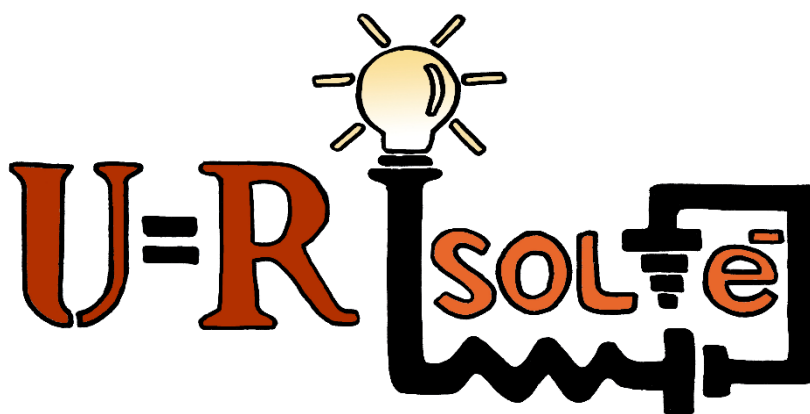


U=RISOLVE – APLICATIVO DE ANÁLISE DE CIRCUITOS ELÉTRICOS SIMULADOS NO QUCS



Lino Manuel Cerqueira Sousa



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2018

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

Candidato: Lino Manuel Cerqueira Sousa, N° 1140355, 1140355@isep.ipp.pt

Orientador: Mário Jorge de Andrade Ferreira Alves (mjf@isep.ipp.pt)

Co-orientador: Francisco José Dias Pereira (fdp@isep.ipp.pt)



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

23 de julho de 2018

Agradecimentos

Expresso nesta secção os meus sinceros agradecimentos a todos aqueles que, de alguma forma, contribuíram para a realização deste projeto:

À minha família, uma fonte de motivação e inspiração inesgotável, por todos os esforços em prol do meu percurso enquanto estudante e por consistir no alicerce de todas as concretizações que eu possa vir a alcançar.

Aos Professores Mário Alves e Francisco Pereira, pelo permanente acompanhamento e orientação, e cuja exigência e excelência possibilitaram a superação deste desafio.

E a todos aqueles que contribuíram para o teste e depuração da aplicação, particularmente os colegas Luís Lima, Tatiana Costa, André Rocha e Manuel Soares.

Resumo

O projeto que aqui é apresentado consiste numa aplicação de análise de circuitos elétricos designada de “U=RISOLVE” e orientada para o âmbito educativo nomeadamente alunos que iniciam o seu percurso académico na Engenharia Eletrotécnica (EE). Trata-se de uma ferramenta que auxilia o aluno na resolução de circuitos elétricos em Corrente Contínua e Corrente Alternada através do Método das Tensões nos Nós (MTN).

A análise para cada circuito é feita recorrendo a um ficheiro de texto proveniente da aplicação de simulação *Quite Universal Circuit Simulator* (QUCS), com informação acerca dos componentes e dos Nós de ligação aos quais esses componentes estão associados. Este ficheiro pode ser gerado sempre que um circuito é simulado no QUCS e tem o nome de *netlist*. Assim, é possível enviar a *netlist* para a aplicação, que será capaz de a tratar de forma a resolver o circuito e apresentar os resultados da análise do MTN.

O que torna esta aplicação dedicada à autoaprendizagem distinguindo-a dos *softwares* de simulação comuns, é a forma como os resultados são apresentados ao aluno. Habitualmente, os simuladores atuam como instrumentos de medição, devolvendo ao utilizador os valores mensurados. No entanto, a U=RISOLVE expõe, passo a passo, as equações matemáticas resultantes do MTN, permitindo ao utilizador obter uma análise teórico-prática essencial para posteriormente confrontar com a simulação obtida no QUCS.

De forma a abranger o máximo número de utilizadores possível, a aplicação foi desenvolvida em *HyperText Markup Language* (HTML) sendo que a análise computacional relativa aos circuitos elétricos foi implementada em JavaScript (JS). Esta arquitetura permite que a aplicação seja executada em *browser*, o que amplia claramente a sua versatilidade tendo em conta a contemporânea facilidade de acesso aos *browsers*.

O alvo deste aplicativo é, como referido, a comunidade estudantil. São os estudantes que poderão vir a utilizar a aplicação no futuro e, portanto, é a opinião dos mesmos que define a qualidade e a utilidade deste projeto. Com efeito, foi solicitado a um grupo de estudantes de EE do Instituto Superior de Engenharia do Porto que testasse e avaliasse esta ferramenta de modo a retirarem as respetivas conclusões.

Abstract

The project presented here refers to an electrical circuit analysis application denominated U=RISOLVE and it's orientated for the educational scope, particularly students starting their academic course through Electrical Engineering (EE). This is a supporting tool that helps the student solve electrical circuits in both direct and alternated current by Nodal Analysis method.

The analysis for each circuit is done by means of a text file derived from the Quite Universal Circuit Simulator (QUCS), that includes information about the electronic components and their connection nodes. This specific file can be generated whenever a simulation is done in QUCS and it's called netlist. Therefore, it is possible to upload the netlist to the app, which will be able to handle it to solve the circuit and display the Nodal Analysis results.

What makes this application dedicated to self-learning is how the results are shown to the student, distinguishing it from the common simulation software. Usually, electronic simulators operate as measuring instruments, providing the calculated values to users. However, U=RISOLVE exhibit, step by step, the Nodal Analysis mathematical equations, granting the user a crucial theoretical study that allows posterior comparisons with QUCS simulations.

To acquire as many students as possible, the application was developed in HyperText Markup Language (HTML) and the computational analysis related to electrical circuits was implemented in JavaScript. This architecture allowed the app to run in browsers, which clearly increased it's versatility, considering the current ease to access browsers.

The app's target, as referred to before, is the student community. The students are the ones who will possibly use U=RISOLVE in the future, hence its their opinion that truly determines the quality and usefulness of this project. In fact, a group of EE students from ISEP was requested to test and evaluate this tool in order to draw conclusions.

Keywords

Nodal Analysis, Self-Learning, Browser App, QUCS, Circuit Equations, Kirchhoff Laws, Simulator, Electrical Engineering, Solver, Javascript, Electric Circuits, Circuit Analysis.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS	XIII
ACRÓNIMOS.....	XV
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	1
1.2. OBJECTIVOS.....	2
1.3. CONTRIBUIÇÕES	4
1.4. CALENDARIZAÇÃO	4
1.5. ESTRUTURA DO RELATÓRIO	4
2. TECNOLOGIA EM SIMULADORES DE CIRCUITOS ELÉTRICOS.....	7
2.1. OS SIMULADORES NO PROJETO E NO ENSINO	7
2.2. OS SIMULADORES EXISTENTES.....	10
2.3. TABELA COMPARATIVA DOS DIFERENTES SIMULADORES.....	17
2.4. O QUCS COMO FERRAMENTA DE SIMULAÇÃO BASE DESTE PROJETO.....	19
3. METODOLOGIAS DE ANÁLISE	21
3.1. MÉTODO DAS TENSÕES NOS NÓS	21
3.2. MODIFIED NODAL ANALYSIS	25
3.3. AS NETLISTS NOS SIMULADORES	29
4. DESCRIÇÃO DA APLICAÇÃO U=RISOLVE	33
4.1. LÓGICA, OBJETIVOS E ARQUITECTURA.....	33
4.2. O SISTEMA DE FICHEIROS DO JAVASCRIPT.....	36
4.3. O ALGORITMO DE ANÁLISE	38
4.4. A GERAÇÃO DE EQUAÇÕES.....	49
4.5. INTERFACE GRÁFICA COM O UTILIZADOR (GUI)	53
5. A UTILIZAÇÃO DA APLICAÇÃO.....	59
5.1. CASOS-EXEMPLO EM CORRENTE CONTÍNUA	59
5.2. CASOS-EXEMPLO EM CORRENTE ALTERNADA	63

6. CONCLUSÕES	67
REFERÊNCIAS DOCUMENTAIS.....	71

Índice de Figuras

Figura 1	Exemplo da aplicação <i>CircuitLab</i> [5]	11
Figura 2	Exemplo da aplicação <i>KTechLab</i> [6]	11
Figura 3	Exemplo do software <i>PSpice</i> [7]	12
Figura 4	Exemplo do software <i>LTSpice</i> [8]	13
Figura 5	Exemplo do software <i>PSIM</i> [9]	14
Figura 6	Exemplo da aplicação <i>Falstad Circuit Simulator</i> [10]	15
Figura 7	Exemplo da aplicação <i>EveryCircuit</i> [11]	15
Figura 8	Exemplo da aplicação <i>SystemVision Cloud</i> [12]	16
Figura 9	Exemplo do software <i>QUCS</i> [13]	17
Figura 10	Exemplo de circuito para o MTN [14]	22
Figura 11	Representação matricial do Método das Tensões nos Nós [14]	24
Figura 12	Circuito exemplo com Super-Nó [15]	25
Figura 13	Exemplo de circuito do <i>QUCS</i> [18]	27
Figura 14	Circuito descrito por uma <i>netlist</i> no <i>PSpice 5.1 (Beta)</i> [19]	29
Figura 15	Lógica de utilização da aplicação <i>U=RISOLVE</i>	34
Figura 16	Exemplo de <i>upload</i> de um ficheiro no <i>HTML5</i>	36
Figura 17	Fluxograma da aplicação <i>U=RISOLVE</i>	39
Figura 18	Aquisição da informação da <i>netlist</i> para uma matriz	40
Figura 19	Método de remoção de instrumentos de medição	40
Figura 20	Remoção de amperímetros	41
Figura 21	Circuito com erros de identificação de Nós	43
Figura 22	Caixa de alerta com erros de identificação de Nós	44
Figura 23	Diagrama de blocos do algoritmo de análise	45
Figura 24	Método de análise de circuitos	46
Figura 25	Circuito com Fontes isoladas de Tensão	48
Figura 26	Geração das equações dos Nós	50
Figura 27	Geração das equações das Correntes	51
Figura 28	Configuração dos botões	54
Figura 29	Configuração da numeração da página de instruções	54
Figura 30	Fração horizontal com uso de <i>CSS</i>	55
Figura 31	Página inicial da aplicação <i>U=RISOLVE</i>	56
Figura 32	Página de instruções	56
Figura 33	Janela de resultados	57
Figura 34	Circuito de nível básico em Corrente Contínua	60

Figura 35	Solução U=RISOLVE do circuito nível básico em Corrente Contínua.....	60
Figura 36	Circuito de nível intermédio em Corrente Contínua	61
Figura 37	Solução U=RISOLVE do circuito nível intermédio em Corrente Contínua.....	61
Figura 38	Circuito de nível avançado em Corrente Contínua.....	62
Figura 39	Solução U=RISOLVE do circuito nível avançado em Corrente Contínua.....	62
Figura 40	Circuito de nível básico em Corrente Alternada	64
Figura 41	Solução U=RISOLVE do circuito nível básico em Corrente Alternada.....	64
Figura 42	Circuito de nível intermédio em Corrente Alternada	65
Figura 43	Solução U=RISOLVE do circuito nível intermédio em Corrente Alternada.....	65
Figura 44	Circuito de nível avançado em Corrente Alternada.....	66
Figura 45	Solução U=RISOLVE do circuito nível avançado em Corrente Alternada.....	66

Índice de Tabelas

Tabela 1	Calendarização do projeto	4
Tabela 2	Análise comparativa dos diferentes simuladores de circuitos elétricos	18

Acrónimos

API	– Application Programming Interface
CANCER	– Circuit Analysis of Non-Linear Circuits, Excluding Radiation
CSS	– Cascading Style Sheets
EE	– Engenharia Eletrotécnica
GUI	– Graphical User Interface
HTML	– HyperText Markup Language
JS	– JavaScript
LKN	– Lei de Kirchhoff dos Nós
MNA	– Modified Nodal Analysis
MTN	– Método das Tensões nos Nós
OSS	– Open-Source Software
PCB	– Printed Circuit Board
QUCS	– Quite Universal Circuit Simulator
SPICE	– Simulation Program with Integrated Circuit Emphasis
W3C	– World Wide Web Consortium

1. INTRODUÇÃO

Neste capítulo introdutório abordar-se-ão as necessidades que culminaram na realização deste trabalho assim como as contribuições e objetivos que se pretendem atingir nas áreas às quais o projeto se destina.

A parte final do capítulo será dedicada à apresentação da organização deste relatório de forma a introduzir uma visão ampla do seguimento que foi atribuído a este projeto, bem como uma secção cronológica que expõe as atividades que levaram à sua concretização.

1.1. CONTEXTUALIZAÇÃO

O projeto aqui apresentado surgiu da necessidade de melhorar todo o processo de ensino na Engenharia Eletrotécnica, quer seja ao lecionar ou ao aprender, com vista a diminuir as dificuldades sentidas pelos alunos na aprendizagem. O primeiro impacto com circuitos elétricos pode ser visto, pelos alunos, como uma temática de conteúdo de complexa análise, o que origina dificuldades em resolver convenientemente os circuitos que lhes são exigidos. As dificuldades passam por identificar corretamente os Ramos ou os Nós, sentidos das Correntes, polaridades de Fontes de Tensão e Corrente, em função do sentido da Corrente que foi estipulado, e tornam-se um problema ainda maior quando é necessário transpor as variáveis para uma equação relativa a um Nó ou uma Malha completa.

Os simuladores de circuitos elétricos são ferramentas essenciais para o processo de ensino-aprendizagem, e embora possam fornecer as soluções para a análise de circuitos, não apresentam as equações dos circuitos que permitem chegar às soluções dos mesmos.

É nesta problemática que se insere a aplicação U=RISOLVE – os softwares de simulação apenas simulam o circuito e retornam os resultados, o seu objetivo é gerar uma solução, não prestando qualquer interface ou serviço de carácter didático. A U=RISOLVE fornece ao aluno uma solução estruturada de forma a que este tenha facilidade em entendê-la e que a possa espontaneamente associar ao circuito que estipulou. A solução apresentada ao aluno envolve o conjunto de equações associadas ao Método das Tensões nos Nós (MTN), equações estas que estão organizadas por ordem crescente de complexidade com o intuito de fomentar a rapidez e eficácia do utilizador na resolução de qualquer circuito.

1.2. OBJETIVOS

A aplicação que este projeto pretende retratar, incide sobre duas temáticas que se consideram fundamentais para uma otimização da atual metodologia de ensino na EE e podem ser considerados os principais objetivos deste trabalho, sendo estes de seguida apresentados:

- Fomentar a autoaprendizagem e a qualidade de ensino na Engenharia Eletrotécnica;
- Colmatar a inexistência de funcionalidades de natureza didática nos simuladores de circuitos elétricos.

É importante referir que a aplicação U=RISOLVE em nada altera o método de ensino, no entanto complementa-o. Por outras palavras, esta aplicação serve-se de informação relativa a cada circuito, extraída do QUCS, que é um simulador correntemente usado no ensino da EE, e frequentemente alvo de estudo e melhoramento devido à sua característica de *Open-Source Software* (OSS), introduzindo uma componente pedagógica, externa a este *software*, de forma a reforçar aprendizagem.

Este projeto dedica-se, tal como referido anteriormente, ao estudo do MTN e são as equações referentes a este método que são providas pela aplicação. A escolha deste método deveu-se muito ao facto de os simuladores atuais utilizarem uma versão aproximada do mesmo para efetuar a análise de circuitos elétricos, a análise nodal modificada – *Modified Nodal Analysis* (MNA). Ora, sendo a aplicação U=RISOLVE baseada e fundamentada em simuladores de circuitos elétricos, complementa-se modelarmente com o Método das

Tensões nos Nós. Não obstante esta similaridade, deixa-se como trabalho futuro a implementação dos restantes métodos de análise abordados no ensino da EE de forma a evoluir ao máximo a ideia que aqui se expõe – a introdução de ferramentas/interfaces de índole didático nos simuladores de circuitos elétricos.

1.3. CONTRIBUIÇÕES

A grande contribuição deste projeto situa-se na área da didática com recurso a uma componente científico-tecnológica. Isto é, o complementar do ensino da EE, estimulando a autoaprendizagem através de uma ferramenta de simulação que auxilia o aluno a solucionar matematicamente circuitos elétricos, com maior rapidez e menor probabilidade de erros de cálculo. É uma aplicação bastante útil principalmente quando se está perante um circuito com vários Nós que, por sua vez, origina várias equações, onde a ocorrência de erros de cálculo se torna uma realidade bastante frequente.

O estudo da tecnologia nos simuladores existentes atualmente é, adicionalmente, uma das contribuições deste projeto, na medida em que se trata de uma importante comparação das tecnologias e de uma eleição do simulador base da aplicação U=RISOLVE.

1.4. CALENDARIZAÇÃO

Este projeto foi desenvolvido ao longo de dezassete semanas e segmentado em diferentes tarefas, que se encontram representadas na Tabela 1. O trabalho teve uma fase inicial de aquisição de conhecimento relacionado com os simuladores de circuitos elétricos existentes e a sua metodologia de análise. Posteriormente foi necessário recordar as ferramentas de programação sob as quais a aplicação iria ser implementada e, por fim, estipular o algoritmo, proceder ao seu desenvolvimento e testar o seu funcionamento.

Tabela 1 Calendarização do projeto

ID	Nome das Etapas	Início	Fim	Duração	Março 2018				Abril 2018				Maio 2018				Junho 2018			
					5/3	12/3	19/3	26/3	2/4	9/4	16/4	23/4	30/4	7/5	14/5	21/5	28/5	4/6	11/6	18/6
1	Estudo dos Simuladores	05/03/18	12/03/18	1w																
2	Estudo do MNA	12/03/18	19/03/18	1w																
3	Estudo do Spice	19/03/18	02/04/18	2w																
4	Análise dos Outputs do QUCS	26/03/18	02/04/18	1w																
5	Revisões do MTN, HTML, CSS e JS	02/04/18	16/04/18	2w																
6	Estudo do Sistema de Ficheiros JavaScript	16/04/18	23/04/18	1w																
7	Desenvolvimento da Aplicação	16/04/18	04/06/18	7w																
8	Fase de Testes	04/06/18	25/06/18	3w																
9	Elaboração do Relatório Final	25/05/18	30/06/18	5w																

1.5. ESTRUTURA DO RELATÓRIO

O relatório está dividido em seis capítulos, e cada um destes capítulos foi formulado para que no final da leitura, proporcionasse uma visão ampla da base, funcionamento e implementação da aplicação em que o relatório incide.

O primeiro capítulo é fundamentalmente uma exposição dos principais motivos que levaram à concretização do projeto, incluindo as contribuições do mesmo do ponto de vista científico, tecnológico e didático. Este capítulo inclui ainda um diagrama de Gantt que revela cronologicamente as tarefas que possibilitaram a realização do trabalho. O segundo capítulo envolve uma componente de estudo focada na tecnologia atual em simuladores de circuitos elétricos, este capítulo evidencia os simuladores de maior relevância da atualidade e fundamenta a escolha do QUCS como a ferramenta de simulação base para este projeto. O capítulo três explicita as metodologias de análise de circuitos elétricos que são relevantes para este projeto e a forma como cada uma delas é introduzida na aplicação U=RISOLVE. O quarto e quinto capítulo são exclusivamente dedicados à aplicação em si, com efeito, o quarto capítulo retrata uma completa descrição da aplicação, isto é, as ferramentas que levaram à sua implementação, a lógica, a arquitetura, o algoritmo de análise e a *Graphical User Interface* (GUI). Vários exemplos de circuitos resolvidos pela aplicação serão ainda apresentados no quinto capítulo. Por último, no capítulo seis, são sintetizados os objetivos que foram cumpridos, bem como as dificuldades que foram surgindo ao longo da realização deste projeto. Neste último capítulo é ainda descrito um

inquérito realizado aos alunos e docentes que contribuíram para a fase de testes da aplicação. A razão deste inquérito é claramente compreensível: o projeto é para os alunos, são eles que vão usufruir da aplicação e, portanto, a sua opinião é imprescindível para avaliar o sucesso deste trabalho.

2. TECNOLOGIA EM SIMULADORES DE CIRCUITOS ELÉTRICOS

Os simuladores de circuitos elétricos são hoje em dia uma ferramenta essencial no ensino. Este capítulo reforça a sua imprescindibilidade e vai ao encontro das suas origens de modo a clarificar as inúmeras vantagens destes aplicativos. Existe, na atualidade, um leque variadíssimo de simuladores. Aqui são apresentados alguns exemplos que se consideram ser os mais relevantes para o processo de ensino na EE, dado que é no âmbito da didática/autoaprendizagem que este projeto se insere.

2.1. OS SIMULADORES NO PROJETO E NO ENSINO

A simulação de circuitos elétricos e eletrónicos detém uma significativa importância no ensino e na execução da Engenharia Eletrotécnica. A sua relevância poderá, no entanto, ter vindo a ser vulgarizada em virtude da crescente evolução tecnológica que se traduz numa maior facilidade de utilização de qualquer tipo de software, e os programas de simulação de circuitos elétricos não são exceção. Efetivamente, cada vez mais se observa uma crescente simplicidade e rapidez no que toca a simular um circuito elétrico, e novos

aplicativos para as mais modernas plataformas digitais vão surgindo, seja com novas funcionalidades de simulação, novos componentes ou até com melhores análises no que diz respeito à exatidão. A título de exemplo, é possível encontrar-se estudos com o intuito de se gerarem modelos matemáticos que permitem aproximar o comportamento de um circuito ou componente ao seu comportamento real [1], deixando de se considerar que o componente se comporta idealmente de acordo com as suas características, tal como se verifica na prática onde ocorrem variações de temperatura, onde existe histerese e vários outros fatores que influenciam o funcionamento dos componentes elétricos e eletrónicos.

De modo a que as ferramentas de simulação não sejam alvo de uma banalização imposta pelo avanço tecnológico torna-se importante salientar que nem sempre a Engenharia Eletrotécnica deteve esta facilidade. O primeiro software de análise de circuitos elétricos a ter grande impacto (não no momento do seu lançamento, mas sim alguns anos mais tarde) foi o tão conhecido *Simulation Program with Integrated Circuit Emphasis* (SPICE). O SPICE foi apresentado à comunidade em abril de 1973 na Universidade da Califórnia, Berkeley e, tal como mencionado anteriormente, nessa altura o programa não obteve a reputação que lhe era merecida. Assim o reconheceu um dos seus fundadores Laurence W. Nagel na sua retrospectiva “*Life Of Spice*” em 1996 [2].

I don't think anybody had a clue of the impact of that paper or the computer program it described. While we all had modest expectations for SPICE what happened was truly phenomenal. Within a few years, SPICE had achieved acceptance at almost all electrical engineering schools, and a cottage industry sprouted to supply SPICE derivatives to the rapidly expanding integrated circuit industry. To this day, twenty-five years later, students at most electrical engineering schools learn how to use SPICE or some derivative.

Ao analisar as palavras de Nagel, verificam-se dois aspetos fulcrais que levaram ao reconhecimento deste programa: a crescente indústria dos circuitos integrados e a evolução do ensino da Engenharia Eletrotécnica.

Relativamente aos circuitos integrados, é necessário voltar atrás no tempo para entender as questões relacionadas com o projeto/design destes circuitos. O primeiro circuito integrado a revolucionar a indústria da eletrónica e computação (não tendo, no entanto, sido o primeiro circuito integrado a ser desenvolvido/patenteado) pertence à Texas Instruments e foi desenvolvido em 1958 pelo Engenheiro Eletrotécnico Jack Kilby (Nobel da Física no ano de 2000) [3]. No início da década de 60, pela primeira vez na história, um circuito

integrado foi aplicado na indústria, mais propriamente num computador desenvolvido para fins militares na Força Aérea dos Estados Unidos da América. Ou seja, foi implementado o primeiro circuito integrado na indústria aproximadamente 10 anos antes da existência do primeiro software de simulação. A questão que aqui se coloca é a seguinte: como era realizado o *design* destes circuitos sem qualquer tipo de simulação?

O método tradicional de projeção de circuitos eletrónicos passava por testar protótipos submetendo-os a diferentes inputs – sinais, mudanças de temperatura, variações de Tensão de alimentação, medindo e analisando a resposta desse protótipo com instrumentos de medida. Este método tinha uma grande desvantagem que impedia a produção em série destes equipamentos, consumia demasiado tempo para projetar o circuito e tinha custos elevados por consequência dos testes que eram realizados aos componentes, danificando-os. No entanto, este método permitia desenvolver competências práticas e fornecia uma visão mais eficaz acerca do comportamento dos circuitos. O que os clientes consideravam uma grande vantagem nos circuitos integrados, era, na realidade, um sério problema para o engenheiro, isto é, as suas diminutas dimensões. Este fator por vezes obrigava a que os circuitos fossem implementados em *breadboard*, o que conduzia a discrepâncias no que diz respeito a medidas e resultados observados tendo em conta os elementos parasitas (perdas) presentes numa *breadboard*, que por sua vez, diferem bastante dos que existem num circuito integrado [4]. Assim, é possível entender que os programas de simulação impulsionaram a evolução da eletrónica e da computação até aos sistemas avançados que estão disponíveis na atualidade.

Voltando à citação de Nagel, o lançamento do seu software despertou o interesse de várias escolas de engenharia para que estas pudessem implementar e usar o Spice. De facto, a simulação de circuitos elétricos e eletrónicos foi sem dúvida uma mais valia no ensino da Engenharia Eletrotécnica. Não só pela capacidade que introduziu na projeção de protótipos, mas também porque funciona como uma ferramenta de solução de cálculo matemático onde os alunos podem verificar os resultados obtidos teoricamente, fomentando assim a autonomia característica de um aluno de engenharia.

Os simuladores são imprescindíveis para uma melhor aprendizagem por várias razões, auxiliam na análise matemática do circuito, permitem uma verificação muito próxima da realidade de como o circuito se comporta de uma forma relativamente acessível, porque se

torna dispendiosa a utilização de instrumentos de medida, e mais importante ainda, evitam situações de erro que levam à danificação de eventuais componentes.

2.2. OS SIMULADORES EXISTENTES

Num ambiente académico exige-se um simulador que seja de simples utilização e compreensão, que possua os recursos necessários para desenvolver qualquer tipo de projeto ou circuito, ou que detenha de uma funcionalidade de criar componentes e modelos para os mesmos. A estas ferramentas impõe-se ainda que tenham a capacidade de fornecer os resultados pretendidos pelo utilizador em diferentes formatos de modo a que possa eleger o que lhe é de maior conveniência tendo em conta o circuito que pretende simular.

As licenças e versões de estudante, nos dias que correm, são serviços fundamentais para os simuladores no geral, assim como a capacidade de abranger diferentes plataformas onde possam operar. Em seguida apresentam-se alguns dos simuladores existentes, evidenciando-se as suas principais características, assim como as vantagens e desvantagens da sua utilização.

CircuitLab

O *CircuitLab* [5], ilustrado na Figura 1, é um simulador em *browser*, implementado totalmente em JavaScript, o que lhe atribui a vantagem de não requerer a instalação de nenhum *plug-in*. Embora as licenças para este *software* variem entre 20 e 80 euros, a versão de teste que é disponibilizada garante as ferramentas e componentes básicos de simulação e uma interface de notável simplicidade e facilidade de utilização. Este simulador possui tecnologias que ampliam a rapidez no momento de desenhar o circuito, seja para efetuar ligações, encontrar componentes, ou visualizar resultados. Contudo, a versão de teste dificilmente poderia ser implementada, quer em ambiente industrial, quer académico porque torna-se uma aplicação muito primária, apenas útil no início do estudo sobre circuitos elétricos. Uma das suas grandes vantagens é a sua comunidade, este aplicativo possui um fórum onde qualquer utilizador pode expor as suas questões ou pesquisar por uma solução, o *CircuitLab* ultrapassa as 100 000 simulações semanais, o que favorece a utilização do fórum.

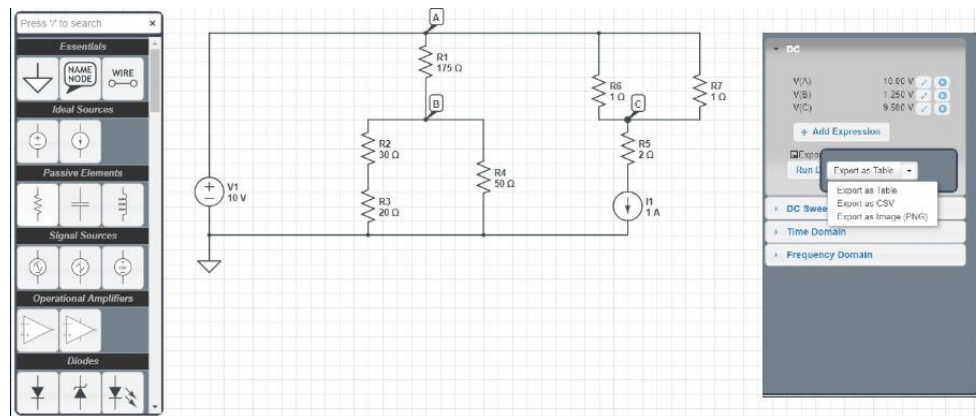


Figura 1 Exemplo da aplicação *CircuitLab* [5]

KTechLab

O *KTechLab*[6] é um projeto recente, com pouco mais de 3000 *downloads*, e trata-se de uma aplicação *open-source*, ainda numa fase *beta*, mas com enorme potencial. Este *software* corre em ambientes *Linux* e, para além de possibilitar a simulação de circuitos analógicos, permite ainda desenvolver e simular simultaneamente projetos de eletrónica digital com microcontroladores, isto é, desenvolver o programa e o circuito e, em tempo real, observar os resultados. A sua interface é também bastante simplificada, tal como é evidenciado na Figura 2, no entanto o conteúdo, em termos de bibliotecas disponíveis é ainda bastante limitado dado a fase em que o projeto se encontra.

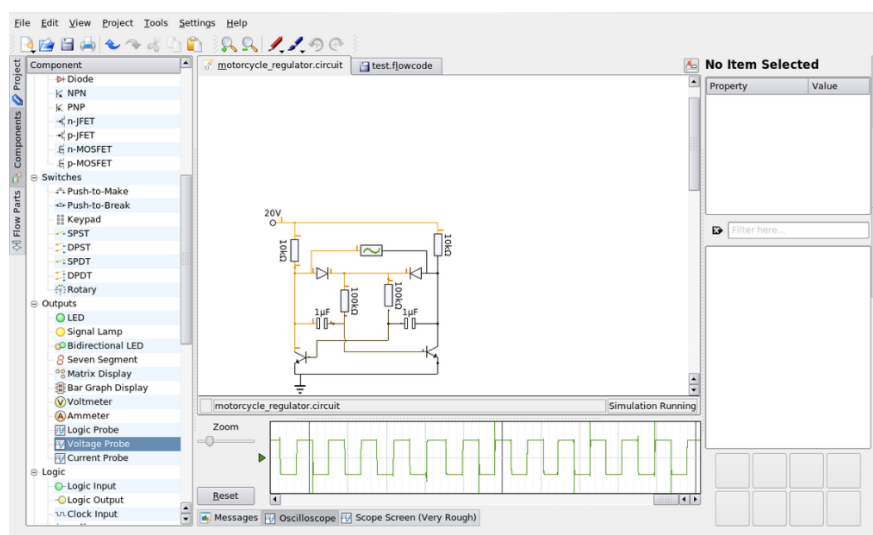


Figura 2 Exemplo da aplicação *KTechLab* [6]

PSpice

O PSpice[7], da Cadence, demonstrado na Figura 3, é talvez o *software* de simulação de circuitos elétricos e eletrônicos mais completo do mercado. Este programa possui uma biblioteca com modelos para inúmeros componentes, e continua em ascensão. Este simulador é dedicado ao *design* e projeto de todo o tipo de circuitos e, portanto, possui um grau de complexidade acentuado. É um *software* pago, embora exista uma versão de estudante e uma *Lite*, que impedem a inclusão de componentes mais complexos. No entanto, o seu ambiente gráfico em nada facilita o processo de aprendizagem, até porque o seu principal objetivo está ligado à indústria de *design* de circuitos integrados e Placas de Circuito Impresso (PCB – *Printed Circuit Boards*), pelo facto de proporcionar métodos de análise de elevada complexidade.

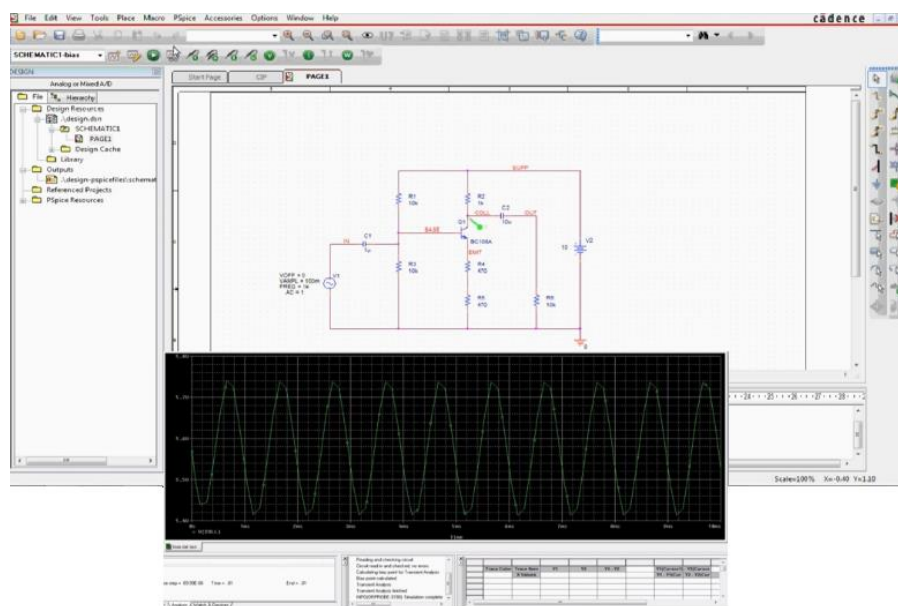


Figura 3 Exemplo do *software* PSpice [7]

LTspice

O LTspice [8] da Linear Technology, retratado na Figura 4, é porventura o *software* mais utilizado pela comunidade académica e até científica, por se tratar de um programa gratuito e possuir grande parte das capacidades que lhe são exigidas pelos estudantes. É um *software* robusto e a sua grande vantagem é a otimização no que diz respeito à utilização do processador do computador. Essa otimização garante-lhe a uma rapidez de simulação superior à maioria dos programas de simulação que usam a ferramenta Spice. A interface

do *LTspice* não é de todo envolvente, sendo o processo de instrução excessivamente demorado. Ainda assim, é compensatório devido ao facto de se tratar de um *software* bastante leve, em termos de memória ocupada, comparativamente com os restantes simuladores.

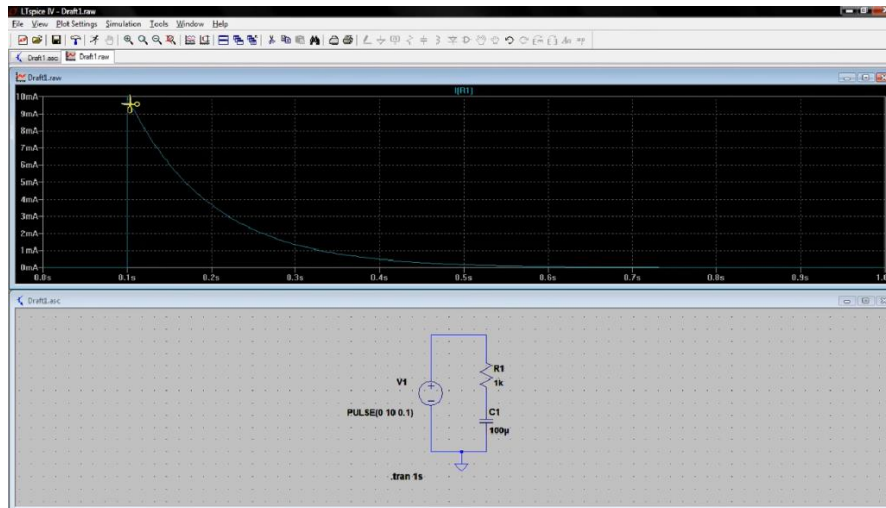


Figura 4 Exemplo do *software LTspice* [8]

PSIM

O PSIM [9] é um programa maioritariamente dedicado à simulação de circuitos de controlo e à eletrónica de potência. Contudo, desde a sua versão 11.1, possui um módulo derivado do Spice utilizado pelo *software LTspice*, que ampliou a sua gama de simulações tornando-o num simulador bastante completo. É de relativamente simples utilização para circuitos de eletrónica de potência, e pode ser testado na totalidade das suas funcionalidades, numa versão *trial* de 30 dias. No entanto, o que destaca este simulador dos restantes é a funcionalidade de permitir uma análise não ideal, isto é, o utilizador pode lidar com perdas, sejam elas elementos parasitas, perdas de condução ou perdas de comutação que aproximam o circuito do seu comportamento real através de uma parametrização da simulação, acrescentando-lhe, assim, um nível de exatidão muito pouco comum nos simuladores atuais. Uma demonstração da sua utilização encontra-se evidenciada na Figura 5.

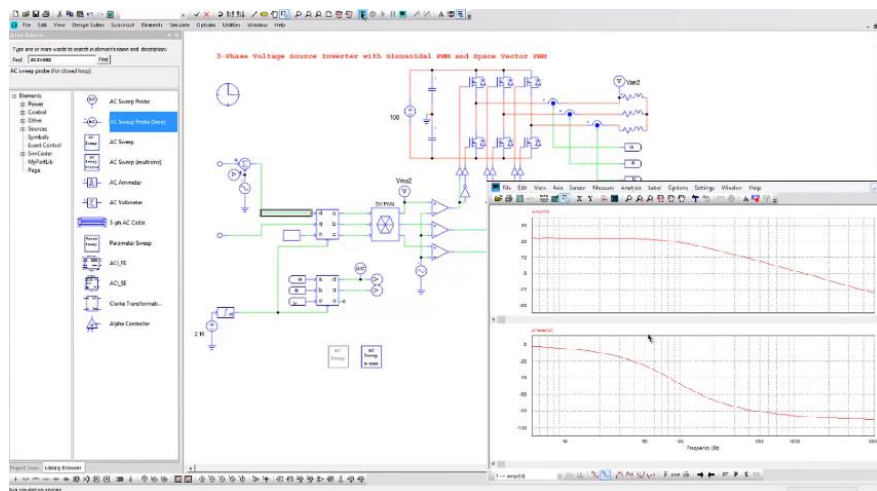


Figura 5 Exemplo do software PSIM [9]

Falstad Circuit Simulator

Um simulador de extrema simplicidade e incrivelmente intuitivo, o *Falstad Circuit Simulator* [10] é um aplicativo implementado em *browser*, também em JS, *open-source* e tem funcionalidades bastante interessantes. Uma delas é o facto de permitir visualizar o fluxo de Corrente e a forma como os componentes o alteram em tempo real. O fluxo de Corrente Elétrica é uma característica dos circuitos elétricos que fica subjacente à sua análise e muitas das vezes os alunos de EE não percebem o significado real deste fenómeno, e esta é uma aplicação que facilita a sua compreensão. A interface é apelativa e contém vários exemplos de circuitos, permitindo controlar o circuito em tempo real, por exemplo, ativar um interruptor. Para além destas características, para verificar a Corrente, Tensão ou Potência basta mover para o ponto que se pretende medir, o indicador do rato, tal como demonstra a Figura 6. A gama de componentes não é muito vasta, mas abrange um considerável número de componentes, o que, numa aplicação totalmente gratuita, é um ponto positivo.

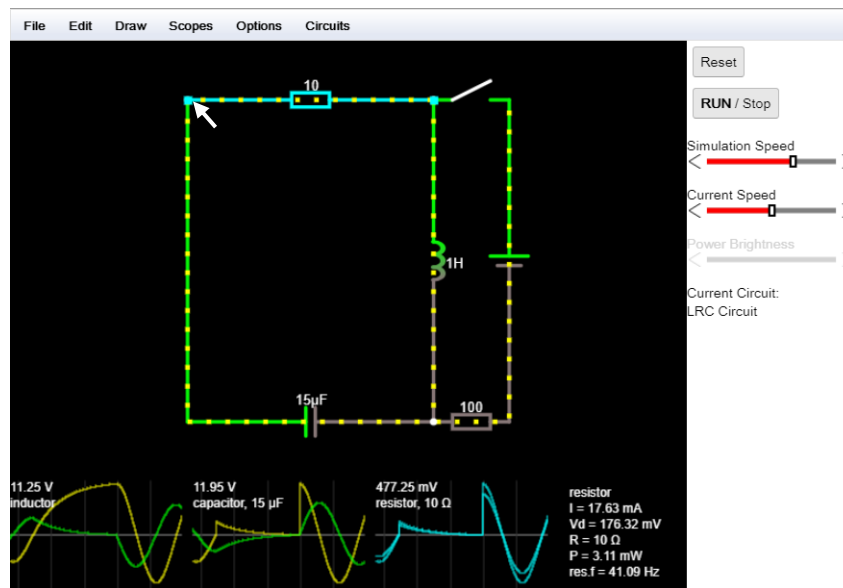


Figura 6 Exemplo da aplicação *Falstad Circuit Simulator* [10]

EveryCircuit

O *EveryCircuit* [11] possui umas das interfaces mais apelativas de todos os simuladores e permite também observar o fluxo de Corrente no circuito. Está disponível para *smartphones Android* e *iOS*, e ainda numa versão em *browser* apresentada na Figura 7. No entanto é uma aplicação paga na versão *browser*, no valor de 15 euros, tendo versões gratuitas apenas nas para *smartphones*. Contudo estas contêm apenas os componentes mais básicos. Permite facilmente criar e simular circuitos, possui uma visualização dinâmica – em tempo real, possibilitando uma simulação em tempo real e uma excelente interatividade.

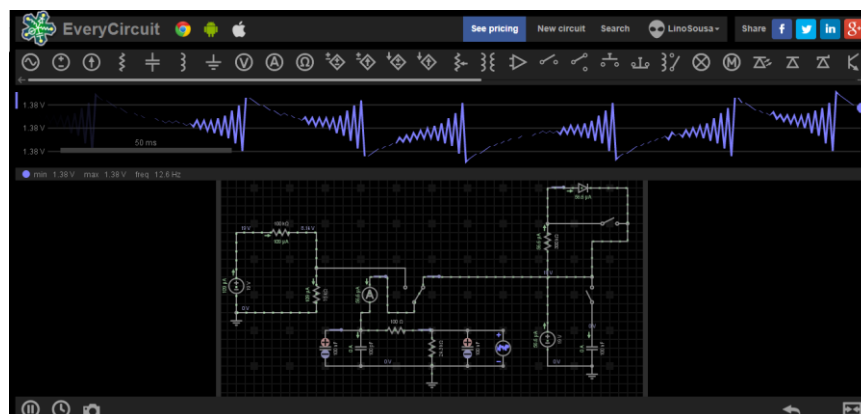


Figura 7 Exemplo da aplicação *EveryCircuit* [11]

System Vision Cloud

O *System Vision Cloud* [12] é um aplicativo gratuito e opera em *browser*. Possui uma interface dinâmica, de simples utilização e um intuitivo método de obtenção de resultados como mostra a Figura 8. No entanto, o seu ponto forte é o sistema baseado em *cloud* que permite guardar e aceder a projetos a qualquer momento e dispositivo. Mas mais importante que isso, permite trabalhar em grupo. O *System Vision Cloud* tem ainda um repositório/fórum com *designs* e circuitos de outros utilizadores que podem ser adquiridos para a própria *cloud* e vice-versa. O único senão desta aplicação é a falta de variedade de componentes, sendo de maior utilidade no âmbito académico. No entanto existem versões pagas de maior complexidade e variedade no que diz respeito a bibliotecas.

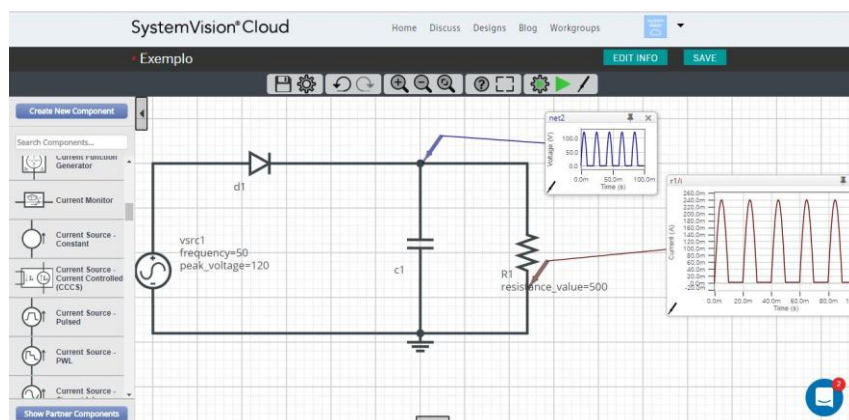


Figura 8 Exemplo da aplicação *System Vision Cloud* [12]

QUCS

O QUCS [13] é um simulador *open-source* lançado em 2003 e que tem vindo constantemente a ser desenvolvido, quer pelos próprios autores, quer por projetos voluntários de índole académico. Desde julho de 2017, este *software* apresenta uma média de 10 mil *downloads* mensais. A sua grande vantagem reside na sua simplicidade e flexibilidade, é um *software* que pode ser executado em praticamente todos os sistemas operativos, possui uma crescente biblioteca de componentes e um leque muito variado nas formas de apresentação de resultados. Tem vindo a acrescentar novos e sofisticados métodos e modelos de análise, possibilitando o desenvolvimento de projetos cada vez mais

abrangentes e complexos. Uma demonstração da sua interface encontra-se exemplificada na Figura 9. Embora se considere uma interface relativamente “*user-friendly*”, os próprios autores do QUCS incentivam os programadores voluntários a melhorarem a sua qualidade gráfica, assim como outras variantes, por exemplo de suporte aos utilizadores como fóruns e tutoriais.

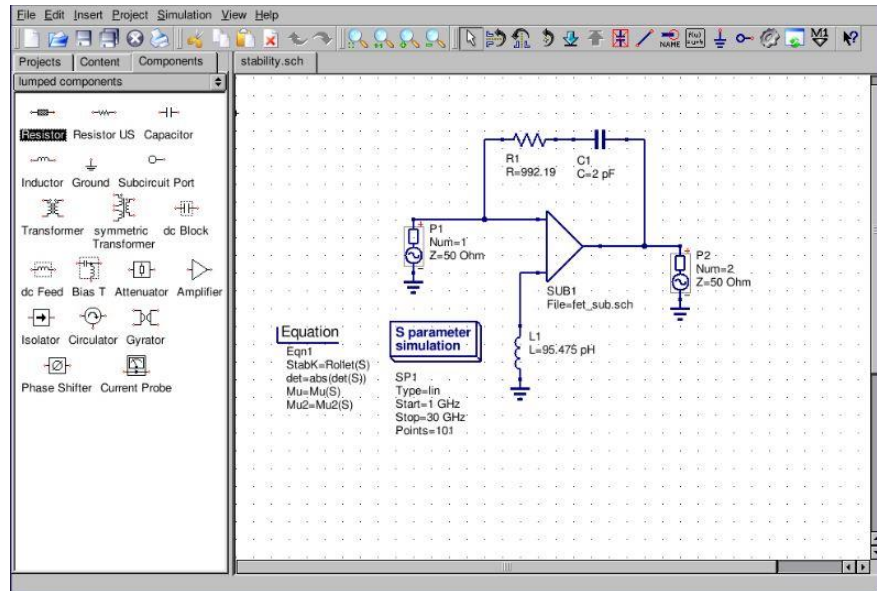


Figura 9 Exemplo do software QUCS [13]

2.3. TABELA COMPARATIVA DOS DIFERENTES SIMULADORES

Os padrões associados à avaliação de circuitos elétricos variam com o contexto onde são inseridos e utilizados. No entanto, como todo este projeto se enquadra principalmente num ambiente académico e com objetivos focados na valorização do ensino, adotaram-se parâmetros de avaliação tais como a complexidade associada à utilização do simulador, os sistemas operativos em que opera, a abrangência das bibliotecas, modelos e metodologias de análise, o tipo de licença a interface e a variedade de métodos de apresentação de resultados que a ferramenta de simulação possui. Todas estas características foram avaliadas e sumarizadas na Tabela 2, assim como foi atribuída uma classificação final em função dos padrões avaliados.

Tabela 2 Análise comparativa dos diferentes simuladores de circuitos elétricos

	Licença	Sistema operativo	Bibliotecas, modelos e metodologias	Complexidade de inicialização	Interface	Análise de resultados	Classificação final
CircuitLab	Pago 20€ / Ano ★★	Linux, Windows, Mac OS, Browser	★★	★★★★★	★★★★	★★★	★★★
KTechLab	GPL ★★★★★	Linux	★★	★★★	★	★★	★★
PSpice	Pago 2 000 € ★	Windows	★★★★★	★	★★	★★★	★★★
LTspice	Freeware ★★★★	Windows, MAC OS	★★★	★★	★	★★	★
PSIM	Pago ★	Windows	★★★★	★★	★★	★★★	★★★
Falstad	GPL ★★★★★	Browser	★★	★★★★	★★★	★★★	★★★
EveryCircuit	Pago 14 € ★★★	Browser, iOS, Android	★★★	★★★	★★★★★	★★★	★★
System Vision	Freeware ★★★★	Browser	★★★	★★★★	★★★★	★★★★	★★★
QUCS	GPL ★★★★★	Linux, Windows, MAC	★★★	★★★★	★★★	★★★★	★★★★

Sintetizando a informação que a tabela anterior apresenta, pode verificar-se que tanto o *PSIM* como o *PSpice* lideram em termos de conteúdo disponível. Contudo tratam-se de *softwares* pagos e de elevada complexidade. O *Falstad* e o *System Vision*, são simuladores com uma interface apelativa e de relativamente simples utilização, contudo estão apenas implementados em *browser*, o que impede a sua aplicabilidade em modo *offline*, o que não impede que possam ser proveitosos para complementar o estudo/simulação. Já o *LTspice*, embora possua uma boa aceitabilidade pela comunidade académica e científica, tem uma interface muito pouco desenvolvida e não dinamiza a sua utilização. O *EveryCircuit* e o *KTechLab* têm bastante potencial de desenvolvimento, mas por enquanto a escassez de versões expandidas a sistemas operativos computacionais impede que sejam utilizados em larga escala. O *CircuitLab* é uma alternativa bastante viável, mas não se encontra ao nível do QUCS, não só pela sua licença ser paga, mas também porque a sua biblioteca de componentes é restrita. Caso contrário estaria no patamar do QUCS que é, de uma forma geral, o simulador mais completo, por todas as razões evidenciadas na comparação:

- *Software open-source*;
- Flexibilidade de funcionamento em vários sistemas operativos;
- Bibliotecas e metodologias relativamente vastas e em constante crescimento;

- Interface *user-friendly*;
- Ampla gama de alternativas de apresentação de resultados.

2.4. O QUCS COMO FERRAMENTA DE SIMULAÇÃO BASE DESTE PROJETO

Na fase inicial deste projeto, revelou-se necessário eleger um *software* que fosse capaz de gerar um *output* pós simulação que identificasse devidamente a conectividade do circuito simulado (*netlist*) para que este pudesse ser utilizado como *input* de informação na aplicação U=RISOLVE. Esta escolha verificou-se óbvia – o QUCS, e não apenas pelo facto de cumprir o requisito da geração do *output*. De facto, após uma investigação realizada acerca dos diferentes simuladores, observou-se que parte destes *softwares* que usa uma versão do *standard* Spice tem a capacidade de gerar e utilizar *netlists*. O QUCS foi elegido como a ferramenta base deste projeto também pela sua constante e prolongada presença neste estabelecimento de ensino em Engenharia Eletrotécnica, seja na unidade curricular de Fundamentos da Engenharia Eletrotécnica, seja em Teoria dos Circuitos. Este *software* é o primeiro contacto que os alunos têm com um simulador de circuitos elétricos, e pouco tempo depois do seu ingresso em EE já se encontram suficientemente familiarizados com o QUCS. Deste modo, se tiverem uma ferramenta que se relaciona com o simulador e que os consegue auxiliar no estudo de uma parte fundamental da matéria, torna-se uma mais valia para a sua aprendizagem. Assim, ao longo deste projeto será possível evidenciar como é que uma aplicação com as características da U=RISOLVE pode complementar a utilização do simulador QUCS no âmbito do ensino e autoaprendizagem.

3. METODOLOGIAS DE ANÁLISE

Os métodos pelos quais as equações dos circuitos elétricos são geradas são a base de um simulador. A sua eficiência na exatidão dos resultados, nos tempos de resposta e no esforço computacional intrínseco à sua análise, são fatores cruciais na implementação de uma metodologia. Neste ponto do relatório são apresentados os principais processos de análise utilizados pelos simuladores de circuitos elétricos, nomeadamente o QUCS, e que estão inteiramente relacionados com o algoritmo de análise elaborado para a aplicação U=RISOLVE.

3.1. MÉTODO DAS TENSÕES NOS NÓS

O MTN é um método organizado que segue um conjunto de técnicas simples para a obtenção das equações do circuito. É baseado na Lei de *Kirchhoff* dos Nós (LKN), isto é, a soma de todas as Correntes que fluem para o Nó deve ser igual à soma das Correntes que saem do mesmo Nó, ou seja, o Nó não acumula carga. Neste método as variáveis a determinar são os potenciais nos Nós, relativamente a um ponto de referência (*ground*).

As etapas a seguir para a execução teórica deste método são as seguintes [14]:

1. Identificar todos os Nós do circuito.
2. Selecionar um Nó de referência, onde o potencial será nulo e, deste modo, obter-se-ão $N-1-T$ incógnitas, sendo N o número total de Nós existentes no circuito, e T o número de Fontes isoladas de Tensão.
3. Arbitrar os sentidos das Correntes.
4. Formular as equações para cada Corrente em função das tensões dos Nós, tendo em conta os sentidos escolhidos (ou seja, substituir cada Corrente através da lei de *Ohm*).
5. Formular as equações dos Nós com base na LKN.
6. Escrever as equações em função dos potenciais nos Nós.

Para exemplificar este método será analisado o circuito da Figura 10 através do procedimento aqui apresentado.

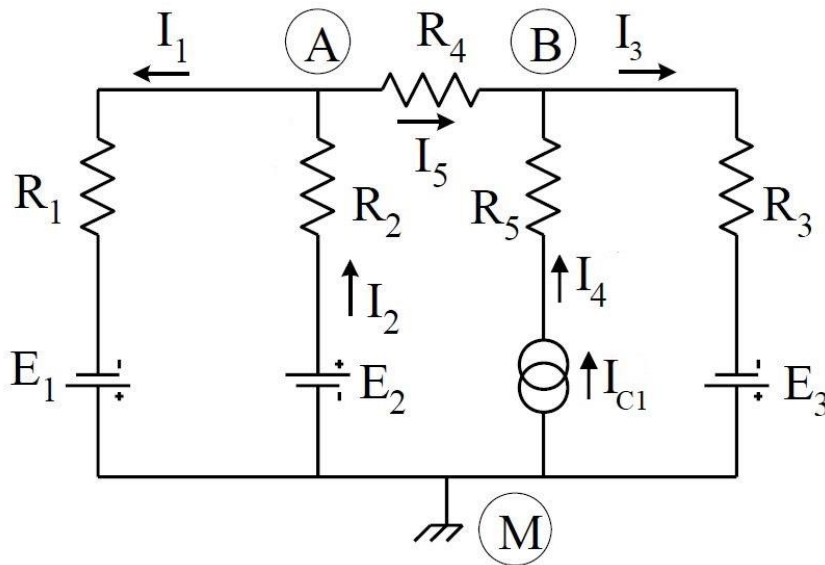


Figura 10 Exemplo de circuito para o MTN [14]

- **Passo 1**

Nós: A, B e M (3 Nós).

Equações: $N - 1 = 2$ equações.

- **Passo 2**

Nó de referência: Nó M.

- **Passo 3**

Sentidos das Correntes: I_1 (A para M), I_2 (M para A), I_3 (B para M), I_4 (M para B), I_5 (A para B).

- **Passo 4**

Equações das Correntes:

$$I_1 = \frac{U_A + E_1}{R_1} \quad (1)$$

$$I_2 = \frac{E_2 - U_A}{R_2} \quad (2)$$

$$I_3 = \frac{U_B + E_3}{R_3} \quad (3)$$

$$I_4 = I_{C1} \quad (4)$$

$$I_5 = \frac{U_A - U_B}{R_4} \quad (5)$$

- **Passo 5**

$$\text{Nó A: } -I_1 + I_2 - I_5 = 0 \quad (6)$$

$$\text{Nó B: } -I_3 + I_5 + I_4 = 0 \quad (7)$$

- **Passo 6**

$$\text{Nó A: } -\frac{U_A - E_1}{R_1} + \frac{E_2 - U_A}{R_2} - \frac{U_A - U_B}{R_4} = 0 \quad (8)$$

$$\text{Nó B: } -\frac{U_B + E_3}{R_3} + \frac{U_A - U_B}{R_4} + I_{C1} = 0 \quad (9)$$

Estão, desta forma, formuladas todas as equações que permitem calcular os potenciais nos Nós e posteriormente as Correntes em todos os Ramos do circuito. Este é sem dúvida um método de fácil interpretação e com uma componente metodológica focada na aprendizagem. O mesmo método poderá, no entanto, ser representado matricialmente e calculado através de equações matriciais, tal como demonstra a Figura 11.

$$\begin{bmatrix} G_{11} & -G_{12} & -G_{13} & \cdots & -G_{1K} \\ -G_{21} & G_{22} & -G_{23} & \cdots & -G_{2K} \\ & & \cdots & & \\ & & & \cdots & \\ -G_{K1} & -G_{K2} & -G_{K3} & \cdots & G_{KK} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \cdots \\ U_K \end{bmatrix} = \begin{bmatrix} I_{11} \\ I_{22} \\ \cdots \\ I_{KK} \end{bmatrix}$$

Figura 11 Representação matricial do Método das Tensões nos Nós [14]

A primeira matriz é preenchida com as condutâncias referentes a cada um dos Nós. Com efeito, na sua diagonal principal, encontram-se a soma das condutâncias que influenciam o Nó a que estão associadas, G_{11} para o Nó 1, G_{22} para o Nó 2 e assim sucessivamente. Estas condutâncias têm sempre sinal positivo. Os coeficientes fora da diagonal do sistema representam as condutâncias comuns entre dois Nós, exemplificando: uma condutância que está conectada ao Nó 1 e 2, deve encontrar-se na posição (1,2) e (2,1) da matriz. No exemplo da Figura 11 correspondem às condutâncias G_{21} e G_{12} respetivamente. As condutâncias fora da diagonal principal têm sempre valores negativos.

A segunda matriz, que multiplica pela matriz das condutâncias, está associada às variáveis que se pretendem calcular: os potenciais nos Nós. A última matriz corresponde à soma das Correntes de curto-circuito injetadas no Nó associado, o seu sentido depende da polarização da Fonte, isto é, o sentido da força eletromotriz das Fontes de Tensão ou o sentido da Corrente das Fontes de Corrente que influenciam diretamente o Nó.

As Fontes isoladas de Tensão, neste método de análise, provocam condutâncias infinitas nos sistemas de equações. A solução para ultrapassar este problema é conhecida como método dos Super-Nós [15] e consiste em remover uma das equações considerando que existe um Super-Nó formado por dois Nós diferentes, tal como é demonstrado na Figura 12.

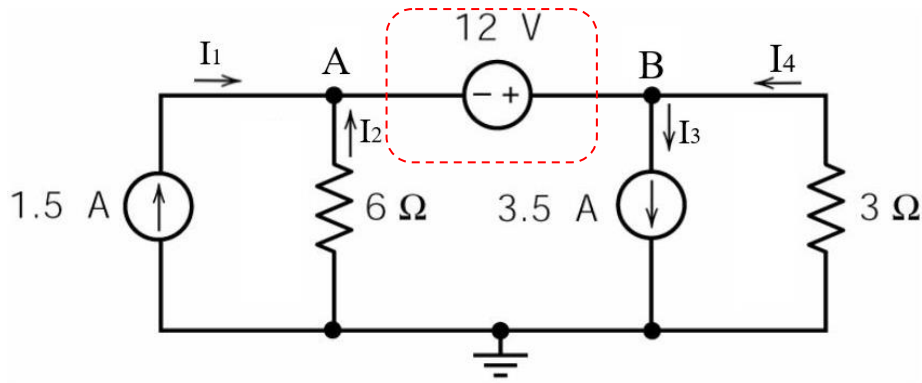


Figura 12 Circuito exemplo com Super-Nó [15]

Considerando que os Nós A e B formam um só Super-Nó o circuito terá apenas uma equação com três Correntes a convergir para o Super-Nó (I_1 , I_2 e I_4) e uma Corrente a divergir (I_3). O que permitirá obter a seguinte equação:

$$\text{Super-Nó AB: } I_1 + I_2 - I_3 + I_4 = 0 \quad (10)$$

No entanto o circuito continua a ter duas incógnitas, os potenciais nos dois Nós: V_A e V_B . A equação que está omissa trata-se da equação do super Nó:

$$\text{Super-Nó AB: } V_B - V_A = 12 \text{ V} \quad (11)$$

No caso de um dos Nós presentes no Super-Nó estar referenciado à terra, a Tensão no Nó que não está referenciado passa a ser automaticamente conhecida, porque se trata do valor da Fonte de Tensão associada, que pode ser positivo ou negativo dependendo da polaridade que está diretamente conectada ao Nó.

O MTN é, assim, um método de fácil compreensão e é devido à sua simplicidade que foi o eleito para ser utilizado pela aplicação U=RISOLVE. As equações são apresentadas segundo o primeiro processo apresentado, derivado da Lei de *Kirchhoff* dos Nós, porque se considera ser a metodologia mais apelativa no que diz respeito ao processo de aprendizagem visto que este é um dos principais focos da aplicação apresentada neste trabalho.

3.2. MODIFIED NODAL ANALYSIS

O *Modified Nodal Analysis* é o algoritmo utilizado nos simuladores de circuitos elétricos, principalmente na análise em Corrente Contínua. Contudo, é também expandido à Corrente Alternada. Este método veio substituir o tradicional *Nodal Analysis* ou Método das

Tensões nos Nós, utilizado inicialmente pelos *softwares* SPICE, de modo a preencher as suas limitações, mas mantendo a sua simplicidade [16].

O grande problema do método tradicional de análise de circuitos elétricos traduz-se na incapacidade de tratar Fontes isoladas de Tensão que não podem ser convertidas para Fontes de Corrente. Embora seja matematicamente possível resolver os circuitos através da análise de Super-Nós, esta metodologia não é propriamente sistemática e por isso não se torna algorítmicamente implementável. Contudo, na aplicação U=RISOLVE, foi estudado um processo de análise de modo a incluir o método dos Super-Nós, que será abordado no quarto capítulo, mas tal só se tornou possível porque as equações são apenas apresentadas, sem resolução das mesmas.

Até ao desenvolvimento do MNA, os simuladores optaram por contornar o problema das Fontes de Tensão isoladas de diferentes formas. Um exemplo de uma solução implementada foi pelo simulador *Computer Analysis of Non-linear Circuits, Excluding Radiation* (CANCER) [17], que substitui todas as Fontes isoladas de Tensão por circuitos equivalentes de Norton de Fontes de Corrente. Esta solução encontra-se cada vez mais otimizada, e hoje em dia, embora a maioria dos *softwares* optem pelo MNA, as alternativas existentes já não possuem qualquer tipo de limitação [18].

O MNA é, no entanto, o método mais rápido e eficaz de todos os restantes. É composto por um conjunto de matrizes que variam das matrizes no método convencional devido ao facto de possuírem informação acerca das Fontes isoladas de Tensão e Corrente. Para uma melhor compreensão desta metodologia é, de seguida, analisado um circuito, representado na Figura 13, através do MNA. De modo a manter uma conexão para as diferentes metodologias e algoritmos utilizados na aplicação U=RISOLVE, este circuito será posteriormente, de novo alvo de análise.

No circuito aqui apresentado, é possível identificar uma característica de alta relevância, que é a existência de dois Nós, um deles real (node_2) e um virtual (node_1). Esta particularidade dos Nós virtuais poderá, inicialmente, causar alguma confusão, mas é essencial para que exista uma análise algorítmica de qualquer circuito elétrico no que diz respeito à correta identificação da conectividade de cada circuito, que será retratada na subsecção seguinte relacionada com as *netlists*.

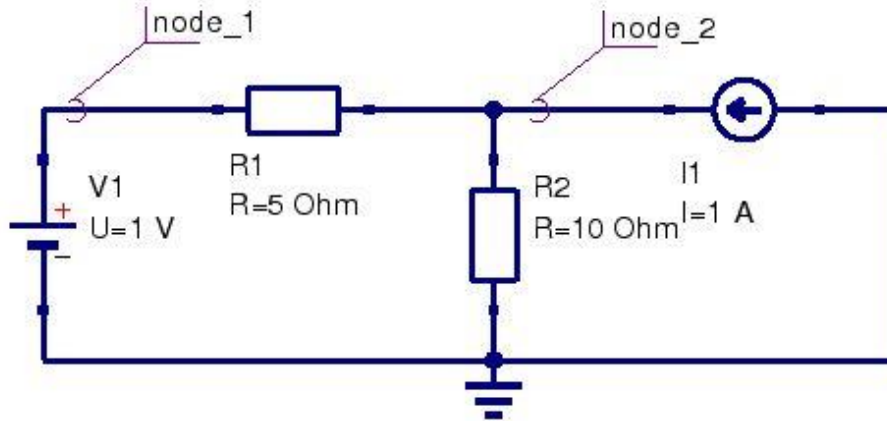


Figura 13 Exemplo de circuito do QUCS [18]

A aplicação do MNA a um circuito elétrico resulta numa equação matricial representada na Equação 12 [19]. A matriz A é composta por quatro diferentes matrizes com informação acerca dos componentes associados a cada Nó e parâmetros de análise referentes à existência de Fontes de Tensão ou Corrente isoladas. A matriz x contém as tensões dos Nós e as Correntes que passam nas Fontes de Tensão isoladas (variáveis desconhecidas que se pretendem calcular). E por fim, a matriz z engloba o somatório dos valores das tensões e Correntes conhecidos no circuito.

$$[A] \times [x] = [z] \quad (12)$$

A matriz A é então dividida em quatro matrizes diferentes: G , B , C e D .

$$A = \begin{bmatrix} G & B \\ C & D \end{bmatrix} \quad (13)$$

A matriz G identifica todas as conectividades entre todos os elementos do circuito. As matrizes B e C englobam as conectividades das Fontes de Tensão. E a matriz D indica as Fontes de Tensão isoladas e, no caso de não existirem, é preenchida com zeros.

Para o circuito exemplo apresentado na figura anterior tem-se a seguinte matriz G :

$$G = \begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_1} \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} \end{bmatrix} = \begin{bmatrix} 0.2 & -0.2 \\ -0.2 & 0.3 \end{bmatrix} \quad (14)$$

Na diagonal principal da matriz encontra-se a soma das condutâncias dos componentes conectados a cada um dos Nós, na primeira posição da diagonal (1,1) para o Nó 1 e na segunda posição da diagonal (2,2) para o Nó 2. Na diagonal secundária da matriz G situam-se as condutâncias negativas dos componentes que estão conectados a ambos os Nós, ou seja, a resistência R_1 que se situa entre os Nós 1 e 2, está localizada nas posições

da matriz (1,2) e (2,1). Até este ponto, o MNA em nada difere do método convencional de análise. As seguintes matrizes mostram as otimizações que este método veio inserir na simulação de circuitos elétricos. Assim, a matriz B, que será igual à matriz transposta de C, terá os valores 0, 1 e -1 dependendo da polaridade das Fontes de Tensão associadas a cada Nó.

$$B = C^T \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (15)$$

Neste exemplo a Fonte de Tensão V_1 tem o terminal positivo ligado ao Nó 1 e, portanto, o primeiro elemento associado a esse Nó contém o valor 1 e o elemento associado ao Nó 2, é 0 porque não existe qualquer Fonte de Tensão conectada ao referido Nó. A matriz que se seguiria para finalizar a matriz A seria a D, no entanto, como neste exemplo não existem Fontes isoladas de Tensão, essa matriz é nula.

Por fim, resta apresentar as matrizes x e z:

$$x = \begin{bmatrix} v_1 \\ v_2 \\ i_{V1} \end{bmatrix} \quad (16)$$

$$z = \begin{bmatrix} 0 \\ I_1 \\ U_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \quad (17)$$

A matriz x engloba, tal como foi referido, as variáveis relativas às tensões nos Nós e as correntes que passam nas Fontes de Tensão. Já na matriz z, nas primeiras duas linhas encontram-se as Correntes que fluem para cada um dos Nós (I_1 para o Nó 2 e nenhuma Fonte de Corrente para o Nó 1), na última linha encontra-se o valor da Fonte de Tensão existente no circuito.

O sistema de equações está agora completo e pronto a ser resolvido, através da equação matricial (1):

$$\begin{bmatrix} 0.2 & -0.2 & 1 \\ -0.2 & 0.3 & 0 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ i_{V1} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \quad (18)$$

Concluindo, este método veio introduzir a capacidade de ultrapassar o problema da condutância infinita provocado pelas Fontes ideais de Tensão sem incluir um elevado grau de complexidade, conferindo-lhe não só elevada eficiência como rapidez de cálculo, visto que são sistemas de equações que facilmente são resolvidos computacionalmente.

3.3. AS NETLISTS NOS SIMULADORES

Grande parte dos simuladores de circuitos elétricos atuais usam uma versão mais ou menos atualizada do Spice e, consequentemente, trabalham com netlists. Uma *netlist* é uma descrição do circuito simulado, que contém um conjunto de informações que descrevem a conectividade de um circuito incluindo componentes, Nós de ligação, polaridades, valores e outros parâmetros que podem influenciar o funcionamento do circuito.

Quando o SPICE foi lançado, as *netlists* eram utilizadas como ficheiros de *input* para simular um circuito [20], devido à inexistência de interfaces gráficas. Na Figura 14 é possível observar uma das primeiras versões do *PSpice*, onde o circuito é descrito através de um ficheiro de texto – a *netlist*. Nos simuladores atuais, as *netlists* continuam a ser uma possível forma de *input* de circuitos, que permite automatizar a criação e simulação de circuitos elétricos, no entanto, na perspetiva do utilizador, uma *netlist* em pouco ou nada auxilia o processo de simulação, face à elevada variedade e evolução das interfaces gráficas deste tipo de *softwares*, tal como foi demonstrado no capítulo 2.

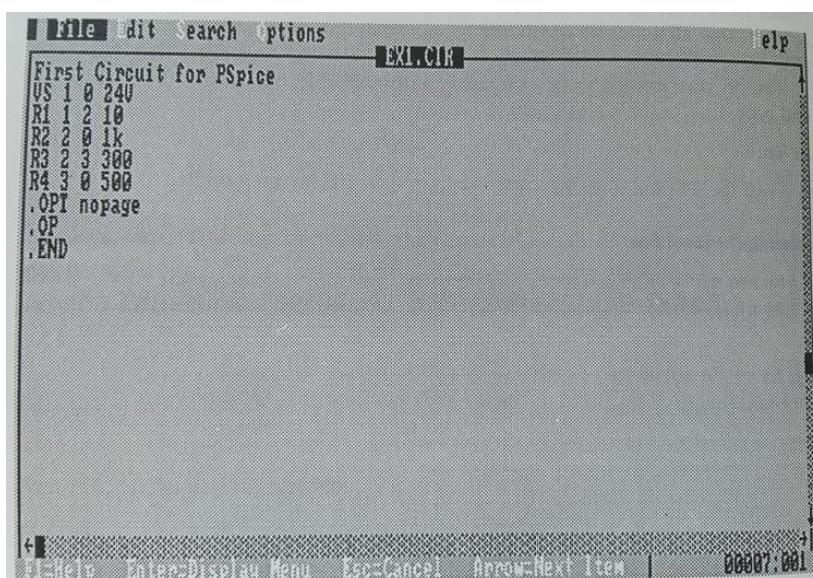


Figura 14 Circuito descrito por uma *netlist* no *PSpice* 5.1 (Beta) [19]

A base de uma *netlist* deve conter, tal como demonstra a Figura 14, a descrição do tipo de componente e o seu nome na primeira coluna. Na segunda e terceira coluna encontram-se os Nós de ligação aos quais cada componente se encontra conectado, e na última coluna o valor numérico de cada componente.

Neste ponto é possível perceber que é essencial para um *software* de simulação que exista um Nó (*net*), para cada ligação de dois ou mais componentes, só assim se torna possível analisar um circuito com eficácia. As *netlists* do QUCS [21] variam um pouco da que foi apresentada na Figura 14, no entanto a base é a mesma. De seguida encontra-se a *netlist* associada ao circuito exemplo da Figura 13, gerada pelo QUCS onde é possível verificar todos os componentes e conectividades entre eles.

```
Idc: I1 node_2 gnd I="1 mA"
R: R1 node_1 node_2 R="50 Ohm" Temp="26.85"
Tc1="0.0" Tc2="0.0" Tnom="26.85"
R: R2 gnd node_2 R="50 Ohm" Temp="26.85"
Tc1="0.0" Tc2="0.0" Tnom="26.85"
Vdc: V1 node_1 gnd U="1 V"
.DC: DC1 Temp="26.85" reltol="0.001" abstol="1 pA"
vntol="1 uV" saveOPs="no" MaxIter="150"
saveAll="no" convHelper="none" Solver="CroutLU"
```

Embora algumas das informações sejam irrelevantes para a análise efetuada pela aplicação U=RISOLVE, são importantes quando se trata de analisar circuitos que serão futuramente implementados na prática, tais como as variações da resistência para diferentes temperaturas e erros de convergência.

Filtrando a informação que é relevante para este estudo obtém-se a seguinte *netlist* (esta filtragem é uma das primeiras ações da aplicação U=RISOLVE, efetuada logo de seguida ao *upload* da *netlist*).

```
Idc: I1      node_2      gnd
R:   R1      node_1      node_2
R:   R2      gnd         node_2
Vdc: V1      node_1      gnd
```

Este simples ficheiro de texto caracteriza univocamente qualquer circuito, no entanto, algumas das informações podem estar subjacentes, por exemplo a identificação de um Nó real, a polaridade das Fontes ou até o número de Ramos do circuito, mas todas estas informações estão presentes na *netlist*:

- **Nós reais e Nós virtuais** – um Nó virtual liga apenas dois componentes, desta forma, só poderá estar presente duas vezes numa *netlist*. Começando pela Fonte de Tensão V_1 , que está conectada à terra e ao Nó 1 (*node_1*), e o único componente conectado ao Nó 1 é a resistência R_1 , logo o Nó 1 é um Nó virtual. A mesma lógica

se segue para o Nó 2 (node_2), este Nó está a ligar três componentes (I_1 , R_1 e R_2), logo aparecerá três vezes na *netlist*, o que faz dele um Nó real. Em suma, Nós reais constam três ou mais vezes numa *netlist* e Nós virtuais apenas duas.

- **Polaridades das Fontes** – nas Fontes de Tensão a polaridade positiva encontra-se sempre na primeira coluna dos Nós, neste caso, V_1 tem a polaridade positiva no Nó 1 e a negativa na terra. Nas Fontes de Corrente, o sentido do fluxo é da segunda coluna de Nós para a primeira, neste exemplo o sentido da Fonte de Corrente é da terra para o Nó 2.
- **Ramos do Circuito** – Um Ramo de um circuito é um percurso entre dois Nós (reais) consecutivos, logo basta fazer a análise da *netlist* para perceber quais são. Começando na Fonte de Corrente I_1 , desde o Nó 2 até à terra, fica completo o primeiro Nó. De seguida o Ramo que contém R_2 , inicia no Nó 2 e termina na terra – segundo Ramo. O último Ramo inicia também no Nó 2, passa por R_2 , até ao Nó 1, no entanto este Nó é virtual, portanto o Ramo não termina aqui, mas sim depois de V_1 , quando se chega novamente à terra, que é um Nó que, embora referenciado, não deixa de ser um Nó real.

Estas informações são apenas algumas das que se podem retirar de uma *netlist* e a lógica que foi feita para as obter nos três pontos anteriores, em muito se relaciona com a lógica utilizada para o desenvolvimento da aplicação. É possível perceber que a partir destes ficheiros de texto se pode gerar um algoritmo de análise, e é esse algoritmo que será estudado no próximo capítulo.

4. DESCRIÇÃO DA APLICAÇÃO U=RISOLVE

Esta é uma aplicação que pretende fomentar o processo de ensino na Engenharia Eletrotécnica focando-se na autoaprendizagem e, para tal, deve ter características que consigam captar a atenção dos utilizadores, que neste caso serão os próprios alunos. Seja a simplicidade na sua utilização, a interface apelativa, a rapidez de resposta, ou a metodologia de apresentação dos resultados, todos estes pormenores se revelam fulcrais para que uma aplicação seja realmente útil à comunidade estudantil. Neste capítulo é descrita a aplicação U=RISOLVE e como esta se relaciona com o utilizador de modo a que os resultados cheguem até ele, assim como as tecnologias que permitem adquirir a informação acerca dos circuitos a analisar. É ainda pormenorizado o processo algorítmico que sustenta a análise dos circuitos desde a aquisição da *netlist* proveniente do QUCS até à geração das equações relativas ao MTN. Por fim, é exposta a forma como a aplicação se relaciona com o utilizador através da *Graphical User Interface* (GUI).

4.1. LÓGICA, OBJETIVOS E ARQUITETURA

A aplicação U=RISOLVE, é uma aplicação implementada em *browser* e em puro *JavaScript*, o que lhe induz a versatilidade e robustez necessária para ser usufruída em

diversas plataformas. Trata-se de um *software* que providencia as equações ao utilizador sem que este necessite de fazer um esforço substancialmente superior ao de simular um circuito no simulador QUCS. De facto, após realizada a simulação, com este aplicativo, o aluno tem a possibilidade de obter uma análise totalmente focada para o ensino/didática à distância de alguns cliques. O processo é extremamente simples e traduz-se em três tarefas.

1. Gerar a *netlist* para o circuito simulado, seja no menu de simulação do QUCS, seja pressionando a tecla F6;
2. Guardar a *netlist* em formato de texto num determinado diretório do computador;
3. Fazer *upload* da *netlist* previamente gerada para a aplicação U=RISOLVE e pressionar o botão “Analisar Circuito”.

Este procedimento encontra-se representado, de uma forma superficial, na Figura 15 que ilustra a simplicidade da metodologia adotada para esta aplicação, uma metodologia necessariamente acessível a todo e qualquer utilizador, a partir do momento que é capaz de utilizar o QUCS como ferramenta de simulação, e tal tarefa, tal como já foi mencionado, é corretamente imposta aos alunos a partir do momento em que iniciam o seu estudo na Engenharia Eletrotécnica.

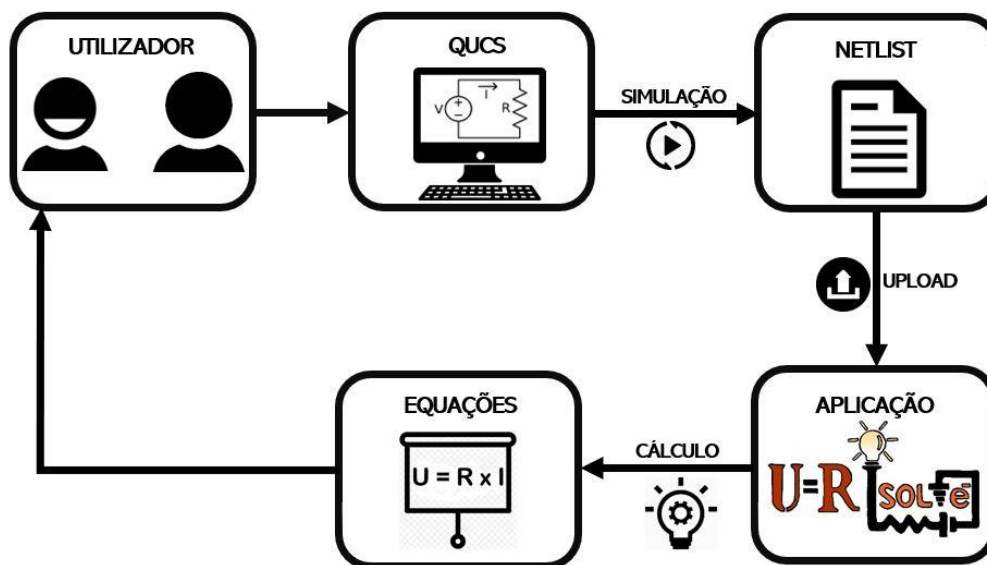


Figura 15 Lógica de utilização da aplicação U=RISOLVE

A apresentação dos resultados ao utilizador é feita numa janela *pop-up* o que permite simular vários circuitos sem que este tenha de abdicar das soluções para os circuitos

anteriores, o que permite não só realizar comparações de resultados como também analisar circuitos consecutivamente sem a necessidade de extrair os resultados das simulações anteriores. As equações são apresentadas por etapas, similarmente ao que acontecia na análise do MTN efetuado no capítulo anterior, permitindo um procedimento de análise de ordem crescente de complexidade de modo a facilitar a compreensão das equações. O processo de apresentação é o seguinte:

1. Inicialmente são apresentadas as equações dos Nós referentes à LKN onde cada Corrente é representada por um “i” seguido de um índice numérico que as diferencia.
2. De seguida são exibidas as equações para cada uma das Correntes em função dos potenciais nos Nós a que estão associadas, juntamente com os sentidos que foram arbitrados automaticamente e aos quais estas correspondem.
3. Por fim, são geradas as equações finais para cada um dos Nós, com todas as Correntes substituídas pelas expressões declaradas no ponto anterior.

Torna-se, portanto, necessário que o utilizador identifique todos os Nós reais do circuito, atribuindo-lhes um nome. Caso contrário, o QUCS atribuiria uma designação, que por *default* é “_netX” onde X representa um índice que é incrementado à medida que o *software* do QUCS deteta um Nó. Desta forma o utilizador não conseguiria identificar nem relacionar os Nós que estariam incluídos nos potenciais das equações geradas. Para superar esta problemática, sempre que o utilizador se equivocar e não nomear um Nó real, ou não referenciar um dos Nós reais, ou identificar/referenciar um Nó virtual, a aplicação reconhecerá essa incorreção e gerará um aviso com a informação acerca do erro que foi cometido. Este sistema permite clarificar a análise dos resultados, assim como obriga o aluno a saber identificar corretamente os Nós, o que numa fase introdutória aos circuitos elétricos, é fundamental.

A aplicação contém, adicionalmente, uma página de instruções para facilitar a introdução à utilização da mesma, e ainda um conjunto adicional de exemplos de circuitos matematicamente resolvidos e simulados no QUCS com a *netlist* pronta a ser carregada para a aplicação. Este conjunto adicional de exercícios tem vários graus de dificuldade e exemplos em Corrente Contínua e Alternada, permitindo, desta forma, motivar os alunos para o estudo.

Pretende-se assim que esta ferramenta seja uma mais valia para os alunos, e todas estas características e funcionalidades aqui apresentadas foram desenvolvidas nesse mesmo âmbito, o de proporcionar auxílio ao processo de aprendizagem.

4.2. O SISTEMA DE FICHEIROS DO JAVASCRIPT

A introdução da versão 5 do HTML, lançada pelo *World Wide Web Consortium* (W3C) em 2008, trouxe com ela um conjunto de *Application Programming Interfaces* (APIs) que revolucionaram a forma como os *websites* são programados e estruturados, conferindo-lhes novas funcionalidades, principalmente relacionadas com a acessibilidade.

A *File API* [22] é uma das interfaces associadas ao HTML5, e tem como objetivo fornecer os mecanismos necessários para manipular uma variada gama de *user inputs*, incluindo ficheiros de texto que os utilizadores tencionem carregar para um *website*/aplicação. Um *user input* é um campo específico num *website* que permite aos utilizadores realizar *uploads* de qualquer tipo de dados, seja um ficheiro, uma imagem ou até um email, tal como está representado no extrato de código seguinte:

```
<input type="file" id="fileToLoad">
```

Desta forma o utilizador visualizará uma secção que lhe permite pesquisar e carregar para a aplicação um ficheiro local tal como demonstra a Figura 16.

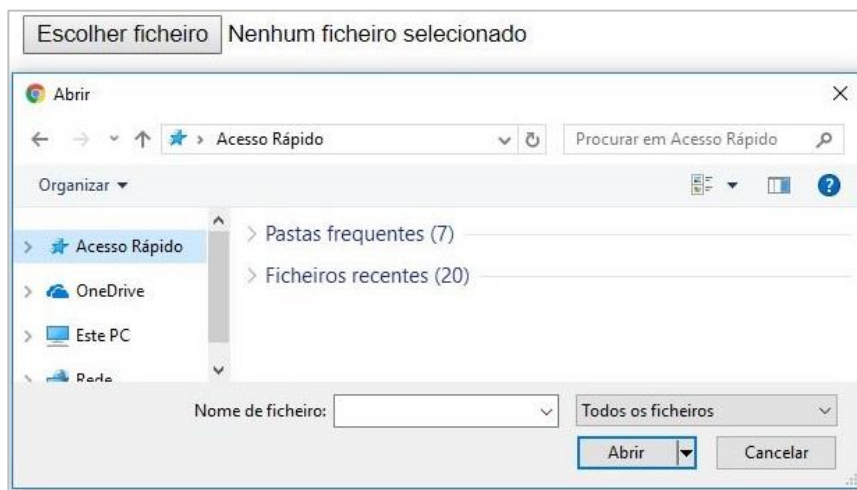


Figura 16 Exemplo do *upload* de um ficheiro no HTML5

Do lado da aplicação, esta conseguirá aceder ao ficheiro através do seu ID, neste exemplo, seria o “fileToLoad”. A *File API* possui ainda duas subinterfaces responsáveis pela manipulação de ficheiros: a *FileList* e a *FileReader*. A primeira, tal como o nome indica,

funciona como um vetor de ficheiros que vai aumentando à medida que o utilizador carrega os mesmos para a aplicação, esta metodologia é bastante útil quando a aplicação permite carregar vários ficheiros em simultâneo (o que não era possível antes desta versão do HTML). A *FileReader* é sem dúvida, a interface mais importante deste projeto, porque permite obter a informação de um ficheiro carregado para uma aplicação, possibilitando ainda executar um processo de monitorização de forma a encontrar erros e verificar se a leitura do ficheiro foi concluída com sucesso. Outra grande vantagem da interface *FileReader* é a de permitir ler em diversos formatos, seja formato de texto (*string* de texto), sequências binárias, formato de objeto ou estrutura de dados e outro tipo de codificações. No seguinte extrato de código (agora em *JavaScript*) pode comprovar-se que todas estas características foram fundamentais para a aplicação U=RISOLVE

```
function loadFileAsText(){
    var fileToLoad = document.getElementById("fileToLoad").files[0];
    var fileReader = new FileReader();

    fileReader.onload = function(fileLoadedEvent){
        var textFromFileLoaded = fileLoadedEvent.target.result;
        var leng = textFromFileLoaded.length;
    }

    fileReader.readAsText(fileToLoad, "UTF-8");
}
```

Inicialmente a função irá adquirir o ficheiro através do seu ID “fileToLoad” que se encontra na posição 0 do vetor da interface *FileList*. De seguida é invocada a função *FileReader*, que no momento que o ficheiro é carregado transfere a informação contida nele para a variável “textFromFileLoaded”.

A partir deste ponto o conteúdo desta variável é uma *string* de texto (devido à função “readAsText” parametrizada no objeto correspondente à interface *FileReader*) que pode ser usada, por exemplo, para calcular o tamanho do ficheiro, tamanho esse que, como é demonstrado no excerto de código, ficou guardado na variável “leng”.

Na aplicação U=RISOLVE o ficheiro a ser carregado trata-se da *netlist* para o circuito que se pretende obter as equações e, tal como foi mencionado anteriormente, a obtenção da informação contida no ficheiro é a primeira tarefa a executar pela aplicação e o primeiro contacto que o utilizador tem com a mesma. Assim, o sistema de ficheiros do *JavaScript* apresentado ao longo desta subsecção, foi fundamental na medida em que forneceu os mecanismos necessários para a aquisição da informação contida nas *netlists*. Adquirida a

informação, torna-se evidente que a próxima etapa será o procedimento de análise dos circuitos.

4.3. O ALGORITMO DE ANÁLISE

Numa conceção inicial não é perceptível a existência de uma metodologia de análise que possa ser automatizada de forma a gerar um algoritmo que resolva circuitos elétricos apenas a partir de um ficheiro de texto que contém a informação acerca da sua conectividade. Neste projeto foi idealizado um processo de análise para chegar até aos resultados desejados e pode ser estruturado em cinco etapas:

- Filtragem e organização da informação contida na *netlist*;
- Remoção de instrumentos de medição;
- Detecção de erros de identificação de Nós;
- Análise do circuito;
- Tratamento de Fontes isoladas de Tensão.

Esta estruturação permitiu organizar o trabalho com objetivos claros e que possibilitou um avanço contínuo da implementação do algoritmo, isto é, a informação que é tratada numa etapa continha, no momento da sua conclusão, o formato que a próxima etapa necessita para executar as suas funções e assim avançar no programa. De seguida serão apresentadas e pormenorizadas todas as fases do algoritmo concebido para sustentar a aplicação U=RISOLVE assim como o fluxograma que o representa, através da Figura 17.

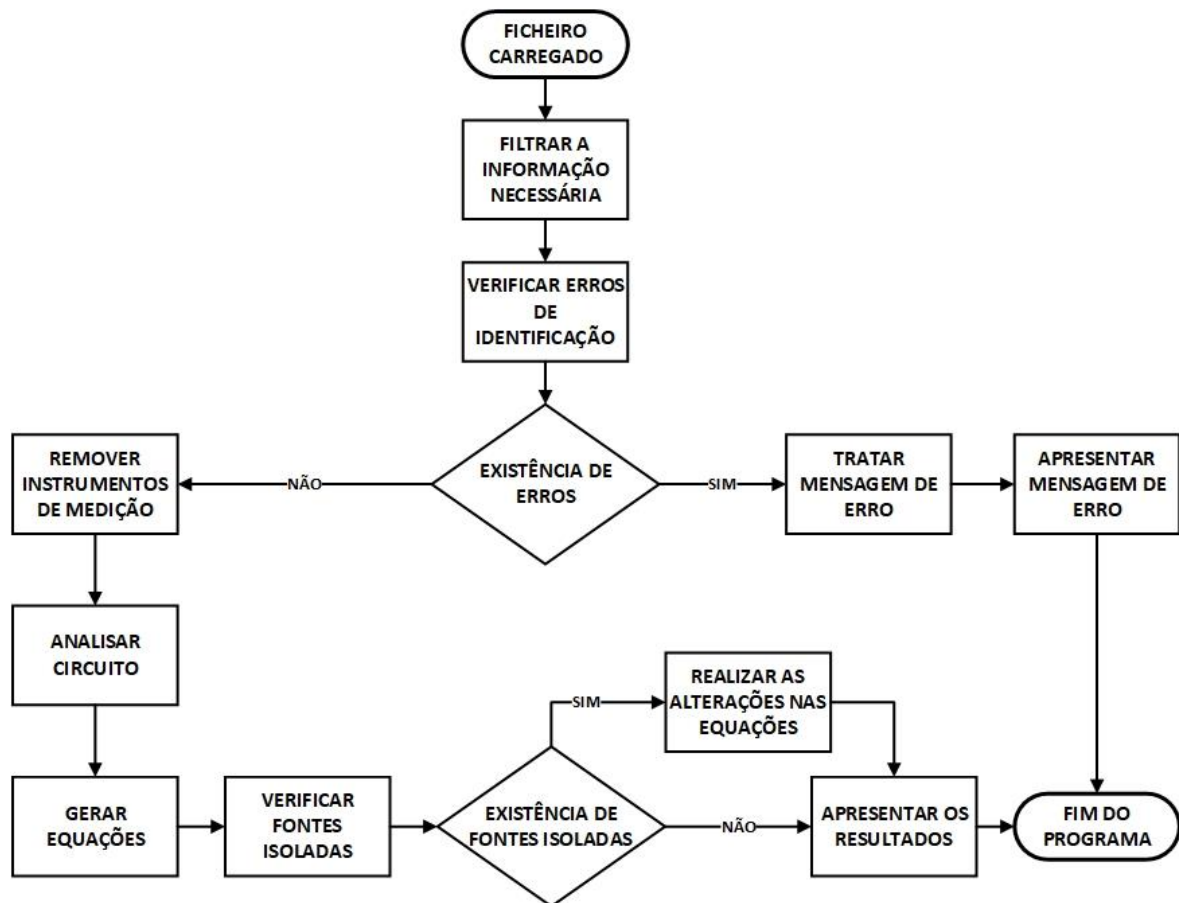


Figura 17 Fluxograma da aplicação U=RISOLVE

4.3.1. FILTRAGEM E ORGANIZAÇÃO DA INFORMAÇÃO

Esta etapa consiste na aquisição da informação, a ser retirada da *netlist*, que realmente será útil para procedimentos futuros. Assim, considerando novamente o circuito da Figura 13 e a sua *netlist* extraída do QUCS e após esta ter sido carregada para aplicação, o *software* irá procurar componentes, sejam Fontes de Tensão, resistências, condensadores, bobines ou Fontes de Corrente e irá agrupá-los. Só se torna possível encontrar os componentes devido ao formato *standard* das *netlists* que identifica distintivamente cada componente, por exemplo cada resistência existente no circuito dá origem a uma referência expressa por “R:” e as Fontes de Tensão e Corrente por “Vdc:” e “Idc:” respetivamente. Os componentes são de seguida agrupados em vetores, e cada componente é constituído por quatro vetores: vetor de nome, vetor do primeiro Nó de ligação, vetor do segundo Nó de ligação e vetor do seu valor numérico. Obtidos estes dados os vetores são agrupados, originando uma matriz de dados que será fundamental para todos os passos seguintes sem

exceção. A Figura 18 sintetiza esta etapa e é possível verificar que as suas linhas e colunas correspondem à junção dos vetores previamente indicados.

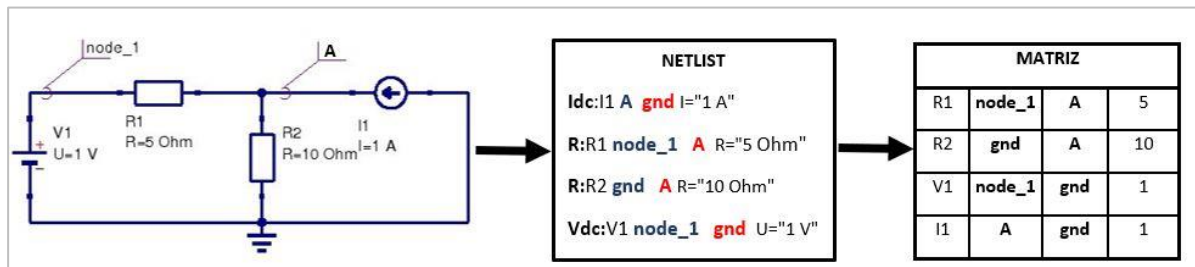


Figura 18 Aquisição da informação da *netlist* para uma matriz

4.3.2. REMOÇÃO DE INSTRUMENTOS DE MEDIÇÃO

No caso de existirem instrumentos de medição como amperímetros e volímetros, visto que são comumente usados nas simulações, é necessário que exista um procedimento de extrair esses componentes sem que essa extração altere as conectividades do circuito, o que conduziria a uma análise errada.

Supondo que, ao mesmo circuito do exemplo anterior (Figura 18), seriam acrescentados um volímetro e um amperímetro, essa inclusão alteraria a matriz que seria gerada pelo programa. Assim, o objetivo desta etapa é reverter a matriz alterada para a original, isto é, sem a influência dos aparelhos de medida.

Na Figura 19 é demonstrada a metodologia utilizada para efetuar a remoção dos aparelhos de medida e esta difere dos volímetros para os amperímetros devido à sua disposição nos circuitos elétricos, isto é, o volímetro encontra-se sempre em paralelo e o amperímetro em série.

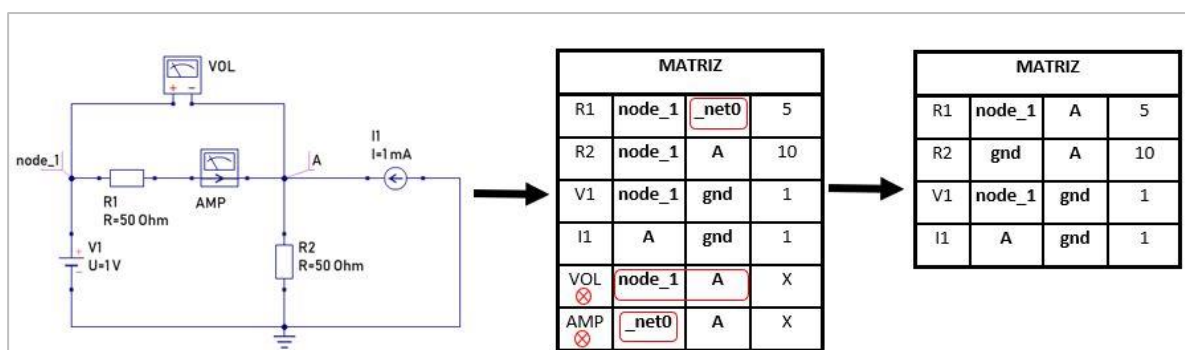


Figura 19 Método de remoção de instrumentos de medição

No caso dos voltímetros, o método de remoção é relativamente simples, porque, visto que se encontra em paralelo no circuito, basta remover a linha completa da matriz, porque o seu Ramo não afetará o normal funcionamento do circuito.

Os amperímetros, estando em série, requerem uma abordagem diferente, porque caso a linha do amperímetro seja removida da matriz, o circuito ficaria em aberto nesse Ramo. Observe-se, através da Figura 20, que o amperímetro se encontra conectado ao Nó A e ao Nó “_net0” (Nó gerado automaticamente pelo QUCS), e caso se remova a linha da matriz que identifica o amperímetro, iria continuar a existir um “_net0” na linha da resistência R1 que não estaria conectado a nenhum outro Nó.

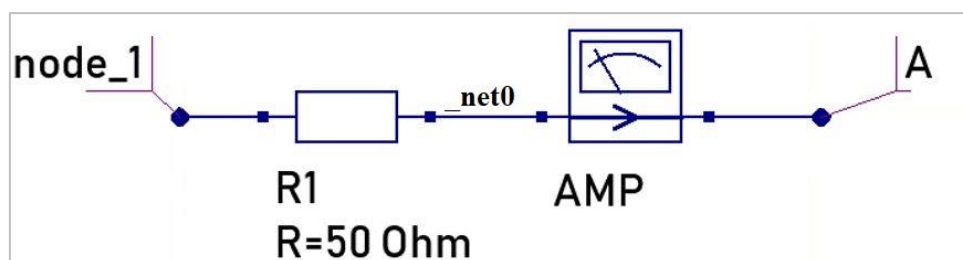


Figura 20 Remoção de amperímetros

Desta forma, é possível perceber que se torna necessário conectar a resistência R1 ao Nó A tal como inicialmente, no circuito sem instrumentos de medição. Para tal foi adotado o seguinte procedimento:

1. Escolher um Nó de ligação de amperímetro que não seja um Nó real, neste exemplo seria o Nó “_net0” e guardar o Nó que não foi escolhido, que no exemplo demonstrado seria o Nó “A”.
2. Procurar esse Nó na matriz, obviamente à exceção da linha do amperímetro, visto que só existirá uma vez. Neste exemplo encontra-se logo na primeira linha associado à resistência R1.
3. Substituir o Nó encontrado pelo Nó guardado, isto é, o Nó “A”.

Desta forma, tal como evidenciado na Figura 18, é possível obter a matriz inicial, sem que os instrumentos de medição influenciassem as conectividades dos componentes. Para efetuar a remoção de uma linha da matriz é usada a função *splice()* do *JavaScript* evidenciada no extrato de código seguinte:

```
matriz[col].splice(indice , 1);
```

Esta função permite aceder à coluna “col” da matriz e à linha “índice” e remover um *item* dessa posição. Visto que cada posição é apenas composta por um valor, seja uma *string* ou um valor numérico, a função removerá a posição inteira. Realizando este procedimento para todas as colunas, a linha ficará completamente removida, seja para os amperímetros ou voltímetros.

A grande vantagem desta função é complementar a flexibilidade da aplicação na medida em que permite que o utilizador simule o circuito sem qualquer tipo de restrições, podendo assim utilizar os instrumentos de medição para confrontar os resultados com as equações oferecidas pela U=RISOLVE.

4.3.3. DETEÇÃO DE ERROS DE IDENTIFICAÇÃO DE NÓS

No intuito de proporcionar uma análise objetiva e de fácil compreensão é necessário impor um conjunto de regras ao utilizador no momento em que está a delinear o circuito elétrico no QUCS. Este conjunto de regras inclui:

- Identificar todos os Nós reais com uma letra;
- Identificar o Nó de referência com o símbolo de *ground* existente no QUCS;
- Não identificar Nós virtuais.

Estas diretivas permitem incitar o utilizador a saber identificar e referenciar os Nós reais corretamente, que é um tópico de elevada importância no início do estudo da EE.

Esta etapa foi implementada novamente com recurso à matriz de dados gerada no momento da filtragem da informação proveniente da *netlist*, e para demonstrar a metodologia utilizada, esta será novamente exemplificada através do circuito das etapas anteriores.

Supondo que o utilizador comete todos os erros que foram apresentados, isto é, não identifica o Nó real, mas identifica o Nó virtual e esquece-se de referenciar um Nó. A matriz gerada será obviamente diferente e terá a informação necessária para identificar os erros cometidos, tal como evidencia a Figura 21.

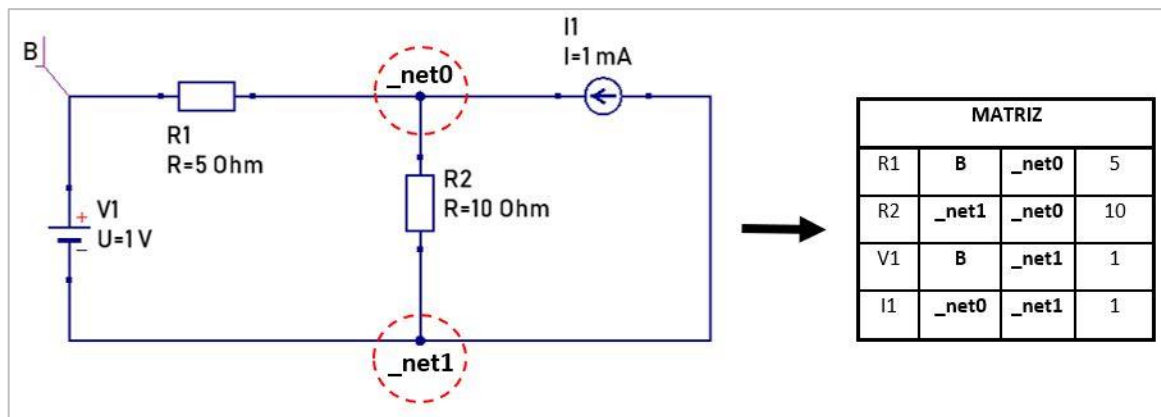


Figura 21 Circuito com erros de identificação de Nós

Neste circuito o Nó “B” está identificado e trata-se de um Nó virtual, e aos dois Nós reais que não foram identificados, foram-lhes atribuídos os nomes “_net0” e “_net1”. Para a aplicação identificar estes problemas é necessário analisar a matriz.

O erro de identificação de Nós virtuais é detetado da seguinte forma: um Nó virtual só tem duas entradas na matriz, portanto todos os Nós que tenham duas entradas na matriz e não comecem com o índice “_net” identificado automaticamente pelo QUCS, tratar-se-á de um Nó virtual identificado incorretamente. No exemplo da Figura 21 o Nó “B” é um Nó virtual que foi identificado indevidamente, porque surge apenas duas vezes na matriz e não começa com o índice “_net”.

Para detetar eventuais Nós reais que não estejam identificados o procedimento é relativamente semelhante ao anterior: um Nó real tem obrigatoriamente mais de duas entradas na matriz, desta forma, qualquer Nó com mais de duas entradas na matriz cujo nome inicie com “_net” será um Nó real do qual a identificação não foi efetuada. Na matriz representada na Figura 21 existem dois Nós que surgem mais que duas vezes na matriz, sendo estes o “_net0” e o “_net1”, e o facto de possuírem esta designação revela que não foram corretamente identificados.

Identificar a falta de *ground* no circuito é a deteção mais simples: basta verificar todas as posições da matriz e verificar se existe ou não existe a *string* “gnd”, e caso exista verificar se é ou não um Nó real.

Resta, no final da verificação, caso existam alguns destes erros, apresentá-los ao utilizador para que este proceda à correção do circuito. A exposição dos erros ao utilizador é realizada através de um método do *JavaScript* designado por *alert()*. Esta função, quando

invocada, gera uma caixa de alerta com uma determinada mensagem. Nesta aplicação a mensagem corresponde a uma *string* dinâmica que no final de cada verificação vai acrescentando o texto referente ao erro que foi detetado. No exemplo da Figura 21, todos os erros foram cometidos e, portanto, todos os erros estariam presentes na caixa de alerta. Este alerta contém ainda uma pequena lista de passos a seguir pelo utilizador de forma a saber como agir quando os erros surgirem, tal como mostra a Figura 22.

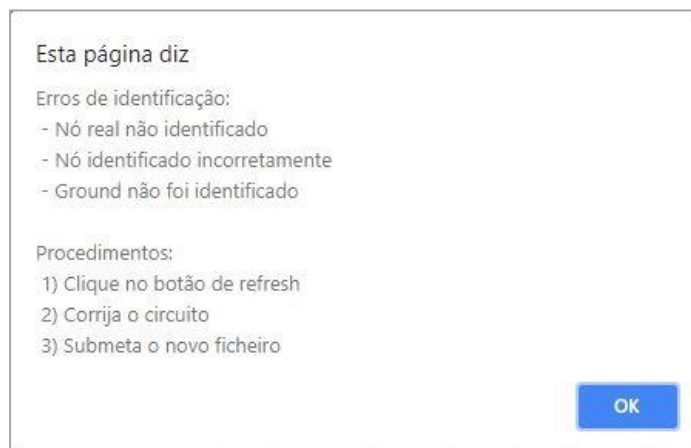


Figura 22 Caixa de alerta com erros de identificação

4.3.4. ANÁLISE DO CIRCUITO

A análise para cada circuito engloba o preenchimento de todas as Correntes do circuito com as informações necessárias para posteriormente proceder ao desenvolvimento das equações do circuito. Na aplicação cada Corrente é considerada um objeto, com diferentes campos que são preenchidos neste etapa, sendo que para cada circuito com n Correntes, existirá um vetor de objetos com n posições. As variáveis presentes em cada Corrente são:

- O nome – uma designação unívoca para cada Corrente;
- A impedância – inclui uma *string* com todas as resistências, condensadores ou bobines presentes nesse Ramo;
- A Tensão – trata-se de uma *string* que contém as Fontes de Tensão no mesmo Ramo, incluindo a polaridade de cada uma;
- A Corrente – é também uma *string* com as Fontes de Corrente que existem no Ramo, juntamente com a sua polaridade;

- O sentido – consiste num vetor de duas posições que contém os Nós que delimitam o Ramo, sendo que a corrente fluirá do Nó da primeira posição para o Nó da segunda posição do referido vetor.

Esta estrutura de dados para cada Corrente está exemplificada no extrato de código seguinte e trata-se de uma metodologia que retrata o circuito de uma forma eficaz e que será importante para toda a análise.

```
correntes[i] = {
    nome: ("i" + i),
    impedancia: "",
    tensao: "",
    corrente: "",
    sentido: new Array(),
};
```

O algoritmo que permite analisar o circuito, e consequentemente preencher todos os campos das Correntes para que se possa proceder à geração das equações do circuito pode, do ponto de vista de programação ser complexo, mas a sua lógica é de fácil compreensão quando acompanhada de um exemplo prático. Na Figura 23 está demonstrado o diagrama de blocos que representa, o algoritmo de análise dos circuitos. De seguida esta metodologia será exemplificada através de um circuito prático, de forma a facilitar a sua compreensão.

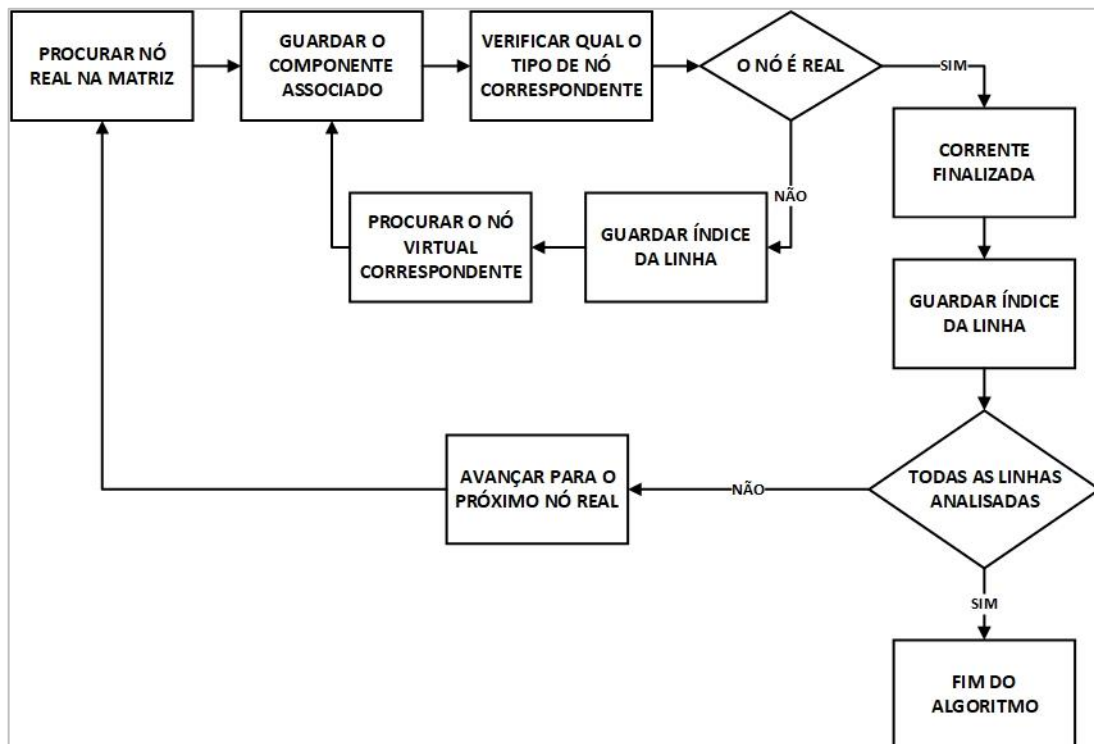


Figura 23 Diagrama de blocos da metodologia de análise

Para exemplificar o algoritmo de análise será utilizado novamente o circuito que tem vindo a servir de exemplo, no entanto, terá agora todos os Nós devidamente identificados, e está representado na Figura 24 com algumas indicações que irão auxiliar a interpretação do algoritmo.

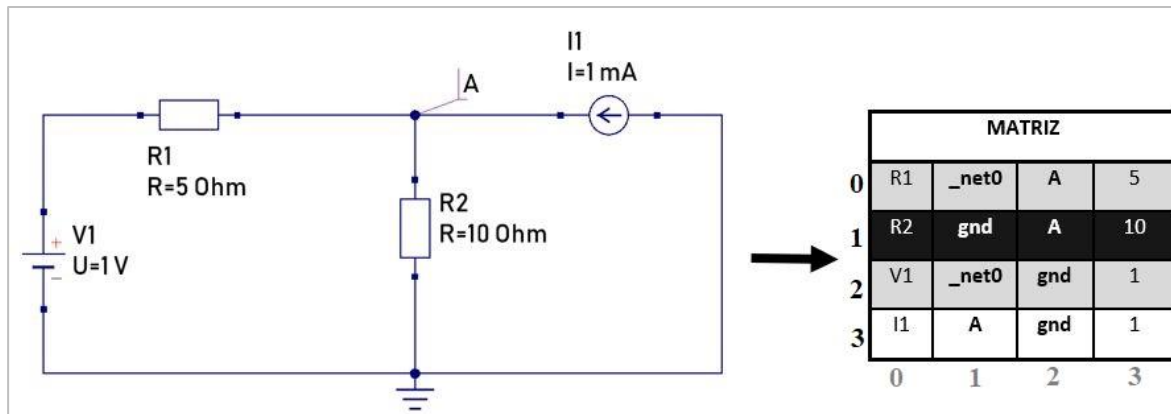


Figura 24 Método de análise de circuitos

O primeiro passo deste algoritmo, e seguindo o diagrama de blocos da Figura 23, trata-se de procurar um Nó real na matriz, assim, percorrendo a primeira coluna de Nós, encontra-se o Nó real “A” na terceira linha (linha preenchida a branco). De seguida trata-se o componente, que neste caso é uma Fonte de Corrente. Esta Fonte de Corrente preencheria o campo “corrente” do objeto de índice zero das Correntes (`correntes[0].corrente = “I1”`) com a polaridade negativa, devido ao facto de o programa começar a Corrente de “A” para “gnd” e a Fonte de Corrente ter o sentido inverso. Tratado o componente, verifica-se o Nó correspondente, isto é, o Nó ao qual a Fonte de Corrente está conectada, que neste exemplo será o Nó “gnd” – a terra. Visto que este Nó se trata de um Nó real (embora referenciado), a Corrente ficará finalizada e o índice de Correntes incrementará. O sentido da Corrente que acabou de ser analisada será de “A” para “gnd”, estes valores serão inseridos no vetor do campo “sentido” da Corrente (`correntes[0].sentido[0] = “A” & correntes[0].sentido[1] = “gnd”`).

Torna-se então necessário que esta linha que acabou de ser analisada (linha 3) seja guardada para que não interfira nas análises posteriores. Para tal é usado um vetor que será atualizado a cada linha que seja verificada (neste momento terá o valor “3” na primeira posição).

O passo seguinte é avançar na matriz para procurar o próximo Nó real, à exceção da linha que já foi analisada. O Nó real que se segue é também o Nó “A” (visto que só existe este

Nó no circuito), e encontra-se localizado na segunda coluna de Nós e na primeira linha. O componente associado é a resistência R_1 , que será introduzida no respetivo campo da Corrente em questão (`correntes[1].resistencia = R1`, note-se que o índice da Corrente já foi incrementado para 1).

Seguindo a lógica da Corrente anterior, é necessário verificar o Nó correspondente, que se trata do Nó “_net0”, que não é um Nó real. Neste caso, para além de guardar o índice da linha (linha 0) de forma a não ser verificada novamente, antes de ser possível avançar para o próximo Nó real, é necessário encontrar a localização do Nó virtual “_net0”. Este Nó encontra-se, tal como é visível na matriz exemplificada na Figura 24, na terceira linha, associada a uma Fonte de Tensão, logo, tratar-se-á a Fonte de Tensão, incluindo-a no campo “tensão” da Corrente em questão e voltar-se-á a verificar o Nó correspondente, que neste caso se trata do “gnd” e, portanto, a Corrente está finalizada. Note-se ainda que, neste momento o vetor de linhas analisadas terá o valor “[3, 0, 2]” porque as linhas de índice 3, 0 e 2 já foram verificadas.

Quanto à polaridade da Fonte de Tensão V_1 , esta será também negativa porque o sentido arbitrado da Corrente é de “A” para “gnd”. O programa deteta as polaridades, sabendo que, o terminal positivo se encontra sempre na primeira coluna de Nós, portanto, se o primeiro Nó que for encontrado, associado a uma Fonte de Tensão ou de Corrente, se encontrar na primeira coluna de Nós, a sua polaridade será negativa, se o contrário acontecer, isto é, o primeiro Nó encontrado relativo à Fonte de Corrente se encontrar na segunda coluna de Nós, tal significará que a polaridade da Fonte será positiva.

Restaria apenas o Ramo correspondente à linha 1 (sombreado a preto), que seria semelhante à situação da Fonte de Corrente, dado que ambos os Nós que delimitam esse Ramo são Nós reais. No final, o vetor de linhas analisadas estaria completo com o valor “[3,0,2,1]”, permitindo ao programa concluir que a análise do circuito estaria terminada. Para concluir, verifica-se que, na matriz, cada cor de sombreado (preto, branco e cinza) corresponde a um Ramo diferente.

Neste exemplo o circuito tem apenas um Nó real e existe apenas um Nó virtual, no entanto, é perceptível, neste ponto, que caso existissem inúmeros Nós virtuais e Nós reais, o processo repetir-se-ia, resolvendo normalmente o circuito. Este aspeto transmite, na realidade, a enorme flexibilidade do algoritmo adotado para efetuar esta análise. No

entanto a importância da metodologia utilizada nesta etapa será retratada na subsecção seguinte referente às geração das equações para o MTN.

4.3.5. TRATAMENTO DE FONTES ISOLADAS DE TENSÃO

Antes de se avançar para a geração das equações, é necessário efetuar a verificação de Fontes isoladas de Tensão de forma a evitar erros de condutâncias infinitas, provocados pelos Ramos que contêm essas Fontes isoladas. A detecção de Fontes isoladas é relativamente simples, e é realizada novamente com recurso à matriz de dados.

Supondo que, ao circuito exemplo utilizado anteriormente, se retirava a resistência R_1 e se acrescentavam dois Ramos, um deles com uma Fonte isolada de Tensão V_2 , tal como demonstra a Figura 25.

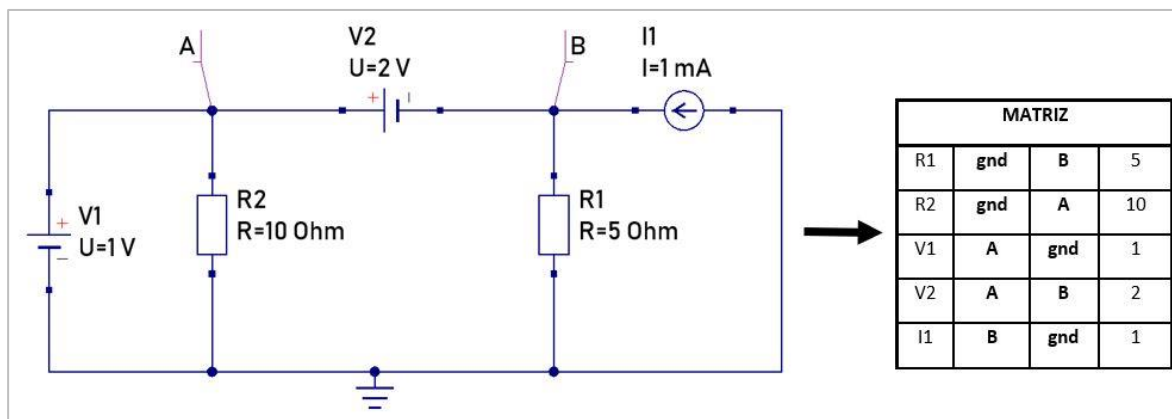


Figura 25 Circuito com Fontes isoladas de Tensão

Ao analisar o circuito, verifica-se que existem duas Fontes isoladas de Tensão, V_1 e V_2 , sendo que V_2 forma um Super-Nó “AB”. Para que a aplicação identifique este tipo de situações, o processo é o seguinte:

1. Procurar uma Fonte de Tensão existente no circuito.
2. Verificar se ambos os Nós de ligação da Fonte encontrada se tratam de Nós reais. Em caso afirmativo, significa que a Fonte está isolada, porque se encontra delimitada pelos dois Nós reais.
3. Em caso de existir Fonte isolada (verificada no ponto 2), apurar se algum dos Nós se trata da terra, isto é, se algum deles está referenciado. Se estiver, trata-se apenas de uma Fonte isolada, tal como acontece com a Fonte V_1 no exemplo da Figura 25.

Se nenhum dos Nós estiver referenciado, tal demonstra que se está perante uma situação de um Super-Nó, como sucede com a Fonte V_2 no exemplo anterior.

4. Repetir os passos anteriores para as restantes Fontes do circuito.

No final desta etapa, toda a informação acerca de Super-Nós e/ou Fontes isoladas será armazenada em variáveis, para que a geração das equações seja convenientemente efetuada, dado que as Fontes isoladas de Tensão alteram por completo as equações do MTN.

Está, desta forma, concluída a descrição do processo algorítmico de análise da aplicação $U=RISOLVE$. A nível de programação, a sua complexidade consistiu muito no alcance de uma abrangência total dos circuitos, isto é, um algoritmo destes deve ser implementado com meios que permitam resolver qualquer circuito, seja com exceções, com instrumentos de medição, com Fontes isoladas, com Fontes de Corrente, com inúmeros Nós e outras situações que se possam prever que venham a acontecer. A próxima fase, ao contrário da que foi apresentada nesta subsecção, está inteiramente relacionada com os resultados em si, ou melhor dizendo, as equações para o Método das Tensões nos Nós.

4.4. GERAÇÃO DAS EQUAÇÕES

Gerar as equações implica apresentar os resultados, e apresentar os resultados requer um elevado grau de clareza e objetividade. Os resultados na aplicação $U=RISOLVE$ são estruturados com um nível crescente de profundidade, isto é, inicialmente apresentam-se as equações dos Nós, segundo a LKN, de seguida são evidenciadas as expressões para cada uma das Correntes, juntamente com o seu sentido e, por fim, as Correntes das equações dos Nós são substituídas pelas expressões de cada Corrente, originando as equações finais. De seguida serão descritos os processos pelos quais os três níveis de equações referidos são calculados e apresentados ao utilizador e ainda a metodologia pela qual são tratadas as Fontes isoladas de Tensão.

4.4.1. EQUAÇÕES DOS NÓS

O cálculo das equações de cada Nó é efetuado através de dois indicadores, um deles é o campo “sentido” de cada Corrente e o outro é um vetor que contém todos os Nós reais no circuito. Assim, primeiramente é definido um Nó real existente no vetor, em seguida procura-se por esse Nó no campo “sentido” de todas as Correntes do circuito, e

dependendo da sua posição, a Corrente ficará no primeiro ou segundo membro da equação do Nó. Na Figura 26 encontra-se exemplificado o processo de obtenção das equações dos Nós, através dos campos referidos das Correntes.

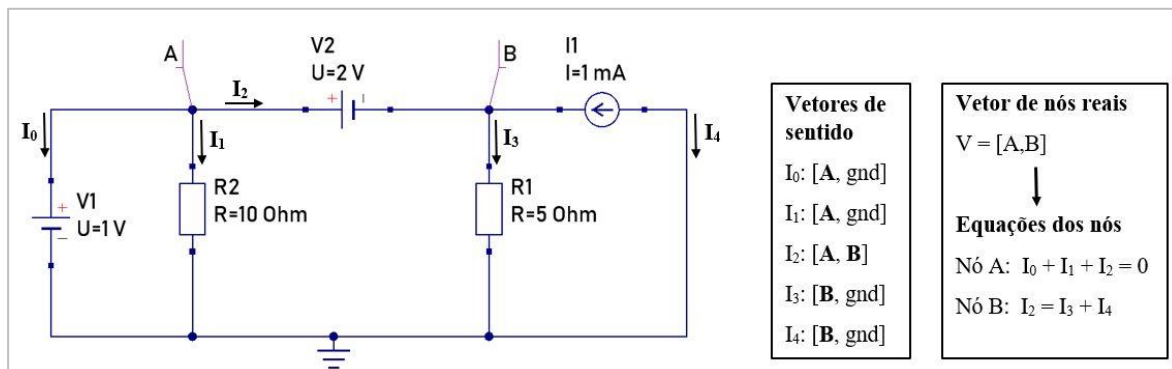


Figura 26 Geração das equações dos Nós

No exemplo apresentado encontram-se os vetores de sentido para cada Corrente e as equações dos Nós derivadas desse vetor. Assim, o método de análise para gerar as equações dos Nós, consiste em fixar inicialmente o Nó “A” e de seguida o Nó “B”, que são os únicos Nós reais do circuito.

Para o Nó “A”, verifica-se que se encontra presente em I_0 , I_1 e I_2 , sempre posicionado na primeira posição do vetor de sentido. A primeira posição deste vetor indica que é o Nó a partir do qual a Corrente parte. Logo, para o Nó “A”, todas as Correntes estão a divergir, e então a equação do Nó terá todas as Correntes num único membro.

Já para o Nó “B”, que se encontra nas Correntes I_2 , I_3 e I_4 , embora que nas Correntes I_3 e I_4 o Nó esteja presente na primeira posição do vetor (Correntes a divergir do Nó), no caso da Corrente I_2 , o Nó “B” encontra-se na segunda posição do vetor, logo estará a convergir para o Nó, ao contrário do que acontece nas Correntes anteriores. Desta forma, num dos membros estará a Corrente I_2 e no outro encontrar-se-ão as Correntes I_3 e I_4 tal como demonstra a equação do Nó “B” na Figura 26.

4.4.2. EQUAÇÕES DAS CORRENTES

Gerar as expressões para cada Corrente implica calcular o seu numerador e denominador de acordo com a lei de *Ohm*. As informações necessárias para calcular as frações encontram-se nos campos “impedância”, “corrente” e “tensão” de cada uma das Correntes do circuito.

Para exemplificar esta etapa, encontra-se representado na Figura 27, um Ramo delimitado por dois Nós “A” e “B” e composto por três resistências e três Fontes de Tensão.

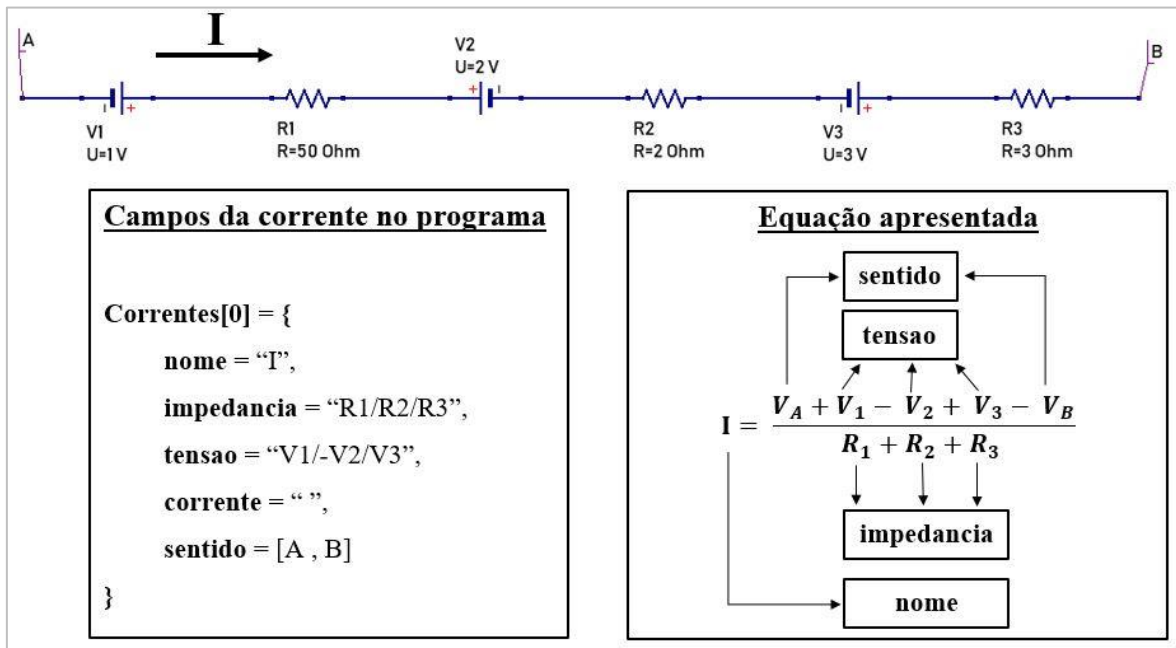


Figura 27 Geração das equações das Correntes

Neste exemplo é possível verificar como é que os campos do objeto “Correntes” influenciam as expressões para cada Corrente, visto que todos eles interferem na equação apresentada. Embora este Ramo não possua uma Fonte de Corrente, no caso da sua existência, a expressão seria substituída pelo nome da Fonte de Corrente, obviamente de acordo com a sua polaridade.

4.4.3. EQUAÇÕES FINAIS

As equações finais tratam-se da junção das equações dos Nós com as expressões das Correntes, isto é, as expressões de cada Corrente são substituídas nas equações dos Nós. A aplicação efetua esta etapa com recurso aos nomes de cada Corrente, ou seja, o programa analisa a equação do Nó que é composta pelos nomes das Correntes, sinais de mais ou menos, um igual e, porventura, um zero, no caso de todas as Correntes serem divergentes ou convergentes para o mesmo Nó.

Sabendo que, a cada Corrente, está associada uma expressão para o numerador e outra para o denominador, basta que o algoritmo identifique o nome de uma Corrente, e automaticamente essa Corrente será substituída pelo numerador e denominador correspondente, sendo que os restantes elementos da equação (sinais e eventuais zeros)

permanecerão intactos, devido ao facto de a análise para cada equação do Nó, ser uma análise progressiva que vai detetando Correntes à medida que avança na equação.

4.4.4. EQUAÇÕES COM FONTES ISOLADAS DE TENSÃO

As Fontes isoladas de Tensão alteram por completo as equações do MTN e, portanto, esta aplicação possui um método que permite resolver o problema das condutâncias infinitas.

No caso de uma Fonte isolada de Tensão, em que um dos Nós está referenciado, tal como demonstrado anteriormente são alteradas as seguintes equações:

- Equação das Correntes: a Corrente que passa na Fonte isolada de Tensão não terá uma expressão associada, e conterà apenas o seu nome, isto é, a Corrente pode ser calculada, no final, através da equação dos Nós, sendo assim uma Corrente incógnita.
- Equação final: nesta equação a Corrente que passa na Fonte isolada, não terá uma expressão associada, e o programa irá detetar esse acontecimento, desta forma, a Corrente não será substituída por uma expressão, mas sim pelo seu nome. Na equação final irá estar presente a Tensão conhecida no Nó ao qual a Fonte de Tensão está diretamente conectada.

Com efeito, irá aumentar o número de variáveis, porque a Corrente que percorre o Ramo da Fonte de Tensão isolada não será conhecida até se calcularem todas as restantes Correntes do circuito, no entanto, o facto de se conhecer a Tensão num dos Nós devido à Fonte isolada de Tensão, equilibra novamente o número de variáveis do circuito.

Na eventualidade de ambos os Nós aos quais a Fonte de Tensão isolada se encontra conectada serem Nós reais e nenhum deles estar referenciado, tal como demonstrado anteriormente, tratar-se-á de um Super-Nó. Os Super-Nós também alteram a disposição das equações e essa modificação representa-se da seguinte forma:

- Equação dos Nós: a Corrente que passa pelo Super-Nó considera-se inexistente (considerando que a Fonte de Tensão se trata de uma Fonte ideal), e todas as Correntes que estiverem associadas aos dois Nós que formam o Super-Nó, passarão a constituir uma única equação, isto é, por exemplo as Correntes que convergem para um Super-Nó “AB” serão a soma das Correntes que convergem para o Nó “A”

com as Correntes que convergem para o Nó “B”, e o mesmo acontece para as Correntes que divergem do Super-Nó.

- Equação das Correntes: a Corrente que passa no Super-Nó estará oculta dos resultados devido ao facto de se considerar que esta é inexistente.
- Equação final: o resultado final engloba as equações dos Nós substituídas pelas Correntes, aqui não há nenhuma alteração, porque equações dos Nós foram previamente reformuladas, e o programa seguirá o seu normal funcionamento. É apenas introduzida, junto com a equação final, a equação do Super-Nó, demonstrada no capítulo 3, em função da Fonte de Tensão isolada.

Termina assim a demonstração de todas as metodologias executadas pela aplicação U=RISOLVE desde a aquisição da informação do circuito, até à geração das equações para o Método da Tensão nos Nós. Todo este processo foi idealizado para facilitar a compreensão do utilizador, portanto, resta demonstrar como é que a aplicação que é apresentada neste projeto se relaciona, graficamente, através da sua interface, com o utilizador.

4.5. INTERFACE GRÁFICA COM O UTILIZADOR (GUI)

A interface gráfica de uma página *web*, principalmente para uma aplicação de âmbito educacional, que irá auxiliar alunos do ensino superior, deve ser o mais apelativa possível. O estilo de um *website* é configurado e personalizado através de uma linguagem designada por *Cascading Style Sheets* (CSS). Uma folha de estilos é um documento que contém mecanismos para editar uma página *web* e, hoje em dia, existem inúmeras possibilidades de edição, desde cores, formas, imagens de fundo, botões, tabelas, divisórias e animações.

À exceção do logótipo da aplicação U=RISOLVE, todos os estilos dos restantes elementos gráficos da aplicação foram editados utilizando CSS, um destes elementos são os botões, que têm uma forma específica e alteram a sua visualização de acordo com a posição do rato. A Figura 28 exemplifica um dos botões editados para a aplicação U=RISOLVE, juntamente com o código CSS e HTML necessário para tornar possível a configuração apresentada.

CSS	HTML
<pre> .button { background-color: #e65c00; color: #4d4d33; padding: 20px 44px; text-align: center; text-decoration: none; display: inline-block; font-size: 18px; border-radius: 12px; cursor: pointer; font-family: "Lucida Console"; font-weight: bold; } .button:hover { background-color: #4d4d33; color: #e65c00; } </pre>	<pre><button class="button"> Analisar Circuito </button></pre>
	RESULTADO  Indicador fora do botão  Indicador em cima do botão

Figura 28 Configuração dos botões

Na página de instruções, também a numeração de cada etapa sofreu alterações através de CSS, sendo-lhe atribuído o formato representado na Figura 29.

CSS	HTML
<pre> .numberCircle { border-radius: 50%; width: 36px; height: 36px; padding: 8px; background: #fff; border: 4px solid #666; color: #666; text-align: center; font: 32px Arial, sans-serif; } </pre>	<pre><div class="numberCircle">1</div></pre>
	RESULTADO 

Figura 29 Configuração da numeração da página de instruções

Na página de resultados, a existência de frações horizontais, deve-se também à utilização de CSS tal como demonstra a Figura 30, sendo sem dúvida o elemento mais importante, pois sem as frações, a apresentação de resultados não teria o mesmo impacto nem seria apazível de analisar.

CSS	HTML
<pre> .fraction { display: inline-block; vertical-align: middle; margin: 0 0.2em 0.4ex; text-align: center; } .fraction > span { display: block; padding-top: 0.15em; } .fraction span.fdn {border-top: thin solid black;} .fraction span.bar {display: none;} </pre>	<pre> <div class="fraction"> 100 500 </div> </pre>
	RESULTADO
	$\frac{100}{500}$

Figura 30 Fração horizontal com uso de CSS

Embora existam outras situações em que o CSS foi utilizado na aplicação, tais como fundos de ecrã, fundos de ecrã repetitivos, posicionamento de divisões e inserção de imagens, as que foram aqui apresentadas são as essenciais na medida em que expõem a grande maioria da programação CSS utilizada.

Quanto à interface gráfica em si, o primeiro contacto que o utilizador tem com a aplicação é a página inicial da U=RISOLVE, onde poderá encontrar, botões de acesso à página de instruções, de *upload* da *netlist*, de analisar o circuito e ainda um botão de *refresh* da página, tal como está evidenciado na Figura 31.



Figura 31 Página inicial da aplicação U=RI SOLVE

Ao carregar no botão de instruções, o utilizador depara-se com uma página com oito simples etapas a seguir para a correta utilização da aplicação. Esta página contém ainda botões de acesso à página inicial, tal como demonstra a Figura 32.

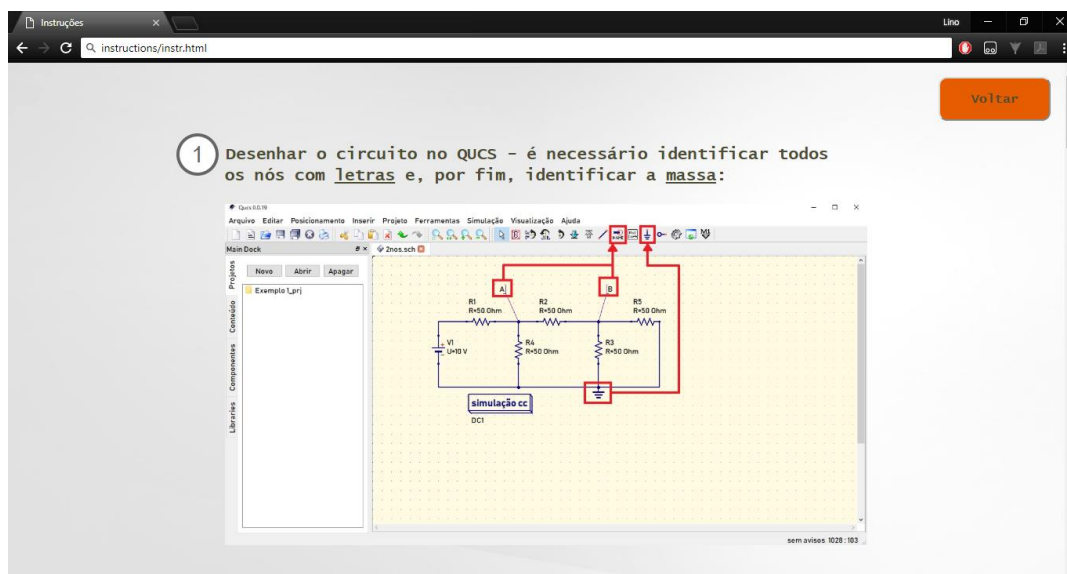


Figura 32 Página de instruções

Para simular um circuito basta que o utilizador pressione o botão selecionador de ficheiros, que se trata da *netlist* do circuito, fazer *upload* do mesmo, e clicar em “Analisar Circuito”. A página de resultados é uma janela *pop-up* e apresenta, tal como mencionado anteriormente, todas as equações dos Nós, das Correntes e equações finais. O exemplo da

Figura 33 remete para um circuito com Fontes de Tensão isoladas e Super-Nós, para que seja possível visualizar como é que a aplicação se comporta nestes casos.

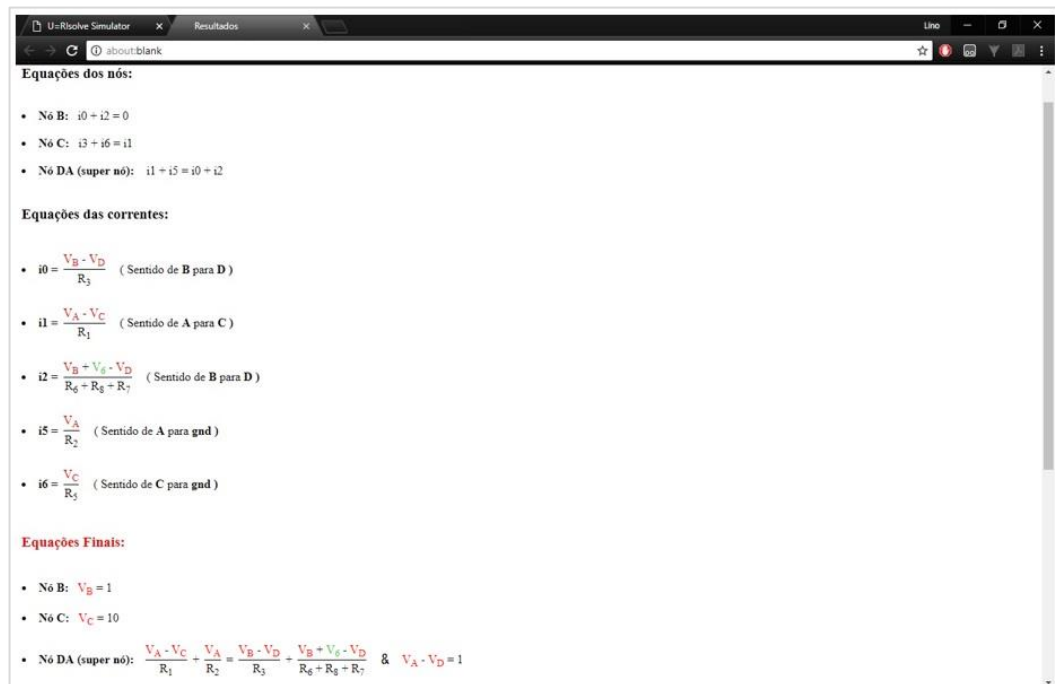


Figura 33 Janela de resultados

É possível verificar que a interface gráfica da aplicação aqui apresentada é de elevada simplicidade e as equações dos circuitos estão à distância de cerca de quatro cliques. Este fator foi tido em grande consideração no momento em que a aplicação estava a ser desenvolvida, pois o objetivo providenciar uma ferramenta de apoio ao estudo, que fomenta não só a rapidez e eficácia de resolução de exercícios, mas também a autoaprendizagem e, portanto, uma interface de ampla simplicidade, torna-se uma mais valia para que os alunos possam realmente utilizar a aplicação U=RESOLVE.

5. A UTILIZAÇÃO DA APLICAÇÃO

Este capítulo é dedicado à demonstração da aplicação, de forma a explorar ao máximo as suas capacidades, com o intuito de comprovar o seu bom funcionamento. Os circuitos que aqui serão apresentados serão disponibilizados na aplicação, não só para que num momento inicial possa atrair os estudantes para a sua utilização, mas também para que a aplicação U=RISOLVE seja vista como uma ferramenta de estudo. Para tal, torna-se imperial fornecer aos alunos um ponto de partida. Assim, todos os circuitos facultados são acompanhados do seu esquemático do QUCS, para possibilitar uma maior rapidez na simulação; a *netlist* do circuito, proporcionando uma instantânea utilização da aplicação; e uma imagem do circuito.

5.1. CASOS-EXEMPLO EM CORRENTE CONTÍNUA

Os circuitos de seguida exibidos foram ordenados por ordem de dificuldade, para oferecer um método de estudo estruturado. O primeiro circuito está representado na Figura 34 e o resultado da aplicação U=RISOLVE encontra-se na Figura 35.

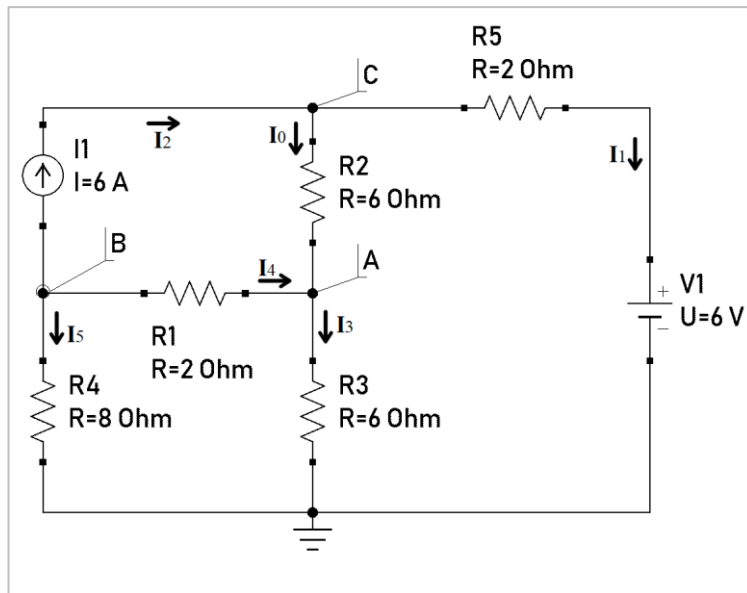


Figura 34 Circuito de nível básico em Corrente Contínua

Equações dos nós:

- Nó C: $i_0 + i_1 = i_2$
- Nó B: $i_2 + i_4 + i_5 = 0$
- Nó A: $i_3 = i_0 + i_4$

- $i_4 = \frac{V_B - V_A}{R_1}$ (Sentido de B para A)

- $i_5 = \frac{V_B}{R_4}$ (Sentido de B para gnd)

Equações das correntes:

- $i_0 = \frac{V_C - V_A}{R_2}$ (Sentido de C para A)

- $i_1 = \frac{V_C - V_1}{R_5}$ (Sentido de C para gnd)

- $i_2 = I_1$ (Sentido de B para C)

- $i_3 = \frac{V_A}{R_3}$ (Sentido de A para gnd)

Equações Finais:

- Nó C: $\frac{V_C - V_A}{R_2} + \frac{V_C - V_1}{R_5} = I_1$

- Nó B: $I_1 + \frac{V_B - V_A}{R_1} + \frac{V_B}{R_4} = 0$

- Nó A: $\frac{V_A}{R_3} = \frac{V_C - V_A}{R_2} + \frac{V_B - V_A}{R_1}$

Figura 35 Solução U=RISOLVE do circuito nível básico em Corrente Contínua

Na Figura 36 encontra-se o segundo circuito em Corrente Contínua (nível intermédio), no qual constam Fontes de Tensão isoladas e Super-Nós. A solução fornecida pela aplicação U=RISOLVE para o respetivo circuito está demonstrada na Figura 37.

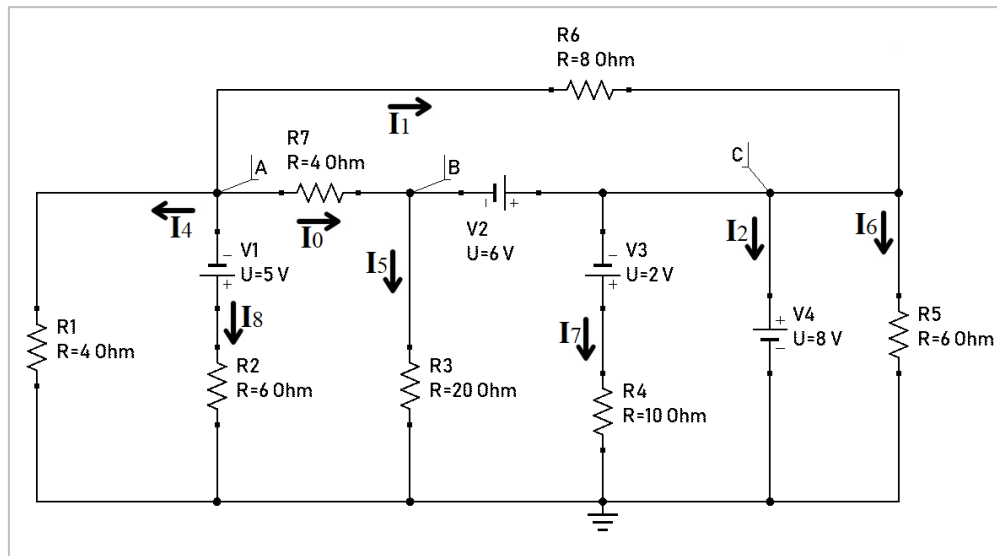


Figura 36 Circuito de nível intermédio em Corrente Contínua

Equações dos nós:

- Nó BC (super nó): $i_0 + i_5 + i_2 + i_6 + i_7 = i_1$
- Nó A: $i_1 + i_4 + i_8 = i_0$
- $i_4 = \frac{V_A}{R_1}$ (Sentido de A para gnd)
- $i_5 = \frac{V_B}{R_3}$ (Sentido de B para gnd)

Equações das correntes:

- $i_0 = \frac{V_B - V_A}{R_7}$ (Sentido de B para A)
- $i_1 = \frac{V_A - V_C}{R_6}$ (Sentido de A para C)
- $i_6 = \frac{V_C}{R_5}$ (Sentido de C para gnd)
- $i_7 = \frac{V_C + V_3}{R_4}$ (Sentido de C para gnd)
- $i_8 = \frac{V_A + V_1}{R_2}$ (Sentido de A para gnd)

Equações Finais:

- Nó BC (super nó): $\frac{V_B - V_A}{R_7} + \frac{V_B}{R_3} + i_2 + \frac{V_C}{R_5} + \frac{V_C + V_3}{R_4} = \frac{V_A - V_C}{R_6}$ & $V_C - V_B = 6$ & $V_C = 8$ V
- Nó A: $\frac{V_A - V_C}{R_6} + \frac{V_A}{R_1} + \frac{V_A + V_1}{R_2} = \frac{V_B - V_A}{R_7}$

Figura 37 Solução U=RISOLVE do circuito nível intermédio em Corrente Contínua

Para finalizar a demonstração da aplicação para circuitos em Corrente Contínua, encontra-se apresentado na Figura 38 o circuito de nível avançado, juntamente com a solução proveniente da aplicação na Figura 39.

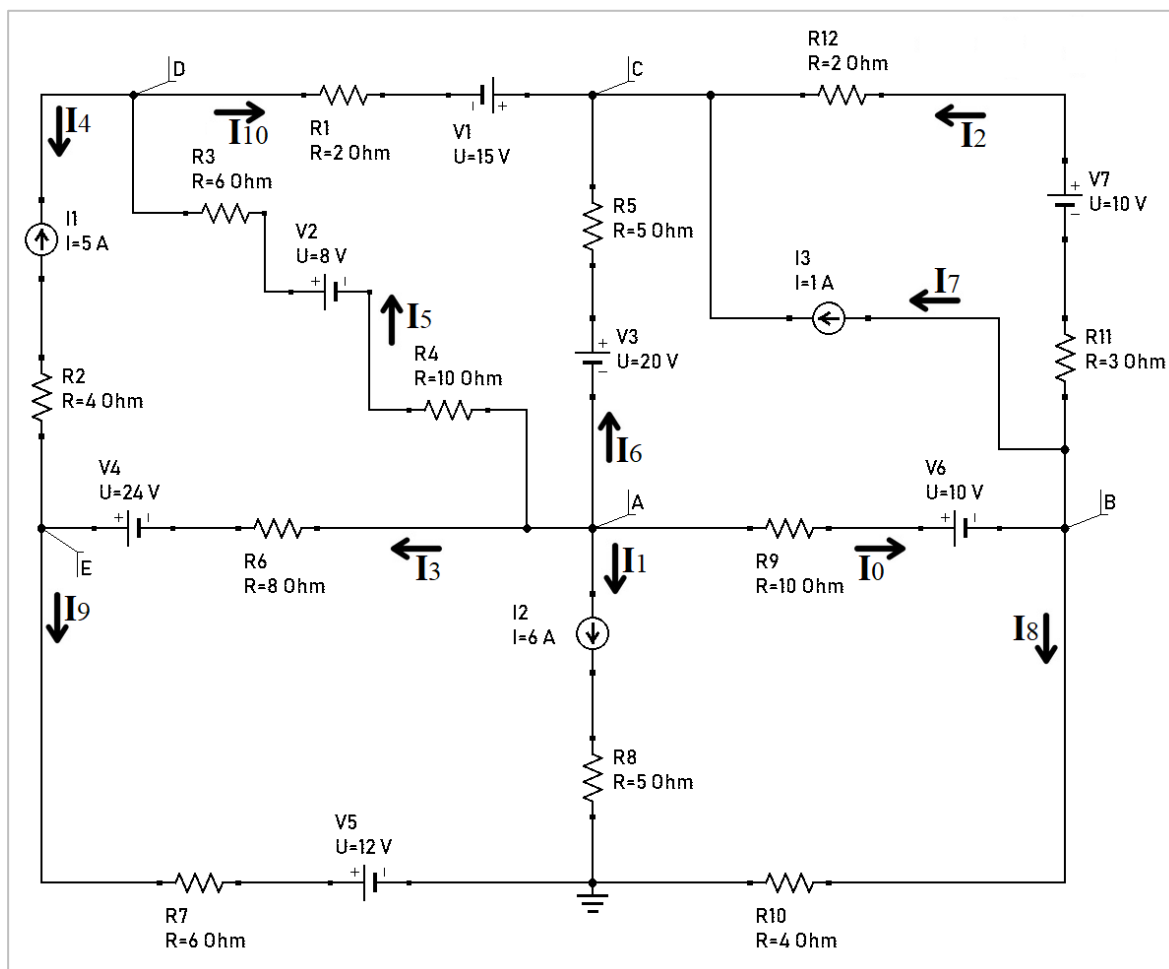


Figura 38 Circuito de nível avançado em Corrente Contínua

Equações dos nós:	
• Nó A: $i_0 + i_1 + i_3 + i_5 + i_6 = 0$	• $i_7 = I_3$ (Sentido de B para C)
• Nó B: $i_2 + i_7 + i_8 = i_0$	• $i_8 = \frac{V_B}{R_{10}}$ (Sentido de B para gnd)
• Nó E: $i_9 = i_3 + i_4$	• $i_9 = \frac{V_E - V_5}{R_7}$ (Sentido de E para gnd)
• Nó D: $i_4 + i_{10} = i_5$	• $i_{10} = \frac{V_D + V_1 - V_C}{R_1}$ (Sentido de D para C)
• Nó C: $0 = i_2 + i_6 + i_7 + i_{10}$	
Equações das correntes:	
• $i_0 = \frac{V_A - V_6 - V_B}{R_9}$ (Sentido de A para B)	Equações Finais:
• $i_1 = I_2$ (Sentido de A para gnd)	• Nó A: $\frac{V_A - V_6 - V_B}{R_9} + I_2 + \frac{V_A + V_4 - V_E}{R_6} + \frac{V_A + V_2 - V_D}{R_4 + R_3} + \frac{V_A + V_3 - V_C}{R_5} = 0$
• $i_2 = \frac{V_B + V_7 - V_C}{R_{11} + R_{12}}$ (Sentido de B para C)	• Nó B: $\frac{V_B + V_7 - V_C}{R_{11} + R_{12}} + I_3 + \frac{V_B}{R_{10}} = \frac{V_A - V_6 - V_B}{R_9}$
• $i_3 = \frac{V_A + V_4 - V_E}{R_6}$ (Sentido de A para E)	• Nó E: $\frac{V_E - V_5}{R_7} = \frac{V_A + V_4 - V_E}{R_6} - I_1$
• $i_4 = -I_1$ (Sentido de D para E)	• Nó D: $-I_1 + \frac{V_D + V_1 - V_C}{R_1} = \frac{V_A + V_2 - V_D}{R_4 + R_3}$
• $i_5 = \frac{V_A + V_2 - V_D}{R_4 + R_3}$ (Sentido de A para D)	• Nó C: $0 = \frac{V_B + V_7 - V_C}{R_{11} + R_{12}} + \frac{V_A + V_3 - V_C}{R_5} + I_3 + \frac{V_D + V_1 - V_C}{R_1}$
• $i_6 = \frac{V_A + V_3 - V_C}{R_5}$ (Sentido de A para C)	

Figura 39 Solução U=RISOLVE do circuito nível avançado em Corrente Contínua

5.2. CASOS-EXEMPLO EM CORRENTE ALTERNADA

Em Corrente Alternada optou-se pela mesma estratégia, sendo apresentados os circuitos novamente por ordem crescente de complexidade. O circuito básico em Corrente Alternada selecionado encontra-se representado na Figura 40, sendo que na Figura 41, tal como aconteceu na subsecção anterior, surge a solução calculada pela aplicação.

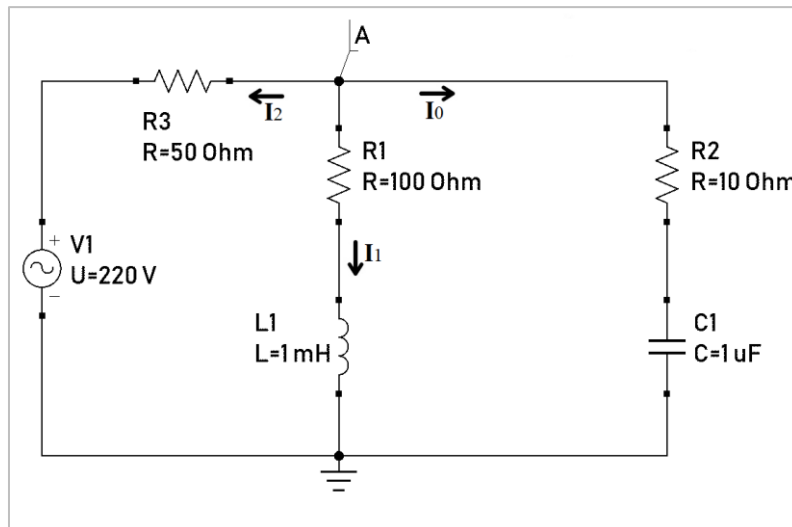


Figura 40 Circuito de nível básico em Corrente Alternada

Equações dos nós:

- Nó A: $i_0 + i_1 + i_2 = 0$

Equações das correntes:

- $i_0 = \frac{V_A}{R_2 + Z_{C1}}$ (Sentido de A para gnd)
- $i_1 = \frac{V_A}{R_1 + Z_{L1}}$ (Sentido de A para gnd)
- $i_2 = \frac{V_A - V_1}{R_3}$ (Sentido de A para gnd)

Equações Finais:

- Nó A: $\frac{V_A}{R_2 + Z_{C1}} + \frac{V_A}{R_1 + Z_{L1}} + \frac{V_A - V_1}{R_3} = 0$

Figura 41 Solução U=RI SOLVE do circuito nível básico em Corrente Alternada

Subindo o nível de complexidade, o circuito da Figura 42 inclui um maior número de Nós e uma Fonte isolada de Tensão, o que faz com que a aplicação produza resultados ligeiramente diferentes, tal como revela a Figura 43.

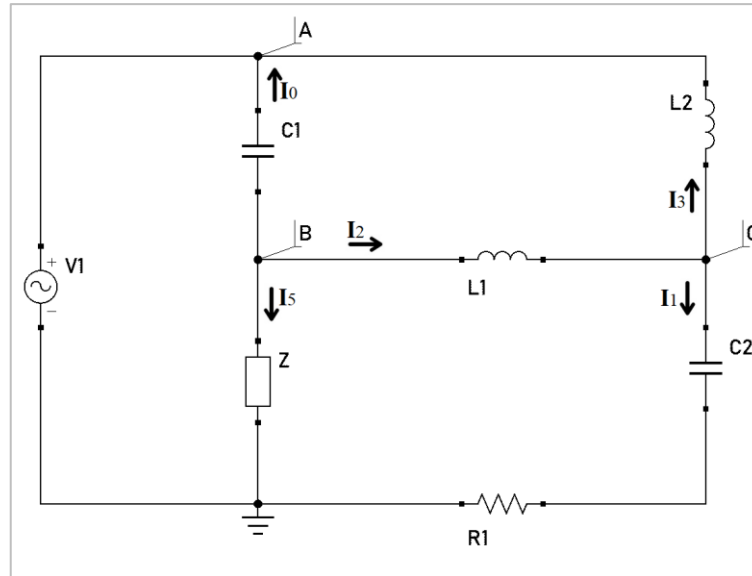


Figura 42 Circuito de nível intermédio em Corrente Alternada

Equações dos nós:

- Nó B: $i_0 + i_2 + i_5 = 0$
- Nó A: $i_4 = i_0 + i_3$
- Nó C: $i_1 + i_3 = i_2$

- $i_3 = \frac{V_C - V_A}{Z_{L2}}$ (Sentido de C para A)
- $i_5 = \frac{V_B}{Z}$ (Sentido de B para gnd)

Equações das correntes:

- $i_0 = \frac{V_B - V_A}{Z_{C1}}$ (Sentido de B para A)
- $i_1 = \frac{V_C}{Z_{C2} + R_1}$ (Sentido de C para gnd)
- $i_2 = \frac{V_B - V_C}{Z_{L1}}$ (Sentido de B para C)

Equações Finais:

- Nó B: $\frac{V_B - V_A}{Z_{C1}} + \frac{V_B - V_C}{Z_{L1}} + \frac{V_B}{Z} = 0$
- Nó A: $V_A = 60 \angle 0^\circ \text{ V}$
- Nó C: $\frac{V_C}{Z_{C2} + R_1} + \frac{V_C - V_A}{Z_{L2}} = \frac{V_B - V_C}{Z_{L1}}$

Figura 43 Solução U=RISOLVE do circuito nível intermédio em Corrente Alternada

Para finalizar a demonstração de resultados desta aplicação, o último circuito em Corrente Alternada é representado através da Figura 44, e a solução encontra-se evidenciada na Figura 45.

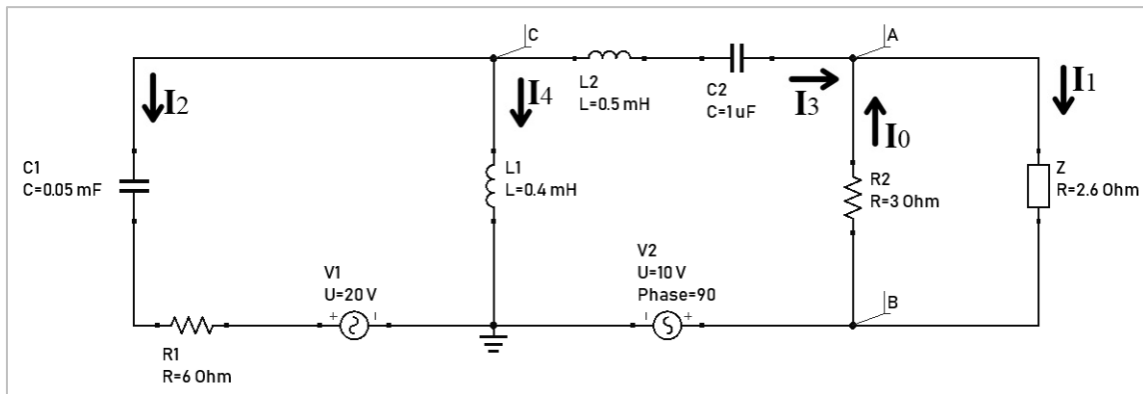


Figura 44 Circuito de nível avançado em Corrente Alternada

Equações dos nós:

- Nó B: $i_0 + i_5 = i_1$
- Nó C: $i_2 + i_3 + i_4 = 0$
- Nó A: $i_1 = i_0 + i_3$

Equações das correntes:

- $i_0 = \frac{V_B - V_A}{R_2}$ (Sentido de B para A)
- $i_1 = \frac{V_A - V_B}{Z}$ (Sentido de A para B)
- $i_2 = \frac{V_C - V_1}{Z_{C1} + R_1}$ (Sentido de C para gnd)

Equações Finais:

- Nó B: $V_B = 10 \angle 90^\circ \text{ V}$
- Nó C: $\frac{V_C - V_1}{Z_{C1} + R_1} + \frac{V_C - V_A}{Z_{L2} + Z_{C2}} + \frac{V_C}{Z_{L1}} = 0$
- Nó A: $\frac{V_A - V_B}{Z} = \frac{V_B - V_A}{R_2} + \frac{V_C - V_A}{Z_{L2} + Z_{C2}}$

Figura 45 Solução U=RISOLVE do circuito nível avançado em Corrente Alternada

A aplicação U=RISOLVE será disponibilizada com este conjunto de casos-exemplo, de modo a que o utilizador, numa fase inicial, usufrua desta ferramenta com maior facilidade.

Embora tenham sido estes os circuitos elegidos para demonstrar a aplicação e servir de apoio aos alunos, o objetivo será continuar a acrescentar novos exemplos com diferentes características, de modo a valorizar este *software* e atribuir-lhe uma maior versatilidade.

6. CONCLUSÕES

Ao longo da execução deste projeto, e dada a sua complexidade, tornou-se imperativo delinear uma planificação adequada das tarefas a realizar, de forma a obter resultados viáveis, em tempo útil e, evidentemente, uma aplicação bem estruturada.

Inicialmente, foi necessário estudar os diversos simuladores existentes e a forma como estes interagem com os seus utilizadores, de modo a averiguar a exequibilidade da aplicação, e se esta seria adequada dado o estado atual da simulação de circuitos elétricos. Comprovou-se que nenhuma das ferramentas atuais de simulação de circuitos elétricos possui a capacidade de oferecer resultados que visem uma componente didática, sejam equações ou metodologias de análise. Esta aferição conduziu à etapa seguinte na elaboração deste projeto, que consistiu na escolha do simulador que serviria de base para a aplicação, isto é, um simulador para esboçar os circuitos.

O QUCS foi, naturalmente, o *software* de eleição, principalmente pelo facto de ser utilizado por centenas de estudante de Engenharia Eletrotécnica no ISEP e, portanto, a sua inserção neste projeto é uma mais valia para a progressão da qualidade de ensino.

No que diz respeito à componente computacional, o primeiro passo consistiu em constatar de que forma é que o QUCS poderia produzir um *output* que fosse passível de ser tratado de forma a analisar univocamente cada circuito. Obtida esta informação, que se trata da

netlist, foi possível implementar todo o processo de análise da aplicação U=RISOLVE, desde a aquisição da *netlist*, até à geração dos resultados.

A última fase deste trabalho incluiu uma fase de teste e avaliação à qual a aplicação foi submetida, realizada por alunos de Engenharia Eletrotécnica do ISEP, cujo *feedback* foi crucial para aperfeiçoar o *software* e torna-lo cada vez mais robusto e preparado para um maior número de possibilidades de circuitos elétricos.

De uma forma geral, este projeto culminou numa aplicação que se considera bem sucedida. Foi, de facto, introduzida uma componente pedagógica à metodologia tradicional de simulação de circuitos elétricos, não só do ponto de vista do aluno, que poderá usufruir da ferramenta como apoio e complemento ao estudo, mas também do ponto de vista do professor, que poderá utilizar a aplicação para formular exercícios.

O modo como a aplicação foi implementada teve uma lógica cujo objetivo é bastante claro: obter uma ferramenta com características que a permitam sujeitar a uma constante evolução, para progressivamente ampliar a sua utilização da comunidade em que se insere. Existem, assim, várias funcionalidades a serem introduzidas na aplicação U=RISOLVE no futuro, tais como: criar uma interface gráfica que facilite a visualização de cada circuito juntamente com as respetivas Correntes; expandir a análise a vários simuladores, visto que a grande maioria é capaz de gerar *netlists*, que apenas variam ligeiramente o seu formato; fornecer resultados para os restantes métodos de análise de circuitos elétricos.

Esta aplicação insere-se num projeto de contribuições aos simuladores de circuitos elétricos, nomeadamente ao QUCS, que tem vindo a ser desenvolvido por vários alunos da Licenciatura em Engenharia Eletrotécnica e Computadores [23][24][25], com o objetivo de introduzir novos módulos e funcionalidades internas e externas ao QUCS no âmbito de melhorar a qualidade de ensino e autoaprendizagem na EE. Assim, pretende-se futuramente elaborar um artigo científico que englobará todas estas contribuições, para que, posteriormente, possa ser submetido a uma revista internacional de relevo na área Educação na Engenharia Eletrotécnica.

O que realmente tornou este projeto em algo absolutamente gratificante foi, não só o quão desafiante este se tornou, mas principalmente o sentimento revelador de que algo de útil acabou de ser criado. O *feedback* positivo de vários alunos que testaram a aplicação permitiu comprovar que o projeto teve qualidade, o que, tendo em conta que os estudantes

são o principal foco do mesmo, nomeadamente a melhoria da sua qualidade de ensino, é determinativo na avaliação do trabalho desenvolvido.

Resta concluir que este projeto permitiu desenvolver importantes capacidades, não só de carácter técnico, de programação *web* ou de análise de circuitos e simuladores, mas também de cariz pessoal, no que diz respeito à organização e planeamento do trabalho que a implementação de uma aplicação deste calibre, requer.

Referências Documentais

- [1] WILSON, Peter; ROSS, Neil; BROWN, Andrew – *Simulation of Magnetic Component Models in Electric Circuits including Dynamic Thermal Effects*. IEEE Transactions on Power Electronics, vol. 17, 2002.
- [2] NAGEL, Lawrence – *Life of Spice*, BCTM'96, 1996, <http://www.designers-guide.org/Perspective/life-of-spice.pdf/>
- [3] *The Chip That Jack Built*, Texas Instruments, c. 2008, (HTML), <http://www.ti.com/>
- [4] ROBERTS, Gordon; SEDRA, Adel – *SPICE for MicroElectronic Circuits*, 3ª Edição. Fort Worth: Saunders College Publishing, pp. 1-32 c. 1992. ISBN 0-03-052617-5.
- [5] CIRCUIT LAB – Circuit Lab, <http://www.circuitlab.com/>
- [6] GITHUB – KTechLab, <http://www.github.com/ktechlab/>
- [7] CADENCE – PSpice, <http://www.pspice.com/>
- [8] ANALOG DEVICES – LTspice, <http://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html/>
- [9] POWERSIM – PSIM, <https://powersimtech.com/products/psim/>
- [10] PAUL FALSTAD – Falstad Circuit Simulator, <https://www.falstad.com/circuit/>
- [11] MUSEMAZE – EveryCircuit, <http://everycircuit.com/>
- [12] MENTOR – System Vision Cloud, <https://www.systemvision.com/>
- [13] SOURCEFORGE – QUCS, <http://qucs.sourceforge.net/>
- [14] VIANA, Ana; PEREIRA, Francisco; ALVES, Mário – FEELE, Métodos de Análise e Simplificação de Circuitos Elétricos (Parte 2), slides 18-30, 2015.
- [15] SVOBODA James, *Supernode*, <http://people.clarkson.edu/~jsvoboda/Syllabi/ES250/MandNeqns/Supernode.pdf/>
- [16] HO, Chung-Wen; RUEHLI, Albert; P. BRENNAN, Pierce – *The Modified Nodal Approach to Network Analysis*, IEEE Transactions on Circuits and Systems, vol. 22, no. 6, pp. 504-509, 1975.
- [17] ROHRER, Ronald; NAGEL, Lawrence; MEYER, Robert; WEBER, Timothy – *CANCER: Computer Analysis of Nonlinear Circuits Excluding Radiation*. IEEE Solid-State Circuits Magazine, vol. 3, no. 2, pp. 40-42, 2011.
- [18] CHATZARAKIS, George – *Nodal Analysis Optimization Based on the Use of Virtual Current Sources: A Powerful New Pedagogical Method*. IEEE Transactions on Education, vol. 52, no. 1, pp. 144-150, 2009.
- [19] SOURCEFORGE – QUCS, *DC Analysis, Modified Nodal Analysis*, <http://qucs.sourceforge.net/tech/node13.html/>

- [20] KEOWN, John – *PSpice and Circuit Analysis*, 2ª Edição, Prentice-Hall, Inc., c.1994, pp. 13-65. ISBN 0-02-363526-6.
- [21] SOURCEFORGE – QUCS, *A Tutorial, Qucs Simulation of SPICE Netlists*, <http://qucs.sourceforge.net/docs/tutorial/spicetoqucs.pdf/>
- [22] WORLD WIDE WEB CONSORTIUM – *File API, W3C Working Draft*, 2017, <https://www.w3.org/TR/file-upload/>
- [23] MACEDO, Pedro – *Análise Comparativa e Melhoramento de Simuladores de Circuitos Eléctricos*. Dissertação de Licenciatura em Engenharia Eletrotécnica e Computadores orientada pelo Eng.º Mário Alves e apresentada no Instituto Superior de Engenharia do Instituto Politécnico do Porto, em 2015.
- [24] MARTINS, Nelson – *Desenvolvimento de Módulos para o Qucs*. Dissertação de Licenciatura em Engenharia Eletrotécnica e Computadores orientada pelo Eng.º Mário Alves e apresentada no Instituto Superior de Engenharia do Instituto Politécnico do Porto, em 2016.
- [25] SILVA, Alberto – *Design, Implementation and Integration of new Functionalities in the QUCS Simulator*. Dissertação de Licenciatura em Engenharia Eletrotécnica e Computadores orientada pelos Eng.º Mário Alves e Francisco Pereira e apresentada no Instituto Superior de Engenharia do Instituto Politécnico do Porto, em 2017.

