



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RÔMULO IORIO FERREIRA DA COSTA

**UTILIZAÇÃO DE INTERATIVIDADE ENTRE *SOFTWARES* MCAD E ECAD PARA
DESENVOLVIMENTO DE EQUIPAMENTOS ELETRÔNICOS: APLICAÇÃO EM
APRESENTADOR MULTIMÍDIA CONTROLADO COM OS PÉS**

FORTALEZA

2019

RÔMULO IORIO FERREIRA DA COSTA

UTILIZAÇÃO DE INTERATIVIDADE ENTRE *SOFTWARES* MCAD E ECAD PARA
DESENVOLVIMENTO DE EQUIPAMENTOS ELETRÔNICOS: APLICAÇÃO EM
APRESENTADOR MULTIMÍDIA CONTROLADO COM OS PÉS

Monografia apresentada ao Curso de Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito para obtenção do título de Bacharel em Engenharia Elétrica.

Área de concentração: Eletrônica

Orientador: Prof. Dr. René Pastor Torrico
Bascope

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- C875u Costa, Rômulo Iorio Ferreira da.
Utilização de interatividade entre softwares mcad e ecad para desenvolvimento de equipamentos eletrônicos : aplicação em apresentador multimídia controlado com os pés / Rômulo Iorio Ferreira da Costa. – 2019.
112 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, 3, Fortaleza, 2019.
Orientação: Prof. Dr. René Pastor Torrico Bascopé.
1. Equipamentos Eletrônicos. 2. CAD Mecânico. 3. CAD Eletrônico. 4. Interatividade entre MCAD e ECAD. 5. Apresentador Multimídia. I. Título.

CDD

RÔMULO IORIO FERREIRA DA COSTA

UTILIZAÇÃO DE INTERATIVIDADE ENTRE *SOFTWARES* MCAD E ECAD PARA
DESENVOLVIMENTO DE EQUIPAMENTOS ELETRÔNICOS: APLICAÇÃO EM
APRESENTADOR MULTIMÍDIA CONTROLADO COM OS PÉS

Trabalho de conclusão de curso apresentado à
banca examinadora como requisito à obtenção
do título de bacharelado em Engenharia Elétrica.

Área de concentração: Eletrônica

Orientador: Prof. Dr. René Pastor Torrico
Bascope

Aprovada em: ___ / ___ / ____.

BANCA EXAMINADORA

Prof. Dr. René Pastor Torrico Bascope (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Ing. Diego de Sousa Madeira
Universidade Federal do Ceará (UFC)

Eng. Reginaldo Florêncio da Silva
Universidade Estadual do Ceará (UECE)

FORTALEZA

2019

A todos aqueles que, na ânsia pelo conhecimento, buscam novas formas de realizar as mesmas tarefas, aprendendo cada vez mais.

AGRADECIMENTOS

À minha noiva Jacqueline Nishimura, por estar sempre comigo em todos os momentos, de bom humor e sempre paciente, me apoiando e colaborando para nosso crescimento e sucesso.

À minha mãe, Rosana Iorio Ferreira, e à minha tia avó, Ana Maria Iorio Dias, por terem oferecido sua experiência na área de educação e me ajudado na escrita e revisão textual do trabalho.

Aos meu avós, Angelina Iorio e Edmilson Barbosa, por todo o apoio e incentivo que dão e sempre deram ao longo de minha vida.

Ao meu pai Roger Caminha Bret, pelo apoio e por todos os ensinamentos ao longo de minha vida.

À toda minha família, por sempre me apoiarem e incentivarem na perseguição dos meus objetivos.

Ao meu Orientador, Prof. Dr. René Pastor Torrico Bascopé pela sua solicitude, seu tempo, paciência, por sempre estar disponível para tirar dúvidas e pela orientação deste trabalho.

Ao meu gestor de estágio e membro da banca, Reginaldo Florêncio da Silva, pela disponibilidade do seu tempo, paciência e orientação, bem como aos ensinamentos que levaram ao desenvolvimento deste trabalho.

Ao Prof. Dr. Ing. Diego de Sousa Madeira, membro da banca, que prontamente aceitou o convite para defesa deste trabalho.

A minha gerente, Lidiane Castro Silva, pelo incentivo e autorização do desenvolvimento deste trabalho.

Ao Prof. Dr. Francisco Carlos de Mattos Brito Oliveira, coordenador geral de pesquisas do Laboratório de Educação a Distância para Pessoas com Deficiência, pelo incentivo ao desenvolvimento deste trabalho.

A todo o pessoal do Laboratório de Educação a Distância para Pessoas com Deficiência e da Dell Technologies, que tornaram possível a minha participação no projeto que levou a realização deste trabalho.

A todos meus amigos que fiz na faculdade e fora dela pelos momentos de descontração, troca de conhecimentos, apoio, incentivo e companheirismo.

A todo o pessoal do Programa de Educação Tutorial (PET), programa que contribuiu imensamente para o meu crescimento pessoal e profissional, sempre prestando apoio, quando

necessário, mesmo após minha saída do programa, e aos membros do Programa que se tornaram grandes amigos.

Aos técnicos de laboratório do Departamento de Engenharia Elétrica da UFC, que muito ajudaram nas práticas de laboratório e pela solicitude, auxílio e ensinamentos que muito contribuíram para minha formação.

A todos os servidores da UFC e à secretaria do curso de Engenharia Elétrica, que me auxiliaram quando necessário durante o curso.

A todo o corpo docente do Departamento de Engenharia Elétrica da UFC, que contribuíram para o meu crescimento e aprendizado durante o curso.

À Dell Technologies e ao IDESCO, pelo fomento e incentivo financeiro ao desenvolvimento do projeto que levou ao desenvolvimento deste trabalho.

"There are naïve questions, tedious questions, ill-phrased questions, questions put after inadequate self-criticism. But every question is a cry to understand the world. There is no such thing as a dumb question." (Carl Sagan)

RESUMO

Este trabalho visa descrever e analisar o método de interatividade entre *softwares Mechanical Computer Aided Design* (MCAD) e *Electronical Computer Aided Design* (ECAD) para desenvolvimento de equipamentos eletrônicos. A utilização conjunta desses dois tipos de *softwares* visa suprir o que apenas os *softwares* ECAD não são capazes de fornecer ao desenvolvimento de projetos, como análise de interferências espaciais no posicionamento dos componentes e simulações térmicas. No trabalho, são explicados os conceitos necessários do uso deles e seus métodos interativos. Além disso, é feita a descrição da aplicação dos métodos a um apresentador multimídia controlado com os pés. Com o qual verificam-se de forma mais prática os benefícios, pois é possível observar o diferencial da utilização cooperativa de um *software* MCAD, possibilitando a definição do posicionamento adequado dos componentes, obtendo uma visão prévia do projeto, através da utilização de modelos 3D. O apresentador multimídia controlado com os pés tem motivação inclusiva, visto que possibilita a utilização de apresentadores multimídia à pessoas com deficiências no movimento das mãos, ou mesmo que não as possuam.

Palavras – Chave: Equipamentos Eletrônicos. CAD Mecânico. CAD Eletrônico. Interatividade MCAD e ECAD. Apresentador Multimídia.

ABSTRACT

This work aims to describe an interactive method between Mechanical and Electrical Computer Aided Design (MCAD and ECAD), as well as highlighting its importance in electronic equipment design. The combined use of these two types of software aims to supplement what ECAD applications alone can't supply, such as spatial interferences analysis in the positioning of components, and thermal simulations. In this work, there are explained the main concepts, how to use the two pieces of software and how to use the interactive method. Besides that, an application of the method is described, in which it is built a feet controlled multimedia presenter, designed following the method, making it possible to see the advantages of using a MCAD software along with ECAD, making it possible to define the appropriate positions of the components, and being able to see a preview of the project, through the use of 3D models. In addition, the feet controlled multimedia presenter is also inclusive, as it makes it possible its usage to people with disabilities in the movement of hands, or even to people without hands.

Key words: Electronic Equipment. Mechanical CAD. Electrical CAD. MCAD - ECAD interactivity. Multimedia Presenter.

LISTA DE FÍGURAS

Figura 1 – Fases de projeto mecânico.....	7
Figura 2 – Metodologia de projeto mecânico.....	9
Figura 3 – Exemplo de diagrama Y para um projeto de um circuito integrado.	16
Figura 4 – Exemplo de diagrama Y para um projeto de um relógio de pulso.....	21
Figura 5 – Apresentador multimídia da empresa <i>Targus</i>	27
Figura 6 – Apresentador multimídia <i>SlClick</i>	28
Figura 7 – Ambiente do <i>Altium Designer 19</i>	30
Figura 8 – Criação de projetos no <i>Altium Designer 19</i> :	31
Figura 9 – Criação de biblioteca integrada.....	31
Figura 10 – Criação de biblioteca de esquemático.	32
Figura 11 – Criação da biblioteca de PCI	33
Figura 12 – Criação de arquivo de esquemático.....	34
Figura 13 – Exemplo de <i>template</i>	34
Figura 14 – Criação de um arquivo de PCI.	35
Figura 15 – Ambiente de visualização 3D do <i>Altium Designer</i>	35
Figura 16 – Ambiente de criação de modelos 3D do <i>Altium</i>	36
Figura 17 – Ambiente de inicialização do <i>Autodesk Inventor</i>	38
Figura 18 – Ambiente de gerenciamento de projetos.....	39
Figura 19 – Criação de peça: (a) <i>menu</i> de criação de arquivos novos; (b) barra de ferramentas superior.....	40
Figura 20 – Barra superior do ambiente de criação de peça do tipo padrão.....	41
Figura 21 – Planos XY, YZ e XZ para criação de um esboço 2D.	41
Figura 22 – Aba esboço (<i>sketch</i>) da barra superior.....	42
Figura 23 – Descrição de um comando com o mouse sobre ele.....	42
Figura 24 – Aba modelo 3D (<i>3D model</i>) da barra superior.	43

Figura 25 – Criação de montagens: (a) <i>menu</i> de criação de arquivos novos; (b) barra de ferramentas superior.....	46
Figura 26 – Barra de ferramentas superior de montagem.	47
Figura 27 – Tipos de comandos de confinamento.....	48
Figura 28 – <i>Menu</i> de parâmetros.	50
Figura 29 – TIP42C com dissipador HS3520_20.....	51
Figura 30 – <i>Menu</i> de ambientes (<i>environments</i>), mostrando a extensão <i>Autodesk Inventor Nastran</i>	51
Figura 31 – Janela de criação de nova análise térmica.	52
Figura 32 – Subitem materiais na árvore de <i>design</i>	53
Figura 33 – Criação de novo material.	53
Figura 34 – Subitem materiais na árvore de <i>design</i> com novo material criado.....	54
Figura 35 – Criação de sólido.	54
Figura 36 – Criação de malhas.....	55
Figura 37 – Criação de carga de geração de calor.....	56
Figura 38 – Criação de carga de convecção.	57
Figura 39 – Resultado da simulação térmica. (a) Com dissipador de calor; (b) Sem dissipador de calor.....	58
Figura 40 – Exportação de um arquivo do <i>Inventor</i> em <i>.step</i>	60
Figura 41 – Exportação de um modelo 3D de um componente.....	60
Figura 42 – Importação de um modelo 3D das dimensões e formato de uma PCI. (a) Importação; (b) Modelo 3D com dimensões e formato da PCI importado.....	61
Figura 43 – PCI montada no <i>Altium Designer</i>	61
Figura 44 – Exportação de um modelo 3D de uma PCI em <i>.step</i> no <i>Altium Designer</i>	62
Figura 45 – Importação da PCI com os componentes para o <i>Autodesk Inventor</i>	62
Figura 46 – <i>Case</i> com PCI de exemplo encaixada.	63
Figura 47 – Invólucro do receptor.	68

Figura 48 – Invólucro do receptor com esboço de PCI.....	68
Figura 49 – Invólucro (caixa metálica) do transmissor.....	69
Figura 50 – Invólucro (caixa metálica) do transmissor com esboço de PCI e componentes de interação com o usuário.	69
Figura 51 – Esquemático do circuito do receptor.....	70
Figura 52 – Esquemático do circuito do transmissor (a) Alimentação e proteção; (b) Microcontrolador e <i>bluetooth</i> ; (c) Entradas e saídas digitais.	71
Figura 53 – Projeto mecânico concluído (a) Receptor; (b) Transmissor.....	74
Figura 54 – Projeto eletrônico concluído (a) Receptor; (b) Transmissor.....	76
Figura 55 – Transmissor do protótipo funcional.....	78
Figura 56 – Receptor do protótipo funcional.....	79
Figura 57 – Vista em destaque das laterais do transmissor.....	79
Figura 58 – Detalhe da troca de bateria.	80
Figura 59 – Luz azul acesa esperando conexão.	80
Figura 60 – RGB central verde indicando a carga e LED direito aceso indicando o botão direito apertado.....	81
Figura 61 – RGB central verde indicando a carga e LED esquerdo aceso indicando o botão esquerdo apertado.	81

LISTA DE ABREVEATURAS E SIGLAS

BOM	Bill of Material
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacture
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNC	Computer Numeric Control
ECAD	Electronic Computer Aided Design
EDA	Electronic Design Automation
FEA	Finite Element Analysis
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IOT	Internet of Things
MCAD	Mechanical Computer Aided Design
MDA	Mechanical Design Automation
NIST	National Institute of Standards and Technology
PCB	Printed Circuit Board
PCI	Placa de Circuito Impresso

SUMÁRIO

INTRODUÇÃO GERAL	1
1 MOTIVAÇÃO, OBJETIVOS E METODOLOGIA DE PESQUISA	3
1.1 Motivação	3
1.2 Objetivos	4
1.2.1 <i>Objetivo geral</i>	4
1.2.2 <i>Objetivos específicos</i>	4
1.3 Metodologia	4
1.4 Estrutura do Trabalho	5
2 ESTADO DA ARTE.....	6
2.1 <i>Softwares MCAD e Projetos Mecânicos</i>	6
2.1.1 <i>Projetos Mecânicos</i>	6
2.1.2 <i>CAD Mecânico</i>	10
2.1.3 <i>Autodesk</i>	12
2.2 <i>Softwares ECAD e Projetos Eletrônicos</i>	14
2.2.1 <i>Projetos Eletrônicos</i>	14
2.2.2 <i>CAD Eletrônico</i>	18
2.2.3 <i>Altium Designer</i>	21
2.3 <i>Placas de Circuito Impresso</i>	22
2.3.1 <i>Breve Histórico</i>	22
2.3.2 <i>Métodos de Fabricação</i>	23
2.4 <i>Metodologia e Tecnologias Utilizadas</i>	25
2.4.1 <i>Interatividade ECAD e MCAD</i>	25
2.4.2 <i>Arduino</i>	26
2.4.1 <i>Módulo Bluetooth HC05</i>	26
2.5 <i>Apresentadores Multimídia</i>	27
3 ANÁLISE QUALITATIVA DO MÉTODO E SOFTWARES UTILIZADOS	29
3.1 <i>Software ECAD Utilizado</i>	29
3.1.1 <i>Características Gerais do Altium Designer</i>	29
3.1.2 <i>Noções Básicas de Criação de Projetos Eletrônicos</i>	30
3.1.3 <i>Limitações para o Desenvolvimento de Equipamentos Eletrônicos</i>	36
3.2 <i>Software MCAD Utilizado</i>	37

3.2.1	<i>Características Gerais do Autodesk Inventor</i>	37
3.2.2	<i>Noções Básicas de Criação de Projetos Mecânicos</i>	38
3.2.3	<i>Simulação Térmica com o Inventor Nastran</i>	50
3.3	<i>Aplicação da Interatividade MCAD e ECAD</i>	59
3.3.1	<i>Complementariedade Entre os Softwares</i>	59
3.3.2	<i>Vantagens do Uso Complementar dos Softwares</i>	63
4	APRESENTADOR MULTIMÍDIA CONTROLADO COM OS PÉS	65
4.1	<i>Introdução Conceitual</i>	65
4.1.1	<i>Breve Descrição do Projeto</i>	65
4.1.2	<i>Diferencial Observado</i>	66
4.2	<i>Pré-Projeto Mecânico do Apresentador Multimídia</i>	67
4.2.1	<i>Adequação Espacial dos Componentes Eletrônicos</i>	67
4.2.2	<i>Projeto dos Invólucros dos Componentes do Apresentador Multimídia</i>	68
4.3	<i>Pré-Projeto Eletrônico do Apresentador Multimídia</i>	70
4.3.1	<i>Definição dos Circuitos Eletrônicos e Criação dos Esquemáticos</i>	70
4.3.2	<i>Programação dos Microcontroladores</i>	72
4.4	<i>Aplicação do Método de Interatividade MCAD e ECAD</i>	73
4.4.1	<i>Pré-projeto Mecânico no Autodesk Inventor</i>	73
4.4.2	<i>Pré-projeto Eletrônico da PCI no Altium Designer</i>	74
4.4.3	<i>Finalização do Projeto Mecânico no Autodesk Inventor</i>	74
4.4.4	<i>Finalização do Projeto Eletrônico da PCI no Altium Designer</i>	75
5	RESULTADOS E TRABALHOS FUTUROS	77
5.1	<i>Discussão Sobre os Softwares Utilizados</i>	77
5.2	<i>Apresentação do Protótipo Funcional da Aplicação</i>	78
5.3	<i>Discussão e Análise dos Resultados Obtidos</i>	81
5.4	<i>Trabalhos Futuros</i>	82
	CONCLUSÃO GERAL	84
	REFERÊNCIAS	85
	ANEXO A – CÓDIGO DO RECEPTOR	88
	ANEXO B – CÓDIGO DO TRANSMISSOR	91

INTRODUÇÃO GERAL

Este trabalho visa analisar a eficácia e a eficiência do método de interatividade entre *softwares Mechanical Computer Aided Design (MCAD)* e *Electronical Computer Aided Design (ECAD)* para desenvolvimento de equipamentos eletrônicos. Este método propõe a utilização conjunta de *softwares MCAD* para planejamento mecânico de projetos deste tipo de equipamento, previamente ao desenvolvimento da Placa de Circuito Impresso (PCI) em *softwares ECAD*, uma vez que essa é apenas um componente de um projeto maior, não sendo o único detalhe importante do projeto.

Este planejamento mecânico prévio do projeto e ambiente que comportará a PCI é realizado com a finalidade de observar e adequar a disposição mecânica dos componentes eletrônicos da PCI, buscando otimizar a posição de componentes importantes à interação com o usuário, como botões, telas, conectores, dentre outros. Em algumas empresas, esta atividade é dividida entre projetistas mecânicos e eletrônicos, o que pode acarretar atrasos operacionais e possíveis dificuldades relacionadas à comunicação entre os profissionais.

Diante disso, percebe-se que o projeto pode ser otimizado caso a parte mecânica menos complexa, quanto à estrutura mecânica da PCI, puder ser também realizada pelo projetista eletrônico, visto que este tipo de atividade requer, dos conceitos mecânicos, somente o conceito de desenho e modelagem 3D. Embora este conceito não seja abordado nos cursos de engenharia elétrica e eletrônica diretamente, pode ser obtido em atividades extracurriculares, não é tão complexo como outros conceitos inerentes a engenharia mecânica, como esforços e transferência de calor, e pode ser de grande valia para o engenheiro eletricista e eletrônico como projetista. Porém, caso o engenheiro eletricista ou eletrônico tenha interesse, e tenha utilidade para o projeto desenvolvido, o que costuma acontecer, no caso de projetos de eletrônica, também é possível realizar simulações térmicas do projeto.

As PCIs são placas com interior constituído de fenolite, um material plástico usado como isolante elétrico, e com superfície de cobre, utilizado como condutor elétrico. São amplamente utilizadas na indústria para desenvolvimento de circuitos eletrônicos com espaço e funcionamento otimizados. Sua utilização se dá por meio de impressão, através de técnicas específicas, daí o nome, de um *layout*, com *pads* e furações e trilhas, para soldar os componentes e conectá-los, a fim de reproduzir na superfície dessa placa os circuitos projetados.

Softwares ECAD, também chamados de *Electronical Design Automation* (EDA), são *softwares* de design eletrônico auxiliado por computador, que possibilitam o desenvolvimento de PCIs. Nestes *softwares*, primeiramente, define-se os esquemáticos do projeto, que são utilizados inclusive em manuais de montagem e manutenção, que devem conter os circuitos do projeto, preferencialmente divididos em blocos estruturais, de acordo com a função do circuito, facilitando o entendimento. Em seguida, é utilizada outra função de ECAD, a de projetar PCIs, definindo a posição dos componentes, furações e trilhas, que possibilitarão que a PCI tenha o funcionamento desejado, após fabricada e soldados os componentes.

Softwares MCAD, também chamados de *Mechanical Design Automation* (MDA), são *softwares* de design mecânico auxiliado por computador, que possibilitam o desenho 3D de objetos. É amplamente utilizado por engenheiros mecânicos, técnicos e projetistas para o desenvolvimento de peças, máquinas e equipamentos. Além disso, estes *softwares* também costumam conter a função de realizar simulações de variáveis mecânicas através de métodos de elementos finitos, para verificar diversas situações de funcionamento, como simulações de esforços, simulações térmicas, dentre outras. Para os engenheiros eletrônicos e eletrônicos, bem como projetistas eletrônicos de forma geral, estes *softwares* podem ser utilizados para auxiliar na organização e simulação espacial dos componentes de um projeto eletrônico, anteriormente ao projeto da PCI no *software* ECAD, e também para simulação térmica, verificando a temperatura prevista de operação dos componentes do circuito.

Para realização deste trabalho, foi aplicado o método de interatividade MCAD e ECAD para desenvolvimento de um apresentador multimídia controlado com os pés. Observou-se a diferença substancial na otimização espacial do projeto quando é aplicado este método, pois pode-se organizar os componentes, buscando o melhor posicionamento de elementos de interação com o usuário, no caso, conectores, botões, LEDs e acesso à bateria. Também foi possível o desenvolvimento, também em 3D, de invólucros para os dois componentes do projeto, tornando-o próximo de um produto real.

1 MOTIVAÇÃO, OBJETIVOS E METODOLOGIA DE PESQUISA

Nesta seção, são apresentadas as motivações que levaram ao desenvolvimento deste trabalho, os objetivos esperados com este, a metodologia de pesquisa utilizada e a estrutura geral do trabalho.

1.1 Motivação

O incentivo deste trabalho surgiu durante o desenvolvimento de um projeto de estágio em um laboratório de pesquisa que é uma colaboração entre uma instituição pública e uma empresa privada da área de indústria eletrônica. Neste projeto, foi desenvolvido um apresentador multimídia controlado com os pés. Este trabalho tem como foco o método utilizado para desenvolvimento do projeto como um equipamento eletrônico, sendo analisado o método de desenvolvimento e o projeto tratado como uma aplicação, a fim de mostrar a funcionalidade do método por meio de um produto criado com sua utilização.

Busca-se mostrar a eficácia e eficiência da utilização do método de interatividade entre *softwares* MCAD e ECAD para desenvolvimento de equipamentos eletrônicos. Para este fim, são apresentados os *softwares*, e comparadas a utilização separada e em conjunto destes. Além disso, apresenta-se como aplicação o desenvolvimento de um apresentador multimídia controlado com os pés, buscando mostrar as melhorias obtidas. Com isso, tem-se como finalidade o incentivo da utilização desse método para melhorar a produtividade de projetistas durante o desenvolvimento de equipamentos eletrônicos, possibilitando uma maior otimização da produção.

Durante a execução do projeto, foi possível perceber as vantagens da aplicação deste método, o que levou ao desenvolvimento deste trabalho. Dentre elas, pode-se destacar a otimização da posição dos componentes. Por exemplo, caso desconsiderada a aplicação do método, os componentes de uma PCI poderiam estar com uma disposição não muito favorável, principalmente referente à posição de elementos de interação com o usuário, ou seja, conectores, botões, LEDs e conector de bateria.. Enquanto isso, com a aplicação do método, otimizou-se e adequou-se a posição dos componentes dessa mesma PCI, sendo possível visualizar através de modelos 3D, uma prévia do produto montado.

1.2 Objetivos

1.2.1 *Objetivo geral*

Comprovar a eficácia e a eficiência da utilização do método de interatividade entre *softwares* MCAD e ECAD para desenvolvimento de equipamentos eletrônicos com diversas finalidades, enfatizando a aplicação em um apresentador multimídia controlado com os pés.

1.2.2 *Objetivos específicos*

- Descrever o método de interatividade entre os *softwares* MCAD e ECAD utilizados para desenvolvimento de equipamentos eletrônicos;
- Analisar semelhanças e diferenças entre os *softwares* MCAD e ECAD e sua utilização no desenvolvimento de equipamentos eletrônicos;
- Apresentar a importância do método para desenvolvimento de equipamentos eletrônicos, destacando a aplicação em um apresentador multimídia controlado com os pés.

1.3 Metodologia

A pesquisa é de um Trabalho de Conclusão de Curso, com metodologia conceitual, qualitativa, iniciada com levantamento de referências bibliográficas, que abordou autores que trabalharam o tema e são referência para conceituação e aprimoramento do assunto. Este trabalho foi realizado durante o desenvolvimento de um projeto de estágio em um laboratório de pesquisa de uma empresa do ramo da indústria eletrônica. Foi utilizado um método específico para desenvolvimento de equipamentos eletrônicos, sendo feita uma aplicação em um apresentador multimídia controlado com os pés. No parágrafo seguinte é descrito o procedimento realizado.

Primeiramente, para desenvolvimento deste trabalho, fez-se um levantamento bibliográfico sobre publicações envolvendo conceitos pertinentes ao trabalho, como projeto e CAD mecânico, projeto e CAD eletrônico, conceito e utilização de *softwares* MCAD e ECAD, e as demais tecnologias utilizadas nas seguintes bases de dados: Google Acadêmico, repositório

on-line do Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), periódicos Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), dentre outros. Em seguida, começou-se o desenvolvimento do projeto e, durante este, através de observação, verificou-se que seria melhor otimizado ao aplicar o método discutido neste trabalho. Então, observou-se as diferenças obtidas com a aplicação do método, verificando os resultados satisfatórios de aplicar o método estudado.

1.4 Estrutura do Trabalho

Esta seção apresenta a estrutura geral do trabalho, bem como uma breve descrição do conteúdo presente em cada seção.

O presente trabalho apresenta a seguinte estrutura:

- Seção I – Introdução inicial ao tema, justificativa, objetivo geral, objetivos específicos e metodologia de pesquisa.
- Seção II – Referencial teórico do estado da arte de conceitos relacionados ao uso da interatividade MCAD e ECAD para desenvolvimento de equipamentos eletrônicos. Apresentação e discussão da importância desses para este trabalho.
- Seção III – Análise qualitativa dos *softwares* utilizados na interatividade MCAD e ECAD para desenvolvimento de equipamentos eletrônicos. Apresentação dos *softwares*, suas funcionalidades básicas e sua importância para este trabalho, bem como as características de interatividade entre os *softwares* utilizados, caracterizando o método.
- Seção IV – Aplicação do método no desenvolvimento de um apresentador multimídia controlado com os pés.
- Seção V – Apresentação dos resultados e conclusões obtidos com o desenvolvimento deste trabalho.

2 ESTADO DA ARTE

Nesta seção é apresentado o estado da arte de conceitos importantes para o desenvolvimento deste trabalho. Isto é, a revisão de referências e material relacionado a trabalhos já feitos em cima destes conceitos.

2.1 *Softwares MCAD e Projetos Mecânicos*

Esta subseção trata de conceitos relacionados a engenharia mecânica, que são abordados de forma mais abrangente, para possibilitar arcabouço teórico para o desenvolvimento do trabalho. Além disso, nessa seção também é feita a apresentação e discussão do *software* MCAD utilizado neste trabalho.

2.1.1 *Projetos Mecânicos*

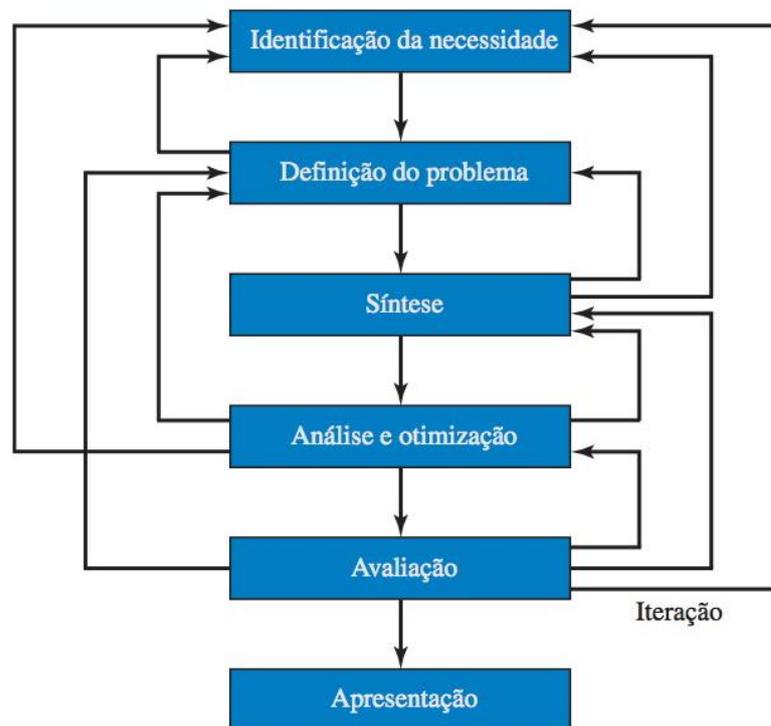
Os *softwares* MCAD são ferramentas computacionais utilizadas para auxílio no desenvolvimento de projetos mecânicos. Um projeto mecânico, como qualquer outro tipo de projeto, é o desenvolvimento de uma solução de um problema dividido em etapas. Pode ser utilizado, por exemplo, para desenvolvimento de um produto inovador, que pode vir a ser comercializado. Diante disso, o projeto deve ser bem executado e otimizado para minimizar despesas, levando a um preço final mais barato e assim melhorando a competitividade, mas também deve ser confiável e próprio para uso, capaz de ser fabricado e replicado. Um fator fundamental de todo projeto é a tomada de decisões. (Budynas; Nisbeth, 2016, p. 4)

Ainda segundo Budynas e Nisbeth (2016, p. 5), um projeto de engenharia mecânica está relacionado à produção e processamento de energia, fornecimento dos meios de produção, ferramentas de transporte e técnicas de automação. Conceitos muito importantes ao desenvolvimento desses projetos são os de características de materiais, resistências, esforços e transferências de forças em objetos, mecânica de sólidos e fluidos e termodinâmica. Assim como qualquer projeto de engenharia, inclusive os de elétrica e eletrônica, um problema pode ser tratado como uma *caixa preta*, com entradas e saídas, que devem ser avaliadas para resolução do problema.

Os autores citados no parágrafo anterior ainda dividem a ideia de um projeto em fases, conforme mostra a Figura 1. Um projeto é iniciado com a identificação da necessidade, isto é, a motivação, o que levou o engenheiro a desenvolver este projeto. Em seguida, definição do problema, é a ideia mais específica do que consiste o projeto, o que ele irá resolver, e todas as especificações de projeto que surgem com isso, algo que se relaciona com a ideia de tratar o projeto como uma “caixa preta”, já mencionada. Após essas fases fundamentais para o início do projeto, vem a síntese, que é o desenvolvimento conceitual do projeto, bem como a pesquisa de mercado de componentes comerciais que possam ser utilizados, facilitando a fabricação do projeto.

A etapa de análise e otimização envolve formulação de um modelo matemático do projeto, que possibilitam simulações teóricas do funcionamento deste. Na etapa de avaliação, já é considerada a criação de um protótipo e realização de testes e simulações reais do projeto. Por fim, a etapa de apresentação representa a transmissão do projeto a terceiros e, possivelmente, ao mercado. Como a imagem também sugere, essas etapas não são seguidas linearmente, e pode haver iterações para melhoria do resultado final projeto.

Figura 1 – Fases de projeto mecânico.



Por fim, como todo projeto, os projetos mecânicos precisam levar em conta algumas considerações de projeto, que são essenciais ao desenvolvimento do projeto, sobretudo na etapa de análise e otimização, pois representam algumas variáveis que devem ser levadas em conta no decorrer deste. Algumas considerações importantes em projetos de engenharia mecânica, segundo Budynas e Nisbeth (2016, p. 8), são funcionalidade, resistência/tensão, distorção/deflexão/rigidez, desgaste, corrosão, segurança, confiabilidade, fabricabilidade, custo, atrito, dentre outras. Pode-se observar que existem tanto considerações mais específicas da área, que fogem ao escopo deste trabalho, quanto considerações mais gerais, que devem ser levadas em conta em todo projeto de engenharia.

Já segundo Norton (2013, p. 4), a ideia principal do projeto de máquinas consiste na função do engenheiro de definir e calcular movimentos, forças e mudanças de energia, relacionadas a produção de trabalho útil e transferência de energia entre elementos de máquina, que são as peças constituintes desta. Isto deve ser feito com a finalidade de determinar as dimensões, formas e materiais necessários para cada peça, bem como testar o funcionamento integrado destas no projeto. Além disso, é de suma importância realizar simulações e testes de características mecânicas como as já citadas por Budynas e Nisbeth (2016) como considerações de projeto, isto é, resistência dos materiais, análises de tensão e deflexão, dentre outras.

Norton (2013, p. 5) também comenta sobre a importância do processo de iteração. Durante o projeto, deve ser definida uma configuração inicial para cada peça e, com essa, realizar uma primeira análise dinâmica do projeto das forças, momentos e torques do projeto, para em seguida calcular as tensões e deformações das peças. O autor fala que normalmente na primeira tentativa, o projeto costuma não se mostrar satisfatório, pois o material não suporta as tensões calculadas. Diante disso, é necessária a iteração, que é refazer o projeto alterando formatos, medidas, matérias e outros fatores das peças, até que o projeto seja aceitável. Observamos que isto tem coerência com o que foi explicado por Budynas e Nisbeth (2016), que diante das etapas de projeto, são necessárias iterações, sobretudo nas etapas de síntese, análise e otimização e avaliação.

Norton (2013) também divide o projeto em etapas, e chama esse processo de metodologia de projetos. Pode-se observar na Figura 2 que as etapas apresentadas por este autor são bem semelhantes as apresentadas por Budynas e Nisbeth (2016), havendo modificações de nomenclatura, e adição de novas etapas, mas, em síntese, segue a mesma ideia. Por exemplo,

as etapas de pesquisa de suporte, definição dos objetivos e especificações de tarefas podem ser tidas como subdivisões da fase de definição do problema definida pelo outro autor. Bem como as etapas de análise, seleção e projeto detalhado podem ser englobadas na fase de análise e otimização do outro autor. O autor ainda reforça a ideia já mencionada da importância da iteração em cima dessas etapas, não sendo normalmente um projeto linear.

Figura 2 – Metodologia de projeto mecânico.

Tabela 1-1 Metodologia de projetos

1	Identificação da necessidade
2	Pesquisa de suporte
3	Definição dos objetivos
4	Especificações de tarefas
5	Síntese
6	Análise
7	Seleção
8	Projeto detalhado
9	Protótipo e teste
10	Produção

Fonte: Norton, 2013, p. 6

Outro detalhe importante citado por Norton (2013, p. 6), é que, considerando que a ideia do processo de iteração é melhorar o projeto, este processo poderia continuar indefinidamente, sempre iterando para obter melhoras, por mais que os processos já tenham sido avaliados como satisfatórios. Diante disso, o autor falar que, nesse caso, deve-se observar que com o tempo os ganhos de função ou redução do custo tenderão a se tornar menos significativos . Por isso, em algum momento o projeto deve ser avaliado como satisfatório e executado. Não há problema deixar margens para melhorias feitas por outros projetistas ou mesmo pelo próprio, com a ideia de criação de versões. Por isso, projetos de objetos que já existem a bastante tempo e já foram melhorados por muitos projetista diferentes, se tornam cada vez mais difíceis de obterem novas melhorias significativas.

2.1.2 CAD Mecânico

Conforme Ullman, et al (1990), CAD é a sigla para *Computer Aided Design*, que pode ser traduzido como “Desenho Auxiliado por Computador”. Assim, *softwares* CAD têm como função ajudar no desenvolvimento de projetos com o uso de um computador e plataformas específicas. Isto é, o que antes do advento destes *softwares* era feito à mão, com uso de réguas, esquadros, transferidores e outras ferramentas de desenho à mão, passou a ser feito em um programa de computador. O CAD Mecânico é a utilização de CAD para desenvolvimento de projetos mecânicos.

Estes programas permitem a facilidade tanto na criação quanto na modificação de um projeto, com comandos para fins específicos e opções de desfazer ou modificar etapas do desenvolvimento. Além disso, esses *softwares* também trazem uma grande facilidade na visualização e apresentação do projeto, facilitando a interação e demonstração desse para clientes ou colegas de equipe. Ainda, segundo Ullman, et al (1990), as formas de auxílio que as ferramentas de CAD fornecem ao processo de desenvolvimento de um projeto mecânico são funcionar como uma ferramenta de rascunho avançada, auxiliar na visualização do *hardware* produzido e os dados relacionados a este, melhorando a organização dos dados e comunicação, e possibilitando maiores análises e cálculos do projeto, como análises cinemáticas, de propriedades de massa, dentre outras.

Conforme Norton (2013, p. 11), os primeiros *softwares* de CAD eram programas de desenho de vistas 2D de um objeto 3D, de forma semelhante ao realizado a mão. Desse modo, era definido apenas um modelo com as bordas da peça, chamado modelo de aramado, ou *wireframe*. Como citado por Xiao-Qing, Jia-Xing (2009), o advento dos CADs 3D foi como uma revolução nessa tecnologia, permitindo que, em vez de apenas múltiplas vistas 2D de um objeto 3D, pudesse agora ser desenvolvido o próprio objeto 3D, sendo possível, quando necessário, exportar as vistas 2D do objeto com cotas e outros detalhes. Além disso, esse autor também ressalta como esta mudança estimulou a inovação, acelerando o processo de prototipagem e diminuindo o ciclo de produção.

Ainda segundo Norton (2013, p. 11-13), outra grande vantagem obtida com os CADs 3D foi a interpretação do objeto 3D como um modelo sólido, com arestas e faces definidas, bem como a informação de “preenchimento”, ou seja, de volume. Isso possibilitou os cálculos

via *software* de informações mais complexas acerca do objeto, como cálculos dependentes da massa, como o centro de massa, de acordo com informações de densidade do material ou mesmo do próprio material, que já contém sua densidade nos bancos de dados do programa. Além disso, outra funcionalidade de suma importância nos projetos mecânicos foi a possibilidade de cálculos de análise por método dos elementos finitos (*Finite Element Analysis*, ou FEA) no próprio *software* CAD, permitindo cálculos de tensões, vibrações e transferência de calor.

Este método, segundo Tavares e Padilha (1998), é uma importante ferramenta que discretiza um sistema, modelando-o por um número finito de elementos menores a fim de obter uma aproximação de certo parâmetro com relação ao sistema com base neste parâmetro com relação a cada um desses elementos menores. Conforme citado por Gil (2015), este método tem uma grande vantagem que é a aproximação de equações diferenciais por um sistema de equações algébricas, o que reduz o esforço computacional de *softwares* de simulação, enquanto mantém uma boa aproximação.

Na verdade, segundo Norton (2013, p. 14) e Budynas e Nisbeth (2016, p. 9), CAD é um subconjunto da Engenharia Auxiliada por Computador (*Computer Aided Engineering*, ou CAE). Além disso, muitos *softwares* CAD de hoje, que possuem mais funcionalidades, como a análise por FEA, são exemplos de CAE. Alguns *softwares* citados pelos autores como *softwares* CAD e/ou CAE são *SolidWorks*, antigo *ProEngineer* (hoje PTC Creo Elements/Pro), o antigo Nastran (hoje incorporado pela *Autodesk* como *Autodesk Nastran* e também *Nastran In-CAD*) e o *Ansys*. Além disso, por conhecimento e certa familiaridade do autor, foi escolhido falar também sobre o *Autodesk Inventor* e o *CATIA*.

Além disso, também existe o conceito de Manufatura Auxiliada por Computador (*Computer Aided Manufacture*, ou CAM), diretamente relacionado com os outros dois e comumente utilizado em conjunto. Analogamente aos anteriores, este conceito consiste na utilização de ferramentas computacionais para auxiliar na manufatura de um projeto. Na verdade, CAM também é um subconjunto de CAE, sendo utilizado de forma complementar e após o desenho do modelo 3D do projeto em um *software* CAD, para produção de protótipos e produtos finais. Diante disso, hoje em dia tornou-se difícil separar estes conceitos, visto que muitos *softwares* satisfazem não apenas um, mas todos estes conceitos.

2.1.3 Autodesk

Autodesk é uma empresa estadunidense, fundada em 1982, que, segundo *Autodesk* (2019) produz *softwares* voltados para o ramo de produção. Os *softwares* dessa empresa são amplamente utilizados no ramo de engenharia, sobretudo como *softwares* CAD e CAE. Esta subseção tratará de dois *softwares* que serão utilizados neste trabalho, o *Inventor* e o *Inventor Nastran*. Uma vantagem dessa empresa é a disponibilização do uso completo e gratuito de muitos de seus *softwares* para estudantes e educadores, sendo possível realizar o cadastro no *site*, indicando sua universidade e posteriormente tendo acesso a *download* de diversas versões dos *softwares*, via solicitação.

2.1.4.1 Inventor

Segundo Waguespack (2013), o *Autodesk Inventor*, foi criado em 1999 como um *software* MCAD de modelagem 3D paramétrica. Desde então, este tem melhorado suas ferramentas de modelagem, desenho, montagem e visualização. A modelagem 3D realizada neste *software* é dita paramétrica, pois parâmetros podem ser criados e utilizados para desenvolvimento e modificação do modelo. Ou seja, a utilização de parâmetros permite que seja criada uma peça base cujas dimensões são funções de um ou mais parâmetros, que, caso alterados, alterarão automaticamente a peça, facilitando a modificação. Porém, há limitações de modificação dos parâmetros dependendo de como a peça foi criada.

Ainda segundo Waguespack (2013), uma peça bem construída depende de um *sketch* 2D, ou esboço 2D bem construído. Em outra seção deste trabalho, será detalhado o conceito de *sketch* 2D utilizado pelo *Inventor*, bem como por outros *softwares* MCAD. Mas, basicamente, os esboços 2D são utilizados para produzir e modificar o modelo 3D da peça. Por isso, se houver um esboço com inconsistências, isso acarretará prejuízo para a peça como um todo, o que justifica a afirmação do autor. O autor também ressalta que uma boa prática para evitar inconsistências é que os esboços sejam mais simplificados.

Além disso, Waguespack (2013), também fala sobre uma funcionalidade diferencial do *software*, que é permitir que uma peça seja criada com base em medidas de outra peça dentro de uma mesma montagem, utilizadas como parâmetros. Ou seja, durante o processo de montagem, ou criação de um *assembly*, é possível a criação de uma peça dependente de uma

ou mais características da outra peça já existente, de modo que, quando a primeira for modificada, a segunda é modificada automaticamente, caso não haja inconsistências. Uma vantagem observável disto é numa situação em que haja encaixes, por exemplo, em que, caso se modifique a peça base, a peça que encaixa nela se modificará automaticamente.

Outra funcionalidade importante do *software*, também comentada por Waguespack (2013), é a possibilidade de alterar a aparência e o material do objeto. Embora a primeira seja apenas visual, é muito importante para apresentação do projeto para clientes e/ou colegas de equipe, auxiliando bastante no entendimento do que está sendo visualizado. Já a última, também possui implicação visual, mas a sua principal implicação é nos cálculos relacionados a massa e análise por elementos finitos, que o *software* também permite. O *Inventor* possui uma biblioteca padrão de aparências e materiais, que contém materiais comuns e suas características físicas como densidade. Mas também é possível criar sua própria aparência ou material personalizado, descrevendo as características físicas que são utilizadas em cálculos de FEA.

2.1.3.2 *Inventor Nastran*

O *Inventor Nastran*, conforme Verma (2018), é um *software* para cálculos de FEA, integrado no *Autodesk Inventor*. Este *software* é uma integração do *Autodesk Nastran* para que este seja utilizado dentro do mesmo ambiente do *Inventor*, e possibilita simulação de diversas áreas, como esforços lineares e não lineares, simulações dinâmicas e de transferência de calor. As simulações são realizadas para estudar os efeitos causados por condições em que um produto será submetido no mundo real. O *software* simula a aplicação de condições predefinidas, através do método de análise por elementos finitos em um modelo e verifica os resultados.

Com estes resultados, é possível prever o funcionamento de um projeto em situações reais, corrigir erros que levaram a falhas e/ou adaptá-lo melhor de acordo com uma necessidade específica. Por exemplo, numa peça que constantemente sofrerá esforços mecânicos devido a movimento ou necessidade de suportar peso, pode ser verificado se essa peça realmente aguentará a carga aplicada ou se romperá, o que, caso ocorra, necessitará alterações no projeto. Outro exemplo de simulação, mais possível de ser utilizado por projetistas eletrônicos, são as simulações térmicas, de transferência de calor, que permitem verificar se a energia térmica de um sistema está sendo corretamente dissipada ou se há acúmulos, que poderiam vir a danificar componentes.

2.2 *Softwares ECAD e Projetos Eletrônicos*

Esta subseção trata de conceitos relacionados a engenharia elétrica, que serão abordados de forma mais abrangente, para possibilitar arcabouço teórico para o desenvolvimento do trabalho. Além disso, nessa seção também é feita a apresentação e discussão sobre alguns *softwares* ECAD comumente utilizados.

2.2.1 *Projetos Eletrônicos*

Um projeto eletrônico tem uma estruturação semelhante a discutida na subseção 2.1.1 (sobre projeto mecânico), também podendo ser dividido em etapas lógicas, visto que também é um projeto de engenharia. Molina (2006) define o projeto industrial como estético e funcionalmente ergonômico, além de um artefato que realize as funções para o qual foi projetado. O autor cita que a função estética também tem uma importância significativa, visto que projetos eletrônicos costumam ser pensados como produtos, que possíveis clientes devam ter interesse em compra-lo. Além disso, é ressaltada a necessidade de ergonomia, devido a limitações legais para comercialização de produtos a serem utilizados por pessoas.

No artigo de Molina (2006), são analisados dois projetos desenvolvidos numa disciplina de uma universidade. Esses dois projetos foram divididas em etapas, que podem ser divididas mais genericamente como: pesquisa bibliográfica; diagrama de blocos e simulações; validação do circuito proposto (em *proto-board*); construção do protótipo; apresentação do projeto e relatório. Pode-se observar a semelhança com as etapas descritas na subseção 2.1.1, sendo a etapa de pesquisa bibliográfica relacionada com as etapas de identificação de necessidade e definição do problema, visto que, ao definir o projeto, deve-se realizar pesquisas bibliográficas para fundamentar a sua realização e auxiliar no seu desenvolvimento, através de projetos semelhantes e satisfatórios já desenvolvidos e documentados.

Ainda comparando as etapas de projeto eletrônico, segundo Molina (2006), e as etapas de projeto mecânico, segundo Budynas e Nisbeth, (2016), observa-se que a etapa de diagrama de blocos e simulações está relacionada com as etapas de síntese e análise e otimização, visto que, um diagrama de blocos é um desenvolvimento conceitual do projeto, assim como na síntese, e as simulações são exatamente a parte matemática e teórica relacionada a etapa de análise e

otimização. Ademais, a etapa de construção de protótipo é exatamente do que trata a etapa de avaliação, que é construir um protótipo funcional do projeto.

Por fim, as etapas de apresentação do projeto e relatório são relacionadas com a etapa de apresentação. Também é importante ressaltar que no projeto eletrônico, como no projeto mecânico e, na verdade, em todo projeto de engenharia, também existe a necessidade de iteração entre as etapas, buscando o aperfeiçoamento e solução de problemas, algo que ocorre comumente durante o desenvolvimento de um projeto. Além disso, bem como no desenvolvimento de um projeto mecânico existem as ferramentas e *softwares* MCAD, para os projetos eletrônicos existem os softwares de ECAD.

Molina (2006) ainda destaca a característica de que ao desenvolver projetos, os engenheiros não devem buscar a solução “perfeita”, mas sim a mais adequada e que resolva bem aquele problema em questão. Ou seja, como discutido na seção 2.1.1, um projeto não deve buscar a perfeição, caso contrário, a etapa de iteração duraria para sempre e o projeto nunca seria concluído. Em vez disso, deve-se observar a relação entre esforços e ganhos de cada nova iteração, e, se for o caso, até adotar a estratégia de criação de versões de um projeto, sendo a primeira mais básica, mas suficiente para resolver o problema abordado.

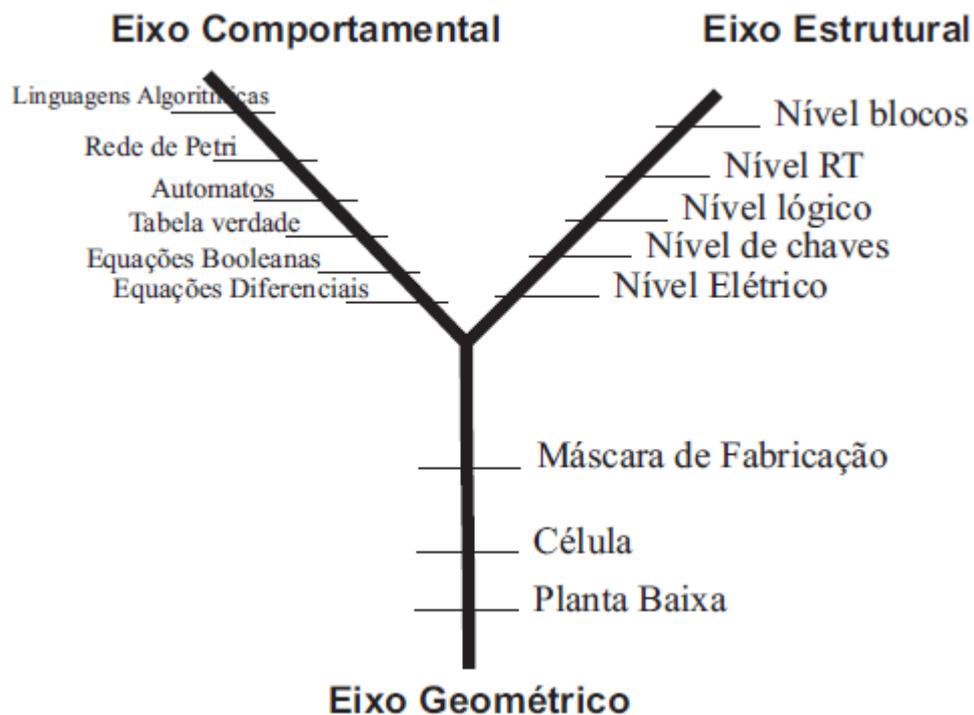
De Oliveira e Shin-Ting (2006), definem essas etapas comuns a projetos de engenharia como “ciclo de vida do projeto”, o que começa com o desenvolvimento do projeto e pode continuar além da implementação deste através de novas versões ou manutenção. Os autores ainda ressaltam a importância do tempo e custo do projeto, além de sua aceitabilidade, e que o tempo pode ser reduzido com a utilização de ferramentas CAD, CAM e CAE, conforme apresentado neste trabalho. Também é discutido sobre como o custo pode ser elevado de acordo com o tempo de desenvolvimento, reforçando a ideia de que a partir de um certo ponto, no que tange à relação entre esforços e ganhos de cada melhoria, a primeira versão de um projeto deve ser entregue, mesmo que ainda haja margem para novas melhorias, que podem ser deixadas para próximas versões.

Os autores citados no parágrafo anterior ainda dizem que é difícil a generalização de um número de etapas de desenvolvimento de um projeto eletrônico e as tarefas realizadas neste. Observa-se que esta afirmação é considerável, pois, só é possível dividir um projeto qualquer em etapas muito generalistas, como já discutido neste trabalho, e, embora essa divisão seja

importante para organização do desenvolvimento de um projeto, ela também tem suas limitações. Como citado pelos autores, as etapas mais específicas de cada projeto são difíceis de ser generalizadas, pois, dependem de diversos fatores, como tecnologia e componentes a serem utilizados, e mesmo a disciplina de trabalho de cada equipe de engenharia de cada empresa.

Desse modo, De Oliveira e Shin-Ting (2006) descrevem o desenvolvimento de um projeto eletrônico através da classificação entre os tipos de atividades realizados, buscando obter uma possibilidade de certa generalização, facilitando a interpretação e execução do projeto, independentemente das diferentes etapas. A classificação descrita é o chamado “Diagrama Y de Gajski-Kuhn”, mostrado na Figura 3, que divide os projetos em três eixos de atividades: eixo comportamental, eixo estrutural e eixo geométrico.

Figura 3 – Exemplo de diagrama Y para um projeto de um circuito integrado.



Fonte: De Oliveira e Shin-Ting (2006)

As atividades de um projeto são organizadas nos eixos de modo que, no extremo externo de cada eixo encontrem-se as descrições mais abstratas de cada tipo de atividade e na junção dos três eixos encontra-se a implementação final do circuito. Ou seja, a interação entre as

atividades de cada eixo aumenta à medida que o desenvolvimento do projeto avança. Como podemos verificar na Figura 3, o eixo comportamental é relacionado às atividades de descrição de como o projeto funciona; o eixo estrutural, como o nome sugere, está relacionado à estrutura do projeto; e o eixo geométrico está relacionado às atividades que dão forma ao projeto.

Os autores De Oliveira e Shin-Ting (2006) também fazem importantes recomendações ao desenvolvimento de projetos mais complexos, destacando a importância de ferramentas computacionais. Porém, mesmo com o uso de boas ferramentas, são dadas outras recomendações, para facilitar o entendimento do próprio projetista, como: projeto igual a documentação, uso de diferentes níveis de abstração e métodos estruturados, descrição precisa e concisa, modularidade, realização dos testes por parte e uso adequado de ferramentas de projeto. Pode-se verificar que estas recomendações são muito importantes e geralmente repassadas como “boas práticas”, auxiliando bastante no desenvolvimento de projetos eletrônicos e/ou de engenharia em geral.

De Oliveira e Shin-Ting (2006) ainda abordam o conceito de Automação do Projeto Eletrônico (do inglês, EDA, *Electronic Design Automation*), um termo amplamente utilizado em conjunto com ou até substituindo ECAD. Este conceito é dividido em três classes de ferramentas: síntese, auxiliando o projetista na criação do projeto, podendo-se destacar ferramentas para desenvolvimento de PCIs, como o *Altium Designer*, *Allegro*, *Ares* e *KiCAD*; análise, auxiliando na verificação e correção, ou seja, destacando-se ferramentas de simulação, como *OrCAD*, *PSIM* e *Proteus Design Suite*; e gerenciamento de informação, facilitando a organização da estrutura de dados de projeto, ação também realizada por alguns dos *softwares* citados, como o *Altium Designer*.

Outro conceito muito importante abordado por De Oliveira e Shin-Ting (2006) é o de validação do projeto, ou seja, realização de testes para confirmação do comportamento do projeto em comparação ao que foi planejado e executado. Destaca-se o termo “testabilidade”, que seria a facilidade de realização de testes adequados em um projeto, e diz-se que, geralmente, essa testabilidade cresce quando o custo de geração ou aplicação de testes num projeto decresce. Desse modo, um “projeto visando testabilidade” integra durante seu desenvolvimento algumas características de testabilidade, que podem ser mantidas e utilizadas durante sua produção/manufatura. Os autores continuam dividindo os testes em tipos de testes e considerando parâmetros de testabilidade.

Os tipos de testes podem ser: paramétricos, avaliação de parâmetros físicos e elétricos, como a medição da tensão sobre e corrente através componentes do circuitos, para verificar se estão em concordância com os cálculos e simulações; exaustivos, em que se testam todas as combinações de estados que o circuito pode assumir, podendo ser amplamente utilizado em circuitos eletrônicos não-sequenciais; estruturais, buscando identificar falhas, exigindo conhecimento da estrutura interna de um circuito; funcional, avaliando um projeto como uma “caixa preta” e verificando se ele cumpre a função para qual foi projetado; testes explícitos, utilizados para aceitação, manutenção e reparos; e técnicas *Ad-hoc*, que buscam melhorar a testabilidade do projeto. Já os parâmetros de testabilidade considerados são: observabilidade, visibilidade e controlabilidade.

Ou seja, pode-se verificar que também existe uma metodologia recomendada para realização de testes no projeto e posterior validação. Embora cada teste tenha sua devida importância, para cada caso deve-se verificar quais tipos de testes devem ser levados em conta e realizados. Se o projeto falhar nesses testes considerados essenciais, deverá voltar a etapa de análise e otimização, possivelmente necessitando simulações, observação do funcionamento do projeto ou até cálculos matemáticos para identificação do problema que levou a falha no teste. Após identificação do problema, este deve ser corrigido, retomada a etapa de testes do ponto que ela havia parado antes de ocorrer o problema. Uma recomendação já citada, que facilita a organização desta etapa, é a realização dos testes por parte, pois facilita a identificação do que levou a uma falha em algum teste.

2.2.2 CAD Eletrônico

Analogamente ao descrito na subseção 2.1.2, CAD Eletrônico, também referido muitas vezes como EDA, é a utilização de CAD para desenvolvimento de projetos eletrônicos. Segundo Birnbaum (2004), EDA é a utilização de ferramentas computacionais para desenvolvimento de circuitos, teste do seu desempenho e verificação do seu comportamento. Ou seja, os *softwares* ECAD são aqueles que facilitam o desenvolvimento de circuitos eletrônicos, seja através do gerenciamento de projeto, simulação ou desenvolvimento de placas de circuito impresso, gerando arquivos para fabricação destas. Comercialmente existem *softwares* que realizam uma ou todas essas funções.

Segundo Jansen (2010), um dos grandes diferenciais que os *softwares* de CAD eletrônico, ou mais abrangente ainda, de CAE eletrônico, é na produção de circuitos integrados com milhões de transistores, o que seria inconcebível sem a utilização dessa tecnologia. O autor ainda cita a possível utilização da sigla CAX (algo como *Computer Aided X*, sendo o X a parte substituível da sigla), onde o X pode ser S de simulação (*simulation*, no inglês), M de produção (*manufacturing*, no inglês), Q de controle de qualidade (*quality assurance*, no inglês), ou P de planejamento (*planning*, no inglês). Desse modo, é possível ver a necessidade e o impacto que a utilização de CAE tem em todas as etapas da produção de equipamentos eletrônicos.

Jansen (2010) também fala sobre a utilização de bibliotecas EDA pré-existentes no desenvolvimento industrial de produtos eletrônicos. Uma biblioteca é um conjunto de informações e elementos de esquemático, simulação e/ou PCB de um ou mais componentes, que existe para facilitar o desenvolvimento de novos circuitos. Como o autor fala, a porcentagem de *design* totalmente novo em um circuito integrado é estimada abaixo de 30%. Além disso, o uso de bibliotecas não se limita a fabricação de circuitos integrados, mas pode auxiliar o desenvolvimento de qualquer tipo de produto eletrônico.

O uso de bibliotecas permite incluir partes já prontas e funcionais, já consolidadas no mercado, em um produto eletrônico com outra parte inovadora. O uso de bibliotecas também é fundamental quanto ao uso de EDA para simulação de circuitos, sendo utilizadas, nesse caso, bibliotecas de componentes já com todos os parâmetros necessários para a simulação. Existem *softwares* que fazem simulações considerando modelos teóricos, com equações aproximadas, de componentes, como o PSIM, e *softwares* que simulam o comportamento mais detalhado de cada componente, como o OrCAD e PSpice, levando em consideração características intrínsecas àquele componente, presentes no *datasheet* deste, que são inseridas em modelos matemáticos.

O autor Jansen (2010) ainda divide as principais áreas de aplicação de CAX em três: aplicações economicamente orientadas; aplicações orientadas ao dispositivo ou a função; e integração de um sistema completamente integrado. As três áreas utilizam a ideia de “modelos de dados”, que são diferentes tipos de informações sobre o produto, em diferentes áreas, mantidos em bases de dados relacionais que podem ser utilizadas em conjunto. Por exemplo, um mesmo produto pode ter vários modelos de dados, referentes a diferentes descrições do produto.

Alguns exemplos de diferentes modelos de dados podem ser listados, referentes: à sua estrutura mecânica, na forma de modelos 3D ou vistas 2D do produto como um todo ou partes deste; à sua estrutura elétrica, na forma de esquemáticos de circuitos; à integração dos sistemas do produto, na forma de desenhos técnicos e instruções de montagem; às funções (ou comportamento) do produto, na forma de manuais, instruções operacionais e de testes; e à características econômicas (ou comerciais), como lista de materiais, preço, e listas de compras.

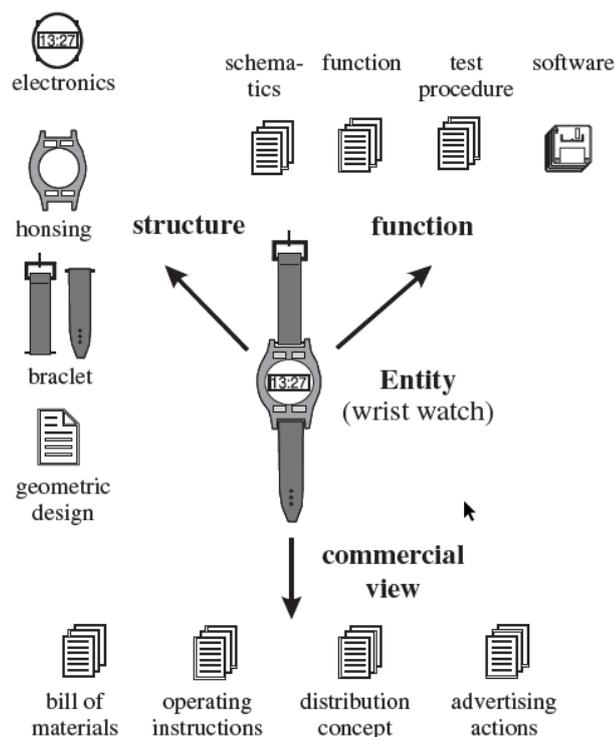
Alguns *softwares* ECAD, como o *Altium Designer*, permitem a integração de quase todos esses modelos de dados num mesmo ambiente, podendo ser exportado para outros *softwares*, caso necessário. Por exemplo, a parte mecânica pode ser exportada no formato universal (.step ou .stp) para *softwares* MCAD, e a parte comercial pode ser exportada como planilha para *softwares* de gerenciamento de dados. Porém, mesmo quando não é possível a utilização de modelos de dados no mesmo *software*, essa abordagem ainda é uma ferramenta diferencial no processo de desenvolvimento de um produto eletrônico.

Além disso, Jansen (2010) também apresenta um modelo de diagrama Y, presente na Figura 4, semelhante ao apresentado por De Oliveira e Shin-Ting (2006), da subseção 2.2.1, presente na Figura 3. O diagrama Y de Jansen (2010), em relação ao apresentado por De Oliveira e Shin-Ting (2006), preserva a ideia dos eixos estruturais e comportamentais (o de função, na figura), mas há a substituição do eixo geométrico pelo eixo comercial. Na verdade, pode-se observar que características mais físicas, que pertenciam ao eixo geométrico, são adicionadas ao eixo estrutural, juntamente com características mecânicas, enquanto os esquemáticos de circuito, que eram considerados do eixo estrutural, no diagrama Y de Jansen (2010), são realocados no eixo comportamental.

Também é considerável ressaltar a importância do eixo comercial do diagrama Y de Jansen (2010), não presente no outro diagrama Y. Esse eixo engloba os modelos de dados comerciais e também alguns comportamentais, como instruções de operação. Pode-se perceber que o diagrama Y apresentado por De Oliveira e Shin-Ting (2006) apresenta uma abordagem mais técnica do produto (no caso do exemplo, um circuito integrado), enquanto o de Jansen (2010), apresenta a parte técnica mais resumida e parte para uma abordagem prática, incluindo características comerciais e de utilização.

Porém, os dois diagramas Y propostos têm importância significativa, sendo cada um adequado para situações diferentes. Por exemplo, o apresentado por De Oliveira e Shin-Ting (2006) pode ser utilizado numa etapa mais técnica do desenvolvimento do projeto, para ajudar a organização e estruturação das tarefas, conforme discutido na subseção 2.2.1, enquanto o diagrama Y de Jansen (2010), pode ser utilizado em situações mais práticas, como apresentação de produtos finais e/ou protótipos, situação em que muitas vezes a parte técnica pode ser abordada de forma mais sucinta, e a parte comercial possui grande importância.

Figura 4 – Exemplo de diagrama Y para um projeto de um relógio de pulso.



Fonte: Jansen (2010)

2.2.3 Altium Designer

Segundo Beltran Jr et al (2014), o *Altium Designer* é um *software* de captura de esquemáticos, simulação e *design* de Placas de Circuito Impresso. O autor apresenta métodos e vantagens do uso desse *software* ECAD no ensino de circuitos eletrônicos. O autor cita a importância do uso de *softwares* de modelagem e simulação, ou seja, ECAD, como base para análise e desenvolvimento de circuitos eletrônicos mais complexos, que apresentariam maior dificuldade para reprodução em laboratório e modelagem matemática sem o auxílio de ferramentas computacionais.

Beltran Jr et al (2014) diz que o *Altium Designer* possibilita dar aos estudantes uma introdução dos conceitos bases do *software*, isto é, *design* de esquemáticos de circuitos e de PCIs e simulação, melhorando a eficiência da etapa de análise. O autor também fala que este *software* é amplamente utilizado para pesquisa, desenvolvimento e experimentação por engenheiros eletrônicos, educadores, desenvolvedores de sistemas embarcados e indústrias ao redor do mundo. Assim como outros *softwares* ECAD, o *Altium Designer* é baseado em ícones que representam os componentes em um circuito, podendo estes ícones serem utilizados com bibliotecas já prontas ou, caso necessário ou desejado, o usuário pode criar suas próprias bibliotecas.

Ainda segundo Beltran Jr et al (2014), o *Altium* tem como vantagem eliminar os passos ditos “desnecessários” na análise de circuitos eletrônicos sem sofrer perda de precisão nem de exatidão dos dados. Os usuários podem, durante o processo de desenvolvimento de circuitos eletrônicos, checar o desempenho e comportamento do sistema. Além disso, no desenvolvimento de PCIs, o *Altium* permite a interação com *softwares* MCAD através da importação de arquivos de modelos 3D dos componentes do circuito, para mostrar o modelo 3D da PCI à medida que esta é montada.

2.3 Placas de Circuito Impresso

2.3.1 Breve Histórico

Atualmente, todo equipamento eletrônico contém uma (ou mais) placas de circuito impresso, que fazem a conexão dos seus componentes para representar os circuitos eletrônicos projetados. Segundo Mouta (2015), o desenvolvimento de processos semelhantes a PCIs começou em meados de 1850, pelo uso de tiras de metal para conectar componentes elétricos em bases de madeira. Posteriormente, as tiras de metal vieram a ser substituídas por fios conectados a terminais e as bases de madeira por estruturas metálicas.

Já em 1925, ainda segundo Mouta (2015), Charles Ducas desenvolveu um método usando processos de galvanização, ou seja, utilização de eletricidade para deposição de material metálico, e um caminho elétrico feito usando uma chapa para estampar e uma tinta condutora. A autora ainda fala que Paul Eisler patenteou um método de impressão de circuitos sobre uma

camada de cobre numa base resistente que não conduzisse eletricidade, começando a ter semelhança com o processo utilizado atualmente.

Segundo Mouta (2015) diz que no fim da Segunda Guerra Mundial, o Instituto Nacional de Padrões e Tecnologia (do inglês, *National Institute of Standards and Technology*, NIST) dos Estados Unidos desenvolveu uma técnica utilizada pelo exército americano semelhante a PCI atual. A técnica consistia em imprimir condutores com pasta de prata e resistores de grafite em substratos cerâmicos, e marcou o início do uso comercial de PCIs. Desde então, foram inventadas e testadas novas técnicas de acordo com as necessidades, buscando melhorar a técnica e os processos de fabricação.

Dentre os exemplos do desenvolvimento de PCIs desde a criação da técnica utilizada pelo exército americano, destacam-se o surgimento de PCIs dupla face, quando a densidade máxima das PCIs simples face alcançaram uma densidade de componentes considerada grande; e, após isso, o aparecimento da técnica de PCIs multicamadas, ou *multilayer PCBs*, por volta de 1956, que permitem a passagem de trilhas em camadas internas das PCIs, permitindo a diminuição subsequente do tamanho das PCIs, e melhor distribuição das trilhas. Além disso, houve melhoras nos materiais utilizados, a fim de melhorar a durabilidade e resistência, bem como as características eletrônicas, das placas.

2.3.2 Métodos de Fabricação

Atualmente, quanto a fabricação de PCIs, existem diferentes métodos, os quais podem ser divididos entre métodos mais adequados para protótipos e métodos mais adequados para produtos finais. Um método comum, utilizado para protótipos ou projetos não comerciais, conforme Branson et al. (2000), consiste na utilização de termo-transferência de tinta em tipos específicos de papel (o tipo de papel pode afetar a qualidade da transferência) para placas de fenolite ou fibra de vidro, com uma fina camada de cobre sobre uma ou ambas as faces (também chamada de “placa de circuito impresso virgem”).

O processo de termo-transferência, também utilizado no ramo de produção têxtil e de roupas, conforme descrito por Dalvey e Nasser (2003), consiste na utilização de tipos específicos de papel, nos quais é feita a impressão da imagem desejada, nesse caso, um *layout* para uma PCI, e uma fonte de calor (que pode ser um ferro de passar ou outro tipo de chapa aquecida), para transferir a imagem impressa no papel para outro substrato, nesse caso, a

superfície da PCI virgem, que contem cobre. Nesse processo, ao ser aquecida, a tinta é transferida de um substrato para outro, saindo do papel e se depositando na superfície com cobre da PCI.

Em seguida, segundo Branson et al. (2000), com o *layout* impresso na PCI, após a termo-transferência, são feitas as devidas correções de pontos que possam não ter sido impressos muito bem, utilizando pincéis de tinta permanente para corrigir trilhas, ou um estilete, para cortar pontos de tinta que não deveriam estar se tocando. Então, é utilizada uma solução aquosa de perclorato de ferro, referida como solução de gravura a água forte (em inglês, *etching solution*), para realizar o processo de corrosão, ou mais genericamente, de gravura a água forte. Nesse processo, a solução de perclorato de ferro, que possui caráter ácido, reagirá com o cobre exposto da PCI virgem, não reagindo com o cobre que está abaixo da tinta, ao final, resultando o *layout* impresso em cobre na PCI.

Este processo, como diz o autor do parágrafo anterior, pode ser acelerado se a solução estiver levemente aquecida, até no máximo 40 °C, e for agitada durante a corrosão. Porém, devem ser tomados os devidos cuidados para evitar que a solução de perclorato de ferro entre em contato com alguns materiais, como pisos ou roupas, visto que esta solução tem propriedade de manchar esses materiais permanentemente. Por se tratar de uma solução ácida, também deve-se tomar os devidos cuidados no contato dessa solução com a pele e olhos. Além disso, se aquecida, a solução pode emitir gases tóxicos, devendo ser utilizado uma máscara protetora para evitar a inalação desses gases, além de a própria solução ser mais reativa quando aquecida, devendo ser redobrados os cuidados.

O processo de corrosão com a solução aquosa de perclorato de ferro é comum aos métodos de fabricação de PCIs para protótipos, mudando apenas o método através do qual a tinta será depositada na superfície com cobre da placa. Outro método também mencionado por Branson et al. (2000), porém referido pelo autor como mais trabalhoso, é o de utilização de tintas sensíveis a luz ultravioleta e lâmpadas de luz ultravioleta para realizar a impressão da tinta no cobre da PCI. Um terceiro método consiste em manualmente, com um pincel de tinta permanente, fazer o *layout* do circuito na placa.

Já outro método, conforme Wise e Engineer (2007), utiliza de uma máquina fresadora tipo CNC (do inglês *Computer Numeric Control*), ou seja, de Controle Numérico

Computadorizado, isto é, uma máquina controlada por algoritmos, que utiliza ferramentas de corte rotatórias para realizar o processo de remoção de material de uma peça, denominado fresagem. O referido método consiste em utilização dos *softwares* de ECAD, como o *Altium Designer*, para transformar o *layout* feito em um tipo específico de arquivo (arquivo *gerber*) que é utilizado para dar comandos a uma fresadora, de modo que ela remova todo o cobre da placa, exceto nos pontos referentes ao *layout* produzido, de modo que ao fim, sobre apenas o *layout* do circuito em cobre na superfície da PCI. Porém, uma desvantagem deste método, é a ausência de anilhas metálicas, ligando trilhas de uma face à outra da PCI.

Além disso, industrialmente também é comum a prática de aplicação de uma camada a mais de material isolante sobre a camada de cobre da PCI, chamada de “máscara”. A aplicação dessa máscara é feita buscando a proteção do circuito contra curto-circuitos acidentais de materiais metálicos nas trilhas do circuito, bem como proteção do próprio circuito contra possíveis descargas eletrostáticas durante o manuseio deste.

2.4 Metodologia e Tecnologias Utilizadas

2.4.1 Interatividade ECAD e MCAD

Segundo Son et al. (2014), quando não há o uso adequado de ferramentas de interatividade MCAD e ECAD, podem haver erros no modelo 3D da PCI, o que pode acarretar erros mais sérios, como furos de fixação de componentes em posições incorretas, interferência espacial entre uma parte mecânica e eletrônica, e até impossibilidade de montagem, devido à falta de espaço adequado entre os componentes. O autor também ressalta a importância da atualização constante entre os modelos 3D dos componentes, também para evitar os erros citados acima.

Man, Pitica e Zolog (2009) falam sobre o Desenvolvimento Integrado de Produto (do inglês *Integrated Product Design*), que basicamente consiste no uso conjunto de ferramentas de MCAD e ECAD, ou seja, na interatividade entre essas ferramentas. Os autores citam que essa interatividade tem sido um fator chave para diminuição de tempos de entrega de produtos e disponibilização do produto ao mercado mais rápido que a concorrência.

Porém, é importante frisar que essa diminuição não necessariamente vem da diminuição do tempo necessário para realizar etapas específicas, mas pela maior integração dos times de desenvolvimento e aplicação de engenharia simultânea. Um exemplo que os autores do parágrafo anterior citam é a análise térmica de PCIs, que, embora demande um tempo maior momentâneo, traz melhorias no desenvolvimento geral do projeto, por diminuir a chance de falhas de origem térmica durante os testes.

Chen e Schaefer (2007), em concordância com os parágrafos anteriores, falam que algumas vantagens da interatividade ECAD e MCAD são reduzir o tempo de produção e disponibilidade para o mercado, permite que os projetistas tenham um maior conhecimento do processo de desenvolvimento como um todo, melhoria na qualidade de produtos e a melhora na produtividade dos projetistas e das empresas. Isso se dá devido ao fato de os projetistas terem conhecimento do projeto como um todo, o que leva ao entendimento das consequências tanto mecânicas quanto eletrônicas do que é feito.

2.4.2 Arduino

Conforme Sirkin, et al (2019), o Arduino é uma plataforma de prototipagem de código aberto (*open source*) baseada em um *hardware* e *software* flexíveis e de fácil uso. Devido a maior facilidade de uso desta plataforma, ela é também amplamente utilizada no ramo do ensino, como cita Sirkin, et al (2019). Além disso, o Arduino é uma das mais conhecidas plataformas de prototipagem utilizadas atualmente, e, dependendo da aplicação, supre bem a necessidade durante a etapa de criação de protótipos e validação de ideias.

2.4.1 Módulo Bluetooth HC05

Segundo Bluetooth (2018), *Bluetooth* é um protocolo de comunicação serial sem fio baseada em ondas de rádio, de baixa potência e baixo alcance. Cada dispositivo *bluetooth* possui um endereço próprio de 48 bits, e existem diversos modos de conexão entre esses dispositivos. O modo utilizado neste trabalho, mestre-escravo (*master-slave*), consiste em um dos dispositivos (mestre) que inicia a conexão com outro (escravo), baseado no endereço deste.

Também é possível utilizar modos de conexão através de senhas, na qual é definida uma senha que deve ser comum aos dois dispositivos para que estes se conectem. Segundo *Bluetooth Module HC05* (2019), o HC05 é um módulo *bluetooth* comumente utilizado para aplicações de curta distância, considerado o método mais barato para transmissão de dados, alcançando velocidades de até 25 Mb/s

2.5 Apresentadores Multimídia

Os apresentadores multimídia disponíveis no mercado, geralmente seguem o formato do presente na Figura 5, da empresa *Targus*, também presente em *Wireless USB Presenter with Laser Pointer (Black)*, 2019. Ou seja, geralmente os dispositivos seguem o formato a um pequeno controle, com botões para desempenhar funções específicas, e um *laser* utilizado com a finalidade de apontar informações na apresentação. As funções dos botões deste apresentador são: avançar e voltar *slides*, escurecer a apresentação, finalizar e iniciar a apresentação de *slides*, e ativar o *laser*, com o botão central. O autor também cita que este equipamento possui um *design* ergonômico.

Além disso, o apresentador multimídia descrito no parágrafo anterior, utiliza radiofrequência da faixa de 2.4 GHz (Giga Hertz) para realizar a comunicação, e possui um dispositivo receptor USB que deve ser conectado ao computador, e segue o conceito de *plug-and-play*, que, segundo o autor, elimina a necessidade de instalação de *drivers* adicionais para a utilização do equipamento. O autor também indica que a distância de uso é de até 50 pés, ou seja, em torno de 15,24 m.

Figura 5 – Apresentador multimídia da empresa *Targus*.



Fonte: *Wireless USB Presenter with Laser Pointer (Black)*, 2019.

Os autores Okuno e Guedes (2017), propõem um apresentador multimídia diferente do disponibilizado no mercado, denominado *SliClick*, consistindo em uma luva, no lugar de um controle, mostrado na Figura 6. Segundo os autores, o projeto foi desenvolvido inspirado em uma situação em que um apresentador multimídia semelhante aos comerciais, no formato de um controle, acarretou confusão durante a apresentação, pois botões foram apertados acidentalmente. A luva desenvolvida contém três funções, avançar e voltar *slides* e ativar um *laser*, para apontar, assim como os apresentadores disponíveis no mercado.

Além disso, uma característica diferencial deste equipamento vestível desenvolvido, é a presença de sensores para detectar os níveis de ansiedade do usuário, e um motor *vibracall* para avisar ao usuário que o tempo de apresentação está próximo do fim. O equipamento citado no parágrafo anterior utiliza para comunicação tecnologia de radiofrequência, assim como os apresentadores comerciais, porém na faixa de 433 MHz (Mega Hertz). Também há o uso de um dispositivo receptor USB, a ser conectado no computador e são utilizadas placas de prototipagem Arduino Nano para o controle dos dispositivos, devido ao tamanho reduzido. Uma diferença é a existência de um aplicativo de servidor desenvolvido em *Java*, que deve ser instalado, e no qual é definido o tempo de apresentação, para a funcionalidade exercida pelo motor *vibracall* de interação com o usuário.

Figura 6 – Apresentador multimídia *SliClick*.



Fonte: Okuno e Guedes (2017).

3 ANÁLISE QUALITATIVA DO MÉTODO E *SOFTWARES* UTILIZADOS

Nesta seção, é feita a análise qualitativa dos *softwares* utilizados, destacando algumas funcionalidades básicas de ambos, necessárias ao desenvolvimento de equipamentos eletrônicos utilizando a colaboração MCAD e ECAD.

3.1 *Software* ECAD Utilizado

Nesta subseção, é feita a análise do *Altium Designer*, já descrito anteriormente na subseção 2.2.3, destacando algumas funcionalidades básicas, necessárias ao desenvolvimento de equipamentos eletrônicos utilizando a colaboração MCAD e ECAD. Maiores detalhamentos acerca da utilização do *software* ECAD *Altium Designer* fogem ao escopo deste trabalho e estão presentes no tutorial em Costa (2019).

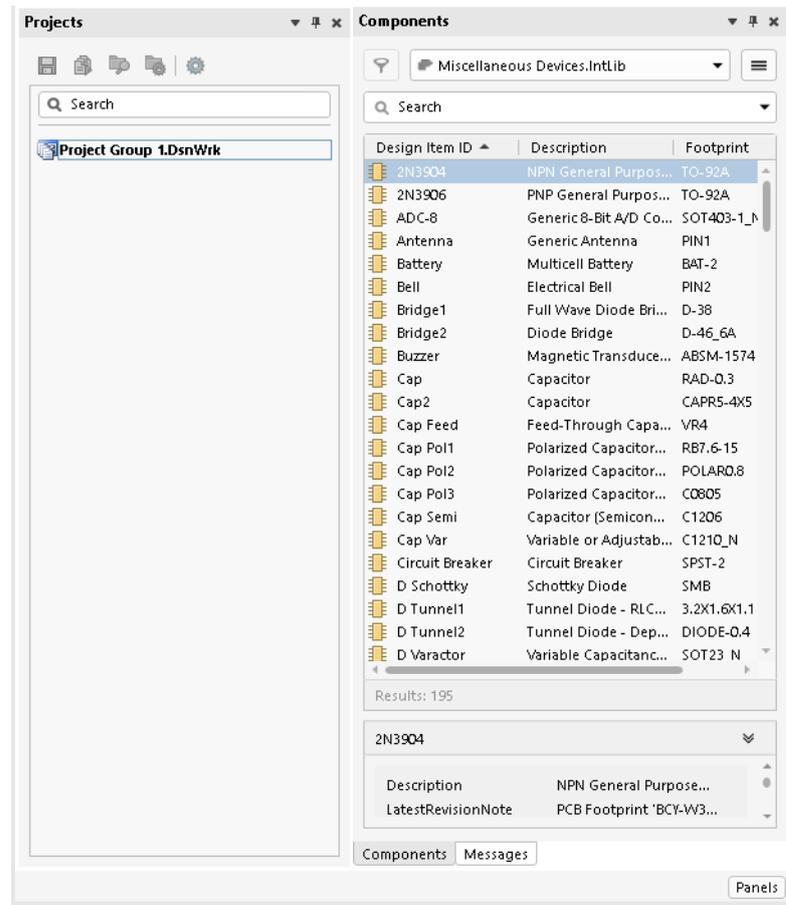
3.1.1 *Características Gerais do Altium Designer*

O *Altium Designer* é um *software* ECAD que permite a criação de projetos de PCIs, através da criação de esquemáticos nos quais são feitas as conexões entre os componentes do circuito e, posteriormente, da criação de um ou mais arquivos de PCB para desenvolvimento do *layout*, através dos circuitos dos arquivos de esquemáticos dos projetos. O *software* também possui a funcionalidade de criação de bibliotecas, nas quais pode ser feitos arquivos dos componentes do circuito, com arquivos de esquemático, no qual são criados os desenhos representativos dos componentes, e arquivos de PCB, no qual são criados os *footprints* dos componentes e os modelos 3D.

Outro diferencial deste *software*, são suas funções de identificação de características específicas de circuitos, durante a criação de esquemáticos, como por exemplo a possibilidade de criação de pares diferenciais. Esta funcionalidade permite que se identifique um par de conexões do circuito que sejam responsáveis por uma comunicação como um par diferencial. Ao se realizar essa identificação, durante a criação do *layout* da PCI, o próprio *software* adiciona uma regra que os comprimentos das trilhas dos pares diferenciais devem ser iguais, para evitar interferências na comunicação.

A Figura 7 mostra em detalhe as duas barras laterais do ambiente de inicialização *Altium Designer* 19. À esquerda, encontra-se o painel de projetos, onde pode-se ver os projetos já abertos e navegar entre eles. À direita, há o painel de componentes, através do qual pode-se ter acesso às bibliotecas, e adicionar componentes de uma determinada biblioteca aos esquemáticos dos projetos.

Figura 7 – Ambiente do *Altium Designer* 19.



Fonte: *Software Altium Designer*.

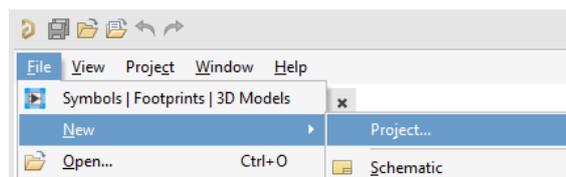
3.1.2 Noções Básicas de Criação de Projetos Eletrônicos

Nesta subseção são demonstradas e explicadas algumas noções e funcionalidade essenciais do *Altium Designer* para criação de projetos de equipamentos eletrônicos.

3.1.2.1 Criação de Projetos

A Figura 8 mostra a funcionalidade criação de projetos no *Altium Designer*, através da qual pode-se selecionar um modelo de projeto ou partir de um modelo vazio, selecionar o local onde o projeto será salvo e o nome deste. A criação de projetos é uma funcionalidade essencial, pois permite a utilização de outras funcionalidades essenciais para o desenvolvimento de equipamentos eletrônicos, como a criação de esquemáticos e PCIs relacionados.

Figura 8 – Criação de projetos no *Altium Designer* 19:

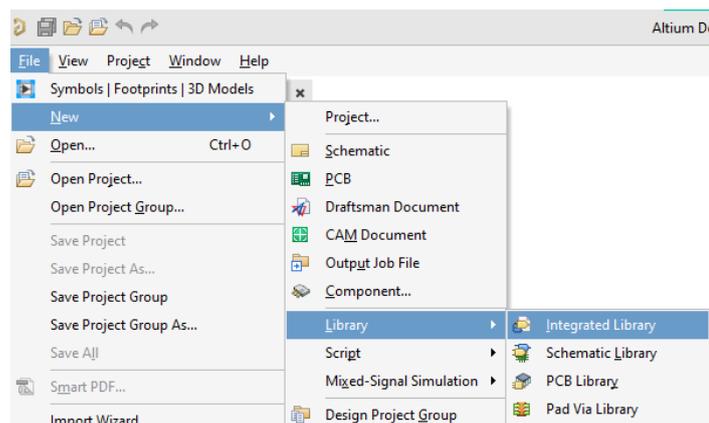


Fonte: *Software Altium Designer*.

3.1.2.2 Bibliotecas

A criação de bibliotecas é uma funcionalidade essencial, pois permite ao projetista ter controle sobre os desenhos de esquemático e *footprints* de PCIs, possibilitando a alteração e atualização, caso necessário, dos componentes do projeto. Além disso, é possível a criação de bibliotecas integradas de esquemático e PCI, que podem ser compartilhadas entre outros projetos, facilitando o desenvolvimento. A Figura 9 mostra a criação de um arquivo de biblioteca integrada.

Figura 9 – Criação de biblioteca integrada.



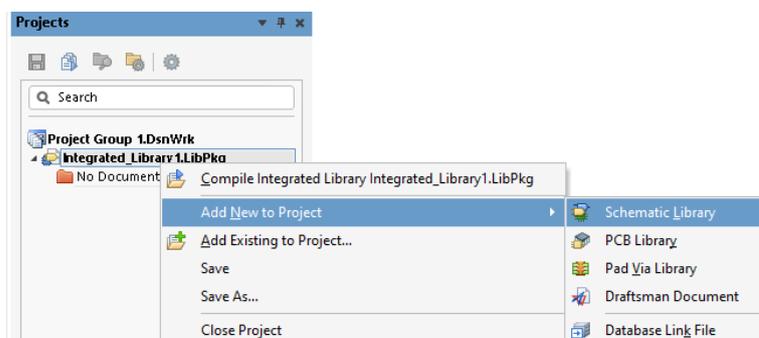
Fonte: *Software Altium Designer*.

3.1.2.2.1 Bibliotecas de Esquemático

As bibliotecas de esquemático são compilados de componentes a serem adicionados em esquemáticos, onde pode ser definido o desenho do componente e pode ser associado o componente correspondente em uma biblioteca de PCB, a fim de, quando o componente for adicionado no esquemático de um projeto, quando for criado o arquivo de PCB do projeto e importadas as conexões dos arquivos de esquemático, serão associado automaticamente os componentes pertencentes a biblioteca de PCB.

Além disso, as bibliotecas de esquemáticos permitem adicionar parâmetros aos componentes, como informações de compra e preço, para geração de um arquivo do tipo lista de materiais (do inglês, BOM, ou *Bill of Material*) A Figura 10 ilustra a criação de uma biblioteca de esquemático.

Figura 10 – Criação de biblioteca de esquemático.

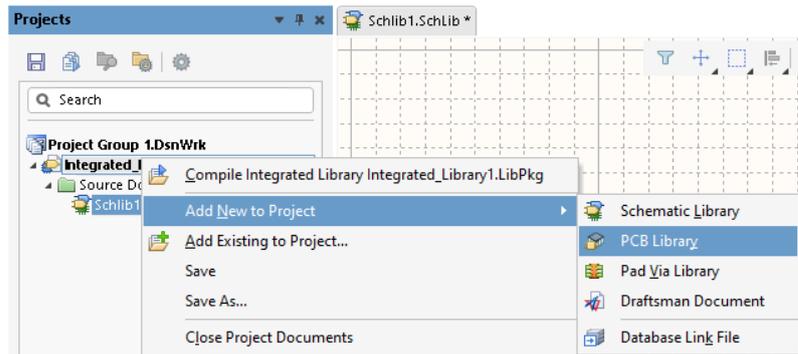


Fonte: Software Altium Designer.

3.1.2.2.2 Bibliotecas de PCI

As bibliotecas de PCI são compilados de componentes, assim como as de esquemático, porém, a serem adicionados em arquivos de PCI, onde é definido o *footprint* do componente, que corresponde aos furos para pinos e fixação dos componentes, ilhas de solda ao redor dos furos de pinos e a *silk*, que é o “contorno” do componente, que será colocado numa PCI industrial sobre a máscara, junto com os *designators*, a fim de identificar o componente. Nesses arquivos é que também é possível associar modelos 3D aos componentes, porém, de forma relativamente limitada. Também é possível o projeto muito simplificado de modelos 3D. A Figura 11 ilustra a criação de uma biblioteca de PCI.

Figura 11 – Criação da biblioteca de PCI



Fonte: *Software Altium Designer.*

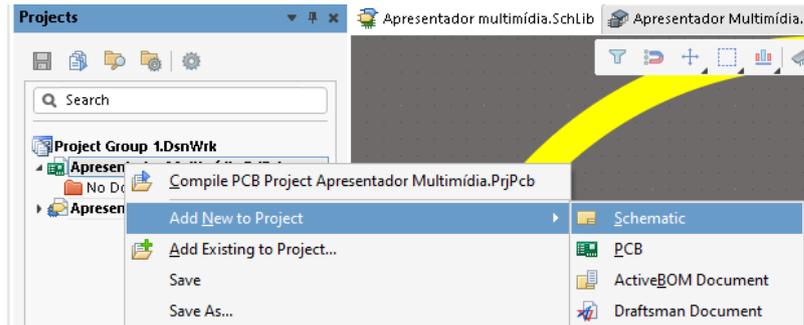
Uma diferença que aparece com relação às bibliotecas de esquemático, é que nas de PCI há uma possibilidade de seleção de abas, que correspondem às camadas (ou *layers*, do inglês) do *footprint* do componente, que influenciarão na montagem do arquivo de PCI. Além disso, também há uma barra de ferramentas na parte superior da área de desenho. Outra diferença é que, nessa biblioteca há uma janela na área superior esquerda da área de desenho que mostra a posição atual do cursor e a camada atual, facilitando a obtenção dessas informações por parte do usuário.

Além disso, diferentemente das bibliotecas de esquemático, que são mais visuais, a criação da biblioteca de PCI não pode ser arbitrária, visto que haverá parâmetros desta que representam as dimensões físicas dos componentes. Ou seja, o tamanho dos furos, distância entre eles e contorno devem ser decididos com base em medidas do componente feitas pelo usuário com o uso de um paquímetro, ou com informações retiradas de *datasheets*. A opção de retirar informações de *datasheets* é mais provisória e as dimensões devem sempre ser conferidas por meio de medição com um paquímetro.

3.1.2.3 Esquemáticos

Os esquemáticos são os arquivos de um projeto onde são colocados os componentes de bibliotecas de esquemático, como as feitas na seção 3.1.2.1, e são feitas as ligações entre esses componentes, montando os circuitos. Além disso, nos arquivos de esquemático é que são definidas algumas regras importantes ao circuito, como por exemplo a criação de pares diferenciais, já comentada. A Figura 12 mostra a criação de um arquivo de esquemático

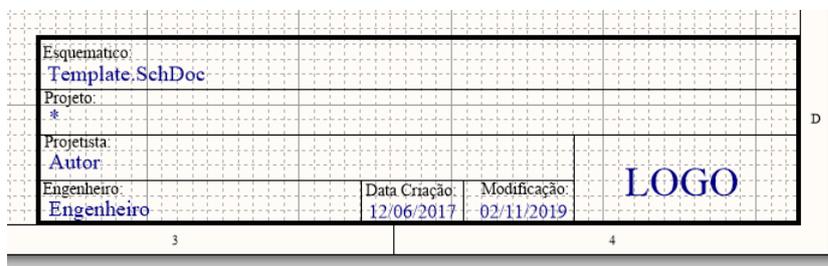
Figura 12 – Criação de arquivo de esquemático.



Fonte: *Software Altium Designer.*

3.1.2.4 Templates

Outra funcionalidade importante para os esquemáticos de um projeto é a utilização de *templates*, que permite a identificação do projeto, com logo da empresa, nome do projetista, data de aprovação, entre outros. O *template* é criado como um arquivo de esquemático vazio, no qual serão adicionados as informações do *template*, utilizando formas geométricas e ferramentas de inserção de texto. Também é importante a inserção de ferramentas de inserção de texto dependente de parâmetros, que são as que serão modificadas em cada arquivo de esquemático ao qual o *template* for adicionado, de acordo com os parâmetros. Na Figura 13 é mostrado o exemplo de um *template* em destaque.

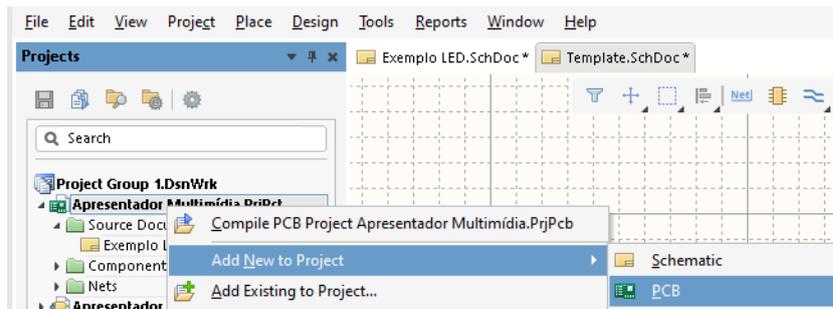
Figura 13 – Exemplo de *template*.

Fonte: *Software Altium Designer.*

3.1.2.5 Placas de Circuito Impresso

Os arquivos de PCI são os arquivos em que são organizados os componentes em um determinado *layout* e feita a conexão desses componentes através de trilhas, sendo possível escolher a posição do componente e das trilhas na PCI, bem como em que face dela estes se localizam. A Figura 14 mostra a criação de um arquivo de PCI.

Figura 14 – Criação de um arquivo de PCI.

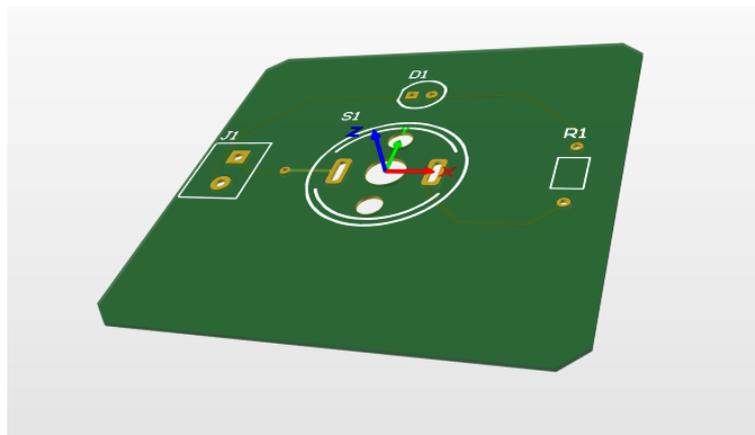


Fonte: *Software Altium Designer.*

3.1.2.6 Visualização 3D

O *software* também possui, no ambiente de criação de PCIs, um ambiente de visualização 3D. Embora limitado, conforme mostrado na Figura 15, o ambiente permite a visualização da PCI montada. Não há nenhum componente, pois os modelos 3D de componentes não foram adicionados na biblioteca. Porém, em outra seção, esse ambiente será novamente abordado, relacionado com a interação MCAD e ECAD.

Figura 15 – Ambiente de visualização 3D do *Altium Designer.*



Fonte: *Software Altium Designer.*

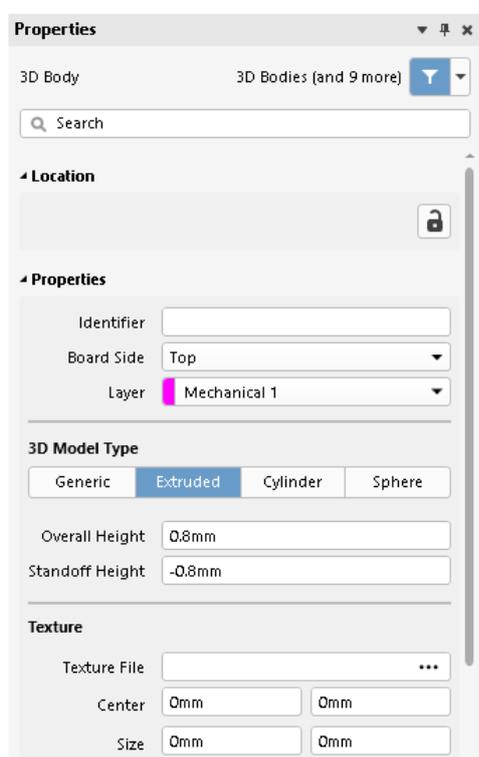
3.1.3 Limitações para o Desenvolvimento de Equipamentos Eletrônicos

Como citado na seção 3.1.2.6, o *software* possui um ambiente de visualização 3D. Porém, observou-se que ele possui algumas limitações, que são discutidas nesta seção.

3.1.3.1 Criação de Modelos 3D e Interação no Ambiente de Visualização 3D

O *Altium Designer*, durante a criação de bibliotecas de PCI, permite a criação de modelos 3D através de formas geométricas básicas, conforme mostra a Figura 16. Porém, esse ambiente não possui ferramentas de modelagem 3D mais complexas, como pode ser visto em outras seções que o *Autodesk Inventor* possui. E também, outra funcionalidade limitada do *Altium designer* é o posicionamento e movimentação dos modelos 3D, que, mesmo podendo ser feito por coordenadas, não é muito intuitivo e não possui ferramentas de criação de relações, que permitem posicionar os componentes com funções em relação a planos, eixos, ou outros componentes. Essas características, não muito comuns em *softwares* ECAD, como o *Altium Designer*, podem ser supridas pelos *softwares* MCAD, como o *Autodesk Inventor*, visto que são funcionalidades básicas destes.

Figura 16 – Ambiente de criação de modelos 3D do *Altium*.



Fonte: *Software Altium Designer*.

3.1.3.2 Ausência de Simulação Térmica

Outra funcionalidade possuída por *softwares* MCAD e que os *softwares* ECAD, como o *Altium Designer* não costumam possuir, é a análise por elementos finitos, visto que é um cálculo mais voltado para a engenharia mecânica. Porém, um tipo de análise por elementos finitos que possui grande importância para a engenharia eletrônica e projetos de equipamentos eletrônicos, é a simulação térmica, visto que a temperatura e dissipação de calor influencia o funcionamento dos componentes eletrônicos. Logo, essa é uma limitação do *Altium Designer* que será suprida utilizando o *Inventor Nastran*.

3.2 Software MCAD Utilizado

Nesta subseção, é feita a análise do *Autodesk Inventor* e do *Inventor Nastran*, que funciona como uma extensão ao *Inventor*, já descritos anteriormente na subseção 2.1.3, destacando algumas funcionalidades básicas, necessárias ao desenvolvimento de equipamentos eletrônicos utilizando a colaboração MCAD e ECAD. Esta seção é focada nas funcionalidades destes *softwares* que são importantes para suprir as limitações dos *softwares* ECAD, levantadas na seção 3.1.3. Maiores detalhamentos acerca da utilização do *software* MCAD *Autodesk Inventor* e suas funcionalidades fogem ao escopo deste trabalho e estão presentes no tutorial em Costa (2019).

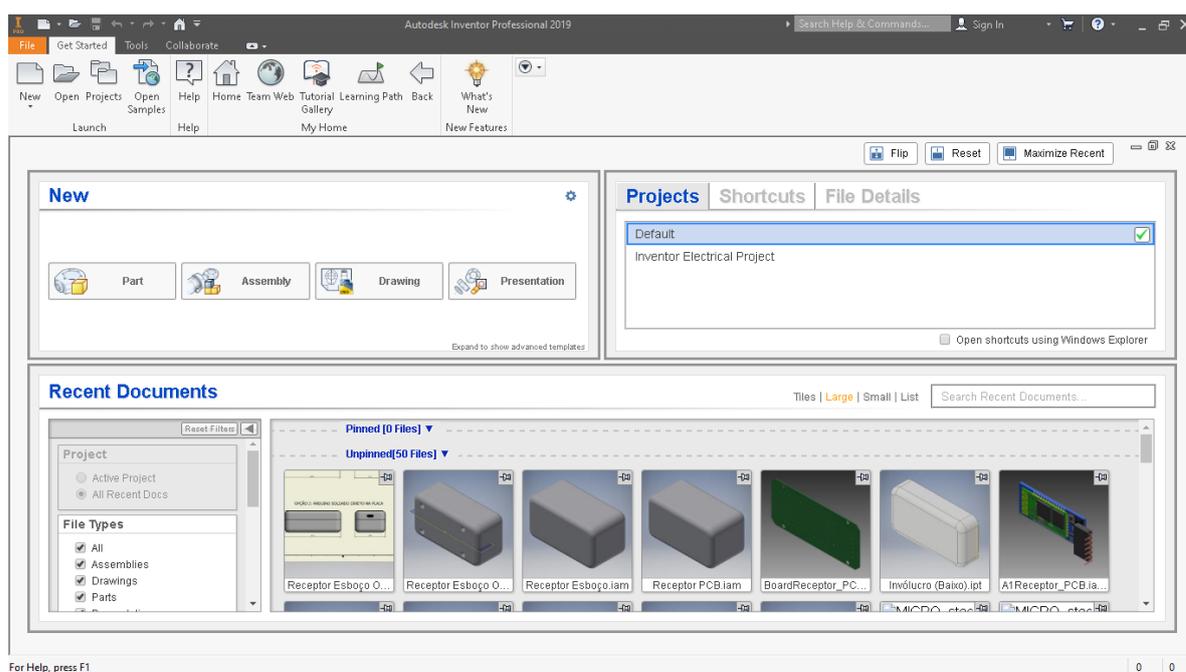
3.2.1 Características Gerais do Autodesk Inventor

O *Autodesk Inventor* é um *software* MCAD que permite a criação de projetos mecânicos em 3D, através da criação de peças (*parts*) 3D, criadas através de esboços 2D com comandos do *software*, e de montagens (*assemblies*), que são arquivos com mais de uma peça conectadas entre si em pontos específicos. O *software* também possui a funcionalidade de cálculos de tensões, esforços e outras análises mecânicas, através de cálculos por análise por elementos finitos. E a utilização do *Inventor Nastran* como extensão fornece ainda mais possibilidades de cálculos por elementos finitos, como as análises térmicas, que são interessantes para os projetos de equipamentos eletrônicos.

A Figura 17 mostra o ambiente de inicialização do *Autodesk Inventor*. À esquerda, no centro, existe o *menu* atalho de criação de novos arquivos, que, da direita pra esquerda são:

peças (*parts*), montagens (*assemblies*), desenho 2D (*drawing*) e apresentação (*presentation*). À sua direita, existem os *menus* de projetos, atalhos e detalhes de arquivos. Abaixo, há o *menu* de documentos recentes, facilitando o acesso a estes. Acima, há uma barra de ferramentas onde da direita para a esquerda, respectivamente, tem-se os botões de: criação de novos arquivos; abrir arquivos; projetos; abrir exemplos; ajuda; voltar ao *menu* inicial (o da Figura 17); página da internet; galeria de tutoriais; tutoriais sequenciais; voltar; e verificação de novas funcionalidades.

Figura 17 – Ambiente de inicialização do *Autodesk Inventor*.



Fonte: *Software Autodesk Inventor*.

3.2.2 Noções Básicas de Criação de Projetos Mecânicos

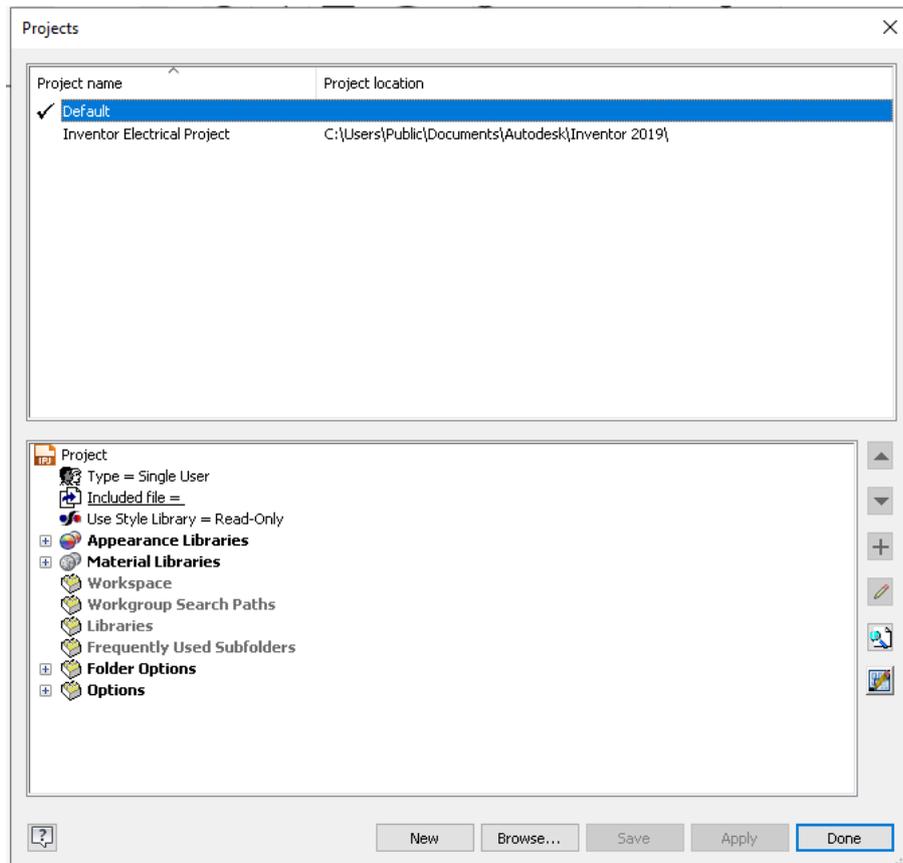
Nesta subseção são demonstradas e explicadas algumas noções e funcionalidades essenciais do *Autodesk Inventor* para criação de projetos mecânico.

3.1.2.1 Criação de Projetos

A Figura 18 mostra o ambiente de criação de projetos, que pode ser aberto através do botão projetos (*projects*), na barra de ferramentas superior dessa Figura. É possível a criação de um novo projeto clicando no botão novo (*new*) no canto inferior esquerdo da janela. Abrirá uma

janela perguntando o tipo de projeto, pode-se criar um projeto de usuário único ou compartilhado. Em seguida, ao clicar em próximo (*next*), abre uma janela na qual é possível escolher o nome e a pasta a ser salvo o projeto. Em seguida, pode-se clicar em terminar (*finish*) e o projeto já aparecerá no ambiente de gerenciamento de projetos. Então, ao fechar a aba de projetos, na tela inicial pode-se ver o novo projeto criado no *menu* de projetos.

Figura 18 – Ambiente de gerenciamento de projetos.

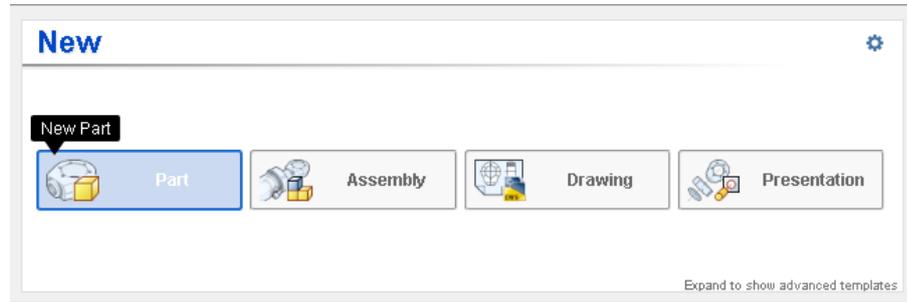


Fonte: *Software Autodesk Inventor*.

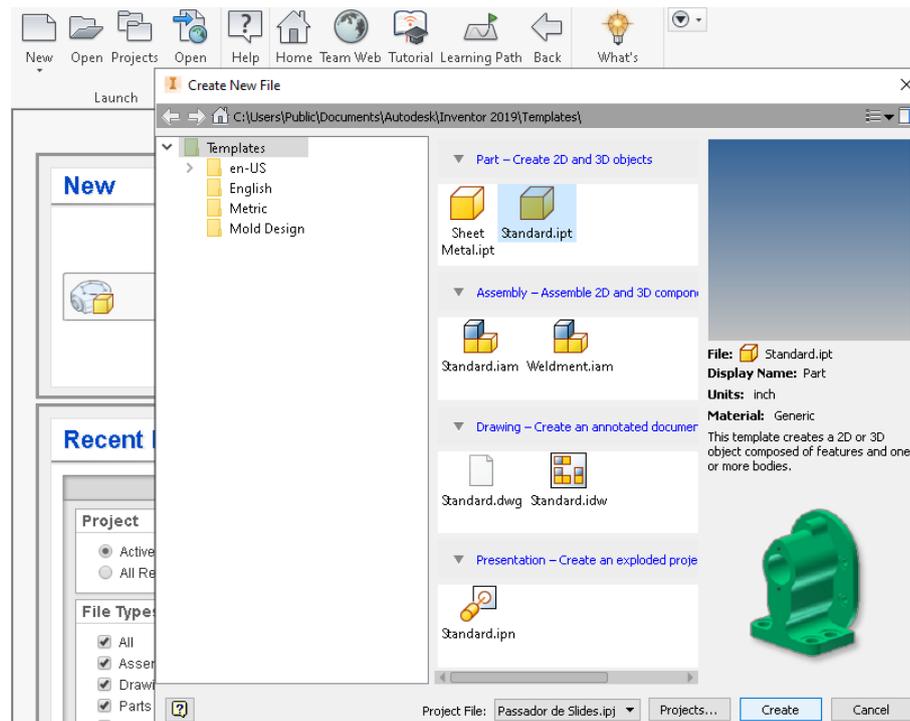
3.2.2.2 Criação de Peças

Nesta seção é abordada a criação de peças, que pode ser feita através do *menu* de criação de arquivos novos, clicando em peças (*parts*), conforme mostrado na Figura 19 (a), ou apertando o botão novo (*new*) na barra de ferramentas superior e escolhendo o tipo de arquivo novo, conforme a Figura 19 (b)(a). Na Figura 19 (b) é possível ver que existem dois tipos de peças, padrão e folha de metal (*sheet metal*). A diferença é que as peças do tipo padrão são feitas por injeção de material, fundição ou impressão 3D, já as peças do tipo folha de metal são feitas com folhas de metal dobradas.

Figura 19 – Criação de peça: (a) *menu* de criação de arquivos novos; (b) barra de ferramentas superior.



(a)



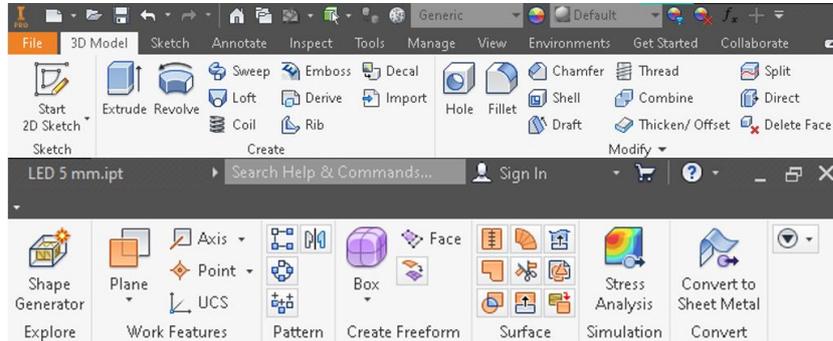
(b)

Fonte: *Software Autodesk Inventor*.

Ao criar a peça, será mostrado o ambiente de criação de peça do tipo padrão. A barra lateral esquerda mostra a árvore de *design* (*design tree*), onde aparecem os comandos que foram utilizados para modelar uma peça. Acima, há várias abas, como pode-se ver em destaque na Figura 20. Como há muitas abas, e foge ao escopo desse trabalho, pois não será utilizado, serão explicadas apenas as duas primeiras, da direita para a esquerda, respectivamente, a de modelo 3D (*3D Model*), que possui diversas funcionalidades para construções de estruturas 3D, e o de

esboço (*sketch*), que é utilizado para desenho dos esboços 2D que servem de base para os comandos de modelo 3D.

Figura 20 – Barra superior do ambiente de criação de peça do tipo padrão.

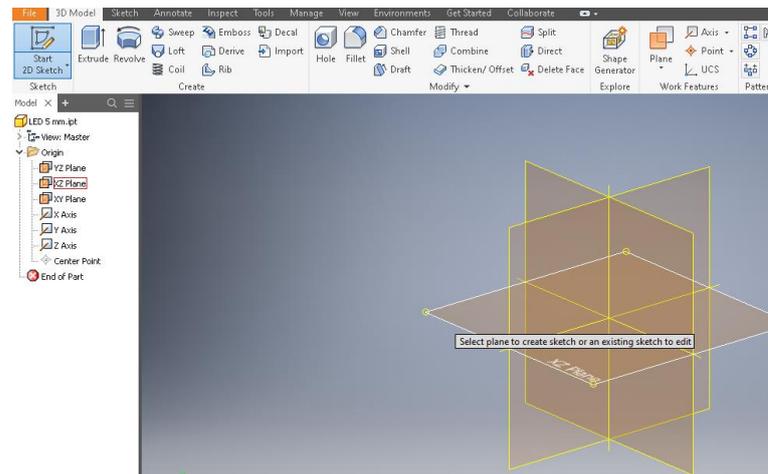


Fonte: *Software Autodesk Inventor.*

3.2.2.2.1 Esboços 2D

Os esboços 2D são partes fundamentais da criação de qualquer peça, visto que os modelos 3D são construídos em função de esboços 2D, interagindo com eles de diversas formas, discutidas na próxima subseção. Esta seção é dedicada à explicação das funcionalidades do ambiente de criação de modelos 2D. Primeiramente, para o uso dos esboços 2D, deve ser criado um esboço 2D em algum plano. Ao criar o primeiro esboço 2D da peça, normalmente não há nenhum plano visível, então ao clicar na funcionalidade criar esboço 2D (*start 2D sketch*), são mostrados os 3 planos básicos da peça, XY, YZ e XZ, conforme a Figura 21.

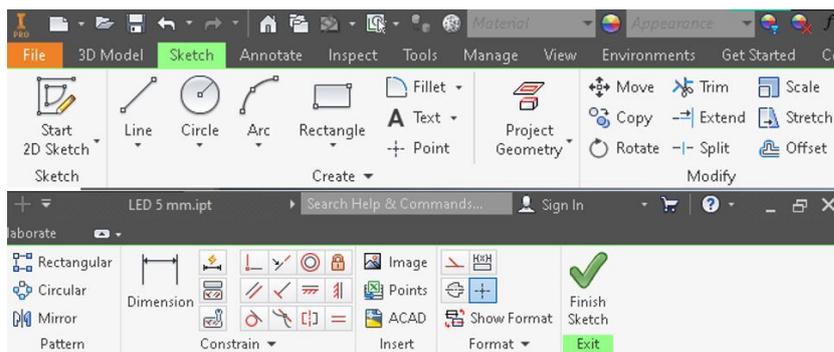
Figura 21 – Planos XY, YZ e XZ para criação de um esboço 2D.



Fonte: *Software Autodesk Inventor.*

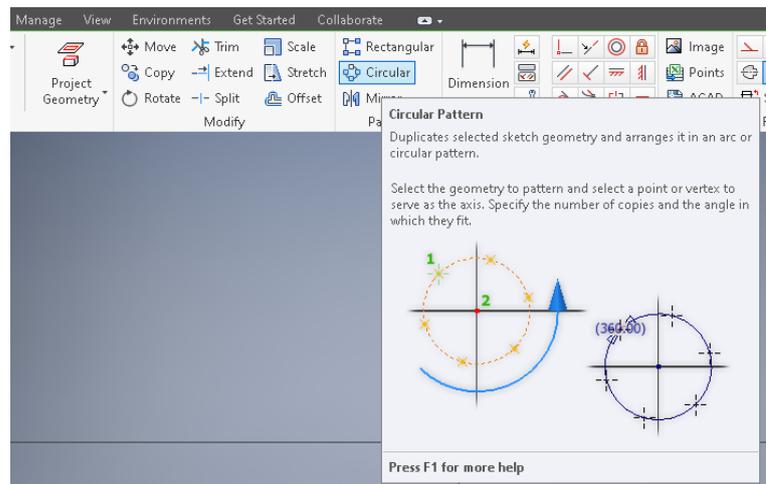
Então, percebe-se que a aba superior de esboço (*sketch*) é selecionada automaticamente e está destacada, conforme a Figura 22. Essa aba possui várias funções de manipulação do esboço 2D, divididas em 6 espaços: criar (*create*); modificar (*modify*); padrões (*pattern*); confinamento (*constraints*); inserir (*insert*); e formato (*format*). Uma funcionalidade útil do *software* é que se você posicionar o mouse sobre um comando e esperar, ele descreve o comando por escrito e mostra uma figura de exemplo, conforme a Figura 23 o que melhora o aprendizado e utilização deste.

Figura 22 – Aba esboço (*sketch*) da barra superior.



Fonte: *Software Autodesk Inventor*.

Figura 23 – Descrição de um comando com o mouse sobre ele.

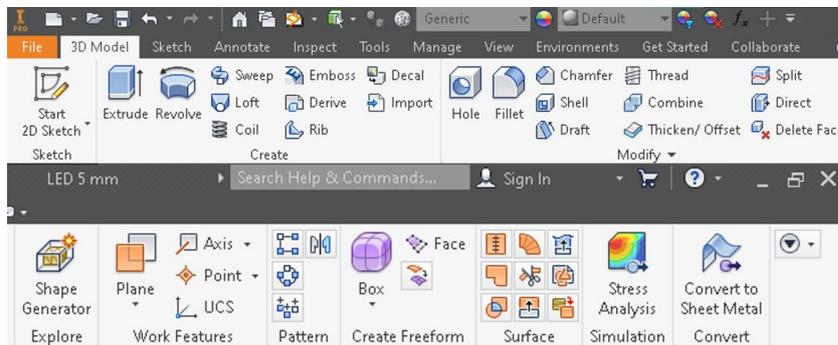


Fonte: *Software Autodesk Inventor*.

3.2.2.2.2 Comandos Básicos de Modelo 3D

Nesta subseção, são discutidos os comandos de criação de modelo 3D, que, assim como os comandos de esboço 2D, são divididos em espaços, no caso, 10 espaços, esboço, criar, modificar, explorar, características de trabalho, padrões, criar forma livre, superfície, simulação e converter. Os utilizados serão o de esboço, criar, modificar e características de trabalho. Embora o de padrões também seja muito útil, não é necessário para este trabalho. Assim como nos esboços 2D, é possível saber o que um comando faz passando o cursor do *mouse* sobre ele, esperando e aparecerá uma descrição do comando e uma imagem de exemplo. A Figura 24 mostra a aba de modelo 3D, com seus diversos comandos.

Figura 24 – Aba modelo 3D (*3D model*) da barra superior.



Fonte: *Software Autodesk Inventor*.

O espaço de esboço serve para a criação de esboço em um determinado plano desejado, abrindo a aba de esboço, já discutida na seção anterior. O espaço de criar possui diversos comandos para criação de modelos 3D baseados em esboços 2D, como: extrusão (*extrusion*), que é basicamente a adição de espessura a uma curva fechada de um esboço 2D, na direção perpendicular ao plano em que ele foi criado; revolução (*revolution*), em que se seleciona um perfil, que corresponde a uma curva fechada, de um esboço 2D, e uma curva, em outro esboço 2D ou 3D, em torno do qual será feita a revolução daquele perfil, formando um sólido tridimensional, também chamado sólido de revolução.

Outros comandos são: envelopamento (*loft*), que permite a seleção de diversos perfis em esboços diferentes, criando um modelo 3D que interseccione os esboços ao passar pelo plano ao qual o esboço pertence, “envelopando” os esboços e pode ser utilizado para criar curvas ou superfícies mais irregulares, sendo mais preciso quantos mais esboços forem

utilizados em sua criação; espiral (*coil*), que usa um perfil e uma direção de um eixo para criação de um modelo 3D correspondente a uma espiral helicoidal do perfil na direção definida, podendo ser definidos diversos parâmetros e pode ser utilizado para criação de molas, bobinas e enrolamentos diversos.

Além desses: gravar em relevo (*emboss*), que é utilizado para criar gravações com relevo em uma determinada face, seja positivo (acima do plano da face) ou negativo (abaixo do plano da face), de um determinado perfil e pode ser utilizado para gravação de nomes, logos, figuras, dentre outros em faces de objetos; derivar (*derive*), que cria na peça um objeto derivado de outra peça, podendo ser alteradas escala e outras características; nervura (*rib*), que cria nervuras de sustentação em uma peça, projetando perfis em uma determinada face da peça e é usado para melhorar a resistência de peças; decalque (*decal*) é utilizado para aplicar uma imagem escolhida em uma face determinada de um objeto e possui função estética; e importar (*import*), importa um arquivo de outros sistemas de CAD num arquivo de peça do *Inventor*.

Do espaço de modificação, os comandos mais relevantes para este trabalho são: furo (*hole*), permite a criação de furos em peças a partir de pontos de trabalho ou circunferências presentes em um esboço 2D, podendo ser definidos diversos parâmetros, como tamanho do furo, tipo de furo e profundidade; arredondamento (*fillet*), semelhante ao comando análogo de esboços 2D, cria um arredondamento de raio definido entre duas faces; casca (*shell*), transforma a peça numa casca, criando uma região oca dentro, podendo ser definida a espessura e a remoção de uma ou mais faces; e *thread*, que cria um perfil helicoidal apenas visualmente em uma determinada superfície cilíndrica, dando o aspecto necessário a furos, parafusos, dentre outros, sem consumir muita memória.

Por fim, o espaço de características de trabalho possui os seguintes comandos: plano (*plane*), permite criar planos de trabalho com diversas condições, distância a outro plano, interseção com pontos, eixos, dentre outros; eixo (*axis*), permite criar eixos de trabalho em diversas condições, de forma semelhante ao comando anterior; ponto de trabalho (*work point*), permite a criação de pontos de trabalho em diversas condições; e sistemas de coordenadas do usuário (UCS ou *User Coordinate System*), que permite a criação pelo usuário de sistemas de coordenadas personalizados internos a peça.

3.2.2.2.3 Material e Aparência da Peça

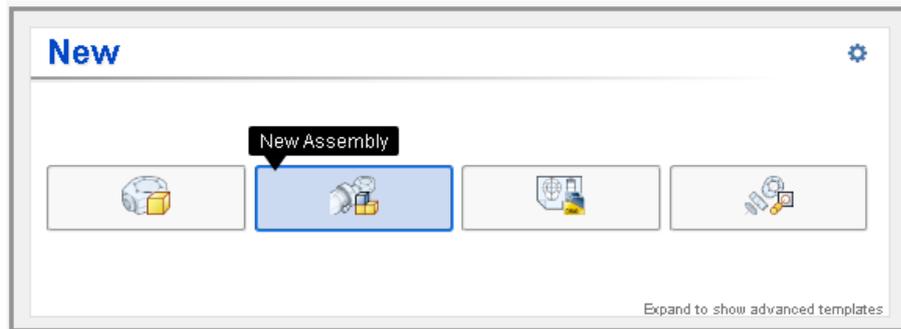
Essas duas funcionalidades permitem mudar a aparência da peça. A ferramenta de aparência (*appearance*) permite a seleção de uma ou mais faces ou características de uma peça e adição de determinada aparência, que pode ser escolhida através de uma extensa lista disponibilizada pelo *software*, ou de outras listas, via *download*, ou até de perfis de aparência criados pelo usuário. O comando material (*material*) possui duas funções, visual e estrutural, pois pode dar aspecto mais metálico, cerâmico, ou até translúcido, como vidro ou plástico, a uma peça, mas também interfere em cálculos de análise por elementos finitos, pois é com o material que é definida a densidade da peça e outras propriedades físicas.

Uma diferença fundamental entre esses dois comandos, além de um ser apenas visual (aparência) e outro visual e estrutural (material), é que uma mesma peça pode ter mais de uma aparência, determinada pelas faces, mas pode ter apenas um único material, comum a toda a peça. A funcionalidade de material é de suma importância na criação de montagens, que são explicadas na próxima seção, pois embora uma peça não possa ser formada por mais de um material, uma montagem pode ter peças de diferentes materiais, o que, novamente, pode ter função visual e estrutural.

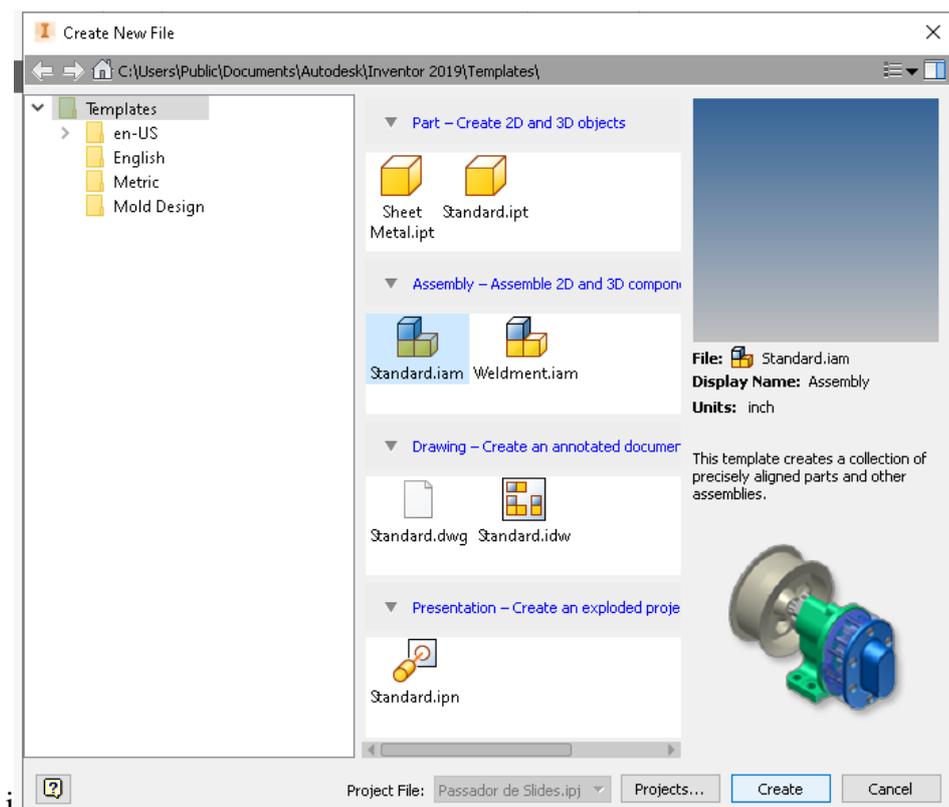
3.2.2.3 Criação de Montagens

Nesta seção é abordada a criação de montagens, que pode ser feita através do *menu* de criação de arquivos novos, clicando em montagens (*assemblies*), conforme mostrado na Figura 25 (a), ou apertando o botão novo (*new*) na barra de ferramentas superior e escolhendo o tipo de arquivo novo, conforme a Figura 25 (b). Na Figura 25 (b) é possível ver que existem dois tipos de montagens, padrão e por soldagem (*weldment*). A diferença é que as montagens do tipo padrão são feitas por união de peças por interferência, parafusos, pinos, ou métodos semelhantes, já as montagens por soldagem são feitas com união das peças através de solda.

Figura 25 – Criação de montagens: (a) *menu* de criação de arquivos novos; (b) barra de ferramentas superior.



(a)

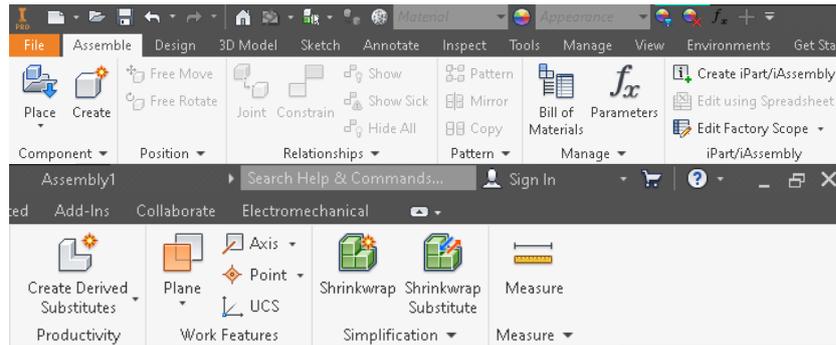


(b)

Fonte: *Software Autodesk Inventor*.

Ao ser criado o arquivo de montagem, pode-se ver a barra de ferramentas superior deste arquivo, conforme a Figura 26, que possui novas abas em relação ao arquivo de peça, sendo essas, a aba de montar (*assemble*) e de *design*. A aba mais utilizada deste modo neste trabalho será a de montar, que permite o uso dos diversos comandos referentes a junção de peças, criando uma peça, ou equipamento, mais complexo.

Figura 26 – Barra de ferramentas superior de montagem.



Fonte: *Software Autodesk Inventor*.

3.2.2.3.1 Comandos Básicos de Montagem

Conforme mencionado na seção anterior, a aba da Figura 26 que é explicada é a aba de montar (*assemble*), dividida em 10 espaços, componente (*component*), posição (*position*), relações (*relationships*), padrões (*pattern*), administração (*manage*), *iPart/iAssembly*, produtividade (*productivity*), características de trabalho (*work features*), simplificação (*simplification*) e medidas (*measure*). Dessas, percebe-se que algumas já foram discutidas anteriormente, no caso, a de padrões e características de trabalho. Além delas, as principais e que são discutidas nesse trabalho são a de componente, relações e administração.

O primeiro espaço abordado é o de componente, pois também é o primeiro espaço que é utilizado durante a criação de uma montagem. Seus comandos são inserir (*place*) e criar (*create*). Como os nomes sugerem, o comando de inserir permite inserir peças ou outras montagens, através de arquivos, nessa montagem, e o comando de criar permite a criação de peças ou montagens dentro dessa montagem, utilizando algum ponto já existente como referência. Com a peça inserida, é possível movê-la livremente, já que não há nada restringindo seu movimento. Porém, numa montagem real, as peças não possuem liberdade total de movimento, as vezes possuem liberdade apenas em alguns eixos ou não possuem nenhuma liberdade.

É para isso que serve o espaço de relações, que possui 2 comandos principais, junta (*joint*) e confinamento (*constraint*). O comando de junta permite criar relações móveis, com limites de movimento definido entre características de duas peças diferentes. Já o de

confinamento, cria relações normalmente fixas, porém, às quais também pode ser dada uma certa liberdade de movimento, motivo pelo qual este comando é suficiente para satisfazer a maioria das necessidades do projetista, e é o que é discutido neste trabalho. Uma boa prática de projetos é sempre deixar uma peça fixa e centralizada em relação à montagem. Normalmente, isso é feito com a peça central de uma montagem, como por exemplo, uma base de um objeto que possui outras peças ligadas a esta base em comum. A visibilidade de elementos pode ser alterada clicando com o botão direito e selecionando a opção visibilidade (*visibility*), ou clicando no plano e usando o comando do teclado “*Alt+V*”.

Então, abre-se o *menu* do comando de confinamento, que possui várias abas, montagem, movimento, transicional e conjunto de confinamentos. Porém, a aba utilizada e explicada neste trabalho, por ser suficiente, é a de montagem. A aba de montagem possui 5 tipos, mostrados em destaque na Figura 27. Da esquerda para a direita, respectivamente, os comandos são: conexão (*mate*), ângulo (*angle*), tangência (*tangent*), inserção (*insert*) e simetria (*symmetry*).

Figura 27 – Tipos de comandos de confinamento.



Fonte: *Software Autodesk Inventor*.

O comando de conexão permite fazer a junção direta de dois planos ou dois eixos, selecionados um após o outro com o clique do *mouse* no plano ou eixo, com opções de direção, escolhidas na opção *solution* e possibilidade de deslocamento (*offset*). Após selecionados os dois planos ou eixos, é mostrada uma prévia de visualização. O comando é finalizado após o clique no botão aplicar (*apply*).

O comando de ângulo permite a definição de um determinado ângulo entre duas características, sejam planos ou eixos. O comando de tangência permite a definição de conexão com tangência entre uma superfície arredondada e outra superfície, arredondada ou não. O comando de inserção é utilizado para inserção de peças, normalmente cilíndricas, como parafusos, pinos ou tubos, em furos. E o comando de simetria permite a definição de simetria entre duas superfícies e um plano de referência. Todos os comandos possuem opção de

deslocamento (*offset*), no mínimo duas soluções possíveis, no mínimo duas seleções e possibilidade de nomear o comando.

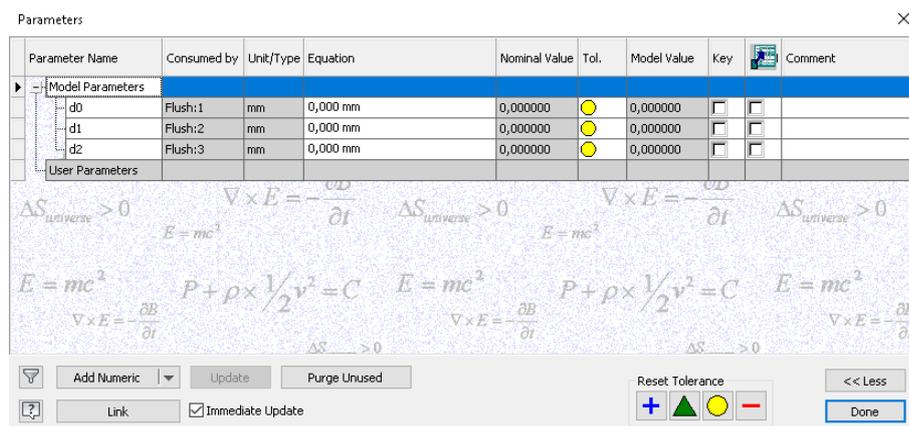
Ao fim da conexão dos três planos básicos do LED com os 3 planos básicos da montagem, ele estará no centro da montagem, e os planos estarão coincidentes, parecendo ser apenas 3, em vez de 6. Além disso, os outros três comandos da aba de relações são: mostrar (*show*), que possibilita mostrar na montagem uma relação escolhida; mostrar problemas (*show sick*), que mostra todas as relações que estejam com algum erro; e esconder todas (*hide all*), que esconde todas as relações mostradas.

O outro espaço importante é o de administração (*manage*), que possui dois comandos. O de lista de materiais (*Bill of Material*, ou BOM), que permite exportar uma lista de materiais, na forma de uma planilha compatível com o *Excel*, que pode conter material das peças, descrição preço, dentre outras informações definidas pelo usuário. O comando de parâmetros (*parameters*) é explicado na seção seguinte.

3.2.2.3.2 Parâmetros

Ao clicar no comando de parâmetros, é aberto o *menu* mostrado na Figura 28, onde é possível ver todos os parâmetros de uma montagem e gerenciá-los, alterando, deletando ou adicionando mais parâmetros. A importância dos parâmetros é permitir uma alteração mais fácil de dimensões, deslocamentos de relações ou quaisquer medidas em gerais que possam ser definidas, tornando estas medidas funções de um parâmetro, ao qual pode ser dado um nome, e acessado e gerenciado facilmente no *menu* de parâmetros. Deste modo, ao ter, por exemplo, várias peças distantes entre si de uma distância comum, pode-se criar um parâmetro e associar o nome dele ao definir a distância, no lugar de números. Também é possível a utilização de equações em função destes parâmetros, aumentando enormemente as possibilidades do projetista.

Figura 28 – Menu de parâmetros.



Fonte: *Software Autodesk Inventor.*

3.2.2.3.3 Adaptabilidade

A funcionalidade de adaptabilidade permite a criação de características em peças que sejam ditas adaptativas a características da mesma ou de outras peças. Ou seja, as características criadas dessa forma serão dinamicamente mutáveis em relação à característica ao qual elas são relacionadas. Por exemplo, o encaixe de uma peça um furo numa peça base, onde outra peça encaixa, pode ser adaptativo à posição da segunda peça, podendo-se alterar a posição da segunda peça sem se preocupar em precisar refazer o furo. Saber utilizar as ferramentas de adaptabilidade associadas à projeção de geometria é uma ferramenta muito robusta para o projetista.

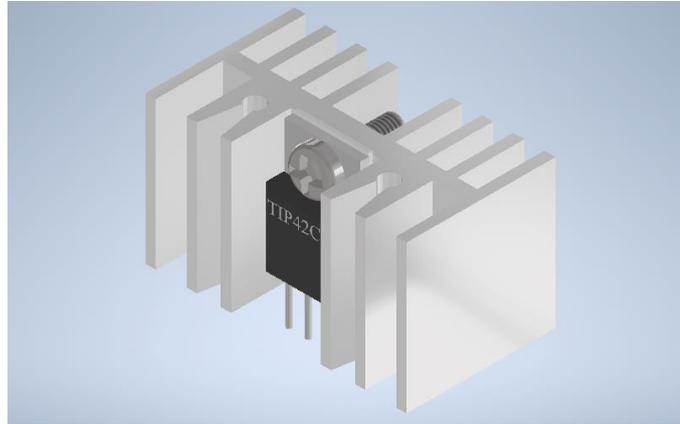
3.2.3 Simulação Térmica com o Inventor Nastran

Nesta seção é abordada a simulação térmica com a extensão *Inventor Nastran* para o *Autodesk Inventor*. O uso dessa funcionalidade permite analisar a variação da temperatura e do fluxo de calor em sistemas, utilizando cálculos por análise por elementos finitos. Isto é muito importante para o projetista eletrônico, pois, caso utilizada corretamente, possibilita ter uma noção se a dissipação de calor dos componentes de um sistema está adequada.

Esta simulação pode evitar problemas de projeto que podem levar a danificação de componentes, visto que o aquecimento exagerado de componentes eletrônicos pode causar danos irreversíveis a estes. Para a simulação térmica, será utilizado um modelo de transistor

TBJ PNP TIP42C, com um dissipador HS3520_20, que possui dimensões de 35 x 20 x 20 mm, modelo mostrado na Figura 29. O material do dissipador é alumínio.

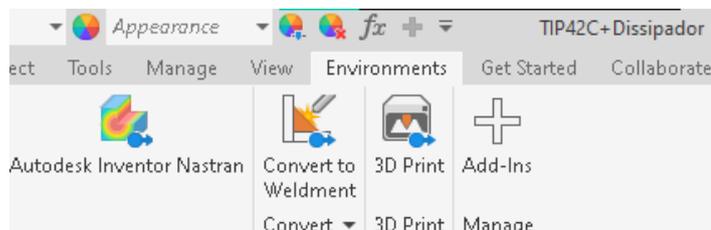
Figura 29 – TIP42C com dissipador HS3520_20.



Fonte: Próprio Autor.

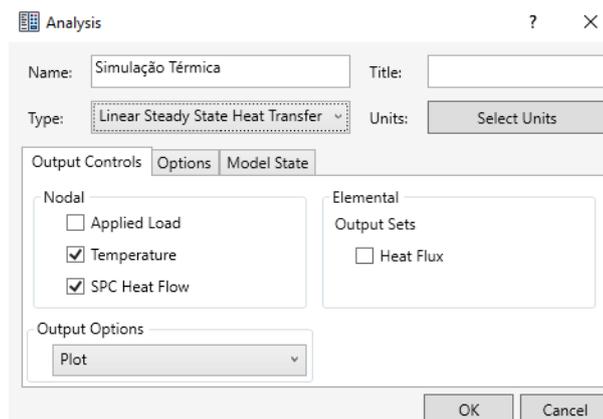
Primeiramente, deve-se abrir a extensão do *Nastran*, o que é feito através do *menu* de ambientes (*environments*), na parte superior do ambiente de trabalho, ilustrado na Figura 30. Em seguida, abrirá uma aba nova no *menu* anterior referente ao *Nastran*. Então, é possível criar uma nova análise, usando o comando novo (*new*), no *menu* superior, que abrirá uma janela, na qual seleciona-se o tipo (*type*) da simulação, escolhendo transferência de calor linear de estado permanente (*linear steady state heat transfer*), mostrado na Figura 31.

Figura 30 – *Menu* de ambientes (*environments*), mostrando a extensão *Autodesk Inventor Nastran*.



Fonte: *Software Autodesk Nastran Inventor*.

Figura 31 – Janela de criação de nova análise térmica.



Fonte: *Software Autodesk Nastran Inventor*.

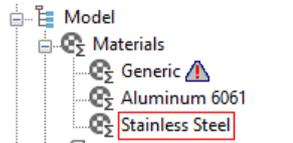
Em seguida, é necessário definir os materiais dos elementos do sistema. O transistor TIP42C foi dividido em 3 materiais, para melhorar os resultados. As considerações escolhidas para os materiais foram: *case* do TIP42C de epóxi; pinos (*leads*) e parte metálica de invar, uma liga de ferro e níquel comumente utilizada em *leads* de componentes, devido ao baixo coeficiente de dilatação térmica; *chip* interno de carbetto de silício dopado (SiC); dissipador de alumínio 6061, liga comumente usada em dissipadores; parafuso M3x10mm de aço inoxidável; e isolante de alumina (óxido de alumínio), com melhores propriedades de isolamento elétrica e dissipação térmica do que a mica.

As características térmicas dos materiais podem ser obtidas no site MatWeb (MatWeb LLC, 2019), um repositório de informações sobre diversos materiais. Então, com as características térmicas dos materiais, é possível selecionar o material na lista de materiais ou criar um novo material, com as propriedades térmicas desejadas. O material do parafuso e do dissipador já existem na lista de materiais do *software*, não sendo necessário criar um novo material, apenas selecionando-os no *menu* de materiais, conforme explicado na seção 3.2.2.2.3. Já os outros 4 materiais, epóxi, invar, SiC e alumina, não existem na lista de materiais, sendo necessário criar um novo material e adicionar as propriedades físicas dele.

No caso, como as únicas propriedades importantes para esta análise são as térmicas, é necessário apenas inserir as propriedades térmicas dos materiais. Porém, caso desejado realizar outros tipos de análises, seriam necessárias outras propriedades, mas foge ao escopo deste trabalho. Na árvore de *design*, à esquerda, agrupado no item modelo (*model*), existe o subitem materiais (*materials*), mostrado na Figura 32, no qual é possível ver os materiais atualmente

utilizados na análise. É possível ver que o material genérico (*generic*) está presente, e ao lado dele um indicador de erro, pois ele não deve ser utilizado, visto que não possui as características físicas definidas.

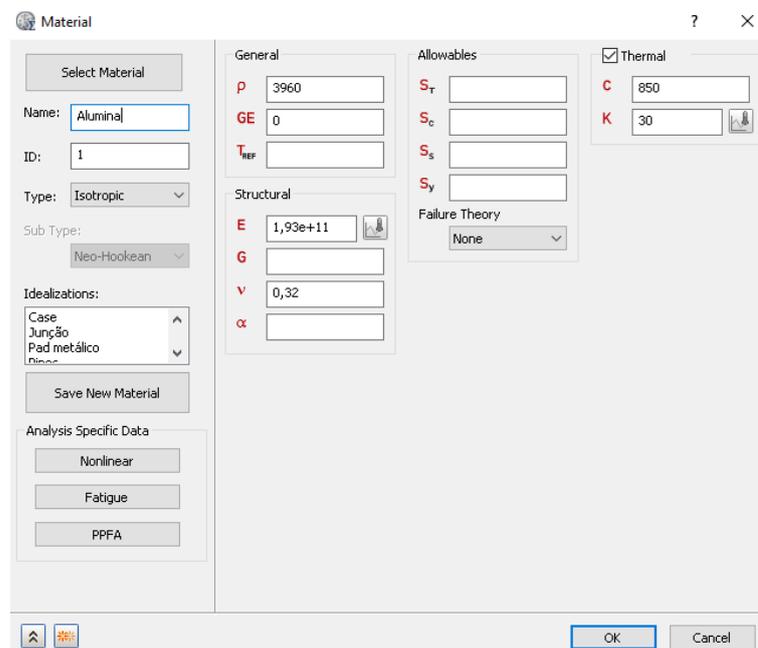
Figura 32 – Subitem materiais na árvore de *design*.



Fonte: Software Autodesk Nastran Inventor.

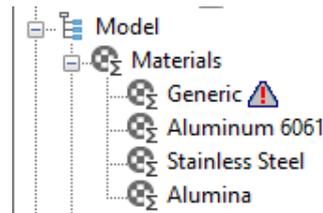
Para criar um novo material, clica-se com o botão direito no subitem materiais, e seleciona-se a opção novo (*new*), que abre a janela mostrada na Figura 33, onde já foi preenchido o nome e as características térmicas para criação do material do isolador, alumina. As características necessárias são a densidade de massa, condutividade térmica e o calor específico, abaixo da região térmica (*thermal*), que deve estar selecionada, como na Figura 33. As outras características não são importantes para esta análise e explicá-las foge ao escopo deste trabalho. Na opção idealizações (*idealizations*), é possível atribuir este material a um dos sólidos do modelo, caso já haja algum sólido criado.

Figura 33 – Criação de novo material.



Fonte: Software Autodesk Nastran Inventor.

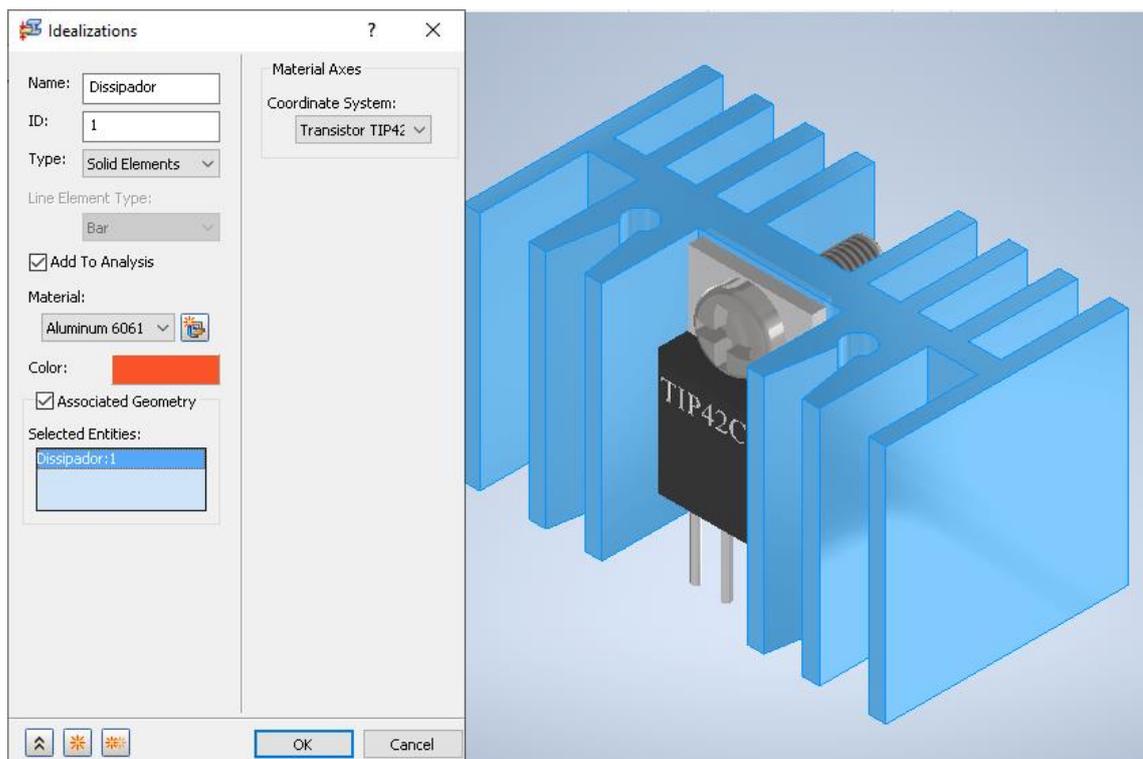
Figura 34 – Subitem materiais na árvore de *design* com novo material criado.



Fonte: *Software Autodesk Nastran Inventor*.

De forma análoga são criados todos os materiais necessários. Outro passo necessário é a criação dos sólidos do modelo, que é feita no subitem idealizações (*idealizations*) do item de modelo da árvore, clicando com o botão direito em sólidos (*solids*) e selecionando a opção novo (*new*), abrindo a janela mostrada na Figura 35, do lado esquerdo, na qual é possível dar um nome ao sólido, selecionar o material na lista de materiais criados no passo anterior. Além disso, marca-se a opção geometria associada (*associated geometry*) e clicando na caixa entidades selecionados (*selected entities*), pode-se selecionar no modelo o sólido desejado, como mostrado na Figura 35, ao lado direito.

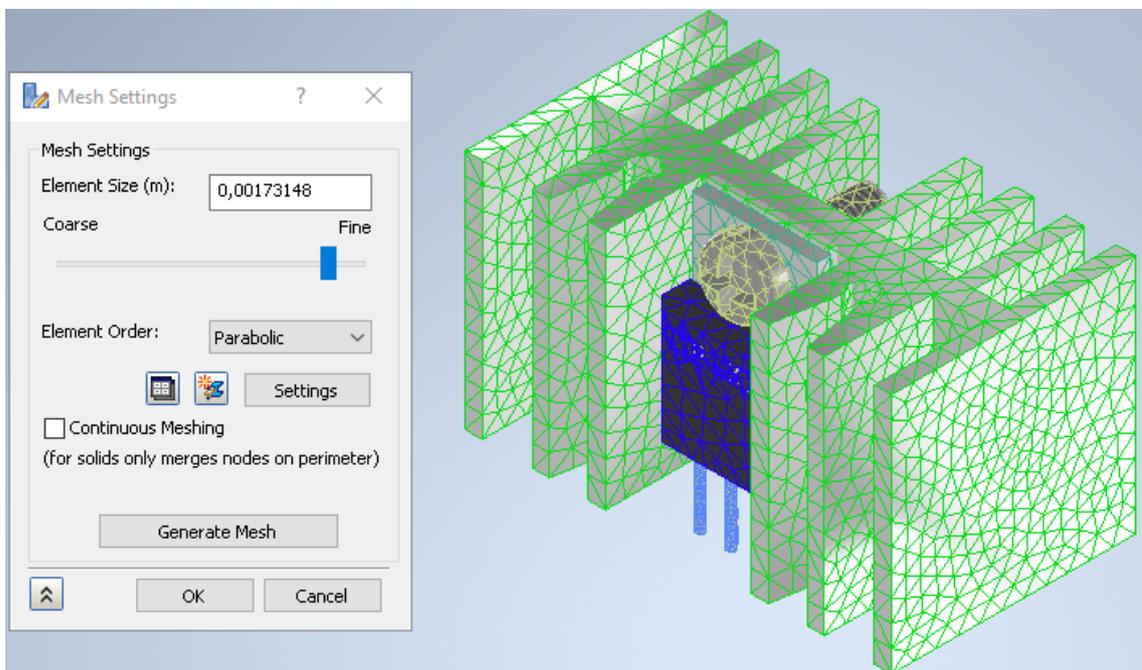
Figura 35 – Criação de sólido.



Fonte: Próprio Autor.

Analogamente, são criados todos os sólidos do modelo. Então, é possível criar a malha na opção opções de malha (*mesh settings*) no *menu* superior, no espaço de malha (*mesh*). É aberta a janela mostrada na Figura 36, no lado esquerdo, na qual clica-se no botão gerar malha (*generate mesh*). Também é possível escolher o tamanho do elemento da malha, refinando o cálculo ou tornando-o menos refinado. O resultado é mostrado no lado direito da Figura 36, onde é possível ver as malhas criadas no modelo. Em seguida, é necessário inserir as cargas (*loads*) da simulação, que numa simulação térmica podem ser: temperaturas, convecção, fluxo de calor, escoamento de calor, geração de calor, radiação e superfícies adiabáticas. É possível perceber a analogia que é comumente feita com circuitos elétricos, onde há uma fonte, seja de tensão ou corrente, que fornece potência, e os componentes, que dissipam ou utilizam essa potência para fins específicos.

Figura 36 – Criação de malhas.

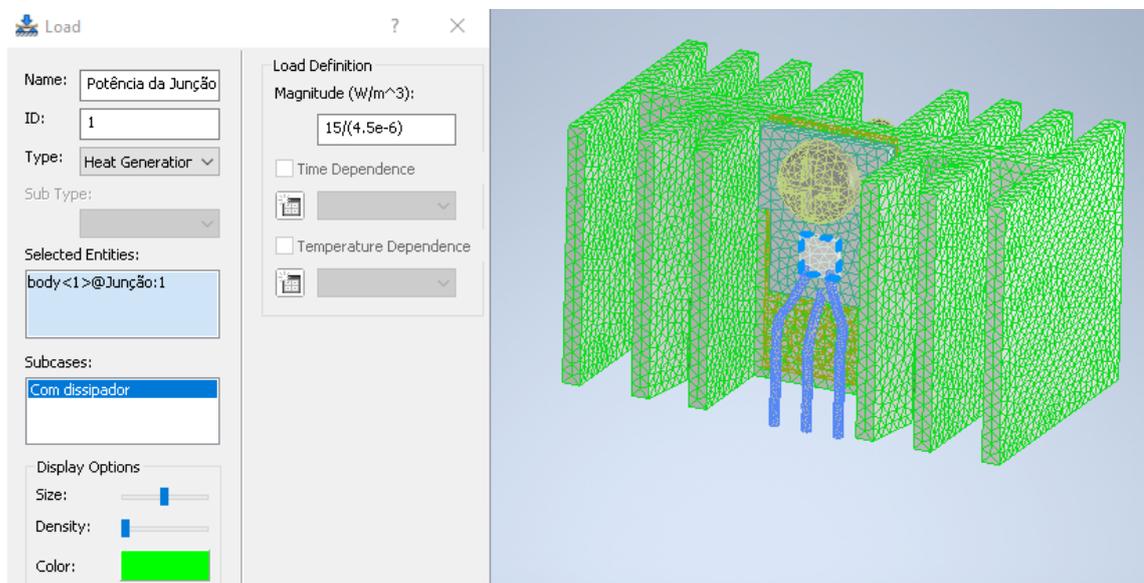


Fonte: Próprio Autor.

A criação de cargas pode ser feita pela barra superior, no espaço configuração (*setup*), clicando em cargas (*loads*), ou através da árvore, no subitem subcasos (*subcases*), clicando com o botão direito em cargas e selecionando novo (*new*). A janela aberta é mostrada na Figura 37, à esquerda, onde seleciona-se o tipo como geração de calor (*heat generation*), pode-se nomear a carga e as entidade selecionadas, clicando no modelo, à direita, onde foi retirada a visibilidade da *case* temporariamente para conseguir selecionar a junção.

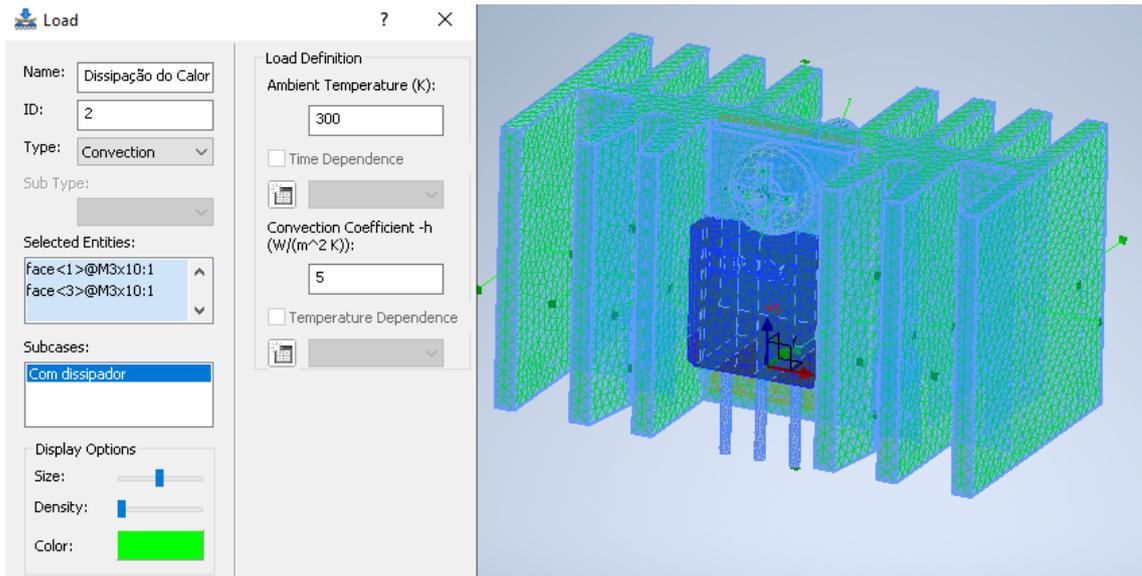
Além disso, é possível escolher a magnitude da potência do calor gerado, em W/m^3 . Como exemplo, são utilizados 15 W, e para conseguir inserir essa potência, pode-se inserir no campo de magnitude de potência, a divisão da potência de 15 W pelo volume da peça, no caso, 4.5 mm^3 , ou seja, $4.5 \cdot 10^{-6} \text{ m}^3$, conforme mostrado na Figura 37. Analogamente, adiciona-se a carga de convecção, considerando a convecção do ar de $5 \text{ W}/(\text{m}^2 \text{ K})$ e a temperatura ambiente de 300 K ($27 \text{ }^\circ\text{C}$) e selecionando todas as faces metálicas do modelo que trocam calor com o ar, ou seja, toda a superfície do dissipador de calor, a chapa metálica e os pinos do componente. O resultado é mostrado na Figura 38, onde é possível ver as cargas, representadas pelas setas verdes direcionadas do dissipador para o ar.

Figura 37 – Criação de carga de geração de calor.



Fonte: Próprio Autor.

Figura 38 – Criação de carga de convecção.

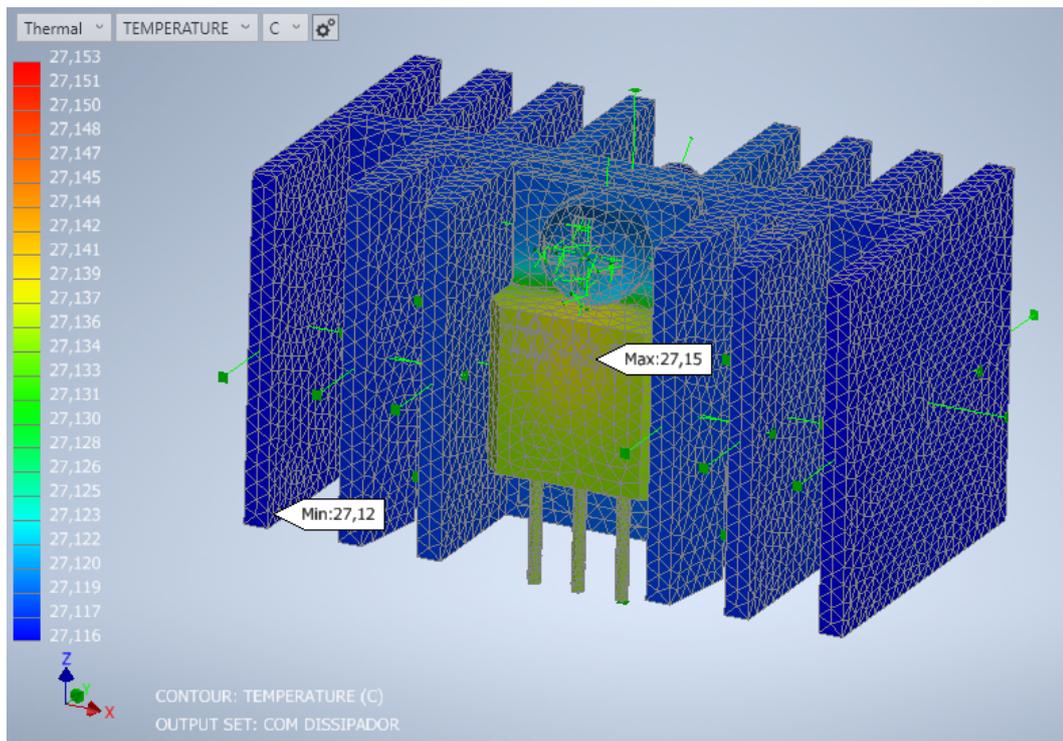


Fonte: Próprio Autor.

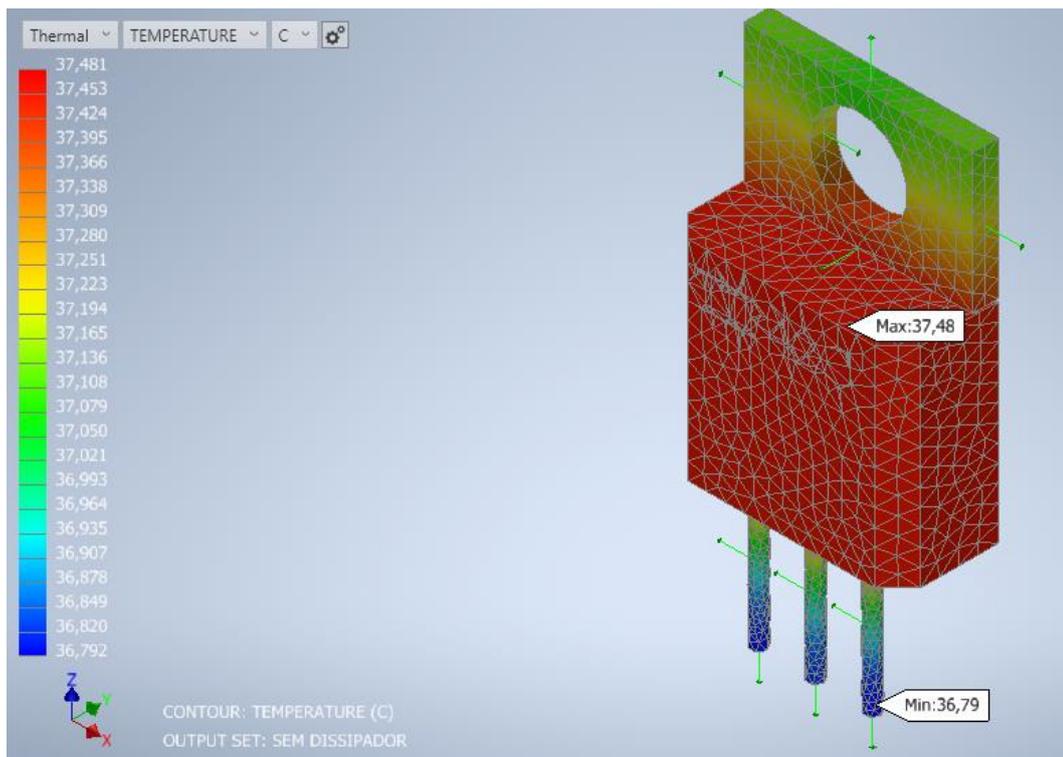
Com as cargas definidas, é possível rodar a simulação, clicando na opção rodar (*run*), no espaço resolver (*solve*) da barra superior. Enquanto o algoritmo é executado, na barra lateral esquerda do ambiente de trabalho, no lugar da árvore de *design* do *Nastran*, pode-se acompanhar os comandos do algoritmo. Ao terminar, automaticamente a barra lateral esquerda volta a mostrar a árvore, com um item novo dentro dos subcasos, de resultados, com o item temperatura abaixo desse. Além disso, é possível visualizar os resultados de temperatura na própria peça, através de cores com uma legenda, conforme a Figura 39.

Além disso, a fim de comparação, é feita uma outra simulação térmica, seguindo os mesmos passos descritos nesta seção e com todos os parâmetros iguais, mas sem a presença de um dissipador de calor conectado ao componente. Os resultados desta simulação são mostrados na Figura 39 (b), e permitem a comparação com os resultados obtidos na simulação com dissipador, mostrada na Figura 39 (a). Através da comparação entre as duas figuras, é possível perceber a diferença entre a simulação com e sem a presença do dissipador de calor, verificando a importância deste e o funcionamento da simulação térmica.

Figura 39 – Resultado da simulação térmica. (a) Com dissipador de calor; (b) Sem dissipador de calor.



(a)



(b)

Fonte: Próprio Autor.

3.3 Aplicação da Interatividade MCAD e ECAD

Esta seção explica a interatividade entre MCAD e ECAD, buscando discutir quais os pontos insuficientes dos *softwares* ECAD que são supridos por *softwares* MCAD, reiterando o que está apresentado em seções anteriores. Além disso, essa seção explica sucintamente o uso dos *softwares* a fim de realizar a interatividade, fornecendo explicações com passo a passo e exemplos para esclarecer o método utilizado.

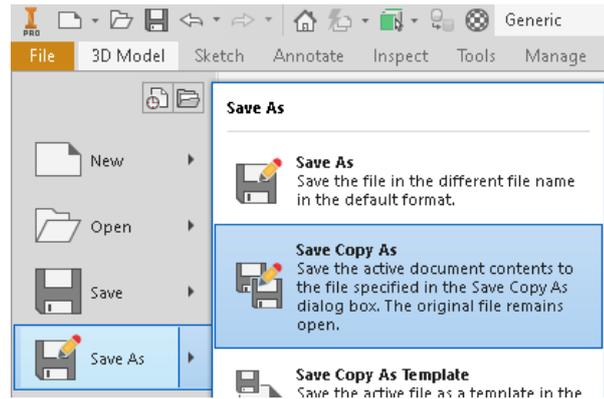
3.3.1 Complementariedade Entre os Softwares

Como apresentado nas seções anteriores, há características do *Altium Designer* que são insuficientes para um projeto mais bem elaborado, inclusive quanto a componentes mecânicos, pois sua possibilidade de criação e interação com modelos 3D, embora exista, é bem limitada. E quanto à possibilidade de simulação térmica, o *Altium Designer* não possui. Por isso, essas são funcionalidades que são muito bem supridas pelo *Autodesk Inventor* e *Inventor Nastran*, como é demonstrado nas seções adiante.

3.3.1.1 Exportação de Modelos 3D para o Altium Designer

Primeiramente, após a criação de modelos 3D no *software* MCAD *Autodesk Inventor*, é necessária a exportação desses arquivos para os arquivos de biblioteca de PCI do *Altium Designer*, começando a interação MCAD e ECAD. O primeiro passo desse processo envolve a exportação do modelo 3D, seja peça ou montagem, no formato que o *Altium Designer* reconhece, os formatos “universais” .step ou .stp. Esse processo, mostrado na Figura 40, é feito através do *menu* de arquivo (*file*) da barra de navegação superior, clicando na seta ao lado da opção salvar como (*save as*) e em salvar cópia como (*save copy as*), processo mostrado na Figura 40. Seleciona-se o formato do arquivo como .step ou .stp e escolhe-se onde salvar, usando o sistema de arquivos do *Windows*.

Figura 40 – Exportação de um arquivo do *Inventor* em *.step*.

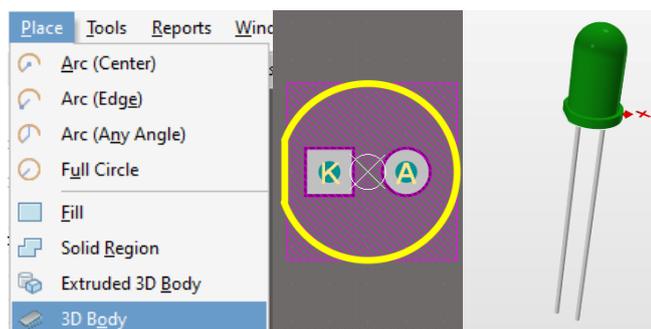


Fonte: *Software Autodesk Inventor*.

No *Altium Designer*, é possível importar arquivos *.step* para dois ambientes diferentes, através do mesmo *menu* de inserir (*place*), na opção de objeto 3D (*3D body*). Pode-se importar o arquivo *.step* de um componente para a biblioteca de PCI deste componente, a fim de associar o modelo 3D ao componente, e sempre que este for utilizado, o modelo 3D será automaticamente colocado junto com o componente, conforme a Figura 41.

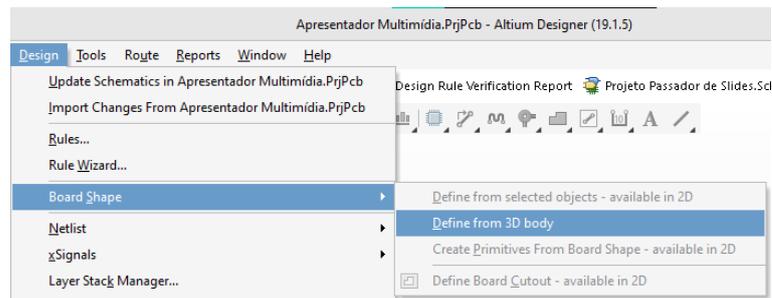
Além disso, também é possível a importação do arquivo *.step* do modelo 3D de uma PCI sem componentes direto para o ambiente de criação de PCI, para importar o formato desta PCI para a PCI criada no *Altium*, possibilitando a criação de PCIs em qualquer formato, através do modo de visualização 3D, no *menu* de *design*, em formato da placa (*board shape*), definir a partir de um objeto 3D (*define from 3D body*), processo ilustrado na Figura 42. Esta funcionalidade é muito importante pois comumente, o espaço limitado e formato da PCI são previamente definidos durante a criação do projeto mecânico do equipamento, e devem ser seguidos.

Figura 41 – Exportação de um modelo 3D de um componente.

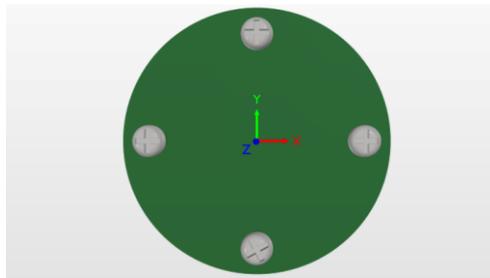


Fonte: *Software Altium Designer*.

Figura 42 – Importação de um modelo 3D das dimensões e formato de uma PCI. (a) Importação; (b) Modelo 3D com dimensões e formato da PCI importado.



(a)



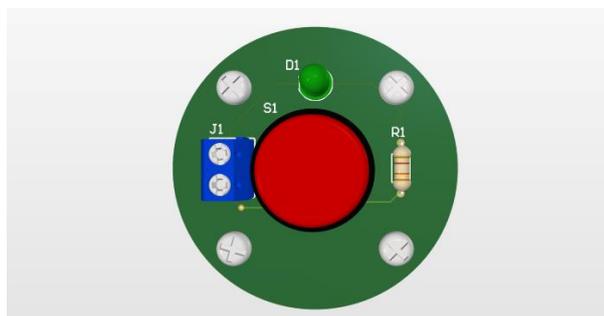
(b)

Fonte: *Software Altium Designer.*

3.3.1.2 Montagem de PCI e Exportação para o Autodesk Inventor

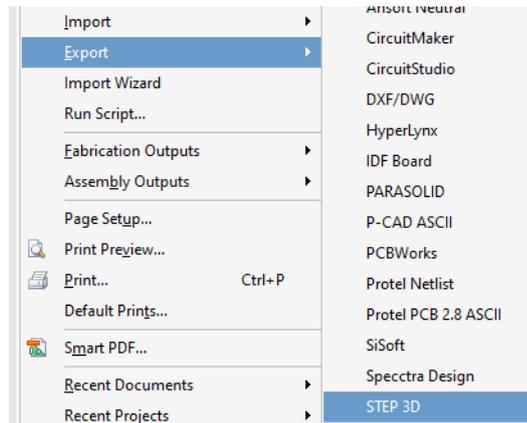
Após a importação do modelo 3D de todos os componentes a serem utilizados e do formato da placa, se for o caso, é feita a montagem da PCI normalmente, conforme os passos descritos em seções anteriores. Ao fim da montagem da PCI, é possível visualizar o modelo 3D da PCI com todos os componentes no *Altium Designer*, mostrado na Figura 43, e exportar esse modelo 3D com todos os componentes no formato .step, para abrir no *Autodesk Inventor*, conforme mostrado na Figura 44, através do *menu* de arquivo (*file*), na opção exportar (*export*) e step 3D

Figura 43 – PCI montada no *Altium Designer*.



Fonte: *Software Altium Designer.*

Figura 44 – Exportação de um modelo 3D de uma PCI em .step no *Altium Designer*.

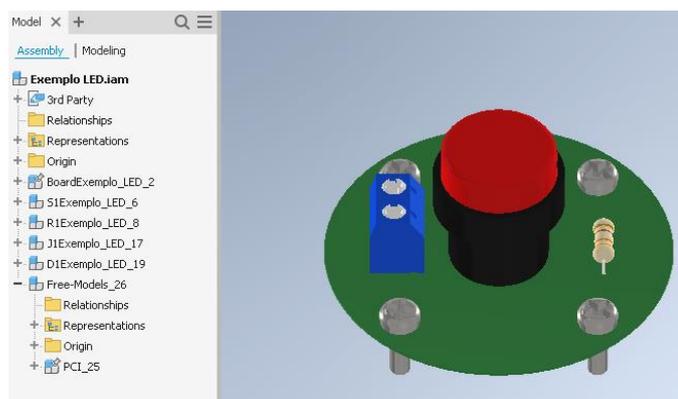


Fonte: *Software Altium Designer*.

3.3.1.3 Adequação dos Componentes via Análise Espacial

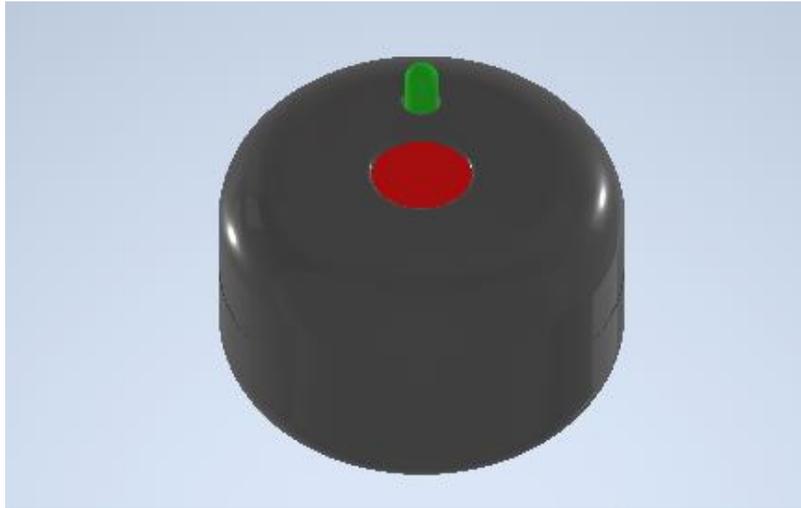
Com o arquivo .step exportado, é possível abri-lo normalmente no *Autodesk Inventor*, no qual é possível ter acesso a placa, com todos os componentes importados e separados na árvore de *design*, e fazer as análises e modificações necessárias, conforme a Figura 45. Nessa etapa, é possível readequar posições de componentes, furos, e outros com maior controle do que no *Altium*, além de ser possível adequar a placa a invólucros (ou *cases*), com posições adequadas de conectores, botões, LEDs, dentre outros. Bem como é possível, se desejado, fazer simulações térmicas e simulações de comportamentos mecânicos em geral, com análise por elementos finitos. Como exemplo, adequa-se a PCI feita a um invólucro feito para exemplificar esta situação, mostrado na Figura 46.

Figura 45 – Importação da PCI com os componentes para o *Autodesk Inventor*.



Fonte: *Software Autodesk Inventor*.

Figura 46 – Case com PCI de exemplo encaixada.



Fonte: *Software Autodesk Inventor*.

3.3.1.4 Alterações de Posições de Componentes no Altium Designer

No exemplo utilizado não houve necessidade, mas, em muitos casos reais, é necessário mudar as posições planejadas no esboço inicial feito no *Altium Designer* após as readequações de posicionamento realizadas no *Autodesk Inventor*. Para este fim, é possível medir as distâncias no *Inventor* com relação ao sistema de coordenadas, para modificação no *Altium*, visto que a modificação é mais precisa caso seja utilizado este método. Além disso, existem casos em que pode ser necessária até mudança de determinados componentes por não se adequarem ao projeto mecânico. Ou caso seja realizada uma simulação térmica e esta não seja satisfatória, é melhor mudar ou mesmo refazer o projeto eletrônico, para não correr o risco de queimar componentes devido a uma dissipação de calor não satisfatória.

3.3.2 Vantagens do Uso Complementar dos Softwares

Na seção anterior é possível ver como utilizar os dois *softwares* de forma interativa, obtendo o melhor de cada um e suprindo os requisitos que um não satisfaz com o uso de outro. Também é possível incrementar mais ainda a interatividade e melhorar o projeto através da inclusão de *softwares* ECAD de simulação eletrônica, verificando o funcionamento adequado do circuito e obtendo as potências dissipadas, o que permitiria uma simulação térmica mais realista no *software* MCAD, porém foge ao escopo planejado neste trabalho.

Ao aplicar este método, é possível ver como o *Autodesk Inventor* é bem mais adequado à criação e manipulação de modelos 3D, otimizando a utilização desses modelos para criação de projetos eletrônicos, uma funcionalidade que o *Altium* apenas introduz e permite limitada exploração do potencial desta. Portanto, conforme já mencionado em seções anteriores, é possível verificar como a utilização deste método traz substancial melhora ao desenvolvimento de projetos eletrônicos, com a utilização e manipulação de modelos 3D, adequação espacial dos componentes de acordo com a necessidade do projeto mecânico e possibilidade de simulações térmicas e dentre outras, através do método dos elementos finitos.

4 APRESENTADOR MULTIMÍDIA CONTROLADO COM OS PÉS

Esta seção faz a descrição e discussão do projeto utilizado como aplicação do método de interatividade entre *softwares* MCAD e ECAD, seguindo os passos descritos nas seções anteriores e mostrando os resultados esperados com a aplicação do método.

4.1 Introdução Conceitual

4.1.1 Breve Descrição do Projeto

O projeto é um dispositivo utilizado para fazer apresentações multimídia, como por exemplo, de *slides*, com a utilização dos pés, no lugar das mãos, como os apresentadores multimídia comuns no mercado, como os dois apresentados em Okuno e Guedes (2017) e *Wireless USB Presenter with Laser Pointer (Black)*, 2019. Assim como os dispositivos citados, este é formado de duas partes, um receptor, que deve ser conectado no computador no qual será realizada a apresentação através de um cabo USB micro, e um transmissor, que também contém as funcionalidades de interação com o usuário. O dispositivo receptor é alimentado diretamente pelo computador, enquanto o transmissor pode ser alimentado por baterias de 9V ou através de uma fonte de 9V com saída P4.

É interessante destacar que o dispositivo desenvolvido segue o conceito de *plug and play*, conforme o dispositivo em *Wireless USB Presenter with Laser Pointer (Black)*, 2019, visto que, não há a necessidade de instalação de nenhum *driver* adicional para o uso. Basta conectar o receptor ao computador e ligar o dispositivo transmissor e o equipamento está pronto para uso. Porém, comparando com os dois equipamentos citados no parágrafo anterior, o equipamento desenvolvido não possui a funcionalidade de apontador *laser*, já que é projetado para uso com os pés, ficando fixo no chão a frente do usuário. Outra diferença consiste na utilização de tecnologia *Bluetooth* para a comunicação entre transmissor e receptor. Além disso, uma funcionalidade presente no equipamento desenvolvido e ausente nos equipamentos citados, é o LED indicador de carga e conexão entre transmissor e receptor, aumentando a interação com o usuário.

O transmissor consiste em: uma caixa metálica robusta, para aguentar o esforço da utilização com os pés; dois botões utilizados em pedaleiras, também buscando a robustez necessária; 2 LEDs indicativos dos botões; um LED indicador de bateria e de conexão com o receptor; um *switch* para ligar e desligar; um suporte para encaixe de baterias de 9V interno; e um conector P4 para uso conectado à tomada. Internamente, o transmissor possui uma PCI que faz o gerenciamento do sistema e é melhor detalhado nas seções seguintes. Além disso, ambos os componentes do dispositivo são controlados por diferentes placas de prototipagem Arduino, um Arduino Pro Micro para o receptor e um Arduino Nano para o transmissor

Para realização da comunicação entre os dois componentes são utilizados módulos *bluetooth* HC05 configurados no modo mestre-escravo (*master-slave*), que permite a conexão automática de um dos dispositivos, considerado o mestre, no caso o transmissor, a apenas outro dispositivo, através do seu endereço, considerado o escravo, no caso o receptor, estando os dois ligados. Esta configuração é utilizada pois previne a conexão dos dispositivos a outros aparelhos com *bluetooth* e conecta assim que os dois dispositivos são ligados, apenas com um pequeno *delay*. Porém, uma desvantagem dessa configuração é que cada transmissor funciona apenas com seu receptor próprio e vice versa.

4.1.2 Diferencial Observado

Este dispositivo possui como principal diferença aos apresentadores multimídia comerciais a não necessidade de utilização das mãos, tornando-o possível de ser utilizado enquanto as mãos do usuário estão livres para desempenhar outras funções, como manipular outros objetos durante a apresentação ou escrever numa lousa, dentre outros. Também é importante destacar que este dispositivo permite a utilização por pessoas com deficiências (PCDs) no movimento dos braços ou mãos ou mesmo no caso de ausência destes, sendo uma solução inclusiva. Além disso, há maior interação com o usuário, devido a presença de um LED que indica se existe conexão entre o transmissor e receptor, e, quando há essa conexão, indica a estimativa do estado de carga da bateria do equipamento, dando ao usuário a informação de quando será necessário trocar as baterias.

4.2 Pré-Projeto Mecânico do Apresentador Multimídia

Esta seção apresenta o projeto mecânico do dispositivo apresentador multimídia, que consiste: no invólucro plástico do receptor, podendo ser produzido por impressão 3D, injeção, ou outros métodos de produção de produtos de plástico; e da caixa metálica do receptor, que foi planejada para ser produzida por placas de metal dobradas, mas também pode ser produzidos por outros métodos de produção de produtos metálicos. Além disso, é também no projeto mecânico que são definidas as posições desejadas dos elementos de interação com o usuário, ou seja, LEDs, botões e conectores, bem como o suporte da bateria.

4.2.1 Adequação Espacial dos Componentes Eletrônicos

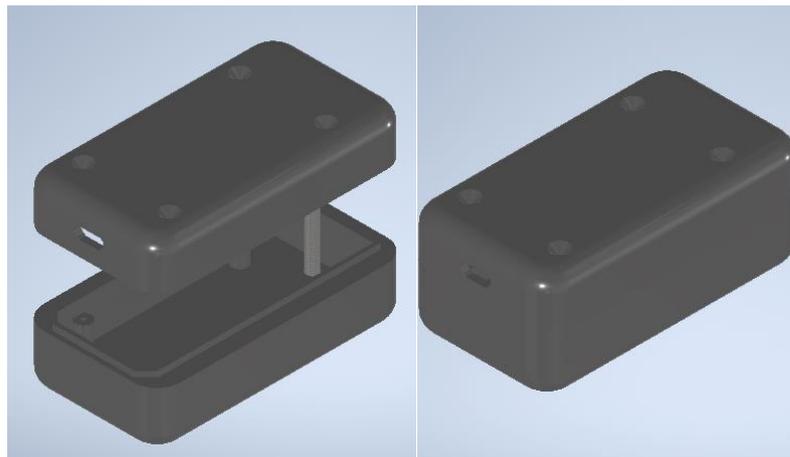
A adequação espacial dos componentes eletrônicos foi feita primeiramente no *software* MCAD, no caso, *Autodesk Inventor*, após a definição dos componentes eletrônicos no projeto eletrônico, embora este seja explicado posteriormente neste trabalho. A adequação se faz necessária pois é preciso definir as posições dos componentes que farão interação com o usuário, bem como a forma que estes serão fixados na caixa metálica. A caixa metálica pode ser considerada como um tipo de painel, sendo utilizados os suportes de componentes ou componentes adequados para fixação em painel, com encaixes com rosca ou através de encaixes por interferência, no caso do *switch* liga/desliga.

É definido que o dispositivo receptor só necessita de um espaço para que o conector USB fique a mostra, sendo o esboço do invólucro realizado na próxima seção em função disso. Quanto ao dispositivo transmissor, é definido que na parte superior, serão localizados os botões e os 3 LEDs, estando os botões a 10 cm de distância, com cada LED correspondente acima do botão e o LED de indicação de bateria e conexão no ponto central entre os outros dois LEDs. Numa lateral é colocado o *switch* liga/desliga e na outra o conector P4 para a fonte de 9V. Embaixo há a tampa removível na qual é presa o suporte de bateria, a fim de ser possível de ser removida para que o usuário tenha acesso à bateria, podendo trocá-la.

4.2.2 Projeto dos Invólucros dos Componentes do Apresentador Multimídia

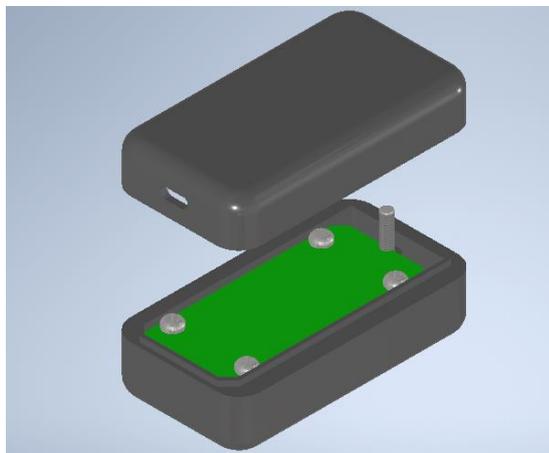
O projeto dos invólucros dos dois componentes do apresentador multimídia é feito com base nas considerações definidas na seção anterior. Primeiramente é feito o projeto do invólucro plástico do receptor, que consiste em duas peças que se encaixam, conforme mostrado na Figura 47. Então, com este invólucro feito, é possível fazer o esboço da PCI, definindo suas dimensões e furações de fixação, conforme mostrado na Figura 48.

Figura 47 – Invólucro do receptor.



Fonte: Próprio autor

Figura 48 – Invólucro do receptor com esboço de PCI.

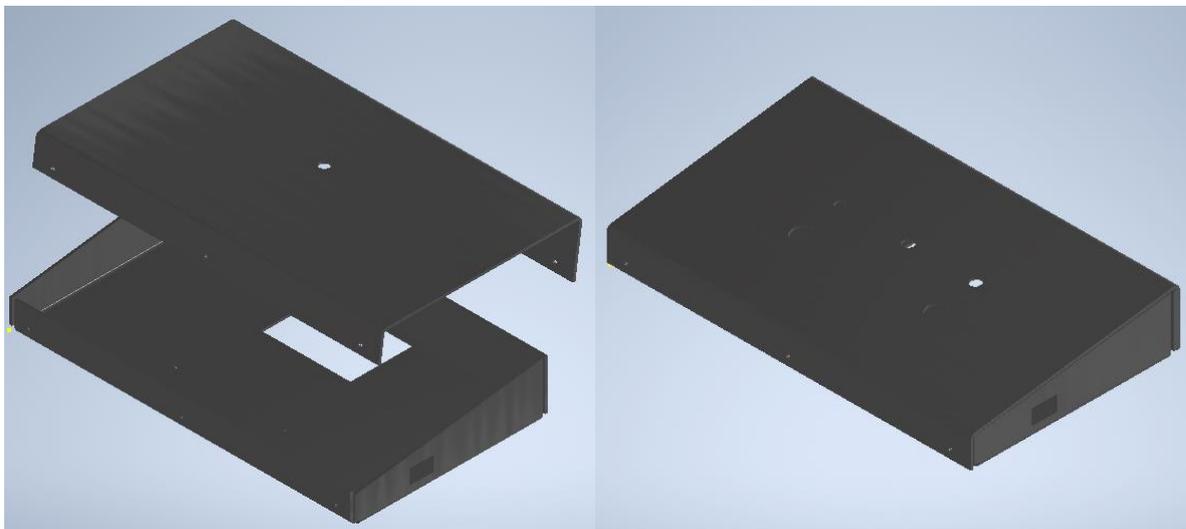


Fonte: Próprio autor

Em seguida, é feito o projeto do invólucro do transmissor, que é uma caixa metálica, com duas peças de metal dobrado que se encaixam, conforme mostrado na Figura 49. Então, com este invólucro feito, é possível fazer o esboço da PCI, definindo suas dimensões e furações

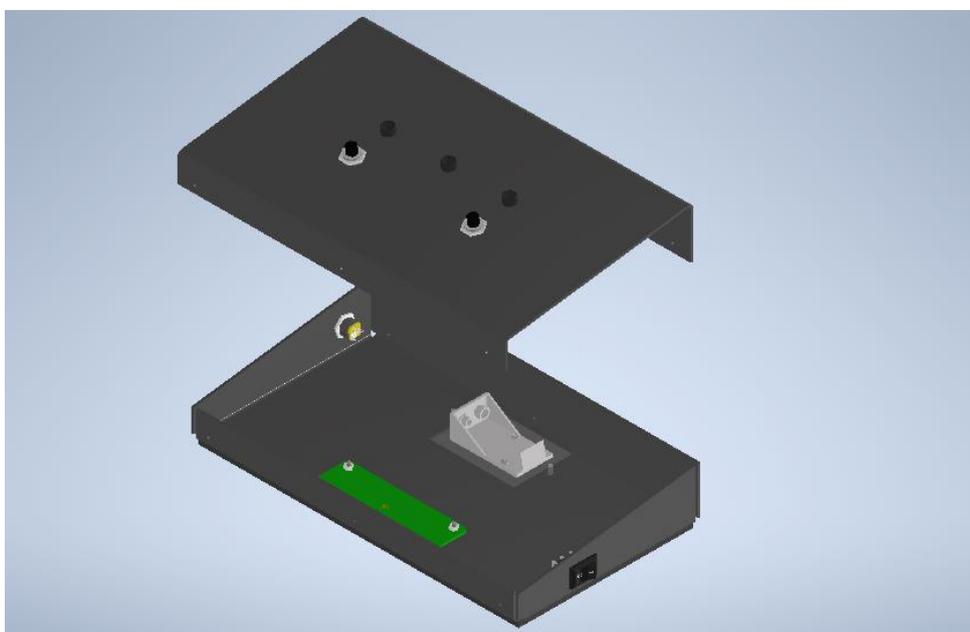
de fixação, bem como a definição dos locais dos componentes de interação com o usuário, conforme mostrado na Figura 50. Neste caso, foram apenas colocados os suportes de alguns dos componentes de interação com o usuário, sendo os componentes inseridos nas próximas seções.

Figura 49 – Invólucro (caixa metálica) do transmissor.



Fonte: Próprio autor

Figura 50 – Invólucro (caixa metálica) do transmissor com esboço de PCI e componentes de interação com o usuário.



Fonte: Próprio autor

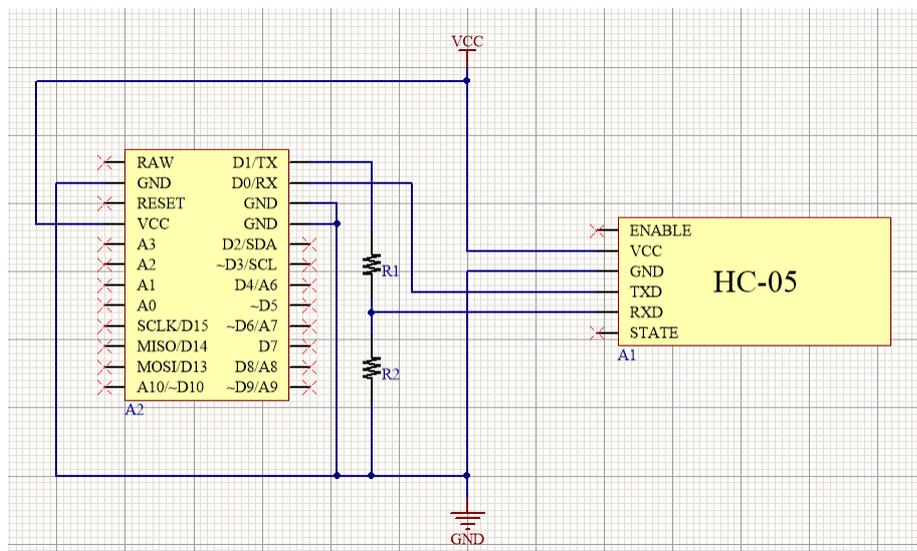
4.3 Pré-Projeto Eletrônico do Apresentador Multimídia

Esta seção apresenta o projeto eletrônico do dispositivo apresentador multimídia, que consiste: nos dois componentes, transmissor e receptor. Nesta seção são mostrados os esquemáticos de projetos feitos no *Altium Designer* e comentado sobre a programação dos microcontroladores.

4.3.1 Definição dos Circuitos Eletrônicos e Criação dos Esquemáticos

Primeiramente, é definido o circuito eletrônico do receptor, que consiste somente no Arduino Pro Micro, no módulo *bluetooth* HC05 utilizado como receptor e escravo (*slave*) e em um divisor de tensão para adequar o nível de tensão do Arduino ao do HC05. O Arduino Pro Micro é utilizado pois possui uma funcionalidade que permite controle do teclado do computador conectado através do código em execução no microcontrolador, permitindo a criação do dispositivo apresentador multimídia em questão da forma como foi planejado. O cabo USB micro do receptor faz a conexão do computador ao Arduino Pro Micro, permitindo esta interface. A Figura 51 mostra o esquemático do circuito do receptor.

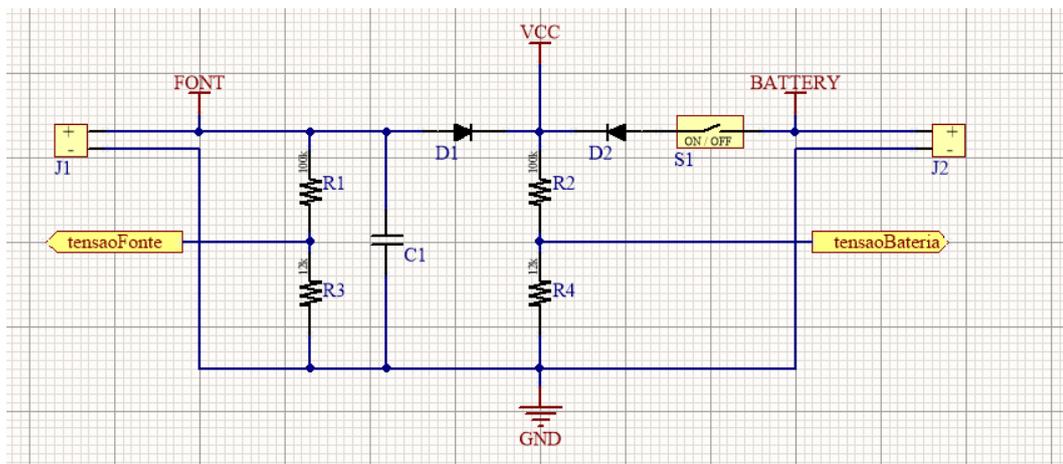
Figura 51 – Esquemático do circuito do receptor.



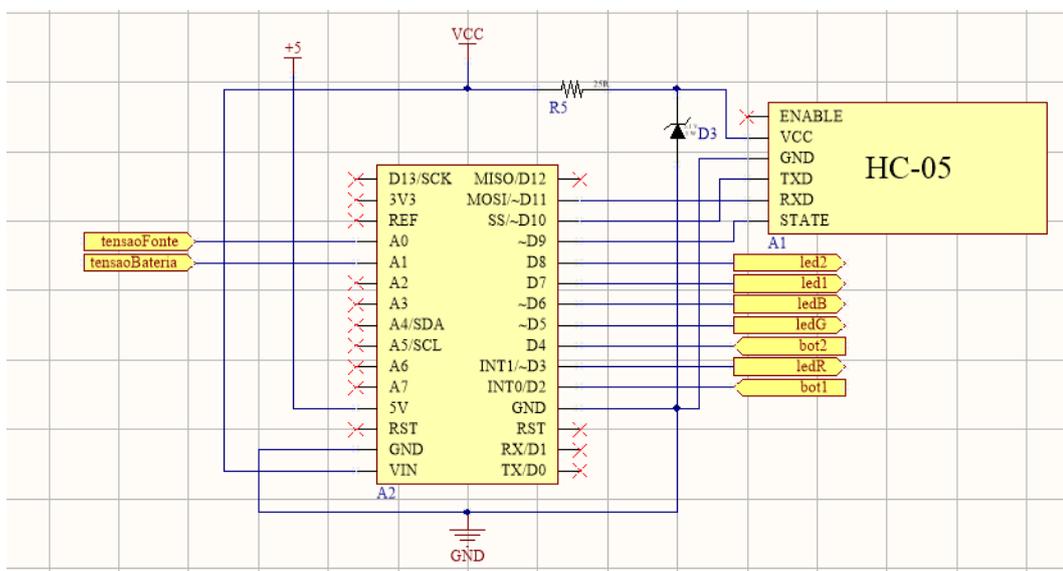
Fonte: Próprio autor

Quanto ao transmissor, o circuito é um pouco mais complexo do que o do receptor. Possui um Arduino Nano como microcontrolador, gerenciando as entradas e saídas, botões, os LEDs, um circuito para monitorar a tensão na bateria e mostrar no LED RGB, e uma parte de proteção da bateria e proteção contra inversões de tensão, utilizando diodos, e um diodo Zener para regular a tensão de alimentação do HC05. Este circuito é dividido em três esquemáticos, um para a parte de alimentação e proteção, um para o circuito central do microcontrolador e do módulo *bluetooth* HC05 e um para as entradas e saídas digitais. Esses esquemáticos são mostrados na Figura 52.

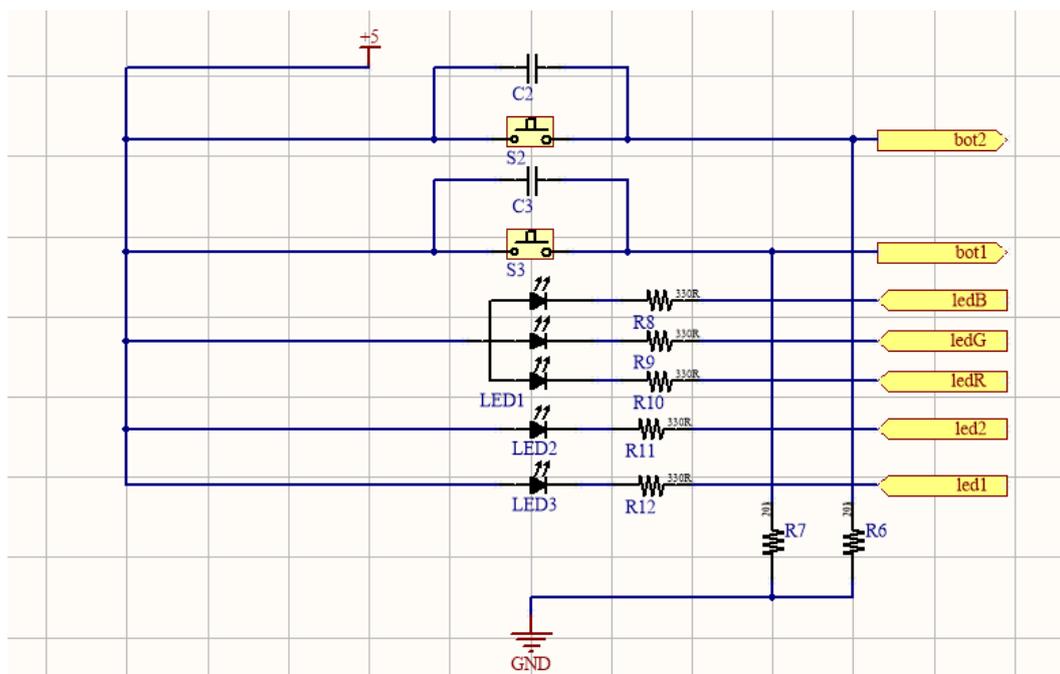
Figura 52 – Esquemático do circuito do transmissor (a) Alimentação e proteção; (b) Microcontrolador e *bluetooth*; (c) Entradas e saídas digitais.



(a)



(b)



(c)

Fonte: Próprio autor

4.2.2 Programação dos Microcontroladores

A programação do receptor também foi simples, bem como seu circuito. Utilizou-se a biblioteca *Keyboard.h* para controle do teclado e o *bluetooth* é tratado no código como uma porta serial comum, tendo suas ligações sido feitas nos pinos da porta serial do Arduino. Desse modo, a lógica é que, quando é recebido um caractere “R” no *bluetooth*, é apertada a seta para a direita do teclado e quando é recebido o caractere “L”, é apertada a seta para a esquerda. Além disso, um caractere “0” serve como caractere de controle, indicando que nenhum botão foi apertado, utilizado para prevenir que sejam passadas várias páginas ao se receber vários “R” ou “L” consecutivos, só podendo passar páginas novamente quando tiver havido pelo menos um “0”. O código do receptor encontra-se no Anexo A.

Quanto à programação do transmissor, que também é um pouco mais complexa que a do receptor, assim como nos circuitos, utilizou-se a biblioteca *SoftwareSerial.h*, que cria uma porta serial virtual com quaisquer pinos escolhidos, no caso, 10 e 11, que são conectados ao módulo *bluetooth*, que também é tratado como uma porta serial comum. Este código possui duas funcionalidades principais independentes, o gerenciamento da função do dispositivo, passar os *slides*, e o gerenciamento da tensão da bateria para ser mostrada no LED RGB.

A função de passar *slides* se dá da seguinte forma: quando o usuário pressiona o botão esquerdo, a porta serial envia ao *bluetooth* um caractere “L” e o LED esquerdo acima do botão acende, quando o botão direito é pressionado, é enviado um “R”, e o LED direito correspondente acende, e quando nenhum botão é pressionado, os LEDs laterais permanecem apagados e é enviado um caractere de controle “0”. Os caracteres são tratados no código do receptor, conforme o parágrafo anterior. Além disso, caso ainda não tenha sido estabelecida conexão *bluetooth*, o que é medido pelo estado do pino estado (*state*) do módulo, ligado ao pino 9 do Arduino, o LED central fica piscando em azul.

Quando uma conexão *bluetooth* é estabelecida, o LED central para de piscar em azul e o dispositivo pode ser utilizado normalmente. Além disso, o LED central passa a, neste momento, mostrar uma estimativa do estado de carga da bateria, medido através da tensão desta, que é proporcional ao estado de carga, fornecendo uma aproximação. São definidos 3 níveis de tensão: 4 a 6 V, definido como carga baixa, e mostrando tons da cor vermelha; 6 a 8 V, considerado carga média, mostrando tons de vermelho e verde em conjunto, formando tons de amarelo; e de 8 a 10 V, considerado carga boa, mostrando tons de verde. O código do receptor encontra-se no Anexo B.

4.4 Aplicação do Método de Interatividade MCAD e ECAD

Nesta seção, é mostrado o resultado dos passos do método de interatividade MCAD e ECAD, descrito nas seções anteriores.

4.4.1 Pré-projeto Mecânico no Autodesk Inventor

É feito o pré-projeto mecânico do apresentador multimídia no *Inventor*, que consiste em definição do invólucro e posições dos componentes principais. Este processo já encontra-se descrito em detalhes na seção 4.2.

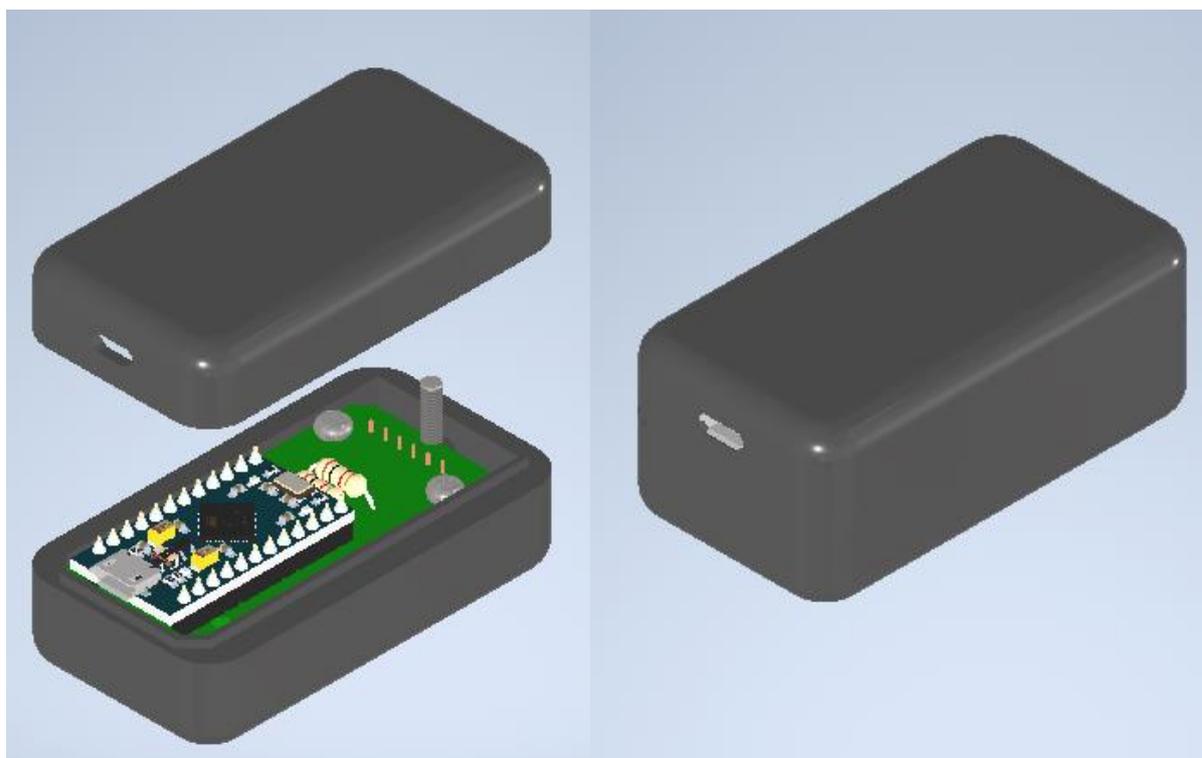
4.4.2 Pré-projeto Eletrônico da PCI no Altium Designer

É feito o pré-projeto eletrônico do apresentador multimídia no *Altium*, que consiste em definição dos circuitos e esquemáticos, com os componentes utilizados. Este processo já encontra-se descrito em detalhes na seção 4.3.

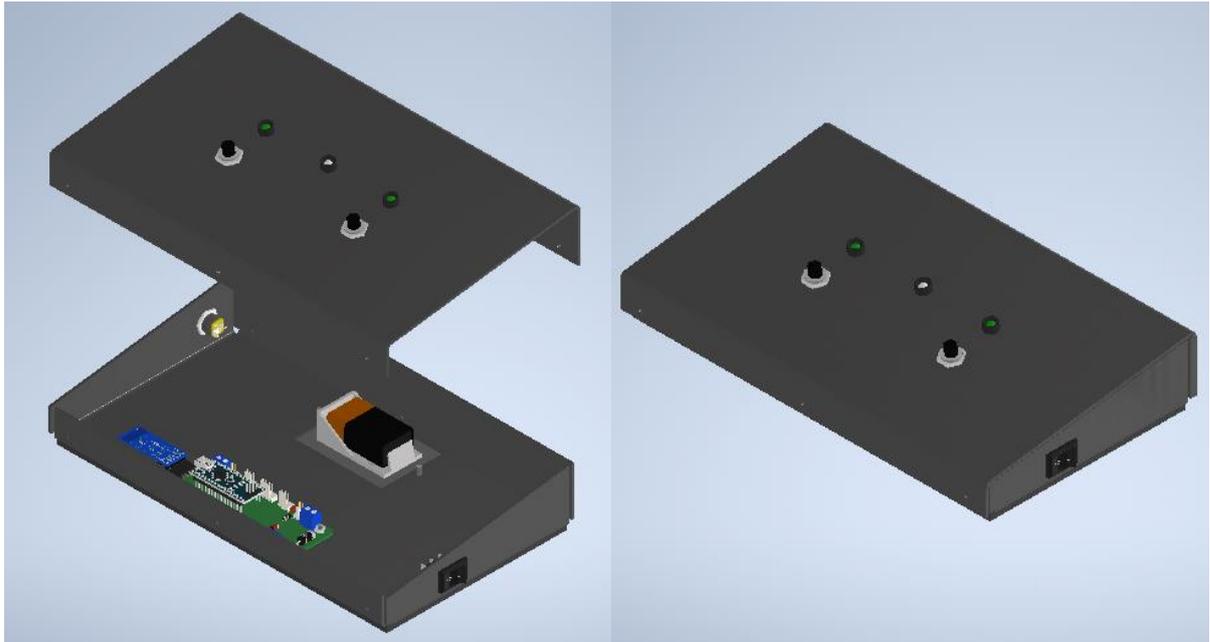
4.4.3 Finalização do Projeto Mecânico no Autodesk Inventor

Então, conforme descrito nas seções anteriores, com o pré-projeto eletrônico realizado, é possível voltar para o *Autodesk Inventor* e finalizar o projeto mecânico, colocando os componentes nos devidos lugares e definindo as melhores posições dos mesmos, conferindo possíveis interferências e corrigindo-as, se existirem. É nessa etapa que também é possível fazer as simulações térmicas e outras, se desejado. A Figura 53 mostra o projeto mecânico concluído, após correções de posicionamento de componentes, se necessário.

Figura 53 – Projeto mecânico concluído (a) Receptor; (b) Transmissor.



(a)



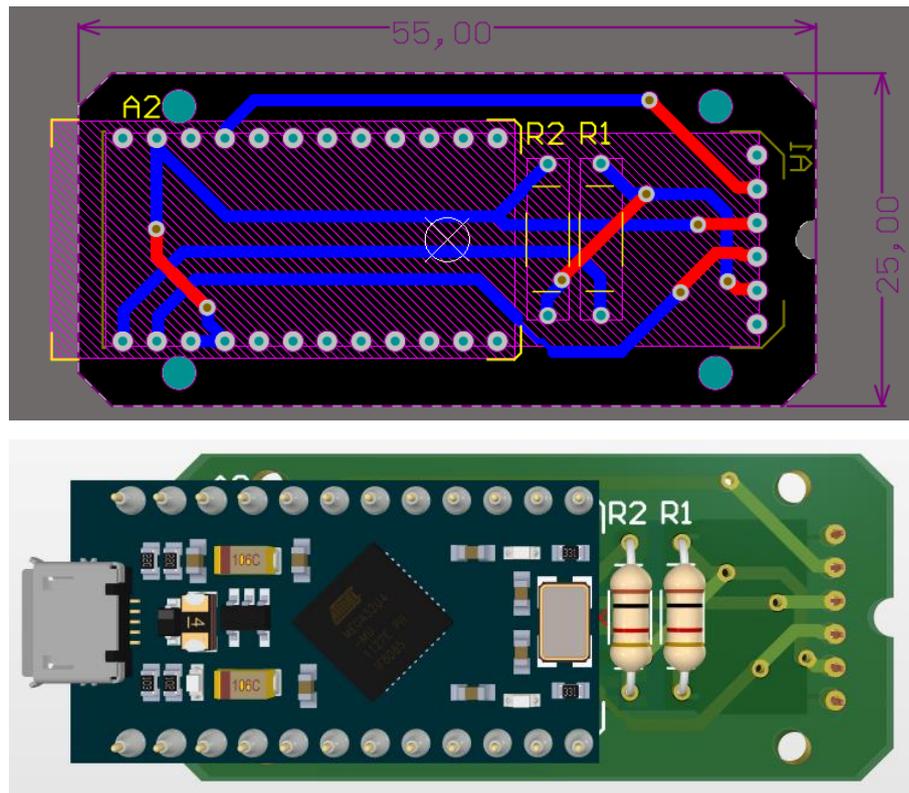
(b)

Fonte: Próprio autor

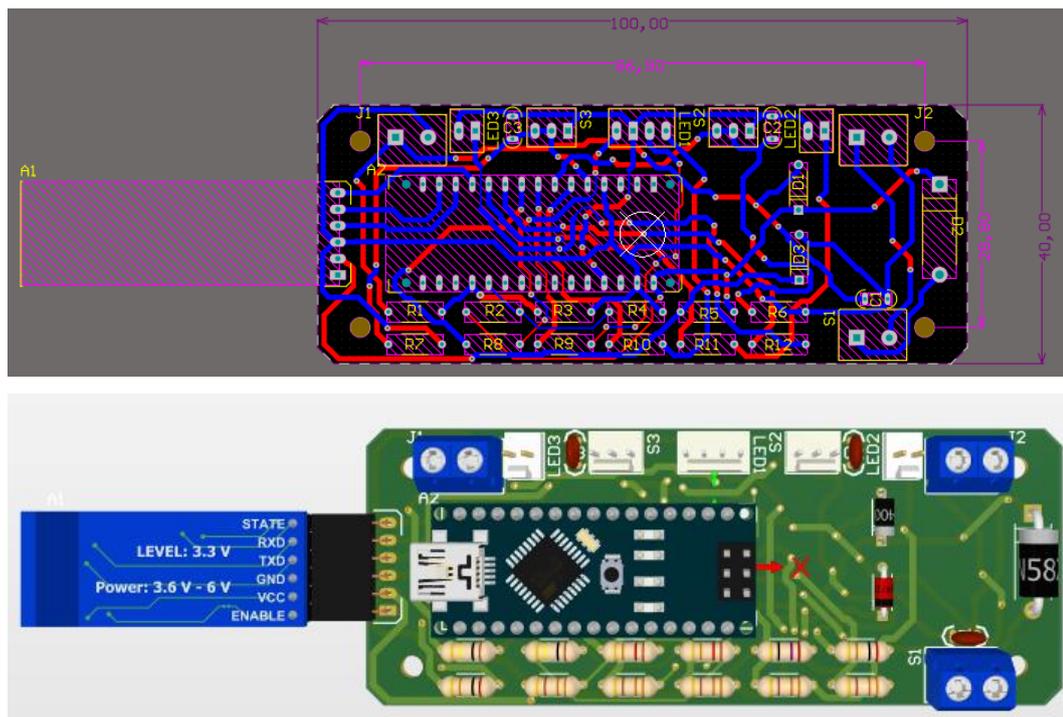
4.4.4 Finalização do Projeto Eletrônico da PCI no Altium Designer

Então, após a conclusão do projeto mecânico no *Inventor*, verificando possíveis problemas, resolvendo-os e fazendo as simulações desejadas, é possível voltar ao *Altium* e finalizar o projeto eletrônico. O que resta do projeto eletrônico a ser finalizado são possíveis mudanças de posicionamento de componentes, e então, o desenho das trilhas. Ao fim, confere-se com a ferramenta de verificação de regras de *design* (*design rules check*), e caso não haja nenhum erro, está finalizado o projeto. A Figura 54 mostra o projeto eletrônico concluído, após desenho das trilhas e verificação.

Figura 54 – Projeto eletrônico concluído (a) Receptor; (b) Transmissor.



(a)



(b)

Fonte: Próprio autor.

5 RESULTADOS E TRABALHOS FUTUROS

Nesta seção, são discutidos os resultados obtidos neste trabalho. Os resultados obtidos podem ser divididos em três seções, discussão sobre os *softwares* utilizados, apresentação do protótipo funcional da aplicação desenvolvida e uma discussão geral dos resultados obtidos com a utilização do método, destacando sua importância.

5.1 Discussão Sobre os *Softwares* Utilizados

Ao longo deste trabalho foram utilizados os *softwares Autodesk Inventor, Autodesk Inventor Nastran e Altium Designer*, sendo os dois primeiros *softwares* MCAD e o último um *software* ECAD. Foi possível perceber a limitação do *Altium Designer* na criação e manipulação de modelos 3D, bem como a ausência de simulação térmica. Porém, estas são funções mais inerentes a *softwares* MCAD, então, como ECAD, o *Altium* é bastante satisfatório, não só suprimindo o mínimo necessário como indo além, e apresentando funcionalidades que outros *softwares* ECAD não possuem, como por exemplo, o ambiente de visualização 3D, mesmo que limitado.

Quanto aos dois *softwares* MCAD, como esperado, não possuem funções de ECAD, sendo um complemento ao *Altium* neste método de interatividade proposto, visto que o foco da engenharia elétrica e eletrônica acaba sendo o desenvolvimento de produtos eletrônicos. Porém, conforme discutido neste trabalho, o projeto eletrônico é apenas uma parte do projeto geral de um produto, bem como o projeto eletromecânico, por assim se dizer, apresentado neste trabalho também é apenas uma parte do projeto geral. Num projeto geral de produto, são levadas em conta outras análises mecânicas, como de esforços, que fogem ao escopo deste trabalho, bem como são levadas em conta análises operacionais e de processos, listas de compras, *design* visual, dentre outros.

Porém, como o objetivo deste trabalho era mostrar que não é necessário ao engenheiro eletricitista e eletrônico um conhecimento muito grande para operar minimamente *softwares* MCAD em conjunto com os ECAD e já incrementar a qualidade dos seus projetos, o apresentado foi satisfatório. Mostrou-se que utilizando modelos 3D para realização da verificação e adequação espacial dos componentes do projeto, e para realização de simulações

térmicas, é possível incrementar consideravelmente a qualidade e confiabilidade de um projeto, sem muito custo.

É notável que há uma certa curva de aprendizado referente aos *softwares* MCAD, não são difíceis de manipular, mas também não são triviais. Mas, isso é uma característica de toda ferramenta, bem como os *softwares* ECAD com os quais os engenheiros eletricitas e eletrônicos estão mais acostumados. Mas, com o grande acervo disponibilizado muitas vezes pelos próprios fabricantes, é possível aprender a utilizar estes *softwares* de forma a melhorar o desempenho do projetista. Desse modo, como um dos objetivos deste trabalho era apresentar o método de interatividade entre os *softwares* e explicar como proceder para utilizá-lo, pode-se dizer que este objetivo foi alcançado de forma satisfatória.

5.2 Apresentação do Protótipo Funcional da Aplicação

Além da análise conceitual, foi feita a montagem de um protótipo funcional da aplicação, o apresentador multimídia para os pés. Nesta seção, são apresentados fotos do protótipo funcional, podendo ser comparadas com os modelos 3D desenvolvidos. A Figura 55 mostra uma visão superior do transmissor apresentador multimídia desligado e a Figura 56 mostra o receptor do protótipo funcional, com o cabo conectado. A Figura 57 mostra uma vista em destaque das laterais do transmissor, destacando o conector da fonte e o *switch* liga/desliga, elementos de interação com o usuário.

Figura 55 – Transmissor do protótipo funcional.



Fonte: Próprio autor

Figura 56 – Receptor do protótipo funcional.



Fonte: Próprio autor

Figura 57 – Vista em destaque das laterais do transmissor.



Fonte: Próprio autor

A Figura 58 à Figura 61 a seguir mostram detalhes do funcionamento e manuseio do transmissor. A Figura 58, mostra em detalhe a troca de bateria, que é feita desparafusando a tampa localizada na parte inferior e removendo com a tampa o suporte de bateria, parafusado nesta. A Figura 59 mostra o LED RGB central aceso em azul, indicando que está esperando conexão do *bluetooth*. A Figura 60 e Figura 61 mostram em detalhe o funcionamento do LED RGB central indicando a carga e dos LEDs verdes correspondentes a cada botão acendendo quando o botão é apertado. A partir dessas Figuras, é possível verificar o funcionamento básico do transmissor.

Figura 58 – Detalhe da troca de bateria.



Fonte: Próprio autor

Figura 59 – Luz azul acesa esperando conexão.



Fonte: Próprio autor

Figura 60 – RGB central verde indicando a carga e LED direito aceso indicando o botão direito apertado.



Fonte: Próprio autor

Figura 61 – RGB central verde indicando a carga e LED esquerdo aceso indicando o botão esquerdo apertado.



Fonte: Próprio autor

5.3 Discussão e Análise dos Resultados Obtidos

Com os resultados obtidos, tanto teóricos, na análise dos *softwares* e do método, quanto práticos, na construção do protótipo funcional e comparação desse com o modelo 3D construído, é possível verificar que os resultados foram satisfatórios. É possível ver que o protótipo funcional está de acordo com os modelos feitos, tanto mecânicos quanto eletrônicos, então os resultados esperados foram alcançados. Este trabalho mostra a importância deste método no desenvolvimento de produtos, que possuem partes eletrônicas e mecânicas que devem funcionar adequadamente e em conjunto, a fim de serem satisfatórias para o usuário, tornando o produto útil, e mesmo, bom.

Um detalhe perceptível é que, sem o projeto mecânico prévio e a análise dos modelos 3D, o projeto eletrônico seria apenas uma placa de circuito impresso, e talvez os componentes estivessem soldados diretamente nesta, dificultando a utilização pelo usuário e mesmo podendo pôr em risco o bem estar do usuário, dependendo dos níveis de tensão e corrente do projeto. Além disso, existem diversos detalhes que só são perceptíveis após a criação de um modelo 3D e que, sem este, só serão perceptíveis após a criação do protótipo, com o caso de este não funcionar adequadamente.

Ou seja, a utilização do método de interatividade MCAD e ECAD evita retrabalho, ao inserir os modelos 3D no projeto, com os quais é possível detectar e solucionar alguns tipos de erros previamente a construção do protótipo. Outro ponto importante que foi discutido neste trabalho é a questão da simulação térmica, uma ferramenta bastante importante ao projetista eletrônico, pois apresenta melhores aproximações do que apenas os cálculos baseados em modelos de resistências térmicas, bem como sendo possível visualizar o fluxo de calor, e saber onde há mais acúmulo deste e, por isso, é necessária a dissipação, a fim de não danificar os componentes.

Quanto ao protótipo funcional construído, também vale salientar a importância da facilidade de utilização pelo usuário, algo que pode ser otimizado com o uso dos modelos 3D. Por exemplo, através do uso de um *software* MCAD, é possível projetar e testar uma situação como a troca da bateria. E, conforme a Figura 58, é possível confirmar que esta situação funcionou na prática conforme o esperado pelo projeto. Dentre outros detalhes importantes, também vale salientar projetar e testar posições de botões e conectores, no caso da aplicação, o apresentador multimídia controlado com os pés, como se utiliza os pés, com os quais se tem menos precisão do que a mão para apertar botões. Desse modo, os botões não poderiam ser muito próximos, caso contrário correria o risco de os dois serem apertados acidentalmente ao apertar apenas um deles.

5.4 Trabalhos Futuros

Para trabalhos futuros, é possível a incrementação do método, adicionando outros *softwares* ECAD para simulações dos circuitos eletrônicos, como por exemplo o PSIM ou OrCAD. Além de validar o funcionamento do circuito, outra vantagem da inclusão do uso desses *softwares* ao método é a possibilidade de simulação das potências dissipadas pelos componentes eletrônicos, sendo possível com esses dados, realizar uma simulação térmica mais fiel do equipamento no *software* MCAD. Além disso, também é possível a inclusão de demais

simulações mecânicas, como simulações de esforços, no caso de equipamentos eletrônicos com partes móveis, a fim de verificar a durabilidade e resistência do equipamento a esforços mecânicos.

CONCLUSÃO GERAL

Neste trabalho foi possível verificar a importância do uso da interatividade entre *softwares* MCAD e ECAD para desenvolvimento de equipamentos eletrônicos. Foi possível perceber a importância do entendimento de que um equipamento desses não consiste apenas do projeto eletrônico, mas sim de um conjunto de projetos menores, dos quais neste trabalho destacou-se o eletrônico e o mecânico, quanto a parte de análise de interferências dimensionais e simulações térmicas. Na verdade, percebeu-se que além disso, existem ainda mais outros projetos que, embora fujam ao escopo deste trabalho em específico, são necessários para um bom projeto geral de um equipamento eletrônico. Dentre estes, caso o equipamento possua peças móveis, vale destacar os estudos e simulações de esforços mecânicos.

Porém, é percebida através deste trabalho a diferença que o uso de um *software* MCAD, mesmo que apenas para as duas análises citadas, acarreta no desenvolvimento de um equipamento eletrônico, quando usado em conjunto de um *software* ECAD. Essa diferença foi percebida tanto ao longo do desenvolvimento do trabalho quanto durante a obtenção dos resultados. Há várias situações que justificam o uso da interatividade MCAD e ECAD, conforme já discutido nas seções anteriores. Desse modo, pode-se dizer que os objetivos esperados com o desenvolvimento deste trabalho foram alcançados, sendo o trabalho satisfatório.

Além disso, nas seções anteriores, também foi bem descrito e explicado o método de interatividade entre *softwares* MCAD e ECAD. Durante esta descrição, foi possível verificar que, embora haja uma curva de aprendizado dos *softwares*, não é algo muito difícil, e com a prática, o projetista se acostuma com o método e pode perceber a diferença que este traz. Quanto a isto, é possível frisar que, os pontos positivos obtidos com a utilização do método compensam os possíveis pontos negativos relacionados à curva de aprendizado e à necessidade de utilização de dois *softwares*, em vez de apenas um.

REFERÊNCIAS

Autodesk: sobre nós / LinkedIn. Disponível em: < <https://www.linkedin.com/company/autodesk/about/>>. Acesso em: 15 de setembro de 2019.

BIRNBAUM, Mark. **Essential electronic design automation (EDA)**. Prentice Hall Professional, 2004.

BELTRAN JR, Angelo A. et al. Teaching Electronic Circuits Using Altium Designer. **International Journal of Scientific Engineering and Technology**, v. 3, n. 10, p. 1239-1243, 2014.

Bluetooth Module HC05. Disponível em: < https://www.fecegypt.com/uploads/dataSheet/1522241762_bluetooth%20module%20hc05.pdf >. Acesso em 18 de novembro de 2019.

BLUETOOTH, S. I. G. *Bluetooth technology*. 2018.

BUDYNAS, Richard G.; NISBETH, J. Keith. **Elementos de Máquinas de Shigley-10ª Edição**. McGraw Hill Brasil, 2016.

BRANSON, Jerry; NABER, John; EDELEN, Glenn. *A simplistic printed circuit board fabrication process for course projects*. **IEEE transactions on education**, v. 43, n. 3, p. 257-261, 2000.

CHEN, Kenway; SCHAEFER, Dirk. MCAD-ECAD integration: overview and future research perspectives. In: **ASME 2007 International Mechanical Engineering Congress and Exposition**. American Society of Mechanical Engineers, 2007. p. 123-132.

COSTA, Rômulo I. F., *Apostila De Interatividade Entre Softwares MCAD E ECAD – Uso Do Método Com Altium Designer E Autodesk Inventor*. Disponível em <<http://www.peteletrica.ufc.br/wp-content/uploads/2019/12/Interatividade-MCAD-e-ECAD-com-Altium-Designer-e-Autodesk-Inventor.pdf>>. Acesso em 12/12/2019.

DALVEY, Jodi A.; NASSER, Nabill F. **Image transfer sheet**. U.S. Patent n. 6,551,692, 22 abr. 2003.

DE OLIVEIRA, José Raimundo; SHIN-TING, Wu. **Projeto de Sistemas Embutidos**. 2006

JANSEN, Dirk (Ed.). **The electronic design automation handbook**. Springer Science & Business Media, 2010.

Gerando Gerber no Altium Designer. Disponível em: <<https://www.micropress.com.br/wp-content/uploads/2015/12/gerando-gerber-no-altium-designer.pdf>> Acesso em: 2 de novembro de 2019.

GIL, Felipe de Souza Lourenço. **Análise da qualidade de malha de elementos finitos e validação de situação de caso real de viga biapoiada**. 2015. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

MOLINA, Paulo SC; PETERSON, Carlos AC. Competição de projetos no ensino de engenharia. In: **CONGRESSO BRASILEIRO DE ENSINO EM ENGENHARIA, Passo Fundo, ABENGE/UPF**. 2006.

MOUTA, Sofia Malheiro da Silva. **PCBA external workbench evaluation procedure**. 2015. Tese de Doutorado.

MAN, Lucian; PITICA, Dan; ZOLOG, Monica. Electrical/mechanical/thermal design integration. In: **2009 15th International Symposium for Design and Technology of Electronics Packages (SIITME)**. IEEE, 2009. p. 161-165.

NORTON, Robert L. **Projeto de máquinas 4ª Edição**. Bookman editora, 2013.

OKUNO, Helder Yukio; GUEDES, Gustavo Paiva. SliClick-Em Direção à Criação de um Apresentador de Slides Vestível. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2017. p. 1280.

Online Materials Information Resouce - MatWeb. Disponível em: <<http://www.matweb.com/index.aspx>> Acesso em: 16 de novembro de 2019.

SIRKIN, David; MARTELARO, Nikolas; JU, Wendy. *Make This! Introduction to Electronics Prototyping Using Arduino*. In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019. p. C11.

SON, Seokbae et al. Collaborative design environment between ECAD and MCAD engineers in high-tech products development. **International Journal of Production Research**, v. 52, n. 20, p. 6161-6174, 2014.

TAVARES, João Manuel Ribeiro Silva; PADILHA, Armando Jorge Monteiro Neves. *Introdução ao método dos elementos finitos*. 1998.

The LED Datasheet | All About LEDs | Adafruit Learning System Disponível em: <<https://learn.adafruit.com/all-about-leds/the-led-datasheet/>>. Acesso em: 1 de novembro de 2019.

ULLMAN, David G.; WOOD, Stephen; CRAIG, David. *The importance of drawing in the mechanical design process*. **Computers & graphics**, v. 14, n. 2, p. 263-274, 1990.

VERMA, Gaurav. **Basics of Autodesk Nastran In-CAD 2018**. CAD/CAM/CAE Works, 2018.

WAGUESPACK, Curtis. **Mastering Autodesk Inventor 2014 and Autodesk Inventor LT 2014: Autodesk Official Press**. John Wiley & Sons, 2013.

Wireless USB Presenter with Laser Pointer (Black). Disponível em <<https://www.targus.com/ap/amp16ap-wireless-presenter-with-laser-pointer>>. Acesso em 12/12/2019.

WISE, Christopher F.; ENGINEER, Departmental. *Fabrication of Printed Circuit Boards Using a Table Top CNC Mill*. **the Technology Interface/Fall 2007 Wise**, 2007.

XIAO-QING, Yang; JIA-XING, Wei. Mechanical creation design based on 3D CAD. In: **2009 IEEE 10th International Conference on Computer-Aided Industrial Design&Conceptual Design**.

ANEXO A – CÓDIGO DO RECEPTOR

```

////////////////////////////////////
//          Receptor Bluetooth do Apresentador Multimídia          //
//          Criadores: Marcelo Martins e Rômulo Iorio              //
//          Modificado em: 11/11/2019                               //
////////////////////////////////////
//          Endereço do HC05 Slave 1: 18:e4:35197d                //
//          Slave 2: 19:06:34e849                                  //
//          Slave 3: 19:06:34ed75                                  //
////////////////////////////////////
//          Bibliotecas                                           //
////////////////////////////////////
#include
<Keyboard.h> //Biblioteca
para controle do teclado
////////////////////////////////////
//          Definição de Constantes                                //
////////////////////////////////////
#define      keyboardLeft      KEY_LEFT_ARROW
#define      keyboardRight     KEY_RIGHT_ARROW
////////////////////////////////////
//          Prototipação de Funções                                //
////////////////////////////////////
void right();
void left();
////////////////////////////////////
//          Criação de Flags                                       //
////////////////////////////////////
bool active = 0;
////////////////////////////////////
//          Criação de Objetos e Variáveis                          //
////////////////////////////////////
int c;
String recep = " ";
////////////////////////////////////
//          Setup: Definições Iniciais                              //
////////////////////////////////////
void setup() {
    Keyboard.begin();
    Serial1.begin(38400);

```

```

    Serial1.setTimeout(0.1);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                               Loop: Código Principal                               //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void loop() {
    //IN = " ";
    c = 0;
    if(Serial1.available() > 0){
        recep = Serial1.readString();
    }
    if(recep[0] == 'L'&&!active) {
        left();
        active = 1;
        Serial.flush();
    }else if(recep[0] == 'R'&&!active){
        right();
        active = 1;
        Serial.flush();
    }else if(recep[0] == '0'&&!active){
        active = 0;
        Serial.flush();
    }
}
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                               Funções                                           //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void left() {
    // new document:
    // Keyboard.releaseAll();
    Keyboard.press(keyboardLeft);
    //Keyboard.press('216');
    delay(100);
    Keyboard.releaseAll();
    delay(100);
}
void right() {
    // new document:
    Keyboard.press(keyboardRight);
    //Keyboard.press('216');
}

```

```
    delay(100);  
    Keyboard.releaseAll();  
    delay(100);  
}
```

ANEXO B – CÓDIGO DO TRANSMISSOR

```

////////////////////////////////////
//          Transmissor Bluetooth do Apresentador Multimídia          //
//          Criadores: Marcelo Martins e Rômulo Iorio                //
//          Modificado em: 11/11/2019                                //
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
//          Bibliotecas                                             //
////////////////////////////////////
#include <SoftwareSerial.h> //
Inclui Biblioteca SoftwareSerial para uso de comunicação Serial em portas
diferentes das RX e TX padrão
////////////////////////////////////
//          Definição de Pinos                                       //
////////////////////////////////////
#define      pinoTensaoFonte          A0
#define      pinoTensaoBateria        A1
#define      ledR                      3
#define      ledG                      5
#define      ledB                      6
#define      botLeft                   2
#define      botRight                  4
#define      ledLeft                   7
#define      ledRight                  8
#define      bluetoothState           9
#define      RX                        10
#define      TX                        11
////////////////////////////////////
//          Definição de Constantes                                   //
////////////////////////////////////
#define      debounceDelay             40 //Tempo
em milissegundos para esperar a estabilização do botão, 40ms
#define      tensaoMaximaBateria       9.0 //Os
valores de tensão máxima e mínima da bateria são estimados para segurança e
devem ser
#define      tensaoMinimaBateria       4.0 //alter
ados de acordo com testes feitos com a bateria,
////////////////////////////////////
//          Prototipação de Funções                                   //
////////////////////////////////////

```

```

float      medirTensaoBateria ();
float      medirTensaoFonte ();
void       testLeft ();
void       testRight ();
void       monitorarCarga ();
void       mostrarCarga ();
void       pedal ();

/////////////////////////////////////////////////////////////////
//                               Criação de Flags                               //
/////////////////////////////////////////////////////////////////

bool       fonteConectada      = 0,      bateriaConectada   = 0;
bool       bluetoothConectado  = 0;
bool       flagLeft            = 0,      flagRight          = 0;

/////////////////////////////////////////////////////////////////
//                               Criação de Objetos e Variáveis                               //
/////////////////////////////////////////////////////////////////

SoftwareSerial HC05 (RX, TX); // RX pino 11 , TX pino 10
short       contBateria        = 0,      contFonte          = 0;
float       tensaoBateria      = 0,      tensaoInstantaneaBateria = 0;
float       tensaoFonte        = 0,      tensaoInstantaneaFonte = 0;
unsigned long tAnteriorPrint   = 0,      tAnteriorFonte     = 0;
unsigned long tAnteriorBateria = 0,      tAnteriorBluetooth = 0;

/////////////////////////////////////////////////////////////////
//                               Setup: Definições Iniciais                               //
/////////////////////////////////////////////////////////////////

void setup()
{
    pinMode (ledG           , OUTPUT);
    pinMode (ledR           , OUTPUT);
    pinMode (ledB           , OUTPUT);
    pinMode (ledLeft        , OUTPUT);
    pinMode (ledRight       , OUTPUT);
    pinMode (botLeft        , INPUT);
    pinMode (botRight       , INPUT);
    pinMode (bluetoothState , INPUT);
    digitalWrite (ledG      , HIGH);
    digitalWrite (ledR      , HIGH);
    digitalWrite (ledB      , HIGH);
    digitalWrite (ledLeft   , HIGH);
    digitalWrite (ledRight  , HIGH);
    analogReference (INTERNAL);

```



```

void pedal(short led1, short led2){
    if (!flagRight&&!flagLeft){ // Se
os dois botões não estiverem pressionados
        HC05.print('0'); // HC05
escreve o caractere na serial
        digitalWrite(led1, HIGH);
        digitalWrite(led2, HIGH);
    }else{
        if (flagLeft){ // Se o botão estiver pressionado
            HC05.print('L'); // HC05
escreve o caractere na serial
            flagLeft = 0;
            digitalWrite(led1, LOW);
        }else{
            digitalWrite(led1, HIGH);
        }
        if (flagRight){ // Se o
botão estiver pressionado
            HC05.print('R'); // HC05
escreve o caractere na serial
            flagRight = 0;
            digitalWrite(led2, LOW);
        }else{
            digitalWrite(led2, HIGH);
        }
    }
}

void testLeft(){
    flagLeft = 1;
}

void testRight(){
    flagRight = 1;
}

float medirTensaoBateria(short pinoTensaoBateria, float
tensaoMediaBateria){
    if((micros() - tAnteriorBateria >= 3000) && (contBateria < 300)){
        tAnteriorBateria = micros();
    }
}

```

```

    tensaoInstantaneaBateria    +=
analogRead(pinoTensaoBateria)*1.1/1024.0;
    contBateria++;

    //Armazena a soma de valores medidos de tensão pelo
sen
}else if(contBateria >= 300){
    contBateria = 0;
    tensaoInstantaneaBateria    /= 300;
    //Ao alcançar as 1000 iterações, tira uma média do
valor medido e converte o valor de tensão medido,
    tensaoMediaBateria          =  tensaoInstantaneaBateria*10/1.1*1.610
328638;
    tensaoInstantaneaBateria    =  0;
}
return tensaoMediaBateria;
}

float medirTensaoFonte(short pinoTensaoFonte, float tensaoMediaFonte){
    if((millis() - tAnteriorFonte >= 1) && (contFonte < 300)){
        tAnteriorFonte          =  millis();
        if(fonteConectada&&bateriaConectada){
            tensaoInstantaneaFonte    +=
analogRead(pinoTensaoFonte)*1.07142857/1024.0*1.0715106820;           //1,0
700116686114352392065344224037
        }else if(fonteConectada&&!bateriaConectada){
            tensaoInstantaneaFonte    +=
analogRead(pinoTensaoFonte)*1.07142857/1024.0*1.0505832759;
        }else{
            tensaoInstantaneaFonte    +=
analogRead(pinoTensaoFonte)*1.07142857/1024.0*1.0647307925;
        }
        contFonte++;

        //Armazena a soma de valores medidos de tensão pelo
sen
    }else if(contFonte >= 300){
        contFonte = 0;
        tensaoInstantaneaFonte    /= 300;
        //Ao alcançar as 1000 iterações, tira uma média do
valor medido e converte o valor de tensão medido,
        tensaoMediaFonte          =  tensaoInstantaneaFonte*10/1.07142857;
    }
}

```

```

    tensaoInstantaneaFonte      = 0;
}
return tensaoMediaFonte;
}

void mostrarCarga(short verde, short vermelho, float tensao){
    if(tensao
    >= 8){
        //Carga boa
        analogWrite(verde, (10 - tensao)*127/2);
        analogWrite(vermelho, 255);
    }else if (tensao >= 6){
        analogWrite(verde, 127 + (8 - tensao)*127/2);
        analogWrite(vermelho, 255 - (8
- tensao)*127/2);
        //Carga média
    }else if (tensao >= 4){
        analogWrite(vermelho, (6
- tensao)*127/2);
        //Carga ruim
        analogWrite(verde, 255);
    }else{
        digitalWrite(verde , HIGH);
        digitalWrite(vermelho , HIGH);
    }
}
}

```