funcional caso o sinal seja proveniente do osciloscópio real. O que este botão permite fazer é parar o pedido de atualização da forma de onda.

Quando o modo de simulação com aquisição do sinal está parado, denomina-se por modo *Hold*. O sinal pode ser retomado ao clicar novamente no botão.

#### 4.2.8 Mensagens de alerta

Tal como no osciloscópio real, neste simulador também existem mensagens de alerta temporários (normalmente duram 2 segundos na tela, sendo que outros poderão ser permanentes até que a ação sugerida seja efetuada) que indicarão ao utilizador alguns pormenores que estejam a acontecer e não possam ser facilmente entendidos. Estas indicações podem por exemplo indicar qual a posição vertical do canal quando esta é modificada, tal como é possível visualizar na Figura 35.

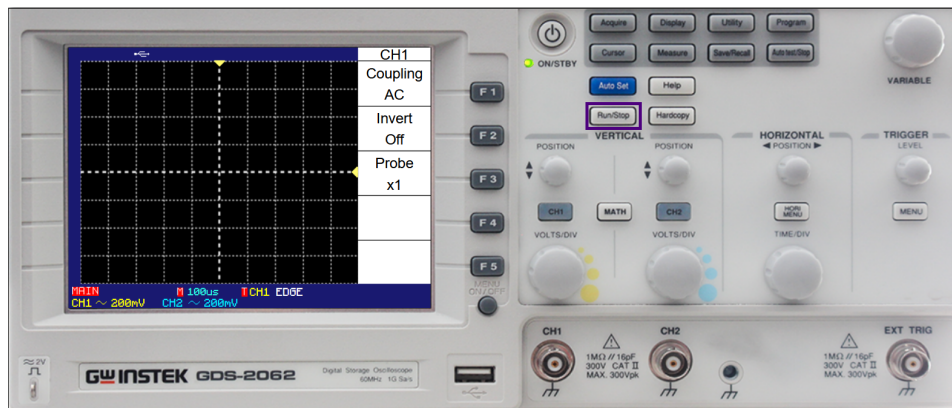
Figura 34: Botão *Run/Stop*

Figura 35: Mensagem de alerta que indica a posição vertical do canal

1

## 4.3 Modo de Simulação

Neste modo é possível estudar formas de onda provenientes de ficheiros CSV, ou do osciloscópio real. Neste modo quem realiza os cálculos para determinar os valores das formas de onda é o computador, exceto se o sinal for proveniente do osciloscópio pois os valores são pedidos e enviados juntamente com a forma de onda.

Todas as funcionalidades que alteram valores no simulador não alteram os do osciloscópio pois neste modo é impossível controlar o osciloscópio, apenas receber valores.

### 4.3.1 *Inputs*

Ao seleccionar os botões que estão destacados na Figura 36, irá aparecer um menu *offcanvas* que permitirá ao utilizador seleccionar qual o sinal que quer analisar, mostrado na Figura 37. Os sinais podem ser provenientes do gerador de sinais virtual, de um ficheiro externo CSV ou também do osciloscópio real.

Após seleccionar o sinal pela primeira vez e caso se pretenda modificá-lo pode-se fazê-lo carregando no botão do *input* com o botão direito do rato. Ao carregar com o botão esquerdo irá ocultar o sinal, carregar novamente irá voltar a mostrar o sinal.

### Calibração

A primeira opção que aparece no menu de seleção apresentado no menu da Figura 37, é o sinal que está disponível no próprio osciloscópio real quando se coloca a ponta de prova em contacto com o que está destacado na Figura 38, sendo que neste caso o sinal está contido num ficheiro CSV.

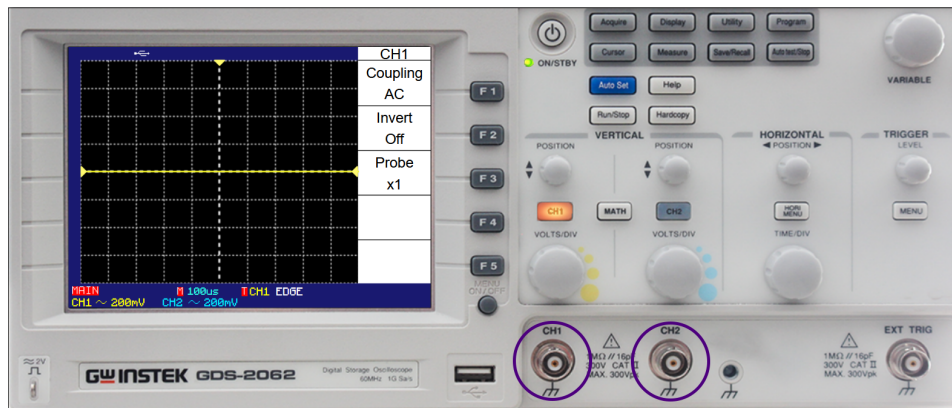


Figura 36: Botões de *input*.

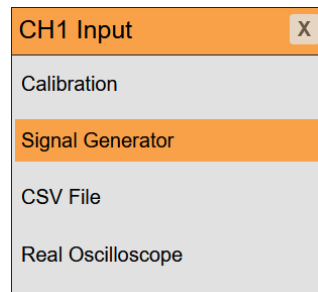


Figura 37: Seleção do sinal de entrada.

### Gerador de Sinais

O gerador de sinais utilizado nesta aplicação tem como objetivo ser um simulador do gerador de sinais real situado nas bancadas do *Laboratório de Circuitos e Sinais Elétricos*, para tal foi necessário tirar uma foto ao gerador de sinais que foi realizada com a câmara do meu telemóvel, mais tarde foi editado e também criado o *imagemap* da imagem para poder ter áreas que fossem interativas.

Na Figura 39, está representado o menu *offcanvas* que é mostrado ao selecionar o *input* "Signal Generator", este tem a possibilidade de ser arrastado e minimizado.

Ao interagir com os botões do gerador de sinais, estes modificam em tempo real o sinal, o que permite ao utilizador modificar e visualizar instantaneamente as alterações efetuadas.

Este gerador de sinais contém as formas de onda sinusoidal, quadrada e triangular tal como no gerador de sinais real e estas formas de onda contém inicialmente 1000Hz e 1V de amplitude, sendo que estes valores de frequência e amplitude podem ser alterados através dos botões da frequência e amplitude.

Após o utilizador selecionar a forma de onda, pode então modificar a frequência, amplitude e offset do sinal.

A frequência contém os botões que funcionam como escalas, que permitem modificar rapidamente a frequência do sinal para valores múltiplos de 10, apenas se

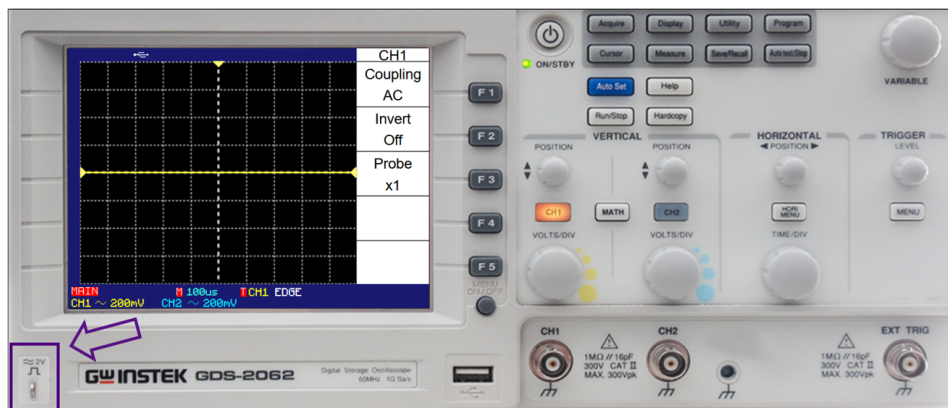


Figura 38: Calibração do osciloscópio



Figura 39: Interface do gerador de sinais

pode ter um destes botões acionados de cada vez.

Para além da seleção da escala, o gerador de sinais também tem um manípulo, mostrado na Figura 43 que indica o múltiplo pelo qual a escala de frequência que está selecionada irá ser multiplicada.

Quando é acionado com o botão esquerdo do rato, aumenta em 0,02 o valor do múltiplo, caso seja acionado com o botão direito do rato, este valor diminuirá em 0,02, sendo que o intervalo de valores possíveis é  $[0,2;2]$ .

Quando o rato estiver sobre o manípulo, pode-se também utilizar o *scroll*, caso deslize para cima, o valor irá ser incrementado em 0,1 de cada vez, caso deslize para baixo, será decrementado em 0,1.

Os manípulos da amplitude e *offset* também incrementam e decrementam da mesma forma que o da frequência, sendo que o que altera neste caso são os limites que agora são de  $[0V;10V]$ .

### Ficheiro CSV

As formas de onda do gerador de sinais e da calibração são provenientes de ficheiros CSV que estão armazenados dentro do código. Este modo de entrada permite estudar outro sinal exterior que esteja representado por um ficheiro CSV.

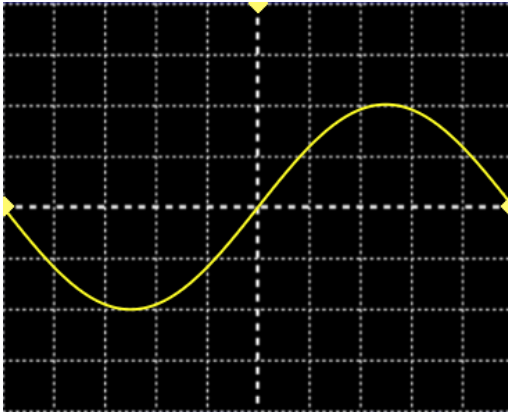


Figura 40: Forma de onda sinusoidal do gerador de sinais

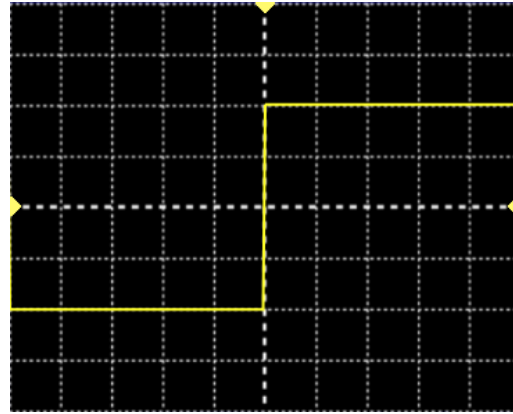


Figura 41: Forma de onda quadrada do gerador de sinais

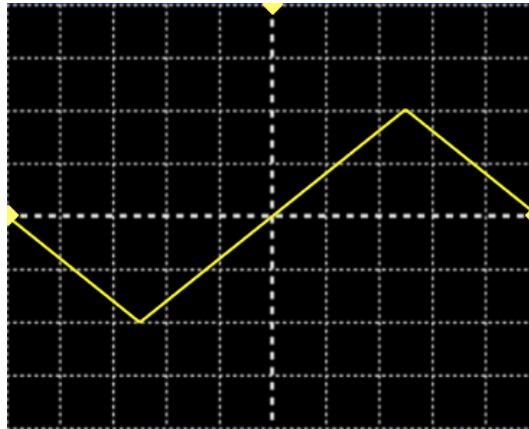


Figura 42: Forma de onda triangular do gerador de sinais

De forma a poder ser considerado um ficheiro válido para análise, o ficheiro precisará de ter:

- Duas colunas

A primeira representa o tempo em segundos a que cada ponto está localizado na escala temporal

O segundo indica a tensão do ponto em volts.

- *Header*

Deve apenas conter a linha "*Time,Amplitude*" pois na aplicação os valores de tempo são tratados como "*Time*" e os valores de tensão como "*Amplitude*".

- Separação por vírgulas

Todos os valores devem ser separados por vírgulas, todos os valores decimais deverão ser separados por pontos.





Figura 43: Manípulo da frequência

- Tamanho

De forma a poder ser possível estudar diferentes tipos de formas de onda facilmente, o tamanho, mais em concreto, o número de pontos que o ficheiro CSV tem, pode ser qualquer valor.

- Um período

No ficheiro, o primeiro ponto deve ser o ponto inicial sendo que este mesmo ponto deverá se repetir no último ponto, fechando assim um período completo da forma de onda. Independentemente do que seja submetido, o código interpretará que o ficheiro CSV contém um período completo da forma de onda.

Um modelo do que é pretendido está apresentado na Figura 44.

1	Time,Amplitude	242	0.00096,-0.2487
2	0,0.0000	243	0.000964,-0.2243
3	0.000004,0.0251	244	0.000968,-0.1997
4	0.000008,0.0502	245	0.000972,-0.1750
5	0.000012,0.0753	246	0.000976,-0.1502
6	0.000016,0.1004	247	0.00098,-0.1253
7	0.00002,0.1253	248	0.000984,-0.1004
8	0.000024,0.1502	249	0.000988,-0.0753
9	0.000028,0.1750	250	0.000992,-0.0502
10	0.000032,0.1997	251	0.000996,-0.0251
11	0.000036,0.2243	252	0.001,0.0000

Figura 44: Ficheiro CSV da forma de onda sinusoidal

### Sinal real proveniente do osciloscópio

Ao seleccionar a última opção do menu, é possível receber o sinal que estiver a ser estudado no osciloscópio. Será recebido pelo cliente os pontos da forma de onda bem como os valores das medidas da forma de onda. Ao estar neste modo também será

possível interromper a atualização da forma de onda pressionando o botão Run/Stop. É possível ter num dos canais o sinal de entrada proveniente do osciloscópio e noutra entrada um sinal simulado, tal como se pretende mostrar na Figura 45.

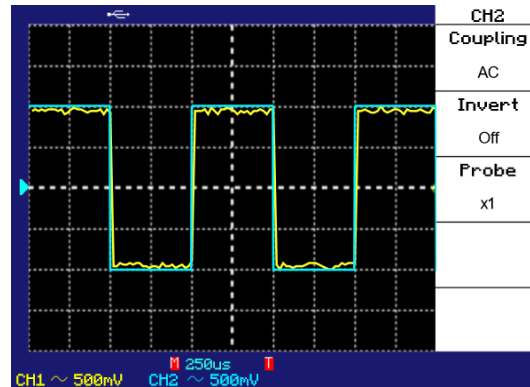


Figura 45: Forma de onda real e simulada apresentadas simultaneamente

## 4.4 Modo de aquisição e controlo remoto

Este modo de funcionamento apenas funciona com um utilizador de cada vez, pois este modo permite um sincronismo total entre osciloscópio e utilizador.

As funcionalidades já apresentadas no modo de simulação são utilizadas de forma semelhante.

Este modo tem como objetivo espelhar a interface do osciloscópio real, o que torna possível manipular o osciloscópio remotamente, fazendo com que grande parte das funcionalidades presentes no osciloscópio real consigam ser representadas.

Para o utilizador, utilizar este modo ou utilizar o osciloscópio de bancada são a mesma experiência.

### 4.4.1 Funcionalidades que não estão implementadas face ao modo simulação

#### *Inputs*

Como o objetivo deste modo é espelhar o osciloscópio, então o *input* dos sinais não é possível de se realizar, pois caso o utilizador pretenda estudar um sinal gerado no computador em conjunto com o do osciloscópio real, pode então mudar para o modo de simulação e selecionar ambos os sinais de entrada em cada canal.

## 4.5 Limitações do osciloscópio

Tanto no modo de aquisição e controlo remoto como no modo de simulação, o que envolve pedidos de valores ao osciloscópio é muito limitado devido ao facto de não existir forma de saber o que pedir. Caso o osciloscópio tivesse forma de conseguir enviar uma informação que dissesse quais valores tinham sido alterados desde o último envio, permitiria que os pedidos ao osciloscópio pudessem ser muito mais rápidos garantindo baixa latência na atualização da informação e também menos sobrecarregamento do osciloscópio.



## Capítulo 5

# Teste, depuração e melhoramentos

### 5.1 Identificação de erros/lacunas

#### 5.1.1 Fase preliminar de desenvolvimento/testes por parte da equipa envolvida

Esta aplicação encontra-se ainda numa fase preliminar de testes por parte da equipa de desenvolvimento.

Ainda não está disponibilizada a possibilidade de haver comunicação externa sobre possíveis erros ou lacunas da aplicação.



## Capítulo 6

# Considerações Finais

### 6.1 Conclusões sobre o Desenvolvimento do Projeto

#### 6.1.1 Aprendizagem e evolução

O desenvolvimento deste projeto representou um marco significativo tanto para o meu percurso acadêmico como no meu desenvolvimento pessoal e profissional, pois é até à data o empreendimento com maior complexidade que realizei.

A realização deste projeto obrigou-me a desenvolver conhecimentos e competências em distintas e diversificadas áreas como a área da programação, as problemáticas associadas à concepção de uma aplicação informática, a gestão do tempo e a definição de prioridades, a gestão dos recursos físicos e até mesmo a atitude, muitas vezes, profissional que tive de desenvolver para levar por diante a realização deste projeto.

No decorrer da realização do projeto fui-me deparando com mudanças de abordagem da execução técnica que ia sendo necessário implementar para resolver os desafios técnicos inerentes ao desenvolvimento de uma aplicação que fosse capaz de comunicar ou substituir virtualmente o funcionamento de um osciloscópio.

Ao mesmo tempo que me deparava com a necessidade de introduzir uma nova abordagem para resolver um problema específico, deparei-me frequentemente que para realizar tal tarefa, iria necessitar de adquirir conhecimentos de tecnologias e utilizar ferramentas que desconhecia e não dominava.

As diferentes aprendizagens que realizei e todo o amadurecimento ocorrido permitem-me hoje constatar o quanto a realização deste projeto me permitiu crescer nas mais

diversas áreas, mas, por outro lado também me permite ter a consciência de que muito mais é possível fazer no âmbito de otimizar e implementar novas funcionalidades à aplicação já desenvolvida.

Sendo o âmbito deste projeto de nível acadêmico e tendo uma limitação temporal pré-definida, com o decorrer do tempo, acaba por se evidenciar a necessidade de definir prioridades de modo a atingir os objetivos essenciais definidos para o projeto.

### 6.1.2 Melhores práticas para futuros projetos

Com a realização deste projeto, concluo que para projetos futuros semelhantes será importante estabelecer as seguintes práticas:

#### Planificação prévia flexível

Deve ser criado um plano detalhado, no qual se definam metas claras e os passos necessários para alcançar essas mesmas metas, contruindo assim uma base sólida para o desenvolvimento do projeto. No entanto, como os desafios técnicos e as solicitações se podem e quase seguramente, se irão alterar com o decorrer do projeto, será necessário que esta planificação inicial possa ser adaptada e seja flexível para que o processo de construção da aplicação possa prosseguir de forma ágil e eficaz.

#### Definição de regras de comunicação e organização

Na linha de pensamento anterior, a definição de regras de comunicação e a implementação de procedimentos de organização entre o aluno e os orientadores podem ser uma excelente ajuda para um desenvolvimento rápido e assertivo do projeto.

Criar um código bem organizado e de fácil leitura não só facilita o entendimento da programação desenvolvida, como também fortalece a capacidade de enfrentar os desafios mais complexos que possam surgir, conjuntamente com os orientadores ou outros colaboradores da aplicação.

### 6.1.3 Avaliação das conquistas

Os objetivos que foram propostos no início do projeto foram sendo cumpridos, no entanto no decorrer do projeto foram se atribuindo novos objetivos mais pormenorizados com vista à realização de uma funcionalidade extra ou tentativa de melhoramento da interface gráfica para que esta se assemelha-se mais à interface real.

Por desconhecimento de alguns aspetos técnicos, certos objetivos que iam sendo propostos acabavam por se manifestar como sendo irrealistas tendo em conta o tempo disponível para desenvolver a aplicação.

No decorrer do desenvolvimento do projeto, por vezes quando atingia uma meta, surgiam mais dois ou três novos pontos a acrescentar à lista de objetivos a atingir. De certa forma, a introdução de mais e novos objetivos, considerando que o tempo



ia ficando sempre mais curto, poderia, se não tivesse existido uma definição e gestão de prioridades, ter criado uma situação que facilmente teria saído do controlo e poria em causa a conclusão deste projeto/aplicação.

## 6.2 Perspetivas de trabalho futuro

Apesar de se ter usado o conhecimento e referências existentes de outros projetos e existentes na documentação do próprio osciloscópio, esta aplicação *web* foi desenvolvida de raiz e deste modo foi possível criar uma aplicação que tem o potencial para:

- Poder ser disponibilizada no U=RIolve [6] *Academy* sendo também possível *quizzes* e outros métodos para motivar o ensino e utilização do osciloscópio aos alunos.
- Pode ser acrescida de uma funcionalidade que utilize novos métodos para analisar as formas de onda, por exemplo, análises espectrais, analisar sinais óticos, sinais de radiofrequência, etc.
- Pode ser acrescida de uma funcionalidade que permita captar sinais de entrada através de placas de som, ou da análise em tempo real de um simulador de circuitos elétricos como o *Qucs*.
- Pode ser também melhorada as semelhanças visuais entre o osciloscópio real e a aplicação; e podem também ser acrescidas à aplicação todas as demais funcionalidades do osciloscópio que ainda não estejam implementadas na aplicação.

Tendo em conta as potencialidades referidas atrás, acredito que esta aplicação pode ser muito mais aproveitada e explorada tanto a nível educativo como profissional.



# Referências

- [1] P. Salgueiro, “Simulador de um osciloscópio analógico,” Master’s thesis, Instituto Superior de Engenharia do Porto, Porto, 2005. [Citado na página 8]
- [2] J. Pereira, “Simulador de um osciloscópio digital,” Master’s thesis, Instituto Superior de Engenharia do Porto, Porto, 2015. [Citado na página 8]
- [3] “Imagemap.” Acedido no dia 5 de setembro de 2023. [Citado na página 15]
- [4] “W3schools.” Acedido no dia 5 de setembro de 2023. [Citado na página 15]
- [5] GW-Instek, *GDS-2000 Series Programming Manual*. [Citado na página 18]
- [6] “U=ríolve.” Acedido no dia 5 de setembro de 2023. [Citado na página 45]



## Anexo A

# Adaptação da preparação do guião de TCIRC

A preparação do guião 2: "Funcionamento do Osciloscópio e Gerador de Sinais", tem o objetivo de introduzir o osciloscópio aos alunos através dos simuladores de osciloscópio que já foram desenvolvidos nos projetos anteriores.

Na Figura 46, encontra-se a atual versão do exercício 13 que utiliza o simulador de osciloscópio digital do João Pereira para reesolver dois exercícios exemplo.

Tendo em conta as diferenças existentes entre aplicações, foi necessário adaptar o exercício para poder ser realizado nesta aplicação. A sugestão do que seria o novo exercício está apresentada em baixo:

### A.1 Novo Exercício 13

Aceda ao simulador de osciloscópio digital, com o *link* do *moodle*. Dica: Entre simulações reinicie a página.

#### A.1.1 Simulação 1

- Ligue o osciloscópio e seleccione o modo de simulação.
- Clique na entrada do canal 1 (CH1)(na parte inferior do osciloscópio).

Faça as seguintes simulações no simulador de osciloscópio digital, disponível no Moodle:

### Simulação 1

- Ative a entrada do Canal 1 (CH1), clicando com o botão esquerdo do rato na entrada (tomada do tipo BNC) desse canal (na parte inferior do Osciloscópio)
- Clicando com o botão direito do rato nessa mesma entrada, selecione a opção "Signal Generator" (deverá aparecer a janela de interface do gerador de sinais)
- Clique no botão Measure, na parte superior do Osciloscópio
- Clique no botão F2 (do lado direito do ecrã do Osciloscópio) até o Volt Type indicar a opção Vrms (tensão eficaz do sinal de entrada)
- Clique no botão F3 (do lado direito do ecrã do Osciloscópio) até o Time Type indicar a opção Frequency (frequência do sinal de entrada)
- Configure o gerador de sinais para um sinal sinusoidal com uma frequência de 200 Hz e uma tensão eficaz de aproximadamente 7 V; o valor eficaz é indicado diretamente no painel direito do ecrã (Vrms)
- Configure o comando de amplificação vertical para 5 V/DIV;
- Para as seguintes velocidades de varrimento: 1 ms/DIV, 500  $\mu$ s/DIV e 250  $\mu$ s/DIV
  - gravar o ecrã do Osciloscópio (menu 'File -> Save -> Screen') e depois inserir as 3 imagens num único documento (e.g. MSWord ou OpenOffice);
- Anexar o documento na caixa abaixo.

### Simulação 2

- Para ambos os canais (CH1 e CH2): garantir que o GND (referência de 0 V) está centrado no ecrã, escolher acoplamento DC e amplificação vertical de 1 V/DIV;
- Na Base de Tempo, escolher a velocidade de varrimento de 250  $\mu$ s/DIV;
- Introduzir no canal 1 (CH1) um sinal sinusoidal (escolher a fonte 'Signal Generator', clicando em cima da tomada BNC do CH1 com o botão direito do rato) com as seguintes características:
  - frequência  $f = 1$  kHz (o valor de frequência gerado aparece diretamente na janela do gerador de sinais);
  - tensão pico-a-pico  $U_{pp} = 4$  V (podem verificar o valor através da indicação digital fornecida pelo osciloscópio, em Grandezas dos sinais de entrada, escolhendo o parâmetro tensão pico-a-pico);
- Introduzir no canal 2 (CH2) o sinal de 'Calibration' (no Osciloscópio real, trata-se de um sinal de referência - onda quadrada - gerado pelo próprio Osciloscópio, que serve para calibrar as Pontas de Prova)
- Regular o posicionamento vertical (botão Position; clique esquerdo do rato sobe, clique direito desce) do CH1 duas divisões para cima e o do CH2 duas divisões para baixo.
- Gravar o ecrã do Osciloscópio (menu 'File -> Save -> Screen') num ficheiro .JPEG
- Anexar a imagem na caixa abaixo.

Figura 46: Exercício 13 da preparação do guião 2

- Selecione a opção "*Signal Generator*" (deverá aparecer a janela de interface do gerador de sinais).
- Sem fechar o gerador de sinais clique sobre o botão "*Measure*" na parte superior do osciloscópio.
- Clique no botão f2 (do lado direito do ecrã do osciloscópio) até o "*volt type*" indicar a opção "*Vrms*" (valor da tensão eficaz).
- Configure o gerador de sinais para um sinal sinusoidal com uma frequência de 200 Hz (com o cursor sobre o manipulo da frequência do lado esquerdo do gerador de sinais deslize o scroll para baixo para diminuir a frequência, ou clique com botão direito do rato); ajuste também a tensão eficaz para aproximadamente 3V (pode verificar ambos os valores no menu lateral do *measure*).

- Configure o comando da amplificação vertical para 2V/Div
- Para as velocidades de varrimento: 1ms/Div; 500us/Div e 250 us/Div:  
Selecione o botão “*Hardcopy*” e depois escolha a opção “*Store All Screen Image*” para guardar a imagem da interface.
- Após guardar as três imagens (uma com cada escala temporal) junte-as num ficheiro *Portable Document Format* (PDF) e anexe na caixa abaixo com o nome “simulacao1”.

### A.1.2 Simulação 2

- Ligue ambos os canais, para o CH1 introduza uma forma de onda proveniente do gerador de sinais e para o CH2 selecione a opção “*CSV File*”.
- Após selecionar a opção “*CSV File*”, deve escolher a forma de onda “*sawtooth*” que está presente na pasta “*waveforms*” no *moodle*.
- No gerador de sinais para o CH1 selecione uma onda triangular com uma frequência de 2500Hz (precisa de primeiro selecionar o botão “10k” e depois ajustar no manípulo da frequência até chegar aos 2500Hz); com uma amplitude pico a pico de 4 volts, consegue visualizar a tensão pico a pico e a frequência no menu “*measure*”.
- De seguida configure o comando de amplificação vertical do CH1 para 1V/Div e do canal 2 para 500mV/Div.
- Ajuste também a velocidade de varrimento para os 100us/Div.
- Selecione o botão “*measure*” e selecione o f1 para alterar o canal fonte para o CH2; selecione o “*Volt Type*” para *Vrms* e o “*Time Type*” para *Period*.
- Selecione o botão “*Hardcopy*” e selecione para gravar o “*Store All Screen Image*”.
- Ajuste a posição vertical do CH1 para 2V (o valor em volts da posição vertical é mostrado nas mensagens de alerta que aparecem no canto inferior esquerdo do osciloscópio ao modificar a posição). E ajuste também a posição vertical do CH2 para -1V.
- Selecione novamente o botão “*Hardcopy*” e desta vez selecione a opção “*Store Waveform Image*”.
- Junte ambas as imagens num ficheiro do formato PDF e submeta no anexo com o título: “simulacao2”.