

U=RI solve APP Implementação do Método das Correntes nos Ramos

Helder Casanova Carvalho

LEEC



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto

Janeiro, 2023

Candidato: Helder Casanova Carvalho, N.º 1171114, 1171114@isep.ipp.pt

Orientação Científica: Francisco Pereira, fdp@isep.ipp.pt

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



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

Janeiro, 2023

Resumo

A aplicação U=RIolve é um simulador de circuitos elétricos que além de gerar os valores das correntes e das quedas de tensão no circuito, apresenta a análise detalhada do circuito através de um método de análise de circuitos.

Neste projeto foi desenvolvida a análise através do método das correntes nos ramos.

Para o uso desta aplicação é necessário que o utilizador forneça uma Netlist gerada noutro simulador de circuitos. Depois de ter a Netlist basta submetê-la e escolher o método que pretende utilizar, quando escolher será lhe mostrado o *output* com todos os resultados finais e também a maneira de chegar a estes. Por fim, se o utilizador quiser guardar a informação obtida são lhe apresentadas várias opções tais como *Portable Document Format* (PDF), *TeX*, *JavaScript Object Notation* (JSON), criar um projeto *Overleaf*[1] ou mesmo imprimir o PDF.

Palavras-Chave: U=RIolve, didática, circuito, output, aplicação, netlist, JSON, aprendizagem.

Abstract

The U=RI solve application is an electric circuit simulator that besides generating the values of the currents and voltage drops in the circuit, it presents the detailed analysis of the circuit through a method of circuit analysis.

In this project was developed the analysis through the method of the currents in branches.

To use this application it is necessary that the user provides a netlist and as such he needs to access another circuit simulator, after having the netlist can just submit it and choose the method he wants to analyze, when chosen, it will be shown the output with all the final results and also the way to get to them. Finally, if the user wants to save the obtained information he will be presented with several options such as PDF, *TeX*, JSON, create an *Overleaf*[1] project or even print the PDF.

Keywords: U=RI solve, circuit, output, application, netlist, didactic.

Índice

Lista de Figuras	vii
Lista de Tabelas	ix
Glossário	xi
Lista de Acrónimos	xiii
1 Introdução	1
1.1 Contextualização	1
1.2 Organização do Relatório	2
1.3 Calendarização do projeto	2
2 Estado da Arte	3
2.1 Qucs	3
2.2 Falstad	4
2.3 Comparação com U=RI solve	5
3 Terminologia e Conceitos Fundamentais	7
3.1 Terminologia	7
3.2 Método de análise	8
3.2.1 <i>Método das Correntes nos Ramos</i> (MCR)	8
3.2.2 Exemplo de resolução MCR	9
4 Algoritmo	11
4.1 Enquadramento	11
4.2 Encapsulamento de funções	13
4.3 Ficheiro JSON	13
4.4 Algoritmo MCR	14
4.4.1 Calcular o Número de Malhas Necessárias	14
4.4.2 Criar as Equações dos Nós	14
4.4.3 Selecionar as Malhas	15
4.4.4 Criar as Equações das Malhas	18
4.4.5 Criar o Sistema de Equações	21
4.4.6 Resolver o Sistema de Equações	21

4.4.7	Atualizar o Ficheiro JSON	21
4.5	Output	23
4.5.1	Estrutura dos Outputs	23
4.5.2	Modal Output	24
4.5.3	Download PDF Output	24
4.5.4	Download JSON	24
4.5.5	Download TeX	24
4.5.6	Open in Overleaf	25
4.5.7	Print	26
5	Conclusões	27
5.1	Trabalho Futuro	28
	Referências	29
Anexo A	Análise de Circuitos Exemplo	31
A.1	Circuito em Corrente Contínua	31
A.1.1	Imagem do Circuito	31
A.1.2	Netlist do Circuito	32
A.1.3	Output em PDF	33
A.2	Circuito em Corrente Contínua com Fonte de Corrente	40
A.2.1	Imagem do Circuito	40
A.2.2	Netlist do Circuito	41
A.2.3	Output em PDF	42
A.3	Circuito em Corrente Alternada	49
A.3.1	Imagem do Circuito	49
A.3.2	Netlist do Circuito	50
A.3.3	Output em PDF	51

Lista de Figuras

2.1	Aplicação Qucs	4
2.2	Aplicação Falstad	4
3.1	Exemplo de um ramo	7
3.2	Exemplo de um nó	8
3.3	Exemplo de malhas [9]	8
3.4	Exemplo de um circuito [11]	9
4.1	Arquitetura do projeto	12
4.2	Exemplo de um ficheiro JSON com as variáveis iniciais de um circuito	13
4.3	Diagrama de blocos do Algoritmo desenvolvido	14
4.4	Fluxograma da formação das equações dos nós	15
4.5	Exemplo de um equação do nó	15
4.6	Circuito a analisar para matrizes [13]	16
4.7	Objeto com todas as malhas[13]	17
4.8	Fluxograma da escolha das malhas	18
4.9	Exemplo de um circuito DC	19
4.10	Campo " <i>analysisObj</i> " do ficheiro JSON final	22
A.1	Imagem do exemplo a ser analisado em corrente contínua	31
A.2	Capa do Output PDF corrente contínua	33
A.3	Pág 1 do Output PDF corrente contínua	34
A.4	Pág 2 do Output PDF corrente contínua	35
A.5	Pág 3 do Output PDF corrente contínua	36
A.6	Pág 4 do Output PDF corrente contínua	37
A.7	Pág 5 do Output PDF corrente contínua	38
A.8	Pág 6 do Output PDF corrente contínua	39
A.9	Imagem do exemplo a ser analisado em corrente Contínua com fonte de Corrente	40
A.10	Capa do Output PDF corrente contínua com fonte de corrente . . .	42
A.11	Pág 1 do Output PDF corrente contínua com fonte de corrente . . .	43
A.12	Pág 2 do Output PDF corrente contínua com fonte de corrente . . .	44
A.13	Pág 3 do Output PDF corrente contínua com fonte de corrente . . .	45
A.14	Pág 4 do Output PDF corrente contínua com fonte de corrente . . .	46

A.15 Pág 5 do Output PDF corrente contínua com fonte de corrente . . .	47
A.16 Pág 6 do Output PDF corrente contínua com fonte de corrente . . .	48
A.17 Imagem do exemplo a ser analisado em corrente Alternada	49
A.18 Capa do Output PDF corrente alternada	51
A.19 Pág 1 do Output PDF corrente alternada	52
A.20 Pág 2 do Output PDF corrente alternada	53
A.21 Pág 3 do Output PDF corrente alternada	54
A.22 Pág 4 do Output PDF corrente alternada	55
A.23 Pág 5 do Output PDF corrente alternada	56
A.24 Pág 6 do Output PDF corrente alternada	57
A.25 Pág 7 do Output PDF corrente alternada	58

Lista de Tabelas

1.1	Calendarização do Projeto	2
4.1	Matriz de adjacência	16
4.2	Matriz de incidência	16
4.3	<i>Array</i> das malhas	17
4.4	<i>Array</i> dos ramos	17

Glossário

Fonte de corrente ideal

Uma fonte de corrente ideal é um dispositivo que fornece uma corrente constante independentemente da carga. O valor da corrente será constante em relação ao tempo e à resistência da carga.[2]

Framework

Framework em programação é uma ferramenta que fornece componentes prontos ou soluções adaptadas para acelerar o desenvolvimento. Uma *framework* pode incluir uma biblioteca, mas é definido pelo princípio de inversão de controle (IoC). Com a programação tradicional, o código personalizado chama a biblioteca para acessar o código reutilizável. Com o IoC, a estrutura chama partes de código personalizadas quando necessário.[3]

Malha

Um circuito fechado que inicia em um nó e termina no mesmo nó.[4]

Netlist

Ficheiro de texto contendo todas as informações sobre um circuito (nomeadamente os componentes que o constituem e as suas ligações), este ficheiro é gerado por um simulador de circuitos.

Nó

Ponto de interligação de três ou mais ramos.[4]

Ramo

ligação entre dois nós que pode contém os mais variados componentes elétricos.

Lista de Acrónimos

FEELE	<i>Fundamentos de Engenharia Eletrotécnica</i>
HTML	<i>HyperText Markup Language</i>
ISEP	<i>Instituto Superior de Engenharia do Porto</i>
JSON	<i>JavaScript Object Notation</i>
LEEC	<i>Licenciatura em Engenharia Eletrotécnica e de Computadores</i>
MCM	<i>Método das Correntes nas Malhas</i>
MCR	<i>Método das Correntes nos Ramos</i>
MTN	<i>Método das Tensões nos Nós</i>
PDF	<i>Portable Document Format</i>

Capítulo 1

Introdução

A rápida análise e compreensão de um circuito elétrico é fundamental para um engenheiro da área de Engenharia Eletrotécnica/Elétrica.

A análise de um circuito elétrico consiste em calcular a corrente elétrica nos ramos e a queda de tensão aos terminais dos componentes.

Neste documento está presente a descrição detalhada daquilo que foi a realização do projeto proposto pelos engenheiros Mário Alves, Francisco Pereira, Lino Sousa e André Rocha e este consiste em introduzir o *Método das Correntes nos Ramos* (MCR) na aplicação $U=RI$ solve que consiste em dar aos estudantes de engenharia, ou mesmo a qualquer pessoa autodidata que se interesse no ramo, uma resolução passo a passo dos diferentes métodos de análise existentes, mais especificamente o *Método das Tensões nos Nós* (MTN) (já presente na aplicação desde o início da realização deste projeto) e o *Método das Correntes nas Malhas* (MCM) (outro projeto a realizar pelo colega de curso Ângelo Pinheiro).

É possível testar o *Método da Corrente nos Ramos* (MCR) aqui

1.1 Contextualização

Existem alunos que a quando da data de ingresso na *Licenciatura em Engenharia Eletrotécnica e de Computadores* (LEEC) no *Instituto Superior de Engenharia do Porto* (ISEP), têm um conhecimento praticamente nulo no que se trata a circuitos elétricos, algo notável pelos professores de *Fundamentos de Engenharia Eletrotécnica* (FEELE) nas primeiras aulas. Para auxiliar os alunos nos seus estudos foi então

criada esta aplicação [5], algo que não existe aquando da escrita deste relatório, existindo apenas aplicações para resolver equações, não apresentando a resolução passo a passo de um circuito sendo-lhe só fornecida a Netlist

1.2 Organização do Relatório

Este relatório é composto por 5 capítulos que contém todo o conteúdo necessário de maneira a que o leitor compreenda o funcionamento não só do projeto mas também da aplicação U=RI solve,

Antes destes 5 capítulos, é apresentado os agradecimentos por parte do autor do relatório e também um pequeno resumo do que consiste o projeto.

Nos capítulos que contém todo o conteúdo, está presente uma breve introdução, o estado da arte, a terminologia e conceitos necessários para a leitura do relatório, todo o algoritmo realizado e por fim as conclusões finais.

Por último estão presente as referências que foram utilizadas e um anexo que consiste na análise de dois circuitos exemplo, um em corrente contínua e um em corrente alternada, do algoritmo criado.

1.3 Calendarização do projeto

Tarefa/Semana	1	2	3	4	5	6	7	8	9	10	11	12	13
Estudo do método													
Introdução a JavaScript													
Estudo da aplicação existente													
Desenvolvimento dos algoritmos													

Tarefa/Semana	14	15	16	17	18	19	20	21	22	23	24
Desenvolvimento dos algoritmos											
Realização dos Outputs											
Redação do Relatório											

Tabela 1.1: Calendarização do Projeto

Capítulo 2

Estado da Arte

Neste capítulo vão ser analisadas as tecnologias no âmbito da aplicação U=RI solve e por fim comparar com a mesma, algumas das aplicações escolhidas foram *QUCS*[6], *Falstad*[7].

2.1 Qucs

O *Qucs* é um simulador circuitos bastante universal que permite configurar um circuito à vontade do utilizador com bastante facilidade exceto quando os componentes são rotacionados, este simulador é utilizável nos sistemas operativos Ubuntu, MAC OS e Windows. O *Qucs* suporta uma grande lista de componentes analógicos, digitais e também sub-circuitos SPICE e é mais simples do que outros simuladores de circuitos, tais como *gEDA* ou *PSPICE*. Contudo esta aplicação não é a melhor ferramenta de estudo/aprendizagem, visto que apenas entrega os resultados finais.

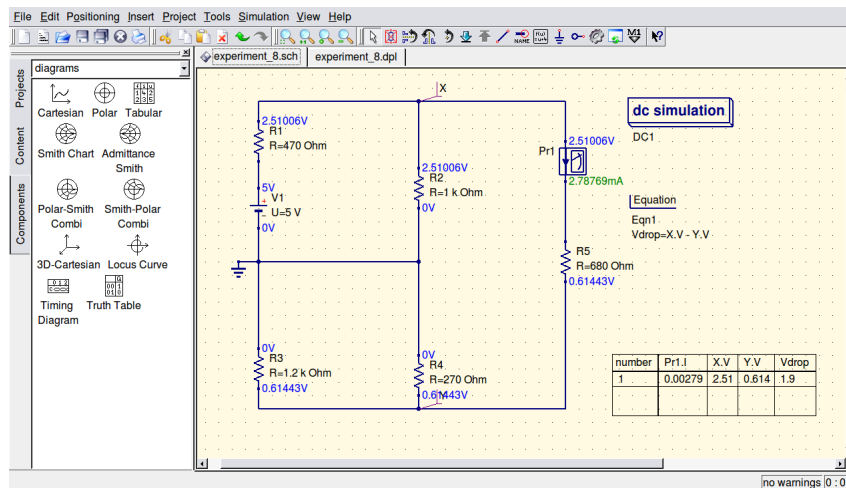


Figura 2.1: Aplicação Qucs

2.2 Falstad

O simulador de circuitos *Falstad* é um simulador de *browser*, logo não é necessário qualquer tipo de *download* para ser utilizado. Tal como o *Qucs*, o *Falstad* também é de fácil utilização e permite que sejam feitas alterações ao circuito mesmo durante a simulação, graficamente é apelativo e é possível analisar o sentido das correntes, visto que uma das características deste software, é a possibilidade de verificar o movimento destas. Infelizmente o *Falstad* só mostra ao utilizador os resultados, não fornecendo a resolução passo a passo do circuito.

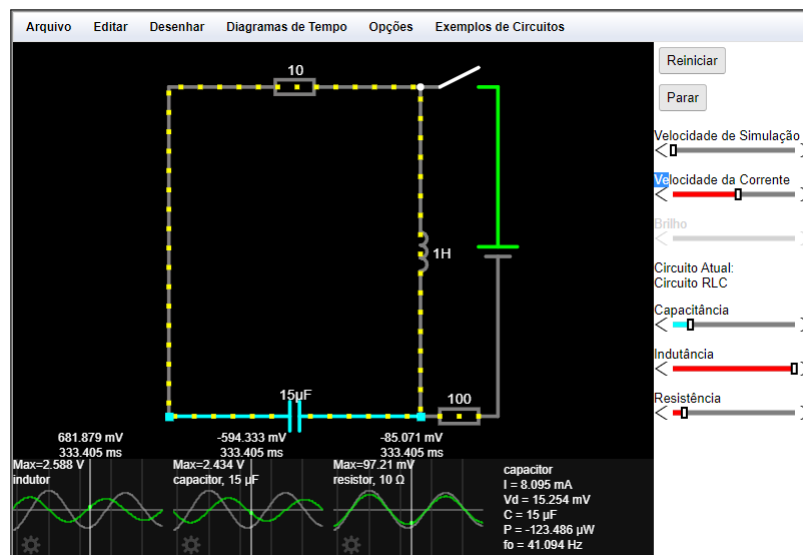


Figura 2.2: Aplicação Falstad

2.3 Comparação com U=RIolve

Tanto U=RIolve, como o *Qucs* e o *Falstad*, são ferramentas de simulação de circuitos. Em termos de facilidade de utilização U=RIolve por si só não funcionaria, uma vez que a edição de circuitos na aplicação ainda não está disponível (à data da realização deste relatório estaria a ser produzido um editor de circuitos para ser introduzido na aplicação U=RIolve, porém ainda não foi terminado) e é necessário recorrer a outra aplicação para obter a Netlist do circuito. U=RIolve peca em termos de complexidade, visto que o número de componentes que são permitidos a ser analisados são muito reduzidos ao contrário dos outros simuladores mencionados (não é possível analisar circuitos com díodos ou tiristores, por exemplo).

Uma característica do U=RIolve é demonstrar ao utilizador não só os resultados, como também uma maneira passo-a-passo de como chegar a eles, o que é fulcral em termos didáticos. Por fim podemos concluir que U=RIolve complementa os demais simuladores devido à funcionalidade didática extra que apresenta.

Capítulo 3

Terminologia e Conceitos Fundamentais

3.1 Terminologia

Alguns dos termos importantes para a compreensão deste relatório são os seguintes:

- **Ramo** – Ramo é um conjunto de um ou mais componentes que normalmente está ligado entre 2 nós como podemos ver na fig, porém é possível ter um circuito sem nós e, neste caso só existe um ramo. [8]

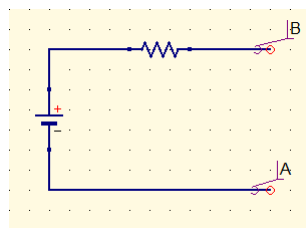


Figura 3.1: Exemplo de um ramo

- **Nó** – Nó é uma junção onde 2 ou mais ramos se conectam é chamada de um nó.[8]

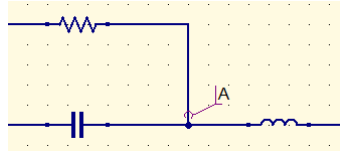


Figura 3.2: Exemplo de um nó

- **Malha** – Malha é um circuito fechado que passa por vários ramos, passando apenas uma vez em cada nó, terminando no nó que começou, nota: não passa em todos os nós do circuito. [8]

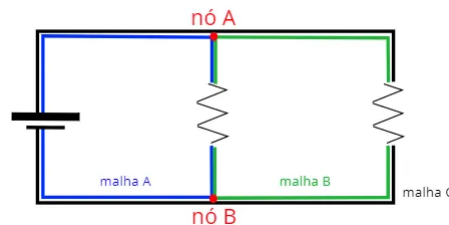


Figura 3.3: Exemplo de malhas [9]

3.2 Método de análise

O método de análise implementado foi o MCR que se baseia na aplicação direta das *Leis de Kirchhoff* das Malhas e dos Nós, segundo um algoritmo bem definido, de forma a sistematizar a sua aplicação em qualquer circuito.

3.2.1 MCR

As incógnitas a determinar são as correntes que percorrem todos os ramos do circuito. Com o valor das correntes determinado, é então possível encontrar o valor da queda de tensão em qualquer componente. O MCR poderá não ser útil quando o circuito contém muitos ramos (visto que quantos mais ramos existirem, maior é o número de incógnitas), porém se estes ramos tiverem fontes de corrente, o número de incógnitas irá diminuir. [10]

A resolução segundo o MCR deve seguir os seguintes passos:

1. Identificar o número de Nós (N), Ramos (R), fontes de corrente ideais (C) e calcular o número de malhas necessárias (M), usando $M = R - C - (N - 1)$;
2. Arbitrar o sentido das incógnitas (correntes);

3. Com base no ponto anterior, escrever $N - 1$ equações linearmente independentes, de acordo com a Lei dos Nós, deixando apenas um nó de fora;
4. Escolher M malhas linearmente independentes, arbitrando o sentido de circulação, ao escolher as malhas deve-se escolher uma primeira malha que não contenha fontes de corrente ideais e a partir daí escolher malhas novas que contenham pelo menos um ramo não utilizado numa malha anterior e que não contenha fontes de corrente ideais também;
5. De acordo com o sentido escolhido anteriormente, escrever M malhas linearmente independentes, de acordo com a Lei das Malhas;
6. Juntar as equações dos nós ($N - 1$) com as equações das malhas (M) e resolver o sistema de R equações.

Nota: Depois de calculadas as correntes, é agora possível calcular as tensões.

3.2.2 Exemplo de resolução MCR

Circuito exemplo:

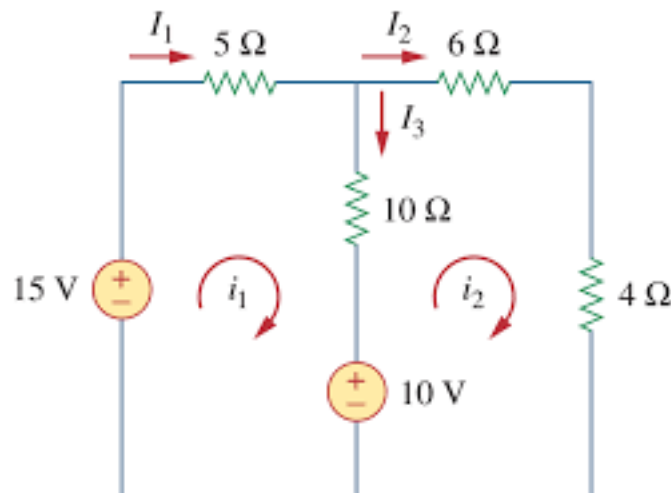


Figura 3.4: Exemplo de um circuito [11]

1. $R=3$, $N=2$, $C=0$, então: $M = R - C - (N - 1)$; $M = 2$;
2. Arbitrar o sentido das correntes como podemos ver na figura 3.4
3. Escolher M malhas linearmente independentes, neste caso 2.
4. Escolhendo o nó superior obtemos a equação: $I_1 + I_2 = I_3$;
5. Equações das malhas:
$$\begin{cases} i_1 : 15 - 10 = I_1 \cdot 5 + I_3 \cdot 10 \\ i_2 : 10 = I_2 \cdot 6 + I_2 \cdot 4 - I_3 \cdot 10 \end{cases}$$
6. Formar o sistema de equações :
$$\begin{cases} 15 - 10 = I_1 \cdot 5 + I_3 \cdot 10 \\ 10 = I_2 \cdot 10 - I_3 \cdot 10 \\ I_1 = I_2 + I_3 \end{cases} \Leftrightarrow \begin{cases} I_1 = 1 \text{ A} \\ I_2 = 1 \text{ A} \\ I_3 = 0 \text{ A} \end{cases}$$

Capítulo 4

Algoritmo

4.1 Enquadramento

A estratégia do desenvolvimento do projeto consistiu em utilizar parte do código já existente (MTN), nomeadamente o início que se trata em retirar os dados necessários da *Netlist* fornecida pelo utilizador, e transformar esta informação num ficheiro *JavaScript Object Notation* (JSON) de maneira a que esta informação possa ser utilizada sempre que necessário.

De seguida, desenvolveu-se o algoritmo do método que utiliza as informações obtidas anteriormente e entrega os dados necessários num ficheiro JSON para realizar os passos seguintes.

Por fim, utiliza-se este último ficheiro JSON para realizar os demais *outputs*.

Na Figura 4.1 podemos observar a arquitetura de todo o projeto realizado e o significado de cada passo.

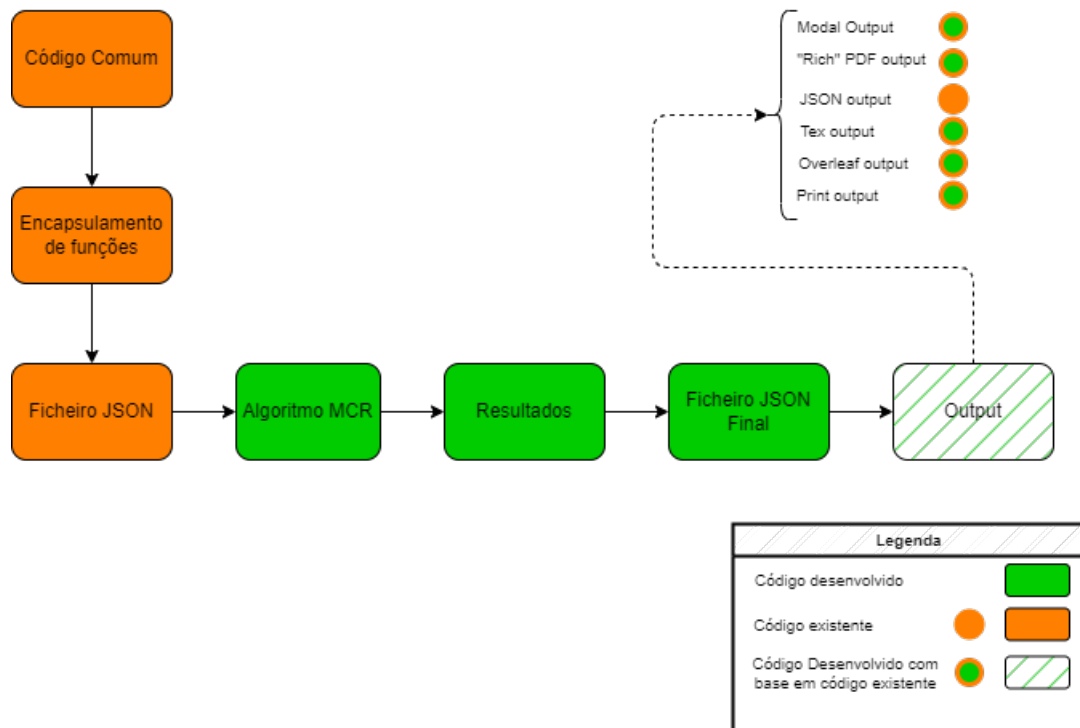


Figura 4.1: Arquitetura do projeto

- **“Código Comum”** – Parte do código existente que é transversal a todos os métodos;
- **“Encapsulamento de funções”** – Alteração do "Código Comum" de maneira a ter a informação no ficheiro JSON necessária a todos os métodos;
- **“Ficheiro JSON”** – Ficheiro com já todos os dados da "Netlist" necessários a realizar os algoritmos dos diferentes métodos;
- **“Algoritmo MCR”** – Referente ao algoritmo necessário que obtém a corrente nos ramos;
- **“Resultados”** – Dados obtidos com a análise do circuito (Algoritmo MCR);
- **“Ficheiro JSON Final”** – O mesmo Ficheiro JSON utilizado anteriormente, porém agora com os dados dos "Resultados";
- **“Output”** – Toda a informação que chega ao utilizador, onde este tem a opção de fazer o que quiser com a mesma, seja só analisar o Modal, fazer o *download* do PDF, fazer o *download* do "ficheiro JSON Final", fazer o *download* do ficheiro "TeX", visualizar o *output* em *Overleaf*, ou mesmo imprimir o PDF do *output*.

4.2 Encapsulamento de funções

Nesta etapa foi desenvolvida pelo colega Ângelo Pinheiro e consistiu em separar o código anterior por funções de maneira ao código ser usado de forma eficiente por todos os métodos.

Estas funções consistem em validar a netlist e imagem enviadas pelo utilizador e criar os dados necessários para a análise, onde são analisados os componentes, verificados os amperímetros, criadas as variáveis sobre os nós e os ramos do circuito.

4.3 Ficheiro JSON

Na Figura 4.2 temos o resultado do encapsulamento em funções do código comum (realizado pelo colega Ângelo Pinheiro) e obtemos dessa maneira todas as variáveis necessárias para desenvolver o algoritmo, sendo o objetivo preencher “*analysisObj*” (explicado com mais detalhe no capítulo 4.4.7). Como é possível observar na figura 4.2 o objeto “*analysisObj*” foi estruturado com toda a informação necessária para serem produzidos os *outputs*, tais como todos os componentes do circuito, a identificação

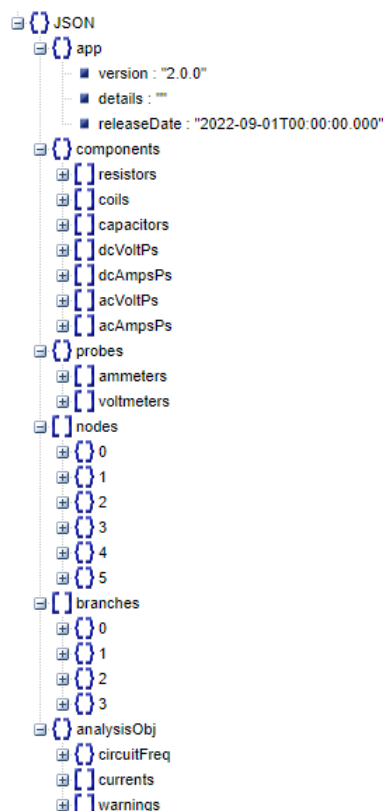


Figura 4.2: Exemplo de um ficheiro JSON com as variáveis iniciais de um circuito

4.4 Algoritmo MCR

Com as variáveis todas determinadas, é desenvolvido o algoritmo para obter a corrente nos ramos. Na Figura 4.3, podemos observar um diagrama de blocos do algoritmo realizado.

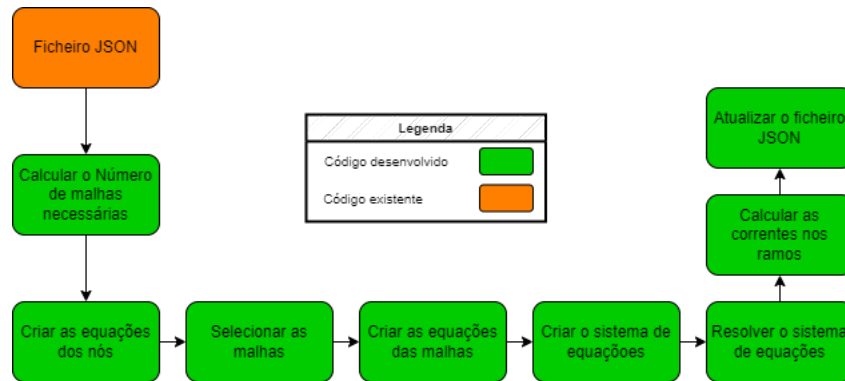


Figura 4.3: Diagrama de blocos do Algoritmo desenvolvido

4.4.1 Calcular o Número de Malhas Necessárias

Para calcular o número de malhas é necessário utilizar a equação $M = R - C - (N - 1)$, sendo R o número de ramos, C o número de fontes de corrente e N o número de nós, estando todos estes valores presentes no Ficheiro JSON.

4.4.2 Criar as Equações dos Nós

O cálculo das $N-1$ equações dos nós começa com o *array* dos nós existente no JSON, segundo o algoritmo presente na Figura 4.4 vai ser sempre desprezado o último nó (por exemplo, se o circuito tiver os nós A, B, C e gnd, vão ser feitas as equações dos nós A, B e C.).

Em seguida, vai ser testado se os ramos ligados no nó têm fonte de corrente, se tiverem, verifica-se o sentido da mesma em comparação com o sentido da corrente arbitrado anteriormente, tendo o mesmo sentido é somado o valor da(s) corrente(s), tendo o sentido contrário o valor é subtraído e guardado numa variável.

Depois de verificadas as fontes de correntes, são analisados os restantes ramos, é feito o mesmo procedimento mas desta vez nada pode ser somado ou subtraído visto que se tratam de variáveis que se quer calcular (correntes), o que acontece é que se o ramo tiver em direção do nó ele é guardado num *array* como estando nesse sentido (“in”), se estiver no sentido contrário (“out”), por fim este *array* com os sentidos vai ajudar a escrever a equação como podemos ver na Figura 4.5. Na Figura temos o fluxograma da formação das equações dos nós.

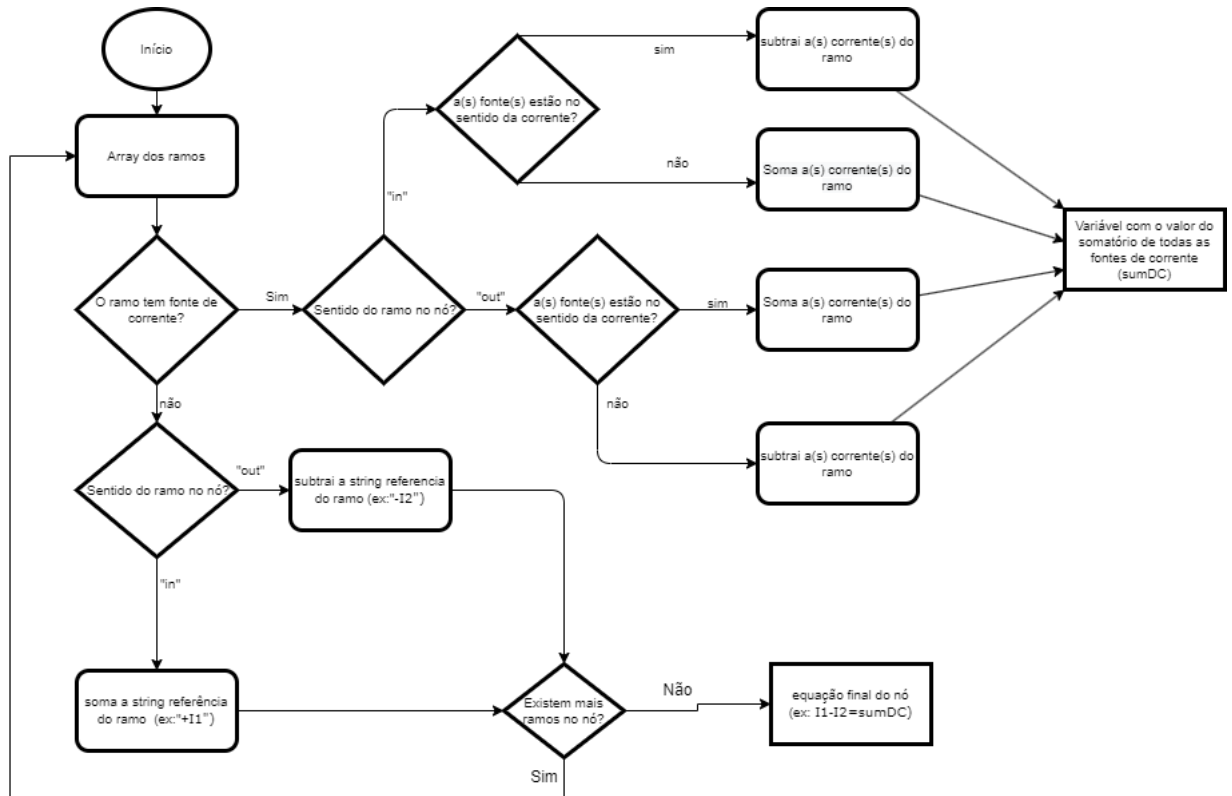


Figura 4.4: Fluxograma da formação das equações dos nós

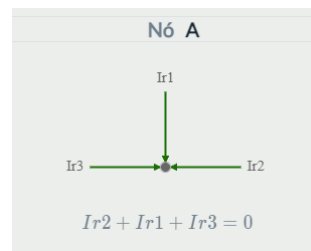


Figura 4.5: Exemplo de uma equação do nó

Nota: Neste caso nenhum ramo continha fonte de corrente e as correntes encontravam-se todas no mesmo sentido, por essa razão, o resultado da soma é igual a zero.

4.4.3 Selecionar as Malhas

Depois de terminadas as equações dos nós, começa a ser feita a escolha das malhas.

Visto que a Netlist não nos transmite a informação de quantas e quais são as malhas possíveis, foi utilizado uma biblioteca com o nome *meshes-finder* [12]. Para utilizar esta biblioteca é necessário fornecer-lhe algumas informações sobre o circuito, a *matriz de adjacência*, a *matriz de incidência*, *array* com referência dos nós e um *array* com referência dos ramos.

- *Matriz de adjacência* – Informa a biblioteca quais os nós que estão ligados entre si (1 se estão ligados, 0 se não).
- *Matriz de incidência* – Informa a biblioteca quais os ramos que estão a incidir em cada nó (1 se incidir, 0 se não).
- *Array dos Nós* – Informa a biblioteca todos os Nós presentes no circuito.
- *Array dos Ramos* – Informa a biblioteca todos os Ramos presentes no circuito.

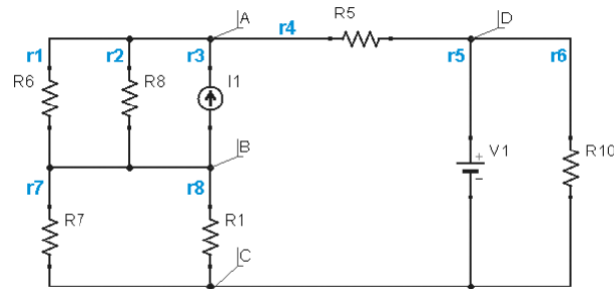


Figura 4.6: Circuito a analisar para matrizes [13]

	A	B	C	D
A	0	1	0	1
B	1	0	1	0
C	0	1	0	1
D	1	0	1	0

Tabela 4.1: Matriz de adjacência

	r1	r2	r3	r4	r5	r6	r7	r8
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

Tabela 4.2: Matriz de incidência

Depois de ter as matrizes e os *arrays*, vão ser utilizadas funções da biblioteca [12], tais como "*MeshesFinder()*" [12], "*initgraph()*" [12] e "*getMeshes()*" [12], o *output* desta última função é um objeto com todas as malhas possíveis do circuito como podemos ver na Figura 4.7

Malhas:
 $M[2][0] = \{r_1, r_2\};$
 $M[2][1] = \{r_1, r_3\};$
 $M[2][2] = \{r_2, r_3\};$
 $M[2][3] = \{r_7, r_8\};$
 $M[2][4] = \{r_5, r_6\};$
 $M[4][0] = \{r_1, r_7, r_5, r_4\};$
 $M[4][1] = \{r_2, r_7, r_5, r_4\};$
 $M[4][2] = \{r_3, r_7, r_5, r_4\};$
 $M[4][3] = \{r_1, r_8, r_5, r_4\};$
 $M[4][4] = \{r_2, r_8, r_5, r_4\};$
 $M[4][5] = \{r_3, r_8, r_5, r_4\};$
 $M[4][6] = \{r_1, r_7, r_6, r_4\};$
 $M[4][7] = \{r_2, r_7, r_6, r_4\};$
 $M[4][8] = \{r_3, r_7, r_6, r_4\};$
 $M[4][9] = \{r_1, r_8, r_6, r_4\};$
 $M[4][10] = \{r_2, r_8, r_6, r_4\};$
 $M[4][11] = \{r_3, r_8, r_6, r_4\};$

Figura 4.7: Objeto com todas as malhas[13]

Depois de ter todas as malhas, é aplicado no código o algoritmo de escolha das malhas necessárias, com os seguintes *arrays*.

M1	...	Mn
0	...	0

Tabela 4.3: *Array* das malhas

Nota:

- 0 = Malha não utilizada;
- 1 = Malha escolhida;
- 2 = Malha com fonte de corrente ou sem ramo novo;

r1	...	rn
0	...	0

Tabela 4.4: *Array* dos ramos

Nota:

- 0 = Ramo não utilizado sem fonte;
- 1 = Ramo com fonte;
- 2 = Ramo utilizado;

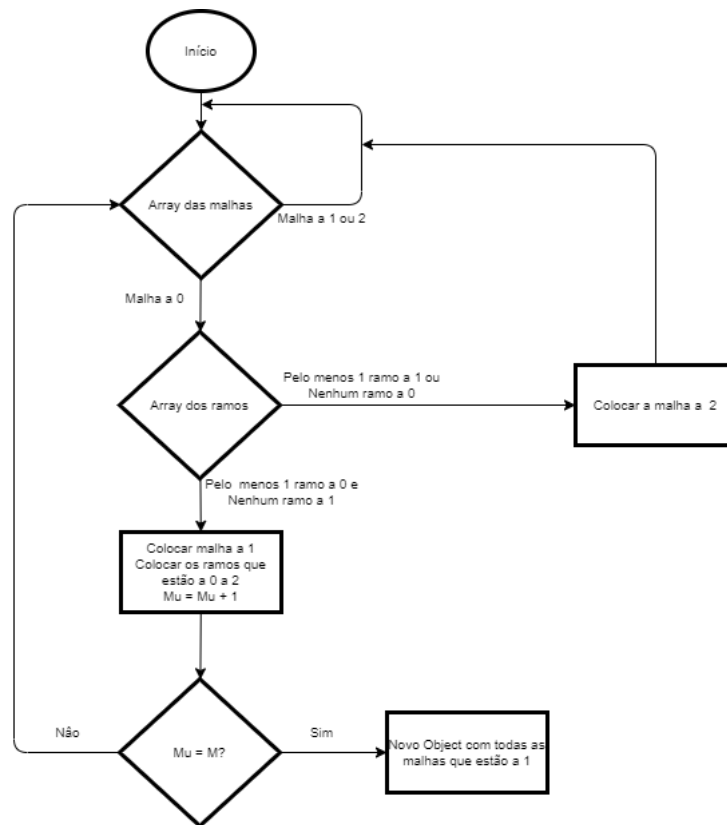


Figura 4.8: Fluxograma da escolha das malhas

Nota:

- Mu é uma variável que diz quantas malhas já foram utilizadas, quando $Mu = M$ o algoritmo pára.
- Como o *output* "*meshes-finder*"[12] entrega a lista das malhas possíveis por ordem crescente do número de ramos que contém, as malhas com menos ramos iram ser favorecidas visto que serão as primeiras a ser testadas.

4.4.4 Criar as Equações das Malhas

Para podermos ter as equações das malhas, é preciso primeiro arbitrar o sentido das mesmas, com o *output* da biblioteca "*meshes-finder*"[12] obtém-se não só todas as malhas possíveis mas também todos os ramos que lhe pertencem. Com esta informação, o passo seguinte é escolher o sentido. O sentido escolhido foi o mesmo que o do primeiro ramo que aparecer na *string* da malha, ou seja se usarmos o exemplo da Figura 4.9 uma das malhas escolhidas é [Ir2, Ir5, Ir1] e o primeiro ramo presente é o ramo com corrente Ir2 logo a malha vai ter o mesmo sentido que este ramo.

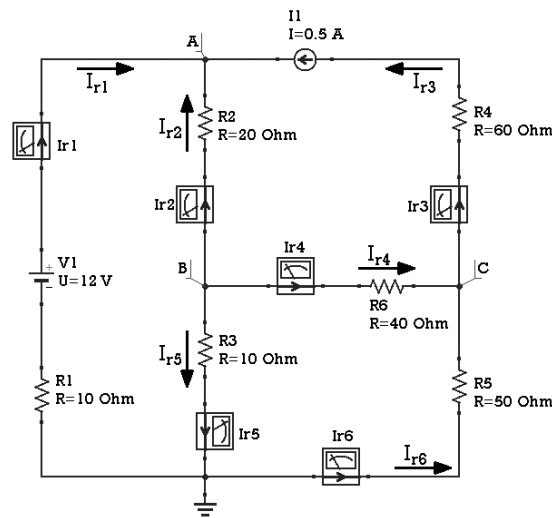


Figura 4.9: Exemplo de um circuito DC

Depois de decidido o sentido da malha, pode-se começar a fazer a equação da mesma. Vão ser construídas duas equações ao mesmo tempo, uma com todos os componentes para apresentar no output ("*equationReal*") e outra substituindo os componentes pelos valores deles deixando apenas as variáveis (correntes). Esta última equação vai servir tanto para o output como para fazer os cálculos finais ("*equationsRevealed*")

Utilizando o mesmo exemplo:

- O ramo com a corrente I_{r2} apresenta apenas uma resistência de 20 ohm logo fica da seguinte forma:

```
1 equationReal = R2 * Ir2
2 equationsRevealed = (20) * Ir2
```

Listagem 4.1: Início das equações

- De seguida, vem o ramo com a corrente I_{r5} (sentido da corrente contrário ao da malha) que contém na mesma uma resistência:

```
1 equationReal = R2 * Ir2 - R3 * Ir5
2 equationsRevealed = (20) * Ir2 - (10) * Ir5
```

Listagem 4.2: Equações com 2 ramos

- Por último, temos o ramo com a corrente IR1 (também com sentido da corrente contrário ao da malha) que contém também uma resistência, porém já é visível uma fonte de tensão e neste caso, estando ela no sentido contrário da malha, podemos subtrair o valor dela ao valor total da soma de todas as fontes de tensão somadas anteriormente (neste caso como é a primeira o valor é 0), e ficam então as equações assim:

```

1      equationReal = R2 * Ir2 - R3 * Ir5 - R1 * Ir1 = - V1
2  equationsRevealed = (20)*Ir2 - (10)*Ir5 - (10)*Ir1 = -12

```

Listagem 4.3: Equações finais

Neste caso os ramos só tinham um componente de carga cada e a simulação é em DC, porém se a simulação fosse em AC e tivesse múltiplos componentes, seria feita o cálculo da impedância equivalente do ramo e o valor seria apresentado em números complexos.

4.4.5 Criar o Sistema de Equações

Depois de formadas as equações das malhas e equações dos nós, já obtivemos todas as equações necessárias para criar o sistema, assim sendo este passo apenas junta todas as equações num só objeto.

4.4.6 Resolver o Sistema de Equações

Quando se obtiver o sistema de equações, vai-se utilizar a biblioteca “*syseqsolver*”[14] para resolver o mesmo. Esta biblioteca não aceita que as variáveis tenham qualquer nome, então é preciso fazer uma pequena mudança nas equações sendo que as variáveis que a biblioteca aceita são as letras do abecedário, tendo em atenção se a letra é maiúscula ou minúscula, visto que aceita as duas porém ficam 2 variáveis distintas :

```
1 Antes = (20)*Ir2 - (10)*Ir5 - (10)*Ir1 = -12
2 Depois = (20)*a - (10)*b -(10)*c = -12
```

Listagem 4.4: Nova Equação

Com as equações alteradas já é possível começar a utilizar as funções da biblioteca, mais precisamente “*addEquation*”[14] que adiciona as equações que nós alteramos ao sistema, “*buildSystem*”[14] que cria o sistema de equações com as equações anteriores e finalmente a função “*solve*”[14] que resolve o sistema e retorna os resultados num objeto.

4.4.7 Atualizar o Ficheiro JSON

O ficheiro JSON é atualizado com as novas informações obtidas através da análise, estas informações vão atualizar o campo “*analysisObj*”. Depois de atualizado, este campo contém as malhas escolhidas e toda a sua informação “*chosenmeshes*”, a frequência do circuito “*circuitFreq*”, as correntes dos ramos “*currents*”, todas as equações utilizadas “*equations*”, o número de equações das malhas “*numMeshEquations*”, o número de equações dos nós “*numNodeEquations*”, os resultados das correntes nos ramos “*result*”, o número total de malhas do circuito “*totalMeshes*” e por fim os avisos que deverão aparecer no output “*warnings*”.

No fim da análise ao circuito da Figura 4.9, o campo “*analysisObj*” do ficheiro JSON encontra-se assim:

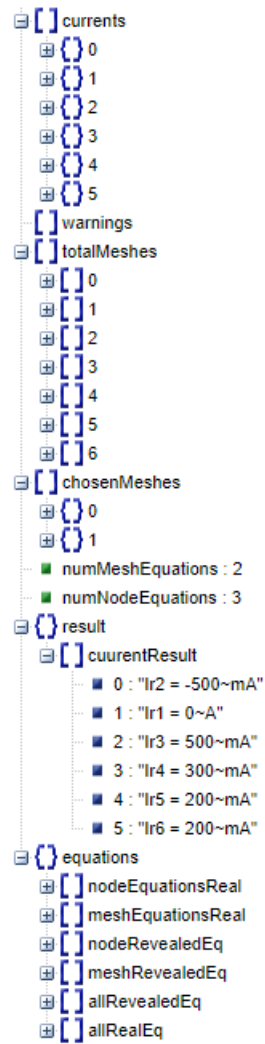


Figura 4.10: Campo "analysisObj" do ficheiro JSON final

4.5 Output

O *output* inicial é o *Modal*¹, que consiste numa janela flutuante sobreposta à página inicial que contém toda a informação necessária à análise do circuito. Porém guardar a informação da análise é complicado, então foram adicionados mais dois métodos aos três já existentes de retirar e guardar a informação, de momento é possível utilizar o “*Download PDF*”, “*Download JSON*”, “*Download TeX*”, “*Open in Overleaf*”, “*Print*”.

4.5.1 Estrutura dos Outputs

Todos os *outputs* contém a mesma informação, visto que derivam do ficheiro JSON final, e a sua estrutura é a seguinte:

- **Capa** – A capa é igual em todos os *outputs* e apresenta o logo da aplicação, os coordenadores do projeto, os desenvolvedores do projeto, o *abstract* do *output* e a data em que o mesmo foi construído.
- **imagem do circuito** – A imagem do circuito é apresentada (se submetida)
- **Variáveis do circuito** – Aqui é apresentado o número de ramos, nós, fontes de corrente e fontes de tensão isoladas.
- **Informação do circuito** – Aqui é exposto o tipo de simulação, a frequência do circuito (se a simulação for AC) e o número de amperímetros presente no circuito.
- **Arbitragem da Corrente nos Ramos** – Onde estão os IDs dos ramos, o seu sentido e componentes.
- **Equações das Correntes** – Aqui estão desenhados os nós e as suas equações.
- **Número de equações das malhas** – Onde é apresentado o cálculo do número de malhas necessárias.
- **Malhas selecionadas** – Neste setor estão desenhadas todas as malhas escolhidas e está escrito as suas equações.
- **Sistema de equações** – Sistema de equações final e os passos para chegar ao mesmo.
- **Resultado** – Por fim é apresentado o valor das correntes nos ramos.

Esta estrutura aplica-se a todos os métodos de output, exceto ao “*Download JSON*”, visto que este apenas vai fazer o download do ficheiro JSON final. O modal Output apresenta esta estrutura com a exceção da capa.

¹A Modal é uma janela popup que é exibida sobre a página atual

4.5.2 Modal Output

O Modal *output* é o *output* principal da aplicação e ele é utilizado pelo facto de conter toda a informação necessária à análise de circuitos e por demonstrar a mesma de uma maneira simples ao utilizador.

Para o desenvolvimento deste método foi utilizada a biblioteca “jQuery”[15] que faz com que a manipulação de documentos HTML se torne bem mais simples, foi também utilizado a *Framework “bootstrap”* [16] que tem um sistema de tabelas/-quadros facilitando não só a edição do output, como também a leitura e interação do mesmo nos mais diversos dispositivos sejam estes computadores, *smartphones* ou *tablets*.

4.5.3 Download PDF Output

O “*download PDF*” *Output* permite ao utilizador fazer o *download* de um *Portable Document Format* (PDF) com toda a informação presente no Modal output (com a adição de uma capa). Este *output* é produzido com base em L^AT_EX[17] e para criar o ficheiro foi utilizada a biblioteca *latexprinter*[18].

Para criar o ficheiro, é necessário uma *string* com a informação toda para construir o PDF, depois de ter o texto, é necessário adicionar as imagens das malhas e dos nós e isto é feito recorrendo à função “*addImgFile()*” existente na biblioteca *latexprinter*[18], com a imagem adicionada ao ficheiro é necessário adicionar a mesma à *string*. Por último, utiliza-se a função “*print()*” que cria o documento PDF e inicia a transferência do mesmo pela parte do utilizador.

4.5.4 Download JSON

Este método consiste no *download* de um ficheiro de texto com todos os dados do ficheiro JSON final, depois de transferido é possível observá-lo de uma maneira mais humana em <http://jsonviewer.stack.hu/>.

Para isso foi criado um elemento *HyperText Markup Language* (HTML) e o seu atributo sendo a *string* "data:text/plain;charset=utf-8," mais o ficheiro JSON final numa *string*. Depois é utilizada a função “*click()*” que inicia a transferência.

Este *output* é fornecido pela possível interação com outras aplicações e não para a leitura humana propriamente dita.

4.5.5 Download TeX

Este método inicializa uma transferência de dois ficheiros de texto por parte do utilizador, um com a *string* codificada em L^AT_EX e outra com as imagens utilizadas. Para a criação desta *string* foram utilizadas as mesmas funções que foram criadas para o *output Open in Overleaf*.

```
1 \newcommand{\inlineimages}[2]{
2 \newwrite\tempfile
3 \immediate\openout\tempfile=#1.base64
4 \immediate\write\tempfile{#2}
5 \immediate\closeout\tempfile
6 \immediate\write18{base64 -d #1.base64 > #1}
7 \includegraphics{#1}
8 }
```

Listagem 4.6: Código desenvolvido pelo Engenheiro André Rocha

4.5.7 Print

Quando o método “*print*” é utilizado, ele abre uma nova janela no browser com um documento PDF e é aberta também uma janela de impressão para o utilizador guardar a informação dessa forma.

Neste método foi utilizada uma biblioteca *jsPDF*[20] que é utilizada para criar ficheiros PDF em *JavaScript*, isto fez com que o ficheiro se parece um bocado diferente dos demais, visto que foi o único em que foi utilizado o *jsPDF*[20], com isto ele é mais rápido a ser construído em comparação ao “*Download PDF*” e ao “*Open in Overleaf*” visto que o PDF é construído instantaneamente, porém tem um pouco menos de qualidade do que os demais.

Para criar o PDF é necessário primeiro determinar a formatação do documento, escolhendo os valores de margens, tamanho do texto do título e dos subtítulos utilizados, e depois é utilizada a variável “line” com a informação da posição vertical no momento dentro de uma página. Esta variável é muito importante porque nos permite que não haja sobreposição no documento, e com ela conseguimos saber se a página tem espaço suficiente para escrever algo que seja necessário de maneira a não haver quebras, no caso de não haver espaço é possível passar à página seguinte com a função “*addPage()*”.

O texto em si é introduzido utilizando as seguintes funções “*.text()*”, “*.printEquation()*” e “*.addImage()*” que como o nome indica, adiciona as imagens ao ficheiro.

É então criada a função “*buildPrintPDFMCR()*” com o ficheiro JSON final e as imagens que contém todas as outras funções da edição de texto.

Capítulo 5

Conclusões

Este projeto começou com o estudo das tecnologias necessárias à realização do mesmo e que já estavam presentes na arquitetura da aplicação tais como javascript e HTML. De seguida foi feito um estudo sobre todo o método a ser inserido na aplicação, o método MCR, de maneira a que este tivesse dominado e que não houvesse dúvidas sobre o mesmo com o decorrer da construção dos algoritmos a serem utilizados, tendo sido a conceção e implementação destes um dos pontos fundamentais e mais importantes na realização do projeto. Alguns dos exemplos destes algoritmos estão presentes na Figura 4.4 e 4.8.

Outra fase importante na realização do projeto foi a criação de todos os *outputs*, visto que isto é uma grande parte das funcionalidades do utilizador, para isto foram reutilizadas algumas funções já presentes na aplicação (do método MTN) e outras criadas pelo colega Ângelo pinheiro (realizou o projeto de implementação do método MCM) e também foram criadas funções para implementar os desejados *outputs*.

Durante todo o projeto foram realizados vários testes à capacidade do algoritmos e do código criado de maneira a melhorar a capacidade dos mesmos.

Por último é importante salientar que é possível melhorar não só a qualidade como aumentar o número de funcionalidades existentes, estando alguns exemplos no capítulo 5.1

5.1 Trabalho Futuro

Depois do projeto estar concluído é possível olhar para a aplicação e apontar algumas melhorias e novas funcionalidades a ser introduzidas na mesma:

- Fazer certas perguntas ao utilizador antes de ser apresentados os resultados finais, tais como o número de ramos, nós e malhas necessárias (algo fundamental para a resolução de análises).
- Apresentar todas as malhas possíveis e deixar o utilizador escolher as que quiser.
- Permitir ao utilizador a escolha dos nós a serem utilizados.

Referências

- [1] J. Hammersley and J. Lees-Miller, “Overleaf.” Available at <https://www.overleaf.com/>. (Último acesso em 03/09/2022). [Citado nas páginas i, iii e 25]
- [2] electricalbaba, “Ideal current source & practical current source explained.” Available at <http://electricalbaba.com/>, 2020. (Último acesso em 01/09/2022). [Citado na página xi]
- [3] R. Ranjan, “What is a framework in programming & why you should use one.” Available at <https://www.netsolutions.com>. (Último acesso em 03/09/2022). [Citado na página xi]
- [4] jaime Santos, ed., *Análise de Circuitos Elétricos*. Publindústria. [Citado na página xi]
- [5] L. M. C. Sousa, “U=risolve: a web-based tool for teaching & self-learning the nodal voltage method,” Master’s thesis, Instituto de Engenharia do Porto, Porto, 2000. [Citado na página 2]
- [6] M. Margraf and S. Jahn, “Qucs.” Available at <http://qucs.sourceforge.net/>. (Último acesso em 03/09/2022). [Citado na página 3]
- [7] P. Falstad, “Falstad.” Available at <http://www.falstad.com/circuit>. (Último acesso em 03/09/2022). [Citado na página 3]
- [8] W. McAllister, “circuit terminology.” Available at <http://www.khanacademy.org/>, 2016. (Último acesso em 01/09/2022). [Citado nas páginas 7 e 8]
- [9] R. Helerbrock, “Leis de kirchhoff.” Available at <http://brasilescola.uol.com.br/>, 2022. (Último acesso em 01/09/2022). [Citado nas páginas vii e 8]
- [10] A. Rocha, L. Sousa, M. Alves, and F. Pereira, “Métodos gerais de análise de circuitos,” fevereiro 2022. [Citado na página 8]
- [11] C. K. Alexander and M. N. Sadiku, eds., *Solutions Manual Fundamentals of Electric Circuits*. McGraw-Hill, 2012. [Citado nas páginas vii e 9]
- [12] A. Rocha, “meshes-finder.” Available at <https://github.com/urisode/meshes-finder>. (Último acesso em 01/09/2022). [Citado nas páginas 15, 16 e 18]

-
- [13] A. Rocha, “meshes-finder algorithm.” (Último acesso em 01/09/2022). [Citado nas páginas vii, 16 e 17]
 - [14] A. Rocha, “syseqsolver.” Available at <https://github.com/txroot/syseqsolver>. (Último acesso em 01/09/2022). [Citado na página 21]
 - [15] J. Resig, “jquery.” Available at <https://jquery.com>. (Último acesso em 02/09/2022). [Citado na página 24]
 - [16] M. Otto and J. Thornton, “Bootstrap.” Available at <https://getbootstrap.com>. (Último acesso em 03/09/2022). [Citado na página 24]
 - [17] L. Lamport, “Latex – a document preparation system.” Available at <https://www.latex-project.org/>. (Último acesso em 03/09/2022). [Citado na página 24]
 - [18] A. Rocha, “Print latex to pdf.” Available at <https://github.com/txroot/latexprinter>. (Último acesso em 03/09/2022). [Citado na página 24]
 - [19] A. Rocha, “Documento de trabalho.” [Citado na página 25]
 - [20] Parallax, “jspdf.” Available at <https://parall.ax/products/jspdf>. (Último acesso em 03/09/2022). [Citado na página 26]

Anexo A

Análise de Circuitos Exemplo

A.1 Circuito em Corrente Contínua

A.1.1 Imagem do Circuito

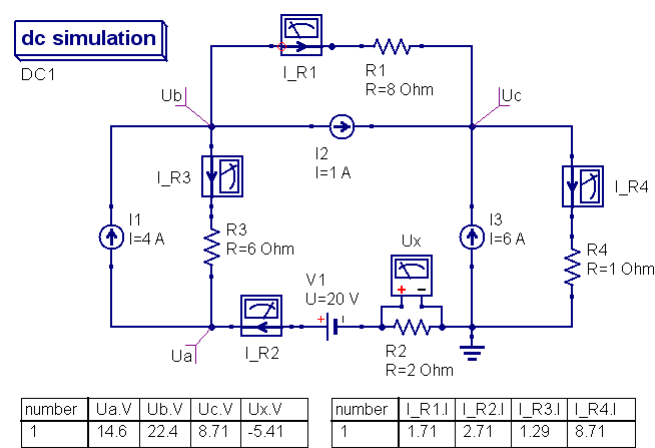


Figura A.1: Imagem do exemplo a ser analisado em corrente contínua

A.1.2 Netlist do Circuito

```

1  Idc:I3 Uc gnd I="6 A"
2  Idc:I2 Uc Ub I="1 A"
3  R:R4 gnd _net0 R="1 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="
    26.85"
4  IProbe:I_R4 Uc _net0
5  R:R1 _net1 Uc R="8 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="
    26.85"
6  IProbe:I_R1 Ub _net1
7  IProbe:I_R3 Ub _net2
8  R:R3 Ua _net2 R="6 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="
    26.85"
9  Idc:I1 Ub Ua I="4 A"
10 IProbe:I_R2 _net3 Ua
11 R:R2 _net4 gnd R="2 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="
    26.85"
12 VProbe:Ux _net4 gnd
13 Vdc:V1 _net3 _net4 U="20 V"
14 .DC:DC1 Temp="26.85" reltol="0.001" abstol="1 pA" vntol="1 uV"
    saveOPs="no" MaxIter="150" saveAll="no" convHelper="none"
    Solver="CroustLU"

```

Listagem A.1: Netlist do circuito a ser analisado em corrente alternada

A.1.3 Output em PDF

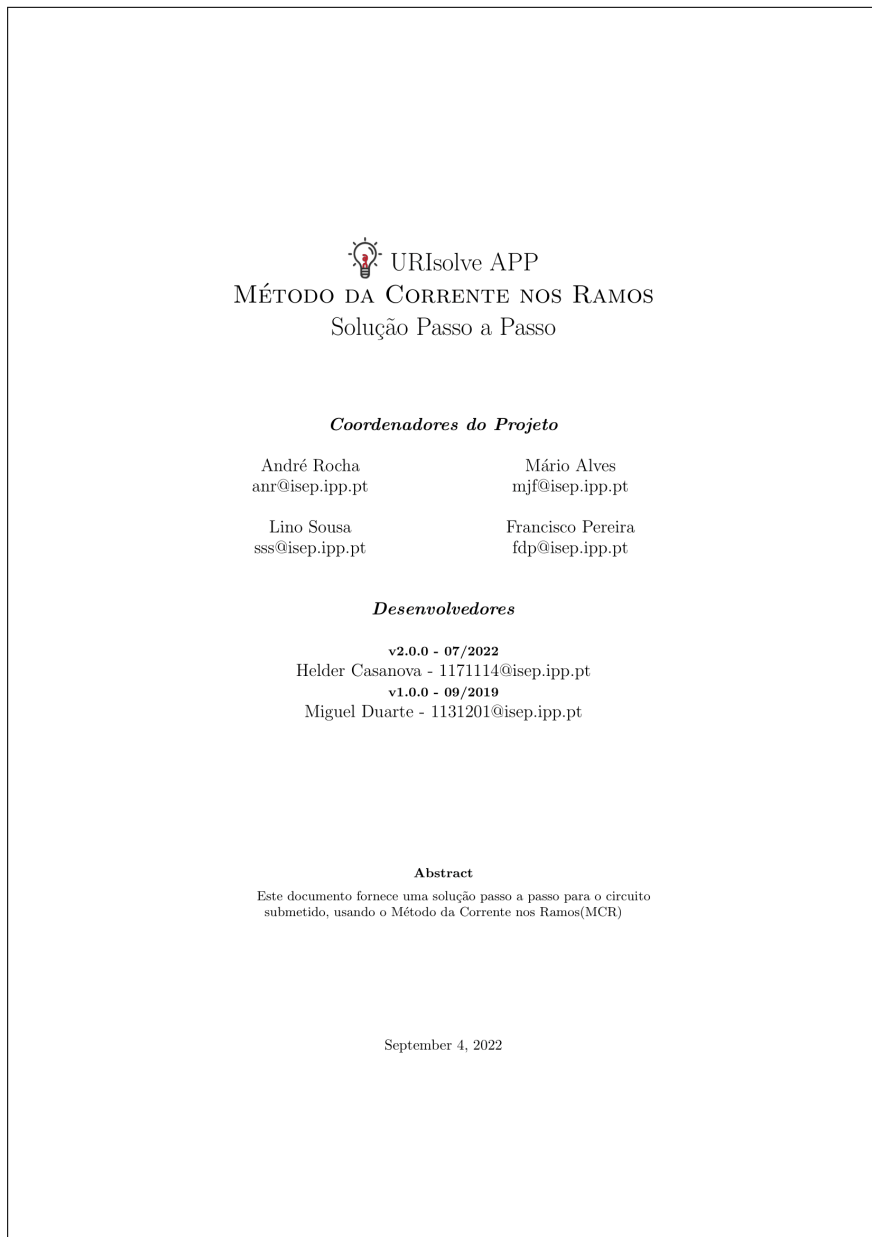


Figura A.2: Capa do Output PDF corrente contínua

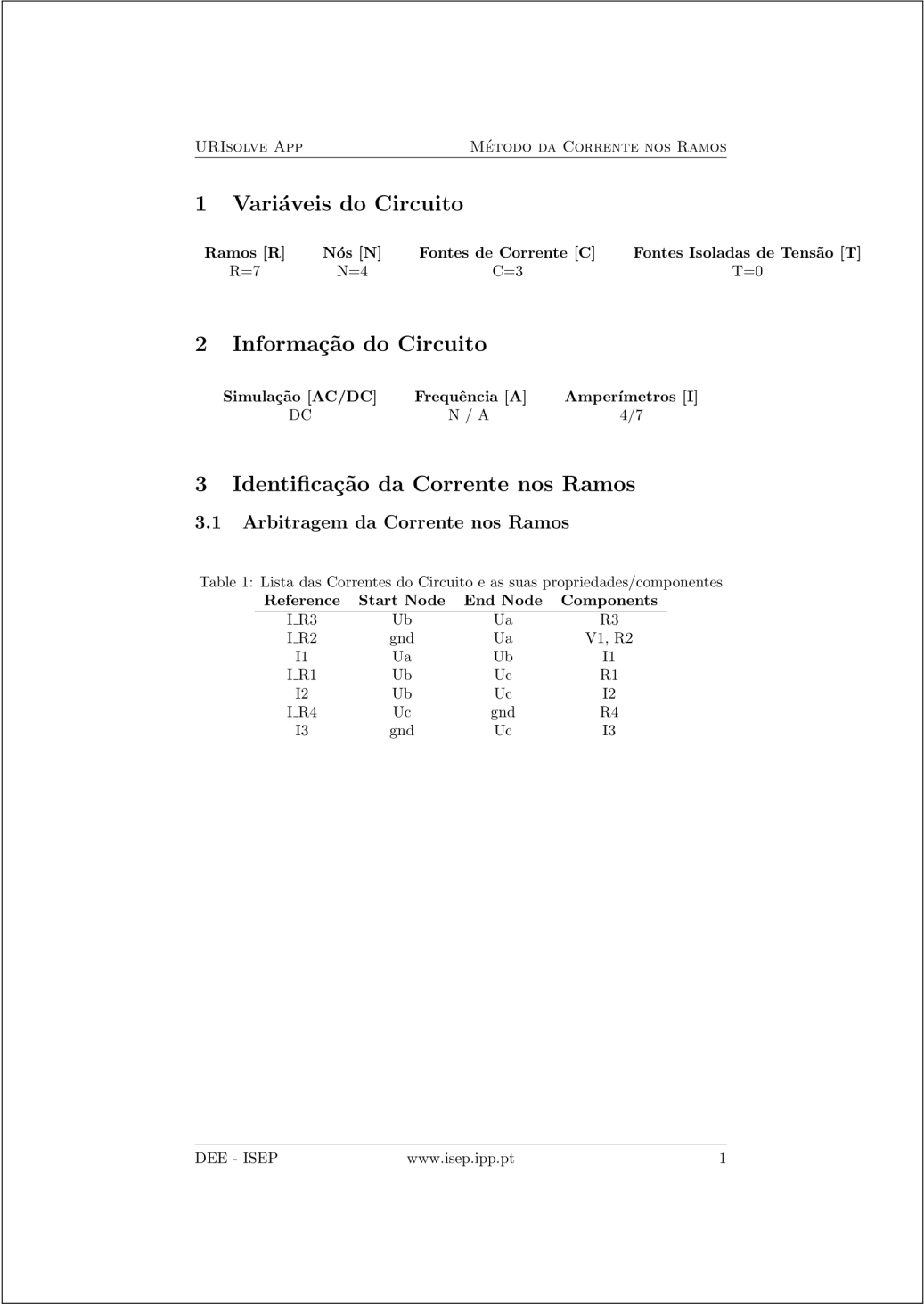
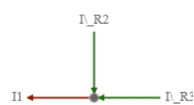


Figura A.3: Pág 1 do Output PDF corrente contínua

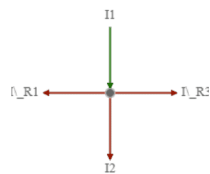
4 Equações das Correntes (Lei de Kirchhoff dos Nós - LKN)

4.1 Nó Ua



$$\text{Equação 1 : } -I_{R3} - I_{R2} = -I1 \quad (1)$$

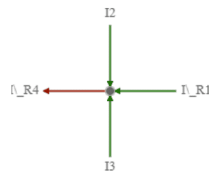
4.2 Nó Ub



$$\text{Equação 2 : } I_{R3} + I_{R1} = +I1 - I2 \quad (2)$$

Figura A.4: Pág 2 do Output PDF corrente contínua

4.3 Nó Uc



Equação 3: $I_{R4} - I_{R1} = +I2 + I3$ (3)

5 Número de (equações das) Malhas necessárias

5.1 Cálculo número de malhas necessárias

$$\begin{aligned} M &= R - (N - 1) - C \Leftrightarrow \\ M &= 7 - (4 - 1) - 3 \Leftrightarrow \\ &\Leftrightarrow M = 1 \end{aligned}$$

Figura A.5: Pág 3 do Output PDF corrente contínua

6 Malhas seleccionadas e equações correspondentes

6.1 Malha 1



Equation : $IM1 : R_3 * I_{R3} - R_1 * I_{R1} - R_4 * I_{R4} - R_2 * I_{R2} = -V_1$ (4)

Figura A.6: Pág 4 do Output PDF corrente contínua

URISOLVE APP

MÉTODO DA CORRENTE NOS RAMOS

7 Sistema de Equações

Equações:

$$\begin{cases} (6) * I_{R3} - (8) * I_{R1} - (1) * I_{R4} - (2) * I_{R2} = -20 \\ -I_{R3} - I_{R2} = -4 \\ I_{R3} + I_{R1} = 3 \\ I_{R4} - I_{R1} = 7 \end{cases}$$

Passos:

Step 1: Sistema de equações inicial

$$\begin{cases} -I_{R3} - I_{R2} = -I1 \\ I_{R3} + I_{R1} = +I1 - I2 \\ I_{R4} - I_{R1} = +I2 + I3 \\ R3 * I_{R3} - R1 * I_{R1} - R4 * I_{R4} - R2 * I_{R2} = -V1 \end{cases}$$

Step 2: Substitua o valor dos componentes do circuito

$$\begin{cases} (6) * I_{R3} - (8) * I_{R1} - (1) * I_{R4} - (2) * I_{R2} = -20 \\ -I_{R3} - I_{R2} = -4 \\ I_{R3} + I_{R1} = 3 \\ I_{R4} - I_{R1} = 7 \end{cases}$$

Figura A.7: Pág 5 do Output PDF corrente contínua

8 Valores das correntes dos ramos

$$\begin{cases} I_{R3} = 1,29 \text{ A} \\ I_{R2} = 2,71 \text{ A} \\ I_1 = 4 \text{ A} \\ I_{R1} = 1,71 \text{ A} \\ I_2 = 1 \text{ A} \\ I_{R4} = 8,71 \text{ A} \\ I_3 = 6 \text{ A} \end{cases}$$

Figura A.8: Pág 6 do Output PDF corrente contínua

A.2 Circuito em Corrente Contínua com Fonte de Corrente

A.2.1 Imagem do Circuito

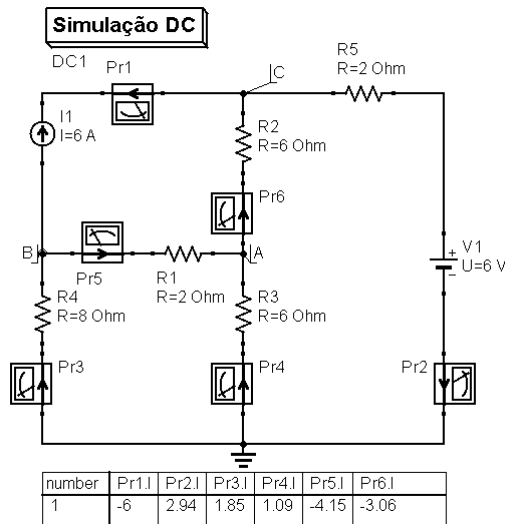


Figura A.9: Imagem do exemplo a ser analisado em corrente Contínua com fonte de Corrente

A.2.2 Netlist do Circuito

```

1  Vdc:V1 _net0 _net1 U="6 V"
2  R:R3 _net2 A R="6 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="
    26.85"
3  IProbe:Pr4 gnd _net2
4  IProbe:Pr2 _net1 gnd
5  R:R5 _net0 C R="2 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="
    26.85"
6  R:R2 C _net3 R="6 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="
    26.85"
7  R:R4 _net4 B R="8 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="
    26.85"
8  Idc:I1 _net5 B I="6 A"
9  IProbe:Pr1 C _net5
10 R:R1 _net6 A R="2 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="
    26.85"
11 IProbe:Pr5 B _net6
12 IProbe:Pr6 A _net3
13 .DC:DC1 Temp="26.85" reltol="0.001" abstol="1 pA" vntol="1 uV"
    saveOPs="no" MaxIter="150" saveAll="no" convHelper="none"
    Solver="CROUTLU"
14 IProbe:Pr3 gnd _net4

```

Listagem A.2: Netlist do Circuito a ser analisado em corrente contínua com fonte de corrente

A.2.3 Output em PDF

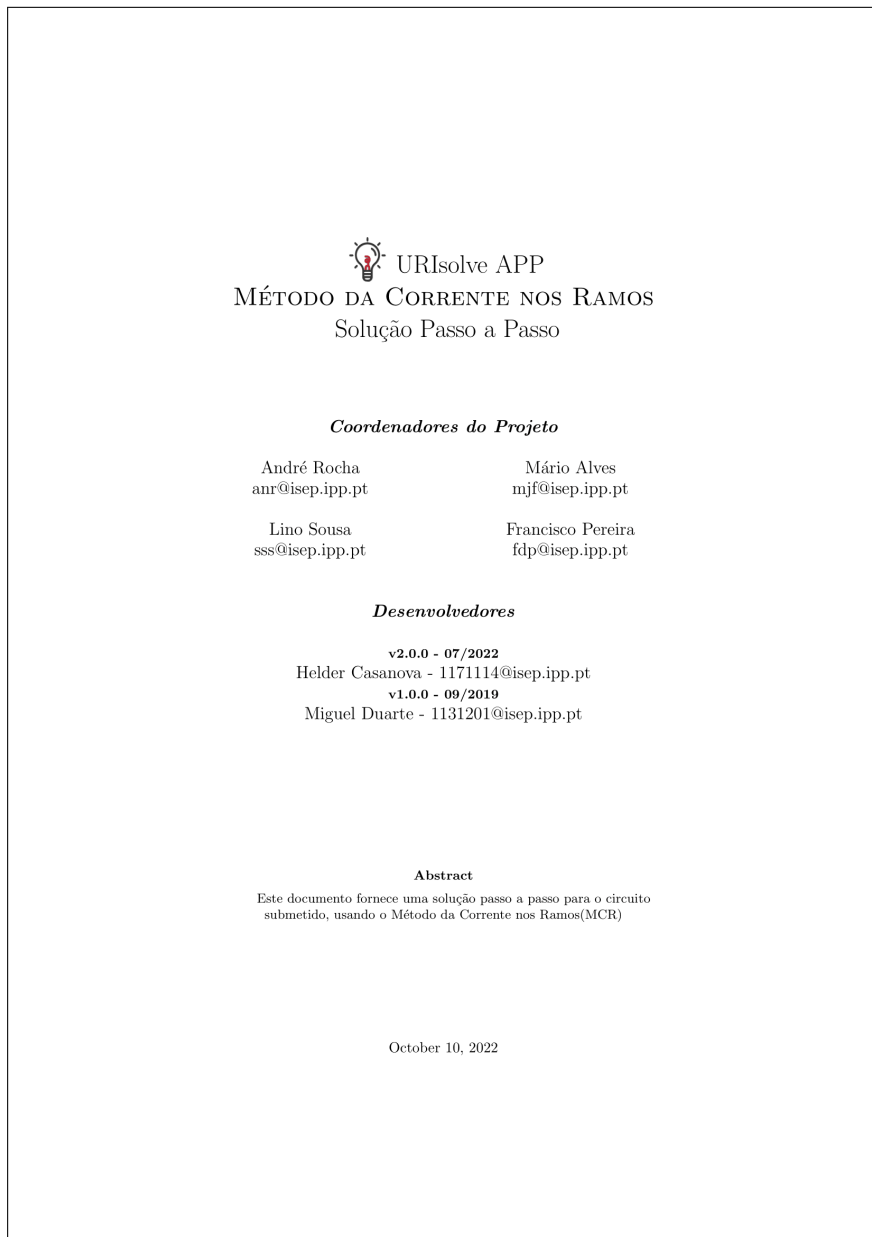


Figura A.10: Capa do Output PDF corrente contínua com fonte de corrente

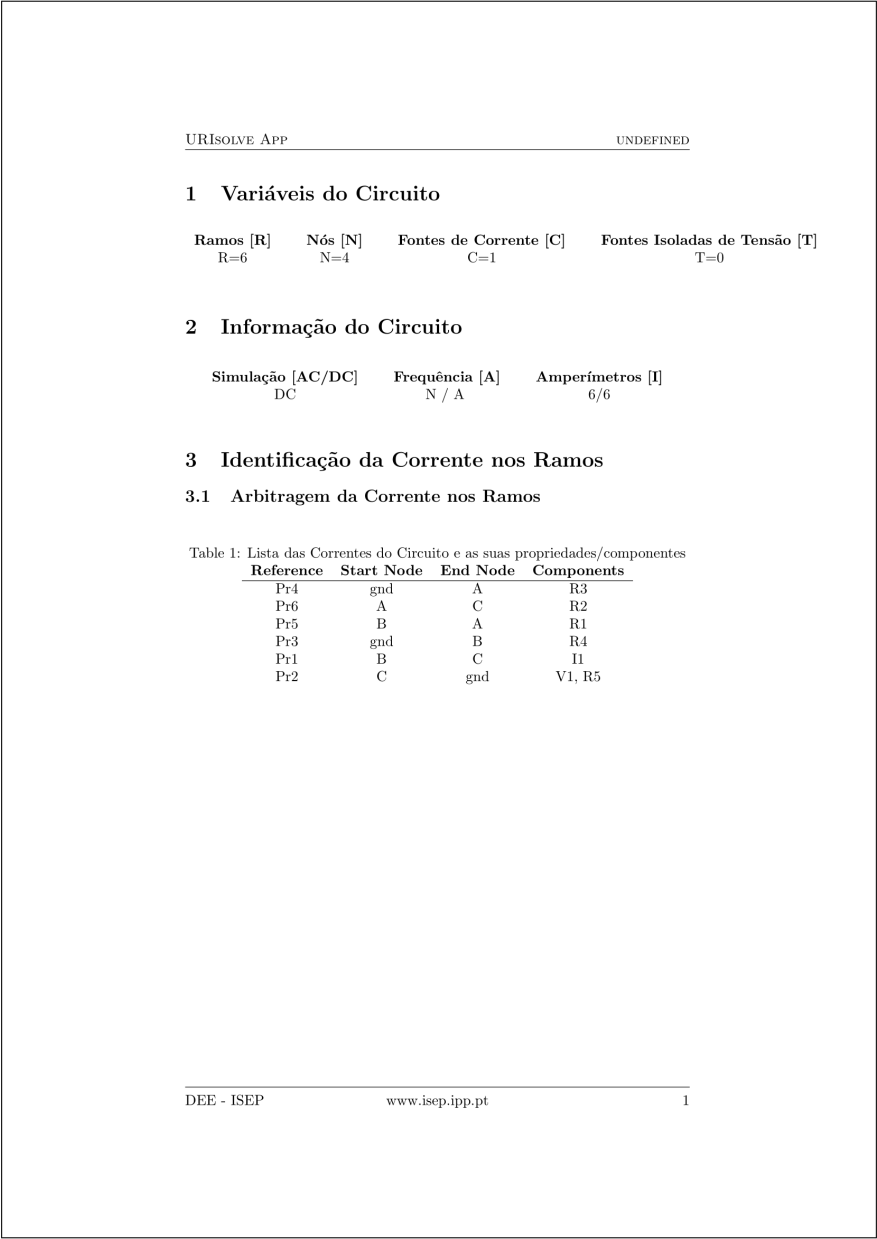


Figura A.11: Pág 1 do Output PDF corrente contínua com fonte de corrente

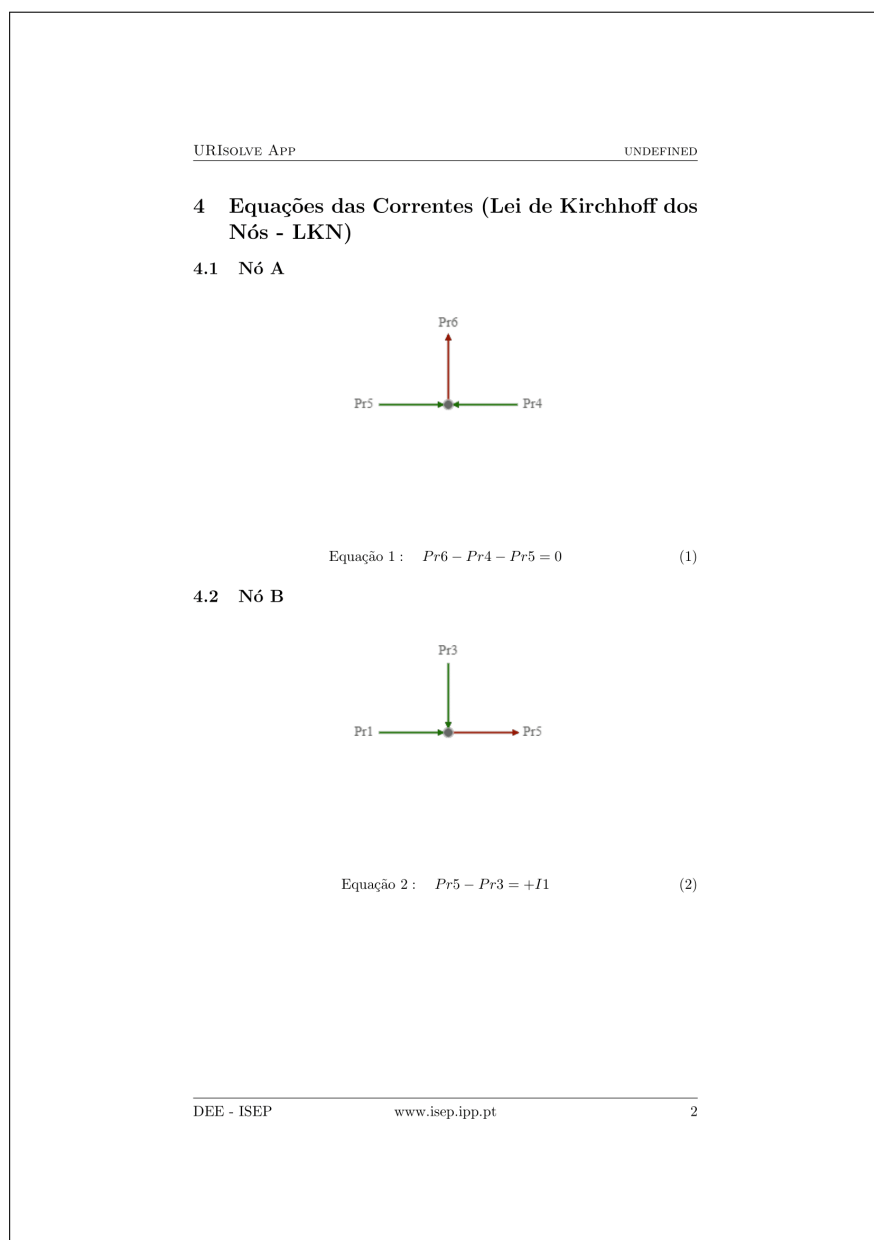


Figura A.12: Pág 2 do Output PDF corrente contínua com fonte de corrente

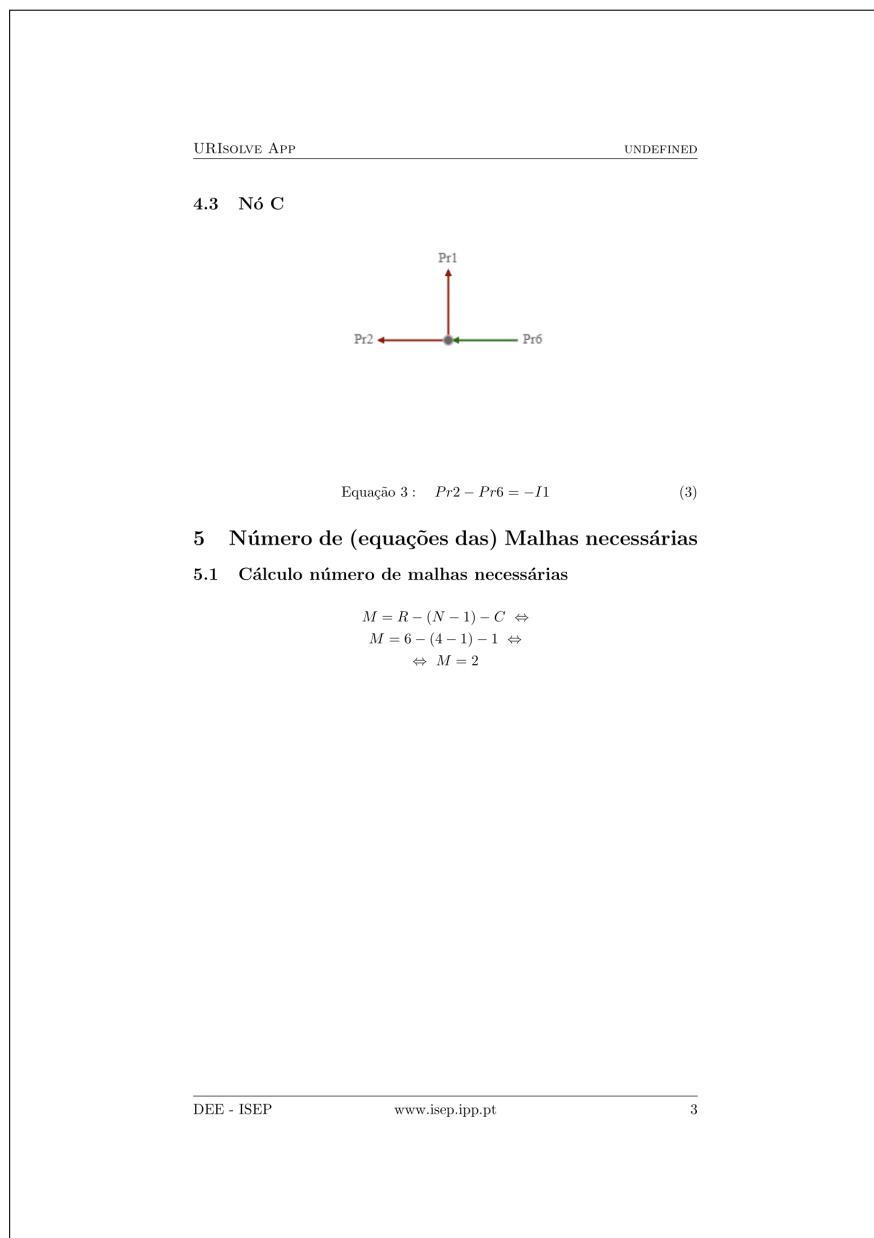


Figura A.13: Pág 3 do Output PDF corrente contínua com fonte de corrente

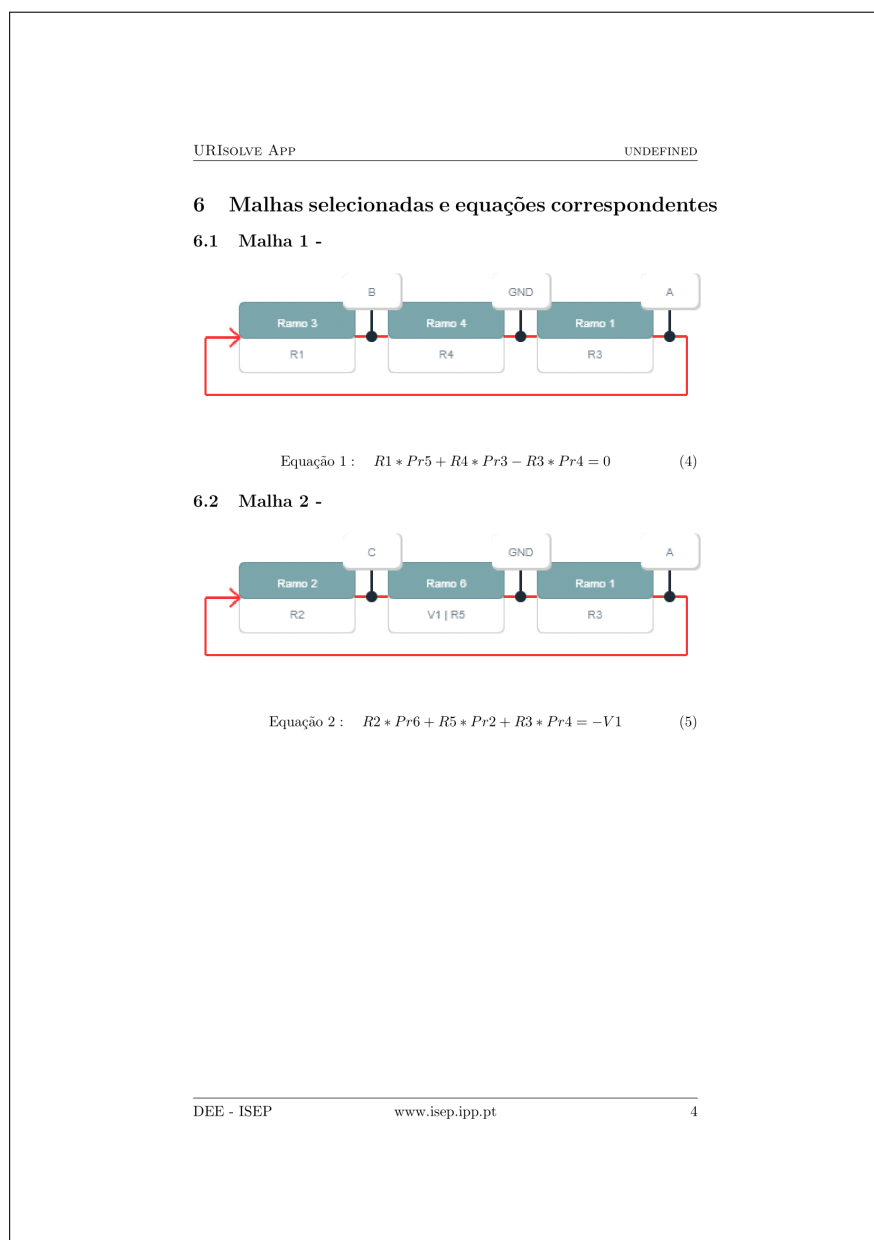


Figura A.14: Pág 4 do Output PDF corrente contínua com fonte de corrente

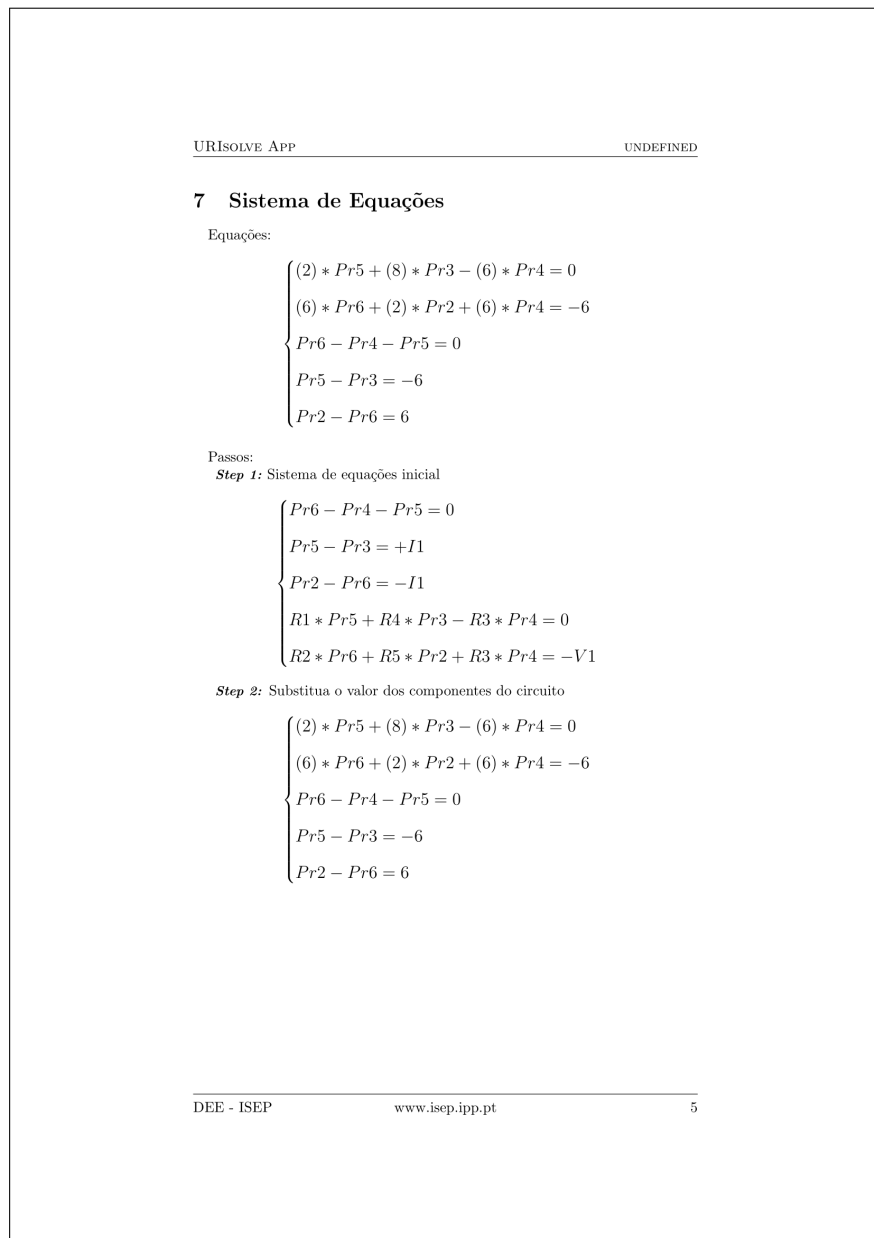


Figura A.15: Pág 5 do Output PDF corrente contínua com fonte de corrente

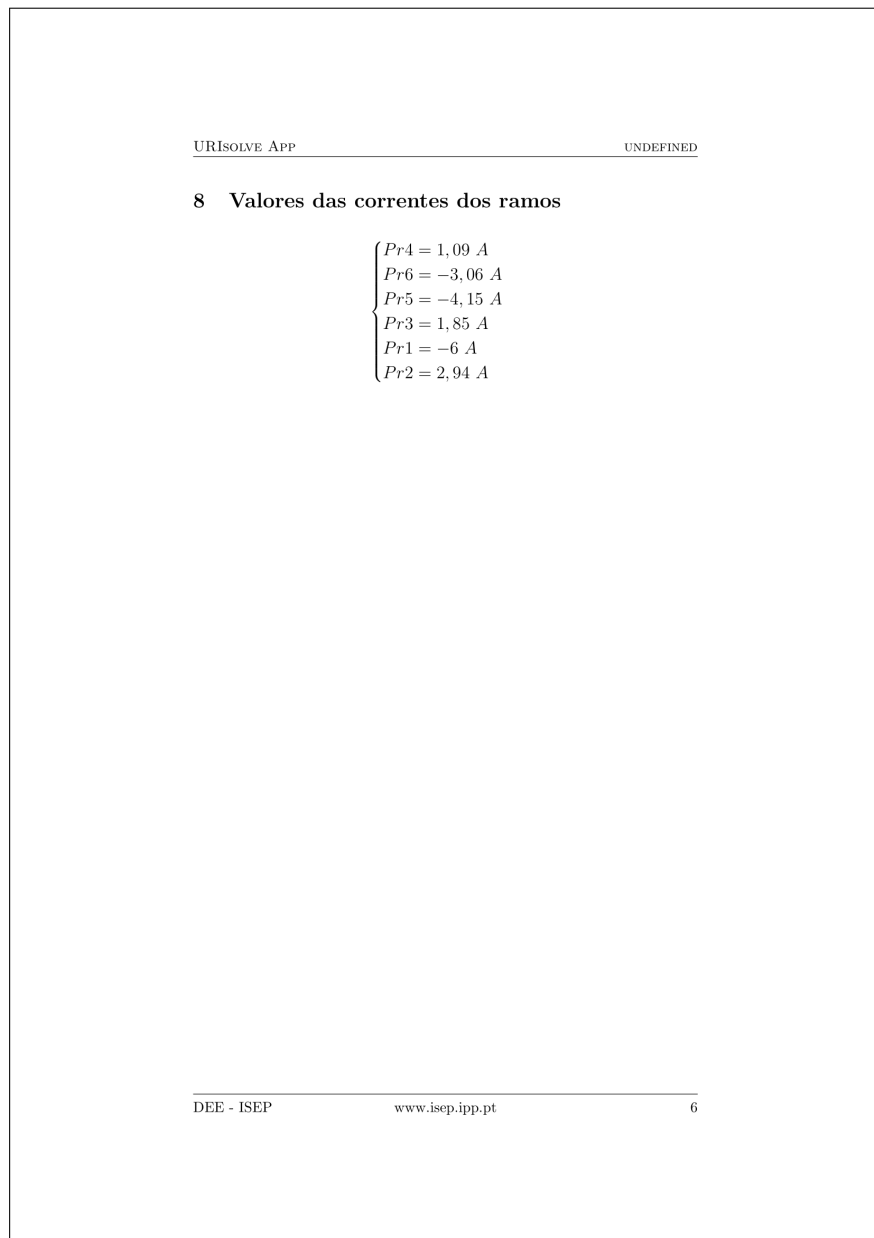


Figura A.16: Pág 6 do Output PDF corrente contínua com fonte de corrente

A.3 Circuito em Corrente Alternada

A.3.1 Imagem do Circuito

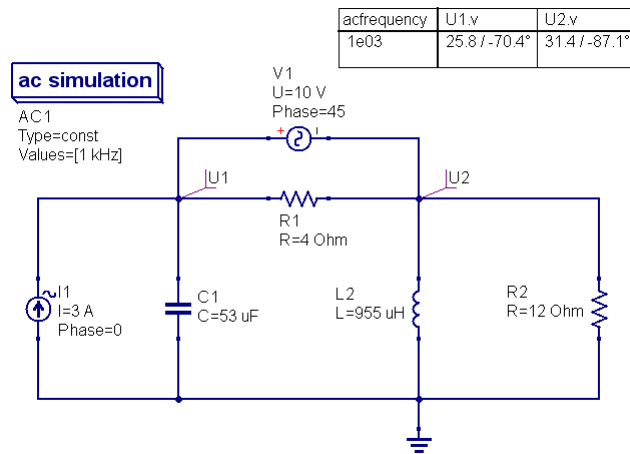


Figura A.17: Imagem do exemplo a ser analisado em corrente Alternada

A.3.2 Netlist do Circuito

```

1  L:L2 gnd U2 L="955 uH" I=""
2
3  R:R1 U1 U2 R="4 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="26.85"
4
5  Iac:I1 U1 gnd I="3 A" f="1 GHz" Phase="0" Theta="0" Ri="0 Ohm"
6
7  Vac:V1 U1_TEMP_V1 U2 U="10 V" f="1 GHz" Phase="45" Theta="0" Ri="0
  Ohm"
8  R:_R_U1_TEMP_V1 U1_TEMP_V1 U1 R="0 Ohm" Temp = "26.85" Tc1 = "0.0"
  Tc2 = "0.0" Tnom = "26.85"
9  .AC:AC1 Type="const" Values="[1 kHz]" Noise="no"
10
11 R:R2 U2 gnd R="12 Ohm" Temp="26.85" Tc1="0.0" Tc2="0.0" Tnom="
  26.85"
12
13 C:C1 gnd U1 C="53 uF" V=""

```

Listagem A.3: Netlist do circuito a ser analisado em corrente alternada

A.3.3 Output em PDF

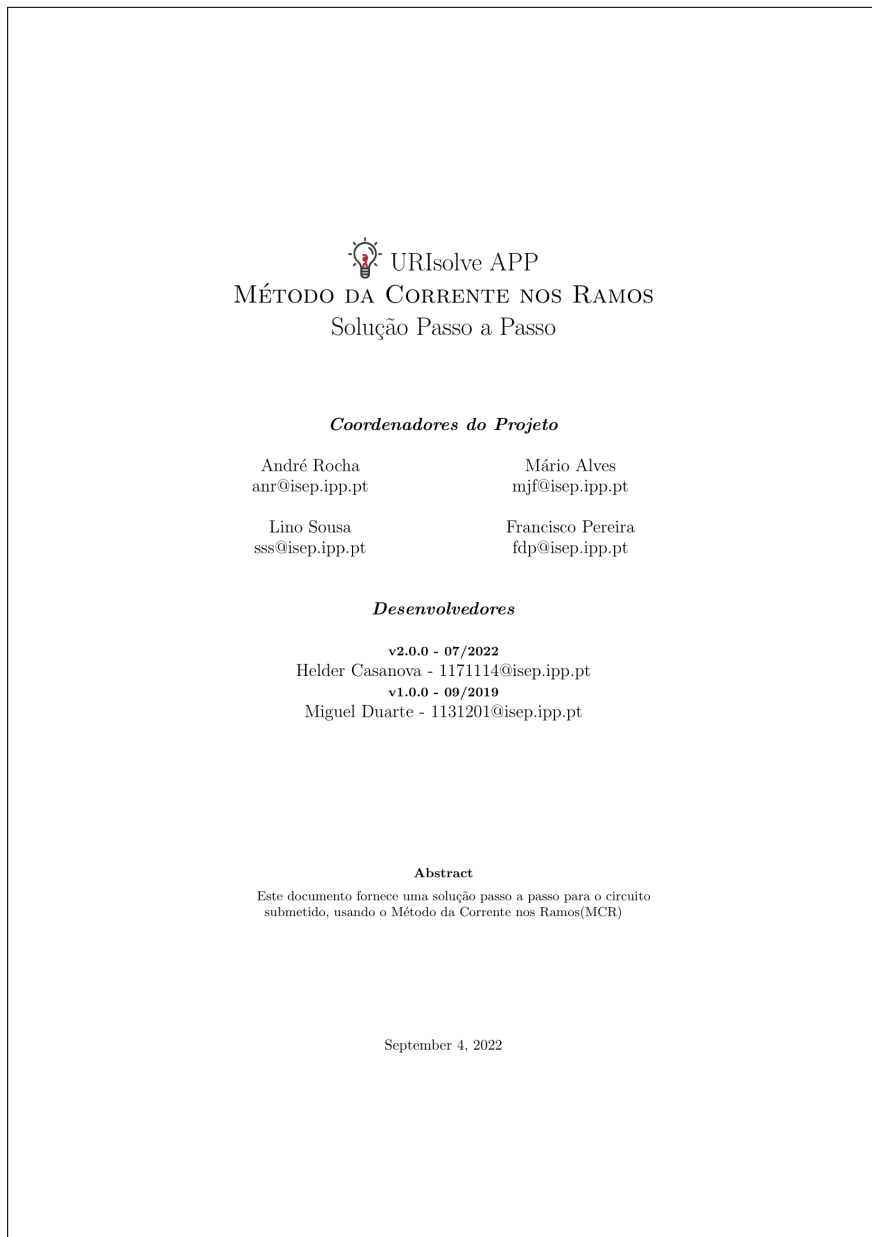


Figura A.18: Capa do Output PDF corrente alternada

URISOLVE APP

MÉTODO DA CORRENTE NOS RAMOS

1 Variáveis do Circuito

Ramos [R]	Nós [N]	Fontes de Corrente [C]	Fontes Isoladas de Tensão [T]
R=6	N=3	C=1	T=1

2 Informação do Circuito

Simulação [AC/DC]	Frequência [A]	Amperímetros [I]
AC	F=1 kHz	0/6

3 Identificação da Corrente nos Ramos

3.1 Arbitragem da Corrente nos Ramos

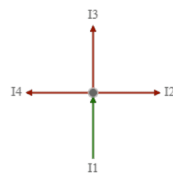
Table 1: Lista das Correntes do Circuito e as suas propriedades/componentes

Reference	Start Node	End Node	Components
I2	U1	U2	R1
I3	U1	gnd	C1
I4	U1	U2	V1
I1	gnd	U1	I1
I5	U2	gnd	R2
I6	U2	gnd	L2

Figura A.19: Pág 1 do Output PDF corrente alternada

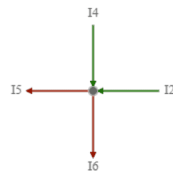
4 Equações das Correntes (Lei de Kirchhoff dos Nós - LKN)

4.1 Nó U1



$$\text{Equação 1 : } I_2 + I_3 + I_4 = +I_1 \quad (1)$$

4.2 Nó U2



$$\text{Equação 2 : } I_5 + I_6 - I_2 - I_4 = 0 \quad (2)$$

Figura A.20: Pág 2 do Output PDF corrente alternada

5 Número de (equações das) Malhas necessárias

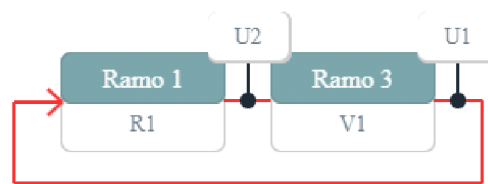
5.1 Cálculo número de malhas necessárias

$$\begin{aligned}M &= R - (N - 1) - C \Leftrightarrow \\M &= 6 - (3 - 1) - 1 \Leftrightarrow \\&\Leftrightarrow M = 3\end{aligned}$$

Figura A.21: Pág 3 do Output PDF corrente alternada

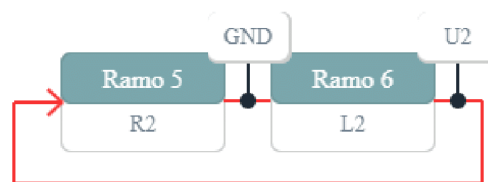
6 Malhas seleccionadas e equações correspondentes

6.1 Malha 1



$$\text{Equation : } IM1 : R1 * I2 = V1 \quad (3)$$

6.2 Malha 2

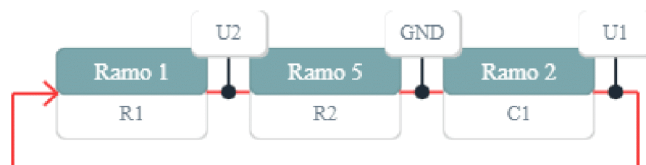


$$\text{Equation : } IM2 : R2 * I5 - L2 * I6 = 0 \quad (4)$$

Figura A.22: Pág 4 do Output PDF corrente alternada

URISOLVE APP

MÉTODO DA CORRENTE NOS RAMOS

6.3 Malha 3

$$\text{Equation : } I_{M3} : R1 * I2 + R2 * I5 - C1 * I3 = 0 \quad (5)$$

Figura A.23: Pág 5 do Output PDF corrente alternada

7 Sistema de Equações

Equações:

$$\begin{cases} (4) * I2 = (7.071 + 7.071i) \\ (12) * I5 - (6i) * I6 = 0 \\ (4) * I2 + (12) * I5 - (-3i) * I3 = 0 \\ I2 + I3 + I4 = +(3.0000) \\ I5 + I6 - I2 - I4 = 0 \end{cases}$$

Passos:

Step 1: Sistema de equações inicial

$$\begin{cases} I2 + I3 + I4 = +I1 \\ I5 + I6 - I2 - I4 = 0 \\ R1 * I2 = V1 \\ R2 * I5 - L2 * I6 = 0 \\ R1 * I2 + R2 * I5 - C1 * I3 = 0 \end{cases}$$

Step 2: Substitua o valor dos componentes do circuito

$$\begin{cases} (4) * I2 = (7.071 + 7.071i) \\ (12) * I5 - (6i) * I6 = 0 \\ (4) * I2 + (12) * I5 - (-3i) * I3 = 0 \\ I2 + I3 + I4 = +(3.0000) \\ I5 + I6 - I2 - I4 = 0 \end{cases}$$

Figura A.24: Pág 6 do Output PDF corrente alternada

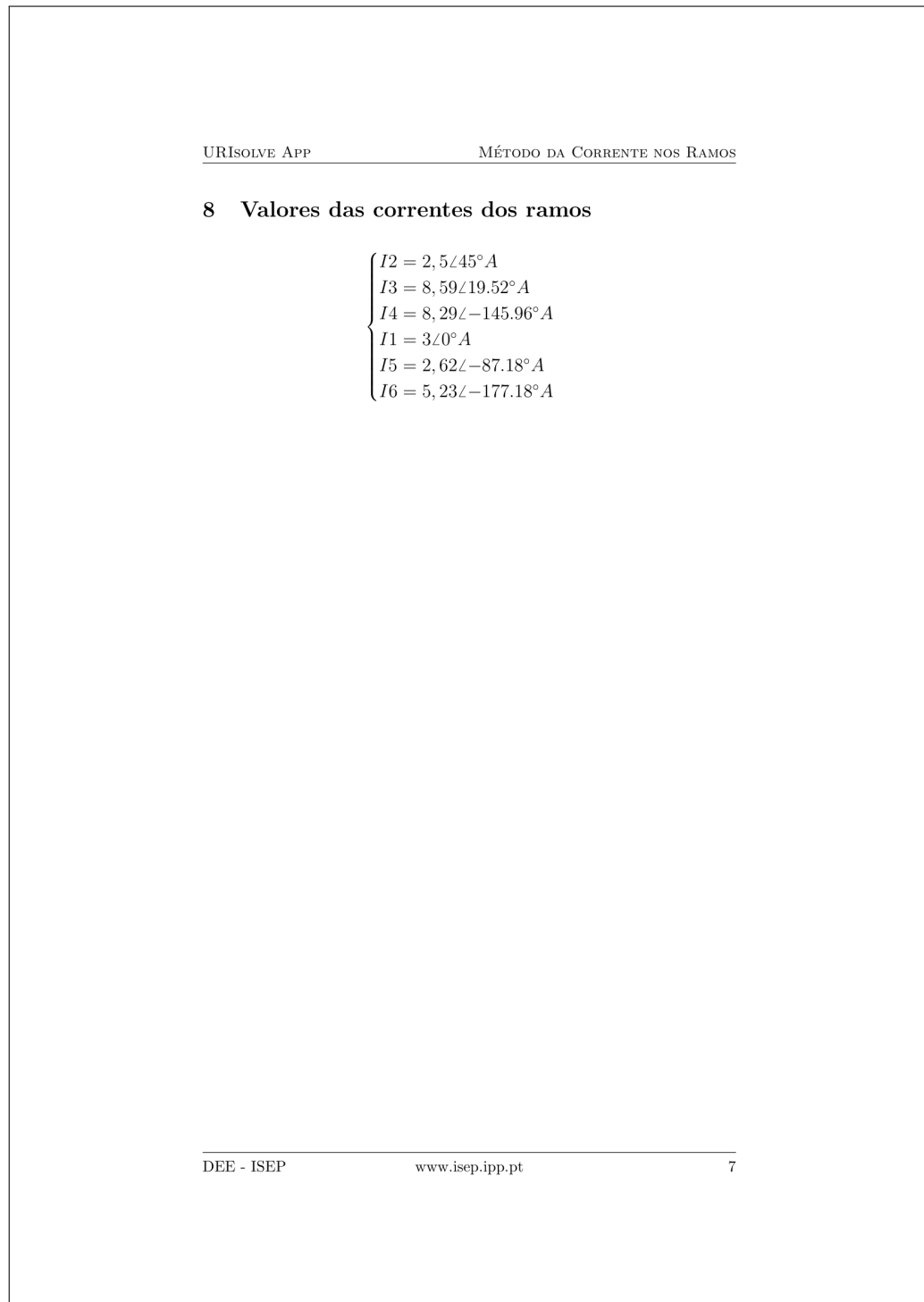


Figura A.25: Pág 7 do Output PDF corrente alternada