

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA FLUMINENSE**

**PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À
ENGENHARIA E GESTÃO**

Raphael Viana Cruz

LABORATÓRIO VIRTUAL DE INSTALAÇÕES ELÉTRICAS

Campos dos Goytacazes/RJ

2023

2023

RAPHAEL VIANA CRUZ

MPSAEG / IFF

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA
E TECNOLOGIA FLUMINENSE**

**PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À
ENGENHARIA E GESTÃO**

RAPHAEL VIANA CRUZ

LABORATÓRIO VIRTUAL DE INSTALAÇÕES ELÉTRICAS

Rodrigo Martins Fernandes

(Orientador)

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão (MPSAEG), como parte dos requisitos necessários à obtenção do título de Mestre em Sistemas Aplicados à Engenharia e Gestão.

Campos dos Goytacazes/RJ
2023

Biblioteca
CIP - Catalogação na Publicação

C9571	<p>Cruz, Raphael Viana Laboratório Virtual de Instalações Elétricas / Raphael Viana Cruz - 2023. 155 f.: il. color.</p> <p>Orientador: Rodrigo Martins Fernandes</p> <p>Dissertação (mestrado) -- Instituto Federal de Educação, Ciência e Tecnologia Fluminense, Campus Campos Centro, Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão, Anton Dakitsch, RJ, 2023. Referências: f. 116 a 118.</p> <p>1. Laboratório Virtual. 2. Instalações Elétricas. 3. Motor de Jogos. 4. Unity. I. Fernandes, Rodrigo Martins, orient. II. Título.</p>
-------	---

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
FLUMINENSE

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS APLICADOS À
ENGENHARIA E GESTÃO

Raphael Viana Cruz

LABORATÓRIO VIRTUAL DE INSTALAÇÕES ELÉTRICAS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, no Curso de Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão (MPSAEG), como parte dos requisitos necessários à obtenção do título de Mestre em Sistemas Aplicados à Engenharia e Gestão.

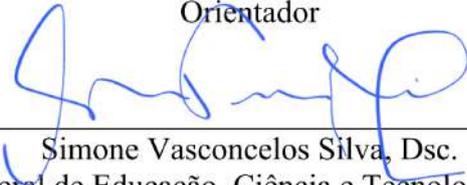
Aprovado em 09 de Fevereiro de 2023.

Banca Examinadora:

RODRIGO MARTINS FERNANDES:07977152746
2746

Assinado digitalmente por RODRIGO MARTINS FERNANDES:07977152746
NO: C=BR, O=ICP-Brasil, OU=Secretaria da Receita Federal do Brasil - RFB, OU=RFB-e-CPF-A3, OU=(EM BRANCO), OU=1187138805262, CN=RODRIGO MARTINS FERNANDES:07977152746
Razão: Eu sou o autor deste documento
Localização:
Data: 2023.04.19 11:19:05-03'07
Foxit PDF Reader Versão: 12.1.1

Rodrigo Martins Fernandes, Dsc.
Instituto Federal de Educação, Ciência e Tecnologia Fluminense
Orientador


Simone Vasconcelos Silva, Dsc.
Instituto Federal de Educação, Ciência e Tecnologia Fluminense
Membro Interno


Jorge Luiz Clemente Gomes, Dsc.
Instituto Federal de Educação, Ciência e Tecnologia Fluminense
Membro Externo

*Dedico este trabalho ao meu filho Arthur, por ser o farol que me guia em momentos
nebulosos.*

AGRADECIMENTOS

Agradeço aos meus pais, Marlo e Cristina, por sempre permitir e incentivar meu crescimento educacional.

Ao meu filho Arthur e minha esposa Priscilla, por compartilhar e me apoiar nos momentos difíceis nesta trajetória.

Agradeço ao Instituto Federal Fluminense, por sempre trazer um ensino de qualidade a população, transformando realidades de vida e fornecendo possibilidade de ascensão da classe mais humilde.

Aos professores do SAEG, por compartilharem seus conhecimentos ao longo do curso.

Agradeço em especial ao meu orientador D.Sc. Rodrigo Martins Fernandes, pelo ensino, confiança e paciência no processo de orientação.

Por fim, a todos os colegas que participaram com palavras de encorajamento e incentivo.

RESUMO

A utilização de ferramentas digitais aplicadas no ensino de disciplinas técnicas e propedêuticas ampliou-se, acelerada pela pandemia de COVID-19 devido às aulas remotas deste período. Os laboratórios virtuais são uma tecnologia consolidada, porém demandam aplicação de recursos financeiros para aquisição, também ocorrendo baixa oferta de soluções voltadas a dispositivos móveis. Este trabalho tem como objetivo o desenvolvimento de laboratório virtual multiplataforma aplicado aos cursos da área elétrica e a disciplina de Instalações Elétricas do Instituto Federal Fluminense, sem custos, replicando o setor e bancada didática utilizada em aula. Como metodologia de desenvolvimento foi aplicado o Modelo-V, utilizando a ferramenta de modelagem tridimensional *Blender* e o motor de jogos *Unity* ao processo. No primeiro momento foi realizada a modelagem 3D de 19 ambientes que compõem o setor, 1 bancada didática e 6 práticas apresentada na disciplina. Após modelagem foi realizada a construção do simulador baseado no método de análise de circuito denominado de Equação de Tableau, assim possibilitando a análise da prática montada pelo discente. No motor de jogos foi realizada a construção de 30 cenas, assim tornando a aplicação modular e facilitando sua manutenção. Para simulação dos comportamentos no ambiente virtualizado foram desenvolvidas 28 classes em C#. Como plataforma destino, a aplicação foi compilada para o sistema operacional *Windows*, podendo ser instalado localmente ou acessado via servidor web e em *Android* para instalação em dispositivos móveis. Nos testes propostos o trabalho demonstrou viabilidade de aplicação e foi aprovado, podendo ser aplicado em atividades presenciais e remotas, trazendo segurança aos discentes e servindo de modelo para os demais laboratórios do curso.

Palavras-chave: Laboratório Virtual, Instalações Elétricas, Motor de Jogos, *Unity*.

ABSTRACT

The use of digital tools applied in the teaching of technical and propaedeutic disciplines has expanded, accelerated by the COVID-19 pandemic due to the remote classes of this period. Virtual laboratories are a consolidated technology, but they require financial resources to be acquired, and there is also a low offer of solutions aimed at mobile devices. This work aims to develop a multiplatform virtual laboratory applied to courses in the electrical area and the Electrical Installations discipline at the Instituto Federal Fluminense, free of charge, replicating the sector and didactic bench used in class. As a development methodology, the V-Model was applied, using the three-dimensional modeling tool Blender and the Unity game engine to the process. In the first moment, 3D modeling of 19 environments that make up the sector, 1 didactic bench and 6 practices presented in the discipline were carried out. After modeling, the construction of the simulator was carried out based on the circuit analysis method called the Tableau Equation, thus enabling the analysis of the practice set up by the student. In the game engine, 30 scenes were built, thus making the application modular and facilitating its maintenance. To simulate behaviors in the virtualized environment, 28 classes were developed in C#. As a target platform, the application was compiled for the Windows operating system, and can be installed locally or accessed via a web server and on Android for installation on mobile devices. In the proposed tests, the work demonstrated the feasibility of application and was approved, being able to be applied in face-to-face and remote activities, bringing security to the students and serving as a model for the other laboratories of the course.

Keywords: Virtual Laboratory, Electrical Installations, Game Engine, Unity.

LISTA DE FIGURAS

Figura 1 – Diagrama para Lei das correntes.....	22
Figura 2 – Diagrama para Lei das tensões.....	23
Figura 3 - Dígrafo de $n = 5$ e $b = 7$	25
Figura 4 - Modelo V.....	29
Figura 5 - Diagrama de Caso de Uso.....	31
Figura 6 - Design de Alto Nível.....	35
Figura 7 - Fluxo geral do ambiente virtual.....	38
Figura 8 - Modelo 3D do corredor do setor de elétrica.....	39
Figura 9 - Jardim de inverno.....	40
Figura 10 - Corredor térreo e superior do setor de elétrica.....	40
Figura 11 - Laboratório de IEBT.....	41
Figura 12 - Modelo 3D da bancada didática.....	42
Figura 13 - Modelo 3D das conexões.....	43
Figura 14 - Diagrama multifilar do Interruptor Simples.....	45
Figura 15 - Diagrama multifilar da Campainha.....	45
Figura 16 - Diagrama multifilar da Cigarra.....	46
Figura 17 - Diagrama multifilar do Interruptor Paralelo.....	46
Figura 18 - Diagrama multifilar do Interruptor Intermediário.....	47
Figura 19 - Diagrama multifilar da Tomada 2P + T.....	48
Figura 20 - Relação de conexões do Interruptor Simples, Campainha e Cigarra.....	49
Figura 21 - Relação de conexões do Interruptor Paralelo.....	50
Figura 22 - Relação de conexões do Interruptor Intermediário.....	51
Figura 23 - Relação de conexões da Tomada 2P + T.....	52
Figura 24 - Modelo elétrico do Interruptor Simples, Cigarra e Campainha.....	53
Figura 25 - Modelo elétrico do Interruptor Paralelo (<i>Three-way</i>).....	54
Figura 26 - Modelo elétrico do Interruptor Intermediário (<i>Four-way</i>).....	57
Figura 27 - Modelo elétrico da Tomada 2P + T.....	61
Figura 28 - Dígrafos dos circuitos implementados.....	61
Figura 29 - <i>Layout menu</i> principal.....	73
Figura 30 - <i>Layout menu</i> de laboratórios.....	74
Figura 31 - <i>Layout</i> dos créditos do projeto.....	74
Figura 32 - <i>Layout</i> das configurações.....	75

Figura 33 - <i>Layout</i> do ambiente em 1ª pessoa.....	76
Figura 34 - <i>Layout</i> da tela de simulação.....	76
Figura 35 - Fluxograma do método <i>Conexao()</i>	78
Figura 36 - Fluxograma dos métodos <i>MaterialPreto()</i> , <i>MaterialVermelho()</i> , <i>MaterialAzul()</i> e <i>MaterialVerde()</i>	79
Figura 37 - Fluxograma do método <i>Selecioneado()</i>	79
Figura 38 - Fluxograma do método <i>OnOff()</i>	80
Figura 39 - Fluxograma do método <i>ValRes()</i>	80
Figura 40 - Fluxograma da classe <i>Branch</i>	81
Figura 41 - Fluxograma do método <i>CountNodes()</i>	82
Figura 42 - Fluxograma do método <i>Checar()</i>	83
Figura 43 - Fluxograma do método <i>Atual_A()</i>	84
Figura 44 - Fluxograma do método <i>MatN()</i>	84
Figura 45 - Fluxograma do método <i>CurtoCirc()</i>	85
Figura 46 - Fluxograma da classe <i>EqTableau</i>	86
Figura 47 - Fluxograma da classe <i>Pratica</i>	87
Figura 48 - Relacionamento entre as classes.....	89
Figura 49 - Estrutura de pastas <i>Unity</i>	92
Figura 50 - <i>Build Settings Unity</i>	93
Figura 51 - Cena do Menu Principal.....	94
Figura 52 - Configuração do botão Simular.....	94
Figura 53 - Tela inicial do corredor de circulação.....	95
Figura 54 - Tela de visualização e nomenclatura dos laboratórios.....	95
Figura 55 - Tela de Créditos.....	96
Figura 56 - Tela dos controles utilizados.....	96
Figura 57 - Tela de ajuste do áudio.....	97
Figura 58 - Acesso ao laboratório de Instalações Elétricas.....	97
Figura 59 - Laboratório de Instalações Elétricas.....	98
Figura 60 - Menu de seleção de prática.....	98
Figura 61 - Cena com a simulação das práticas.....	99
Figura 62 - Conexão sem material definido.....	100
Figura 63 - Circuito conectado e em funcionamento.....	100
Figura 64 - <i>Feedback</i> de circuito correto.....	102
Figura 65 - <i>Feedback</i> do diagrama.....	102

Figura 66 - <i>Feedback</i> de circuito incorreto.....	103
Figura 67 - <i>Feedback</i> de curto-circuito.....	104
Figura 68 - <i>Feedback</i> de tempo.....	104
Figura 69 - Variável booleana visualizada no <i>inspector</i> do <i>Unity</i>	105
Figura 70 - <i>Debug</i> da quantidade de ramos visualizados no console do <i>Unity</i>	105
Figura 71 - <i>Debug</i> da matriz $w(t)$ visualizado no console do <i>Unity</i>	105

LISTA DE QUADROS

Quadro 1 - Requisitos funcionais.....	33
Quadro 2 - Requisitos não funcionais.....	34
Quadro 3 - Matriz de incidência do Interruptor Simples, Campainha e Cigarra.....	62
Quadro 4 - Matriz de incidência do Interruptor Paralelo.....	63
Quadro 5 - Matriz de incidência do Interruptor Intermediário.....	63
Quadro 6 - Matriz de incidência da Tomada 2P+T.....	64
Quadro 7 - Testes realizados no subsistema <i>Corredor</i>	106
Quadro 8 - Testes realizados no subsistema <i>Laboratórios</i>	107
Quadro 9 - Testes realizados no subsistema <i>Simulação das Práticas</i>	108
Quadro 10 - Testes realizados no subsistema <i>Áudio</i>	109
Quadro 11 - Testes realizados no sistema em 1ª pessoa.....	109

LISTA DE TABELAS

Tabela 1 - Valor da resistência modelada de acordo com a posição da chave para o Interruptor Simples, Campainha e Cigarra.....	54
Tabela 2 - Valor da resistência modelada de acordo com a posição da chave para o Interruptor Paralelo.....	55
Tabela 3 - Valor da resistência modelada de acordo com a posição da chave para o Interruptor Intermediário.....	58
Tabela 4 - Resolução dos Equacionamentos.....	71

LISTA DE ABREVIATURAS E SIGLAS

2D	Ambiente bidimensional
API	<i>Application Programming Interface</i>
BT	Baixa Tensão
CLP	Controlador Lógico Programável
ConOps	Conceito das Operações
ddp	Diferença de potencial
FBX	<i>Filmbox</i>
HD	<i>High Definition</i>
IEBT	Instalações Elétricas de Baixa Tensão
IFF	Instituto Federal de Educação, Ciência e Tecnologia Fluminense
LKC	Lei de Kirchhoff das Correntes
LKT	Lei de Kirchhoff das Tensões
NBR	Norma Brasileira Regulamentadora
RA	Realidade Aumentada
RV	Realidade Virtual
SO	Sistema Operacional
TI	Tecnologia da Informação
UI	<i>User Interface</i>
USB	<i>Universal Serial Bus</i>

SUMÁRIO

1 INTRODUÇÃO.....	18
1.1 Justificativa.....	19
1.2 Objetivo Geral.....	20
1.3 Objetivo Específico.....	20
2 REFERENCIAL TEÓRICO.....	22
2.1 Laboratório Virtual.....	22
2.2 Lei de Kirchoff das correntes – LKC.....	22
2.3 Lei de Kirchoff das Tensões – LKT.....	23
2.4 Dígrafos.....	24
2.5 Equação de Tableau.....	25
2.6 Trabalhos Relacionados.....	26
3 METODOLOGIA.....	29
4 DESENVOLVIMENTO E RESULTADOS.....	31
4.1 Conceito das Operações.....	31
4.2 Especificação do Sistema.....	32
4.2.1 Requisitos funcionais.....	33
4.2.2 Requisitos não funcionais.....	34
4.3 Design de Alto Nível.....	35
4.4 Projeto Detalhado.....	36
4.4.1 Ferramentas computacionais.....	36
4.4.1.1 Blender.....	36
4.4.1.2 <i>Unity</i>	36
4.4.1.3 CADe SIMU.....	37
4.4.1.4 Gimp.....	37
4.4.1.5 Scilab.....	37
4.4.1.6 <i>Visual Code Studio</i>	37
4.4.2 Fluxo do programa.....	38
4.4.3 Modelagem 3D.....	39
4.4.4 Modelagem elétrica e equacionamentos.....	44
4.4.4.1 Diagrama dos circuitos modelados.....	44
4.4.4.2 Cálculo das possibilidades de conexões para cada circuito modelado.....	48
4.4.4.3 Modelo elétrico.....	52

4.4.4.4	Construção dos dígrafos.....	61
4.4.4.5	Construção da matriz de incidência.....	62
4.4.4.6	Equacionamento da LKC para os circuitos dos modelos elétricos.....	64
4.4.4.7	Equacionamento da LKT para os circuitos dos modelos elétricos.....	65
4.4.4.8	Equação dos ramos.....	67
4.4.4.9	Montagem das matrizes de Tableau.....	69
4.4.4.10	Solução dos circuitos modelados utilizando a equação de Tableau.....	71
4.4.5	Projeto de <i>front end</i>	72
4.5	Desenvolvimento de Software.....	77
4.5.1	Classes.....	77
4.5.1.1	Classes dependentes.....	77
4.5.1.1.1	<i>OnClick</i>	77
4.5.1.1.2	<i>Wires</i>	77
4.5.1.1.3	<i>SelecaoObjeto</i>	78
4.5.1.1.4	<i>LuzDestaque</i>	79
4.5.1.1.5	<i>TeclaOnOff</i>	80
4.5.1.1.6	<i>Branch</i>	81
4.5.1.1.7	<i>Nodes</i>	81
4.5.1.1.8	<i>Booleanas</i>	82
4.5.1.1.9	<i>Corrigir</i>	82
4.5.1.1.10	<i>Feedback</i>	83
4.5.1.1.11	<i>Update_A</i>	83
4.5.1.1.12	<i>MatrizN</i>	84
4.5.1.1.13	<i>CurtoCircuito</i>	84
4.5.1.1.14	<i>Output</i>	85
4.5.1.1.15	<i>EqTableau</i>	85
4.5.1.1.16	<i>Pratica</i>	86
4.5.1.1.17	<i>IrParaCena</i>	87
4.5.1.1.18	<i>SalvarPosic</i>	87
4.5.1.1.19	<i>Bancada</i>	88
4.5.1.1.20	<i>Trigger</i>	88
4.5.1.1.21	<i>Mensagem</i>	88
4.5.1.1.22	<i>AudioPlayInterruptor</i>	88
4.5.1.2	Classes sem dependências.....	90

4.5.1.2.1 <i>Timer</i>	90
4.5.1.2.2 <i>Buttons</i>	90
4.5.1.2.3 <i>MoveZoom</i>	90
4.5.1.2.4 <i>MusicPlayer</i>	91
4.5.1.2.5 <i>VolumeControl</i>	91
4.5.1.2.6 <i>DestroyGameObject</i>	91
4.5.2 Implementação no motor de jogos <i>Unity</i>	92
4.6 Testes Unitários.....	104
4.7 Teste do Subsistema.....	106
4.7.1 Subsistema Corredor.....	106
4.7.2 Subsistema Laboratórios.....	106
4.7.3 Subsistema Simulação das Práticas.....	107
4.7.4 Subsistema Controles.....	108
4.7.5 Subsistema Áudio.....	108
4.8 Teste de Sistemas.....	109
4.8.1 Sistema Ambiente em 1ª pessoa.....	109
4.8.2 Sistema Configurações.....	110
4.9 Teste de Aceitação.....	111
5 CONCLUSÃO.....	113
REFERÊNCIAS.....	115
APÊNDICE A – DESCRIÇÃO DOS CENÁRIOS.....	119
APÊNDICE B – PROJETO BANCADA DIDÁTICA.....	122
APÊNDICE C – QUADRO DE SÍMBOLOS.....	123
APÊNDICE D – DIAGRAMAS DE CLASSES.....	124
APÊNDICE E – LABORATÓRIOS QUE COMPÕEM O SETOR ELÉTRICO.....	125
APÊNDICE F – QUADRO COM TESTES UNITÁRIOS.....	128
ANEXO A – PESQUISA DE ACESSIBILIDADE REMOTA.....	145
ANEXO B – PLANTA BAIXA DO BLOCO B.....	153

1 INTRODUÇÃO

A evolução da tecnologia da informação (TI), impulsionada pelo aprimoramento dos computadores pessoais, dispositivos móveis (*smartphones*) e internet de banda larga, influenciou um meio para a disseminação de conhecimento antes restrito às salas de aula, agora em ambiente digital. A pandemia de coronavírus 2019 (COVID-19) evidenciou a necessidade e urgência na implementação de metodologias de ensino que promovam maior engajamento dos alunos, tanto em aulas presenciais quanto remotas, por meio da inserção de tecnologias digitais.

A utilização dos simuladores virtuais ampliou-se, sendo uma modelo de aprendizado já consolidado, aprimorando a capacitação e condicionamento dos profissionais na área militar, como o Simulador de Tiro Militar (*Assault Rifle Simulator – VICTRIX*) e o Simulador de Veículos Terrestre (*Ground Vehicles Simulation – SIMVEH*) desenvolvidos pela empresa Indra, o simulador para guindastes portuários e *offshore* e os simuladores para linha amarela desenvolvido pela empresa Oniria para as áreas portuária e industrial (INDRA, 2022; ONIRIA, 2022).

Além dos exemplos já mencionados, há os laboratórios virtuais, que são plataformas que permitem que os alunos realizem experimentos, testem hipóteses e explorem fenômenos de forma segura, sem a necessidade de um espaço físico de laboratório. O objetivo dessas plataformas é proporcionar uma experiência realista e imersiva para os alunos, permitindo que eles aprendam habilidades práticas e teóricas em um ambiente acessível e interativo. As simulações em laboratórios virtuais podem incluir a modelagem de sistemas complexos, a visualização de processos abstratos e a interação com equipamentos virtuais, proporcionando aos alunos uma oportunidade única de explorar e compreender conhecimentos científicos e técnicos.

Na área educacional empresas como DeLorenzo com seus simuladores de Comandos Elétricos e Máquinas Elétricas, Exsto com seu aplicativo de realidade aumentada (RA) para Comandos elétricos e a Algetec com laboratórios voltados a área de engenharia e saúde desenvolveram aplicações virtuais de suas bancadas didáticas físicas, assim ampliando seu portfólio de produtos e ocupando essa fatia importante no mercado. Uma aplicação também utilizada pelo Senai do Rio Grande do Sul é o Simulador de Planta Industrial, voltado para instaladores de quadro de comandos, onde é possível realizar a montagem de painéis industriais em um ambiente que simula os procedimentos necessários para um adequado funcionamento do quadro industrial em um ambiente em 1ª pessoa (ALGETEC, 2022; DLB, 2022; EXSTO, 2022; FIERGS, SENAI, 2022).

Outro exemplo de êxito na vertente educacional é a plataforma Phet – Simulações Interativas da Universidade de Colorado Boulder, fundada em 2002 pelo prêmio Nobel Carl Wieman, com mais de 1,1 bilhão de simulações executadas em todo mundo, abrangendo experimentos das áreas de física, química, matemática, ciências da terra e biologia (PHET, 2022).

1.1 Justificativa

Fundado em 1910, pelo então presidente da república Nilo Peçanha, como escola de Aprendizizes e Artífices, o Instituto Federal Fluminense (IFF) conta com 12 *campi*, além de um Polo de Inovação, um Centro de Referência em Tecnologia, Informação e Comunicação na Educação, da Unidade de Formação de Cordeiro e da Reitoria, distribuídos pelo estado do Rio de Janeiro, oferecendo uma estrutura verticalizada, com a formação inicial e continuada, os Cursos Técnicos de Nível Médio, as Licenciaturas, os Cursos Superiores de Tecnologia e os Bacharelados, estendendo-se a oferta de formação até a pós-graduação *lato e stricto sensu* (IFF, 2021).

Em 1966 foi criado o curso Técnico em Eletrotécnica, sendo o precursor das modalidades da área elétrica no instituto. Após reformulações em sua estrutura, foi instituído o *campus* Campos Centro e nele os cursos da área elétrica, sendo três de nível médio (Eletrotécnica Integrado, Concomitante e Proeja) e um de nível superior (Bacharelado em Engenharia Elétrica). Hoje estes constam com 16 laboratórios, divididos em 6 grandes áreas: Instalações Elétricas Prediais, Projetos Elétricos, Instalações Elétricas Industriais, Eletrônica, Sistemas Elétricos de Potência e Manutenção Industrial (IFF, 2021).

O Laboratório de Instalações Elétricas é responsável por fornecer a infraestrutura para ensinar os conceitos de Instalações Elétricas de Baixa Tensão (IEBT) com foco em residências. Para esse fim, o laboratório usa seis box para simular a distribuição de dispositivos eletroeletrônicos e uma bancada didática desenvolvido pelos alunos, apresentado na 26ª Feira do Saber-Fazer-Saber.

Assim, o ensino da IEBT tem se voltado para a metodologia tradicional, centrada no professor em aulas expositivas, sendo complementado pelo uso dos laboratórios físicos e bancadas didáticas que compõem os ambientes disponibilizados pela instituição.

Os laboratórios físicos demandam investimentos na aquisição e atualização dos equipamentos utilizados nas simulações da vida laboral de um profissional, assim como um fluxo contínuo de reposição de materiais de consumo necessários à complementação da atividade realizada.

Outro ponto que requer atenção é a disponibilidade do laboratório e de profissional para supervisão das práticas realizadas, devido ao risco de acidentes decorrentes da operação de equipamentos elétricos, podendo acarretar a inviabilidade do uso recorrente por parte do corpo discente.

Para sanar os problemas relacionados aos laboratórios físicos, pode-se utilizar os simuladores virtuais, porém também necessitando de investimento na aquisição das licenças de utilização. Outro ponto a ser analisado é que essas aplicações são construídas de acordo com a bancada física também comercializada pelas empresas, trazendo um descolamento na familiaridade dos discentes em aprendizado com a bancada do laboratório, caso este não tenha sido adquirido, também possuindo baixa oferta de produtos voltados a dispositivos móveis.

Em 2020 o IFF realizou pesquisa interna de acessibilidade remota que verificou a situação do acesso dos discentes a recursos de TI, para o então proposto ensino remoto e qual seria este recurso. Esta estudo evidenciou que 95,2% dos alunos dos cursos técnicos possuíam *smartphones* com acesso à internet contra 64% com acesso a computadores pessoais, este documento pode ser visualizada no [Anexo A](#). Em pesquisa realizada pela *Bain & Company* (2020), mostrou que 90% dos *smartphones* utilizados no Brasil possuem como sistema operacional (SO) o *Android* e 80% dos desenvolvedores trabalham com o mesmo SO.

O presente trabalho se justifica por criar um laboratório virtual, sem demanda de recursos financeiros por parte da instituição, para que os alunos possam desenvolver suas atividades remotamente por meio de dispositivos móveis (*smartphones*) ou computadores, complementando as atividades de ensino presenciais e remotas.

1.2 Objetivo Geral

Este trabalho tem como objetivo o desenvolvimento de um laboratório virtual multiplataforma sem custos adicionais para a instituição de ensino e voltado para utilização na disciplina de Instalações Elétricas de Baixa Tensão, emulando o ambiente físico onde se encontram os laboratórios das coordenações da área elétrica do Instituto Federal Fluminense – *campus* Campos Centro e a bancada didática utilizada para demonstração dos circuitos elétricos.

1.3 Objetivo Específico

O projeto tem como principais objetivos específicos a construção de modelos 3D dos ambientes físicos e da bancada didática, bem como o desenvolvimento de um simulador para análise de circuitos elétricos. Além disso, pretende-se implementar o ambiente virtual e as simulações no software Unity, visando à criação de um ambiente imersivo e interativo para os

usuários. Por fim, será necessário desenvolver e testar as simulações de seis práticas do laboratório, a fim de garantir a eficácia do simulador.

2 REFERENCIAL TEÓRICO

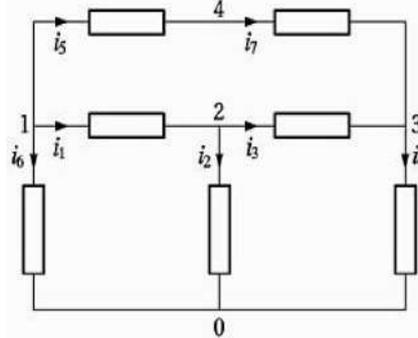
2.1 Laboratório Virtual

O laboratório virtual é um ambiente que simula objetos concretos, projetados através de linguagens computacionais, sendo modelos ideais para representação dos comportamentos dos objetos reais utilizados em meio material (PORTO, 2015).

2.2 Lei de Kirchhoff das correntes – LKC

Em um circuito elétrico as cargas não podem ser extintas ou criadas, assim a soma das correntes que saem de um nó é nula, considerando que as correntes que entram no nó têm sinal negativo. Em um circuito com n nós numerados de 0 a $n-1$, podemos representar n equações de corrente, tendo coeficientes $+1$, -1 e 0 sendo elas algébricas lineares, também podendo ser considerado que a soma das correntes que entram em uma região, sistema ou nó, tem que ser igual a soma das correntes que saem desta mesma região, sistema ou nó. Na Figura 1 mostra-se um circuito com 5 nós com os respectivos sentidos das correntes, assim possibilitando elaborar as equações características para cada nó, resultando na Equação 1 (BURIAN JÚNIOR, 2006; BOYLESTAD, 2004).

FIGURA 1 – DIAGRAMA PARA LEI DAS CORRENTES



Fonte: Burian Júnior (2006)

$$\begin{aligned}
 -i_2 - i_4 - i_6 &= 0 \\
 i_1 + i_5 + i_6 &= 0 \\
 -i_1 + i_2 + i_3 &= 0 \\
 -i_3 + i_4 - i_7 &= 0 \\
 -i_5 + i_7 &= 0
 \end{aligned} \tag{1}$$

Agrupando as correntes em um vetor coluna pode-se representar as equações independentes de forma matricial, de acordo com a Equação 2, onde a matriz A é nomeada de

matriz de incidência do circuito. Na Equação 3 mostra-se o equacionamento da LKC para o circuito demonstrado na Figura 1 (BURIAN JÚNIOR, 2006).

$$[A] \cdot [i] = 0 \quad (2)$$

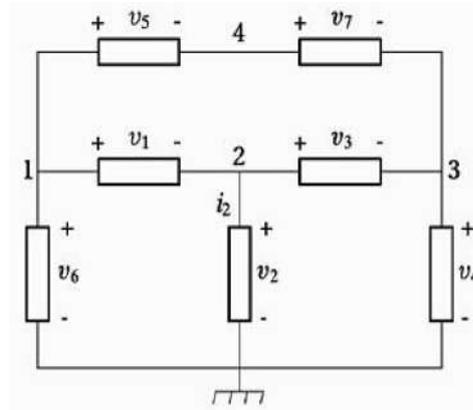
$$\begin{bmatrix} +1 & 0 & 0 & 0 & +1 & +1 & 0 \\ -1 & +1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & +1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & +1 \end{bmatrix} \cdot \begin{bmatrix} i1 \\ i2 \\ i3 \\ i4 \\ i5 \\ i6 \\ i7 \end{bmatrix} = 0 \quad (3)$$

2.3 Lei de Kirchhoff das Tensões – LKT

Assim como as tensões entre os terminais dos elementos do circuito, também devemos destacar as tensões dos nós, definindo um nó de referência com tensão nula, pode-se calcular a tensão de um bipolo como a diferença da tensão no terminal positivo do dispositivo e a tensão do nó onde o terminal negativo está conectado (BURIAN JÚNIOR, 2006).

Deve-se destacar que a soma algébrica das elevações e quedas de tensão de uma malha fechada deve ser igual a zero. Assim para o circuito da Figura 2, temos as equações características da tensão para cada bipolo, como demonstrado na Equação 4 (BOYLESTAD, 2004).

FIGURA 2 – DIAGRAMA PARA LEI DAS TENSÕES



Fonte: Burian Júnior (2006)

$$\begin{aligned}
 v_1 &= e_1 - e_2 \\
 v_2 &= e_2 \\
 v_3 &= e_2 - e_3 \\
 v_4 &= e_3 \\
 v_5 &= e_1 - e_4 \\
 v_6 &= e_1 \\
 v_7 &= e_4 - e_3
 \end{aligned} \tag{4}$$

Semelhante a lei das correntes, podemos expressar as tensões de um bipolo de forma matricial, agrupando-as nos vetores v e e , onde A^T é a matriz de incidência transposta, na Equação 6 mostra-se a expressão matricial das tensões referente ao circuito da Figura 1 e 2, de acordo com a Equação 5 (BURIAN JÚNIOR, 2006).

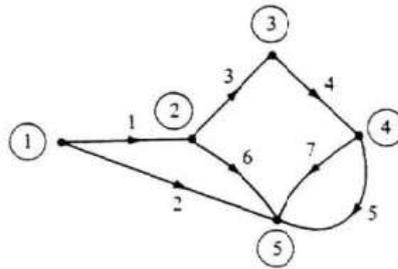
$$v = A^T \cdot e \tag{5}$$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} = \begin{bmatrix} +1 & -1 & 0 & 0 \\ 0 & +1 & 0 & 0 \\ 0 & +1 & -1 & 0 \\ 0 & 0 & +1 & 0 \\ +1 & 0 & 0 & -1 \\ +1 & 0 & 0 & 0 \\ 0 & 0 & -1 & +1 \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} \tag{6}$$

2.4 Dígrafos

As propriedades de interconexão de um circuito podem ser demonstradas por meio de um gráfico do circuito que suprime as informações dos componentes, mas mantém as informações dos ramos e sentidos das correntes. Ao realizar esse procedimento para todos os elementos do circuito, obtemos um dígrafo contendo todas as informações necessárias para análise e formulação das equações LKC e LKT. Na Figura 3, é apresentado um exemplo de dígrafo com 5 nós (n) e 7 ramos (b) (CHUA; DESOER; KUH, 1987).

FIGURA 3 - DÍGRAFO DE N = 5 E B = 7



Fonte: Chua, Desoer e Kuh (1987)

2.5 Equação de Tableau

Este método consiste em escrever todas as equações elaboradas pelas LKC e LKT, para circuitos lineares independentes e as equações dos ramos, assim todos os três vetores e , v e i são representados como variáveis.

Para formulação da equação de Tableau, Chua, Desoer e Kuh (1987) descrevem as etapas do algoritmo:

1. Desenhar o dígrafo do circuito, escolher um nó de referência arbitrário e formular a matriz de incidência A (excluindo-se o nó de referência).
2. Escrever um conjunto completo de equações LKC linearmente independentes, como demonstrado na Equação 7.

$$A \cdot i(t) = 0 \quad (7)$$

3. Escrever um conjunto completo de equações LKT linearmente independentes, como demonstrado na Equação 8.

$$v(t) - A^T \cdot e(t) = 0 \quad (8)$$

4. Escrever as equações dos ramos. Como o circuito é linear, essas equações sempre podem ser reformuladas na forma da Equação 9.

$$M(t) \cdot v(t) + N(t) \cdot i(t) = u_s(t) \quad (9)$$

Considerando que o dígrafo possui n nós e b ramos, a matriz de Tableau terá as dimensões de $n-1$, b e b , respectivamente, como pode ser visto na Equação 10. Como os

vetores e , v e i também contêm $n-1$, b e b variáveis a equação de Tableau sempre consistirá de $(n-1) + 2b$ equações lineares e $(n-1) + 2b$ variáveis. A Equação 11 demonstra a expressão matemática de Tableau.

$$\begin{matrix} & \overbrace{\hspace{1.5cm}}^{n-1} & \overbrace{\hspace{1.5cm}}^b & \overbrace{\hspace{1.5cm}}^b \\ n-1 & \left\{ \begin{array}{ccc} \mathbf{0} & \mathbf{0} & A \\ -A^T & 1 & \mathbf{0} \\ \mathbf{0} & M(t) & N(t) \end{array} \right. & & \end{matrix} \quad (10)$$

$$\begin{matrix} & T(t) \cdot w(t) = u(t) \\ \left[\begin{array}{ccc} 0 & 0 & A \\ -A^T & 1 & 0 \\ 0 & M(t) & N(t) \end{array} \right] \cdot \underbrace{\begin{bmatrix} e(t) \\ v(t) \\ i(t) \end{bmatrix}}_{w(t)} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ u_s(t) \end{bmatrix}}_{u(t)} & & \end{matrix} \quad (11)$$

2.6 Trabalhos Relacionados

Pankeda et al. (2021) tem como objetivo fornecer uma solução de Realidade Virtual (RV) para o ensino e aprendizado de ciências para alunos do ensino médio, desenvolvendo um aplicativo *Android* no motor de jogos *Unity* para aprendizado dos fundamentos da 1ª Lei de Ohm, mostrando potencial na quebra de barreiras onde os alunos não possuem equipamentos laboratoriais, mas necessitam da realização dos experimentos, podendo ser executado através de um *smartphone* e um óculos de RV.

Daineko et al. (2021) demonstra em sua pesquisa o desenvolvimento do laboratório virtual no motor de jogos *Unity* que permite a descoberta e experiências de um Engenheiro de Rádio. Passando por uma revisão da literatura, indicação de ferramentas de programação, as características da *Unity* e o processo de desenvolvimento.

Tanaka, De Almeida e Gouveia (2021) tem como objetivo desenvolver um simulador de realidade virtual para treinamento de eletricitistas de rede de distribuição, oferecendo situações realistas, muitas vezes não vistas em treinamento convencional. Após avaliação optou-se por um ambiente *multiplayer* dispendo de ambiente realista na capacitação em poda de árvore especializada. Conclui-se que o simulador possui potencial para aprimoramento da qualidade dos eletricitistas de rede.

O projeto de Halker e Vantanabe (2020) tem como objetivo o desenvolvimento de um simulador de máquinas elétricas, a fim de disponibilizar aos discentes treino e repetição, além

de trazer maior segurança aos envolvidos nas práticas. Para isso são utilizados o *Blender* e o *Unity*, estes sendo escolhidos por serem gratuitos e por possuir um amplo material de apoio na internet. Conclui-se que os objetivos foram alcançados e o teste de conceito foi cumprido para o curso de Engenharia de Instrumentação, Automação e Robótica.

Lima (2020) desenvolve em seu trabalho um simulador virtual para o ensino de análises e caracterização de materiais na área da construção civil, voltado para o técnico em edificações, inicialmente aplicando um questionário a 5 professores e 113 alunos para diagnóstico e posterior desenvolvimento dos modelos 3D no *Sketup* e a montagem do ambiente no motor de jogos *Unity*. Após este período, o produto foi submetido a 35 pessoas durante exposição e eventos, somado a questionário avaliativo. Concluindo que o projeto alcançou seus objetivos no ensino e aprendizagem do discente através da aplicação desenvolvida.

Para o ensino de física Daineko et al. (2018) desenvolve o laboratório virtual, contendo quatro experimentos, com interfaces amigáveis e replicando a funcionalidade e requisitos dos experimentos reais, permitindo a exportação dos resultados alcançados. Foi utilizado o *Blender* para modelagem 3D, a linguagem C# para reprodução dos comportamentos e o *Unity* para desenvolvimento do programa. Mostrando que os laboratórios virtuais educacionais são um bom exemplo de aplicação das tecnologias da informação na formação discente.

Zhu et al. (2018) apresenta neste trabalho a nova versão do *Virtual Lab*, responsável pela simulação do Controlador Lógico Programável (CLP) inicialmente desenvolvido no *Microsoft XNA Game Studio* e podendo ser utilizado apenas componentes mecânicos e elétricos, foi atualizado e ampliado no *Unity* suportando uma maior variedade de componentes e inclusão dos dispositivos pneumáticos. A nova versão desenvolvida também permite a ação de arrastar e soltar objetos 3D e mais opções de câmeras e interface mais amigável. Em pesquisa realizada com alunos, após utilização em curso, ficou evidenciado a eficácia do programa e a necessidade de melhoria da interface.

Uma outra aplicação voltada para o ensino de física foi demonstrada por González et al. (2017), o objetivo deste trabalho é a apresentação de um novo laboratório virtual 2D e 3D, baseado no *Unity*, assim ampliando as opções de ferramentas para o estudo dos fenômenos físicos com 17 experimentos entre eles o estudo das Leis de Kirchhoff, da resistividade e da 2ª Lei de Newton.

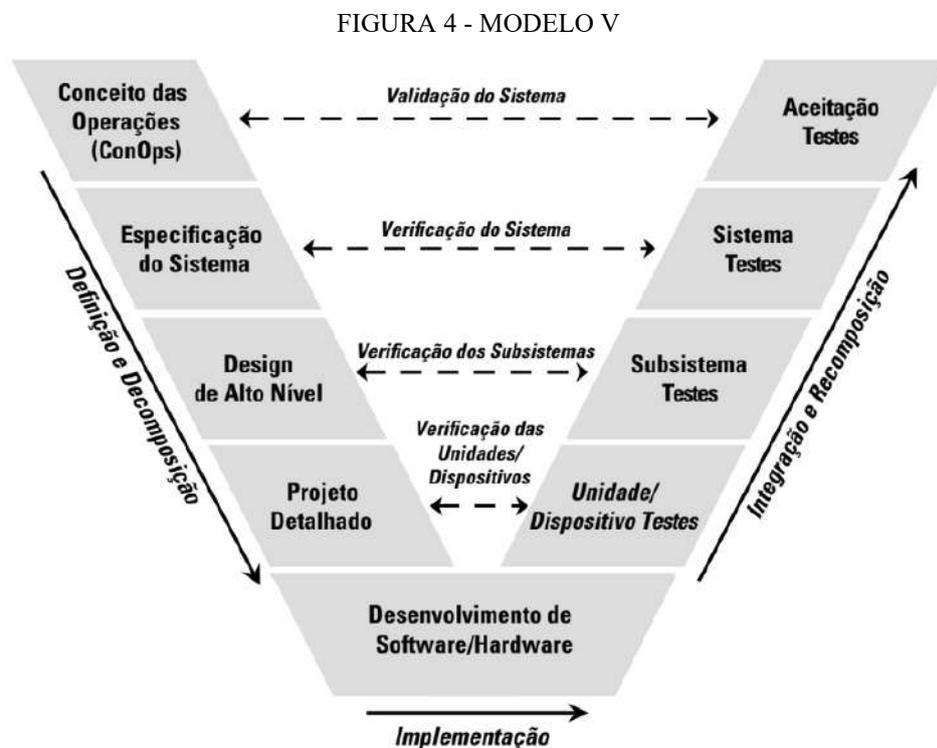
Cnop (2016) desenvolve ambiente de RV para visita técnica no reator Argonauta, sendo um ambiente controlado, assim permitindo livre exploração e esclarecimento das fases

que compõem a segurança, produção e controle de um reator nuclear, utilizando o *Unity* como motor de jogos e evidenciando as possibilidades da virtualização de visitas técnicas restritas, devido seu caráter perigoso.

Jing (2016) elabora o Campus virtual utilizando *Unity*, criando cenas e toda a movimentação no ambiente virtual, tendo um ambiente com baixa manutenção. Demonstrando os princípios do *design* de sistema, cenas e praticidade do sistema.

3 METODOLOGIA

De acordo com Shamieh (2011), a utilização de modelos para o desenvolvimento de sistemas pode torná-los mais baratos e economizar tempo de desenvolvimento. Assim, como procedimento metodológico foi utilizado o Modelo em V (*V-Model*), sendo uma variação gráfica do modelo cascata, demonstrando a relação entre as etapas de desenvolvimento e seus respectivos testes, onde do lado esquerdo do V, temos as definições de requisitos e o refinamento progressivo dos problemas e técnicas aplicados na solução. Do lado direito do modelo é representado as ações de garantia da qualidade que validará os processos realizados do lado esquerdo do modelo, como pode ser visualizado na Figura 4 (PRESSMAN, 2011).



Nos parágrafos seguintes serão descritas as etapas do modelo V segundo Shamieh (2011), partindo da etapa de Conceito das Operações (ConOps), onde é realizado a identificação das principais necessidades dos interessados, as funções, responsabilidades do projeto, seu conceito e para validação do sistema é definido as medidas de desempenho do projeto.

Na sequência temos a etapa de Especificação do Sistema, cabendo a esta a descrição dos requisitos verificáveis definidos na etapa anterior.

Continuando descendo a lado esquerdo do modelo, temos a etapa do Design de Alto Nível, onde será modelada a arquitetura de alto nível do sistema, observando para os requisitos, manutenção, atualização e integração dos demais sistemas.

O Projeto Detalhado é a etapa subsequente, sendo responsável pelo desenvolvimento e detalhamento dos requisitos dos componentes do sistema.

A base do modelo é a etapa de Desenvolvimento de Software, onde será realizado a busca e definição da tecnologia apropriada para o sistema em desenvolvimento, como também, o desenvolvimento da codificação de acordo com as especificações do projeto.

Seguindo para o lado direito do Modelo V temos o Teste de Unidade/Dispositivo, onde será realizado o teste de cada unidade que compõe o sistema.

Ascendendo pelo lado direito, temos a etapa de Teste do Subsistema, assim realizando a verificação de cada subsistema de acordo com os requisitos de alto nível e quando necessário a verificação de integração do hardware com o software.

No Teste de Sistema é verificado a integração dos subsistemas, também sendo testado todo o sistema tendo como parâmetro os requisitos estabelecidos. Outro ponto a ser verificado nesta etapa é a adequação de todas as interfaces.

Fechando as etapas de verificação está o Teste de Aceitação, onde será verificado se a aplicação atende os requisitos estabelecidos e se as metas foram atingidas.

No capítulo 4 será demonstrada a aplicação do modelo V, sendo descrito suas etapas de desenvolvimento associadas aos resultados gerados.

4 DESENVOLVIMENTO E RESULTADOS

4.1 Conceito das Operações

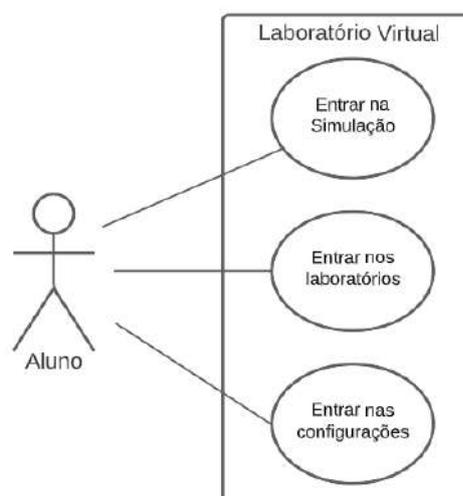
Para alcançar os objetivos propostos o simulador deve ser um ambiente virtual que possibilite a utilização dos discentes e docentes a qualquer momento através das ferramentas digitais como computadores pessoais e dispositivos móveis (*smartphones*), trazendo uma familiaridade com os ambientes físicos utilizados na instituição permitindo a inclusão e adaptação de novas simulações. Outro ponto a se ressaltar é a possibilidade da repetição para maior assimilação dos conceitos vistos em aula, ocorrendo *feedbacks* do sistema tanto no indicativo da correta ligação, como também na impossibilidade de efetuar a conexão, assim condicionando o usuário a não realização em um momento presencial no laboratório físico.

Para tornar o ambiente amigável, deve possuir sonorização de fundo e todos os efeitos necessários para maior imersão do usuário como: tema sonoro, sons dos dispositivos elétricos, sons de movimentação do personagem e sonorização dos *feedbacks*. Também deve ser possível a configuração e os ajustes de volume dos áudios inseridos no sistema.

Destaca-se que a aplicação deve possibilitar a migração para novas tecnologias, como a realidade virtual e o metaverso, onde este vem se tornando uma grande revolução na forma de lidar com os ambientes virtuais e trazem uma vivência mais imersiva.

Dentro desta perspectiva foram divididos os casos de uso demonstrado na Figura 5 e as descrições dos casos podem ser visualizados no [Apêndice A](#).

FIGURA 5 - DIAGRAMA DE CASO DE USO



Fonte: Elaboração própria (2023)

Como proposta de modelo de ambiente, foram selecionados 6 experimentos realizado e demonstrado na bancada física para aplicação no ambiente digitalizado, sendo elas:

- 1 – Interruptor Simples;
- 2 – Campainha;
- 3 – Cigarra;
- 4 – Interruptor Paralelo;
- 5 – Interruptor Intermediário;
- 6 – Tomada 2P + T.

O interruptor simples foi selecionado pelo fato de ser o ponto de partida da disciplina, sendo a primeira prática realizada no laboratório e conseqüentemente, a primeira demonstração na bancada didática, cabendo a esta ligação a introdução dos elementos que compõem um circuito residencial, padronização das cores de identificação dos condutores e critérios de segurança na instalação dos dispositivos.

A campainha e a cigarra foram selecionadas para demonstração da simulação dos dispositivos que possuem como carga um elemento diferente da lâmpada, onde seu circuito de ligação se assemelha ao de um interruptor simples, porém ocorrendo o acionamento de uma saída sonora. Outra diferença é o mecanismo de comando da carga, sendo utilizado um contato sem retenção no lugar de um com retenção.

As práticas do interruptor paralelo e intermediário se relacionam com a sequência didática da disciplina e são circuitos mais complexos para análise, devido a grande quantidade de possibilidades de conexões, assim demonstrando a efetividade do método de análise proposto.

Outro ponto necessário de demonstração é a utilização de instrumentos de medição e a realização de conexões em derivação pelo ambiente virtualizado, assim, a prática da tomada foi selecionada devido a utilização destes dispositivos para verificação do funcionamento da ligação e realização da conexão deste em paralelo com o circuito.

4.2 Especificação do Sistema

A elicitação de requisitos é parte fundamental na construção de um sistema e tem como objetivo a descoberta das necessidades que o programa deve atender, assim mitigando os erros de entrega (POHL; RUPP, 2015).

De acordo com o proposto no ConOps os requisitos aplicados foram divididos em requisitos funcionais e não funcionais, sendo definidos e demonstrados nos itens seguintes.

4.2.1 Requisitos funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, seu comportamento fornecido por uma função do sistema (POHL; RUPP, 2015; SOMMERVILLE, 2011, VAZQUEZ; SIMÕES, 2016).

No Quadro 1 mostram-se os 12 requisitos elencados com sua respectiva descrição e o caso de uso a que se relaciona.

QUADRO 1 - REQUISITOS FUNCIONAIS (CONTINUA)

Id	Requisito	Descrição	Caso de Uso
RF01	Ambiente em primeira pessoa	O sistema deve permitir a circulação e movimentação do aluno com a visão do personagem em 1ª pessoa, assim o discente deverá ter uma perspectiva gráfica do ponto de vista do personagem, sem visualização do próprio corpo.	Entrar na simulação.
RF02	Ajuste de volume da sonorização	O sistema deve permitir ao usuário o ajuste dos volumes dos áudios contidos em cada ambiente.	Entrar na simulação e Entrar nas configurações.
RF03	Visualização das funcionalidades dos botões de movimentação	O sistema deve permitir ao usuário a visualização das funcionalidades dos botões de movimentação e interação com o ambiente virtual.	Entrar na simulação e Entrar nas configurações.
RF04	Indicação dos laboratórios	O sistema deve indicar de forma interativa o nome de cada laboratório.	Entrar na simulação.
RF05	Direcionamento de ações	O sistema deve indicar de forma interativa as ações necessárias para continuidade da simulação.	Entrar na simulação.
RF06	Seleção de prática	O sistema deve permitir a seleção da prática a ser realizada através de menu.	Entrar na simulação.
RF07	Escolha do laboratório	O sistema deve permitir que o usuário possa ser direcionado diretamente ao laboratório desejado.	Entrar nos laboratórios.
RF08	Reiniciar simulação da prática	O sistema deve permitir ao usuário o reinício da simulação da prática.	Entrar na simulação.

QUADRO 1 - REQUISITOS FUNCIONAIS (CONCLUSÃO)

Id	Requisito	Descrição	Caso de Uso
RF09	Identificação das partes passíveis de interação	O sistema deve identificar os objetos passíveis de manipulação por parte do usuário em cada prática realizada.	Entrar na simulação.
RF10	Seleção de cor	O sistema deve permitir que o usuário possa selecionar a cor de cada conexão (cabo com pino banana) inserida na prática.	Entrar na simulação.
RF11	Verificação do circuito	O sistema deve permitir que o usuário verifique se o circuito montado está de acordo com a norma técnica em vigor.	Entrar na simulação.
RF12	Indicação de defeito	O sistema deve informar de forma interativa a ocorrência de um curto-circuito na ligação montada na bancada.	Entrar na simulação.

Fonte: Elaboração própria (2023)

4.2.2 Requisitos não funcionais

Os requisitos não funcionais são os que estão diretamente ligados aos serviços oferecidos pelo sistema, estando relacionado a aspectos de ambiente, organização, implementação e qualidade destes serviços ofertados ao usuário, descrevendo limitações gerais complementando a especificação do *software* (POHL; RUPP, 2015; SOMMERVILLE, 2011, VAZQUEZ; SIMÕES, 2016).

No Quadro 2 mostram-se os 5 requisitos elencados com sua respectiva descrição.

QUADRO 2 - REQUISITOS NÃO FUNCIONAIS (CONTINUA)

Id	Requisito	Descrição
RNF01	Multiplataforma	<p>O sistema deve possibilitar a instalação em dispositivos móveis (<i>smartphones</i>) baseados no SO <i>Android</i>, assim cobrindo a maior parte dos alunos vinculados à instituição.</p> <p>Também deve permitir o acesso através de navegador <i>web online</i> através de computadores, assim trazendo mais um recurso de utilização sem necessidade de instalação e conseqüentemente ocupação de espaço em memória do dispositivo.</p> <p>Para situações em que há o acesso a computadores, porém sem conexão com a internet, o sistema deve permitir a instalação no SO <i>Windows</i>.</p>

QUADRO 2 - REQUISITOS NÃO FUNCIONAIS (CONCLUSÃO)

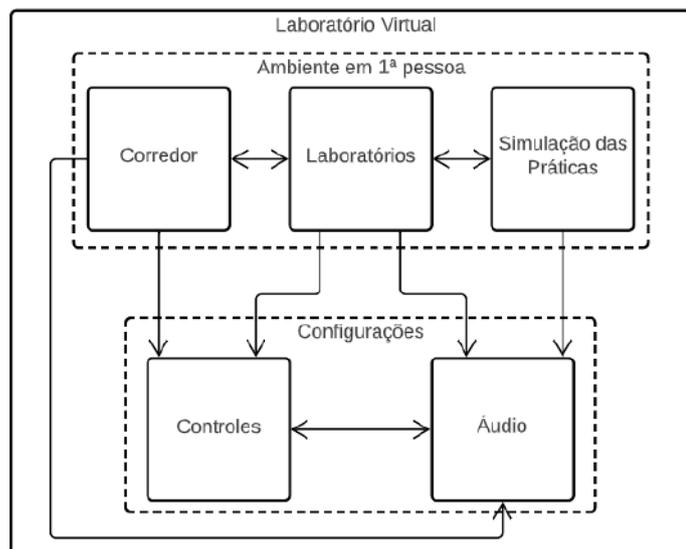
Id	Requisito	Descrição
RNF02	Modularidade	Para facilitar futuras manutenções e atualizações o sistema deve ser dividido em cenas, devendo cada laboratório, corredor de circulação, e práticas ser um módulo individualizado.
RNF03	Replicar setor de elétrica	Sendo o público-alvo da aplicação os alunos do IFF, o sistema deve trazer de forma virtualizada os ambientes que compõem o setor de elétrica, respeitando as proporções e detalhes arquitetônicos, para um deslocamento verossímilhante com o ambiente físico.
RNF04	Sem custo para instituição	O projeto deverá utilizar em seu desenvolvimento, programas de plataforma aberta ou em sua versão gratuita, assim não demandando aquisição de <i>softwares</i> para a elaboração da aplicação e consequentemente delimitando os custos aos insumos necessários para utilização destes, como: mão de obra, energia elétrica e <i>internet</i> .
RNF05	Adaptação a novas tecnologias	A possibilidade de migração para novas plataformas deve ser observada para futuras versões, assim a utilização de um programa que traz todas essas possibilidades incorporadas ou através de instalação de complementos de RV e o modo <i>multiplayer</i> deve ser observado na escolha do programa construtor.

Fonte: Elaboração própria (2023)

4.3 Design de Alto Nível

Desta forma o protótipo foi dividido em sistemas em primeira pessoa e sistemas de configuração, como pode ser visto na Figura 6.

FIGURA 6 - DESIGN DE ALTO NÍVEL



Fonte: Elaboração própria (2023)

No subsistema *Corredor*, o usuário poderá circular pelo setor onde se concentra os laboratórios da área elétrica, cabendo a este ambiente o papel de introduzir o setor para novos alunos, podendo ser utilizado como ambiente de visita guiada e utilizado na recepção de novos discentes do curso, mas também a visitantes. Deste subsistema será possível acessar os laboratórios de forma similar ao ambiente real.

No subsistema *Laboratórios* é possível acessar diretamente a cena do ambiente laboratorial pretendido, assim trazendo agilidade para alunos que já conhecem o ambiente e apenas pretendem acessar os laboratórios para praticar os circuitos.

No subsistema *Simulação das Práticas*, será possível realizar a montagem do circuito, após acessar a cena com o laboratório de Instalações Elétricas e ativar o menu suspenso ao clicar ou tocar na bancada, sendo o principal módulo da aplicação.

No sistema *Configurações* encontram-se os subsistemas necessários para o ajuste de volume dos áudios inseridos e um *menu* com a indicação das funções dos botões de controle do personagem e interação com o ambiente.

4.4 Projeto Detalhado

Neste tópico será detalhado os *softwares* utilizados para o desenvolvimento do projeto, assim como sua arquitetura.

4.4.1 Ferramentas computacionais

4.4.1.1 Blender

Partindo do requisito de custo reduzido na aquisição dos softwares utilizados no desenvolvimento do projeto, o Blender na sua versão 2.93 foi selecionado, sendo uma poderosa ferramenta *open source* de computação gráfica 3D capaz de modelar, renderizar e criar animações de personagem e objetos, podendo ser utilizado a interface de programação de aplicações (*Application Programming Interface – API*) para *scripts Python* na personalização de ferramentas, além da possibilidade de utilização em vários SO, como *Windows*, Linux e Mac (BLAIN, 2021; BLENDER, 2022; BRITO, 2008).

4.4.1.2 Unity

Devido às características necessárias para desenvolvimento do laboratório virtual, o motor de jogos *Unity* foi selecionado e utilizado em sua versão 2020.3.21f1, oferecendo suporte a gráficos 3D, realidade virtual, possibilidade de desenvolvimento multiplataforma e

capacidade de *networking*, assim possibilitando a integração de múltiplos usuários conectados pela *internet* (SILVA FILHO et al., 2018).

Para simulação dos comportamentos específicos de cada aplicação é possível realizar a programação através da linguagem *C#*, criando *scripts* que herdam da classe base *MonoBehavior*, cabendo aos códigos derivarem desta classe.

4.4.1.3 CADe SIMU

Na confecção dos diagramas dos circuitos elétricos foi utilizado o software gratuito CADe SIMU na sua versão 4.0, este destinado a criação de diagramas de comandos elétricos e possibilita o desenvolvimento de qualquer circuito, sendo amplamente utilizado na área de eletrotécnica (CADe SIMU, 2022).

4.4.1.4 Gimp

Devido à necessidade de edição das imagens inseridas no simulador, como a aplicação de canal alfa na simbologia dos componentes elétricos, foi utilizado o programa Gimp na sua versão 2.10.28, também sendo um *software open source* e multiplataforma, permite a manipulação de imagens, desenvolvimento de elementos de *design* gráfico e permitindo alta flexibilidade e extensibilidade devido sua integração com linguagens de programação como o *Python*, assim permitindo uma alta customização das suas funções (GIMP, 2022).

4.4.1.5 Scilab

Na verificação dos valores calculados pelo *script* que realiza a operação da equação de Tableau, foi utilizado o programa *open source* Scilab na versão 6.1.1, permitindo a manipulação matemática de matrizes e outras aplicações usuais de engenharia (SCILAB, 2022).

4.4.1.6 Visual Code Studio

Como ambiente de desenvolvimento integrado (*Integrated Development Environment - IDE*) de programação foi utilizado o *Visual Code Studio* em sua versão 2019, sendo este programa compatível com o motor de jogos selecionado.

4.4.2 Fluxo do programa

Nesta etapa foi definido o fluxo do programa, seguindo os conceitos de fluxo do jogo onde segundo Eng (2015), demonstra desde o início do programa até a finalização dele, dando uma visão geral da aplicação.

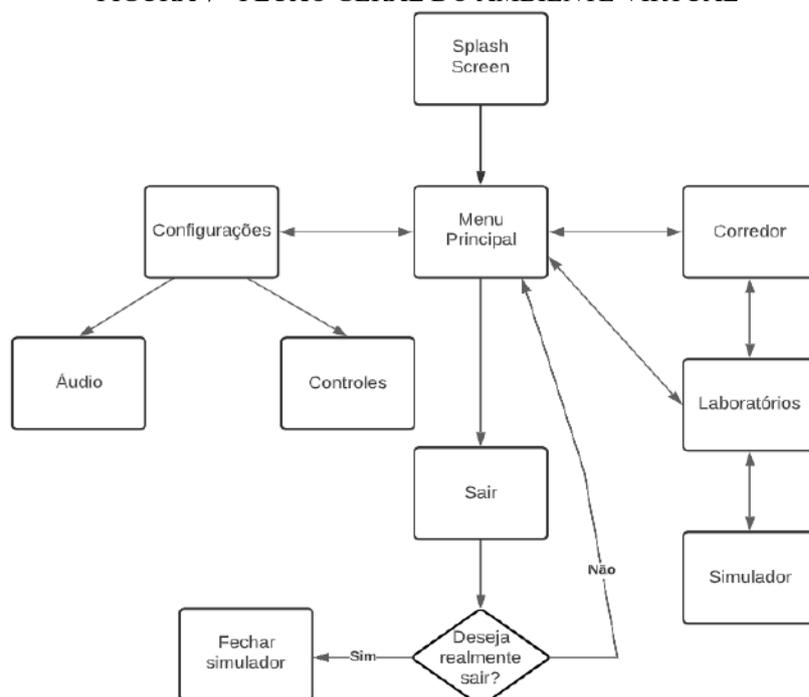
O ponto de partida do fluxo é a tela inicial, comumente denominada de “tela de *splash*”, responsável pela abertura do simulador com as informações do motor de jogos construtor da aplicação e os créditos do desenvolvedor, na sequência o usuário é direcionado ao menu principal, este responsável pelo fornecimento das opções do programa, podendo o usuário ser encaminhado para o corredor do setor de elétrica, para o menu de laboratórios do setor e para as configurações do sistema, também cabendo a esta janela a disponibilização da opção de saída do programa após confirmação da ação.

Ainda no corredor é possível o acesso aos laboratórios, seguindo o fluxo semelhante ao realizado fisicamente no instituto, cabendo a realização das simulações ao laboratório de instalações elétricas.

Nas configurações é possível a visualização dos controles de locomoção do personagem em primeira pessoa, assim como os ajustes dos volumes dos áudios inseridos no sistema.

O fluxo geral do ambiente virtual pode ser visualizado na Figura 7, onde as setas estão indicando o estado de navegação para determinadas condições e as setas bidirecionais a navegação dupla entre estados (ENG, 2015).

FIGURA 7 - FLUXO GERAL DO AMBIENTE VIRTUAL



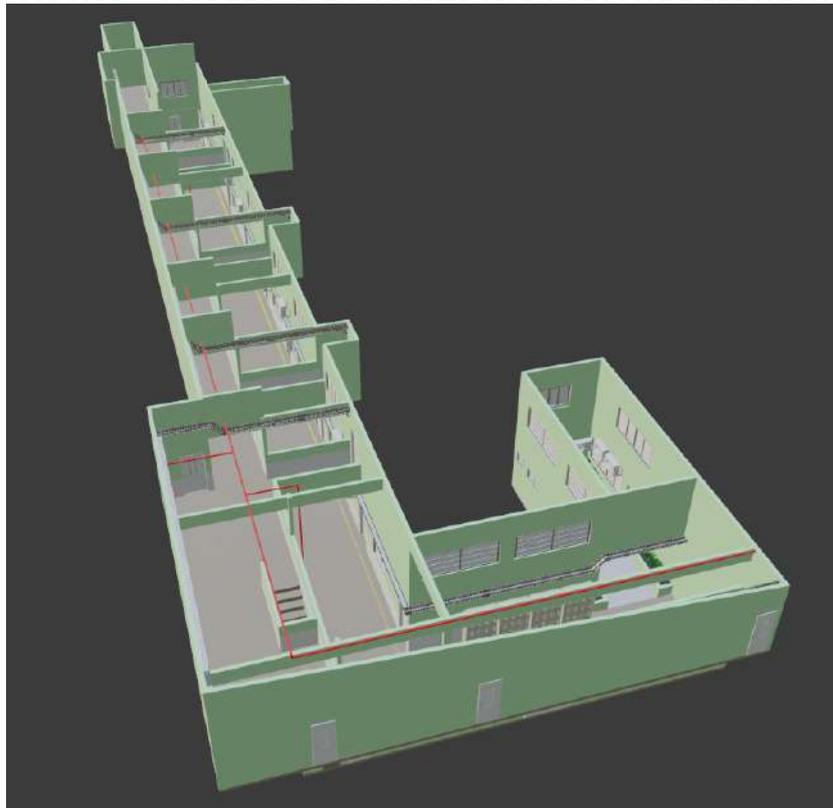
Fonte: Elaboração própria (2023)

4.4.3 Modelagem 3D

A área de circulação dos cursos de eletricidade foi modelada para uso no ambiente virtual, proporcionando familiaridade aos alunos que já conhecem o setor e apresentando-o aos novos alunos. A planta baixa disponibilizada pela Diretoria de Infraestrutura do *campus*, que pode ser visualizada no [Anexo B](#), foi utilizado como base para modelagem.

Na Figura 8 é demonstrado o corredor de circulação do setor, possuindo o andar térreo com acesso a 14 laboratórios e o andar superior com acesso a 5 ambientes. Para esta cena foi realizada a modelagem 3D excluindo-se alguns detalhes estruturais, como a não representação das colunas, para facilitação na movimentação do personagem em primeira pessoa, também houve o alargamento de portas e vãos de passagem, quando necessário, com o mesmo objetivo.

FIGURA 8 - MODELO 3D DO CORREDOR DO SETOR DE ELÉTRICA



Fonte: Elaboração própria (2023)

Um dos locais de maior utilização dos alunos neste ambiente é o jardim de inverno, sendo utilizado para interação social, realização das refeições e aguardo na troca das aulas e contra turnos, este ambiente foi escolhido como ponto de partida da aplicação e pode ser visualizado na Figura 9a o ambiente físico e na Figura 9b o modelo 3D.

FIGURA 9 - JARDIM DE INVERNO



(a) Ambiente físico

(b) Ambiente modelado

Fonte: Elaboração própria (2023)

Na Figura 10a é demonstrado o corredor térreo do setor e seu respectivo modelo 3D é visualizado na Figura 10b. A Figura 10c demonstra o corredor superior do bloco sendo a Figura 10d sua representação tridimensional. Também pode ser visualizado na Figura 10e o detalhe do painel com desenhos ilustrativos contido no setor e sua representação virtualizada na Figura 10f. Sendo estes corredores os pontos de circulação e acesso aos laboratórios.

FIGURA 10 - CORREDOR TÉRREO E SUPERIOR DO SETOR DE ELÉTRICA (CONTINUA)



(a) Foto do corredor térreo

(b) Modelo do corredor térreo



(c) Foto do corredor superior

(d) Modelo do corredor superior

FIGURA 10 - CORREDOR TÉRREO E SUPERIOR DO SETOR DE ELÉTRICA (CONCLUSÃO)



(e) Foto de detalhe do corredor superior

(f) Modelo de detalhe do corredor superior

Fonte: Elaboração própria (2023)

Como alvo do trabalho, o laboratório de instalações elétricas foi modelado de acordo com os detalhes existentes *in loco*, possuindo seis *box* responsável pela simulação física da instalação dos dispositivos de uma edificação alimentada em baixa tensão (BT), sendo dois alimentados com circuito trifásico, dois com circuito bifásico e dois com circuito monofásico, a disposição dos equipamentos e móveis pode ser visualizada na Figura 11a e na Figura 11b o modelo 3D do laboratório.

FIGURA 11 - LABORATÓRIO DE IEBT



(a) Foto do ambiente

(b) Modelo do ambiente

Fonte: Elaboração própria (2023)

Compondo a infraestrutura didática do ambiente está a bancada desenvolvida e montada para demonstração dos circuitos de ligação dos dispositivos ensinado em aula, contando com 28 componentes, dentre eles interruptores, campainha, instrumentos de medição, dispositivos de proteção, ventilador de teto, sensor de presença e chave boia, o detalhamento dos dispositivos com seus respectivos espaçamentos e localização pode ser visualizado no [Apêndice B](#).

Seguindo as dimensões e espaçamentos dos componentes, foi realizado a modelagem 3D da bancada base e inclusão dos objetos previamente baixados de repositórios gratuitos como o GrabCad *Community*, Sketchfab, 3D *ContentCentral*[®] e o *bimobject*[®], sendo um recurso importante de difusão e compartilhamento dos modelos 3D pela comunidade. Cabe ressaltar

que a disposição dos soquetes e cores do objetos foram ajustadas em relação ao modelo físico para melhor operação no ambiente virtualizado, na Figura 12a é demonstrado a bancada física utilizada em laboratório e o modelo 3D da bancada é visualizado na Figura 12b.

FIGURA 12 - MODELO 3D DA BANCADA DIDÁTICA



(a) Foto da bancada didática



(b) Modelo da bancada didática

Fonte: Elaboração própria (2023)

A definição do comprimento do modelo dos cabos condutores do tipo pino banana levou em consideração as distâncias dos soquetes dos demais modelos (interruptores, lâmpadas, etc), de forma a prover uma experiência mais orgânica para o usuário no manuseio e visualização dos cabos. Para traçado do modelo, foi utilizado o objeto *Bézier*, onde posteriormente no modo de edição do *Blender* são excluídos os vértices do objeto e realizado o redesenho do traçado do condutor de acordo com o formato desejado. Para aumentar o

diâmetro do modelo elaborado, foi alterado o valor da propriedade profundidade do objeto para 0,0016 m, assim trazendo semelhança com o condutor utilizado em laboratório, também ocorrendo a inclusão do material de cor preta. O resultado da modelagem dos cabos podem ser visualizados nas seguintes figuras: *Interruptor Simples* (Figura 13a), *Campainha* (Figura 13b), *Cigarra* (Figura 13c), *Interruptor Paralelo* (Figura 13d), *Interruptor Intermediário* (Figura 13e) e da *Tomada 2P + T* (Figura 13f).

FIGURA 13 - MODELO 3D DAS CONEXÕES



(a) Interruptor Simples



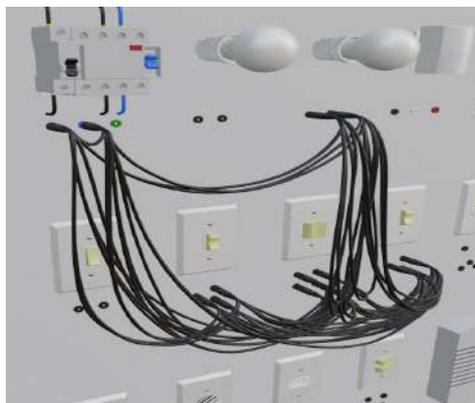
(b) Campainha



(c) Cigarra



(d) Interruptor Paralelo



(e) Interruptor Intermediário



(f) Tomada 2P+T

Fonte: Elaboração própria (2023)

Com todos os modelos construídos foi realizada a exportação dos objetos no formato *Filmbbox* (FBX), sendo este compatível com o motor de jogos *Unity*.

4.4.4 Modelagem elétrica e equacionamentos

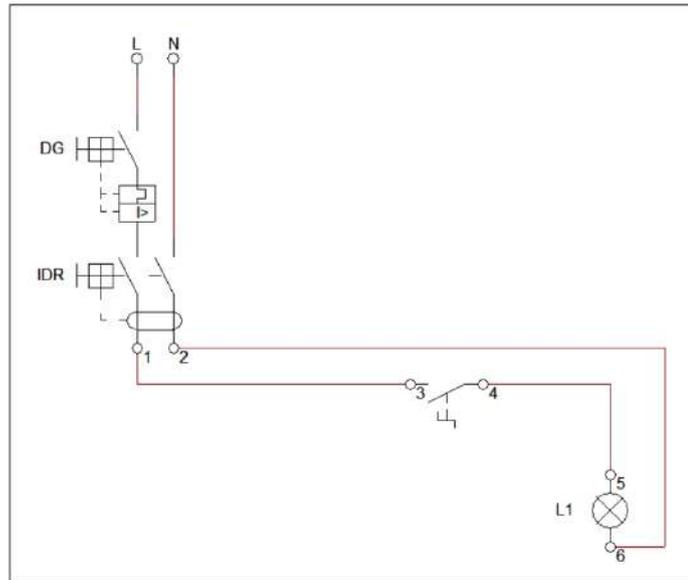
Com a montagem e modelagem 3D dos objetos e ambientes é necessário a realização e implementação da análise do circuito que será simulado pelo discente, para esta tarefa foi escolhido o método de análise de circuitos nomeado de Equação de Tableau, cabendo inicialmente a modelagem elétrica do circuito, construção de seu dígrafo e aplicação das equações características de *Kirchhoff* para cálculo das correntes e tensões dos ramos e nós do circuito alvo.

Desta forma as práticas foram modeladas de acordo com circuitos alimentados por fonte independente de corrente contínua, os interruptores como resistores variáveis e as cargas como resistores fixos. Como o alvo da análise são as correntes e tensões dos ramos e nós, não é necessária uma modelagem voltada para corrente alternada, sendo este o sinal de alimentação de uma IEBT.

4.4.4.1 Diagrama dos circuitos modelados

a) *Interruptor Simples*: Para o circuito é realizado a construção do diagrama multifilar, partindo da fonte de alimentação tendo seus terminais numerados em ordem crescente, cabendo aos terminais 1 e 2 o fornecimento da tensão de alimentação em 127 volts (V) entre os condutores fase e neutro, de acordo com a tensão de alimentação de nossa região. Os terminais 3 e 4 compõem os pontos de conexão do interruptor simples, estes sendo uma chave com retenção, de acordo com Franchi (2008) é um dispositivo que quando acionado só retorna a posição inicial após um novo acionamento, os terminais 5 e 6 são os pontos de conexão da lâmpada, como pode ser visualizado na Figura 14.

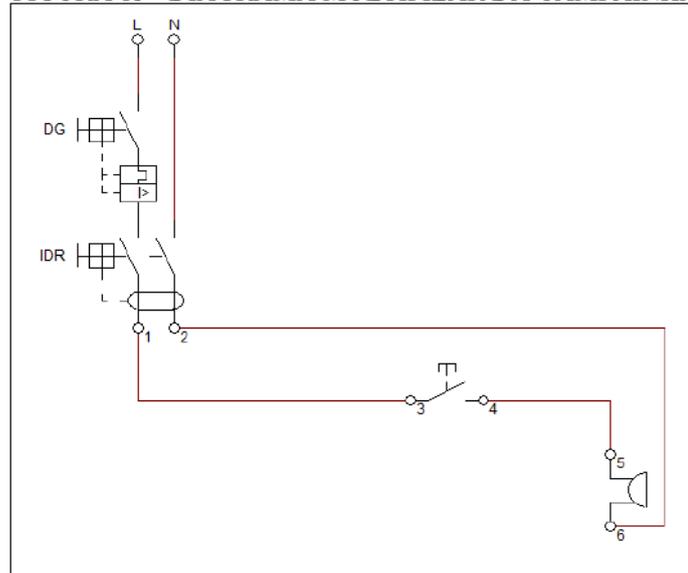
FIGURA 14 - DIAGRAMA MULTIFILAR DO INTERRUPTOR SIMPLES



Fonte: Elaboração própria (2023)

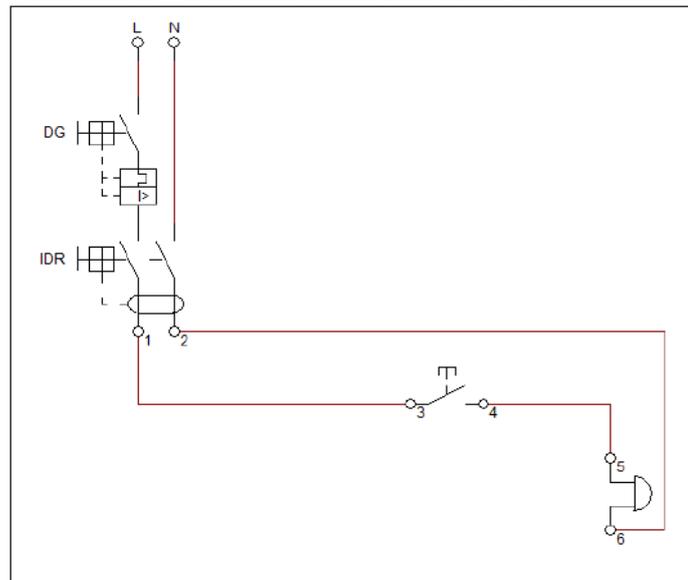
b) *Campainha e Cigarra*: Para os circuitos multifilares, visto nas Figuras 15 e 16, a alimentação é mantida de acordo com a Figura 14, onde os terminais 3 e 4 representam os pontos de conexão do pulsador normalmente aberto (NA), onde segundo Franchi (2008), são contatos sem retenção, assim só permanecendo fechado durante aplicação de força externa e quando cessado, retorna a posição inicial, os terminais 5 e 6 são os pontos de conexão dos dispositivos sonoros.

FIGURA 15 - DIAGRAMA MULTIFILAR DA CAMPAINHA



Fonte: Elaboração própria (2023)

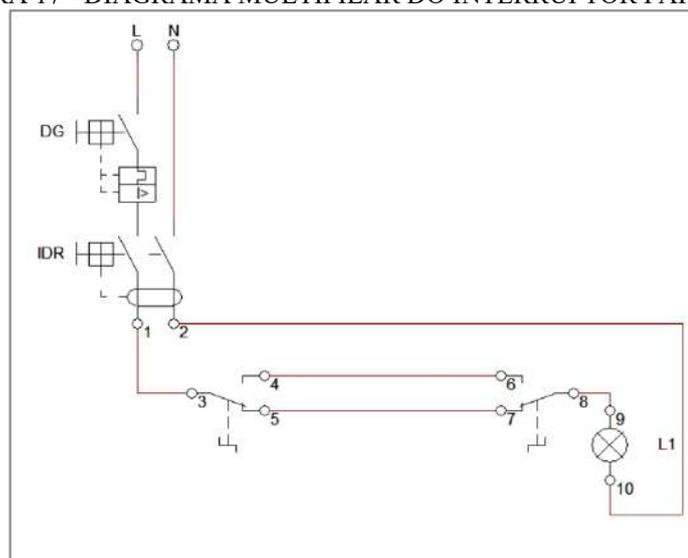
FIGURA 16 - DIAGRAMA MULTIFILAR DA CIGARRA



Fonte: Elaboração própria (2023)

c) *Interruptor Paralelo (Three-way)*: Mantendo-se a alimentação nos terminais 1 e 2, o diagrama multifilar demonstrado na Figura 17, utiliza como interruptor uma chave seletora, onde de acordo com Franchi (2008), possui duas posições com um contato comum, sendo uma chave de função composta e com retenção. Nos terminais 3 (comum), 4 e 5 temos o primeiro dispositivo de comando e os terminais 6, 7 e 8 (comum) compõem os pontos de conexão do segundo dispositivo de chaveamento, sempre atuando em pares para o correto funcionamento. Os terminais 9 e 10 representam os pontos de conexão da lâmpada.

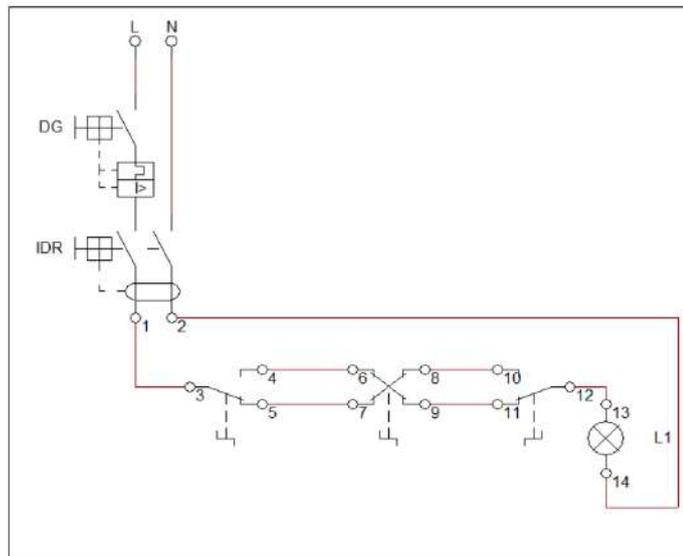
FIGURA 17 - DIAGRAMA MULTIFILAR DO INTERRUPTOR PARALELO



Fonte: Elaboração própria (2023)

d) *Interruptor Intermediário (Four-way)*: Mostra-se na Figura 18 o diagrama multifilar, onde os terminais 6 e 7 são os pontos de conexão da entrada do dispositivo e os terminais 8 e 9 a saída. Este dispositivo deve ser instalado entre dois *Interruptores Paralelos*, representado pelos terminais 3, 4, 5, 10, 11 e 12 e tendo como função a mudança do estado lógico da lâmpada de qualquer ponto acionado (CERVELIN; CAVALIN, 2008). Os terminais 13 e 14 compõem os pontos de conexão da lâmpada.

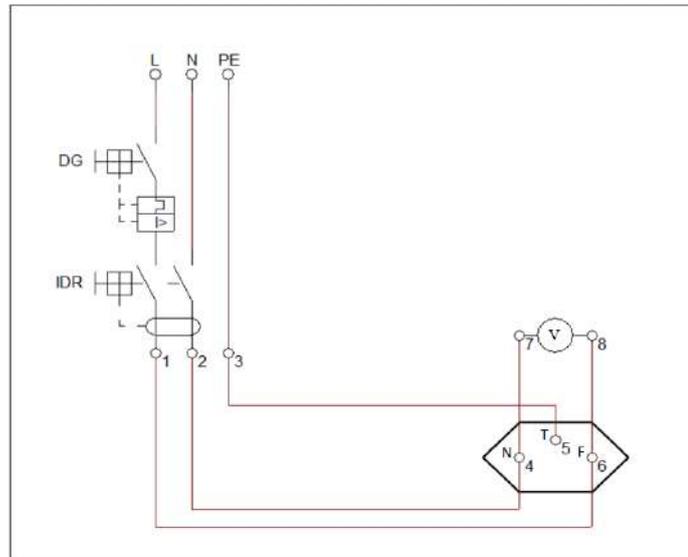
FIGURA 18 - DIAGRAMA MULTIFILAR DO INTERRUPTOR INTERMEDIÁRIO



Fonte: Elaboração própria (2023)

e) *Tomada 2P+T*: Seguindo a NBR14136:2002 – Plugues e tomadas para uso doméstico e análogo até 20 A/250 V em corrente alternada – Padronização, para correta alimentação da tomada é acrescentado ao circuito o terminal de proteção, representado pelo ponto 3 da Figura 19. O terminal 4 é o ponto de interligação do neutro, o ponto 5 fica responsável pelo condutor terra e o terminal 6 a conexão com a fase do circuito, assim encerrando a alimentação do dispositivo. Para verificação da diferença de potencial (ddp) no dispositivo é incluído o voltímetro ao circuito, sendo os terminais 7 e 8 os pontos de conexão do instrumento.

FIGURA 19 - DIAGRAMA MULTIFILAR DA TOMADA 2P + T



Fonte: Elaboração própria (2023)

Todas as simbologias apresentadas nos diagramas podem ser vistos no [Apêndice C](#).

4.4.4.2 Cálculo das possibilidades de conexões para cada circuito modelado

Com a identificação dos terminais passíveis de interação na simulação, foi realizado o cálculo do número de possibilidades de conexões para cada circuito modelado, através do conceito de análise combinatória, demonstrado na Equação 12. Para condicionamento das ações dos discentes na prática em bancada física foi realizado a exclusão das ligações entre os terminais de um mesmo dispositivo (n_e) e posteriormente calculado o número final de conexões a serem modeladas (n_c) e utilizadas no circuito, de acordo com a Equação 13.

$$C_{n,p} = \frac{n!}{p!(n-p)!} \quad (12)$$

$$n_c = C_{n,p} - n_e \quad (13)$$

a) *Interruptor Simples, Campainha e Cigarra*: Para os 6 terminais de conexão utilizados nos circuitos do *Interruptor Simples, Campainha e Cigarra* e a utilização sempre em pares o cálculo da Equação 12 fornece 15 combinações, como pode ser visto na Equação 14.

$$C_{6,2} = \frac{6!}{2!(6-2)!} = 15 \quad (14)$$

Excluindo as interligações entre os terminais da fonte (1 com 2), do interruptor (3 com 4) e do dispositivo de saída (5 com 6) o cálculo da Equação 13 fornece 12 combinações a serem modeladas, como pode ser visto na Equação 15.

$$n_c = 15 - 3 = 12 \quad (15)$$

Na Figura 20 mostram-se todas as 12 combinações de conexões possíveis do *Interruptor Simples, Campainha e Cigarra*.

FIGURA 20 - RELAÇÃO DE CONEXÕES DO INTERRUPTOR SIMPLES, CAMPAINHA E CIGARRA



Fonte: Elaboração própria (2023)

b) *Interruptor Paralelo (Three-way)*: Para 10 terminais de conexão utilizados no circuito do *Interruptor Paralelo* e a utilização sempre em pares o cálculo da Equação 12 fornece 45 combinações, como pode ser visto na Equação 16.

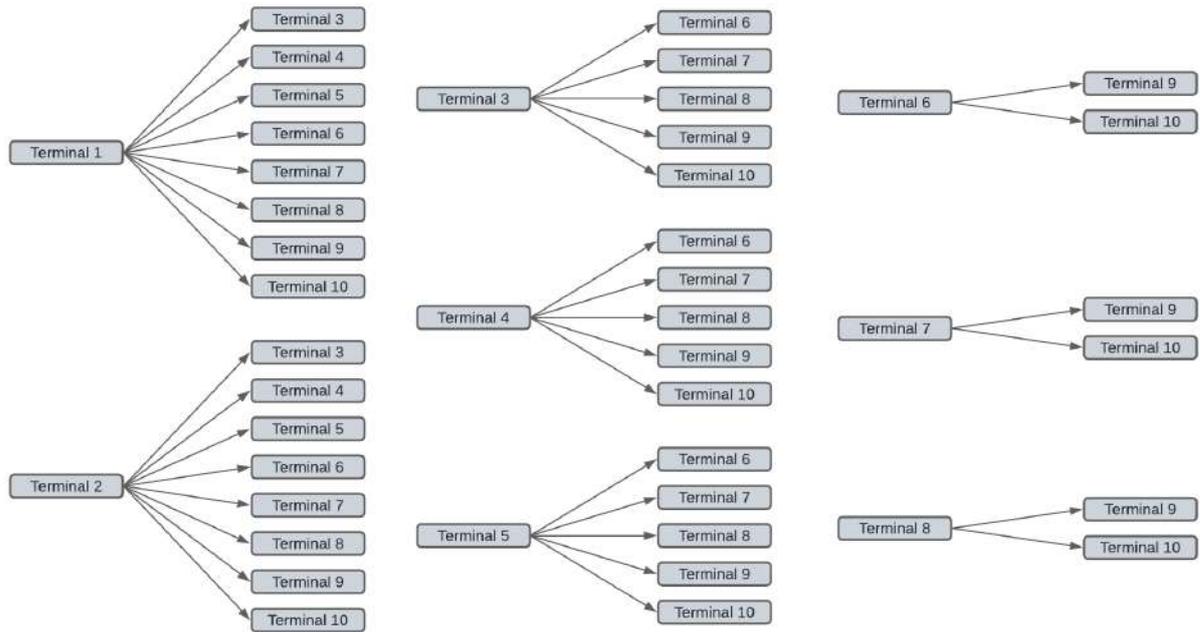
$$C_{10,2} = \frac{10!}{2!(10-2)!} = 45 \quad (16)$$

Excluindo as interligações entre os terminais da fonte (1 com 2), dos interruptores (3 com 4, 3 com 5, 4 com 5, 6 com 7, 6 com 8 e 7 com 8) e da lâmpada (9 com 10) o cálculo da Equação 13 fornece 37 combinações a serem modeladas, como pode ser visto na Equação 17.

$$n_c = 45 - 8 = 37 \quad (17)$$

Na Figura 21 mostram-se todas as 37 combinações de conexões possíveis do *Interruptor Paralelo (Three-way)*.

FIGURA 21 - RELAÇÃO DE CONEXÕES DO INTERRUPTOR PARALELO



Fonte: Elaboração própria (2023)

c) *Interruptor Intermediário (Four-way)*: Para 14 terminais de conexão utilizados no circuito do *Interruptor Intermediário (Four-way)* e a utilização sempre em pares o cálculo da Equação 12 fornece 91 combinações, como pode ser visto na Equação 18.

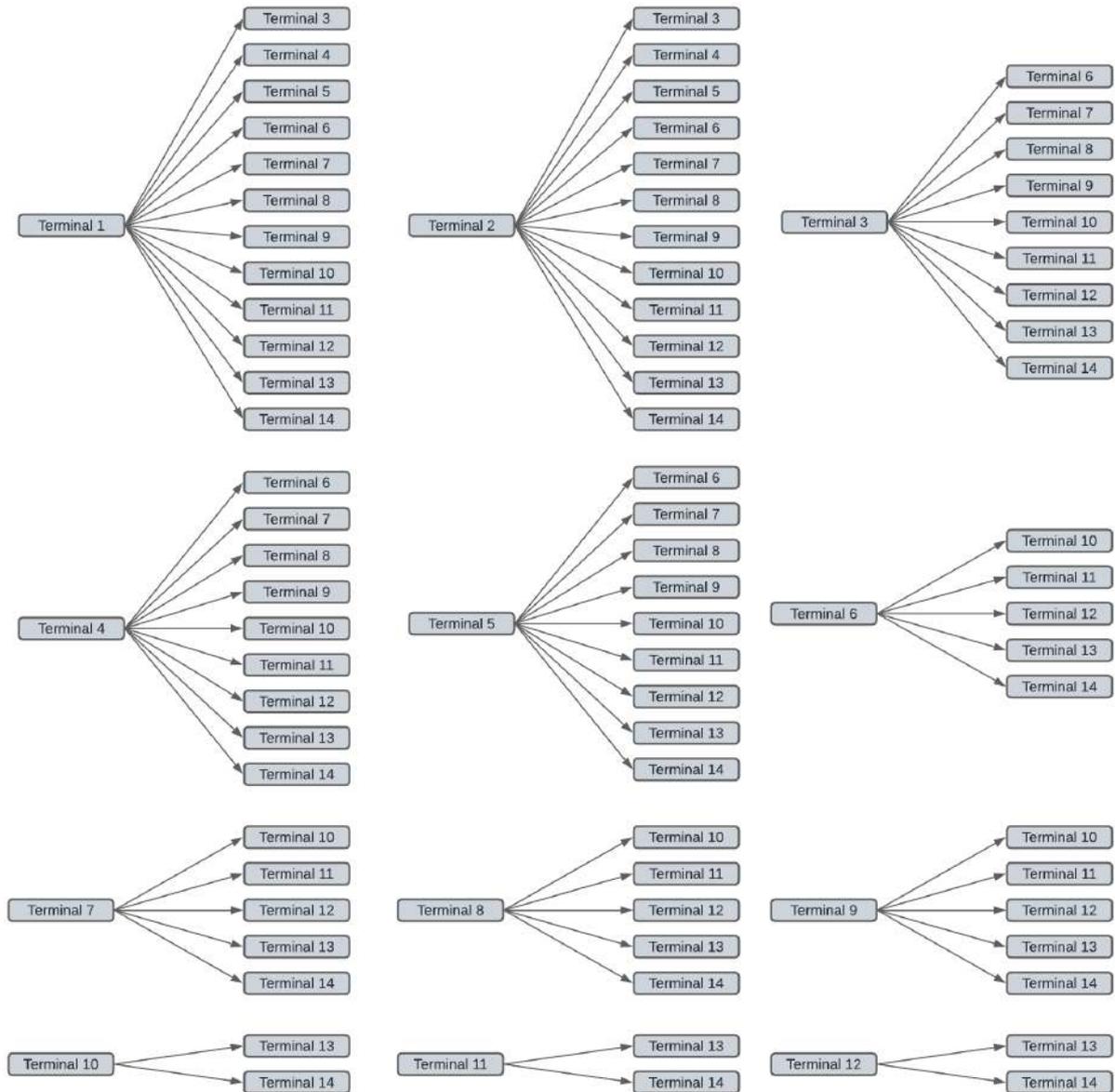
$$C_{14,2} = \frac{14!}{2!(14-2)!} = 91 \quad (18)$$

Excluindo as interligações entre os terminais da fonte (1 com 2), dos interruptores (3 com 4, 3 com 5, 4 com 5, 6 com 7, 6 com 8, 6 com 9, 7 com 8, 7 com 9, 8 com 9, 10 com 11, 10 com 12 e 11 com 12) e da lâmpada (13 com 14) o cálculo da Equação 13 fornece 77 combinações a serem modeladas, como pode ser visto na Equação 19.

$$n_c = 91 - 14 = 77 \quad (19)$$

Na Figura 22 mostram-se todas as 77 combinações de conexões possíveis do *Interruptor Intermediário (Four-way)*.

FIGURA 22 - RELAÇÃO DE CONEXÕES DO INTERRUPTOR INTERMEDIÁRIO



Fonte: Elaboração própria (2023)

d) *Tomada 2P+T*: Para 8 terminais de conexão utilizados no circuito da *Tomada 2P + T* e a utilização sempre em pares o cálculo da Equação 12 fornece 28 combinações, como pode ser visto na Equação 20.

$$C_{8,2} = \frac{8!}{2!(8-2)!} = 28 \quad (20)$$

Excluindo as interligações entre os terminais da fonte (1 com 2, 1 com 3 e 2 com 3), da tomada (4 com 5, 4 com 6 e 5 com 6) e do medidor (7 com 8) e como o objetivo da inclusão do voltímetro é a medição da tensão na tomada, também foi excluído as conexões

com a fonte (7 com 1, 7 com 2, 7 com 3, 8 com 1, 8 com 2 e 8 com 3) e com o ponto de terra na tomada (7 com 4 e 8 com 4) o cálculo da Equação 13 fornece 13 combinações a serem modeladas, como pode ser visto na Equação 21.

$$n_c = 28 - 15 = 13 \quad (21)$$

Na Figura 23 mostram-se todas as 13 combinações de conexões possíveis da Tomada $2P + T$.



Fonte: Elaboração própria (2023)

4.4.4.3 Modelo elétrico

Para análise das configurações possíveis de serem executadas e suas respectivas consequências elétricas, foram criados modelos baseados em circuitos independentes, sendo considerados os interruptores como resistores variáveis, onde no estado aberto sua resistência tem valor elevado de 1 Mega ohm ($M\Omega$) e no estado fechado sua resistência tende a zero recebendo o valor de 1 micro ohm ($\mu\Omega$).

Na modelagem das lâmpadas foi considerado um valor de resistência de 269 ohm (Ω), correspondendo a um dispositivo de 60 watts (W). Como alimentação do circuito foi considerado uma fonte de tensão de 127 volts (V), valor padronizado entre fase e neutro na rede elétrica da instituição.

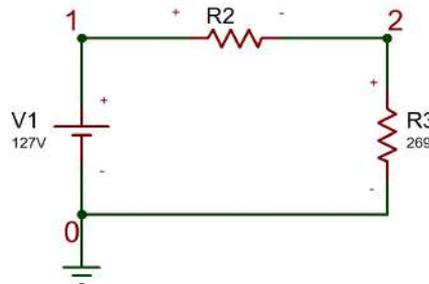
a) *Interruptor Simples, Campainha e Cigarra*: Para os circuitos das Figuras 14, 15 e 16 a lâmpada ou os dispositivos sonoros devem receber um valor de corrente de 0.4721189 ampère (A) e a fonte de tensão deve fornecer o mesmo valor de corrente. Embora sejam circuitos elétricos para aplicações distintas, o modelo do circuito elétrico equivalente é o mesmo para o interruptor simples, a campainha e a cigarra.

Para o correto funcionamento do circuito do *Interruptor Simples, Campainha e Cigarra*, foi utilizado um ponto de comando (variável de entrada A), assim este circuito terá 2

($2^n = 2^1$, onde n representa o número de variáveis de entrada) possibilidades de acionamento (DE LOURENÇO et al., 2009).

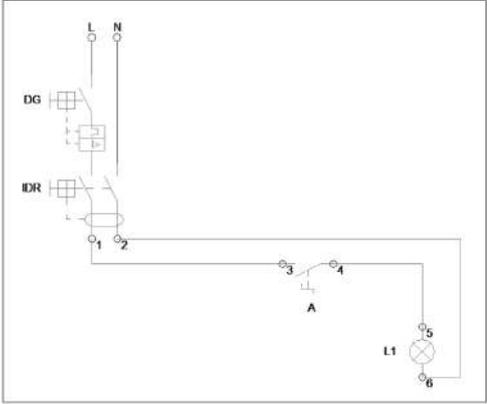
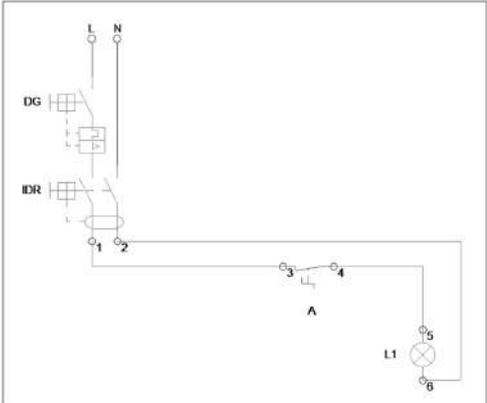
Considerando o interruptor simples, campainha e cigarra como um dispositivo de lógica binária, foi realizado a modelagem elétrica do circuito, visualizado na Figura 24, como um resistor (R2) que em situação de chave aberta (estado lógico 0) recebe o valor de $1 \text{ M}\Omega$. Já na situação de chave fechada (estado lógico 1) é atribuído o valor de $1 \mu\Omega$ no resistor (R2). Para o circuito, a carga é modelada por um resistor (R3) de 269Ω , podendo ser visualizado na Tabela 1 a posição da chave do interruptor simples (podendo a análise ser extrapolada para a campainha e cigarra) no diagrama multifilar e o valor da resistência elétrica modelada de acordo com seu respectivo estado lógico.

FIGURA 24 - MODELO ELÉTRICO DO INTERRUPTOR SIMPLES, CIGARRA E CAMPAINHA



Fonte: Elaboração própria (2023)

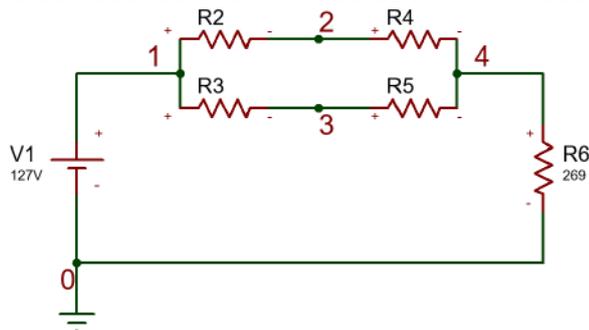
TABELA 1 - VALOR DA RESISTÊNCIA MODELADA DE ACORDO COM A POSIÇÃO DA CHAVE PARA O INTERRUPTOR SIMPLES, CAMPAINHA E CIGARRA

Entrada	Valor de resistência atribuído ao modelo	Diagrama multifilar correspondente
A	R2 (Ω)	
0	1 M	
1	1 μ	

Fonte: Elaboração própria (2023)

b) *Interruptor Paralelo*: Para o modelo elétrico do *Interruptor Paralelo*, visualizado na Figura 25, sendo uma chave seletora, foi representado por dois resistores (R2 e R3) para o primeiro dispositivo de acionamento (variável de entrada A) e por mais dois resistores (R4 e R5) para o segundo elemento de comando (variável de entrada B). Para o circuito, a carga é modelada por um resistor (R6) de 269 Ω .

FIGURA 25 - MODELO ELÉTRICO DO INTERRUPTOR PARALELO (*THREE-WAY*)



Fonte: Elaboração própria (2023)

Para o correto funcionamento do circuito do *Interruptor Paralelo*, foram utilizados dois pontos de entrada, assim este circuito terá 4 ($2^n = 2^2$) possibilidades de acionamento (DE LOURENÇO et al., 2009).

Para fins de análise foi considerado como estado lógico 0 quando a chave seletora encontra-se na posição inferior e como estado lógico 1 quando a chave seletora encontra-se na posição superior, sendo aplicado este padrão nas duas entradas do circuito elétrico.

Os resistores que compõem o modelo elétrico do dispositivo sempre assumirão valores distintos, pois o contato elétrico do interruptor físico somente permite a passagem de corrente elétrica em um dos ramos (superior ou inferior) por vez. Desta forma, quando a chave seletora está posicionado para baixo (estado lógico 0) foi atribuído aos resistores que compõem o ramo inferior do modelo elétrico (R3 e R5) o valor de $1 \mu\Omega$ e os resistores que compõem o ramo superior do modelo elétrico (R2 e R4) recebem o valor de $1 M\Omega$.

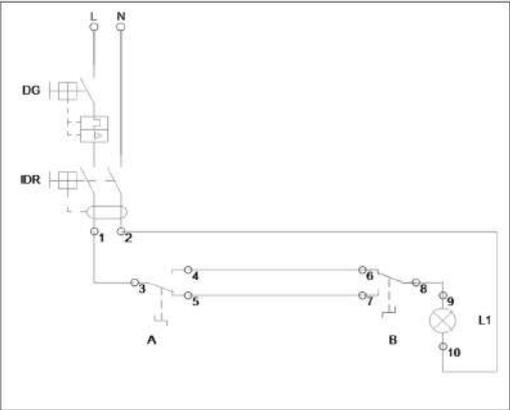
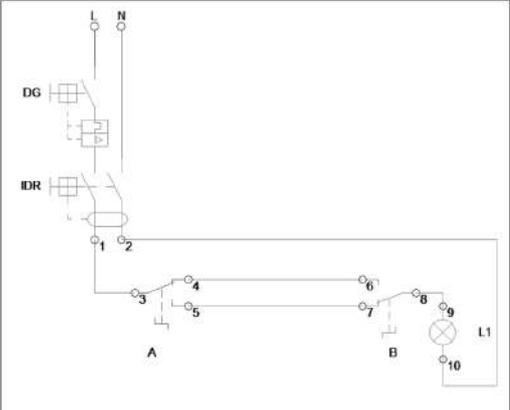
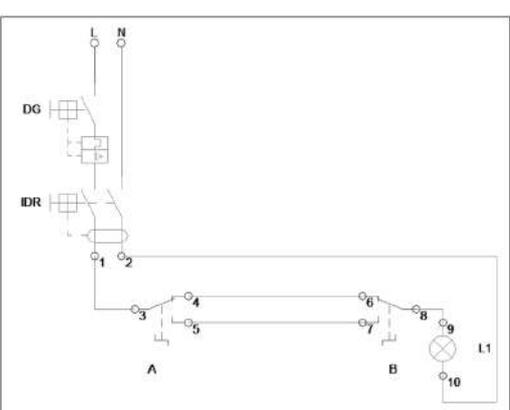
Quando a chave seletora encontra-se posicionado para cima (estado lógico 1) foi atribuído aos resistores que compõem o ramo inferior do modelo elétrico (R3 e R5) o valor de $1 M\Omega$ e os resistores que compõem o ramo superior do modelo elétrico (R2 e R4) recebem o valor de $1 \mu\Omega$.

Na Tabela 2 pode ser visualizado a posição da chave no diagrama multifilar e o valor da resistência elétrica modelada de acordo com seu respectivo estado lógico.

TABELA 2 - VALOR DA RESISTÊNCIA MODELADA DE ACORDO COM A POSIÇÃO DA CHAVE PARA O INTERRUPTOR PARALELO (CONTINUA)

Entrada		Valor de resistência atribuído ao modelo				Diagrama multifilar correspondente
A	B	R2 (Ω)	R3 (Ω)	R4 (Ω)	R5 (Ω)	
0	0	1 M	1μ	1 M	1μ	

TABELA 2 - VALOR DA RESISTÊNCIA MODELADA DE ACORDO COM A POSIÇÃO DA CHAVE PARA O INTERRUPTOR PARALELO (CONCLUSÃO)

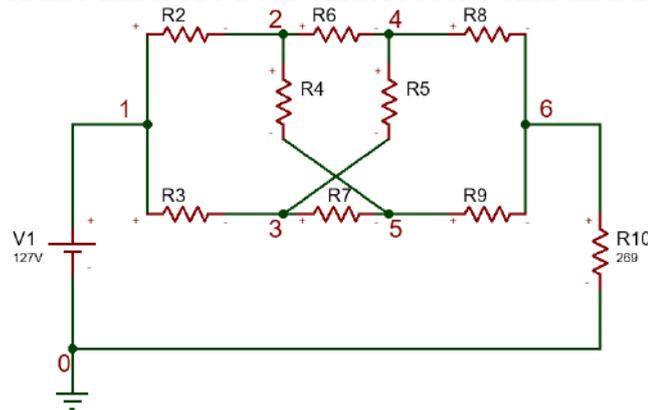
Entrada		Valor de resistência atribuído ao modelo				Diagrama multifilar correspondente
A	B	R2 (Ω)	R3 (Ω)	R4 (Ω)	R5 (Ω)	
0	1	1 M	1 μ	1 μ	1 M	
1	0	1 μ	1 M	1 M	1 μ	
1	1	1 μ	1 M	1 μ	1 M	

Fonte: Elaboração própria (2023)

c) *Interruptor Intermediário*: Para o modelo elétrico do *Interruptor Intermediário*, visualizado na Figura 26, foram utilizados quatro resistores (R4, R5, R6 e R7), representando 2 ligações distintas, onde R4 e R5 representam a circulação de corrente cruzada (estado lógico 1) e R6 e R7 representam a circulação de corrente em paralelo (estado lógico 0), onde esses

pares de resistores devem sempre receber o mesmo valor de resistência. Para a representação dos interruptores paralelos, foram modelados dois resistores (R2 e R3) para simulação do comportamento do primeiro dispositivo e dois resistores (R8 e R9) para o segundo *three-way*. No circuito, a carga é modelada por um resistor (R10) de 269 Ω .

FIGURA 26 - MODELO ELÉTRICO DO INTERRUPTOR INTERMEDIÁRIO (*FOUR-WAY*)



Fonte: Elaboração própria (2023)

Para o correto funcionamento do circuito do *Interruptor Intermediário*, foram utilizados três pontos de comando (dois interruptores paralelos, correspondendo às variáveis de entrada A e C respectivamente e um interruptor intermediário, que corresponde a variável de entrada B), assim este circuito terá 8 ($2^n = 2^3$) possibilidades de acionamento (DE LOURENÇO et al., 2009).

Neste circuito os modelos elétricos dos interruptores paralelos instalados irão se comportar de forma semelhante ao apresentado no item *b*, assim, quando a chave seletora está posicionado para baixo (estado lógico 0) foi atribuído aos resistores que compõem o ramo inferior do modelo elétrico (R3 e R9) o valor de 1 $\mu\Omega$ e os resistores que compõem o ramo superior do modelo elétrico (R2 e R8) recebem o valor de 1 $M\Omega$.

Quando a chave seletora encontra-se posicionado para cima (estado lógico 1) foi atribuído aos resistores que compõem o ramo inferior do modelo elétrico (R3 e R9) o valor de 1 $M\Omega$ e os resistores que compõem o ramo superior do modelo elétrico (R2 e R8) recebem o valor de 1 $\mu\Omega$. Sendo aplicado este padrão nas entradas A e C (interruptores paralelos) do circuito elétrico.

No modelo elétrico do dispositivo intermediário no estado lógico 0, é atribuído o valor de 1 $\mu\Omega$ aos resistores R6 e R7 e o valor de 1 $M\Omega$ aos resistores R4 e R5, assim permitindo a circulação de corrente pelos ramos paralelos do dispositivo. Já quando o dispositivo se encontra no estado lógico 1, é atribuído o valor de 1 $M\Omega$ aos resistores R6 e R7 e o valor de 1

$\mu\Omega$ aos resistores R4 e R5, assim permitindo a circulação de corrente pelos ramos cruzados do dispositivo.

Na Tabela 3 pode ser visualizado a posição da chave no diagrama multifilar e o valor da resistência elétrica modelada de acordo com seu respectivo estado lógico.

TABELA 3 - VALOR DA RESISTÊNCIA MODELADA DE ACORDO COM A POSIÇÃO DA CHAVE PARA O INTERRUPTOR INTERMEDIÁRIO (CONTINUA)

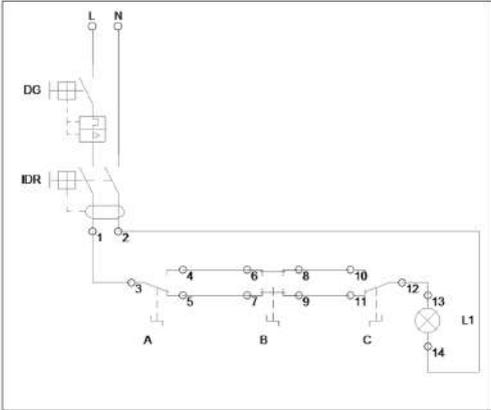
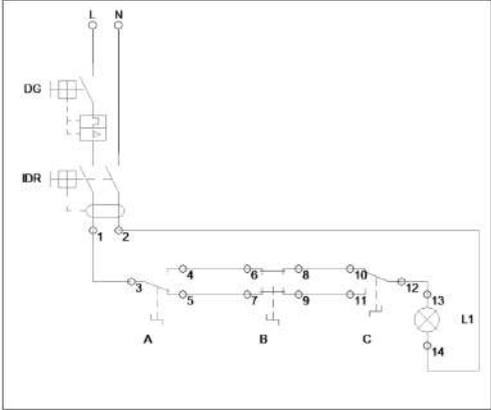
Entrada			Valor de resistência atribuído ao modelo									Diagrama multifilar correspondente
A	B	C	R2 (Ω)	R3 (Ω)	R4 (Ω)	R5 (Ω)	R6 (Ω)	R7 (Ω)	R8 (Ω)	R9 (Ω)		
0	0	0	1M	1 μ	1M	1M	1 μ	1 μ	1M	1 μ		
0	0	1	1M	1 μ	1M	1M	1 μ	1 μ	1 μ	1M		

TABELA 3 - VALOR DA RESISTÊNCIA MODELADA DE ACORDO COM A POSIÇÃO DA CHAVE PARA O INTERRUPTOR INTERMEDIÁRIO (CONTINUAÇÃO)

Entrada			Valor de resistência atribuído ao modelo								Diagrama multifilar correspondente
A	B	C	R2 (Ω)	R3 (Ω)	R4 (Ω)	R5 (Ω)	R6 (Ω)	R7 (Ω)	R8 (Ω)	R9 (Ω)	
0	1	0	1M	1 μ	1 μ	1 μ	1M	1M	1M	1 μ	
0	1	1	1M	1 μ	1 μ	1 μ	1M	1M	1 μ	1M	
1	0	0	1 μ	1M	1M	1M	1 μ	1 μ	1M	1 μ	

TABELA 3 - VALOR DA RESISTÊNCIA MODELADA DE ACORDO COM A POSIÇÃO DA CHAVE PARA O INTERRUPTOR INTERMEDIÁRIO (CONCLUSÃO)

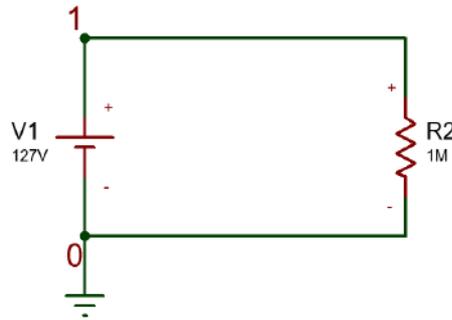
Entrada			Valor de resistência atribuído ao modelo									Diagrama multifilar correspondente
A	B	C	R2 (Ω)	R3 (Ω)	R4 (Ω)	R5 (Ω)	R6 (Ω)	R7 (Ω)	R8 (Ω)	R9 (Ω)		
1	0	1	1 μ	1 M	1 M	1 M	1 μ	1 μ	1 μ	1 M		
1	1	0	1 μ	1 M	1 μ	1 μ	1 M	1 M	1 M	1 μ		
1	1	1	1 μ	1 M	1 μ	1 μ	1 M	1 M	1 μ	1 M		

Fonte: Elaboração própria (2023)

d) *Tomada 2P+T*: Este circuito foi modelado eletricamente como sendo um circuito ideal, assim a tomada não representa uma carga, pois não provoca nenhuma queda de tensão. Para que seja possível sua análise foi considerado como carga o próprio instrumento de medição de

tensão (voltímetro), que por característica segundo Sadiku, Musa e Alexander (2013) deve possuir alta resistência interna (para um modelo ideal deve ser infinita) para interferir minimamente nas leituras de corrente, assim o voltímetro foi representado como um resistor (R2) de $1\text{ M}\Omega$, este modelo elétrico pode ser visualizado na Figura 27.

FIGURA 27 - MODELO ELÉTRICO DA TOMADA 2P + T

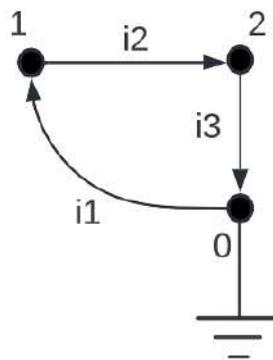


Fonte: Elaboração própria (2023)

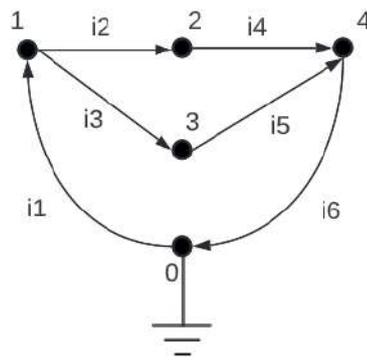
4.4.4.4 Construção dos dígrafos

Para cada modelo elétrico do circuito é realizada a representação do seu dígrafo, demonstrado na Figura 28a o dígrafo do *Interruptor Simples, Cigarra e Campainha*, na Figura 28b o dígrafo do *Interruptor Paralelo*, na Figura 28c o dígrafo do *Interruptor Intermediário* e na Figura 28d o dígrafo da *Tomada 2P+T*, onde é possível a visualização dos nós e o sentido da corrente nos ramos, assim permitindo a montagem da matriz de incidência dos circuitos.

FIGURA 28 - DÍGRAFOS DOS CIRCUITOS IMPLEMENTADOS (CONTINUA)

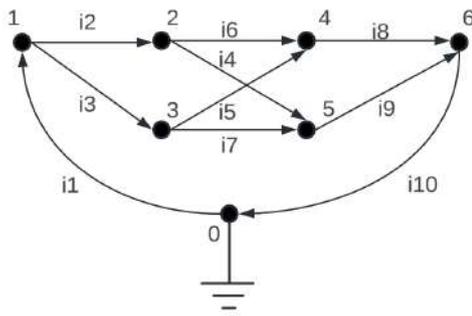


(a) Interruptor Simples, Campainha e Cigarra

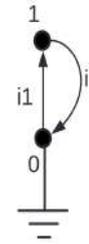


(b) Interruptor Paralelo

FIGURA 28 - DÍGRAFOS DOS CIRCUITOS IMPLEMENTADOS (CONCLUSÃO)



(c) Interruptor Intermediário



(d) Tomada 2P+T

Fonte: Elaboração própria (2023)

4.4.4.5 Construção da matriz de incidência

Utilizando a técnica da montagem da matriz de incidência (A), apresentado no item 2.4, onde as linhas da matriz correspondem ao nó do circuito (n), as colunas correspondem ao ramo (b) e considerando as correntes que entram no nó tendo sinal negativo, construiu-se as matrizes para cada um dos dígrafos, onde A terá as dimensões de $(n - 1) \times b$.

a) *Interrupto Simples, Campainha e Cigarra*: Para o circuito composto por 3 nós ($n = 3$) e 3 ramos ($b = 3$), a matriz de incidência (A) terá as dimensões de 2 linhas ($n-1$) e 3 colunas, sendo demonstrado no Quadro 3 o modelo elétrico do circuito com seu respectivo dígrafo e matriz de incidência.

QUADRO 3 - MATRIZ DE INCIDÊNCIA DO INTERRUPTOR SIMPLES, CAMPAINHA E CIGARRA

Modelo Elétrico	Dígrafo	Matriz de Incidência ($A_{n-1,b}$)
		$\begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$

Fonte: Elaboração própria (2023)

b) *Interruptor Paralelo*: Para o circuito composto por 5 nós ($n = 5$) e 6 ramos ($b = 6$), a matriz de incidência (A) terá as dimensões de 4 linhas ($n-1$) e 6 colunas, sendo demonstrado no Quadro 4 o modelo elétrico do circuito com seu respectivo dígrafo e matriz de incidência.

QUADRO 4 - MATRIZ DE INCIDÊNCIA DO INTERRUPTOR PARALELO

Modelo Elétrico	Dígrafo	Matriz de Incidência ($A_{n-1,b}$)
		$\begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix}$

Fonte: Elaboração própria (2023)

c) *Interruptor Intermediário*: Para o circuito composto por 7 nós ($n = 7$) e 10 ramos ($b = 10$), a matriz de incidência (A) terá as dimensões de 6 linhas ($n-1$) e 10 colunas, sendo demonstrado no Quadro 5 o modelo elétrico do circuito com seu respectivo dígrafo e matriz de incidência.

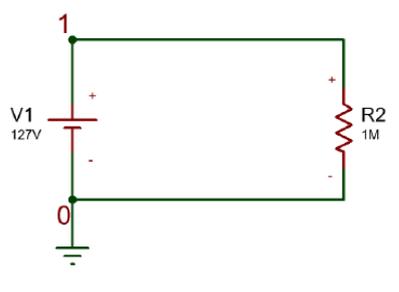
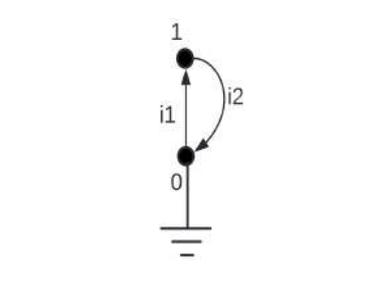
QUADRO 5 - MATRIZ DE INCIDÊNCIA DO INTERRUPTOR INTERMEDIÁRIO

Modelo Elétrico	Dígrafo	Matriz de Incidência ($A_{n-1,b}$)
		$\begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix}$

Fonte: Elaboração própria (2023)

d) *Tomada 2P+T*: Para o circuito composto por 2 nós ($n = 2$) e 2 ramos ($b = 2$), a matriz de incidência (A) terá as dimensões de 1 linha ($n - 1$) e 2 colunas, sendo demonstrado no Quadro 6 o modelo elétrico do circuito com seu respectivo dígrafo e matriz de incidência.

QUADRO 6 - MATRIZ DE INCIDÊNCIA DA TOMADA 2P+T

Modelo Elétrico	Dígrafo	Matriz de Incidência ($A_{n-1,b}$)
		$\begin{bmatrix} -1 & 1 \end{bmatrix}$

Fonte: Elaboração própria (2023)

4.4.4.6 Equacionamento da LKC para os circuitos dos modelos elétricos

Para montagem da equação de Kirchhoff das correntes, foi realizada a construção do vetor de correntes dos ramos ($i(t)$), sendo um vetor coluna contendo todas as correntes dos ramos para cada circuito modelado. Todas as equações da LKC podem ser vistas nos itens subsequentes.

a) *Interrupto Simples, Campainha e Cigarra*: Para o circuito com 3 ramos o vetor $i(t)$ será composto por 3 linhas e 1 coluna. Utilizando a matriz de incidência de acordo com o Quadro 3 e aplicando a Equação 7 para cálculo das correntes dos ramos, temos o equacionamento demonstrado na Equação 22.

$$LKC \rightarrow A \cdot i(t) = 0 \rightarrow \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} i1 \\ i2 \\ i3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (22)$$

b) *Interruptor Paralelo*: Para o circuito com 6 ramos o vetor $i(t)$ será composto por 6 linhas e 1 coluna. Utilizando a matriz de incidência de acordo com o Quadro 4 e aplicando a Equação 7 para cálculo das correntes dos ramos, temos o equacionamento demonstrado na Equação 23.

$$LKC \rightarrow A \cdot i(t) = 0 \rightarrow \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} i1 \\ i2 \\ i3 \\ i4 \\ i5 \\ i6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (23)$$

c) *Interruptor Intermediário*: Para o circuito com 10 ramos o vetor $i(t)$ será composto por 10 linhas e 1 coluna. Utilizando a matriz de incidência de acordo com o Quadro 5 e aplicando a

Equação 7 para cálculo das correntes dos ramos, temos o equacionamento demonstrado na Equação 24.

$$LKC \rightarrow A \cdot i(t) = 0 \rightarrow \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} i1 \\ i2 \\ i3 \\ i4 \\ i5 \\ i6 \\ i7 \\ i8 \\ i9 \\ i10 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (24)$$

d) *Tomada 2P+T*: Para o circuito com 2 ramos o vetor $i(t)$ será composto por 2 linhas e 1 coluna. Utilizando a matriz de incidência de acordo com o Quadro 6 e aplicando a Equação 7 para cálculo das correntes dos ramos, temos o equacionamento demonstrado na Equação 25.

$$LKC \rightarrow A \cdot i(t) = 0 \rightarrow [-1 \quad 1] \cdot \begin{bmatrix} i1 \\ i2 \end{bmatrix} = 0 \quad (25)$$

4.4.4.7 Equacionamento da LKT para os circuitos dos modelos elétricos

Para montagem da equação de *Kirchhoff* das tensões foi realizada a construção dos vetores de tensão dos ramos (v) e tensão dos nós (e). Também deve-se transpor as matrizes de incidência (A^T) para cada circuito modelado.

O vetor v é uma matriz do tipo coluna contendo todas as tensões dos dispositivos conectados ao circuito. No vetor e são inseridos os valores das tensões de cada nó que compõem o arranjo, excluindo-se a tensão do nó de referência. Todas as equações da LKT podem ser vistas nos itens subsequentes.

a) *Interrupto Simples, Campainha e Cigarra*: De acordo com o dígrafo apresentado na Figura 28a, este composto por 3 ramos e 2 nós (excluindo-se o nó de referência), assim o vetor v será de 3 linhas por 1 coluna e o vetor e será de 2 linhas por 1 coluna. Aplicando a Equação 8 para cálculo das tensões dos ramos e dos nós, temos o equacionamento demonstrado na Equação 26.

$$LKT \rightarrow v - A^T \cdot e = 0 \rightarrow \begin{bmatrix} v1 \\ v2 \\ v3 \end{bmatrix} - \begin{bmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e1 \\ e2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (26)$$

b) *Interruptor Paralelo*: De acordo com o dígrafo apresentado na Figura 28b, este composto por 6 ramos e 4 nós (excluindo-se o nó de referência), assim o vetor v será de 6 linhas por 1 coluna e o vetor e será de 4 linhas por 1 coluna. Aplicando a Equação 8 para cálculo das tensões dos ramos e dos nós, temos o equacionamento demonstrado na Equação 27.

$$LKT \rightarrow v - A^T \cdot e = 0 \rightarrow \begin{bmatrix} v1 \\ v2 \\ v3 \\ v4 \\ v5 \\ v6 \end{bmatrix} - \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e1 \\ e2 \\ e3 \\ e4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (27)$$

c) *Interruptor Intermediário*: De acordo com o dígrafo apresentado na Figura 28c, este composto por 10 ramos e 6 nós (excluindo-se o nó de referência), assim o vetor v será de 10 linhas por 1 coluna e o vetor e será de 6 linhas por 1 coluna. Aplicando a Equação 8 para cálculo das tensões dos ramos e dos nós, temos o equacionamento demonstrado na Equação 28.

$$LKT \rightarrow v - A^T \cdot e = 0 \rightarrow \begin{bmatrix} v1 \\ v2 \\ v3 \\ v4 \\ v5 \\ v6 \\ v7 \\ v8 \\ v9 \\ v10 \end{bmatrix} - \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e1 \\ e2 \\ e3 \\ e4 \\ e5 \\ e6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (28)$$

d) *Tomada 2P+T*: De acordo com o dígrafo apresentado na Figura 28d, este composto por 2 ramos e 1 nó (excluindo-se o nó de referência), assim o vetor v será de 2 linhas por 1 coluna e o vetor e será de 1 linha por 1 coluna. Aplicando a Equação 8 para cálculo das tensões dos ramos e dos nós, temos o equacionamento demonstrado na Equação 29.

$$LKT \rightarrow v - A^T \cdot e = 0 \rightarrow \begin{bmatrix} v1 \\ v2 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} \cdot [e1] = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (29)$$

4.4.4.8 Equação dos ramos

Também foram formuladas as equações para cada ramo dos circuitos modelados de acordo com a 1ª Lei de Ohm, onde a queda de tensão no dispositivo é igual ao produto da resistência que compõem o ramo pela corrente que circula por ele.

Sendo essas equações lineares, é possível a reescrita como demonstrado na Equação 9, assim possibilitando a visualização das matrizes $M(t)$, que será multiplicada pelo vetor de tensão dos ramos e a matriz $N(t)$, que será multiplicada pelo vetor de corrente dos ramos.

a) *Interruptor Simples, Campainha e Cigarra*: Utilizando o vetor $i(t)$ demonstrado na Equação 22, o vetor v visualizado na Equação 26 e mantendo como referência para análise o dígrafo da Figura 28a, foram formuladas as equações para cada ramo do circuito, temos os equacionamentos demonstrados nas Equações 30 e 31.

$$\begin{aligned} \text{Equação dos Ramos} \rightarrow v1 &= 127V \\ v2 &= R2 \cdot i2 \rightarrow v2 - R2 \cdot i2 = 0 \\ v3 &= R3 \cdot i3 \rightarrow v3 - R3 \cdot i3 = 0 \end{aligned} \quad (30)$$

$$M(t) \cdot v(t) + N(t) \cdot i(t) = u_s(t) \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v1 \\ v2 \\ v3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & -R2 & 0 \\ 0 & 0 & -R3 \end{bmatrix} \cdot \begin{bmatrix} i1 \\ i2 \\ i3 \end{bmatrix} = \begin{bmatrix} 127 \\ 0 \\ 0 \end{bmatrix} \quad (31)$$

b) *Interruptor Paralelo*: Utilizando o vetor $i(t)$ demonstrado na Equação 23, o vetor v visualizado na Equação 27 e mantendo como referência para análise o dígrafo da Figura 28b, foram formuladas as equações para cada ramo do circuito, temos os equacionamentos demonstrados nas Equações 32 e 36.

$$\begin{aligned} \text{Equação dos Ramos} \rightarrow v1 &= 127V \\ v2 &= R2 \cdot i2 \rightarrow v2 - R2 \cdot i2 = 0 \\ v3 &= R3 \cdot i3 \rightarrow v3 - R3 \cdot i3 = 0 \\ v4 &= R4 \cdot i4 \rightarrow v4 - R4 \cdot i4 = 0 \\ v5 &= R5 \cdot i5 \rightarrow v5 - R5 \cdot i5 = 0 \\ v6 &= R6 \cdot i6 \rightarrow v6 - R6 \cdot i6 = 0 \end{aligned} \quad (32)$$

$$M(t) \cdot v(t) + N(t) \cdot i(t) = u_s(t) \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v1 \\ v2 \\ v3 \\ v4 \\ v5 \\ v6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -R2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -R3 & 0 & 0 & 0 \\ 0 & 0 & 0 & -R4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -R5 & 0 \\ 0 & 0 & 0 & 0 & 0 & -R6 \end{bmatrix} \cdot \begin{bmatrix} i1 \\ i2 \\ i3 \\ i4 \\ i5 \\ i6 \end{bmatrix} = \begin{bmatrix} 127 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (33)$$

c) *Interruptor Intermediário*: Utilizando o vetor $i(t)$ demonstrado na Equação 24, o vetor v visualizado na Equação 28 e mantendo como referência para análise o dígrafo da Figura 28c, foram formuladas as equações para cada ramo do circuito, temos os equacionamentos demonstrados nas Equações 34 e 35.

$$\begin{aligned} v1 &= 127V \\ v2 &= R2 \cdot i2 \rightarrow v2 - R2 \cdot i2 = 0 \\ v3 &= R3 \cdot i3 \rightarrow v3 - R3 \cdot i3 = 0 \\ v4 &= R4 \cdot i4 \rightarrow v4 - R4 \cdot i4 = 0 \\ \text{Equação dos Ramos} \rightarrow v5 &= R5 \cdot i5 \rightarrow v5 - R5 \cdot i5 = 0 \\ v6 &= R6 \cdot i6 \rightarrow v6 - R6 \cdot i6 = 0 \\ v7 &= R7 \cdot i7 \rightarrow v7 - R7 \cdot i7 = 0 \\ v8 &= R8 \cdot i8 \rightarrow v8 - R8 \cdot i8 = 0 \\ v9 &= R9 \cdot i9 \rightarrow v9 - R9 \cdot i9 = 0 \\ v10 &= R10 \cdot i10 \rightarrow v10 - R10 \cdot i10 = 0 \end{aligned} \quad (34)$$

$$M(t) \cdot v(t) + N(t) \cdot i(t) = u_s(t) \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v1 \\ v2 \\ v3 \\ v4 \\ v5 \\ v6 \\ v7 \\ v8 \\ v9 \\ v10 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -R2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -R3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -R4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -R5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -R6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -R7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R10 \end{bmatrix} \cdot \begin{bmatrix} i1 \\ i2 \\ i3 \\ i4 \\ i5 \\ i6 \\ i7 \\ i8 \\ i9 \\ i10 \end{bmatrix} = \begin{bmatrix} 127 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (35)$$

d) *Tomada 2P+T*: Utilizando o vetor $i(t)$ demonstrado na Equação 25, o vetor v visualizado na Equação 29 e mantendo como referência para análise o dígrafo da Figura 28d, foram formuladas as equações para cada ramo do circuito, temos os equacionamentos demonstrados nas Equações 36 e 37.

$$\begin{aligned} \text{Equação dos Ramos} \rightarrow v1 &= 127V \\ v2 &= R2 \cdot i2 \rightarrow v2 - R2 \cdot i2 = 0 \end{aligned} \quad (36)$$

$$M(t) \cdot v(t) + N(t) \cdot i(t) = u_s(t) \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v1 \\ v2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -R2 \end{bmatrix} \cdot \begin{bmatrix} i1 \\ i2 \end{bmatrix} = \begin{bmatrix} 127 \\ 0 \end{bmatrix} \quad (37)$$

4.4.4.9 Montagem das matrizes de Tableau

Tendo desenvolvido e equacionado a LKC, a LKT e as equações dos ramos, é possível a visualização das matrizes que compõem a equação de Tableau de acordo com a Equação 11.

A matriz de Tableau ($T(t)$) sempre terá as dimensões de $(n-1) + 2b$ linhas por $(n-1) + 2b$ colunas, sendo uma matriz quadrada e variando seu tamanho de acordo com o circuito analisado. O vetor $w(t)$ terá as dimensões de $(n-1) + 2b$ linhas por 1 coluna e é composto pelos vetores de tensão dos nós ($e(t)$), tensão dos ramos ($v(t)$) e corrente dos ramos ($i(t)$). O vetor $u(t)$ é composto por vetores nulos nas suas duas primeiras posições e pelo vetor $u_s(t)$ na sua última posição e também possui dimensões de $(n-1) + 2b$ linhas e 1 coluna.

a) *Interruptor Simples, Campainha e Cigarra*: Para o circuito composto por 3 nós e 3 ramos de acordo com a Figura 28a, temos uma matriz $T(t)$ de 8 linhas por 8 colunas, e os vetores colunas $w(t)$ e $u(t)$ composto por 8 linhas. Na Equação 38 mostra-se a montagem da equação de Tableau para o circuito com a chave fechada (estado lógico 1) visto na Tabela 1, assim R2 tendendo a zero e em destaque as posições e tamanhos de cada matriz que compõem a equação.

$$\underbrace{\begin{bmatrix} 0 & 0 & A \\ -A^T & 1 & 0 \\ 0 & M(t) & N(t) \end{bmatrix}}_{T(t)} \cdot \underbrace{\begin{bmatrix} e(t) \\ v(t) \\ i(t) \end{bmatrix}}_{w(t)} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ u_s(t) \end{bmatrix}}_{u(t)} \rightarrow \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & & 0 & 0 & 0 & 0 & -1 & 1 \\ \hline 1 & 0 & & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & & 0 & 0 & 1 & 0 & 0 & -269 \end{array} \right] \cdot \begin{bmatrix} e1 \\ e2 \\ v1 \\ v2 \\ v3 \\ i1 \\ i2 \\ i3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \hline 127 \\ 0 \\ 0 \end{bmatrix} \quad (38)$$

b) *Interruptor Paralelo*: Para o circuito composto por 5 nós e 6 ramos de acordo com a Figura 28b, temos uma matriz $T(t)$ de 16 linhas por 16 colunas, e os vetores colunas $w(t)$ e $u(t)$ composto por 16 linhas. Na Equação 39 mostra-se a montagem da equação de Tableau para o circuito considerando a chave A = 1 e a chave B = 1, visto na Tabela 2, assim R2 e R4 tendendo a zero e em destaque as posições e tamanhos de cada matriz que compõe a equação.

$$\underbrace{\begin{bmatrix} 0 & 0 & A \\ -A^T & 1 & 0 \\ 0 & M(t) & N(t) \end{bmatrix}}_{T(t)} \cdot \underbrace{\begin{bmatrix} e(t) \\ v(t) \\ i(t) \end{bmatrix}}_{w(t)} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ u_s(t) \end{bmatrix}}_{u(t)} \rightarrow \left[\begin{array}{ccc|cc} 0 & 0 & 0 & -1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1000000 \end{array} \right] \cdot \begin{bmatrix} e1 \\ v1 \\ v2 \\ i1 \\ i2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 127 \\ 0 \end{bmatrix} \quad (41)$$

4.4.4.10 Solução dos circuitos modelados utilizando a equação de Tableau

A solução dos equacionamentos matriciais para cada circuito do modelo elétrico é obtido realizando-se a operação de inversão da matriz de Tableau e multiplicando pelo vetor $u(t)$, assim se obtendo o valor da tensão dos nós, dos ramos e a corrente dos ramos, conforme a Equação 42. Para realização desta operação foi utilizado o *software Scilab*, através de suas bibliotecas de cálculo matricial, podendo ser visualizado no Tabela 4 a solução dos cálculos efetuados.

$$w(t) = u(t) \cdot T(t)^{-1} \quad (42)$$

TABELA 4 - RESOLUÇÃO DOS EQUACIONAMENTOS (CONTINUA)

Circuito	Composição da matriz $w(t)$	Solução da matriz $w(t)$
Interruptor Simples, Campainha e Cigarra	$\begin{bmatrix} e1 \\ e2 \\ v1 \\ v2 \\ v3 \\ i1 \\ i2 \\ i3 \end{bmatrix}$	<pre>--> w w = -127. -127. 127. 0. -127. -0.4721190 -0.4721190 -0.4721190</pre>
Interruptor Paralelo	$\begin{bmatrix} e1 \\ e2 \\ e3 \\ e4 \\ v1 \\ v2 \\ v3 \\ v4 \\ v5 \\ v6 \\ i1 \\ i2 \\ i3 \\ i4 \\ i5 \\ i6 \end{bmatrix}$	<pre>--> w w = -127. -127. -127. -127. 127. 127. 0. 0. 0. 0. -127. -0.4721190 -0.4721190 0. -0.4721190 0. -0.4721190</pre>

TABELA 4 - RESOLUÇÃO DOS EQUACIONAMENTOS (CONCLUSÃO)

Circuito	Composição da matriz $w(t)$	Solução da matriz $w(t)$
Interruptor Intermediário	$\begin{bmatrix} e1 \\ e2 \\ e3 \\ e4 \\ e5 \\ e6 \\ v1 \\ v2 \\ v3 \\ v4 \\ v5 \\ v6 \\ v7 \\ v8 \\ v9 \\ v10 \\ i1 \\ i2 \\ i3 \\ i4 \\ i5 \\ i6 \\ i7 \\ i8 \\ i9 \\ i10 \end{bmatrix}$	<pre>--> w w = -127. -127. -127. -127. -127. -127. 127. 0. 0. 0. 0. 0. 0. 0. 0. 0. -127. -0.4721190 -0.4721190 0. 0. 0. -0.4721190 0. -0.4721190 0. -0.4721190</pre>
Tomada 2P+T	$\begin{bmatrix} e1 \\ v1 \\ v2 \\ i1 \\ i2 \end{bmatrix}$	<pre>--> w w = -127. 127. -127. -0.000127 -0.000127</pre>

Fonte: Elaboração própria (2023)

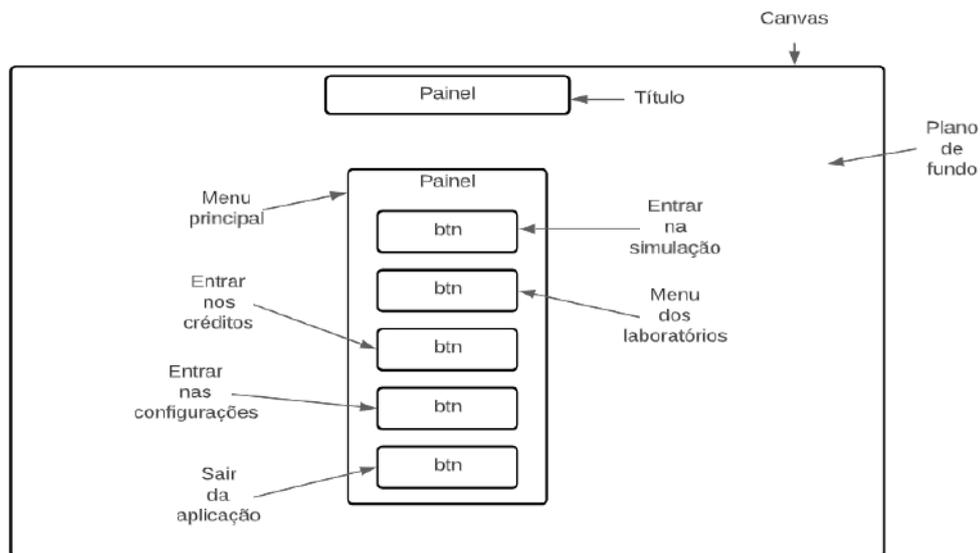
4.4.5 Projeto de *front end*

Para definição das telas que serão apresentadas em cena (*front end*), foi realizado o projeto dos *layouts* para a resolução em *Full High Definition* (HD), com 1920 *pixels* de resolução vertical por 1080 linhas horizontais, cabendo os ajustes necessários para responsividade das mesmas. Ressalta-se que o *Unity* possui como componentes elementos de interface do usuário (*User Interface – UI*), assim facilitando este desenvolvimento. Esses elementos sempre que inseridos em cena devem estar associados a um *Canvas*, sendo o elemento pai destes objetos. Desta forma, as telas projetadas utilizam esses componentes disponíveis no motor de jogos como: *Canvas*, *Panel*, *Button* e *Slider*.

Na Figura 29, mostra-se o layout do *menu* principal, estando na parte superior centralizada na tela o nome da aplicação e na parte central o painel responsável pela

organização dos 5 botões funcionais. O primeiro botão é responsável pelo direcionamento para a cena de circulação no corredor do setor, o segundo botão, direciona o aluno para a cena com o menu dos laboratórios, seguindo na descendente, temos o terceiro botão, que leva a cena com os créditos do projeto, o quarto e penúltimo botão é responsável pelo encaminhamento para cena de configurações e o quinto e último, o botão de saída da aplicação. Para uma tela mais amigável, também é projetado a inserção de imagem como plano de fundo, sendo utilizada em todas as cenas de *menu*.

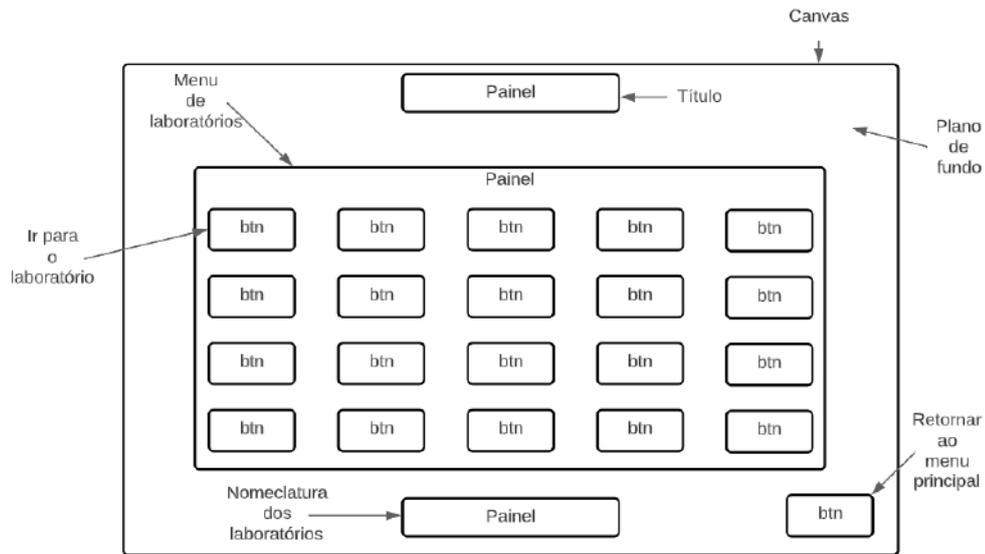
FIGURA 29 - LAYOUT MENU PRINCIPAL



Fonte: Elaboração própria (2023)

Na cena responsável pela indicação e direcionamento aos laboratórios, foi projetado em sua parte central o painel responsável pela organização dos botões que encaminham para os laboratórios distintos e na parte inferior centralizada do *canvas*, o espaço destinado para os painéis com a nomenclatura do laboratório selecionado. Voltando-se para parte inferior direita, é inserido o botão de retorno ao *menu* principal, seguindo assim o fluxo geral da aplicação demonstrado na Figura 7, o *front end* projetado e detalhado acima pode ser visto na Figura 30.

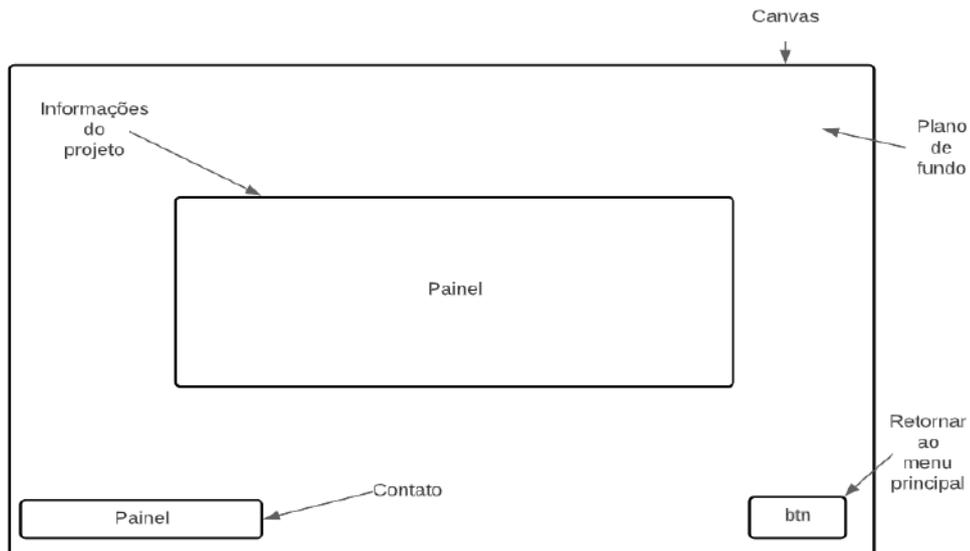
FIGURA 30 - LAYOUT MENU DE LABORATÓRIOS



Fonte: Elaboração própria (2023)

Outra cena projetada para a aplicação, demonstrada na Figura 31, é a de créditos, contendo o contato do autor do projeto, para sugestões de melhoria, estando essa informação no painel inferior esquerdo do *canvas*. Na parte central da tela, foi planejado o espaço para descrição da finalidade do projeto, nome do autor e a vinculação do programa de ensino. Também é incluído nesta cena o botão de retorno a tela inicial da aplicação, estando localizado na mesma posição do layout da Figura 30.

FIGURA 31 - LAYOUT DOS CRÉDITOS DO PROJETO

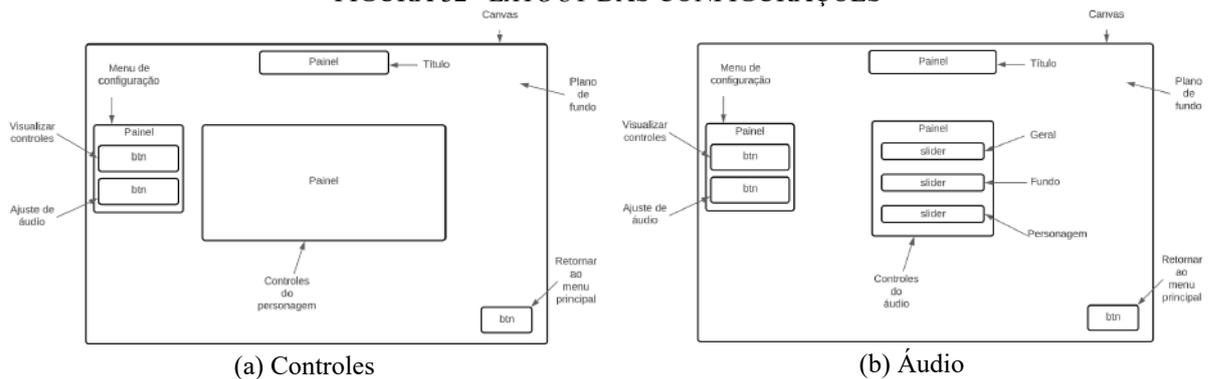


Fonte: Elaboração própria (2023)

O *front end* de configuração foi dividido entre as duas funções deste caso de uso, estando na mesma cena, como base encontra-se o painel responsável pela organização das

funções contidas nas configurações, podendo ser visualizado à esquerda dos *canvas*. Na parte superior central está o painel com o nome da aplicação e no canto inferior direito, o botão de retorno ao menu principal, semelhante às telas anteriormente descritas. Na Figura 32a, encontra-se na parte central o espaço contendo os controles do personagem, sendo o painel inicial desta cena. Na Figura 32b, mostra-se o painel com os controles de ajuste de volume do sistema, este elemento da UI será ativado ao clicar/tocar no botão do *menu* de configurações.

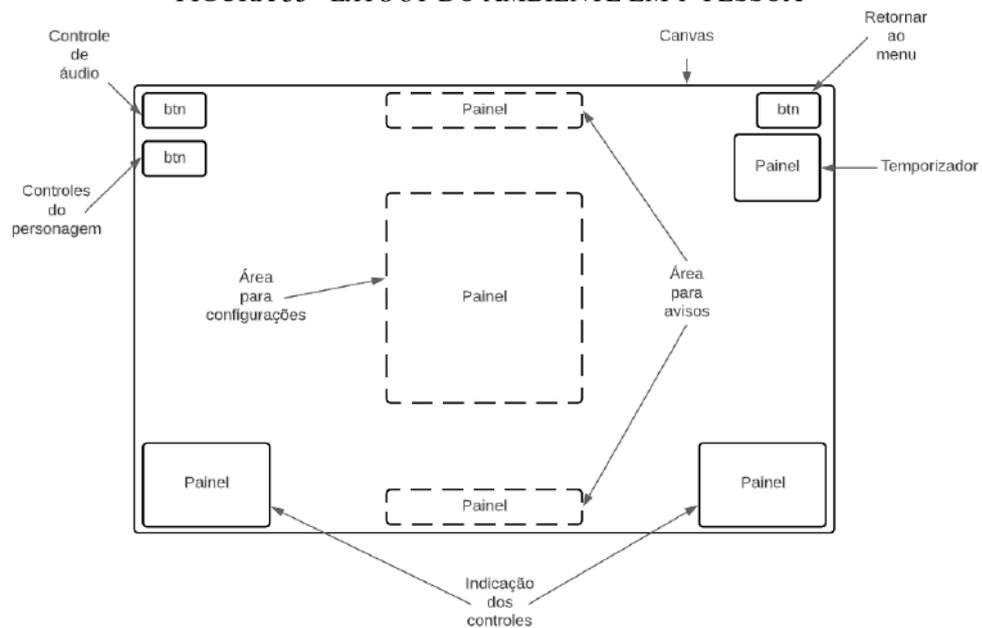
FIGURA 32 - LAYOUT DAS CONFIGURAÇÕES



Fonte: Elaboração própria (2023)

Para o ambiente em primeira pessoa, foi projetada uma tela contendo os botões de configuração na parte superior esquerda, e quando selecionados, as opções utilizam a área central da tela para demonstração de suas funcionalidades. Ainda na área superior do *canvas*, porém do lado direito, foi inserido o botão de retorno ao *menu* inicial, estando o painel com o cronômetro localizado abaixo deste. O temporizador inserido neste ambiente tem apenas a função de indicação do tempo de circulação, não trazendo nenhuma sanção ao aluno durante o deslocamento nesta cena. Na zona inferior esquerda e direita, foi inserido os painéis contendo todos os botões de movimentação e interação do usuário, para versão *Standalone* da aplicação e para a versão *Android*, esta área disponibiliza os controles *touchscreen* do personagem. Também é idealizado para esta tela duas áreas para notificações e avisos ao usuário, sendo estes pontos de *feedback* do sistema, como exemplo de utilização destas áreas, temos as indicações dos nomes dos laboratórios do setor, onde a tela projetada pode ser visualizada na Figura 33.

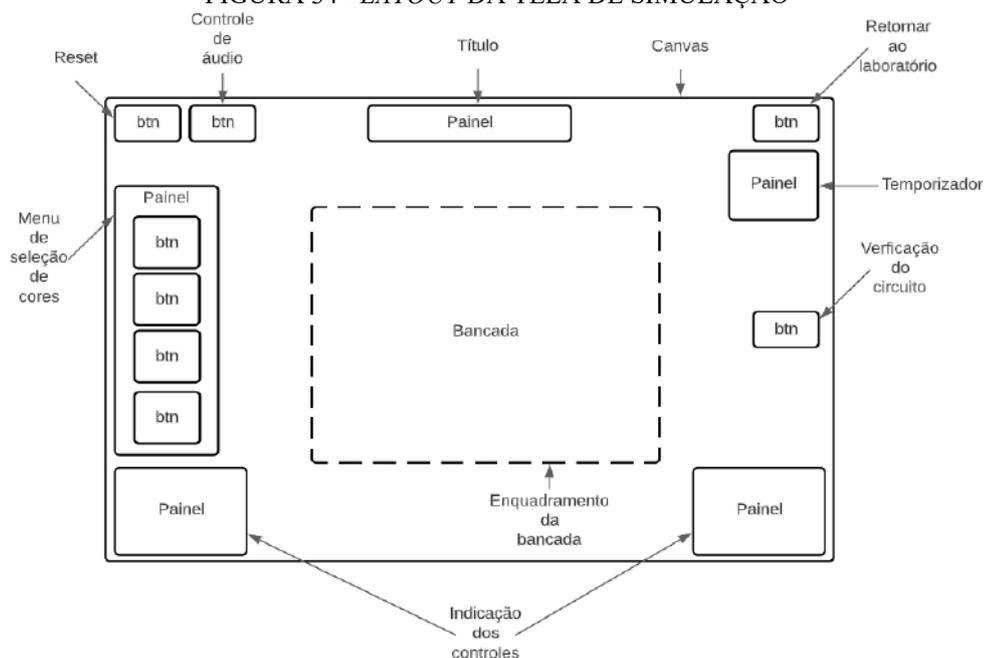
FIGURA 33 - LAYOUT DO AMBIENTE EM 1ª PESSOA



Fonte: Elaboração própria (2023)

Na Figura 34, mostra-se a tela projetada para a cena de simulação dos circuitos na bancada didática, seguindo a disposição base demonstrada na Figura 33, sendo adicionado o painel para seleção das cores dos cabos inseridos pelo aluno e o botão de checagem do circuito, estando posicionados à esquerda e à direita da tela, respectivamente. Os botões posicionados na parte superior esquerda do *canvas* tem a função de *reset*, assim realizando o reinício da simulação, e ao seu lado direito, o botão de ajuste do volume do áudio.

FIGURA 34 - LAYOUT DA TELA DE SIMULAÇÃO



Fonte: Elaboração própria (2023)

4.5 Desenvolvimento de Software

4.5.1 Classes

Para simulação do comportamento e dinâmica do programa, é necessário o desenvolvimento de *scripts*, onde o motor de jogos *Unity* trabalha com a linguagem *C#*, utilizando o paradigma da orientação ao objeto. Assim a aplicação foi dividida em classes com seus respectivos métodos e atributos, essas responsáveis pelo molde que serão aplicados, sendo a forma mais básica para definição de um objeto, como também de abstrair características similares das entidades (CARVALHO, 2016).

Esta subseção foi dividida entre as classes que possuem dependências, assim, ocorrendo relacionamento entre elas, e classes que atuam de forma isolada, replicando um comportamento ou agregando alguma função ao sistema, totalizando 28 classes desenvolvidas e aplicadas no sistema.

4.5.1.1 Classes dependentes

4.5.1.1.1 *OnClick*

A montagem do circuito é dependente da seleção dos soquetes dos dispositivos utilizados em cada simulação, assim cabendo a classe *OnClick* a tarefa de identificação do clique/toque no objeto previamente configurado com o componente *Collider* e o armazenamento do respectivo *status* da ação em variáveis booleanas. Para realização desta verificação são utilizados os métodos *OnMouseEnter()*, *OnMouseExit()*, *OnMouseDown()* e *OnMouseUp()*, todos fornecidos pela classe *MonoBehavior* do *Unity*.

4.5.1.1.2 *Wires*

Na ativação da conexão deve ser verificado se há alguma outra ligação ocupando o soquete selecionado, assim o método *Conexao()* realiza a checagem e ativação do objeto que representa o fio de interligação entre os pontos caso essas condições sejam satisfeitas, onde o fluxo de análise decisão pode ser visto na Figura 35.

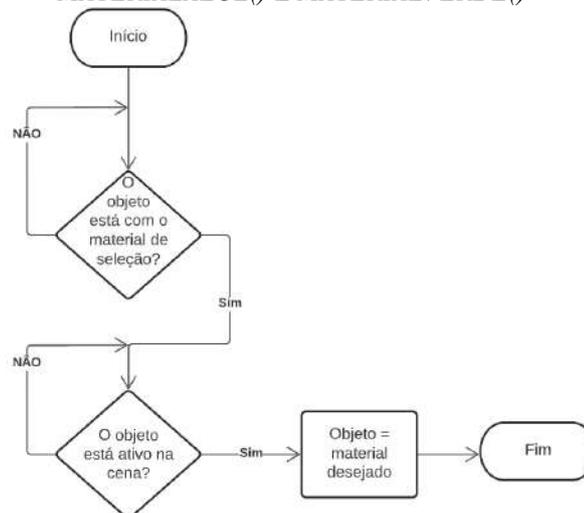
FIGURA 35 - FLUXOGRAMA DO MÉTODO *CONEXAO()*

Fonte: Elaboração própria (2023)

4.5.1.1.3 *SelecaoObjeto*

Um dos requisitos de segurança em uma instalação de BT é a identificação das funções do condutor utilizado na infraestrutura, sendo a indicação por cor uma das alternativas estabelecido pela NBR5410 e que foi utilizado na aplicação. Quando o condutor é instanciado em cena ele contém o material que representa uma cor genérica, também sendo esta cor um requisito para alteração da identificação da conexão, assim os métodos *MaterialPreto()*, *MaterialVermelho()*, *MaterialAzul()* e *MaterialVerde()* verificam o material atual do objeto através do parâmetro *sharedMaterial* contido no componente *Renderer* do *GameObject* e altera a renderização de acordo com a seleção do aluno através dos botões contidos no menu suspenso em tela, onde cada botão é responsável por chamar o método correspondente a sua cor, na Figura 36 mostra-se o fluxo dos métodos supracitados.

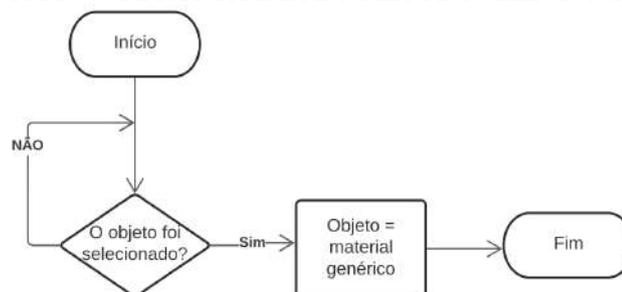
FIGURA 36 - FLUXOGRAMA DOS MÉTODOS *MATERIALPRETO()*, *MATERIALVERMELHO()*, *MATERIALAZUL()* E *MATERIALVERDE()*



Fonte: Elaboração própria (2023)

Caso ocorra erro na escolha da cor de identificação é possível realizar a correção através do método *Selecionado()*, onde o aluno deverá realizar o clique/toque no condutor que deseja a alteração do material, esse estando previamente configurado com o *MeshCollider*, assim sendo identificado a seleção e mudança para o material genérico, permitindo a realização do procedimento de alteração de cor anteriormente mencionado, sendo demonstrado na Figura 37.

FIGURA 37 - FLUXOGRAMA DO MÉTODO *SELECIONADO()*



Fonte: Elaboração própria (2023)

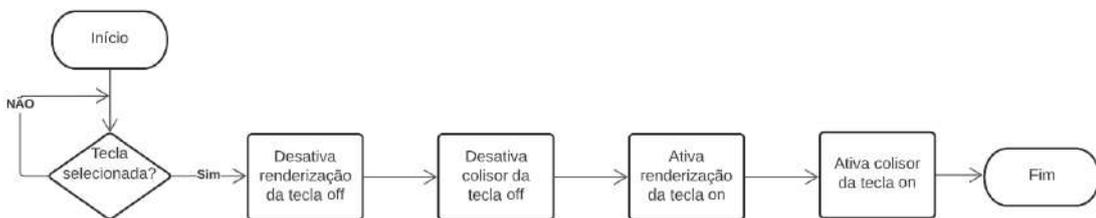
4.5.1.1.4 *LuzDestaque*

Como ponto de direcionamento e indicação dos objetos que podem ser manipulados na cena foi inserido um *spotlight* nos soquetes e teclas da bancada, cabendo a esta classe a ativação ou desativação da iluminação através do componente *Ligth* do *GameObjetc* de acordo com o clique/toque ou posicionamento do cursor do mouse sobre o objeto, devendo esta classe ser instanciada como componente do *spotlight*.

4.5.1.1.5 TeclaOnOff

Na Figura 38, demonstra-se o fluxo do método *OnOff()* da classe *TeclaOnOff*, onde os interruptores e dispositivos de proteção contidos neste protótipo são binários, estando abertos ou fechados de acordo com a posição de suas teclas, assim este método simula a mudança de posição dos interruptores, que ativa ou desativa a renderização da tecla, através do componente *MeshRenderer* e para evitar a ocorrência de *bug* também é realizado a ativação ou desativação do *BoxCollider* inserido em cada tecla. A verificação da seleção da tecla é realizada pela classe *OnClick*, assim cabendo a inclusão desta como componente dos objetos.

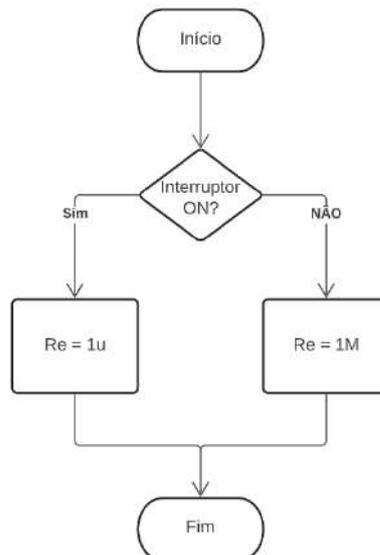
FIGURA 38 - FLUXOGRAMA DO MÉTODO *ONOFF()*



Fonte: Elaboração própria (2023)

Na Figura 39, mostra-se o fluxograma do outro método contido nesta classe, nomeado como *ValRes()*, este é responsável pela mudança dos valores da resistência no modelo elétrico do circuito, de acordo com a posição das teclas dos interruptores o modelo receberá o valor de $1\ \mu\Omega$ para chave fechada e $1\ M\Omega$ para chave aberta.

FIGURA 39 - FLUXOGRAMA DO MÉTODO *VALRES()*

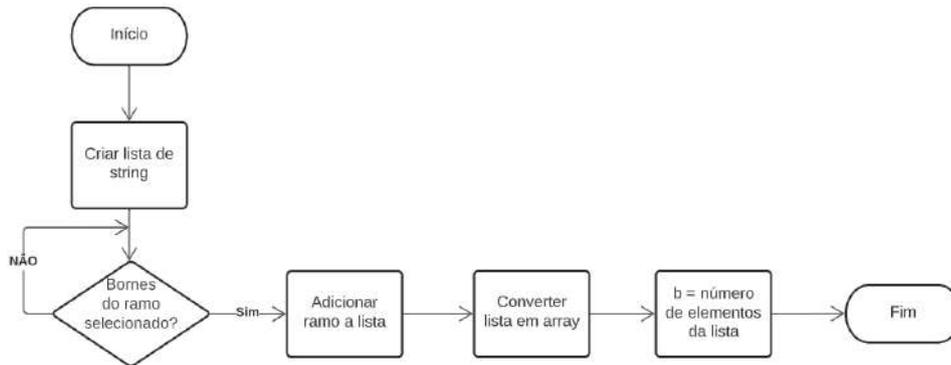


Fonte: Elaboração própria (2023)

4.5.1.1.6 Branch

De acordo com o modelo de análise de circuito, é de fundamental importância a capacidade de identificação dos ramos inseridos na ligação, para isto foi criada uma lista de *strings* que será responsável pelo armazenamento dos ramos dos circuitos através de seus nomes. Para inclusão destas nomenclaturas foi desenvolvido o método *InsertBranches()* contendo como parâmetros os soquetes que compõem o ramo e sua respectiva numeração, posteriormente é realizada a contabilização dos elementos desta lista, assim se obtendo o valor de *b* através do método *CountBranches()*, também é realizada a conversão da lista em *array*, para futura manipulação dos dados contidos na lista de controle, onde o fluxo da classe pode ser vista na Figura 40.

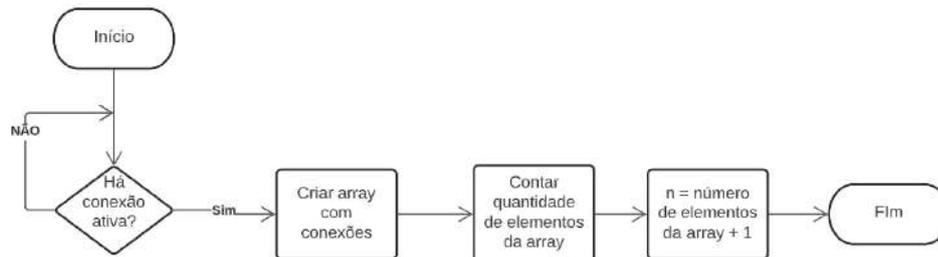
FIGURA 40 - FLUXOGRAMA DA CLASSE *BRANCH*



Fonte: Elaboração própria (2023)

4.5.1.1.7 Nodes

A verificação do número de nós do circuito simulado foi realizada através do método *CountNodes()*, sendo responsável pela chamada do método contido na classe *MonoBehavior FindGameObjectsWithTag*, onde previamente as conexões das práticas são classificadas com a *tag* “Conexão”, quando executado, este método cria uma *array* com os objetos ativos em cena que possuam a *tag* alvo, posteriormente é contabilizado o número de elementos da *array*, assim se obtendo o valor de *n*, este fluxo pode ser visualizado na Figura 41. Vale ressaltar que as conexões do neutro não são classificadas com a *tag*, pois será utilizado como nó de referência do circuito.

FIGURA 41 - FLUXOGRAMA DO MÉTODO *COUNTNODES()*

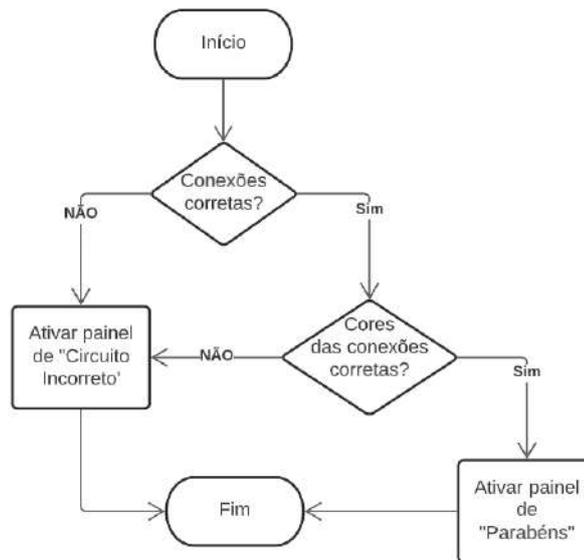
Fonte: Elaboração própria (2023)

4.5.1.1.8 *Booleanas*

Devido a necessidade de verificação do *status* de ativação dos objetos que representam as conexões, da situação de seleção dos soquetes e da renderização das teclas utilizadas em cena, foi desenvolvido a classe *Booleanas*, assim possuindo 2 *arrays* de tamanho variável para acesso aos objetos *conexao* e *soquetes*, para acesso às teclas é utilizado a *array* da classe *TeclasOnOff*. Assim a matriz de booleanas *C* armazena a situação de ativação dos objetos contidos na variável *conexao*, através da propriedade *activeSelf* do *GameObject*. Para armazenamento da ação de clique nos soquetes inseridos na *array soquete* foi criado outra matriz de booleanas denominada *S*, este acessa a booleana *ClicouNoObjeto* da classe *OnClick* inserido como componente dos soquetes. Para verificação da renderização das teclas foi criado a matriz de booleanas denominado de *R*, assim acessando os objetos vinculados na *array teclas* da classe *TeclasOnOff* e armazenando a situação de renderização em cena através da propriedade *enabled* do componente *MeshRenderer*.

4.5.1.1.9 *Corrigir*

Após a realização das conexões na simulação e seleção das cores dos fios é possível realizar a verificação e correção do circuito através do botão *Checar*, este responsável por instanciar a classe que realiza a análise de acordo com a NBR5410:2004. Para isto, a classe verifica por meio do método homônimo do botão cada conexão ativa em cena de acordo com a prática realizada e a propriedade *sharedMaterial* do componente *Renderer* contido nos objetos-alvo da análise. Estando de acordo com as normas é chamado o método *CircuitoCorreto* da classe *Feedback*, não estando de acordo, o método *CircuitoIncorreto* é acionado, podendo ser visualizado na Figura 42 o fluxo de decisão da classe.

FIGURA 42 - FLUXOGRAMA DO MÉTODO *CHECAR()*

Fonte: Elaboração própria (2023)

4.5.1.1.10 *Feedback*

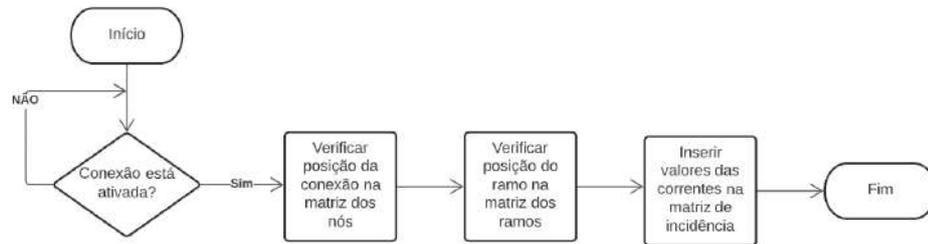
Essa classe realiza a ativação dos painéis de retorno da montagem do circuito, possuindo dois métodos *CircuitoCorreto()* e *CircuitoIncorreto()*, este sendo acionados pela classe *Corrigir*.

4.5.1.1.11 *Update_A*

Esta classe é responsável por atualizar os valores da corrente na matriz de incidência, de acordo com as conexões realizadas pelo discente, assim representando o arranjo do circuito modelado.

Para realização desta atualização é rastreado na matriz de controle de nós a posição da conexão ativada pelo aluno, assim identificando o nó conectado e na matriz de controle dos ramos do circuito a posição de cada ramo inserido, esta verificação é realizada através do método *IndexOf* contido na classe *Array* da biblioteca *System* da linguagem C#. Após as verificações são inseridos os valores da corrente na matriz de incidência de acordo com a LKC na posição rastreada pelo método anteriormente citado, na Figura 43 mostra-se o fluxo descrito.

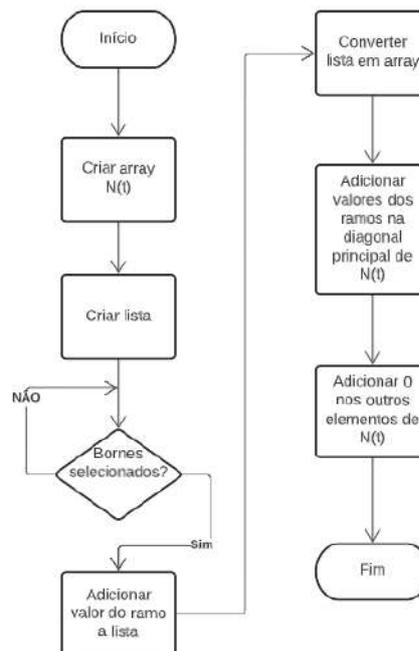
Esse rastreamento é necessário devido a característica de montagem do circuito, estando o aluno livre para conectar os dispositivos na ordem que desejar, assim podendo alterar o nome do nó de acordo com esta ordem de interligação.

FIGURA 43 - FLUXOGRAMA DO MÉTODO *ATUAL A()*

Fonte: Elaboração própria (2023)

4.5.1.1.12 *MatrizN*

Esta classe cria e preenche a matriz $N(t)$, com os valores dos ramos inseridos no circuito pelo aluno, assim permitindo o correto cálculo da matriz de Tableau. Mostra-se na Figura 44 que para essa verificação é realizado a checagem do clique/toque nos soquetes e são inseridos os valores das resistências que compõem o modelo elétrico de acordo com a inserção dos ramos.

FIGURA 44 - FLUXOGRAMA DO MÉTODO *MATN()*

Fonte: Elaboração própria (2023)

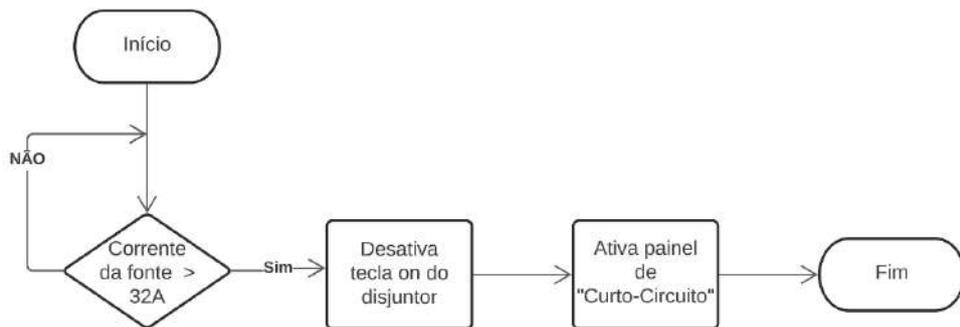
4.5.1.1.13 *CurtoCircuito*

Um ponto importante de análise e retorno do sistema é a ocorrência de curto-circuito na simulação, assim esta classe é responsável por verificar a corrente da fonte, esta, calculada pela equação de Tableau estando na matriz $w(t)$, sempre na posição $(n-1) + b$.

Como parâmetro será considerado correntes acima de 32 A como defeito, assim em consonância com a proteção instalada na bancada física, estando a fonte fornecendo valores

acima do estipulado será realizado os procedimentos de *feedback* com a atuação do dispositivo de proteção (disjuntor) e a sinalização visual da ocorrência através do painel de aviso, combinado a sonorização característica do defeito, sendo demonstrado o fluxo de verificação na Figura 45.

FIGURA 45 - FLUXOGRAMA DO MÉTODO *CURTOCIRC()*



Fonte: Elaboração própria (2023)

4.5.1.1.14 *Output*

Esta classe realiza as ações que ativam ou desativam as saídas de acordo com a prática selecionada (acende ou apaga a lâmpada, ativa ou desativa o áudio da campainha/cigarra e rotaciona o ponteiro do voltímetro) na simulação do circuito. Para realização destes procedimentos foi criado o método *OnOff()*, tendo como parâmetros o tipo de carga a ser acionada, o número de variáveis do circuito e a matriz $w(t)$ para verificação do módulo da corrente ou tensão dos ramos. Sendo a carga uma lâmpada é realizada a ativação da lâmpada acesa e desativação da lâmpada apagada através da propriedade *SetActive* dos objetos. No caso da carga ser um dispositivo sonoro é necessário a ativação do áudio inserido na cena. Para a prática da tomada a demonstração da correta ligação é através do ponteiro do voltímetro analógico, assim o método insere a nova rotação do ponteiro através do componente *Transform* adicionando um novo *Vector3* no *eulerAngles*.

4.5.1.1.15 *EqTableau*

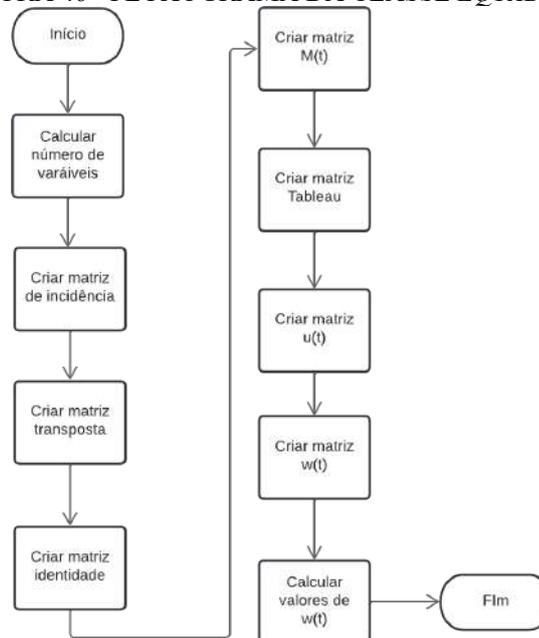
Esta classe é responsável pelo cálculo das correntes e tensões dos ramos em tempo de execução utilizando a equação de Tableau como método de análise do circuito, parte-se do cálculo do número de variáveis do circuito, estando diretamente relacionado ao número de ramos ativos no circuito, representado pela letra b , e o número de nós, representado pela letra n , sabendo que a matriz de Tableau será quadrada e terá o tamanho de $(n-1) + 2b$.

Com os valores de n e b calculados pelas classes *Nodes* e *Branch*, respectivamente, calcula-se o número de variáveis do circuito e gera-se as matrizes necessárias para o cálculo das correntes e tensões dos ramos, alterando os valores de entrada, também será recalculado os valores bases e geradas novas matrizes para a correta análise, onde mostra-se o fluxo de execução da classe na Figura 46.

Para realização das operações matriciais em C#, foi inserido o *framework Math.Net Numerics*, através do gerenciador de complementos NuGet para *Unity*.

Uma importante função desta classe é fornecer os valores das correntes dos ramos, sendo esses, responsáveis pela dinâmica do funcionamento dos circuitos.

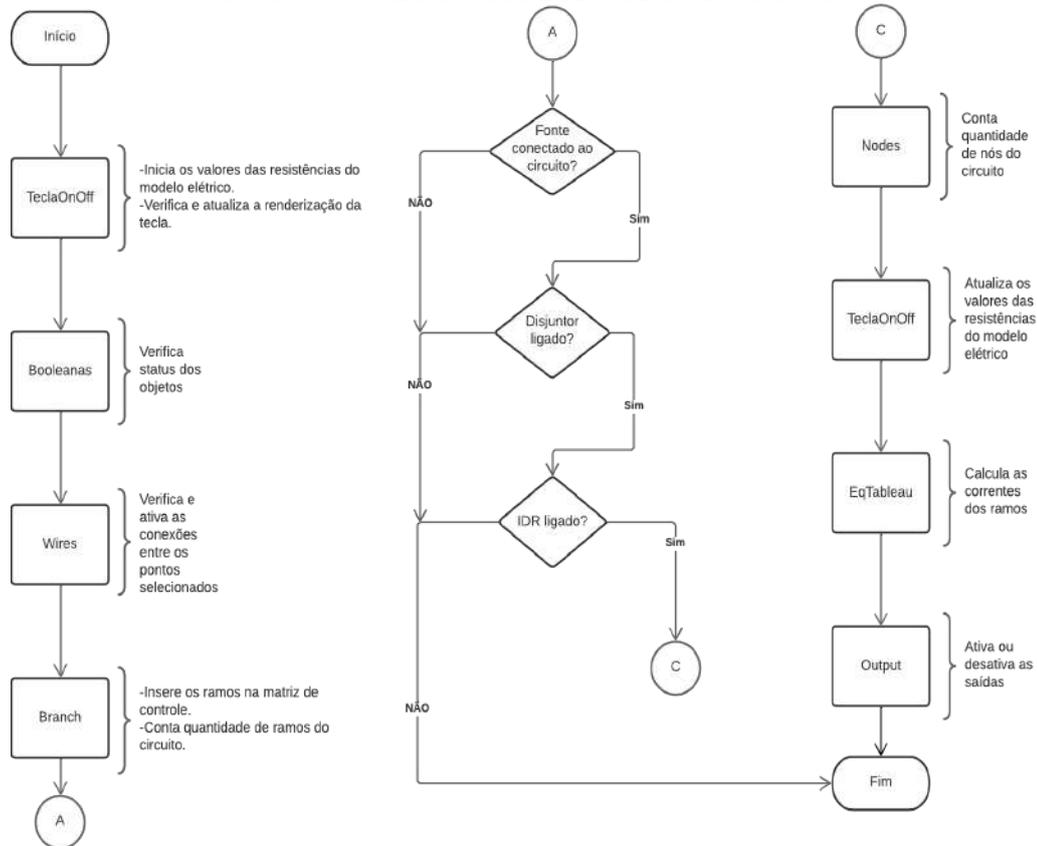
FIGURA 46 - FLUXOGRAMA DA CLASSE *EQTABLEAU*



Fonte: Elaboração própria (2023)

4.5.1.1.16 Pratica

A classe *Pratica* é responsável pela concentração dos métodos contidos nas classes anteriormente citadas, assim permitindo o correto funcionamento das simulações e permitindo o retorno adequado para cada ação realizada, ficando a cargo da principal integração do sistema, pois é responsável pela execução dos métodos a cada quadro (*frame*) da simulação e determina a sequência correta do algoritmo. Na Figura 47, podemos verificar o fluxograma que representa esta sequência.

FIGURA 47 - FLUXOGRAMA DA CLASSE *PRATICA*

Fonte: Elaboração própria (2023)

4.5.1.1.17 *IrParaCena*

Devido a construção dividida em cenas é fundamental poder realizar a mudança destas de acordo com a necessidade do usuário, assim foi desenvolvido a classe *IrParaCena*, onde através do método *ChamarCena()* é realizado o carregamento da cena previamente configurada no *inspector* da *Unity*, através de uma *string* pública, esta variável deve receber o nome da cena desejada, essas ações são possíveis devido a utilização do *SceneManager* do *Unity.Engine*. Também é utilizado nesta classe o método *Update()*, devido à necessidade de verificação em tempo de execução na demanda de mudança de ambiente na movimentação do usuário pelo corredor e laboratórios.

4.5.1.1.18 *SalvarPosic*

Outra necessidade de um sistema com múltiplas cenas é o de armazenar a última posição do personagem no momento de transição dos ambientes, para quando houver o retorno este esteja na mesma posição.

Assim a classe *SalvarPosic* através dos métodos *Awake()*, *Start()* e *SalvarLocalizacao()*, possibilita a correta transição de posição em cenas, onde o primeiro

método armazena o nome da cena atual na variável pública *nomeDaCena*, através do gerenciador de cenas do motor de jogos, este método sempre é executado antes do primeiro *frame* da cena que foi instanciado. No segundo método é verificado através do *PlayerPrefs.HasKey* se já houve o armazenamento da posição do personagem em outro momento, assim ocorrendo, este recebe a posição salva na chave. No último método é realizado a criação de 3 chaves para armazenamento da posição do personagem nas coordenadas x, y e z, através do *PlayerPrefs.SetFloat*, assim essas chaves são carregadas no método *Start()*, este método público pode ser chamado toda vez que há transição de cena e a classe deve ser adicionada como componente do personagem para correto funcionamento.

4.5.1.1.19 Bancada

Ao localizar a bancada didática no laboratório, o aluno deve clicar/tocar sobre ela, assim surgindo um menu suspenso com as práticas disponíveis para simulação, cabendo a classe *Bancada* o controle da ativação do *menu*, mas também as ações de bloqueio da câmera e da movimentação do personagem, permitindo uma melhor experiência do usuário, para retorno das funções bloqueadas é necessário o encerramento das opções de simulação.

Cabe a cada opção de simulação realizar o carregamento da cena desejada, para isto, é utilizado a classe *IrParaCena*, citada no item 4.5.1.17.

4.5.1.1.20 Trigger

Para detecção e controle de colisões que acionam funções do sistema, foi desenvolvido a classe *Trigger*, contendo os métodos *OnTriggerEnter()* e *OnTriggerExit()*, fornecido pelo motor de jogos, assim é possível armazenar em sua variável booleana *trigger* a ocorrência e o fim da colisão, podendo esta informação ser utilizada por outras classes do sistema.

4.5.1.1.21 Mensagem

Para realização da ativação e desativação das mensagens de aviso nas portas, janelas e da bancada didática, assim orientando as ações do discente no ambiente digital, foi criado a classe *Mensagem*, onde através da detecção de colisão realizada pela classe *Trigger* é acessado a propriedade *SetActive* da variável *aviso*, assim realizando a ação necessária.

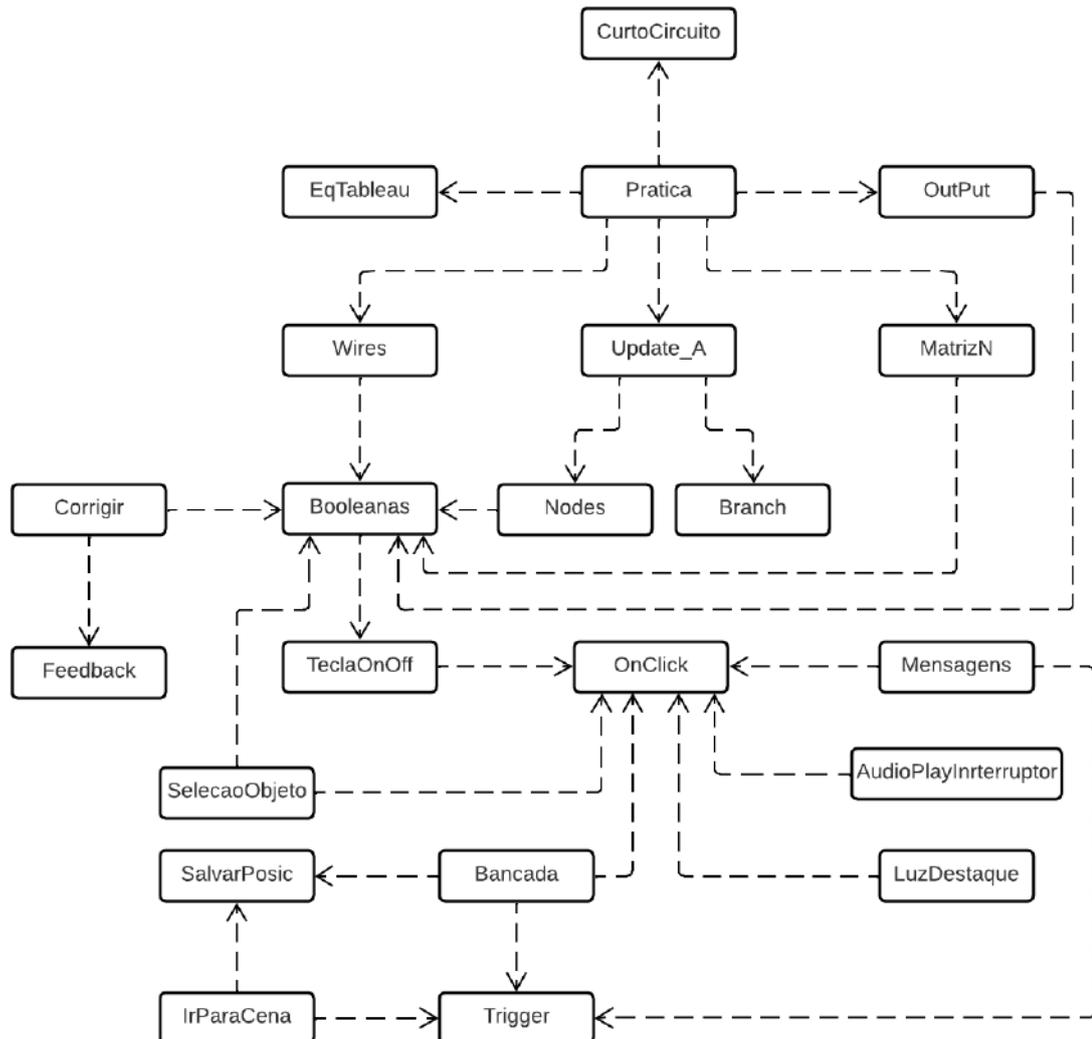
4.5.1.1.22 AudioPlayInterruptor

Como ponto de realismo da simulação foi incluído como componente de cada tecla utilizada na prática a sonorização característica de clique nestes objetos e o controle do acionamento

desta áudio é realizado por esta classe, de acordo com a booleana *audioplay* da classe *OnClick* também inserida como componente dos objetos. Desta forma, quando realizado o clique/toque na tecla a booleana *audioplay* recebe o valor de *true* e consequentemente o áudio é ativado através do método *GetComponent<AudioSource>().Play()*, também ocorrendo o retorno da booleana para o valor inicial *false*, evitando a contínua reprodução da sonorização.

Todas as classes demonstradas nesta seção possuem um relacionamento de dependência, sendo demonstrado na Figura 48, onde as linhas tracejadas representam o sentido unidirecional da dependência. O detalhamento do projeto pode ser visualizado no [Apêndice D](#), através do diagrama de classes, contendo os atributos e métodos para cada classe.

FIGURA 48 - RELACIONAMENTO ENTRE AS CLASSES



Fonte: Elaboração própria (2023)

4.5.1.2 Classes sem dependências

4.5.1.2.1 *Timer*

Devido a característica de transição em cenas da aplicação é necessário que o tempo decorrido durante a cena anterior seja armazenado e posteriormente carregada na nova cena, para isso é utilizado a classe *PlayerPrefs* do *Unity*, assim é criada uma chave para armazenamento do tempo decorrido na cena, ocorrendo a transição para um novo ambiente é verificado se existe a chave correspondente, havendo a chave armazenada, é carregado o tempo decorrido na cena anterior, não existindo a chave, é transcorrido a contabilização do tempo de forma progressiva partindo de 0 segundos. Essas ações ocorrem nas cenas referentes ao corredor do setor e nos laboratórios.

Como a característica de contabilização do tempo nas práticas ocorre de forma regressiva é inserido como variável booleana as opções de seleção no *Inspector* do *Unity* se a contagem deve ser progressiva ou regressiva, onde para a característica progressiva é armazenado na variável *timer_geral* o tempo contabilizado pelo *Time.deltaTime* utilizando o operador $+=$. Para a característica regressiva é armazenado na variável *timer_pratica* o tempo contabilizado pelo *Time.deltaTime* utilizando o operador $-=$, assim decrescendo o tempo previamente ajustado em 120 segundos. Quando o tempo é igualado a 0 a classe ativa o painel de aviso de tempo encerrado e finaliza a contagem.

O método *DesativaTimer()* é utilizado para interromper a contagem do tempo no momento de *feedback* de circuito correto e incorreto, para a segunda opção a cronometragem do tempo retorna quando o painel de aviso é fechado, através do método *AtivaTimer()*.

4.5.1.2.2 *Buttons*

Esta classe é responsável pelos métodos que fecham o programa após clique/toque no botão sair do menu principal e o método que exclui todas as chaves armazenadas através do *PlayerPrefs*, assim deletando as informações salvas ao longo da simulação.

4.5.1.2.3 *MoveZoom*

Para movimentação da câmera inserida nas cenas referentes às práticas, se fez necessário o desenvolvimento desta classe, onde através do método *Update()* verifica-se a ocorrência da entrada de comando, esta podendo ocorrer de duas formas: por meio dos botões do teclado e pelo toque em tela para a versão *Android*, assim sendo necessário duas análises distintas.

Para as versões destinadas a computadores, foram utilizados as diretivas *UNITY_STANDALONE* e *UNITY_WEBGL*, estas têm como função indicar qual trecho de

código deve ser compilado de acordo com a plataforma utilizada, assim permitindo a criação de um único *script* que pode ser utilizado nas diferentes plataformas. No trecho de código delimitado por estas diretivas, a entrada analisada é o teclado, assim, os botões destinados a movimentação do personagem são: deslocamento vertical fica a cargo dos botões W (para cima) e S (para baixo) e o deslocamento horizontal é realizado pelos botões D (para direita) e A (para esquerda) do teclado.

Para a versão *Android*, foi utilizado a diretiva `UNITY_ANDROID`, onde a forma de entrada para deslocamento da câmera é o toque em tela. Neste trecho de código delimitada pela diretiva, é realizado a verificação da ocorrência de toque em tela através do método `Input.touchCount`, sendo superior a 0, é indicado que o toque único em deslocamento (movimento de arrastar) deve ser analisado através do método `Input.GetTouch(0).Moved`.

Para ambas formas de entrada, o método realiza o deslocamento suave da câmera no sentido alvo, assim ocorrendo a alteração da posição no eixo desejado e mantendo-se a posição nos demais eixos.

4.5.1.2.4 *MusicPlayer*

Para ambientação do discente na simulação foi incluído uma sonorização em segundo plano, e devido a construção modularizada em cenas da aplicação, criou-se a necessidade de um gerenciador de áudio, que verifica o andamento do *playback* para que na mudança de cena estas configurações sejam mantidas e não tenha um reinício da música previamente iniciada em outra cena.

4.5.1.2.5 *VolumeControl*

Para ajuste do volume dos áudios inseridos no ambiente, esta classe foi criada, assim verificando a posição dos slides de volume e alterando de acordo com esta posição, também cabe a esta classe salvar os ajustes realizados pelo aluno e mantê-los caso haja mudança de cena.

4.5.1.2.6 *DestroyGameObject*

Esta classe tem como objetivo destruir os objetos que só podem ser visualizados uma única vez, assim não permitindo a criação de uma instância caso a cena seja carregada novamente, como exemplo deste tipo de dinâmica há o painel de boas vindas, visualizado no início da cena referente ao corredor do setor. Para isto, é verificado se há armazenado através do `PlayerPrefs` alguma chave de posição, assim indicando que esta cena já foi acessada

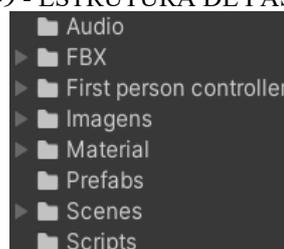
previamente e armazenada pela classe *SalvPosic*, havendo esta ocorrência, a classe identifica e destrói o objeto para não visualização do mesmo. Esta classe deve ser inserida como componente do objeto a ser destruído.

4.5.2 Implementação no motor de jogos *Unity*

No motor de jogos foram criados três projetos distintos com *template* 3D, o primeiro para configuração do ambiente *Standalone (Windows)*, o segundo para ambiente *Android*, e o terceiro para compilação para servidores *Web*. Apesar da estrutura do projeto ser a mesma, este procedimento se fez necessário devido aos ajustes de *layout* da aplicação, podendo haver variações para melhor utilização nos diferentes ambientes, tipos de *input* para controle do personagem diferenciados, além das características específicas de configuração para compilação do *software*.

Desta forma o projeto é organizado em pastas, estas servindo para alocação dos recursos necessários para aplicação. A pasta *Scenes* é a primeira a ser criada pelo motor de jogos, cabendo ao armazenamento das cenas construídas no ambiente. Na sequência foi inserido a pasta *FBX*, que foi responsável por armazenar todos os modelos 3D construídos e exportados do *Blender*. Outra pasta fundamental na dinâmica da aplicação é a *Script*, responsável pela organização de todos os códigos desenvolvidos para o simulador. Na pasta *Prefabs*, são incluídas estruturas modelos que serão reaproveitadas em cenas diferentes, como a base para os menus, o gerenciador de classes e áudio, assim evitando a nova construção em cada cena. As pastas *Audio* e *Imagens*, ficam os ativos correspondentes a cada nomenclatura. Outra pasta fundamental na organização é a *Material*, onde fica contido os materiais criados e inseridos nos modelos *FBX*, podendo ser extraídos destes e novamente vinculados ao objeto, assim evitando materiais repetidos e consequentemente reduzindo o tamanho da aplicação. Na pasta *First person controller* ficam os ativos do personagem em primeira pessoa, este podendo ser inserido no projeto através da *Asset Store* da *Unity* e contendo todos os recursos necessários para seu funcionamento, esta estrutura pode ser vista na Figura 49.

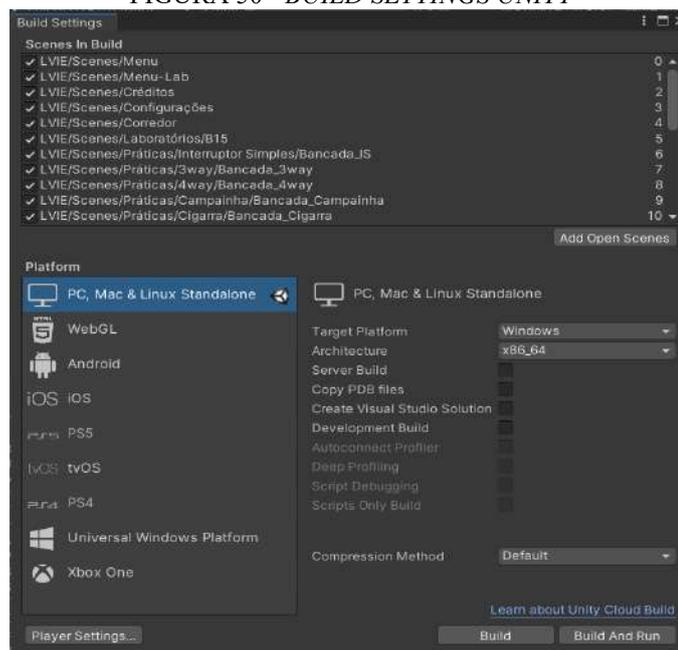
FIGURA 49 - ESTRUTURA DE PASTAS *UNITY*



Fonte: Elaboração própria (2023)

Para atender ao requisito de modularidade, o projeto foi dividido em 29 cenas e todas devem ser incluídos nas configurações de construção, para que possam ser efetivamente utilizadas, tanto após a criação do executável/instalador, quanto para a mudança das cenas no *play mode* do *Unity*, necessitando que a cena inicial seja a primeira da lista, como demonstrado na Figura 50.

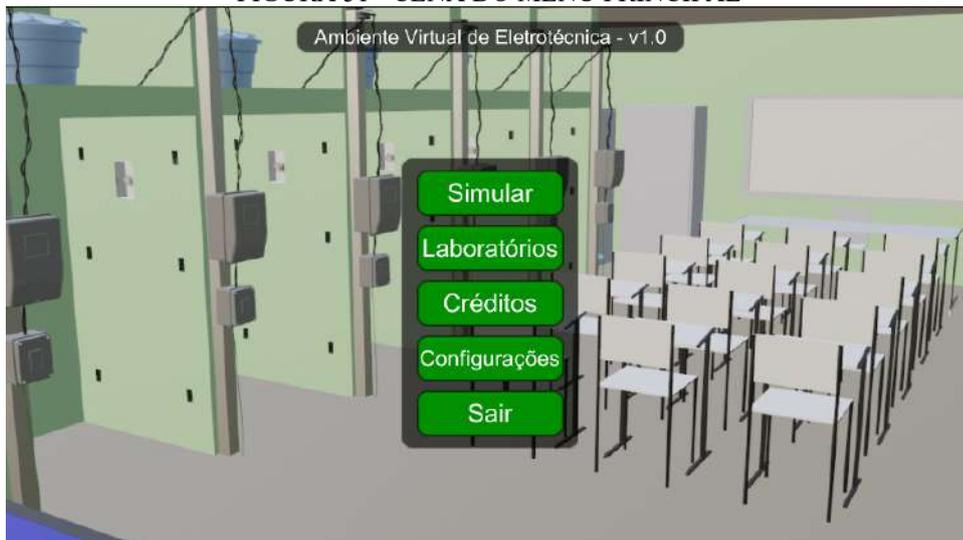
FIGURA 50 - BUILD SETTINGS UNITY



Fonte: Elaboração própria (2023)

Como primeira cena está o menu principal, construído através do componente *Canvas* e possuindo os botões necessários para o direcionamento das demais cenas do projeto, seguindo o fluxo demonstrado na Figura 7. Como plano de fundo da cena está a imagem do laboratório de instalações elétricas, previamente salva na pasta *Imagens*, sendo necessário a mudança do tipo da textura para *Sprite*. Devido ao uso dos componentes *Button* na cena, também deve-se inserir o componente *EventSystem*, este sendo responsável pela detecção do evento de clique neste elemento de UI, o menu pode ser visualizado na Figura 51.

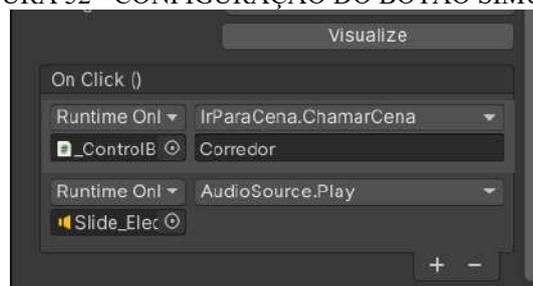
FIGURA 51 - CENA DO MENU PRINCIPAL



Fonte: Elaboração própria (2023)

Para realização da mudança de cena após o clique/toque no botão do menu, foi criado um *GameObject* vazio e inserido como componente o *script IrParaCena*, este *GameObject* tem como objetivo gerenciar e instanciar a classe em cena e foi utilizada para referência nos botões, assim permitindo a utilização do método *ChamarCena()*, como pode ser visto na Figura 67. Em todas as cenas que possuem botões que acionam a mudança de cena foi realizado o mesmo procedimento, além da inclusão de áudio de interação com o objeto.

FIGURA 52 - CONFIGURAÇÃO DO BOTÃO SIMULAR



Fonte: Elaboração própria (2023)

Na opção *Simular* o aluno é direcionado para o corredor de circulação da área elétrica, sendo indicado o objetivo da simulação, cabendo ao mesmo o início da movimentação e busca do laboratório alvo do projeto. Neste ambiente o aluno terá uma perspectiva de visão em primeira pessoa, como demonstrado na Figura 53a versão para computador/*Web* e na Figura 53b a versão para *Android*. Para facilitar o uso nas diferentes plataformas, os botões responsáveis pelas ações do personagem foram nomeados de acordo com as teclas do computador. No painel destinado a contabilização do tempo decorrido, é adicionado como

componente o *script* que contém a classe *Timer*, assim sendo possível instanciar a classe em cena. Nesta cena o cronômetro será progressivo, sendo ajustado no *inspector* do *Unity*.

FIGURA 53 - TELA INICIAL DO CORREDOR DE CIRCULAÇÃO



(a) Versão para computador/*Web*

(b) Versão *Android*

Fonte: Elaboração própria (2023)

No botão *Laboratórios* o aluno pode visualizar os nomes dos ambientes, sua numeração, como também ser direcionado diretamente ao ambiente desejado demonstrado na Figura 54. Nesta cena está inserido nos botões referentes aos laboratórios as classes *Mensagem* e *OnClick*. Para direcionamento à cena desejada é instanciado a classe *IrParaCena* como componente do objeto no objeto gerenciador, assim possibilitando a utilização do método *ChamarCena()* nos botões da cena.

FIGURA 54 - TELA DE VISUALIZAÇÃO E NOMENCLATURA DOS LABORATÓRIOS

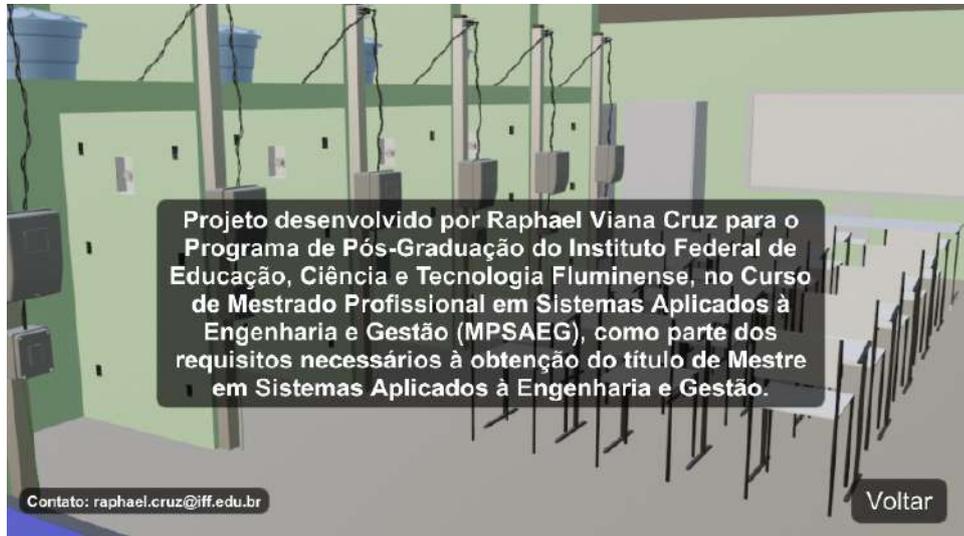


Fonte: Elaboração própria (2023)

A opção *Créditos* contém a indicação do autor do projeto, sua finalidade e o contato para sugestões de melhoria, visualizado na Figura 55. Nesta cena também está instanciado a

classe *IrParaCena* como componente do objeto gerenciador, assim possibilitando a utilização do método *ChamarCena()* no botão *Voltar*.

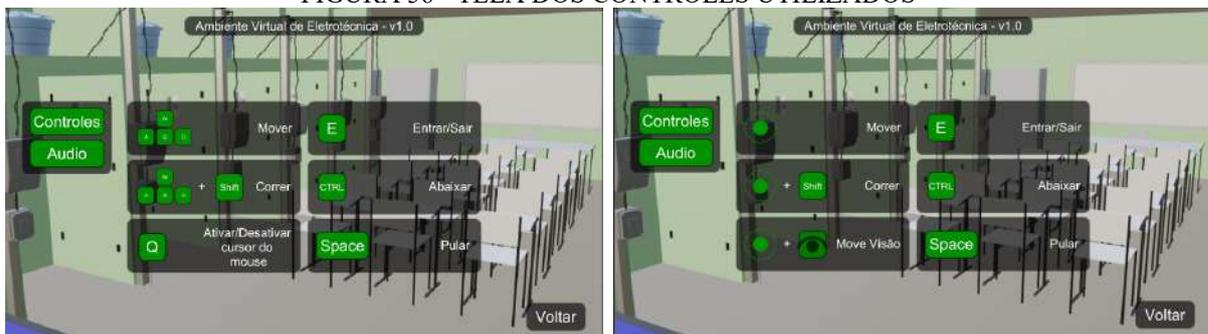
FIGURA 55 - TELA DE CRÉDITOS



Fonte: Elaboração própria (2023)

No botão *Configurações*, o jogador pode acessar a indicação de cada função dos botões de controle, onde na Figura 56a temos a tela da versão para computadores/*Web* e na Figura 56b a tela da versão *Android*. Nesta cena também pode-se ajustar o volume do som total do jogo no *slider* denominado *Master*, para ajuste do volume da trilha sonora de fundo através do *slider Background* e controlar o volume dos sons de movimentação do *player* e sons ambientes através do *slider Player*, demonstrado na Figura 57. Ao clicar nos botões disponibilizados como opção para cena, é realizada a ativação do painel que contém as informações/configurações desejadas e a desativação do painel não desejado, este procedimento foi configurado no método *OnClick()* dos botões *Controles* e *Audio*.

FIGURA 56 - TELA DOS CONTROLES UTILIZADOS

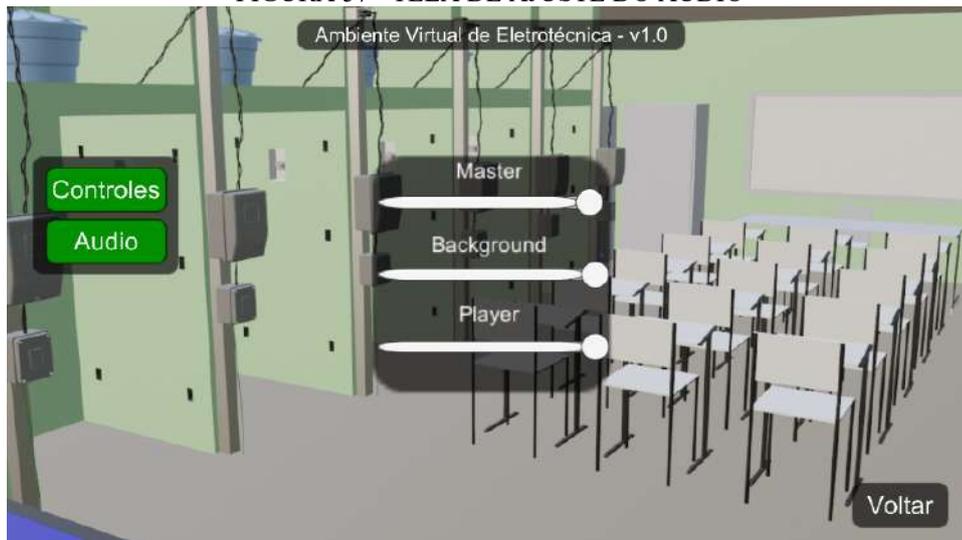


(a) Versão para computador/*Web*

(b) Versão *Android*

Fonte: Elaboração própria (2023)

FIGURA 57 - TELA DE AJUSTE DO ÁUDIO

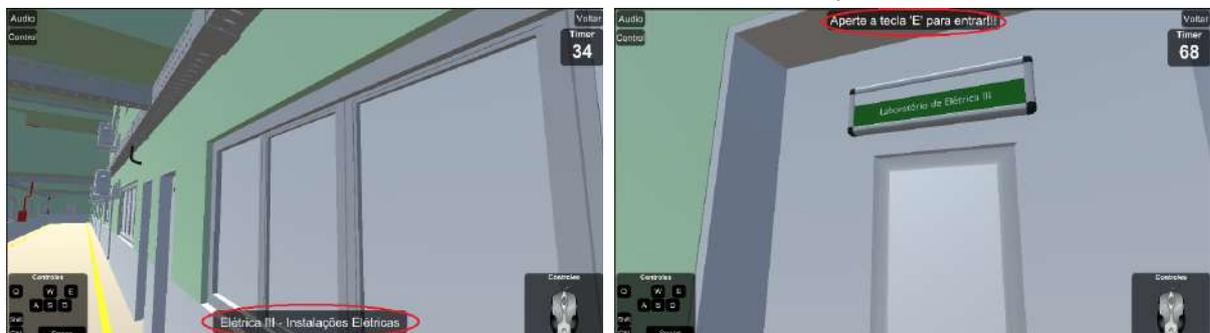


Fonte: Elaboração própria (2023)

No botão *Voltar* é referenciado o objeto responsável pelo gerenciamento da classe *IrParaCena*, sendo possível acessar o método público *ChamarCena()*, direcionando para a cena com o menu principal. Este procedimento foi realizado em todos os botões de mesmo nome, apenas ocorrendo a diferenciação da cena de retorno, de acordo com o ambiente ativo.

Após circulação nos corredores o aluno é direcionado através de sinalizações do sistema, demonstrado e em destaque na Figura 58a, possibilitando a localização do laboratório de Instalações Elétricas, desta forma, o aluno deve acessá-lo como indicativo no destaque da Figura 58b, assim ocorrendo a transição para a cena correspondente. Outra possibilidade é o acesso aos outros 18 ambientes contidos no setor, onde é possível a visualização destes ambientes no [Apêndice E](#).

FIGURA 58 - ACESSO AO LABORATÓRIO DE INSTALAÇÕES ELÉTRICAS

(a) *Feedback da janela*(b) *Feedback da porta*

Fonte: Elaboração própria (2023)

Ao acessar a cena que representa o ambiente laboratorial, o aluno deve se direcionar até a bancada para realizar as simulações utilizadas em aula como demonstrado na Figura 59.

Quando localizado e posicionado a frente da bancada, o discente deve clicar/tocar na bancada, ativando o *menu* de seleção com as prática disponíveis, cabendo o gerenciamento a classe *Bancada*, onde mostra-se na Figura 60 as opções de montagem. Cada botão do menu de circuito é responsável por chamar a cena referente a sua prática, este procedimento é realizado pela classe *IrParaCena*, através de seu método *ChamarCena()*, estando instanciado no objeto gerenciador.

FIGURA 59 - LABORATÓRIO DE INSTALAÇÕES ELÉTRICAS



Fonte: Elaboração própria (2023)

FIGURA 60 - MENU DE SELEÇÃO DE PRÁTICA

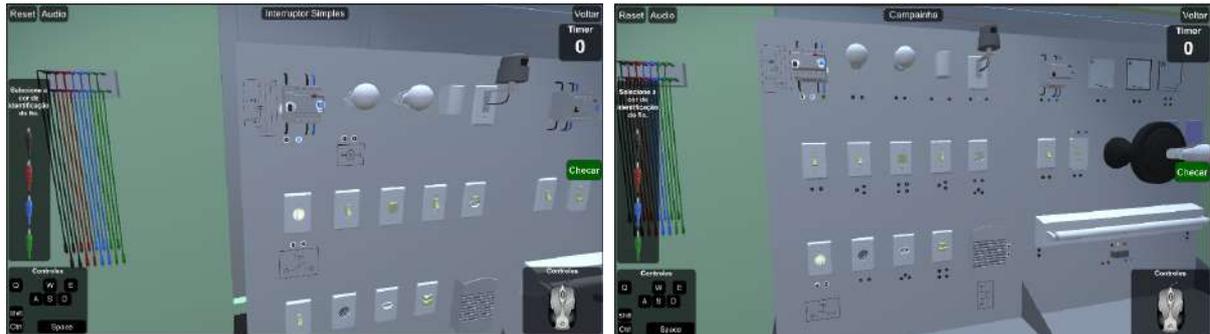


Fonte: Elaboração própria (2023)

Ao acessar a prática do Interruptor Simples (Figura 61a), Campainha (Figura 61b), Cigarra (Figura 61c), Interruptor Paralelo (Figura 61d), Interruptor Intermediário (Figura 61e) ou da Tomada 2P+T (Figura 61f), há na parte superior central há a indicação do circuito

escolhido, mantendo-se na parte superior, porém, à esquerda, há duas opções, a primeira tem como função reiniciar a simulação para correção de algum erro e a segunda traz novamente os controles de ajuste de volume dos áudios que compõem o ambiente. Direcionando para a parte superior direita, temos a opção de retorno ao laboratório e logo abaixo, o *timer* regressivo, estando ajustado em 120 segundos em todas as simulações.

FIGURA 61 - CENA COM A SIMULAÇÃO DAS PRÁTICAS



(a) Interruptor Simples

(b) Campainha



(c) Cigarra



(d) Interruptor Paralelo



(e) Interruptor Intermediário



(f) Tomada 2P+T

Fonte: Elaboração própria (2023)

No ambiente de simulação do circuito selecionado pelo discente, as partes passíveis de interação ficam destacadas com um foco de luz e após sua seleção a luz se apaga, indicando a ação ocorrida, sendo esta ação executada pela classe *LuzDestaque*. Outro ponto de *feedback* é a indicação da simbologia dos dispositivos a serem utilizados, trazendo mais um ponto de

direcionamento para o aluno. Na parte lateral esquerda fica o painel de seleção de cores dos fios de conexão entre os bornes selecionados, cabendo a cada botão executar o método referente sua cor, estes instanciando a classe *SelecaoObjeto*, sendo esta indicação parte do processo de avaliação do circuito. Como exemplo, mostra-se na Figura 62 a prática do interruptor simples, estando em destaque a conexão com material genérico.

FIGURA 62 - CONEXÃO SEM MATERIAL DEFINIDO



Fonte: Elaboração própria (2023)

Após correta montagem e acionamento das proteções (disjuntor e IDR), é possível a verificação do funcionamento do circuito do Interruptor Simples (Figura 63a e 63b), Interruptor Paralelo (Figura 63c e 63d), Interruptor Intermediário (Figura 63e e 63f), Campainha (Figura 63g), Cigarra (Figura 63h) ou Tomada 2P+T (Figura 63i e 63j).

FIGURA 63 - CIRCUITO CONECTADO E EM FUNCIONAMENTO (CONTINUA)



(a) Interruptor Simples montado

(b) Interruptor Simples aceso

FIGURA 63 - CIRCUITO CONECTADO E EM FUNCIONAMENTO (CONCLUSÃO)



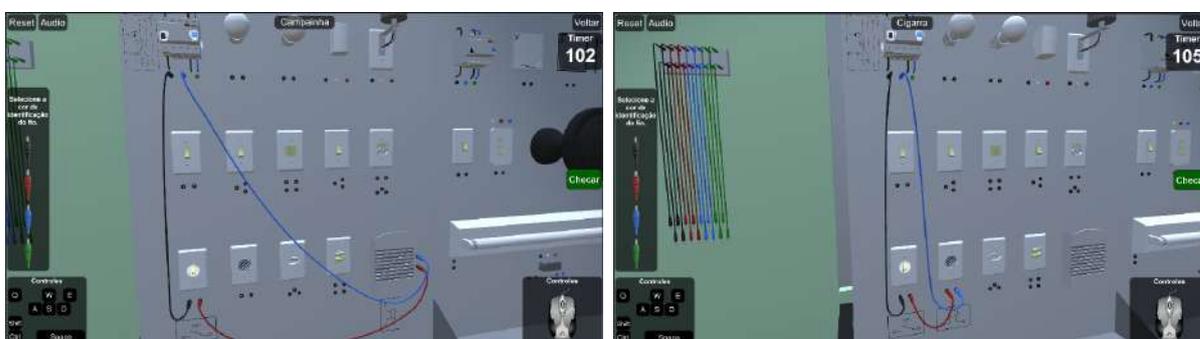
(c) Interruptor Paralelo montado

(d) Interruptor Paralelo aceso



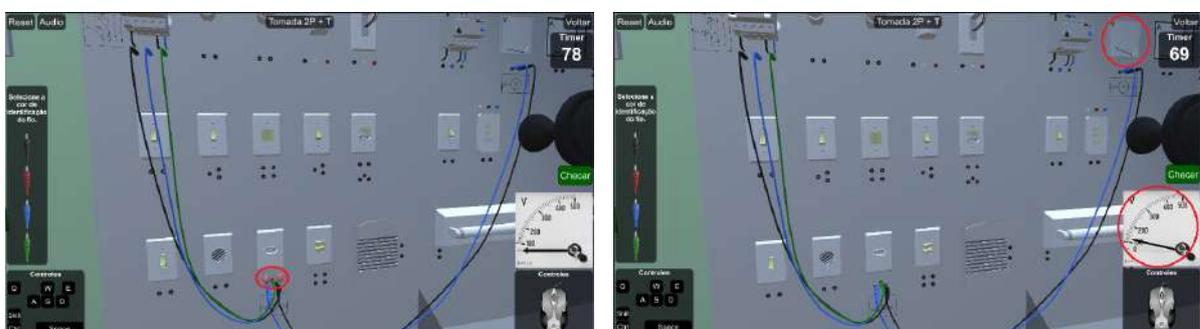
(e) Interruptor Intermediário montado

(f) Interruptor Intermediário aceso



(g) Campainha montada

(h) Cigarra montada



(i) Tomada 2P+T montada

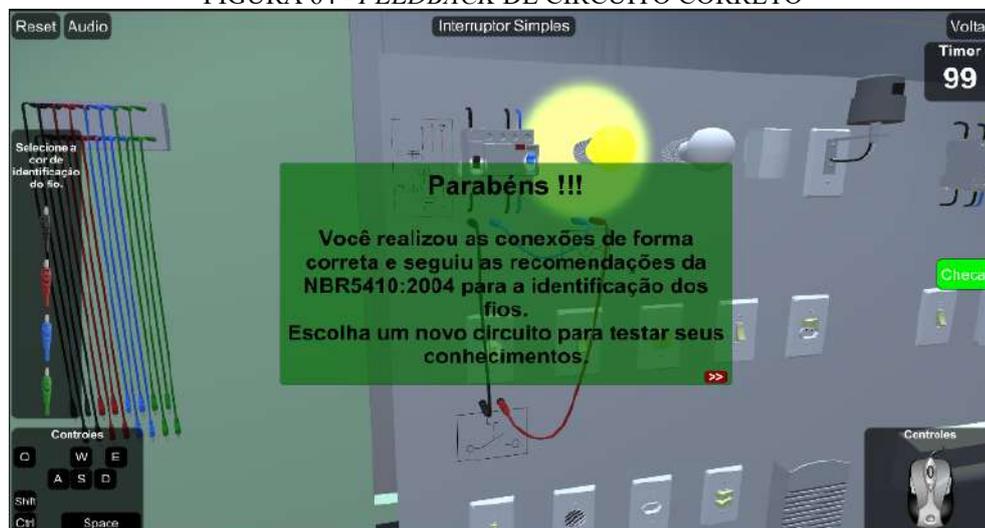
(j) Tomada 2P+T energizada

Fonte: Elaboração própria (2023)

Caso as conexões e cores estejam corretos, o aluno recebe um *feedback* positivo após o clique/toque no botão *Checar*, através do painel demonstrado na Figura 64, e o correto

diagrama do circuito de ligação, demonstrado na Figura 65a do *Interruptor Simples*, Figura 65b da *Campainha*, Figura 65c da *Cigarras*, Figura 65d do *Interruptor Paralelo*, Figura 65e do *Interruptor Intermediário* e Figura 65f da *Tomada 2P + T*.

FIGURA 64 - FEEDBACK DE CIRCUITO CORRETO



Fonte: Elaboração própria (2023)

FIGURA 65 - FEEDBACK DO DIAGRAMA (CONTINUA)



(a) Interruptor Simples

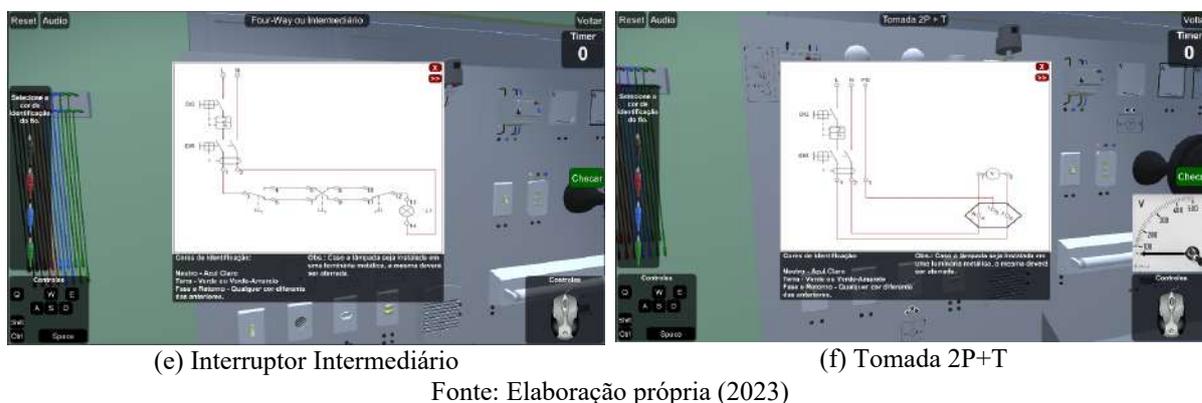
(b) Campainha



(c) Cigarras

(d) Interruptor Paralelo

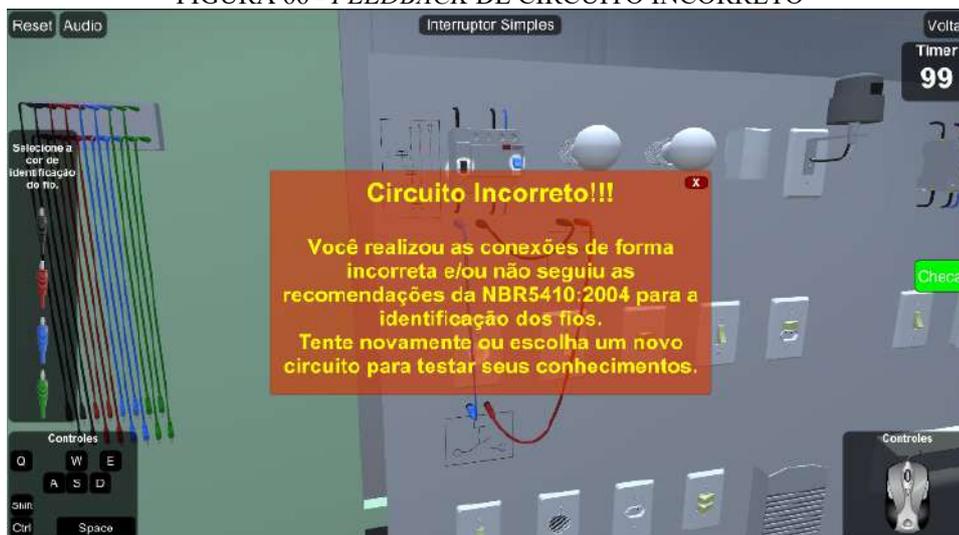
FIGURA 66 - FEEDBACK DO DIAGRAMA (CONCLUSÃO)



Para que o cronômetro interrompa sua regressão, o usuário deve clicar/tocar no botão *Checar*, assim desativando a classe *Timer*.

Outra situação possível é o erro nas conexões ou utilização inadequada das cores dos fios, assim ocorrendo, o aluno receberá uma indicação do erro e consequentemente a sugestão de uma nova tentativa para montagem da ligação dentro da norma vigente, como demonstrado na Figura 66. Optou-se pela não indicação explícita do erro, assim estimulando a pesquisa e consulta da norma e do circuito, fortalecendo o aprendizado do discente.

FIGURA 66 - FEEDBACK DE CIRCUITO INCORRETO



Ocorrendo um fechamento que cause curto-circuito, o aluno receberá um retorno através de um painel informando a ocorrência do defeito e o desarme da proteção responsável por salvaguardar o circuito em conjunto com a sonorização característica desta falha, como pode ser visualizado na Figura 67 e em destaque a proteção desligada.

FIGURA 67 - FEEDBACK DE CURTO-CIRCUITO



Fonte: Elaboração própria (2023)

Caso o tempo programado para montagem da prática encerre, é ativado o painel informativo da ocorrência, sendo bloqueado a movimentação da câmera e reiniciando a cena após fechamento do painel, mostra-se na Figura 68 a ação realizada ao término do tempo, em destaque o cronômetro regressivo zerado.

FIGURA 68 - FEEDBACK DE TEMPO



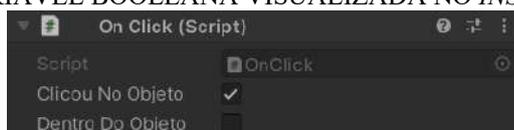
Fonte: Elaboração própria (2023)

4.6 Testes Unitários

Os testes unitários foram realizados de forma manual de acordo com cada classe e seus métodos, assim ocorrendo a verificação de todas as ações executadas nos *scripts*, podendo ser visualizado o detalhamento dos testes no [Apêndice F](#).

Para verificação das mudanças das variáveis foi realizada a observação pelo *inspector* do motor de jogos, onde é possível visualizar as propriedades dos objetos em cena, após sua seleção dentro da hierarquia da aplicação, na Figura 69 pode-se visualizar um exemplo de verificação da mudança da booleana *ClicouNoObjeto* para *true*, assim aparecendo em sua *checkbox* como marcada.

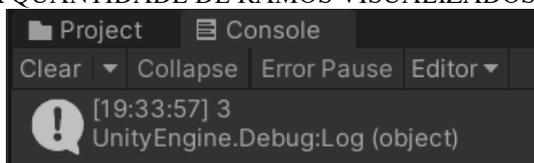
FIGURA 69 - VARIÁVEL BOOLEANA VISUALIZADA NO *INSPECTOR* DO *UNITY*



Fonte: Elaboração própria (2023)

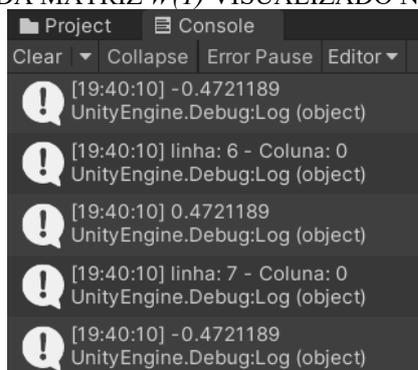
Também foi utilizado o método *Debug.Log()*, assim possibilitando a visualização das matrizes instanciadas, quantidade de nós, ramos do circuito montado e qualquer unidade a ser testada e validada no console do motor de jogos, na Figura 70 mostra-se a quantidade de ramos ativos na simulação e na Figura 71 pode ser visualizado parte da matriz $w(t)$ gerada na prática do *Interruptor Simples* com a lâmpada acesa e de acordo com circuito da Figura 14.

FIGURA 70 - *DEBUG* DA QUANTIDADE DE RAMOS VISUALIZADOS NO *CONSOLE* DO *UNITY*



Fonte: Elaboração própria (2023)

FIGURA 71 - *DEBUG* DA MATRIZ $W(T)$ VISUALIZADO NO *CONSOLE* DO *UNITY*



Fonte: Elaboração própria (2023)

Para realização dos testes da versão para *Android*, foi instalado em dispositivo móvel o aplicativo *Unity Remote 5*, assim possibilitando o controle do personagem e verificação das ações após toque na tela, através de conexão pela porta serial universal (*Universal Serial Bus*

– USB). Para possibilitar esta utilização o celular deve estar com o modo de depuração USB ativado. Esta ação foi realizada para evitar a construção do arquivo de instalação após cada alteração ocorrida devido aos testes realizados, assim otimizando o processo de teste.

Para os testes da versão *Web*, foi criado um servidor local através do *prompt* de comando usando *Python* na versão 3.10.5, após seleção da pasta foi digitado o comando *python -m http.server*, assim permitindo o acesso por navegador *web* do arquivo *.html* gerado na construção do programa pelo *Unity*. No navegador foi digitado o endereço *http://localhost:8000*, assim possibilitando o acesso à pasta contendo o arquivo a ser testado.

4.7 Teste do Subsistema

4.7.1 Subsistema Corredor

Neste subsistema foram realizados os testes de colisão, de movimentação do personagem, de contagem progressiva do tempo transcorrido, de ativação e desativação dos avisos contidos no ambiente e de falhas graves contidas nas renderizações dos objetos modelados e inseridos no motor de jogos. No Quadro 7 é possível a visualização dos testes realizados para este subsistema, ocorrendo a aprovação em todos os casos.

QUADRO 7 - TESTES REALIZADOS NO SUBSISTEMA *CORREDOR*

Componente	Teste realizado	Situação
Colisores	Verificação dos colisores das paredes, pisos, portas, armários, bancos do jardim de inverno e escada.	ok
Timer	Verificação da contagem progressiva.	ok
Trigger	Verificação da ativação dos painéis de aviso nas portas e janelas.	ok
Mesh	Verificação da renderização dos objetos em cena.	ok
Player	Verificação da movimentação do personagem em primeira pessoa.	ok

Fonte: Elaboração própria (2023)

4.7.2 Subsistema Laboratórios

O subsistema laboratórios contém o *menu* de direcionamento aos ambientes laboratórias e suas respectivas cenas correlatas, assim ocorrendo os testes demonstrado no Quadro 8, sendo verificado as ações de indicação da nomenclatura dos ambientes para cada botão em cena.

Para análise das cenas que replicam os laboratórios, foram realizados os testes de colisão, de movimentação do personagem, de contagem progressiva do tempo transcorrido, de ativação e desativação dos avisos contidos no ambiente e de falhas graves contidas nas renderizações dos objetos modelados e inseridos no motor de jogos.

QUADRO 8 - TESTES REALIZADOS NO SUBSISTEMA *LABORATÓRIOS*

Componente	Teste realizado	Situação
Menu de laboratórios	Verificação da indicação do nome do laboratório após posicionamento do mouse sobre os botões.	ok
Colisor	Verificação dos colisores das paredes, pisos, portas e móveis.	ok
Timer	Verificação da contagem progressiva.	ok
Trigger	Verificação da ativação e desativação dos painéis de aviso nas portas e na bancada didática.	ok
Mesh	Verificação da renderização dos objetos em cena.	ok
Player	Verificação da movimentação do personagem em primeira pessoa.	ok

Fonte: Elaboração própria (2023)

4.7.3 Subsistema Simulação das Práticas

Este subsistema é responsável por rodar todas as práticas contidas no escopo do projeto, assim, foram realizados os testes demonstrados no Quadro 9, que verificam os processos contidos em cada cena, também cabendo a avaliação das matrizes geradas após a montagem para cada circuito modelado tendo como referência os cálculos realizados e demonstrado na Tabela 4.

QUADRO 9 - TESTES REALIZADOS NO SUBSISTEMA *SIMULAÇÃO DAS PRÁTICAS*

Práticas	Teste realizado	Situação
Interruptor Simples, Paralelo, Intermediário, Campainha, Cigarra e Tomada 2P+T.	Verificação da renderização dos objetos em cena.	ok
	Verificação dos colisores dos soquetes, teclas e fios.	ok
	Verificação das luzes de destaque.	ok
	Verificação dos circuitos corretos.	ok
	Verificação dos circuitos incorretos.	ok
	Verificação de curto-circuito.	ok
	Verificação do <i>reset</i> da cena.	ok
	Verificação da movimentação e zoom da câmera.	ok
	Verificação do cronômetro regressivo.	ok
	Verificação do <i>feedback</i> de tempo.	ok
	Verificação da mudança de cor dos fios.	ok
	Verificação dos botões que fecham os painéis de <i>feedback</i> .	ok
Verificação da matriz $w(t)$ de acordo com o circuito montado.	ok	

Fonte: Elaboração própria (2023)

4.7.4 Subsistema Controles

Este subsistema é composto pelo painel que indica a funcionalidade de cada botão para deslocamento e interação com o ambiente, desta forma, foi verificado a ativação do painel contendo as funções dos botões de controle do personagem, assim como sua correta relação com sua aplicação, sendo aprovado nestas avaliações.

4.7.5 Subsistema Áudio

O subsistema de áudio é responsável pelo gerenciamento do volume dos componentes sonoros em cena e pode ser acessado pelos subsistemas corredor, laboratórios e simulação das práticas, assim ocorrendo a verificação da variação do volume para cada cena. Estes ajustes podem ocorrer em menu específico de configuração acessado através da cena inicial do ambiente virtual ou pelas 25 cenas contidas nos subsistemas citados anteriormente, sendo aprovado em todos os casos de teste relacionados no Quadro 10.

Outra importante funcionalidade deste subsistema é armazenar o valor do volume ajustado pelo usuário e carregá-lo na transição entre as cenas, este caso de teste foi aplicado nos testes de integração entre os sistemas, assim demonstrado posteriormente no item 4.8.

QUADRO 10 - TESTES REALIZADOS NO SUBSISTEMA ÁUDIO

Cenas	Teste realizado	Situação
Menu de Áudio, laboratórios B10, B13, B15, B16, B17, B18, B19a, B19b, B20, B21, B22a, B22b, B22b-postes, B23, B110, B111, B112, B118-1, B118-2, prática do interruptor simples, paralelo, intermediário, campainha, cigarra e tomada 2P+T.	Verificação do ajuste do volume da sonorização de fundo (<i>background</i>) do sistema.	ok
	Verificação do ajuste do volume geral (<i>Master</i>) do sistema.	ok
	Verificação do ajuste do volume do personagem.	ok

Fonte: Elaboração própria (2023)

4.8 Teste de Sistemas

4.8.1 Sistema Ambiente em 1ª pessoa

Para verificação deste sistema foi realizado o teste de integração entre os ambientes desenvolvidos, assim testando a transição entre a cena do Corredor para os laboratórios que compõem o setor, transição entre a cena do laboratório de instalações elétricas para as práticas implementadas e vice e versa, também ocorrendo a verificação do armazenamento e carregamento do tempo, posição e volume do áudio, tendo aprovação em todas as situações analisadas, sendo demonstrado os casos de testes no Quadro 11.

QUADRO 11 - TESTES REALIZADOS NO SISTEMA EM 1ª PESSOA (CONTINUA)

Subsistema	Cena	Teste realizado	Situação
Corredor	Corredor	Verificação da transição para os laboratórios B10, B13, B15, B16, B17, B18, B19a, B20, B21, B22a, B22b, B23, B110, B111, B112, B118-1 e B118-2 com o tempo armazenado e volume do áudio armazenado.	ok
		Verificação da ativação das configurações de áudio.	ok
		Verificação da ativação ao painel de função dos botões de controle.	ok

QUADRO 11 - TESTES REALIZADOS NO SISTEMA EM 1ª PESSOA (CONCLUSÃO)

Subsistema	Cena	Teste realizado	Situação	
Laboratórios	Laboratórios B10, B13, B15, B16, B17, B18, B19a, B20, B21, B22a, B22b, B23, B110, B111, B112, B118-1 e B118-2.	Verificação da transição para o corredor na posição, tempo e volume do áudio armazenado.	ok	
	Laboratório B15	Verificação da transição para a prática do interruptor simples, campainha, cigarra, interruptor paralelo, interruptor intermediário e tomada 2P+T.	ok	
	Laboratório B19a	Verificação da transição para o laboratório B19b com o tempo e volume do áudio armazenado.	ok	
	Laboratórios B20 e B22a.	Verificação da transição para a área dos postes do laboratório B22b com o tempo e volume do áudio armazenado.	ok	
	Laboratório B22b		Verificação da transição para o laboratório B22a com o tempo e volume do áudio armazenado.	ok
			Verificação da transição para a área dos postes do laboratório B22b com o tempo e volume do áudio armazenado.	ok
Simulação das Práticas	Prática Interruptor Simples, Paralelo, Intermediário, Campainha, Cigarra e Tomada 2P+T.	Verificação da transição para o laboratório B15 na posição, tempo e volume do áudio armazenado.	ok	

Fonte: Elaboração própria (2023)

4.8.2 Sistema Configurações

Este sistema é composto pelos subsistemas de Controles e Áudio, ocorrendo a integração mútua, assim, para os testes foram realizados a verificação da ativação do painel de controles e de ajuste de volume dos áudios, ocorrendo a aprovação nos testes de transição entre os subsistemas.

4.9 Teste de Aceitação

O teste de aceitação, de acordo com Sommerville (2011), também nomeado de teste alfa, verifica se os requisitos elencados pelo cliente estão de acordo com o desenvolvimento apresentado, se relacionando com a etapa de ConOps do modelo V.

De acordo com esses parâmetros o programa cumpre com o requisito de construção para múltiplas plataformas, sendo construído em versões para computador, podendo ser instalado em SO *Windows*, mas também permitindo a utilização via plataforma *web*, sem necessidade de instalação por parte do aluno. Para aplicação em dispositivos móveis o ambiente virtual também foi construído para o SO *Android*, assim possibilitando a utilização de mais de 90% dos discentes matriculados no instituto e permitindo a repetição das simulações.

Para cumprir a necessidade de familiaridade com os ambientes do setor, foram criados modelos 3D utilizando como base a planta baixa e fotos do bloco onde se localizam os laboratórios para aplicação de detalhes similares aos visualizados nos ambientes. Outro ponto de réplica foi a construção do modelo da bancada didática de acordo com os detalhes de projeto e montagem do recurso utilizado na rotina de demonstração e testes dos circuitos ministrados na disciplina.

Estes ambientes foram divididos em 30 cenas, onde 4 tem a função de menus do sistema, 20 ambientes de circulação em 1ª pessoa e 6 práticas simuladas, assim permitindo a ampliação do escopo de práticas, atualização dos ambientes para adequação com as novas configurações físicas e devida correção necessária. Esta divisão em cenas distintas, torna o programa modular, cabendo a modificação quando necessária, apenas a cena alvo da operação, não necessitando modificação dos demais ambientes.

Foi incluído *feedbacks* de direcionamento do objetivo da aplicação, ações necessárias para continuidade da simulação, nomenclatura dos ambientes, indicação da situação das práticas realizadas, término do tempo e ocorrência de falha do circuito montado pelo aluno. Também foi incluído nas cenas referentes às práticas dos circuitos pontos focais de luz, para indicação dos dispositivos passíveis de interação, assim atendendo outro ponto apresentado na primeira etapa do modelo V.

A inclusão de sonorização ambiente foi destacada para maior integração do discente com o ambiente, mas também sendo necessário para demonstração de correto funcionamento dos dispositivos sonoros simulados em bancada. Assim foram incluídos sons de fundo, movimentação do personagem, interação com os elementos de UI, indicativos de correta ligação do circuito, falha crítica no circuito e finalização do tempo de montagem da prática.

Todos esses áudios podem ser controlados e ajustados através do menu de configurações, atendendo outra demanda requisitada.

A necessidade de desenvolvimento sem oneração da instituição se cumpriu com a utilização dos programas Blender, *Unity*, Gimp, CadeSimu e Scilab, estes, se enquadrando nas versões gratuitas ou *opensource*.

Um importante requisito estipulado foi a possibilidade de migração para novos ambientes emergentes, sendo atendido pela utilização do motor de jogos *Unity*, onde há a possibilidade instalação de complementos para utilização em RV e em modo *multiplayer*, assim permitindo a ampliação do ambiente para o metaverso.

De acordo com os requisitos propostos na concepção do programa, foi verificado que foi atendido às necessidades iniciais, podendo ser considerado o Laboratório Virtual de Instalações Elétricas como aprovado no teste de aceitação.

5 CONCLUSÃO

Este trabalho teve como objeto o desenvolvimento de um laboratório virtual multiplataforma para aplicação na disciplina de Instalações Elétricas de Baixa Tensão, ministrada nos cursos técnicos de nível médio em Eletrotécnica e no curso superior em Engenharia Elétrica, onde se replica o setor localizado no Instituto Federal Fluminense – *campus* Campos Centro com o intuito de possibilitar o aprimoramento dos conceitos demonstrado em laboratório por parte do corpo discente da instituição.

Como metodologia de desenvolvimento foi utilizado o modelo V, contendo 5 etapas de planejamento e desenvolvimento associados a 4 etapas de testes para verificação e validação do projeto, trazendo organização na concepção da solução e padrão no sistema de avaliação do simulador.

Para viabilização do objetivo proposto foi utilizado o programa *Blender* na modelagem 3D de 1 ambiente comum de circulação, 19 laboratórios, 1 bancada didática e 6 circuitos práticos. Para aplicação dos modelos projetados e implementação das dinâmicas do simulador foi utilizado motor de jogos *Unity*, este fornecendo recursos como: elementos de interface de usuário, aplicação de física em objetos, assim replicando comportamentos reais, ampla loja de ativos que aceleram o desenvolvimento do programa e capacidade de compilação para plataformas distintas.

Devido à necessidade e não existência dentro do motor de jogos *Unity* de um módulo para análise dos circuitos elétricos montados na bancada pelo discente no ambiente virtual, foi utilizado o método analítico denominado de Equação de Tableau, aplicando este algoritmo de resolução através de código desenvolvido em C# e compatível com a *engine*, para equacionar as tensões e correntes de cada ramo constituinte do circuito elétrico modelado, além das tensões dos respectivos nós de conexão, possibilitando assim, a simulação dos circuitos propostos.

As 4 etapas de testes propostas neste trabalho, de acordo com o modelo V (teste unitário, de subsistema, sistema e aceitação), foram executadas e bem-sucedidas e eventuais problemas identificados foram corrigidos. O laboratório virtual desenvolvido também atendeu de forma positiva aos requisitos estipulados nas fases de planejamento como: similaridade com os ambientes físicos da instituição, possibilidade de utilização multiplataforma (computador local, computador em servidor *web* e *Android*), capacidade de repetição das práticas simuladas, retornos do sistema (para circuito correto, incorreto, curto-circuito, término do tempo de montagem e direcionamento de ações), capacidade de expansão e migração para novas tecnologias.

O laboratório virtual desenvolvido tem potencial de aplicação nos cursos da área elétrica, trazendo segurança e aprimoramento dos conceitos aos discentes, cabendo a utilização como complemento e apoio às aulas presenciais, mas também podendo ser aplicado em dinâmicas remotas de ensino.

A aplicação desenvolvida possibilita a expansão para os demais laboratórios contidos no setor, servindo como base e modelo para ampliação do uso de simuladores virtualizados personalizados nos cursos que englobam a área elétrica do IFF.

Tendo em vista o contínuo aprimoramento do trabalho proposto, em ações futuras é indicado a inclusão das demais práticas ministradas na bancada didática não contempladas no escopo deste trabalho (interruptor de duas seções, fotocélula, sensor de presença, ventilador de teto entre outros), realização de testes unitários automatizados, possibilitando assim a detecção de falhas não identificadas nos testes manuais, aplicação de teste beta com os alunos da disciplina, inclusão de módulo de avaliação discente, para suporte e compreensão da curva de aprendizagem de cada aluno, e compilação de versão *light*, para dispositivos com pouco recurso computacional.

REFERÊNCIAS

- ALGETEC. **Laboratórios Virtuais**. Disponível em: <<https://www.algetec.com.br/br/produto/laboratoriosvirtuais>>. Acesso em: 21 dez. 2022.
- BAIN AND COMPANY. **Impacto econômico e social do Android no Brasil**. 2020. Disponível em: <<https://www.bain.com/contentassets/a9200a057a0241b8963c05a9b09e33fe/impactos-do-android-no-brasil.pdf>>. Acesso em: 31 ago. 2022.
- BLAIN, J. M. **The complete guide to blender graphics: computer modeling & animation**. Sixth edition ed. Boca Raton, FL: CRC Press, 2021.
- BLENDER. **Home of the Blender project - Free and Open 3D Creation Software**. Disponível em: <<https://www.blender.org/>>. Acesso em: 9 nov. 2021.
- BOYLESTAD, R. L. **Introdução à análise de circuitos**. 10. ed. São Paulo: Pearson Education do Brasil, 2004.
- BRITO, A. **Blender 3D: architecture, buildings, and scenery: create photorealistic 3D architectural visualizations of buildings, interiors, and environmental scenery**. Birmingham: Packt Publ, 2008.
- BURIAN JÚNIOR, Y. **Circuitos elétricos**. Sao Paulo: Pearson Prentice Hall, 2006.
- CADE SIMU. **Curso e Download Gratuito**. Disponível em: <<https://www.cadesimu.net/>>. Acesso em: 2 set. 2022.
- CARVALHO, T. L. E. **Orientação a Objetos - Aprenda seus conceitos e suas aplicabilidades de forma efetiva**. 1ª ed. Casa do Código, 2016.
- CERVELIN, S.; CAVALIN, G. **Instalações elétricas prediais - teoria e prática**. 22. ed. Curitiba: Base Livros Didáticos Ltda., 2008.
- CHUA, L. O.; DESOER, C. A.; KUH, E. S. **Linear and nonlinear Circuits**. New York: McGraw-Hill. v. 1, 1987.
- CNOP, A. C. **Simulação virtual de visita técnica no reator argonauta para fins de divulgação científica**. Rio de Janeiro - RJ: UFRJ, 2016.
- DAINEKO, Y.; IPALAKOVA, M.; TSOY, D.; BOLAOV, Z.; YELGONDY, Y.; SHAIPIITEN, A. **Development of the educational software based on a game engine**. 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT). **Anais...** Em: 2018 IEEE 12TH INTERNATIONAL CONFERENCE ON APPLICATION OF INFORMATION AND COMMUNICATION TECHNOLOGIES (AICT). out. 2018.

DAINEKO, Y.; KOZHAKHMETOVA, B. A.; KULAKAYEVA, A. E.; TSOY, D.; AITMAGAMBETOV, A. Z.; GUBSKY, D. S.; IPALAKOVA, M. T.; SEITNUR, A. M. **Development of Virtual Laboratory Work on the Base of Unity Game Engine for the Study of Radio Engineering Disciplines.** Em: 8TH INTERNATIONAL CONFERENCE ON AUGMENTED REALITY, VIRTUAL REALITY AND COMPUTER GRAPHICS, AVR. 2021. . Acesso em: 25 ago. 2022

DE LOURENÇO, A. C.; CRUZ, E. C. A.; FERREIRA, S. R.; CHOUERI JÚNIOR, S. **Circuitos Digitais.** 9ª ed. São Paulo: Érica, 2009.

DLB. **DLB REALIDADE VIRTUAL - Simuladores.** Disponível em: <<https://sites.google.com/view/dlb-realidade-virtual-para-ead/simuladores>>. Acesso em: 26 ago. 2022.

ENG, L. Z. **Building a game with Unity and Blender: learn how to build a complete 3D game using the industry-leading Unity game development engine and Blender, the graphics software that gives life to your ideas.** Birmingham, UK: Packt Publishing, 2015.

EXSTO. **Exsto Hub – Apps no Google Play.** Disponível em: <https://play.google.com/store/apps/details?id=br.com.exsto.ExstoHub&hl=pt_BR&gl=US>. Acesso em: 26 ago. 2022.

FIERGS; SENAI. **Ambiente Virtual de Aprendizagem para cursos EAD.** Disponível em: <https://ead.senairs.org.br/cursos/#portal_recurso>. Acesso em: 26 ago. 2022.

FRANCHI, C. M. **Acionamentos elétricos.** 4. ed. São Paulo: Érica, 2008.

GIMP. **GIMP - The GNU Image Manipulation Program: The Free and Open Source Image Editor.** Disponível em: <<https://www.gimp.org/>>. Acesso em: 2 set. 2022.

GONZÁLEZ, J. D.; ESCOBAR, J. H.; SÁNCHEZ, H.; HOZ, J. DE LA; BELTRÁN, J. R. **2D and 3D virtual interactive laboratories of physics on Unity platform.** Journal of Physics: Conference Series, v. 935, n. 1, p. 012069, 1 dez. 2017.

HALKER, N. S. M.; VATANABE, S. L. **Desenvolvimento de um Laboratório Virtual Didático para ensino de Máquinas Elétricas.** p. 18, 2020.

IFF. **Conheça o IFF.** Disponível em: <<https://portal1.iff.edu.br/conheca-o-iff/fluminense/conheca-o-iff/fluminense>>. Acesso em: 9 nov. 2021.

INDRA. **Sistemas de Treinamento.** Disponível em: <<https://www.indracompany.com/pt-br/sistemas-treinamento-1?business=1>>. Acesso em: 26 ago. 2022.

JING, X. **Design and Implementation of 3D Virtual Digital Campus - Based on Unity3D.** 2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). **Anais...** Em: 2016 EIGHTH INTERNATIONAL CONFERENCE ON MEASURING TECHNOLOGY AND MECHATRONICS AUTOMATION (ICMTMA). mar. 2016.

LIMA, R. O. M. DE. **LABEDIF: Uma Proposta de Laboratório Virtual para os Cursos Profissionais na Área da Construção Civil.** 5 maio 2020.

ONIRIA. **Simulação Virtual para Treinamento e Desenvolvimento de Pessoas.** Disponível em: <<https://oniria.com.br/simulacao-virtual/>>. Acesso em: 26 ago. 2022.

PANDEKA, N. P.; OWOLAWI, P. A.; MAPAYI, T.; MALELE, V.; AIYETORO, G.; OJO, J. S. **Mobile Virtual Reality (VR) for Science Projects: Ohm's Law Laboratory.** 2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME). **Anais...** Em: 2021 International conference on electrical, computer, communications and mechatronics engineering (ICECCME). out. 2021.

PHET. **Simulações Interativas.** Disponível em: <https://phet.colorado.edu/pt_BR/>. Acesso em: 25 ago. 2022.

POHL, K.; RUPP, C. **Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam, foundation level, IREB compliant.** Second edition ed. Santa Barbara, CA: Rocky Nook, 2015.

PORTO, S. C. C. **Laboratório Virtual X Laboratório Material: Investigando a Natureza do Entendimento Construído em Dois Ambientes de Aprendizagem.** 2015.

PRESSMAN, R. S. **Software engineering: a practitioner's approach.** 7th ed. New York: McGraw-Hill Higher Education, 2011.

SADIKU, M. N. O.; MUSA, S. M.; ALEXANDER, C. K. **Applied circuit analysis.** New York: McGraw-Hill, 2013.

SCILAB. **What is Scilab ?** Disponível em: <<https://www.scilab.org/>>. Acesso em: 2 set. 2022.

SHAMIEH, C. **Engenharia de sistemas para leigos.** Limitada ed. Indianapolis: Wiley Publishing, 2011.

SILVA FILHO, G. O., FURTADO, P. G. F., LEITE, F. L. & NEVES FILHO, H. B. (2018). **Unity: Criando jogos e outras aplicações multi-plataforma.** Em H. B. Neves Filho, L. A. B. Freitas & N. C. C. Quinta (Orgs.). Introdução ao desenvolvimento de softwares para analistas do comportamento (pp. 138-155). Campinas: ABPMC.

SOMMERVILLE, I. **Engenharia de software.** 9. ed. São Paulo: Pearson Prentice Hall, 2011.

TANAKA, E. H.; DE ALMEIDA, L.; GOUVEIA, G. S. F. **Ambiente virtual e colaborativo para a capacitação de eletricitistas de rede de distribuição.** p. 9, 2021.

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de Requisitos: software orientado ao negócio.** 1ª ed. Rio de Janeiro: Brasport, 2016.

ZHU, Z.; SHEN, Y.; G Y. LIN P.E., C.; REN, S.; SMITH, K.; W. DEAN, A. **Design of a Virtual Laboratory for Automation Control**. ASEE Annual Conference and Exposition, Conference Proceedings. **Anais...** Em: 125TH ASEE ANNUAL CONFERENCE AND EXPOSITION. Salt Lake City, UT: American Society for Engineering Education, 2018. Disponível em: <file:///C:/Users/viana/Downloads/design-of-a-virtual-laboratory-for-automation-control.pdf>. Acesso em: 25 ago. 2022

APÊNDICE A – DESCRIÇÃO DOS CENÁRIOS

Caso de Uso: Entrar na Simulação

Ator: Aluno

Fluxo Principal

- 1 – O sistema dá boas vindas e informa o objetivo da simulação.
- 2 – O aluno fecha o aviso.
- 3 – O aluno se desloca até o laboratório de Instalações Elétricas. [FA1][FA2][FA3][FA4]
- 4 – O sistema informa o nome do laboratório.
- 5 – O sistema informa a ação para entrar no laboratório.
- 6 – O aluno acessa o laboratório.
- 7 – O aluno se desloca até a bancada didática.
- 8 – O sistema informa que a bancada deve ser clicada/tocada.
- 9 – O aluno clica/toca na bancada.
- 10 – O sistema mostra as opções de práticas.
- 11 – O aluno escolhe a prática do Interruptor Simples. [FA5][FA6][FA7][FA8][FA9]
- 12 – O aluno monta o circuito. [FA2][FA10][FA11]
- 13 – O aluno simula o funcionamento do circuito. [FA2][FA11][FA12]
- 14 – O aluno checa se o circuito está correto. [FA2]
- 15 – O sistema informa que o circuito está correto. [FA13]
- 16 – O aluno clica/toca no botão de passar.
- 17 – O sistema mostra o diagrama de ligação correto.
- 18 – O aluno clica/toca no botão de passar para próxima prática. [FA14]

Fluxo Alternativo 1 – FA1

No item 3, o aluno retornar ao menu inicial.

Fluxo Alternativo 2 – FA2

No item 3, 13 e 14, o aluno acessa o menu de configurações para ajustar o volume do ambiente, do personagem e geral.

Fluxo Alternativo 3 – FA3

No item 3, o aluno acessa o menu de controles e visualiza as funções dos botões de movimentação e interação do personagem.

Fluxo Alternativo 4 – FA4

No item 3, o aluno se desloca para outro laboratório do setor.

Fluxo Alternativo 5 – FA5

No item 11, o aluno escolhe a prática da Campanha.

Fluxo Alternativo 6 – FA6

No item 11, o aluno escolhe a prática da Cigarra.

Fluxo Alternativo 7 – FA7

No item 11, o aluno escolhe a prática do Interruptor Paralelo.

Fluxo Alternativo 8 – FA8

No item 11, o aluno escolhe a prática do Interruptor Intermediário.

Fluxo Alternativo 9 – FA9

No item 11, o aluno escolhe a prática da Tomada 2P+T.

Fluxo Alternativo 10 – FA10

No item 12, o sistema informa que o tempo de montagem encerrou e realiza o *reset* do circuito.

Fluxo Alternativa 11 – FA11

No item 12 e 13, o aluno realiza o *reset* do circuito.

Fluxo Alternativo 12 – FA12

No item 13, o sistema informa a ocorrência de curto-circuito e realiza o *reset* do circuito.

Fluxo Alternativo 13 – FA13

No item 15, o sistema informa que o circuito está incorreto.

Fluxo Alternativo 14 – FA14

No item 18, o aluno fecha o diagrama.

Caso de Uso: Entrar no Laboratório

Ator: Aluno

Fluxo Principal

- 1 – O sistema mostra as opções de laboratórios do setor. [FA1]
- 2 – O aluno navega pelas opções.
- 3 – O sistema mostra o nome dos laboratórios.
- 4 – O aluno acessa o laboratório selecionado. [FA1]

Fluxo Alternativo 1 – FA1

No item 1 e 4, o aluno retorna ao menu inicial.

Caso de Uso: Entrar nas Configurações

Ator: Aluno

Fluxo Principal

- 1 – O sistema mostra as opções de configuração de áudio. [FA1][FA4]
- 2 – O aluno ajusta o volume do ambiente. [FA1][FA2][FA3][FA4]

Fluxo Alternativo 1 – FA1

No item 1 e 2, o aluno seleciona as configurações de controles e o sistema mostra as funções dos botões de movimentação e interação do personagem.

Fluxo Alternativo 2 – FA2

No item 2, o aluno ajusta o volume do personagem.

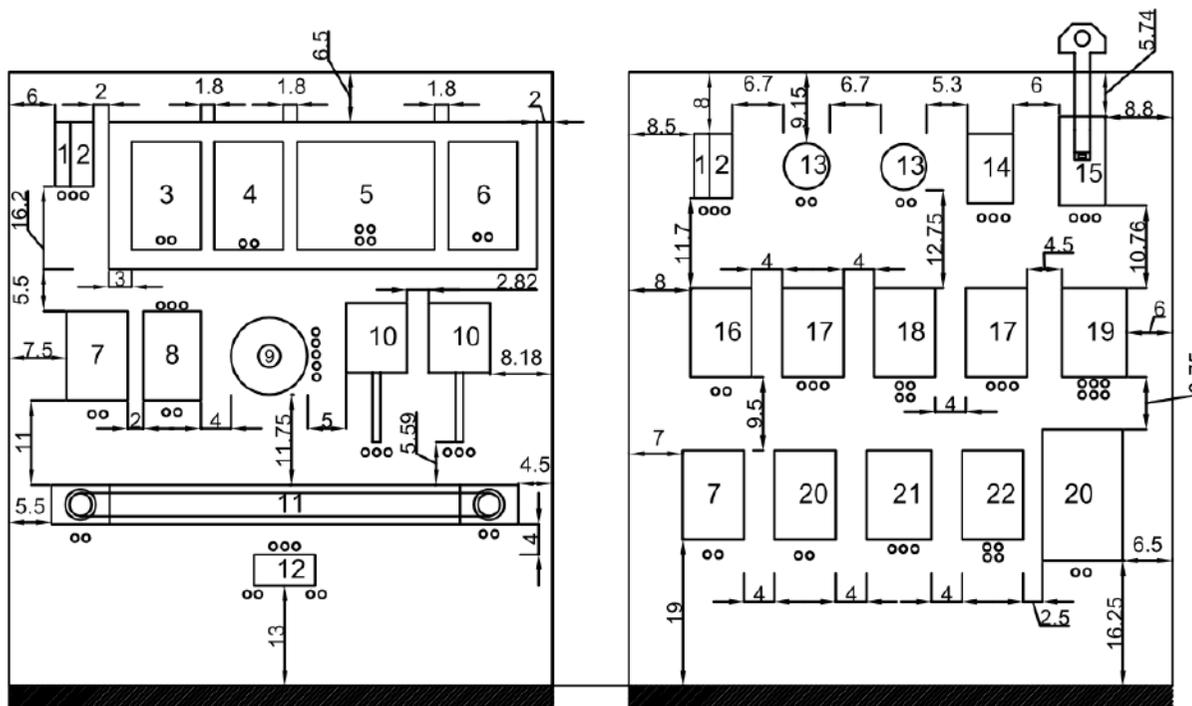
Fluxo Alternativo 3 – FA3

No item 2, o aluno ajusta o volume geral.

Fluxo Alternativo 4 – FA4

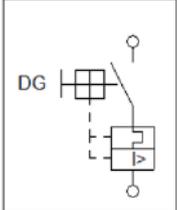
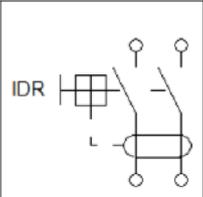
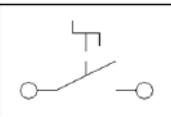
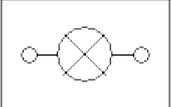
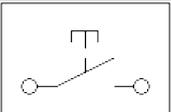
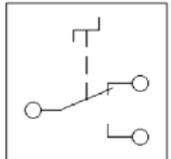
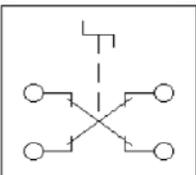
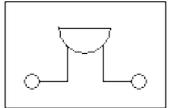
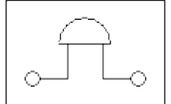
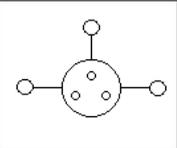
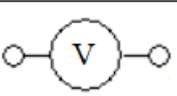
No item 1 e 2, o aluno retorna ao menu inicial.

APÊNDICE B – PROJETO BANCADA DIDÁTICA

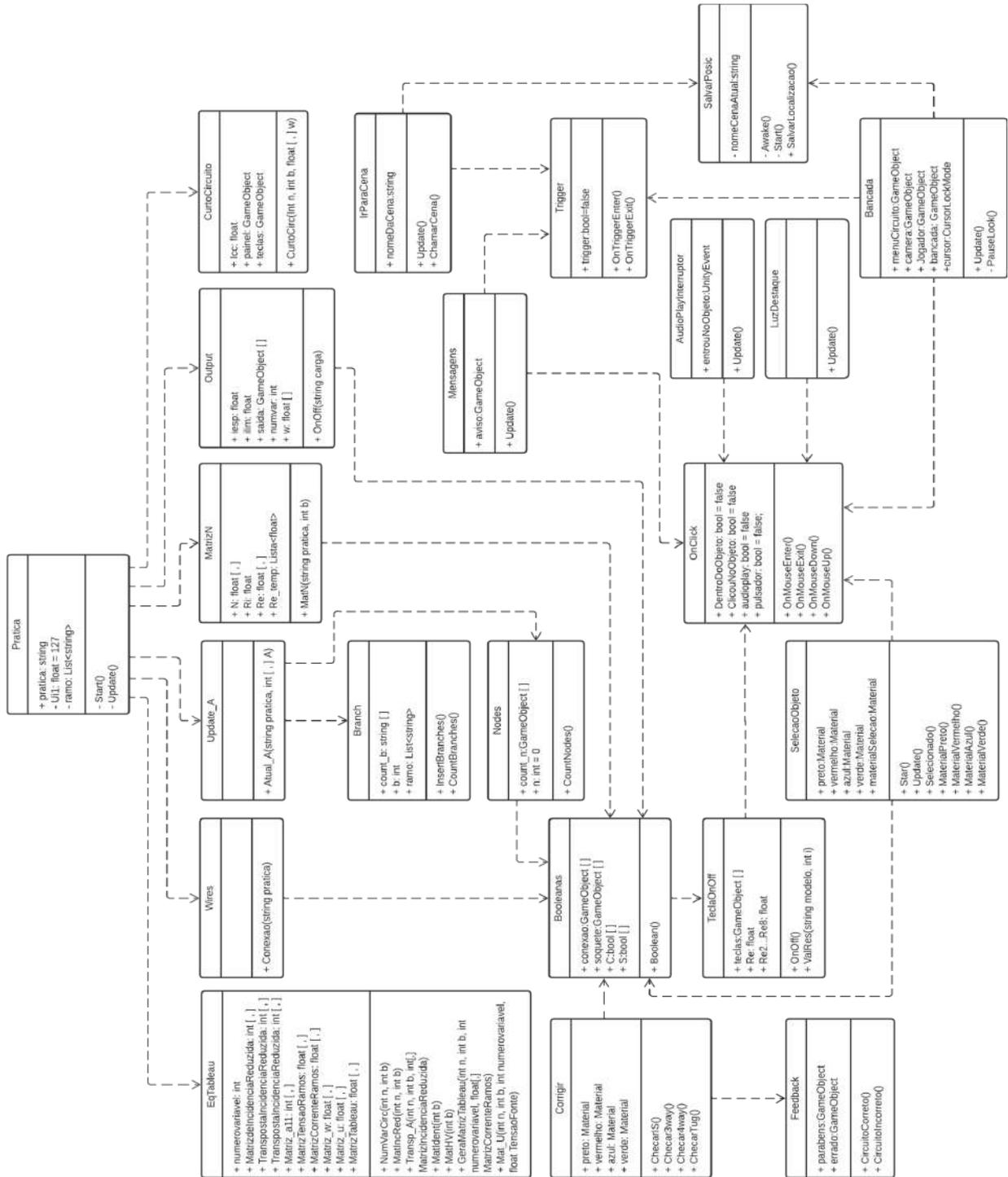


- 1 - Disjuntor
- 2 - IDR
- 3 - Voltímetro
- 4 - Amperímetro
- 5 - Wattímetro
- 6 - Freqüencímetro
- 7 - Pulsador
- 8 - Interruptor de ventilador de teto
- 9 - Receptáculo acoplado à motor de ventilador de teto
- 10 - Chave boia
- 11 - Lâmpada fluorescente
- 12 - Reator
- 13 - Receptáculo
- 14 - Sensor de presença
- 15 - Fotocélula
- 16 - Interruptor de uma seção
- 17 - Interruptor three way
- 18 - Interruptor four way
- 19 - Interruptor de uma seção acoplado a uma tomada
- 20 - Campainha
- 21 - Tomada 2P + T
- 22 - Dois interruptores de uma seção cada

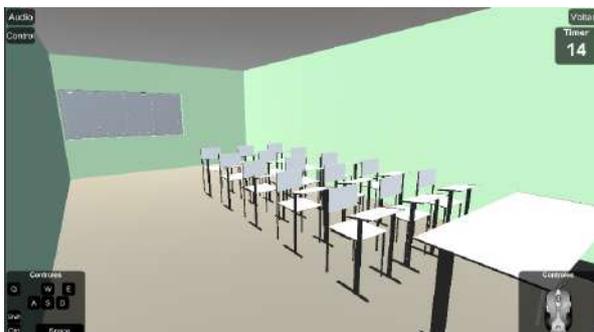
APÊNDICE C – QUADRO DE SÍMBOLOS

Simbologia	Descrição
	Disjuntor Termomagnético
	Interruptor Diferencial Residual
	Interruptor Simples
	Lâmpada
	Pulsador
	Interruptor Paralelo (<i>Three-way</i>)
	Interruptor Intermediário (<i>Four-way</i>)
	Campainha
	Cigarra
	Tomada 2P + T
	Voltímetro

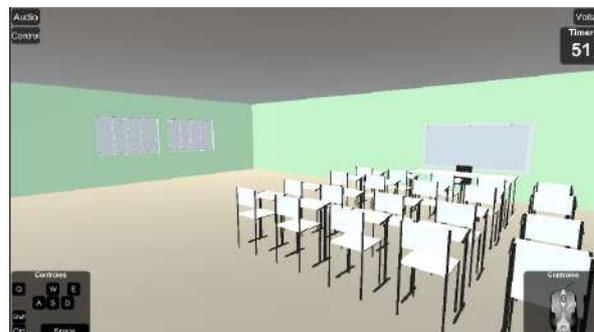
APÊNDICE D – DIAGRAMAS DE CLASSES



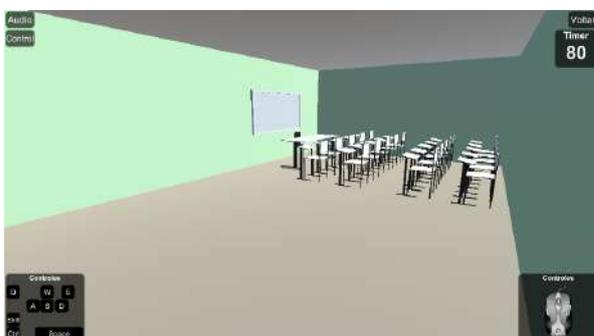
APÊNDICE E – LABORATÓRIOS QUE COMPÕEM O SETOR ELÉTRICO



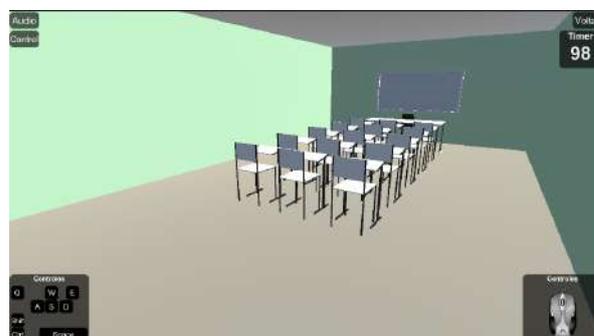
(a) B10 – Elétrica I – Fontes Renováveis



(b) B13 – Elétrica II – Comandos e Proteção



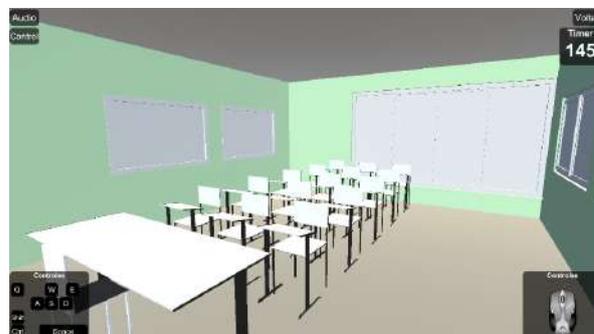
(c) B16 – Elétrica IV – CLP e Redes Industriais



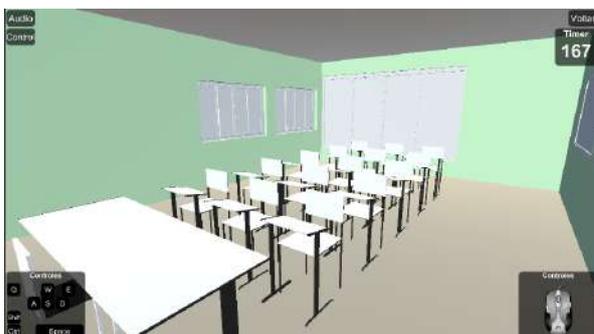
(d) B17 – Elétrica V – Máquinas Elétricas



(e) B18 – Elétrica VI e VII – Automação Predial



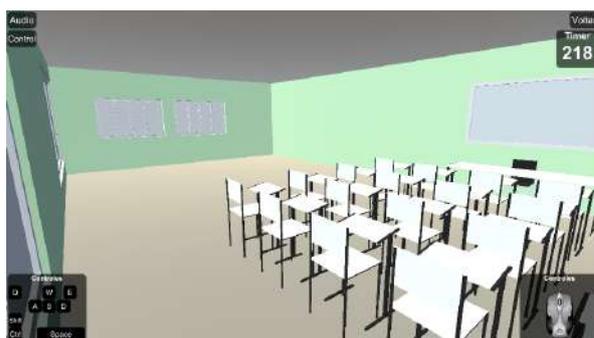
(f) B19a – Elétrica VIII – Segurança em Instalações Elétricas



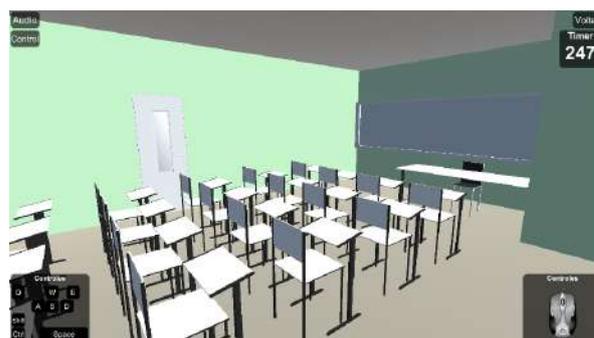
(g) B19b – Eletrotécnica I – Eletrotécnica CC e CA



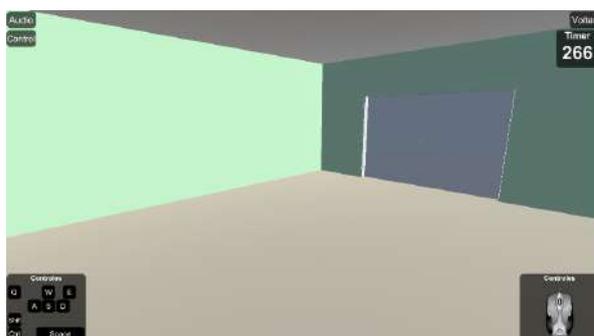
(h) B20 – Elétrica IX – Manutenção Elétrica



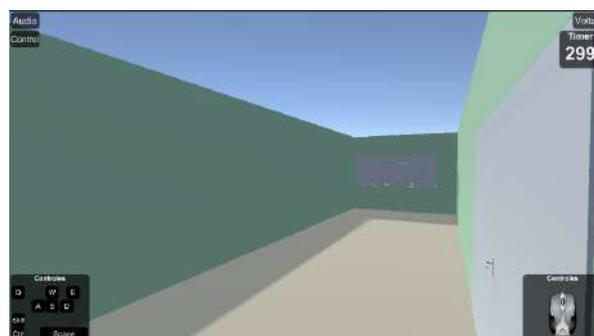
(i) B21 – Eletrotécnica II – Eletrotécnica CC e CA



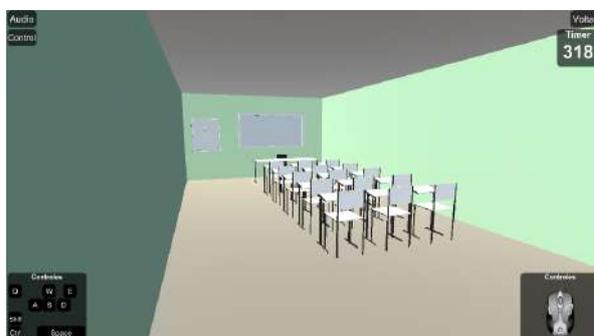
(j) B22a – Elétrica X – Redes e Trafos



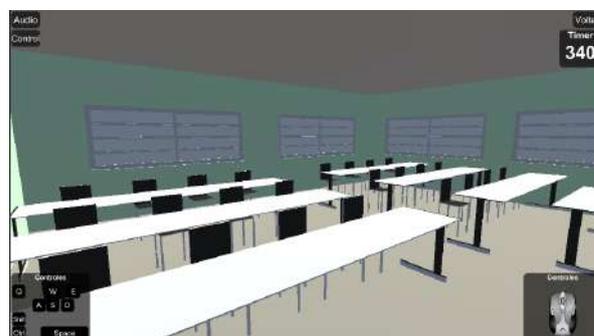
(k) B22b – Elétrica X – Redes e Trafos



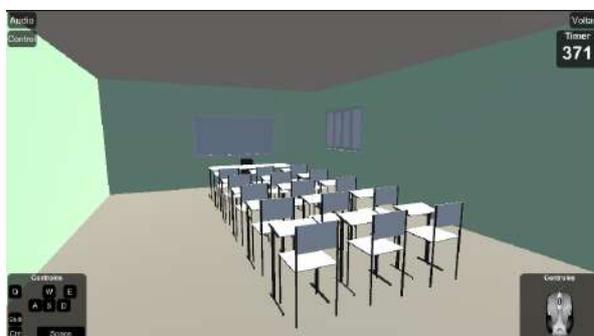
(l) B22b – Anexo



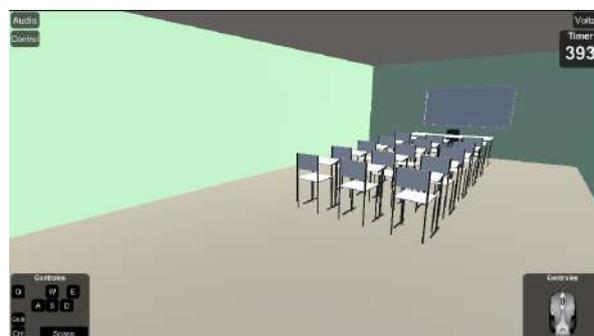
(m) B23 – Eletrotécnica III – Eletrotécnica CC e CA



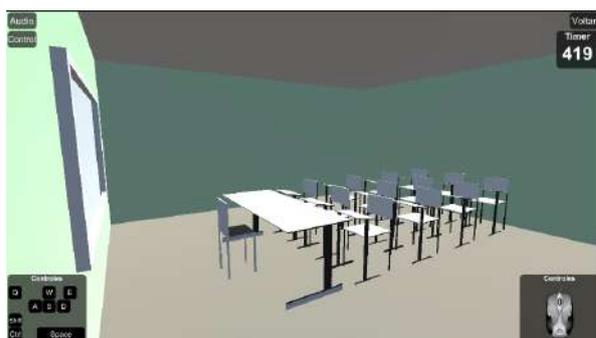
(n) B110 – Informática e CAD



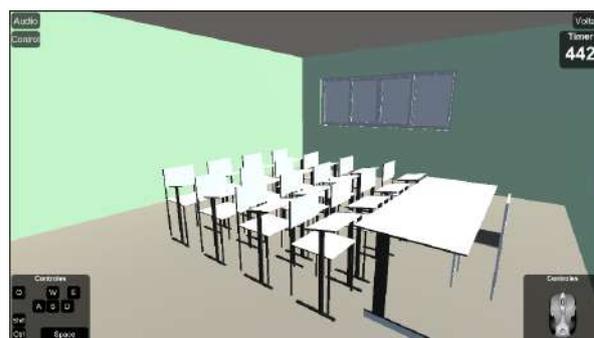
(o) B111 – Eletrônica I – Eletrônica Analógica



(p) B112 – Eletrônica II – Eletrônica Digital



(q) B118-1 – Eletrônica III – Eletrônica Industrial



(r) B118-2 – Eletrônica IV – Manutenção Eletrônica

APÊNDICE F – QUADRO COM TESTES UNITÁRIOS

Classe	Método	Teste realizado	Situação
AudioPlayInterruptor	Update()	Play do áudio após clique.	ok
		Reset da booleana audioplay.	ok
Bancada	Update()	Ativação do menu após clique.	ok
		Armazenamento da posição.	ok
	PauseLook()	Verificação da desativação do componente <i>FirstPersonLook</i> após ativação do menu de práticas.	ok
		Verificação da desativação do componente <i>FirstPersonMovement</i> após ativação do menu de práticas.	ok
		Verificação da desativação do componente <i>Jump</i> após ativação do menu de práticas.	ok
		Verificação da desativação do componente <i>Crouch</i> após ativação do menu de práticas.	ok
		Verificação da ativação do componente <i>FirstPersonLook</i> após desativação do menu de práticas.	ok
		Verificação da ativação do componente <i>FirstPersonMovement</i> após desativação do menu de práticas.	ok
		Verificação da ativação do componente <i>Jump</i> após desativação do menu de práticas.	ok
		Verificação da ativação do componente <i>Crouch</i> após desativação do menu de práticas.	ok
Booleanas	Boolean()	Verificação da mudança da booleana <i>C</i> de acordo com sua conexão de referência.	ok
		Verificação da mudança da booleana <i>S</i> de acordo com seu soquete de referência.	ok

Classe	Método	Teste realizado	Situação
Booleanas	Boolean()	Verificação da mudança da booleana <i>R</i> de acordo com sua tecla de referência.	ok
Branch	InsertBranches()	Verificação da inserção do ramo na lista de controle	ok
Branch	CountBranches()	Verificação da contagem de itens na lista de controle.	ok
Button	Sair()	Verificação do encerramento da aplicação.	ok
	ResetPlayerPrefs()	Verificação da exclusão das chaves de armazenamento.	ok
Corrigir	ChecarIS()	Verificação do <i>feedback</i> de erro.	ok
		Verificação do <i>feedback</i> de acerto.	ok
	Checar3way()	Verificação do <i>feedback</i> de erro.	ok
		Verificação do <i>feedback</i> de acerto.	ok
	Checar4way()	Verificação do <i>feedback</i> de erro.	ok
		Verificação do <i>feedback</i> de acerto.	ok
	ChecarTug()	Verificação do <i>feedback</i> de erro.	ok
		Verificação do <i>feedback</i> de acerto.	ok
CurtoCircuito	CurtoCirc()	Verificação da ativação do aviso após fechamento de curto-circuito.	ok
		Verificação da mudança para <i>true</i> da booleana <i>ClicouNoObjeto</i> .	ok
DestroyGameObject	Awake()	Verificação da atribuição do nome da cena ativa na variável <i>nomeCenaAtual</i> .	ok
	Start()	Verificação da destruição de objeto em cena já acessada.	ok
EqTableau	NumVarCirc()	Verificação da contabilização dos números de variáveis do circuito de acordo com os ramos e nós inseridos no circuito.	ok
	MatIncRed()	Verificação do tamanho da matriz	ok

Classe	Método	Teste realizado	Situação
EqTableau		instanciada de acordo com o número de ramos e nós inseridos no circuito.	
	Transp_A()	Verificação da transposição da matriz de incidência reduzida.	ok
	MatIdent()	Verificação do preenchimento da diagonal principal com o valor 1 da matriz identidade.	ok
		Verificação do preenchimento da triangular superior e inferior da matriz identidade com o valor 0.	ok
	MatM()	Verificação do preenchimento da diagonal principal com o valor 1 da matriz M(t).	ok
		Verificação do preenchimento da triangular superior e inferior da matriz M(t) com o valor 0.	ok
EqTableau	GeraMatrizTableau()	Verificação da criação da matriz de Tableau de acordo com o número de variáveis do circuito.	ok
		Verificação do preenchimento da matriz de Tableau com o valor 0.	ok
		Verificação da inserção da matriz de incidência reduzida na posição correta da matriz de Tableau.	ok
		Verificação da inserção da matriz de incidência reduzida transposta na posição correta da matriz de Tableau.	ok
		Verificação da inserção da matriz identidade na posição correta da matriz de Tableau.	ok
		Verificação da inserção da matriz M(t) na posição correta da matriz de Tableau.	ok
		Verificação da inserção da matriz N(t) na posição correta da matriz de Tableau.	ok
EqTableau	Mat_U()	Verificação da criação da matriz u(t) de	ok

Classe	Método	Teste realizado	Situação
		acordo com o número de variáveis do circuito.	
EqTableau	Mat_U()	Verificação do preenchimento da matriz de $u(t)$ com o valor 0.	ok
		Verificação da inserção da tensão da fonte na posição correta da matriz $u(t)$.	ok
	EqTab()	Verificação da criação da matriz $w(t)$ de acordo com o número de variáveis do circuito.	ok
		Verificação da inversão da matriz de Tableau.	ok
		Verificação da multiplicação da matriz $u(t)$ pela matriz de Tableau inversa.	ok
		Verificação do preenchimento da matriz $w(t)$ com o resultado da multiplicação das matrizes $u(t)$ e Tableau inversa.	ok
Feedback	CircuitoCorreto()	Verificação da ativação do painel de aviso.	ok
	CircuitoIncorreto()	Verificação da ativação do painel de aviso.	ok
IrParaCena	ChamarCena()	Verificação do carregamento da cena.	ok
	Update()	Verificação de mudança de cena após <i>trigger</i> na porta e <i>input</i> do botão.	ok
LuzDestaque	Update()	Verificação da desativação do componente <i>Light</i> após clique no objeto.	ok
		Verificação da desativação do componente <i>Light</i> após posicionamento do cursor do mouse no objeto.	ok
		Verificação da ativação do componente <i>Light</i> após novo clique no objeto.	ok
		Verificação da ativação do componente <i>Light</i> após posicionamento do cursor do mouse fora do objeto.	ok

Classe	Método	Teste realizado	Situação
MatrizN	MatN()	Verificação do tamanho da matriz $N(t)$ de acordo com o número de ramos da prática do Interruptor Simples.	ok
		Verificação da criação da lista de resistores da prática do Interruptor Simples.	ok
		Verificação da inserção do valor 0 na lista de resistores após clique/toque nos soquetes 1 e 2 da prática do Interruptor Simples.	ok
		Verificação da inserção do valor do resistor 2 na lista de resistores após clique/toque nos soquetes 3 e 4 da prática do Interruptor Simples.	ok
		Verificação da inserção do valor do resistor 3 na lista de resistores após clique/toque nos soquetes 5 e 6 da prática do Interruptor Simples.	ok
		Verificação da conversão da lista de resistores em <i>array</i> da prática do Interruptor Simples.	ok
		Verificação do preenchimento da diagonal principal da matriz $N(t)$ com os valores contidos na matriz de resistores da prática do Interruptor Simples.	ok
		Verificação do preenchimento das posições diferentes da diagonal principal com o valor 0 da matriz $N(t)$ da prática do Interruptor Simples.	ok
		Verificação do tamanho da matriz $N(t)$ de acordo com o número de ramos da prática Cigarra.	ok
		Verificação da criação da lista de resistores da prática Cigarra.	ok

Classe	Método	Teste realizado	Situação
MatrizN	MatN()	Verificação da inserção do valor 0 na lista de resistores após clique/toque nos soquetes 1 e 2 da prática Cigarra.	ok
		Verificação da inserção do valor do resistor 2 na lista de resistores após clique/toque nos soquetes 3 e 4 da prática Cigarra.	ok
		Verificação da inserção do valor do resistor 3 na lista de resistores após clique/toque nos soquetes 5 e 6 da prática Cigarra.	ok
		Verificação da conversão da lista de resistores em <i>array</i> da prática Cigarra.	ok
		Verificação do preenchimento da diagonal principal da matriz N(t) com os valores contidos na matriz de resistores da prática Cigarra.	ok
		Verificação do preenchimento das posições diferentes da diagonal principal com o valor 0 da matriz N(t) da prática Cigarra.	ok
		Verificação do tamanho da matriz N(t) de acordo com o número de ramos da prática Campainha.	ok
		Verificação da criação da lista de resistores da prática Campainha.	ok
		Verificação da inserção do valor 0 na lista de resistores após clique/toque nos soquetes 1 e 2 da prática Campainha.	ok
		Verificação da inserção do valor do resistor 2 na lista de resistores após clique/toque nos soquetes 3 e 4 da prática Campainha.	ok

Classe	Método	Teste realizado	Situação
MatrizN	MatN()	Verificação da inserção do valor do resistor 3 na lista de resistores após clique/toque nos soquetes 5 e 6 da prática Campainha.	ok
		Verificação da conversão da lista de resistores em <i>array</i> da prática Campainha.	ok
		Verificação do preenchimento da diagonal principal da matriz N(t) com os valores contidos na matriz de resistores da prática Campainha.	ok
		Verificação do preenchimento das posições diferentes da diagonal principal com o valor 0 da matriz N(t) da prática Campainha.	ok
		Verificação do tamanho da matriz N(t) de acordo com o número de ramos da prática do Interruptor Paralelo.	ok
		Verificação da criação da lista de resistores da prática do Interruptor Paralelo.	ok
		Verificação da inserção do valor 0 na lista de resistores após clique/toque nos soquetes 1 e 2 da prática do Interruptor Paralelo.	ok
		Verificação da inserção do valor do resistor 2 na lista de resistores após clique/toque nos soquetes 3 e 4 da prática do Interruptor Paralelo.	ok
		Verificação da inserção do valor do resistor 3 na lista de resistores após clique/toque nos soquetes 3 e 5 da prática do Interruptor Paralelo.	ok

Classe	Método	Teste realizado	Situação
MatrizN	MatN()	Verificação da inserção do valor do resistor 8 na lista de resistores após clique/toque nos soquetes 6 e 8 da prática do Interruptor Paralelo.	ok
		Verificação da inserção do valor do resistor 9 na lista de resistores após clique/toque nos soquetes 7 e 8 da prática do Interruptor Paralelo.	ok
		Verificação da inserção do valor do resistor 6 na lista de resistores após clique/toque nos soquetes 9 e 10 da prática do Interruptor Paralelo.	ok
		Verificação do preenchimento da diagonal principal da matriz $N(t)$ com os valores contidos na matriz de resistores da prática do Interruptor Paralelo.	ok
		Verificação do preenchimento das posições diferentes da diagonal principal com o valor 0 da matriz $N(t)$ da prática do Interruptor Paralelo.	ok
		Verificação do tamanho da matriz $N(t)$ de acordo com o número de ramos da prática do Interruptor Intermediário.	ok
		Verificação da criação da lista de resistores da prática do Interruptor Intermediário.	ok
		Verificação da inserção do valor 0 na lista de resistores após clique/toque nos soquetes 1 e 2 da prática do Interruptor Intermediário.	ok
		Verificação da inserção do valor do resistor 2 na lista de resistores após clique/toque nos soquetes 3 e 4 da prática do Interruptor Intermediário.	ok

Classe	Método	Teste realizado	Situação
MatrizN	MatN()	Verificação da inserção do valor do resistor 3 na lista de resistores após clique/toque nos soquetes 3 e 5 da prática do Interruptor Intermediário.	ok
		Verificação da inserção do valor do resistor 4 na lista de resistores após clique/toque nos soquetes 6 e 9 da prática do Interruptor Intermediário.	ok
		Verificação da inserção do valor do resistor 5 na lista de resistores após clique/toque nos soquetes 7 e 8 da prática do Interruptor Intermediário.	ok
		Verificação da inserção do valor do resistor 6 na lista de resistores após clique/toque nos soquetes 6 e 8 da prática do Interruptor Intermediário.	ok
		Verificação da inserção do valor do resistor 7 na lista de resistores após clique/toque nos soquetes 7 e 9 da prática do Interruptor Intermediário.	ok
		Verificação da inserção do valor do resistor 8 na lista de resistores após clique/toque nos soquetes 10 e 12 da prática do Interruptor Intermediário.	ok
		Verificação da inserção do valor do resistor 10 na lista de resistores após clique/toque nos soquetes 13 e 14 da prática do Interruptor Intermediário.	ok
		Verificação do preenchimento da diagonal principal da matriz N(t) com os valores contidos na matriz de resistores da prática do Interruptor Intermediário.	ok

Classe	Método	Teste realizado	Situação
MatrizN	MatN()	Verificação do preenchimento das posições diferentes da diagonal principal com o valor 0 da matriz N(t) da prática do Interruptor Intermediário.	ok
		Verificação do tamanho da matriz N(t) de acordo com o número de ramos da prática Tomada 2P+T.	ok
		Verificação da criação da lista de resistores da prática Tomada 2P+T.	ok
		Verificação da inserção do valor 0 na lista de resistores após clique/toque nos soquetes 1 e 2 da prática Tomada 2P+T.	ok
		Verificação da inserção do valor do resistor 2 na lista de resistores após clique/toque nos soquetes 4 e 6 da prática Tomada 2P+T.	ok
		Verificação do preenchimento da diagonal principal da matriz N(t) com os valores contidos na matriz de resistores da prática Tomada 2P+T.	ok
		Verificação do preenchimento das posições diferentes da diagonal principal com o valor 0 da matriz N(t) da prática Tomada 2P+T.	ok
Mensagens	Star()	Verificação da desativação do painel de aviso.	ok
	Update()	Verificação da ativação do painel de aviso após o <i>trigger</i> .	ok
		Verificação da desativação do painel de aviso após o fim do <i>trigger</i> .	ok
		Verificação da ativação do painel de aviso após o posicionamento do cursor do mouse sobre o objeto.	ok

Classe	Método	Teste realizado	Situação
Mensagens	Update()	Verificação da desativação do painel de aviso após o posicionamento do cursor do mouse fora do objeto.	ok
MoveZoom	Update()	Verificação da movimentação da câmera para a direita após <i>input</i> da tecla D.	ok
		Verificação da movimentação da câmera para a direita após deslocamento do dedo em tela <i>touch</i> .	ok
		Verificação da movimentação da câmera para a esquerda após <i>input</i> da tecla A.	ok
		Verificação da movimentação da câmera para a esquerda após deslocamento do dedo em tela <i>touch</i> .	ok
		Verificação da movimentação da câmera para cima após <i>input</i> da tecla W.	ok
		Verificação da movimentação da câmera para cima após deslocamento do dedo em tela <i>touch</i> .	ok
		Verificação da movimentação da câmera para baixo após <i>input</i> da tecla S.	ok
		Verificação da movimentação da câmera para baixo após deslocamento do dedo em tela <i>touch</i> .	ok
MusicPlayer	StartMusicPlayer()	Verificação da continuação do fluxo de áudio após transição de cena.	ok
Nodes	CountNodes()	Verificação da inserção dos nós na lista de controle.	ok
		Verificação da contagem dos elementos da lista.	ok
OnClick	OnMouseDown()	Verificação da mudança da booleana após clique/toque no objeto.	ok

Classe	Método	Teste realizado	Situação
OnClick	OnMouseUp()	Verificação da mudança da booleana após finalização do clique/toque no objeto.	ok
	OnMouseEnter()	Verificação da mudança da booleana após posicionamento do cursor do mouse sobre o objeto.	ok
	OnMouseExit()	Verificação da mudança da booleana após posicionamento do cursor do mouse fora do objeto.	ok
Output	OnOff()	Verificação da desativação da lâmpada apagada para corrente no valor esperado e menor que corrente máxima estipulada.	ok
		Verificação da ativação da lâmpada acesa para corrente no valor esperado e menor que corrente máxima estipulada.	ok
		Verificação da ativação da lâmpada apagada para corrente superior a corrente máxima estipulada.	ok
		Verificação da desativação da lâmpada acesa para corrente superior a corrente máxima estipulada.	ok
		Verificação da ativação da lâmpada apagada para corrente inferior a corrente esperada.	ok
		Verificação da desativação da lâmpada acesa para corrente inferior a corrente esperada.	ok
		Verificação da ativação do áudio para corrente no valor esperado e menor que corrente máxima estipulada.	ok
		Verificação da desativação do áudio para corrente superior a corrente máxima estipulada.	ok
		Verificação da desativação do áudio para corrente inferior a corrente esperada.	ok

Classe	Método	Teste realizado	Situação
Output	OnOff()	Verificação da rotação do ponteiro do voltímetro da bancada para posição de circuito energizado.	ok
		Verificação da rotação do ponteiro do voltímetro do <i>canvas</i> para posição de circuito energizado.	ok
		Verificação da rotação do ponteiro do voltímetro da bancada para posição de circuito desenergizado.	ok
		Verificação da rotação do ponteiro do voltímetro do <i>canvas</i> para posição de circuito desenergizado.	ok
Pratica	Start() e Update()	Verificação do fluxo de integração da classe.	ok
SalvarPosic()	Awake()	Verificação do armazenamento do nome da cena ativa em variável.	ok
	Start()	Verificação de carregamento das posições salvas nas chaves do eixo X, Y e Z.	ok
	SalvarLocalizacao()	Verificação do armazenamento de chave com a posição do eixo X.	ok
		Verificação do armazenamento de chave com a posição do eixo Y.	ok
		Verificação do armazenamento de chave com a posição do eixo Z.	ok
SelecaoObjeto	Selecionado()	Verificação da aplicação do material de seleção após clique/toque no objeto.	ok
	MaterialPreto()	Verificação da aplicação do material preto no objeto selecionado.	ok
	MaterialVermelho()	Verificação da aplicação do material vermelho no objeto selecionado.	ok
	MaterialAzul()	Verificação da aplicação do material azul no objeto selecionado.	ok

Classe	Método	Teste realizado	Situação
SelecaoObjeto	MaterialVerde()	Verificação da aplicação do material verde no objeto selecionado.	ok
TeclaOnOff	OnOff()	Verificação da desativação do componente <i>MeshRenderer</i> da tecla <i>off</i> após clique/toque no objeto.	ok
		Verificação da desativação do componente <i>BoxCollider</i> da tecla <i>off</i> após clique/toque no objeto.	ok
		Verificação da ativação do componente <i>MeshRenderer</i> da tecla <i>on</i> após clique/toque na tecla <i>off</i> .	ok
		Verificação da ativação do componente <i>BoxCollider</i> da tecla <i>on</i> após clique/toque na tecla <i>off</i> .	ok
		Verificação do <i>reset</i> da booleana <i>ClicouNoObjeto</i> após clique/toque na tecla <i>off</i> .	ok
		Verificação do <i>reset</i> da booleana <i>ClicouNoObjeto</i> da tecla <i>on</i> após clique/toque na tecla <i>off</i> .	ok
		Verificação da ativação do componente <i>MeshRenderer</i> da tecla <i>off</i> após clique/toque na tecla <i>on</i> .	ok
		Verificação da ativação do componente <i>BoxCollider</i> da tecla <i>off</i> após clique/toque na tecla <i>on</i> .	ok
		Verificação da desativação do componente <i>MeshRenderer</i> da tecla <i>on</i> após clique/toque no objeto.	ok
		Verificação da desativação do componente <i>BoxCollider</i> da tecla <i>on</i> após clique/toque no objeto.	ok

Classe	Método	Teste realizado	Situação
TeclaOnOff	OnOff()	Verificação do <i>reset</i> da booleana <i>ClicouNoObjeto</i> após clique/toque na tecla <i>on</i> .	ok
		Verificação do <i>reset</i> da booleana <i>ClicouNoObjeto</i> da tecla <i>off</i> após clique/toque na tecla <i>on</i> .	ok
		Verificação da desativação do componente <i>MeshRenderer</i> do pulsador na posição <i>off</i> após clique/toque no objeto.	ok
		Verificação da ativação do componente <i>MeshRenderer</i> do pulsador na posição <i>on</i> após clique/toque no objeto.	ok
		Verificação da ativação do componente <i>MeshRenderer</i> do pulsador na posição <i>off</i> após término do clique/toque no objeto.	ok
	Verificação da desativação do componente <i>MeshRenderer</i> do pulsador na posição <i>on</i> após término do clique/toque no objeto.	ok	
	ValRes()	Verificação do valor da resistência atribuída ao modelo elétrico de acordo com a posição da tecla dos interruptores.	ok
Timer	Start()	Verificação do carregamento do tempo armazenado na cena anterior.	ok
		Verificação do início de contagem em 0 segundos caso não haja tempo armazenado.	ok
	Update()	Verificação da contagem retroativa para a booleana regressivo igual <i>true</i> .	ok
		Verificação da contagem progressiva para a booleana regressivo igual a <i>false</i> .	ok
		Verificação do armazenamento do tempo decorrido.	ok

Classe	Método	Teste realizado	Situação
Timer	DesativaTimer()	Verificação da desativação do script Timer.	ok
	AtivaTimer()	Verificação da ativação do script Timer.	ok
Trigger	OnTriggerEnter()	Verificação da mudança da variável booleana <i>trigger</i> para <i>true</i> após colisão com objeto.	ok
	OnTriggerExit()	Verificação da mudança da variável booleana <i>trigger</i> para <i>false</i> após fim da colisão com objeto.	ok
Update_A	Atual_A()	Verificação da inserção da corrente na matriz de incidência de acordo com a ordem de conexão dos ramos e nós.	ok
VolumeControl	Start()	Verificação do carregamento do volume <i>Master</i> armazenado na chave.	ok
		Verificação do carregamento do volume <i>Master default</i> quando não possuir valor armazenado.	ok
		Verificação do carregamento do volume <i>Player</i> armazenado na chave.	ok
		Verificação do carregamento do volume <i>Player default</i> quando não possuir valor armazenado.	ok
		Verificação do carregamento do volume <i>Background</i> armazenado na chave.	ok
		Verificação do carregamento do volume <i>Background default</i> quando não possuir valor armazenado.	ok
	VolumeMaster()	Verificação do ajuste do volume após mudança do <i>slider</i> correspondente.	ok
		Verificação do armazenamento do volume ajustado.	ok
	VolumePlayer()	Verificação do ajuste do volume após mudança do <i>slider</i> correspondente.	ok

Classe	Método	Teste realizado	Situação
VolumeControl	VolumePlayer()	Verificação do armazenamento do volume ajustado.	ok
	VolumeBackGround()	Verificação do ajuste do volume após mudança do <i>slider</i> correspondente.	ok
		Verificação do armazenamento do volume ajustado.	ok
Wires	Conexao()	Verificação da ativação das 163 conexões após clique/toque dos soquetes.	ok
		Verificação da desativação das 163 conexões após novo clique/toque dos soquetes.	ok

ANEXO A – PESQUISA DE ACESSIBILIDADE REMOTA



MINISTÉRIO DA
EDUCAÇÃO



DEBPCC-PESQUISA DE ACESSIBILIDADE REMOTA -2020.1

Cursos Técnicos Integrados ao Ensino Médio



- CURSO TÉCNICO EM AUTOMAÇÃO INDUSTRIAL
- CURSO TÉCNICO EM EDIFICAÇÕES
- CURSO TÉCNICO EM ELETROTÉCNICA
- CURSO TÉCNICO EM ELETROTÉCNICA-PROEJA
- CURSO TÉCNICO EM INFORMÁTICA
- CURSO TÉCNICO EM MECÂNICA

Cursos Técnicos Concomitantes



- CURSO TÉCNICO EM AUTOMAÇÃO INDUSTRIAL
- CURSO TÉCNICO EM EDIFICAÇÕES
- CURSO TÉCNICO EM ELETROTÉCNICA
- CURSO TÉCNICO EM ESTRADAS
- CURSO TÉCNICO EM INFORMÁTICA
- CURSO TÉCNICO EM MECÂNICA
- CURSO TÉCNICO EM QUÍMICA
- CURSO TÉCNICO EM TELECOMUNICAÇÕES

Cursos Técnicos Subsequentes



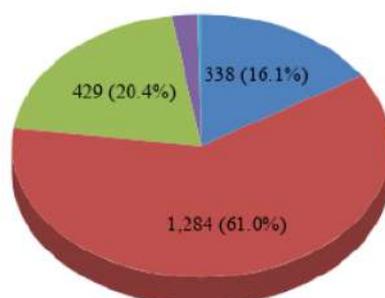
- CURSO TÉCNICO EM SEGURANÇA DO TRABALHO

CURSOS TÉCNICOS *CAMPUS* CAMPOS CENTRO

PESQUISA DE ACESSIBILIDADE REMOTA 2020

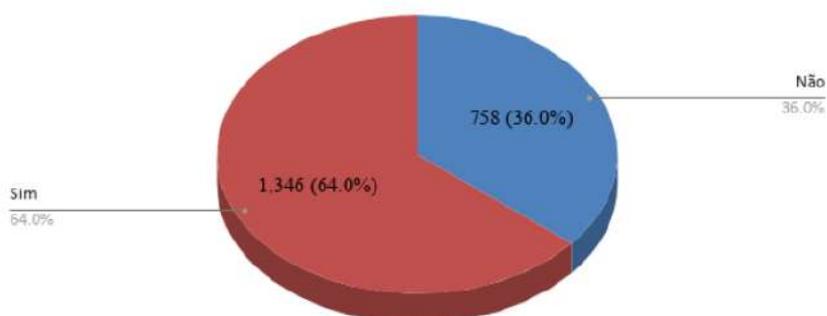
Respondentes: 81% dos matriculados nos Cursos Técnicos do campus Campos Centro em 2020.1

1. Onde você está residindo durante o período de pandemia?



● Em um município vizinho, no Estado do RJ
 ● Na zona urbana de Campos
 ● Na zona rural de Campos
● Em um município distante, no Estado do RJ
 ● Em outro estado

2. Na sua residência durante o período de pandemia, você possui computador com internet disponível para estudar?





INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Fluminense

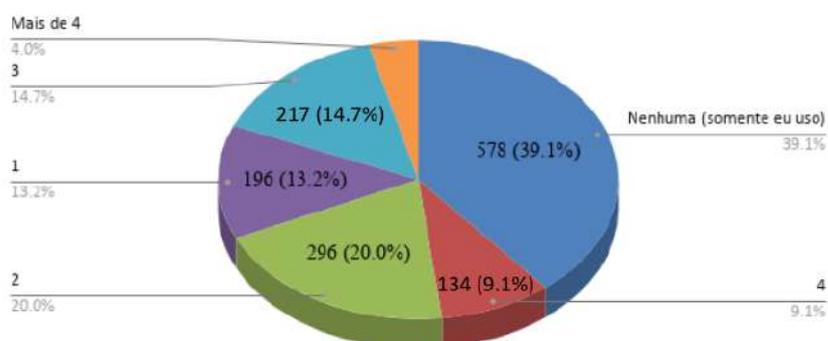
MINISTÉRIO DA
EDUCAÇÃO



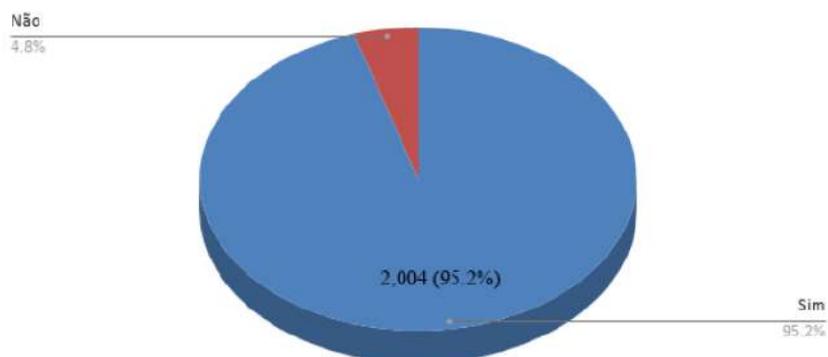
CURSOS TÉCNICOS CAMPUS CAMPOS CENTRO
PESQUISA DE ACESSIBILIDADE REMOTA 2020

Respondentes: 81% dos matriculados nos Cursos Técnicos do campus Campos Centro em 2020.1

3. Caso possua computador disponível, ele é compartilhado com quantas pessoas?



4. Você possui telefone celular (smartphone) com internet?





INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Fluminense

MINISTÉRIO DA
EDUCAÇÃO

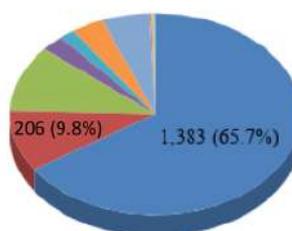


CURSOS TÉCNICOS *CAMPUS* CAMPOS CENTRO

PESQUISA DE ACESSIBILIDADE REMOTA 2020

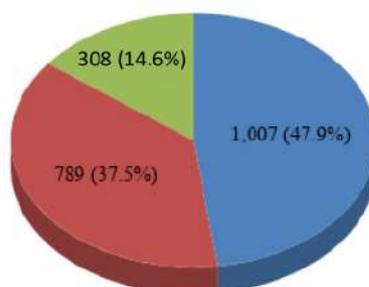
Respondentes: 81% dos matriculados nos Cursos Técnicos do campus Campos Centro em 2020.1

5. Na sua residência durante o período de pandemia, que tipo de conexão de internet você possui?



- Banda larga fixa
- Dados móveis (3G ou 4G) com pacote de dados pequeno
- Não sei o tipo de conexão que possuo
- Dados móveis (3G ou 4G) com pacote de dados grande
- Não possuo conexão de internet
- Banda larga fixa, Dados móveis (3G ou 4G) com pacote de dados grande
- Banda larga fixa, Dados móveis (3G ou 4G) com pacote de dados pequeno

6. Você se sente em condições de estudar por meio de Ensino à Distância (EaD) na sua residência durante o período de pandemia?



- Talvez eu consiga, posso tentar
- Tenho condições suficientes
- Não tenho condições



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Fluminense

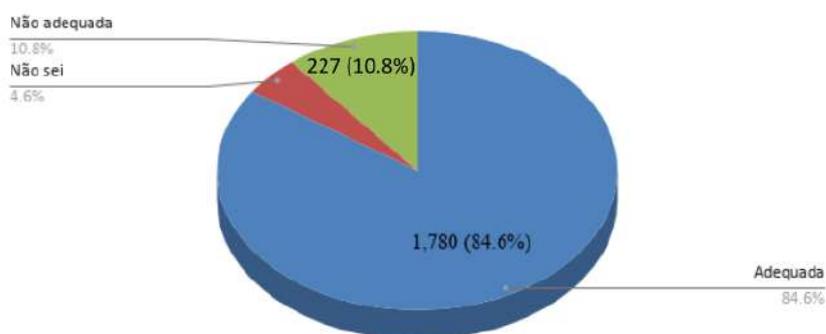
MINISTÉRIO DA
EDUCAÇÃO



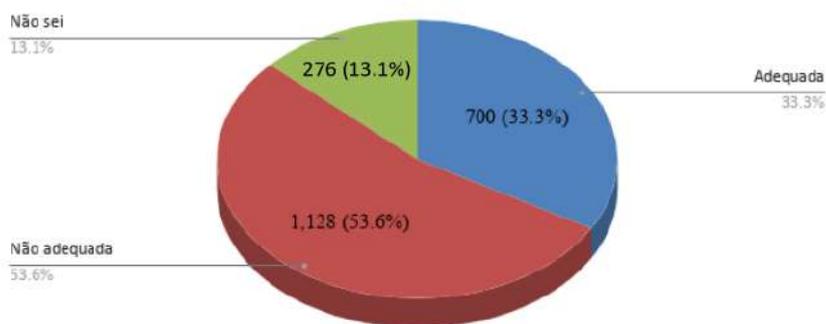
CURSOS TÉCNICOS CAMPUS CAMPOS CENTRO
PESQUISA DE ACESSIBILIDADE REMOTA 2020

Respondentes: 81% dos matriculados nos Cursos Técnicos do campus Campos Centro em 2020.1

7. De acordo com as condições que tem em sua residência, indique sua opinião sobre uso de Vídeos na internet (links)



8. De acordo com as condições que tem em sua residência, indique sua opinião sobre uso de material copiado em meio digital

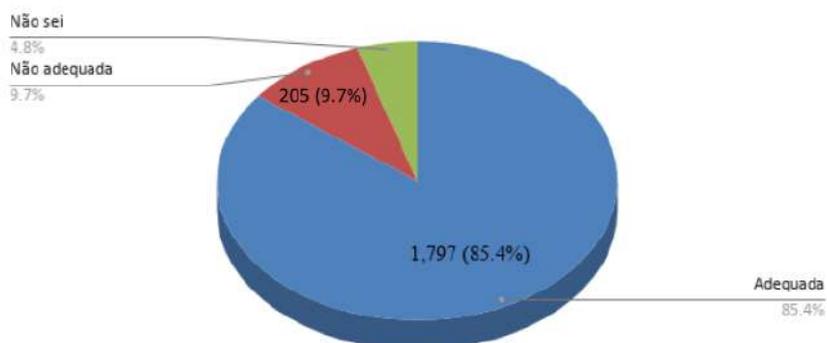


CURSOS TÉCNICOS *CAMPUS* CAMPOS CENTRO

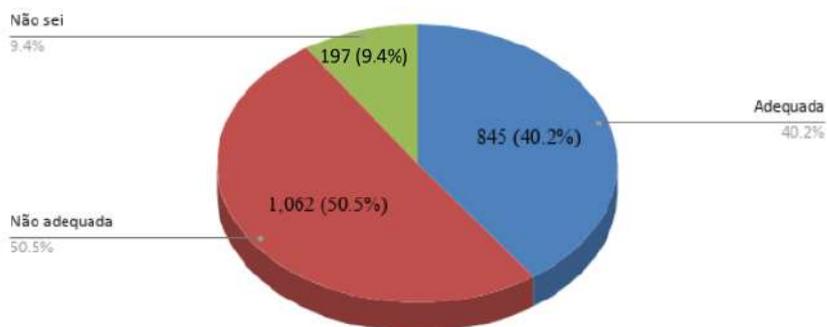
PESQUISA DE ACESSIBILIDADE REMOTA 2020

Respondentes: 81% dos matriculados nos Cursos Técnicos do campus Campos Centro em 2020.1

9. De acordo com as condições que tem em sua residência, indique sua opinião sobre o uso de material para download



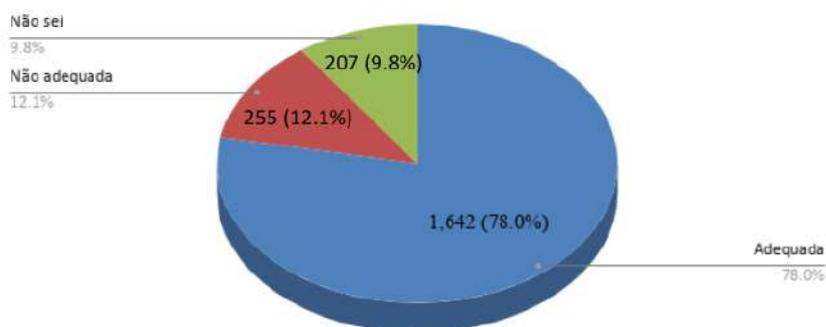
10. De acordo com suas condições, indique sua opinião sobre uso de material impresso a ser retirado no campus



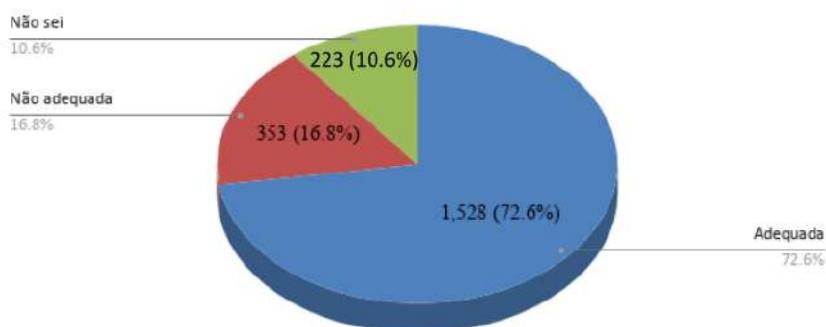
CURSOS TÉCNICOS *CAMPUS* CAMPOS CENTRO
PESQUISA DE ACESSIBILIDADE REMOTA 2020

Respondentes: 81% dos matriculados nos Cursos Técnicos do campus Campos Centro em 2020.1

11. De acordo com suas condições, indique sua opinião sobre uso de material no Sistema Acadêmico



12. De acordo com suas condições, indique sua opinião sobre uso de material em Ambiente Virtual de Aprendizagem

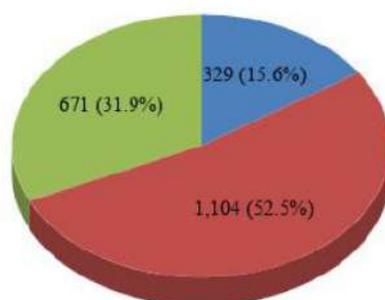


CURSOS TÉCNICOS *CAMPUS* CAMPOS CENTRO

PESQUISA DE ACESSIBILIDADE REMOTA 2020

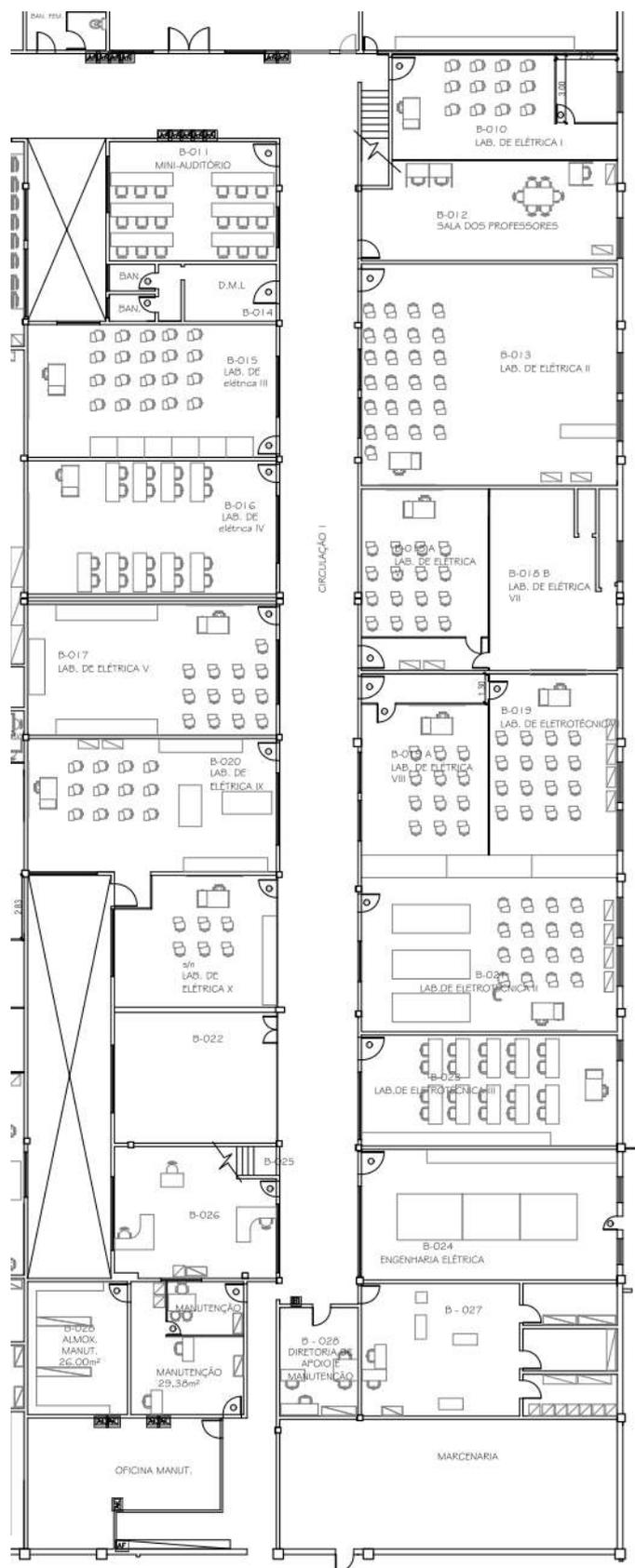
Respondentes: 81% dos matriculados nos Cursos Técnicos do campus Campos Centro em 2020.1

13. Qual é a sua opinião em relação à situação atual do seu curso?

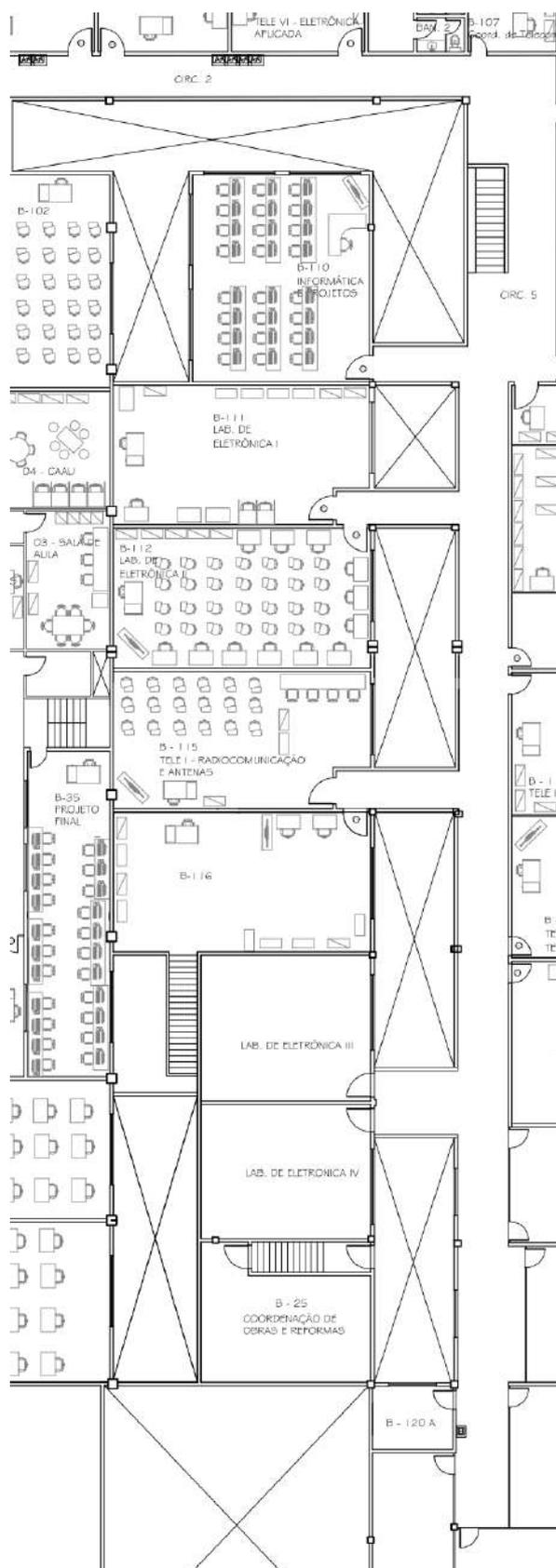


- Não tenho opinião formada sobre isso
- Eu prefiro aguardar a liberação das atividades presenciais
- Eu gostaria de iniciar imediatamente as aulas, mesmo na modalidade à distância

ANEXO B – PLANTA BAIXA DO BLOCO B



Bloco B Térreo



Bloco B 1º Pavimento