

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS**  
**UNIDADE ACADÊMICA DE GRADUAÇÃO**  
**CURSO DE ENGENHARIA ELÉTRICA**

**THALES AUGUSTO GELINGER**

**SIMULADOR DE CIRCUITOS ONLINE COLABORATIVO**

**São Leopoldo**

**2022**

THALES AUGUSTO GELINGER

**SIMULADOR DE CIRCUITOS ONLINE COLABORATIVO**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica, pelo Curso de Engenharia Elétrica da Universidade do Vale do Rio dos Sinos (UNISINOS).

Orientador: Prof. Dr. João Olegario de Oliveira de Souza

São Leopoldo

2022

## **AGRADECIMENTOS**

Quero expressar meus agradecimentos:

Primeiramente a Deus, sem ele nada disso seria possível.

Ao professor João Olegario de Oliveira de Souza por ter apoiado a ideia e sempre ajudado a melhorar.

A minha noiva Letícia, por estar comigo durante toda essa jornada e ter me apoiado sempre.

Aos meus pais por sempre terem dado suporte e compreensão.

Aos meus amigos e colegas em especial o Henrique e o Michael que me apoiaram a levar a ideia para frente e chegar no resultado e deram muitas ideias durante o desenvolvimento do projeto.

E a todos os demais que fizeram parte do meu caminho até aqui.

## RESUMO

O uso de ambiente digital para simular fenômenos reais já vem sendo utilizado com muita frequência para os mais diversos propósitos, dentre esses para eletrônica, especificadamente circuitos elétricos. A pandemia foi um fator que fez com que aumentasse a demanda por bons ambientes virtuais que possam suprir esse tipo de demanda, mas que também contribuam para um desenvolvimento colaborativo. A proposta deste trabalho foi desenvolver um simulador de circuitos online, que processe a simulação de forma performática, sem pesar na demanda de dados, e que atualize em tempo real toda a modificação feita no circuito para todos os usuários conectados. Para isso, foi utilizado a biblioteca React para a criação de interfaces, a ferramenta Ngspice para a simulação SPICE junto com Webassembly para permitir a execução dela pela web, e o *realtime database* do Firebase para permitir o compartilhamento em tempo real e o salvamento automático do circuito. Foram feitas algumas simulações e testes com usuários, e o simulador apresentou valores condizentes com a realidade e com simuladores já estabelecidos no mercado como o LTSpice, se mostrando bastante promissor para o desenvolvimento de circuitos junto com outras pessoas.

**Palavras-chave:** Simulador de circuitos, Colaboração, Virtual.

## LISTA DE FIGURAS

Figura 1 - Diagrama esquemático circuito de exemplo.....	14
Figura 2 - Interface do Multisim.....	15
Figura 3 - Interface do Proteus.....	16
Figura 4 - Interface do LTSpice.....	18
Figura 5 - Interface do QUCS.....	19
Figura 6 - Interface do Tinkercad .....	20
Figura 7 - Exemplo de sintaxe html.....	22
Figura 8 – Anatomia do CSS.....	22
Figura 9 - Hello World em javascript. ....	23
Figura 10 - Ilustração das principais tecnologias web.....	24
Figura 11 - Exemplo de código ReactJS.....	26
Figura 12 - Arquitetura de interpretação do webassembly.....	27
Figura 13 - Exemplo de estrutura JSON. ....	28
Figura 14 - Arquitetura geral do sistema. ....	34
Figura 15 - Interface do figma com o design do sistema. ....	35
Figura 16 - Circuito em esquema e em JSON. ....	36
Figura 17 - Comunicação em tempo real.....	37
Figura 18 - Entrada e saída NGSPICE. ....	38
Figura 19 – Login do simulador.....	40
Figura 20 - Dashboard vazio. ....	41
Figura 21 - Área de trabalho vazia. ....	42
Figura 22 - Área de trabalho com componentes.....	43
Figura 23 – Caixa de edição de componente.....	43
Figura 24 – Ação de rotacionar componente .....	44
Figura 25 – Ação de conectar .....	45
Figura 26 – Componente sem conexão .....	46
Figura 27 – Componente com conexão .....	46
Figura 28 - Área de trabalho com componentes conectado. ....	47
Figura 29 – Ação de remover.....	47
Figura 30 – Barra de componentes antes e de ferramentas depois .....	48
Figura 31 – Ícone de simular antes e durante a simulação.....	49
Figura 32 - Área de trabalho com voltímetro conectado ao circuito.....	49

Figura 33 – Voltímetro apresentando medição .....	50
Figura 34 – Conexão de circuito com osciloscópio .....	51
Figura 35 – Valor no osciloscópio ao iniciar simulação.....	51
Figura 36 - Área de trabalho com osciloscópio conectado. ....	52
Figura 37 – Ícone de compartilhar.....	52
Figura 38 – Mensagem ao clicar ícone de compartilhar .....	53
Figura 39 – Tarja de quem roda a simulação.....	53
Figura 40 – Tarja de quem está no mesmo circuito, mas não roda a simulação.....	54
Figura 41 – Visão de quem roda a simulação.....	54
Figura 42 - Visão de quem está no mesmo circuito, mas não roda a simulação.....	55
Figura 43 – Ícone para imprimir circuito .....	55
Figura 44 – Menu para imprimir circuito.....	56
Figura 45 - Dashboard com circuito adicionado.....	57
Figura 46 – Circuito CC com duas fontes de tensão LTSpice .....	59
Figura 47 - Circuito CC com duas fontes de tensão Colaborativo .....	60
Figura 48 - Circuito com tempo de carga no capacitor LTSpice .....	61
Figura 49 - Circuito com tempo de carga no capacitor Colaborativo .....	61
Figura 50 – Circuito com onda retificada LTSpice .....	62
Figura 51 - Circuito com onda retificada colaborativo. ....	62
Figura 52 – Avaliação dos usuários sobre a usabilidade do simulador apresentado	63
Figura 53 – Avaliação dos usuários sobre a função de colaboração.....	64

## LISTA DE QUADROS

Quadro 1 – Tecnologias utilizadas .....	31
Quadro 2 – Componentes utilizados .....	32
Quadro 3 – Ferramentas de medição utilizados .....	33
Quadro 4 – Comparação de valores medidos entre o LTSPice e o Colaborativo.....	59

## **LISTA DE TABELAS**

Tabela 1 - Comparação de funcionalidades dos simuladores.	21
---	----



## LISTA DE SIGLAS

- BAAS *Backend as a servisse* (Processo interno como serviço)
- CC Corrente Continua
- CSS *Cascading Style Sheets* (Folhas de estilo em cascata)
- HTML *HyperText Markup Language* (Linguagem de marcação de hipertexto)
- JSON *Javascript object notation* (Notação de objeto Javascript)
- QUCS *Quite Universal Circuit Simulator* (Simulador de circuito muito universal)
- SPICE *Simulation Program with Integrated Circuit Emphasis* (Programa de simulação com Ênfase em Circuito Integrado)

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>11</b>
1.1 OBJETIVOS .....	12
1.1.1 Objetivo geral .....	12
1.1.2 Objetivos específicos .....	12
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>13</b>
2.1 SIMULADORES DE CIRCUITOS ELÉTRICOS .....	13
2.1.1 SPICE.....	13
2.1.2 SIMULADORES COMERCIAIS .....	15
2.1.3 SIMULADORES GRATUITOS.....	17
2.1.4 SIMULADORES DE CÓDIGO LIVRE.....	18
2.1.5 SIMULADORES ONLINE .....	20
2.2 TECNOLOGIAS WEB .....	21
2.2.1 HTML .....	21
2.2.2 CSS .....	22
2.2.3 JAVASCRIPT.....	23
2.2.4 REACTJS.....	25
2.2.5 WEBASSEMBLY .....	26
2.2.6 FIREBASE .....	27
<b>3 METODOLOGIA</b> .....	<b>30</b>
3.1 FERRAMENTAS .....	30
3.2 INSTRUMENTOS E SIMBOLOGIAS .....	31
3.3 ARQUITETURA DO <i>SOFTWARE</i> .....	33
3.3.1 CLIENTE .....	34
3.3.2 SERVIDOR.....	35
3.3.3 CÁLCULOS DA SIMULAÇÃO .....	37
<b>4 ANÁLISE DOS RESULTADOS</b> .....	<b>40</b>
4.1 FLUXO DE LOGIN .....	40
4.2 FLUXO DE CIRCUITO .....	41
4.2.1 Inserir componentes na área de trabalho .....	42
4.2.2 Conectar componentes .....	45
4.2.3 Remover Componentes e Conexões.....	47
4.2.4 Simulação .....	48

4.2.4.1 Osciloscópio .....	50
4.2.4.2 Colaboração .....	52
4.2.4.3 Imprimir e Voltar .....	55
4.3 COMPARAÇÃO COM OUTROS SISTEMAS .....	57
<b>4.3.1 Comparação de simulações com o LTSPice .....</b>	<b>59</b>
4.3.1.1 Simulação de circuito CC com duas fontes de tensão.....	59
4.3.1.2 Simulação de circuito medindo tempo de carga no capacitor.....	60
4.4 AVALIAÇÃO DE USUÁRIOS .....	63
<b>4.4.1 Análise da usabilidade.....</b>	<b>63</b>
<b>4.4.2 Análise do compartilhamento em tempo real.....</b>	<b>64</b>
<b>4.4.3 Análise da simulação.....</b>	<b>65</b>
<b>5 CONCLUSÃO .....</b>	<b>66</b>
5.1 TRABALHOS FUTUROS .....	66
<b>REFERÊNCIAS.....</b>	<b>68</b>

## 1 INTRODUÇÃO

Desde o momento que a computação se tornou mais presente na vida de todos, é cada vez mais notável o uso da mesma para simulador como seriam situações da vida real em um ambiente virtual, dentre elas simulações de circuitos, de peças que envolvem um hardware, para identificar o comportamento disso antes de aplicar na prática de fato, a fim de evitar possíveis acidentes em uma montagem errada, e poder realizar vários testes com componentes diferentes antes de efetivamente comprar e realizar a montagem do circuito.

Com a situação que a pandemia trouxe onde vivemos em mundo muito mais digital e a tecnologia para a colaboração se torna mais necessária nas vidas das pessoas, nota-se uma necessidade em algo que abrange uma necessidade nos estudantes de hardware, entusiastas ou times de desenvolvimento, que precisam desenvolver algum circuito em conjunto com outras pessoas, e que seja de fácil acesso.

Além disso, pensando primariamente em estudantes, por mais que existam uma infinidade de boas ferramentas para a simulações, as melhores acabam por ser pagas e muitas vezes até complexas de serem operadas levando a uma curva de aprendizado muito grande para o que as vezes é apenas uma operação pontual sem uma total necessidade de uma variedade complexa de ferramentas disponíveis.

Muitos estudantes das áreas de engenharia elétrica, eletrônica e afins, ou até mesmo entusiastas do mundo do desenvolvimento de hardware, precisam fazer uso de simuladores de circuitos pois não há a gama de componentes disponíveis em um laboratório em casa, entretanto, é sempre complicado de encontrar um bom simulador, simuladores grátis muitas vezes não possuem muitos recursos, e não possuem boa usabilidade, e muitas vezes também é necessário de outros simuladores para diferentes fins, sejam circuitos simples ou mais complexos que envolvam microcontroladores, e simuladores estes que são comerciais, e muitos acabam optando pelo uso ilegal do mesmo simplesmente pelo fato de não haver opção que supra a necessidade do indivíduo.

A escolha do desenvolvimento desta pesquisa e protótipo é válida, pois se vê a necessidade de uma ferramenta que faça com que a experiência de trabalhos de circuitos em grupos seja melhor, e que não haja necessidade de aprender múltiplos simuladores para múltiplas situações, e que o mesmo utilizado em um sistema

operacional, possa ser utilizado em outro usufruindo de todo o fluxo de trabalho adquirido anteriormente.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo geral

Desenvolver uma ferramenta de simulação que seja gratuita, de código livre, multiplataforma, e que supra simulações com circuitos de componentes básicos, e permita um ambiente colaborativo com atualização do circuito em tempo real entre vários colaboradores.

### 1.1.2 Objetivos específicos

Os objetivos específicos foram elencados a partir da subdivisão do objetivo geral em atividades específicas e com finalidade única. Pode-se citar como objetivos específicos.

- a) Pesquisar formas de simulação de circuitos;
- b) Identificar *softwares* de simulação de circuitos mais utilizados;
- c) Identificar tecnologias multiplataforma;
- d) Identificar meios de efetuar simulações com alta performance;
- e) Desenvolver um protótipo que faça simulações e as compartilhe em tempo real;
- f) Validar o uso do sistema através de um grupo de voluntários.

O trabalho é dividido nos seguintes capítulos: o capítulo 2 é o capítulo de fundamentação teórica, onde encontra-se todas as tecnologias e sistemas existentes de inspiração para este; o capítulo 3 apresenta o desenvolvimento do projeto em si, e como cada etapa é realizada; e por fim, o capítulo 4 demonstra a projeção do tempo para o desenvolvimento de cada etapa do protótipo.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo são apresentados os conteúdos para a execução do projeto, tanto com referências dos mais variados tipos de simuladores já existentes, pontuando suas funcionalidades que serviram de inspiração para este projeto, quanto sobre as tecnologias envolvidas no processo do desenvolvimento propriamente dito do mesmo.

### 2.1 SIMULADORES DE CIRCUITOS ELÉTRICOS

Simuladores de circuitos elétricos, são ferramentas utilizadas por vários tipos de pessoas, sejam elas estudantes, entusiastas, profissionais ou amadores, onde há um circuito que se deseja montar, porém se gostaria de analisar o funcionamento do mesmo e avaliar se ele se comporta como deve, e para isso, essas ferramentas se fazem necessárias para poder montar o circuito em um ambiente virtual controlado, onde se possa ter uma visualização do circuito, em regime estacionário ou também em funcionamento dinâmico, avaliando todas as possibilidades de simulação antes de uma montagem propriamente dita em um placa de prototipação ou diretamente na placa de circuitos.

Nos seguintes tópicos é possível obter um vislumbre de quais são os tipos de simuladores disponíveis no mercado e algumas características.

#### 2.1.1 SPICE

Simuladores SPICE (Programa de simulação com ênfase em circuitos integrados) foram lançados em 1973 e foram os programas de simulação que obtiveram maior notoriedade documentada, e a partir dos mesmos que muitos simuladores comerciais ou não do mercado passaram a ser desenvolvidos, construindo ao longo do tempo uma biblioteca sólida de componentes, onde até os próprios fornecedores disponibilizam componentes no formato de utilização SPICE.

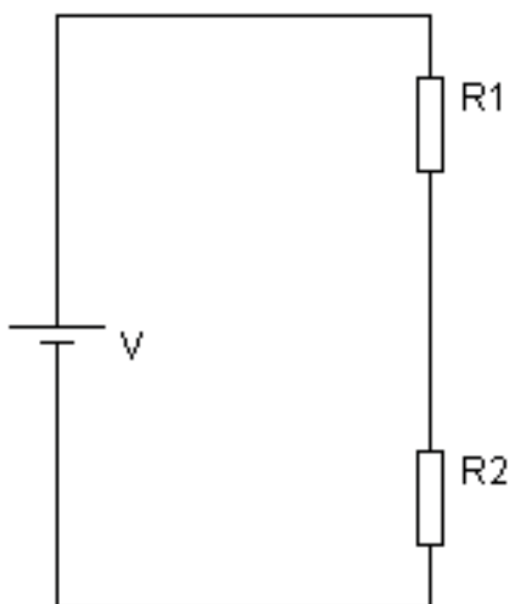
“SPICE é um programa de simulação e modelagem de computador usado por engenheiros para prever matematicamente o comportamento de circuitos eletrônicos.” (National Instruments, 2019, *online*).

Segundo a National Instruments (2019), o SPICE tem capacidade de simular uma grande variedade de componentes sejam eles básicos como resistores ou mais complexos como MOSFETs, e com o uso desses componentes como blocos para outros, se possibilitou fabricantes disponibilizassem seus próprios componentes em formato SPICE já nas especificações adequadas.

Simulações de circuito em SPICE precisam ser informadas como uma *netlist* (lista de conexões), entre os nós de cada componente, e essa lista é literalmente a representação de um circuito descrito na forma de texto (National Instruments, 2019, online).

Na Figura 1 é possível ver um exemplo de um circuito divisor de tensão e como é a representação do mesmo em uma *netlist*.

Figura 1 - Diagrama esquemático circuito de exemplo



Fonte: Rodrigo Cesar Vertulo (2022, online).

#### Quadro 1 - Exemplo de lista de conexões

\*Qualquer texto após o asterisco é ignorado por SPICE

*Divisor de tensão
V1 1 0 12
R1 1 2 1000
R2 2 0 2000
.OP * executa uma análise de ponto operacional DC

Fonte: National Instruments (2022, online).

De acordo com National Instruments (2019, online);

Embora os modelos SPICE usados em uma simulação SPICE possam afetar muito a precisão dos resultados, as configurações da simulação também contribuem para vários graus de precisão. As opções de simulação SPICE geralmente permitem que o usuário obtenha mais precisão nos resultados ao custo da velocidade da simulação.

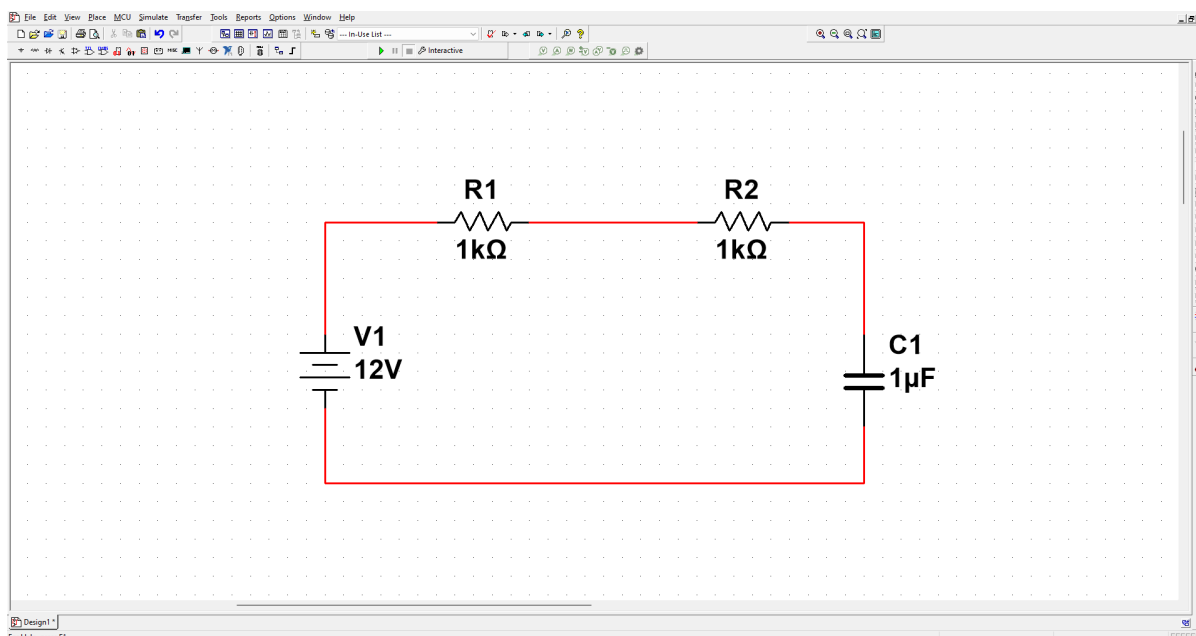
Apesar da precisão do uso da simulação SPICE, ela sozinha se torna um tanto quanto complexa de se utilizar principalmente em circuitos maiores e mais complexos, e o fato de precisar descrever a *netlist* para efetuar a simulação faz com que seja necessária uma grande curva de aprendizado para dominar as formas de montar uma *netlist* e efetuar a simulação propriamente dita, e tendo isto em vista, muitas ferramentas comerciais e não comerciais foram criadas com base no SPICE, mas com uma interface que possibilite que o usuário apenas desenhe o circuito e a própria ferramenta transcreve o desenho para uma *netlist* para a simulação.

### 2.1.2 SIMULADORES COMERCIAIS

Simuladores de circuitos comerciais são aqueles disponibilizados pelas empresas como produtos pagos, são produtos com boa qualidade de simulação e muitos são bem consolidados no mercado e conhecidos por quem busca este tipo de *software*, como exemplo temos os mais famosos, como o Multisim e o Proteus, onde ambos utilizam da plataforma SPICE para realizar as simulações. Na Figura 2 está a interface de usuário do Multisim.

Figura 2 - Interface do Multisim





Fonte: O Autor (2022).

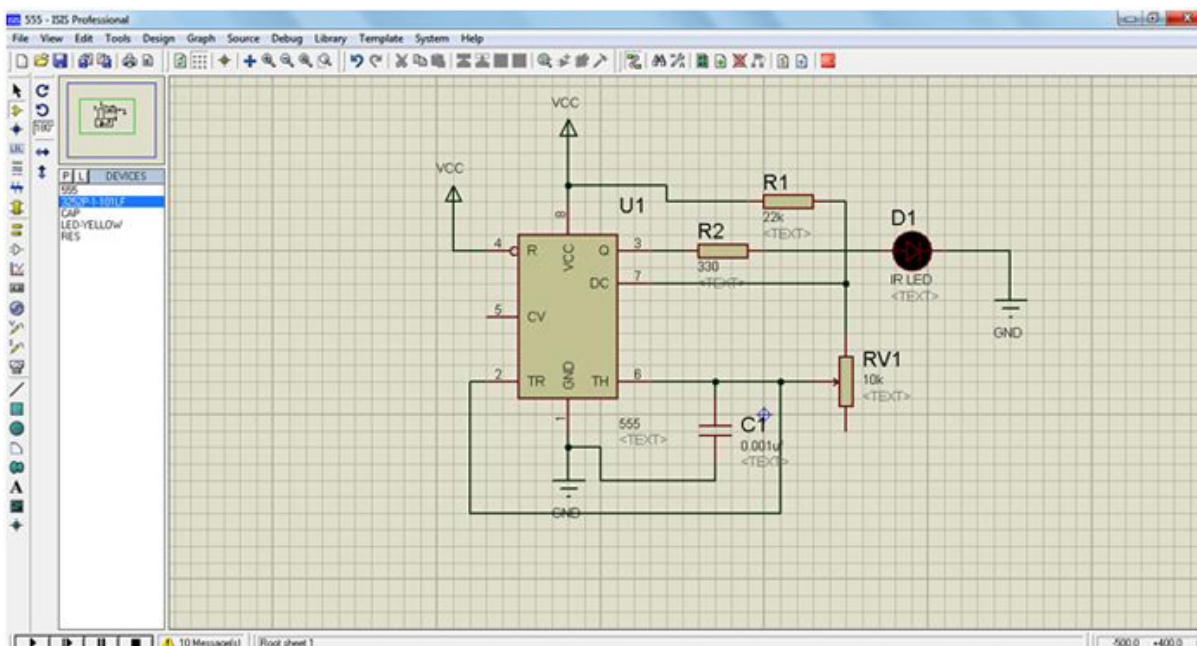
De acordo com National Instruments (2019, online);

O Multisim integra a simulação SPICE padrão da indústria com um ambiente esquemático interativo para visualizar e analisar instantaneamente o comportamento de circuitos eletrônicos. Sua interface intuitiva ajuda os educadores a reforçar a teoria dos circuitos e melhorar a compreensão e retenção dos conceitos teóricos em todo o currículo de engenharia. Ao acrescentar a poderosa simulação e análise de circuitos ao fluxo do projeto, o Multisim ajuda pesquisadores e projetistas a reduzir as iterações dos protótipos de placas de circuito impresso (PCB) e economizar seus custos de desenvolvimento.

O Multisim é uma ferramenta de simulação que precisa ser instalada na máquina do usuário, e possui uma série de diferenciais que agregam ainda mais no SPICE, por se tratar de uma interface mais amigável, é possível selecionar entre uma variada lista de componentes, fontes, instrumentos de medição, executar simulações em tempo real (Instituto newton c. braga, 2021, *online*).

Outro simulador que tem destaque é o Proteus demonstrado com sua interface na Figura 3, porém além de possibilitar a simulação de circuitos também em cima da plataforma SPICE, ele também é capaz de simular *firmwares* e *softwares* interagindo com o circuito montado, dando mais possibilidades para o processo de prototipagem de um equipamento.

Figura 3 - Interface do Proteus



Fonte: Vinod Bandlamudi (2022, online).

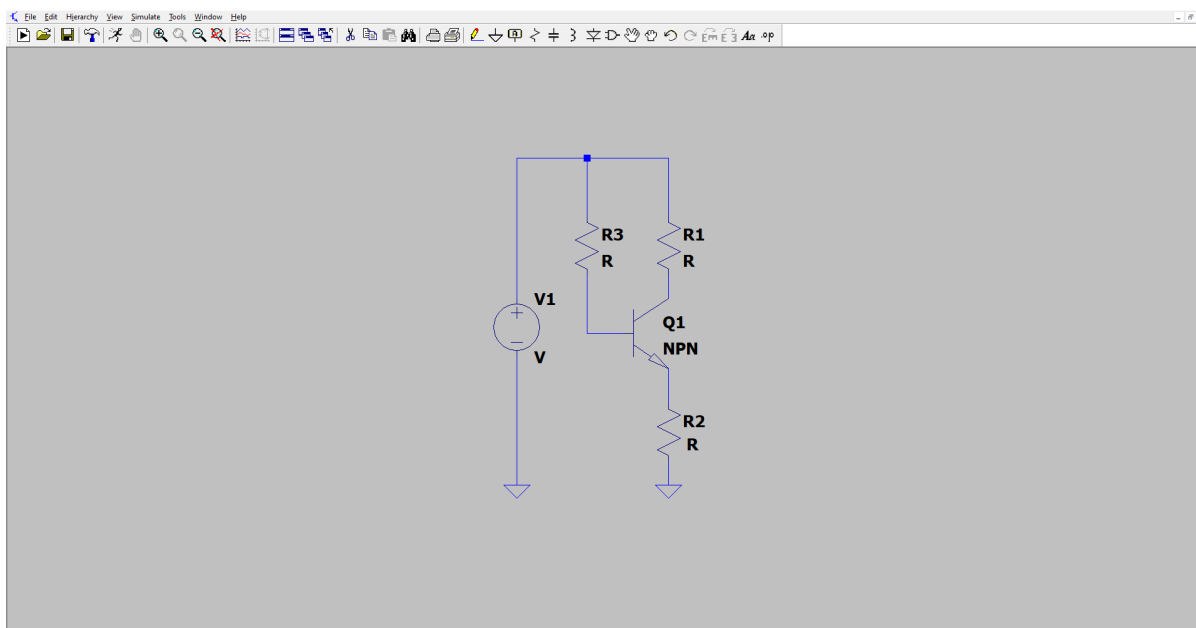
Segundo Labcenter (2021), o *Proteus Virtual System Modeling* (VSM) combina simulação SPICE de modo misto com simulação de microcontrolador rápido líder mundial. Ele permite a prototipagem rápida de designs de *hardware* e *firmware*, em *software*.

Assim como o Multisim, o Proteus também carrega consigo muitas funcionalidades diferentes, dentre elas se destacam a capacidade do uso de microcontroladores dos mais variados tipos, permitindo uma gama ainda maior de possibilidades de circuitos simulados, além de trazer junto uma interface com animações nos circuitos, demonstrando por exemplo a rotação de um motor.

### 2.1.3 SIMULADORES GRATUITOS

Dentre os simuladores gratuitos, existem várias ferramentas disponíveis, e são aqueles normalmente mais simples mas que cumprem bem o seu papel se tratando de um simulador para algo específico, dentre os exemplos mais famosos existe o consagrado LTSpice demonstrado na Figura 4, que é baseado na plataforma de simulação SPICE, o maior problema dentre os simuladores gratuitos está nas atualizações, onde possuem um código fonte fechado que não permita que a comunidade de sugestões de melhorias e atualizações para o sistema.

Figura 4 - Interface do LTSpice



Fonte: O Autor (2022).

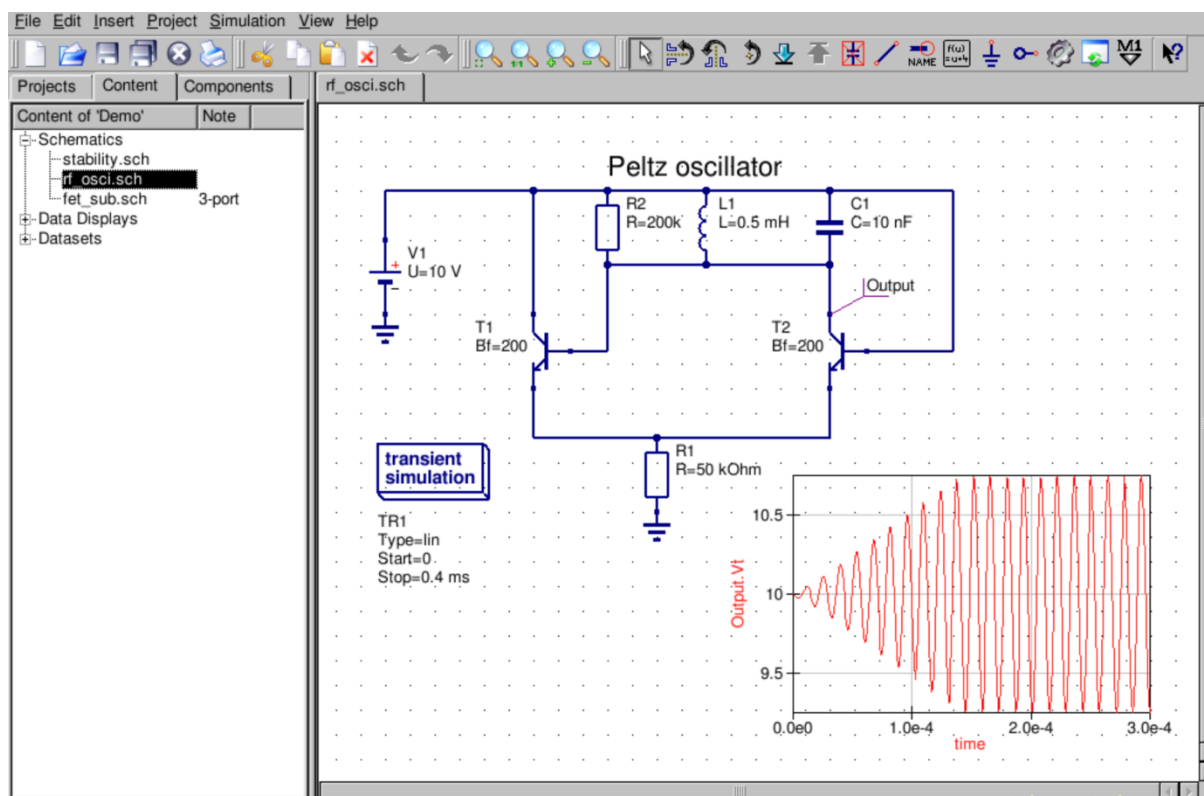
O LTSpice é o *software* gratuito mais conhecido dentre todos que necessitam deste tipo de ferramenta pois é gratuito, e muito poderoso por usar também da simulação SPICE, é um *software* de simulação de alto desempenho, captura esquemática e visualizador de forma de onda com aprimoramentos e modelos para facilitar a simulação de circuitos analógicos, entretanto, por mais que ele possui muitos componentes e funções diferentes, sua interface é um pouco complexa de utilizar, e raramente possui atualizações, fazendo com que sua usabilidade comprometa a experiência do uso do simulador (Analog Devices, 2021, *online*).

#### 2.1.4 SIMULADORES DE CÓDIGO LIVRE

Seguindo os simuladores gratuitos, existe uma derivação desta modalidade que são os simuladores de código aberto, ou seja, ele possui o diferencial direto dos simuladores simplesmente gratuitos, e permite uma maior liberdade da comunidade de *hardware*. Dentre estes são poucas as iniciativas que de fato chegaram a ser comentadas na comunidade, o maior exemplo é o QUCS, que utiliza também de simuladores SPICE, e é uma interface para sua utilização, que por mais que funcione,

é desenvolvido com uma tecnologia não tão comum para o desenvolvimento de interfaces, afastando quem talvez se interesse em possuir uma boa ferramenta de simulação e ajudar a contribuir para a evolução dela, sua interface de usuário pode ser visualizada na Figura 5.

Figura 5 - Interface do QUCS



Fonte: Stefan Jahn (2022, online).

De acordo com a página oficial do QUCS (2017, online)

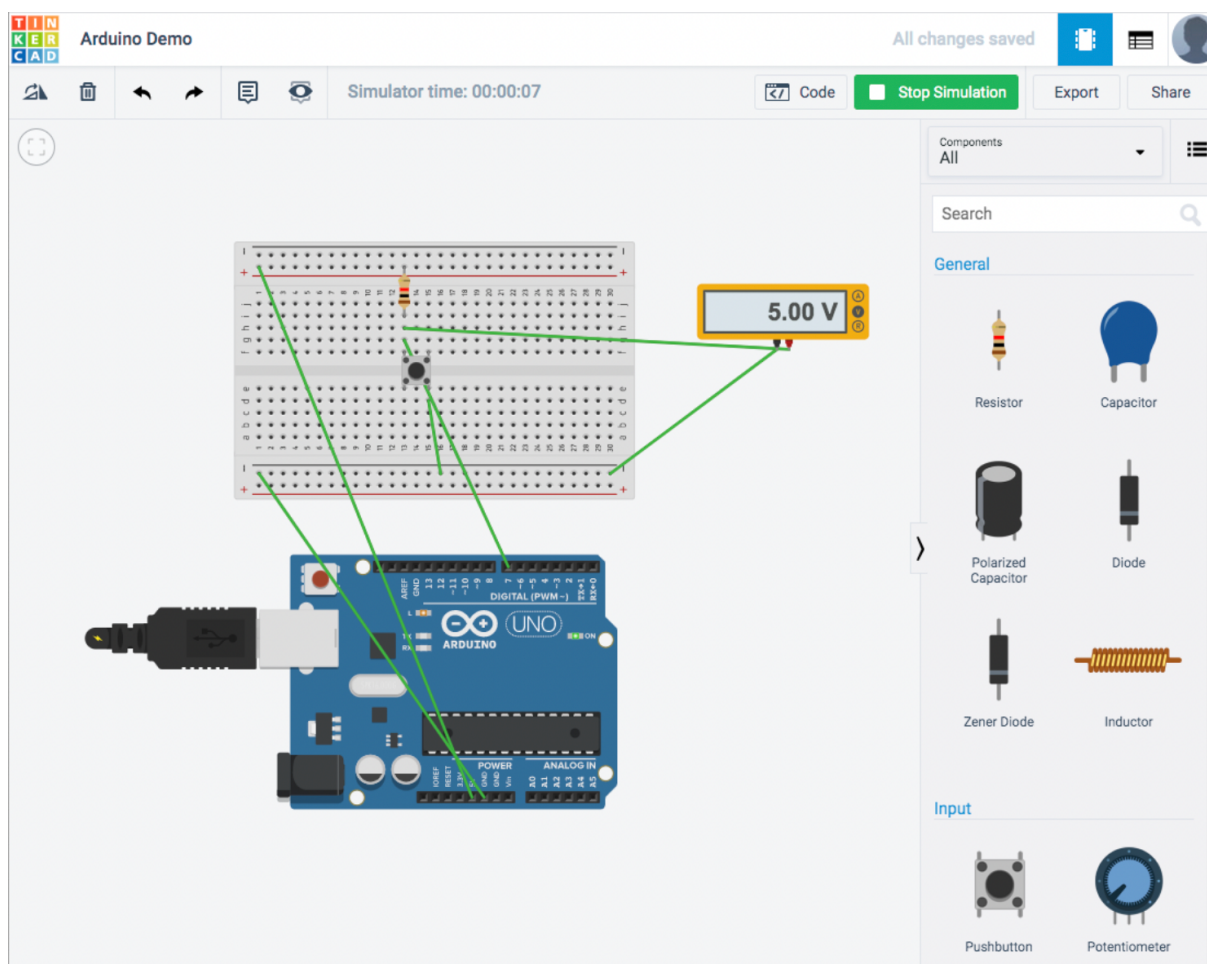
Qucs é um simulador de circuito integrado, o que significa que você é capaz de configurar um circuito com uma interface gráfica de usuário (GUI) e simular o comportamento de sinal grande, sinal fraco e ruído do circuito. Após o término da simulação, você pode visualizar os resultados da simulação em uma página ou janela de apresentação.

O QUCS foi uma iniciativa de simulador de código livre que não possui atualizações desde 2017, e por mais que possua um simulador SPICE rodando por trás, e tem uma interface, tanto a forma como é apresentada a interface e a forma como foi desenvolvida, a escolha da tecnologia, influenciou para a dificuldade deste projeto seguir em frente para uso.

### 2.1.5 SIMULADORES ONLINE

Simuladores online são aqueles que rodam dentro do navegador, onde não é necessário realizar o *download* e a instalação da ferramenta, evitando o gasto de memória da máquina do usuário e muitas vezes proporcionando um auto salvamento do trabalho realizado permitindo o acesso por qualquer local que possua internet. Dos maiores exemplos de simuladores online, existe o Tinkercad demonstrado na Figura 6.

Figura 6 - Interface do Tinkercad



Fonte: Adaptado de Tinkercad (2021, *online*).

Tinkercad é um aplicativo web gratuito que serve tanto para desenho em 3D quanto para simulação de circuitos e programação, e por ser uma ferramenta online de simulação, o Tinkercad tem o benefício de executar no navegador, ou seja, é possível simular os circuitos em qualquer sistema operacional. Entretanto, ele utiliza

de uma visualização de componentes em um formato mais parecido com a vida real, inclusive disponibilizando uma *protoboard* para simular os encaixes dos componentes, é um simulador que funciona porém limitado pela quantidade de componente e por não se tratar de um simulador SPICE faz com que eles tenham cada vez mais complicações para ser utilizado com circuitos mais complexos (Autodesk, 2021).

No Tabela 1, há uma comparação entre todos os simuladores citados até então, demarcando suas funcionalidades e diferenças, entre aqueles que possuem e marcando com um 'X', e aqueles que não possuem, deixando o campo vazio.

Tabela 1 - Comparação de funcionalidades dos simuladores

Funcionalidades	Tinkercad	Multisim	Proteus	LtSpice	Qucs	Simulador Deste trabalho
Spice	X	X	X	X	X	X
Online	X					X
Colaborativo						X
Gratuito	X			X	X	X
Open Source					X	X
Tecnologia com alta adesão						X

Fonte: O Autor (2022).

## 2.2 TECNOLOGIAS WEB

Para se alcançar as funcionalidades requeridas para o sistema, é necessário o uso de tecnologias que se comprovam consolidadas no mercado do desenvolvimento web, tal como sua popularidade, visando alcançar o maior número de pessoas para a contribuição no projeto *open source* no longo prazo.

### 2.2.1 HTML

O HTML é uma linguagem descritiva que especifica a estrutura de uma página web, onde um documento .html é um arquivo de texto com elementos estruturados com a abertura e fechamento de *tags*, cada *tag* começa e termina com colchetes angulares (<>), e existem *tags* vazias que não necessariamente possuem conteúdo entre as *tags* mas sim nas configuração dos atributos dela (MDN, 2021).

Na Figura 7 é demonstrado como é escrito um elemento em HTML.

Figura 7 - Exemplo de sintaxe html



Fonte: MDN (2021).

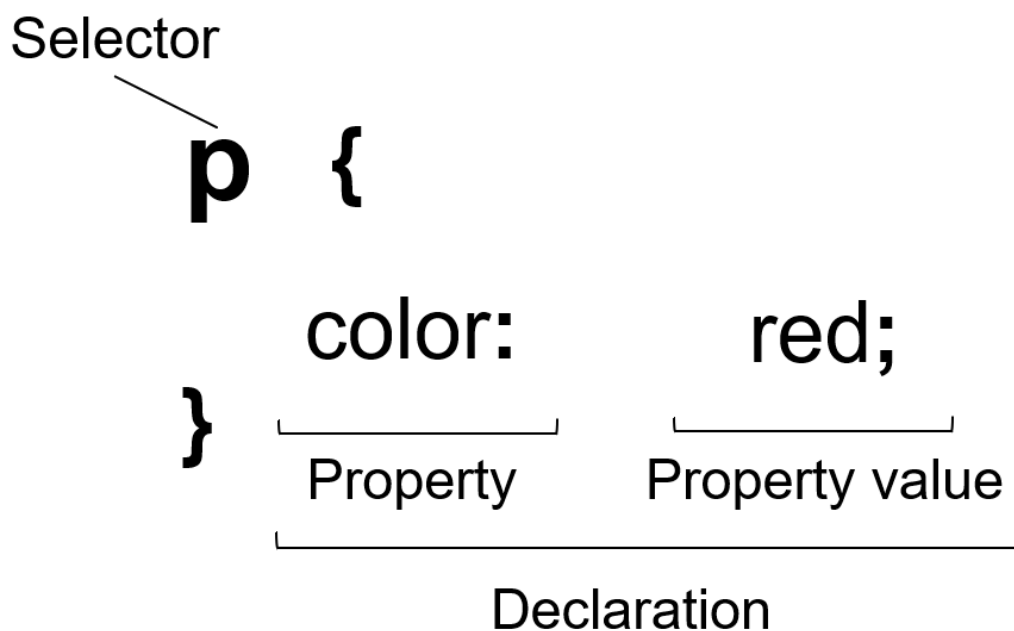
Os documentos HTML são salvos com a extensão .html e são executados pelo Navegador (MDN, 2021).

### 2.2.2 CSS

O CSS é uma linguagem de marcação que trabalha em conjunto com o HTML, e serve para dar estilos aos elementos estruturados na apresentação no navegador, onde ele aplica tais estilizações de acordo em como foram selecionados os elementos no arquivo com extensão .css conectado ao .html, e em um arquivo .css além dos seletores possui também as propriedades e os valores para que seja possível variar a estilização para cada elemento (MDN, 2021).

Na Figura 8, é demonstrado a estrutura de como estilizar um elemento p em HTML utilizando CSS.

Figura 8 – Anatomia do CSS



Fonte: Adaptado pelo autor de MDN (2022).

O CSS como no próprio nome da diz, *Cascading*, que significa cascata, está se referindo a como as regras de estilização se aplicam seguindo uma estrutura de hierarquia, onde um seletor possui maior valor que outro podendo sobrescrever a regra de um elemento utilizando de seletores mais específicos (MDN, 2021).

### 2.2.3 JAVASCRIPT

O javascript é uma linguagem de programação interpretada utilizada na web, que tem como responsabilidade dar mobilidade e funções para o HTML e o CSS, fazendo com que a página web não seja apenas um site estático, e sim uma aplicação funcional, é conhecida como uma linguagem de *script* da web, e possui diversos paradigmas de se trabalhar (MDN, 2021).

Pode-se ver na Figura 9 um exemplo de como ter uma saída na tela com um clássico “*Hello, World!*”, comumente utilizado quando se começa em uma nova linguagem de programação.

Figura 9 - *Hello World* em javascript





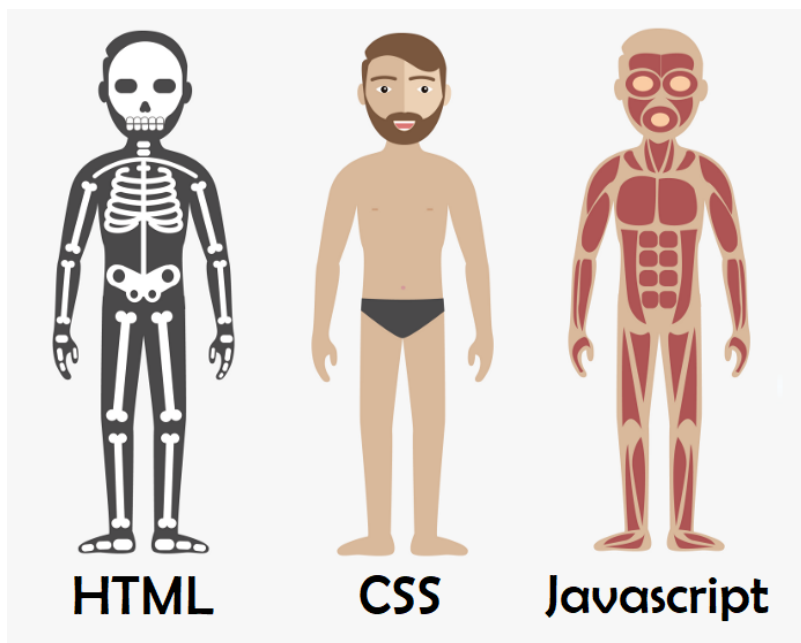
Fonte: O Autor (2022).

Entretanto, por mais que o Javascript seja reconhecido como uma linguagem web, com o passar do tempo, e visto as possibilidades da linguagem, criou-se a plataforma nodejs que fez com que fosse possível a utilização do Javascript no lado servidor, o que abriu as portas para a disseminação do Javascript entre todas as plataformas, seja na criação de sites mais complexos, de servidores, jogos e aplicativos (OpenJS Foundation, 2021, *online*).

Ou seja, todas as tecnologias web apresentadas até aqui servem cada uma para seu propósito específico, e pode ser exemplificado através da imagem na figura 8, em uma analogia direta, o HTML dá o esqueleto e uma página, o CSS dá a casca de visualização, e o Javascript dá vida a tudo isso permitindo a mobilidade e funcionalidade de todo esse conjunto (Balerini, 2021, *online*).

Na Figura 10, temos uma ilustração de qual seria o papel de cada uma dessas tecnologias dentro da web.

Figura 10 - Ilustração das principais tecnologias web



Fonte: Balerini (2021).

#### 2.2.4 REACTJS

ReactJS é uma biblioteca Javascript desenvolvida pelo Facebook para a criação de interfaces de usuário (UI), o objetivo da existência dela, é de facilitar o desenvolvimento de interfaces utilizando das tecnologias já conhecidas como o HTML CSS e o Javascript, porém de forma a possibilitar uma maior escalabilidade e facilidade na construção de telas mais complexas (Artemij Fedosekev, 2015).

Por padrão, a web com Javascript trabalha de forma imperativa, ou seja, com uma interação diretamente no documento de elementos do browser (DOM), com o React, é possível trabalhar de forma declarativa, tornando o código com uma legibilidade melhor e também melhorando sua depuração, o React também trabalha com componentização, ou seja, a possibilidade extrair pedaços do código e pedaços da interface em componentes específicos em arquivos separado, permitindo a importação em diversos locais, evitando a repetição de código e auxiliando na reutilização de código, e possui uma virtual DOM, o que traz mais performance dentro do navegador, pois possui uma gerencia de estado que apenas executa uma nova renderização em um pedaço específico da tela (Artemij Fedosekev, 2015).

Na Figura 11, temos o exemplo de como seria a construção de um componente para mostrar algo em tela dentro do ReactJS.

Figura 11 - Exemplo de código ReactJS

A screenshot of a code editor window with a light blue background. The editor has three colored window control buttons (red, yellow, green) in the top left corner. The code is written in a dark font with syntax highlighting. The code defines a functional component named 'Hello' and renders it with the name 'World' into a root element with the ID 'root'.

```
const Hello = ({ name }) => {  
  return <h1> Hello {name} </h1>;  
};  
  
ReactDOM.render(<Hello name="World" />, document.getElementById("root"));
```

Fonte: Adaptado pelo autor a partir do livro React.js Essentials (2021).

O React utiliza o Javascript em conjunto com o HTML, então por não usar modelos tal como outras bibliotecas fazem, isso permite uma maior integração entre os dados e o que é apresentado na tela, e por ter essa integração entre essas duas linguagens que são o Javascript e o HTML, é comum os arquivos que possuem código em React possuírem a extensão .jsx que significa Javascript XML (Artemij Fedosekev, 2015).

### 2.2.5 WEBASSEMBLY

O Webassembly permite com que aplicações web sejam executadas de forma como se fossem aplicativos nativos, o que implica em um aumento de performance em navegadores, abrindo a possibilidade dos mais variados tipos de aplicações (MDN, 2021).

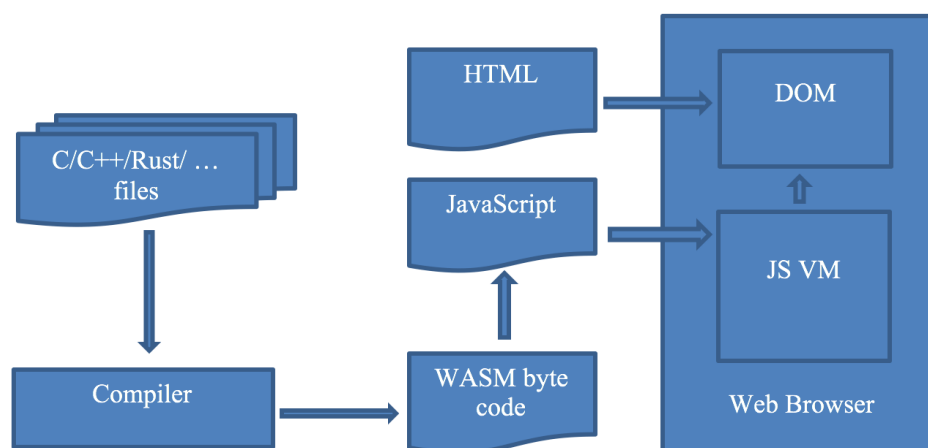
O WebAssembly é um novo tipo de código que pode ser executado em browsers modernos — se trata de uma linguagem de baixo nível como assembly, com um formato binário compacto que executa com performance quase nativa e que fornece um novo alvo de compilação para linguagens como C/C++, para que possam ser executadas na web. Também

foi projetado para executar em conjunto com o JavaScript, permitindo que ambos trabalhem juntos. (MDN, 2021)

Como o Webassembly é um *bytecode* (código binário), é muito difícil um ser humano digitar propriamente dito neste formato de arquivo, e para isso se utiliza compiladores, que usam de linguagens de baixo nível, como C, C++, Rust e as compilam para um arquivo com extensão `.wasm` que posteriormente é importado para a utilização junto ao Javascript dentro da web, logo a baixo pode-se visualizar uma abstração desse funcionamento (Webassembly, 2021).

Na Figura 12, temos um diagrama representando como é o fluxo para incorporar um Webassembly dentro de um navegador.

Figura 12 - Arquitetura de interpretação do Webassembly



Fonte: David Maciejak (2017).

## 2.2.6 FIREBASE

O Firebase é uma plataforma para criar um servidor como um serviço, comumente conhecido como BAAS é um conjunto de ferramentas disponibilizadas pelo Google, que permite o desenvolvimento de vários tipos de aplicações, ele possui em si ferramentas para autenticação, bancos de dados relacionais, e não relacionais, métricas de aplicativos, gerenciamento de propagandas, e mais uma variedade quantidade de serviços, dentre eles um destaque para o *realtime database* (banco de dados em tempo real), que é um serviço de base de dados não relacional em formato

de documento, que é sincronizado em tempo real com todas as aplicações conectadas a ele, ou seja, com várias aplicações conectadas e apresentando a mesma base, a alteração que ocorre em uma automaticamente ocorre em outra, o que traz um maior dinamismo em aplicações de tempo real (GeekHunter, 2021).

Na Figura 13, há um exemplo de como é a aparência de uma estrutura de dados JSON.

Figura 13 - Exemplo de estrutura JSON



Fonte: Adaptado pelo autor de AWS Trabalhando com JSON (2021).

Um banco de dados não relacional em formato de documento, trabalha basicamente no formato de um JSON, permitindo todo o tipo de estrutura a ser salva dentro dele, sem necessitar de regras específicas e colunas específicas para cada tipo de dado, sendo assim um banco de dados com uma maior versatilidade em aplicações que não se tem certeza de qual será o formato de dados que irá seguir (AWS, 2021).

### 2.2.7 UX

O termo UX é uma abreviação para User Experience, que nada mais é do que um conjunto de técnicas desenvolvidas para melhorar a experiência do cliente com um produto ou serviço. User Experience está em tudo o que consumimos, desde a fruta que compramos na feira perto de casa até um celular novo. Independentemente de existir uma equipe focada nisso ou não, a UX vai acontecer de toda forma. O que as empresas estão fazendo atualmente é unindo diversas áreas do conhecimento para tornar essa experiência inesquecível. Usabilidade, arquitetura da informação, User Interface, Marketing e atendimento são algumas das ferramentas usadas para fazer com que o cliente se encante desde o primeiro contato. Assim, a UX vai além da jornada de compra, ela é desenhada para permanecer com o cliente durante toda a vida útil daquele produto ou serviço. Uma UX bem planejada pode causar efeitos ainda mais duradouros e influenciar positivamente vários aspectos da vida do cliente. A venda de uma simples camiseta pode gerar um grande impacto dependendo da forma que a experiência foi desenhada. Muito além de gerar lucros, a aplicação de UX traz consumidores fiéis para a marca, o que em um mundo social como o de hoje, é extremamente valioso. (HostGator, 2021)

### 3 METODOLOGIA

Neste capítulo será abordado sobre as tecnologias utilizadas para o desenvolvimento do protótipo como ferramentas utilizadas, arquitetura implementada, linguagens, bibliotecas, e instrumentos utilizados.

Para este projeto, o protótipo é construído como uma aplicação web multiplataforma para principalmente sistemas *desktop*, que também possa ser instalada na máquina se assim o usuário preferir.

#### 3.1 FERRAMENTAS

Como o sistema é um *software* voltado principalmente para a web, utilizou-se da tríade das linguagens web tal como a biblioteca do ReactJS em conjunto com Typescript para adicionar tipagem aos tipos de dados utilizados, também foram utilizadas algumas funcionalidades disponíveis no Firebase que são o *realtime database* e a autenticação para a criação de usuários. E para a simulação em si, foram utilizados componentes SPICE tal como uma biblioteca capaz de interpretar circuitos escritos dessa forma.

No Quadro 1 pode-se ver as tecnologias versão e o contexto no qual foram utilizadas.

Quadro 1 – Tecnologias utilizadas

Nome	Versão	Descrição
HTML	5	Linguagem de marcação
CSS	3	Linguagem de estilização
TYPESCRIPT	4.5.2	Linguagem de programação derivada do Javascript porém com tipagem nas variáveis
REACTJS	17.0.2	Biblioteca para criação de interfaces de usuário na WEB
WEBASSEMBLY	1.0	Bytecode para implementação de códigos de baixo nível para serem executados no navegador
Firebase RealTime Database	1.0.0	Banco de dados não relacional de tempo real, para a sincronização dos circuitos montados
NGSPICE	35	Ferramenta para simulação



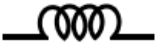




Fonte: O Autor (2022).

### 3.2 INSTRUMENTOS E SIMBOLOGIAS

Com o objetivo de diminuir o máximo possível a curva de aprendizado no uso do sistema, foram utilizadas simbologias de componentes já conhecidas no mercado no contexto de criação de esquemáticos, o padrão escolhido foi o americano, e o símbolo e a descrição dos componentes mais comuns é possível visualizar na Quadro 2.






Quadro 2 – Componentes utilizados

Símbolo	Nome
	Diodo
	Capacitor
	Indutor
	Resistor
	Fonte de alimentação DC
	Fonte de alimentação AC
	Fonte alimentação por pulso

Fonte: O Autor (2022).

Também foram utilizados de alguns componentes para representar os instrumentos disponíveis para uso dentro do sistema, estes não possuem um padrão específico em vista que cada simulador adota aquilo que achar mais conveniente, então para este, utilizou-se de formas geométricas simples com terminais para identificar instrumentos tal como voltímetro, amperímetro, e até mesmo os mais sofisticados como um osciloscópio.

Quadro 3 – Ferramentas de medição utilizados

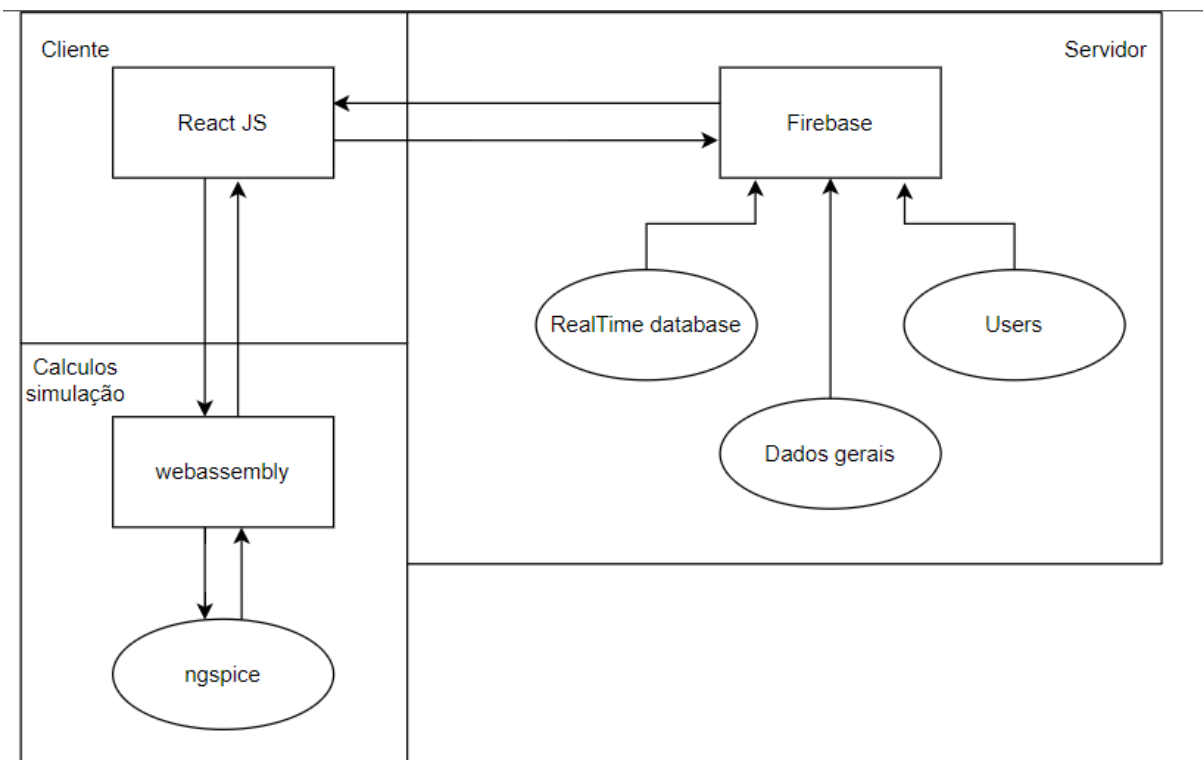
Símbolo	Nome
	Voltímetro
	Amperímetro
	Osciloscópio

Fonte: O Autor (2022).

### 3.3 ARQUITETURA DO SOFTWARE

O sistema pode ser explicado separando-o em 3 módulos diferentes cada um responsável por uma função específica, uma parte é responsável pela parte do cliente, onde é visualizado toda a interface e usabilidade do sistema, outra é responsável pela parte do servidor, onde ocorre toda a parte de autenticação de usuários e sincronização dos dados que são apresentados, e outra responsável pela simulação propriamente dita, onde pega os dados montados do circuito e realiza todos os cálculos dentro do hardware do cliente para apresentar os valores em tela, esta arquitetura é representada conforme a Figura 14.

Figura 14 - Arquitetura geral do sistema

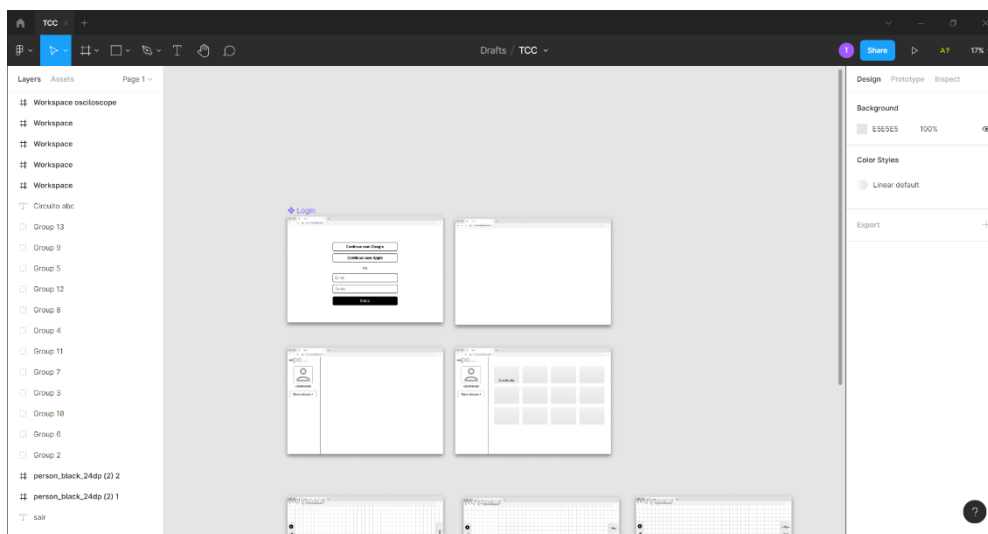


Fonte: O Autor (2022).

### 3.3.1 CLIENTE

O lado do cliente é o responsável pela parte com a qual o usuário terá um contato direto, e é onde ocorre toda a experiência do usuário com o uso do sistema, que é desenvolvido com base na biblioteca do ReactJS como base principal, a interface primeiramente é feito um protótipo de tela como apresentado na Figura 15, sem nenhum código envolvido, apenas para servir como guia para o desenvolvimento em si, em um *software* de prototipação de telas chamado Figma.

Figura 15 - Interface do Figma com o design do sistema



Fonte: O Autor (2022).

A tela em si é feita seguindo o conceito de componentes onde cada pedaço presente nela é um pedaço modular, o que promove um aumento na performance com relação a renderização, e leitura de código, para futuras implementações de novas funcionalidades ou até mesmo manutenção do código.

### 3.3.2 SERVIDOR

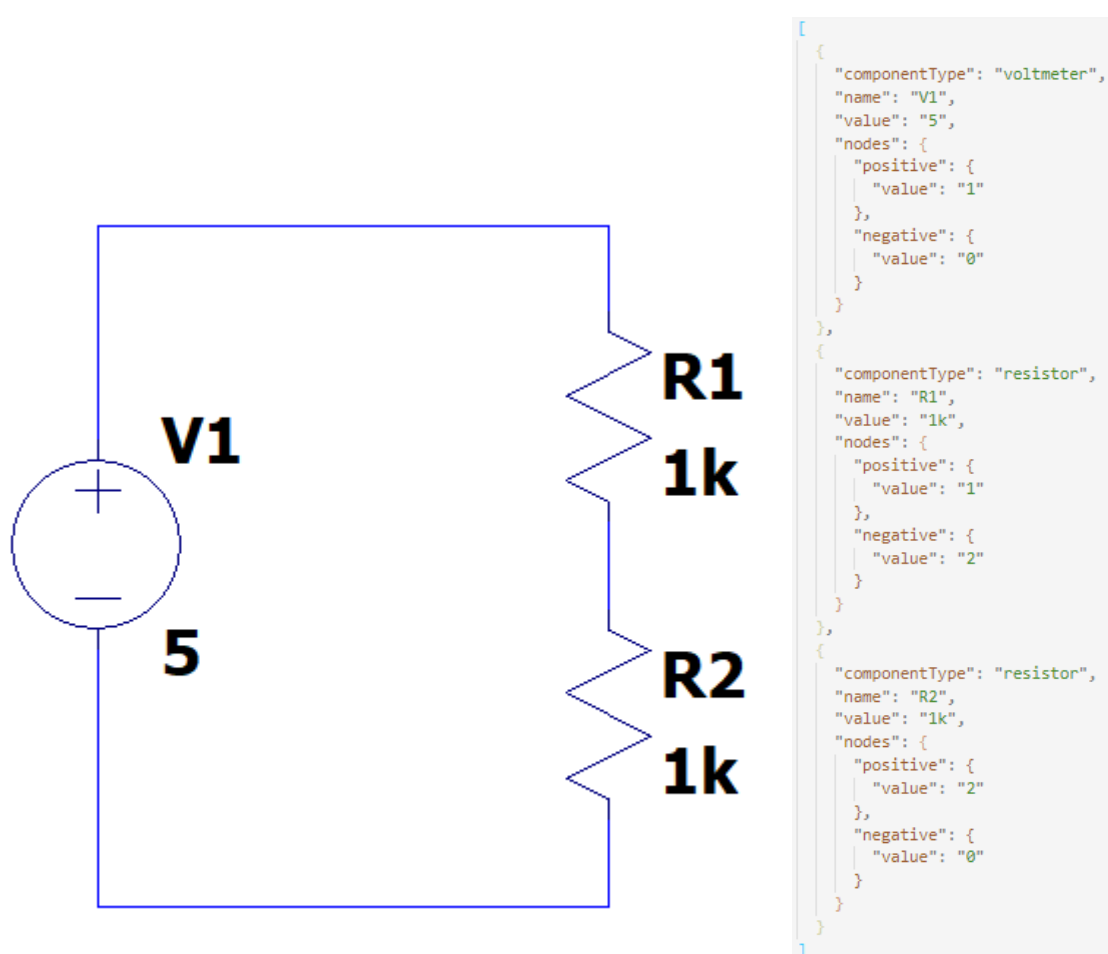
O lado responsável pela comunicação entre telas e por toda a parte colaborativa do simulador e permitir a experiencia de construção de circuitos com outras pessoas em tempo real, é o lado do servidor também conhecido como *backend*, que neste caso é utilizado um BAAS que é o Firebase.

Com o intuito de ser um serviço utilizado como *backend*, usa-se principalmente de duas funções do Firebase, a autenticação de usuários, e o banco de dados em tempo real.

A autenticação de usuários, pode ser feita tanto com o cadastro de um e-mail e senha, mas também com um login social como google ou Facebook por exemplo, dando uma facilidade maior para quem for entrar pela primeira vez no sistema, e isso serve também para a identificação de usuários para poder vincular os circuitos e salvar, assim garantindo que o trabalho feito não seja perdido.

E se tratando do banco de dados em tempo real não relacional, ela serve da seguinte maneira, existindo o banco configurado no Firebase, ao criar um circuito, um novo documento é criado, esse documento possui um formato JSON, e pode ser modificado de diversas formas para simbolizar a conexão entre os componentes no circuito, e na Figura 16, é possível visualizar como fica um exemplo de uma representação de um circuito em um formato JSON.

Figura 16 - Circuito em esquema e em JSON

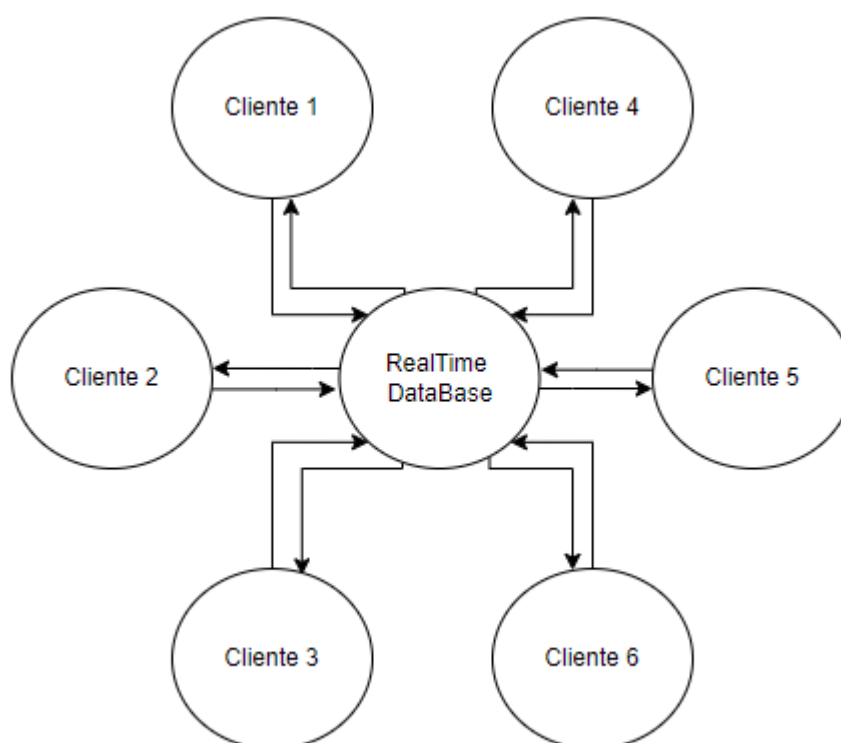


Fonte: O Autor (2022).

Com um documento criado, este terá um identificador único, o que permitirá com que o circuito seja compartilhado com outra pessoa, e ambos terem a visão do mesmo circuito, e por ser um banco de dados em tempo real, o lado cliente fica sempre escutando o documento com o identificador específico, e toda vez que o mesmo sofrer

alguma modificação automaticamente todos os cliente conectados vão ser atualizados com esta atualização, onde o exemplo dessa comunicação é demonstrado na Figura 17, permitindo assim com que a atualização de qualquer pessoa seja visualizada por qualquer um que esteja conectado no mesmo circuito.

Figura 17 - Comunicação em tempo real



Fonte: O Autor (2022).

Por fim, além das funções descritas, o servidor também é responsável pelo armazenamento de todas as informações dos componentes, tal como especificações, símbolo, e todo o tipo de dado necessário que precise ser armazenado dinamicamente para a construção da interface.

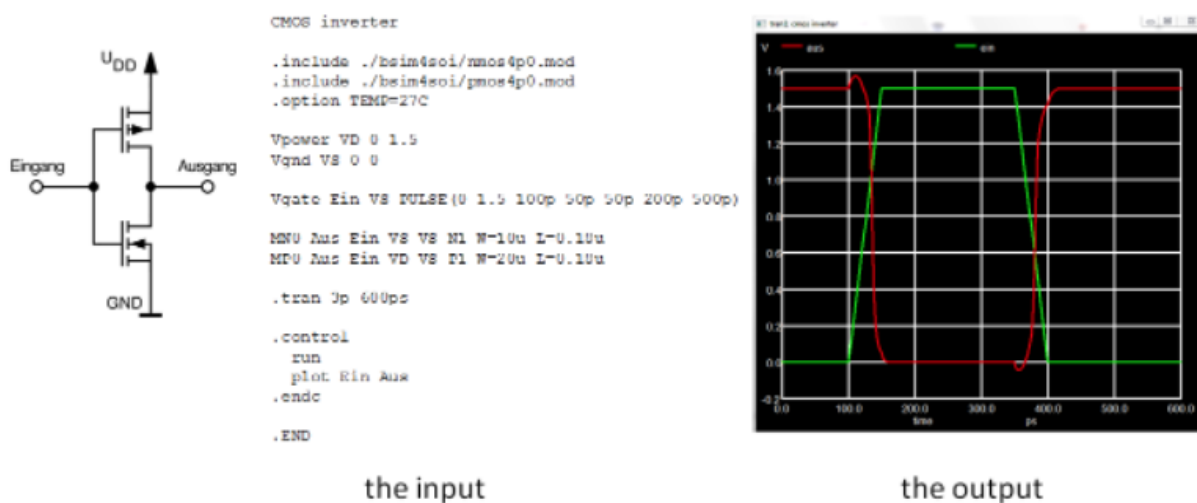
### 3.3.3 CÁLCULOS DA SIMULAÇÃO

Por se tratar de um sistema de simulação de circuitos, é requerido que muitos cálculos sejam executados muitas vezes, o que pode impactar na performance com a

qual o usuário interaja com o sistema, e para minimizar qualquer tipo de atraso e má experiência com relação ao tempo de execução, utiliza-se do Webassembly, que é responsável não só por permitir que a execução dos cálculos utilize o processador da máquina do usuário, mas também, por dar a liberdade de compilar linguagens diferentes para o uso na web, tais como linguagens de baixo nível como C, C++ ou Rust, fazendo com que seja possível, usufruir de bibliotecas já existentes para a simulação de circuitos construídas em C, que já se provaram competentes para o papel de simulação de circuitos elétricos.

Tendo uma tecnologia capaz de compilar código baixo nível para sua utilização na web, optou-se por utilizar de uma plataforma também já consolidada que é a simulação com componentes SPICE, pois as próprias fabricantes já fornecem documentação de componentes nessa plataforma, fazendo com que a curva entre o que já existe e este novo *software* seja a mínima possível em questão de compatibilidade, e para suprir essa necessidade de simulação SPICE, optou-se por utilizar do Ngspice, que é um simulador SPICE de linha de comando, que possui bons resultados com relação a análise, porém por se tratar de um ferramenta de linha de comando, muitas pessoas optam por se distanciar, fazendo assim com que não seja tão atrativo o uso da mesmo pela maioria da população, na Figura 18 há um exemplo de como o é a entrada e saída de dados computados no Ngspice.

Figura 18 - Entrada e saída NGSPICE



Fonte: NGSPICE (2022).

Então, para efetuar todos os cálculos e dados da simulação, como o servidor já possui o formato de como o circuito deve ser montado, e os clientes já estão montando devidamente a apresentação dele, utiliza-se da estratégia do Ngspice como Webassembly, para com que cada cliente conectado utilize do hardware do seu próprio sistema para efetuar os cálculos dos valores de simulações então apresentados em tela.



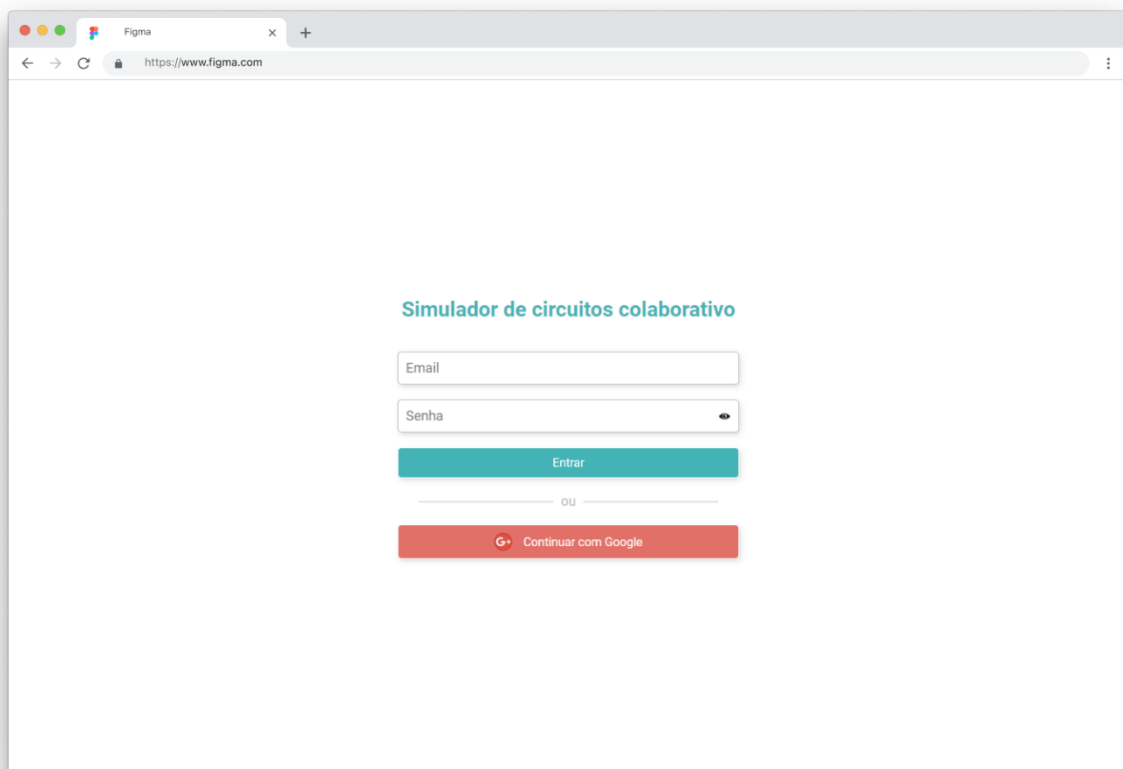
## 4 ANÁLISE DOS RESULTADOS

Neste capítulo é apresentado o *software* e seu funcionamento, e avaliações de usuários habituados a utilizarem simuladores de circuitos, sendo eles em sua maioria estudantes das áreas de engenharia e estudantes de ensino técnico em áreas relacionadas ao desenvolvimento de circuitos.

### 4.1 FLUXO DE LOGIN

Para começar a utilizar o sistema é necessário, que o usuário efetue um login para que o sistema possa vincular os projetos seguintes a quem está fazendo, na Figura 19, é apresentado que este login pode ser feito, tanto com um e-mail e senha do usuário, quanto com um login social com o Google, facilitando a inserção de novos usuários.

Figura 19 – Login do simulador

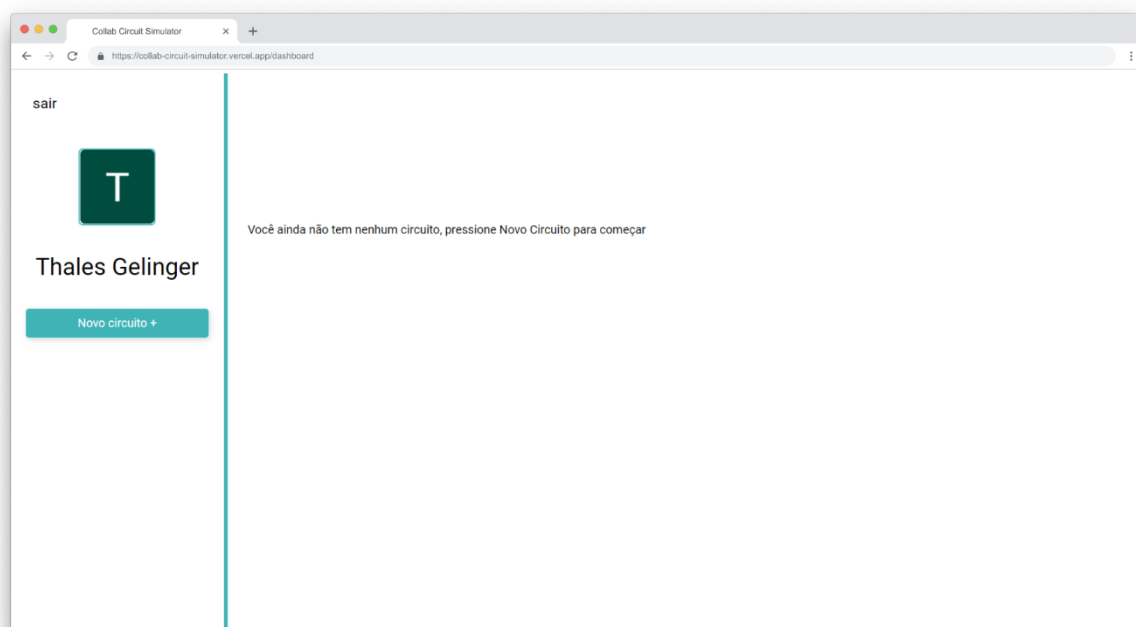


Fonte: O Autor (2022).

## 4.2 FLUXO DE CIRCUITO

Ao entrar no sistema, é apresentado a tela na Figura 20, onde é possível ver alguns dados do usuário, como o nome e a foto fornecida pelo provedor do login, tal como todos os projetos vinculados a ele e um botão para criar um circuito, se o usuário é novo no sistema então ele exibe uma mensagem escrita, “Você ainda não tem nenhum circuito, pressione Novo Circuito para começar”.

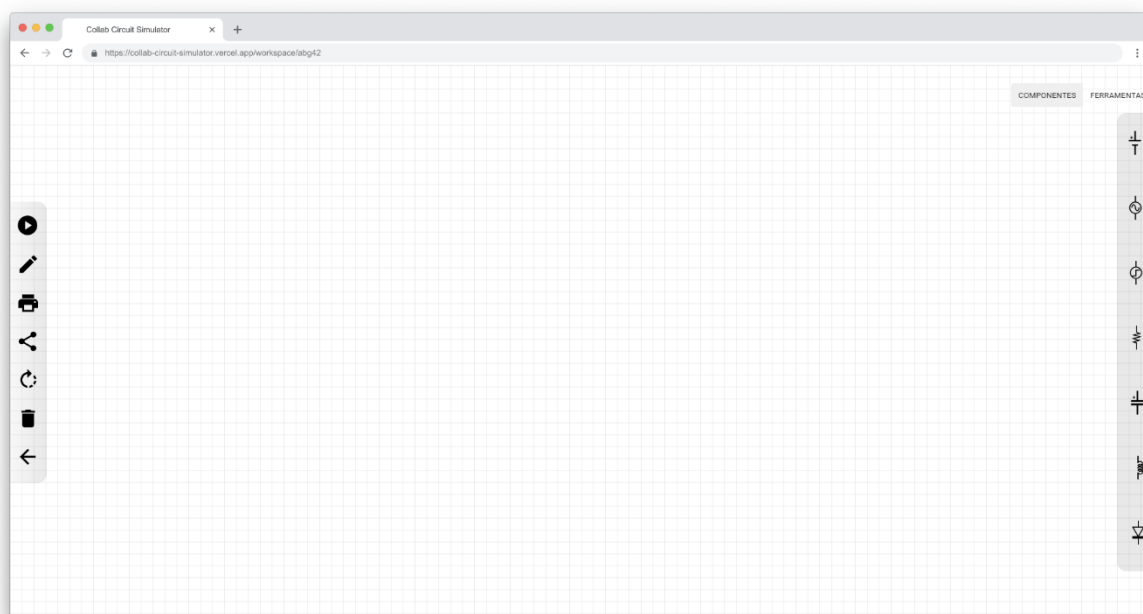
Figura 20 - Dashboard vazio



Fonte: O Autor (2022).

Ao criar um projeto, é apresentado então o espaço de trabalho do simulador em si vazia, como está na Figura 21, a qual é possível criar circuitos, simulá-los e compartilhá-los para um desenvolvimento cooperativo em tempo real, e a partir do momento em que um projeto é criado e o circuito começa a ser construído, toda e qualquer alteração é salva automaticamente em um banco de dados remoto sem a necessidade de clicar em algum lugar para salvar, ou correr o risco de perder o trabalho feito.

Figura 21 - Área de trabalho vazia



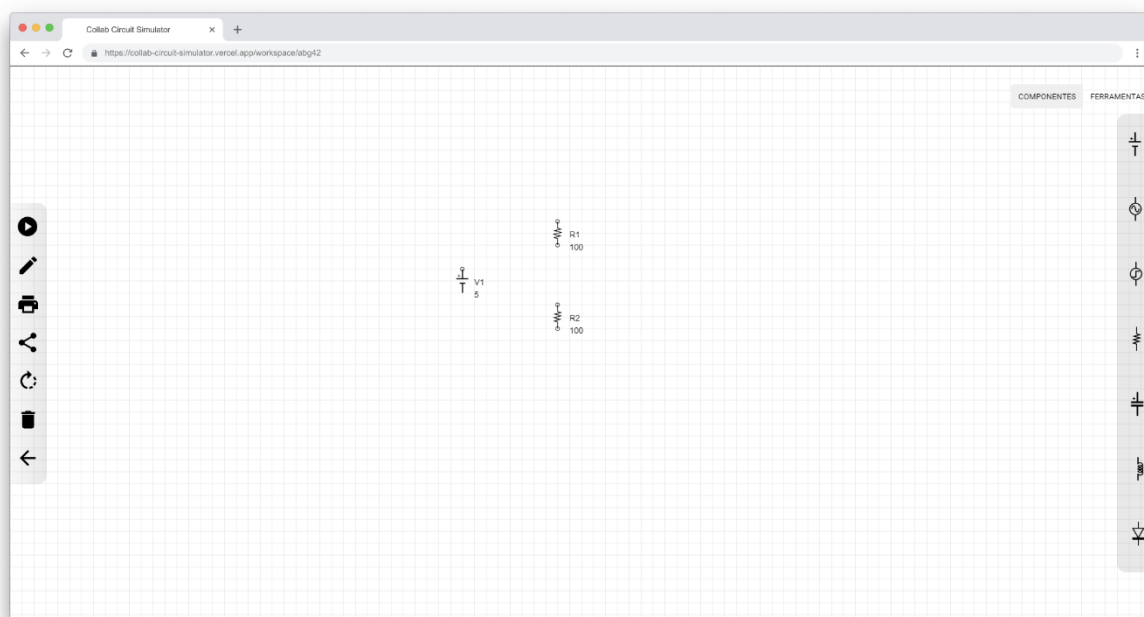
Fonte: O Autor (2022).

Toda a área quadriculada do sistema é uma área útil para o posicionamento de componentes, esta área é limitada ao tamanho da tela do usuário, possui também no lado esquerdo uma barra de ferramentas relacionadas a área de trabalho e a simulação, e na área direita uma barra com os componentes disponíveis para simulação que pode ser substituída por uma barra de ferramentas caso o usuário selecione no botão acima por FERRAMENTAS.

#### 4.2.1 Inserir componentes na área de trabalho

Para inserir componentes na área de trabalho, o usuário deve se dirigir até a área lateral direita, verificar se está selecionada para mostrar componentes, mover o cursor do mouse até o componente desejado e segurar o componente e arrastá-lo até a posição desejada, por fim com os componentes posicionados em tela haverá algo como o exemplo na Figura 22.

Figura 22 - Área de trabalho com componentes

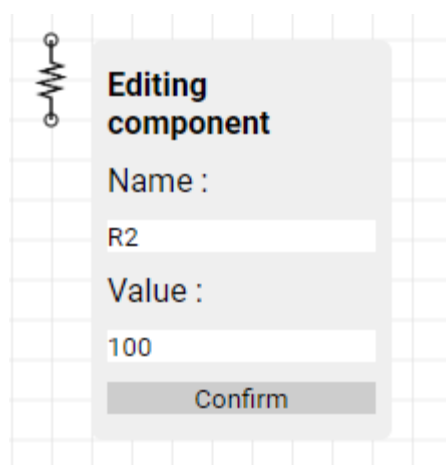


Fonte: O Autor (2022).

Com os componentes adicionados a área de trabalho, o usuário pode movê-lo da mesma forma como achar necessário para qualquer lugar dentro da área quadricula.

Para mudar os valores dos componentes, basta soltá-lo na área de trabalho e clicar duas vezes sob o rótulo ao lado, que isso fará abrir uma caixa de texto como na Figura 23, permitindo a troca de valor e de nome do componente selecionado.

Figura 23 – Caixa de edição de componente



Fonte: O Autor (2022).

Além da mudança de valores, o usuário também pode rotacionar o componente para melhor encaixá-lo no circuito, para isto existe uma ferramenta para rotacionar na aba na lateral esquerda, com um ícone de rotação conforme Figura 24, que ao selecionar basta clicar em cima do componente já posicionado na area de trabalho e o mesmo ira rotacionar para até 4 angulos diferentes, 0, 90, 180, 270 graus.

Figura 24 – Ação de rotacionar componente



Fonte: O Autor (2022).

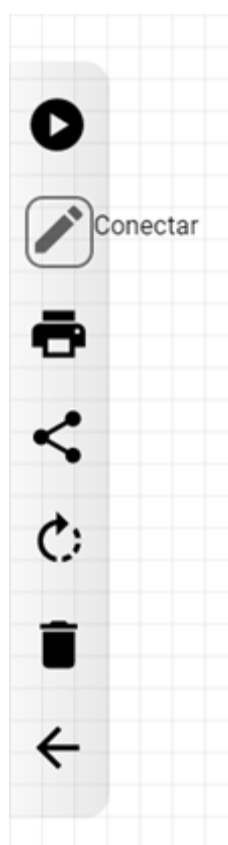
Ao selecionar qualquer item na aba de ferramentas na esquerda, todos possuem confirmações visuais para o usuário, ao passar o ponteiro do mouse por cima do ícone, o mesmo apresenta um texto ao lado representando o significado dele para deixar mais claro o significado daquela função, e ao clicar se o componente não está selecionado, ele faz a seleção e apresenta um contorno representando qual função

está ativa no momento, e para desativar, pode-se clicar novamente no ícone, ou simplesmente pressionar a tecla escape, que isto faz com que o sistema volte para o comportamento padrão que é de mover componentes.

#### 4.2.2 Conectar componentes

Com os componentes devidamente posicionados, é possível conectá-los com o uso da ferramenta “Conectar” também encontrada na barra esquerda, representada por um ícone de lápis como na Figura 25.

Figura 25 – Ação de conectar

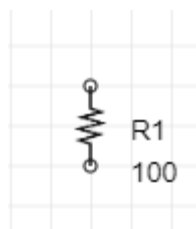


Fonte: O Autor (2022).

Para o usuário criar uma nova conexão, após verificar que a ferramenta conectar foi selecionada, é necessário clicar em algum ponto onde seja possível

adicionar uma conexão, ou seja, em um terminal de um componente como na Figura 26, ou em outra conexão que já esteja posicionada dentro do sistema.

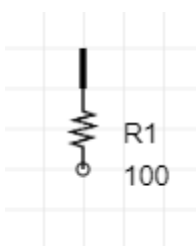
Figura 26 – Componente sem conexão



Fonte: O Autor (2022).

É possível notar também, que o componente possui um ícone de um círculo vazio em seu terminal quando ele não possui conexão, desta forma o usuário pode perceber visualmente quando um componente ainda não foi devidamente conectado, e verificar uma linha sendo formada a partir do ponto selecionado como na Figura 27 e seguindo o ponteiro do mouse até que usuário clique em outro ponto de conexão permitido.

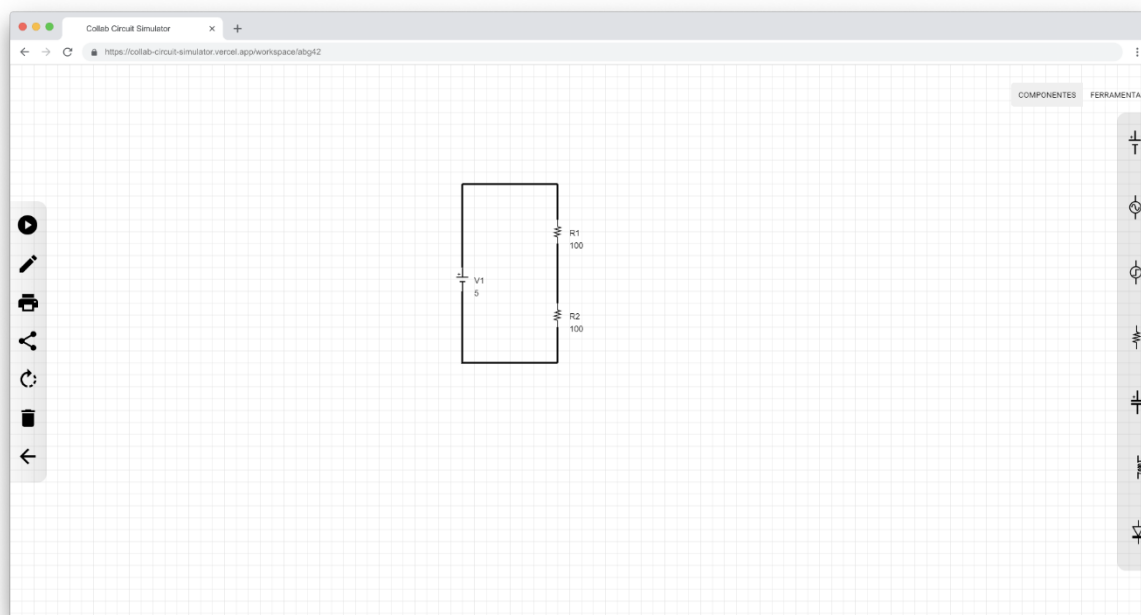
Figura 27 – Componente com conexão



Fonte: O Autor (2022).

Por fim, na Figura 28 pode-se ver que com as linhas adicionadas, o circuito terá então seus respectivos nós ligando cada componente.

Figura 28 - Área de trabalho com componentes conectado



Fonte: O Autor (2022).

### 4.2.3 Remover Componentes e Conexões

Caso o usuário tenha de adicionar um componente a mais, ou fez uma conexão errada, o simulador dispõe de uma ferramenta de remoção com um ícone de lixeira como na Figura 29, que pode tanto apagar um componente quanto uma conexão permitindo-o que mude o que precisar ser mudado no circuito.

Figura 29 – Ação de remover



Fonte: O Autor (2022).

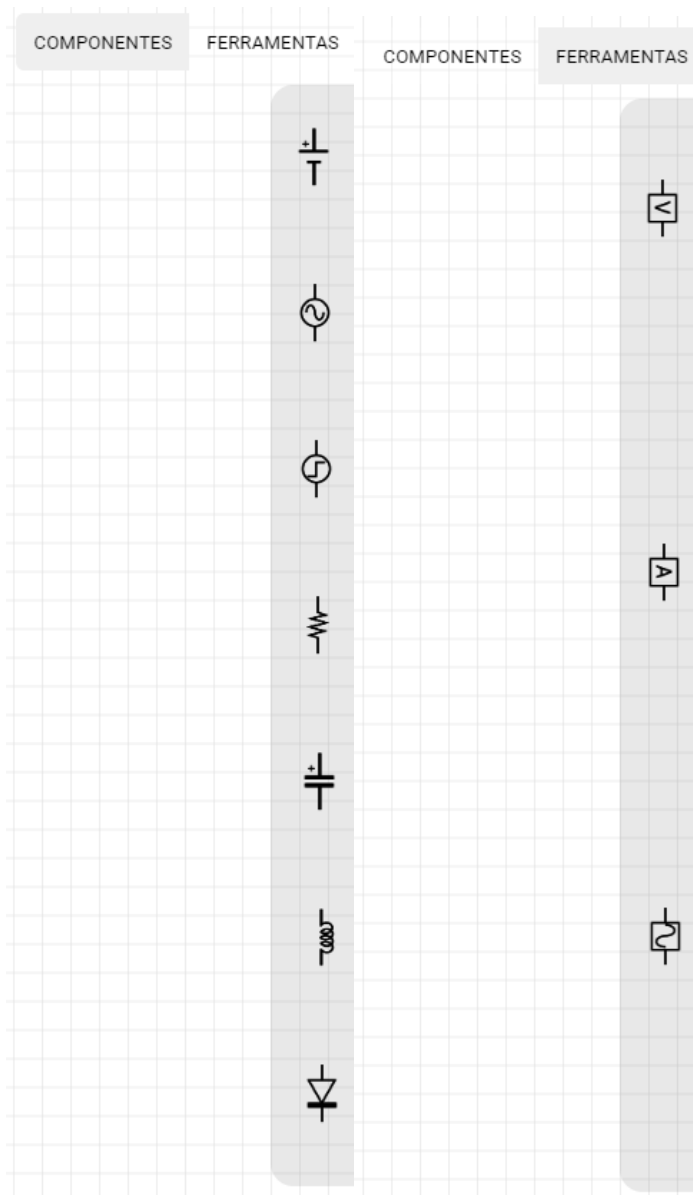
Com esta funcionalidade de remoção, conclui-se as ferramentas básicas de inserção, edição e remoção dentro do circuito.



#### 4.2.4 Simulação

Com o circuito devidamente montado e conectado, para efetuar a simulação, é necessário primeiro definir aquilo que se deseja medir, como tensão, ou corrente por exemplo, e para efetuar tais medições, selecionamos na aba da direita a opção de ferramentas, onde ela substitui os componentes por ferramentas de medições como na Figura 30, que são voltímetro, amperímetro e osciloscópio, então basta fazer o mesmo que é feito com os componentes, arrastar a ferramenta até a posição desejada e conectar os terminais nos pontos de medição no circuito,

Figura 30 – Barra de componentes antes e de ferramentas depois



Fonte: O Autor (2022).

Então para simular, é necessário selecionar o botão de simulação com ícone de play no canto superior da barra esquerda, que a simulação começará, e para um feedback visual, o ícone de play do botão de simular é substituído por um ícone de pause como demonstrado na Figura 31.

Figura 31 – Ícone de simular antes e durante a simulação

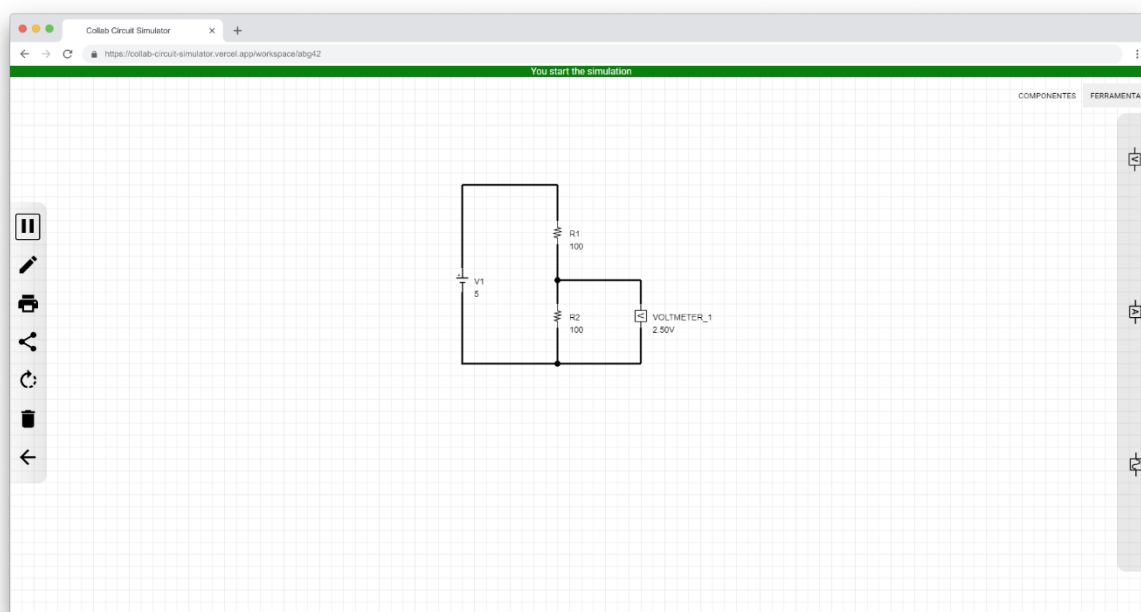


Fonte: O Autor (2022).

Com a simulação rodando, é possível notar uma tarja verde acima de tudo com a mensagem de que você começou a simulação, e os valores nas ferramentas sendo substituído pelos valores lidos no circuito.

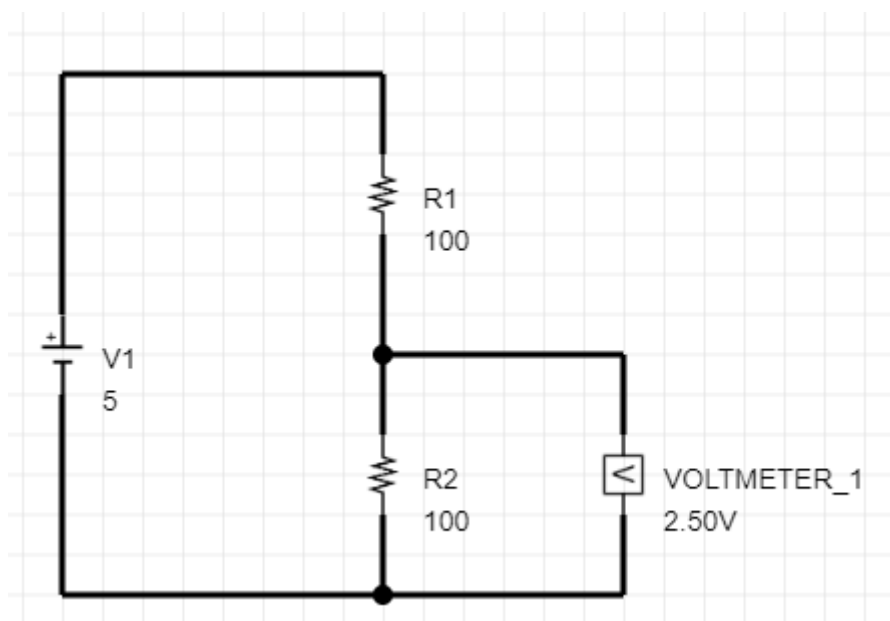
Essas modificações podem ser notadas nas Figuras 32 e 33.

Figura 32 - Área de trabalho com voltímetro conectado ao circuito



Fonte: O Autor (2022).

Figura 33 – Voltímetro apresentando medição



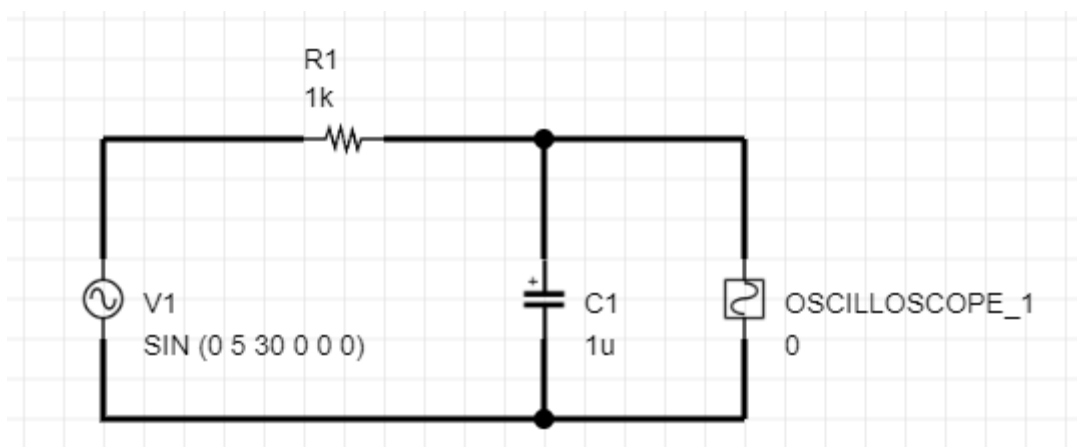
Fonte: O Autor (2022).

Como pode-se ver na Figura 33 acima, considerando um circuito simples de um divisor de tensão com uma fonte de 5V, temos o valor esperado de 2,5V.

#### 4.2.4.1 Osciloscópio

Em muitos circuitos não basta apenas os valores de tensão e corrente, mas sim ter uma visão detalhada em um gráfico de osciloscópio, e para adicionar isto, é necessário seguir o mesmo processo das outras ferramentas de medição, acessar a barra de ferramentas, seleccionar o osciloscópio, posicioná-lo no local desejado, e conectar os seus terminais no ponto que se deseja medir no circuito como no exemplo na Figura 34.

Figura 34 – Conexão de circuito com osciloscópio



Fonte: O Autor (2022).

Então ao iniciar a simulação, na Figura 35 é mostrado que o valor apresentado no osciloscópio muda para uma mensagem solicitando ao usuário para clicar sob o osciloscópio e revelar o valor medido.

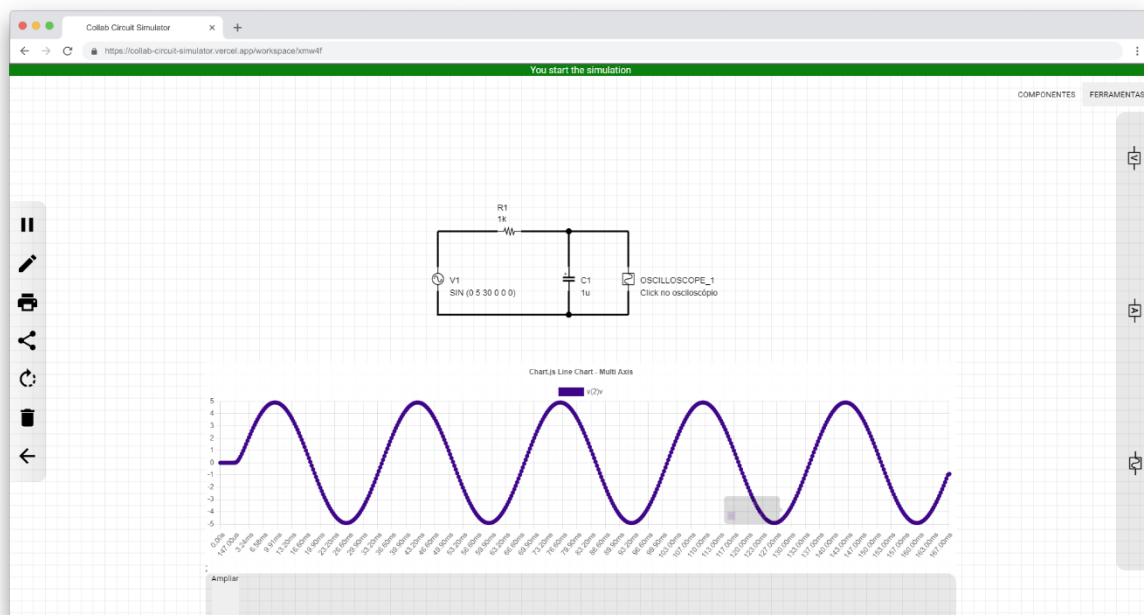
Figura 35 – Valor no osciloscópio ao iniciar simulação



Fonte: O Autor (2022).

Ao clicar duas vezes sob o osciloscópio, é apresentado um gráfico semelhante ao da Figura 36, na seção inferior da tela apresentando a onda medida no osciloscópio acoplado.

Figura 36 - Área de trabalho com osciloscópio conectado



Fonte: O Autor (2022).

#### 4.2.4.2 Colaboração

Com o objetivo de um ambiente colaborativo, para permitir com que outras pessoas participem da mesma área de trabalho contribuindo para o desenvolvimento do circuito, foi desenvolvido um método para o compartilhamento de circuitos de forma facilitada, e para acessar essa função, o usuário pode copiar a URL do circuito que ele está desenvolvendo e enviar para outra pessoa acessar, seja por meio de e-mail ou por um *chat*, ou clicar no botão de compartilhar demonstrado na Figura 37, na barra de ferramentas a esquerda.

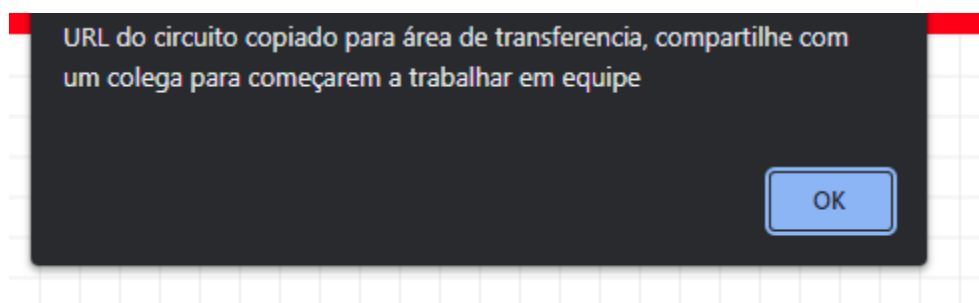
Figura 37 – Ícone de compartilhar



Fonte: O Autor (2022).

Ao clicar no botão de compartilhamento, ele também copiará a URL e logo após, emitirá uma mensagem como na Figura 38, informando para o usuário compartilhar com alguém.

Figura 38 – Mensagem ao clicar ícone de compartilhar

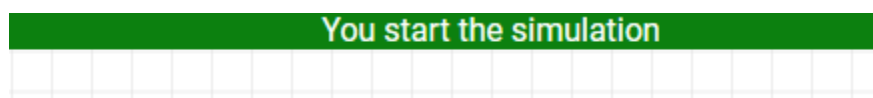


Fonte: O Autor (2022).

Ao enviar a URL para outra pessoa trabalhar no mesmo circuito, o usuário que a receber e cola-la no navegador, será direcionado para a página de login para se identificar, e logo após é redirecionado para a URL que foi colada, tendo a mesma visão do circuito daquele que enviou, e tendo também acesso a todas as funcionalidades antes descritas para a execução do circuito.

Entretanto, há algumas diferenças para o que cada usuário está olhando com relação a simulação, para poder avisar que um usuário está simulando, há uma regra para a aparência da tarja verde apresentada quando a simulação inicia. A regra basicamente, é de que o usuário que iniciou a simulação, vê uma tarja verde em seu monitor como na Figura 39, enquanto o outro vê uma tarja vermelha, informando que outra pessoa começou a simulação como na Figura 40.

Figura 39 – Tarja de quem roda a simulação



Fonte: O Autor (2022).

Figura 40 – Tarja de quem está no mesmo circuito, mas não roda a simulação



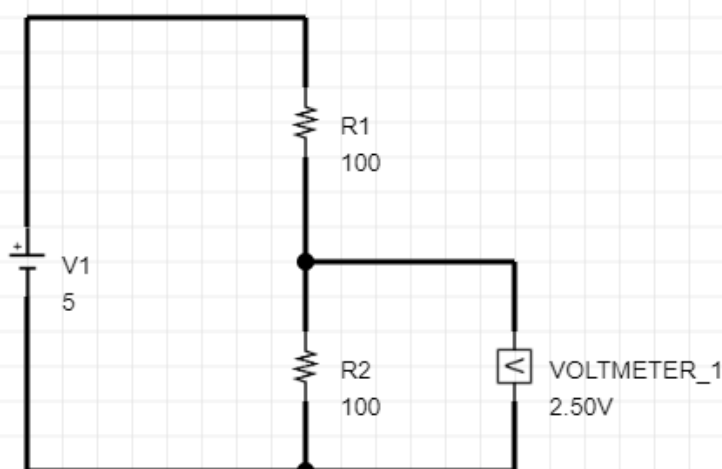

Fonte: O Autor (2022).

Dessa forma, é possível avisar aos colegas quando alguém está testando o circuito para não fazer nenhuma mudança nesse momento, porém, isso não impede o outro usuário de fazer alterações, é apenas um aviso visual de que outra pessoa está executando a simulação naquele exato momento.

Além disso, também é importante ressaltar, que ao rodar a simulação, os valores são apresentados apenas ao usuário que de fato está rodando a simulação, pois a simulação roda com webassembly, ou seja, cada execução da simulação é individual por máquina de cada usuário.

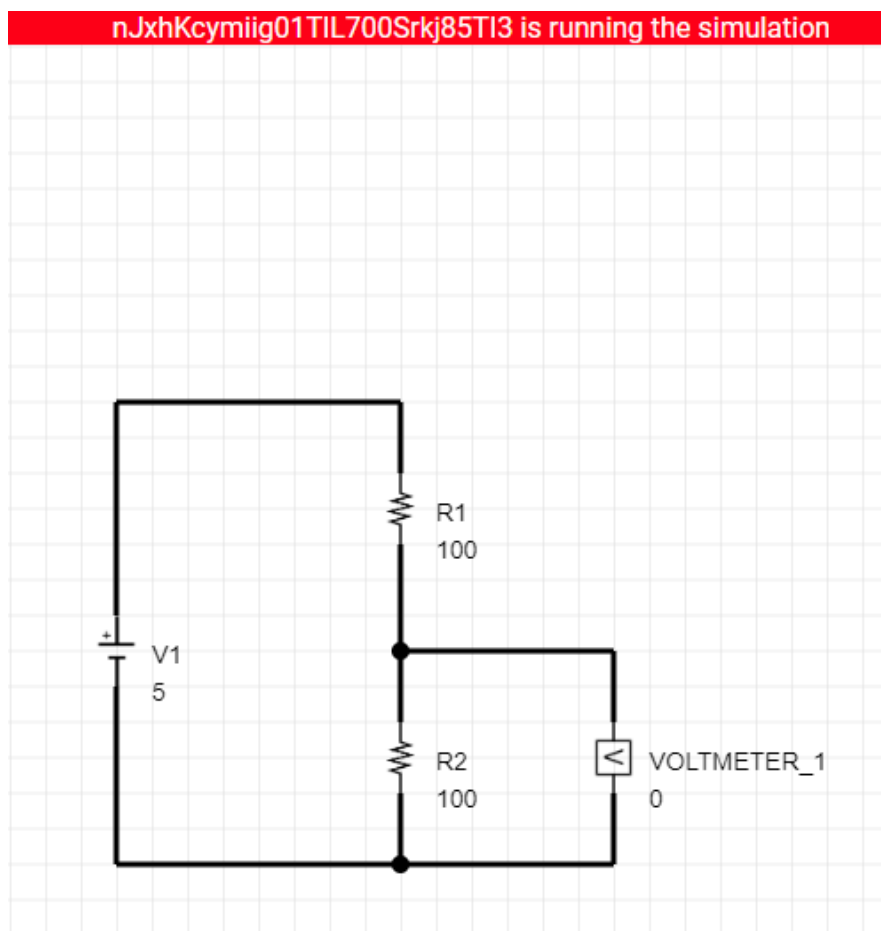
Pode-se visualizar nas Figuras 41 e 42 a diferença da visão do usuário que está rodando a simulação do usuário que não está.

Figura 41 – Visão de quem roda a simulação



Fonte: O Autor (2022).

Figura 42 - Visão de quem está no mesmo circuito, mas não roda a simulação



Fonte: O Autor (2022).

#### 4.2.4.3 Imprimir e Voltar

Além das funções já apresentadas, a barra esquerda ainda possui outras duas, uma para imprimir o circuito com um ícone de impressora como na Figura 43, que serve basicamente para o que o próprio nome diz.

Figura 43 – Ícone para imprimir circuito

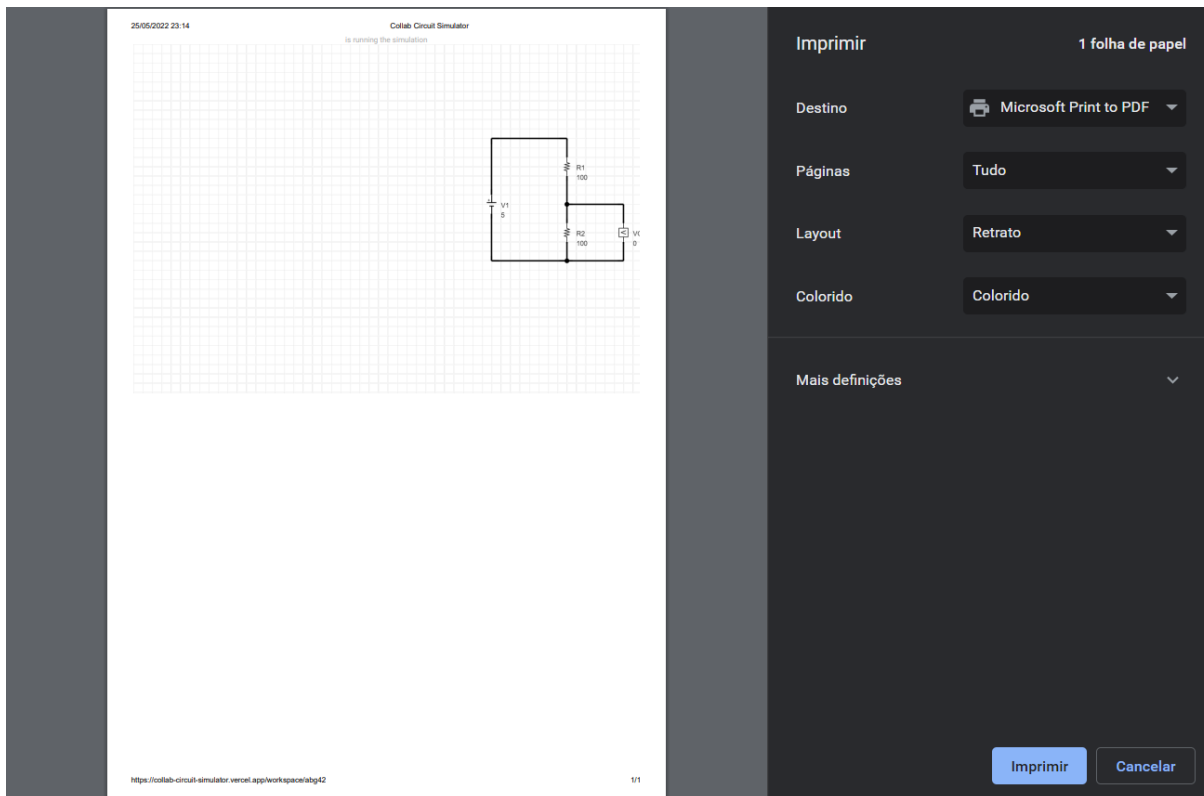


Fonte: O Autor (2022).



Ao selecioná-la abre-se um menu semelhante ao da Figura 44, para imprimir o circuito desenvolvido.

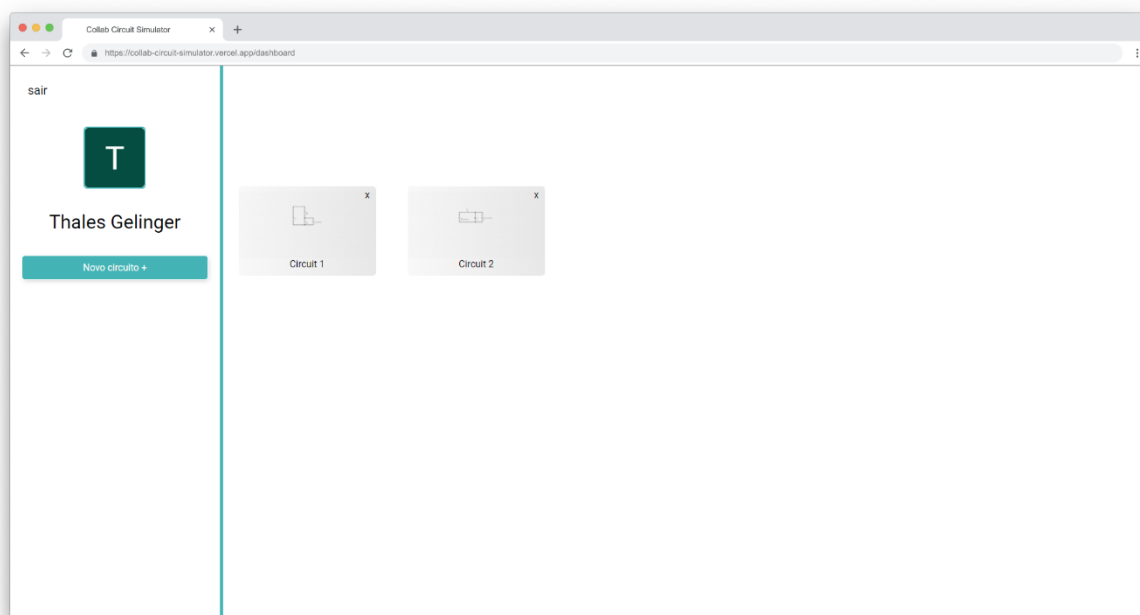
Figura 44 – Menu para imprimir circuito



Fonte: O Autor (2022).

E a outra função serve basicamente para voltar para a tela de dashboard, e possuir uma imagem do circuito como capa na tela de dashboard, dando assim um resultado como o na Figura 45, mostrando todos os circuitos desenvolvidos pelo usuário.

Figura 45 - Dashboard com circuito adicionado



Fonte: O Autor (2022).

### 4.3 COMPARAÇÃO COM OUTROS SISTEMAS

Dentre os *softwares* para simulação de circuitos já existentes, há aqueles com proposta similares na questão de serem executados dentro de um navegador, e proporcionar experiência multiplataformas, dentre eles temos o Multisim live, o Tinkercad e o Falstad, ambos com seus pontos fortes e fracos, porém com limitações que o *software* apresentado neste trabalho visa solucionar.

Comparando diretamente com o Tinkercad, temos um *software* de simulação que possui uma boa quantidade de componentes, e um grande diferencial que é a simulação com uso de Arduino, o que dá uma gama maior de possibilidades dentro do simulador, entre tanto, o que deixa a desejar no mesmo, é que ele possui um visual muito mais de ferramenta de simulação para coisas muito básicas do que de fato algo técnico que possa ser utilizado num meio acadêmico mais profissional, seja pelo fato das medições serem limitados quanto ao uso do osciloscópio, ou pelo fato da apresentação dos componentes serem uma representação 3D do que é montado de fato e não um circuito esquemático com simbologias conhecidas por quem trabalha

na área, e além disto, outro ponto fraco está na questão do peso que muitas vezes para coisas simples ele requer uma boa internet o que acaba dificultando para quem usa, e a opção de compartilhamento há usuários que relatam um funcionamento diferente do proposto, o *software* apresentado aqui, visa sanar justamente estas questões de possuir uma visualização de esquemático de fato, dar mais opções de medição de configuração do osciloscópio, e possuir uma opção de compartilhamento para um trabalho colaborativo no mesmo.

Com o Multisim live, possuímos um sistema muito completo com questão de medição apresentação e simplicidade, seu maior problema é o fato de ele ser limitado na sua versão gratuita, possuindo um limite de tamanho de área útil para a construção do circuito, e um limite de componentes que possam ser incluídos no circuito, limitação essas que são um grande complicador no caso de circuitos mais complexos, forçando o usuário a compra do sistema para a utilização, o que faz com que entusiastas e estudantes da área tendem a evitar na maioria das vezes, e o simulador aqui proposto, é ilimitado para ambas as questões de forma gratuita, tanto para o tamanho do circuito quanto pela área de trabalho disponível por ser uma área de fato infinita para uso, dando muito mais possibilidades para os mais variados tipos de circuitos independentemente do quão grande e complexo possa ser.

Se tratando do Falstad por mais que ele possua uma visão de circuito mesmo em forma de esquemático, e várias ferramentas para edição e medição do circuito, ele é uma ferramenta que não possui uma interface amigável, não apenas ao visualizar, mas ao utilizar o sistema não são ferramentas intuitivas, ou com um design que faça jus ao que precisa ser feito, com o simulador apresentado, um dos principais focos é justamente a questão de UI e UX para trazer pessoas para utilizarem-na e fazerem evoluir.

É notável que todas as ferramentas juntas possuem sim vários pontos fortes, mas também pontos fracos que possam ser empecilhos para algumas coisas, ou as vezes que poderiam ter alguma funcionalidade a mais que podem agregar ainda mais nas possibilidades do que pode ser feito com elas, funcionalidades essas que visam ser agregadas no *software* apresentado neste trabalho.

### 4.3.1 Comparação de simulações com o LTSpice

Em função de comparar a confiabilidade dos valores simulados, foram efetuados alguns testes comparando circuitos montados e medidos no LTSpice e no simulador apresentado.

#### 4.3.1.1 Simulação de circuito CC com duas fontes de tensão.

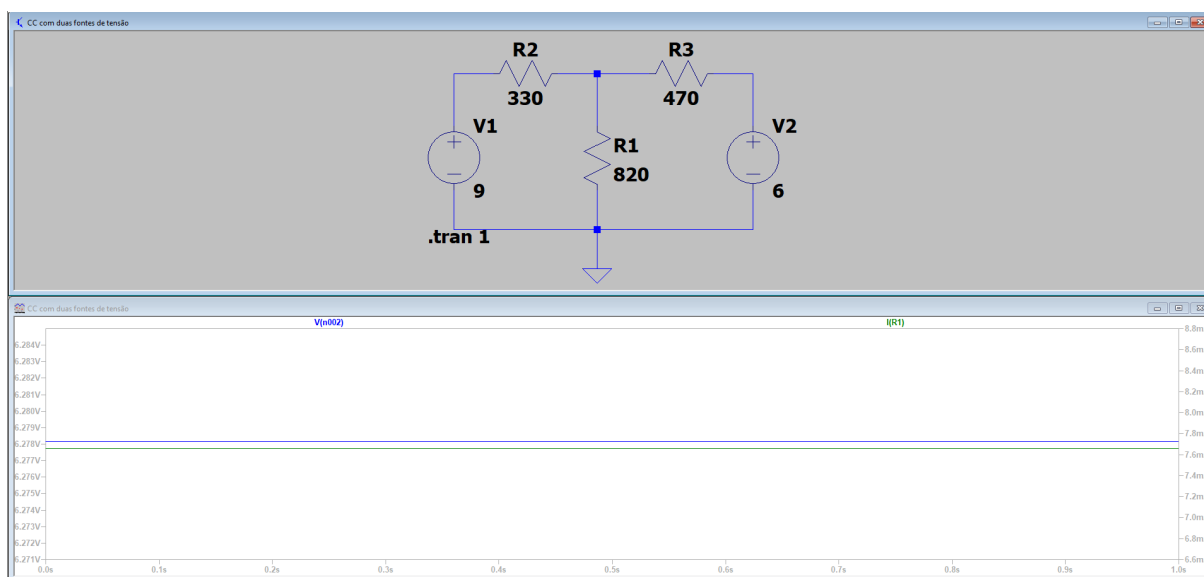
O primeiro teste foi de um circuito contendo duas fontes de tensão e medindo os valores em cima do R1, como pode-se ver no Quadro 4, e nas Figuras 46 e 47, ambos tiveram resultados muito semelhantes.

Quadro 4 – Comparação de valores medidos entre o LTSpice e o Colaborativo

Valores Medidos	LTSpice	Simulador Colaborativo
Vr1	6,278V	6,28V
Ir1	7,6mA	7,66mA

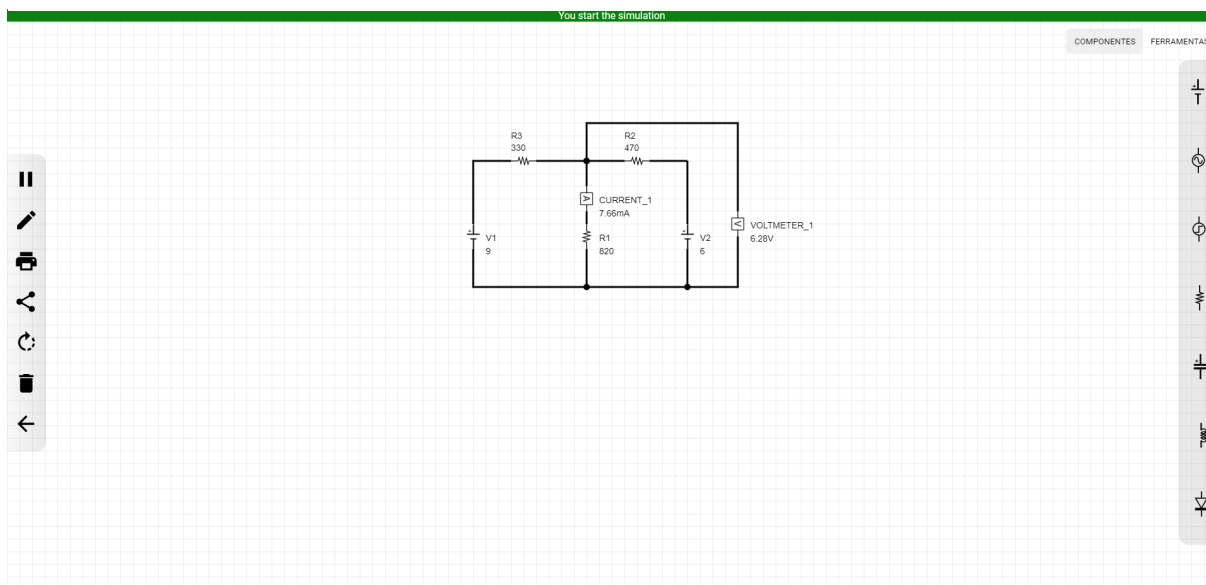
Fonte: O Autor (2022).

Figura 46 – Circuito CC com duas fontes de tensão LTSpice



Fonte: O Autor (2022).

Figura 47 - Circuito CC com duas fontes de tensão Colaborativo



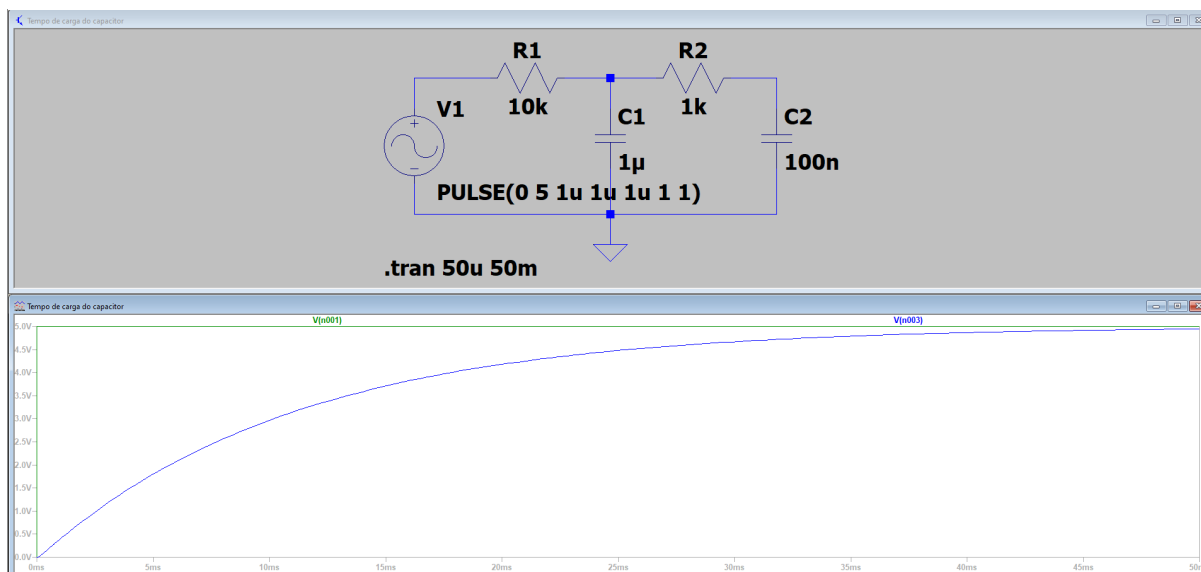
Fonte: O Autor (2022).

Nota-se uma pequena mudança de resultado entre o Simulador apresentado no projeto e o LTSpice no momento de medir a tensão em cima de R1, pois o simulador há um arredondamento para cima para mostrar com apenas duas casas decimais, mostrando assim essa diferença de 0,01V.

#### 4.3.1.2 Simulação de circuito medindo tempo de carga no capacitor

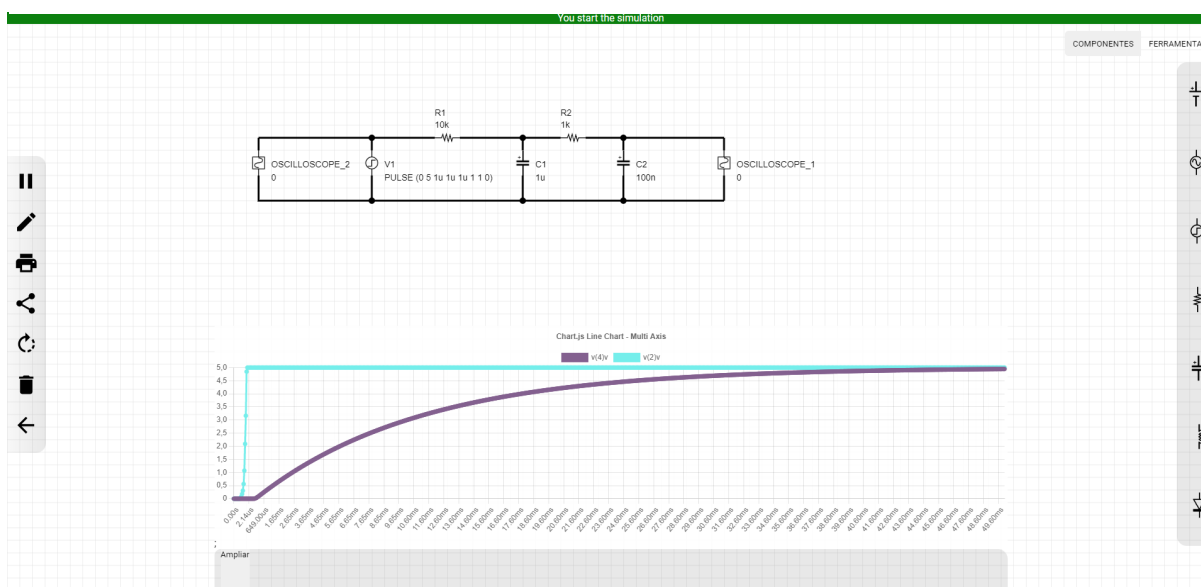
O segundo teste foi para verificar a carga em um capacitor utilizando uma fonte de sinal do tipo pulso, então para isso foram montados os mesmos circuitos com os exatos mesmos componentes, e o mesmo tempo de transiente, tendo assim os resultados como demonstrados nas Figuras 48 e 49.

Figura 48 - Circuito com tempo de carga no capacitor LTSpice



Fonte: O Autor (2022).

Figura 49 - Circuito com tempo de carga no capacitor Colaborativo



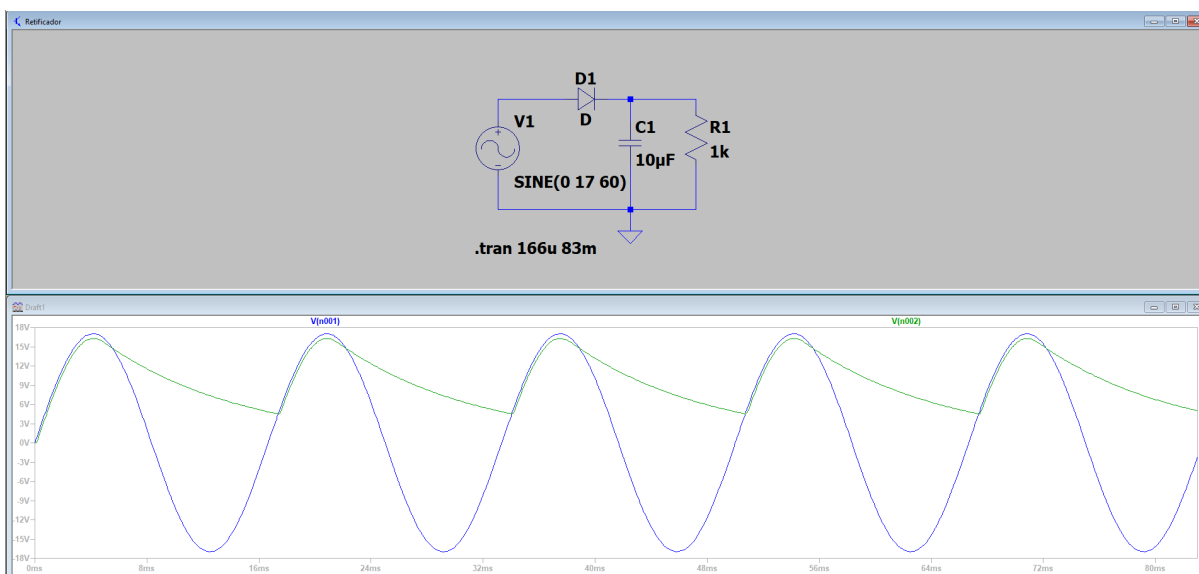
Fonte: O Autor (2022).

Nota-se uma leve demora no momento que de fato começa a subir tanto o sinal da fonte quanto no capacitor quando analisado ao simulador apresentado, isso se deve ao fato de que o número de pontos no gráfico do simulador, começa a partir do 0,0 e não possui tantos pontos no eixo x, e considerando que a fonte foi configurada para ter uma demora de 1us para inicializar, isso faz com que no simulador do projeto, esse tempo para subida fique mais evidente.

### 4.3.1.3 Simulação com circuito retificador

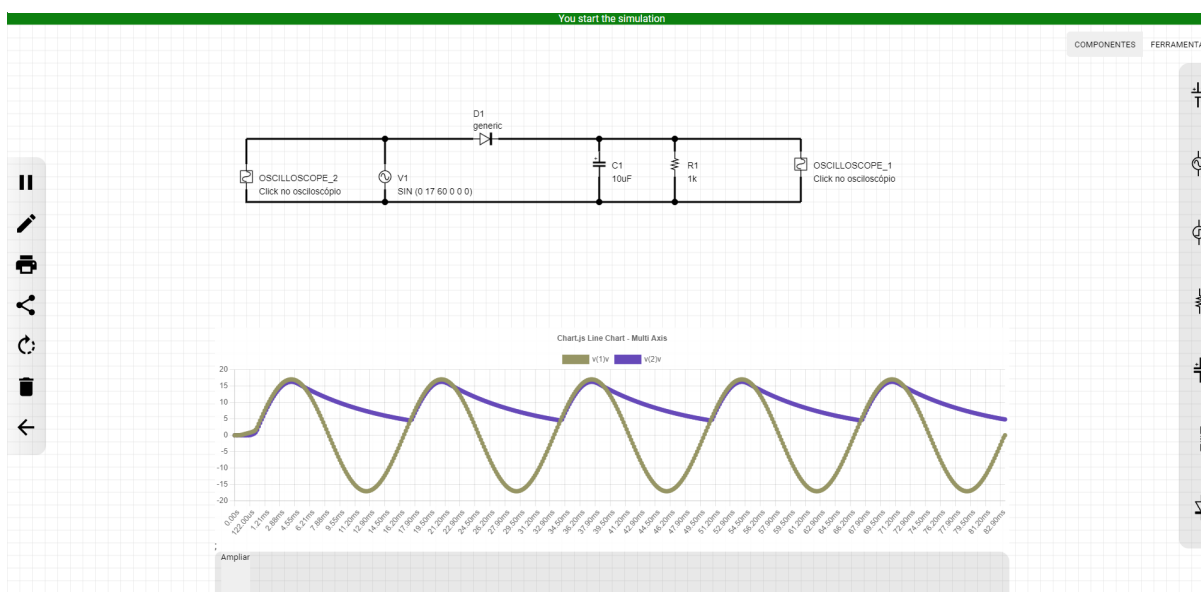
Por último, foi desenvolvido um circuito contendo um capacitor e um diodo para demonstrar a retificação e o tempo de descarga, e nota-se nos gráficos das Figuras 50 e 51 que eles tiveram um resultado semelhante com os mesmos valores.

Figura 50 – Circuito com onda retificada LTSpice



Fonte: O Autor (2022).

Figura 51 - Circuito com onda retificada colaborativo



Fonte: O Autor (2022).

Nota-se novamente uma leve demora no início no gráfico do simulador apresentado, também devido ao número de pontos e o fato dele começar no 0,0 pois entre o 0,0 e até o próximo ponto, ele não possui uma subida linear até de fato ter o primeiro sinal de subida, tornando mais evidente essa quebra no primeiro momento em que é feita a medição.

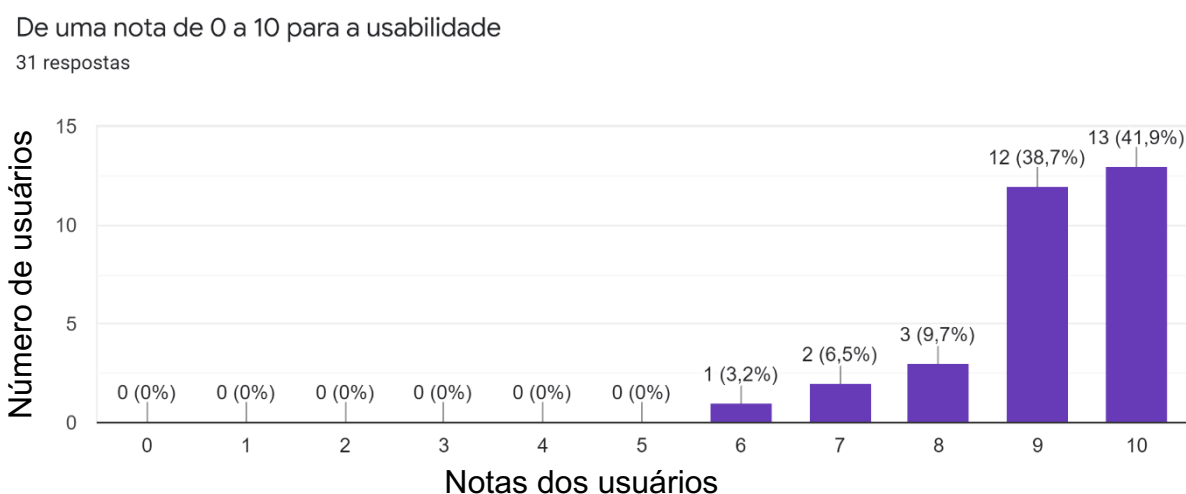
#### 4.4 AVALIAÇÃO DE USUÁRIOS

Os testes foram realizados com 31 usuários, e foi questionado sobre quais os simuladores que eles já possuíam algum contato, os simuladores mais comentados foram efetivamente aqueles citados no referencial teórico sendo o Proteus o mais popular entre os usuários, e tendo o conhecimento desses simuladores, os usuários responderam a um questionário sobre o simulador apresentado.

##### 4.4.1 Análise da usabilidade

Alguns usuários relataram complexidade na utilização de alguns simuladores, e pensando nisso, foi solicitado a eles avaliarem com uma nota de 0 a 10 o que acharam sobre a interface e usabilidade do simulador, esse resultado é possível verificar no gráfico abaixo na Figura 52.

Figura 52 – Avaliação dos usuários sobre a usabilidade do simulador apresentado



Fonte: O Autor (2022).



Analisando as notas dadas, é possível analisar que foram bem positivas entre a maioria dos usuários, entretanto alguns ainda encontraram alguns pontos na interface a se melhorar tais como algumas falhas visuais e detalhes de implementação, mas se tratando apenas da forma como se deve utilizar o simulador, foi uma resposta bem positiva.

#### 4.4.2 Análise do compartilhamento em tempo real

Dentre todas as funcionalidades aquela que mais traz um diferencial é o fato de poder trabalhar mais de uma pessoa no mesmo circuito ao mesmo tempo, e poder salvar o trabalho de forma automática, os usuários foram submetidos a está este teste, seja com outra pessoa de fato, ou com a sugestão do formulário de abrir o mesmo circuito em dois navegadores para visualizar a edição em tempo real, com isso, é possível visualizar na Figura 53 as notas dos usuários para a ferramenta de colaboração.

Figura 53 – Avaliação dos usuários sobre a função de colaboração



Fonte: O Autor (2022).

Nota-se que os usuários de fato aprovaram a opção de compartilhamento em tempo real, sendo que demonstraram um grande interesse em este tipo de funcionalidade, as notas mais baixas se deram ao fato de que alguns usuários relataram problemas na simulação envolvendo o compartilhamento de linhas

principalmente que ao apagar um a outra não apagava, mas além deste caso isolado, outros usuário conseguiram ter uma boa experiência com essa funcionalidade.

#### **4.4.3 Análise da simulação**

Como proposto, o simulador é capaz de resolver simulações simples com componentes passivos básicos, e dentre os testes realizados com usuários, isso foi possível, os usuários testaram em sua maioria simulações que envolviam fontes CC, alguns usuários foram mais além e testaram com a utilização de um osciloscópio, a maioria conseguiu ter um funcionamento esperado na simulação embora alguns sofreram com algumas instabilidades principalmente no momento de fazer as conexões, o que é aceitável tendo em vista que um *software* pode eventualmente ter alguns problemas que são corrigidos conforme vai ganhando mais maturidade.

## 5 CONCLUSÃO

Resgatando elementos do início do trabalho, nota-se que, o objetivo geral foi de desenvolver um simulador de circuitos online e colaborativo, que permita simulações simples com componentes essenciais para circuitos básicos, com uma interação em tempo real com mais de um usuário permitindo uma melhor dinâmica no desenvolvimento de circuitos.

Os usuários de fato sentem a necessidade de um simulador mais simples que cumpra o objetivo e seja mais amigável com o usuário, possuindo uma interface mais fluida e que remova todos os empecilhos de curva de aprendizado no uso, podendo simplesmente começar um novo circuito e poder desenvolvê-lo sem quaisquer complicações.

Dentre as funcionalidades, nota-se também que a de compartilhamento de circuitos em tempo real, é de fato algo que faz diferença no dia a dia do estudante, e permite com que discussões e o aprendizado sobre circuitos seja mais dinâmico, assim como em sala de aula circuitos são montados em duplas, ter essa mesma experiência fora da sala de aula, permite maior interação dos alunos e maior otimização no momento do desenvolvimento.

Levando em conta o objetivo geral e os específicos, é notável que o sistema desenvolvido, atendeu as expectativas, pois permitiu a interação em tempo real, e realizou efetivamente simulações simples, nota-se também que apesar de toda a funcionalidade, e levando em consideração o tempo de desenvolvimento do projeto, que alguns problemas podem ocorrer, e esses de fato ocorrem, porém foram erros superficiais que em sua maioria não apresentaram nenhum problema na usabilidade em si, foram erros esporádicos, e como qualquer *software* sempre há a possibilidade de melhorar e incrementar suas funcionalidades.

### 5.1 TRABALHOS FUTUROS

Tratando-se de trabalhos futuros, por ser um projeto de código livre, o objetivo então é divulgar e trazer mais usuários e desenvolvedores para a ideia do projeto e fazê-lo evoluir como um todo para poder suprir cada vez mais casos de simulação, entretanto em um primeiro momento, alguns usuários trouxeram ideias muito

relevantes, dentre elas comentam sobre um *chat* para dentro do próprio simulador poder se comunicar com o colega, a possibilidade de utilizar em um smartphone, que apesar de funcionar as simulações em ambiente móvel, a sua interface não é totalmente responsiva para uma melhor experiência.

Além disso, um grande foco é trabalhar em todos os possíveis problemas que possam ter passados despercebidos dentro da fase inicial, e incrementar com mais tipos de simulações, dando mais liberdade ao que o usuário desejar fazer.

E então, por ser um simulador SPICE, todos os modelos SPICE criados por diversos fabricantes podem ser inseridos ao projeto. Entretanto, da forma como o projeto está hoje, esse acoplamento poderá ser feito apenas adicionando no código, o que para a maioria dos usuários não é conveniente, fazendo assim com que uma funcionalidade de exportação seja necessária para poder usufruir de toda a possibilidade que o SPICE proporciona.

## REFERÊNCIAS

- Analog Devices. **LTSpice simulator**. 2021. Disponível em: <<https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>>. Acesso em 23 ago. 2021.
- Autodesk. **Tinkercad**. 2021. Disponível em: <<https://www.tinkercad.com/>>. Acesso em 27 ago. 2021.
- AWS. **Trabalhar com o JSON**. 2021. Disponível em: <[https://docs.aws.amazon.com/pt\\_br/sdk-for-javascript/v2/developer-guide/working-with-json.html](https://docs.aws.amazon.com/pt_br/sdk-for-javascript/v2/developer-guide/working-with-json.html)>. Acesso em 2 out. 2021.
- Balerini, Rafaela. **HTML, CSS e Javascript, quais as diferenças**. 2021. Disponível em: <<https://www.alura.com.br/artigos/html-css-e-js-definicoes>>. Acesso em 12 set. 2021.
- Bandlamudi, Vinod. **PCB Layout Design with Proteus**. Disponível em: <<https://www.engineersgarage.com/pcb-layout-design-with-proteus/>>. Acessado em 22 de mar. de 2022.
- Fedosekev, Artemij. **React.js Essentials: a fast-paces guide to designing and building scalable and maintainable web apps with React.js**. Packt Publishing. 2015
- GeekHunter. **Firestore: o que é e quando usa no desenvolvimento mobile?**. 2021. Disponível em: <<https://blog.geekhunter.com.br/firebase-o-que-e-e-quando-usar-no-desenvolvimento-mobile/>>. Acesso em 28 set. 2021.
- HostGator. **UX: O que é, sua importância e como aplicar**. 2021. Disponível em: <[https://www.hostgator.com.br/blog/ux-e-sua-importancia-e-como-aplicar/?gclid=CjwKCAjwzeqVBhAoEiwAOrEmzTQk0up5aivP4LcZ3jVQ4oY5BBZkJBf\\_eGDu\\_LVAnK1\\_zLQX6YdzSBoCd-8QAvD\\_BwE](https://www.hostgator.com.br/blog/ux-e-sua-importancia-e-como-aplicar/?gclid=CjwKCAjwzeqVBhAoEiwAOrEmzTQk0up5aivP4LcZ3jVQ4oY5BBZkJBf_eGDu_LVAnK1_zLQX6YdzSBoCd-8QAvD_BwE) />. Acesso em 28 de jun. 2022.
- Instituto Newton C. **Braga. Lição 1 conhecendo o Multisim 11**. 2021. Disponível em: <<https://www.newtoncbraga.com.br/index.php/cursos-on-line/95-cursos/multisim-11/3654-licao-1-conhecendo-o-multisim-11.html> >. Acesso em 22 ago. 2021.
- Jahn, Stefan. **Quite Universal Circuit Simulator**. 2021. Disponível em: <<http://qucs.sourceforge.net/>>. Acesso em 23 ago. 2021.
- Labcenter Electronics. **Proteus**. 2021. Disponível em: <<https://www.labcenter.com/>>. Acesso em 22 ago. 2021.

Maciejak, David. **WebAssembly 101: Bringing Bytecode to the Web**. 2017. . Disponível em: <<https://www.fortinet.com/blog/threat-research/webassembly-101-bringing-bytecode-to-the-web>>. Acesso em 27 mai. 2022.

MDN. **MDN Web Docs**. 2021. Disponível em: <<https://developer.mozilla.org/en-US/>>. Acesso em 30 ago. 2021.

National Instruments. **SPICE simulation Fundamentals**. 2021. Disponível em: <<https://www.ni.com/pt-br/innovations/white-papers/06/spice-simulation-fundamentals.html>>. Acesso em 20 ago. 2021.

NGSPICE. **Tutorial: ngspice for beginners**. 2021. Disponível em: <<http://ngspice.sourceforge.net/ngspice-tutorial.html>>. Acesso em 20 nov. 2021.

OpenJs Foundation. **NodeJs Certifications**. 2021. Disponível em: <<https://openjsf.org/>>. Acesso em 10 set. 2021.

Vertulo, Rodrigo Cesar. **Como funciona o Divisor de Tensão**. Disponível em: <<http://labdeeletronica.com.br/como-funciona-o-divisor-de-tensao/>>. Acessado em 27 de mai. de 2022.

Webassembly. **Webassembly**. 2021. Disponível em: <<https://webassembly.org/>>. Acesso em 18 de set. de 2021