



PROJETO, IMPLEMENTAÇÃO E INTEGRAÇÃO
DO MÉTODO DAS CORRENTES DE MALHA
NA APLICAÇÃO $U=RI$ SOLVE

Licenciatura em Engenharia Eletrotécnica e de
Computadores

Professor Orientador

Mário Alves
mjf@isep.ipp.pt

Professor Coorientador

André Rocha
anr@isep.ipp.pt

Autor

Ângelo Pinheiro
1190398@isep.ipp.pt

8 de julho de 2022

Agradecimentos

Declaro nesta secção os meus agradecimentos a todos os que contribuíram para o desenvolvimento do projeto:

Aos professores Mário Alves e André Rocha cuja orientação de excelência foi indispensável para a realização e aperfeiçoamento do projeto.

À minha namorada pelo apoio prestado em momentos menos fáceis e por acreditar sempre em mim e na capacidade de superar problemas.

Ao meu colega e amigo Alexandre Nunes pela ajuda e “BrainStorming” com questões mais técnicas de implementação de certos algoritmos.

E por fim a todos os que participaram na fase de testes do método e que assim permitiram a resolução de problemas.

“The power and beauty of physical laws is that they apply everywhere, whether or not you choose to believe in them. In other words, after the laws of physics, everything else is opinion.”

Neil de Grasse Tyson

Resumo

A aplicação *U=RSolve* é uma ferramenta de simulação para fins de ensino e autoaprendizagem dos métodos de análise de circuitos elétricos, disponibilizando uma resolução passo-a-passo, completa e intuitiva, ao contrário da esmagadora maioria dos simuladores de circuitos, que se limitam a mostrar os valores/resultados da simulação, não explicando portanto o processo de obtenção desses resultados. Esta característica está presente no *U=RSolve* e é fundamental no início do desenvolvimento de um engenheiro eletrotécnico pois fornece-lhe um apoio extra na resolução e estudo de circuitos elétricos.

Anteriormente à realização deste projeto, a *app* dispunha apenas do *Método da Tensão nos Nós (MTN)* como opção para simulação e resolução de circuitos [23]. Este documento descreve o processo de projeto, implementação e integração do *Método das Correntes nas Malhas (MCM)* na aplicação já existente.

A análise de qualquer circuito inicia-se pela criação digital do seu esquema elétrico, de maneira a obter uma “Netlist”, que é um ficheiro de texto que contém os componentes que integram o circuito e as relações de conexão entre eles. Após este passo resta submeter o ficheiro na aplicação, acompanhado ou não de uma imagem do esquema elétrico do circuito, e escolher um dos métodos de análise [23]. O resultado é uma análise detalhada do circuito, com diversas informações tais como as suas variáveis fundamentais (ramos, (Nós), fontes de corrente e fontes de tensão isoladas), frequência do circuito/simulação, detalhes sobre as equações a construir em cada um dos métodos (Equações dos Nós no caso do *MTN*, ou Equações das Malhas no caso do *MCM*), sistema de equações, resultados das correntes, etc. Há ainda a possibilidade de vários outputs, sendo o Portable Document Format (PDF), o TeX e o JavaScript Object Notation (JSON) os principais.

A escolha da arquitetura em que a *app* assenta teve como critério a acessibilidade e facilidade de uso. Para cumprir estes objetivos a ferramenta funciona em qualquer browser pois é baseada em HTML e CSS no que toca a apresentação de resultados, e *JavaScript* para realizar todas as computações necessárias até os obter.

Abstract

The *U=RI solve* application is an electrical circuit simulation and analysis tool for (self) learning and teaching electrical circuit analysis. It provides a step-by-step, complete and intuitive resolution, unlike most circuit simulators that “just” output results. We believe that, similarly to other tools such as the Khan Academy, this turns out to be crucial at an early stage in learning electrical circuits analysis.

Prior to this project, The *U=RI solve* application only featured the *Node Voltage Method* (NVM). Hereby we outline the design, implementation and integration of the *Mesh Current Method* (MCM) in the already existent app.

Any circuit analysis begins with the digital creation of its electrical diagram, in order to obtain a “Netlist” which is a text file containing the components that constitute the circuit as well as the connection relationships between them. After this step it remains to submit the file in the application, accompanied or not by an image of the electrical diagram, and to choose one of the analysis methods. The result is a detailed analysis of the circuit, with various informations such as the fundamental variables (branches, nodes, current sources and isolated voltage sources), circuit/simulation frequency, details about the to be built equations according to the method used (Node equations in NVM, or Mesh equations in MCM), equation system, current results, etc. There is even the possibility of various outputs, with Portable Document Format (PDF), TeX and JavaScript Object Notation (JSON) being the main types.

The choice of architecture in which the app is based on had the accessibility and ease of use as the main criteria. In order to fulfill those objectives, the tool works in any browser since it's based on HyperText Markup Language (HTML) and Cascading Style Sheets (CSS) as far as the results presentation goes, and *JavaScript* to perform all the needed computations until obtaining them.

Conteúdo

1	Introdução	10
1.1	Contextualização	10
1.2	Definição de Objetivos	10
1.3	Descrição da Abordagem e Tarefas	10
1.4	Contribuições	11
1.5	Organização do Relatório	11
2	Terminologia e Conceitos Fundamentais	13
2.1	Terminologia	13
2.2	Métodos gerais de análise de circuitos elétricos	14
2.2.1	Método da Corrente nos Ramos (MCR)	15
2.2.2	Método da Tensão nos Nós (MTN)	16
2.2.3	Método da Corrente nas Malhas (MCM)	18
3	Análise de Tecnologias relacionadas	21
3.1	<i>Symbolab</i>	21
3.2	<i>Microsoft Math Solver</i>	21
3.3	<i>WolframAlpha</i>	22
3.4	<i>Falstad</i>	23
3.5	<i>PhotoMath</i>	23
3.6	Comparação com a ferramenta <i>U=RI solve</i>	24
4	Algoritmo MCM	25
4.1	Aplicação a um circuito-exemplo (contendo uma fonte ideal de corrente)	25
4.2	Enquadramento Prático	26
4.3	Encapsulamento em funções de código comum	28
4.4	Implementação do Algoritmo	29
4.4.1	Cálculo do Número de Malhas Principais	30
4.4.2	Determinação das malhas do circuito	30
4.4.3	Algoritmo de escolha de malhas	31
4.4.4	Tratamento das malhas	38
4.4.5	Construção e Resolução do Sistema de Equações	40
4.4.6	Resolução do sistema de Equações	42
4.4.7	Computação das Correntes nos Ramos	42
4.5	Atualizar Ficheiro JSON	43
4.6	<i>Output</i>	44
4.6.1	Capa e estrutura do <i>Output</i>	44
4.6.2	<i>Modal Output</i>	46
4.6.3	<i>“Rich”PDF Output</i>	47
4.6.4	<i>Json Output</i>	48
4.6.5	<i>TeX Output</i>	49
4.6.6	<i>Open in Overleaf</i>	50
4.6.7	<i>Print Output</i>	52

5	Outras funções de suporte ao algoritmo	55
5.1	Arredondamento de Resultados	55
5.2	Desenho das Malhas do circuito	55
6	Conclusões	57
A	Análise MCM de vários circuitos-exemplo	60
A.1	Circuito-exemplo em Corrente Contínua	60
A.2	Circuito-exemplo em Corrente Alternada	69

Lista de Tabelas

1	Calendário do projeto	11
2	Comparação entre as ferramentas apresentadas e U=RI solve . . .	24
3	Formato JSON	32
4	Formato vetor	32
5	Exemplo “ <i>All Variable Equation</i> ” (LaTeX em cima e texto corrido correspondente em baixo)	41
6	Exemplo “ <i>Revealed Current Source</i> ” (LaTeX em cima e texto corrido correspondente em baixo)	41
7	Exemplo “ <i>All Revealed Equation</i> ” (LaTeX em cima e texto corrido correspondente em baixo)	42
8	Exemplo “ <i>Solver Equation</i> ”	42

Lista de Figuras

1	Exemplo de um ramo com três componentes	13
2	Exemplo de um Nó com quatro ramos interligados	13
3	Exemplo de malhas num circuito	14
4	Circuito Exemplo a ser analisado por cada método	15
5	Circuito com as marcações necessárias ao MCR	16
6	Circuito a ser analisado usando o MTN	18
7	Circuito com as marcações necessárias ao MCM	19
8	Interface do Symbolab	21
9	Microsoft Math Solver INterface	22
10	Interface do WolframAlfa	22
11	Interface do Simulador de circuitos do Falstad	23
12	Interface do PhotoMath	24
13	Circuito Exemplo	25
14	Diagrama de blocos ilustrativo do plano/arquitetura do projeto .	27
15	Diagrama de blocos do algoritmo do método da corrente nas malhas	30
16	Exemplo de um Circuito e o Grafo correspondente	31
17	Seleção de malhas menor que M	33
18	Seleção de malhas usando M malhas principais	33
19	Seleção de malhas que não cobre todos os ramos	34
20	Seleção de malhas que cobre todos os ramos	34
21	Fluxograma descritivo da filtragem de malhas virtualmente impossíveis	35
22	Fluxograma descritivo da escolha de “Malhas Auxiliares”	36
23	Fluxograma descritivo da filtragem de malhas que contém fontes de corrente	37
24	Fluxograma descritivo da escolha de “Malhas Principais”	38
25	Exemplo de de uma malha a ser construída	39
26	Exemplo de correntes de malha num componente do circuito . .	41
27	Exemplo de construção de equações das corrente nos ramos de um circuito	43
29	Exemplo de um sistema imprimido usando “jsPDF”	54
30	Exemplo do desenho de uma malha de um circuito	55

Acrónimos

MCM *Método das Correntes nas Malhas.*

MCR *Método das Correntes nos Ramos.*

MTN *Método da Tensão nos Nós.*

CSS Cascading Style Sheets.

HTML HyperText Markup Language.

JPG Joint Photographic Experts Group.

JSON JavaScript Object Notation.

PDF Portable Document Format.

PNG Portable Network Graphic.

SVG Scalable Vector Graphics.

URI Uniform Resource Identifier.

Glossário

Base64 Método de conversão de dados binários arbitrários, dados compostos por mais que caracteres de texto simples, em uma mensagem de texto a ser transferida através de uma rede de computadores.

Fonte Ideal de Corrente Componente elétrico que define a corrente elétrica no ramo que está inserido a cada instante de tempo. Sendo ideal mantém a tensão aos seus terminais constante dependendo esta apenas dos restantes componentes do circuito. [6].

Fonte Isolada de Tensão Fonte de tensão cujos terminais são dois nós do circuito, ou seja, por si só a fonte constitui um ramo.

Grafo É um conjunto de vértices (nós) e de arestas (entre dois vértices) que representam a forma como estes estão (inter)ligados. [20].

Malha percurso elétrico fechado, composto por ramos delimitados por nós, sem repetição nem de ramos nem de nós.

Método de Análise Algoritmo com passos concretos e bem delineados, com o objetivo de descobrir ou as tensões nos nós ou as correntes nos ramos de um circuito.

Netlist Ficheiro de texto representativo de um circuito elétrico montado num simulador de circuitos, contendo as informações dos componentes que integram o circuito bem como as relações de interligação.

Nó Ponto de interligação de três ou mais componentes.

Ramo Conjunto de um ou mais componentes interligados em série first.

String Sequência ordenada de caracteres.

Super-Nó Conjuntos de nós interligados apenas por fontes isoladas de tensão.

Teorema da Sobreposição Consiste na análise separada do mesmo circuito para cada fonte de tensão existente no circuito, onde as correntes nos ramos do circuito são obtidas pela soma algébrica das várias correntes nos ramos para cada análise.

1 Introdução

Um dos principais fundamentos de Engenharia Elétrica/Eletrotécnica é a resolução/análise de circuitos elétricos, isto é, dado um conjunto de componentes e as relações de ligação entre eles, ter a capacidade de descobrir as grandezas físicas *Corrente* e *Tensão* elétricas a que cada componente está sujeito [24]. Exposto de outra forma, analisar um circuito elétrico é determinar as *Correntes nos Ramos* e as *Tensões nos Nós* de um dos seus esquemas elétricos. Uma vez que é possível obter as tensões nos Nós a partir das correntes nos ramos, ao analisar o circuito, basta determinar apenas as correntes nos ramos [24].

Ter habilidade de resolver circuitos de forma manuscrita é de extrema importância no desenvolvimento de um engenheiro, na medida em que imprime uma forma de ver o circuito mais natural e instintiva [1], e permite perceber que interações físicas e mecanismos estão presentes e o fazem ter o comportamento que tem.

1.1 Contextualização

A aplicação surge da necessidade de acompanhamento autónomo passo a passo na resolução de circuitos elétricos aos estudantes de Engenharia Eletrotécnica e de Computadores. As ferramentas já existentes ou são apenas simuladores de circuitos e ajudam somente na confirmação de resultados, ou se são ferramentas passo a passo são mais úteis no âmbito da matemática do que propriamente de resolução de circuitos.

1.2 Definição de Objetivos

O propósito global deste projeto prende-se na integração de mais um método, o *Método das Correntes nas Malhas (MCM)* na aplicação *U=RI solve* já existente [23]. Para isso será necessário fazer um estudo inicial do “U=RI solve” de modo a entender o seu funcionamento e arquitetura, bem como as tecnologias nele usadas, tais como as linguagens de programação “JavaScript”, HTML e CSS. De seguida será delineado todo o trabalho referente à implementação do algoritmo em si. Este inclui o cálculo do número de malhas necessário, a computação das malhas existentes no circuito, a escolha das melhores malhas para escrever equações, a escrita e resolução das equações, cálculo das correntes nos ramos e por fim os diferentes tipos de *output*.

Disponibiliza-se [aqui](#) o link utilizado nos testes da aplicação, contendo a versão mais recente.

1.3 Descrição da Abordagem e Tarefas

De modo a ter uma organização mais eficiente do trabalho, as tarefas descritas no ponto anterior foram ainda calendarizadas no espaço de tempo disponível, obtendo-se assim um ponto de vista global acerca do encaminhamento do projeto.

Objetivos / Semana	1	2	3	4	5	6	7	8	9	10
Ambientação a JavaScript										
Estudo do Projeto Existente										
Calcular Número de Malhas										
Escolha de C malhas auxiliares										
Escolher M malhas principais										
Escrever equações										

Objetivos / Semana	8	9	10	11	12	13	14	15	16	17
Calcular correntes										
Output										
Redação do Relatório										

Tabela 1: Calendário do projeto

1.4 Contribuições

Este projeto tem como maior foco de contribuição a área de Engenharia Eletrotécnica, nomeadamente a comunidade aprendiz da mesma. Destina-se, por isso, aos estudantes e auto-didatas e serve como complemento à aprendizagem, proporcionando não só os resultados de uma análise de um circuito, mas também quais os passos para lá chegar, obtendo assim um carácter pedagógico e educativo.

1.5 Organização do Relatório

Este relatório encontra-se estruturado em seis capítulos cuja visão global permite dar a entender ao leitor o funcionamento base da aplicação em questão bem como a implementação de um novo método de análise na mesma.

No primeiro, segundo e terceiro capítulos são apresentados, respetivamente, os agradecimentos, o resumo e o “abstract” do projeto.

O quarto capítulo consiste numa parte introdutória que contextualiza o leitor na aplicação e descreve em concreto os objetivos a serem implementados neste relatório.

O quinto capítulo oferece a explicação de vários conceitos teóricos de máxima relevância para o projeto tais como o conceito de “Ramo” e “Malha” e “Netlist”. É também feita neste capítulo a explicação detalhada dos principais métodos de análise, incluindo o *Método das Correntes nas Malhas (MCM)* cuja implementação é o objetivo principal do projeto.

No sexto capítulo é feita a análise de várias ferramentas relevantes no âmbito da engenharia, nomeadamente as que possuem um carácter mais didático. Estas são ainda comparadas com a aplicação “U=RIssolve”.

No sétimo capítulo é feita a análise de um circuito exemplo utilizando o *Método das Correntes nas Malhas (MCM)* de forma a consolidar os conceitos expostos no capítulo quinto, bem como oferecer um modelo exemplo do que a aplicação terá de ser capaz de executar.

O oitavo capítulo refere-se ao trabalho desenvolvido, isto é, à implementação em si do método na aplicação. Encontra-se dividido em vários subcapítulos que descrevem os vários passos realizados no decorrer da elaboração do projeto.

No nono capítulo são ainda apresentadas algumas funções desenvolvidas, úteis ao projeto, nomeadamente de que forma é feito o desenho das malhas do circuito e como é feito o arredondamento dos resultados.

Por fim, no décimo capítulo são apresentados dois exemplos reais de *output* da aplicação: um em corrente contínua e outro em corrente alternada.

2 Terminologia e Conceitos Fundamentais

2.1 Terminologia

Existem vários tipos de ligação num circuito elétrico, cujos conceitos serão de extrema importância no decorrer do projeto:

- *Ramo* - Define-se *Ramo* como sendo um conjunto de um ou mais componentes ligados em série [10][12].

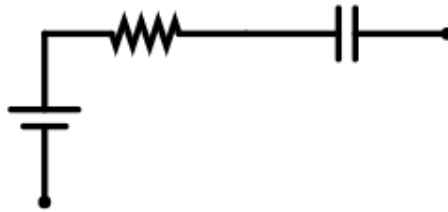


Figura 1: Exemplo de um ramo com três componentes

- *Nó* - Define-se *Nó* como sendo o ponto de interligação de três ou mais *Ramos* [10][12].

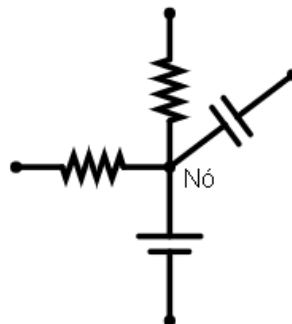


Figura 2: Exemplo de um Nó com quatro ramos interligados

- *Malha* - Define-se *Malha* como sendo um circuito elétrico fechado, composto por *Ramos*, que são delimitados por *Nós*, sem repetição nem de ramos nem de Nós. De outra forma, e utilizando teoria dos grafos (utilizada mais a frente), uma malha é um “loop” do tipo simples [10][12]. Salienta-se a distinção entre malhas simples e complexas, em que as simples não contêm ramos internos, enquanto que as complexas contêm (Na figura 1 e 3 são malhas simples e 2 e 4 complexas).

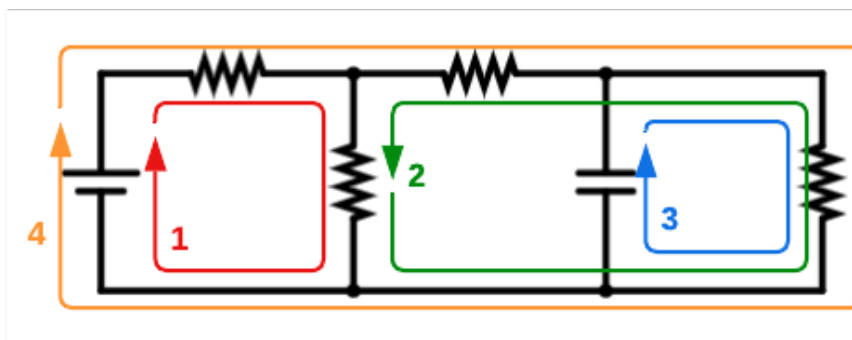


Figura 3: Exemplo de malhas num circuito

Salienta-se ainda um exemplo de Netlist na Listagem 1 referente ao circuito do “Exemplo DC” constante no fim deste relatório:

```

1  # Qucs 0.0.19 C:/PathToTheFile/Ex3.sch
2
3  Vdc:V3 _net4 D U="1 V"
4  Idc:I1 B A I="2 A"
5  R:R1 A B R="5 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
6  Vdc:V1 A _net0 U="3 V"
7  R:R2 gnd _net0 R="2 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
8  Vdc:V2 _net1 B U="4 V"
9  R:R3 gnd _net1 R="3 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
10 R:R4 gnd B R="1 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
11 Idc:I2 B _net2 I="5 A"
12 R:R5 _net2 C R="2 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
13 R:R7 C _net4 R="6 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
14 R:R6 _net3 C R="7 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
15 Idc:I3 _net3 gnd I="1 A"
16 R:R8 gnd _net5 R="2 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
17 Vdc:V4 D _net5 U="2 V"
18 R:R9 gnd _net6 R="2 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
19 Idc:I4 _net7 _net6 I="3 A"
20 R:R10 _net7 D R="4 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
21 .DC:DC1 Temp="26.85" reltol="0.001" abstol="1 pA" vntol="1 uV" save0Ps="no"
    ↪ MaxIter="150" saveAll="no" convHelper="none" Solver="CroutLU"

```

Listing 1: Estrutura do ficheiro JSON

2.2 Métodos gerais de análise de circuitos elétricos

Os Método de Análise mais usualmente ensinados/utilizados para analisar circuitos com qualquer dimensão e complexidade, incluindo circuitos com mais do que uma fonte de alimentação (tensão e/ou corrente) são o *Método das Correntes nos Ramos*, *Método da Tensão nos Nós*, o *Método das Correntes nas Malhas* e o método segundo o *Teorema da Sobreposição*. Todos eles se baseiam nos princípios gerais da *Lei de Ohm* e das *Leis de Kirchhoff* (dos Nós e das Malhas), tendo cada um o seu algoritmo, com as suas particularidades.

Estes métodos serão apresentados de seguida, bem como um exemplo de aplicação simples para ilustrar o funcionamento de cada um (circuito da figura

4).

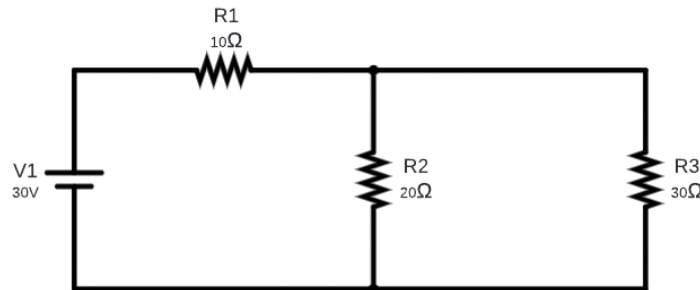


Figura 4: Circuito Exemplo a ser analisado por cada método

2.2.1 Método da Corrente nos Ramos (MCR)

Este método consiste na aplicação direta das *Leis de Kirchhoff* e por esse motivo é o método que é mais trabalhoso no que toca à matemática envolvida.

O sistema de equações reduz uma equação a cada fonte de corrente que o circuito contenha, pois é uma corrente num ramo que já é conhecida. Logo, sistema de equações terá tantas equações quantas correntes nos ramos **desconhecidas** houverem [24].

Os passos para a resolução de um circuito utilizando este método são os seguintes [21][24][14]:

1. Cálculo do número de equações necessárias, identificando as variáveis base relevantes do circuito: **Ramos (R)**, **Nós (N)**, e **Fontes de Corrente (C)**:
 - $(N - 1)$ Equações dos Nós
 - $M = R - (N - 1) - C$ Equações das malhas
2. Escolher um dos Nós para ser tomado como referência, isto é, será o Nó cuja Lei não será incluída. Escrever as equações dos Nós restantes de acordo com a *Lei de Kirchhoff dos Nós*;
3. Escolher as M malhas definindo um sentido de circulação, tomando em atenção não incluir nenhuma que contenha uma fonte de corrente;
4. Escrever as equações as malhas escolhidas de acordo com a *Lei de Kirchhoff das Malhas*;
5. Resolver o sistema de R equações, obtendo os valores das correntes nos ramos;

Exemplo

O circuito encontra-se representado novamente de seguida, desta vez já com as marcações de correntes nos ramos e nas malhas necessárias.

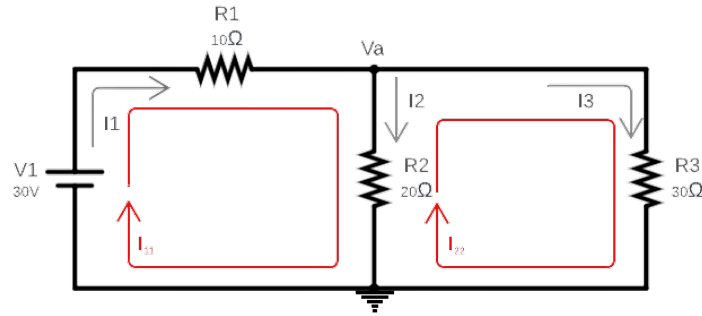


Figura 5: Circuito com as marcações necessárias ao MCR

- $R = 3; N = 1$, logo:
 - * $(N - 1) = (2 - 1) = 1$ Equações dos Nós
 - * $M = R - (N - 1) - C \Leftrightarrow M = 3 - (2 - 1) - 0 \Leftrightarrow M = 2$ Equações das malhas
- Nó superior escolhido. Equação: $I_1 = I_2 + I_3$
- Escolhidas as duas malhas representadas a vermelho na figura 5;
- Construir as equações das malhas:

$$\begin{cases} V_1 = R_1 I_1 + R_2 I_2 \\ 0 = -R_2 I_2 + R_3 I_3 \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} 30 = 30 \cdot I_1 + 20 \cdot I_2 \\ 0 = -20 \cdot I_2 + 30 \cdot I_3 \end{cases}$$

- Juntar as equações dos nós e malhas num sistema e resolvê-lo:

$$\begin{cases} I_1 = I_2 + I_3 \\ 30 = 30 \cdot I_1 + 20 \cdot I_2 \\ 0 = -20 \cdot I_2 + 30 \cdot I_3 \end{cases} \Leftrightarrow \begin{cases} I_1 = 1,36 \text{ A} \\ I_2 = 818 \text{ mA} \\ I_3 = 545 \text{ mA} \end{cases}$$

2.2.2 Método da Tensão nos Nós (MTN)

Este método consiste em determinar os potenciais de tensão em cada Nó do circuito, relativamente a um Nó de referência (de massa, $V = 0V$). Após obter as tensões nos Nós é simples calcular as correntes nos ramos do circuito utilizando a *Lei de Ohm*.

O sistema de equações reduz uma equação a cada fonte de tensão isolada que o circuito contém, pois o potencial de um lado da fonte corresponde ao potencial do outro lado +/- o valor de tensão da fonte. É introduzido o conceito de *Super-Nó* [24], que consiste em criar uma camada de abstração em que se considera toda a fonte de tensão e os seus Nós extremidade um único Nó. Assim, a

equação resultante da aplicação da *Lei de Kirchhoff dos Nós* neste *Super-Nó* engloba todas as correntes que “entram” e “saem” dos dois Nós extremidade [24].

Os passos para resolver um circuito usando este método são os seguintes [21][24][16][14]:

1. Cálculo do número de equações necessárias (S), identificando as variáveis base relevantes do circuito: **Nós (N)**, e **Fontes de Tensão isoladas (T)**, e aplicando a expressão:

$$S = N - 1 - T$$

2. Se existirem Nós interligados por apenas fontes de tensão isolada, formar um *Super-Nó*;
3. Escolher (aterrar) um Nó do circuito que servirá como referência para os potenciais de tensão dos restantes Nós do circuito. Seguir as seguintes regras:

Se $T = 0$, selecionar um Nó qualquer (sem obedecer a nenhum critério);

Se $T = 1$, escolher um dos dois Nós ligados à fonte de tensão isolada;

Se $T \geq 1$, escolher um Nó do super-Nó com maior número de Nós.

Se estas regras forem seguidas garante-se uma solução mais simples, pois impõe-se um máximo número de Nós cujo potencial é automaticamente conhecido.

4. Para cada Fonte Isolada de Tensão em cada *Super-Nó* escrever a expressão que a relaciona com os restantes Nós desse *Super-Nó* com um Nó que é escolhido como referência. Caso o *Super-Nó* seja aterrado, escolher o Nó da massa como referência, caso contrário escolher um Nó qualquer;
5. Arbitrar sentidos para as correntes nos ramos;
6. Construir as S equações recorrendo à *Lei de Kirchhoff dos Nós*;
7. Para cada corrente nas equações, substituí-la pela relação entre a resistência total desse ramo e a sua queda de tensão, recorrendo à *Lei de Ohm*:

$$I = \frac{U}{R}$$

exceto os ramos com fontes de corrente, visto que a corrente nesse ramo já é conhecida;

8. Resolver o sistema, encontrando as tensões nos Nós do circuito;
9. Usando as mesmas relações computadas no ponto 7 determinar as correntes nos ramos do circuito.

Exemplo

Neste caso não existem fontes de tensão isoladas, logo é um circuito de resolução mais simples.

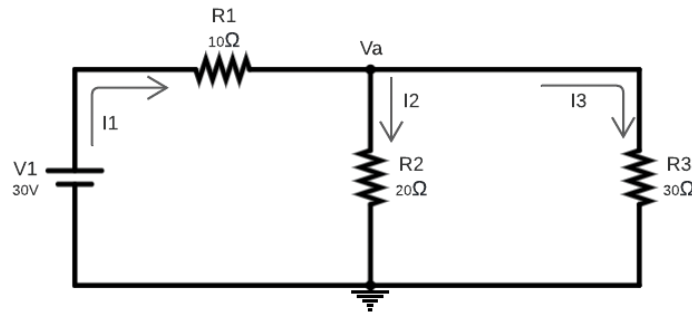


Figura 6: Circuito a ser analisado usando o MTN

- Número de equações necessárias (S):

$$S = N - 1 - T \Leftrightarrow S = 2 - 1 - 0$$

- Nó inferior escolhido como referência
- Para cada Fonte Isolada de Tensão em cada
- Sentido das correntes representadas na figura 6
- Equação do nó superior: $I_1 = I_2 + I_3$
- Equação com as substituições utilizando a *Lei de Ohm*:

$$\frac{V_1 - V_{N1}}{R_1} = \frac{V_{N1}}{R_2} + \frac{V_{N1}}{R_3}$$

- Resolver a equação:

$$V_{N1} = 16,36 \text{ V}$$

- Calcular as correntes nos ramos:

$$\begin{cases} I_1 = \frac{V_1 - V_{N1}}{R_1} \\ I_2 = \frac{V_{N1}}{R_2} \\ I_3 = \frac{V_{N1}}{R_3} \end{cases} \Leftrightarrow \begin{cases} I_1 = \frac{30 - 16,36}{10} \\ I_2 = \frac{16,36}{20} \\ I_3 = \frac{16,36}{30} \end{cases} \Leftrightarrow \begin{cases} I_1 = 1,36 \text{ A} \\ I_2 = 818 \text{ mA} \\ I_3 = 545 \text{ mA} \end{cases}$$

2.2.3 Método da Corrente nas Malhas (MCM)

Este método consiste em considerar correntes fictícias, que circulam nas malhas do circuito e que portanto não possuem significado real. Deste conceito é possível extrair um sistema de equações cujo resultado é o valor das correntes de malha. Resta apenas calcular os valores das correntes nos ramos a partir da soma algébrica das correntes nas malhas.

O sistema de equações reduz uma equação a cada fonte de corrente existente no circuito, pois é mais uma malha auxiliar que será escolhida, que é uma malha cujo valor de corrente é conhecido logo não é preciso construir a sua equação [24].

De forma mais concisa e algorítmica, os passos para a resolução de um circuito utilizando este método são os seguintes [21][24][14]:

1. Cálculo do número de equações necessárias, identificando as variáveis base relevantes do circuito: **Ramos (R)**, **Nós (N)**, e **Fontes de Corrente (C)**, e aplicando a expressão:

$$M_p = R - (N - 1) - C$$

2. Selecionar e marcar um número C de malhas auxiliares, cada uma delas incluindo uma e uma só fonte de corrente;
3. Selecionar e marcar um número M de malhas principais, linearmente independentes, e que não contenham nenhuma fonte de corrente ($I_{M1}, I_{M2}, \dots, I_{MM}$). Cada malha escolhida deve seguir duas regras:

- (a) Ter pelo menos um ramo em comum com outra malha;
- (b) Ter pelo menos um ramo que não pertença a outra malha;

Desta forma garante-se que o sistema é linearmente independente e que são incluídos todos os ramos;

4. Construir as M equações de malha, com base na *Lei de Kirchhoff das malhas*;
5. Resolver o sistema, encontrando os valores de $I_{M1}, I_{M2}, \dots, I_{MM}$;
6. Calcular as correntes nos ramos ($I_1, I_2, \dots, I_R$) através da soma algébrica das correntes das malhas que incluem cada ramo. Cada parcela nesta soma deve ter o sinal de acordo com o sentido relativo entre a corrente no ramo e a corrente de malha em questão: sentidos contrários sinal negativo e vice-versa.

Exemplo

Irá ser realizada uma análise de um circuito mais complexo, incluindo uma Fonte Ideal de Corrente, num capítulo posterior. No entanto do mesmo modo que os outros métodos apresentados, consta aqui a resolução do mesmo circuito exemplo utilizando o *MCM*. A figura seguinte corresponde ao mesmo circuito da figura 4, mas com todas as marcações referentes ao *Método das Correntes nas Malhas (MCM)* incluídas. Segue-se a resolução.

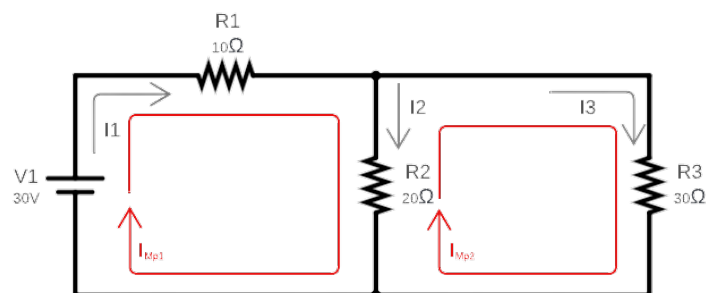


Figura 7: Circuito com as marcações necessárias ao MCM

1. Número de equações principais necessárias:

$$M_p = R - (N - 1) - C \Leftrightarrow M_p = 3 - (2 - 1) - 0 \Leftrightarrow M_p = 2$$

2. Malhas principais assinaladas na figura 7
3. Sistema de equações construído com base nas malhas escolhidas:

$$\begin{aligned} & \begin{cases} V_1 = R_1 I_{Mp1} + R_2 (I_{Mp1} - I_{Mp2}) \\ 0 = R_2 (I_{Mp2} - I_{Mp1} + R_3 I_{Mp2}) \end{cases} \Leftrightarrow \\ & \Leftrightarrow \begin{cases} 10 = 10 \cdot I_{Mp1} + 20 \cdot (I_{Mp1} - I_{Mp2}) \\ 0 = 20 \cdot (I_{Mp2} - I_{Mp1}) + 30 \cdot I_{Mp2} \end{cases} \end{aligned}$$

4. Resolver o sistema:

$$\begin{cases} I_{Mp1} = 1,36 \text{ A} \\ I_{Mp2} = 545 \text{ mA} \end{cases}$$

5. Calcular as correntes nos ramos através das correntes de malha:

$$\begin{cases} I_1 = I_{Mp1} \\ I_2 = I_{Mp2} - I_{Mp1} \\ I_3 = I_{Mp2} \end{cases} \Leftrightarrow \begin{cases} I_1 = 1,36 \text{ A} \\ I_2 = 818 \text{ mA} \\ I_3 = 545 \text{ mA} \end{cases}$$

3 Análise de Tecnologias relacionadas

Foram selecionadas algumas tecnologias cujo objetivo, para além de fornecer o resultado, passa também por proporcionar mais informação que ajuda na compreensão dos vários conceitos envolvidos. Estas ferramentas são: o *Symbolab Math Solver*, o *Microsoft Math Solver*, o *WolframAlfa* o *Falstad* e o *PhotoMath*.

Os três primeiros apresentados são mais voltados à matemática, produzindo resoluções passo a passo para diversos problemas. Já o último é um simulador de circuitos elétricos com características visuais vantajosas à aprendizagem.

3.1 *Symbolab*

O Symbolab [2] é uma ferramenta de resolução passo-a-passo de expressões matemáticas introduzidas pelo utilizador. Existe uma versão grátis que permite resolver praticamente qualquer tipo de expressão, praticar problemas por tópicos e oferece os passos principais de resolução [4]. A versão paga, com vários tipos de subscrição, inclui também passos intermédios de resolução, *quizzes*, registo de progresso de aprendizagem e problemas personalizados. Tal como ilustra a figura 8, a interface gráfica de utilização é intuitiva e a aplicação completa no que toca a operadores matemáticos [4].

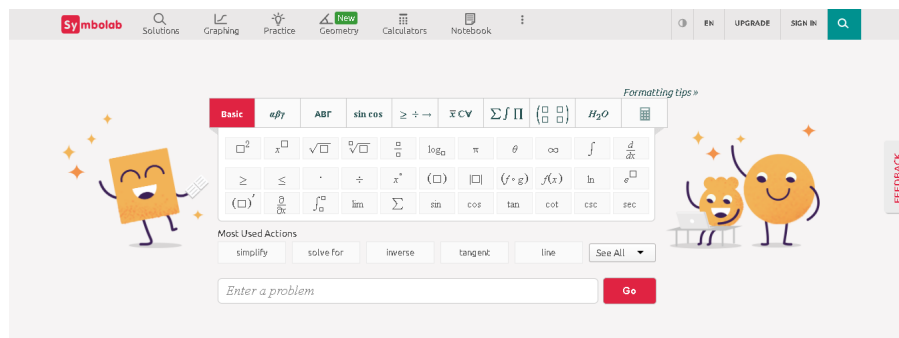


Figura 8: Interface do Symbolab

3.2 *Microsoft Math Solver*

O Microsoft Math Solver [13] é uma ferramenta de resolução de expressões matemáticas, da *Microsoft*, que se destaca por permitir introduzir uma expressão utilizando uma ferramenta ao estilo duma caixa de seleção em qualquer parte do browser [8], tornando o processo de submissão mais rápido e eficiente.

É uma ferramenta bastante completa no que toca a símbolos matemáticos e a interface, em apresentação na Figura 9 favorece a aprendizagem.

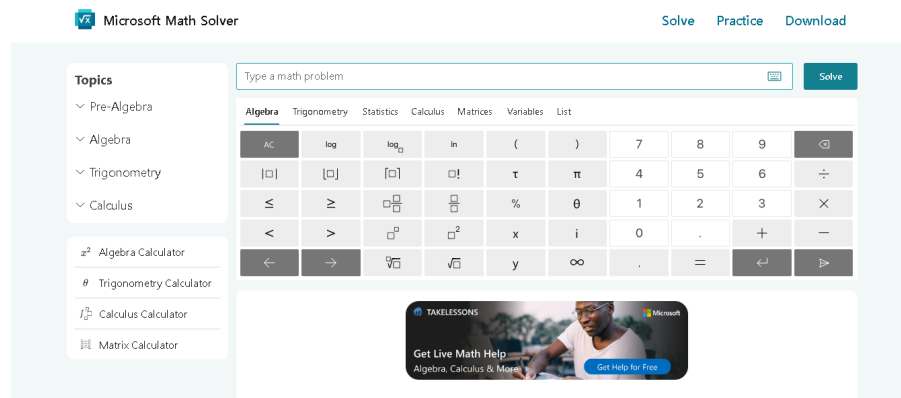


Figura 9: Microsoft Math Solver Interface

3.3 WolframAlpha

O WolframAlpha [25] é uma ferramenta extremamente completa, não só para a engenharia mas para virtualmente tudo o que possa envolver cálculos ou computação de algum género. No que toca a engenharia e matemática, é também uma excelente ferramenta com capacidades de resolução passo-a-passo, cálculo de grandezas específicas (Lei de Ohm, paralelo de resistências ou frequência de ressonância, por exemplo). Existe um modo para inserção de caracteres matemáticos semelhante ao [Symbolab](#) e uma linguagem natural já característica da ferramenta, que pode ser desafiante para aprender de raiz. Contudo, a interface de utilização é simples, apelativa e encontra-se representada na Figura 10.

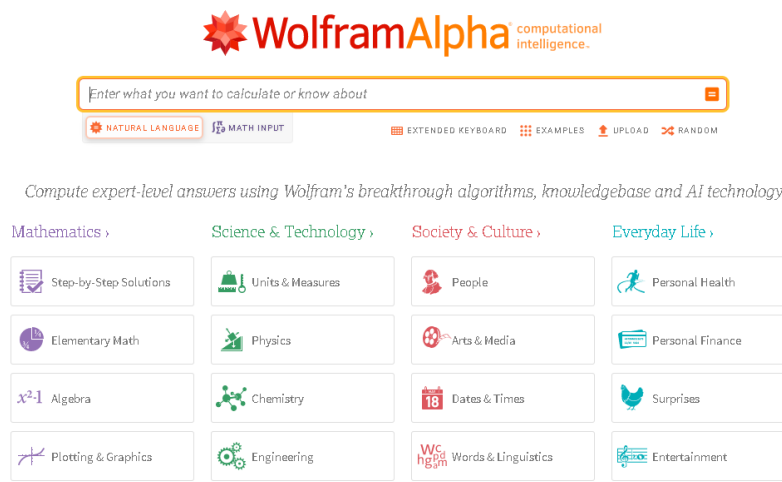


Figura 10: Interface do WolframAlfa

3.4 *Falstad*

O Simulador de circuitos do Falstad [7] não tem uma vertente tão didática, no que toca a explicar os conceitos passo-a-passo. No entanto é um simulador com algumas particularidades gráficas que ajudam na compreensão dos fenómenos físicos associados a circuitos elétricos. Tomando um simples divisor de corrente como exemplo (Figura 11): é possível observar o fluxo da corrente pelos ramos do circuito (pontos amarelos) e perceber que por unidade de tempo o fluxo de eletrões na resistência menor é maior. Além da corrente é possível visualizar o potencial em cada ponto do circuito pelo gradiente de cores de vermelho a verde (potencial negativo a potencial positivo), e a potência através do mesmo gradiente de cores, agora salientando que componentes estão a dissipar ou fornecer energia, por unidade de tempo.

É uma ferramenta bastante simples de usar e interativa ao ponto de ser possível editar o circuito sem parar a simulação, possibilitando a visualização instantânea da alteração dos valores de corrente nos ramos, por exemplo.

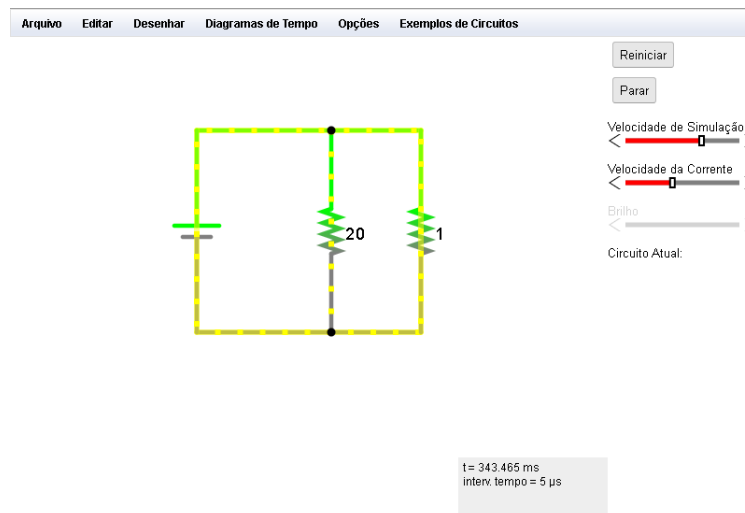


Figura 11: Interface do Simulador de circuitos do Falstad

3.5 *PhotoMath*

Em contraste com as ferramentas apresentadas até agora, o PhotoMath [22] é uma aplicação para dispositivos móveis, com suporte “Android” e “iOS”. Esta utiliza a câmara do dispositivo para reconhecer expressões matemáticas (mesmo em papel e manuscritas) [9]. Fornece todos os passos de maneira intuitiva e clara, como é possível inferir da Figura 12 e é totalmente gratuito.

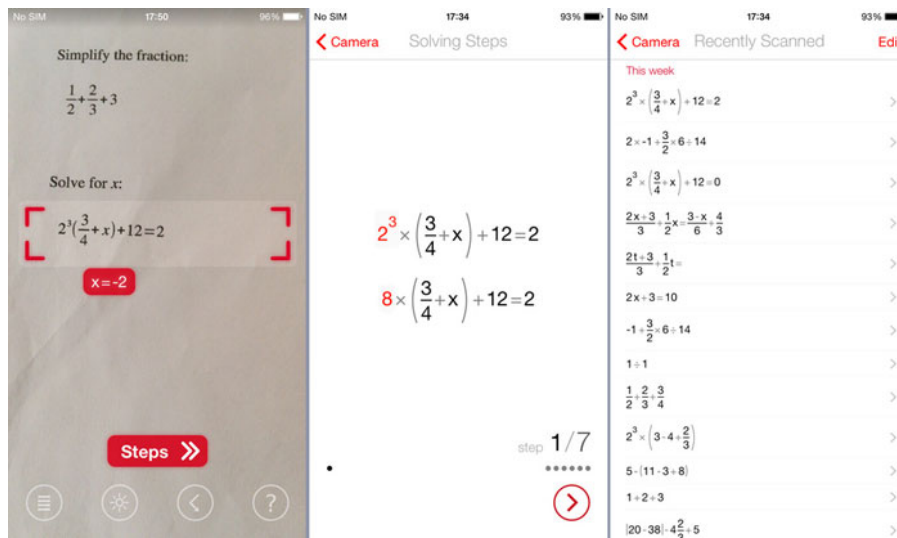


Figura 12: Interface do PhotoMath

3.6 Comparação com a ferramenta *U=RSolve*

Foram criados os seguintes critérios que tentam classificar as aplicações apresentadas de forma mais objetiva e quantitativa: interface, qualidade passo a passo, preço, plataforma, se o passo a passo é relacionado com circuitos elétricos (Circ. Elét.). Estes encontram-se na tabela 2 onde são comparadas as várias ferramentas, incluindo o *U=RSolve*, atribuindo uma classificação de 1 a 5 estrelas, nos pontos qualitativos.

	Interface	Passo a Passo	Custo anual	Plataforma	PassoaPasso Circ. Elét.
Symbolab	★★★★★	★★★★	30USD	Browser	Não é
Microsoft Math Solver	★★★★	★★★★★	Grátis	Browser	Não é
WolframAlfa Pro Student	★★★★★	★★★★★	63USD	Browser	Não é
Falstad	★★★★	★	Grátis	Browser	Não tem
PhotoMath	★★★★★	★★★★★	Grátis	Mobile	Não é
U=RSolve	★★★★★	★★★★★	Grátis	Browser	É

Tabela 2: Comparação entre as ferramentas apresentadas e U=RSolve

4 Algoritmo MCM

4.1 Aplicação a um circuito-exemplo (contendo uma fonte ideal de corrente)

De seguida descreve-se o *Método das Correntes nas Malhas* (MCM) [15], que é o alvo de implementação na aplicação U=RIssolve, no contexto deste projeto.

No seguinte exemplo é posto em prática o algoritmo em questão [21][14]:

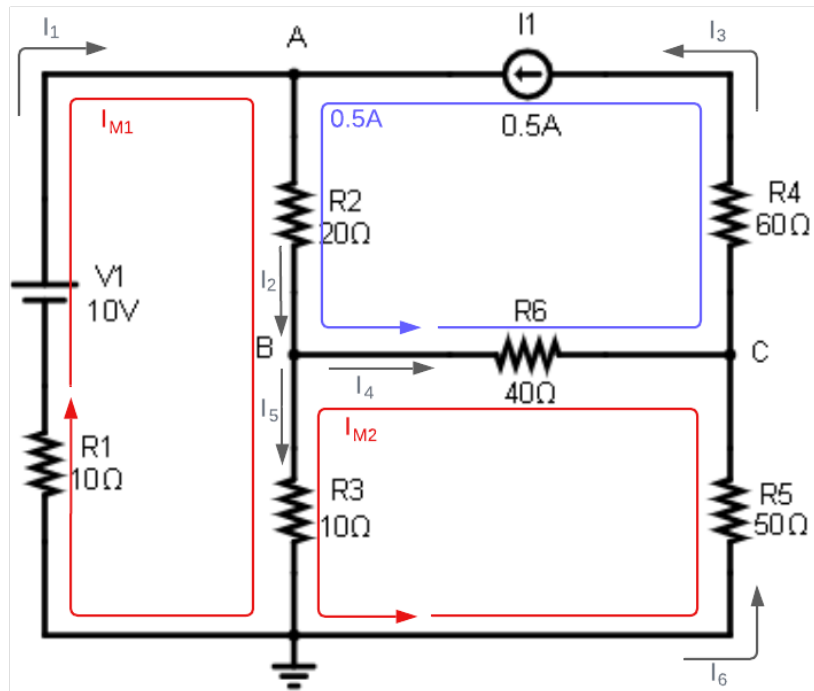


Figura 13: Circuito Exemplo

1. Este circuito é composto por 6 Ramos, 4 Nós e 1 Fonte Ideal de Corrente, isto é, $R = 6$, $N = 4$ e $C = 1$. Aplicando a expressão vem:

$$\begin{aligned} M_p &= R - (N - 1) - C \Leftrightarrow \\ \Leftrightarrow M_p &= 6 - (4 - 1) - 1 \Leftrightarrow \\ \Leftrightarrow M_p &= 2 \end{aligned}$$

2. Escolher uma malha auxiliar, que contenha a fonte de corrente. A malha escolhida encontra-se a azul marcada na Figura 13.
3. Escolher duas malhas principais, que sigam as regras estabelecidas. As malhas escolhidas encontram-se a vermelho marcadas na Figura 13.

4. Construir o sistema de equações:

$$\begin{cases} V_1 = R_1 I_{M1} + R_2(I_{M1} + 0,5) + R_3(I_{M1} + I_{M2}) \\ 0 = R_3(I_{M2} + I_{M1}) + R_6(I_{M2} - 0,5) + R_5 I_{M2} \end{cases} \Leftrightarrow \begin{cases} 10 = 10 \cdot I_{M1} + 20 \cdot (I_{M1} + 0,5) + 10 \cdot (I_{M1} + I_{M2}) \\ 0 = 10 \cdot (I_{M2} + I_{M1}) + 40 \cdot (I_{M2} - 0,5) + 50 \cdot I_{M2} \end{cases} \quad (1)$$

5. Resolver o sistema:

$$\begin{cases} I_{M1} = 0 \text{ A} \\ I_{M2} = 200 \text{ mA} \end{cases} \quad (2)$$

6. Após marcar as correntes nos Ramos (marcações presentes na Figura 13), arbitrando os seus sentidos, resta apenas inferir as suas equações em função das correntes de Malha, e calcular:

$$\begin{cases} I_1 = I_{M1} \\ I_2 = I_{M1} + 0.5 \\ I_3 = 0.5 \text{ A} \\ I_4 = -I_{M2} + 0.5 \\ I_5 = I_{M1} + I_{M2} \\ I_6 = I_{M2} \end{cases} \Leftrightarrow \begin{cases} I_1 = 0 \text{ A} \\ I_2 = 500 \text{ mA} \\ I_3 = 500 \text{ mA} \\ I_4 = 300 \text{ mA} \\ I_5 = 200 \text{ mA} \\ I_6 = 200 \text{ mA} \end{cases} \quad (3)$$

4.2 Enquadramento Prático

O plano para o desenvolvimento do projeto toma como base a utilização parcial do código do *Método da Tensão nos Nós (MTN)* já implementado, nomeadamente de porções de código comum a todos os métodos, que deveriam fazer parte de uma função que pode ser chamada por qualquer um deles. A saída deste código será um ficheiro JSON com as informações da “Netlist” já extraídas, validadas e trabalhadas, prontas a serem analisadas.

Segue-se o desenvolvimento do algoritmo do método em si, que envolve toda a computação desde o cálculo do número de equações necessárias até serem obtidas as correntes nos ramos do circuito.

Todas estas novas informações serão posteriormente atualizadas no ficheiro JavaScript Object Notation (JSON) previamente criado, num campo denominado “analysisObj”. Assim, o ficheiro atualizado com todas as novas informações de análise é utilizado posteriormente para a produção dos diversos tipos de “outputs”, garantindo que todos eles apresentam a mesma informação.

Na figura 14 encontram-se representados vários blocos da arquitetura a implementar, fornecendo também a informação de quais blocos foram desenvolvidos *vs* quais blocos já existiam:

1. “Código MTN” $\Rightarrow$ Referente ao código inicial já existente relativo ao algoritmo MTN.

2. “Código Comum” $\Rightarrow$ Referente à porção de código, presente no “Código MTN” que é transversal a todos os métodos e por isso faz mais sentido do ponto de vista funcional ser desagregado do algoritmo de cada método.
3. “Encapsulamento em Funções” $\Rightarrow$ Referente ao ato de alterar a arquitetura do “Código Comum” de texto corrido para várias funções com objetivos segmentados. Salientam-se os exemplos de funções: “LoadFile()”, “ValidateNetlist()”, etc.
4. “Ficheiro JSON” $\Rightarrow$ Referente à criação de um ficheiro de dados do tipo JSON contendo todas as informações do modelo do circuito construídas e trabalhadas até este ponto, isto é, componentes do circuito, ramos, nós, informações das correntes nos ramos, etc.
5. “Algoritmo MCM” $\Rightarrow$ Referente ao algoritmo em questão, isto é, às computações necessárias realizar com o objetivo de calcular o valor das correntes nos ramos do circuito.
6. “Dados Resultado” $\Rightarrow$ Referente aos dados relevantes recolhidos durante a análise do circuito.
7. “Ficheiro JSON Atualizado” $\Rightarrow$ Referente ao mesmo ficheiro JSON anterior, agora fundido com as informações recolhidas durante a análise, isto é, os “Dados Resultado”
8. “Output” $\Rightarrow$ Referente às várias possibilidades de saída de informação para o utilizador. Inclui a informação apresentada imediatamente no ecrã, transferência de PDF, transferência do ficheiro TeX, abrir diretamente no *Overleaf* e imprimir.

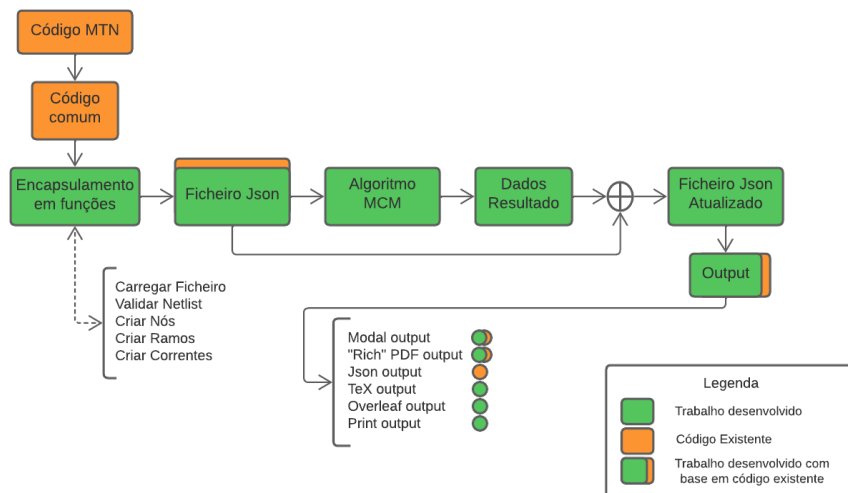


Figura 14: Diagrama de blocos ilustrativo do plano/arquitetura do projeto

Este formato, que é ilustrado na figura 14 garante que todas as computações são realizadas antes de ser feita a saída da informação para o utilizador.

4.3 Encapsulamento em funções de código comum

O código relativo ao *Método da Tensão nos Nós (MTN)* [23] consistia apenas numa função principal, que tratava de carregar o ficheiro, validá-lo e importar os dados para variáveis locais (embora houvesse a chamada a algumas sub-funções, por exemplo “validateNetlist()”). Esta arquitetura não é a melhor do ponto de vista global da aplicação pois estes passos são comuns a todos os métodos e não faz sentido repetir código em todos eles. Do ponto de vista funcional é muito mais prático existir uma função comum que é chamada no início de qualquer análise, e apenas após o seu retorno, o código divergir para a análise de cada método.

A primeira etapa do projeto consistiu na melhoria de algum código existente, relativo ao *Método da Tensão nos Nós (MTN)* [23]: com a adição de um novo método, paralelo ao existente, existem blocos de código que são agora comuns. Por este motivo, tendo em foco a eficiência global da aplicação e a utilização de um mínimo de recursos computacionais, o código comum será “encapsulado” em funções comuns e executado previamente ao algoritmo de cada método.

O primeiro passo foi identificar o código em questão e dividi-lo em vários trechos, cada um com um propósito mais objetivo em que cada um constituirá uma função com esse propósito. Chama-se a este passo o “encapsulamento” de funções.

Foi adotado um formato padrão para o objeto retorno de todas as funções, em que o primeiro parâmetro é um “booleano”, sinalizando se houve erro ou não, o segundo é o código de erro (um número), e o terceiro contém os dados de retorno caso não existam erros ou mais informações sobre os erros caso existam.

Salientam-se as funções “encapsuladas” e o seu papel no algoritmo:

- “loadFile()” - Carregamento do ficheiro: trata de validar a “Netlist”, e constrói e apresenta no output os erros caso existam. Caso o ficheiro seja válido retorna a “Netlist” acompanhada de avisos caso existam;
- “importData()” - Importa os dados da “Netlist” para variáveis locais utilizando o método “acquireCpData()” e testando de que tipo é o componente (resistência, condensador, fonte, etc). A cada componente cria um objeto com as suas informações e adiciona-o a um *array* de componentes do mesmo tipo (*array* de resistências, condensadores, etc);
- “manageAmmeters()” - Guarda a informação de posição dos amperímetros, e dos componentes em série com eles;
- “findNodes()” - Encontra os Nós do circuito e classifica-os como Reais (tipo 0) ou Virtuais (tipo 1);
- “makeBranches()” - Cria e constrói os ramos do circuito com todos os componentes constituintes, criando um objeto correspondente para cada um;
- “branchCurrents()” - Atribui uma direção às correntes nos ramos do circuito. Começa por utilizar informação do amperímetro no ramo, caso existam, tomando o seu sentido como sentido da corrente. Caso não exista amperímetro passa a utilizar o sentido de uma possível fonte de corrente que exista no ramo. Caso não exista nem amperímetro nem fonte de corrente, o sentido da corrente no ramo é arbitrado;

- “getBranchInfo()” - Calcula vários valores relativos a cada ramo, nomeadamente a sua impedância equivalente, a sua tensão equivalente e os Nós extremidade das fontes de tensão e corrente.
- “buildJson()” - Constrói o ficheiro JSON de “output” com as várias informações recolhidas ao longo do “parsing” dos dados. Neste estágio ele apresenta o seguinte formato:

```
1      {
2          'app':{
3              'version': '',
4              'details': '',
5              'releaseDate': ''
6          },
7          'components':{
8              'resistors':[ ],
9              'coils':[ ],
10             'capacitors':[ ],
11             'dcVoltPs':[ ],
12             'dcAmpsPs':[ ],
13             'acVoltPs':[ ],
14             'acAmpsPs':[ ]
15         },
16         'probes':{
17             'ammeters':[ ],
18             'voltmeters':[ ]
19         },
20         'nodes':[ ],
21         'branches':[ ],
22         'analysisObj':{
23             'circuitFreq':{
24                 'value': ,
25                 'mult': ''
26             },
27             'currents':[ ],
28             'warnings':[ ]
29         }
30     }
```

Listing 2: Estrutura do ficheiro JSON

4.4 Implementação do Algoritmo

O algoritmo para a resolução de um circuito pelo *Método das Correntes nas Malhas (MCM)* tem objetivamente os mesmos passos e alguns adicionais para compensar a desigualdade Homem/Máquina, isto é, certas ações que são extremamente naturais para o cérebro humano realizar são mais complexas quando se passa para a visão da máquina. Um exemplo claro é a escolha das malhas para construir as equações: o ser humano apenas necessita olhar para o circuito e

escolher as malhas, enquanto que a máquina precisa de um passo intermédio de obter todas as malhas possíveis do mesmo e só depois pode dar início à escolha.

Os passos gerais do algoritmo encontram-se em representação na Figura 15:

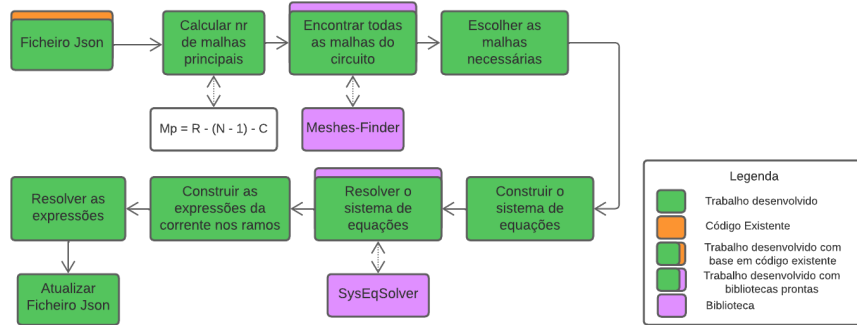


Figura 15: Diagrama de blocos do algoritmo do método da corrente nas malhas

4.4.1 Cálculo do Número de Malhas Principais

Para o cálculo do número de malhas principais (Mp) é necessário em primeiro lugar obter as variáveis fundamentais relevantes. Estas são: os ramos (R), os Nós (N) e as fontes de corrente (C). O cálculo é feito usando a seguinte expressão.

$$Mp = R - (N - 1) - C \quad (4)$$

4.4.2 Determinação das malhas do circuito

O próximo passo é determinar quais são todas as malhas do circuito, de forma a poder fazer a escolha das melhores para construir o sistema de equações.

Numa versão anterior, mais primitiva da aplicação o *Método das Correntes nas Malhas* (*MCM*) já estava implementado e a determinação das malhas do circuito era realizada através de análise combinatória, seguida da filtragem de malhas fisicamente impossíveis no âmbito de circuitos elétricos [26]. Este cálculo é exponencialmente mais complexo consoante aumenta a dimensão do circuito a analisar, consumindo mais recursos computacionais que os necessários.

Assim, a determinação das malhas existentes no circuito é feita com uma biblioteca já desenvolvida denominada *Meshes-Finder* [18] e que faz uso da teoria dos (Grafos) para analisar a rede do circuito.

A utilização correta desta biblioteca requer a criação de duas matrizes, a de adjacência e a de incidência, que descrevem as ligações existentes no circuito:

- *Matriz de adjacência* - Diz-se que dois Nós (ou vértices) são adjacentes se existe um ramo (ou aresta) que os liga.

$$\begin{matrix} & A & B & C & D \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

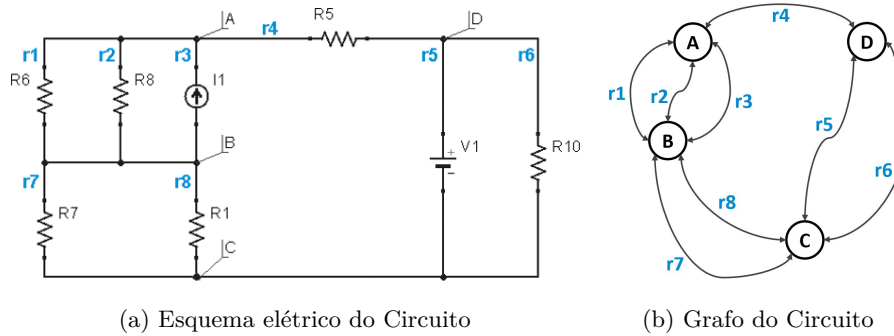


Figura 16: Exemplo de um Circuito e o Grafo correspondente

Na matriz considera-se um “1” caso exista adjacência e “0” caso não exista.

- *Matriz de incidência* - Diz-se que um ramo (aresta) é incidente a um Nó (vértice) caso esteja conectado a ele, isto é, caso esse Nós (vértice) seja uma das extremidades do ramo (aresta).

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
A	1	1	1	1	0	0	0	0
B	1	1	1	0	0	0	1	1
C	0	0	0	0	1	1	1	1
D	0	0	0	1	1	1	0	0

Na matriz considera-se um “1” caso exista incidência e “0” caso não exista.

Para obter as malhas do circuito basta chamar o método “*initGraph()*” [18], passando as matrizes criadas como argumento, bem como as referências dos Nós e dos ramos, seguido do método “*getMeshes()*” [18], que retorna um ficheiro JSON contendo todas as malhas do circuito, organizadas por ordem crescente de número de ramos (arestas).

Por último foi efetuada uma reorganização da informação resultante de forma a facilitar o próximo passo que é a escolha de malhas principais. O ficheiro resultante do *Meshes-Finder* consiste numa estrutura JSON organizada por campos cada um contendo as malhas de uma certa ordem (ver figura 3). Este formato foi alterado para um simples vetor de vetores (ver figura 4).

O próximo passo será escolher malhas principais onde será necessário percorrer as malhas várias vezes, daí ser escolhido o formato da Figura 4, pois é um formato mais simples de ser “varrido”. Este vetor será chamado “vetor das malhas” deste ponto em diante.

4.4.3 Algoritmo de escolha de malhas

A escolha de malhas é feita essencialmente em dois passos principais. Num momento inicial serão escolhidas as malhas auxiliares, e depois as principais [21]. Ocorrerão alguns processos de filtragem de modo a otimizar o algoritmo.

```

}order}: {
  0: [ ]
  1: [ ]
  2: [
    0: [1, 4]
    1: [1, 3]
    2: [5, 6]
  ]
  3: [
    0: [2, 5, 1]
    1: [2, 5, 3]
    2: [2, 6, 1]
    3: [2, 5, 3]
    4: [4, 5, 1]
    5: [4, 5, 3]
    6: [4, 6, 1]
    7: [4, 6, 3]
  ]
}

}malhas_arr}: [
  0: [1, 4]
  0: [1, 3]
  0: [5, 6]
  0: [2, 5, 1]
  0: [2, 5, 3]
  0: [2, 6, 1]
  0: [2, 6, 3]
  0: [4, 5, 1]
  0: [4, 5, 1]
  0: [4, 6, 3]
  0: [4, 6, 1]
]

```

Tabela 4: Formato vetor

Tabela 3: Formato JSON

O interesse base é escolher as malhas mais simples possíveis, e por esse motivo a estratégia adotada consiste em percorrer o “vetor das malhas” desde o início, escolhendo a primeira malha que satisfaça as condições exigidas, repetindo este procedimento até serem obtidas todas as malhas necessárias. Deste modo serão sempre escolhidas malhas com menos ramos e que á partida darão origem a um sistema e resolução do circuito mais simples. As condições exigidas, acima referidas, são as condições constantes na descrição do funcionamento do método:

1. Sistema deve conter M equações, sendo $M = R - (N - 1) - C$;
2. Cada fonte de corrente C deve pertencer a uma e uma só malha “Auxiliar”;
3. As restantes malhas (“Principais”) devem incluir todos os ramos, exceto os que contêm fontes de corrente, verificando:
 - (a) Ter pelo menos um ramo em comum com outra malha;
 - (b) Ter pelo menos um ramo que não pertença a outra malha;

No entanto do ponto de vista algorítmico não é necessário testar-se cada malha desta forma, logo, foi concebida uma análise do problema em questão com o objetivo de simplificar as condições que consiste na aprovação das regras anteriores de forma separada, usando exemplos por contradição.

Número de malhas menor que M , não cumprimento da regra 1

Veja-se o circuito seguinte e a sua seleção de malhas:

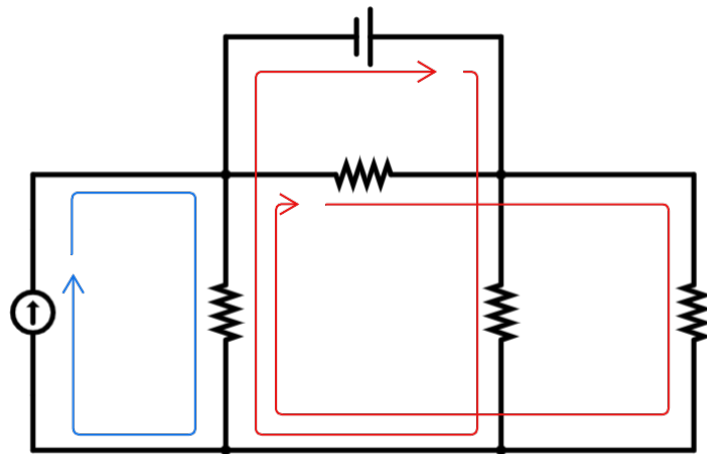


Figura 17: Seleção de malhas menor que M

Todas as regras impostas são cumpridas, exceto a nº 1, o que leva a uma resolução errada do circuito.

Usando número de Malhas Principais igual a M

Veja-se o mesmo circuito que na figura 17, agora com um número correto de malhas selecionadas:

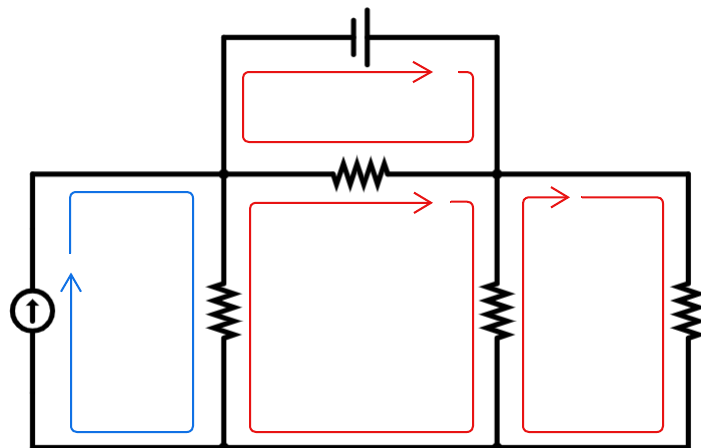


Figura 18: Seleção de malhas usando M malhas principais

Todas as regras impostas são cumpridas, exceto a alínea b do ponto 3, isto é, nem todas as malhas têm um ramo que não integra outra malha. No entanto esta escolha de malhas leva a uma resolução correta do circuito.

Seleção de malhas que não cobre todos os ramos

Veja-se o seguinte circuito, e a seleção de malhas:

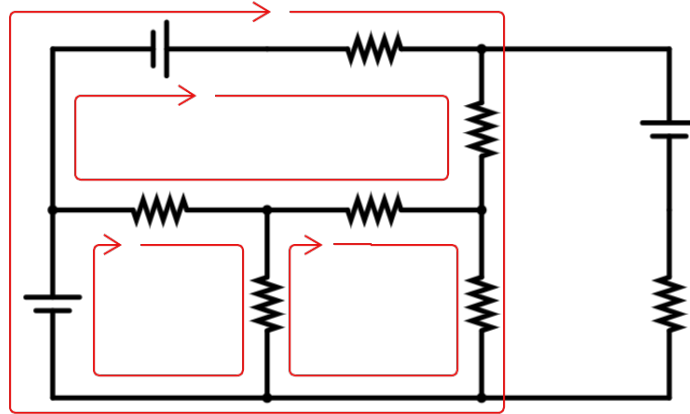


Figura 19: Seleção de malhas que não cobre todos os ramos

Todas as regras impostas são cumpridas, à exceção da inclusão de todos os ramos no sistema (sobrando o ramo à direita). Esta seleção de malhas leva a uma resolução errada do circuito, visto que o sistema não é linearmente independente.

Seleção de malhas que cobre todos os ramos

Segue-se o mesmo circuito que o da figura 19, agora com uma seleção de malhas que faz inclusão de todos os ramos:

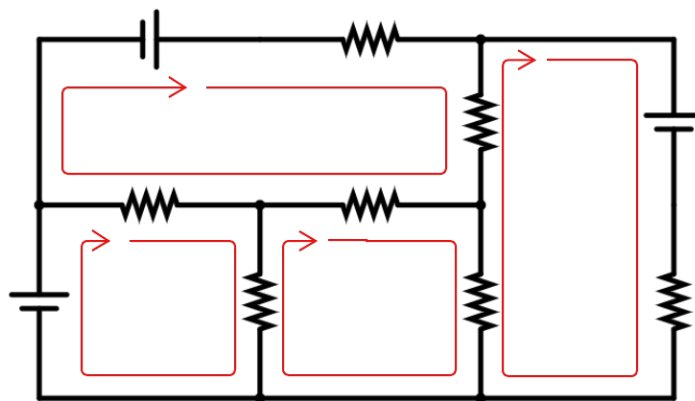


Figura 20: Seleção de malhas que cobre todos os ramos

Todas as regras impostas são cumpridas, levando a um sistema de equações

linearmente independente e a uma resolução correta do circuito.

Assim, conclui-se que o grupo final de malhas escolhidas terá que satisfazer as seguintes condições:

1. Conter um número Mp equações;
2. Cada fonte de corrente C , ou corrente conhecida, deve pertencer apenas e só a uma malha (auxiliar);
3. As restantes malhas (principais) devem cobrir todos os ramos do circuito exceto os que contêm fontes de corrente.

Foi ainda criado um vetor de “flags”, que guardam a informação do estado dos ramos: se estão ou não ocupados, isto é, incluídos em alguma malha. Esta informação será importante na escolha de Malhas Principais. Marcar um certo ramo como ocupado significa guardar um “1” no respetivo lugar do vetor correspondente a esse ramo.

4.4.3.1 Malhas Auxiliares (Ma)

O algoritmo começa com a filtragem (diagrama de blocos da figura 21) de algumas malhas que são fisicamente impossíveis de considerar uma corrente que as percorre: malhas que contêm mais que uma fonte de corrente. Estas são retiradas do “vetor das malhas”.

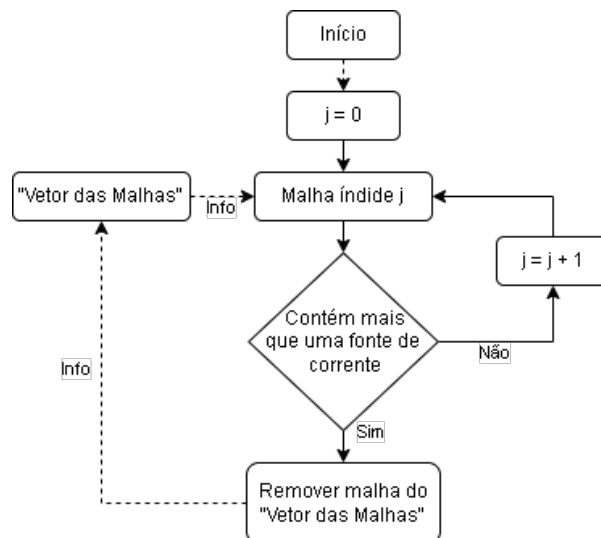


Figura 21: Fluxograma descritivo da filtragem de malhas virtualmente impossíveis

Procede-se a escolha de “Malhas Auxiliares”. Sabe-se que é necessário escolher uma malha auxiliar por cada fonte de corrente existente no circuito, e que cada fonte de corrente apenas pode pertencer a uma e uma só malha. A estratégia de resolução deste problema consiste em criar um ciclo do tipo “for”, em que a cada ciclo de funcionamento uma malha é escolhida. Este ciclo é

programado para executar tantas vezes quantas fontes de corrente houverem, garantindo assim que são escolhidas C fontes de corrente.

A cada ciclo de funcionamento o “vetor das malhas” é percorrido desde o início utilizando outro ciclo do tipo “for”. Neste ciclo, a cada malha, é testado se algum dos ramos inclui a fonte de corrente em questão. Caso inclua a malha é escolhida, isto é, é criado um objeto malha com as suas diversas informações. Caso não inclua, a malha é passada à frente sem executar nenhuma ação.

O diagrama de blocos da Figura 22 descreve visualmente o funcionamento do algoritmo até este ponto.

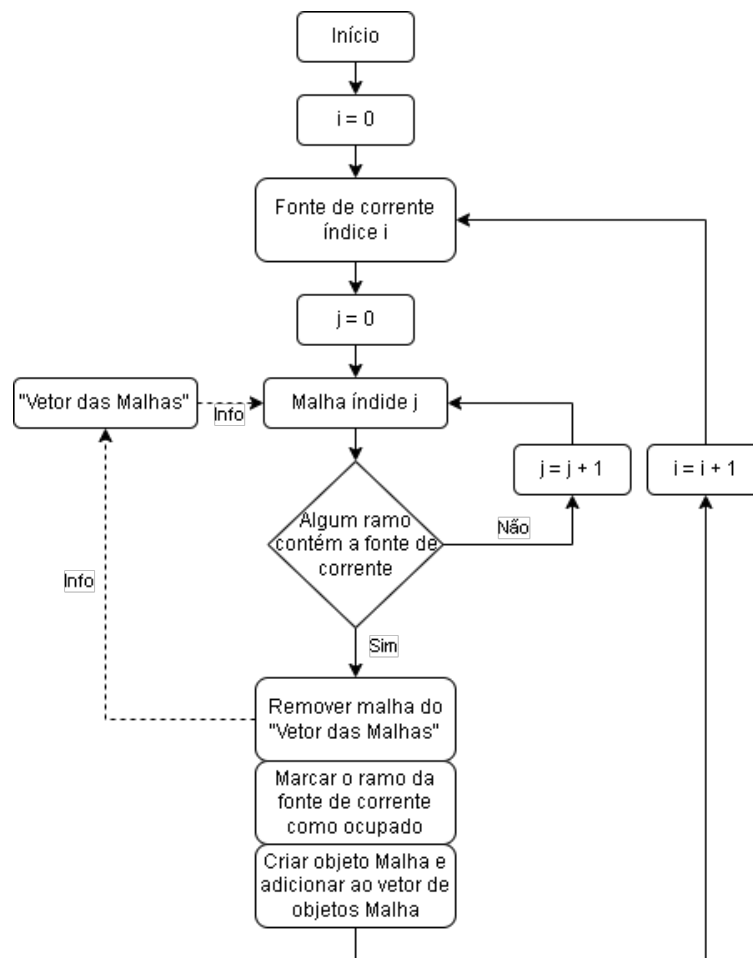


Figura 22: Fluxograma descritivo da escolha de “Malhas Auxiliares”

4.4.3.2 Malhas Principais (Mp)

A escolha de Malhas Principais implica selecionar malhas que não contêm fontes de corrente. Por este motivo é previamente feita uma filtragem intermédia, que retira do “vetor das malhas” todas as que contêm fontes de corrente (não escolhidas). Esta filtragem é semelhante à filtragem inicial e encontra-se representada pelo diagrama de blocos da figura 23.

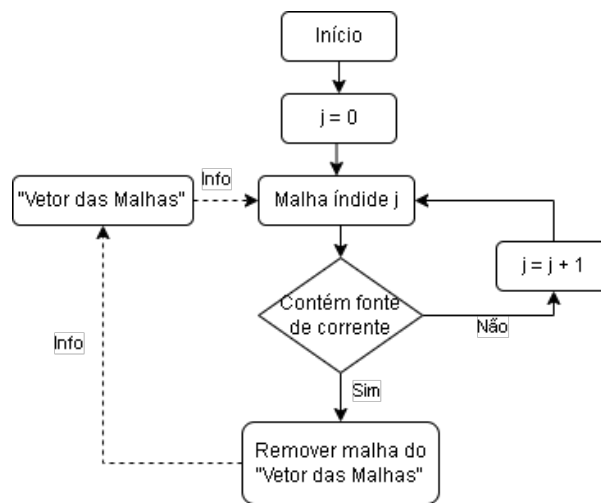


Figura 23: Fluxograma descritivo da filtragem de malhas que contém fontes de corrente

Deste modo é possível fazer a seleção de Malhas Principais, sabendo que é necessário um número M_p de malhas. Será utilizado um ciclo do tipo “while”, que só acabará quando duas condições forem cumpridas:

1. Existe um número M de malhas selecionadas;
2. Todos os ramos do circuito (exceto os que contém fontes de corrente) estão incluídos em malhas principais.

A cada ciclo é testado se a malha contém algum ramo novo, isto é, um ramo ainda não incluído em nenhuma malha (ramo não ocupado). Caso não traga, o algoritmo prossegue para a próxima malha. Caso traga, a malha é retirada do “vetor das malhas”, os ramos dessa malha são marcados como “ocupados” e é criado o objeto malha adicionando-o ao vetor de objetos malha.

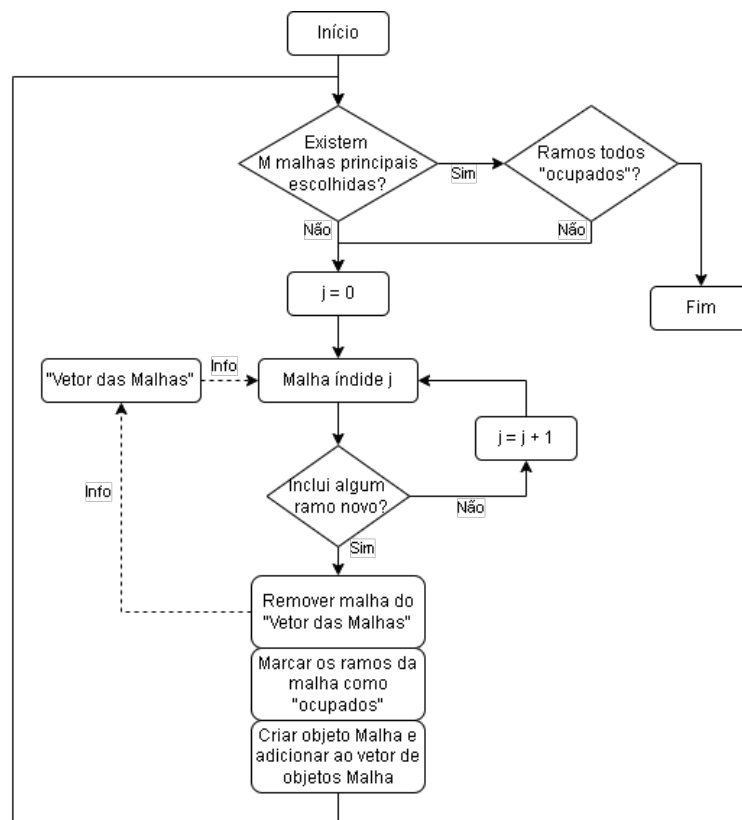


Figura 24: Fluxograma descritivo da escolha de “Malhas Principais”

Desta forma foram escolhidas C Malhas Auxiliares e Mp Malhas Principais, e criados os respetivos objetos malha. O resultado é o vetor de objetos malha.

4.4.4 Tratamento das malhas

4.4.4.1 Algoritmo

Neste ponto do algoritmo cada ramo tem atribuído um “startNode” e um “endNode”, representados pelas letras S e E na figura 25 (que contém um circuito exemplo). As setas cinza representam o sentido “start” → “endNode” (estas setas podem ou não já representar o sentido da corrente no ramo, no entanto essa informação não é relevante para esta computação).

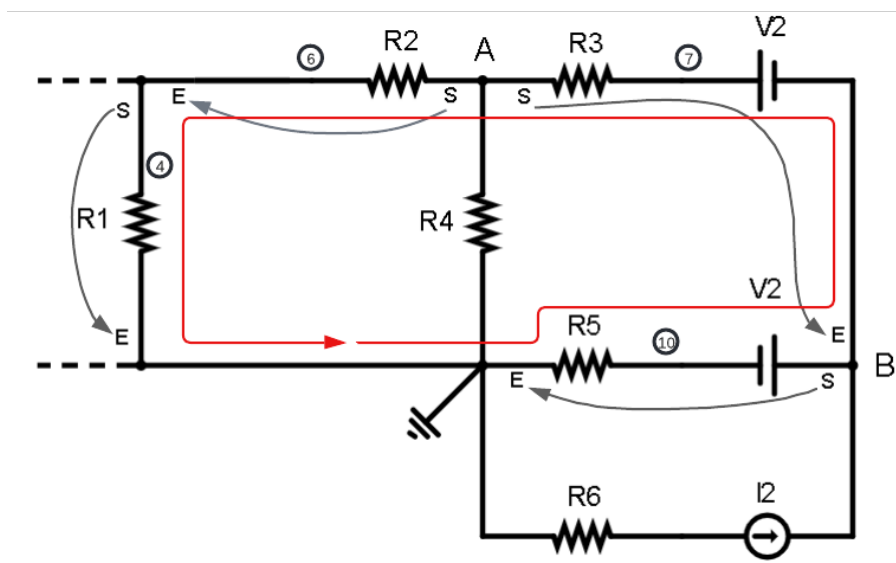


Figura 25: Exemplo de de uma malha a ser construída

A forma mais intuitiva de construir as equações das malhas segundo a *Lei de Kirchhoff das malhas* [15] é percorrer a malha e adicionar os componentes pela ordem em que eles surgem. Esta é uma tarefa fácil para o ser humano, no entanto para a máquina não é tão simples, pois toda a informação que ela dispõe neste momento é a ordem em que os ramos aparecem ao percorrer a malha num dado sentido. Esta informação foi obtida do ficheiro resultante da análise com a biblioteca “Meshes-Finder”, pois cada vetor que contém os ramos da malha está ordenado sequencialmente.

Todavia, no caso de haver múltiplos componentes no mesmo ramo não há maneira de saber se o sentido “startNode” → “endNode” é o mesmo sentido que a malha, logo, não há como adicionar estes componentes à equação pela ordem correta que aparecem no ramo.

É porém possível computar esta informação, e obter um “vetor de direções” que informa se determinado ramo tem o seu sentido “startNode” → “endNode” concordante com o sentido da malha (com o número 1) ou contra o sentido da malha (com o número -1). Para isso é feita uma pesquisa de seguimento de ramos cujos passos são descritos de seguida:

1. Arbitrar um Nó de pesquisa inicial (“startNode” do primeiro ramo);
2. Pesquisar e encontrar qual o único outro ramo que partilha esse Nó;
3. Atualizar o Nó de pesquisa: caso o Nó de pesquisa atual seja o “startNode” do ramo encontrado, atualizá-lo para o “endNode” do ramo encontrado, e vice-versa.
4. Consoante o Nó de pesquisa seja o “startNode” ou o “endNode” o próximo Nó de pesquisa, adicionar ao “vetor de direções” dos ramos “1” ou “-1”, respetivamente;

5. Repetir os passos 2, 3 e 4 até se chegar ao Nó de pesquisa inicial, terminando a pesquisa.

Assim é obtido um “vetor de direções” que descreve os sentidos dos ramos em relação á malha.

É ainda feito um passo adicional caso a malha seja auxiliar. É benéfico que o sentido da malha coincida com o da sua fonte de corrente, então é testado se o ramo que contém a fonte de corrente tem a sua direção a favor ou contra o sentido da malha, conjugando esta informação com o sentido do ramo anteriormente computado. Caso a fonte de corrente e a malha tenham sentidos opostos, o vetor dos ramos da malha é espelhado e o “vetor de direções” dos ramos é alterado para o seu “simétrico”, isto é, onde havia “1” passa a “-1” e vice-versa.

4.4.4.2 Exemplo

Para clarificar todo o algoritmo desenvolvido será agora analisado o caso do circuito exemplo da Figura 25, tratando as informações da malha até ser possível construir a malha azul.

No início do algoritmo a máquina tem um vetor dos ramos da malha ordenados sequencialmente (figura 3):

Após a aplicação do “algoritmo de seguimento de ramos” é encontrado o seguinte “vetor de direções”:

```
1 "branches": [6, 7, 10, 4]
```

Listing 3: Vetor dos Ramos da malha

Pela Figura 3 é possível observar que onde consta “1” no “vetor de direções” corresponde a uma direção relativa concordante entre o ramo e a malha enquanto que onde consta “-1”, esta relação é oposta.

Por fim, como esta é uma malha auxiliar é ainda feito o último passo de teste de fonte de corrente: o sentido da malha e da fonte de corrente são opostos, logo o vetor dos ramos da malha é espelhado e o “vetor das direções” dos ramos é substituído pelo seu “simétrico”. O resultado final é o seguinte:

```
1 "branches": [4, 10, 7, 6]
2 "branchesDir" : [1, -1, -1, 1]
```

Listing 4: Vetor dos Ramos da malha com direções

Este resultado consegue descrever a malha azul da figura 25 que era o objetivo do algoritmo.

4.4.5 Construção e Resolução do Sistema de Equações

Como primeiro passo é feita a obtenção dos componentes em cada ramo, utilizando uma estratégia de seguimento semelhante aos passos 1 para obter

os componentes por ordem no ramo, do “startNode” para o “endNode”. As equações foram construídas utilizando esta informação e conjugando-a com o “vetor de direções” dos ramos.

A construção das equações é constituída por várias parcelas de soma, em que cada uma corresponde a um componente do circuito. Tome-se a figura 26 como exemplo: a parcela na equação da malha I_{Mp2} , relativa à resistência R3 vem: $+R3(I_{Mp2} - I_{Ma1})$.

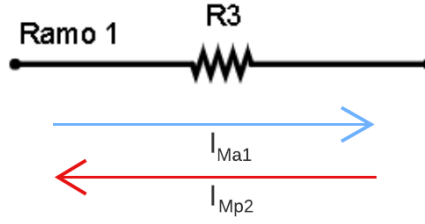


Figura 26: Exemplo de correntes de malha num componente do circuito

Para cada malha principal foram construídas quatro tipos de equações, cada uma com o seu propósito:

- “*All Variable Equation*” - Equação relativa ao primeiro passo do sistema de equações, sem nenhuns valores numéricos, isto é, apenas com os nomes/referências de todos os componentes. Formatada em *LaTeX* para poder ser apresentada nos diferentes outputs diretamente.

$-V2 = R4 \cdot (I_{Ma2} + I_{Mp1} + I_{Mp2}) + R3 \cdot (I_{Mp1})$
$-V2=R4\cdot(I_{M_{a2}}+I_{M_{p1}}+I_{Mp2})+R3\cdot(I_{M_{p1}})$

Tabela 5: Exemplo “*All Variable Equation*” (LaTeX em cima e texto corrido correspondente em baixo)

- “*Revealed Current Source*” - Equação relativa ao segundo passo do sistema de equações, com os valores de fonte de corrente revelados. Formatada em *LaTeX* para poder ser apresentada nos diferentes outputs diretamente.

$-V2 = R4 \cdot (5 + I_{Mp1} + I_{Mp2}) + R3 \cdot (I_{Mp1})$
$-V2=R4\cdot(5+I_{M_{p1}}+I_{Mp2})+R3\cdot(I_{M_{p1}})$

Tabela 6: Exemplo “*Revealed Current Source*” (LaTeX em cima e texto corrido correspondente em baixo)

- “*All Revealed Equation*” - Equação com todas os valores revelados, a excessão das incógnitas do sistema. Formatada em *LaTeX* para poder ser apresentada nos diferentes outputs diretamente. Ver exemplo da figura 7

$-4 = 1 \cdot (5 + I_{M_{p1}} + I_{M_{p2}}) + 3 \cdot (I_{M_{p1}})$
$-4=1\cdot(5+I_{M_{p1}}+I_{M_{p2}})+3\cdot(I_{M_{p1}})$

Tabela 7: Exemplo “*All Revealed Equation*” (LaTeX em cima e texto corrido correspondente em baixo)

- “*Solver Equation*” - Equação equivalente á “*All Revealed Equation*”, porém não formatada em *LaTeX*. Expressão puramente matemática que servirá de “input” para a biblioteca de resolução de sistemas de equações.

$4+1(5+a+b)+3(a)$

Tabela 8: Exemplo “*Solver Equation*”

Estas equações são suficientes para construir e apresentar ao utilizador um sistema com 3 passos de resolução, sendo eles:

- Passo 1 - “Sistema de equações inicial” - Contém todas as equações “*All Variable Equation*”;
- Passo 2 - “Substitua o valor das correntes de malha conhecidas” - Contém todas as equações “*Revealed Current Source*”;
- Passo 3 - “Substitua o valor dos componentes do circuito” - Contém todas as equações “*All Revealed Equation*”;

4.4.6 Resolução do sistema de Equações

É utilizada uma biblioteca para a resolução do sistema de equações, denominada “*Syseqsolver*” [19]. Escolha da sua utilização prendeu-se na simplicidade do seu uso e integração e capacidade de resolução de sistemas tanto reais como com componentes complexas.

Esta biblioteca oferece vários métodos que simplificam a tarefa de resolver um sistema de equações adicionando uma camada de abstração na sua utilização. Os métodos pertencem á classe “*linearEqSystem*” [19] e os que foram utilizados no projeto são os seguintes: método “*addEquation()*” [19] para adicionar uma equação ao sistema, método “*buildSystem()*” [19] para construir o sistema de equações com as equações adicionadas, método “*solve()*” [19] que resolve o sistema e retorna um objeto contendo os resultados.

Os resultados das correntes nas malhas são armazenados no próprio objeto *Malha*.

4.4.7 Computação das Correntes nos Ramos

O cálculo das correntes nos ramos a partir das correntes nas malhas é realizado através de uma simples soma algébrica. Caso a frequência do circuito seja diferente de zero, são somadas as componentes reais e imaginárias das correntes, algebricamente e em separado.

A informação de que correntes de malha fluem em cada ramo, bem como o sentido relativo entre a corrente no ramo e as várias possíveis correntes de malha que nele existem é obtida simplesmente percorrendo todas os ramos, e para cada ramo, percorrendo todas as malhas e testando se cada malha inclui cada ramo. Caso inclua essa malha é guardada no objeto Ramo bem como a direção relativa entre o ramo e a malha.

Diga-se que se quer calcular a corrente do ramo 1 de um circuito genérico (figura 27).

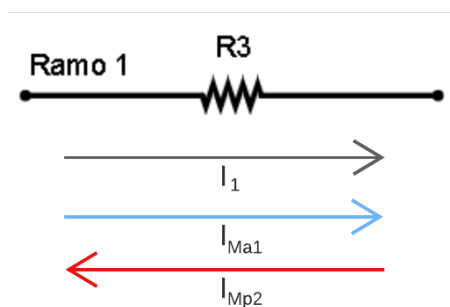


Figura 27: Exemplo de construção de equações das corrente nos ramos de um circuito

Arbitrou-se a corrente I_1 , representada pela seta cinza na figura. Para o cálculo basta apenas somar o valor das correntes nas malhas que contém esse ramo, tomando em atenção o sentido das mesmas: caso o sentido de uma dada malha nesse ramo seja o mesmo soma-se a malha, caso seja contrário subtrai-se a malha. Ao mesmo tempo é construída a expressão da soma para ser depois apresentada. Para o ramo 1, a corrente 1 exemplo da figura teria como equação:

$$I_1 = I_{Ma1} - I_{Mp2} \quad (5)$$

Os resultados das correntes nos ramos são armazenados num objeto que contém todas as informações dessa corrente, bem como a sua expressão. Aqui termina o algoritmo do *Método das Correntes nas Malhas (MCM)*.

4.5 Atualizar Ficheiro JSON

O ficheiro JSON é atualizado com todas as novas informações. Nomeadamente é atualizado o seu campo relativo às informações da análise: “analysisObj”.

São adicionadas as correntes nos ramos, o vetor dos objetos malha que foram escolhidas no algoritmo, o vetor de todas as malhas do circuito e as equações do sistema de equações, como consta na Listagem 5.

```

1      'chosenMeshes': [mesh, mesh, mesh, mesh, mesh, mesh]
2      'circuitFreq': {value: 0, mult: ''}
3      'currents': [{...}, {...}, {...}, {...}, {...}, {...},
   ↪  {...}, {...}, {...}, {...}]
4      'equations': {'allVariableEq': Array(2),
   ↪  'meshCurrRevealedEq': Array(2), 'allRevealed':
   ↪  Array(2), 'allVariableEq': Array(2)}
5      'totalMeshes': [.....]
6      'warnings': [ ]

```

Listing 5: Exemplo *Campo* “analysisObj”

Todas estas novas informações serão utilizadas no próximo passo que é apresentar a informação da análise ao utilizador.

4.6 Output

O Objeto *Output* principal é a *Modal*¹ que é apresentada ao utilizador sempre que é clicado o botão de simulação. No entanto por si só este modo de *Output* não permite ao utilizador guardar a informação com facilidade. Por esse motivo já existiam três formas adicionais: e foram ainda adicionadas duas, num total de cinco *Outputs* diferentes: *Download PDF*, *Download JSON*, *Download TeX*, *Open in Overleaf* e *Print*. Todos eles partem do ficheiro JSON atualizado ou seja todos eles fornecem a mesma informação ao utilizador.

4.6.1 Capa e estrutura do *Output*

A capa do documento de todos os modos de “output” (à exceção do “Modal Output”) tem o mesmo formato qualquer que seja o circuito submetido a análise, mas difere consoante o método que for utilizado. Contém as informações do utilizador caso este as tenha introduzido antes de transferir o ficheiro.

¹Consiste numa janela flutuante que é sobreposta ao ecrã principal da aplicação e que possui capacidades de interação com o utilizador. Salienta-se o uso da framework *Bootstrap* na sua criação

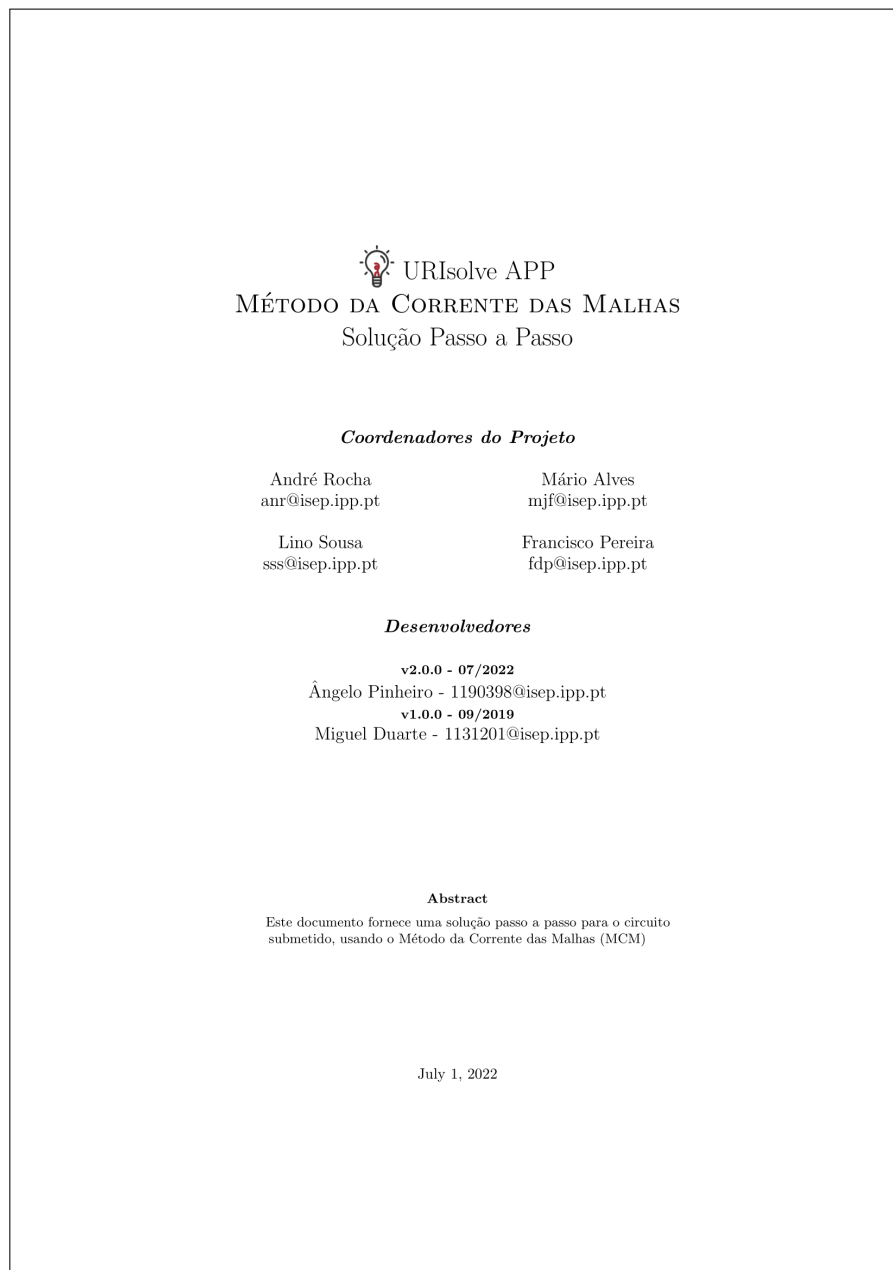


Figura 28: Capa

O restante documento apresenta sempre o mesmo formato qualquer que seja o circuito submetido para análise:

1. Imagem do circuito (caso tenha sido submetida);
2. Varáveis do Circuito. Inclui o número de ramos, nós, fontes de corrente e fontes isoladas de tensão;

3. Informação do Circuito. Inclui tipo de simulação (AC/DC), Frequência e número de amperímetros utilizados;
4. Número de Malhas Principais e Auxiliares;
5. Malhas selecionadas (Auxiliares e Principais) e Equações correspondentes. Inclui diagramas detalhados de cada uma onde se encontram representados os ramos, componentes e direção da malha;
6. Sistema de Equações. Inclui três fases de resolução: Sistema inicial, fontes de corrente substituídas e apenas incógnitas;
7. Valores das Correntes de Malha;
8. Identificação das Correntes nos Ramos. Inclui informações sobre as correntes: “start” e “endNodes”, componentes e identificação e o valor da mesma.

4.6.2 Modal Output

O *Modal Output* é a primeira saída de informação para o utilizador, imediatamente apresentada após ao fim da análise do circuito. Consiste numa janela flutuante onde todos os resultados é apresentada de forma limpa e concisa de forma a proporcionar o máximo de compreensão e clareza ao utilizador.

Tal como em toda a aplicação foi utilizada uma biblioteca denominada [jQuery](#): uma ferramenta baseada em “JavaScript” que interage com o HTML, com o propósito de simplificar o código necessário para desenvolvimento de *web-sites*.

A sintaxe básica para o uso da biblioteca consiste primeiramente em selecionar o elemento HTML pretendido utilizando os seletores CSS e em executar alguma ação. Por exemplo, é possível selecionar um elemento do tipo “div”, com o “id” “circuitInfo” e alterar o seu texto com a seguinte linha de código:

```
1  $('#meuId').html("NovoTexto");
```

Listing 6: Sintaxe básica da biblioteca “jQuery”

Foi também utilizada uma “framework” de desenvolvimento de aplicações em HTML nomeadamente componentes de interface com o utilizador e “frontend” denominada “Bootstrap” [11]. Tem por base um sistema de quadro/tabela, onde é possível, com facilidade, construir o conteúdo pretendido nas diferentes posições do quadro, seja ele que elemento HTML for.

Sendo assim, criadas as várias secções na janela flutuante e as várias () da informação a apresentar em cada uma e usar este método para definir o conteúdo. Para dar origem aos conteúdos de cada secção foram criadas funções que recebem como argumento o ficheiro JSON atualizado e retornam a String. Desta forma o *output* para algumas das secções do projeto é feito da seguinte forma:

```
1 //circuit fundamental variables
2 $('#fundamentalVars').html(outCircuitFundamentalsMCM(jsonFile
  ↪ ));
3
4 //circuit info MCM
5 $('#circuitInfo').html(outCircuitInfoMCM(jsonFile));
6
7 //circuit equations info
8 $('#meshEquations').html(outEquationCalcMCM(jsonFile));
```

Listing 7: Exemplo de aplicação da biblioteca “jQuery” no projeto

4.6.3 “Rich” PDF Output

Este output [23] consiste na transferência de um PDF de máxima qualidade, criado com base em LaTeX. Para isso foi utilizada a biblioteca *LatexPrinter* [17] para a criação do ficheiro.

O uso da biblioteca requer em primeiro lugar a criação de uma variável do tipo “latexprinter” que inicializa um novo ficheiro. Após este passo é necessária a criação de um ficheiro em LaTeX, criando para isso uma String e adicionando as informações necessárias até ter o documento construído. Para fazer adição de imagens é necessário utilizar o método “addImgFile()”, passando-lhe como argumento o objeto imagem que se pretende incluir. O nome da imagem deve ser o mesmo que o especificado no ficheiro LaTeX para poder ser incluído. Por fim é utilizado o método “print()” que constrói o documento PDF e transfere o mesmo.

É necessário ainda fazer a conversão as imagens, que se encontram em formato SVG para formato JPG ou PNG. Foram usados para isso o evento “onload” para executar um trecho de código quando a imagem carrega no ecrã.

```

1  // Instanciate printer object
2  docToPrint = new latexprinter(null, 'printLnk',
   ↪ 'pdfPrintButton');
3  // Add the desired Latex Source Code
4  docToPrint.setTexFile(TeX);
5  // Add Logo Image
6  let sampleimg = base64imgselect({logo});
7  docToPrint.addImgFile('logo.jpg', sampleimg);
8  // Add Circuit Image
9  if(fileContents[0]){
10     let imageObj = new Image();
11     imageObj.src = fileContents[0];
12     sampleimg = resize(imageObj);
13     docToPrint.addImgFile('circuit.jpg', sampleimg);
14 }
15 // Add Canvas Images
16 for(let i = 0; i < canvasObjects.length; i++){
17     let canvas = document.createElement('canvas');
18     canvas.width = canvasObjects[i].width;
19     canvas.height = canvasObjects[i].height;
20     let context = canvas.getContext('2d');
21     let image = new Image();
22     image.onload = function(){
23         let resizedImage = new Image();
24         resizedImage.onload = function(){
25             context.drawImage(resizedImage, 0, 0,
   ↪ resizedImage.width, resizedImage.height);
26             let finalImage = canvas.toDataURL();
27             docToPrint.addImgFile({meshImage}+canvasObjects[i]
   ↪ .id+'.jpg', finalImage);
28         }
29         resizedImage.src = resizeMCM(image, 400).data;
30     }
31     image.src = canvasObjects[i].imageData;
32 }
33 docToPrint.print();

```

Listing 8: Botão “Download PDF”

4.6.4 *Json Output*

Este modo de *output* [23] é o mais simples, pois não envolve criação de nenhum ficheiro novo. Consiste apenas em transferir diretamente o ficheiro JSON criado. Para esse efeito foi criado um novo elemento HTML e definido o atributo como sendo a string composta por “data:text/plain;charset=utf-8,” mais o ficheiro JSON em formato *string* e codificado numa sequência de caracteres do tipo URI. De seguida é definido um nome para o atributo “download” e resta apenas apender o elemento ao corpo do HTML, clicar no elemento (usando o

método “.click”). O ficheiro será transferido.

```
1 // Export JSON File
2 $(document).off().on('click', function() {
3     const filename = 'urisolve_results.json';
4     let element = document.createElement('a');
5     element.setAttribute('href',
6         ↪ 'data:text/plain;charset=utf-8,'
7         + encodeURIComponent(JSON.stringify(jsonFile)));
8     element.setAttribute('download', filename);
9     element.style.display = 'none';
10    document.body.appendChild(element);
11    element.click();
12    document.body.removeChild(element);
13 });
```

Listing 9: Botão “Download JSON”

4.6.5 TeX Output

Este tipo de *output* [23] consiste num botão que quando é clicado transfere dois ficheiros “.tex”: o primeiro sendo o documento principal e o segundo o documento que contém as imagens.

Quanto ao documento principal, foi criado um ficheiro LaTeX, à semelhança do ficheiro criado para o “*Download PDF*” *output*. O documento das imagens consiste na criação de um novo comando para cada imagem, contendo-a em codificada em Base64. No documento principal basta utilizar o pacote “*inlineImages*” e executar o comando correspondente à imagem que se quer inserir.

Criados os dois documentos resta apenas transferi-los utilizando o mesmo método do “*Download Json*” *output*: criar dois elementos, definir os atributos necessários e usar o método “.click()”. Os dois ficheiros serão transferidos.

```

1      let TeX = buildTeX0v(jsonFile, canvasObjects);
2      let ImagesTeX = buildImTeX(canvasObjects);
3
4      //Print TeX (Temporary - Index 1432 - texfile cannot be
5      ↪ change before it)
6      if(studNumber.length>1 && studLastname.length > 1 &&
7      ↪ studNumber.length>1){
8          let string = ``\vspace{0.5cm}\centering{ \r\n
9          ↪ Simulation performed by: \textbf{ "+studName+"
10          ↪ "+studLastname+" (" +studNumber+" )"}";
11          string += `` at " + hourstr + ``\r\n";
12          TeX = TeX.slice(0,1660) + string + TeX.slice(1661);
13      }
14
15      let element = document.createElement('a');
16      element.setAttribute('href',
17      ↪ 'data:text/plain;charset=utf-8,' +
18      ↪ encodeURIComponent(TeX));
19      element.setAttribute('download', filename);
20      let element2 = document.createElement('a');
21      element2.setAttribute('href',
22      ↪ 'data:text/plain;charset=utf-8,' +
23      ↪ encodeURIComponent(ImagesTeX));
24      element2.setAttribute('download', imagesFilename);
25      element.style.display = 'none';
26      element2.style.display = 'none';
27      document.body.appendChild(element);
28      element.click();
29      document.body.removeChild(element);
30      document.body.appendChild(element2);
31      element2.click();
32      document.body.removeChild(element2);

```

Listing 10: Botão “Download TeX”

4.6.6 Open in Overleaf

Este método de *output* consiste num botão que quando clicado abre o ficheiro diretamente na aplicação online [Overleaf](#).

O método de criação do documento é exatamente o mesmo que o “*Download TeX*” *output*, onde são na mesma criados dois ficheiros, um principal e um de imagens (são utilizadas até as mesmas funções). A diferença prende-se no destino dos ficheiros, que ao invés de serem transferidos, é utilizada a funcionalidade do *Overleaf* “One-click to open in Overleaf”.

A funcionalidade em questão consiste em enviar um pedido do tipo “POST” com o objetivo de estabelecer uma ligação e enviar dois “snippets”, isto é, objetos de informação, que o *Overleaf* consegue abrir. Segue-se o código HTML utilizado, onde são criados os elementos *input* necessários.

```
1 <form id='overleaf' action='https://www.overleaf.com/docs'
  ↳ method='post' target='_blank' class='btn btn-secondary
  ↳ font-weight-light align-bottom col-md-2'>
2   <input id='main' type='hidden' name='encoded_snip'>
3   <input id='images' type='hidden' name='encoded_snip'>
4   <input type='hidden' name='snip_name[]' value='main.tex'>
5   <input type='hidden' name='snip_name[]'
  ↳ value='images.tex'>
6   <img src='img/overleaf.png' style='width:
  ↳ 22px;'></img>&nbsp;&nbsp; Open in Overleaf
7 </form>
```

Listing 11: Código HTML referente ao “Open in Overleaf”

Do lado do “JavaScript” basta criar os ficheiros necessários normalmente, fazer a codificação URI de cada um, definir os valores dos elementos HTML como sendo essas Strings URI e chamar o método “submit()”. O pedido será enviado e os documentos abrirão num novo separador do browser.

```

1   let studName = document.getElementById('output-name').value;
2   let studLastname =
3     ↳ document.getElementById('output-lastname').value;
4   let studNumber =
5     ↳ document.getElementById('output-number').value
6   // Get Simulation Time
7   let hourstr = new Date().getHours();
8   let minstr = new Date().getMinutes();
9   if(hourstr.toString().length < 2)
10    hourstr = }0} + hourstr;
11   if(minstr.toString().length < 2)
12    minstr = }0} + minstr;
13   hourstr = hourstr + ':' + minstr;
14   //Print TeX (Temporary - Index 1432 - texfile cannot be
15   ↳ change before it)
16   if(studNumber.length>1 && studLastname.length > 1 &&
17   ↳ studNumber.length>1){
18     let string = '\\vspace{0.5cm}\\centering{ \\r\\n
19     ↳ Simulation performed by: \\textbf{ '+studName+'
20     ↳ '+studLastname+' ('+studNumber+')' } ' ;
21     string += ' at ' + hourstr + '\\r\\n';
22     TeX = TeX.slice(0,1660) + string + TeX.slice(1661);
23   }
24   TeX = TeX.replaceAll('[latin1]', '');
25   document.getElementById('main').value =
26   ↳ encodeURIComponent(TeX);
27   document.getElementById('images').value =
28   ↳ encodeURIComponent(ImagesTeX);
29   document.getElementById('overleaf').submit();

```

Listing 12: Botão “Open in Overleaf”

4.6.7 Print Output

Utilizou-se a biblioteca *jsPDF* [5], uma biblioteca para criação de ficheiros PDF, em “JavaScript”. A sua utilização justifica-se pela capacidade de abrir uma nova janela do browser com o PDF criado, e ainda abrir automaticamente a janela de impressão, que é o objetivo deste *output*.

O documento criado é em tudo semelhante ao *Download PDF*, tendo sido construído de raiz utilizando as funções da biblioteca. O ficheiro resultante é de criação mais rápida, sacrificando um pouco a qualidade em comparação com o *Output* em LaTeX.

A utilização da biblioteca prende-se na utilização de vários métodos *built-in*. Inicialmente é criado um novo objeto do tipo “jsPDF”, inicializando assim o documento. A criação deste objeto pode receber um objeto com várias propriedades que definem o formato do PDF sendo as mais importantes para este projeto a orientação (vertical ou horizontal), o formato (a3, a4, etc), as unidades utilizadas para posicionar elementos nas páginas (mm, pt, px, etc).

Os seguintes métodos representam 90% dos métodos utilizados no projeto: “addPage()”, “setFont()”, “fontSize()”, “addImage()”, “text()” e “line()”. Cada um tem os seus parâmetros específicos, havendo no entanto dois parâmetros comuns a todos: a posição em x e em y.

No projeto foram criadas diversas variáveis tais como “marginTop”, “marginBottom” e “marginSides” que auxiliam na criação do documento, nomeadamente no posicionamento dos diversos elementos. Foi criada também a variável “line” que é incrementada a cada adição que é feita ao documento, mantendo assim guardada a posição vertical de até onde a página do documento está escrita.

Por exemplo, para criar o título da imagem do circuito fez-se:

```
1      doc.setFontSize(subtitleSize);  
2      doc.text('1. ' + lang._circuitImage ,  
      ↪ marginSides*width, line+=35, null, null, 'left');
```

Listing 13: Exemplo de impressão do título “Imagem do circuito”

onde se define em primeiro lugar o tamanho de letra a utilizar, neste caso “subtitleSize” (igual a 16 definido previamente). De seguida é chamado o método “text()” e são-lhe passados os seguintes argumentos: string do texto a adicionar, posição em x, posição em y, rotação/inclinação nula, e alinhamento ao lado esquerdo.

Antes de fazer a adição de um bloco de texto ou imagem é calculado previamente quanto é que ele irá ocupar. Este valor é comparado com o espaço vertical restante na página (calculado usando: $height - line$) e se for menor do que o espaço do objeto a adicionar é criada uma nova página antes de o fazer (usando o método “addPage()”). A cada nova página que é criada é também chamada uma função que imprime o cabeçalho e rodapé.

Uma limitação da biblioteca no âmbito deste projeto é a incapacidade de trabalhar com “LaTeX” embora seja necessário imprimir as equações da resolução do método que é assim que estão escritas/formatadas. Por este motivo foi criada uma função “printEquation()” que recebe a equação formatada em “LaTeX” e imprime-a no documento e que funciona da seguinte forma: A string em “LaTeX” é percorrida do início e cada carácter é testado. Caso não seja nenhum especial é porque é uma letra e é simplesmente imprimido no documento. Caso seja “\” são testados os próximos caracteres em busca de alguma palavra reservada de “LaTeX” como é o exemplo de “cdot” ou “angle” e são imprimidos os símbolos correspondentes.

Segue-se na Figura 29 um exemplo da secção do sistema de equações de um circuito, imprimido com recurso a este método, ilustrando assim o resultado do funcionamento da função “printEquation()”:

6. Sistema de Equações

Equações:

$$\begin{cases} (10 \angle 45^\circ) = 4 \cdot (\underline{I_{Mp1}} - \underline{I_{Mp3}}) \\ 0 = 12 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + 6i \cdot (\underline{I_{Mp2}}) \\ 0 = -3i \cdot ((3 \angle 0^\circ) + \underline{I_{Mp3}}) + 12 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + 4 \cdot (-\underline{I_{Mp1}} + \underline{I_{Mp3}}) \end{cases}$$

Passo 1 - Sistema de equações inicial

$$\begin{cases} \underline{V1} = R1 \cdot (\underline{I_{Mp1}} - \underline{I_{Mp3}}) \\ 0 = R2 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + Z_{L2} \cdot (\underline{I_{Mp2}}) \\ 0 = Z_{C1} \cdot (\underline{I_{Ma1}} + \underline{I_{Mp3}}) + R2 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + R1 \cdot (-\underline{I_{Mp1}} + \underline{I_{Mp3}}) \end{cases}$$

Passo 2 - Substitua o valor das correntes de malha conhecidas

$$\begin{cases} \underline{V1} = R1 \cdot (\underline{I_{Mp1}} - \underline{I_{Mp3}}) \\ 0 = R2 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + Z_{L2} \cdot (\underline{I_{Mp2}}) \\ 0 = Z_{C1} \cdot ((3 \angle 0^\circ) + \underline{I_{Mp3}}) + R2 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + R1 \cdot (-\underline{I_{Mp1}} + \underline{I_{Mp3}}) \end{cases}$$

Passo 3 - Substitua o valor dos componentes do circuito

$$\begin{cases} (10 \angle 45^\circ) = 4 \cdot (\underline{I_{Mp1}} - \underline{I_{Mp3}}) \\ 0 = 12 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + 6i \cdot (\underline{I_{Mp2}}) \\ 0 = -3i \cdot ((3 \angle 0^\circ) + \underline{I_{Mp3}}) + 12 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + 4 \cdot (-\underline{I_{Mp1}} + \underline{I_{Mp3}}) \end{cases}$$

Figura 29: Exemplo de um sistema imprimido usando “jsPDF”

No fim, para criar um ficheiro PDF e abrir automaticamente um novo separador e a janela de impressão basta chamar a função “buildPrintPDF()”, passando-lhe o ficheiro JSON e o objeto que contém as imagens a ser incluídas. Isto é feito caso o botão “Print” seja clicado:

```

1      // Print
2      $('#print').off().on('click', function() {
3          buildPrintPDF(jsonFile, canvasObjects);
4      });

```

Listing 14: Botão “Print”

5 Outras funções de suporte ao algoritmo

5.1 Arredondamento de Resultados

Foi criada uma função cujo objetivo é receber um número e retornar informação suficiente para o apresentar devidamente. O protótipo da função é o seguinte:

```
1 function resultDecimals(number, targetDec, isangle);
```

Listing 15: Protótipo da função “resultDecimals()”

Os argumentos da função são: o “number” que é o número a ser apresentado, “targetDec” que é o número de casas decimais pretendidas e “isangle” que é um valor booleano que informa se o número é ou não um ângulo.

A função retorna um objeto com os seguintes campos: “value” que é a string com o valor arredondado, “unit” que é o prefixo da unidade (“m” para mili, “p” para pico, por exemplo), e “TeXUnit” que é o sufixo em notação científica do valor (“*10⁻³” para mili, “*10⁻⁹” para pico, por exemplo).

Segue-se um exemplo de arredondamento realizado pela função, onde nos diversos *outputs* é apresentado:

$$\underline{Pr2} = 1,27\angle 89,997^\circ \text{ uA}$$

ao invés de:

$$\underline{Pr2} = 0.00000127\angle 89,997^\circ \text{ A}$$

promovendo assim ainda mais compreensão do utilizador dado que os valores são de mais simples entendimento utilizando os prefixos da unidade.

5.2 Desenho das Malhas do circuito

Na finalidade de melhorar a qualidade de apresentação dos resultados foi necessário proporcionar uma componente visual/gráfica das malhas do circuito (Exemplo na figura 30). Para isso foram criadas várias funções que recebem como *input* simplesmente o objeto Malha e criam e anexam um elemento SVG, contendo a imagem, à *modal* de resultados. Retornam ainda as imagens em formato Base64 (svg+xml) para serem utilizados por outros tipos de *output* caso seja necessário.

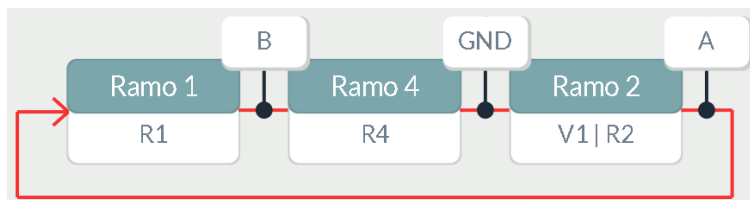


Figura 30: Exemplo do desenho de uma malha de um circuito

O código concebido tem por base o uso de uma biblioteca gráfica, de simples utilização, denominada [D3](#) [3]. Esta facilita a criação de elementos visuais tais como quadrados, círculos, linhas, ou até texto, e a sua anexação aos elementos SVG. A sintaxe base para a criação de um dos quadrados de cada bloco de ramo é a seguinte [3]:

```
1  svg.append("rect")
2      .attr("width", w+2)
3      .attr("height", h/2+2)
4      .attr("x", x-1)
5      .attr("y", y-1)
6      .attr("rx", 6)
7      .attr("ry", 6)
8      .attr("fill", "#6f969a")
```

Listing 16: Exemplo de criação de um quadrado usando o D3

O formato do desenho final da malha é altamente customizável, pois há um nível de abstração em que é possível, por exemplo: alterar a variável da largura de um bloco de ramo e a largura de toda a malha e restantes componentes ser alterada de forma automática.

Na finalidade, para fazer o desenho de uma malha é apenas necessário chamar a função “createMesh()” passando-lhe todos os valores necessários e definidos para o seu desenho, por exemplo: os objetos malha e ramos, a largura e altura de cada bloco, a largura e altura de cada *label* de Nó, o id a atribuir ao elemento SVG criado, etc.

6 Conclusões

Este projeto solicitou a existência de várias competências não só a nível técnico mas também de gestão de projeto. Neste sentido houve a necessidade de delinear um planeamento das várias tarefas a concluir de forma a proporcionar uma perspetiva global do andamento do projeto.

De seguida e também de extrema importância foi feito o estudo das tecnologias existentes e em utilização no projeto já em vigor até ao momento bem como a arquitetura global da aplicação, de modo a poder fazer o planeamento do trabalho a realizar de forma concordante e simbiótica. Salientam-se todas as temáticas do âmbito da programação em JavaScript, HTML e CSS, bem como o estudo do formato do ficheiro “input” da aplicação, a Netlist.

Numa fase inicial foi analisado o código existente, referente ao *Método da Tensão nos Nós* (MTN), em busca de segmentos comuns a todos os métodos. Estes blocos de código foram posteriormente empacotados em funções de modo a poderem ser usados por todos os métodos, aumentando assim a eficiência e utilização de recursos por parte da aplicação.

Uma vez que em alguns pontos do projeto era possível a implementação através de diferentes opções, foi necessário executar uma pesquisa de várias tecnologias bem como fazer o seu teste de forma a encontrar a que melhor satisfaz a relação: cumprimento do objetivo requerido/rapidez de concretização. Acentuam-se os pontos relativos ao “Print output”, utilização da biblioteca “jsPDF”, e do desenho das malhas, uso da biblioteca “D3”.

O desenvolvimento da componente computacional relativa à implementação do método foi uma das fases cruciais do projeto, tendo sido desenvolvidos vários algoritmos críticos ao seu funcionamento, nomeadamente a construção das equações e a escolha de malhas.

Já relativamente ao “output”, para além dos já em utilização na aplicação e a sua adaptação, foram ainda desenvolvidos dois adicionais: o “Print output” já referido, e o “Open in Overleaf”, que oferece uma forma de “output” rápida e com possibilidades de edição posterior sem a necessidade de ter um editor “LaTeX” instalado na máquina. Neste ponto foram ainda desenvolvidos mecanismos de arredondamento e apresentação de resultados, bem como formatação de imagem. Este último ponto constituiu um desafio devido à natureza assíncrona do código envolvido e às conversões entre formatos de imagem.

Por fim decorreu uma fase de testes por parte de professores e alunos de Engenharia Eletrotécnica, onde foram reportados vários erros, “bugs” e sugestões de melhoria, aumentando assim a capacidade de análise da aplicação e a sua qualidade e robustez no geral.

O trabalho desenvolvido tem ainda a possibilidade e flexibilidade de ser melhorado e construído sobre. Uma melhoria simples e que promove a interatividade é a possibilidade de o utilizador escolher as malhas para análise e construção do sistema de equações.

Em suma, resta apontar as competências pessoais adquiridas ao longo de todo o projeto, tanto específicas e técnicas como transversais de ambiente de projetos colaborativos. Salientam-se a programação em JavaScript e criação de documentação de qualidade em “LaTeX”.

Referências

- [1] Khan Academy. Circuit analysis overview. *Khan Academy*, 2020.
- [2] Michal Avny. Symbolab math solver - step by step calculator.
- [3] Mike Bostock. D3.js - data-driven documents.
- [4] Pamela Brittain. Symbolab all-too-easy tool for solving, and seeing, step-by-step math solutions. 2019.
- [5] Lawrence Dudley, Andy Fitch, Dario Grandich, James Hall, and Tom Faller. jspdf - html5 pdf generator — parallax.
- [6] electricalbaba. ideal practical current source explained. *electricalbaba*, 2022.
- [7] Paul Falstad. Falstad circuito simulator.
- [8] Carrie Garges. Microsoft math solver answer key and helpful videos for tricky homework questions. 2022.
- [9] Debbie Gorrell. Photomath one-trick pony displays equation solutions with unique digital tool. 2014.
- [10] Rafael HELERBROCK. Leis de kirchhoff. *electricalbaba*, 2022.
- [11] @mndo, @fat, and comunidade. Bootstrap · the most popular html, css, and js library in the world.
- [12] Vítor Meireles. *Circuitos Elétricos*. LIDEL, 2007.
- [13] Microsoft. Microsoft math solver - math problem solver & calculator.
- [14] Ramakrishna Natarajan. *Mathematical Analysis of Electronic Circuits*. PhD thesis, University of Wyoming, 1948.
- [15] Thyago Ribeiro. Leis de kirchhoff. *eletricidade*.
- [16] Wilton Robert. *Transient Response of Four-Terminal Networks*. PhD thesis, Iowa State College, 1945.
- [17] André Rocha. Github - txroot/latexprinter: Print latex to pdf.
- [18] André Rocha. Github - txroot/meshes-finder: Meshes-finder.
- [19] André Rocha. Github - txroot/syseqsolver: Linear equations system solver (unique solution and equations = unknowns).
- [20] André Rocha. Meshes-finder algorithm, 2021.
- [21] André Rocha, Lino Sousa, Mário Alves, and Francisco Pereira. Métodos gerais de análise de circuitos. 2022.
- [22] Damir Sabol. Math explained, step-by-step.
- [23] Lino Manuel Cerqueira Sousa. U=rresolve: a web-based tool for teaching & self-learning the nodal voltage method. 2020.

- [24] Ana Viana, Mário Alves, and Francisco Pereira. Métodos de análise e simplificação de circuitos eléctricos (parte 2). 2015.
- [25] Stephen Wolfram. Wolfram—alpha: Making the world’s knowledge computable.
- [26] Miguel Ângelo da Rocha Duarte. Aplicação para geração da metodologia/equações do método das correntes nas malhas, a partir do qucs. 2019.

A Análise MCM de vários circuitos-exemplo

A.1 Circuito-exemplo em Corrente Contínua

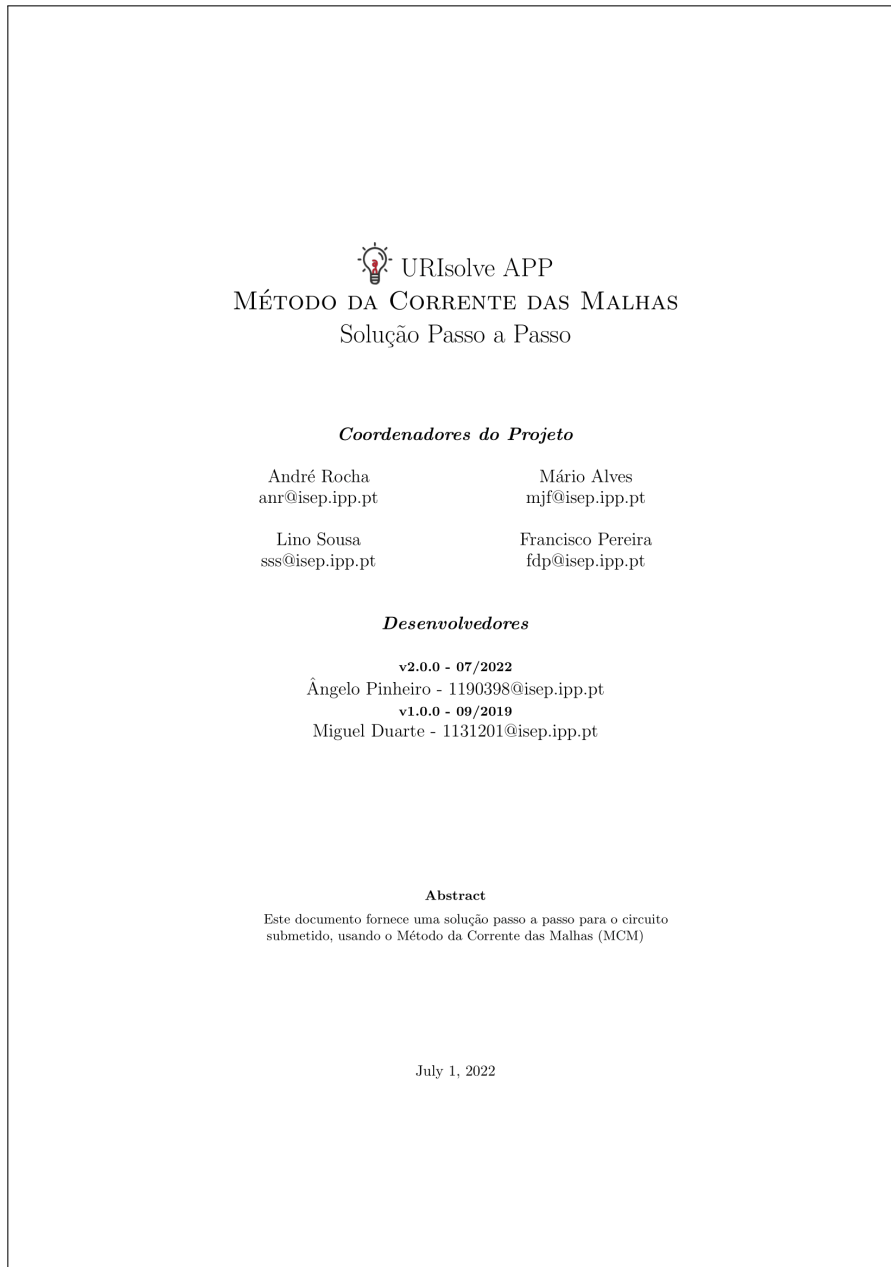


Figura 31: Capa

1 Imagem do Circuito

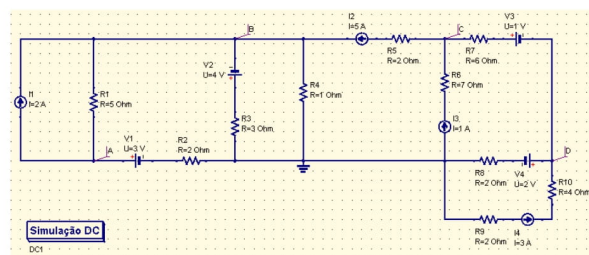


Figure 1: Imagem do Circuito

2 Variáveis do Circuito

Ramos [R]	Nós [N]	Fontes de Corrente [C]	Fontes Isoladas de Tensão [T]
R=10	N=5	C=4	T=0

3 Informação do Circuito

Simulação [AC/DC]	Frequência [A]	Amperímetros [I]
DC	N / A	0/10

4 Número de malhas Principais e Auxiliares

4.1 Malhas Principais

$$\begin{aligned} M_p &= R - (N - 1) - C \Leftrightarrow \\ M_p &= 10 - (5 - 1) - 4 \Leftrightarrow \\ &\Leftrightarrow M_p = 2 \end{aligned}$$

Nota: O número de malhas principais (Mp) calculado corresponde ao número de equações das malhas (do sistema de equações), em que as incógnitas são as correntes (fictícias) das malhas.

Figura 32: Página 1 Exemplo DC

4.2 Malhas Auxiliares

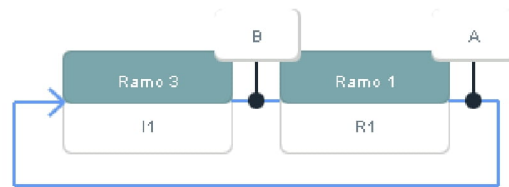
Nota: O número de malhas auxiliares é o mesmo que o número de fontes de corrente (FC). Cada malha auxiliar passa numa única FC, normalmente escolhendo-se o sentido de circulação da (corrente da) malha igual de forma a alinhar-se no sentido da corrente da FC, assumindo assim o seu próprio valor (positivo).

$$C = 4 \implies M_a = 4$$

Figura 33: Página 2 Exemplo DC

5 Malhas seleccionadas (Auxiliares e Principais) e equações correspondentes

5.1 Malha 1 - Auxiliar



$$\text{Value : } I_{Ma1} : 2A \quad (1)$$

5.2 Malha 2 - Auxiliar



$$\text{Value : } I_{Ma2} : 5A \quad (2)$$

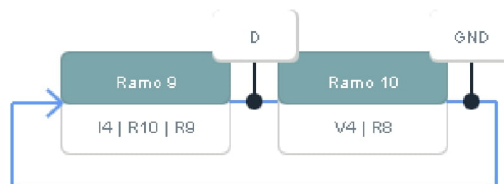
Figura 34: Página 3 Exemplo DC

5.3 Malha 3 - Auxiliar



$$\text{Value : } I_{Ma3} : 1A \quad (3)$$

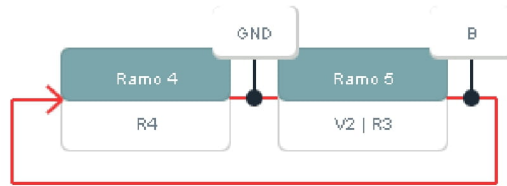
5.4 Malha 4 - Auxiliar



$$\text{Value : } I_{Ma4} : 3A \quad (4)$$

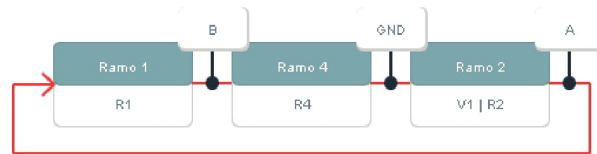
Figura 35: Página 4 Exemplo DC

5.5 Malha 5 - Principal



$$\text{Equation : } I_{Mp1} : -V2 = R4 \cdot (I_{Ma2} + I_{Mp1} + I_{Mp2}) + R3 \cdot (I_{Mp1}) \quad (5)$$

5.6 Malha 6 - Principal



$$\text{Equation : } I_{Mp2} : V1 = R1 \cdot (-I_{Ma1} + I_{Mp2}) + R4 \cdot (I_{Ma2} + I_{Mp1} + I_{Mp2}) + R2 \cdot (I_{Mp2}) \quad (6)$$

Figura 36: Página 5 Exemplo DC

6 Sistema de Equações

Equações:

$$\begin{cases} -4 = 1 \cdot (5 + I_{Mp1} + I_{Mp2}) + 3 \cdot (I_{Mp1}) \\ 3 = 5 \cdot (-2 + I_{Mp2}) + 1 \cdot (5 + I_{Mp1} + I_{Mp2}) + 2 \cdot (I_{Mp2}) \end{cases}$$

Passos:

Step 1: Initial equation system

$$\begin{cases} -V2 = R4 \cdot (I_{Ma2} + I_{Mp1} + I_{Mp2}) + R3 \cdot (I_{Mp1}) \\ V1 = R1 \cdot (-I_{Ma1} + I_{Mp2}) + R4 \cdot (I_{Ma2} + I_{Mp1} + I_{Mp2}) + R2 \cdot (I_{Mp2}) \end{cases}$$

Step 2: Substitute the mesh current values

$$\begin{cases} -V2 = R4 \cdot (5 + I_{Mp1} + I_{Mp2}) + R3 \cdot (I_{Mp1}) \\ V1 = R1 \cdot (-2 + I_{Mp2}) + R4 \cdot (5 + I_{Mp1} + I_{Mp2}) + R2 \cdot (I_{Mp2}) \end{cases}$$

Step 3: Substitute the circuit component values

$$\begin{cases} -4 = 1 \cdot (5 + I_{Mp1} + I_{Mp2}) + 3 \cdot (I_{Mp1}) \\ 3 = 5 \cdot (-2 + I_{Mp2}) + 1 \cdot (5 + I_{Mp1} + I_{Mp2}) + 2 \cdot (I_{Mp2}) \end{cases}$$

Figura 37: Página 6 Exemplo DC

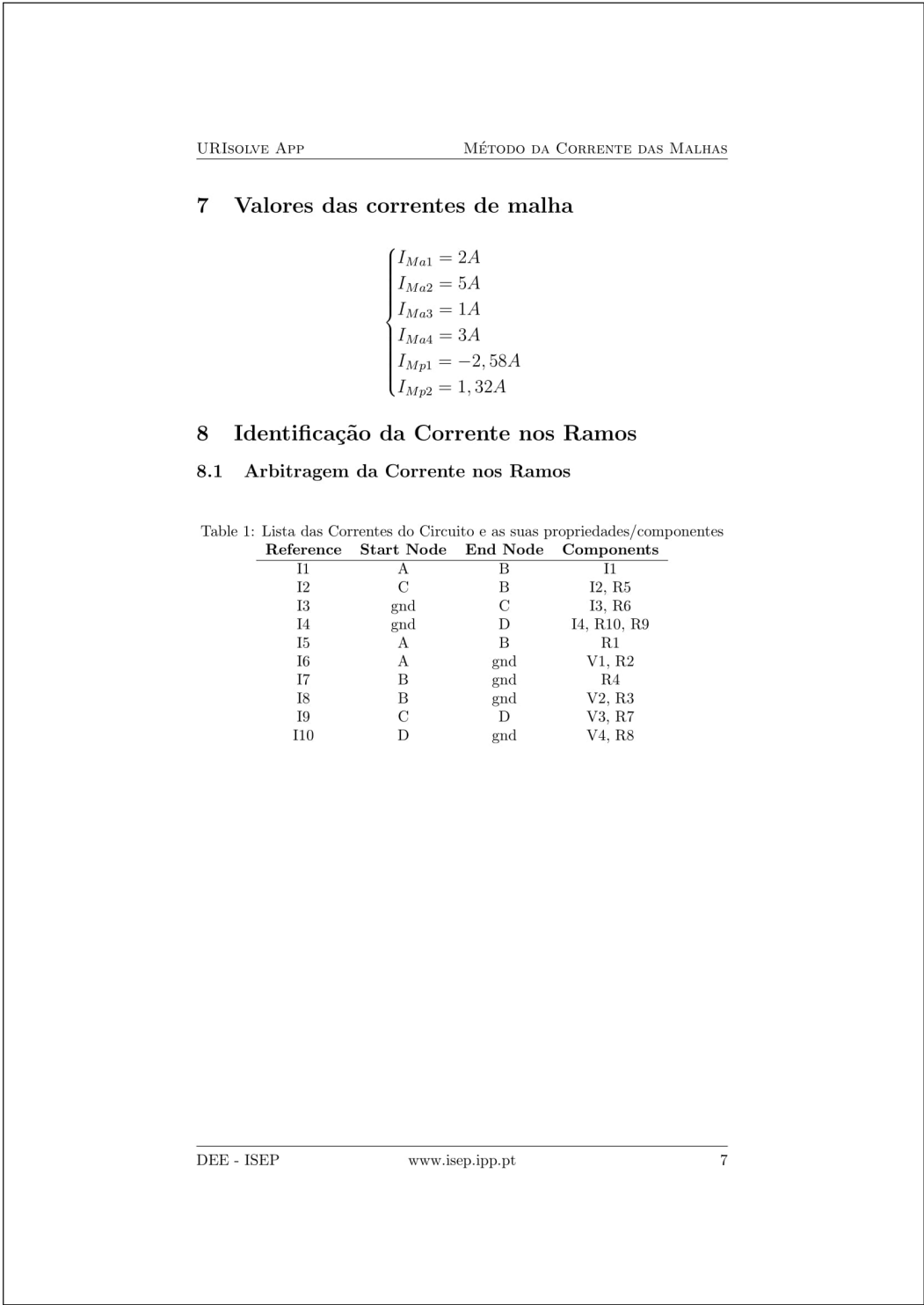


Figura 38: Página 7 Exemplo DC

8.2 Computação da Corrente nos Ramos

$$\left\{ \begin{array}{l} I1 = I_{Ma1} \\ I2 = I_{Ma2} \\ I3 = I_{Ma3} \\ I4 = I_{Ma4} \\ I5 = -I_{Ma1} + I_{Mp2} \\ I6 = -I_{Mp2} \\ I7 = I_{Ma2} + I_{Mp1} + I_{Mp2} \\ I8 = -I_{Mp1} \\ I9 = -I_{Ma2} + I_{Ma3} \\ I10 = -I_{Ma2} + I_{Ma3} + I_{Ma4} \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} I1 = 2 \text{ A} \\ I2 = 5 \text{ A} \\ I3 = 1 \text{ A} \\ I4 = 3 \text{ A} \\ I5 = -677,42 \text{ mA} \\ I6 = -1,32 \text{ A} \\ I7 = 3,74 \text{ A} \\ I8 = 2,58 \text{ A} \\ I9 = -4 \text{ A} \\ I10 = -1 \text{ A} \end{array} \right.$$

Note: cada corrente no ramo foi obtida através da soma algébrica das correntes de malha que existem nesse ramo.

Figura 39: Página 8 Exemplo DC

A.2 Circuito-exemplo em Corrente Alternada

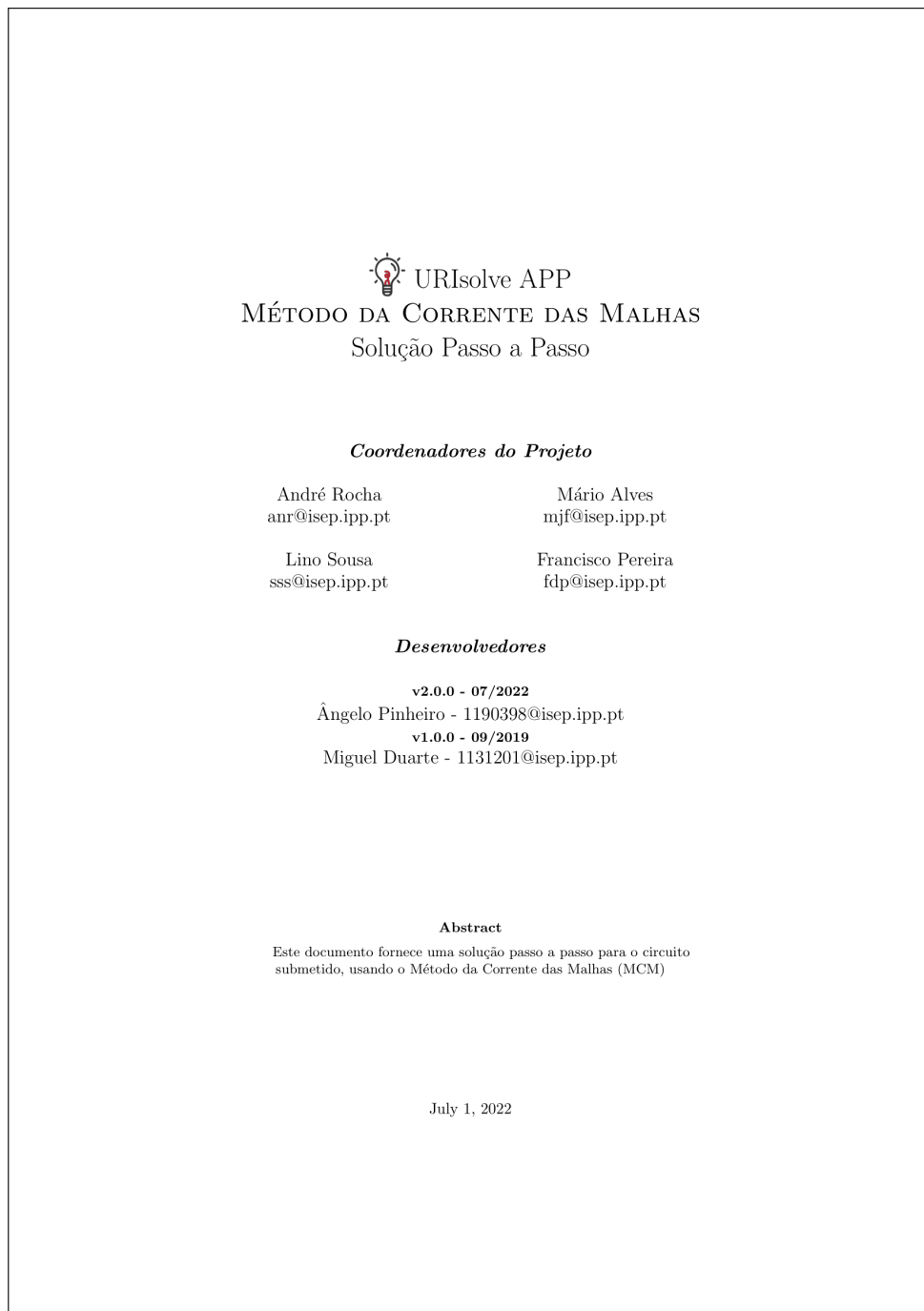
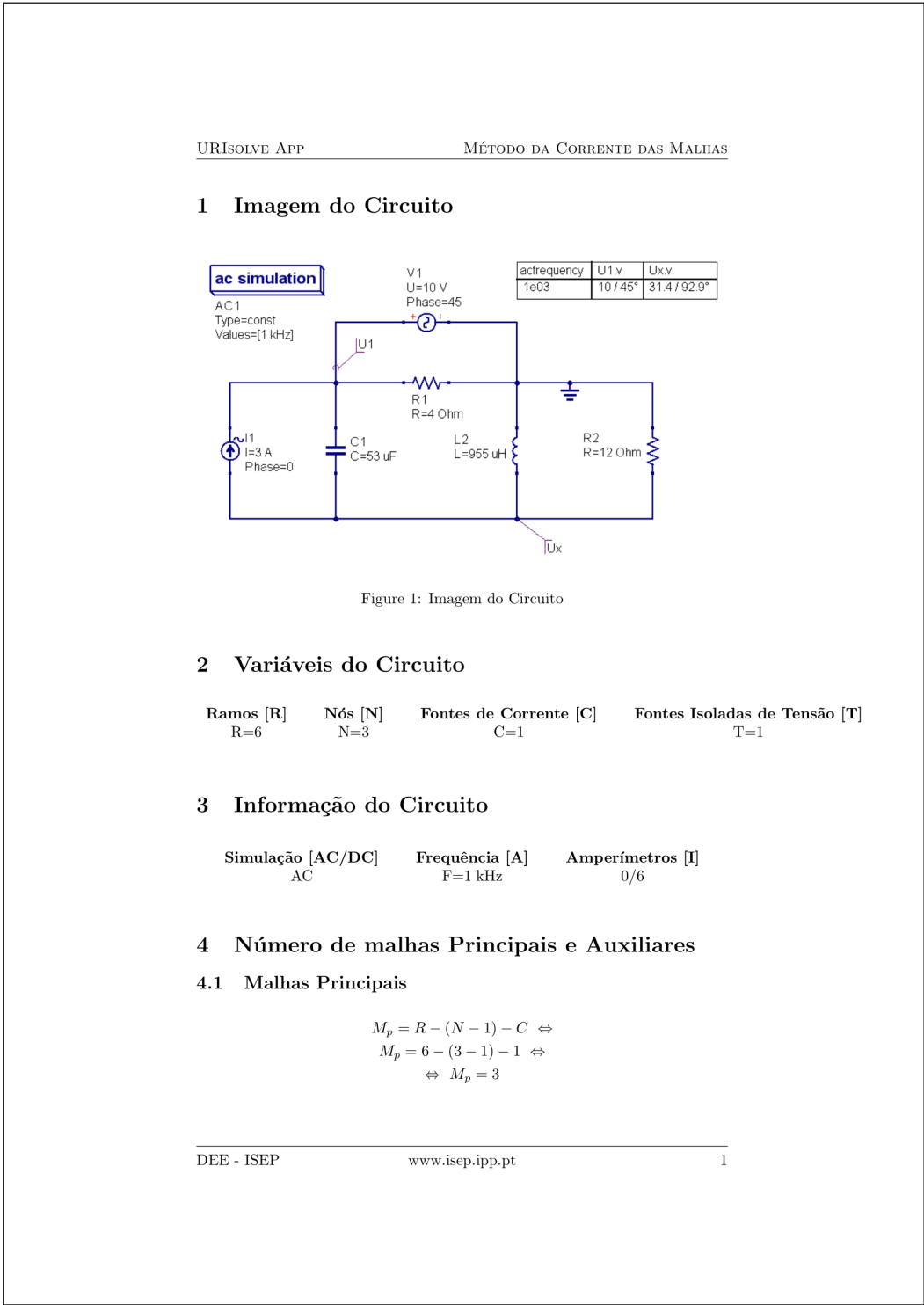


Figura 40: Capa



Nota: O número de malhas principais (Mp) calculado corresponde ao número de equações das malhas (do sistema de equações), em que as incógnitas são as correntes (fictícias) das malhas.

4.2 Malhas Auxiliares

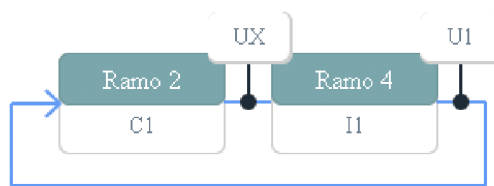
Nota: O número de malhas auxiliares é o mesmo que o número de fontes de corrente (FC). Cada malha auxiliar passa numa única FC, normalmente escolhendo-se o sentido de circulação da (corrente da) malha igual de forma a alinhar-se no sentido da corrente da FC, assumindo assim o seu próprio valor (positivo).

$$C = 1 \implies M_a = 1$$

Figura 42: Página 2 Exemplo AC

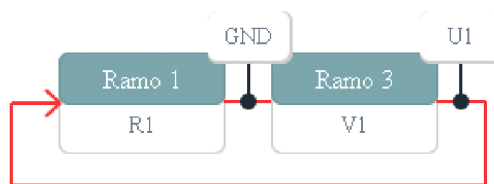
5 Malhas seleccionadas (Auxiliares e Principais) e equações correspondentes

5.1 Malha 1 - Auxiliar



$$\text{Value : } \underline{I_{Ma1}} : 3\angle 0^\circ A \quad (1)$$

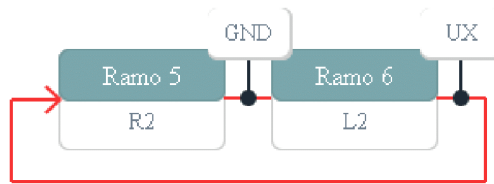
5.2 Malha 2 - Principal



$$\text{Equation : } \underline{I_{Mp1}} : \underline{V1} = R1 \cdot (\underline{I_{Mp1}} - \underline{I_{Mp3}}) \quad (2)$$

Figura 43: Página 3 Exemplo AC

5.3 Malha 3 - Principal



$$\text{Equation : } \quad \underline{I_{Mp2}} : 0 = R2 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + \underline{Z_{L2}} \cdot (\underline{I_{Mp2}}) \quad (3)$$

5.4 Malha 4 - Principal



$$\text{Equation : } \quad \underline{I_{Mp3}} : 0 = \underline{Z_{C1}} \cdot (\underline{I_{Ma1}} + \underline{I_{Mp3}}) + R2 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + R1 \cdot (-\underline{I_{Mp1}} + \underline{I_{Mp3}}) \quad (4)$$

Figura 44: Página 4 Exemplo AC

6 Sistema de Equações

Equações:

$$\begin{cases} (10\angle 45^\circ) = 4 \cdot (\underline{I_{Mp1}} - \underline{I_{Mp3}}) \\ 0 = 12 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + 6i \cdot (\underline{I_{Mp2}}) \\ 0 = -3i \cdot ((3\angle 0^\circ) + \underline{I_{Mp3}}) + 12 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + 4 \cdot (-\underline{I_{Mp1}} + \underline{I_{Mp3}}) \end{cases}$$

Passos:

Passo 1: Sistema de equações inicial

$$\begin{cases} \underline{V1} = R1 \cdot (\underline{I_{Mp1}} - \underline{I_{Mp3}}) \\ 0 = R2 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + \underline{Z_{L2}} \cdot (\underline{I_{Mp2}}) \\ 0 = \underline{Z_{C1}} \cdot (\underline{I_{Ma1}} + \underline{I_{Mp3}}) + R2 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + R1 \cdot (-\underline{I_{Mp1}} + \underline{I_{Mp3}}) \end{cases}$$

Passo 2: Substitua o valor das correntes de malha conhecidas

$$\begin{cases} \underline{V1} = R1 \cdot (\underline{I_{Mp1}} - \underline{I_{Mp3}}) \\ 0 = R2 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + \underline{Z_{L2}} \cdot (\underline{I_{Mp2}}) \\ 0 = \underline{Z_{C1}} \cdot ((3\angle 0^\circ) + \underline{I_{Mp3}}) + R2 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + R1 \cdot (-\underline{I_{Mp1}} + \underline{I_{Mp3}}) \end{cases}$$

Passo 3: Substitua o valor dos componentes do circuito

$$\begin{cases} (10\angle 45^\circ) = 4 \cdot (\underline{I_{Mp1}} - \underline{I_{Mp3}}) \\ 0 = 12 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + 6i \cdot (\underline{I_{Mp2}}) \\ 0 = -3i \cdot ((3\angle 0^\circ) + \underline{I_{Mp3}}) + 12 \cdot (\underline{I_{Mp2}} + \underline{I_{Mp3}}) + 4 \cdot (-\underline{I_{Mp1}} + \underline{I_{Mp3}}) \end{cases}$$

Figura 45: Página 5 Exemplo AC

7 Valores das correntes de malha

$$\begin{cases} \underline{I_{Ma1}} = 3\angle 0^\circ \text{ A} \\ \underline{I_{Mp1}} = 8,29\angle 34,076^\circ \text{ A} \\ \underline{I_{Mp2}} = 5,24\angle 2,871^\circ \text{ A} \\ \underline{I_{Mp3}} = 5,86\angle 29,438^\circ \text{ A} \end{cases}$$

8 Identificação da Corrente nos Ramos

8.1 Arbitragem da Corrente nos Ramos

Table 1: Lista das Correntes do Circuito e as suas propriedades/componentes

Reference	Start Node	End Node	Components
I1	Ux	U1	I1
I2	U1	gnd	R1
I3	U1	Ux	C1
I4	U1	gnd	V1
I5	Ux	gnd	R2
I6	Ux	gnd	L2

Figura 46: Página 6 Exemplo AC

8.2 Computação da Corrente nos Ramos

$$\left\{ \begin{array}{l} \underline{I1} = \underline{I_{Ma1}} \\ \underline{I2} = \underline{I_{Mp1}} - \underline{I_{Mp3}} \\ \underline{I3} = \underline{I_{Ma1}} + \underline{I_{Mp3}} \\ \underline{I4} = -\underline{I_{Mp1}} \\ \underline{I5} = \underline{I_{Mp2}} + \underline{I_{Mp3}} \\ \underline{I6} = -\underline{I_{Mp2}} \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} \underline{I1} = 3\angle 0^\circ \text{ A} \\ \underline{I2} = 2,5\angle 45^\circ \text{ A} \\ \underline{I3} = 8,6\angle 19,563^\circ \text{ A} \\ \underline{I4} = 8,29\angle 34,076^\circ \text{ A} \\ \underline{I5} = 2,62\angle -87,129^\circ \text{ A} \\ \underline{I6} = 5,24\angle 2,871^\circ \text{ A} \end{array} \right.$$

Note: cada corrente no ramo foi obtida através da soma algébrica das correntes de malha que existem nesse ramo.

Figura 47: Página 7 Exemplo AC