

Maicon Augusto Tibola

**ENSINO DE PROGRAMAÇÃO E ELETRÔNICA
POR MEIO DA CONSTRUÇÃO DE
INSTRUMENTOS MUSICAIS DIGITAIS DE
BAIXO CUSTO**

Curitiba - PR, Brasil

2020

Maicon Augusto Tibola

**ENSINO DE PROGRAMAÇÃO E ELETRÔNICA POR
MEIO DA CONSTRUÇÃO DE INSTRUMENTOS
MUSICAIS DIGITAIS DE BAIXO CUSTO**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Pontifícia Universidade Católica do Paraná - PUCPR

Programa de Pós-Graduação em Informática - PPGIa

Orientador: Carlos Nascimento Silla Junior

Coorientador: Dilmeire Sant'Anna Ramos Vosgerau

Curitiba - PR, Brasil

2020

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Luci Eduarda Wielganczuk – CRB 9/1118

T554e
2024 Tibola, Maicon Augusto
Ensino de programação e eletrônica por meio da construção de instrumentos musicais digitais de baixo custo / Maicon Augusto Tibola ; orientador: Carlos Nascimento Silla Junior ; coorientador: Dilmeire Sant'Anna Ramos Vosgerau. – 2024.
241 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2024
Bibliografia: f. 203-219

1. Informática. 2. Programação (Computadores) – Estudo e ensino.
3. Eletrônica – Estudo e ensino. 4. Instrumentos musicais digitais. I. Silla Junior, Carlos Nascimento. II. Vosgerau, Dilmeire Sant'Anna Ramos. III. Pontifícia Universidade Católica do Pós-Graduação em Informática. IV. Título.

CDD. 20. ed. – 004

Curitiba, 25 de outubro de 2024.


88-2024

DECLARAÇÃO

Declaro para os devidos fins, que **Maicon Augusto Tibola** defendeu a dissertação de Mestrado intitulada “**ENSINO DE PROGRAMAÇÃO E ELETRÔNICA POR MEIO DA CONSTRUÇÃO DE INSTRUMENTOS MUSICAIS DIGITAIS DE BAIXO CUSTO**”, na área de concentração Ciência da Computação no dia 29 de janeiro de 2020, no qual foi aprovado.

Declaro ainda, que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade firmo a presente declaração.

Documento assinado digitalmente
 **EMERSON CABRERA PARAISO**
Data: 28/10/2024 09:37:00-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Emerson Cabrera Paraiso
Coordenador do Programa de Pós-Graduação em Informática

Agradecimentos

Agradeço a Deus, pela força e sabedoria concedidas ao longo deste percurso. Agradeço ao meu orientador Carlos Nascimento Silla Junior pelo suporte e incentivos no desenvolvimento deste trabalho de pesquisa. Agradeço também minha coorientadora Dilmeire Sant'Anna Ramos Vosgerau pelas contribuições durante todo o projeto.

Aos meus pais Milton Tibola e Silvana Aparecida de Souza Tibola que me incentivam e apoiam todas minhas decisões. Aos meus irmãos Maiara Carine Tibola e Marlon Cesar Tibola por toda atenção e apoio.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Resumo

Contexto: Este trabalho apresenta uma abordagem de ensino integrado de programação, música e eletrônica por meio da construção de instrumentos musicais digitais (DMIs) de baixo custo. Essa abordagem foi aplicada a quatro participantes em formação como educadores musicais. Há uma preocupação sobre como atrair e motivar os alunos para as áreas de ciência, tecnologia, engenharia e matemática (STEM) devido às oportunidades e habilidades que podem ser adquiridas. A maioria dos trabalhos na literatura utilizam a robótica com foco nas áreas STEM. No entanto, há uma tendência de incluir o A de Artes, com novas maneiras de envolver os alunos, formando o STEAM. **Motivação:** As abordagens com DMIs comerciais têm obtido resultados positivos no ensino de programação, porém seu alto custo representa uma barreira na aplicação dessa abordagem, especialmente para escolas de baixa renda. **Objetivos:** Verificar a viabilidade de construir DMIs de baixo custo que possam ser usados para fins didáticos. Desenvolver e executar um curso que integre a construção de DMIs de baixo custo e o ensino de programação e eletrônica. **Metodologia:** Para alcançar os objetivos, enfrentaram-se dois desafios principais. O primeiro foi projetar, avaliar e selecionar DMIs que pudessem ser usados para ensinar programação e eletrônica e que também sejam de baixo custo. O segundo, desenvolver e avaliar um programa de formação para professores utilizando essas tecnologias de forma integrada. **Conclusões:** Foram construídos e analisados diversos DMIs de baixo custo e selecionados quatro considerados adequados para o uso no ensino de programação e eletrônica. Além disso, foi desenvolvido e aplicado um programa de formação de professores que integrou os DMIs selecionados, sendo avaliado por quatro participantes sem formação tecnológica. Os resultados indicam que o programa foi eficaz para ensinar programação e eletrônica, capacitando participantes sem formação em Ciências da Computação ou Engenharia. Os participantes não só foram capazes de construir seus próprios DMIs de baixo custo, mas também adquiriram conhecimentos nos conteúdos de programação e eletrônica.

Palavras-chave: Instrumentos Musicais Digitais, DMI de baixo custo, Ensino de Programação, Ensino de Eletrônica, STEAM Education.

Abstract

Background: This work presents an integrated teaching approach that combines programming, music, and electronics through the construction of low-cost digital musical instruments (DMIs). This approach was applied to four participants training as music educators. There is a concern with how to attract and motivate students toward science, technology, engineering, and mathematics (STEM) fields due to the opportunities and skills that can be acquired. Most studies in the literature use robotics with a focus on STEM areas. However, there is a trend to include the "A" for Arts, introducing new ways to engage students and forming the STEAM approach. **Motivation:** Approaches using commercial DMIs have shown positive results in teaching programming, however, their high cost presents a barrier to applying this approach, especially in low-income schools. **Objectives:** To verify the feasibility of building low-cost DMIs that can be used for educational purposes. To develop and deliver a course that integrates low-cost DMI construction with the teaching of programming and electronics. **Method:** To achieve the objectives, two main challenges were faced. The first was to design, evaluate, and select low-cost DMIs that could be used to teach programming and electronics. The second was to develop and evaluate a teacher's training program that integrates these technologies. **Conclusions:** Several low-cost DMIs were built and analyzed, and four were selected as suitable for use in teaching programming and electronics. Additionally, a teacher's training program was developed and implemented, integrating the selected DMIs, and it was evaluated by four participants with no prior technological training. The results indicate that the training program was effective in teaching programming and electronics, equipping participants without a background in Computer Science or Engineering. The participants were not only able to build their own low-cost DMIs but also acquired knowledge in programming and electronics content.

Keywords: Digital Music Instruments, Low Cost DMI, Programming Teaching, Electronics Teaching, STEAM Education.

Lista de ilustrações

Figura 1 – The Extended Clarinet	32
Figura 2 – Seaboard	33
Figura 3 – TED - Tangible Electrophonic	33
Figura 4 – Neurohedron	34
Figura 5 – Representação em blocos de um DMI	34
Figura 6 – Striso	36
Figura 7 – The Ghost	36
Figura 8 – Histórico da Robótica Pedagógica	38
Figura 9 – Placa Arduino UNO	39
Figura 10 – Botão ou Push-Button	40
Figura 11 – Potenciômetro	40
Figura 12 – Sensor Piezoelétrico	41
Figura 13 – Etapas da Pesquisa Desenvolvimento de Objeto	44
Figura 14 – Etapas da Pesquisa Desenvolvimento de Objeto Adaptada	45
Figura 15 – Etapas da Conceptualização do Objeto 1	56
Figura 16 – Circuito Teremim na Protoboard	57
Figura 17 – Circuito Teclado/Xilofone/Bateria de Botão	58
Figura 18 – Circuito Teclado/Xilofone/Bateria com Piezoelétricos	58
Figura 19 – Circuito Teclado/Xilofone com Sensores Capacitivos	59
Figura 20 – Circuito Harpa Laser na Protoboard	60
Figura 21 – Circuito Controlador MIDI na Protoboard	61
Figura 22 – Circuito Teclado com Cartão SD na Protoboard	62
Figura 23 – Circuito Xilofone e Bateria com Cartão SD na Protoboard	62
Figura 24 – Teclado Yamaha PSR-EW410	63
Figura 25 – Organização das Notas em um Teclado	64
Figura 26 – Bateria Convencional com 8 partes	65
Figura 27 – Bateria Digital com 10 pads e módulo.	66
Figura 28 – Xilofone Simples	67
Figura 29 – Xilofone mais complexo	67
Figura 30 – Xilofone Digital	67
Figura 31 – Harpa	68
Figura 32 – Harpa Laser	68
Figura 33 – Teremim – Moog Music Etherwave Plus Theremin	69
Figura 34 – Diagrama de configuração do DMI utilizando PWM	72
Figura 35 – Diagrama de configuração do DMI utilizando Cartão SD e PWM	72
Figura 36 – Estratégia para construção de DMIs de baixo custo	73

Figura 37 – Circuito Teclado de botões	74
Figura 38 – Alto falante utilizado no Teclado de botões	74
Figura 39 – Circuito Xilofone de Botões	75
Figura 40 – Cartão SD utilizado no Xilofone de botões	76
Figura 41 – Circuito Xilofone com Piezo	76
Figura 42 – Teclado com piezoelétricos e PVC	77
Figura 43 – Teclado capacitivo com isopor	77
Figura 44 – LoopMIDI configurado com a porta virtual ArduinoPort	78
Figura 45 – Hairless MIDI Serial Bridge conectado à porta MIDI virtual	79
Figura 46 – Virtual MIDI Piano Keyboard	79
Figura 47 – Jumper da Placa Arduino	80
Figura 48 – FreePiano utilizado para reproduzir os comandos MIDI	81
Figura 49 – Bateria digital utilizando tampas de plástico	81
Figura 50 – Xilofone digital utilizando papelão	82
Figura 51 – Circuito Bateria/Xilofone Digital	82
Figura 52 – Xilofone com piezoelétrico e palitos de madeira	83
Figura 53 – Visão do circuito xilofone com piezoelétrico e palitos de madeira	83
Figura 54 – Bateria com tampas de plástico e espumas	84
Figura 55 – Teremim Digital em protoboard	85
Figura 56 – Circuito Teremim Digital	85
Figura 57 – Harpa Laser teste 1	86
Figura 58 – Circuito Harpa Laser	87
Figura 59 – Quadro da Harpa Laser	88
Figura 60 – LDR fixados no Quadro	88
Figura 61 – Lasers fixados no Quadro	88
Figura 62 – Placa com circuito da Harpa Laser	88
Figura 63 – Circuito Xilofone Capacitivo	89
Figura 64 – Circuito Teremim Capacitivo	90
Figura 65 – Controlador MIDI	91
Figura 66 – Placa do circuito Controlador MIDI	91
Figura 67 – Modelo para aulas da etapa de Formação	101
Figura 68 – Circuito Teclado digital com Botões	112
Figura 69 – Circuito Xilofone e Bateria digital com piezoelétrico	112
Figura 70 – Circuito Teclado digital capacitivo	113
Figura 71 – Circuito Teremim Capacitivo	113
Figura 72 – Placa Arduino com número identificador dos componentes	118
Figura 73 – Circuito 1 - DMI 1 – Estudante E1	143
Figura 74 – Circuito 1 - DMI 1 – Estudante E2	143
Figura 75 – Circuito 1 - DMI 1 – Estudante E3	143

Figura 76 – Circuito 1 - DMI 1 – Estudante E4	143
Figura 77 – Circuito primeiro DMI aula 04	143
Figura 78 – Circuito 2 - Botões e LEDs – Estudante E1	150
Figura 79 – Circuito 2 - Botões e LEDs – Estudante E2	150
Figura 80 – Circuito 2 - Botões e LEDs – Estudante E3	150
Figura 81 – Circuito 2 - Botões e LEDs – Estudante E4	150
Figura 82 – Circuito 3 - Botões e LEDs – Estudante E1	156
Figura 83 – Circuito 3 - Botões e LEDs – Estudante E2	156
Figura 84 – Circuito 3 - Botões e LEDs – Estudante E3	156
Figura 85 – Circuito 4 - Piezo – Estudante E1	169
Figura 86 – Circuito 4 - Piezo – Estudante E2	169
Figura 87 – Circuito 4 - Piezo – Estudante E4	170

Lista de tabelas

Tabela 1 – Preços dos DMIs em Silla, Przybysz e Leal (2016) e Silla et al. (2018)	49
Tabela 2 – Análise do Contexto Objeto 1 - Construção de DMIs de Baixo Custo	51
Tabela 3 – Levantamento NIME 2010-2016	53
Tabela 4 – Sensores definidos para cada DMI	71
Tabela 5 – Custo para construção do teclado digital	92
Tabela 6 – Custo para construção da Bateria/Xilofone digital	92
Tabela 7 – Custo para construção do Teremim Digital	93
Tabela 8 – Custo para construção da Harpa Laser	93
Tabela 9 – Custo para construção do Xilofone Capacitivo	94
Tabela 10 – Custo para construção do Teremim Capacitivo	94
Tabela 11 – Custo para construção do Controlador MIDI	94
Tabela 12 – Análise do Contexto do Objeto 2 - Formação.	98
Tabela 13 – Plano de Ensino etapa de Formação	102
Tabela 14 – Respostas do Questionário de Inscrição	110
Tabela 15 – Estrutura da Aula 01	114
Tabela 16 – Perguntas do Kahoot 01	115
Tabela 17 – Resultados do Kahoot 01	117
Tabela 18 – Perguntas do Kahoot 02	118
Tabela 19 – Resultados do Kahoot 02	119
Tabela 20 – Estrutura da Aula 02	120
Tabela 21 – Perguntas do Kahoot 03	121
Tabela 22 – Resultados do Kahoot 03	122
Tabela 23 – Perguntas do Kahoot 04	123
Tabela 24 – Resultados do Kahoot 04	125
Tabela 25 – Estrutura da Aula 03	127
Tabela 26 – Perguntas do Kahoot 05	129
Tabela 27 – Resultados do Kahoot 05	132
Tabela 28 – Perguntas do Kahoot 06	133
Tabela 29 – Resultados do Kahoot 06	136
Tabela 30 – Estrutura da Aula 04	138
Tabela 31 – Perguntas do Kahoot 07	139
Tabela 32 – Resultados do Kahoot 07	141
Tabela 33 – Estrutura da Aula 05	144
Tabela 34 – Perguntas do Kahoot 08	147
Tabela 35 – Resultados do Kahoot 08	148
Tabela 36 – Estrutura da Aula 06	151

Tabela 37 – Perguntas do Kahoot 09	152
Tabela 38 – Resultados do Kahoot 09	155
Tabela 39 – Estrutura da Aula 07	157
Tabela 40 – Perguntas do Kahoot 10	158
Tabela 41 – Resultados do Kahoot 10	160
Tabela 42 – Perguntas do Kahoot 11	161
Tabela 43 – Resultados do Kahoot 11	162
Tabela 44 – Estrutura da Aula 08	163
Tabela 45 – Perguntas do Kahoot 12	165
Tabela 46 – Resultados do Kahoot 12	167
Tabela 47 – Perguntas do Kahoot 13	167
Tabela 48 – Resultados do Kahoot 13	169
Tabela 49 – Estrutura da Aula 09	171
Tabela 50 – Perguntas do Kahoot 14	172
Tabela 51 – Resultados do Kahoot 14	174
Tabela 52 – Perguntas do Kahoot 15	175
Tabela 53 – Resultados do Kahoot 15	188
Tabela 54 – Resultados do Kahoot 15 comparados aos outros Kahoots	190
Tabela 55 – Taxa de acerto dos Kahoots	191
Tabela 56 – Taxa de acerto dos Kahoots por categoria	191
Tabela 57 – Resultados Questionário sobre a percepção dos Estudantes	192

Lista de abreviaturas e siglas

AMI	Augmented Musical Instrument
DMI	Digital Musical Instrument
E-TEXTILE	Electronic Textile
IDE	Integrated Development Environment
IMA	Instrumento Musical Aumentado
IMD	Instrumento Musical Digital
Laser	Light Amplification by Stimulated Emission of Radiation
LDR	Light Dependent Resistor
LED	Light-Emitting Diode
MIDI	Musical Instrument Digital Interface
NIED	Núcleo de Informática Aplicada à Educação
NIME	New Interfaces for Musical Expression
NSF	National Science Foundation
OBR	Olimpíada Brasileira de Robótica
PVC	Policloreto de Vinila
PWM	Pulse Width Modulation
SMET	Science, Mathematics, Engineering and Technology
STEAM	Science, Technology, Engineering, Arts and Mathematics
STEM	Science, Technology, Engineering and Mathematics
STREAM	Science, Technology, Robotics, Engineering and Mathematics
TED	Tangible Electrophonic Drumstick
USB	Universal Serial Bus
WRE	Workshop de Robótica Pedagógica

Sumário

1	INTRODUÇÃO	23
1.1	Problematização	25
1.2	Justificativa	28
1.3	Objetivos	29
1.4	Estrutura do Trabalho	29
2	REFERENCIAL TEÓRICO	31
2.1	A Concepção de DMIs	31
2.2	O Ensino de Programação e Eletrônica	36
3	METODOLOGIA	43
3.1	Etapas da Pesquisa Desenvolvimento de Objeto	43
4	CONSTRUÇÃO DE DMIS DE BAIXO CUSTO (OBJETO 1)	47
4.1	Análise do Contexto do Objeto 1	47
4.2	Análise do Objeto 1	51
4.2.1	Conceptualização do Objeto 1	51
4.2.2	Modelagem do Objeto 1	57
4.3	Preparação do Objeto 1	63
4.4	Desenvolvimento do Objeto 1	74
4.4.1	Prototipação	74
4.4.2	Prototipação - Adaptações e Modificações	78
4.5	Análise de custo e funcionalidade dos protótipos	91
5	FORMAÇÃO (OBJETO 2)	97
5.1	Análise do Contexto do Objeto 2	97
5.2	Análise do Objeto 2	99
5.2.1	Conceptualização do Objeto 2	99
5.2.2	Modelagem do Objeto 2	101
5.3	Preparação do Objeto 2	101
5.4	Desenvolvimento do Objeto 2	109
5.5	Análises da Formação	114
5.5.1	Aula 01 - Introdução ao DMI e Arduino	114
5.5.1.1	Resultados de Aprendizagem da Aula 01.	114
5.5.1.2	Estrutura da Aula 01	114
5.5.1.3	Resultados dos Kahoots 01 e 02	115

5.5.1.3.1	Kahoot #01 – É um DMI?	115
5.5.1.3.2	Kahoot #02 – Placa Arduino	117
5.5.1.4	Observações realizadas durante a aula 01	119
5.5.1.5	Considerações sobre a aula	119
5.5.2	Aula 02 - Reproduzindo Notas Musicais	120
5.5.2.1	Resultados de Aprendizagem da Aula 02	120
5.5.2.2	Estrutura da Aula 02:	120
5.5.2.3	Resultados dos Kahoots 03 e 04	121
5.5.2.3.1	Kahoot #03 – Buzzer	121
5.5.2.3.2	Kahoot #04 – Código Buzzer	123
5.5.2.4	Observações realizadas durante a aula 02	125
5.5.2.5	Considerações sobre a aula	126
5.5.3	Aula 03 - Qualidade do Som	127
5.5.3.1	Resultados de Aprendizagem da Aula 03.	127
5.5.3.2	Estrutura da Aula 03.	127
5.5.3.3	Resultados dos Kahoots 05 e 06	129
5.5.3.3.1	Kahoot #05 – Jack J2 e protoboard.	129
5.5.3.3.2	Kahoot #06 – Potenciômetro	133
5.5.3.4	Observações realizadas durante a aula 03.	136
5.5.3.5	Considerações sobre a aula	137
5.5.4	Aula 04 - Botões	138
5.5.4.1	Resultados de Aprendizagem da Aula 04	138
5.5.4.2	Estrutura da Aula 04	138
5.5.4.3	Resultados do Kahoot 07	139
5.5.4.3.1	Kahoot #07 – Botões	139
5.5.4.4	Observações realizadas durante a aula 04	142
5.5.4.5	Considerações sobre a aula	142
5.5.5	Aula 05 - Comandos de Decisão	144
5.5.5.1	Resultados de Aprendizagem da Aula 05	144
5.5.5.2	Estrutura da Aula 05	144
5.5.5.3	Resultados do Kahoot 08	146
5.5.5.3.1	Kahoot #8 – Comandos de decisão	146
5.5.5.4	Observações realizadas durante a aula 05	149
5.5.5.5	Considerações sobre a aula	149
5.5.6	Aula 06 - Protocolo MIDI	150
5.5.6.1	Resultados de Aprendizagem da Aula 06.	150
5.5.6.2	Estrutura da Aula 06.	151
5.5.6.3	Resultados do Kahoot 09	152
5.5.6.3.1	Kahoot #09 – Protocolo MIDI	152

5.5.6.4	Observações realizadas durante a aula 06	155
5.5.6.5	Considerações sobre a aula	156
5.5.7	Aula 07 - Variáveis	157
5.5.7.1	Resultados de Aprendizagem da Aula 07.	157
5.5.7.2	Estrutura da Aula 07.	157
5.5.7.3	Resultados dos Kahoots 10 e 11	158
5.5.7.3.1	Kahoot #10 – Variáveis	158
5.5.7.3.2	Kahoot #11 – Funções	160
5.5.7.4	Observações realizadas durante a aula 07	162
5.5.7.5	Considerações sobre a aula	162
5.5.8	Aula 08 - Detectando Pressão	163
5.5.8.1	Resultados de Aprendizagem da Aula 08.	163
5.5.8.2	Estrutura da Aula 08.	163
5.5.8.3	Resultados dos Kahoots 12 e 13	165
5.5.8.3.1	Kahoot #12 – Piezo	165
5.5.8.3.2	Kahoot #13 – Função Map e Millis	167
5.5.8.4	Observações realizadas durante a aula 08	169
5.5.8.5	Considerações sobre a aula	170
5.5.9	Aula 09 - Sensor Capacitivo	170
5.5.9.1	Resultados de Aprendizagem da Aula 09.	170
5.5.9.2	Estrutura da Aula 09.	171
5.5.9.3	Resultados do Kahoot 14	171
5.5.9.3.1	Kahoot #14 – Sensor capacitivo	171
5.5.9.4	Observações realizadas durante a aula 09.	174
5.5.9.5	Considerações sobre a aula	174
5.5.10	Aula 10 - Finalização	175
5.5.10.1	Resultados do Kahoot 15	175
5.5.10.1.1	Kahoot #15 – Todo o conteúdo	175
5.5.10.2	Observações realizadas durante a aula 10	192
5.5.10.3	Considerações sobre a aula	198
6	CONCLUSÃO	201
	REFERÊNCIAS	205
	APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ES- CLARECIDO - TCLE	223
	APÊNDICE B – CÓDIGOS DOS DMIS CONSTRUÍDOS	227

1 Introdução

O ensino de programação proporciona aos estudantes o aprimoramento de habilidades como resolução de problemas, raciocínio lógico e criatividade (SAELI et al., 2011; SCAICO et al., 2013; ARIMOTO; OLIVEIRA, 2019). A utilidade destas habilidades tem aumentado a procura por cursos de introdução à programação, bem como a inclusão da disciplina de programação em currículos do ensino fundamental e médio (SCAICO et al., 2013; SERAFINI, 2011; KALELIOĞLU, 2015; MERKOURIS; CHORIANOPOULOS; KAMEAS, 2017a).

A programação pode auxiliar os estudantes de diversas maneiras, mas não significa que todos os estudantes se tornarão profissionais da área de programação necessariamente, mas sim que terão habilidades úteis para o seu crescimento pessoal e social (GARNELI; GIANNAKOS; CHORIANOPOULOS, 2015; ARIMOTO; CRUZ, 2020). Assim, não é necessário que o estudante inicie o aprendizado de programação por uma linguagem de programação específica (SHEIN, 2014).

Muitas vezes os estudantes consideram que aprender a programar é uma tarefa complexa devido aos diversos conceitos agregados (ARIMOTO; OLIVEIRA, 2019). Junto a isso, algumas abordagens para o ensino de programação ainda utilizam metodologias que revelam ser insuficientes para engajar os estudantes, o que causa desmotivação, reprovação e evasão (RAPKIEWICZ et al., 2006; GOMES; HENRIQUES; MENDES, 2008; SILVA; MELO; TEDESCO, 2018).

A motivação é um fator que afeta o desempenho dos estudantes gerando consequências no aprendizado, por isso é necessário compreender quais abordagens podem contribuir para promover e manter a motivação dos estudantes de programação (SANTANA; FIGUEIRÊDO; BITTENCOURT, 2017).

São encontradas na literatura diversas abordagens que utilizam ferramentas como o Alice¹ (WANG et al., 2009), Scratch² (KALELIOĞLU; GÜLBAHAR, 2014) e Greenfoot³ (BEGOSSO et al., 2012). Outras abordagens unem o aprendizado de programação com outros assuntos, tais como jogos (HERNANDEZ et al., 2010), matemática (FÖRSTER; FÖRSTER; LÖWE, 2018), física (LIN; WANG; WU, 2019), eletrônica (PECH; NOVÁK, 2020) e robótica (MERKOURIS; CHORIANOPOULOS; KAMEAS, 2017b).

Observa-se que o uso de ferramentas para o ensino de programação auxilia na motivação dos estudantes (KANIKA; CHAKRAVERTY; CHAKRABORTY, 2020). Diante

¹ <https://www.alice.org/>

² <https://scratch.mit.edu/>

³ <https://www.greenfoot.org/>

disso, a maioria dos trabalhos têm o foco em abordagens envolvendo disciplinas do *Science, Technology, Engineering and Mathematics* (STEM⁴), como aprendizado de geometria e física para construção de robôs (WATSON, A; WATSON, G, 2013). No entanto, deve ser considerado a integração do "A" de "Artes" ao STEM formando o *Science, Technology, Engineering, Arts and Mathematics* (STEAM⁵) (WATSON, A; WATSON, G, 2013). A inclusão das Artes pode contribuir auxiliando os professores no uso de tecnologias e encorajando criatividade dos estudantes (YAKMAN, 2008; CONRADTY; BOGNER, 2018; SULLIVAN; BERS, 2018).

As ferramentas voltadas para a área de música apresentam resultados positivos para o STEAM *education* como *Music Maker* (RAO, 2017), *EarSketch* (SIVA et al., 2018) e *Digital Musical Instrument* (DMI⁶) (SILLA et al., 2018; SILLA; PRZYBYSZ; LEAL, 2016). Os DMIs são dispositivos que podem ser considerados como sendo instrumentos musicais que utilizam sons gerados por computador (MIRANDA; WANDERLEY, 2006). Assim, os DMIs são dispositivos que podem ser utilizados em contextos educacionais em abordagens envolvendo programação e eletrônica (HARRIMAN, 2015b).

Apesar dos resultados positivos, os DMIs comerciais possuem um alto custo para serem utilizados no ensino de programação, o que dificulta a escalabilidade deste tipo de abordagem principalmente em lugares com baixo poder aquisitivo como as escolas públicas (SILLA et al., 2018; SILLA; PRZYBYSZ; LEAL, 2016). Além disso, é necessário desenvolver abordagens utilizando DMIs que possibilitem explorar os diferentes tipos de DMIs, quais softwares e hardwares são necessários, quais conceitos de programação, eletrônica e música podem ser utilizados (HARRIMAN, 2015b).

Neste sentido, o interesse do autor em desenvolver esta pesquisa iniciou a partir da busca por um estágio que envolvesse hardware. Com isso, ao final de 2016 em uma conversa sobre projetos envolvendo música, o professor Carlos Nascimento Silla Junior ofereceu a oportunidade para trabalhar com a construção de DMIs dentro das instalações da Pontifícia Universidade Católica do Paraná (PUCPR). Durante três meses foram construídas diversas versões de DMIs, a partir dos instrumentos Piano, Xilofone e Bateria, utilizando a placa Arduino e alguns sensores, o que serviu como base para ingressar no mestrado. Por estes motivos, esta pesquisa está relacionada aos projetos anteriores de Silla et al. (2018) e Silla, Przybysz e Leal (2016), porém, neste trabalho buscou-se associar o ensino de programação e eletrônica por meio da construção de DMIs de baixo custo. Assim, para analisar como a construção de DMIs de baixo custo pode auxiliar no aprendizado de programação e eletrônica foi realizada uma formação para estudantes de Licenciatura em Música que não possuíam conhecimentos prévios de programação e/ou eletrônica.

⁴ STEM em português: Ciência, Tecnologia, Engenharia e Matemática.

⁵ STEAM em português: Ciência, Tecnologia, Engenharia, Artes e Matemática.

⁶ DMI em português: Instrumento Musical Digital

1.1 Problematização

O ensino de programação e eletrônica é uma combinação que amplia o aprendizado dos estudantes, permitindo desenvolver habilidades de programação e ao mesmo tempo aplicar em situações do dia a dia (FIGUEIREDO; CIFREDO-CHACÓN; GONÇALVES, 2016). A eletrônica oferece um ambiente rico para o aprendizado, e assim como a programação, proporciona os benefícios de desenvolver habilidades como resolução de problemas, pensamento criativo e trabalho em equipe (BARAK, 2018).

Os conceitos de programação e eletrônica muitas vezes são apresentados por meio da robótica (BEZERRA JUNIOR; QUEIROZ; LIMA, 2018). No entanto, a maioria dos trabalhos envolvendo robótica focam em elementos da programação para a construção de robôs, deixando de lado a eletrônica como parte do ensino (KYNIGOS; GRIZIOTI; NIKITOPOULOU, 2017; GUPTA; TEJOVANTH; MURTHY, 2012). Apesar disso, a robótica proporciona resultados positivos em manter estudantes motivados e promover materiais para atividades práticas, uma vez que se tem aplicações da robótica nos contextos de ensino e aprendizado, referenciado como "Robótica Educacional" (JUNG; WON, 2018).

Ainda que a robótica educacional esteja direcionada para áreas do STEM alguns trabalhos se concentram mais na integração de elementos sociais, culturais ou estéticos, do que diretamente nos benefícios da robótica educacional (ANWAR et al., 2019). Contudo, o *STEM education* não significa robótica unicamente, a robótica é utilizada com o objetivo de atrair novos estudantes da educação básica para as áreas do STEM e melhorar o engajamento de estudantes no ensino superior (ANWAR et al., 2019). Neste contexto, torna-se difícil encontrar uma única definição para o *STEM education*, pois trata-se de um movimento multidisciplinar.

STEM é um acrônimo para se referir a *Science, Technology, Engineering and Mathematics* e foi utilizado nos anos 90 pela agência americana *National Science Foundation* (NSF). Inicialmente foi nomeado como SMET (*Science, Mathematics, Engineering and Technology*). Porém, no início dos anos 2000, foi alterado para STEM, popularizando-se como *STEM education* referindo-se ao ensino dessas grandes áreas (SANDERS, 2009).

A definição do que é *STEM education* não se encontra bem estabelecida na literatura. Há definições que se referem ao *STEM education* como sendo uma abordagem que é utilizada para aumentar a compreensão de como as coisas funcionam incluindo o uso de tecnologias, e não apenas apresentar matérias de matemática e ciência (BYBEE, 2010). Outros definem *STEM education* como uma abordagem que inclui todas as áreas do STEM como uma só (IOANNOU; BRATITSIS, 2016).

Contudo, são encontradas na literatura diversas estratégias que confirmam esta divergência de definição sobre o que é a *STEM education*. Algumas abordagens buscam integrar conteúdos de STEM em matérias escolares como Matemática (NICOLETE

et al., 2015; PETERS; RUECKERT; SERUGA, 2018; REAMER et al., 2015) e Física (BUNYAMIN; FINLEY, 2016), outras utilizam formas para atrair a atenção dos estudantes através de vídeo games (FONSECA et al., 2018) e robótica (BELTER et al., 2015; KARIM; LEMAIGNAN; MONDADA, 2016; KIM et al., 2015; MATARIĆ; KOENIG; FEIL-SEIFER, 2007).

Na robótica é onde se encontra a maior parte dos trabalhos com STEM (ANWAR et al., 2019). Em alguns casos acrescentam a robótica como parte do acrônimo STEM criando o STREAM (*Science, Technology, Robotics, Engineering and Mathematics*) (STUBBS; YANCO, 2009). Além disso, há o termo “robótica educacional” o qual pode ser considerado um assunto independente e, devido a multidisciplinaridade, ao mesmo tempo pode ser uma subdisciplina para os conceitos do STEM (JUNG; WON, 2018).

A robótica é também utilizada como ferramenta para aumentar a criatividade dos estudantes (ALIMISIS, 2013). No entanto, se tem dificuldades em mensurar a criatividade dentro do escopo STEM e por este motivo há uma tendência em integrar a criatividade tradicional (Artes), acrescentando o A de artes ao STEM, formando as áreas de *Science, Technology, Engineering, Arts and Mathematics* (STEAM) (CONRADTY; BOGNER, 2018; SULLIVAN; BERS, 2018).

O STEAM *education* é um modelo educacional que busca aproximar as áreas tradicionais do STEM acrescentando Artes ao currículo (YAKMAN, 2008). A arte e o design podem transformar a economia no século 21 da mesma maneira que as áreas do STEM fizeram no século passado (MAEDA, 2013).

O movimento STEAM vem se popularizando em escolas, com isso, diversas abordagens pedagógicas são usadas para integrar as Artes em aulas do STEM. Por conta disso, há diferentes interpretações para Artes no STEAM. A partir disso, é verificado que pode-se classificar Artes do STEAM em três categorias principais: Educação Artística; Artes como qualquer disciplina que não seja STEM; Artes como um sinônimo de aprendizado baseado em projetos, problemas, tecnologias ou que utilize construção de algo (PERIGNAT; KATZ-BUONINCONTRO, 2019).

Com isso, no movimento STEAM *education* são encontrados trabalhos envolvendo a área de Artes em diversos formatos, como a estilização de roupas com a Eletrônica chamados de E-Textiles (*Electronic textiles*) (PEPPLER, 2013; QIU et al., 2013), realizando peças de teatro com robótica (BARNES et al., 2017; KO et al., 2020) e outros utilizam a música como motivação para atrair estudantes para as áreas do STEM (BARNES et al., 2019; BARNES et al., 2017; MAGERKO et al., 2016; SULLIVAN; BERS, 2018).

Observa-se que a integração das Artes ao STEM promove aos estudantes uma maior variedade e possibilita explorar um nicho no qual mais se identifique (YAKMAN, 2008). A música possui como base princípios matemáticos e aqueles que gostam da lógica

matemática gostam do estudo sobre a teoria musical, por este motivo a área de Artes se encaixa nas disciplinas do STEM (DESIMONE, 2014). A música pode ser utilizada para engajar estudantes em nível pessoal e ainda aprofundar a compreensão de conceitos científicos, no entanto, há a necessidade de se desenvolver novas abordagens para isso (CROWTHER et al., 2016).

Neste contexto, Silla, Przybysz e Leal (2016) apresentam uma abordagem educacional utilizando DMIs comerciais (Guitarra, Baixo, Bateria e Teclado) para o ensino de programação e música de forma integrada. O método de ensino utilizado foi de reuniões semanais com o objetivo de criar jogos que interagissem com instrumentos musicais reais. O projeto contou com 4 estudantes do ensino médio durante um período de 36 semanas.

Na referida abordagem, a música é utilizada como um fator motivacional e ao mesmo tempo como um conteúdo da área de Artes por meio de teoria musical. Com o objetivo de integrar conceitos de software e hardware os estudantes foram apresentados ao Greenfoot⁷, o qual e permitiu que os estudantes aprendessem a linguagem de programação Java. Com isso, desenvolveram jogos (software) para interagir com os instrumentos musicais (hardware).

Um dos objetivos originais do projeto citado foi de elaborar um projeto que fosse capaz de ser desenvolvido em qualquer universidade ou escola, por este motivo foram utilizados instrumentos musicais reais (hardware) do tipo digital (DMI). Estes dispositivos possuem as vantagens de lidar com a detecção de notas, a possibilidade de conectá-los a um computador para manipulação dos sons e a possibilidade de concentrar vários estudantes no mesmo ambiente sem que haja interferências nas atividades de cada um.

Os resultados apresentados por Silla, Przybysz e Leal (2016) demonstram que o interesse dos estudantes em realizar seus projetos foi aumentando ao longo de cada semana e que ao final todos conseguiram desenvolver seu próprio jogo musical com interação com os DMIs.

Em 2018 o projeto de Silla, Przybysz e Leal (2016) foi modificado e aplicado para estudantes do ensino médio de escolas públicas, desta vez focado no público feminino, mantendo os mesmos DMIs comerciais (Guitarra, Baixo, Bateria e Teclado). A primeira modificação foi o acréscimo de um dia nos encontros semanais, dividindo entre assuntos técnicos e teoria musical. Outra modificação foi iniciar o projeto com conceitos lúdicos sobre lógica por meio de enigmas e desafios dos sites lightbot.com e code.org para posteriormente utilizarem o Greenfoot. A metodologia utilizada foi de reuniões realizadas 2 vezes por semana, contou com 4 estudantes pré-universitárias, durante um período de 36 semanas. Ao final foram obtidas as opiniões dos estudantes para uma análise qualitativa. Assim, a análise demonstrou que haviam muitos outros estudantes interessados em participar do

⁷ <https://www.greenfoot.org>

projeto (SILLA et al., 2018).

Apesar dos resultados positivos, os trabalhos de Silla et al. (2018), Silla, Przybysz e Leal (2016) possuem a limitação de escalabilidade devido a quantidade de DMIs disponíveis para cada estudante. Esta limitação se deve ao alto custo dos DMIs comerciais utilizados. O preço médio dos DMIs utilizados nos trabalhos citados pode variar entre R\$3.000,00 e R\$6.000,00 o que dificulta utilizar este tipo de abordagem para grandes números de estudantes, principalmente em lugares com baixo poder aquisitivo como as escolas públicas. Como uma solução para essa limitação de escalabilidade, Silla et al. (2018) propõem como trabalho futuro ensinar os estudantes a construir seus próprios DMIs utilizando dispositivos educacionais como Arduino⁸ e/ou Raspberry PI⁹.

Esta pesquisa é inspirada nos projetos de Silla et al. (2018), Silla, Przybysz e Leal (2016) os quais utilizaram-se de DMIs comerciais para o ensino de programação e música. No entanto, nesta pesquisa buscou-se associar o ensino de programação e eletrônica por meio da construção de instrumentos musicais digitais de baixo custo. Portanto, tem-se a seguinte questão norteadora: **Como a construção de instrumentos musicais digitais de baixo custo pode auxiliar no aprendizado de programação e eletrônica?**

1.2 Justificativa

Esta pesquisa de mestrado está delimitada em desenvolver instrumentos musicais digitais (DMIs) buscando utilizar materiais de baixo custo, e ensinar como construí-los em um curso de formação para estudantes universitários. Tem como contribuição uma abordagem pedagógica para o ensino de programação introdutória, música e eletrônica com o foco na construção de DMIs de baixo custo. Além disso, a abordagem proposta foi concebida com os educadores musicais em mente que, por sua vez, podem se tornar multiplicadores da abordagem proposta.

A maioria das escolas possuem limitações orçamentarias para o ensino de música, geralmente não possuem instrumentos musicais para os estudantes, local apropriado e recursos didáticos (PIZZATO; HENTSCHKE, 2014; SILVA; ALMEIDA, 2018). Além disso, é importante desenvolver nos estudantes a motivação para que sejam desenvolvedores/-criadores de tecnologias e não apenas usuários (SÁEZ-LÓPEZ; ROMÁN-GONZÁLEZ; VÁZQUEZ-CANO, 2016).

Portanto seguindo como motivação os trabalhos de Silla et al. (2018) e Silla, Przybysz e Leal (2016) esta pesquisa possui relevância trazendo uma abordagem inédita a qual contribuí para a melhoria da educação em lugares com baixo poder aquisitivo como as escolas públicas. Apesar de existirem trabalhos na literatura que foquem na

⁸ <https://www.arduino.cc/>

⁹ <https://www.raspberrypi.org/>

construção de DMIs (ROSAS; BEHAR, 2018; CHUNG; WU, 2017; CALEGARIO et al., 2017; CHUNG; CARTWRIGHT; CHUNG, 2014; HARRIMAN, 2015a; SAWYER et al., 2013), até o presente momento se tem conhecimento de nenhum trabalho que foque no desenvolvimento de DMIs de baixo custo, que possam ser utilizados tanto para o ensino de programação e eletrônica, quanto nas aulas de música.

1.3 Objetivos

Com base no cenário apresentado na justificativa, bem como no contexto problematizador dessa pesquisa, este trabalho tem como objetivo geral: Analisar como a construção de DMIs de baixo custo pode auxiliar no aprendizado de programação e eletrônica. Para que o objetivo geral seja alcançado, são propostos os seguintes objetivos específicos:

- a) Avaliar a viabilidade de construção de DMIs de baixo custo;
- b) Verificar os desafios impostos no planejamento de curso que relacione o ensino de programação e eletrônica a partir da construção de DMIs de baixo custo;
- c) Identificar os limites e possibilidades da integração do ensino de programação e eletrônica a partir da construção de DMIs de baixo custo.

1.4 Estrutura do Trabalho

O capítulo 2 apresenta o referencial teórico referente aos trabalhos encontrados na literatura com os principais conceitos utilizados para a construção de DMIs e para o ensino de programação e eletrônica. O capítulo 3 apresenta a metodologia deste trabalho baseada no método qualitativo de Pesquisa de Desenvolvimento de Objeto de Van der Maren (2004). O capítulo 4 apresenta Objeto 1 com a construção de DMIs de baixo custo. O capítulo 5 apresenta Objeto 2 com a formação. No capítulo 6 são apresentadas as conclusões e os trabalhos futuros para esta pesquisa.

2 Referencial Teórico

Neste capítulo são apresentados os principais conceitos e trabalhos encontrados na literatura que são relacionados com o escopo desta pesquisa. A seção 2.1 A Concepção de DMIs, apresenta os conceitos relacionados à construção de DMIs encontradas na literatura, bem como os tipos de DMIs, tecnologias utilizadas e suas aplicações. Na seção 2.2 O Ensino de Programação e Eletrônica, são apresentadas as principais abordagens educacionais utilizadas no ensino de programação e eletrônica.

2.1 A Concepção de DMIs

Um instrumento musical digital ou DMI - Acrônimo em inglês para *Digital Musical Instrument*, pode ser considerado como sendo um instrumento musical que utiliza sons gerados por computador (MIRANDA; WANDERLEY, 2006). A construção de um DMI consiste em um projeto multidisciplinar que pode envolver etapas de programação, eletrônica e música (JORDÁ, 2005). Desta forma, a utilização de DMIs no contexto educacional está relacionado ao STEAM *education*, onde Artes é representada na forma de música e as áreas de Tecnologia e Engenharia na forma de programação e eletrônica respectivamente.

Um projeto que visa construir um DMI pode ser denominado de luteria¹ digital, pois inclui um vasto número de tecnologias como eletrônica, sensores, sintetizadores², programação e áreas associadas ao ser humano como Psicologia, Fisiologia e Ergonomia (JORDÁ, 2005).

Os DMIs podem ser representados como dispositivos formados por dois módulos separados fisicamente. Um módulo contendo os dispositivos de entrada (ou controladores gestuais) e outro com a unidade de geração sonora. A conexão entre os módulos é realizada por meio de estratégias de mapeamento (MIRANDA; WANDERLEY, 2006).

Os DMIs permitem que o intérprete altere sons e a relação entre gestos e parâmetros sonoros, enquanto em um instrumento acústico o intérprete atua diretamente no mecanismo de produção sonora, o qual faz parte do corpo do instrumento, limitando o instrumento ao som e formato que foram construído (MARSHALL et al., 2009).

¹ A palavra Luteria tem origem no instrumento conhecido como Alaúde, difundido na Europa a partir século XV. O Alaúde recebeu o nome de "Luth" na França dando origem ao termo "Luthier" para se referir aos construtores de Alaúdes que por sua vez deriva a palavra "Luthieria" referindo-se à construção de alaúdes (MOURÃO; MORAIS, 2016). Neste contexto, o termo é comumente utilizado para referir-se a arte de construir instrumentos musicais independente do seu tipo como citado em (JORDÁ, 2005).

² Um sintetizador é um dispositivo capaz de gerar sons mediante aos elementos de tom, timbre e volume. Podem ser considerados um instrumento musical eletrônico por serem formados por uma coleção de circuitos eletrônicos. Em alguns casos possuem o formato de um teclado (CROMBIE, 1982).

Essa representação do DMI demonstra que esses dispositivos são como instrumentos musicais convencionais, mas se diferem de outros pelas suas particularidades em relação à forma com que o instrumento converte gestos físicos em sons. Enquanto em um instrumento tradicional o som é gerado a partir da vibração de partes do corpo do instrumento, o DMI utiliza um som gerado por um computador. Isso permite que os DMIs tenham maior liberdade quanto ao seu formato se comparados a outros instrumentos, uma vez que não dependem de sua estrutura física para gerar sons.

Ainda que um DMI possa ser construído em diferentes formatos, alguns trabalhos apresentam DMIs inspirados em instrumentos convencionais com modificações tanto no design quanto no som como em (COLLECCHIA; SOMEN; MCELROY, 2014; FAN; ESSL, 2013; LEVINSON, 2011), outros trabalhos buscam manter o formato original, mas acrescentar novas funções (LAMB; ROBERTSON, 2011; ROTHMAN, 2010).

Os DMIs podem ser classificados em quatro (4) grandes grupos baseados no controlador gestual (MIRANDA; WANDERLEY, 2006): Instrumentos Musicais Aumentados; Controladores gestuais em forma de instrumento; Controladores gestuais inspirados em instrumentos; Controladores gestuais alternativos.

- a. *Augmented Musical Instruments* (AMI): Instrumentos Musicais Aumentados (IMA) são instrumentos acústicos ou elétricos que possuem suas características principais mantidas, como formato e timbre, mas são acrescentados sensores que possibilitam “aumentar” (*augmented*) as funcionalidades do instrumento como efeitos e novos sons. Na Figura 1 é apresentado o “The Extended Clarinet” (NORMARK et al., 2016) que utiliza sensores para melhorar a performance musical e expressiva de um Clarinete.



Figura 1 – The Extended Clarinet

Fonte: (NORMARK et al., 2016)

- b. *Instrument-Like gestural controllers*: Controladores gestuais semelhantes a instrumentos são aqueles que reproduzem as características de instrumentos acústico, porém são construídos com diferentes sensores e componentes eletrônicos. Como por exemplo o “Seaboard” (LAMB; ROBERTSON, 2011) apresentado na Figura 2 que mantém o design de um teclado musical, porém possui a capacidade de manipular sons em tempo real;



Figura 2 – Seaboard

Fonte: (LAMB; ROBERTSON, 2011)

- c. *Instrument-Inspired gestural controllers*: Controladores gestuais inspirados em instrumentos, pretendem superar as limitações de outros instrumentos existentes, mas não possuem a intenção de copiá-los. Um exemplo é o TED – “Tangible Electrophonic Drumstick” (LEVINSON, 2011) apresentado na Figura 3 que funciona como uma baqueta para bateria musical, mas que permite gravar sons e manipulá-los de forma independente da superfície que é tocado;



Figura 3 – TED - Tangible Electrophonic

Fonte: (LEVINSON, 2011)

- d. *Alternate gestural controllers*: Controladores gestuais alternativos possuem características próprias com grandes diferenças de quaisquer outros instrumentos existentes. A Figura 4 apresenta o Neurohedron (HAYES, 2010), um instrumento em formato de dodecaedro que funciona como um sequenciador, onde cada face é um sensor de entrada e saída.

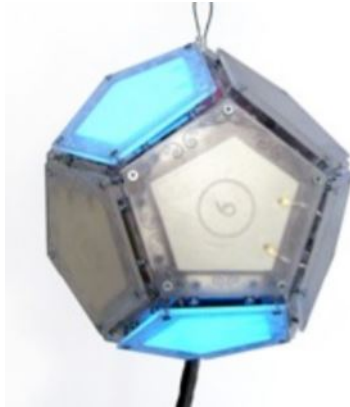


Figura 4 – Neurohedron

Fonte: (HAYES, 2010)

A facilidade de acesso à informação proporciona a criação de comunidades que visam compartilhar conhecimentos sobre a criação de DMIs, como o congresso *New Interfaces for Musical Expression* (NIME) e a cultura “Maker”.

Dentro da cultura Maker são encontrados diversos DMIs, pois a comunidade disponibiliza as formas de como construí-los, alcançando um maior número de pessoas interessadas em construir seu próprio DMI. Alguns exemplos são encontrados em [Arduino Project Hub \(2020\)](#) e [Instructables \(2020\)](#).

Como pode ser visto na Figura 5, um DMI pode ser representado por dispositivo de entrada (ou controle gestual) que interpreta um gesto físico, uma unidade de produção sonora e uma unidade de mapeamento que determina os sons para cada gesto de entrada (MIRANDA; WANDERLEY, 2006).

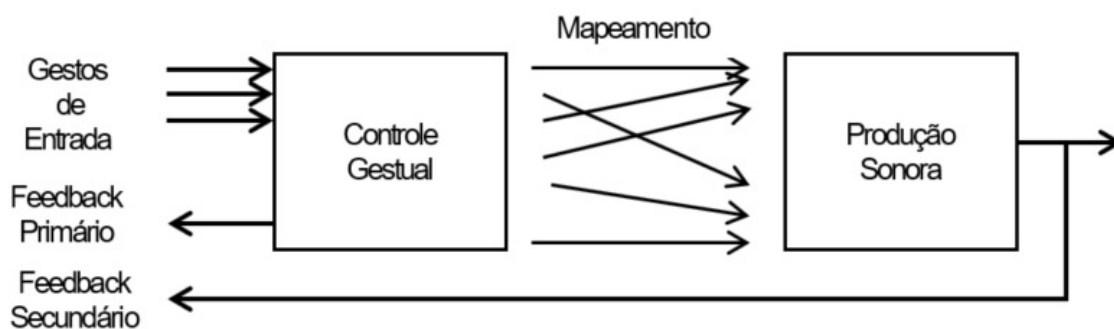


Figura 5 – Representação em blocos de um DMI

Fonte: (MIRANDA; WANDERLEY, 2006)

Esta representação (Figura 5) se inicia a partir do gesto de entrada, o qual representa um gesto físico reproduzido por quem está tocando o DMI. Assim, o controle gestual recebe um gesto físico gerando uma resposta inicial que pode ser visual, auditiva ou tátil-quinestésica, apresentado como Feedback Primário. Com isso, este gesto é convertido em um sinal elétrico por meio de um sensor.

Após concluir a leitura dos gestos de entrada, os sensores enviam os sinais para a etapa de mapeamento. Nesta etapa é realizada a definição do que fazer com cada sinal obtido. O mapeamento pode ser utilizado para identificar quando reproduzir uma nota musical após um sinal recebido ou mesmo reproduzir um som pré-definido. Por fim, a geração de som é responsável por dar a resposta ao dispositivo de entrada que estão mapeados, representado como Feedback Secundário (MIRANDA; WANDERLEY, 2006).

Utilizando desta abordagem (Figura 5) é possível sequenciar as etapas para a construção de um DMI em cinco instruções (MIRANDA; WANDERLEY, 2006):

1. Escolha dos gestos físicos serão utilizados;
2. Definir dispositivos de entrada com base nos gestos físicos;
3. Definir componente de geração sonora com base no som desejado;
4. Mapear os dispositivos para cada som desejado;
5. Escolher componentes para os feedbacks primários e secundários.

Estas etapas podem ser observadas nos DMIs Striso (VAN DER TORREN, 2014) e The Ghost (ROTHMAN, 2010) os quais possuem o design semelhante à instrumentos convencionais como a sanfona e a guitarra, mas com algumas alterações no formato e nos efeitos sonoros.

O Striso (Figura 6) é um DMI com aspectos de instrumento acústico³, prototipado com Arduino como unidade de mapeamento e posteriormente substituído pela placa STM32F4-DISCOVERY. O Striso possui sons próprios provindos de um sintetizador interno e por meio de um computador, foi construído com o objetivo de ser um instrumento que possibilite ser tocado em qualquer lugar e de forma intuitiva (VAN DER TORREN, 2014).

³ Instrumentos acústicos são aqueles que são formados por uma fonte que pode oscilar, de formas diferentes, a partir de gestos realizados por quem o controla, e possuem um sistema de ressonância que vibra em resposta às oscilações e junta-se à atmosfera (MIRANDA; WANDERLEY, 2006).



Figura 6 – Striso

Fonte: (VAN DER TORREN, 2014)

O The Ghost (Figura 7) é um DMI com o formato de uma guitarra, porém com botões no lugar das cordas. Foi prototipado em módulos para as funções dos botões e LEDs, outro separado para controle dos dados. Cada módulo possui um microcontrolador Atmega168 e o controle dos dados utiliza um Atmega1280, todos programados com a IDE do Arduino (ROTHMAN, 2010).



Figura 7 – The Ghost

Fonte: (ROTHMAN, 2010)

2.2 O Ensino de Programação e Eletrônica

Observa-se na literatura que a programação e eletrônica são utilizadas como conteúdos de aprendizado, na maioria das vezes, direcionados para áreas do STEM utilizando como motivação a robótica, muitas vezes citada como Robótica Educacional ou Pedagógica (BEZERRA JUNIOR; QUEIROZ; LIMA, 2018).

A Robótica Educacional ou Robótica Pedagógica consiste em uma abordagem de ensino que reúne diferentes áreas e tecnologias em um ambiente com o objetivo de construir, programar e controlar robôs (DI LIETO et al., 2017; JUNIOR et al., 2013). Neste contexto, um robô pode ser considerado qualquer dispositivo computacional eletromecânico, ou

seja, um objeto físico que interage com o ambiente de forma autônoma ou controlada (THOMAZ et al., 2009).

O surgimento da robótica na educação se dá a partir da integração de computadores nas escolas, por volta dos anos 70 nos Estados Unidos e chegando no Brasil nos anos 80, com atividades de programação com linguagem LOGO difundida por Seymour Papert (AZEVEDO; AGLAÉ; PITTA, 2010). Com esta linguagem Seymour Papert tornou a programação mais acessível para todas as idades. Então para comprovar que isso era realmente possível criou um dispositivo chamado de tartaruga de solo, o qual se tratava de uma tartaruga mecânica controlada por meio da linguagem LOGO que com auxílio de uma caneta desenhava seu percurso no chão (PAPERT, 1985). Ao realizar uma parceria com a empresa de brinquedos LEGO, Seymour Papert cria o termo construcionismo, o qual refere-se ao modelo educacional onde o estudante passa a construir um protótipo, sem a limitação de ser uma tartaruga, além de escrever o programa (AZEVEDO; AGLAÉ; PITTA, 2010).

No Brasil, a robótica pedagógica iniciou a partir da linguagem LOGO, em 1987 no Núcleo de Informática Aplicada à educação (NIED) localizado na UNICAMP, com projetos que utilizavam o computador para controlar robôs, inicialmente com a Tartaruga de Solo e o Traçador de Gráfico Educacional e posteriormente com os brinquedos LEGO, dando início em 1989 à primeira oficina de robótica pedagógica com o objetivo de formar pesquisadores na área. Ao longo dos anos 90 foram sendo desenvolvidas novas interfaces e tecnologias a fim de formar pesquisadores e professores com o objetivo de implementar a robótica pedagógica em escolas no Brasil. Assim, em 1997 se tem a versão da linguagem LOGO para os computadores pessoais, o TcLogo, o qual possibilita as primeiras aplicações da robótica em algumas instituições no Brasil, sendo utilizado pelo NIED com conceitos de automação para fabricas (D'ABREU et al., 2020, p.47).

Contudo, em 2000 percebe-se os avanços da robótica pedagógica criando-se incentivos como a Olimpíada Brasileira de Robótica (OBR) e o Workshop de Robótica Pedagógica (WRE) que possibilitam a consolidação da robótica em escolas apresentando ao público as aplicações, produtos e tendências da robótica, formando assim uma cultura (D'ABREU et al., 2020). Na Figura 8 é apresentado um histórico da robótica pedagógica baseado nas informações de (D'ABREU et al., 2020).

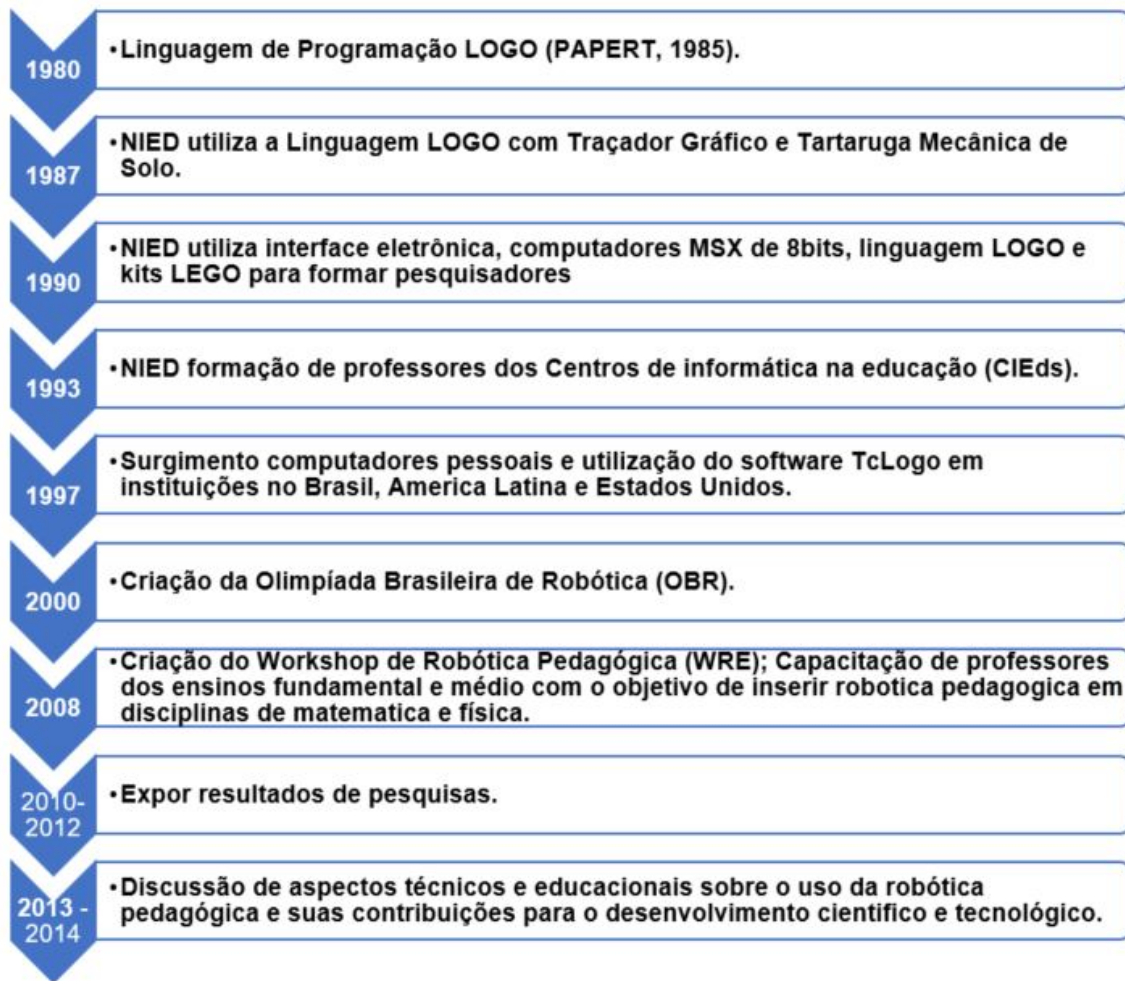


Figura 8 – Histórico da Robótica Pedagógica

Fonte: (D'ABREU et al., 2020)

A robótica na educação evoluiu ao longo dos anos podendo ser utilizada para integrar estudantes com dificuldades visuais utilizando peças de LEGO. Isso é possível com a construção de diferentes formas de acomodações como a utilização de bandejas com separadores para facilitar o acesso às peças, marcações com as instruções para programar e simplificar o reuso de códigos (LUDI; BERNSTEIN; MUTCH-JONES, 2018).

Em algumas abordagens são utilizadas apenas a etapa de programação por meio de softwares específicos, como é o caso do Earsketch⁴. O EarSketch é um *software* educacional projetado para interação entre alunos de diversas áreas utilizado em cursos introdutórios de computação através da interação com a programação e composição musical (SIVA et al., 2018). Outro *software* semelhante é o Music Maker⁵, que possui uma linguagem própria, Music Macro Language, que permite realizar a composição musical por meio de códigos (RAO, 2017).

⁴ <https://earsketch.gatech.edu/landing/>

⁵ <https://www.magix.com/us/music/music-maker/>

Devido à facilidade de acesso à informação e a criação de comunidades Maker, novos meios de levar a robótica para a educação vem surgindo. Além do LEGO, os microcontroladores como os que são utilizados na placa Arduino são responsáveis por facilitar o acesso à robótica.

A placa Arduino é um dispositivo de desenvolvimento de código aberto utilizado para protótipos com hardware e software. Possui uma interface de programação no computador na qual é permitido programar de maneira simples o funcionamento de entradas e saídas do seu hardware. A primeira placa de Arduino foi lançada em 2005 para ajudar estudantes de design com o objetivo de ser uma ferramenta fácil e ágil para prototipação (ARDUINO.CC, 2018). Atualmente existem diversos modelos baseados na linha de microcontroladores ATmega e com especificações que atendem uma grande parte dos usuários.

Uma das placas Arduino mais utilizadas é a Arduino UNO R3 (Figura 9) que utiliza como microcontrolador o Atmega-328p, além disso, possui o chip Atmega16 μ 2 nas versões mais recentes e o Atmega8 μ 2 nas versões anteriores (SCHUBERT; D'AUSILIO; CANTO, 2013). A utilização de um chip dedicado a comunicação USB permite à placa Arduino receber novos protocolos como o USB-MIDI para utilização do Arduino como dispositivo MIDI⁶ (DIAKOPOULOS; KAPUR, 2011).



Figura 9 – Placa Arduino UNO

Fonte: <https://store.arduino.cc/usa/arduino-uno-rev3> (Acesso em: 09/11/2020)

As placas de desenvolvimento geralmente são utilizadas para o aprendizado e/ou prototipagem, porém não possuem transdutores acoplados na placa. Desta forma o usuário não fica restrito a uma única aplicação podendo utilizá-la para inovar em áreas como a música, jogos, brinquedos, etc.

Transdutores são dispositivos que realizam a conversão de um tipo de energia em outro, por exemplo um microfone que transforma a energia sonora em energia elétrica. Quando ocorre uma conversão de um tipo de energia para uma tensão ou corrente elétrica esse transdutor passa a ser chamado de sensor. Por outro lado, ao se converter uma tensão

⁶ MIDI: Musical Instrument Digital Interface, em português: Interface Digital de Instrumentos Musicais. Trata-se de um protocolo de comunicação em tempo real para dispositivos musicais, ou seja é uma linguagem que transmite dados referentes aos sons de um instrumento, o que permite conectar instrumentos em computadores, interfaces digitais ou em outros instrumentos (LOY, 1985).

ou corrente elétrica em outra forma de energia o transdutor passa a ser chamado de atuador, como por exemplo, no caso de um alto-falante que converte energia elétrica em som (SINCLAIR, 2001). A placa Arduino possui portas digitais e analógicas de entrada e saída, o que permite a utilização de diferentes transdutores como botões, potenciômetros, piezoelétricos, entre outros.

Botões/Push-Button (Figura 10) são sensores digitais que possuem geralmente quatro terminais e um contato. Geralmente possuem dois componentes metálicos internamente que estão ligados aos terminais, quando pressionado, o contato liga os componentes metálicos fazendo com que o circuito se complete conduzindo corrente elétrica. Desta forma um microcontrolador é capaz de realizar a leitura de quando este botão é pressionado. São componentes mecânicos e por conta disso sofrem ruídos externos, para minimizar tais interferências são utilizados resistores de pull-up ou pull-down, porém para isso a placa de Arduino possui em seu hardware resistores internos que podem ser configurados via código. (PATSKO, 2006).



Figura 10 – Botão ou Push-Button

Potenciômetros (Figura 11) podem ser definidos como resistores com resistência ajustável. Possuem apenas 3 terminais que podem ser combinados dependendo da sua finalidade, geralmente utilizado para ajuste de parâmetros, desta forma funciona como um sensor analógico. O microcontrolador realiza a leitura da tensão que este componente ajusta e então realiza uma conversão dos valores de tensão para valores entre 0 e 1023, podem ser encontrados em formato logarítmico e linear (PATSKO, 2006).



Figura 11 – Potenciômetro

Piezoelétrico (Figura 12) possui a capacidade de converter uma força mecânica aplicada em energia elétrica. São construídos a partir de cristais de quartzo e fosfato de gálio que alteram a simetria de suas cargas elétricas ao ser aplicado alguma força. Um

sensor piezo tem capacidade de gerar tensão dependendo do seu tamanho, possuindo apenas dois conectores funcionam como sensor e atuador, geralmente encontrados em buzzer para emissão de sons. No Arduino podem ser utilizados como sensores digitais ou analógicos, desta forma este dispositivo pode ser utilizado em várias áreas ([CARSTENS, 1993](#)).



Figura 12 – Sensor Piezoelétrico

3 Metodologia

A presente pesquisa tem como objetivo geral “Analisar como a construção de DMIs de baixo custo pode auxiliar a aprendizagem de programação e eletrônica”, assim, caracteriza-se como uma abordagem qualitativa. Neste sentido, é apropriado utilizar a pesquisa qualitativa quando precisa-se explorar um problema ou uma questão (CRESWELL, 2010).

Como método qualitativo foi adotado a Pesquisa Desenvolvimento (VAN DER MAREN, 2004). Esta metodologia pode assumir três formas: 1)Desenvolvimento de conceito; 2)Desenvolvimento de objeto; 3)Desenvolvimento/aperfeiçoamento de habilidades pessoais como ferramentas profissionais (VAN DER MAREN, 2004). Desenvolvimento de conceito consiste em buscar por aplicações ou ferramentas que levem a teorias para de fato promover uma nova ideia. Desenvolvimento de Objeto visa a solução de problemas encontrados na prática diária. As abordagens que pertencem ao campo da didática e tecnologia educacional são áreas onde este tipo de pesquisa é mais comum. Desenvolvimento ou aperfeiçoamento de habilidades pessoais como ferramentas profissionais consiste em desenvolver conhecimentos com proposito profissional (VAN DER MAREN, 2004).

Esta pesquisa se enquadra na forma de Pesquisa de Desenvolvimento de Objeto. A qual o objeto é uma solução encontrada para os problemas encontrados em um contexto. O método escolhido se justifica pela construção de DMIs de baixo custo para o ensino de programação e eletrônica por meio de uma formação para estudantes do curso de Licenciatura em Música.

3.1 Etapas da Pesquisa Desenvolvimento de Objeto

As etapas da Pesquisa Desenvolvimento de objeto são: Análise do Contexto, Análise do Objeto, Preparação e Desenvolvimento/Implantação. A Figura 13 apresenta as etapas da Pesquisa Desenvolvimento de Van der Maren (2004), no entanto foi adaptada para que a Análise do Contexto fosse posicionada a esquerda no diagrama, pois o método se inicia pela Análise do Contexto passando para a Análise do Objeto, Preparação e por fim Desenvolvimento/Implantação.

A Análise do Contexto consiste em observar as demandas e necessidades que levaram a construção do objeto, que conhecimentos e motivações foram desejados, que objetivos e intenções foram alcançados, em que contexto está inserido, sobre qual conteúdo é tratado, com quais processos foi executado, com que meios foi contemplado e quem é o público alvo (VAN DER MAREN, 2004).

A Análise do Objeto consiste em conceitualizar o objeto a fim de desenvolver um modelo que o represente com os componentes e as restrições que devem ser atendidas. Além disso, deve ser estabelecida uma lista com ordem de prioridade para os componentes e restrições. Desta forma, pode-se identificar os elementos que são passíveis de serem sacrificados durante a fase de implementação caso dificuldades sejam encontradas no processo (VAN DER MAREN, 2004). Logo, esta etapa é composta por conceptualização e modelagem do objeto.

Assim que seja estabelecida a representação do objeto, é realizada a etapa de Preparação, na qual são elaboradas estratégias e alternativas para criação de um protótipo do objeto, sabendo que este será um compromisso entre o que é desejado e o que é possível (VAN DER MAREN, 2004). Por fim, tem-se a etapa de Desenvolvimento/Implantação do objeto que executa os procedimentos para criação do protótipo realizando adaptações, modificações e avaliações para que por fim realizar a implantação do objeto.

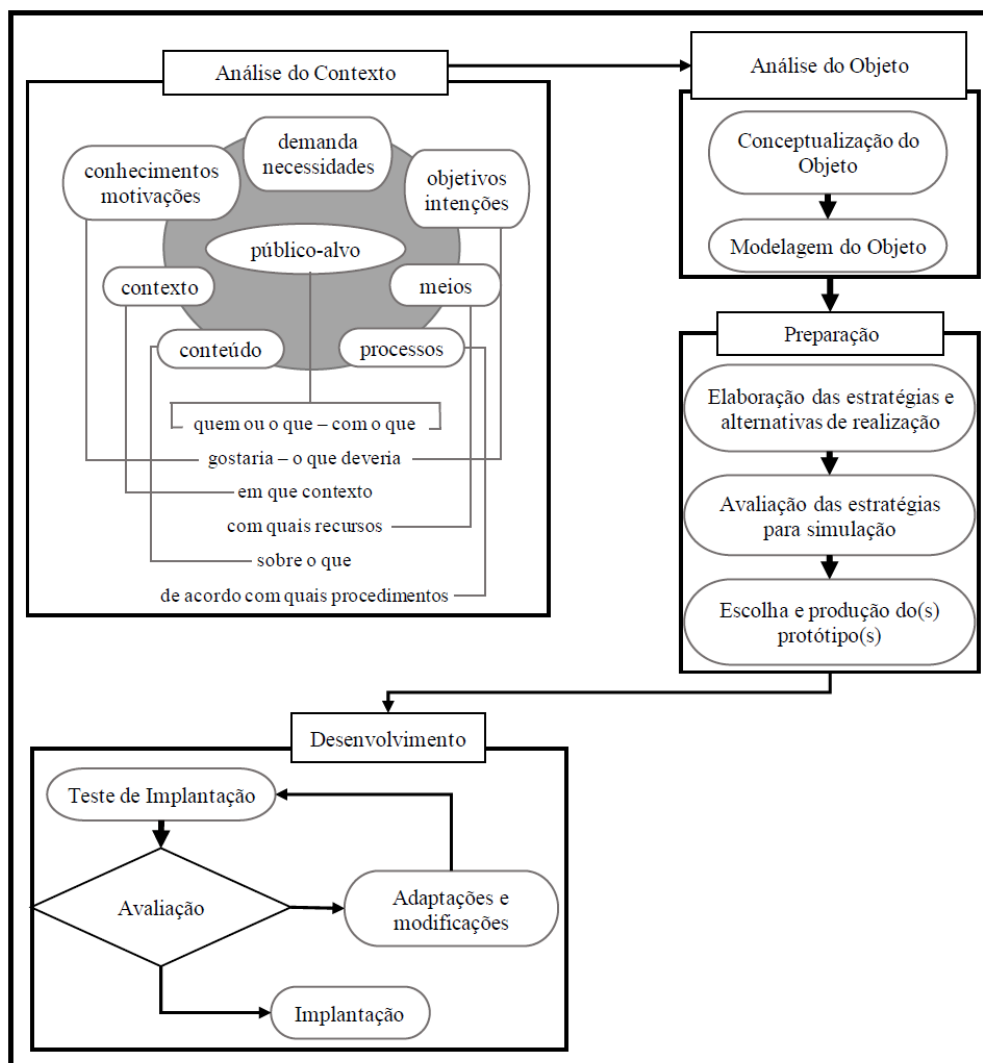


Figura 13 – Etapas da Pesquisa Desenvolvimento de Objeto

Fonte: Adaptado de (VAN DER MAREN, 2004)

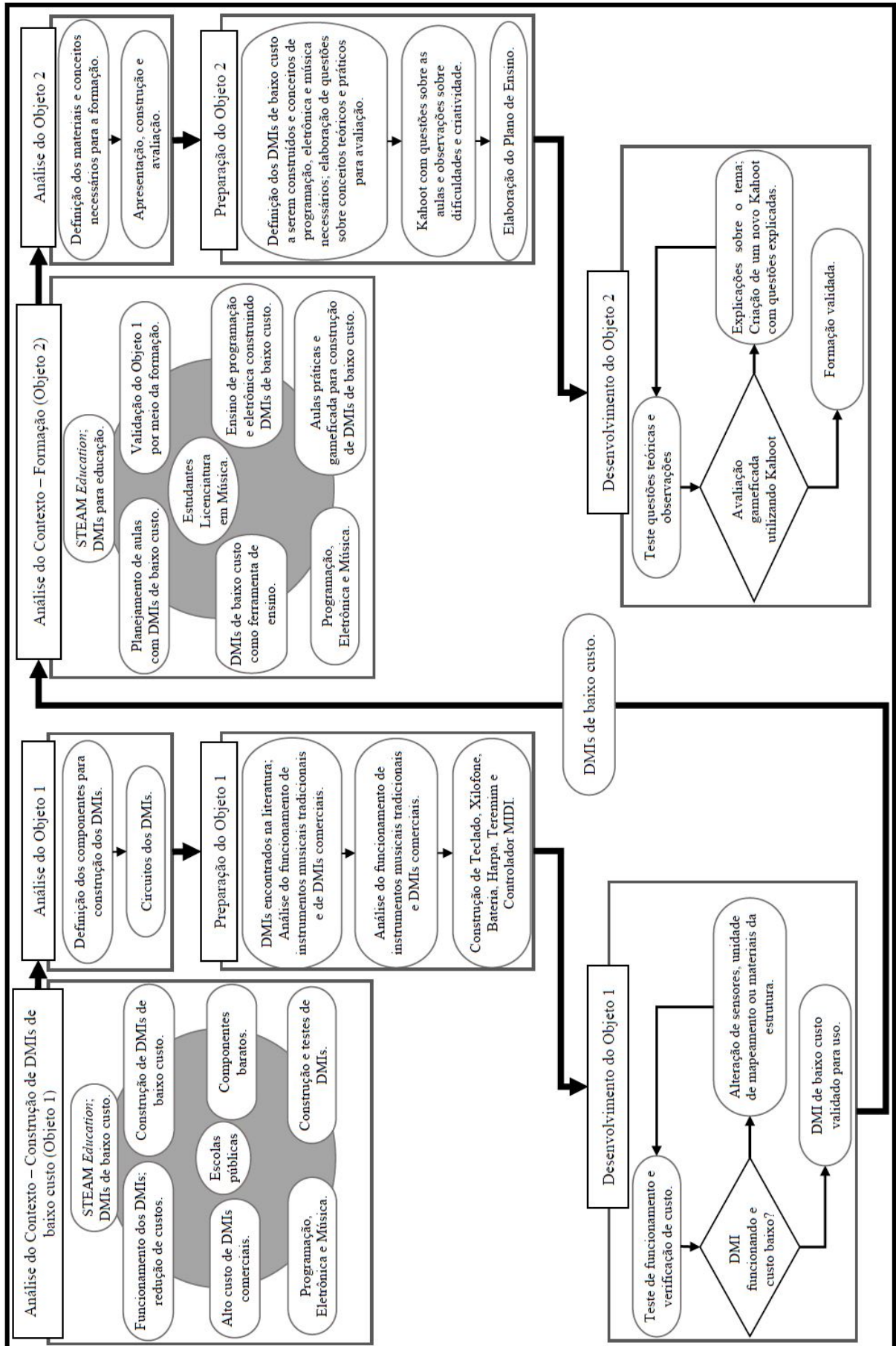


Figura 14 – Etapas da Pesquisa Desenvolvimento de Objeto Adaptada

Fonte: O autor 2020

A partir do objetivo geral desta pesquisa são observados dois contextos diferentes, a construção de DMIs de baixo custo e o aprendizado de programação e eletrônica. Por este motivo as etapas da Pesquisa Desenvolvimento de Objeto foram realizadas duas vezes, uma para a construção de DMIs de baixo custo (Objeto 1) e outra para a Formação (Objeto 2) como é apresentado na Figura 14. Com isso, o diagrama da Pesquisa Desenvolvimento de Objeto de Van der Maren (2004) (Figura 13) foi adaptado para que o Objeto 2 inicie ao final do Desenvolvimento do Objeto 1. O diagrama da Figura 14 se inicia pela Análise do Contexto do Objeto 1, passando para Análise do Objeto 1, Preparação do Objeto 1 e Desenvolvimento do Objeto 1. Com isso, se tem como resultado os DMIs de baixo custo, os quais dão início à Análise do Contexto do Objeto 2, passando para Análise do Objeto 2, Preparação do Objeto 2 e Desenvolvimento do Objeto 2 que tem como resultado a Formação. Cada etapa é detalhada nos capítulos 4 Construção de DMIs de Baixo Custo (Objeto 1) e 5 Formação (Objeto 2).

4 Construção de DMIs de Baixo Custo (Objeto 1)

Neste capítulo são apresentadas as etapas percorridas para a construção dos DMIs de baixo custo, de acordo com as etapas de pesquisa desenvolvimento de objeto apresentadas no capítulo 3. Este capítulo está organizado de acordo com as principais etapas da metodologia de desenvolvimento de objeto, a seção 4.1 apresenta a Análise do Contexto do Objeto 1. A seção 4.2 apresenta a Análise do Objeto 1. A seção 4.3 apresenta a Preparação do Objeto 1. A seção 4.4 apresenta o Desenvolvimento do Objeto 1.

4.1 Análise do Contexto do Objeto 1

A Análise do Contexto do Objeto 1 se deu pelos resultados dos trabalhos desenvolvidos por Silla, Przybysz e Leal (2016) em “*Music Education Meets Computer Science and Engineering Education*” e Silla et al. (2018) em “*Girls, Music and Computer Science*”. Em Silla, Przybysz e Leal (2016) foi proposta uma abordagem de ensino envolvendo música e lógica de programação de forma integrada. Neste contexto, utilizou-se uma forma para atrair e motivar estudantes do ensino médio para os cursos de Engenharia e Ciência da Computação. Assim, o referido estudo teve como objetivo ensinar programação e música por meio do desenvolvimento de jogos que interagissem com Instrumentos Musicais Digitais (DMIs) comerciais.

Originalmente Silla, Przybysz e Leal (2016) tiveram como objetivo desenvolver um projeto que fosse possível utilizar em qualquer escola ou universidade. Por este motivo foram escolhidos DMIs como ferramenta para o projeto, uma vez que estes dispositivos oferecem vantagens em se manipular os sons por meio de um computador e não necessitar de uma sala isolada ou a prova de sons. O projeto citado utilizou quatro DMIs comerciais (Baixo, Guitarra, Bateria e Teclado) e contou com quatro estudantes do ensino médio para um curso semanal. Os estudantes participaram de 36 encontros para trabalharem em projetos envolvendo música e programação de jogos. Os encontros foram organizados em 5 fases:

1. Ensino de Programação Java;
2. Ensino de Teoria e prática musical;
3. Comunicação dos instrumentos musicais com o código em Java;
4. Desenvolvimento da interface dos jogos;

5. Desenvolvimento dos projetos individuais pelos estudantes.

A abordagem consistiu em ensinar programação orientada objetos utilizando Java com a IDE Greenfoot¹, possibilitando que cada estudante desenvolvesse um jogo que interagisse com os DMIs conectados ao computador. O interesse dos estudantes foi crescendo ao longo das fases devido ao aumento na motivação em relação aos projetos individuais. Após 36 semanas todos os estudantes conseguiram finalizar seus projetos. Os autores destacam que a partir da segunda fase do projeto os estudantes iam ao laboratório cerca de 2 ou 3 dias da semana.

Com base nos resultados do trabalho de 2016, Silla et al. (2018) realizaram uma versão melhorada do projeto anterior, o qual teve como motivação atrair mais estudantes do sexo feminino para as áreas de Engenharia e Ciência da Computação. Esta segunda edição permaneceu com a mesma proposta do trabalho anterior em utilizar *hardware* (DMI) e *software* (desenvolvimento de jogos) para o ensino de programação e música de forma integrada (SILLA et al., 2018).

Os resultados dos trabalhos referidos demonstram que é possível ensinar música e programação simultaneamente, combinando DMIs comerciais e jogos desenvolvidos pelos estudantes. Tais projetos são contribuições para um cenário global onde se busca alternativas para atrair, motivar e engajar mais estudantes para as áreas de Engenharia e Ciências da Computação. Neste contexto, a utilização de música com a computação faz parte de uma série de cursos conhecidos como “*Performamatics*” (RUTHMANN et al., 2010).

“*Performamatics*” é um projeto que inicialmente foi criado com o objetivo de melhorar o engajamento de estudantes do curso de Ciência da Computação por meio da música com aplicações do mundo real conectando a teoria e a prática (GREHER; HEINES, 2008). Originalmente o “*Performamatics*” foi projetado para aumentar o engajamento de estudantes de Ciências da Computação, mas devido a parceria entre as áreas de Artes, Música e Inglês levou aplicações do mundo real aos estudantes de todas as áreas (HEINES; GREHER; KUHN, 2009)

Neste contexto, o “*Scratch*” pode ser uma alternativa para o ensino de programação para estudantes do curso de Música, o que permite utilizar dispositivos como o “*InchiBoards*” o qual possui diversos sensores, tornando possível a interação gestual com a música no “*Scratch*” (RUTHMANN et al., 2010).

É possível observar que as abordagens do “*Performamatics*” estão inseridas no contexto do movimento STEAM *education*, uma vez que “*Performamatics*” inclui a área de Artes de forma integrada a Tecnologia. Além disso, são encontradas diferentes formas de ensino relacionadas as Artes que integram as áreas do movimento STEAM *education*,

¹ <https://www.greenfoot.org/>

como confecção de tecidos com eletrônica, (BUECHLEY et al., 2008; LAU et al., 2009; PEPPLER, 2013), teatro de robôs (BARNES et al., 2017; JEON et al., 2016; KO et al., 2020), jogos de futebol com robôs (DIAS et al., 2015; KRISHNAN; ASWATH; UDUPA, 2014; YOON; BAEK, 2018), conceitos musicais (BURG; ROMNEY; SCHWARTZ, 2013; MEYERS et al., 2009; MISRA; BLANK; KUMAR, 2009) e teatro musical (BARNES et al., 2019; MCLAMORE, 2016).

Pode se notar que a grande maioria das abordagens utilizam a robótica de alguma forma. Mesmo no contexto STEAM *education* envolvendo abordagens do “*Performamatics*”, os robôs estão presentes. Tais trabalhos possuem resultados positivos, mas tornam-se um nicho específico para um único público alvo, fazendo com que o objetivo dos trabalhos se confunda com as ferramentas da robótica.

São encontradas na literatura abordagens educacionais que utilizam DMIs com a motivação de atrair e engajar estudantes para as áreas do STEAM *education*. Algumas destas buscam utilizar os DMIs para melhorar a educação musical (MYLLYKOSKI; TUURI, 2015; RIGLER; SELDESS, 2007; ANTUNES; TEIXEIRA; PIREDDU, 2017), outros trabalhos utilizam kits de robótica e programação em blocos para construir DMIs como incentivo à criatividade e ao aprendizado de conceitos MAKERS (SAWYER et al., 2013; HARRIMAN, 2015a; ROSAS; BEHAR, 2018).

As ferramentas educacionais que utilizam a construção de DMIs devem considerar as abordagens de ensino, os equipamentos e os custos necessários (HARRIMAN, 2015b). Portanto, esta pesquisa está inserida em um contexto que visa o ensino de programação (Tecnologia) e eletrônica (Engenharia) com a música (Artes) como fator motivacional. Neste seguimento, a demanda desta pesquisa se enquadra no movimento STEAM *education* e possui como motivação tornar acessível DMIs para o ensino de programação.

Apesar dos resultados positivos, a utilização de DMIs comerciais em abordagens educacionais possui uma limitação de escalabilidade do projeto devido aos elevados custos de cada DMI. Os trabalhos de Silla et al. (2018), Silla, Przybysz e Leal (2016) contaram com quatro DMIs (Bateria, Teclado, Guitarra e Baixo) apresentados na Tabela 1.

Tabela 1 – Preços dos DMIs em Silla, Przybysz e Leal (2016) e Silla et al. (2018)

Instrumento Musical Digital	Preço Médio (R\$)
Bateria Roland Digital Drums Kit Model: HD – 3	3000,00
Teclado Roland GW – 8 Keyboard	3650,00
Guitarra (padrão) + Captador GK-3 Roland + Roland GR-55 Sintetizador	6000,00
Baixo (padrão) + Captador GK-3B Roland + Roland GR-55 Sintetizador	6000,00

Fonte: O autor, 2020 - Preço médio consultado no dia 15/04/2020

Pode-se verificar na Tabela 1 que a guitarra e o baixo foram utilizados em conjunto com um captador específico para cada instrumento musical (“*GK-3 Roland*” e “*GK-3B Roland*”) e um sintetizador para ambos os instrumentos. Silla, Przybysz e Leal (2016) relatam que para reduzir custos foram adquiridos uma guitarra padrão e um baixo padrão, mas para utilizá-los em um computador foi necessário adaptar o captador “*GK3 Roland*” para guitarra e o captador “*GK-3B Roland*” para o baixo e, em seguida, conecta-los ao sintetizador e então conectar os DMIs ao computador. Além disso, o sintetizador permitia conectar apenas um DMI por vez.

Nesta etapa identificou-se a necessidade de reduzir os custos dos DMIs que apesar de existirem trabalhos na literatura que foquem na construção de DMIs, até o presente momento se tem conhecimento de nenhum trabalho que foque no desenvolvimento de DMIs de baixo custo, que possam ser utilizados tanto para o ensino de programação e eletrônica, quanto nas aulas de música. Com isso, foi elaborado o primeiro questionamento: “É possível construir DMIs com baixo custo?”. Tal questionamento deu origem ao primeiro objeto desta Pesquisa Desenvolvimento, ou seja, é apresentada uma solução, a construção de DMIs de baixo custo, para o problema de escalabilidade em abordagens educacionais que utilizam DMIs comerciais para o ensino de programação e música, encontrado na Análise do Contexto do Objeto 1. Neste seguimento, a demanda desta pesquisa se enquadra no movimento *STEAM education*.

Com isso, os conhecimentos desejados são relacionados a como construir DMIs com baixo custo tendo como motivação tornar acessíveis os DMIs para abordagens educacionais com baixo poder aquisitivo voltadas para a programação e eletrônica. Os objetivos e intenções são de criar DMIs de baixo custo. Para isso, os conteúdos tratados são programação, eletrônica e música. Logo, os meios adotados a fim de solucionar os problemas identificados no contexto foram construir diversos DMIs buscando o baixo custo por meio de componentes eletrônicos, softwares gratuitos e materiais reutilizáveis. De acordo com os procedimentos de construir e testar DMIs que sejam tocados por meio de toques manuais, movimento e por proximidade.

Verifica-se que há dificuldades de escalabilidade em projetos que utilizam DMIs comerciais para o ensino de programação e/ou eletrônica. Outros trabalhos na literatura utilizam a construção de DMIs em abordagens educacionais. No entanto, estes trabalhos buscam por melhorias na área de educação musical (ROSAS; BEHAR, 2018), incentivo à criatividade (HARRIMAN, 2015b) ou em conceitos Makers (SAWYER et al., 2013). Por estes motivos, o Objeto 1 trata-se da construção de DMIs de baixo custo como solução para a dificuldade de escalabilidade de trabalhos que utilizam DMIs comerciais no contexto *STEAM education*. Os resultados da Análise do Contexto do Objeto 1 são apresentados na Tabela 2.

Tabela 2 – Análise do Contexto Objeto 1 - Construção de DMIs de Baixo Custo

Etapas	Análise do Contexto
Demanda Necessidades	STEAM <i>education</i> ; Reduzir Custos dos DMIs.
Conhecimentos e Motivações	Como funcionam os DMIs comerciais; Como construir de DMIs de baixo custo; Tornar acessível DMIs para o ensino de programação e eletrônica.
Objetivos e Intenções	Criar DMIs de baixo custo para o ensino de programação e eletrônica.
Contexto	DMIs comerciais possuem o custo muito alto para o ensino de programação, dificultando a escalabilidade de projetos com este tema e prejudicando a acesso desses conhecimentos em lugares com baixo poder aquisitivo como escolas públicas.
Meios	Construir DMIs de baixo custo utilizando programação, eletrônica e materiais baratos.
Conteúdos	Programação, Eletrônica e Música.
Processos	Construir e testar diferentes DMIs de baixo custo.
Público Alvo	Abordagens educacionais com baixo poder aquisitivo voltadas para a programação e eletrônica.

Fonte: O Autor, 2020

4.2 Análise do Objeto 1

Com base na Análise do Contexto tem-se como resposta o Objeto 1 - A Construção de DMIs de Baixo Custo. Com isso, a Análise do Objeto 1 é composta pela Conceptualização do Objeto 1 e pela Modelagem do Objeto 1. A seção 4.2.1 Conceptualização do Objeto 1 refere-se às etapas realizadas para identificar os componentes necessários para construir DMIs de baixo custo. A seção 4.2.2 Modelagem do Objeto 1, apresenta uma representação dos DMIs com base na conceptualização do Objeto 1.

4.2.1 Conceptualização do Objeto 1

Um DMI pode apresentar diferentes características quanto ao tamanho, formato e som, podendo assumir formas inspiradas em instrumentos musicais convencionais como violão, piano, bateria, bem como formatos criados a partir da imaginação ou de especificações próprias de quem o constrói. Quando se acrescenta sensores em um instrumento musical convencional pode classifica-lo como *Augmented Musical Instruments* (AMI). Neste caso, pode-se aproveitar da estrutura de um instrumento musical convencional e adicionar

efeitos e adaptações. Com isso, A conceptualização do objeto 1 foi realizada a partir da seguinte questão - Como podem ser construídos os DMIs?

Para analisar quais formatos os DMIs poderiam ser construídos foi realizado um levantamento dos trabalhos encontrados no congresso NIME no período entre 2010 e 2016. Esta conferência foi escolhida devido a diversidade de trabalhos envolvendo instrumentos musicais. Com base neste levantamento, foram observadas as principais características dos instrumentos musicais encontradas na literatura, as quais pudessem auxiliar na etapa de prototipação dos DMIs. Assim, foram analisadas características sobre o formato, sensores, geração sonora. A apuração deste levantamento é apresentada na Tabela 3.

A Tabela 3 está organizada em Autor, Instrumento (DMI-AMI), Sensores e Mapeamento e Gerador Sonoro. Na coluna Instrumento – (DMI/AMI) são apresentados os nomes dos instrumentos musicais, utilizando os mesmos nomes que os autores se referem em cada trabalho. Na coluna sensores são apresentados os tipos de sensores utilizados para a detecção de gestos do usuário, quanto ao funcionamento prático, por este motivo foram classificados em sensores de:

- Toque: Botões, Piezoelétrico, Resistivos, Teclados, Capacitivos e Indutivos;
- Movimento: Acelerômetro, Magnetômetro, Giroscópio e Motores;
- Luz/Câmera: Luminoso, Infravermelho, Câmera, Projetor, Câmera Laptop e Kinect;
- Microfone: Microfones, Microfone Celular e Microfone do Computador;
- Proximidade: Ultrassom, Radio Frequência e Infravermelho.

Na coluna Mapeamento e Gerador Sonoro apresenta o meio utilizado na construção do DMI para mapear os sinais dos sensores e para os dispositivos de geração sonora. Na maioria dos trabalhos o mapeamento e geração sonora são construídas com o mesmo dispositivo, por este motivo estão organizados na mesma coluna. Foram classificados em:

- Microcontroladores Atmel (Arduino UNO, Nano, Mega, CI Atmega);
- Computador (PC), Raspberry PI (RASP);
- Kinect - (Sensores de Movimento);
- Lego (Kits de robótica da empresa LEGO);
- Celulares, Iphone, Ipad, Ipod (Dispositivos da empresa Apple);
- Computadores Mac e processadores ARM.
- Não Info (Não Informado no Trabalho).

Tabela 3 – Levantamento NIME 2010-2016

Autor	Instrumento (DMI-AMI)	Sensores					Mapeamento e Gerador Sonoro
		Toque	Movimento	Luz/Câmera	Microfone	Proximidade	
(MCPHERSON; KIM, 2010)	Acoustic Piano (AMI)	X	X				ATM
(TAYLOR; HOOK, 2010)	FerroSynth (DMI)	X	X				PC
(HAYES, 2010)	The Neurohedron (DMI)	X	X				ATM e PC
(MIYAMA, 2010)	Peacock (DMI)			X			ATM e PC
(CASSINELLI et al., 2010)	ScoreLight (DMI)			X			ATM
(ROTHMAN, 2010)	The Ghost (DMI)	X	X				ATM
(NAGASHIMA, 2010)	Peller Min (DMI)			X			ATM
(MARSHALL; WANDERLEY, 2011)	Feedback actuators (DMI)	X	X	X	X	X	PC
(BERDAHL; JU, 2011)	Satellite CCRMA (DMI)						ATM e PC
(LAMB; ROBERTSON, 2011)	Seaboard – Piano (DMI)	X					Não info
(RAMKISSOON, 2011)	Bass Sleeve (AMI)	X	X				ATM
(KAPUR et al., 2011)	The Karmetik (DMI)						ATM
(HENRIQUES, 2012)	SONIK SPRING (DMI)		X				MIDITron
(SCHIESSER; SCHACHER, 2012)	SABRe (AMI)	X	X				PC
(LEVINSON, 2011)	TED – Drum (DMI)	X			X		PC
(MCPHERSON, 2012)	TouchKeys (AMI)	X					PC
(LEEuw, 2012)	Electrumpet (AMI)	X				X	Iphone
(FAN; ESSL, 2013)	Air Violin (DMI)			X			Kinect e PC
(HONG; YEO, 2013)	Laptap (DMI)				X		PC
(JACKIE et al., 2013)	SoloTouch (DMI)	X					ATM
(JENKINS et al., 2013)	EROSS – Trumpet (AMI)			X			ATM
(BARBOSA et al., 2013)	Illusio (DMI)	X					Não info
(KLEINBERGER, 2013)	PAMDI(DMI)	X					ATM
(NAKANISHI; MATSUMURA; ARAKAWA, 2013)	Powder Box (DMI)	X	X				ATM

(NAM, 2013)	mPoi(DMI)		X				ATM
(SCHACHER, 2013)	Quarterstaff (DMI)	X	X				ATM
(PORTNER, 2014)	The Chimaera (DMI)		X				ARMCortex
(MAMEDES et al., 2014)	Intonaspacio (DMI)	X	X		X		ATM e PC
(OVERHOLT; GELINECK, 2014)	Augmented violin (AMI)		X				Ipod e Mac
(GUREVICH, 2014)	Mechatronic (DMI)						Não info
(JOHNSTON et al., 2014)	Polus (DMI)						ATM e PC
(ARELLANO; MCPHERSON, 2014)	Radear(DMI)					X	ATM e PC
(VAN DER TORREN, 2014)	The Striso (DMI)	X					ATM e PC
(SNYDER; RYAN, 2014)	The Birl (DMI)						ATM e PC
(BARENCA; CORAK, 2014)	Manipuller II (DMI)	X					ATM
(LONG, 2014)	Robotic Taishogoto (DMI)	X					ATM e PC
(COLLECCHIA; SOMEN; MCELROY, 2014)	The Siren Organ (DMI)	X					ATM
(BERTHAUT; KNIBBE, 2014)	Wubbles (DMI)						Kinect e PC
(FLØ; WILMERS, 2015)	Doppelganger (DMI)	X				X	ATM
(PIEPENBRINK et al., 2015)	Bistable Resonator Cymbal (DMI)				X		Não info
(MCCLOSKEY et al., 2015)	Accessibility (DMI)						ATM e PC
(HARRIMAN, 2015b)	Modular-muse (DMI)	X	X			X	ATM
(D’ALESSANDRO et al., 2015)	AirPiano (DMI)			X			Não info
(MARLEY; WARD, 2015)	Reactable Gestroviser (DMI)						PC
(HATTWICK; WANDERLEY, 2015)	The Pearl (DMI)	X				X	Não info
(PON et al., 2015)	Womba (DMI)	X			X		PC
(BLESSING; BERDAHL, 2015)	LapBox (AMI)	X					RASP
(BALDWIN et al., 2016)	Tromba (AMI)				X		Não info
(SELLO, 2016)	Hexenkessel (AMI)			X		X	PC
(RIEGER; TOPEL, 2016)	Driftwood (DMI)	X					ATM
(NORMARK et al., 2016)	Clarinet (AMI)	X	X				ATM
(NAGASHIMA, 2016)	Multi Rubbing Tactile (DMI)	X					ATM
(NASH, 2016)	keyboard QWERTY (DMI)	X	X				ATM
(MEACHAM; KANNAN; WANG, 2016)	The Laptop Accordion (DMI)			X			PC

(JAKOBSEN et al., 2016)	Hitmachine (DMI)	X	X	X	X	X	LEGO
(MICHON et al., 2016)	BladeAxe (AMI)	X	X	X	X	X	iPad

Fonte: O Autor, 2017.

Ao todo foram analisados 56 artigos com um DMI/AMI cada, desenvolvidos em vários formatos e com diferentes componentes como Arduino (Atmel), Raspberry PI, Lego entre outros. Com isso foi verificado quais tipos de DMIs poderiam ser construídos, quais componentes são mais utilizados e o que poderia ser adaptado para um DMI de baixo custo. Com base nisso, foi determinado que seriam construídos DMIs baseados no formato de instrumentos musicais convencionais mas não seriam AMIs. Assim, inicialmente foram escolhidos 5 DMIs para serem prototipados nos formatos de Teclado, Xilofone, Bateria, Harpa e Teremim.

Para construir os DMIs foi escolhida a representação de DMI apresentada por [Miranda e Wanderley \(2006\)](#), que tem como componentes principais o controle gestual, mapeamento e geração sonora. Para cada um dos tipos de instrumentos (Teclado, Xilofone, Bateria, Harpa e Teremim) foram escolhidos diferentes tipos de controles gestuais, sendo estes: Botões, Sensor Capacitivo, Piezoelétrico, Laser, Sensor de Luminosidade, Potenciômetros, Sensor de Proximidade e os circuitos com osciladores e amplificador operacional. Para unidade de mapeamento foram escolhidas as placas Arduino UNO e Arduino Nano, devido às portas digitais e analógicas disponíveis na mesma placa, além da praticidade para programar e gravar o código que a IDE Arduino dispõe quando comparadas a outras placas, além do baixo custo. Quanto a geração sonora foram escolhidas as placa Arduino UNO e Arduino Nano por conta das portas PWM² que permitem gerar sons por meio de um alto falante ou Buzzer e por conta da possibilidade de se usar um cartão SD para armazenar sons dos instrumentos convencionais. Além destes componentes, foi idealizado utilizar materiais reutilizáveis para a estrutura dos DMIs como papelão, isopor, pvc, palitos de madeira, caixas de plástico e placas de metal.

Inicialmente foi idealizado que seriam construídos DMIs que não necessitassem do computador para serem tocados, mas sim utilizassem as placas Arduino para mapeamento e geração sonora. No entanto, a utilização do Arduino requer o uso de um computador para abordagens de ensino de programação e para gravação dos códigos, por estes motivos foi idealizado que os DMIs escolhidos seriam também prototipados utilizando o computador como gerador sonoro. Com isso, foi acrescentado o computador como gerador sonoro aos DMIs citados (Teclado, Xilofone, Bateria, Harpa e Teremim) e foi acrescentado um DMI denominado de Controlador MIDI³, reunindo os sensores utilizados nos 5 DMIs escolhidos

² PWM: *Pulse Width Modulation*, em português: Modulação por Largura de Pulso. Trata-se de uma técnica para obter sinais de saída analógica em meios digitais ([BARR, 2001](#)).

³ Controladores MIDI são dispositivos que possuem a capacidade de enviar dados MIDI para outro dispositivo, como um teclado, computador ou sintetizador. Por esse motivo não geram sons, mas sim

inicialmente em um único dispositivo. Com o computador como fonte de geração sonora poderia-se desenvolver um software capaz de receber dados por meio da conexão USB do Arduino. No entanto, o protocolo MIDI foi escolhido por ser um protocolo já utilizado na área de Música e por permitir utilizar softwares gratuitos de geração sonora.

Com o intuito de se verificar a funcionalidade dos sensores de entrada para cada tipo de instrumento musical, foram utilizados diferentes tipos de sensores para cada DMI:

- Teclado: Botões, Piezoelétrico e Capacitivo;
- Xilofone: Botões, Piezoelétrico e Capacitivo;
- Bateria: Botões, Piezoelétrico e Capacitivo;
- Teremin: Circuito Oscilador e Amplificador operacional;
- Harpa: Laser e Sensor de Luminosidade;
- Controlador MIDI: Botões, Piezoelétrico, Potenciômetro, Sensor de Proximidade.

Além destes materiais apresentados, foram utilizadas as ferramentas básicas para trabalhar com os circuitos elétricos como protoboards, fios de cabo de rede, fios maleáveis, placas perfuradas, ferro de solda, resistores e capacitores. Os processos realizados para a etapa de conceptualização do Objeto 1 são apresentados na Figura 15.

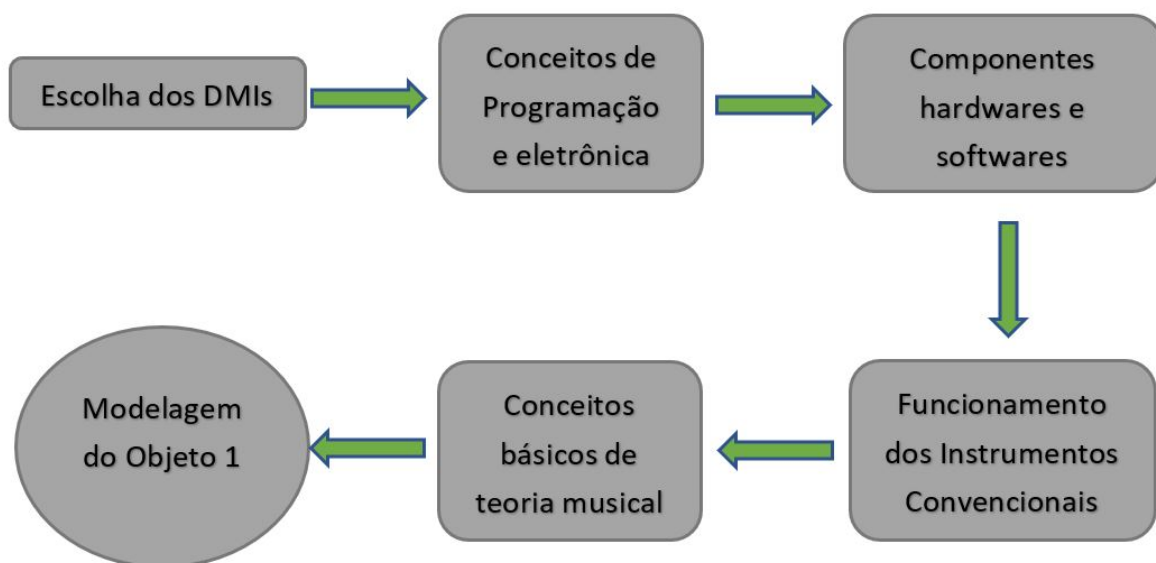


Figura 15 – Etapas da Conceptualização do Objeto 1

Fonte: O autor, 2020

enviam os dados para um outro equipamento que possui a capacidade de geração sonora. Em um computador o som é gerado por meio de um software capaz de traduzir os comandos MIDI em sons, conhecidos como *Digital Audio Workstation* (DAW) (CHAN, 2019).

4.2.2 Modelagem do Objeto 1

A modelagem do Objeto 1 trata-se dos desenhos técnicos de cada DMI idealizado. Nas Figuras 16, 17, 18, 19, 20, 21, 22 e 23 são apresentados os desenhos com todos os componentes que foram idealizados para construir os DMIs de baixo custo.

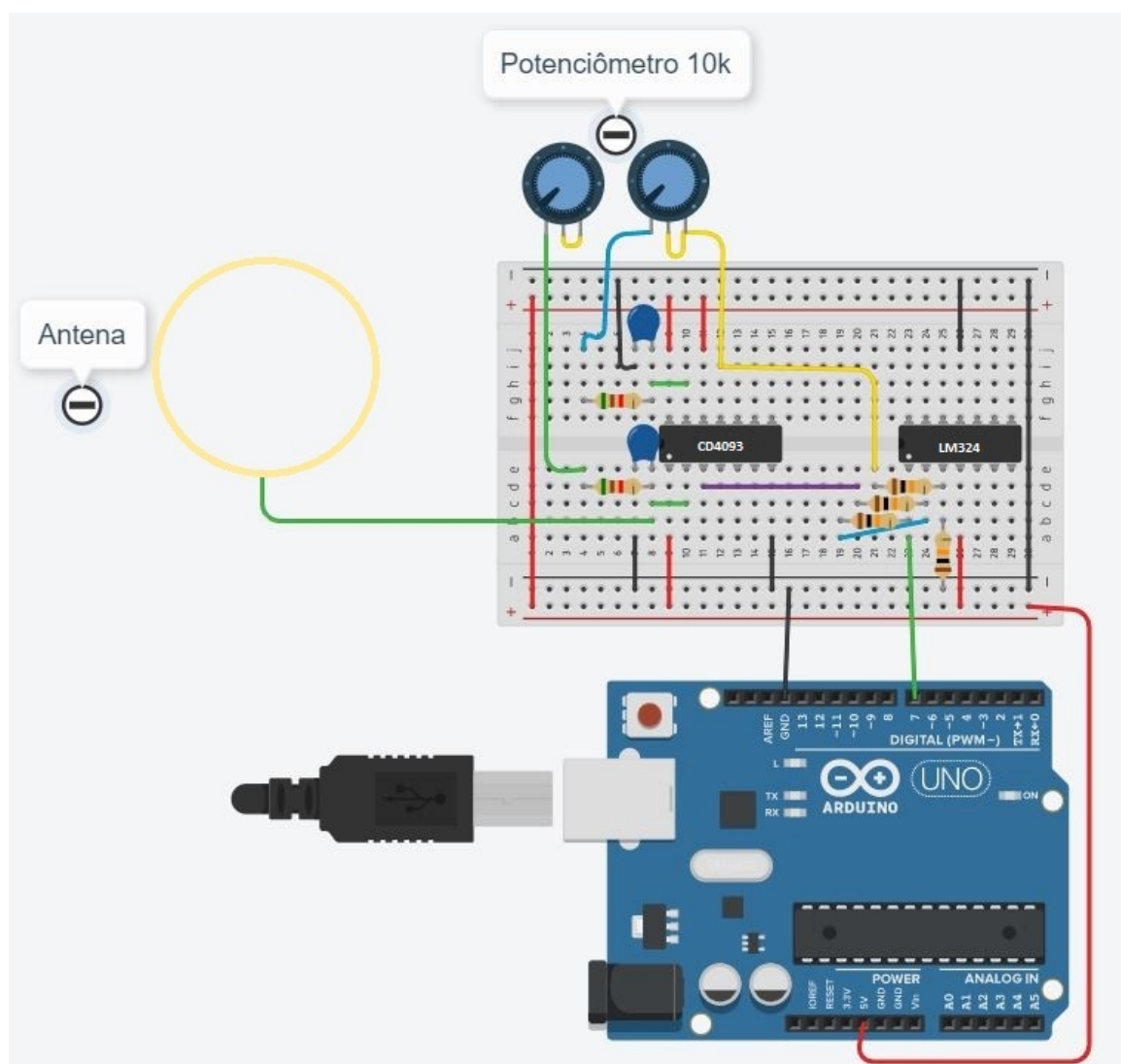


Figura 16 – Circuito Teremim na Protoboard

Fonte: O autor, 2020

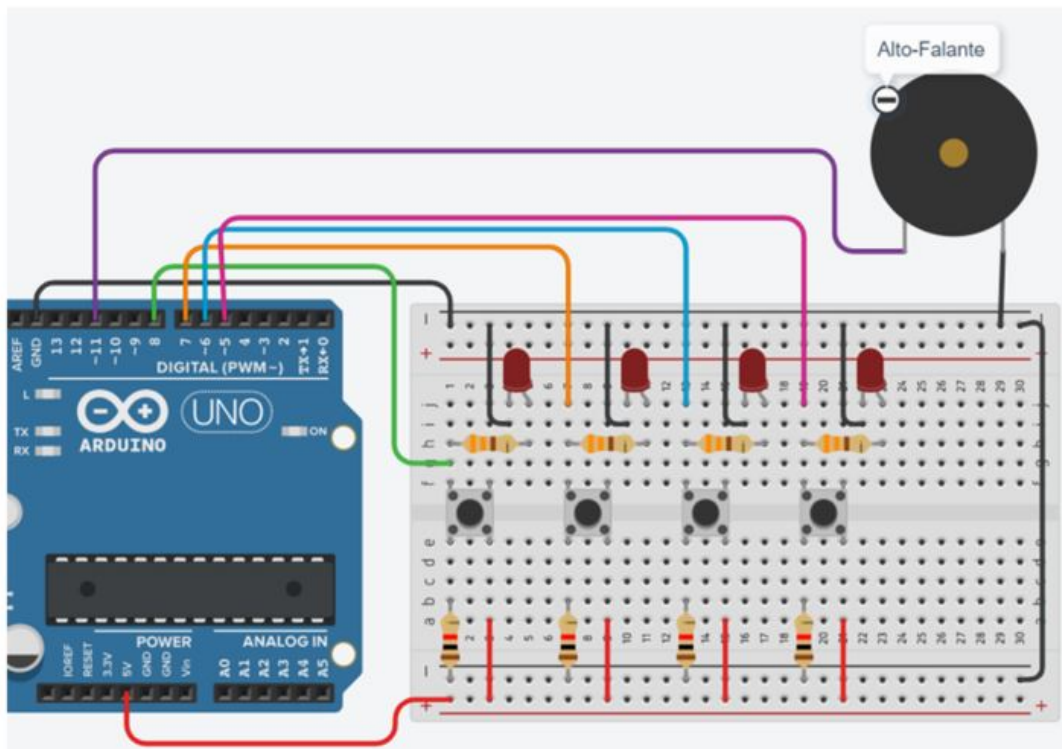


Figura 17 – Circuito Teclado/Xilofone/Bateria de Botão

Fonte: O autor, 2020

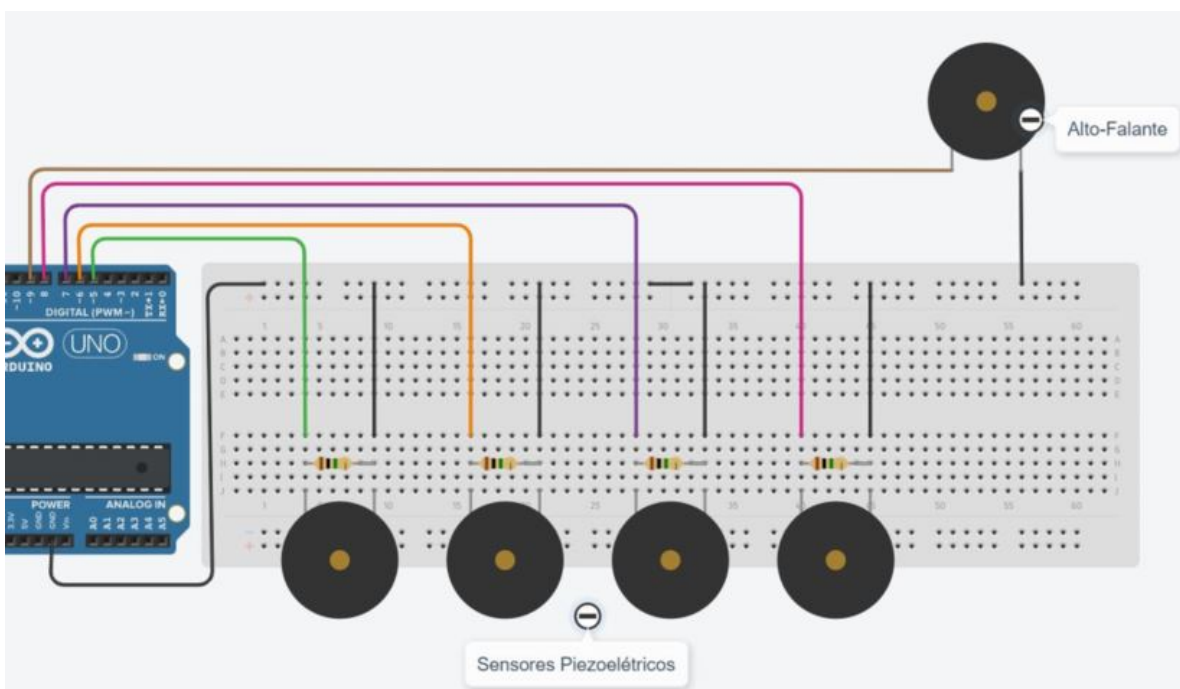


Figura 18 – Circuito Teclado/Xilofone/Bateria com Piezoelétricos

Fonte: O autor, 2020

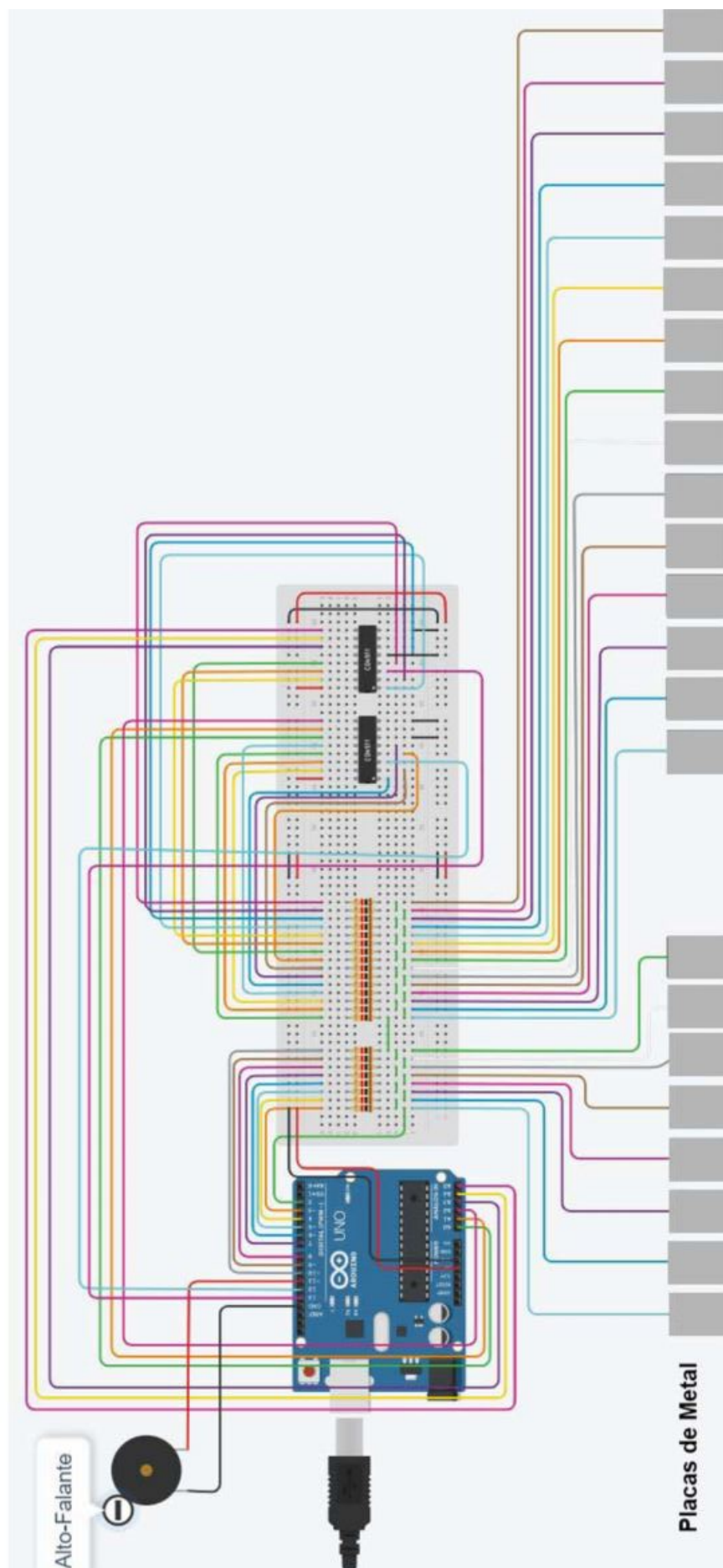


Figura 19 – Circuito Teclado/Xilofone com Sensores Capacitivos

Fonte: O autor, 2020

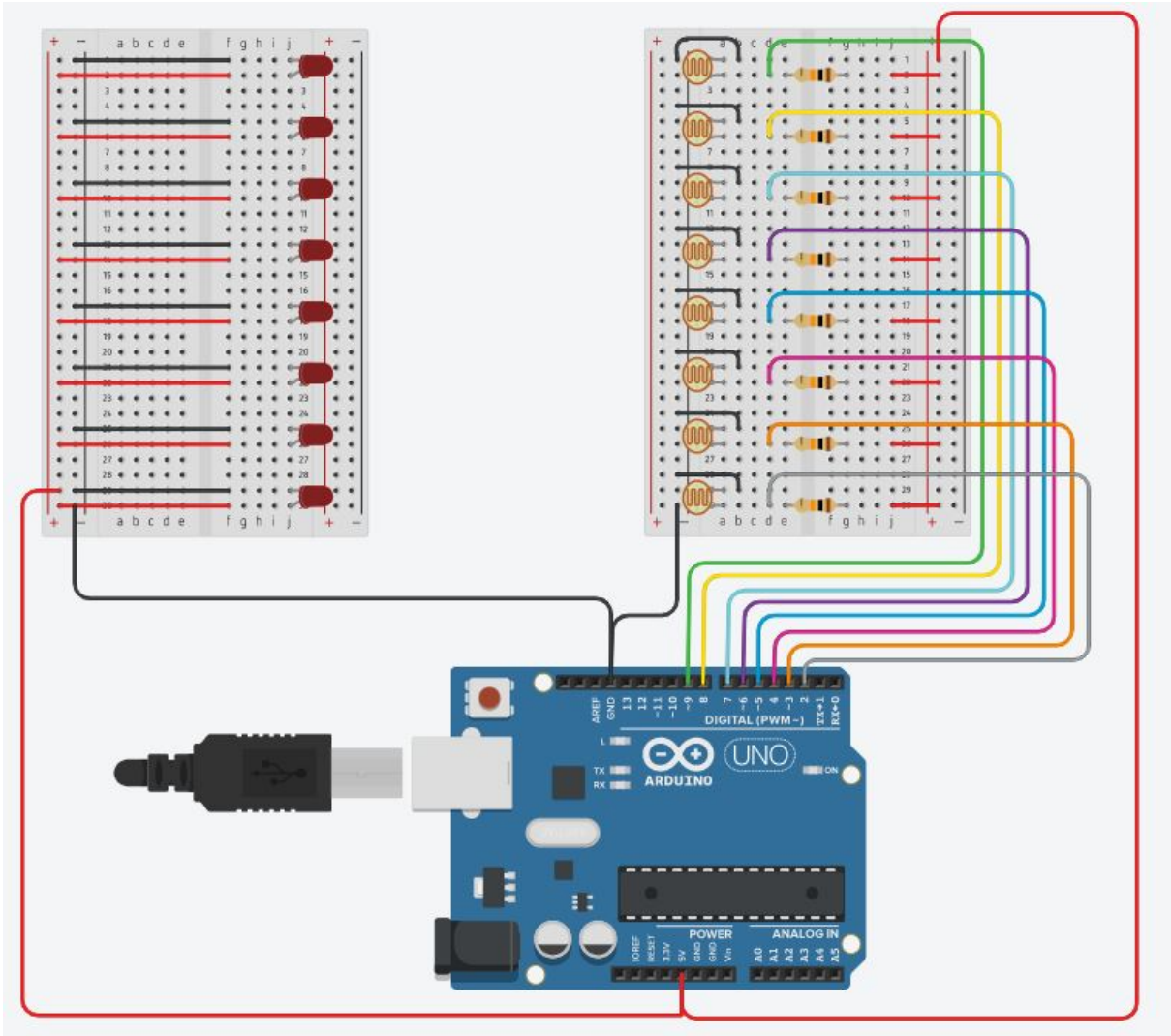


Figura 20 – Circuito Harpa Laser na Protoboard

Fonte: O autor, 2020

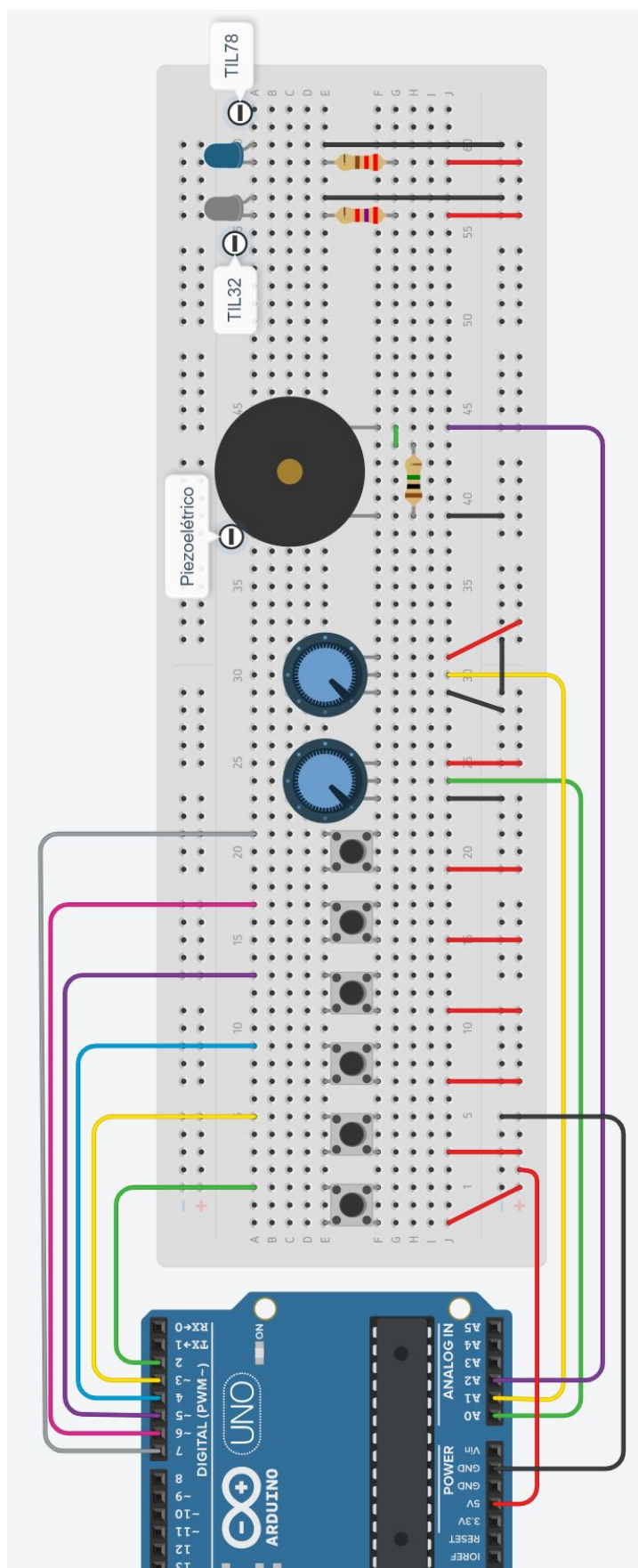


Figura 21 – Circuito Controlador MIDI na Protoboard

Fonte: O autor, 2020

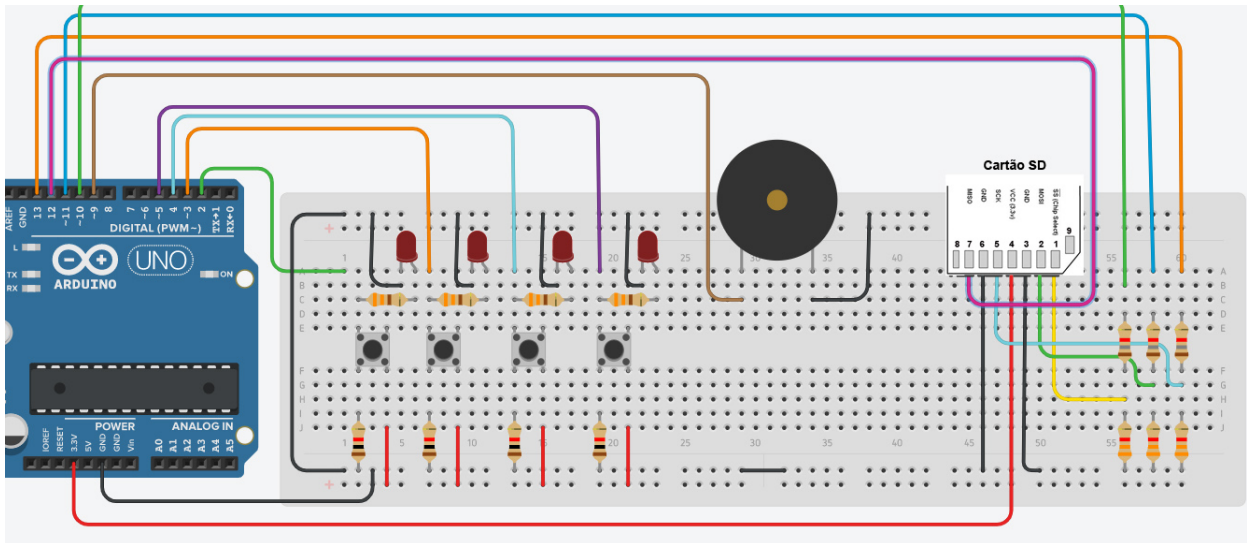


Figura 22 – Circuito Teclado com Cartão SD na Protoboard

Fonte: O autor, 2020

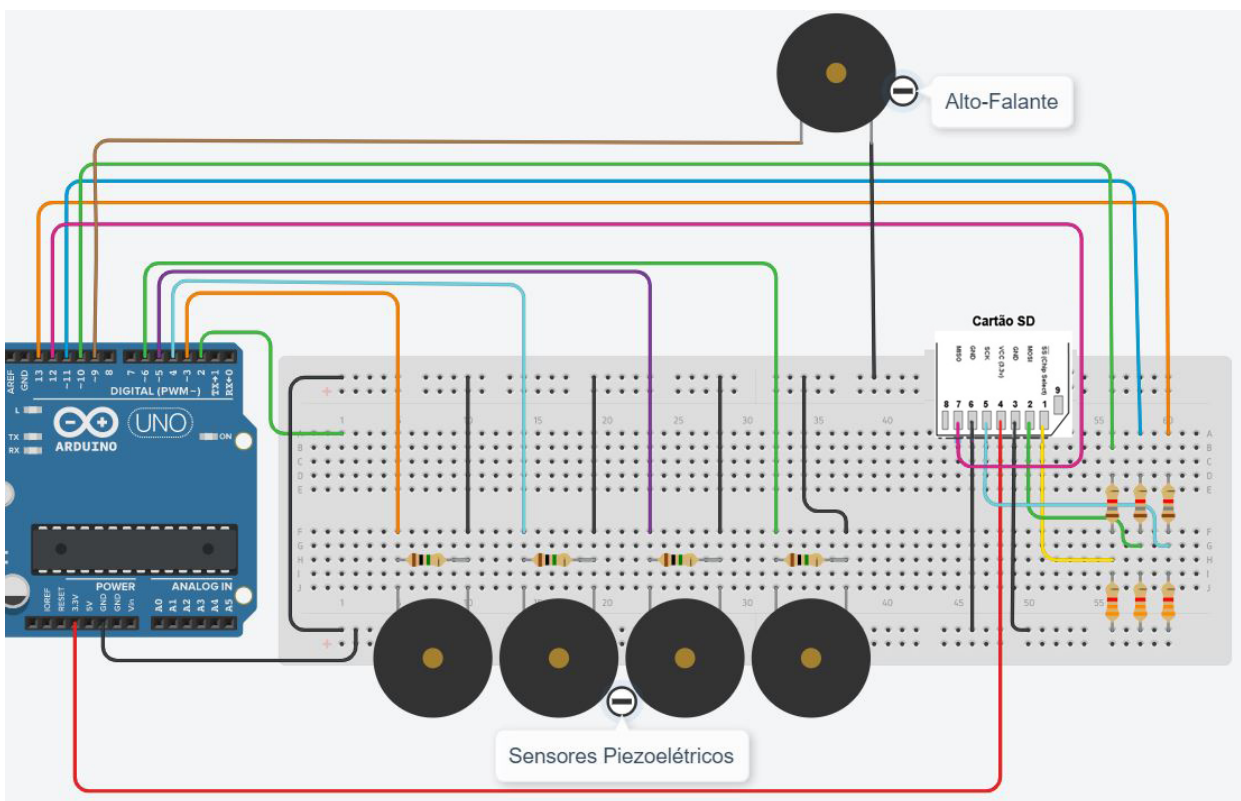


Figura 23 – Circuito Xilofone e Bateria com Cartão SD na Protoboard

Fonte: O autor, 2020

4.3 Preparação do Objeto 1

Nesta etapa, foram criadas as estratégias de realização para a construção de DMIs de baixo custo, bem como a avaliação dessas estratégias e a escolha dos processos para produção. Assim, são apresentados os procedimentos adotados para alcançar o primeiro objetivo específico desta pesquisa - Avaliar a viabilidade de construção de DMIs de baixo custo.

A estratégia elaborada foi de construir seis DMIs com diferentes componentes e avaliá-los. Para isso, as etapas idealizadas foram de analisar as principais características e como funcionam os instrumentos musicais convencionais o Teclado/Piano, Xilofone, Bateria, Harpa e Teremim, bem como o seu DMI equivalente encontrado na literatura. A primeira característica levada em consideração foi o sensor de entrada do DMI e como é o funcionamento do instrumento musical convencional equivalente.

No teclado os sensores de entrada são as teclas. Tal componente recebe gestos físicos que enviam os comandos para ligar ou desligar uma nota. Já a bateria possui sensores de pressão que recebem impactos, enquanto a guitarra e o baixo possuem cordas, mas necessitam de captadores para detectar o movimento de cada corda. Assim, verifica-se que o teclado e a bateria, em comparação com a guitarra e o baixo, apresentam funcionamento mais simples de ser compreendido, por este motivo foram pesquisados o funcionamento do teclado e bateria inicialmente.

O teclado é um dos instrumentos musicais mais conhecidos, é um DMI que possui diversos tamanhos com quantidade de teclas que variam de acordo com o fabricante, na Figura 24 é apresentado um teclado do fabricante Yamaha. São semelhantes a um piano, exceto pelo fato que o piano utiliza cordas para gerar o som enquanto o teclado gera os sons digitalmente, permitindo também adicionar efeitos, ajustes de volume, diferentes timbres, além de apresentar um tamanho mais compacto em relação ao piano (YAMAHA, 2020a).



Figura 24 – Teclado Yamaha PSR-EW410

As notas musicais são basicamente frequências de vibração, e cada tecla possui uma nota musical (Equipe Descomplicando a Música, 2020). Na Figura 25 é possível verificar como as notas são organizadas em um teclado. As teclas brancas indicam as notas naturais Dó, Ré, Mi, Fá, Sol, Lá, Si e as teclas pretas indicam as notas Dó#, Ré#, Fá#, Sol# e Lá#. O símbolo de “#” em cada nota representa uma alteração, conhecida como sustenido, trata-se de uma nota com uma frequência intermediária, podem também ser representadas pelo símbolo "b"bemol indicando as notas Réb, Mib, Solb, Láb, Sib. Estas formas de representar as teclas pretas são utilizadas para diferenciar quando se está tocando de forma sequencial da esquerda para a direita (sustenidos) ou da direita para a esquerda (bemol) (TESSMANN, 2018).

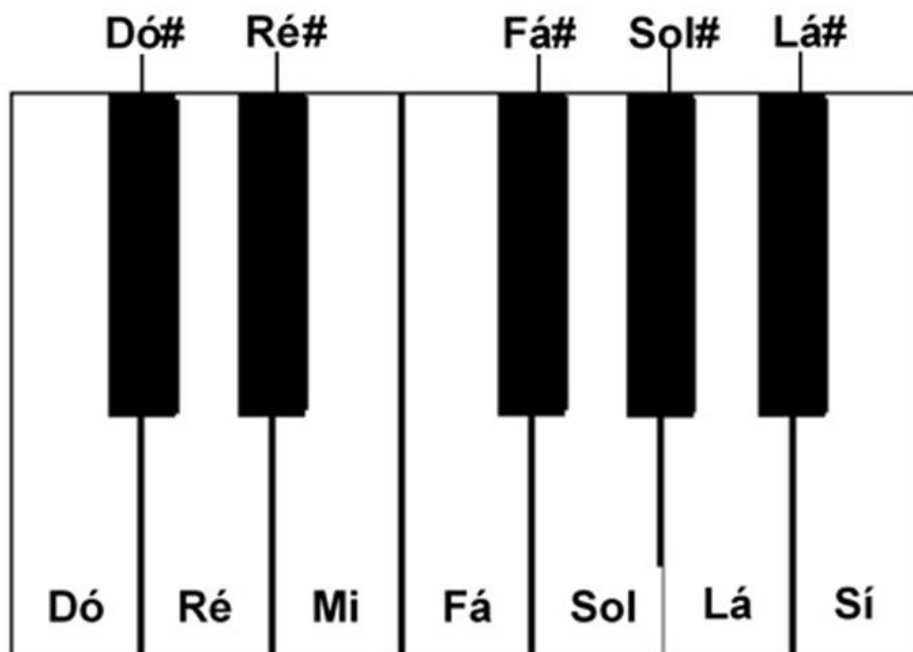


Figura 25 – Organização das Notas em um Teclado

Fonte: O autor, 2020

Deste modo, o teclado possui o funcionamento básico bastante intuitivo. No entanto para que o DMI a ser construído seja mais parecido com um teclado convencional, foram escolhidas algumas características que podem ser configuradas:

- Ao pressionar uma tecla uma nota será emitida com o timbre de piano;
- O som de uma nota permanece tocando continuamente enquanto a tecla é pressionada;
- O som de uma nota se encerra segundos após uma tecla ser pressionada;
- Quando mais de uma tecla for pressionada ao mesmo tempo serão gerados os sons de cada uma das notas.

Uma bateria é um instrumento musical acústico formado por um grupo de instrumentos de percussão⁴ posicionados de forma que possam ser tocados por uma única pessoa por meio de baquetas, geralmente construídas de madeira. A Figura 26 apresenta um kit de bateria com *Floor Tom*, *Crash Cymbal*, *Tom-Toms*, *Ride Cymbal*, *Snare*, *Hi-Hat*, *Bass* e *Cowbell*. As posições de cada parte do kit de bateria podem variar dependendo do gênero musical, bem como a quantidade componentes. A composição básica de um kit de bateria convencional contém principalmente um tambor grande, um tambor pequeno e um prato (YAMAHA, 2020b).



Figura 26 – Bateria Convencional com 8 partes

Fonte: (LAURITSEN, 2014)

A bateria digital (Figura 27) possui diversas peças denominadas de pad que se assemelham aos instrumentos de percussão da bateria convencional, geralmente são construídas com almofada de borracha ou tecido para receber impacto. Em cada bateria pode variar o número de pads, pois cada pad gera um som característico. Além dos pads a bateria digital contém um módulo que recebe os sinais enviados pelos pads.

⁴ Instrumentos de percussão são aqueles que produzem sons quando são percutidos, ou seja quando recebem algum impacto (HASHIMOTO et al., 2003).



Figura 27 – Bateria Digital com 10 pads e módulo.

Fonte: <https://www.sweetwater.com/insync/electronic-drums-buying-guide/> - acesso em 20/04/2020

Uma bateria digital possui pelo menos um sensor em cada pad que envia um sinal elétrico para o módulo. Alguns pads possuem mais de um sensor, o que permite gerar mais de um som. Por exemplo, um pad poderia gerar o som de um sino e de um prato dependendo da posição em que o pad recebe uma batida. Este módulo recebe o sinal elétrico do pad e o converte para um som pré-configurado pelo usuário, gerando assim um som para cada pad, podendo emitir sons de bumbo, caixa, tons, chimbau, entre outros. (SWEETWATER, 2020).

Contudo, o funcionamento básico da bateria digital depende de um pad receber um impacto para que o sensor envie um sinal para o módulo, e então converter um sinal elétrico em um som pré-configurado. Assim, para construção de um DMI com as características de uma bateria foram escolhidas as configurações:

- A bateria deve conter 6 pads;
- O som deve ser gerado com diferentes níveis de intensidade de impacto;
- Deve ser emitido sons de 2 pads quando tocados simultaneamente.

O xilofone é um instrumento musical de percussão que utiliza-se baquetas para tocar cada nota. Geralmente este instrumento musical é construído com uma sequência de placas de madeira dispostas de modo análogo às teclas de um piano. Os xilofones mais

simples (Figura 28) encontrados comercialmente, possuem apenas as notas equivalentes às teclas brancas (notas naturais) de um piano/teclado, diferente dos xilofones mais sofisticados que, conforme mostra a Figura 29, possuem também o equivalente às notas pretas (sustenidos/bemol) de um piano/teclado.



Figura 28 – Xilofone Simples

Fonte: <https://kforyou.com.br/loja/xilofone-de-madeira/> - acesso 20/04/2020



Figura 29 – Xilofone mais complexo

Fonte: <https://www.amazon.com/Stagg-XYLO-SET-37-HG-Professional/dp/B00PMDEVVE> -

acesso 20/04/2020



Figura 30 – Xilofone Digital

Fonte: <http://www.wernick.net/xylosynth/bubinga-xylosynth> - acesso em 20/04/2020

O xilofone digital (Figura 30) possui o formato de um xilofone mais complexo, porém a geração sonora é realizada por um sintetizador interno ou por um dispositivo externo conectado à porta MIDI. Assim, para construção de um DMI com as características de um xilofone foram escolhidas as configurações:

- Ao receber um impacto o som será emitido com o timbre de xilofone;
- O xilofone deve conter 6 notas;
- O som deve ser gerado com diferentes níveis de intensidade de impacto;
- Deve ser emitido sons de 2 notas quando tocadas simultaneamente.

Observou-se que o xilofone possui semelhanças entre o teclado e bateria em relação as suas funcionalidades. O xilofone possui as notas naturais em sequência assim como teclado/piano, mas que para ser tocado necessita de um impacto como a bateria. Por estes motivos as configurações escolhidas são combinações das configurações do teclado e bateria.

Outro DMI escolhido para ser prototipado foi a harpa laser⁵ inspirada nos trabalhos de Magun, Gerrero e Jiménez (2006), Wiley e Kapur (2009) e Sawyer et al. (2013). A harpa é um instrumento musical que utiliza cordas para cada nota musical.

Apesar de ser construída a partir de cordas, a harpa convencional (Figura 31) possui a configuração das notas musicais semelhantes as teclas brancas de um teclado (notas naturais), mas não possuem cordas específicas para as teclas pretas (sustenidos/bemol). Algumas harpas possuem pedais que regulam o quanto uma corda é esticada, formando os sustenidos/bemol (PHILHARMONIA, 2020). Diferentemente, uma Harpa Laser (Figura 32) utiliza lasers que simulam as cordas de uma Harpa convencional.

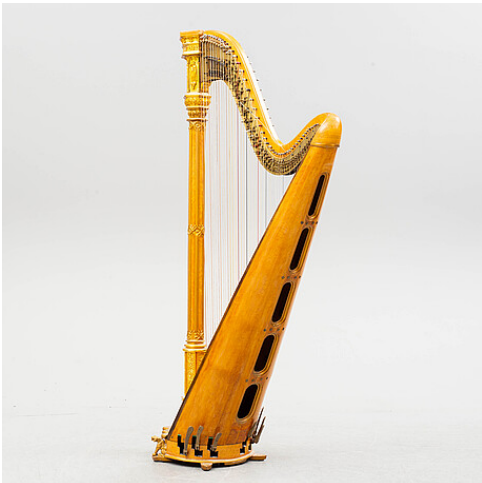


Figura 31 – Harpa

Fonte:
www.bukowskis.com/en/lots/1182352-harpa-erard-paris acesso em 20/04/2020



Figura 32 – Harpa Laser

Fonte: (SAWYER et al., 2013)

O funcionamento da harpa laser utiliza sensores de luz que possuem a capacidade de identificar a intensidade luminosa emitida pelos lasers. Desta forma, o laser e o sensor de luz trabalham em conjunto formando uma “corda de laser” para a harpa. Assim, quando um laser é interrompido de iluminar o sensor de luz, um sinal é enviado para uma unidade de mapeamento e uma nota pode ser tocada para cada “corda de laser” (MAGUN;

⁵ Laser é uma sigla em inglês para “*Light Amplification by Stimulated Emission of Radiation*” (COLWELL, 2003), em português: “Amplificação da Luz por Emissão Estimulada de Radiação”

GERRERO; JIMÉNEZ, 2006; WILEY; KAPUR, 2009). Assim, para construção de um DMI com as características de uma harpa laser foram escolhidas as configurações:

- Ao interromper um laser deve ser emitido o som com o timbre de harpa;
- A harpa deve conter 6 notas;
- O som de uma nota se encerra segundos após a interrupção do laser;
- Deve ser emitido sons de 2 notas quando tocadas simultaneamente.

O teremim (Figura 33) é um instrumento musical eletrônico que não necessita de contato físico para ser tocado. Este instrumento possui duas antenas, uma controla as notas musicais e outra controla o volume (COLWELL, 2003). Para tocar um teremim é necessário apenas aproximar uma das mãos até a antena, isso altera a frequência emitida pelo instrumento gerando uma nota.



Figura 33 – Teremim – Moog Music Etherwave Plus Theremin

Fonte: <http://www.thereminworld.com/moog-music-etherwave-plus-theremin> - acesso em 21/04/2020

O teremim não tem marcações para cada nota musical, o seu funcionamento depende da aproximação da mão. Quando se aproxima a mão da antena o som gerado será mais agudo e quanto mais afastado da antena mais grave. A antena para o volume possui o mesmo funcionamento, reduzindo o volume ao aproximar a mão e aumentando quando se afasta. Assim, para construção de um DMI com as características de um teremim foram escolhidas as configurações:

- Deve conter duas antenas, uma para o volume e outra para as notas musicais;

- Ao aproximar a mão da antena das notas gerar uma nota mais aguda e quando afastar gerar uma nota mais grave;
- Ao aproximar a mão da antena de volume reduzir o volume e quando afastar a mão da antena aumentar o volume;
- O timbre deve ser semelhante ao de um teremim eletrônico.

Seguindo a abordagem de representação de um DMI proposta por [Miranda e Wanderley \(2006\)](#), os três principais componentes do DMI são: sensores para converter gestos de entrada, mapeamento que determina o que é feito com o sinal de entrada e unidade de geração sonora que converte o sinal mapeado em sons. Desta forma, a segunda etapa da preparação trata-se da definição dos componentes dos DMIs.

Para determinar os sensores que serão utilizados nos DMIs, antes é necessário definir como serão tocados fisicamente. Para isso, os tipos de gestos físicos definidos para construção de cada DMI foi baseada nos instrumentos musicais convencionais:

- Teclado: pressão com dedos das mãos;
- Bateria: pressão com baquetas;
- Xilofone: pressão com baquetas;
- Harpa: movimento das cordas com os dedos das mãos;
- Teremim: movimento das mãos.

Com a escolha dos gestos físicos para cada DMI, foram definidos os sensores de entrada. Um sensor é um componente que mede um estímulo externo ao seu ambiente ([KARVINEN; KARVINEN, 2014](#)). Assim, os sensores necessários para a construção dos DMIs devem ser capazes de medir os gestos físicos definidos. Além disso, devem converter uma energia física em energia elétrica para que sejam utilizados em conjunto do Arduino.

Existem vários tipos de sensores que podem ser utilizados nos DMIs e cada um é utilizado para um tipo de gesto físico realizado. Por exemplo, se o gesto físico é realizar pressão em uma superfície, então podem ser utilizados sensores do tipo: piezoelétricos, sensores de toque ou até botões. Logo, para realizar a escolha dos sensores foi levado em consideração principalmente o gesto físico que se desejava para cada DMI. Os sensores escolhidos para prototipar cada DMI são apresentados na Tabela 4.

Pode-se perceber pela Tabela 4 que foram escolhidos mais de um tipo de sensor para cada DMI. Isso se deve primeiramente à algumas configurações de sensores necessitarem de um conjunto de componentes para seu funcionamento, como é o caso da Harpa e do

Teremim. Para os outros DMIs foram escolhidos diversos sensores para a verificação da funcionalidade de cada um.

Tabela 4 – Sensores definidos para cada DMI

DMI	Sensores
Teclado	Botão; Sensor Capacitivo; Piezoelétrico.
Xilofone	Botão; Sensor Capacitivo; Piezoelétrico.
Bateria	Botão; Piezoelétrico
Harpa	Laser; Sensor de Luminosidade
Teremim	Potenciômetro; Amplificador; Oscilador, Sensor Capacitivo.

Fonte: O Autor, 2020

A unidade de mapeamento escolhida para realizar a leitura dos valores convertidos pelos sensores foi a placa Arduino. O Arduino é uma plataforma de hardware e software *open-source*, o que significa que qualquer pessoa pode construir sua própria placa Arduino (BANZI; SHILOH, 2011). O Arduino é muito conhecido por ser utilizado em projetos de robótica, porém originalmente foi desenvolvido com o intuito de ajudar estudantes de design, que não possuíam conhecimentos de programação e eletrônica, a desenvolverem projetos conectando o mundo físico ao mundo digital (GIBB, 2010).

Um dos modelos mais comuns é o Arduino UNO que possui um microcontrolador Atmega-328p. Esta placa possui 14 entradas/saídas digitais e 6 entradas analógicas. Com isso é possível utilizar até 18 sensores de entrada. A placa permite comunicação com um computador por meio do cabo USB ou pelos terminais RX e TX. Além disso possui 6 terminais com saída PWM (BANZI; SHILOH, 2011).

Outra placa escolhida foi a Arduino Nano, que possui o microcontrolador Atmega-328. Esta placa possui 22 entradas/saídas digitais sendo que 8 podem ser utilizadas como entradas analógicas e 6 podem ser utilizadas como saídas PWM. Esta placa tem a vantagem de ser mais barata e possui o tamanho reduzido, podendo ser utilizada diretamente na protoboard (NANO, 2020).

Existem diversas placas de desenvolvimento que podem ser utilizadas como unidade de mapeamento como a serie Raspberry PI⁶, Micro Bit⁷, Calliope Mini⁸, LilyPad⁹, LEGO¹⁰,

⁶ Raspberry PI: é um computador construído em uma única placa, desenvolvido para o ensino de computação <www.raspberrypi.org/products>.

⁷ Micro Bit: é uma série de placas desenvolvidas para crianças e iniciantes de programação e eletrônica <<https://www.adafruit.com/category/1009>>.

⁸ Calliope Mini: é uma placa de desenvolvimento focada para o ensino de programação e eletrônica para estudantes do ensino fundamental <<https://calliope.cc/los-geht-s/einfuehrung#yellow>>.

⁹ LilyPad: é uma placa de desenvolvimento derivada das placas Arduino, utilizada para projetos *wearables* (BUECHLEY; EISENBERG, 2008).

¹⁰ LEGO: a empresa de brinquedos LEGO possui kits de desenvolvimento que utilizam programação e eletrônica em conjunto de brinquedos LEGO como por exemplo o kit LEGO MINDSTORMS <<https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-core-set/5003400#lego-mindstorms-education-ev3>>.

entre outras. No entanto há algumas vantagens em se utilizar as placas Arduino como baixo custo, fácil usabilidade, confiabilidade no desenvolvimento, menor consumo de energia, maior facilidade de utilizar as portas de entrada/saída sem adaptações (MONK, 2016). Além disso, a placa Arduino possui uma conexão USB com chip dedicado, o que permite à placa Arduino receber novos protocolos como o USB-MIDI para utilização do Arduino como dispositivo MIDI (DIAKOPOULOS; KAPUR, 2011).

Como unidade de produção sonora foram escolhidas a placa Arduino UNO e o Computador. Com o Arduino UNO é possível gerar notas musicais a partir dos terminais de saída PWM conectados a um alto falante. Desta forma, pode-se gerar notas musicais a partir de frequências programadas pelo Arduino utilizando a configuração apresentada Figura 34.

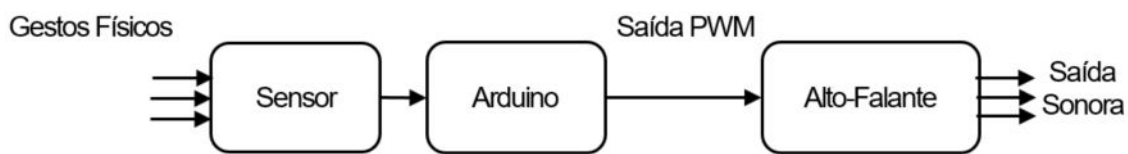


Figura 34 – Diagrama de configuração do DMI utilizando PWM

Fonte: O autor, 2020

Além disso, o Arduino UNO permite utilizar um cartão SD para armazenamento de dados, o que possibilita reproduzir sons a partir de um arquivo, como apresentado na Figura 35.

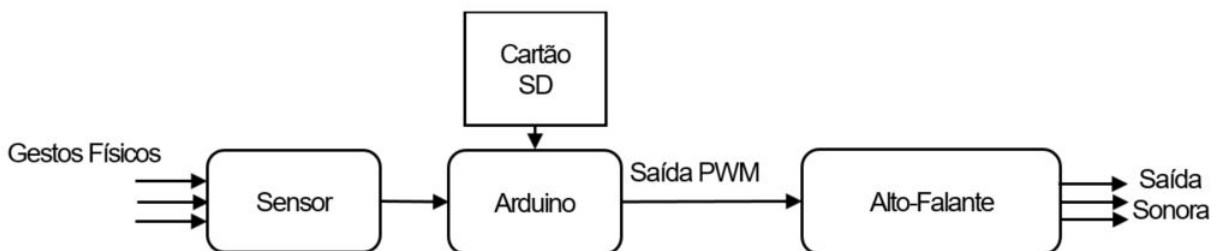


Figura 35 – Diagrama de configuração do DMI utilizando Cartão SD e PWM

Fonte: O autor, 2020

Outra possibilidade de produção sonora é utilizar o computador. A placa Arduino permite utilizar a comunicação USB para enviar e receber dados do computador. Logo, utilizando o Arduino como unidade de mapeamento é possível enviar dados para computador e tocar diferentes sons. Uma vez estabelecido que o computador pode ser fonte de geração sonora dos DMIs, torna-se viável utilizar o protocolo MIDI.

Com o protocolo MIDI pode-se fazer com que os DMIs possuam uma qualidade sonora mais próxima de um instrumento convencional do que utilizando apenas o Arduino.

O protocolo MIDI é um padrão utilizado para transmitir os dados MIDI. São pacotes de mensagens enviadas em sequência a uma taxa de 31250 bits por segundo (bps) com informações sobre a nota, volume, instrumento etc (LOY, 1985).

O protocolo MIDI não envia sons, mas sim dados. Quando uma tecla é pressionada em um teclado, é enviado um sinal elétrico para a interface MIDI, então este sinal é convertido em mensagens MIDI (LOY, 1985). Estes sinais são enviados ao computador e com auxílio de um DAW, os valores são interpretados e a nota especificada é gerada. As mensagens MIDI são representadas por valores numéricos, geralmente são formadas por três componentes, canal, evento e duração. Podem ser de dois tipos, mensagens do sistema e mensagens do canal. Mensagens como tempo, transmissão, parar, iniciar, são denominadas como mensagens do sistema, pois enviam comandos para o funcionamento do sistema MIDI. As mensagens do canal são mais comumente utilizadas, pois estão relacionadas ao funcionamento dos sons, como nota, duração, instrumento etc (LOY, 1985).

Para enviar uma mensagem MIDI deve ser informado um dos 16 canais para qual será enviado o comando, em seguida qual evento será executado e a velocidade. Para ser tocada a nota C3 no canal 1 devem ser enviados os comandos 144 (Canal 1), 48 (nota C3), 100 (duração) nesta sequência, sendo que a velocidade pode ser variada de 0 a 127. Os 16 canais podem ser utilizados para receber dados de qualquer tipo de instrumento (Teclado, Xilofone, Flauta etc), com exceção do canal 10 que é específico para sons de bateria (LOY, 1985). Os comandos de evento MIDI para ligar e desligar notas podem ser:

- Ligar canal (1 à 16): valores de 144 à 159 respectivamente;
- Desligar canal (1 à 16): valores de 144 à 159 respectivamente;
- Nota: valores de 12 à 127;
- Duração: valores de 0 à 127.

Com base nas características determinadas para construção dos DMIs, tem-se a estratégia elaborada para construção dos DMIs de baixo custo na Figura 36.

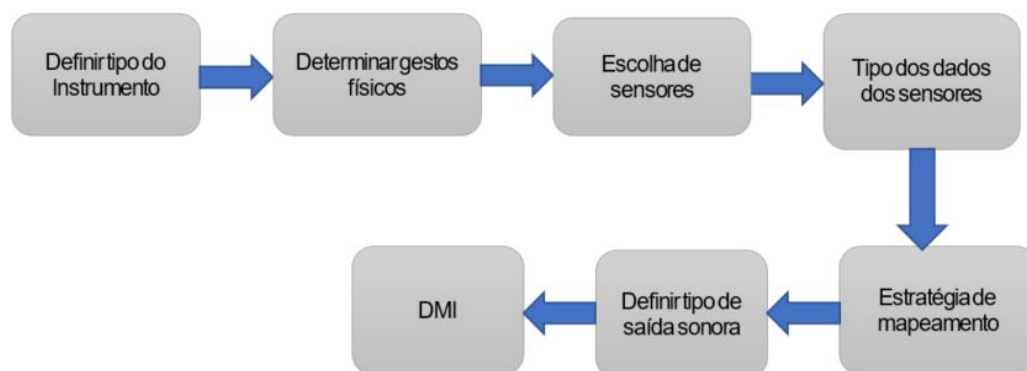


Figura 36 – Estratégia para construção de DMIs de baixo custo

4.4 Desenvolvimento do Objeto 1

Nesta etapa são apresentados os processos para construção de DMIs de baixo custo, os códigos desenvolvidos nesta etapa são apresentados no apêndice B. Esta etapa foi necessária porque somente desta forma seria possível perceber as possibilidades e dificuldades no processo, bem como ter a percepção de quais conteúdos relacionados a programação, eletrônica e música seriam necessários para construir os DMIs e que poderiam ser utilizados para o ensino dessas áreas. A seção 4.4.1 Prototipação apresenta os processos para construção dos DMIs iniciais. A seção 4.4.2 Prototipação - Adaptações e Modificações apresenta as melhorias realizadas nos DMIs anteriores, bem como a construção de outros DMIs. Na seção 4.5 - Análise de Custo e Funcionalidade dos Protótipos são apresentados os custos para construção de cada DMI e sua funcionalidade.

4.4.1 Prototipação

Inicialmente foram desenvolvidos três DMIs: Teclado, Xilofone e Bateria. Posteriormente foram construídos outros quatro DMIs utilizando outros materiais: Xilofone, Bateria, Teremim e Harpa. A construção destes DMIs foi necessária porque somente desta forma seria possível perceber o nível de dificuldade durante o processo, bem como ter a percepção de quais conteúdos relacionados a programação, eletrônica e música poderiam ser aplicados na abordagem educacional.

Com o intuito de aprender a construir os DMIs, foi construído um primeiro teclado utilizando a placa Arduino e botões para representar as notas. Este DMI (Figura 37) foi construído em protoboard, contendo 4 botões e 4 resistores de *pulldown*¹¹ e um alto falante (Figura 38) para reproduzir as notas. Também foram adicionados 4 LEDs que acendem quando uma nota era pressionada.

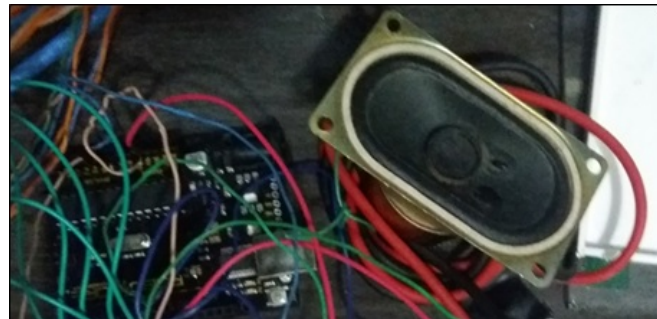
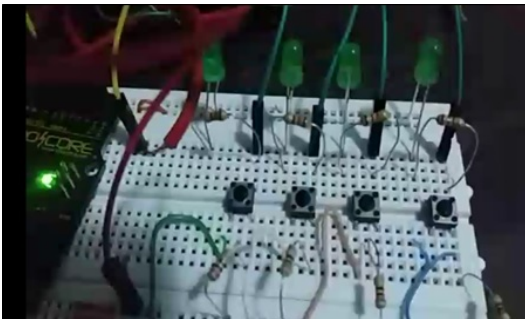


Figura 37 – Circuito Teclado de botões Figura 38 – Alto falante utilizado no Teclado de botões

Fonte: O Autor, 2017

¹¹ Resistor de *pulldown* trata-se de um resistor conectado ao GND. Quando conectado ao 5v é denominado *pullup* (BELVEDERE, 2018).

Este teclado era tocado através dos botões, quando pressionados era enviado um sinal para a placa Arduino que interpretava esse sinal identificando qual botão foi pressionado, gerando assim um sinal em determinada frequência que era enviado a uma porta da placa Arduino, a qual estava conectado o alto falante, gerando assim uma nota específica para cada botão e ainda acendendo um LED para cada botão através de um outro sinal. O som gerado então dependia da frequência de trabalho das portas da placa Arduino, gerando as notas musicais Dó, Ré, Mi e Fá.

O circuito do teclado de botão é apresentado na Figura 39. Foram utilizados resistores de 330Ω para cada LED com intuito de reduzir a corrente elétrica e por consequência reduzir o brilho dos LEDs. Para cada botão foi adicionado um resistor de $10K\Omega$ conectando ao terra (GND) para que estejam em funcionalidade de *pulldown*. Os botões e os LEDs foram conectados aos terminais digitais 5, 6, 7 e 8, bem como conectados a alimentação de 5V e GND da placa Arduino.

Posteriormente foi adicionado ao circuito anterior um cartão SD (Figura 40) o qual serviu para armazenar sons de quatro notas (Dó, Ré, Mi e Fá) de um piano. Foi necessário utilizar um adaptador de cartão SD e modificá-lo com "pinos poste" para que pudesse ser utilizado no circuito da protoboard. Isso permitiu que o timbre fosse alterado, podendo ter sons de teclado, xilofone e bateria.

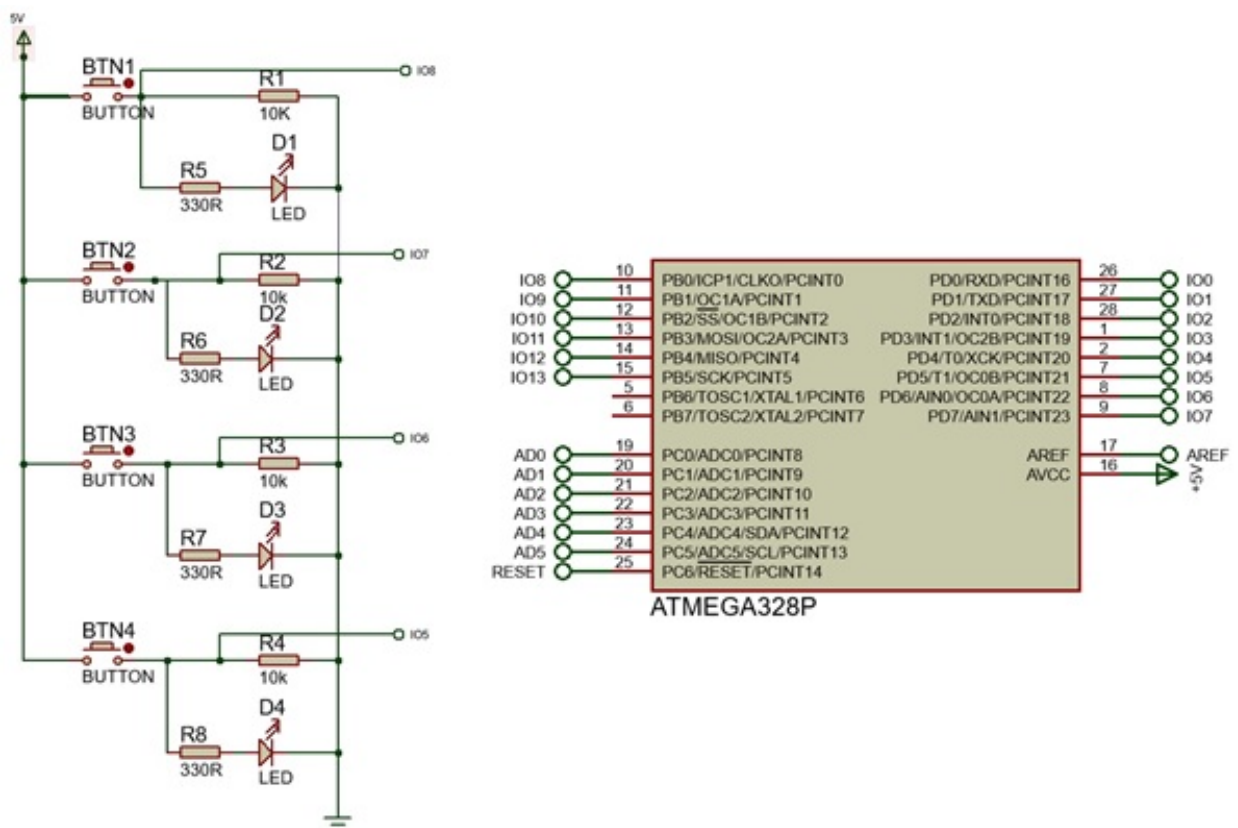


Figura 39 – Circuito Xilofone de Botões

Fonte: O Autor, 2017

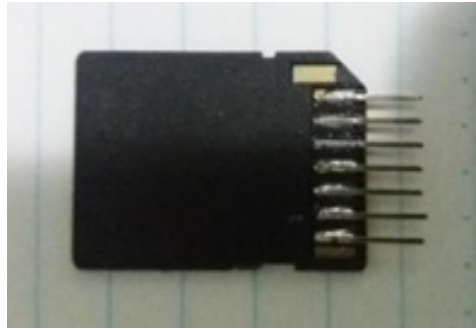


Figura 40 – Cartão SD utilizado no Xilofone de botões

Fonte: O Autor, 2017

Para se aproximar mais do xilofone convencional, os botões foram substituídos por sensores de pressão piezoelétricos, desta forma este DMI (Figura 41) funcionou como um instrumento musical de percussão podendo receber impactos. Utilizou-se 5 sensores piezoelétricos e por consequência cinco notas (Dó, Ré, Mi, Fá e Sol).

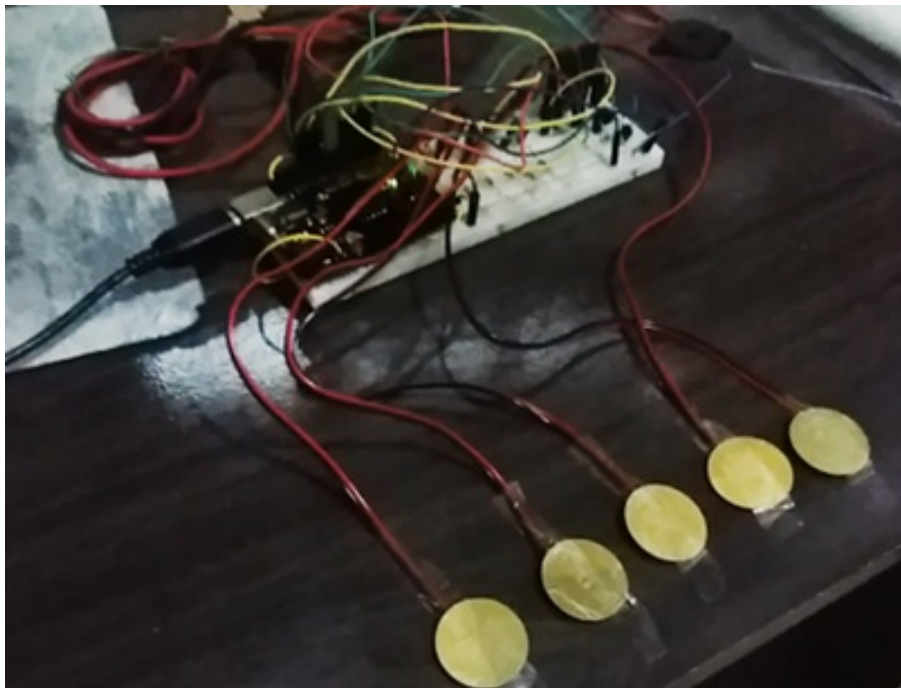


Figura 41 – Circuito Xilofone com Piezo

Fonte: O Autor, 2017

O segundo DMI construído foi um teclado com piezoelétricos. O teclado é um instrumento musical que funciona utilizando teclas, porém com a finalidade de se aprender mais sobre os DMIs e reutilizar os materiais do DMIs anteriores, foi construído utilizando piezoelétrico e cartão SD (Figura 42).

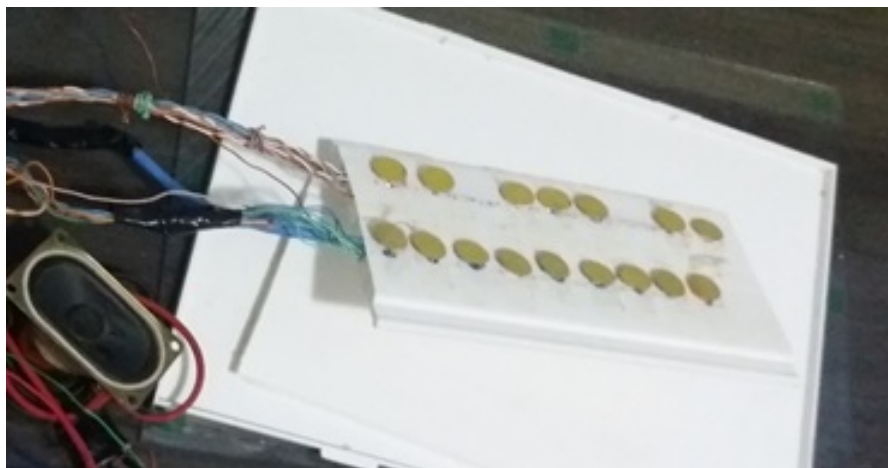


Figura 42 – Teclado com piezoelétricos e PVC

Fonte: O Autor, 2017

Os sensores piezoelétricos foram utilizados para representar cada tecla do teclado, porém os sensores necessitavam de um impacto com uma força maior em comparação com a força aplicada a uma tecla de teclado devido ao tamanho dos sensores. O pvc é um material que permite modelar a sua forma de maneira fácil, porém ao utilizar pvc com os piezoelétricos causaram vibrações ao tocar uma nota, resultando em ruídos de outras notas ao mesmo tempo. Foi construído uma segunda versão do teclado substituindo os piezos por placas de metal formando 32 notas e para fixa-las foi utilizado isopor (Figura 43).

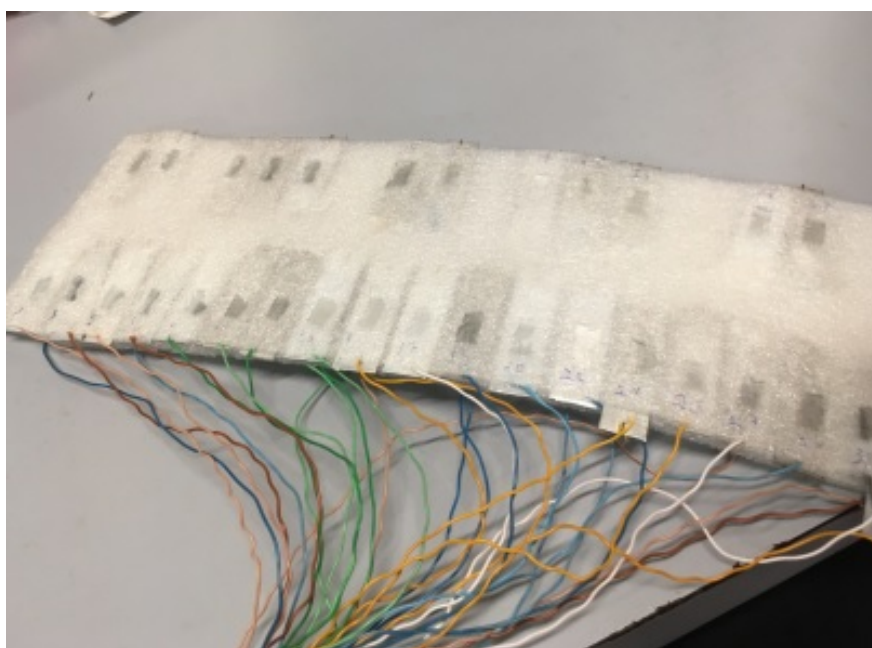


Figura 43 – Teclado capacitivo com isopor

Fonte: O Autor, 2017

Cada peça de metal funciona como sensores capacitivos que são interpretados pelo Arduino. Este DMI gera sons de teclado por meio do cartão SD. O número de teclas era muito superior o número de portas de entrada do Arduino, por este motivo foi utilizado um circuito multiplexador. Porém, tinha-se um atraso do som quando mais de uma nota era pressionada devido a lentidão na leitura do cartão SD e leitura das teclas. Estes motivos fizeram com que fosse realizada uma modificação nos DMIs acrescentando o computador como fonte de geração sonora.

4.4.2 Prototipação - Adaptações e Modificações

A primeira adaptação foi utilizar o protocolo MIDI em conjunto do computador para geração sonora. Para que seja possível enviar comandos MIDI para o computador foram necessários utilizar 3 softwares: LoopMIDI; Hairless MIDI to Serial Bridge; Virtual MIDI Piano Keyboard.

Foi utilizado o software LoopMIDI o qual cria uma porta MIDI, com um nome especificado pelo usuário, que aguarda o recebimento de dados. Com este software é possível identificar o nome da porta MIDI da forma que desejar, na Figura 44 é apresentado um exemplo de configuração do LoopMIDI com nome para a porta de Arduino.

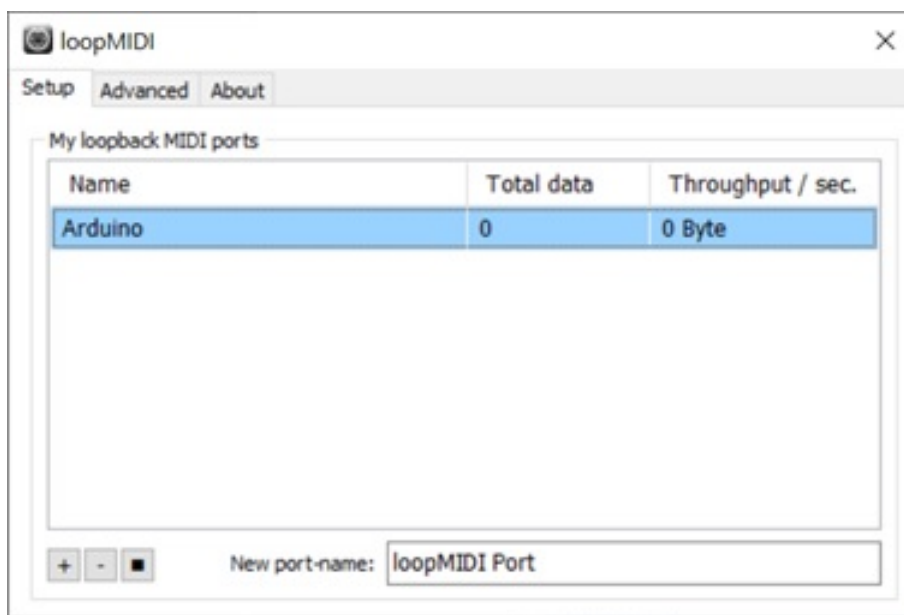


Figura 44 – LoopMIDI configurado com a porta virtual ArduinoPort

Fonte: <https://www.tobias-erichsen.de/software/loopmidi.html>

O segundo software foi o Hairless MIDI to Serial Bridge. Este software faz a ligação entre a porta serial e a porta MIDI, desta forma é possível enviar os dados da porta USB onde está conectada o Arduino e a porta MIDI virtual criada com o LoopMIDI. Na Figura 45 é apresentado a configuração do Hairless MIDI conectando a USB Serial a porta MIDI virtual com o nome Arduino.

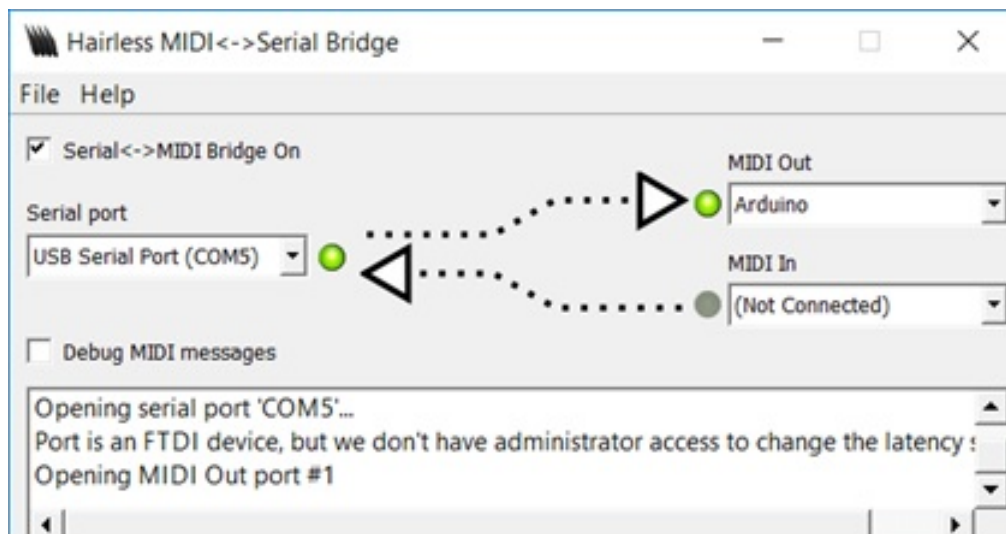


Figura 45 – Hairless MIDI Serial Bridge conectado à porta MIDI virtual

Fonte: <https://projectgus.github.io/hairless-midiserial/>

Para reproduzir os comandos MIDI foi utilizado o Virtual MIDI Piano Keyboard. Este software disponibiliza um teclado virtual o que permite tocar as notas musicais e visualizá-las em tempo real. Na Figura 46 é apresentado o Virtual MIDI Piano Keyboard configurado para tocar o instrumento *Xylophone*.

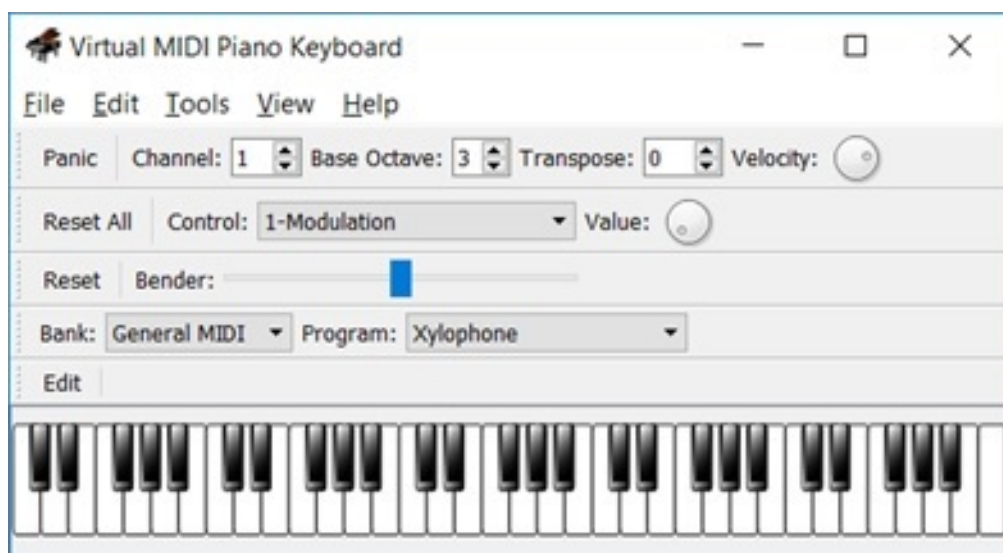


Figura 46 – Virtual MIDI Piano Keyboard

Fonte: <https://vmpk.sourceforge.io/>

Com estes três softwares foi possível construir DMIs que reproduzem sons com os timbres disponíveis no Virtual MIDI Piano Keyboard, principalmente os timbres de xilofone, teclado e bateria. Contudo, verificou-se que os computadores dos laboratórios para o curso de formação não comportavam tais softwares, por este motivo buscou-se outra

alternativa realizando uma modificação no *firmware* no chip do USB da placa Arduino UNO e acrescentando um jumper (Figura 47).

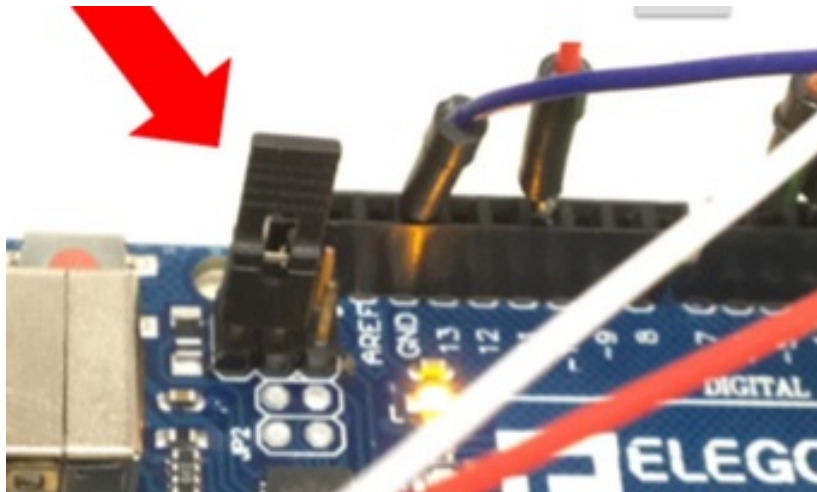


Figura 47 – Jumper da Placa Arduino

Fonte: O Autor, 2020

A placa Arduino UNO possui dois microcontroladores, o primeiro trata-se do Atmega328p que é utilizado para gravar o código e realizar as funções do Arduino, o segundo trata-se de um chip Atmega16 μ 2 utilizado para conversão serial para USB. Neste chip para USB é possível alterar seu firmware para que seja reconhecido como um dispositivo MIDI quando plugado ao computador. Para alterar o firmware do chip Atmega16 μ 2 presente no Arduino uno foi utilizado o Hiduino¹² que permite a alteração no Arduino mantendo as funcionalidades básicas.

Para utilizar o Arduino como placa de desenvolvimento foi acrescentando um jumper¹³ como mostrado na Figura 47. Quando o Jumper está conectado no Arduino, o computador identifica como sendo a placa Arduino e permite programá-lo, porém quando se retira o Jumper o computador identifica como sendo um dispositivo MIDI. Desta forma o computador está apto para receber comandos MIDI. Porém, é necessário utilizar um DAW para gerar as notas.

Para geração sonora poderia ser utilizado o Virtual MIDI Piano Keyboard como foi construído inicialmente, porém este software necessita instalação. Logo, para utilizar um software para geração sonora foi escolhido o Freepiano (Figura 48) que não necessita instalação.

¹² <https://github.com/ddiakopoulos/hiduino>

¹³ Jumper: é um conector que interliga dois pontos



Figura 48 – FreePiano utilizado para reproduzir os comandos MIDI

Fonte: <https://freepiano.tiwb.com/en/>

O FreePiano possui um teclado para tocar as notas musicais e todas as configurações MIDI necessárias. Desta forma, foi possível utilizar o computador com a placa Arduino modificada nos DMIs construídos. Além disso, os DMIs Xilofone e Bateria construídos anteriormente foram modificados utilizando materiais reutilizáveis (tampas e caixas de plástico e papelão) para esconder os sensores. Além disso, foram construídas placas para os circuitos elétricos com conectores e sensores piezoelétricos.

A bateria digital com tampas (Figura 49) foi construído buscando verificar o funcionamento das portas analógicas que permitem medir a intensidade de força aplicada pelo usuário. Foram adicionadas tampas plásticas para representarem os pads da bateria e utilizado lápis com borracha na ponta para baqueta.



Figura 49 – Bateria digital utilizando tampas de plástico

Fonte: O Autor, 2020

Da mesma forma que a bateria com tampas, foi construído o xilofone utilizando papelão no lugar das peças de madeira do xilofone convencional (Figura 50) e alterando o som gerado pelo FreePiano de bateria para xilofone.

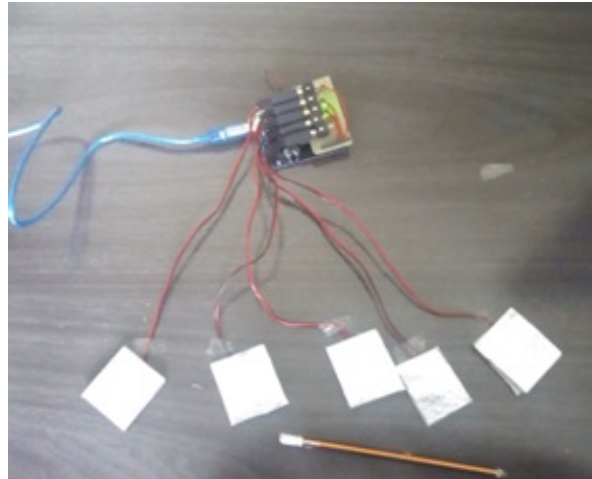


Figura 50 – Xilofone digital utilizando papelão

Fonte: O Autor, 2020

Como a Bateria Digital (Figura 49) e o Xilofone Digital (Figura 50) possuem o funcionamento semelhante de receber um impacto para tocar um som, utilizou-se o mesmo circuito (Figura 51). Com isso, foi construído uma placa com os componentes necessários e pinos poste para facilitar a montagem na placa Arduino.

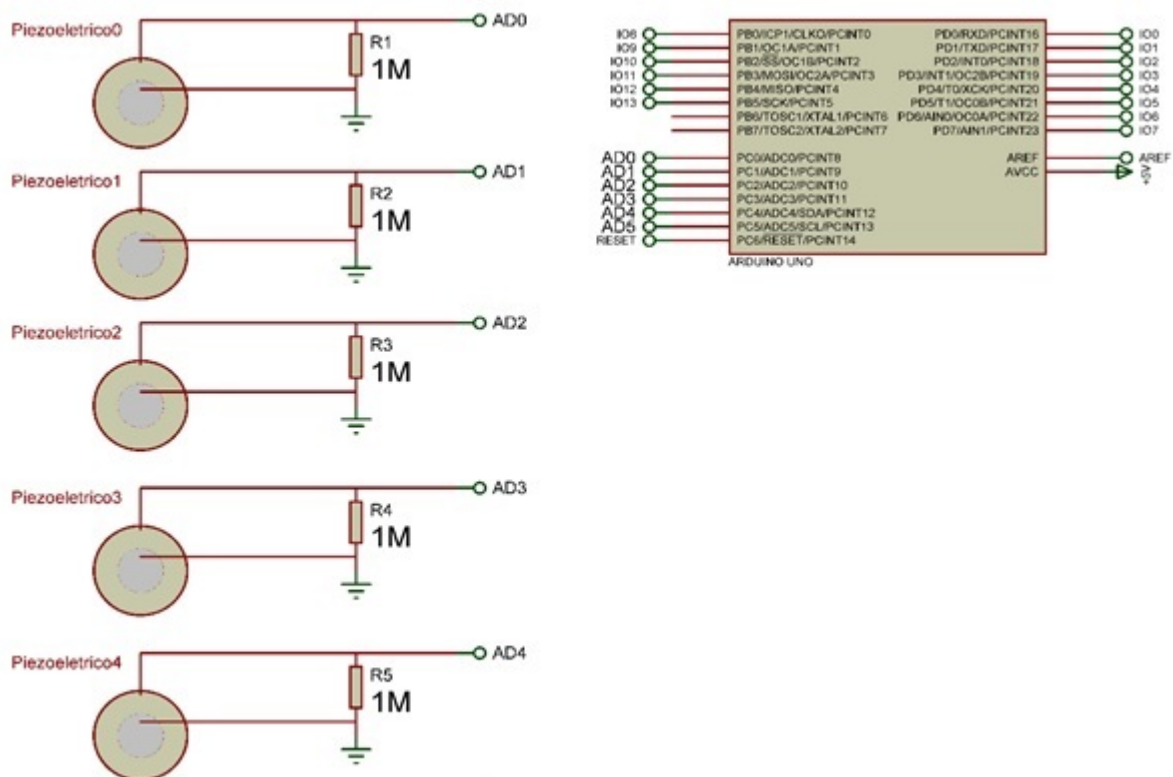


Figura 51 – Circuito Bateria/Xilofone Digital

Fonte: O Autor, 2020

O circuito Bateria/Xilofone digital (Figura 51) é formado por cinco sensores

piezoelétricos conectados às entradas analógicas do Arduino e um resistor de $1M\Omega$ com a finalidade de se reduzir ruídos. Cada piezoelétrico é conectado à uma porta analógica do Arduino (AD0, AD1, AD2, AD3, AD4), para que seja possível identificar a intensidade do impacto.

Outra modificação realizada foi a utilização de palitos de madeira para construção do xilofone com piezoelétrico (Figura 52 e 53) com a intenção deixar mais parecido fisicamente com o instrumento convencional. Os palitos de madeira foram pintados com tinta spray e os piezoelétricos foram colados com cola quente. Observa-se que é possível utilizar materiais reutilizáveis para construção dos DMIs.



Figura 52 – Xilofone com piezoelétrico e palitos de madeira

Fonte: O Autor, 2020

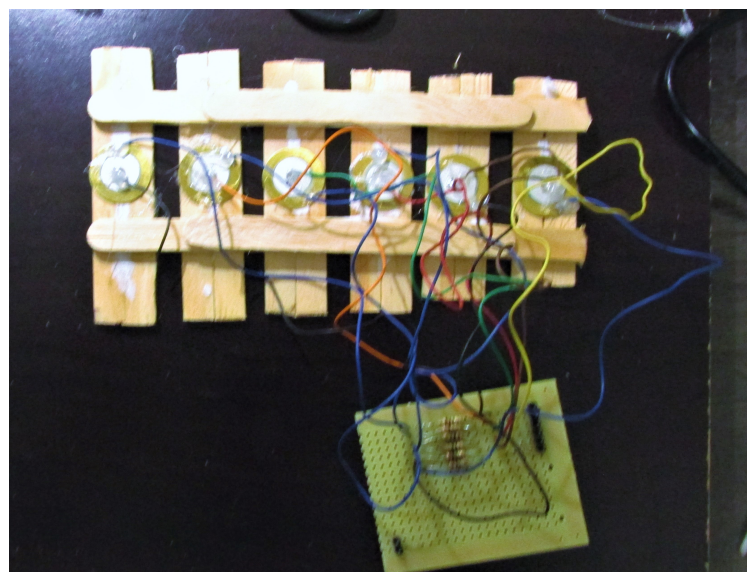


Figura 53 – Visão do circuito xilofone com piezoelétrico e palitos de madeira

Fonte: O Autor, 2020

A utilização de palitos de madeira para os instrumentos de percussão dificulta na geração sonora, uma vez que a madeira gera ruídos quando recebe impacto, da mesma maneira que as tampas de plástico. Da mesma maneira ocorre com os plásticos utilizados na bateria. Com isso, para a bateria foram adicionadas "espumas para case" reaproveitadas de caixas de componentes (Figura 54). Desta forma, o ruído gerado pelo plástico é diminuído, mas se perde um pouco da sensibilidade do sensor.



Figura 54 – Bateria com tampas de plástico e espumas

Fonte: O Autor, 2020

A construção do teremim foi baseada no teremim comercial que utiliza dois osciladores de mesma frequência, fazendo com que as antenas sofram interferências ao se aproximar delas, gerando notas musicais ou controlando o volume. No entanto, nesta etapa a antena de controle de volume não foi construída.

No teremim digital (Figura 55) foi construído os osciladores através do circuito integrado 4093, essas duas frequências foram subtraídas através de um circuito analógico utilizando opAmp Lm324, desta forma é obtido uma nova frequência que é lida pelo Arduino e reproduzido um som com base na aproximação da mão do usuário à antena (Figura 56). Este DMI no entanto, possui a desvantagem de ser necessário calibrar todas as vezes que o circuito é ligado, pois sofre interferências do ambiente. Além disso, deve-se configurar as notas na placa Arduino após a calibração. Foi utilizado o Arduino Nano, pois fica fixado à protoboard o que reduz os ruídos gerados pelo manuseio da placa.

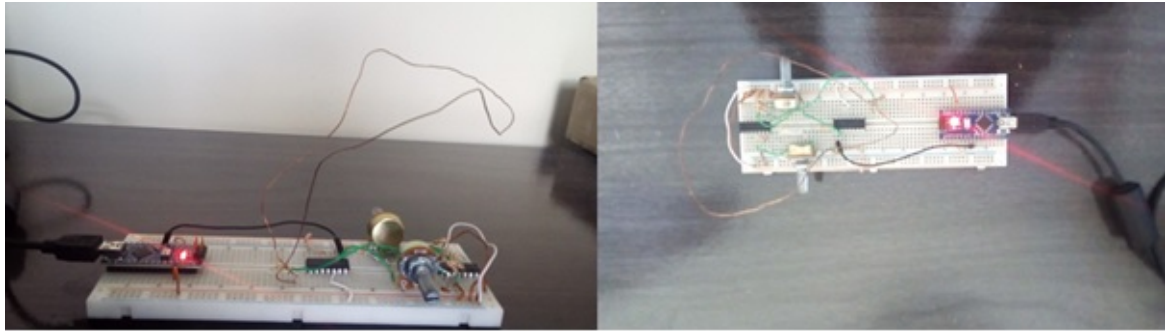


Figura 55 – Teremin Digital em protoboard

Fonte: O Autor, 2020

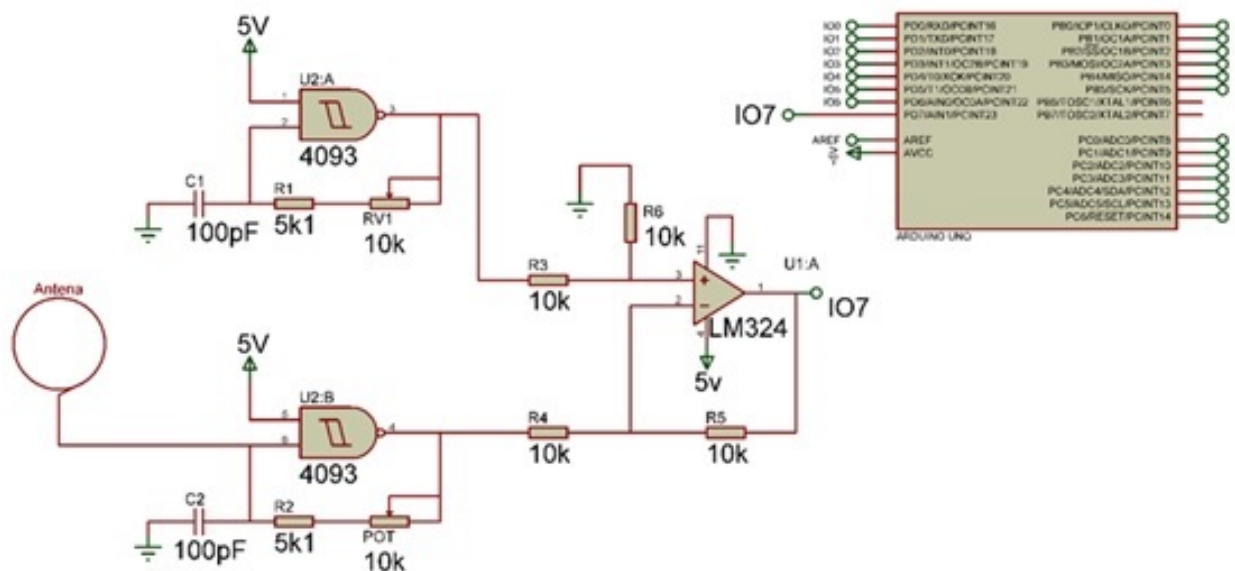


Figura 56 – Circuito Teremin Digital

Fonte: O Autor, 2020

A Harpa convencional é um instrumento que utiliza cordas para reproduzir som. Visando construir uma Harpa digital foram utilizados lasers e sensores LDR (*Light Dependent Resistor*) que simulam as cordas de uma Harpa real. Os sensores LDR são componentes resistivos que variam seu valor dependendo da quantidade de luz que é emitida sobre eles. Os LDRs utilizados são de 3mm e quando não estão recebendo nenhuma fonte de luz, estes componentes apresentam um valor de resistência de $50M\Omega$, já quando há uma luz sobre eles, variam de $5K\Omega$ e $10K\Omega$. Desta forma os lasers são alinhados aos sensores LDRs e os valores são lidos pelo Arduino.

Foram utilizados 8 lasers posicionados em um pedaço de PVC (Figura 57) e 8 sensores posicionados em uma folha de papel para representarem as cordas, por este motivo foi necessário utilizar um resistor de $10K\Omega$ juntamente com cada sensor LDR na configuração de divisor de tensão, pois foi necessário utilizar as portas digitais do Arduino.

As entradas digitais do Arduino conseguem ler valores binários, 0 para quando a tensão de entrada é menor que 3,3V e 1 para quando a tensão é maior ou igual a 3.3V. Assim, é possível identificar quando um laser é interrompido ou não.

O valor medido do LDR enquanto estava com o laser foi de $10K\Omega$ e o resistor utilizado de $10K\Omega$, portanto o divisor de tensão resulta num valor de aproximadamente 2,5V, logo o Arduino considera um valor digital de 0 quando os lasers são apontados para os sensores, fazendo com que quando a luz é interrompida a resistência do LDR aumenta obtendo assim uma tensão maior do que 2,5V a qual resulta em um valor digital 1 para o Arduino. Desta forma é determinado quando uma nota é tocada ou não enviando o comando MIDI pela porta Serial. O circuito da Harpa Laser é apresentado na Figura 58.

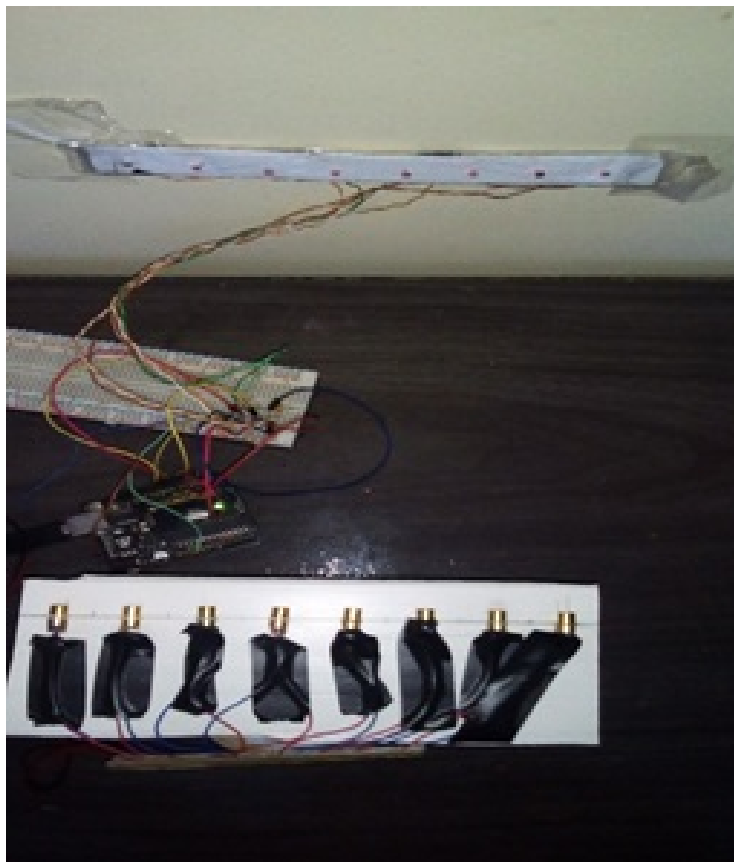


Figura 57 – Harpa Laser teste 1

Fonte: O Autor, 2020

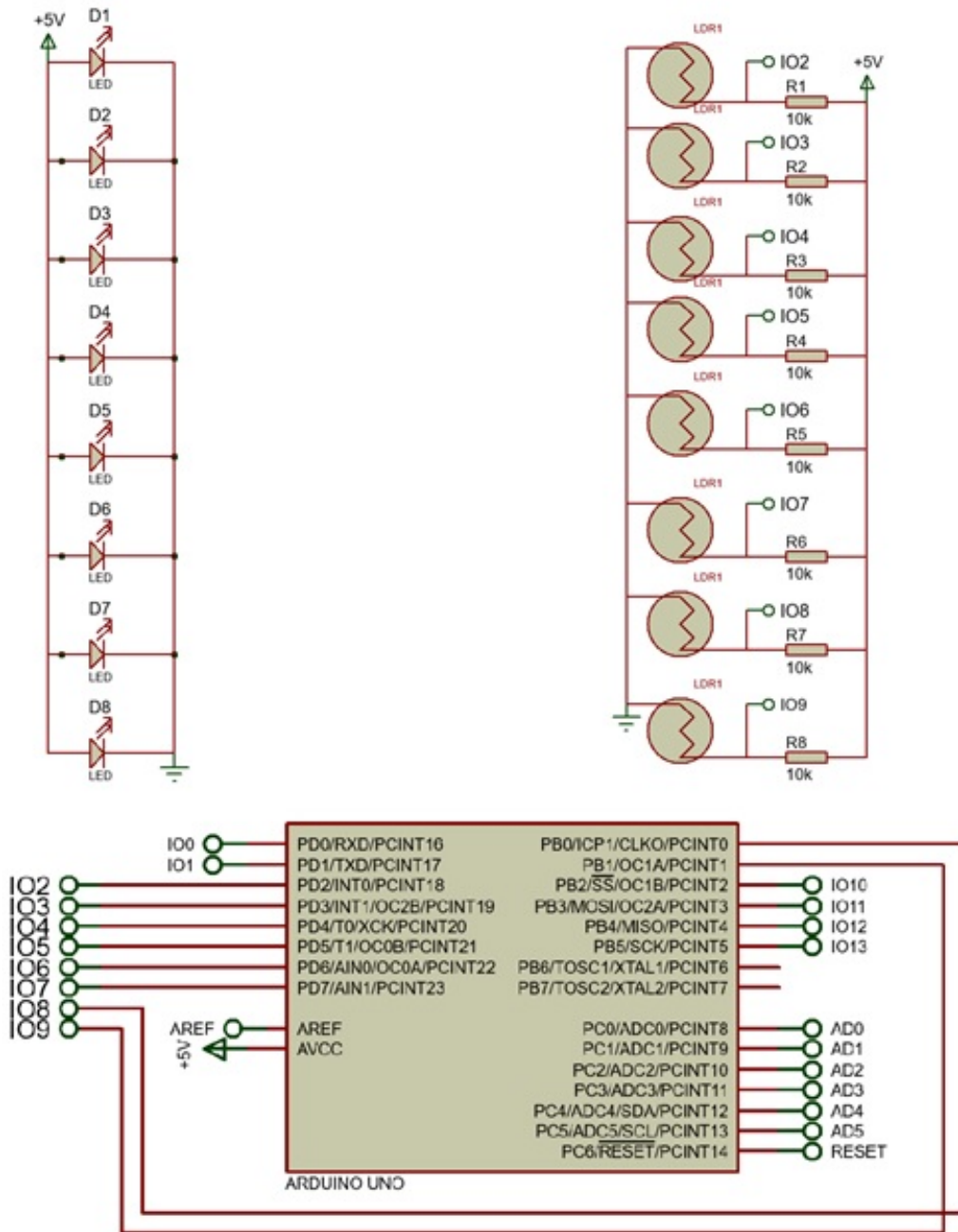


Figura 58 – Circuito Harpa Laser

Fonte: O Autor, 2020

Após realizar os testes com os circuitos, a harpa laser foi construída utilizando um quadro (Figura 59). Os LDR foram dispostos de um lado do quadro (Figura 60) apontados para os lasers posicionados do outro lado (Figura 61). Além disso foi construída uma placa (Figura 62) com os componentes necessários para utilizar como shield na placa Arduino.

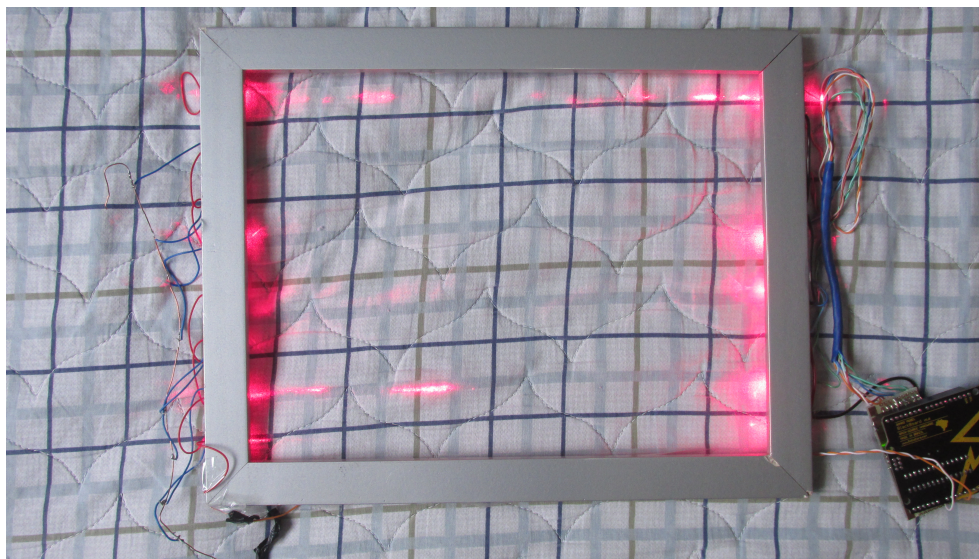


Figura 59 – Quadro da Harpa Laser

Fonte: O Autor, 2020

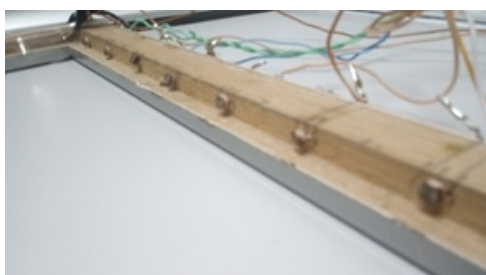


Figura 60 – LDR fixados no Quadro



Figura 61 – Lasers fixados no Quadro

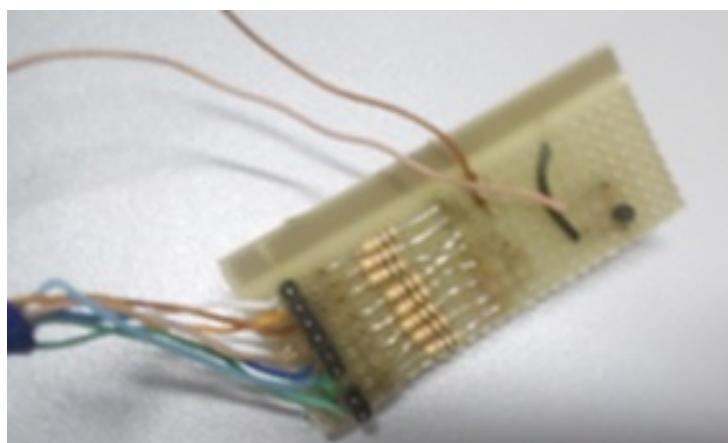


Figura 62 – Placa com circuito da Harpa Laser

Fonte: O Autor, 2020

O Xilofone Capacitivo trata-se de um DMI que utiliza um sensor capacitivo como tecla. Para a construção deste DMI foi utilizado um clipe de papel, um resistor de $10M\Omega$

e a biblioteca `CapacitiveSensor`¹⁴. O circuito necessário para a construção deste DMI é apresentado na Figura 63. O clipe de papel foi utilizado como material metálico com o objetivo de reduzir os custos do projeto, porém pode ser utilizado qualquer material que conduz eletricidade.

A biblioteca `CapacitiveSensor` é responsável por transformar as portas do Arduino em um sensor capacitivo, ou seja pode ser utilizada para detectar a capacitância do corpo humano. São necessárias 2 portas do Arduino para cada sensor utilizado, pois a biblioteca funciona enviando um sinal para uma das portas do Arduino denominada de emissor (E), então aguarda-se a variação da capacitância através da porta denominada receptor (R). Para definir cada uma das portas é necessário informar a função “`CapacitiveSensor(E, R)`”.

O resistor é utilizado para reduzir ruídos indesejados o que causa um atraso na recepção do sinal. É recomendado pelo desenvolvedor da biblioteca utilizar um resistor no valor entre $100\text{K}\Omega$ e $50\text{M}\Omega$. Nos testes realizados com este circuito foi observado que quanto maior o resistor maior é o atraso do sinal, logo o valor que trouxe um melhor resultado foi o de $10\text{M}\Omega$. Também foi observado que materiais com uma área maior possuem mais sensibilidade, muitas vezes era possível detectar os sinais antes de encostar no material metálico, por este motivo foi utilizado um clipe de papel.

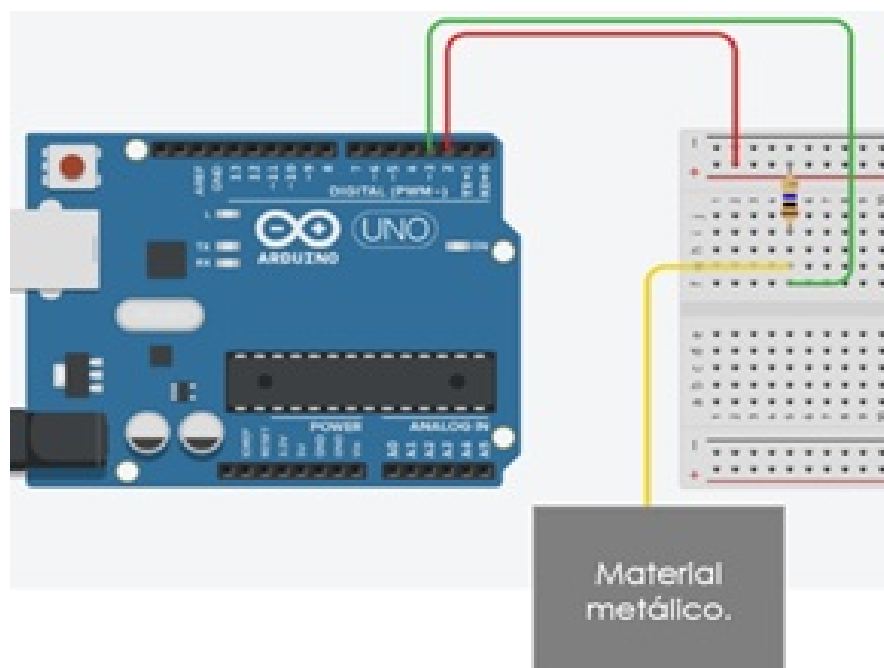


Figura 63 – Circuito Xilofone Capacitivo

Fonte: O Autor, 2020

Este DMI foi resultado do Xilofone Capacitivo, seu funcionamento utiliza o mesmo princípio porém são utilizados mais resistores no circuito. A Figura 64 apresenta os

¹⁴ <https://playground.arduino.cc/Main/CapacitiveSensor/>

componentes conectados, são utilizados 3 resistores de $10M\Omega$ em série, desta forma as resistências são somadas, logo são $30M\Omega$ adicionados ao circuito. O Resistor de $1K\Omega$ foi utilizado para reduzir ruídos e melhorar a precisão. Com isso o sensor capacitivo fica mais sensível podendo ser detectado a uma distância aproximada de 10cm. Assim este DMI possui o mesmo funcionamento do Teremim original podendo ser tocado sem contato. A principal diferença entre os Teremim Digital apresentado anteriormente é que este possui menos componentes e por consequência se tornou mais barato.

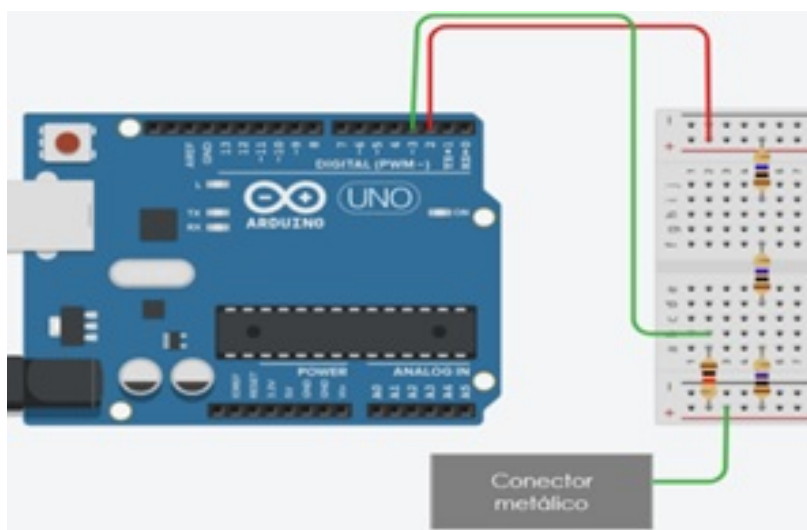


Figura 64 – Circuito Teremim Capacitivo

Fonte: O Autor, 2020

Controladores MIDI são dispositivos que conseguem enviar e receber dados MIDI. Estes dispositivos não geram sons apenas trabalham com dados numéricos que são utilizados para tocar notas musicais ou para controle de efeitos como volume, modulação, etc. Este DMI foi construído com o objetivo de agrupar vários sensores em um único dispositivo.

Como pode ser verificado na Figura 65 o controlador MIDI possui 6 botões que tocam uma nota musical cada, 1 sensor piezoelétrico que toca o som de um chimbau de bateria, 2 potenciômetros para acionar efeitos de volume e modulação. Com isso o Arduino interpreta cada sinal dos sensores e envia um sinal para o computador no formato do protocolo MIDI, então o software virtual Keyboard interpreta o sinal e executa o comando.

A Figura 66 apresenta o circuito construído em placa perfurada do controlador MIDI, com o objetivo de ser fácil de montar na placa Arduino. Foi utilizado uma patola para fixar todos os componentes.



Figura 65 – Controlador MIDI

Fonte: O Autor, 2020

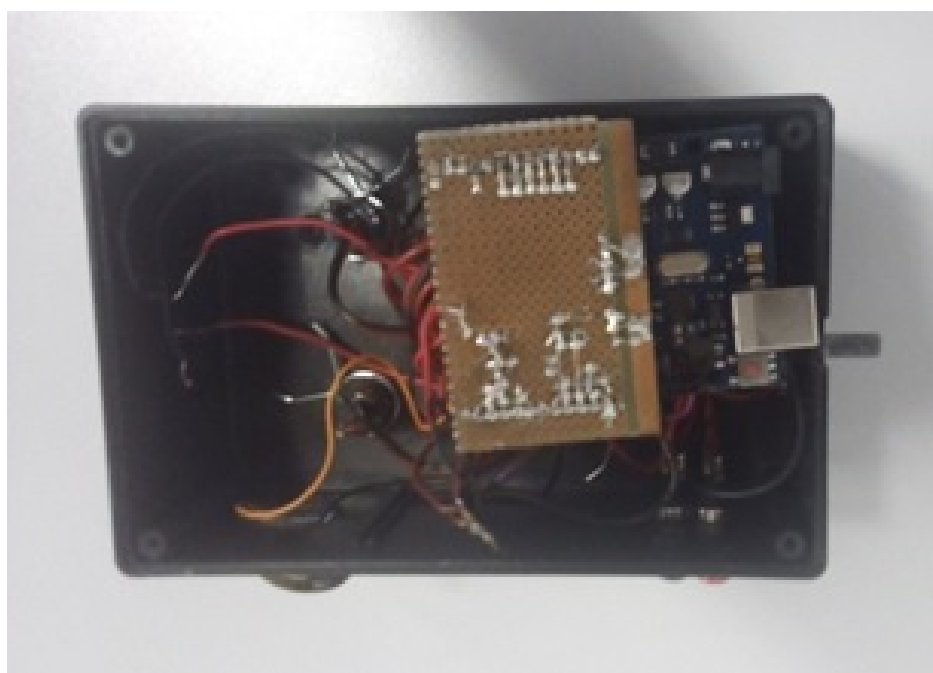


Figura 66 – Placa do circuito Controlador MIDI

Fonte: O Autor, 2020

4.5 Análise de custo e funcionalidade dos protótipos

Uma das principais motivações deste projeto é construir DMIs utilizando componentes que tenham um custo baixo quando comparados aos DMIs comercializados. Isso foi determinante para escolher os componentes do projeto, principalmente a placa Arduino. Ao todo foram construídos seis DMIs, porém alguns DMIs tiveram modificações e adaptações, por este motivo há alterações no custo.

O teclado digital (Figura 37) foi o primeiro DMI construído por ser o mais simples e intuitivo. Apesar da simplicidade do circuito, este DMI possui diversas teorias básicas de programação eletrônica que podem ser utilizadas em outros DMIs, como leitura e saída digital do Arduino, resistores *pullup* a *pulldown*, saída PWM, estruturas de controle IF e ELSE e laços de repetição. O custo deste DMI é apresentado na Tabela 5.

Tabela 5 – Custo para construção do teclado digital

Teclado digital			
Componente	Quantidade	Preço unitário	Preço
Arduino UNO	1	R\$ 51,57	R\$ 51,57
Botões chave táctil	6	R\$ 0,45	R\$ 2,70
Buzzer	1	R\$ 1,42	R\$ 1,42
Potenciômetro 10k Ω	1	R\$ 1,79	R\$ 1,79
Resistor 330 Ω	5	R\$ 0,06	R\$ 0,30
Resistor 10k Ω	5	R\$ 0,06	R\$ 0,30
LED difuso	5	R\$ 0,24	R\$ 2,58
Conector jack j2	1	R\$ 0,85	R\$ 0,85
Total Quantidade	25	Total	R\$ 61,51

Fonte: O Autor, 2020

A bateria e o xilofone (Figuras 49 e 50) possuem o mesmo princípio de funcionamento de percussão, ou seja, necessita de um impacto para se obter o som. Desta forma os dois instrumentos foram construídos da mesma forma, porém o som que o instrumento musical emite é diferente. Por este motivo os dois instrumentos tiveram o mesmo custo como visto na Tabela 6, uma vez que o som é emitido pelo computador através de um software. O Arduino e piezoelétrico são os principais componentes deste DMI, assim o xilofone e bateria podem ser construídos com esses componentes e com auxílio de uma protoboard.

Tabela 6 – Custo para construção da Bateria/Xilofone digital

Bateria/Xilofone			
Componente	Quantidade	Preço unitário	Preço
Arduino UNO	1	R\$ 51,57	R\$ 51,57
Piezo 20mm	6	R\$ 0,34	R\$ 2,04
Resistor 1M Ω	6	R\$ 0,07	R\$ 0,42
Placa perfurada	1	R\$ 3,00	R\$ 3,00
Conector jack F	6	R\$ 0,77	R\$ 4,62
Conector jack M	6	R\$ 0,85	R\$ 5,10
Total Quantidade	26	Total	R\$ 66,75

Fonte: O Autor, 2020

O Teremim (Figura 55) foi construído pensando no funcionamento de dispositivos capacitivos e foi desenvolvido utilizando o Arduino Nano com o objetivo de comparar o seu funcionamento em relação ao Arduino Uno. O preço visto na Tabela 7 relata que este foi o protótipo mais barato, no entanto isso não está relacionado apenas com o Arduino Nano, pois mesmo se o Teremim for construído com os mesmos componentes e com o Arduino Uno, o preço ainda estaria mais barato que os outros protótipos. O fato é que este preço mais barato está relacionado ao menor número de componentes utilizados, inclusive os circuitos integrados possuem a vantagem de ter mais do que uma funcionalidade, permitindo construir todo o protótipo apenas com um circuito integrado.

Tabela 7 – Custo para construção do Teremim Digital

Teremim Digital			
Componente	Quantidade	Preço unitário	Preço
Arduino NANO	1	R\$ 33,66	R\$ 33,66
CI4093	1	R\$ 1,40	R\$ 1,40
LM324	1	R\$ 1,42	R\$ 1,42
Resistor 10k Ω	4	R\$ 0,07	R\$ 0,28
Resistor 5.1k Ω	2	R\$ 0,07	R\$ 0,14
Potenciômetro 1k Ω	2	R\$ 1,30	R\$ 2,60
Capacitor 100pF	2	R\$ 0,09	R\$ 0,18
Total Quantidade	13	Total	R\$ 39,68

Fonte: O Autor, 2020

A Harpa laser (Figura 32) por outro lado, como pode ser visto na Tabela 8 se tornou o protótipo mais caro devido ao uso dos diodos lasers. Nesta primeira etapa foram utilizados 8 diodos lasers, representando uma corda cada.

Tabela 8 – Custo para construção da Harpa Laser

Harpa Laser			
Componente	Quantidade	Preço unitário	Preço
Arduino UNO	1	R\$ 51,57	R\$ 51,57
Diodo Laser	8	R\$ 2,65	R\$ 21,20
Resistor 220 Ω	8	R\$ 0,07	R\$ 0,56
Resistor 10k Ω	8	R\$ 0,07	R\$ 0,56
LDR	8	R\$ 2,47	R\$ 19,76
Total Quantidade	33	Total	R\$ 93,65

Fonte: O Autor, 2020

Os DMIs Xilofone Capacitivo (Figura 43) e Teremim Capacitivo (Figura 55) possuem um custo baixo em relação aos outros por utilizar apenas 1 componente eletrônico

além do Arduino (Tabelas 9 e 10). Há necessidade de utilizar um material metálico que influencia no custo, como foi demonstrado é possível utilizar um clipe de papel, o que torna esses DMIs muito barato e fácil de construí-lo.

Tabela 9 – Custo para construção do Xilofone Capacitivo

Xilofone Capacitivo			
Componente	Quantidade	Preço unitário	Preço
Arduino UNO	1	R\$ 51,57	R\$ 51,57
Resistor 1M Ω	5	R\$ 0,07	R\$ 0,35
Clipe de papel	5	R\$ 0,05	R\$ 0,25
Total Quantidade	11	Total	R\$ 52,17

Fonte: O Autor, 2020

Tabela 10 – Custo para construção do Teremim Capacitivo

Teremim Capacitivo			
Componente	Quantidade	Preço unitário	Preço
Arduino UNO	1	R\$ 51,57	R\$ 51,57
Resistor 10M Ω	3	R\$ 0,07	R\$ 0,21
Clipe de papel	1	R\$ 0,05	R\$ 0,05
Total Quantidade	5	Total	R\$ 51,83

Fonte: O Autor, 2020

Como visto na Tabela 11 o controlador MIDI (Figura 65) foi um dos DMIs mais caros devido a utilização de diversos sensores diferentes, além da caixa plástica (patola) e da placa perfurada. O objetivo de construir um DMI neste formato foi verificar a funcionalidade desses componentes em conjunto e construí-lo como um produto, contendo uma estrutura fechada e com sensores dispostos para serem tocados.

Tabela 11 – Custo para construção do Controlador MIDI

Controlador MIDI			
Componente	Quantidade	Preço unitário	Preço
Arduino UNO	1	R\$ 51,57	R\$ 51,57
Resistor 1M Ω	1	R\$ 0,07	R\$ 0,07
Piezoelétrico 20mm	1	R\$ 0,34	R\$ 0,34
Botões Coloridos	4	R\$ 1,50	R\$ 6,20
Chave Táctil	2	R\$ 2,90	R\$ 5,80
Potenciômetro Linear 10k Ω	1	R\$ 1,50	R\$ 1,50

Potenciômetro Deslizante 30mm 10k Ω	1	R\$ 5,20	R\$ 5,20
Patola PB-114	1	R\$ 14,76	R\$ 14,76
Placa perfurada	1	R\$ 3,00	R\$ 3,00
Total Quantidade	13	Total	R\$ 88,44

Fonte: O Autor, 2020

5 Formação (Objeto 2)

Neste capítulo são apresentadas as etapas percorridas para a elaboração da formação, de acordo com as etapas da pesquisa desenvolvimento de objeto apresentadas no capítulo 3. Com base no desenvolvimento dos DMIs de baixo custo, era necessário fazer a avaliação dos DMIs como ferramenta pedagógica. Para isso, foi feita a formação utilizando os DMIs de baixo custo construídos para o ensino de programação e eletrônica. Este capítulo está organizado de acordo com as principais etapas da metodologia de desenvolvimento de objeto, a seção 5.1 apresenta a Análise do Contexto do Objeto 2. A seção 5.2 apresenta a Análise do Objeto 2. A seção 5.3 apresenta a Preparação do Objeto 2. A seção 5.4 apresenta o Desenvolvimento do Objeto 2.

5.1 Análise do Contexto do Objeto 2

A Análise do Contexto do Objeto 2 se deu a partir do Desenvolvimento do Objeto 1. Assim, foi observado que para verificar se era possível utilizar os DMIs de baixo custo construídos (Objeto 1) para ensinar programação e eletrônica era necessário que estes dispositivos fossem utilizados como ferramentas pedagógicas de forma prática.

A demanda do objeto 2, assim como no objeto 1, trata-se do *STEAM education*, que faz a integração dos DMIs (Artes), com o ensino de programação (Tecnologia) e eletrônica (Engenharia) no contexto educacional. A abordagem escolhida foi de ensinar os estudantes a construir seus próprios DMIs. Por este motivo a Eletrônica foi incluída como conteúdo de ensino.

A necessidade trata-se de verificar a possibilidade de se utilizar DMIs de baixo custo para o ensino de programação. Com isso, tem-se como motivação construir uma abordagem que permita utilizar DMIs para o ensino de programação e fazer com que os estudantes sejam desenvolvedores de tecnologia.

Os conhecimentos necessários para realizar a etapa de formação são de elaboração de um plano de ensino para executar a etapa de formação, bem como os assuntos de lógica de programação, eletrônica, música e a construção de DMIs. A etapa de formação tem como objetivos e intenções:

- Verificar os desafios impostos no planejamento de curso que relacione o ensino de programação e eletrônica a partir da construção de instrumentos musicais digitais de baixo custo;
- Identificar os limites e possibilidades da integração do ensino de programação e

eletrônica a partir da construção de instrumentos musicais digitais de baixo custo;

- Validar a proposta de utilizar DMIs de baixo custo para o ensino de programação e eletrônica;
- Construir uma abordagem que permita levar o ensino de programação para estudantes sem conhecimentos prévios de programação e eletrônica.

Como meios de execução optou-se por realizar um curso que tem como método ensinar conceitos básicos de programação e eletrônica aos estudantes por meio da construção seus próprios DMIs de baixo custo. Para isso, os conteúdos necessários são de lógica de programação, eletrônica e música.

Levando em consideração as intenções desta etapa, escolheu-se como público-alvo estudantes do curso de Licenciatura em Música sem conhecimentos prévios de programação e/ou eletrônica. Estes estudantes possuem um perfil de formadores, por este motivo o objeto 2 é denominado como Formação, esta escolha é justificada pelas motivações desta pesquisa e por conta destes estudantes serem possíveis multiplicadores dessa abordagem educacional.

Os procedimentos adotados para alcançar o objeto 2 foi a elaboração de um plano de ensino baseado em uma abordagem gameficada¹, contendo os conteúdos de programação, eletrônica e música, os resultados de aprendizagem, as atividades a serem realizadas, materiais a serem utilizados e formas de avaliação.

Cada tópico da análise do contexto é apresentado na Tabela 12. Dessa forma, foi idealizado que seria ofertado um curso de formação a fim de validar o Objeto 1.

Tabela 12 – Análise do Contexto do Objeto 2 - Formação.

Etapas	Análise do Contexto
Demanda Necessidades	STEAM <i>education</i> . Verificar a possibilidade de se utilizar DMIs de baixo custo para o ensino de programação e eletrônica.
Conhecimentos e Motivações	Elaborar plano de aula, lógica de programação, eletrônica e música; Demonstrar a aplicabilidade de DMIs de baixo custo na educação.

¹ Gameificação se refere a forma de se conduzir as aulas a partir da mecânica dos games fora do contexto dos games, onde o aprendizado esta cercado de desafios, prazer e entretenimento (ALVES; MINHO; DINIZ, 2014)

Objetivos e Intenções	Verificar os desafios impostos no planejamento de curso que relacione o ensino de programação e eletrônica a partir da construção de instrumentos musicais digitais de baixo custo; Identificar os limites e possibilidades da integração do ensino de programação e eletrônica a partir da construção de instrumentos musicais digitais de baixo custo; Validar a proposta de utilizar DMIs de baixo custo para o ensino de programação e eletrônica para estudantes sem conhecimentos prévios.
Contexto	Para que a etapa de construção de DMIs de baixo custo (objeto 1) seja validada é necessário realizar uma segunda etapa utilizando o objeto 1 como ferramenta de ensino, por meio de uma etapa de formação.
Meios	Realizar um curso de formação de ensino de programação e eletrônica com os DMIs de baixo custo.
Conteúdos	Programação, Eletrônica e Música.
Processos	Ensinar programação e eletrônica utilizando DMIs como fator motivacional.
Público Alvo	Estudantes de Licenciatura em Música sem conhecimentos prévios de programação e eletrônica.

Fonte: O Autor, 2020

5.2 Análise do Objeto 2

Com base na Análise do Contexto do Objeto 2 tem-se como resposta o objeto 2 - A Formação. Com isso, a Análise do Objeto 2 é composta pela Conceptualização do Objeto 2 e pela Modelagem do Objeto 2. A seção 5.2.1 Conceptualização do Objeto 2 refere-se ao processo idealizado para a construção do curso de formação. A seção 5.2.2 Modelagem do Objeto 2 apresenta como foi idealizado o ciclo das aulas do curso de formação.

5.2.1 Conceptualização do Objeto 2

Para construção da etapa de formação foi tido como premissa que os estudantes teriam como atividades práticas a construção de diferentes tipos de DMIs de baixo custo. Para isso, foram escolhidos cinco DMIs (Teclado, Xilofone, Bateria, Teremim e Controlador MIDI) para serem construídos ao longo da etapa de formação. A escolha destes DMIs se deu pelos resultados obtidos na etapa de desenvolvimento do Objeto 1, levando em conta fatores como o grau de complexidade para construção do DMI, conceitos sobre

programação e eletrônica que poderiam ser ensinados, e funcionalidade do DMI para que fosse apresentado um dispositivo que motivasse o estudante.

Foram idealizadas dez aulas para serem ministradas em curso de extensão contendo conceitos de programação e eletrônica visando construir os DMIs. As aulas foram ordenadas com a intenção de demonstrar que cada conceito apresentado é necessário e não só mais um conceito. Assim, enquanto os DMIs foram sendo construídos foram apresentados problemas de funcionalidade para que posteriormente fossem solucionados com algum conceito de programação e eletrônica. Desta forma, os conteúdos de lógica de programação e eletrônica foram apresentados de acordo com a limitação de cada DMI desenvolvido.

Partindo do pressuposto que os estudantes possuem o conhecimento de área de Música, esperava-se que não apresentassem dificuldades em criar/reproduzir diferentes sons com os conceitos apresentados, por este motivo conceitos relacionados a esta área não foram aprofundados, mas utilizados de forma prática. Para avaliar o desempenho de cada estudante, ao final de cada aula responderiam a um questionário referente ao conteúdo daquela aula de forma gamificada por meio da ferramenta Kahoot. Além disso, em cada etapa serão observadas suas motivações e dificuldades realizando anotações pontuais sobre as dificuldades e sugestões.

Com estas definições os estudantes obteriam diversos benefícios, tais como: aprendizado ou aperfeiçoamento de lógica de programação, aprender sobre conceitos de eletrônica na prática, aprender como integrar as áreas de programação e eletrônica com a música por meio da criação de um DMI, desenvolver a motivação para ser um desenvolvedor/criador.

Para a elaboração do curso de extensão foi necessário selecionar os DMIs construídos que possuem um custo baixo e que sejam possíveis de ensinar estudantes sem conhecimentos prévios a construí-los. Assim, são apresentados os DMIs e suas características escolhidas:

- Teclado e Xilofone de Botão: Utiliza componentes fáceis de serem montados e possibilitar o ensino de conceitos básicos de programação e eletrônica;
- Xilofone e Bateria com Piezoelétricos: Permite utilizar as funcionalidades analógicas do Arduino e geração de corrente elétrica que possibilita a apresentação de conceitos sobre eletrônica;
- Xilofone Capacitivo: Permite ser construído com qualquer tipo de material metálico e possibilita a utilização de biblioteca no Arduino;
- Teremim: Permitir a construção de um DMI que não necessita de contato físico;
- Controlador MIDI: Permite a união dos sensores aprendidos.

5.2.2 Modelagem do Objeto 2

O modelo elaborado para a realização de cada aula da Formação é apresentado na Figura 67. O modelo se inicia pela escolha do DMI e como será seu funcionamento. Com isso, tem-se o principal comportamento do DMI o que permite definir quais serão os sensores de entrada, permitindo assim construir o circuito eletrônico. A próxima etapa deve ser o código com a lógica no Arduino com a lógica do mapeamento e saída sonora. Na etapa de teste tem-se um DMI propriamente dito, mas apresentando alguns problemas como qualidade sonora ruim, dificuldades para tocar determinadas notas ou tocar várias notas ao mesmo tempo. Feito isso, o ciclo se completa com a apresentação das possíveis soluções para corrigir os problemas e então tem-se o mesmo DMI melhorado. Este novo DMI é reutilizado para dar início ao próximo DMI.

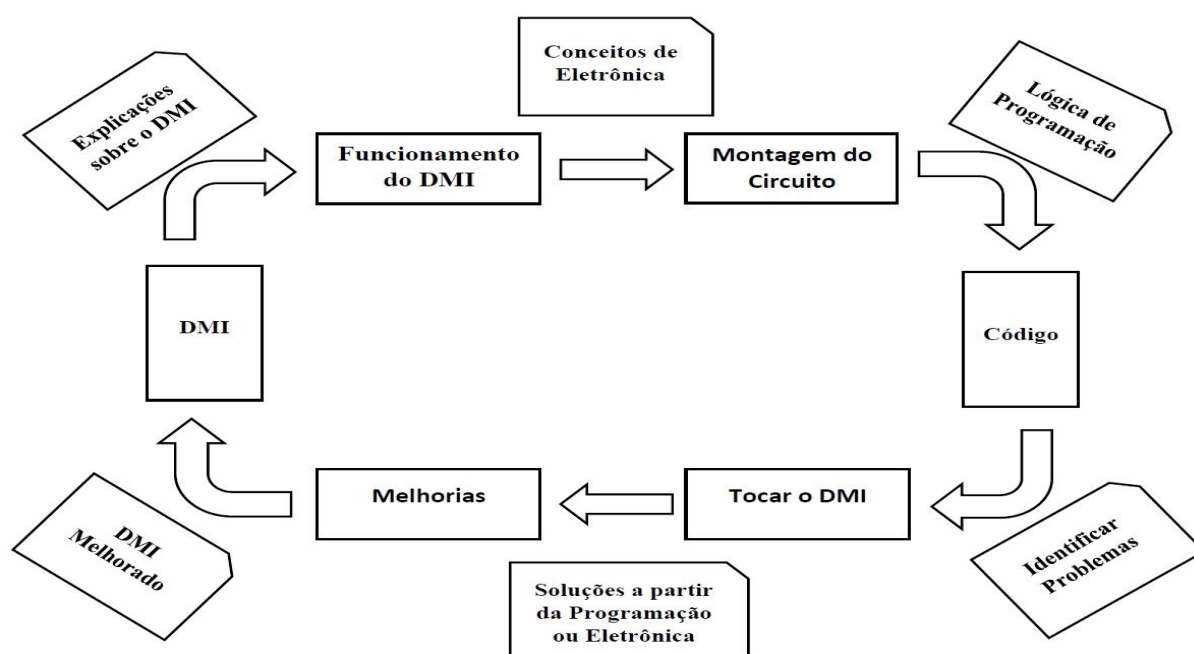


Figura 67 – Modelo para aulas da etapa de Formação

Fonte: O autor, 2020

5.3 Preparação do Objeto 2

A estratégia para realização da formação é representada em forma de plano de ensino (Tabela 13). Ao todo foram elaboradas 10 aulas com 2 horas e 15 minutos cada. Os conteúdos são um resumo do que cada aula apresenta. Os resultados de aprendizagem são os tópicos que os estudantes irão compreender ao final de cada aula. As atividades de ensino apresentam cada atividade prática e teórica que será apresentada durante a aula. Os materiais de apoio são todos os itens utilizados tanto pelos estudantes quanto pelo professor durante a aula. As atividades avaliativas são as ferramentas utilizadas para avaliar os estudantes ao final da aula.

Tabela 13 – Plano de Ensino etapa de Formação

Aula 01:	Introdução aos DMIs e Arduino
Carga Horária:	2:15h
Conteúdos:	<ul style="list-style-type: none"> • Visão Geral do Curso; • Instrumentos Musicais Digitais (DMIs); • Aferição do entendimento sobre DMIs; • Placa Arduino; • Aferição do entendimento sobre a Placa Arduino.
Resultados de Aprendizagem:	<ul style="list-style-type: none"> • R.A.1.1: Distinguir entre DMI e não DMI; • R.A.1.2: Identificar os componentes da placa do Arduino; • R.A.1.3: Gravar no Arduino o código feito no computador.
Atividade de Ensino:	<ul style="list-style-type: none"> • Apresentação e Discussão; • Perguntas com imagens de DMIs e instrumentos musicais convencionais e respostas do tipo Verdadeiro e Falso; • 6 Perguntas sobre a placa Arduino com imagens e alternativas; • Atividade prática para testar o Arduino com o código Blink.
Materiais de Apoio:	<ul style="list-style-type: none"> • Slides; • Placa Arduino • Computador com IDE do Arduino.
Atividade Avaliativa:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot.
Aula 02:	Reproduzindo Notas Musicais
Carga Horária:	2:15h
Conteúdos:	<ul style="list-style-type: none"> • Buzzer; • Aferição do entendimento sobre o Buzzer; • Código do Buzzer; • Aferição do entendimento sobre o código do Buzzer; • Programar música de sua escolha.
Resultados de Aprendizagem:	<ul style="list-style-type: none"> • R.A.2.1: Conectar o Buzzer na placa Arduino; • R.A.2.2: Utilizar o comando #define; • R.A.2.3: Configurar uma porta como saída; • R.A.2.4: Utilizar o comando tone para reproduzir notas musicais; • R.A.2.5: Utilizar o comando delay para pausar o programa.

Atividade de Ensino:	<ul style="list-style-type: none"> • Apresentação, imagens do Buzzer e montagem na placa Arduino; • 4 Perguntas sobre circuito do Buzzer com 4 alternativas cada; • Apresentação, imagens com código de exemplo; • 8 Perguntas sobre código do Buzzer com 4 alternativas cada; • Atividade prática para programar uma música.
Materiais de Apoio:	<ul style="list-style-type: none"> • A Slides; • Placa Arduino; • Buzzer; • Computador com IDE do Arduino.
Atividade Avaliativa:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot.
Aula 03:	Qualidade do Som
Carga Horária:	2:15h
Conteúdos:	<ul style="list-style-type: none"> • Programar música de sua escolha; • Conector Jack J2, Protoboard e Resistor; • Aferição do entendimento sobre Jack J2, Protoboard, Resistores e Fios; • Utilizar fones de ouvido no Arduino; • Potenciômetro; • Aferição do entendimento sobre potenciômetro; • Controle do volume.
Resultados de Aprendizagem:	<ul style="list-style-type: none"> • R.A.3.1: Utilizar fone de ouvidos no circuito; • R.A.3.2: Montar circuitos na protoboard; • R.A.3.3: Utilizar resistores de diferentes valores; • R.A.3.4: Utilizar potenciômetro no circuito.
Atividade de Ensino:	<ul style="list-style-type: none"> • Finalização da Atividade prática da aula anterior; • Apresentação, Imagens dos circuitos em protoboard; • 9 Perguntas sobre circuito com fone de ouvido, sendo 4 de verdadeiro ou falso e 5 com 4 alternativas; • Atividade prática para montar o circuito com o jack j2, fone de ouvido, resistores e protoboard; • Apresentação, imagens do potenciômetro e discussão; • 8 Perguntas sobre o código com 4 alternativas cada; • Atividade prática para montar o potenciômetro com circuito anterior.

Materiais de Apoio:	<ul style="list-style-type: none"> • Slides; • Placa Arduino; • Jack J2; • Protoboard; • Resistores; • Fios; • fones de ouvido; • potenciômetro.
Atividade Avaliativa:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot..
Aula 04:	Botões
Carga Horária:	2:15h
Conteúdos:	<ul style="list-style-type: none"> • Botões; • Aferição do entendimento sobre os Botões; • Construindo primeiro DMI.
Resultados de Aprendizagem:	<ul style="list-style-type: none"> • R.A.4.1: Conectar botões na protoboard; • R.A.4.2: Programar o Arduino para ler valores dos botões; • R.A.4.3: Reproduzir notas musicais em tempo real.
Atividade de Ensino:	<ul style="list-style-type: none"> • Apresentação, imagens do botão, conexão na protoboard e código necessário; • 9 Perguntas sobre circuito Botões com 4 alternativas cada. • Atividade prática para montar 5 botões na protoboard e programá-los para executar uma nota cada um.
Materiais de Apoio:	<ul style="list-style-type: none"> • Slides; • Placa Arduino; • Jack J2; • Protoboard; • Resistores; • Fios; • fones de ouvido; • potenciômetro; • Computador com IDE Arduino ; • 5 botões.
Atividade Avaliativa:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot.
Aula 05:	Comandos de Decisão

Carga Horária:	2:15h
Conteúdos:	<ul style="list-style-type: none"> • Comando de decisão IF; • Teste prático dos botões; • Acrescentar 1 botão em outra protoboard; • Controlar LED; • Código do LED e comando ELSE; • Aferição do entendimento sobre os comandos de decisão IF e ELSE; • Melhorando a precisão do primeiro DMI e controle de LED.
Resultados de Aprendizagem:	<ul style="list-style-type: none"> • R.A.5.1: Utilizar comandos IF e ELSE para identificar os sinais dos botões; • R.A.5.2: Conectar circuitos em 2 protoboards; • R.A.5.3: Utilizar LEDs em circuitos; • R.A.5.4: Utilizar comandos digitalWrite para controlar LEDs; • R.A.5.5: Utilizar operador booleano && com comandos IF.
Atividade de Ensino:	<ul style="list-style-type: none"> • Apresentação, imagens do código necessário; • Atividade prática, verificar se o comando IF melhorou a precisão; • Apresentação, imagens do circuito; • Apresentação, imagens do LED e circuito; • Apresentação, imagens do código para o LED; • 9 Perguntas sobre código IF e Else com 4 alternativas cada. • Atividade prática para alterar o código anterior utilizando comandos de decisão, montar mais 1 botão e 5 LEDs.
Materiais de Apoio:	<ul style="list-style-type: none"> • Slides; • Placa Arduino; • Jack J2; • 2 Protoboards; Resistores; • Fios; • fones de ouvido • potenciômetro; • Computador com IDE Arduino; • 6 botões; • 5 LEDs.
Atividade Avaliativa:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot..
Aula 06:	Protocolo MIDI

Carga Horária:	2:15h
Conteúdos:	<ul style="list-style-type: none"> • Protocolo MIDI; • Aferição do entendimento sobre protocolo MIDI; • Melhorando a qualidade sonora.
Resultados de Aprendizagem:	<ul style="list-style-type: none"> • R.A.6.1: Configurar o computador para executar comandos MIDI; • R.A.6.2: Enviar comandos MIDI do Arduino ao computador.
Atividade de Ensino:	<ul style="list-style-type: none"> • Apresentação, imagens dos códigos necessários. • 11 Perguntas sobre protocolo MIDI, sendo 10 com 4 alternativas cada e 1 com verdadeiro ou falso. • Atividade prática para programar o Arduino com comandos do protocolo MIDI.
Materiais de Apoio:	<ul style="list-style-type: none"> • Slides; • Placa Arduino; • 2 Protoboards; • Resistores; • Fios; • fones de ouvido; • Computador com IDE Arduino; • 6 botões; • 5 LEDs; • Software FreePiano.
Atividade Avaliativa:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot.
Aula 07:	Variáveis
Carga Horária:	2:15h
Conteúdos:	<ul style="list-style-type: none"> • Variáveis; • Aferição do entendimento sobre variáveis; • Tocando múltiplas teclas ao mesmo tempo; • Funções; • Aferição do entendimento sobre funções; • Melhorando o código.
Resultados de Aprendizagem:	<ul style="list-style-type: none"> • R.A.7.1: Utilizar variáveis com Arduino; • R.A.7.2: Ler e enviar sinais de vários botões ao mesmo tempo; • R.A.7.3: Utilizar funções com Arduino.

Atividade de Ensino:	<ul style="list-style-type: none"> • Apresentação, imagens dos códigos necessários; • 8 Perguntas sobre variáveis com 4 alternativas cada; • Atividade prática para programar o Arduino com variáveis; • Apresentação, imagens dos códigos necessários; • 5 Perguntas sobre funções com 4 alternativas cada; • Atividade prática para programar as funções para enviar comandos MIDI.
Materiais de Apoio:	<ul style="list-style-type: none"> • Slides; • Placa Arduino; • Protoboard; • Resistores; • Fios; • fones de ouvido; • Computador com IDE Arduino; • 5 botões; • Software FreePiano.
Atividade Avaliativa:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot.
Aula 08:	Detectando Pressão
Carga Horária:	2:15h
Conteúdos:	<ul style="list-style-type: none"> • Sensor Piezoelétrico; • Programar Piezoelétrico; • Diodo; • Aferição do entendimento sobre piezoelétricos; • Detectando pressão; • Comandos Map e Millis; • Aferição do entendimento sobre map e millis; • Melhorar precisão.
Resultados de Aprendizagem:	<ul style="list-style-type: none"> • R.A.8.1: Utilizar sensor piezoelétrico na protoboard; • R.A.8.2: Utilizar o comando analogRead para ler sinais dos piezoelétricos; • R.A.8.3: Melhorar o sinal do piezoelétrico com diodo; • R.A.8.4: Detectar pressão do piezoelétrico; • R.A.8.5: Mapear os valores lidos de entradas analógicas com função map; • R.A.8.6: Verificar valores com base no tempo com a função Millis.

Atividade de Ensino:	<ul style="list-style-type: none"> • Apresentação, imagens do circuito piezoelétrico; • Apresentação, imagens do código; • Apresentação, imagens do circuito com piezoelétrico e diodo; • 8 Perguntas sobre circuito piezoelétrico com 4 alternativas cada; • Atividade prática para detectar pressão com piezoelétrico; • Apresentação, imagens do código necessário; • 6 Perguntas sobre o código do piezoelétrico com 4 alternativas cada; • Atividade prática modificando o código com as funções map e millis.
Materiais de Apoio:	<ul style="list-style-type: none"> • Slides; • Placa Arduino; • Protoboard; • Fios; • fones de ouvido; • Computador com IDE Arduino; • 5 botões; • Software FreePiano; • 3 piezoelétricos.
Atividade Avaliativa:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot.
Aula 09:	Sensor Capacitivo
Carga Horária:	2:15h
Conteúdos:	<ul style="list-style-type: none"> • Sensor Capacitivo; • Aferição do entendimento sobre sensor capacitivo; • Construindo DMI Capacitivo e Teremim.
Resultados de Aprendizagem:	<ul style="list-style-type: none"> • R.A.9.1: Utilizar materiais metálicos como sensor capacitivo; • R.A.9.2: Instalar uma biblioteca no Arduino e utilizar suas funções; • R.A.9.3: Programar o Arduino para ler sensores capacitivos.
Atividade de Ensino:	<ul style="list-style-type: none"> • Apresentação, imagens do circuito piezoelétrico e código. • 12 Perguntas sobre circuito do sensor capacitivo com 4 alternativas cada; • Atividade prática para construir o DMI capacitivo e o Teremim.

Materiais de Apoio:	<ul style="list-style-type: none"> • Slides; • Placa Arduino; • Protoboard; • Fios; • fones de ouvido; • Computador com IDE Arduino; • Software FreePiano; • clipes de papel; • fita isolante.
Atividade Avaliativa:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot..
Aula 10:	Finalização
Carga Horária:	2:15h
Conteúdos:	<ul style="list-style-type: none"> • Todo conteúdo do curso.
Resultados de Aprendizagem:	<ul style="list-style-type: none"> • R.A.10: Aprendizado e Retenção dos conhecimentos.
Atividade de Ensino:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot com as perguntas dos Kahoots anteriores.
Materiais de Apoio:	<ul style="list-style-type: none"> • Plataforma Kahoot.
Atividade Avaliativa:	<ul style="list-style-type: none"> • Utilizando a plataforma Kahoot.

Fonte: O Autor, 2020

5.4 Desenvolvimento do Objeto 2

Nesta etapa foi realizado a aplicação da formação utilizando os DMIs de baixo custo. Para isso, esta pesquisa foi registrada na Plataforma Brasil sendo aprovada pelo Sistema CEP/CONEP com número CAAE: 21320019.6.0000.0020 e parecer 3.607.208. Após isso, foi divulgado como sendo um curso de extensão para estudantes do curso de Licenciatura em Música.

Foi determinado que os estudantes seriam maiores de 18 anos e para realização do curso de formação foi elaborado um Termo de Consentimento Livre e Esclarecido (TCLE) para cada estudante (Apêndice A) assinar e enviar juntamente com a inscrição. Todos os materiais necessários foram disponibilizados para os estudantes durante o período das aulas. O curso contou com as instalações do laboratório de informática da Pontifícia

Universidade Católica do Paraná – (PUCPR).

Além do TCLE, no momento de inscrição os estudantes responderam a um questionário contendo 3 questões referentes as suas percepções sobre o curso e seu conteúdo. Foram obtidas inscrições de 4 estudantes (E1, E2, E3 e E4), as perguntas e respostas deste questionário são apresentadas na Tabela 14.

Pode ser observado que 3 estudantes (E1, E2 e E3) se inscreveram devido a proposta envolver programação e eletrônica e por estar ligado a área de música. É possível verificar que apenas 2 estudantes possuíam algum tipo de conhecimento prévio sobre programação, porém nenhum dos estudantes possuíam conhecimentos prévios sobre o Arduino.

Tabela 14 – Respostas do Questionário de Inscrição

Questão	E1	E2	E3	E4
Qual sua motivação para participar deste curso?	Sempre tive interesse na área, e tenho vontade de desenvolver minhas habilidades além do que é trabalhado na faculdade.	Tenho muito interesse na parte tecnológica associada a música, assim como em circuitos e programação em geral.	A proposta do curso me interessou.	indicação do Carlos.
Você sabe o que é um Arduino? Se sim já utilizou?	Sim, porém não tenho contato.	Sei o que é mas nunca utilizei.	Sim, mas nunca utilizei.	Não
Você já teve contato com alguma linguagem de Programação? Se sim, nos conte mais sobre essa experiência. (Qual a linguagem, qual o contexto)	Tenho curso de programação de 112 horas, que fiz durante o ensino médio, e outros cursos nessa área (digitação, etc...).	Cursei dois semestres de Ciência da Computação, onde tive contato com diversas linguagens, sendo C e Pascal as principais.	Não	Não sei.

Fonte: O Autor, 2020

Cada aula do curso de extensão sobre construção de DMIs de baixo de custo teve

uma atividade prática e questionários referentes aos conteúdos teóricos. Foram organizadas 2 aulas por semana durante 5 semanas com duração de aproximadamente 2 horas cada aula. Todas as aulas foram ministradas dentro da Pontifícia Universidade Católica do Paraná campus Curitiba. Assim, os tópicos definidos para as aulas são:

- Aula 01 – Introdução ao DMI e Arduino;
- Aula 02 – Reproduzindo Notas Musicais;
- Aula 03 – Qualidade do Som;
- Aula 04 – Botões;
- Aula 05 – Comandos de Decisão;
- Aula 06 – Protocolo MIDI;
- Aula 07 – Variáveis;
- Aula 08 – Detectando Pressão;
- Aula 09 – Sensor Capacitivo;
- Aula 10 – Finalização.

Além dos computadores disponibilizados nos laboratórios da Pontifícia Universidade Católica do Paraná campus Curitiba os materiais utilizados para desenvolvimento dos DMIs formaram 4 kits didáticos onde cada kit contém:

- 1 Arduino UNO R3;
- 1 Conector Jumper;
- 10 Resistores de 330 Ω , 1k Ω , 10k Ω , 100k Ω , 1M Ω , 10M Ω , 10k Ω ;
- 6 botões do tipo Chave Táctil 12x12x4,3mm 4 Terminais;
- 5 LEDs Difuso 5mm
- 1 Buzzer to tipo passivo;
- 70 Cabos;
- 1 conector Jack J2;
- 1 Protoboard 800 furos;
- 1 Protoboard 400 furos;
- 5 piezoelétricos 27mm;
- 5 clipes de papel.

Para a etapa de formação foi necessário elaborar um conjunto de conteúdos teóricos a serem ensinados sobre os componentes, programação e eletrônica. Desta forma para iniciar foram apresentados conteúdos referentes a placa Arduino, lógica de programação e eletrônica. Os principais DMIs escolhidos para serem construídos durante a etapa de formação são o Teclado com botões (Figura 68), Xilofone e Bateria com piezoelétricos (Figura 69), capacitivo (Figura 70) e Teremim capacitivo (Figura 71). Tais DMIs foram escolhidos com o objetivo de ensinar programação e eletrônica, por este motivo não foram adicionados os materiais reciclados para construção.

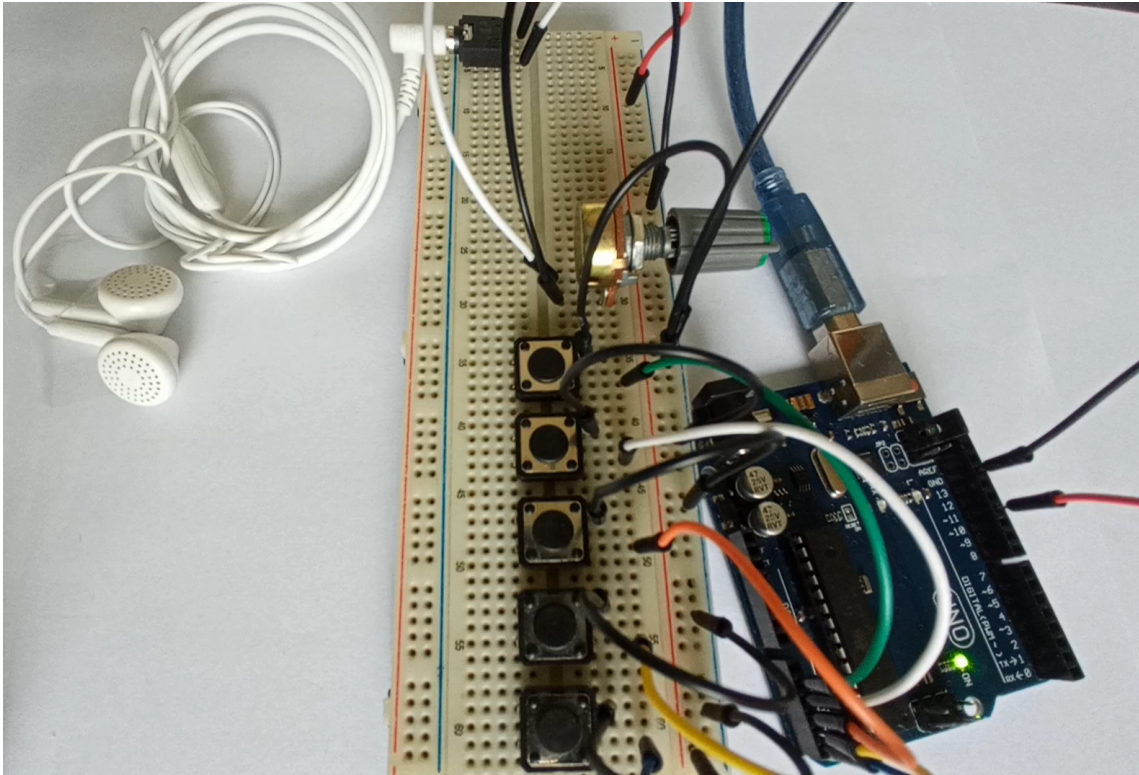


Figura 68 – Circuito Teclado digital com Botões

Fonte: O autor, 2020

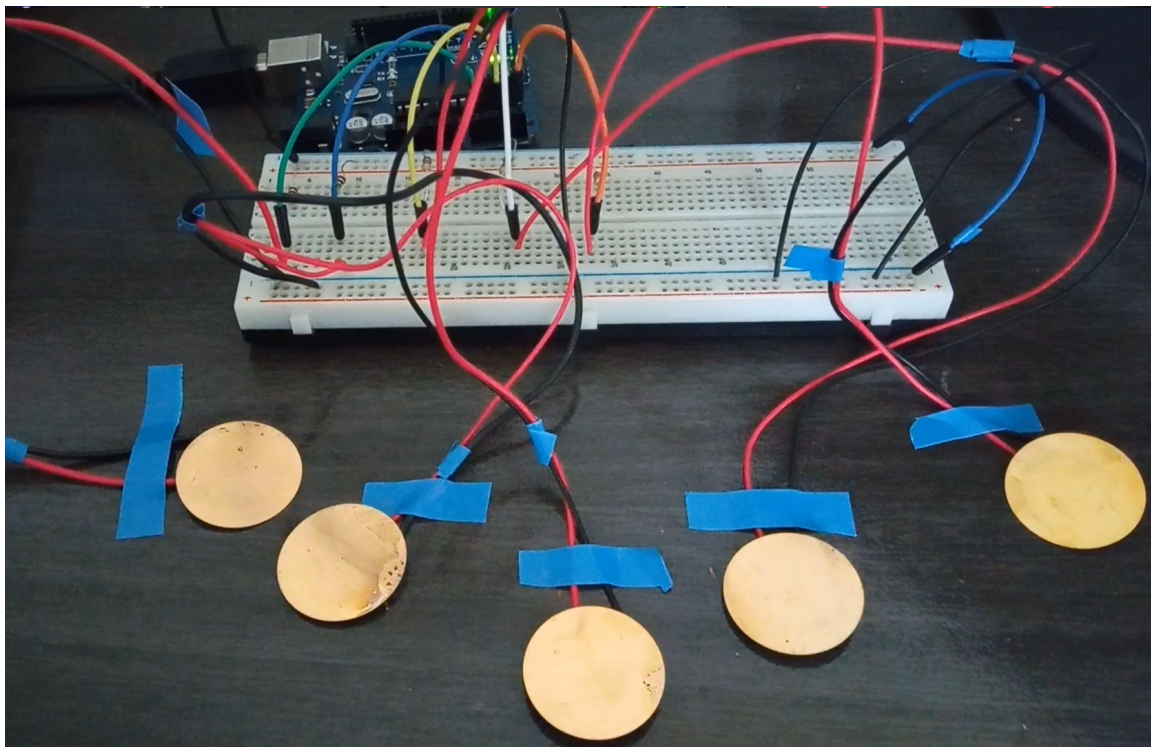


Figura 69 – Circuito Xilofone e Bateria digital com piezoelétrico

Fonte: O autor, 2020

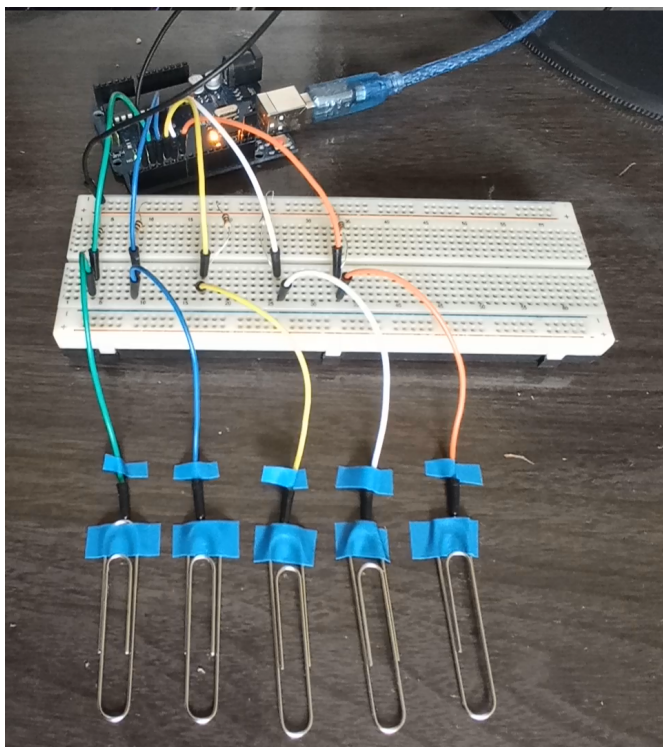


Figura 70 – Circuito Teclado digital capacitivo

Fonte: O autor, 2020

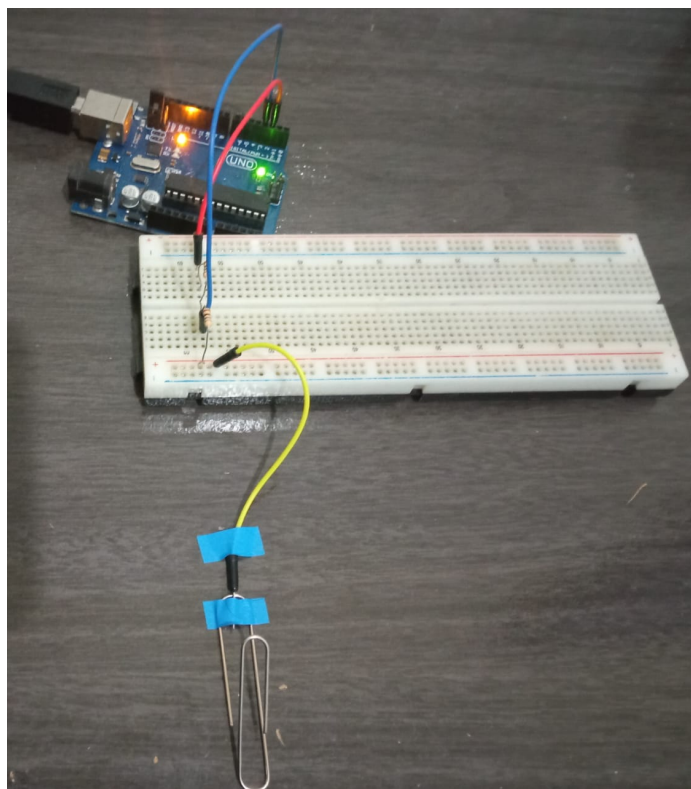


Figura 71 – Circuito Teremin Capacitivo

Fonte: O Autor, 2020

5.5 Análises da Formação

Nesta etapa são apresentadas as análises da formação. Para cada aula são apresentados os resultados de aprendizagem, a estrutura da aula, os kahoots utilizados, as observações e considerações.

5.5.1 Aula 01 - Introdução ao DMI e Arduino

5.5.1.1 Resultados de Aprendizagem da Aula 01.

A primeira aula do curso teve como resultados de aprendizagem (R.A):

- R.A.1.1: Distinguir entre DMI e não DMI;
- R.A.1.2: Identificar os componentes da placa do Arduino;
- R.A.1.3: Gravar na placa Arduino o código feito no computador.

5.5.1.2 Estrutura da Aula 01

Tabela 15 – Estrutura da Aula 01

Momento	Tópicos	Estratégia Instrucional	Recursos
1º	Visão Geral do Curso	Apresentação e Discussão	Slides
2º	Instrumentos Musicais Digitais (DMIs)	Apresentação	Slides
3º	Aferição do entendimento sobre DMIs	Perguntas com imagens e respostas do tipo Verdadeiro e Falso.	Kahoot 01

4º	Placa do Arduino	Apresentação e Discussão	Slides e Placa Arduino
5º	Aferição do entendimento sobre a Placa Arduino	6 Perguntas com imagens e alternativas	Kahoot 02
6º	Placa do Arduino	Atividade prática para testar o Arduino com o código do Blink	Slides, Placa Arduino, Computador com IDE do Arduino.

Fonte: O Autor, 2020

5.5.1.3 Resultados dos Kahoots 01 e 02

Nesta etapa foram realizados 2 Kahoots, desta forma serão analisados os resultados de cada um deles.






5.5.1.3.1 Kahoot #01 – É um DMI?

Objetivo: Verificar se os estudantes compreenderam como diferenciar um DMI de um instrumento musical comum.

Conteúdo do Kahoot 1: Este Kahoot é formado por cinco questões contendo imagens de dois DMIs e de três instrumentos musicais comuns. Como apresentado no Quadro 2, cada questão foi composta de uma imagem de um dos instrumentos musicais nesta ordem: Q1.1 Bateria, Q1.2 Bateria Digital, Q1.3 Flauta, Q1.4 Piano e Q1.5 Teclado Digital. Cada imagem foi demonstrada na tela do projetor com a afirmação “É um DMI”, desta forma os estudantes responderam em seus celulares/computadores “Verdadeiro” para as imagens de DMIs e “Falso” para imagens dos instrumentos musicais comuns.

Tabela 16 – Perguntas do Kahoot 01

Nº	Questão	Alternativas
----	---------	--------------

Q1.1	<p>É um DMI</p> 	Verdadeiro	Falso
Q1.2	<p>É um DMI</p> 	Verdadeiro	Falso
Q1.3	<p>É um DMI</p> 	Verdadeiro	Falso
Q1.4	<p>É um DMI</p> 	Verdadeiro	Falso
Q1.5	<p>É um DMI</p> 	Verdadeiro	Falso

Resultados: 100% dos estudantes compreenderam o que são os DMIs e como diferenciá-los de um instrumento musical comum, visto que acertaram todas as questões do primeiro Kahoot como apresentado na Tabela 17. Após o primeiro Kahoot foi dado sequência ao conteúdo sobre a placa Arduino e seus componentes e posteriormente realizado o Kahoot 2.

Tabela 17 – Resultados do Kahoot 01

Kahoot 1				
	E1	E2	E3	E4
Q1				
Q2				
Q3				
Q4				
Q5				

Correto

Incorreto

Fonte: O Autor, 2020

5.5.1.3.2 Kahoot #02 – Placa Arduino

Objetivo: Verificar se os estudantes compreenderam o que é cada componente da placa Arduino.

Conteúdo do Kahoot 2: Este Kahoot foi composto por seis questões contendo uma imagem da placa Arduino com um número identificador em cada componente da placa Arduino. Na Tabela 18 são demonstradas todas as questões realizadas. As cinco primeiras questões fazem uma pergunta relacionada a identificação dos componentes da placa Arduino e contém 4 alternativas de respostas com apenas 1 correta, já a questão 6 possui uma afirmação com respostas de verdadeiro ou falso. A Figura 72 foi utilizada para responder as questões Q2.1, Q2.2, Q2.3, Q2.4 e Q2.5. Os componentes da placa Arduino com seu respectivo número identificador são:

1. Botão Reset, utilizado para reiniciar o código gravado;
2. Portas de entrada/saída, utilizadas para enviar/receber sinais de sensores;
3. Porta USB, utilizado para gravar o código na placa Arduino;
4. LED no pino 13; Emissor de luz utilizado para testes da placa Arduino;
5. Microprocessador Atmega328p, realiza o processamento das informações;
6. Conector de Fonte externa; recebe energia de uma fonte externa diferente do USB.

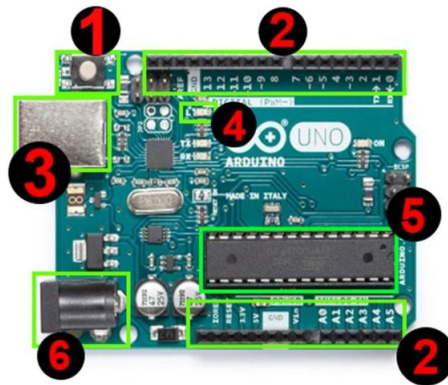


Figura 72 – Placa Arduino com número identificador dos componentes

Fonte: O Autor, 2020

Tabela 18 – Perguntas do Kahoot 02

Nº	Questão	Alternativas	
Q2.1	Qual componente da placa Arduino é utilizado para enviar/receber sinais de sensores?	A) 1	B) 2
		C) 3	D) 4
Q2.2	Qual componente que realiza o processamento o das informações da placa?	A) 1	B) 3
		C) 4	D) 5
Q2.3	Qual componente é o LED no pino 13?	A) 1	B) 3
		C) 4	D) 6
Q2.4	Qual componente faz conexão com o computador utilizado para gravar o código?	A) 1	B) 3
		C) 4	D) 6
Q2.5	Qual componente é utilizado para iniciar o código novamente?	A) 1	B) 2
		C) 3	D) 4
Q2.6	O botão reset apaga o código do Arduino.	Verdadeiro	
		Falso	

Fonte: O Autor, 2020

Cada estudantes recebeu uma placa Arduino durante a aula teórica e poderia consultá-la durante o Kahoot.

Resultados: É possível verificar na Tabela 19 que apenas 2 estudantes conseguiram identificar todos os componentes da placa Arduino. Duas questões foram respondidas de forma incorreta (Q2.1 e Q2.6). No geral todos os estudantes foram capazes de identificar 4 dos 5 componentes da placa Arduino.

Por terem respondido de forma incorreta as questões Q2.1 e Q2.2 foram explicadas novamente durante a execução deste Kahoot para que todos os estudantes realizassem a atividade prática sem dúvidas em relação a estes conteúdos.

A partir dessas informações é possível dizer que os estudantes conseguiram identificar a maioria dos componentes da placa Arduino apenas com o a explicação do conteúdo teórico.

Tabela 19 – Resultados do Kahoot 02

Kahoot 2				
	E1	E2	E3	E4
Q1	Correto	Correto	Correto	Incorreto
Q2	Correto	Correto	Correto	Correto
Q3	Correto	Correto	Correto	Correto
Q4	Correto	Correto	Correto	Correto
Q5	Correto	Correto	Correto	Correto
Q6	Correto	Correto	Incorreto	Correto

	Correto
✗	Incorreto

Fonte: O Autor, 2020

5.5.1.4 Observações realizadas durante a aula 01

Antes das explicações sobre os DMIs foi perguntado aos estudantes se conheciam o que eram DMIs. Alguns estudantes relataram que controladores MIDI e sintetizadores poderiam ser DMIs.

Sobre o segundo momento da aula os estudantes foram questionados se já conheciam e se haviam manuseado a placa Arduino. Três estudantes (E1, E2 e E3) já conheciam a placa Arduino, mas nunca haviam manuseado, já o estudante E4 não conhecia a Placa Arduino. Os estudantes E1 e E2 possuíam mais facilidade em gravar o código na placa Arduino, pois realizaram a atividade de forma rápida conseguindo realizar alterações. Também realizaram questionamentos sobre a linguagem de programação como: “A linguagem utilizada no Arduino é a linguagem C?”, demonstrando familiaridade com o código apresentado.

O estudante E3 relatou que não era difícil gravar o código na placa Arduino e que era simples modificar o código de exemplo para piscar o LED. Já o estudante E4 teve algumas dificuldades no momento de gravar o código, pois o computador não reconheceu o cabo USB na primeira vez e apresentou uma mensagem de erro na tela. Isso foi corrigido conectando o USB em outra porta do computador.

5.5.1.5 Considerações sobre a aula

Ao final desta aula todos os estudantes conseguiram gravar o código de exemplo Blink do Arduino e posteriormente conseguiram modifica-lo para que o LED piscasse em frequências diferentes. Com isso foi possível verificar que todos os estudantes atingiram os

resultados de aprendizagem, uma vez que souberam diferenciar os DMIs de não DMIs, identificar os componentes da placa Arduino e gravar o código na placa Arduino sem muitas dificuldades.

5.5.2 Aula 02 - Reproduzindo Notas Musicais

5.5.2.1 Resultados de Aprendizagem da Aula 02

A segunda aula do curso teve como resultados de aprendizagem (R.A):

- R.A.2.1: Conectar o Buzzer na placa Arduino;
- R.A.2.2: Utilizar o comando #define;
- R.A.2.3: Configurar uma porta como saída;
- R.A.2.4: Utilizar o comando tone para reproduzir notas musicais.
- R.A.2.5: Utilizar o comando delay para pausar o programa;

5.5.2.2 Estrutura da Aula 02:

Tabela 20 – Estrutura da Aula 02

Momento	Tópicos	Estratégia Instrucional	Recursos
1º	Buzzer	Apresentação, imagens do Buzzer e montagem na placa Arduino	Slides, Placa Arduino, e Buzzer
2º	Aferição do entendimento sobre o Buzzer	4 Perguntas com 4 alternativas cada	Kahoot03
3º	Código do Buzzer	Apresentação, imagens com código de exemplo	Slides, Placa Arduino, Buzzer e Computador com IDE do Arduino
4º	Aferição do entendimento sobre o código do Buzzer	8 Perguntas com 4 alternativas cada.	Kahoot04

5º	Programar música de sua escolha	Atividade prática para programar uma música	Slides, Placa Arduino, Buzzer e Computador com IDE do Arduino
----	---------------------------------	---	---

Fonte: O Autor, 2020

5.5.2.3 Resultados dos Kahoots 03 e 04

Nesta aula foram realizados 2 Kahoots, o primeiro para o funcionamento do Buzzer e o segundo para o código para reproduzir notas musicais com o Buzzer na placa Arduino. Nessa seção serão analisados resultados de cada um deles.

5.5.2.3.1 Kahoot #03 – Buzzer

Objetivo: Verificar se os estudantes aprenderam como utilizar o Buzzer com placa Arduino.

Conteúdo do Kahoot 3: Este Kahoot foi composto de quatro questões referentes ao sensor Buzzer, as quais estão descritas na Tabela 21. Cada questão foi direcionada para que os estudantes aprendessem sobre o funcionamento do Buzzer, como utiliza-lo com a placa Arduino visando que os estudantes tivessem mais familiaridade com o componente antes da atividade prática. Vale ressaltar que cada estudante possuía um sensor Buzzer em suas mãos entregue durante a aula teórica, podendo assim consulta-lo para responder as questões se achasse necessário.

Tabela 21 – Perguntas do Kahoot 03

Nº	Questão	Alternativas	
Q3.1	Como identificar qual terminal do Buzzer é positivo e negativo?	A) Através das cores dos terminais	B) Testando na prática
		C) Retirando o Ade-sivo	D) Através do símbolo de +
Q3.2	Quais portas do Arduino poderiam ser utilizadas?	A) (GND, 12)	B) (GND, 11)
		C) (GND, 3)	D) (AREF, 10)
Q3.3	Porque deve-se utilizar as portas PWM com o Buzzer?	A) Porque elas ligam e desligam o Buzzer	B) Porque são as portas que cabem o Buzzer
		C) Porque variam a frequência	D) Porque são mais rápidas

Q3.5	Porque o Buzzer é chamado de sensor de saída?	A) Porque o Buzzer liga e desliga	B) Porque o Buzzer recebe um sinal e emite um som.
		C) Porque o Buzzer varia a frequência	D) Porque o Buzzer envia um sinal para o Arduino.

Fonte: O Autor, 2020

Resultados: Através da análise da Tabela 22 é possível verificar que os estudantes conseguiram compreender como utilizar o Buzzer, uma vez que responderam corretamente a maioria das questões. Um dos estudantes respondeu corretamente todas as questões e os outros obtiveram 1 erro entre as 4 questões.

As questões respondidas de forma incorreta Q3.2 e Q3.3 foram explicadas novamente e demonstrado quais eram as respostas corretas, principalmente na questão Q3.2 a qual se refere a uma funcionalidade importante para a experiência prática. Na questão Q3.3 era esperado que os estudantes respondessem a alternativa “Porque variam a frequência”, porém o estudante E3 respondeu “Porque elas ligam e desligam o Buzzer”, e o estudante E4 respondeu “Porque são as portas que cabem o Buzzer.”.

Tabela 22 – Resultados do Kahoot 03

Kahoot 3				
	E1	E2	E3	E4
Q1	Correto	Correto	Correto	Correto
Q2	Incorreto	Correto	Correto	Correto
Q3	Correto	Correto	Incorreto	Incorreto
Q4	Correto	Correto	Correto	Correto

Correto
Incorreto

Fonte: O Autor, 2020

Após a conclusão deste Kahoot os estudantes conectaram o Buzzer ao Arduino e foi dada continuidade na explicação teórica, dando início ao conteúdo sobre o código para o Buzzer.

5.5.2.3.2 Kahoot #04 – Código Buzzer

Objetivo: Verificar a compreensão dos estudantes sobre as funções base do Arduino e como programar o Buzzer para emitir notas musicais.

Conteúdo do Kahoot 4: Este Kahoot foi composto por oito questões referentes ao código para reproduzir notas musicais com o Buzzer. Como pode ser visualizado na Tabela 20 as questões são relacionadas aos códigos base do Arduino (Q4.1 e Q4.2) e às funções necessárias para reproduzir notas musicais com o Buzzer (Q4.4, Q4.6 e Q4.7) bem como os intervalos de tempo necessários (Q4.5) e as configurações iniciais (Q4.3). Além disso buscou-se verificar o raciocínio lógico dos estudantes através da questão Q4.8, que possui um código que foi utilizado como exemplo na aula, mas que contém uma alteração. Esta foi a primeira experiência dos estudantes programando o Arduino do zero, por este motivo optou-se por verificar sua compreensão dos temas listados.

Tabela 23 – Perguntas do Kahoot 04

Nº	Questão	Alternativas	
Q4.1	Para que serve a função setup?	A) É utilizada para configurações iniciais	B) É utilizada para instalar o Arduino.
		C) É utilizada para ligar o Buzzer	D) É utilizada para executar o pinMode.
Q4.2	Para que serve a função loop?	A) É utilizada para configurar o Buzzer	B) É utilizada para executar a lógica.
		C) Não é utilizada	D) É utilizada para ligar a função tone
Q4.3	Quais parâmetros do pinMode configura o Buzzer como saída?	A) pinMode(Buzzer, OUTPUT)	B) pinMode (OUTPUT, Buzzer)
		C) pinMode(Buzzer, INPUT)	D) pinMode(Buzzer, SAIDA).
Q4.4	Para que é utilizada a função tone?	A) Pausar o programa por 1 segundo	B) Desligar o Buzzer por 1 segundo
		C) Gerar uma determinada frequência	D) Ligar e desligar o Buzzer
Q4.5	Quantos segundos a função delay (15000) pausa o programa?	A) 15 milissegundos	B) 1500 segundos.
		C) 1,5 segundos	D) 15 segundos.
Q4.6	Para que é utilizado o 3º parâmetro da função tone?	A) Para gerar uma frequência	B) Pausar o programa por um tempo.

		C) É a duração que a frequência será executada	D) É o pino do Arduino que será executado.
Q4.7	Para que é utilizada a função noTone?	A) Desligar o Buzzer	B) Verifica se o Buzzer está funcionando.
		C) Gerar uma nota no Buzzer	D) Pausar o programa por um tempo.
Q4.8	<p>O que esse código faz?</p> <pre>void loop(){ tone(buzzer,NOTE_C3,1000); delay(1000); noTone(buzzer); delay(1000); }</pre>	A) O código dá um erro na verificação	B) Não funciona porque está dentro da função
		C) Executa nota C3 no Buzzer apenas 1 vez	D) Executa a nota C3 várias vezes

Fonte: O Autor, 2020

Resultados: Como é possível verificar na Tabela 24 entre as questões listadas apenas a questão Q4.6 foi compreendida por todos os estudantes. As questões Q4.1, Q4.3 e Q4.4 foram respondidas incorretamente por apenas um estudante (E4). Já a questão Q4.5 e Q4.7 dois dos estudantes não conseguiram compreendê-las.

De forma geral este foi o questionário que teve o maior número de respostas incorretas, contendo 1 questão em que todos os estudantes não conseguiram compreender (Q4.8) e 1 questão (Q4.2) em que apenas um estudante (E2) respondeu corretamente. Todos os estudantes obtiveram seu menor desempenho neste Kahoot e ainda um dos estudantes (E4) obteve um desempenho muito abaixo respondendo corretamente apenas 2 questões entre 8.

Tabela 24 – Resultados do Kahoot 04

Kahoot 4				
	E1	E2	E3	E4
Q1	Correto	Correto	Correto	Incorreto
Q2	Incorreto	Correto	Incorreto	Incorreto
Q3	Correto	Correto	Correto	Incorreto
Q4	Correto	Correto	Correto	Incorreto
Q5	Correto	Incorreto	Incorreto	Correto
Q6	Correto	Correto	Correto	Correto
Q7	Incorreto	Incorreto	Incorreto	Incorreto
Q8	Incorreto	Incorreto	Incorreto	Incorreto

Correto	Correto
Incorreto	Incorreto

Fonte: O Autor, 2020

Ao final deste Kahoot foi dada continuidade a aula a qual os estudantes realizaram a atividade prática de programar o Buzzer para executar notas musicais. Após testarem diferentes notas musicais foi pedido como atividade para que os estudantes programassem o Arduino para tocar uma música de sua escolha.

5.5.2.4 Observações realizadas durante a aula 02

Durante a atividade prática do código do Buzzer o estudante E2 realizou os seguintes questionamentos sobre o código: “O comando “#define” é o mesmo da linguagem C?”, “Para comentar o código usa os símbolos “//”?”, “É possível utilizar os símbolos “/* */” para comentar um bloco do código?”. Tais questionamentos foram esclarecidos pelo autor. Isso confirma as informações fornecidas pelo estudante E2 no formulário de inscrição, em que o estudante informou que possuía conhecimentos prévios. Quanto a atividade prática para programar uma música utilizando os conhecimentos adquiridos o estudante procurou alterar o código utilizando exemplos que encontrou no site do Arduino, porém não obteve sucesso em gravar o código, pois o código encontrado não tinha as funções "setup" e "loop". Isso foi corrigido adicionando as funções ao código após pedir ajuda para o autor. Com isso os estudantes E1 e E3 utilizaram das explicações para comentar seus códigos para gravarem códigos de exemplos encontrados na internet. Isso demonstra que os estudantes estavam interagindo entre si para conseguir realizar a atividade e conhecer melhor a plataforma do Arduino. Após as explicações os estudantes E1, E2 e E3 conseguiram encontrar códigos prontos de músicas como Asa Branca, Tema do Super Mario, Tema de Star Wars, entre outras que gostavam, conseguindo gravar os códigos e reproduzir utilizando o Buzzer.

O estudante E4 apresentou um erro com a IDE do Arduino, a qual não reconheceu a placa Arduino ao conectar o USB ao computador, o que foi solucionado alterando a porta USB do computador. Quando copiou o código da explicação repetiu as funções Setup e Loop, o que gerou um erro de sintaxe na IDE do Arduino durante a gravação. Isso foi corrigido apagando as linhas “Setup” e “Loop” repetidas. Mesmo após a correção o Buzzer não reproduziu a nota programada, então o estudante perguntou novamente sobre os terminais do Buzzer, “Qual era o positivo e negativo?”. Após a explicação o estudante verificou que conectou o Buzzer de forma errada, então ao corrigir o problema conseguiu executar as notas musicais programadas.

Ao fazer alguns testes o estudante E4 identificou que as notas musicais programadas não reproduziam as notas musicais corretas. Utilizando o celular com um aplicativo de afinação percebeu que a nota programa C3 reproduzia na verdade a nota B7. Para corrigir o problema primeiro foi removido o adesivo de cima do Buzzer, o que alterou a nota para D3, mas não corrigiu o problema. Então foi trocado o Buzzer por outro, o que fez com que a nota musical programada se aproximasse da nota C3 no afinador.

Outra questão levantada pelos estudantes foi que quando duas notas eram programadas apenas uma nota musical podia ser ouvida. O que foi corrigido alterando os valores do “Delay” e da duração na função tone. A partir disso os estudantes identificaram quais valores para o “Delay” e para a duração na função “tone” funcionariam melhor para seus experimentos.

5.5.2.5 Considerações sobre a aula

Esta foi a primeira vez em que todos os estudantes programaram a placa Arduino com um código iniciado do zero, ou seja cada linha de código foi escrita pelos estudantes seguindo o exemplo dado na aula teórica. Durante a atividade prática os estudantes gravaram o código de exemplo e testaram se estava correto, com isso as dúvidas geradas foram respondidas como relatado anteriormente até que todos os estudantes conseguissem realizar corretamente as atividades.

Na atividade prática os estudantes poderiam escolher qualquer música para programar na placa Arduino e verificar o funcionamento do código, isso gerou dúvidas nos estudantes sobre qual música escolher, por este motivo pesquisaram na internet músicas já prontas. A programação dessa música escolhida foi iniciada nesta aula (Aula 02), porém foi concluída no primeiro momento da aula seguinte (Aula 03).

Como os estudantes possuem maior familiaridade com o tema Música, foi possível perceber aspectos musicais que não haviam sido planejados que os estudantes contribuíram para melhoria das aulas. Os estudantes utilizaram um aplicativo de afinação² no celular

² Existem diversos aplicativos de afinação, cada estudante utilizou um aplicativo que já estava familiarizado, como: Cifra Club Tuner, Fine Chromatic Tuner e Guitar Tuner, Bass, Violin.

para identificar quais notas estão sendo tocadas, podendo conferir quais notas estão sendo reproduzidas. Com isso foi identificado que um dos pacotes de Buzzer que foram adquiridos possuem uma qualidade inferior ao outro em relação a reprodução das notas musicais. Outro aspecto em que os estudantes contribuíram foram as definições de tempo para cada nota, como semibreve, mínima, semínima, colcheia, semicolcheia.

Todos os estudantes conseguiram alcançar um bom resultado de aprendizagem, uma vez que além de programar o exemplo apresentado, iniciaram a atividade programando uma música que gostavam e também adicionaram seus conhecimentos musicais na atividade contribuindo com a qualidade da aula em si.

Ao final todas as placas de Arduino foram recolhidas com Buzzer removido para que na próxima aula os estudantes montassem novamente o circuito com a intenção de fixar o conhecimento adquirido. Também foi observado que os estudantes E1, E2 e E3 demonstraram grande interesse na placa Arduino ao perguntarem sobre preços, tipos e locais para compra.

5.5.3 Aula 03 - Qualidade do Som

5.5.3.1 Resultados de Aprendizagem da Aula 03.

- R.A.3.1: Utilizar fone de ouvidos no circuito;
- R.A.3.2: Montar circuitos na protoboard;
- R.A.3.3: Utilizar resistores de diferentes valores;
- R.A.3.4: Utilizar potenciômetro no circuito.

5.5.3.2 Estrutura da Aula 03.

Tabela 25 – Estrutura da Aula 03

Momento	Tópicos	Estratégia Instrucional	Recursos
1º	Programar música de sua escolha	Finalização da Atividade prática da aula anterior	Slides, Placa Arduino, Computador com IDE do Arduino e Buzzer.

2º	Conector Jack J2, Protoboard e Resistor	Apresentação, Imagens dos circuitos em protoboard	Slides, Placa Arduino, Jack J2, Protoboard, Resistores, Fios, e fones de ouvido
3º	Aferição do entendimento sobre Jack J2, Protoboard, Resistores e Fios.	9 Perguntas, sendo 4 de verdadeiro ou falso e 5 com 4 alternativas.	Kahoot
4º	Utilizar fones de ouvido no Arduino	Atividade prática para montar o circuito com o jack j2, fone de ouvido, resistores e protoboard	Slides, Placa Arduino, Jack J2, Protoboard, Resistores, Fios e fones de ouvido.
5º	Potenciômetro	Apresentação, imagens do potenciômetro e discussão	Slides, Placa Arduino, Jack J2, Protoboard, Resistores, Fios, fones de ouvido e potenciômetro.
6º	Aferição do entendimento sobre potenciômetro	8 Perguntas com 4 alternativas cada.	Kahoot

7º	Controle do volume	Atividade prática para montar o potenciômetro com circuito anterior.	Slides, Placa Arduino, Jack J2, Protoboard, Resistores, Fios, fones de ouvido e potenciômetro.
----	--------------------	--	--

Fonte: O Autor, 2020

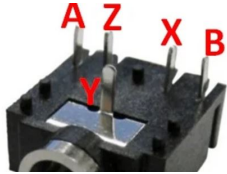
5.5.3.3 Resultados dos Kahoots 05 e 06

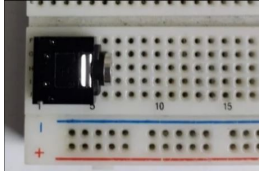
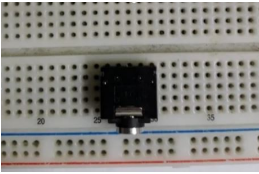
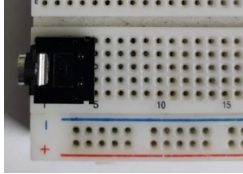
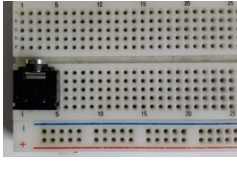
5.5.3.3.1 Kahoot #05 – Jack J2 e protoboard.


Objetivo: Verificar a compreensão dos estudantes sobre o uso do conector Jack J2 na protoboard e sobre o funcionamento do resistor?

Conteúdo do Kahoot 5: Neste Kahoot os estudantes responderam nove questões sobre os componentes necessários para melhorar a qualidade do som. Como pode ser verificado na Tabela 23 as questões são referentes ao conector Jack J2 (Q5.1 a Q5), procurando reforçar em como utilizar a protoboard de maneira correta e quais terminais devem ser utilizados para reproduzir sons de maneira correta. Outro tópico abordado é o de funcionamento dos resistores (Q5.7 a Q5.9) utilizados no circuito para redução do volume. Também foi questionado sobre as cores de fios utilizados (Q5.6), pois é considerado uma boa prática manter padrões de cores para montagem de circuitos eletrônicos.

Tabela 26 – Perguntas do Kahoot 05

Nº	Questão	Alternativas	
Q5.1	O conector J2 possui 2 terminais duplicados, quais são eles? 	A) (A e B ; Z e X)	B) (A e Y ; B e Y)
		C) (Z e Y ; X e Y)	D) (A e Z ; X e B)

Q5.2	<p>O conector nesta posição irá funcionar.</p> 	Verdadeiro	Falso
Q5.3	<p>O conector nesta posição irá funcionar.</p> 	Verdadeiro	Falso
Q5.4	<p>O conector nesta posição irá funcionar.</p> 	Verdadeiro	Falso
Q5.5	<p>O conector nesta posição irá funcionar.</p> 	Verdadeiro	Falso
Q5.6	<p>Quais cores de fios são utilizados para os terminais positivo e negativo respectivamente?</p>	A) Azul e Preto	B) Vermelho e Preto
		C) Azul e Vermelho	D) Vermelho e Roxo

Q5.7	Para qual finalidade é utilizado um resistor entre o conector j2 e o Arduino?	A) Para diminuir o volume	B) Para facilitar a conexão
		C) Para aumentar o volume	D) Para aumentar a potência do som
Q5.8	Porque o resistor diminui o volume?	A) Porque gera energia	B) Porque diminui a corrente elétrica do Arduino
		C) O resistor aumenta o volume	D) Porque o Arduino sabe que precisa diminuir o som.
Q5.9	Como fica a ordem dos resistores do MENOR para o MAIOR? 	A) 10kΩ, 100Ω, 1kΩ	B) 1kΩ, 100Ω, 10kΩ
		C) 100Ω, 1kΩ, 10kΩ	D) 100Ω, 10kΩ, 1kΩ

Fonte: O Autor, 2020

A questão Q5.1 apresentou uma imagem contendo o conector Jack J2 com os seus terminais nomeados, desta forma os estudantes responderam utilizando a imagem e o conector em suas mãos. Nas questões Q5.2, Q5.3, Q5.4 e Q5.5 foram apresentadas imagens do Jack J2 conectado na protoboard em diversas posições, assim foi respondido “Verdadeiro” para as conexões que estavam corretas e “Falso” para conexões incorretas. As questões Q5.6, Q5.7 e Q5.8 são referentes as funcionalidades dos componentes. Já a questão Q5.9 apresentou uma imagem contendo 3 resistores diferentes os quais deveriam ser colocados em ordem de grandeza.

Resultados: Como resultado era esperado que os estudantes compreendessem ao menos a lógica de funcionamento da protoboard, pois todos os componentes são

dependentes da protoboard para realização das atividades práticas. Também foi levado em conta que nenhum dos estudantes tinha experiência prévias com componentes eletrônicos, principalmente a protoboard.

Ao observar a Tabela 27 é possível verificar que este Kahoot teve um melhor desempenho do que o Kahoot 4 (Tabela 24) mesmo que o assunto fosse mais complexo num primeiro momento. As questões respondidas incorretamente foram Q5.1, Q5.5, Q5.6 e Q5.9.

Tabela 27 – Resultados do Kahoot 05

Kahoot 5				
	E1	E2	E3	E4
Q1	Correto	Correto	Correto	Incorreto
Q2	Correto	Correto	Correto	Correto
Q3	Correto	Correto	Correto	Correto
Q4	Correto	Correto	Correto	Correto
Q5	Incorreto	Correto	Correto	Incorreto
Q6	Correto	Correto	Incorreto	Correto
Q7	Correto	Correto	Correto	Correto
Q8	Correto	Correto	Correto	Correto
Q9	Correto	Incorreto	Correto	Incorreto

	Correto
	Incorreto

Fonte: O Autor, 2020

Apenas 1 estudante (E4) respondeu incorretamente à questão Q5.1, a qual foi explicada novamente para reduzir erros na montagem do circuito. A Questão Q5 foi respondida incorretamente por 2 estudantes (E1 e E4), a qual trata-se de uma conexão do Jack J2 que em teoria pode funcionar, porém dificulta a montagem, por este motivo foi considerada incorreta. Na questão Q5.6 apenas 1 estudante (E3) respondeu incorretamente. Para a questão Q5.6 foi obtido como resposta “Vermelho e Roxo” sendo que em nenhum momento foi mencionado a cor roxo para os fios, enquanto a resposta correta seria "Vermelho e Preto". Por fim 2 estudantes (E2 e E4) responderam incorretamente à questão Q5.9 a qual trata dos valores dos resistores. Foram obtidos como resposta de Q5.9 “10K, 100R e 1K”. Observando a figura utilizada na questão Q5.9 é possível verificar que os estudantes responderam na ordem que estavam visualizando, porém a pergunta se tratava da ordem de grandeza da menor resistência para a maior resistência.

É possível dizer que os estudantes compreenderam os assuntos abordados neste Kahoot visto que todos responderam corretamente as questões Q5.2, Q5.3, Q5.4, Q5.7 e

Q5.8, e para as questões respondidas incorretamente obtiveram novamente a explicação durante a execução do Kahoot.

Após a finalização do Kahoot 5 os estudantes realizaram a atividade prática de montar um circuito com o conector Jack J2, um resistor e o fone de ouvido na protoboard. Ao finalizar esta etapa foi dada continuidade a aula teórica que abordou uma introdução aos conceitos relacionados ao potenciômetro.



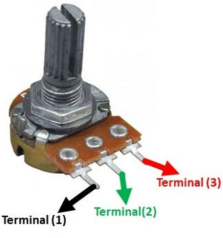
5.5.3.3.2 Kahoot #06 – Potenciômetro

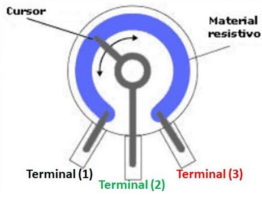
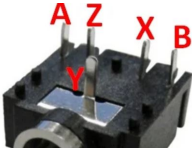
Objetivo: Verificar se os estudantes compreenderam como utilizar resistores e potenciômetro para reduzir o volume.

Conteúdo do Kahoot 6: Este Kahoot foi composto por oito questões envolvendo os conceitos necessários para controlar o volume dos fones de ouvido através da variação de resistência. Como pode ser visualizado na Tabela 28 as primeiras questões (Q6.1 a Q6.4) são referentes ao funcionamento do resistor para controlar o volume dos fones de ouvido, sendo que as questões Q6.1 e Q6.2 procuram reforçar o conceito sobre qual resistor possui uma resistência maior ou menor. As questões Q6.3 e Q6.4 foram compostas de imagens de resistores com o intuito de identificar as resistências através das cores dos resistores. Nas questões Q6.5, Q6.6 e Q6.7 tratam das funcionalidades de um potenciômetro, já a questão Q6.9 foi composta de uma imagem do Jack J2 com uma nova configuração para reproduzir sons dos dois lados do fone de ouvido.

Tabela 28 – Perguntas do Kahoot 06

Nº	Questão	Alternativas	
Q6.1	O que acontece quando se troca o resistor de $1k\Omega$ por um $10k\Omega$?	A) O volume aumenta	B) O volume continua o mesmo.
		C) O volume diminui	D) O volume começa baixo mas depois aumenta.
Q6.2	O que acontece quando troca-se o resistor de $1k\Omega$ por um de 100Ω ?	A) O volume aumenta	B) O volume continua o mesmo.
		C) O volume diminui	D) O volume começa baixo mas depois aumenta.

Q6.3	<p>Qual é o valor do resistor?</p> 	A) 10kΩ	B) 1MΩ
Q6.4	<p>Qual é o valor do resistor?</p> 	A) 10kΩ	B) 1MΩ
Q6.5	O que é um potenciômetro?	A) Sensor de energia	B) Resistor variável.
Q6.6	Quais terminais são necessários para controlar o volume?	C) Controlador do Arduino	D) Reprodutor de áudio
Q6.7	<p>Se o potenciômetro estiver na resistência máxima o cursor estará em:</p> 	A) 1 e 3	B) 1, 2 e 3
Q6.7	Se o potenciômetro estiver na resistência máxima o cursor estará em:	C) 2 e 3	D) 1 e 2
Q6.7	Se o potenciômetro estiver na resistência máxima o cursor estará em:	A) No terminal 1 a esquerda	B) No terminal 3 a direita

		C) No terminal 2 no centro	D) Em qualquer posição
Q6.8	<p>Quais terminais são utilizados para que o som saia pelos 2 lados?</p> 	A) A e Z	B) Z e Y
		C) A e B	D) B e Y

Fonte: O Autor, 2020

Resultados: Como pode-se observar na Tabela 29 todos os estudantes responderam corretamente as questões Q6.4, Q6.5 e Q6.7. Também é verificado que os estudantes E1, E2, e E4 responderam incorretamente apenas 1 questão e o estudante E3 apenas 2 questões. Para as questões Q6.1 e Q6.2 era esperado que os estudantes compreendessem a relação entre a resistência e o volume, como observado, dois estudantes (E3 e E4) responderam incorretamente. Na questão Q6.3 foi obtido 1 resposta incorreta (E3) a qual trata-se de uma questão considerada mais complexa devido a lógica para leitura das cores dos resistores. Na questão Q6.6 foi perguntado sobre como utilizar o potenciômetro para controlar o volume de forma correta, apenas 1 estudante respondeu incorretamente (E2). Por fim a questão Q6.8 está relacionada ao Jack J2 que já havia sido utilizado anteriormente em outra configuração, porém um estudante (E1) respondeu de forma incorreta.

Tabela 29 – Resultados do Kahoot 06

Kahoot 6				
	E1	E2	E3	E4
Q1	Correto	Correto	Correto	Incorreto
Q2	Correto	Correto	Incorreto	Correto
Q3	Correto	Correto	Incorreto	Correto
Q4	Correto	Correto	Correto	Correto
Q5	Correto	Correto	Correto	Correto
Q6	Correto	Incorreto	Correto	Correto
Q7	Correto	Correto	Correto	Correto
Q8	Incorreto	Correto	Correto	Correto

Correto	Correto
Incorreto	Incorreto

Fonte: O Autor, 2020

Após a realização deste Kahoot os estudantes realizaram a atividade prática montando o circuito contendo os resistores de diferentes valores e o potenciômetro para controlar o volume.

5.5.3.4 Observações realizadas durante a aula 03.

O início desta aula foi utilizado para que os estudantes concluíssem a programação da música escolhida na aula anterior. Com a placa Arduino e o Buzzer em mãos os estudantes interagiram entre si para identificar os terminais do Buzzer para conecta-lo à placa Arduino.

O estudante E2 concluiu a programação da música mais rapidamente, o que permitiu que organizasse o código, o qual utilizou valores para cada tempo das notas musicais. Assim o estudante E2 representou tais valores utilizando o comando “#define” com os nomes dos tempos como por exemplo: “#define SEMIBREVE 1000”, “#define MINIMA 500”. O que gerou curiosidade dos estudantes E1 e E3 que utilizaram a mesma forma de representar os tempos das notas. Com isso os estudantes perceberam que o código estava muito grande com todos os comandos “#define” para as notas, assim encontraram um arquivo chamado “pitches.h” disponível no site do Arduino. Este arquivo trata-se de um arquivo “header”, utilizado para organizar o código, onde é programado o cabeçalho do código, ou seja as definições iniciais, neste caso o “pitches.h” contém todas as notas reunidas em um só lugar com os “#define” pré-programados,. Assim os estudantes E1, E2 e E3 utilizaram o arquivo “pitches.h” contendo as notas musicais e o alteraram acrescentando os seus próprios códigos feitos durante a aula. Após os 30 minutos iniciais da aula todos os estudantes haviam concluído a atividade de programar uma música no Arduino. Assim, os estudantes enviaram seus códigos por e-mail.

Na atividade desta aula os estudantes montaram pela primeira vez um circuito na protoboard. Os estudantes conseguiram utilizar facilmente a protoboard, ao realizar a atividade prática montaram o circuito com diferentes resistores para testar o volume reproduzido pelo Arduino e posteriormente trocaram o resistor pelo potenciômetro.

Nos primeiros momentos os estudantes cometeram alguns erros como conectar os componentes em uma trilha da protoboard que não estava ligada, ou esquecer de conectar um fio, porém esse tipo de erro era esperado por se tratar da primeira vez utilizando um novo componente. Nesta etapa foi demonstrado como encontrar erros no circuito através de uma revisão do circuito montado. Outras vezes o projeto não funcionava, mas as conexões estavam corretas, então foi constatado que os conectores jack J2 não fixavam muito bem na protoboard e se soltavam quando eram colocados os fones de ouvido. Isso gerou uma rotina por parte dos estudantes de sempre verificar se o conector não estava com um mal contato e se todo o circuito estava correto.

5.5.3.5 Considerações sobre a aula

Esta foi a primeira vez que os estudantes utilizaram a protoboard e não foram utilizados componentes de kits didáticos como Lego e módulos de sensores, por isso acreditava-se que teriam grandes dificuldades para montar o circuito, porém todos os estudantes conseguiram realizar as atividades sem muitas dificuldades. A protoboard foi utilizada nesta etapa devido a necessidade de acrescentar mais componentes no circuito como o Jack J2, potenciômetro e resistores, uma vez que somente com o Arduino não é possível montar circuitos mais complexos. Ao final da atividade os estudantes reproduziram a música que programaram anteriormente, agora utilizando fones de ouvido.

Ao final desta aula os estudantes responderam a um formulário que continha uma pergunta: “Este projeto que foi desenvolvido é um DMI? Se não, o que falta para ser?” e foram obtidas as seguintes respostas:

- E1: “Falta uma interface para que os sons sejam reproduzidos digitalmente”;
- E2: “É necessário que ele seja sensível ao toque / movimentos. Sem a necessidade de programar as notas antes de cada música”;
- E3: “É necessário a manipulação dos sons a serem processados pelo Arduino, manipular a saída e também o valor da corrente a ser reproduzida pelo dispositivo de saída”;
- E4: “Ele já é um DMI, porém pro projeto final estar pronto falta alguns ajustes”.

Os estudantes E2 e E3 responderam de forma correta. Nesta etapa ainda não se tinha um DMI, faltava um sinal de entrada que controla em tempo real os sons. Na aula seguinte isso foi discutido novamente e as dúvidas foram esclarecidas. Pode-se perceber que alguns estudantes responderam sobre o meio digital como saída, não considerando o Arduino como tal.

5.5.4 Aula 04 - Botões

5.5.4.1 Resultados de Aprendizagem da Aula 04

- R.A.4.1: Conectar botões na protoboard;
- R.A.4.2: Programar o Arduino para ler valores dos botões;
- R.A.4.3: Reproduzir notas musicais em tempo real.

5.5.4.2 Estrutura da Aula 04

Tabela 30 – Estrutura da Aula 04

Momento	Tópicos	Estratégia Instrucional	Recursos
1º	Botões	Apresentação, imagens do botão, conexão na protoboard e código necessário.	Slides, Placa Arduino, Jack J2, Protoboard, Resistores, Fios, fones de ouvido, potenciômetro, Computador com IDE Arduino e botão.
2º	Aferição do entendimento sobre os Botões.	9 Perguntas com 4 alternativas cada.	Kahoot

3º	Construindo primeiro DMI	Atividade prática para montar 5 botões na protoboard e programa-los para executar uma nota cada um.	Slides, Placa Arduino, Jack J2, Protoboard, Resistores, Fios, fones de ouvido, potenciômetro, Computador com IDE Arduino e 5 botões
----	--------------------------	---	---

Fonte: O Autor, 2020

5.5.4.3 Resultados do Kahoot 07

5.5.4.3.1 Kahoot #07 – Botões

Objetivo: Verificar se os estudantes compreenderam como montar os circuitos contendo botões e como programa-los no Arduino.

Conteúdo do Kahoot 7: Este Kahoot é composto por nove questões envolvendo os conceitos sobre o botão. Como pode ser verificado na Tabela 31 as questões estão divididas entre os assuntos sobre o circuito eletrônico (Q7.1 a Q7.5) e a programação (Q7.6 a Q7.9). As questões Q7.1 e Q7.2 são referentes ao funcionamento do botão e foi utilizado o nome “Pushbutton” no lugar de “botão” por ser um nome amplamente utilizado em manuais, tutoriais, lojas de componentes, etc. Na questão Q7.3 é retomado o assunto do resistor, porém nesta etapa funcionando em conjunto com o botão. Nas questões Q7.4 e Q7.5 são relacionadas as portas do Arduino que podem ser utilizadas em conjunto do botão. Nas questões envolvendo programação foram realizadas perguntas sobre as principais funções e símbolo necessários para utilizar o botão no Arduino.

Tabela 31 – Perguntas do Kahoot 07

Nº	Questão	Alternativas	
Q7.1	No Arduino que tipo e sensor é um Pushbutton?	A) Sensor de entrada	B) Sensor de saída
		C) Sensor de interação	D) Sensor de pressão

Q7.2	O Arduino interpreta os comandos do Pushbutton como...	A) Sinal analógico, ou seja valor 0 ou 1	B) Sinal digital, ou seja valor 0 até 1023
		C) Sinal analógico, ou seja valor 0 até 1023	D) Sinal digital, ou seja valor 0 ou 1
Q7.3	Qual a diferença entre o circuito do botão com resistor e sem o resistor?	A) Com resistor o circuito é mais eficiente	B) Sem o resistor o circuito é mais eficiente.
		C) Não há diferença na prática, apenas facilita a montagem	D) Depende do objetivo do projeto.
Q7.4	Quais desses terminais do Arduino podem ser utilizados para ligar o botão?	A) (1, GND)	B) (GND, 8)
		C) (5V, 8)	D) (AREF, GND)
Q7.5	Quais desses terminais do Arduino NÃO podem ser utilizados para ligar o botão?	A) (1, 5V)	B) (GND, A2)
		C) (2, GND)	D) (GND, 11)
Q7.6	Para que é utilizado parâmetro INPUT_PULLUP?	A) Para utilizar o resistor interno do Arduino no botão	B) Para utilizar um resistor com o botão
		C) Para identificar o botão como entrada	D) Para configurar o botão como digital.
Q7.7	Qual símbolo utilizado para inverter o valor de uma função?	A) A) (barra) /	B) (Letra N) N
		C) (asterisco) *	D) (exclamação) !
Q7.8	O que faz a função digitalRead?	A) Aciona uma nota no fone	B) Le um valor analógico em um pino.
		C) Le um valor digital em um pino	D) Envia um sinal para um pino

Q7.9	O que faz o símbolo * na função?	A) Multiplica o valor NOTE_C3 com digitalRead(botão)	B) Divide o valor NOTE_C3 com digitalRead(botão)
		C) Soma o valor NOTE_C3 com digitalRead(botão)	D) Subtrai o valor NOTE_C3 com digitalRead(botão)

Fonte: O Autor, 2020

Resultados: Observando a Tabela 32 é possível verificar que em uma questão (Q7.3) todos os estudantes responderam incorretamente. Nesta questão as respostas obtidas foram “Com o resistor o circuito é mais eficiente” respondidas pelos estudantes E2, E3 e E4, e “Sem o resistor o circuito é mais eficiente” pelo estudante E1, porém a resposta correta não está ligada a eficiência do circuito, mas sim com a facilidade da montagem, tendo como resposta correta “Não há diferença na prática, apenas facilita a montagem”. A questão Q7.1 obteve 1 resposta incorreta (E4) assim como a questão Q7.4 (E1). Na questão Q7.2 dois estudantes responderam incorretamente (E1 e E3), obtendo como resposta “Sinal analógico, ou seja valor 0 ou 1” quando era esperado a resposta “Sinal digital, ou seja valor 0 ou 1”. O que demonstra uma confusão com os termos Sinal digital e Sinal analógico. A questão Q7.5 foi respondida incorretamente por 2 estudantes (E3 e E4) com respostas “(GND, A2)” relatando que as portas analógicas dos Arduino não poderiam ser utilizadas com o botão, o que não ocorre na prática, desta forma a resposta correta seria a alternativa com os valores “(1, 5V)”. Por fim as questões Q7.6 foram respondidas incorretamente por 2 estudantes (E3 e E4), obtendo como respostas “Para identificar o botão como entrada”, a qual é uma resposta verdadeira, porém incompleta, desta forma a resposta correta seria “Para utilizar o resistor interno do Arduino no botão”.

Tabela 32 – Resultados do Kahoot 07

Kahoot 7				
	E1	E2	E3	E4
Q1	Correto	Correto	Correto	Incorreto
Q2	Incorreto	Correto	Incorreto	Correto
Q3	Incorreto	Incorreto	Incorreto	Incorreto
Q4	Incorreto	Correto	Correto	Correto
Q5	Correto	Correto	Incorreto	Incorreto
Q6	Correto	Correto	Incorreto	Incorreto
Q7	Correto	Correto	Correto	Correto
Q8	Correto	Correto	Correto	Correto
Q9	Correto	Correto	Correto	Correto

	Correto
	Incorreto

Fonte: O Autor, 2020

Todas as respostas incorretas foram explicadas novamente durante a execução deste Kahoot para que todos os estudantes realizassem a atividade prática compreendendo todos os assuntos. Após a conclusão deste Kahoot os estudantes realizaram a atividade prática sobre o assunto montando o circuito contendo botões, potenciômetro e fone de ouvido, assim construindo seu primeiro DMI.

5.5.4.4 Observações realizadas durante a aula 04

Nesta aula os estudantes utilizaram os primeiros 30 min da aula para montar os circuitos da aula anterior. Neste momento surgiram algumas dúvidas em relação a montagem. O Estudante E3 conectou de forma incorreta o potenciômetro, o que impediu o controle do volume, além do mal contato com o Jack J2. Já o estudante E2 não conectou as trilhas do GND em conjunto. Para corrigir esses problemas foi explicado novamente o funcionamento correto dos componentes.

Na atividade desta aula os estudantes construíram seus primeiros DMIs. Todos os estudantes conseguiram realizar as atividades práticas com a protoboard sem muitas dificuldades. Isso pode ser observado através dos circuitos desenvolvidos pelos estudantes nesta aula e que são apresentados nas Figuras 73, 74, 75 e 76. Pode-se verificar que os estudantes seguiram os padrões de cores de fios estabelecidos nas aulas e por conta própria organizaram os circuitos da maneira que gostariam.

Ao adicionar os botões ocorreram alguns erros como conectar os componentes em uma trilha da protoboard que não estava ligada, porém esse tipo de erro foi resolvido fazendo uma revisão do circuito montado. Outras vezes o projeto não funcionava, mas as conexões estavam corretas, então foi constatado que os conectores jack J2 não fixavam muito bem na protoboard e se soltavam quando era colocado os fones de ouvido. Isso gerou uma rotina por parte dos estudantes de sempre verificar se o conector não estava com um mal contato.

Nesta aula os estudantes E1, E2 e E3 já haviam comprado uma placa Arduino cada. É possível perceber na Figura 75 que o estudante E3 utilizou sua placa Arduino nesta aula. Isso demonstra que os estudantes estavam motivados para aprender os conceitos do curso. Apesar de terem comprado suas placas Arduino, os estudantes E1 e E2 preferiram não utiliza-las durante essas aulas.

5.5.4.5 Considerações sobre a aula

As fotos das Figuras 73, 74, 75, 76 foram tiradas pelos próprios estudantes com o intuito de utilizarem nas aulas seguintes para montar os circuitos novamente. O que contribuiu para o curso, pois os estudantes criaram maior familiaridade com o circuito. Enquanto na apresentação da aula foi mostrado apenas o circuito básico, como é mostrado

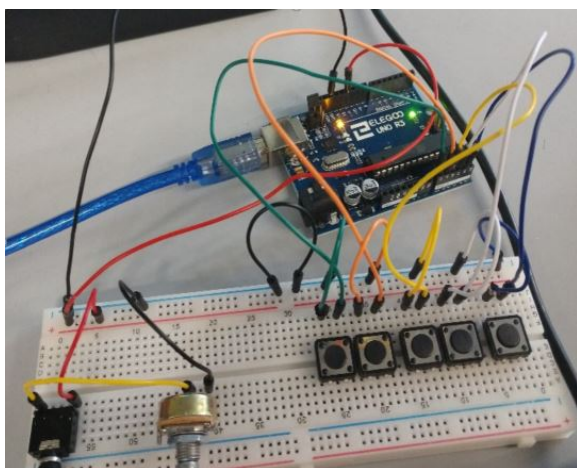


Figura 73 – Circuito 1 - DMI 1 – Estudante E1

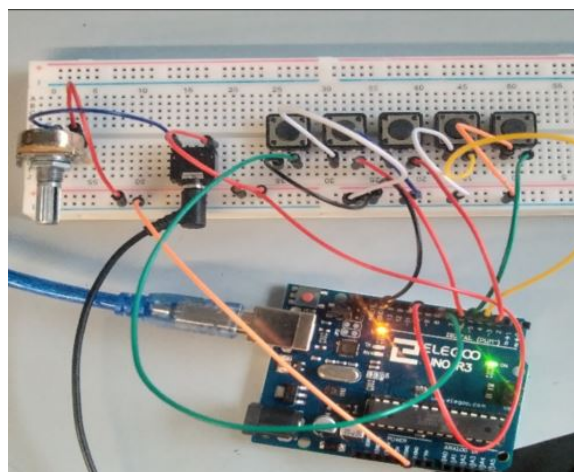


Figura 74 – Circuito 1 - DMI 1 – Estudante E2

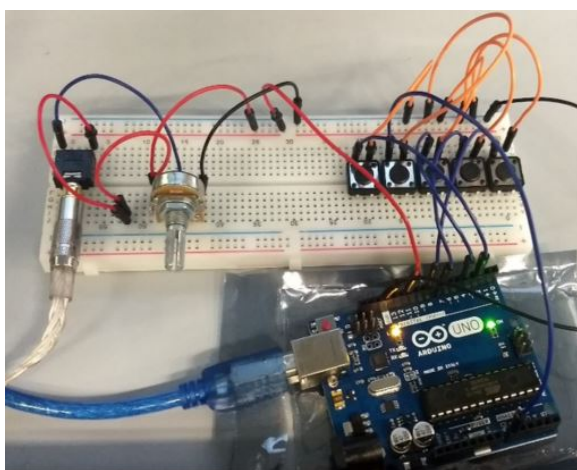


Figura 75 – Circuito 1 - DMI 1 – Estudante E3

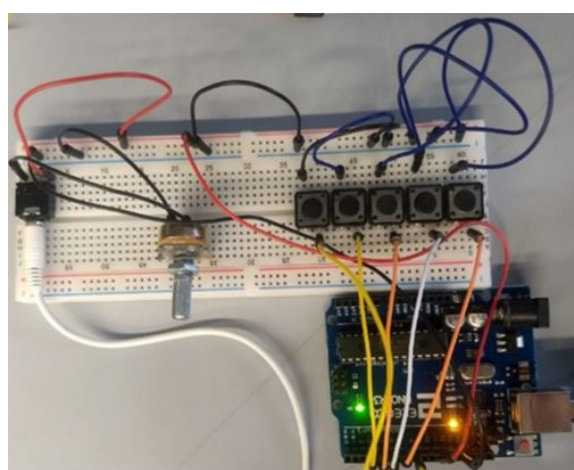


Figura 76 – Circuito 1 - DMI 1 – Estudante E4

na Figura 77. Nas aulas seguintes os estudantes montaram os circuitos mais rapidamente utilizando as fotos dos circuitos que tiraram.

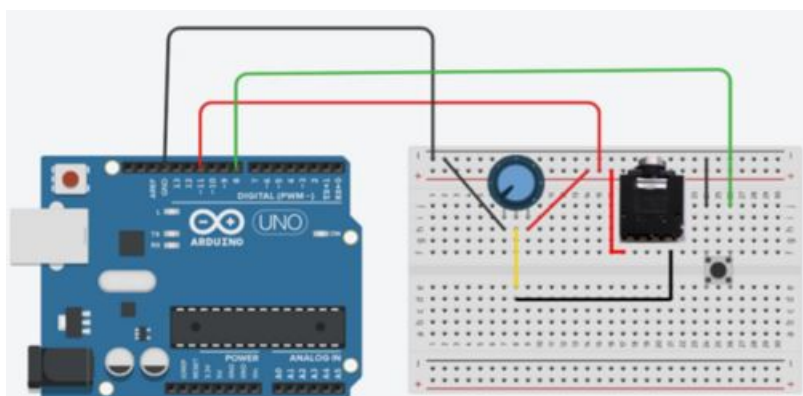


Figura 77 – Circuito primeiro DMI aula 04

5.5.5 Aula 05 - Comandos de Decisão

5.5.5.1 Resultados de Aprendizagem da Aula 05

- R.A.5.1: Utilizar comandos IF e ELSE para identificar os sinais dos botões;
- R.A.5.2: Conectar circuitos em 2 protoboards;
- R.A.5.3: Utilizar LEDs em circuitos;
- R.A.5.4: Utilizar comandos digitalWrite para controlar LEDs;
- R.A.5.5: Utilizar operador booleano && com comandos IF.

5.5.5.2 Estrutura da Aula 05

Tabela 33 – Estrutura da Aula 05

Momento	Tópicos	Estratégia Instrucional	Recursos
1º	Comando de decisão IF	Apresentação, imagens do botão, conexão na protoboard e código necessário.	Slides, Placa Arduino, Jack J2, Protoboard, Resistores, Fios, fones de ouvido, potenciômetro, Computador com IDE Arduino, 5 botões.
2º	Teste prático dos botões	Atividade prática, verificar se o comando IF melhorou a precisão	Slides, Placa Arduino, Jack J2, Protoboard, Resistores, Fios, fones de ouvido, potenciômetro, Computador com IDE Arduino, 5 botões

3º	Acrescentar 1 botão em outra protoboard	Apresentação, imagens do circuito	Slides, Placa Arduino, Jack J2, 2 Protoboards, Resistores, Fios, fones de ouvido, potenciômetro, Computador com IDE Arduino, 6 botões.
4º	Controlar LED	Apresentação, imagens do LED e circuito	Slides, Placa Arduino, Jack J2, 2 Protoboards, Resistores, Fios, fones de ouvido, potenciômetro, Computador com IDE Arduino, 6 botões, e LED.
5º	Código do LED e comando ELSE.	Apresentação, imagens do código para o LED.	Slides, Placa Arduino, Jack J2, 2 Protoboards, Resistores, Fios, fones de ouvido, potenciômetro, Computador com IDE Arduino, 6 botões, e LED

6º	Aferição do entendimento sobre os comandos de decisão IF e ELSE.	9 Perguntas com 4 alternativas cada.	Kahoot
7º	Melhorando a precisão do primeiro DMI e controle de LED.	Atividade prática para alterar o código anterior utilizando comandos de decisão, montar mais 1 botão e 5 LEDs.	Slides, Placa Arduino, Jack J2, 2 Protoboards, Resistores, Fios, fones de ouvido, potenciômetro, Computador com IDE Arduino, 6 botões, e 5 LEDs.

Fonte: O Autor, 2020

5.5.5.3 Resultados do Kahoot 08

5.5.5.3.1 Kahoot #8 – Comandos de decisão

Objetivo: Verificar se os estudantes compreenderam como utilizar os comandos IF e ELSE e como funciona o componente LED.

Conteúdo do Kahoot 8: Este Kahoot foi formado por nove questões envolvendo os comandos “IF” e “ELSE” utilizados para melhorar a precisão dos botões utilizados anteriormente, e como utilizar o LED no Arduino para responder a comandos enviados pelos botões. Na Tabela 34 é possível verificar que as questões Q8.1, Q8.2, Q8.3 e Q8.7 são relacionadas ao funcionamento do comando “IF” em como utiliza-lo no Arduino em conjunto da questão Q8 sobre o comando “ELSE”. Já as questões Q8.4, Q8.5 e Q8.6 fazem referência a montagem do circuito com o LED, sendo que para a questão Q8.4 foi apresentado uma imagem de um LED para auxiliar na resposta, em conjunto da questão Q8.9 direcionada a programação para enviar um comando ao LED.

Tabela 34 – Perguntas do Kahoot 08

Nº	Questão	Alternativas	
Q8.1	Para que é utilizada a função IF?	A) Para executar o Buzzer no tempo certo	B) Para verificar se o Buzzer está tocando
		C) Para verificar quando a nota é tocada	D) Para verificar quando uma condição é satisfeita
Q8.2	Porque tem o número 1 na função IF?	A) Para executar apenas 1 vez	B) Para verificar se o valor de botão é 0
		C) Para verificar se o valor do botão é 1	D) Para que o Buzzer ligue
Q8.3	Porque são utilizados 2 símbolos de igual na função IF?	A) Para verificar se é igual.	B) Para verificar se é diferente..
		C) Para atribuir o valor 1 ao botão	D) Para verificar a nota1.
Q8.4	Qual terminal do LED é o POSITIVO?	A) O LED não possui positivo e negativo.	B) A parte de cima do LED é o positivo.
		C) O terminal mais comprido é o positivo	D) O terminal mais curto é o positivo.
Q8.5	Como identifica quais são os terminais do LED se os terminais forem cortados?	A) O chanfro do LED indica o terminal negativo.	B) O chanfro do LED indica o lado positivo.
		C) Cortando um dos terminais para ficar maior)	D) Não tem como identificar.
Q8.6	Qual terminal do Led deve-se colocar o resistor?	A) Terminal negativo.	B) Terminal positivo.
		C) Qualquer um dos terminais.	D) Nenhum dos terminais.
Q8.7	Para que serve os símbolos && na função IF?	A) Para inverter a lógica.	B) Para verificar mais de uma condição
		C) Para multiplicar o digitalRead	D) Para apertar 2 botões.

Q8.8	Para que é utilizada a função ELSE?	A) Para verificar se o botão foi solto	B) Para verificar se o botão foi pressionado
		C) Executa um comando se a condição do IF	D) Envia um comando para uma porta do Arduino
Q8.9	O que faz o símbolo * na função?	A) Liga apenas LEDs	B) Envia um comando para uma porta do Arduino
		C) Desliga os LEDs	D) Verifica se um LED está acesso.

Fonte: O Autor, 2020

Resultados: É possível verificar na Tabela 35 que todos os estudantes responderam corretamente 5 questões (Q8.3, Q8.4, Q8.5, Q8.7 e Q8.8). É observado que apenas 1 estudante (E4) respondeu incorretamente à questão Q8.1, enquanto 2 estudantes (E3 e E4) responderam incorretamente à questão Q8.9. Já a questão Q2 apenas 1 estudante (E2) respondeu corretamente. Na questão Q8.6 todos os estudantes responderam incorretamente. Esta questão procurou demonstrar que não existe uma regra para utilizar o resistor no LED, sendo assim possível utilizar o resistor em qualquer terminal do LED.

Tabela 35 – Resultados do Kahoot 08

Kahoot 8				
	E1	E2	E3	E4
Q1	Correto	Correto	Correto	Incorreto
Q2	Incorreto	Correto	Incorreto	Incorreto
Q3	Correto	Correto	Correto	Correto
Q4	Correto	Correto	Correto	Correto
Q5	Correto	Correto	Correto	Correto
Q6	Incorreto	Incorreto	Incorreto	Incorreto
Q7	Correto	Correto	Correto	Correto
Q8	Correto	Correto	Correto	Correto
Q9	Correto	Correto	Incorreto	Incorreto

	Correto
	Incorreto

Fonte: O Autor, 2020

Todas as respostas incorretas foram explicadas novamente durante a execução deste Kahoot para que todos os estudantes realizassem a atividade prática compreendendo todos os assuntos. Após a conclusão deste Kahoot os estudantes realizaram a atividade prática sobre o assunto, acrescentando ao circuito anterior LEDs e programando os comandos de decisão.

5.5.5.4 Observações realizadas durante a aula 05

Na atividade prática foram observadas algumas dificuldades em relação aos símbolos de abertura e fechamento de chaves (`{ }`) dos comandos “IF”. Os estudantes E3 e E4 que não possuem conhecimentos prévios em programação, esqueciam muitas vezes de fechar as chaves de um bloco de comando, o que fazia os botões ficarem executando as notas de forma ininterruptas. Então foi necessário explicar novamente sobre o funcionamento das chaves dentro do programa.

No segundo momento da aula os estudantes testaram o código contendo os comandos IF. Todos os estudantes conseguiram realizar este teste de forma correta, tocando algumas músicas que conheciam e que eram possíveis de serem tocadas utilizando apenas cinco notas.

Na segunda atividade prática desta aula os estudantes acrescentaram ao seu circuito 5 LEDs com 1 resistor cada. Esta atividade foi considerada a mais difícil de montar e programar, por ter muitos componentes e muita programação.

O estudante E2 ao perceber que o circuito estava muito grande questionou se poderia ser utilizado as portas analógicas como saídas para conectar os LED. Então foi explicado que poderiam ser utilizados como qualquer outra porta do Arduino, logo todos os estudantes começaram a utilizar estas portas para algum componente, o que não ocorria antes devido à falta de explicação sobre tal componente. No início da atividade dos LEDs não foi programado para que apagassem de forma proposital, porém o estudante E1 que possuía conhecimentos prévios em programação conseguiu fazer os LEDs apagarem antes mesmo da explicação.

5.5.5.5 Considerações sobre a aula

Nesta aula os estudantes continuaram com o circuito anterior com 5 botões, controle do volume e o fone de ouvido, porém o código foi modificado para melhorar a precisão dos botões. Da maneira em que havia sido programado era difícil tocar uma nota após a outra, pois haviam pausas no programa que faziam "enroscar" o som. Para corrigir este problema foi necessário utilizar o comando de decisão IF, podendo assim tocar uma música de forma confortável.

Com a intenção de adicionar sensores de saída, foram utilizados LEDs para fun-

cionarem em conjunto dos botões. Com isso foi necessário introduzir os conceitos do uso do operador booleano. O funcionamento desse circuito deveria seguir a regra que cada botão acenderia um LED, porém o LED só será aceso quando forem pressionados simultaneamente o botão de uma nota e o botão do LED. Por este motivo o operador booleano foi necessário nesta etapa. Os circuitos montados pelos estudantes podem ser vistos nas Figuras 78, 79, 80 e 81.

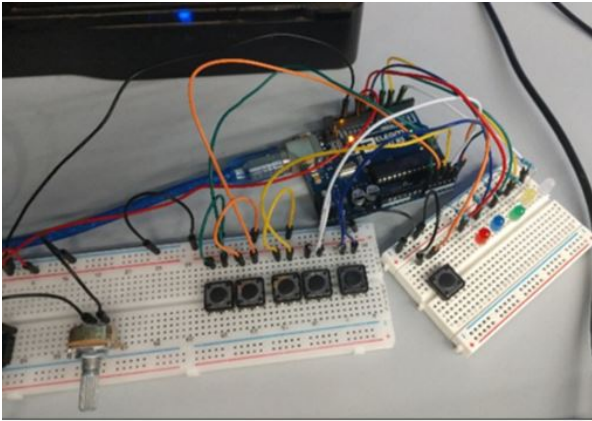


Figura 78 – Circuito 2 - Botões e LEDs – Estudante E1

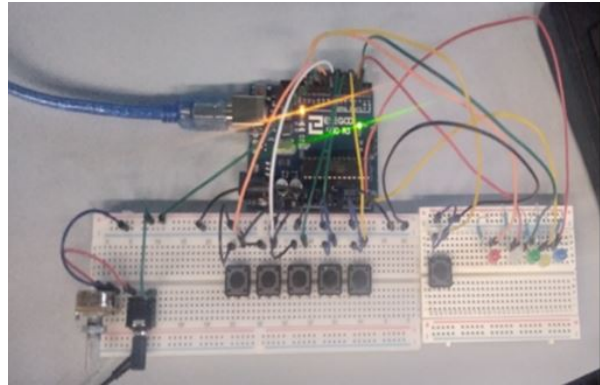


Figura 79 – Circuito 2 - Botões e LEDs – Estudante E2

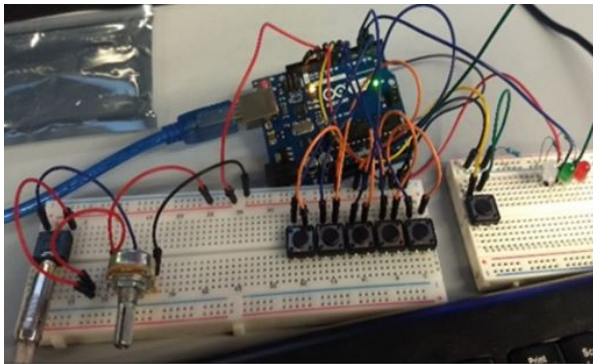


Figura 80 – Circuito 2 - Botões e LEDs – Estudante E3

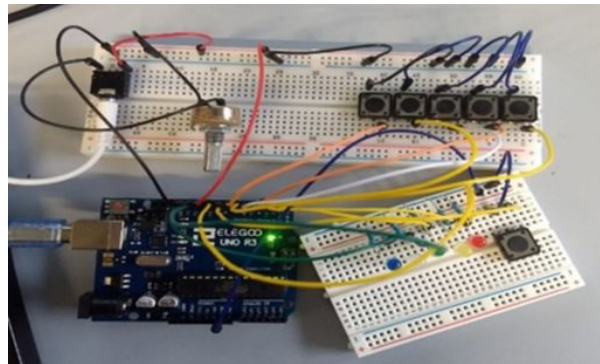


Figura 81 – Circuito 2 - Botões e LEDs – Estudante E4

5.5.6 Aula 06 - Protocolo MIDI

5.5.6.1 Resultados de Aprendizagem da Aula 06.

- R.A.6.1: Configurar o computador para executar comandos MIDI;
- R.A.6.2: Enviar comandos MIDI do Arduino ao computador.

5.5.6.2 Estrutura da Aula 06.

Tabela 36 – Estrutura da Aula 06

Momento	Tópicos	Estratégia Instrucional	Recursos
1º	Protocolo MIDI	Apresentação, imagens dos códigos necessários.	Slides, Placa Arduino, 2 Protoboards, Resistores, Fios, fones de ouvido, Computador com IDE Arduino, 6 botões, 5 LEDs, Software FreePiano.
2º	Aferição do entendimento sobre protocolo MIDI	11 Perguntas sendo 10 com 4 alternativas cada e 1 com verdadeiro ou falso.	Kahoot
3º	Melhorando a qualidade sonora	Atividade prática para programar o Arduino com comandos do protocolo MIDI	Slides, Placa Arduino, 2 Protoboards, Resistores, Fios, fones de ouvido, Computador com IDE Arduino, 6 botões, 5 LEDs, Software FreePiano.

5.5.6.3 Resultados do Kahoot 09

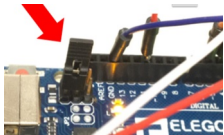
5.5.6.3.1 Kahoot #09 – Protocolo MIDI

Objetivo: : Verificar se os estudantes compreenderam como utilizar o protocolo MIDI com o Arduino.

Conteúdo Kahoot 9: Neste Kahoot os estudantes responderam onze questões referentes ao protocolo MIDI. Na Tabela 37 é demonstrado que as questões Q9.1, Q9.2, Q9.3 e Q9.5 são questões que envolvem o funcionamento do protocolo MIDI, já as questões Q9.4, Q9.6, Q9.9 e Q9.11 são relacionadas aos comandos que o Arduino envia para o funcionamento do protocolo MIDI. As questões Q9.7, Q9.8 e Q9.10 são relacionadas as configurações iniciais que devem ser feitas no Arduino.

Tabela 37 – Perguntas do Kahoot 09

Nº	Questão	Alternativas	
Q9.1	No protocolo MIDI são transmitidos...	A) Somente dados	B) Somente som
		C) Som e Dados	D) Somente sinais
Q9.2	Qual significado de MIDI?	A) Instrumentos musicais digitais	B) Instrumentos de interface Digital musical
		C) Interface Digital para instrumentos musicais.	D) Interface Musical para instrumentos digitais
Q9.3	Qual canal MIDI é utilizado apenas para sons de bateria?	A) Canal 1	B) Canal 10
		C) Canal 11	D) Canal 9

Q9.4	Qual comando deve ser enviado para ligar a nota no canal 1?	A) 48	B) 153
		C) 100	D) 144
Q9.5	Qual ordem dos comandos MIDI para tocar uma nota?	A) NOTA, LIGAR e CANAL, DURAÇÃO	B) CANAL, NOTA, DURAÇÃO
		C) NOTA, DURAÇÃO, CANAL	D) LIGAR e CANAL, NOTA, DURAÇÃO
Q9.6	Quais os comandos para tocar a nota C3 no canal 2?	A) 144,48,100	B) 153,48,100
		C) 145,48,100	D) 143,48,100
Q9.7	<p>Para que o Arduino seja reconhecido como interface MIDI o jumper deve estar conectado?</p> 	Verdadeiro	Falso
Q9.8	Quais os pinos devem ser colocado o jumper para programar o Arduino?	A) 1 e 2	B) 4 e 5
		C) 1 e 4	D) 4 e 1

Q9.9	Qual comando é utilizado para definir a velocidade da comunicação MIDI?	A) Serial.begin(31250)	B) Serial.begin(31150)
		C) Serial.println(31250)	D) Begin.serial(31250)
Q9.10	Qual a velocidade da comunicação MIDI?	A) 31000 bps	B) 31200 bps
		C) 31250 bps	D) 31050 bps

Fonte: O Autor, 2020

As questões Q9.7 e Q9.8 utilizaram imagens da placa Arduino mostrando o conector jumper na sua posição correta e os números dos terminais em que deveria ser conectado. Por este motivo na questão Q9.7 obteve respostas de verdadeiro e falso.

Resultados: Como pode ser observado na Tabela 38 todos os estudantes responderam corretamente as questões Q.92, Q9.8, Q9.9, Q9.10 e Q9.11. As questões Q9.4 e Q9.7 obtiveram apenas 1 resposta incorreta dos estudantes (E3 e E1). As questões Q9.1, Q9.3, Q9.5 e Q9.6 obtiveram 2 respostas incorretas cada.

Tabela 38 – Resultados do Kahoot 09

Kahoot 9				
	E1	E2	E3	E4
Q1	Correto	Correto	Incorreto	Incorreto
Q2	Correto	Correto	Correto	Correto
Q3	Correto	Incorreto	Incorreto	Correto
Q4	Correto	Correto	Incorreto	Correto
Q5	Incorreto	Incorreto	Correto	Correto
Q6	Correto	Incorreto	Correto	Incorreto
Q7	Incorreto	Correto	Correto	Correto
Q8	Correto	Correto	Correto	Correto
Q9	Correto	Correto	Correto	Correto
Q10	Correto	Correto	Correto	Correto
Q11	Correto	Correto	Correto	Correto

Correto	Correto
Incorreto	Incorreto

Fonte: O Autor, 2020

Após a conclusão deste Kahoot os estudantes executaram a atividade prática configurando o circuito da placa Arduino e programando o código com o protocolo MIDI.

5.5.6.4 Observações realizadas durante a aula 06

Nesta aula todos os estudantes conseguiram enviar comandos MIDI ao computador. Com isso conseguiram construir um segundo DMI que possibilitou gerar diversos timbres diferentes devido ao protocolo MIDI possibilitar escolher qual instrumento musical será tocado.

Uma reclamação de todos os estudantes foi que para gravar o código e poder testa-lo era necessário criar uma rotina de retirar o jumper e o conector USB diversas vezes, o que se tornava cansativo.

É possível verificar nas Figuras 82, 83 e 84 que os estudantes melhoraram a organização dos circuitos. Não foi possível obter a foto do circuito do estudante E4 desta aula. A partir desta aula os estudantes estavam mais dispersos, pois se iniciou o período de provas no curso de Licenciatura em Música. Isso contribuiu para que faltasse imagens deste circuito.

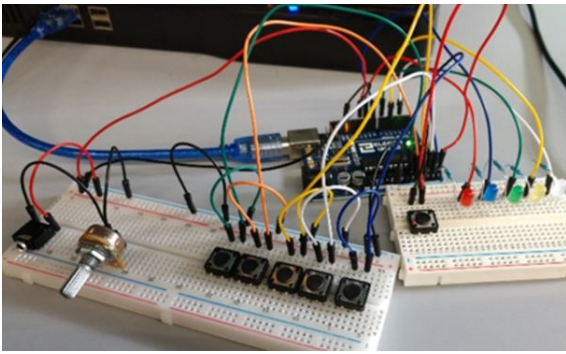


Figura 82 – Circuito 3 - Botões e LEDs – Estudante E1

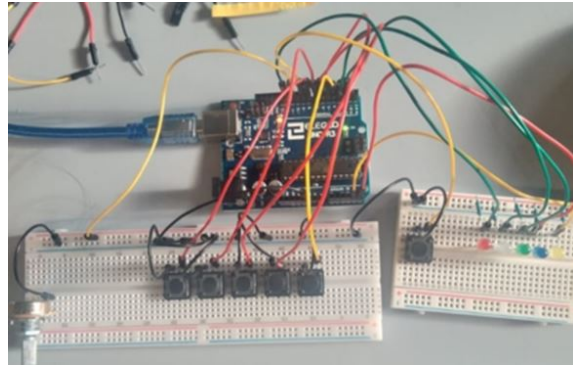


Figura 83 – Circuito 3 - Botões e LEDs – Estudante E2

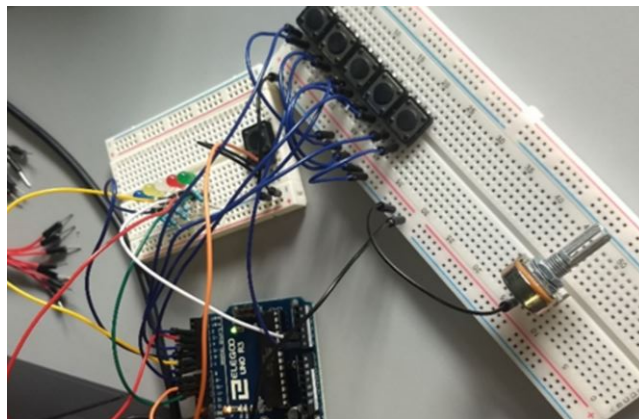


Figura 84 – Circuito 3 - Botões e LEDs – Estudante E3

5.5.6.5 Considerações sobre a aula

Nesta aula foram introduzidos os conceitos sobre o protocolo MIDI. Para que fosse possível utilizar os comandos do protocolo MIDI com o Arduino e o computador, foi necessário modificar o firmware do microcontrolador Atmega16 μ 2, como já explicado na fundamentação teórica deste trabalho. O funcionamento desse firmware necessita de um conector (jumper) na placa Arduino para identificar quando será feita uma gravação de um código ou quando será utilizado como dispositivo MIDI. Para isso foi necessário explicar o funcionamento do jumper. Isso gerou um conflito em relação às placas Arduino adquiridas pelos estudantes E1, E2 e E3, uma vez que é necessário alterar o firmware do Arduino. Para corrigir isso o autor modificou os firmwares das placas dos estudantes.

Outra adaptação necessária foi utilizar o software Freepiano para reproduzir os comandos MIDI, apresentado na seção 4.4.2 Prototipação Utilizando Computador. É importante ressaltar que esse procedimento precisou ser realizado para que funcionasse os comandos relacionados ao canal 10 que contém os sons de bateria. Esta aula deu início a outro módulo do curso, o qual teve maior foco no protocolo MIDI e as formas de enviar os comandos necessários para o computador, mantendo o circuito sem modificações.

5.5.7 Aula 07 - Variáveis

5.5.7.1 Resultados de Aprendizagem da Aula 07.

- R.A.7.1: Utilizar variáveis com Arduino;
- R.A.7.2: Ler e enviar sinais de vários botões ao mesmo tempo;
- R.A.7.3: Utilizar funções com Arduino.

5.5.7.2 Estrutura da Aula 07.

Tabela 39 – Estrutura da Aula 07

Momento	Tópicos	Estratégia Instrucional	Recursos
1º	Variáveis	Apresentação, imagens dos códigos necessários.	Slides, Placa Arduino, Protoboard, Fios, fones de ouvido, Computador com IDE Arduino, 5 botões, Software FreePiano.
2º	Aferição do entendimento sobre variáveis	8 Perguntas com 4 alternativas cada.	Kahoot
3º	Tocando múltiplas teclas ao mesmo tempo.	Atividade prática para programar o Arduino com variáveis.	Slides, Placa Arduino, Protoboard, Resistores, Fios, fones de ouvido, Computador com IDE Arduino, 5 botões, Software FreePiano.

4º	Funções	Apresentação, imagens dos códigos necessários	Slides, Placa Arduino, Protoboard, Resistores, Fios, fones de ouvido, Computador com IDE Arduino, 5 botões, Software FreePiano.
5º	Aferição do entendimento sobre funções.	5 Perguntas com 4 alternativas cada.	Kahoot
6º	Melhorando o código	Atividade prática para programar as funções para enviar comandos MIDI	Slides, Placa Arduino, Protoboard, Resistores, Fios, fones de ouvido, Computador com IDE Arduino, 5 botões, Software FreePiano.

Fonte: O Autor, 2020

5.5.7.3 Resultados dos Kahoots 10 e 11

5.5.7.3.1 Kahoot #10 – Variáveis

Objetivo: verificar se os estudantes compreenderam como utilizar variáveis no Arduino.

Conteúdo do Kahoot 10: Este Kahoot foi composto por oito questões referentes ao assunto de variáveis. Como pode ser verificado na Tabela 40 que as questões Q10.1, Q10.2, Q10.3, Q10.4, e Q10.8 são direcionadas ao funcionamento de uma variável. Enquanto as questões Q10.5, Q10.6 e Q10.7 são referentes ao código de exemplo.

Tabela 40 – Perguntas do Kahoot 10

Nº	Questão	Alternativas	
Q10.1	Para declarar uma variável é necessário...	A)Nome, Tipo, Valor	B)Tipo, Valor, Nome

		C) Tipo, Nome, Valor	D) Valor, Nome, Tipo
Q10.2	O que é a palavra int na declaração de uma variável?	A) Nome	B) Tipo
		C) Valor	D) memória
Q10.3	A palavra int significa...	A) Intermédio	B) Tipo
		C) Intervalo	D) Inteiro
Q10.4	Porque a variável é iniciada antes da função void loop?	A) Porque a variável dentro da função loop iria sempre ter o valor 0	B) Porque variáveis devem ser declaradas somente antes da função loop
		C) Está errado, deve ser declarado dentro da função loop	D) Porque a variável inicia com 0
Q10.5	Porque deve-se mudar o estadoAnt para 1 depois que toca a nota?	A) Para tocar as notas várias vezes	B) Porque é necessário desligar a nota.
		C) Porque é necessário guardar o valor anterior	D) Porque é necessário atualizar o valor do estado anterior.
Q10.6	Porque a função else está depois do primeiro IF e não do segundo IF?	A) Porque é quando o botão está solto	B) Porque o segundo IF somente liga a nota.
		C) Porque o segundo IF está dentro de outro IF e não pode usar ELSE	D) Porque o estado anterior está dentro do primeiro IF
Q10.7	Porque no primeiro IF é verificado se o valor é igual a 1 e no segundo IF é verificado se é igual a 0?	A) Está errado desta forma, deve ser invertida a lógica.	B) O 1º IF verifica o estado anterior do botão, o 2º verifica o estado atual
		C) O 1º IF verifica o estado atual do botão, o 2º verifica o estado anterior	D) Porque no 1º IF é verificado o botão e no 2º é ligado a nota.
Q10.8	O que é uma variável do tipo boolean?	A) Variável que possui vários valores diferentes.	B) Variável que possui valores verdadeiros ou falso.
		C) Variável que possui valores 1 ou 0	D) Variável que possui valores de estado.

Fonte, O Autor, 2020

Resultados: Na Tabela 41 é possível verificar que todos os estudantes responderam corretamente as questões Q10.1, Q10.2 e Q10.8. Enquanto apenas 1 estudante (E4) respondeu incorretamente as questões Q10.3 e Q10.5, e também a questão Q10.7 por E3. A questão Q10.4 foi respondida incorretamente por 2 estudantes (E2 e E3), assim como a questão Q10.6 pelos estudantes E1 e E4.

Tabela 41 – Resultados do Kahoot 10

Kahoot 10				
	E1	E2	E3	E4
Q1	Correto	Correto	Correto	Correto
Q2	Correto	Correto	Correto	Correto
Q3	Correto	Correto	Correto	Incorreto
Q4	Correto	Incorreto	Incorreto	Correto
Q5	Correto	Correto	Correto	Incorreto
Q6	Incorreto	Correto	Correto	Incorreto
Q7	Correto	Correto	Incorreto	Correto
Q8	Correto	Correto	Correto	Correto

	Correto
✗	Incorreto

Fonte: O Autor, 2020

Ao final deste Kahoot os estudantes realizaram a atividade prática de acrescentar ao código as variáveis. Então foi dada continuidade a explicação teórica sobre funções.

5.5.7.3.2 Kahoot #11 – Funções

Objetivo: Verificar se os estudantes compreenderam como utilizar as funções no Arduino.

Conteúdo do Kahoot 11: Este Kahoot foi composto por cinco questões sobre como utilizar funções no Arduino. Na Tabela 42 são demonstradas todas as questões envolvendo as configurações de uma função (Q11.2, Q11.3 e Q11.5) e como elas funcionam (Q11.1 e Q11.4).

Tabela 42 – Perguntas do Kahoot 11

Nº	Questão	Alternativas	
Q11.1	O que é uma função no Arduino?	A) Um conjunto de comandos que realiza uma tarefa	B) Um conjunto de dados que repetem infinitamente
		C) Um conjunto de comandos que configura o MIDI	D) Um conjunto de comandos que enviam dados MIDI
Q11.2	Onde devem ser declaradas as funções?	A) Dentro da função Setup	B) Depois da função Loop
		C) Entre a função Setup e Loop	D) Dentro da função Loop
Q11.3	Qual é a configuração de uma função?	A) TIPO, PARAMETROS, NOME	B) NOME, PARAMETRO, TIPO
		C) TIPO, NOME, PARAMETROS	D) PARAMETRO, TIPO, NOME
Q11.4	Funções com nomes compostos devem ser...	A) Primeira palavra inicia com MAIÚSCULA e a segunda com minúscula	B) Iniciar com todas Maiúsculas e terminar com todas Minúsculas
		C) Todas letras maiúsculas	D) Primeira palavra inicia com minúscula e a segunda com maiúscula
Q11.5	O que são os parâmetros de uma função?	A) São valores que a função precisa.	B) São variáveis que todo código pode acessar
		C) São valores que a função recebe para serem utilizados na lógica	D) São valores que recebem apenas comandos MIDI.

Fonte: O Autor, 2020

Todas as questões foram compostas com uma imagem da função utilizada como exemplo para enviar comandos MIDI demonstradas na explicação teórica.

Resultados: Era esperado que os estudantes compreendessem o funcionamento das funções no Arduino respondendo corretamente ao menos as questões referentes a configuração das funções (Q11.2, Q11.3 e Q11.5). É possível verificar na Tabela 43 que

todos os estudantes responderam corretamente as questões Q11.1 e Q11.5. Além disso os estudantes E1 e E3 responderam corretamente todas as questões. A questão Q2 obteve apenas 1 resposta incorreta por E4 e assim como a questão Q11.3 por E2. Na questão Q11.5 dois estudantes (E2 e E4) responderam de forma incorreta.

Tabela 43 – Resultados do Kahoot 11

Kahoot 11				
	E1	E2	E3	E4
Q1	Correto	Correto	Correto	Correto
Q2	Correto	Correto	Correto	Incorreto
Q3	Correto	Incorreto	Correto	Correto
Q4	Correto	Correto	Correto	Correto
Q5	Correto	Incorreto	Correto	Incorreto

Correto	Correto
Incorreto	Incorreto

Fonte: O Autor, 2020

Ao final deste Kahoot os estudantes realizaram a atividade prática de programar a função no código do circuito 3 - Botões e LEDs.

5.5.7.4 Observações realizadas durante a aula 07

Nesta aula não houve problemas no circuito, principalmente por não ter necessidade de modificar o circuito antigo. Em relação ao código o estudante E4 demonstrou dificuldades em relação ao uso das chaves ({ }) para o fechamento de blocos de código. Isso já havia sido explicado anteriormente, porém foi necessário reforçar mais uma vez como utilizar os símbolos de chaves. Além disso criou apenas variáveis com o mesmo nome para o estado anterior. O que foi corrigido explicando como cada nome deve ser único e como reconhecer a mensagem desse erro na IDE do Arduino.

O estudante E3 utilizou seu próprio Arduino e seu notebook para realizar as atividades a partir desta aula. Assim, utilizou softwares MIDI de seu gosto.

5.5.7.5 Considerações sobre a aula

Nesta aula os estudantes aprenderam como utilizar variáveis para armazenar o estado do botão, além disso foi apresentado como utilizar uma função para enviar os comandos MIDI com o objetivo de diminuir a repetição das linhas de comandos MIDI organizando melhor o código. Com isso os estudantes concluíram o DMI chamado de Xilofone de Botão MIDI, o qual reproduzia sons de piano e xilofone conforme era escolhido

no software MIDI. Este foi o DMI que os estudantes mais gostaram, como será apresentado na aula 10 na subseção 5.5.10.2 na Tabela 57. Foi composto dos 5 botões que enviavam comandos MIDI em tempo real para o computador e também foi possível utilizar múltiplos botões ao mesmo tempo.

5.5.8 Aula 08 - Detectando Pressão

5.5.8.1 Resultados de Aprendizagem da Aula 08.

- R.A.8.1: Utilizar sensor piezoelétrico na protoboard;
- R.A.8.2: Utilizar o comando analogRead para ler sinais dos piezoelétricos;
- R.A.8.3: Melhorar o sinal do piezoelétrico com diodo;
- R.A.8.4: Detectar pressão do piezoelétrico;
- R.A.8.5: Mapear os valores lidos de entradas analógicas com função map;
- R.A.8.6: Verificar valores com base no tempo com a função Millis.

5.5.8.2 Estrutura da Aula 08.

Tabela 44 – Estrutura da Aula 08

Momento	Tópicos	Estratégia Instrucional	Recursos
1º	Sensor Piezoelétrico	Apresentação, imagens do circuito piezoelétrico.	Slides, Placa Arduino, Protoboard, Fios, fones de ouvido, Computador com IDE Arduino, Software FreePiano e piezoelétrico.
2º	Programar Piezoelétrico	Apresentação, imagens do código necessário.	Slides, Placa Arduino, Protoboard, Fios, fones de ouvido, Computador com IDE Arduino, Software FreePiano e piezoelétrico.

3º	Construindo DMI Capacitivo e Teremim	Atividade prática para construir o DMI capacitivo e o Teremim..	Slides, Placa Arduino, Protoboard, Fios, fones de ouvido, Computador com IDE Arduino, 5 botões, Software FreePiano, 3 clipes de papel e fita isolante.
3º	Diodo	Apresentação, imagens do circuito com piezoelétrico e diodo.	Slides, Placa Arduino, Protoboard, Fios, fones de ouvido, Computador com IDE Arduino, Software FreePiano, piezoelétrico e diodo.
4º	Aferição do entendimento sobre piezoelétricos	8 Perguntas com 4 alternativas cada.	Kahoot
5º	Detectando pressão	Atividade prática para detectar pressão com piezoelétrico	Slides, Placa Arduino, Protoboard, Fios, fones de ouvido, Computador com IDE Arduino, Software FreePiano, 3 piezoelétrico e diodo.
6º	Comandos Map e Millis	Apresentação, imagens do código necessário	Slides, Placa Arduino, Protoboard, Fios, fones de ouvido, Computador com IDE Arduino, Software FreePiano, 3 piezoelétricos.

7º	Aferição do entendimento sobre map e millis	6 Perguntas com 4 alternativas cada.	Kahoot
8ª	Melhorar precisão	Atividade prática modificando o código com as funções map e millis	Slides, Placa Arduino, Protoboard, Fios, fones de ouvido, Computador com IDE Arduino, 5 botões, Software FreePiano, 3 piezoelétricos.

Fonte: O Autor, 2020

5.5.8.3 Resultados dos Kahoots 12 e 13

5.5.8.3.1 Kahoot #12 – Piezo

Objetivo: : Verificar se os estudantes compreenderam como utilizar o piezo no Arduino e como programa-lo.

Conteúdo do Kahoot 12: Este Kahoot foi composto de oito questões envolvendo as funcionalidades do piezo e como programa-lo no Arduino. Na Tabela 45 são demonstradas todas as questões deste Kahoot sendo que as questões Q12.1 e Q12.2 são referentes as funcionalidades do piezo. A questão Q12.3 é direcionada a placa Arduino. Nas questões Q12.4 e Q12.6 foram direcionadas a forma de comunicação do Arduino com o computador. Já as questões Q12.5 e Q12.8 são relacionadas a programação do piezo no Arduino. Por fim a questão Q12.7 relacionada ao componente Diodo utilizado em conjunto do piezo.

Tabela 45 – Perguntas do Kahoot 12

Nº	Questão	Alternativas	
Q12.1	Os piezoelétricos são possuem...	A) Saída	B)Entrada
		C) Sinal Digital	D) PULLUP
Q12.2	Os piezoelétricos possuem valores...	A) Digitais, ou seja valores entre 0 e 1023	B) Digitais, ou seja valores 0 ou 1.
		C) Analógicos, ou seja valores entre 0 e 1023	D) Analógicos ou seja valores 0 ou 1.

Q12.3	Quais portas do Arduino podem ser utilizadas para ler os valores do piezo?	A) Somente portas analógicas do A0 ao A5	B) Somente portas PWM($\tilde{\quad}$)
		C) Somente portas digitais do 2 ao 13.	D) Qualquer porta pode ser utilizada.
Q12.4	Para que serve a função <code>serial.println</code> ?	A) Para imprimir um valor	B) Para mandar um comando MIDI.
		C) Para imprimir um valor/palavra e pular uma linha	D) Para pular uma linha
Q12.5	O que faz a função <code>analogRead</code> ?	A) Lê um valor na porta analógica	B) Escreve um valor na porta analógica.
		C) Le um valor na porta digital	D) Escreve um valor na porta digital.
Q12.6	O que é o Monitor Serial?	A) É onde é monitorado o Arduino	B) É onde o piezo envia os valores.
		C) É como o Arduino faz gráficos.	D) É a forma de visualizar os valores da porta
Q12.7	O que é o Diodo?	A) É um componente que controla o sentido da corrente elétrica	B) É um componente que amplifica o sinal do piezo
		C) É um componente que emite luz	D) É um componente que transforma energia em calor.
Q12.8	Qual símbolo utilizado para verificar se um valor é maior dentro do IF?	A) Símbolo “M”	B) Símbolo “==”
		C) Símbolo “->”	D) Símbolo “>”

Fonte: O Autor, 2020

Resultados: Na Tabela 46 é verificado que o estudante E1 respondeu corretamente todas as questões, enquanto os estudantes E2 e E3 responderam incorretamente apenas uma questão (Q12.4). No caso do estudante E4 foi obtido um baixo desempenho neste Kahoot respondendo corretamente apenas três questões (Q12.3, Q12.4 e Q12.8).

Tabela 46 – Resultados do Kahoot 12

Kahoot 12				
	E1	E2	E3	E4
Q1	■	■	■	■
Q2	■	■	■	■
Q3	■	■	■	■
Q4	■	■	■	■
Q5	■	■	■	■
Q6	■	■	■	■
Q7	■	■	■	■
Q8	■	■	■	■

■	Correto
■	Incorreto

Fonte: O Autor, 2020

Após este Kahoot os estudantes realizaram a atividade prática montando o circuito contendo o piezo e programando o Arduino com o código explicado. Após a conclusão da atividade foi dada continuidade na explicação teórica sobre a função MAP, utilizada para mapear valores.

5.5.8.3.2 Kahoot #13 – Função Map e Millis

Objetivo: Verificar se os estudantes compreenderam como utilizar as funções MAP e Millis.

Conteúdo do Kahoot 13: Este Kahoot foi composto por seis questões envolvendo as funcionalidades da função map e millis. Na Tabela 47 é possível verificar que as questões Q13.1 e Q13.2 são relacionadas a função map. Já as questões Q13.3, Q13.4, Q13.5 e Q13.6 são ligadas a função millis. Nas questões Q13.5 e Q13.6 foram utilizadas imagens dos códigos para facilitar o entendimento da pergunta.

Tabela 47 – Perguntas do Kahoot 13

Nº	Questão	Alternativas	
Q13.1	O que faz a função map?	A) Altera os valores lidos do piezo	B) Mapeia os valores analógicos entre um valor mínimo e máximo
		C) Mapeia as notas musicais	D) Corrige os valores

Q13.2	Quais são os parâmetros da função map?	A) Valor, MIN, MAX, novoMIN, novoMAX	B) MIN, MAX, novoMAX, novoMIN, valor
		C) Valor, MIN, MAX, novoMAX, novoMIN	D) Valor, novoMIN, novoMAX, MAX, MIN.
Q13.3	Para que serve a variável delayP1?	A) Para armazenar o valor do piezo1	B) Para armazenar o valor do map
		C) Para armazenar o tempo de espera para tocar a próxima nota	D) Para pausar o programa
Q13.4	O que faz a função millis()?	A) Para pausar o programa	B) Conta o tempo desde o momento que o Arduino está ligado.
		C) Conta valores a cada 1000 segundos	D) Atualiza o tempo
Q13.5	Porque deve ser subtraído o valor millis() – delayP1?	A) Para comparar o tempo atual com o tempo anterior	B) Para atribuir o tempo na variável delayP1.
		C) Para zerar o cronometro	D) Para pausar o programa.
Q13.6	Para que serve o teste > 100 na comparação?	A) Para verificar se o piezo foi pressionado em um valor maior que 100	B) Para pausar o programa em 100 milissegundos
		C) Para que o cronometro comece em 100	D) Para verificar quando passou pelo menos 100 milissegundos.

Fonte: O Autor, 2020

Resultados: Era esperado como resultado que os estudantes compreendessem ao menos os conteúdos relacionados as questões Q13.2, Q13.4 e Q13.5, pois são direcionadas a atividade prática. Na Tabela 48 é verificado que os estudantes E2 e E3 responderam corretamente todas as questões enquanto o estudante E1 respondeu incorretamente apenas uma questão (Q13.1). O estudante E4 respondeu corretamente apenas duas questões (Q13.1 e Q13.2)

Tabela 48 – Resultados do Kahoot 13

Kahoot 13				
	E1	E2	E3	E4
Q1	✘			
Q2				
Q3				✘
Q4				✘
Q5				✘
Q6				✘

	Correto
✘	Incorreto

Fonte: O Autor, 2020

Após a conclusão deste Kahoot os estudantes realizaram a atividade prática adicionando ao código as funções map e millis.

5.5.8.4 Observações realizadas durante a aula 08

Os estudantes demonstraram bastante curiosidade com o sensor piezoelétrico, pois foi o primeiro contato deles com este tipo de sensor. Foi observado que os estudantes começaram a interagir de diversas formas com o piezoelétrico além da atividade proposta. O estudante E1 conseguiu visualizar na tela a variação de sua voz colocando o piezoelétrico no pescoço, além disso visualizou a variação de batimentos cardíacos colocando o piezoelétrico no pulso. Os circuitos podem ser vistos nas Figuras 85, 86 e 87.

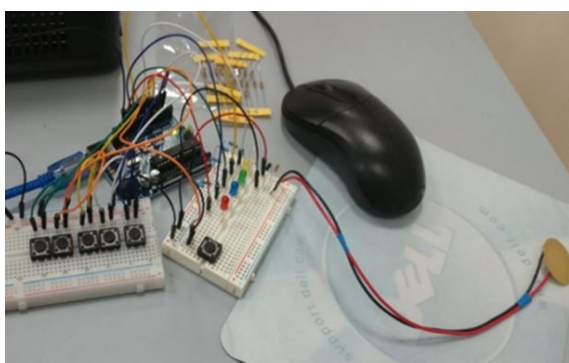


Figura 85 – Circuito 4 - Piezo – Estudante E1

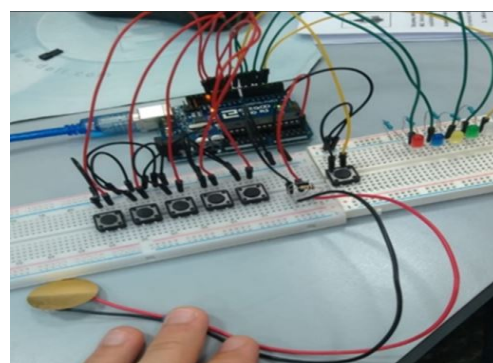


Figura 86 – Circuito 4 - Piezo – Estudante E2

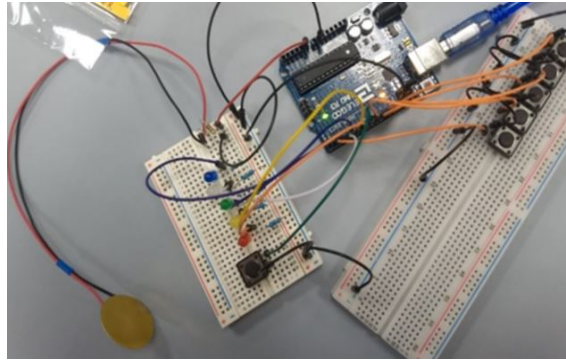


Figura 87 – Circuito 4 - Piezo – Estudante E4

Devido a conflitos de horário nesta aula os estudantes adicionaram apenas um piezoelétrico ao seu circuito como pode ser visto nas Figuras 85, 86 e 87. Por este motivo o estudante E3 não possui imagens de seu circuito, porém realizou a atividade prática conseguindo construir este DMI. Com este sensor os estudantes construíram um DMI mais próximo a um xilofone, podendo receber diferentes batidas para reproduzir sons. Também foi possível utilizar o canal 10 do protocolo MIDI para reproduzir sons de bateria.

5.5.8.5 Considerações sobre a aula

Nesta aula foi apresentado o sensor piezoelétrico que utiliza a pressão para gerar energia. Este sensor sofre interferências do ambiente e possui grande sensibilidade a qualquer movimento, o que gera muito ruído no sinal quando utilizado pelo Arduino. Quando esta aula foi preparada foi utilizado um diodo em conjunto de um resistor e o piezoelétrico com o intuito de reduzir o ruído, o que de fato funcionou. No entanto quando esta aula foi levada para o curso de extensão os estudantes obtiveram um sinal sem ruídos, fazendo com que o diodo não fosse mais necessário. O problema era gerado apenas no notebook e não nos computadores das salas do curso de extensão. Após isso foi pesquisado sobre esse comportamento e foi identificado que o aterramento do computador em que está sendo utilizado o Arduino deve estar corretamente ligado. Como as aulas foram elaboradas em um notebook, foi verificado que o fio de aterramento estava rompido, após consertar este fio, todo o ruído que era gerado desapareceu. Isso implica em remover o diodo destas aulas, apesar de ser um componente importante em aulas de eletrônica, não faz sentido para esta etapa do curso, uma vez que foi utilizado apenas neste circuito.

5.5.9 Aula 09 - Sensor Capacitivo

5.5.9.1 Resultados de Aprendizagem da Aula 09.

- R.A.9.1: Utilizar materiais metálicos como sensor capacitivo;
- R.A.9.2: Instalar uma biblioteca no Arduino e utilizar suas funções;

- R.A.9.3: Programar o Arduino para ler sensores capacitivos.

5.5.9.2 Estrutura da Aula 09.

Tabela 49 – Estrutura da Aula 09

Momento	Tópicos	Estratégia Instrucional	Recursos
1º	Sensor Capacitivo	Apresentação, imagens do circuito piezoelétrico e código necessário.	Slides, Placa Arduino, Protoboard, Fios, fones de ouvido, Computador com IDE Arduino, Software FreePiano, clipes de papel e fita isolante.
2º	Aferição do entendimento sobre sensor capacitivo	12 Perguntas com 4 alternativas cada	Kahoot
3º	Construindo DMI Capacitivo e Teremim	Atividade prática para construir o DMI capacitivo e o Teremim..	Slides, Placa Arduino, Protoboard, Fios, fones de ouvido, Computador com IDE Arduino, 5 botões, Software FreePiano, 3 clipes de papel e fita isolante.

Fonte: O Autor, 2020

5.5.9.3 Resultados do Kahoot 14

5.5.9.3.1 Kahoot #14 – Sensor capacitivo

Objetivo: Verificar se os estudantes compreenderam como utilizar um sensor capacitivo com o Arduino.

Conteúdo do Kahoot 14: Este Kahoot foi composto de doze questões sobre o sensor capacitivo. Na Tabela 50 é possível verificar que as questões Q14.1, Q14.2, Q14.3

Q14.4 e Q14.5 são referentes ao funcionamento do sensor capacitivo. As questões Q14.6, Q14.8 e Q14.9 são referentes a programação necessária para o funcionamento do sensor no Arduino. Já as questões Q14.7, Q14.10 e Q14.12 são direcionadas ao circuito do sensor capacitivo e por fim a questão Q14.11 trata do Teremim que foi o instrumento musical utilizado como inspiração para a atividade prática.

Tabela 50 – Perguntas do Kahoot 14

Nº	Questão	Alternativas	
Q14.1	O que é um sensor capacitivo?	A) Sensor que detecta a mudança de resistência	B) Sensor que detecta a mudança de capacitância
		C) Sensor que utiliza capacitor	D) Sensor que armazena energia
Q14.2	O sensor capacitivo é um...	A) Sensor de saída	B) Sensor de entrada
		C) Sensor de intensidade	D) Sensor de pressão
Q14.3	O que é um capacitor?	A) Componente que armazena energia	B) Sensor que detecta a mudança de capacitância
		C) Sensor de energia elétrica	D) Componente que amplifica energia
Q14.4	Qual é a unidade de medida do capacitor?	A) Volv(V)	B) Ohm(Ω)
		C) Farad (F)	D) Hertz (H)
Q14.5	O que é o Dielétrico?	A) Componente que armazena energia	B) Componente Isolante
		C) Componente de metal	D) Componente condutor
Q14.6	O que são os valores 2 e 3 na função?	A) Envio e Recebimento	B) Recebimento e Envio
		C) Início e Final	D) Capacitância e Resistencia.
Q14.7	Em qual dos terminais deve ser colocado o componente metálico para tocar?	A) Terminal de envio	B) Terminal GND
		C) Qualquer terminal	D) Terminal de Recebimento
Q14.8	Porque é necessário utilizar a biblioteca CapacitiveSensor?	A) Porque possui as funções para o sensor capacitivo	B) Porque é necessário para utilizar capacitores.

		C) Porque melhora o sinal do sensor	D) Porque utiliza campo elétrico.
Q14.9	O que faz a função <code>sensor1.capacitiveSenro(30)</code> ?	A) Envia o sinal para o sensor	B) Limpa o sinal do sensor
		C) Le os valores recebidos pelo sensor	D) Calibra o sinal do sensor.
Q14.10	Qual a finalidade do resistor de 1k no circuito capacitivo?	A) Para reduzir interferências de estática	B) Para zerar o sinal de capacitância
		C) Para aumentar o sinal de capacitância	D) Para aumentar as interferências estáticas.
Q14.11	O que é um Teremim?	A) Instrumento musical capacitivo	B) Instrumento de medição
		C) Instrumento Musical Digital	D) Instrumento Musical Eletrônico.
Q14.12	Para que servem os 3 resistores de 10M no Teremim?	A) Para que servem os 3 resistores de 10M no Teremim	B) Para aumentar a sensibilidade do instrumento
		C) Para evitar ruídos	D) Para variar a capacitância.

Fonte: O Autor, 2020

Resultados: É possível verificar na Tabela 51 que todos os estudantes responderam corretamente as questões Q14.1, Q14.3, Q14.4, Q14.5, Q14.6 e Q14.8. As questões Q14.2 e Q14.10 foram respondidas incorretamente apenas por 1 estudante (E4). A questão Q14.9 foi respondida incorretamente por E1 e E4. Já a questão Q14.12 foi respondida corretamente apenas por 1 estudante (E2). Por fim as questões Q14.7 e Q14.11 foram respondidas incorretamente por todos os estudantes.

Tabela 51 – Resultados do Kahoot 14

Kahoot 14				
	E1	E2	E3	E4
Q1	■	■	■	■
Q2	■	■	■	❌
Q3	■	■	■	■
Q4	■	■	■	■
Q5	■	■	■	■
Q6	■	■	■	■
Q7	❌	❌	❌	❌
Q8	■	■	■	■
Q9	❌	■	■	❌
Q10	■	■	■	❌
Q11	❌	❌	❌	❌
Q12	❌	■	❌	❌

■	Correto
❌	Incorreto

Fonte: O Autor, 2020

Após a finalização deste Kahoot os estudantes realizaram a atividade prática de montar um circuito contendo o sensor capacitivo e programá-lo no Arduino.

5.5.9.4 Observações realizadas durante a aula 09.

Todos os estudantes conseguiram montar os 2 circuitos propostos nesta aula. O primeiro foi um DMI chamado Xilofone capacitivo por utilizar um sensor capacitivo. Uma reclamação dos estudantes foi que este DMI sofria muitas interferências e que isso precisava ser corrigido através do código. Então quando o circuito era montado em outro lugar eles teriam que alterar o código novamente.

5.5.9.5 Considerações sobre a aula

Esta aula foi ministrada em 2 dias diferentes devido a conflitos de horários dos estudantes. Isso contribuiu para que esta aula fosse adaptada a fim de ensinar os principais conceitos para os estudantes e que eles conseguissem realizar a atividade prática.

Para construir os DMIs desta aula foi necessário utilizar a biblioteca Capacitive-Sensor disponível no gerenciador de bibliotecas do Arduino. Os DMIs construídos nesta aula são os mais baratos, pois utilizam apenas resistores e cliques de papel. Este DMI sofre

com ruídos da mesma maneira que o piezoelétrico, por este motivo foi necessário verificar o fio de aterramento do notebook antes de desenvolvê-lo.

5.5.10 Aula 10 - Finalização

Na última aula do curso, foi realizado um Kahoot contendo todo o conteúdo do curso, de forma a averiguar a evolução dos estudantes ao longo do mesmo.

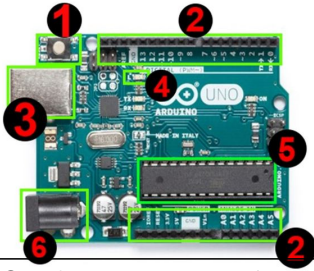
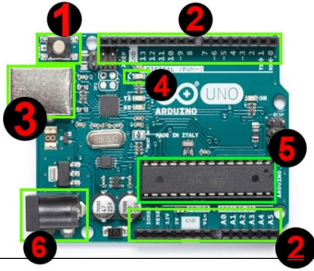
5.5.10.1 Resultados do Kahoot 15

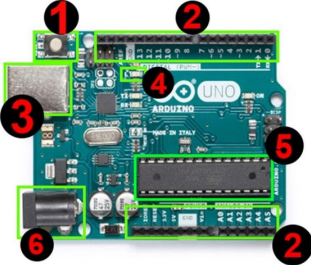
5.5.10.1.1 Kahoot #15 – Todo o conteúdo

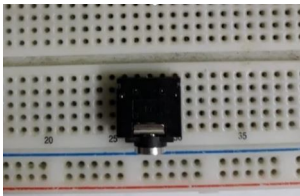
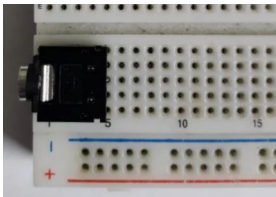
Objetivo: Verificar o quanto os estudantes compreenderam ao longo de todo o curso.



Conteúdo do Kahoot 15: Este Kahoot foi composto por quarenta e oito questões sobre todos os assuntos abordados nos Kahoots anteriores. Foram escolhidas as questões consideradas mais difíceis dos 14 Kahoots anteriores. Apenas do Kahoot 1 não foi utilizada nenhuma questão. Na Tabela 52 é possível verificar quais questões foram realizadas e de quais Kahoots anteriores foram retiradas. As questões não foram modificadas, porém a ordem das alternativas foi modificada de forma aleatória.

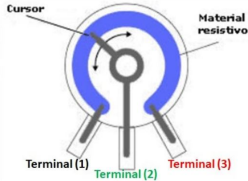
Tabela 52 – Perguntas do Kahoot 15

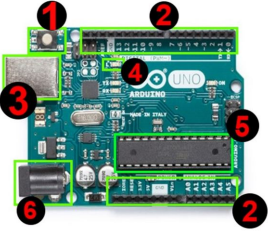
Nº	Questão	Alternativas	
Q15.1	Qual componente é o LED no pino 13? 	A) 1	B) 3
		C) 4	D) 6
Q15.2	Qual componente é utilizado para gravar o código? 	A) 2	B) 3
		C) 4	D) 6

Q15.3	<p>Qual componente é utilizado para iniciar o código novamente?</p> 	A) 1	B) 2
		C) 3	D) 4
Q15.4	Quais pinos do Arduino poderiam ser utilizados?	A) (GND, 12)	B) (GND, ~11)
		C) (GND, ~3)	D) (AREF, ~10)
Q15.5	Porque o Buzzer é chamado de sensor de saída?	A) Porque o buzzer liga e desliga	B) Porque o buzzer recebe um sinal e emite um som
		C) Porque o Buzzer varia a frequência	D) Porque o Buzzer envia um sinal para o Arduino
Q15.6	Para que serve a função setup?	A) É utilizada para configurações iniciais	B) É utilizada para instalar o arduino
		C) É utilizada para ligar o buzzer	D) É utilizada para executar o pinMode
Q15.7	Para que serve a função Loop?	A) É utilizada para configurar o buzzer	B) É utilizada para executar a logica

		C) Não é utilizada	D) É utilizada para ligar a função tone
Q15.8	Quantos segundos a função delay (15000) pausa o programa?	A) 15 milissegundos	B) 1500 segundos
		C) 1,5 segundos	D) 15 segundos
Q15.9	<p>O que esse código faz?</p> <pre>void loop(){ tone(buzzer,NOTE_C3,1000); delay(1000); noTone(buzzer); delay(1000); }</pre>	A) O código dá erro na verificação	B) Não funciona porque está dentro da função setup
		C) Executa a nota C3 no Buzzer apenas 1 vez	D) Executa a nota C3 várias vezes
Q15.10	<p>O conector nesta posição irá funcionar.</p> 	Verdadeiro	Falso
Q15.11	<p>O conector nesta posição irá funcionar</p> 	Verdadeiro	Falso

Q15.12	<p>Como fica a ordem dos resistores do MENOR para o MAIOR</p> 	A) 10kΩ,100Ω,1kΩ	B) 1kΩ,100Ω,10kΩ
		C) 100Ω,1kΩ,10kΩ	D) 100Ω,10kΩ,1kΩ
Q15.13	<p>O que acontece quando troca-se o resistor de 1kΩ por um de 10kΩ no circuito com fone de ouvidos?</p>	A) O volume aumenta	B) O volume continua o mesmo
		C) O volume diminui	D) O volume começa baixo mas depois aumenta
Q15.14	<p>Qual é o valor do resistor?</p> 	A) 10kΩ	B) 1MΩ
		C) 1kΩ	D) 100Ω
Q15.15	<p>O que é um potenciômetro?</p>	A) Sensor de energia	B) Resistor variável
		C) Controlador do Arduino	D) Reprodutor de áudio
Q15.16	<p>Se o potenciômetro estiver na resistência máxima o cursor estará em:</p>	A) No terminal 1 a esquerda	B) No terminal 3 a direita

		C) No terminal 2 no centro	D) Em qualquer posição
Q15.17	Quais terminais são utilizados para que o som saia pelos 2 lados do fone?	A) A e Z	B) Z e Y
		C) A e B	D) B e Y
Q15.18	No Arduino que tipo de sensor é um Pushbutton?	A) Sensor de entrada	B) Sensor de saída
		C) Sensor de interação	D) Sensor de pressão
Q15.19	O Arduino interpreta os comandos do Pushbutton como...	A) Sinal analógico, ou seja valor 0 ou 1	B) Sinal digital, ou seja valor 0 até 1023
		C) Sinal analógico, ou seja valor 0 até 1023	D) Sinal digital, ou seja valor 0 ou 1
Q15.20	Qual a diferença entre o circuito do BOTÃO com resistor e sem o resistor?	A) Com resistor o circuito é mais eficiente	B) Sem o resistor o circuito é mais eficiente

		C) Não há diferença na pratica, apenas facilita a montagem	D) Depende do objetivo do projeto
Q15.21	Quais desses terminais do Arduino NÃO podem ser utilizados para	A) (1, 5V)	B) (GND, A2)
		C) (GND, ~11)	D) (2, GND)
Q15.22	Para que é utilizado parâmetro INPUT_PULLUP?	A) Para utilizar o resistor interno do Arduino no botão	B) Para utilizar um resistor com o botão
		C) Para identificar o botão como entrada	D) Para configurar o botão como digital
Q15.23	O que faz a função digitalRead?	A) Aciona uma nota no fone	B) Le um valor analógico em um pino
		C) Le um valor digital em um pino	D) Envia um sinal para um pino
Q15.24	O que faz o símbolo * na função?	A) Multiplica o valor NOTE_C3 com digitalRead(botao)	B) Divide o valor NOTE_C3 com digitalRead(botao)

		C) Soma o valor NOTE_C3 com digitalRead(botao)	D) Subtrai o valor NOTE_C3 com digitalRead(botão)
Q15.25	Para que é utilizada a função IF?	A) Para executar o buzzer no tempo certo	B) Para verificar se o buzzer está tocando
		C) Para verificar quando a nota é tocada	D) Para verificar quando uma condição é satisfeita
Q15.26	Porque tem o número 1 na função IF? <code>if(digitalRead(NotaC3)==1)</code> { . . . }	A) Para executar apenas 1 vez	B) Para verificar se o valor do botão é 0
		C) Para verificar se o valor do botão é 1	D) Para que o buzzer ligue
Q15.27	Porque são utilizados 2 símbolos de igual na função IF?	A) Para verificar se é igual	B) Para verificar se é diferente
		C) Para atribuir o valor 1 ao botao	D) Para verificar a nota
Q15.28	Como identificar quais são os terminais do LED se os terminais forem cortados?	A) O Chanfro no LED indica o terminal negativo	B) O chanfro no LED indica o lado positivo

		C) Cortando um dos terminais para ficar maior	D) Não tem como identificar
Q15.29	Qual terminal do LED é o POSITIVO?	A) O LED não possui positivo e negativo	B) A parte de cima do LED é o positivo
		C) O terminal mais comprido é o positivo	D) O terminal mais curto é o positivo
Q15.30	Para que é utilizada a função ELSE?	A) Para verificar se o botão foi solto	B) Para verificar se o botão foi pressionado
		C) Executa um comando se a condição do IF não foi satisfeita	D) Para ligar um LED
Q15.31	O que faz o comando digitalWrite?	A) Liga apenas LEDs	B) Envia um comando para uma porta do Arduino
		C) Desliga os LEDs	D) Verifica se um LED está ligado
Q15.32	No protocolo MIDI são transmitidos...	A) Somente dados	B) Somente som

		C) Som e dados	D) Somente Sinais
Q15.33	Quais os comandos para tocar a nota C3 no canal 2?	A) 144,48,100	B) 153,48,100
		C) 145,48,100	D) 143,48,100
Q15.34	A palavra int significa...	A) Intermédio	B) Tipo
		C) Intervalo	D) Inteiro
Q15.35	<p>Porque a função ELSE está depois do PRIMEIRO IF e não do SEGUNDO IF?</p> <pre> if(digitalRead(NotaC3)==1){ if(estadoAntC3 == 0){ ... } }else { ... } </pre>	A) Porque é quando o botao esta solto	B) Porque o segundo IF somente liga a nota
		C) Porque o segundo IF esta dentro de outro IF e não pode usar ELSE	D) Porque o estado anterior esta dentro do primeiro IF

Q15.36	<p>Porque no primeiro IF é verificado se o valor é igual a 1 e no segundo IF é verificado se é igual a 0?</p> <pre> if(digitalRead(NotaC3)==1){ ... } }else { ... } </pre>	A) Esta errado desta forma, deve ser invertida a logica	B) O 1º IF verifica o estado anterior do botao, o 2º verifica o estado atual
		C) O 1º IF verifica o estado atual do botao, o 2º verifica o estado anterior	D) Porque no 1º IF é verificado o botao e no 2º é ligado a nota
Q15.37	Onde devem ser declaradas as funções?	A) Dentro da função setup	B) Depois da função loop
		C) Entre a função setup e loop	D) Dentro da função loop
Q15.38	Qual é a configuração de uma função?	A) TIPO, PARAMETROS, NOME	B) NOME, PARAMETRO, TIPO
		C) TIPO, NOME, PARAMETROS	D) PARAMETRO, TIPO, NOME
Q15.39	O que são os parâmetros de uma função?	A) São valores que a função precisa	B) São variáveis que todo código pode acessar

		C) São valores que a função recebe para serem utilizados na logica	D) São valores que recebem apenas comandos MIDI
Q15.40	Os piezoelétricos possuem valores...	A) Digitais, ou seja valores entre 0 e 1023	B) Digitais, ou seja valores 0 ou 1
		C) Analógicos, ou seja valores entre 0 e 1023	D) Analógicos, ou seja valores 0 ou 1
Q15.41	Para que serve a função Serial.println?	A) Para imprimir um valor	B) Para mandar um comando MIDI
		C) Para imprimir um valor/palavra e pular uma linha	D) Para pular uma linha
Q15.42	O que é o Monitor Serial?	A) É onde é monitorado o Arduino	B) É onde o piezo envia os valores
		C) É como Arduino faz gráficos	D) É a forma e visualizar os valores da porta serial
Q15.43	Os piezoelétricos possuem valores...Para que serve a variável delayP1?	A) Para armazenar o valor do piezo1	B) Para armazenar o valor do map

		C) Para armazenar o tempo de espera para tocar a próxima nota	D) Para pausar o programa
Q15.44	O que faz a função millis()?	A) Para pausar o programa	B) Conta o tempo desde o momento que o Arduino está ligado
		C) Conta valores a 1000 segundos	D) Atualiza o tempo
Q15.45	Em qual dos terminais do Arduino deve ser colocado o componente metálico para tocar (DMI Xilofone Capacitivo)?	A) Terminal de envio	B) Terminal GND
		C) Qualquer Terminal	D) Terminal de Recebimento
Q15.46	O que é um Teremim?	A) Instrumento musical capacitivo	B) Instrumento de medição
		C) Instrumento musical Digital	D) Instrumento Musical Eletrônico
Q15.47	O Sensor Capacitivo é um...	A) Sensor de saída	B) Sensor de entrada

		C) Sensor de intensidade	D) Sensor de pressão
Q15.47	Para que servem os 3 resistores de 10M no Teremim?	A) Para diminuir a capacitância	B) Para aumentar a sensibilidade do instrumento
		C) Para evitar ruídos	D) Para variar a capacitância

Fonte: O Autor, 2020

Resultados: É possível observar na Tabela 53 que as questões respondidas incorretamente foram Q15.33, Q15.43 e Q15.48 pelo estudante E1, enquanto Q15.12, Q15.14, Q15.16, Q15.33 e Q15.36 pelo estudante E2, já para o estudante E3 foram as questões Q15.18, Q15.31, Q15.33, Q15.35, Q15.36, Q15.42 e Q15.44, por fim Q15.1, Q15.5, Q15.8, Q15.14, Q15.516, Q15.19, Q15.33, Q15.34, Q15.36, Q15.40, e Q15.44 para o estudante E4. A questão Q15.33 obteve o pior desempenho, visto que todos os estudantes responderam incorretamente, seguida da questão Q15.36 a qual apenas 1 estudante respondeu corretamente.

Tabela 53 – Resultados do Kahoot 15

Kahoot 15				
	E1	E2	E3	E4
Q1				✗
Q2				
Q3				
Q4				
Q5				✗
Q6				
Q7				
Q8				✗
Q9				
Q10				
Q11				
Q12		✗		
Q13				
Q14		✗		✗
Q15				
Q16		✗		✗
Q17				
Q18			✗	
Q19				✗
Q20				
Q21				
Q22				
Q23				
Q24				
Q25				
Q26				
Q27				
Q28				
Q29				
Q30				
Q31			✗	
Q32				
Q33	✗	✗	✗	✗
Q34				
Q35			✗	
Q36		✗	✗	✗
Q37				
Q38				
Q39				
Q40				✗
Q41				
Q42			✗	
Q43	✗			
Q44			✗	✗
Q45				
Q46				
Q47				
Q48	✗			

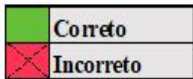
	Correto
✗	Incorreto

Ao final deste Kahoot os estudantes responderam a um formulário com perguntas referentes as suas percepções sobre o curso.

Na Tabela 54 é possível verificar os resultados dos estudantes da primeira vez que responderam as questões na coluna K1 e comparar com os resultados da segunda vez que responderam as mesmas questões na coluna K2. Desta forma foi observado que houve um aumento de questões respondidas corretamente na segunda vez que este Kahoot foi aplicado, indicando que houve aprendizado e retenção do conhecimento adquirido ao longo das aulas.

Tabela 54 – Resultados do Kahoot 15 comparados aos outros Kahoots

	K1	K2	K1	K2	K1	K2	K1	K2
	E1		E2		E3		E4	
Q1								
Q2								
Q3								
Q4								
Q5								
Q6								
Q7								
Q8								
Q9								
Q10								
Q11								
Q12								
Q13								
Q14								
Q15								
Q16								
Q17								
Q18								
Q19								
Q20								
Q21								
Q22								
Q23								
Q24								
Q25								
Q26								
Q27								
Q28								
Q29								
Q30								
Q31								
Q32								
Q33								
Q34								
Q35								
Q36								
Q37								
Q38								
Q39								
Q40								
Q41								
Q42								
Q43								
Q44								
Q45								
Q46								
Q47								
Q48								



Na Tabela 55 é possível verificar que todos os estudantes aumentaram a taxa de acerto das questões quando aplicadas pela segunda vez, obtendo um valor igual ou superior a 77% no final do curso.

Tabela 55 – Taxa de acerto dos Kahoots

	E1	E2	E3	E4
K1	75%	79%	69%	44%
K2	94%	90%	85%	77%

Fonte: O Autor, 2020

Na Tabela 56 são demonstrados os progressos dos estudantes divididas em 4 categorias, com base no resultado da primeira vez que respondeu o Kahoot e o resultado da segunda vez. As categorias para cada estudante são:

(EA) = Errou da primeira vez e acertou na segunda vez;

(EE) = Errou da primeira vez e errou na segunda vez;

(AE) = Acertou da primeira vez e errou na segunda vez;

(AA) = Acertou da primeira vez e acertou na segunda vez

Tabela 56 – Taxa de acerto dos Kahoots por categoria

	E1	E2	E3	E4
EA	23%	17%	27%	46%
EE	2%	4%	4%	10%
AE	4%	6%	10%	13%
AA	71%	73%	58%	31%

Fonte: O Autor, 2020

É possível verificar que os estudantes E1, E2 e E3 obtiveram um maior número de questões com AA, o que indica que os estudantes compreenderam os conteúdos da primeira vez e continuaram dominando o assunto ao longo das aulas na maioria das questões. Já o estudante E4 obteve um número maior de questões na categoria EA, o que indica que na primeira vez o estudante não havia compreendido o assunto, porém ao final das aulas após as explicações e de realizar todas as atividades práticas, o estudante conseguiu evoluir seus conhecimentos. Todos os estudantes obtiveram um valor menor que 10% na categoria EE, o que indica que foram poucas questões em que os estudantes erraram da primeira vez e continuaram errando. Enquanto na categoria AE foi menor ou igual a 13%, o que indica que os estudantes haviam acertado a questão na primeira vez, porém erraram na segunda vez.

5.5.10.2 Observações realizadas durante a aula 10

Ao final desta aula foi realizado um questionário com a intenção de verificar a percepção dos estudantes em relação ao conteúdo do curso. As respostas de cada estudante são apresentadas na Tabela 57

Tabela 57 – Resultados Questionário sobre a percepção dos Estudantes

Pergunta	E1	E2	E3	E4
Sobre a afirmação: “No início não achava que conseguiria realizar as atividades”	Discordo parcialmente	Discordo totalmente	Concordo parcialmente	Não concordo nem discordo
Sobre a afirmação: “Acredito que utilizarei o que aprendi no curso em outras atividades”.	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
Acredito que o que aprendi é útil para meu conhecimento.	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
O que você considera que foi mais difícil de ser realizado durante o curso? Justifique.	A montagem e calibragem dos piezos	O mais difícil foi compreender o funcionamento dos circuitos elétricos	Realizar a montagem dos componentes eletrônicos, e entender algumas funções mais complexas do Arduino	não teve nada tão difícil, eu só não conhecia nada sobre o Arduino e achei o curso top

O que você considera que foi mais fácil de ser realizado durante o curso? Justifique.	Montagem dos componentes (até chegar nos LEDs)	O mais fácil de ser realizado foi entender a linguagem de programação.	Aplicar os conceitos musicais para a programação.	A parte mais fácil era de fazer montar e desmontar as pecinhas no Arduino e fazer a musiquinha nas primeiras aulas
Sobre a afirmação “Aprendi muito sobre programação” você considera que:	Aprendi todo conteúdo	Aprendi todo conteúdo	Aprendi mais ou menos	Aprendi quase todo conteúdo
Sobre a afirmação “Aprendi muito sobre eletrônica” você considera que:	Aprendi quase todo conteúdo	Aprendi quase todo conteúdo	Aprendi mais ou menos	Aprendi quase todo conteúdo
Sobre a afirmação “Aprendi muito sobre música” você considera que:	Aprendi quase todo conteúdo	Aprendi todo conteúdo	Aprendi pouco	Aprendi mais ou menos
Após a finalização do curso você se considera capacitado para construir seu próprio projeto?	Muito capaz	Muito capaz	Muito capaz	Extremamente capaz

Sobre a afirmação: “A compreensão dos assuntos apresentados é importante para mim”.	Concordo totalmente	Concordo totalmente	Concordo parcialmente	Concordo totalmente
Sobre a afirmação: “Acredito saber construir um Instrumento Musical Digital (DMI)”	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
Sobre a afirmação: “Os softwares utilizados como Arduino IDE e Freepiano são fáceis de serem manuseados”.	Concordo totalmente	Concordo parcialmente	Concordo totalmente	Concordo totalmente
Sobre a afirmação: “Os componentes utilizados como placa Arduino, protoboard, resistores, etc. são fáceis de serem manuseados”.	Concordo totalmente	Concordo parcialmente	Concordo parcialmente	Concordo totalmente

Sobre a afirmação: “Gostaria de retornar em uma próxima versão do curso”.	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
Sobre a afirmação: “Gostei do que aprendi neste curso”.	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente
Sobre a afirmação: “O xilofone de botão com Buzzer foi o DMI que mais gostei de construir”.	Concordo parcialmente	Discordo totalmente	Concordo parcialmente	Concordo parcialmente
Sobre a afirmação: “O xilofone de botão com MIDI foi o DMI que mais gostei de construir”.	Concordo totalmente	Discordo parcialmente	Concordo totalmente	Concordo totalmente
Sobre a afirmação: “O xilofone de piezo foi o DMI que mais gostei de construir”.	Não concordo nem discordo	Discordo parcialmente	Concordo totalmente	Não concordo nem discordo

Sobre a afirmação: “O xilofone capacitivo foi o DMI que mais gostei de construir”.	Concordo parcialmente	Concordo totalmente	Concordo totalmente	Não concordo nem discordo
Sobre a afirmação: “O teremim capacitivo foi o DMI que mais gostei de construir”.	Não concordo nem discordo	Discordo totalmente	Concordo parcialmente	Não concordo nem discordo

Fonte: O Autor, 2020

Ao perguntar se antes de iniciar o curso os estudantes acreditavam conseguir realizar as atividades os estudantes que possuem conhecimentos prévios afirmaram conseguir realizar as atividades sem muitas dificuldades, enquanto aqueles que não possuem conhecimentos prévios não tinham certeza se iriam conseguir.

Apesar da proposta ser diferente de sua área de conhecimento, nenhum dos estudantes respondeu que acreditavam não conseguir realizar as atividades.

Todos os estudantes consideraram que o conteúdo aprendido será útil para seu desenvolvimento e que o utilizarão em outras atividades além do curso.

Foram realizadas duas perguntas com respostas em aberto sobre as dificuldades e facilidades na percepção dos estudantes em relação as dificuldades, as respostas foram:

1. “O mais difícil foi compreender o funcionamento dos circuitos elétricos”;
2. “Realizar a montagem dos componentes eletrônicos, e entender algumas funções mais complexas do Arduino”;
3. “A montagem e calibragem dos piezos”;
4. “Não teve nada tão difícil, eu só não conhecia nada sobre o Arduino e achei o curso top”.

É possível perceber que a maioria dos estudantes encontrou alguma dificuldade na parte de eletrônica. Considerando que este foi o primeiro contato prático com componentes

eletrônicos dos estudantes é comum ter uma dificuldade maior nesta etapa. Alguns dos sensores utilizados, como o piezoelétrico, necessitam de uma calibração prévia, isso foi relatado como uma dificuldade, pois estes sensores mudam conforme o ambiente e não são todos idênticos, o que demanda mais tempo para calibrar e quanto mais sensores são utilizados mais testes devem ser realizados. Outra dificuldade foi adicionar LEDs ao circuito os quais necessitam de um resistor cada um, portanto o circuito fica mais complexo. Esta atividade utiliza as luzes e a música respondendo aos comandos de um botão. O material foi estruturado de forma que as aulas estejam interligadas, assim a partir da 2ª aula os estudantes montavam novamente o circuito da aula anterior com auxílio do material e de fotos tiradas por eles mesmos. Apesar de relatarem a dificuldade na etapa de eletrônica todos os estudantes utilizavam os 30 minutos iniciais para remontarem sozinhos todo o circuito.

Em relação as facilidades as respostas foram:

1. “O mais fácil de ser realizado foi entender a linguagem de programação.”
2. “Aplicar os conceitos musicais para a programação.”
3. “Montagem dos componentes (até chegar nos Leds).”
4. “A parte mais fácil era de fazer montar e desmontar as pecinhas no Arduino e fazer a musiquinha nas primeiras aulas.”

Os estudantes relataram ter mais facilidade em conceitos musicais aplicados na programação do Arduino, como configurar o MIDI, configurar tempos e notas musicais, construir músicas utilizando a programação. Exceto a etapa com LEDs, dois dos estudantes consideraram mais fácil montar e desmontar os componentes do que programar.

Os estudantes foram questionados sobre sua retenção de aprendizado sobre os conteúdos. Sobre programação dois estudantes relataram terem aprendido todo o conteúdo, outro acredita ter aprendido quase todo o conteúdo e outro acredita ter aprendido mais ou menos. Comparando com o primeiro questionário (Tabela 14) nota-se que os estudantes que consideraram ter aprendido todo o conteúdo são os mesmos com experiências anteriores em programação. Com relação a eletrônica nenhum dos estudantes consideraram ter aprendido todo o conteúdo. Com essas duas perguntas correlacionadas as perguntas abertas, é possível perceber que a maioria dos estudantes consideram que aprenderam sobre os conteúdos e que aprenderam menos nos tópicos com mais dificuldades. Um dos estudantes relatou ter aprendido mais ou menos sobre programação e eletrônica, em comparação com as respostas em aberto este estudante considerou difícil compreender algumas funções mais complexas e montar os circuitos. Mesmo com as dificuldades apresentadas todos os estudantes se consideram ao menos muito capazes de construir seu próprio projeto inclusive um instrumento musical digital.

Todos os estudantes consideraram os softwares e hardwares fáceis de serem manuseados e configurados. O interessante desta resposta é que os softwares utilizados são gratuitos e alguns dos estudantes testaram seus próprios softwares para a etapa musical. Enquanto os componentes utilizados não são materiais didáticos, ou seja, não possuem uma placa pronta que facilitaria a montagem, com exceção da placa Arduino.

Ao todo os estudantes construíram 5 instrumentos musicais digitais diferentes. A maioria dos instrumentos construídos foram chamados de Xilofone, pois foram programadas utilizando apenas as cinco notas naturais (sem os sustenidos), o que se assemelha com um Xilofone simples:

- Xilofone de botão: Instrumento que possuía 5 botões e gerava sons através de um buzzer/fone;
- Xilofone de botão MIDI: Continuação do instrumento anterior, mas com utilizando comandos MIDI e o computador para executar o som;
- Xilofone de piezo: Instrumento utilizando o sensor piezoelétrico, o qual se aproxima de um Xilofone real;
- Xilofone capacitivo: Instrumento musical utilizando sensores capacitivos com funcionamento de toque;
- Teremim capacitivo: Instrumento musical utilizando sensores capacitivos que funciona através da aproximação das mãos.

O Xilofone de piezo e o Teremim foram os DMIs que os estudantes menos gostaram de construir. São os instrumentos que mais necessitam de calibração e sofrem mais interferências do ambiente, por este motivo levam mais tempo para montar. Enquanto o Xilofone de botão MIDI e o Xilofone capacitivo foram os DMIs que os estudantes gostaram mais de construir.

5.5.10.3 Considerações sobre a aula

Esta aula foi composta pelo Kahoot 15 contendo perguntas combinadas de todos os Kahoots anteriores, pelo formulário apresentado na Tabela 52 e por fim foi realizada uma discussão sobre o curso. Na discussão os estudantes relataram que o curso foi muito curto, apenas 10 aulas e com assuntos muito básicos. Também foi proposto o desenvolvimento de uma apostila para poder acompanhar as aulas e realizar as atividades práticas, além de criar uma pasta compartilhada contendo os arquivos. Outra proposta foi disponibilizar um kit contendo os componentes e placa Arduino para serem comprados no início do curso. Com relação as aulas os estudantes relataram que os Kahoots foram divertidos de realizar, sendo uma forma de avaliação melhor do que provas convencionais. Como ideia para o

curso foi proposto utilizar LEDs em conjunto das notas musicais para ensinar conceitos musicais. Com relação as dificuldades, os estudantes (E1 e E2) com conhecimento prévio em programação consideraram não ter grandes dificuldades com o código, os estudantes E3 e E4 relataram ter um pouco de dificuldades para organizar o código. Já em relação aos temas de eletrônica, todos os estudantes relataram ter dificuldade de compreender o funcionamento dos componentes eletrônicos.

6 Conclusão

Esta pesquisa emerge da necessidade de desenvolver abordagens educacionais, voltadas para o ensino de programação e eletrônica, que possibilitem estudantes serem desenvolvedores/criadores de tecnologia. Para isso, a maioria dos trabalhos encontrados na literatura utilizam a robótica pedagógica, mas tornam-se um nicho específico dentro do contexto do *STEM education*. Contudo, outros trabalhos apresentam resultados positivos utilizando Instrumentos Musicais Digitais (DMIs) comerciais. Assim, é acrescentando o A de Artes ao *STEM* formando o *STEAM education*.

A utilização de DMIs comerciais em abordagens educacionais para o ensino de programação, eletrônica e/ou música possuem dificuldades de escalabilidade devido aos altos custos dos DMIs comerciais. Isso dificulta a utilização destas abordagens em lugares com baixo poder aquisitivo como as escolas públicas. Desta forma, esta pesquisa contribui com uma abordagem inédita no contexto do *STEAM education* para o ensino de programação e eletrônica por meio da construção de instrumentos musicais digitais de baixo custo.

Diante disso, esta pesquisa teve como objetivo geral analisar como a construção de DMIs de baixo custo pode auxiliar no aprendizado de programação e eletrônica. Assim, verifica-se que o objetivo geral foi atingido a partir da prototipação de seis DMIs (Teclado, Xilofone, Bateria, Teremim, Harpa e Controlador MIDI) e da validação dos DMIs por meio de um curso de formação para estudantes de Licenciatura em Música sem os conhecimentos prévios de programação e eletrônica.

Para alcançar os objetivos da pesquisa utilizou-se do método de Pesquisa de Desenvolvimento de Objeto ([VAN DER MAREN, 2004](#)), no qual cria-se um objeto para atender à demanda encontrada a partir da análise do contexto. Observou-se que a demanda se trata do *STEAM education* e o objeto a construção de DMIs de baixo custo para o ensino de programação e eletrônica. No entanto, verificou-se que o método precisava ser reproduzido duas vezes. Uma para a construção dos protótipos dos DMIs de baixo custo que possui o contexto de criação dos protótipos de baixo custo e outra para a aplicação dos DMIs no ensino de programação e eletrônica que possui o contexto de validar os protótipos construídos em uma abordagem educacional. Com isso, foram desenvolvidos dois objetos pedagógicos.

O primeiro objeto foi a construção de DMIs de baixo custo, o que proporcionou uma avaliação sobre suas funcionalidades, dificuldades e os custos. O segundo objeto foi o curso de formação, onde foram observados o desempenho dos estudantes sobre os conteúdos de programação e eletrônica. Constata-se que é possível utilizar DMIs de baixo custo para o ensino programação e eletrônica. Além disso, a motivação de construir DMIs para fazer

música durante as atividades auxiliou no aprendizado dos estudantes.

O objeto 1 responde o primeiro objetivo específico desta pesquisa - Avaliar a viabilidade de construção de DMIs de baixo custo. Foram construídos 6 DMIs diferentes (Teclado, Xilofone, Bateria, Harpa, Teremim e Controlador MIDI) utilizando a placa Arduino e diferentes tipos de sensores e materiais reutilizáveis. Foram avaliados o funcionamento dos DMIs com base na complexidade de construção, capacidade de geração sonora e na possibilidade de reutilização de códigos e circuitos. Diante disso, os DMIs foram testados com diferentes combinações de componentes a fim de escolher a melhor combinação para prototipação. Assim, foram testados um total de 13 protótipos no formato dos 6 DMIs definidos inicialmente, mas com diferentes componentes. Destes 13 protótipos, foram escolhidas 7 com melhor desempenho para avaliação dos custos.

Verifica-se que foi possível construir diferentes DMIs funcionais com custo baixo, utilizando a placa de desenvolvimento Arduino UNO e diferentes combinações de sensores (toque, pressão, luz e movimento), bem como a utilização de materiais reutilizáveis (tampa de plástico, papelão, cliques de papel, palitos de madeira, quadro de madeira). A escolha destes componentes possibilitou reduzir os custos para construção dos DMIs ficando em torno de R\$40 e R\$94, em comparação com os DMIs comerciais utilizados em outras abordagens ficando em torno de R\$3000 e R\$6000.

O objeto 2 responde ao segundo objetivo específico desta pesquisa - Verificar os desafios impostos no planejamento de curso que relacione o ensino de programação e eletrônica a partir da construção de DMIs de baixo custo. A constatação do objetivo é observado pelo plano de ensino elaborado para o curso de formação que contempla os conteúdos sobre programação e eletrônica necessários para alcançar os objetivos das aulas, os resultados de aprendizagem que atesta quais conhecimentos os estudantes são capazes de adquirir ao final do processo, as atividades de ensino que apresenta de que forma são realizadas as aulas, os materiais de apoio que contém o que será utilizado em cada aula e as atividades avaliativa que apresenta a forma utilizada para medir o desempenho dos estudantes.

O público-alvo escolhido para participar do curso de formação foram estudantes de Licenciatura em Música que não possuíam o conhecimento prévio de programação e eletrônica. Esta escolha se deu devido a estes estudantes possuírem um perfil de formadores e, por conta disso, serem possíveis multiplicadores desse tipo de abordagem educacional. Foram elaboradas 10 aulas, onde cada aula foi planejada para que os estudantes construíssem DMIs como motivação para aprender o básico de programação e eletrônica. Foram escolhidos 4 DMIs (Teclado, Xilofone, Bateria e Teremim) para serem construídos ao longo das aulas. Foi utilizada uma abordagem gameficada, na qual os estudantes eram expostos aos conteúdos teóricos de programação e eletrônica, construíam na prática o DMI, faziam música e identificavam falhas ou possíveis melhorias no DMI. Além disso, foi

utilizada a ferramenta Kahoot com os conteúdos teóricos apresentados, a fim de mensurar o desempenho dos estudantes, em cada aula e na última aula foi elaborado um Kahoot contendo os conteúdos de todas as aulas a fim de verificar a evolução dos estudantes em comparação aos Kahoots anteriores.

De forma geral, verifica-se que os 4 estudantes conseguiram aprender e conseguiram reter os conteúdos sobre programação e eletrônica. Ainda que os estudantes não tivessem conhecimentos prévios destas áreas, obtiveram um desempenho satisfatório nos Kahoots com média de acerto de 70%. Observou-se que todos os estudantes conseguiram desenvolver todos os protótipos propostos nas atividades sem grandes dificuldades. Por meio de observações realizadas durante as aulas, verificou-se que os estudantes estavam motivados em aprender os assuntos abordados, visto que todos os estudantes voltariam a fazer uma nova versão do curso, além disso este trabalho contribuiu para que ao menos três estudantes adquirissem, de forma espontânea, os componentes utilizados no curso com a intenção de utilizarem durante as aulas e para seus projetos particulares.

A partir dos resultados observados no objeto 2, verifica-se o terceiro objetivo específico - Identificar os limites e possibilidades da integração do ensino de programação e eletrônica a partir da construção de DMIs de baixo custo. As limitações identificadas nesta etapa foram que para o formador replicar esta abordagem, é necessário ter um conhecimento básico de programação, eletrônica e música. É importante reservar um tempo para que os estudantes construam os circuitos da aula anterior, pois auxilia no aprendizado, principalmente na organização dos circuitos e memorização por repetição. É necessário separar um espaço para armazenamento dos componentes de forma organizada para que os estudantes possam identificá-los. Os DMIs escolhidos para o curso de formação não possuíam uma etapa para construção de acabamento, como caixas, pinturas e suportes. Alguns sensores escolhidos (capacitivo e piezoelétrico) sofrem com ruídos do ambiente, o que demanda de calibração por diversas vezes.

A partir das observações realizadas durante o curso de formação e das sugestões dos estudantes foram identificadas possibilidades de construção de um material didático com as instruções para o formador e para o estudante utilizarem durante as aulas. Utilizar softwares para afinação de instrumentos para auxiliar na construção dos DMIs que utilizem Buzzer.

Os resultados apresentados demonstram que a construção de DMIs de baixo custo soma-se ao ensino de programação e eletrônica como um fator motivacional para o aprendizado dos estudantes e apresenta uma aplicação do mundo real utilizando as teorias de programação e eletrônica de forma análoga à robótica pedagógica.

Referências

- ALIMISIS, D. Educational robotics : Open questions and new challenges. *Themes in Science and Technology Education*, v. 6, n. 1, p. 63–71, 2013. Citado na página 26.
- ALVES, L. R. G.; MINHO, M. R. d. S.; DINIZ, M. V. C. Gamificação: diálogos com a educação. *FADEL, Luciane Maria et al. (Org.). Gamificação na educação. São Paulo: Pimenta Cultural*, p. 74–97, 2014. Citado na página 98.
- ANTUNES, S. F.; TEIXEIRA, A. C.; PIREDDU, M. Robótica livre como alternativa didática para a aprendizagem de música. *RENOTE-Revista Novas Tecnologias na Educação*, v. 15, n. 2, 2017. Citado na página 49.
- ANWAR, S.; BASCOU, N. A.; MENEKSE, M.; KARDGAR, A. A Systematic Review of Studies on Educational Robotics. *Journal of Pre-College Engineering Education Research (J-PEER)*, Purdue University Press, v. 9, n. 2, p. 19–42, jul 2019. ISSN 2157-9288. Disponível em: <<https://docs.lib.purdue.edu/jpeer/vol9/iss2/2>>. Citado 2 vezes nas páginas 25 e 26.
- Arduino Project Hub. *Arduino Project Hub*. 2020. Disponível em: <<https://create.arduino.cc/projecthub/projects/tags/music>>. Citado na página 34.
- ARDUINO.CC. *ARDUINO*. 2018. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Citado na página 39.
- ARELLANO, D. G.; MCPHERSON, A. P. Radear : A Tangible Spinning Music Sequencer. *NIME 2014 Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 84–85, 2014. Citado na página 54.
- ARIMOTO, M.; OLIVEIRA, W. Dificuldades no processo de aprendizagem de programação de computadores: um survey com estudantes de cursos da área de computação. In: SBC. *Anais do XXVII Workshop sobre Educação em Computação*. [S.l.], 2019. p. 244–254. Citado na página 23.
- ARIMOTO, M. M.; CRUZ, J. H. R. Ensino de lógica e programação no ensino médio por meio de uma abordagem lúdica e gamificada. In: SBC. *Anais do XXVIII Workshop sobre Educação em Computação*. [S.l.], 2020. p. 166–170. Citado na página 23.
- AZEVEDO, S.; AGLAÉ, A.; PITTA, R. Minicurso : Introdução a Robótica Educacional. *62a. Reunião Anual da SBPC - 25 a 30 de julho de 2010.*, p. 1–41, 2010. Disponível em: <<http://www.sbpcnet.org.br/livro/62ra/minicursos/MCSamuelAzevedo.pdf>>. Citado na página 37.
- BALDWIN, A.; HAMMER, T.; PECHIULIS, E.; WILLIAMS, P.; OVERHOLT, D.; SERAFIN, S. Tromba Moderna: A Digitally Augmented Medieval Instrument. *Proceedings of the International Conference on New Interfaces for Musical Expression*, v. 16, p. 14–19, 2016. Disponível em: <http://www.nime.org/proceedings/2016/nime2016_paper0004.pdf>. Citado na página 54.

- BANZI, M.; SHILOH, M. *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*. [S.l.]: Make Community, LLC, 2011. ISBN 9781449363291. Citado na página 71.
- BARAK, M. Teaching Electronics: From Building Circuits to Systems Thinking and Programming. In: *Handbook of technology education*. [S.l.: s.n.], 2018. p. 337–360. ISBN 9783319388892. Citado na página 25.
- BARBOSA, J.; CALEGARIO, F.; TEICHRIB, V.; RAMALHO, G.; CABRAL, G. Illusio: A Drawing-Based Digital Music Instrument. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2013. Citado na página 53.
- BARENCA, A.; CORAK, M. The Manipuller II: Strings within a Force Sensing Ring. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 589–592, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_364.pdf>. Citado na página 54.
- BARNES, J.; FAKHRHOSSEINI, M.; VASEY, E.; DUFORD, Z.; RYAN, J.; JEON, M. Child-robot Theater: STEAM education in an afterschool program. *ACM/IEEE International Conference on Human-Robot Interaction*, n. 2015, p. 404, 2017. ISSN 21672148. Citado 2 vezes nas páginas 26 e 49.
- BARNES, J.; VASEY, E.; JEON, M.; Maryam FakhrHosseini, S.; PARK, C. H. Informal stEAM education case study: Child-robot musical theater. *Conference on Human Factors in Computing Systems - Proceedings*, p. 1–6, 2019. Citado 2 vezes nas páginas 26 e 49.
- BARR, M. Pulse width modulation. *Embedded Systems Programming*, v. 14, n. 10, p. 103–104, 2001. Citado na página 55.
- BEGOSSO, L. C.; BEGOSSO, L. R.; GONÇALVES, E. M.; GONÇALVES, J. R. An approach for teaching algorithms and computer programming using greenfoot and python. In: IEEE. *2012 Frontiers in Education Conference Proceedings*. [S.l.], 2012. p. 1–6. Citado na página 23.
- BELTER, D.; SKRZYPCZYNSKI, P.; WALAS, K.; WLODKOWIC, D. Affordable Multi-legged Robots for Research and STEM Education: A Case Study of Design and Technological Aspects. *Advances in Intelligent Systems and Computing*, v. 351, p. 181–190, 2015. ISSN 21945357. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84924690402&partnerID=tZOtx>>. Citado na página 26.
- BELVEDERE, P. *Arduino UNO – Fundamentos e aplicações*. SENAI-SP Editora, 2018. (Eletroeletrônica). ISBN 9788583938576. Disponível em: <<https://books.google.com.br/books?id=ADxFDwAAQBAJ>>. Citado na página 74.
- BERDAHL, E.; JU, W. Satellite CCRMA: A Musical Interaction and Sound Synthesis Platform. *NIME 2011 Proceedings of the International Conference on New Interfaces for Musical Expression*, n. June, p. 173–178, 2011. Citado na página 53.
- BERTHAUT, F.; KNIBBE, J. Wubbles: A Collaborative Ephemeral Musical Instrument. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 499–500, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_334.pdf>. Citado na página 54.

- BEZERRA JUNIOR, J. E.; QUEIROZ, P. G. G.; LIMA, R. W. de. A study of the publications of educational robotics: A Systematic Review of Literature. *IEEE Latin America Transactions*, v. 16, n. 4, p. 1193–1199, apr 2018. ISSN 1548-0992. Disponível em: <<https://ieeexplore.ieee.org/document/8362156/>>. Citado 2 vezes nas páginas 25 e 36.
- BLESSING, M.; BERDAHL, E. Textural Crossfader. *NIME 2015 Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 4–5, 2015. Citado na página 54.
- BUECHLEY, L.; EISENBERG, M. The lilypad arduino: Toward wearable engineering for everyone. *IEEE Pervasive Computing*, IEEE, v. 7, n. 2, p. 12–15, 2008. Citado na página 71.
- BUECHLEY, L.; EISENBERG, M.; CATCHEN, J.; CROCKETT, A. The LilyPad Arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. *Conference on Human Factors in Computing Systems - Proceedings*, p. 423–432, 2008. Citado na página 49.
- BUNYAMIN, M. A. H.; FINLEY, F. STEM Education in Malaysia: Reviewing the Current Physics Curriculum. *International Conference of Association for Science Teacher Education*, 2016. Citado na página 26.
- BURG, J.; ROMNEY, J.; SCHWARTZ, E. Computer science "big ideas" play well in digital sound and music. *SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, p. 663–668, 2013. Citado na página 49.
- BYBEE, R. W. What Is STEM Education? *Science*, v. 329, n. 5995, p. 996–996, aug 2010. ISSN 0036-8075. Disponível em: <<https://www.sciencemag.org/lookup/doi/10.1126/science.1194998>>. Citado na página 25.
- CALEGARIO, F.; WANDERLEY, M. M.; HUOT, S.; CABRAL, G.; RAMALHO, G. A method and toolkit for digital musical instruments: generating ideas and prototypes. *IEEE MultiMedia*, IEEE, v. 24, n. 1, p. 63–71, 2017. Citado na página 29.
- CARSTENS, J. R. *Electrical sensors and transducers*. [S.l.]: Regents/Prentice Hall Englewood Cliffs, NJ., 1993. Citado na página 41.
- CASSINELLI, A.; KURIBARA, Y.; ZERROUG, A.; ISHIKAWA, M.; MANABE, D. scoreLight: Playing with a Human-Sized Laser Pick-Up. *NIME 2010 Proceedings of the International Conference on New Interfaces for Musical Expression*, v. 10, n. June 2010, p. 144–149, 2010. Citado na página 53.
- CHAN, T. *The Best MIDI Keyboard Controllers*. 2019. <<https://www.rollingstone.com/product-recommendations/electronics/best-midi-keyboard-controller-885672/>>. Acesso: 04/04/2020. Citado na página 56.
- CHUNG, C. C. J.; CARTWRIGHT, C.; CHUNG, C. Robot music camp 2013: An experiment to promote stem and computer science. In: IEEE. *2014 IEEE Integrated STEM Education Conference*. [S.l.], 2014. p. 1–7. Citado na página 29.
- CHUNG, S.-M.; WU, C.-T. Creating a teaching and learning experience for designing interactive applications: Digital musical instruments. p. 448–452, 2017. Citado na página 29.

COLLECCHIA, R.; SOMEN, D.; MCELROY, K. The Siren Organ. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 391–394, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_558.pdf>. Citado 2 vezes nas páginas 32 e 54.

COLWELL, B. Me and my theremin. *Computer*, v. 36, n. 2, p. 8–9, 2003. ISSN 00189162. Citado 2 vezes nas páginas 68 e 69.

CONRADTY, C.; BOGNER, F. X. From STEM to STEAM: How to Monitor Creativity. *Creativity Research Journal*, Taylor & Francis, v. 30, n. 3, p. 233–240, 2018. ISSN 10400419. Disponível em: <<https://doi.org/10.1080/10400419.2018.1488195>>. Citado 2 vezes nas páginas 24 e 26.

CRESWELL, J. W. *Creswell, J.W. projeto de pesquisa- método qualitativo, quantitativo e misto. Tradução de Luciana de oliveira da rocha. 2 Ed. Porto Alegre-artmed, 2007.* [s.n.], 2010. 296 p. ISBN 9788536323008. Disponível em: <<http://ir.obihiro.ac.jp/dspace/handle/10322/3933>>. Citado na página 43.

CROMBIE, D. *THE COMPLETE SYNTHESIZER: A comprehensive Guide by David Crombie (1982).* [S.l.]: Omnibus Pr & Schirmer Trade Books (August 1, 1982), 1982. Citado na página 31.

CROWTHER, G. J.; MCFADDEN, T.; FLEMING, J. S.; DAVIS, K. Leveraging the power of music to improve science education. *International Journal of Science Education*, v. 38, n. 1, p. 73–95, jan 2016. ISSN 0950-0693. Disponível em: <<http://www.tandfonline.com/doi/full/10.1080/09500693.2015.1126001>>. Citado na página 27.

D'ABREU, J. V. V.; RAMOS, J. J.; ROCHA, A. P.; BEZZON, G.; XAVIER, S.; SOUZA, J. L. de. *Uma experiência de implementação de robótica e computação física no Brasil.* Penso Editora, 2020. 46–64 p. (Tecnologia e Inovação na Educação Brasileira). ISBN 9788584291892. Disponível em: <<https://books.google.com.br/books?id=PnCrDwAAQBAJ>>. Citado 2 vezes nas páginas 37 e 38.

D'ALESSANDRO, N.; TILMANNE, J.; MOREAU, A.; PULEO, A. Modelling dependencies of IT infrastructure elements. *Proceedings of the 2015 Conference on New Interfaces for Musical Expression (NIME 2015)*, 2015. Citado na página 54.

DESIMONE, C. The necessity of including the arts in STEM. *ISEC 2014 - 4th IEEE Integrated STEM Education Conference*, 2014. Citado na página 27.

DI LIETO, M. C.; INGUAGGIATO, E.; CASTRO, E.; CECCHI, F.; CIONI, G.; DELL'OMO, M.; LASCHI, C.; PECINI, C.; SANTERINI, G.; SGANDURRA, G.; DARIO, P. Educational Robotics intervention on Executive Functions in preschool children: A pilot study. *Computers in Human Behavior*, Elsevier B.V., v. 71, p. 16–23, 2017. ISSN 07475632. Citado na página 36.

DIAKOPOULOS, D.; KAPUR, A. Hiduino: A firmware for building driverless usb-midi devices using the arduino microcontroller. In: CITESEER. *NIME*. [S.l.], 2011. p. 405–408. Citado 2 vezes nas páginas 39 e 72.

DIAS, R.; SILVA, J.; AZEVEDO, J. L.; CUNHA, B.; NEVES, A. J.; LAU, N. Improving the kicking accuracy in a soccer robot. *Proceedings of the ACM Symposium on Applied Computing*, v. 13-17-April, p. 295–300, 2015. Citado na página 49.

Equipe Descomplicando a Música. *Notas Musicais*. 2020. Disponível em: <<https://www.descomplicandoamusica.com/notas-musicais/>>. Citado na página 64.

FAN, X.; ESSL, G. Air Violin : A Body-centric Style Musical Instrument. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2013. Citado 2 vezes nas páginas 32 e 53.

FIGUEIREDO, M.; CIFREDO-CHACÓN, M.-Á.; GONÇALVES, V. Learning programming and electronics with augmented reality. In: SPRINGER. *International Conference on Universal Access in Human-Computer Interaction*. [S.l.], 2016. p. 57–64. Citado na página 25.

FLØ, A. B.; WILMERS, H. Doppelgänger : A solenoid-based large-scale sound installation. *NIME 2015 Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 61–64, 2015. Citado na página 54.

FONSECA, S.; GONZÁLEZ, S.; RODRÍGUEZ, B.; SEDA, M.; BOBONIS, J.; CANELA, A.; ROLÓN, C.; BOSQUE, J.; MÉNDEZ, L.; CARTAGENA, A.; FUENTES, T.; SANTIAGO, N. G.; NIEVES, A. Educational Nanotechnology Video Game to Inspire Middle and High School Students to Pursue STEM Related Professional Careers. In: *2018 IEEE Frontiers in Education Conference (FIE)*. [S.l.: s.n.], 2018. p. 1–5. ISBN 9781538611746. Citado na página 26.

FÖRSTER, E.-C.; FÖRSTER, K.-T.; LÖWE, T. Teaching programming skills in primary school mathematics classes: An evaluation using game programming. In: IEEE. *2018 IEEE Global Engineering Education Conference (EDUCON)*. [S.l.], 2018. p. 1504–1513. Citado na página 23.

GARNELI, V.; GIANNAKOS, M. N.; CHORIANOPOULOS, K. Computing education in k-12 schools: A review of the literature. In: IEEE. *2015 IEEE Global Engineering Education Conference (EDUCON)*. [S.l.], 2015. p. 543–551. Citado na página 23.

GIBB, A. M. New media art, design, and the arduino microcontroller: A malleable tool. *Diss. Pratt Institute*, 2010. Citado na página 71.

GOMES, A. J.; HENRIQUES, J.; MENDES, A. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *EFT: Educação, Formação & Tecnologias*, Educom-Associação Portuguesa de Telemática Educativa, v. 1, n. 1, p. 93–103, 2008. Citado na página 23.

GREHER, G.; HEINES, J. Interdisciplinary Collaboration : Connecting Computer Science and Music Students to the Benefit of Both. *Assoc. for Technology in Music Instruction*. Atlanta, GA, p. 1–4, 2008. Citado na página 48.

GUPTA, N.; TEJOVANTH, N.; MURTHY, P. Learning by creating: Interactive programming for indian high schools. In: IEEE. *2012 IEEE International Conference on Technology Enhanced Education (ICTEE)*. [S.l.], 2012. p. 1–3. Citado na página 25.

GUREVICH, M. Distributed Control in a Mechatronic Musical Instrument. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 487–490, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_449.pdf>. Citado na página 54.

HARRIMAN, J. Pd Poems and Teaching Tools. *NIME 2015 Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 331–334, 2015. Citado 2 vezes nas páginas 29 e 49.

HARRIMAN, J. Start ‘ em Young : Digital Music Instruments for Education. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 6–9, 2015. Citado 4 vezes nas páginas 24, 49, 50 e 54.

HASHIMOTO, F. A. d. A. et al. Análise musical de "estudo para instrumentos de percussão", 1953, m. camargo guarneri; primeira peça escrita somente para instrumentos de percussão no Brasil. 2003. Citado na página 65.

HATTWICK, I.; WANDERLEY, M. M. Interactive Lighting in the Pearl: Considerations and Implementation. *Proceedings of the Conference on New Interfaces for Musical Expression*, p. 201–204, 2015. Citado na página 54.

HAYES, T. Neurohedron: A Nonlinear Sequencer Interface. *NIME 2010 Proceedings of the International Conference on New Interfaces for Musical Expression*, n. June, p. 23–25, 2010. Citado 2 vezes nas páginas 34 e 53.

HEINES, J. M.; GREHER, G. R.; KUHN, S. Music performamatics: Interdisciplinary interaction. *SIGCSE Bulletin Inroads*, v. 41, n. 1, p. 478–482, 2009. ISSN 10963936. Citado na página 48.

HENRIQUES, T. Sonik spring. *Conference on Human Factors in Computing Systems - Proceedings*, p. 1019–1022, 2012. Citado na página 53.

HERNANDEZ, C. C.; SILVA, L.; Alencar Segura, R.; SCHIMIGUEL, J.; Paradelas Ledón, M. F.; Mendes Bezerra, L. N.; Frango Silveira, I. Teaching Programming Principles through a Game Engine. *CLEI Electronic Journal*, v. 13, n. 2, p. 1–8, aug 2010. ISSN 0717-5000. Disponível em: <<http://192.168.10.19:8080/cleiej/index.php/cleiej/article/view/197>>. Citado na página 23.

HONG, D. R.; YEO, W. S. Laptop: Laptop Computer as a Musical Instrument using Audio Feedback. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 233–236, 2013. Disponível em: <http://www.nime.org/proceedings/2013/nime2013_137.pdf>. Citado na página 53.

INSTRUCTABLES. *Search: Digital Music Instruments*. 2020. Disponível em: <<https://www.instructables.com/circuits/howto/digital+musical+instrument/>>. Citado na página 34.

IOANNOU, M.; BRATITSIS, T. Utilizing Sphero for a speed related STEM activity in Kindergarten. *Hellenic Conference on Innovating STEM Education*, 2016. Citado na página 25.

JACKIE; CHUI, Y. T.; MARAFA, M.; SAMSON; YOUNG, K. F. SoloTouch: A Capacitive Touch Controller with Lick-based Note Selector. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 389–393, 2013. Disponível em: <http://nime.org/proceedings/2013/nime2013_130.pdf>. Citado na página 53.

JAKOBSEN, K. B.; WINGE, J.; PETERSEN, M. G.; STOUGAARD, J.; GROENBAEK, J. E.; RASMUSSEN, M. K. Hitmachine: Collective Musical Expressivity for Novices. *Proceedings of the International Conference on New Interfaces for Musical Expression*, v. 16, p. 241–246, 2016. Disponível em: <http://www.nime.org/proceedings/2016/nime2016_paper0048.pdf>. Citado na página 55.

JENKINS, L.; TRAIL, S.; TZANETAKIS, G.; DRIESSEN, P.; PAGE, W. An Easily Removable, wireless Optical Sensing System (EROSS) for the Trumpet. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 352–357, 2013. Disponível em: <http://nime.org/proceedings/2013/nime2013_261.pdf>. Citado na página 53.

JEON, M.; FAKHRHOSSEINI, M.; BARNES, J.; DUFORD, Z.; ZHANG, R.; RYAN, J.; VASEY, E. Making Live Theatre with Multiple Robots as Actors. *Human-Robot Interaction*, n. 1, p. 445–446, 2016. Disponível em: <<http://dl.acm.org.ezproxy1.lib.asu.edu/citation.cfm?id=2906919>>. Citado na página 49.

JOHNSTON, B.; THRUSH, H. D.; KAPUR, A.; MURPHY, J.; MOLETA, T. Polus: The Design and Development of a New, Mechanically Bowed String Instrument Ensemble. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 557–560, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_355.pdf>. Citado na página 54.

JORDÁ, S. Digital Lutherie Crafting musical computers for new musics' performance and improvisation. *Departament de Tecnologia*, v. 26, n. 3, p. 531, 2005. Disponível em: <<http://dialnet.unirioja.es/servlet/tesis?codigo=19509>>. Citado na página 31.

JUNG, S. E.; WON, E. S. Systematic review of research trends in robotics education for young children. *Sustainability (Switzerland)*, v. 10, n. 4, p. 905, 2018. ISSN 20711050. Citado 2 vezes nas páginas 25 e 26.

JUNIOR, L. A.; NETO, O. T.; HERNANDEZ, M. F.; MARTINS, P. S.; ROGER, L. L.; GUERRA, F. A. A Low-Cost and Simple Arduino-Based Educational Robotics Kit. *Journal of Selected Areas in Robotics and Control (JSRC)*, v. 3, n. 12, p. 12, 2013. Disponível em: <<http://blog.minibloq.org/>>. Citado na página 36.

KALELIOĞLU, F. A new way of teaching programming skills to k-12 students: Code. org. *Computers in Human Behavior*, Elsevier, v. 52, p. 200–210, 2015. Citado na página 23.

KALELIOĞLU, F.; GÜLBAHAR, Y. The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, Vilnius University Institute of Mathematics and Informatics, Lithuanian . . . , v. 13, n. 1, p. 33–50, 2014. Citado na página 23.

KANIKA; CHAKRAVERTY, S.; CHAKRABORTY, P. Tools and Techniques for Teaching Computer Programming: A Review. *Journal of Educational Technology*

- Systems*, v. 49, n. 2, p. 170–198, dec 2020. ISSN 0047-2395. Disponível em: <<http://journals.sagepub.com/doi/10.1177/0047239520926971>>. Citado na página 23.
- KAPUR, A.; DARLING, M.; MURPHY, J.; HOCHENBAUM, J.; DIAKOPOULOS, D.; TRIMPIN, T. The KarmetiK NotomotoN : A New Breed of Musical Robot for Teaching and Performance. *Proceedings of the International Conference on New Interfaces for Musical Expression*, n. June, p. 228–231, 2011. ISSN 2220-4806. Disponível em: <http://www.nime.org/proceedings/2011/nime2011_228.pdf>. Citado na página 53.
- KARIM, M. E.; LEMAIGNAN, S.; MONDADA, F. A review: Can robots reshape K-12 STEM education? *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts, ARSO*, v. 2016-March, 2016. ISSN 21627576. Citado na página 26.
- KARVINEN, K.; KARVINEN, T. *Primeiros Passos com Sensores: Perceba o mundo usando eletrônica, Arduino e Raspberry Pi*. Novatec Editora, 2014. ISBN 9788575224021. Disponível em: <<https://books.google.com.br/books?id=ew-sBAAAQBAJ>>. Citado na página 70.
- KIM, C.; KIM, D.; YUAN, J.; HILL, R. B.; DOSHI, P.; THAI, C. N. Robotics to promote elementary education pre-service teachers' STEM engagement, learning, and teaching. *Computers and Education*, Elsevier Ltd, v. 91, n. December, p. 14–31, 2015. ISSN 03601315. Disponível em: <<http://dx.doi.org/10.1016/j.compedu.2015.08.005>>. Citado na página 26.
- KLEINBERGER, R. PAMDI Music Box. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 2–3, 2013. Citado na página 53.
- KO, S.; SWAIM, H.; SANGHAVI, H.; DONG, J.; NADRI, C.; JEON, M. Robot-theater programs for different age groups to promote steam education and robotics research. *ACM/IEEE International Conference on Human-Robot Interaction*, v. 3378353, p. 299–301, 2020. ISSN 21672148. Citado 2 vezes nas páginas 26 e 49.
- KRISHNAN, A. B.; ASWATH, S.; UDUPA, G. Real time vision based soccer playing humanoid robotic platform. *ACM International Conference Proceeding Series*, v. 10-11-Octo, 2014. Citado na página 49.
- KYNIGOS, C.; GRIZIOTI, M.; NIKITOPOULOU, S. RobIn: A Half-baked Robot for Electronics in a STEM Context. In: *Proceedings of the 2017 Conference on Interaction Design and Children*. New York, NY, USA: ACM, 2017. p. 521–526. ISBN 9781450349215. Disponível em: <<https://dl.acm.org/doi/10.1145/3078072.3084319>>. Citado na página 25.
- LAMB, R.; ROBERTSON, A. Seaboard : a New Piano Keyboard-related Interface Combining Discrete and Continuous Control. *Proceedings of the International Conference on New Interfaces for Musical Expression*, n. June, p. 503–506, 2011. ISSN 2220-4806. Disponível em: <http://www.nime.org/proceedings/2011/nime2011_503.pdf>. Citado 3 vezes nas páginas 32, 33 e 53.
- LAU, W. W.; NGAI, G.; CHAN, S. C.; CHEUNG, J. C. Learning programming through fashion and design: A pilot summer course in wearable computing for middle school students. *SIGCSE Bulletin Inroads*, v. 41, n. 1, p. 504–508, 2009. ISSN 10963936. Citado na página 49.

- LAURITSEN, C. *Drum Lessons: Teach Yourself How to Play Drums (Free Audio Available)*. [S.l.]: LearnToPlayMusic.com, 2014. (Progressive). ISBN 9789825322382. Citado na página 65.
- LEEuw, H. The electrumptet , additions and revisions. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 258–259, 2012. Citado na página 53.
- LEVINSON, C. TED: a Tangible Electrophonic Drumstick. *Proceedings of the 2011 Conference on New Interfaces for Musical Expression (NIME 2011)*, 2011. Citado 3 vezes nas páginas 32, 33 e 53.
- LIN, Y.-T.; WANG, M.-T.; WU, C.-C. Design and implementation of interdisciplinary stem instruction: Teaching programming by computational physics. *The Asia-Pacific Education Researcher*, Springer, v. 28, n. 1, p. 77–91, 2019. Citado na página 23.
- LONG, J. The Robotic Taishogoto: A New Plug 'n Play Desktop Performance Instrument. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 479–482, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_313.pdf>. Citado na página 54.
- LOY, G. Musicians make a standard: the midi phenomenon. *Computer Music Journal*, JSTOR, v. 9, n. 4, p. 8–26, 1985. Citado 2 vezes nas páginas 39 e 73.
- LUDI, S.; BERNSTEIN, D.; MUTCH-JONES, K. Enhanced robotics! Improving building and programming learning experiences for students with visual impairments. *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, v. 2018-January, p. 372–377, 2018. Citado na página 38.
- MAEDA, J. STEM + Art = STEAM. *Steam*, v. 1, n. 1, p. 1–3, 2013. ISSN 2327-2074. Citado na página 26.
- MAGERKO, B.; FREEMAN, J.; MCKLIN, T.; REILLY, M.; LIVINGSTON, E.; MCCOID, S.; CREWS-BROWN, A. EarSketch: A STEAM-based approach for underrepresented populations in high school computer science education. *ACM Transactions on Computing Education*, v. 16, n. 4, 2016. ISSN 19466226. Citado na página 26.
- MAGUN, I.; GERRERO, M. F. E.; JIMÉNEZ, E. J. Gestural Control Using a Laser Harp. *Communications*, n. Conielectcomp, p. 3–7, 2006. Citado 2 vezes nas páginas 68 e 69.
- MAMEDES, C.; RODRIGUES, M.; WANDERLEY, M.; MANZOLLI, J.; GARCIA, D. H. L.; FERREIRA-LOPES, P. Composing for DMIs - Entoa, Music for Intonaspacio. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 509—512, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_411.pdf>. Citado na página 54.
- MARLEY, W.; WARD, N. Gestroviser : Toward Collaborative Agency in Digital Musical Instruments . *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 140–143, 2015. Citado na página 54.

MARSHALL, M.; WANDERLEY, M. Examining the effects of embedded vibrotactile feedback on the feel of a digital musical instrument. *Proceedings of the 11th International Conference on New Interfaces for Musical Expression (NIME11)*, n. June, p. 399–404, 2011. Citado na página 53.

MARSHALL, M. T.; HARTSHORN, M.; WANDERLEY, M. M.; LEVITIN, D. J. Sensor Choice for Parameter Modulations in Digital Musical Instruments: Empirical Evidence from Pitch Modulation. *Journal of New Music Research*, v. 38, n. 3, p. 241–253, 2009. ISSN 0929-8215. Citado na página 31.

MATARIĆ, M. J.; KOENIG, N.; FEIL-SEIFER, D. Materials for enabling hands-on robotics and STEM education. *AAAI Spring Symposium - Technical Report*, SS-07-09, p. 99–102, 2007. Citado na página 26.

MCCLOSKEY, B.; BT, L. D.; BRIDGES, B.; BT, L. D.; LYONS, F.; BT, L. D. Accessibility and dimensionality : enhanced real time creative independence for digital musicians with quadriplegic cerebral palsy. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 24–27, 2015. Citado na página 54.

MCLAMORE, A. Musical Theater. *Musical Theater*, IEEE, p. 366, 2016. Citado na página 49.

MCPHERSON, A. TouchKeys: Capacitive Multi-touch Sensing on a Physical Keyboard. *NIME 2012 Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 419–432, 2012. Citado na página 53.

MCPHERSON, A. P.; KIM, Y. Augmenting the Acoustic Piano with Electromagnetic String Actuation and Continuous Key Position Sensing. *Signal Processing*, n. Nime, p. 217–222, 2010. Disponível em: <http://www.nime.org/proceedings/2010/nime2010_217.pdf>. Citado na página 53.

MEACHAM, A.; KANNAN, S.; WANG, G. The Laptop Accordion. *NIME 2016 Proceedings of the International Conference on New Interfaces for Musical Expression*, v. 16, n. July 2016, p. 236–240, 2016. Disponível em: <http://www.nime.org/proceedings/2016/nime2016_paper0047.pdf>. Citado na página 54.

MERKOURIS, A.; CHORIANOPOULOS, K.; KAMEAS, A. Teaching programming in secondary education through embodied computing platforms: Robotics and wearables. *ACM Transactions on Computing Education (TOCE)*, ACM New York, NY, USA, v. 17, n. 2, p. 1–22, 2017. Citado na página 23.

MERKOURIS, A.; CHORIANOPOULOS, K.; KAMEAS, A. Teaching programming in secondary education through embodied computing platforms: Robotics and wearables. *ACM Transactions on Computing Education*, v. 17, n. 2, 2017. ISSN 19466226. Citado na página 23.

MEYERS, A.; COLE, M. C.; KORTH, E.; PLUTA, S. Musicomputation: Teaching computer science to teenage musicians. *C and C 2009 - Proceedings of the 2009 ACM SIGCHI Conference on Creativity and Cognition*, p. 29–38, 2009. Citado na página 49.

- MICHON, R.; SMITH, J. O.; WRIGHT, M.; CHAFE, C. Augmenting the iPad: the BladeAxe. *Proceedings of the International Conference on New Interfaces for Musical Expression*, v. 16, p. 247–252, 2016. Disponível em: <http://www.nime.org/proceedings/2016/nime2016_paper0049.pdf>. Citado na página 55.
- MIRANDA, E. R.; WANDERLEY, M. M. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. A-R Editions, 2006. (Computer Music and Digital Audio Series). ISBN 9780895795854. Disponível em: <<https://books.google.com.br/books?id=CGEwXZ7hcIoC>>. Citado 7 vezes nas páginas 24, 31, 32, 34, 35, 55 e 70.
- MISRA, A.; BLANK, D.; KUMAR, D. A music context for teaching introductory computing. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, v. 3, n. September, p. 248–252, 2009. Citado na página 49.
- MIYAMA, C. Peacock: A non-haptic 3D performance interface. *International Computer Music Conference, ICMC 2010*, p. 443–445, 2010. Citado na página 53.
- MONK, S. *Make: Action: Movement, Light, and Sound with Arduino and Raspberry Pi*. Make Community, LLC, 2016. ISBN 9781457187759. Disponível em: <<https://books.google.com.br/books?id=g2eDCwAAQBAJ>>. Citado na página 72.
- MOURÃO, S. L.; MORAIS, R. G. de. A LUTERIA COMO PROPOSTA PEDAGÓGICA PARA O ENSINO DE MÚSICA. p. 242–262, 2016. Citado na página 31.
- MYLLYKOSKI, M.; TUURI, K. Prototyping hand-based wearable music education technology. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 182–183, 2015. Citado na página 49.
- NAGASHIMA, Y. Untouchable instrument"peller-min". In: *NIME*. [S.l.: s.n.], 2010. p. 387–390. Citado na página 53.
- NAGASHIMA, Y. Multi Rubbing Tactile Instrument. *NIME 2016 Proceedings of the International Conference on New Interfaces for Musical Expression*, v. 16, p. 168–169, 2016. ISSN 00207608. Disponível em: <http://www.nime.org/proceedings/2016/nime2016_paper0035.pdf>. Citado na página 54.
- NAKANISHI, Y.; MATSUMURA, S.; ARAKAWA, C. POWDER BOX: An Interactive Device with Sensor Based Replaceable Interface For Musical Session. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2013. Disponível em: <<http://nime2013.kaist.ac.kr/>>. Citado na página 53.
- NAM, S. Musical Poi (mPoi). *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 148–151, 2013. Disponível em: <<http://nime2013.kaist.ac.kr/>>. Citado na página 54.
- NANO, A. *Overview placa Arduino Nano*. 2020. Disponível em: <<https://store-usa.arduino.cc/collections/boards/products/arduino-nano>>. Citado na página 71.
- NASH, C. The 'E' in QWERTY: Musical Expression with Old Computer Interfaces. *Proceedings of the International Conference on New Interfaces for Musical Expression*, v. 16, n. July, p. 224–229, 2016. Disponível em: <http://www.nime.org/proceedings/2016/nime2016_paper0045.pdf>. Citado na página 54.

- NICOLETE, P. C.; da Silva Cristiano, M. A.; SILVA, J. B. da; BILESSIMO, S. M. S.; SIMÃO, J. P. S.; BENAVIDES, K. D. R.; da Mota Alves, J. B. Mathematics and technological integration in the Brazilian basic education as motivation to STEM. In: *Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM '15*. New York, New York, USA: ACM Press, 2015. p. 591–598. ISBN 9781450334426. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2808580.2808671>>. Citado na página 26.
- NORMARK, C. J.; PARNES, P.; EK, R.; ANDERSSON, H. The extended clarinet. *NIME 2016 Proceedings of the International Conference on New Interfaces for Musical Expression*, v. 16, p. 162–167, 2016. Disponível em: <http://www.nime.org/proceedings/2016/nime2016_paper0034.pdf>. Citado 2 vezes nas páginas 32 e 54.
- OVERHOLT, D.; GELINECK, S. Design & Evaluation of an Accessible Hybrid Violin Platform. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 122–125, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_470.pdf>. Citado na página 54.
- PAPERT, S. LOGO: Computadores e Educação.: Brasiliense. *São Paulo (SP)*, 1985. Citado na página 37.
- PATSKO, L. Tutorial – Aplicações, Funcionamento e Utilização de Sensores. *Maxwell Bohr: Instrumentação eletrônica*, p. 84, 2006. Citado na página 40.
- PECH, J.; NOVÁK, M. Use Arduino and Micro : bit as Teaching Platform for the Education Programming and Electronics on the STEM Basis. *2020 V International Conference on Information Technologies in Engineering Education (Inforino)*, p. 1–4, 2020. Citado na página 23.
- PEPPLER, K. STEAM-powered computing education: Using E-textiles to integrate the arts and STEM. *Computer*, v. 46, n. 9, p. 38–43, 2013. ISSN 00189162. Citado 2 vezes nas páginas 26 e 49.
- PERIGNAT, E.; KATZ-BUONINCONTRO, J. STEAM in practice and research: An integrative literature review. *Thinking Skills and Creativity*, Elsevier Ltd, v. 31, p. 31–43, 2019. ISSN 18711871. Disponível em: <<https://doi.org/10.1016/j.tsc.2018.10.002>>. Citado na página 26.
- PETERS, G.; RUECKERT, T.; SERUGA, J. A Framework Supporting Literacy in Mathematics and Software Programming - Addressing Some Challenges in STEM Education. In: *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*. SCITEPRESS - Science and Technology Publications, 2018. v. 1, n. Iceis 2018, p. 497–506. ISBN 978-989-758-298-1. Disponível em: <<http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006629304970506>>. Citado na página 26.
- PHILHARMONIA. *Instrument: Harp*. 2020. Disponível em: <<https://philharmonia.co.uk/resources/instruments/viola-2/>>. Citado na página 68.
- PIEPENBRINK, A.; DRIVE, C.; BARBARA, S.; WRIGHT, M.; ART, P. The Bistable Resonator Cymbal : An Actuated Acoustic Instrument Displaying Physical Audio Effects.

Proceedings of the International Conference on New Interfaces for Musical Expression, p. 227–230, 2015. Citado na página 54.

PIZZATO, M. S.; HENTSCHKE, L. Motivação para aprender música na escola. *Revista da ABEM*, v. 18, n. 23, 2014. Citado na página 28.

PON, A.; WANG, J.; RADFORD, L.; CARPENDALE, S. Womba: A Musical Instrument for an Unborn Child. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 87–90, 2015. Disponível em: <http://www.nime.org/proceedings/2015/nime2015_304.pdf>. Citado na página 54.

PORTNER, H. CHIMAERA The Poly-Magneto-Phonic Theremin An Expressive Touch-Less Hall-Effect Sensor Array. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 501–504, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_397.pdf>. Citado na página 54.

QIU, K.; BUECHLEY, L.; BAAFI, E.; DUBOW, W. A curriculum for teaching computer science through computational textiles. *ACM International Conference Proceeding Series*, n. March, p. 20–27, 2013. Citado na página 26.

RAMKISSOON, I. The Bass Sleeve: A real-time multimedia gestural controller for augmented electric bass performance. *Proceedings of the International Conference on New Interfaces for Musical Expression*, n. June, p. 224–227, 2011. ISSN 22204806. Disponível em: <<http://www.nime2011.org/proceedings/papers/G04-Ramkissoon.pdf>>. Citado na página 53.

RAO, D. M. Music maker: Using music to introduce coding and concurrency to young learners. *Proceedings - Frontiers in Education Conference, FIE*, v. 2017-Octob, p. 1–6, 2017. ISSN 15394565. Citado 2 vezes nas páginas 24 e 38.

RAPKIEWICZ, C. E.; FALKEMBACH, G.; SEIXAS, L.; ROSA, N. dos S.; CUNHA, V. V. da; KLEMMANN, M. Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. *RENOTE-Revista Novas Tecnologias na Educação*, v. 4, n. 2, 2006. Citado na página 23.

REAMER, A. C.; IVY, J. S.; VILA-PARRISH, A. R.; YOUNG, R. E. Understanding the evolution of mathematics performance in primary education and the implications for STEM learning: A Markovian approach. *Computers in Human Behavior*, Elsevier Ltd, v. 47, p. 4–17, jun 2015. ISSN 07475632. Disponível em: <<http://dx.doi.org/10.1016/j.chb.2014.09.037https://linkinghub.elsevier.com/retrieve/pii/S0747563214004956>>. Citado na página 26.

RIEGER, A.; TOPEL, S. Driftwood: Redefining Sound Sculpture Controllers. *NIME 2016 Proceedings of the International Conference on New Interfaces for Musical Expression*, v. 16, p. 158–159, 2016. Disponível em: <http://www.nime.org/proceedings/2016/nime2016_paper0032.pdf>. Citado na página 54.

RIGLER, J.; SELDESS, Z. The music Cre8tor: An interactive system for musical exploration and education. *Proceedings of the 7th International Conference on New Interfaces for Musical Expression, NIME '07*, p. 415–416, 2007. Citado na página 49.

- ROSAS, F. W.; BEHAR, P. A. Construção e uso de instrumentos musicais digitais no contexto educacional: um olhar a partir da dimensão socioafetiva. *Currículo sem Fronteiras*, v. 18, n. 3, p. 1108–1128, 2018. Citado 3 vezes nas páginas 29, 49 e 50.
- ROTHMAN, P. The Ghost: an open-source , user programmable MIDI performance controller. *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, n. Nime, p. 431–435, 2010. Citado 4 vezes nas páginas 32, 35, 36 e 53.
- RUTHMANN, A.; HEINES, J. M.; GREHER, G. R.; LAIDLER, P.; SAULTERS, C. Teaching computational thinking through musical live coding in Scratch. *SIGCSE'10 - Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, p. 351–355, 2010. Citado na página 48.
- SAELI, M.; PERRENET, J.; JOCHEMS, W. M.; ZWANEVELD, B. Teaching programming in secondary school: A pedagogical content knowledge perspective. *Informatics in education*, Vilnius University Institute of Mathematics and Informatics, Lithuanian . . . , v. 10, n. 1, p. 73–88, 2011. Citado na página 23.
- SÁEZ-LÓPEZ, J.-M.; ROMÁN-GONZÁLEZ, M.; VÁZQUEZ-CANO, E. Visual programming languages integrated across the curriculum in elementary school: A two year case study using “scratch” in five schools. *Computers & Education*, Elsevier, v. 97, p. 129–141, 2016. Citado na página 28.
- SANDERS, M. Stem, stem education, stemmania. *The Technology Teacher*, p. 20–27, 2009. Disponível em: <<https://vtechworks.lib.vt.edu/bitstream/handle/10919/51616/STEMmania.pdf?sequence=1&isAllowed>>. Citado na página 25.
- SANTANA, B. L.; FIGUERÊDO, J. S. L.; BITTENCOURT, R. A. Motivação de estudantes non-majors em uma disciplina de programação. In: SBC. *Anais do XXV Workshop sobre Educação em Computação*. [S.l.], 2017. Citado na página 23.
- SAWYER, B.; FORSYTH, J.; O’CONNOR, T.; BORTZ, B.; FINN, T.; BAUM, L.; BUKVIC, I. I.; KNAPP, B.; WEBSTER, D. Form, function and performances in a musical instrument MAKERS camp. *SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, p. 669–674, 2013. Citado 4 vezes nas páginas 29, 49, 50 e 68.
- SCAICO, P. D.; LIMA, A. A. de; AZEVEDO, S.; SILVA, J. B. B. da; RAPOSO, E. H.; ALENCAR, Y.; MENDES, J. P.; SCAICO, A. et al. Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch. *Revista Brasileira de Informática na Educação*, v. 21, n. 02, p. 92, 2013. Citado na página 23.
- SCHACHER, J. C. The Quarterstaff, a Gestural Sensor Instrument. *Proceedings of the International Conference on New Interfaces for Musical Expression 2013 (NIME 2013)*, p. 535–540, 2013. Disponível em: <<http://nime2013.kaist.ac.kr/>>. Citado na página 54.
- SCHIESSER, S.; SCHACHER, J. C. SABRe: The Augmented Bass Clarinet. *International Conference on New Interface for Musical Expression*, 2012. Disponível em: <http://www.eecs.umich.edu/nime2012/Proceedings/papers/193_Final_Manuscript.pdf>. Citado na página 53.

- SCHUBERT, T. W.; D'AUSILIO, A.; CANTO, R. Using arduino microcontroller boards to measure response latencies. *Behavior research methods*, Springer, v. 45, n. 4, p. 1332–1346, 2013. Citado na página 39.
- SELLO, J. T. The Hexenkessel: A Hybrid Musical Instrument for Multimedia Performances. *NIME 2016 Proceedings of the International Conference on New Interfaces for Musical Expression*, v. 16, p. 122–131, 2016. Disponível em: <http://www.nime.org/proceedings/2016/nime2016_paper0025.pdf>. Citado na página 54.
- SERAFINI, G. Teaching programming at primary schools: visions, experiences, and long-term research prospects. In: SPRINGER. *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*. [S.l.], 2011. p. 143–154. Citado na página 23.
- SHEIN, E. *Should everybody learn to code?* [S.l.]: ACM New York, NY, USA, 2014. Citado na página 23.
- SILLA, C. N.; PRZYBYSZ, A. L.; LEAL, W. V. Music education meets computer science and engineering education. In: *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2016. v. 2016-Novem, p. 1–7. ISBN 978-1-5090-1790-4. ISSN 15394565. Disponível em: <<http://ieeexplore.ieee.org/document/7757413/>>. Citado 8 vezes nas páginas 15, 24, 27, 28, 47, 48, 49 e 50.
- SILLA, C. N.; PRZYBYSZ, A. L.; RIVOLLI, A.; GIMENEZ, T.; BARROSO, C.; MACHADO, J. Girls, Music and Computer Science. In: *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2018. p. 1–6. ISBN 978-1-5386-1174-6. Disponível em: <<https://ieeexplore.ieee.org/document/8658724/>>. Citado 7 vezes nas páginas 15, 24, 27, 28, 47, 48 e 49.
- SILVA, C. V. da; ALMEIDA, C. M. G. de. Educação musical e inclusão: um estudo sobre as práticas de professores de música no ensino fundamental. *Revista Educação, Artes e Inclusão*, v. 14, n. 4, p. 078–100, 2018. Citado na página 28.
- SILVA, T. S. C. da; MELO, J. C. B. de; TEDESCO, P. C. d. A. R. Um modelo para promover o engajamento estudantil no aprendizado de programação utilizando gamification. *Revista Brasileira de Informática na Educação*, v. 26, n. 03, p. 120, 2018. Citado na página 23.
- SINCLAIR, I. R. *Sensors and Transducers Third edition*. Elsevier, 2001. 319 p. ISBN 0 7506 4932 1. Disponível em: <<http://senofficial.yolasite.com/resources/sensorsandtransducers.pdf>>. Citado na página 40.
- SIVA, S.; IM, T.; MCKLIN, T.; FREEMAN, J.; MAGERKO, B. Using music to engage students in an introductory undergraduate programming course for non-Majors. *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, v. 2018-Janua, n. May 2020, p. 975–980, 2018. Citado 2 vezes nas páginas 24 e 38.
- SNYDER, J.; RYAN, D. The Birl: An Electronic Wind Instrument Based on an Artificial Neural Network Parameter Mapping Structure. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 585–588, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_540.pdf>. Citado na página 54.

STUBBS, K. N.; YANCO, H. A. STREAM: A Workshop on the Use of Robotics in K–12 STEM Education. *IEEE Robotics & Automation Magazine*, v. 16, n. June, p. 17–19, 2009. Citado na página 26.

SULLIVAN, A.; BERS, M. U. Dancing robots: integrating art, music, and robotics in Singapore’s early childhood centers. *International Journal of Technology and Design Education*, Springer Netherlands, v. 28, n. 2, p. 325–346, jun 2018. ISSN 0957-7572. Disponível em: <<http://link.springer.com/10.1007/s10798-017-9397-0>>. Citado 2 vezes nas páginas 24 e 26.

SWEETWATER. *Electronic Drums Buying Guide*. 2020. Disponível em: <<https://www.sweetwater.com/insync/electronic-drums-buying-guide/>>. Citado na página 66.

TAYLOR, S.; HOOK, J. FerroSynth: A Ferromagnetic Music Interface. *NIME 2010 Proceedings of the International Conference on New Interfaces for Musical Expression*, n. Nime, p. 463–466, 2010. Citado na página 53.

TESSMANN, R. *Como Tocar Teclado Guia Definitivo Para Iniciantes*. 2018. Disponível em: <<https://aprendateclado.com/como-tocar-teclado-guia-definitivo/>>. Citado na página 64.

THOMAZ, S.; AGLAÉ, A.; FERNANDES, C.; PITTA, R.; AZEVEDO, S.; BURLAMAQUI, A.; SILVA, A.; GONÇALVES, L. M. RoboEduc: A Pedagogical Tool to support Educational Robotics. *Proceedings - Frontiers in Education Conference, FIE*, p. 1–6, 2009. ISSN 15394565. Citado na página 37.

VAN DER MAREN, J.-M. *Méthodes de recherche pour l ’ éducation. Éducation et formation. Fondements*. 2. ed. Montréal: Les Presses de l’Université de Montréal; DeBoeck Université, 2004. 231–238 p. Citado 5 vezes nas páginas 29, 43, 44, 46 e 201.

VAN DER TORREN, P. T. Striso, a Compact Expressive Instrument Based on a New Isomorphic Note Layout. *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 615–620, 2014. Disponível em: <http://www.nime.org/proceedings/2014/nime2014_442.pdf>. Citado 3 vezes nas páginas 35, 36 e 54.

WANG, T.-C.; MEI, W.-H.; LIN, S.-L.; CHIU, S.-K.; LIN, J. M.-C. Teaching programming concepts to high school students with alice. In: IEEE. *2009 39th IEEE Frontiers in Education Conference*. [S.l.], 2009. p. 1–6. Citado na página 23.

WATSON, A, D.; WATSON, G, H. Transitioning stem to steam: Reformation of engineering education. *Journal for Quality and Participation*, v. 36, n. 3, p. 1–5, 2013. Citado na página 24.

WILEY, M.; KAPUR, A. Multi-Laser Gestural Interface — Solutions for Cost-Effective and Open Source Controllers. *NIME 2009 Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 43–44, 2009. Citado 2 vezes nas páginas 68 e 69.

YAKMAN, G. STEAM EDUCATION an overview of creating a model of integrative education. *Foreign Affairs*, v. 91, n. 5, p. 28, 2008. ISSN 00157120. Citado 2 vezes nas páginas 24 e 26.

YAMAHA, C. *Choosing a Piano Acoustic or Digital?* 2020. Disponível em: <https://www.yamaha.com/en/musical_instrument_guide/piano/selection/selection002.html>. Citado na página 63.

YAMAHA, C. *The Structure of the Drum - The drum kit-a collection of percussion instruments.* 2020. Disponível em: <https://www.yamaha.com/en/musical_instrument_guide/drums/mechanism/>. Citado na página 65.

YOON, M. B.; BAEK, J. E. Development and application of the STEAM education program based on the soccer robot for elementary students. *International Journal of Mobile and Blended Learning*, v. 10, n. 3, p. 11–22, 2018. ISSN 19418655. Citado na página 49.

APÊNDICE A – Termo de Consentimento Livre e Esclarecido - TCLE

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO - TCLE

Você está sendo convidado(a) como voluntário(a) a participar do estudo Construção de Instrumentos Musicais Digitais para Educação, que tem como objetivo desenvolver instrumentos musicais digitais (IMDs) de baixo custo que possam ser utilizados em abordagens pedagógicas para o ensino integrado de programação, música e eletrônica. Acreditamos que esta pesquisa seja importante porque a maioria das escolas possuem limitações orçamentárias no ensino de música, geralmente não possuem instrumentos musicais para os estudantes, em outros casos não possuem um instrumento por estudante. Além disso, é importante desenvolver nos jovens a motivação para que sejam desenvolvedores/criadores de tecnologias e não apenas usuários das mesmas.

PARTICIPAÇÃO NO ESTUDO

A sua participação no referido estudo será de participar das aulas do curso de extensão sobre construção de IMDs de baixo de custo, realizar atividades e responder aos questionários referentes aos conteúdos das aulas. Você participará de 2 aulas por semana durante 4 semanas com duração de aproximadamente 2 horas cada aula. Todas as aulas serão ministradas dentro da Pontifícia Universidade Católica do Paraná campus Curitiba, bloco 2 segundo andar no laboratório de informática 11. Os tópicos abordados em cada aula são:

- Aula 01 – Introdução sobre DMIs, Arduino e IDE de Programação do Arduino;
- Aula 02 – Gerando notas musicais com Arduino e Buzzer;
- Aula 03 – Melhorando Qualidade sonora utilizando fones de ouvido;
- Aula 04 – Utilizando sensores de entrada - Primeiro DMI;
- Aula 05 – Melhorando precisão dos botões com comandos IF;
- Aula 06 – Enviando comandos ao computador com protocolo MIDI;
- Aula 07 – Utilizando múltiplas teclas ao mesmo tempo;
- Aula 08 – Introdução as portas analógicas, construindo controle de efeitos;

- Aula 09 – Sensores Infravermelho, construindo Harpa Laser e Teremin;
- Aula 10 – Sensores Capacitivos, construindo teclados touch e finalização.

RISCOS E BENEFÍCIOS

Através deste Termo de Consentimento Livre e Esclarecido você está sendo alertado de que, da pesquisa a se realizar, pode esperar alguns benefícios, tais como: aprendizado ou aperfeiçoamento de lógica de programação, aprender sobre conceitos de eletrônica na prática, aprender como integrar as áreas de programação e eletrônica com a música através da criação de um instrumento musical digital, certificado de horas complementares. Bem como, também que é possível que aconteçam os seguintes desconfortos ou riscos em sua participação, tais como constrangimentos por não conseguir realizar atividades, cansaço devido ao tempo utilizado para realizar atividades, reconhecimento na divulgação dos resultados ainda que seja preservada sua identidade, desconfortos gerados devido a problemas técnicos como computador estragado, componentes queimados ou softwares com mal funcionamento. Para minimizar tais riscos, nós pesquisadores tomaremos as seguintes medidas: Para minimizar os riscos de constrangimento todos os procedimentos das atividades serão disponibilizados no material em forma de texto e imagens, caso o participante não consiga realizar uma atividade o pesquisador retomará o conteúdo das aulas e demonstrará de forma mais detalhada cada passo da atividade de forma individual. Para evitar riscos de cansaço dos participantes haverá um intervalo em cada aula, caso as atividades se tornem cansativas as aulas serão encerradas mais cedo e continuadas na próxima semana. Os dados da pesquisa serão utilizados para análise de resultados posteriormente e nenhuma informação pessoal será divulgada e qualquer informação referente aos participantes serão removidas em casos onde o participante se reconheça na divulgação dos resultados. Todos os equipamentos possuem sobressalentes e serão testados antes de cada aula para evitar atrasos, caso algum equipamento esteja com mal funcionamento, este será substituído por outro em perfeitas condições. Os participantes terão a liberdade de se retirar do curso a qualquer momento sem nenhum ônus.

SIGILO E PRIVACIDADE

Nós pesquisadores garantiremos a você que sua privacidade será respeitada, ou seja, seu nome ou qualquer outro dado ou elemento que possa, de qualquer forma, lhe identificar, será mantido em sigilo. Nós pesquisadores nos responsabilizaremos pela guarda e confidencialidade dos dados, bem como a não exposição dos dados de pesquisa.

AUTONOMIA

Nós lhe asseguramos assistência durante toda pesquisa, bem como garantiremos seu livre acesso a todas as informações e esclarecimentos adicionais sobre o estudo e suas consequências, enfim, tudo o que você queira saber antes, durante e depois de sua

participação. Também informamos que você pode se recusar a participar do estudo, ou retirar seu consentimento a qualquer momento, sem precisar justificar, e de, por desejar sair da pesquisa, não sofrerá qualquer prejuízo à assistência que vem recebendo.

RESSARCIMENTO E INDENIZAÇÃO

No entanto, caso tenha qualquer despesa decorrente da participação nesta pesquisa, tais como transporte, alimentação entre outros, bem como de seu acompanhante (se for o caso), haverá ressarcimento dos valores gastos na forma seguinte: mediante depósito em conta corrente. De igual maneira, caso ocorra algum dano decorrente de sua participação no estudo, você será devidamente indenizado, conforme determina a lei.

CONTATO

Os pesquisadores envolvidos com o referido projeto são Maicon Augusto Tibola da Pontifícia Universidade Católica do Paraná – PUCPR vinculado ao Programa de Pós-Graduação em Informática (PPGIa) e com eles você poderá manter contato pelos telefones (41) 99874-8329.

O Comitê de Ética em Pesquisa em Seres Humanos (CEP) é composto por um grupo de pessoas que estão trabalhando para garantir que seus direitos como participante de pesquisa sejam respeitados. Ele tem a obrigação de avaliar se a pesquisa foi planejada e se está sendo executada de forma ética. Se você achar que a pesquisa não está sendo realizada da forma como você imaginou ou que está sendo prejudicado de alguma forma, você pode entrar em contato com o Comitê de Ética em Pesquisa da PUCPR (CEP) pelo telefone (41) 3271-2103 entre segunda e sexta-feira das 08h00 às 17h30 ou pelo e-mail nep@pucpr.br.

DECLARAÇÃO

Declaro que li e entendi todas as informações presentes neste Termo de Consentimento Livre e Esclarecido e tive a oportunidade de discutir as informações deste termo. Todas as minhas perguntas foram respondidas e eu estou satisfeito com as respostas. Entendo que receberei uma via assinada e datada deste documento e que outra via assinada e datada será arquivada nos pelo pesquisador responsável do estudo.

Enfim, tendo sido orientado quanto ao teor de todo o aqui mencionado e compreendido a natureza e o objetivo do já referido estudo, manifesto meu livre consentimento em participar, estando totalmente ciente de que não há nenhum valor econômico, a receber ou a pagar, por minha participação.

Dados do participante da pesquisa	
Nome:	
Telefone:	
e-mail:	

Curitiba, _____ de _____ de _____

Assinatura do participante da pesquisa

Assinatura do Pesquisador**USO DE IMAGEM E/OU ÁUDIO**

Autorizo o uso de minha imagem e áudio para fins da pesquisa, sendo seu uso restrito a foto, áudio, bem como suas formas de descarte.

Assinatura do participante da pesquisa

Assinatura do Pesquisador

APÊNDICE B – Códigos dos DMIs

Construídos

```

1 //*****
  //Codigo Teclado de botoes
3 //*****
  //definicao da notas retiradas de https://github.com/codebendercc/
    arduino-library-files/blob/master/examples/02.Digital/toneMelody/
      pitches.h
5 //*****
#define DO  262  // Nota Do
7 #define RE  294  // Nota Re
  #define MI  330  // Nota Mi
9 #define FA  349  // Nota Fa
  //*****
11 //VARIABLES GLOBAIS
  //*****
13 int keys[]  = {11,10,9,8};          //pinos de entrada dos 4 botoes
  int tones[] = {DO,RE,MI,FA};       //notas para cada botao
15 int pino    = 13;                  //alto-falante conectado na porta 13
  //*****
17 //CONFIGURACAO ENTRADA E SAIDA
  //*****
19 void setup() {
    pinMode(pino,OUTPUT);              //define pino 13 como saida
21    for(int i = 0; i <= 3; i++){
      pinMode(keys[i],INPUT);         //define os pinos 11,10,9,8 como entrada
23    }
  }
25 //*****
  //LOOP INFINITO
27 //*****
  void loop() {
29    verificarBotao();                //chama funcao verificarBotao
  }
31
33 //*****
  //FUNCAO VERIFICAR BOTAO QUANDO PRESSIONADO
35 //*****
  void verificarBotao(){
37    int lastState = LOW;              //estado anterior em baixo
      for(int i = 0; i <= 3 ; i++){

```

```

39   while(digitalRead(keys[i])==HIGH){ //verifica se o botao foi
    precionado
    tone(pino,tones[i]); //envia sinal para o pino 13 do alto-valente
    tocando a nota no vetor tones
41   delay(200);           //aguarda 200ms para evitar ruídos
    lastState=HIGH;      //altera estado anterior do botao para alto
43  }

45   if(lastState==HIGH){//se o estado anterior estava em alto
    lastState=LOW;      //altera o estado anterior para baixo
47   noTone(pino);      //encerra o sinal enviado para o pino 13
    }
49  }
}

```

Código Teclado de Botão

```

//*****
2 //Codigo Xilofone/Teclado/Bateria com Piezo,
//*****
4 #include <SD.h>           //biblioteca cartao SD com audios do xilofone
#include <TMRpcm.h>       //biblioteca para tocar os audios no cartao SD
6
//*****
8 #define SD_ChipSelectPin 10           //pino 10 conectado ao SD
#define tempo 10                       //tempo para ler piezo
10 //*****
int piezoStatus[] = {0,0,0,0,0};      //vetor de status do piezo
12 int lastTime = 0;

14 //para tocar as notas musicais do xilofone remova as "//" da linha a
    baixo
char* notas [] ={"C3mx.wav","D3mx.wav","E3mx.wav","F3mx.wav","G3mx.wav","
    A3mx.wav"};
16
//para tocar as notas musicais da bateria remova as "//" da linha a
    baixo
18 //char* notas [] = {"cymbal.wav","kick.wav","r1.wav","r2.wav","ride.wav
    ","snare.wav"};

20 TMRpcm music; //objeto music para ser utilizado em todo o codigo

22 //*****
//SETUP
24 //*****
void setup() {
26   Serial.begin(9600);           //baud rate porta serial
    if(!SD.begin(SD_ChipSelectPin)) { //se o cartao nao esta pronto

```

```

28     Serial.println("SD fail");           //imprimir msg de erro
        return;
30 }
    else{
32     Serial.println("SD ok");           //se nao, imprimir sd ok
        music.speakerPin = 9;           //alto falante pino 9
34     pinMode(music.speakerPin, OUTPUT); //define o pino 9 como saida
        music.volume(12);               //define o volume da saida
36 }

38     for(int i = 2; i <= 6; i++){
        pinMode(i,INPUT);               //define os pinos 2,3,4,5,6 como entrada
40     }
    }
42 //*****
//LOOP INFINITO
44 //*****
void loop() {
46     if(millis() - lastTime >= tempo){ //se passou o tempo para leitura
        lerPiezo();                     //chama funcao para leitura
48     lastTime = millis();              //atualiza valor do tempo
    }
50     tocarNota();                       //chama funcao de tocar nota
    }
52

54 //*****
//funcao para ler piezo quando pressionado
56 //*****
void lerPiezo(){
58     for(int i = 2 ; i <= 6; i++){
        if(digitalRead(i) == 1){        //se piezo pressionado
60         piezoStatus[i-2] = 1;         //atualiza o vetor piezoStatus
        }
62     }
    }
64

//*****
66 //funcao para tocar a nota musical
//*****
68 void tocarNota(){
    for(int i = 0; i < 5; i++){
70     if(piezoStatus[i] == 1){          //se o vetor piezoStatus foi alterado
        music.play(notas[i]);           //toca o arquivo de audio da nota
72     piezoStatus[i] = 0;               //atualiza o vetor de piezoStatus
    }
74 }

```

```
}

```

Código Xilofone com Piezo

```

1 //*****
//Codigo Teclado capacitivo com 32 teclas
3 //Todas as teclas devem estar conectadas,
//se qualquer uma estiver desconectada pode nao funcionar corretamente
5 //*****
#include <SD.h> //biblioteca cartao SD
7 #include <TMRpcm.h> //biblioteca para tocar os audios
#include <CapacitiveSensor.h> //biblioteca sensor capacitivo
9 //*****
#define SD_ChipSelectPin 10 //pino 10 conectado ao SD
11 #define mediaInicial 11 //valor inicial das medias
#define sensor 30 //define sensibilidade das teclas
13 #define notes 31 //numero de notas/teclas
#define meanSize 3 //quantidade de medias que serao lidas
15 #define ratio 0.05 //taxa de erro para calculo de leitura
//*****
17 long means[notes][meanSize]; //matriz valor medio lido das teclas
int count = 0; //contador das medias
19 //vetor notas musicais
char* notas[]={ "C2mx.wav", "Cs2mx.wav", "D2mx.wav", "Ds2mx.wav", "E2mx.wav",
21 "F2mx.wav", "Fs2mx.wav", "G2mx.wav", "Gs2mx.wav", "A2mx.wav",
" As2mx.wav", "C3mx.wav", "Cs3mx.wav", "D3mx.wav", "Ds3mx.wav",
23 "E3mx.wav", "F3mx.wav", "Fs3mx.wav", "G3mx.wav", "Gs3mx.wav",
"A3mx.wav", "As3mx.wav", "C4mx.wav", "Cs4mx.wav", "D4mx.wav",
25 "Ds4mx.wav", "E4mx.wav", "F4mx.wav", "Fs4mx.wav", "G4mx.wav",
"Gs4mx.wav", "A4mx.wav"};
27
//*****
29 TMRpcm music; //objeto music para ser utilizado em todo o codigo

31 boolean tecla = false; //variavel status da tecla
boolean previous[notes]; //vetor status anterior
33 boolean tocarNota[notes]; //vetor notas a serem tocadas
CapacitiveSensor* sensors[notes]; //vetor com sensores capacitivos
35
//*****
37 //SETUP
//*****
39 void setup(){
Serial.begin(9600); //baud rate em 9600
41 //pinos conectado ao circuito multiplexador
pinMode(8, OUTPUT); //define pino 8 como saida
43 pinMode(9, OUTPUT); //define pino 9 como saida
pinMode(10, OUTPUT); //define pino 10 como saida

```



```

45 for(int i = 0; i < notes; i++){
47   for(int j = 0; j < meanSize; j++){
49     means[i][j] = mediaInicial;           //inicia valor das medias
51   }
52   previous[i] = false;                    //status anterior das teclas
53   tocarNota[i] = false;                  //status das teclas a serem
54   tocadas
55   sensors[i] = new CapacitiveSensor(2,4); //instancia sensor
56   sensors[i]->set_CS_Autocal_Millis(0x000000A0); //calibragem
57   automatica
58   sensors[i]->set_CS_Timeout_Millis(0x00000005); //tempo millis
59   inicial
60 }
61
62 music.speakerPin = 9;                    //alto falante pino 9
63 if (!SD.begin(SD_ChipSelectPin)) { //se o cartao sd nao esta
64   inicializado
65   Serial.println("SD fail");            //imprimir mensagem fail
66   return;
67 }
68 else{
69   Serial.println("SD ok");              //se nao, imprimir ok
70 }
71 music.volume(12);                        //configura volume
72 pinMode(music.speakerPin, OUTPUT); //alto falante como saida
73 }
74 //*****
75 //LOOP INFINITO
76 //*****
77 void loop(){
78   for(int i = 0; i < notes; i++){
79     digitalWrite(10, i%2);              //sinal no pino 10 para multiplexador
80     digitalWrite(9, (i%4)>1);           //sinal no pino 9 para multiplexador
81     digitalWrite(8, (i%8)>3);           //sinal no pino 8 para multiplexador
82     delay(1);                           //aguarda 1ms
83     long valorLido = sensors[i]->capacitiveSensor(sensor);
84     //calcula novo valor medio
85     means[i][count] = means[i][count]*(1.0 - ratio) - valorLido*ratio;
86     float novaMedia = 0;
87     for(int j = 0; j < meanSize; j++){
88       novaMedia+=means[i][j];           //soma o valor das medias
89     }
90     novaMedia/=meanSize;                //calcula a media novamente
91
92     if (valorLido > novaMedia)//se valor for maior que a nova media

```

```

    tecla = true;          //tecla pressionada, tecla = verdadeiro
89 else
    tecla = false;       //se nao, tecla = a falso
91
    tocarNota[i] = tecla; //atualiza vetor de teclas a serem tocadas
93
    previous[i] = tecla; //atualiza vetor do valor anterior da tecla
95    delay(1);           //aguarda 1ms
}
97 for(int i = 0;i < notes;i++) {
    if(tocarNota[i] == true){ //se o vetor tocarNota e igual a true
99        music.play(notas[i]); //toca o arquivo de audio da nota musical
        previous[i] = 0;      //atualiza o vetor de piezoStatus
101    }
    if(count >= meanSize) //se contador for maior que a media
103        count = 0;         //contador = 0
    else
105        count++;          //se nao soma contador
}
107 }

```

Código Teclado capacitivo

```

1 //*****
//Xilofone e Bateria com piezoeletrico e MIDI
3 //*****
#define piezo1 A0          //Piezo1 no pino A0
5 #define piezo2 A1       //Piezo2 no pino A1
#define piezo3 A2         //Piezo3 no pino A2
7 #define piezo4 A3       //Piezo4 no pino A3
#define piezo5 A4         //Piezo5 no pino A4
9
#define NOTE_MIDI_C3 48    //Nota MIDI C3
11 #define NOTE_MIDI_D3 50 //Nota MIDI D3
#define NOTE_MIDI_E3 52   //Nota MIDI E3
13 #define NOTE_MIDI_F3 53 //Nota MIDI F3
#define NOTE_MIDI_G3 55   //Nota MIDI G3
15 //*****
//VARIABLES GLOBAIS
17 //*****
int val1 = 0;             //variavel para armazenar o valor do piezo1
19 int val2 = 0;           //variavel para armazenar o valor do piezo2
int val3 = 0;             //variavel para armazenar o valor do piezo3
21 int val4 = 0;           //variavel para armazenar o valor do piezo4
int val5 = 0;             //variavel para armazenar o valor do piezo5
23
boolean estadoAnt_p1 = false; //estado anterior do piezo1
25 boolean estadoAnt_p2 = false; //estado anterior do piezo2

```

```

boolean estadoAnt_p3 = false;           //estado anterior do piezo3
27 boolean estadoAnt_p4 = false;        //estado anterior do piezo4
boolean estadoAnt_p5 = false;          //estado anterior do piezo5
29
//*****
31 //SETUP
//*****
33 void setup() {
    Serial.begin(31250);                //baud rate de 31250
35 }
//*****
37 //LOOP INFINITO
//*****
39 void loop() {

    val1 = analogRead(piezo1);          //Le o valor do piezo1
//se o valor do piezo esta entre 128 e 50 e estado anterior e falso
43 if(val1 <= 128 && val1 >=50 && estadoAnt_p1 == false) {
    enviarComando(144,NOTE_MIDI_C3,val1); //Chama a funcao enviar
comando
45     estadoAnt_p1 = true;              //Atualiza o valor do estado anterior
    delay(100);                          //Espera 100 millis
47 }
    else if(estadoAnt_p1 == true){      //se o valor anterior e
verdadeiro
49     enviarComando(128,NOTE_MIDI_C3,0); //Chama a funcao enviar
comando
    estadoAnt_p1 = false;                //Atualiza o valor do estado
anterior
51     delay(100);                       //Espera 100 millis
    }

53     val2 = analogRead(piezo2);        //Le o valor do piezo1
55     //Verifica se o valor do piezo esta entre 128 e 50 e se o estado
anterior e falso
    if(val2 <= 128 && val2 >=50 && estadoAnt_p2 == false) {
57     enviarComando(144,NOTE_MIDI_D3,val2);//Chama a funcao enviar
comando
    estadoAnt_p2 = true;                 //Atualiza o valor do estado anterior do
piezo
59     delay(100);                       //Espera 100 millis
    }
61     else if(estadoAnt_p2 == true){ //Verifica se o valor anterior e
verdadeiro
    enviarComando(128,NOTE_MIDI_D3,0); //Chama a funcao enviar comando
63     estadoAnt_p2 = false;              //Atualiza o valor do estado
anterior do piezo

```

```
65     delay(100);                //Espera 100 millis
66     }
67     val3 = analogRead(piezo3);    //Le o valor do piezo1
68     //Verifica se o valor do piezo esta entre 128 e 50 e se o estado
69     anterior e falso
70     if(val3 <= 128 && val3 >=50 && estadoAnt_p3 == false) {
71         enviarComando(144,NOTE_MIDI_E3,val3); //Chama a funcao enviar
72         comando
73         estadoAnt_p3 = true;        //Atualiza o valor do
74         estado anterior do piezo
75         delay(100);                //Espera 100 millis
76     }
77     else if(estadoAnt_p3 == true){ //Verifica se o valor anterior e
78     verdadeiro
79     enviarComando(128,NOTE_MIDI_E3,0); //Chama a funcao enviar comando
80     estadoAnt_p3 = false;          //Atualiza o valor do estado
81     anterior do piezo
82     delay(100);                //Espera 100 millis
83     }
84     val4 = analogRead(piezo4);    //Le o valor do piezo1
85     //Verifica se o valor do piezo esta entre 128 e 50 e se o estado
86     anterior e falso
87     if(val4 <= 128 && val4 >=50 && estadoAnt_p4 == false) {
88         enviarComando(144,NOTE_MIDI_F3,val4); //Chama a funcao enviar
89         comando
90         estadoAnt_p4 = true;        //Atualiza o valor do estado
91         anterior do piezo
92         delay(100);                //Espera 100 millis
93     }
94     else if(estadoAnt_p4 == true){ //Verifica se o valor anterior
95     e verdadeiro
96     enviarComando(128,NOTE_MIDI_F3,0); //Chama a funcao enviar comando
97     estadoAnt_p4 = false;          //Atualiza o valor do estado
98     anterior do piezo
99     delay(100);                //Espera 100 millis
100    }
101    val5 = analogRead(piezo5);    //Le o valor do piezo1
102    //Verifica se o valor do piezo esta entre 128 e 50 e se o estado
103    anterior e falso
104    if(val5 <= 128 && val5 >=50 && estadoAnt_p5 == false) {
105        enviarComando(144,NOTE_MIDI_G3,val5); //Chama a funcao enviar
106        comando
107        estadoAnt_p5 = true;        //Atualiza o valor do
108        estado anterior do piezo
109        delay(100);                //Espera 100 millis
```

```

    }
99     else if(estadoAnt_p5 == true){           //Verifica se o valor
anterior e verdadeiro
        enviarComando(128,NOTE_MIDI_G3,0);    //Chama a funcao enviar
comando
101     estadoAnt_p5 = false;                   //Atualiza o valor do estado
anterior do piezo
        delay(100);                           //Espera 100 millis
103     }
}
105 //*****
//Funcao enviar comandos MIDI
107 //*****
void enviarComando(int evento, int nota, int duracao){
109     Serial.write(evento);                   //comando evento
        Serial.write(nota);                  //comando nota
111     Serial.write(duracao);                 //comando duracao
}

```

Código Xilofone/Bateria com piezo e MIDI

```

//*****
2 //Codigo Teremin MIDI
//*****
4 #include <CapacitiveSensor.h> //biblioteca sensor capacitivo
#define NOTAC6 72 //Nota MIDI C3
6 #define NOTAD6 74 //Nota MIDI D3
#define NOTAE6 76 //Nota MIDI E3
8 #define NOTAF6 77 //Nota MIDI F3
#define NOTAG6 79 //Nota MIDI G3
10
/*antena conectada entre o pino 4 e 2
12 para aumentar a sensibilidade da antena deve ser utilizado
resistor entre 40M e 50M, pode ser utilizado resistores
14 de 10M em serie. Antena deve ser conectada no pino 2
*/
16 CapacitiveSensor antena = CapacitiveSensor(4,2);
void setup(){
18     Serial.begin(31250); //baud rate em 31250
        antena.set_CS_AutocaL_Millis(0x0000FFFF); //autocalibracao
20 }
22
int nota = 0;
24 int notaAnterior = 1;
void loop(){
26     long distancia = antena.capacitiveSensor(150); //leitura da distancia
        distancia /= 1000; //divisao por 1000

```

```

28     if(distancia <= 5 ) nota = 0;                               //nota desligada
30     if(distancia > 5 && distancia <= 10) nota = NOTAC6; //nota D0
32     if(distancia > 10 && distancia <= 15) nota = NOTAD6; //nota RE
34     if(distancia > 15 && distancia <= 20) nota = NOTAE6; //nota MI
36     if(distancia > 20 && distancia <= 30) nota = NOTAF6; //nota FA
38     if(distancia > 30 ) nota = NOTAG6;                          //nota SOL

36     if(nota != notaAnterior){//se nota for diferente da anterior
38         if(nota != 0){          //se nota for diferente de 0
40             enviarNotas(144,nota,127); //envia comando ligar a nota
42         }
44         //envia comando desligar a nota anterior
46         enviarNotas(128,notaAnterior,127);
48         notaAnterior = nota;      //atualiza notaAnterior;
50     }
52 }

// *****
46 //Funcao enviar notas
// *****
48 void enviarNotas(byte com, byte nota, byte velo){
50     Serial.write(com);          //envia comando ligar/desligar
52     Serial.write(nota);         //envia nota a ser tocada
54     Serial.write(velo);        //envia velocidade
56 }

```

Código Teremin

```

// *****
2 //Codigo Harpa Laser
// *****
4 #define pinLdr2 2          //LDR no pino 2
6 #define pinLdr3 3          //LDR no pino 3
8 #define pinLdr4 4          //LDR no pino 4
10 #define pinLdr5 5          //LDR no pino 5
12 #define pinLdr6 6          //LDR no pino 6
14 #define pinLdr7 7          //LDR no pino 7
16 #define pinLdr8 8          //LDR no pino 8
18 #define pinLdr9 9          //LDR no pino 9

// *****
14 //VARIaVEIS GLOBAIS
// *****
16
18 int anteriorLdr2 = 0;      //status anterior do botao 2
20 int anteriorLdr3 = 0;      //status anterior do botao 3
22 int anteriorLdr4 = 0;      //status anterior do botao 4
24 int anteriorLdr5 = 0;      //status anterior do botao 5

```

```
int anteriorLdr6 = 0;           //status anterior do botao 6
22 int anteriorLdr7 = 0;        //status anterior do botao 7
int anteriorLdr8 = 0;          //status anterior do botao 8
24 int anteriorLdr9 = 0;        //status anterior do botao 9

26
//*****
28 //SETUP
//*****
30 void setup() {
    Serial.begin(9600);         //baud rate 9600
32 }
34
//*****
36 //LOOP INFINITO
//*****
38 void loop() {
    //se o pinLdr2 foi pressionado e anterior era 0
40 if((!digitalRead(pinLdr2)) && (anteriorLdr2 == 0)){
        tocarNota(0x90, 0x37, 0x60);           //chama funcao tocar nota
42     anteriorLdr2=1;                           //atualiza status anterior do
        botao
    }
44 else if((digitalRead(pinLdr2)) ){             //se botao esta
        desligado
        tocarNota(0x90, 0x37, 0x00);           //enviar comando
        desligar
46     anteriorLdr2=0;                           //atualiza anterior para 0
    }
48 //se o pinLdr3 foi pressionado e anterior era 0
    if((!digitalRead(pinLdr3)) && (anteriorLdr3 == 0)){
50     tocarNota(0x90, 0x38, 0x60);             //chama funcao
        tocar nota
        anteriorLdr3=1;                         //atualiza status anterior do
        botao
52     }
54 else if((digitalRead(pinLdr3)) ){             //se botao esta
        desligado
        tocarNota(0x90, 0x38, 0x00);           //enviar comando
        desligar
56     anteriorLdr3=0;                           //atualiza anterior para 0
    }
58 //se o pinLdr4 foi pressionado e anterior era 0
    if((!digitalRead(pinLdr4)) && (anteriorLdr4 == 0)){
60     tocarNota(0x90, 0x39, 0x60);             //chama funcao
```

```
        tocar nota
            anteriorLdr4=1;                                //atualiza status anterior do
        botao
62    }
64    else if((digitalRead(pinLdr4)) ){                    //se botao esta
        desligado
            tocarNota(0x90, 0x39, 0x00);                //enviar comando
        desligar
66    anteriorLdr4=0;                                    //atualiza anterior para 0
    }
68    //se o pinLdr5 foi pressionado e anterior era 0
    if((!digitalRead(pinLdr5)) && (anteriorLdr5 == 0)){
70    tocarNota(0x90, 0x3A, 0x60);                        //chama funcao
        tocar nota
            anteriorLdr5=1;                                //atualiza status anterior do
        botao
72    }
    else if((digitalRead(pinLdr5)) ){                    //se botao esta
        desligado
74    tocarNota(0x90, 0x3A, 0x00);                        //enviar comando
        desligar
            anteriorLdr5=0;                                //atualiza anterior para 0
76    }
    //se o pinLdr6 foi pressionado e anterior era 0
78    if((!digitalRead(pinLdr6)) && (anteriorLdr6 == 0)){
        tocarNota(0x90, 0x3B, 0x60);                    //chama funcao
        tocar nota
80    anteriorLdr6=1;                                    //atualiza status anterior do
        botao
    }
82    else if((digitalRead(pinLdr6)) ){                    //se botao esta
        desligado
            tocarNota(0x90, 0x3B, 0x00);                //enviar comando
        desligar
84    anteriorLdr6=0;                                    //atualiza anterior para 0
    }
86    //se o pinLdr7 foi pressionado e anterior era 0
    if((!digitalRead(pinLdr7)) && (anteriorLdr7 == 0)){
88    tocarNota(0x90, 0x3C, 0x60);                        //chama funcao
        tocar nota
            anteriorLdr7=1;                                //atualiza status anterior do
        botao
90    }
    else if((digitalRead(pinLdr7)) ){                    //se botao esta
        desligado
92    tocarNota(0x90, 0x3C, 0x00);                        //enviar comando
```



```

desligar
  anteriorLdr7=0;                               //atualiza anterior para 0
94  }
//se o pinLdr7 foi pressionado e anterior era 0
96  if((digitalRead(pinLdr8)) && (anteriorLdr8 == 0)){
      tocarNota(0x90, 0x3D, 0x60);             //chama funcao
      tocar nota
98      anteriorLdr8=1;                           //atualiza status anterior do
      botao
100  }
  else if((!digitalRead(pinLdr8)) ){           //se botao esta
      desligado
102      tocarNota(0x90, 0x3D, 0x00);             //enviar comando
      desligar
      anteriorLdr8=0;                           //atualiza anterior para 0
104  }
//se o pinLdr7 foi pressionado e anterior era 0
106  if((digitalRead(pinLdr9)) && (anteriorLdr9 == 0)){
      tocarNota(0x90, 0x3E, 0x60);             //chama funcao
      tocar nota
108      anteriorLdr9=1;                           //atualiza status anterior do
      botao
110  }
  else if((!digitalRead(pinLdr9)) ){           //se botao esta
      desligado
112      tocarNota(0x90, 0x3E, 0x00);             //enviar comando
      desligar
      anteriorLdr9=0;                           //atualiza anterior para 0
114  }

116  delay(5);
}
118
//*****
120 //funcao para enviar comando MIDI para tocar nota musical
//*****
122 void tocarNota(int command, int note, int velo) {
      Serial.write(command);                   //enviar comando ligar ou desligar
124      Serial.write(note);                     //enviar comando nota
      Serial.write(velo);                      //enviar comando velocidade
126 }

128 //*****
//funcao para enviar comando MIDI para slide
130 //*****

```

```

132 void controlChange(byte command, byte ef, byte val){
    Serial.write(command); //enviar comando slide
    Serial.write(ef); //enviar comando qual slide
134 Serial.write(val); //enviar valor
    delay(100);
136 }
138
//*****
140 //funcao para enviar comando MIDI para trocar o instrumento musical
//*****
142 void configInstrument(byte command, byte inst) {
    Serial.write(command); //enviar comando trocar instrumento
144 Serial.write(inst); //enviar comando qual instrumento
}

```

Código Harpa Laser

```

1 //*****
//Codigo Controlador MIDI com todos sensores
3 //*****
#define btn2 2 //botao 1 no pino 2
5 #define btn3 3 //botao 2 no pino 3
#define btn4 4 //botao 3 no pino 4
7 #define btn5 5 //botao 4 no pino 5
#define btn6 6 //botao 5 no pino 6
9 #define btn7 7 //botao 6 no pino 7

11 //*****
//VARIÁVEIS GLOBAIS
13 //*****
int anteriorBtn2 = 0; //status anterior do botao 2
15 int anteriorBtn3 = 0; //status anterior do botao 3
int anteriorBtn4 = 0; //status anterior do botao 4
17 int anteriorBtn5 = 0; //status anterior do botao 5
int anteriorBtn6 = 0; //status anterior do botao 6
19 int anteriorBtn7 = 0; //status anterior do botao 7

21 //*****
//SETUP
23 //*****
void setup() {
25 Serial.begin(9600); //baud rate em 9600
pinMode(btn2, INPUT_PULLUP); //btn2 como input e pullup
27 pinMode(btn3, INPUT_PULLUP); //btn3 como input e pullup
pinMode(btn4, INPUT_PULLUP); //btn4 como input e pullup
29 pinMode(btn5, INPUT_PULLUP); //btn5 como input e pullup
pinMode(btn6, INPUT_PULLUP); //btn6 como input e pullup

```

```
31  pinMode(btn7, INPUT_PULLUP);           //btn7 como input e pullup
    }
33
//*****
35 //LOOP INFINITO
//*****
37 void loop() {
    //se o btn2 foi pressionado e anterior era 0
39  if(!digitalRead(btn2) && (anteriorBtn2 == 0)){
        tocarNota(0x90, 0x37, 0x60);           //chama funcao tocar nota
41  anteriorBtn2=1;           //atualiza status anterior do botao
    }
43  else if((digitalRead(btn2)) ){           //se botao esta desligado
        tocarNota(0x90, 0x37, 0x00);           //enviar comando desligar
45  anteriorBtn2=0;           //atualiza anterior para 0
    }
47  //se o btn3 foi pressionado e anterior era 0
    if(!digitalRead(btn3) && (anteriorBtn3 == 0)){
49  tocarNota(0x90, 0x38, 0x60);           //chama funcao tocar nota
        anteriorBtn3=1;           //atualiza status anterior do botao
51
    }
53  else if((digitalRead(btn3)) ){           //se botao esta desligado
        tocarNota(0x90, 0x38, 0x00);           //enviar comando desligar
55  anteriorBtn3=0;           //atualiza anterior para 0
    }
57  //se o btn4 foi pressionado e anterior era 0
    if(!digitalRead(btn4) && (anteriorBtn4 == 0)){
59  tocarNota(0x90, 0x39, 0x60);           //chama funcao tocar nota
        anteriorBtn4=1;           //atualiza status anterior do botao
61
    }
63  else if((digitalRead(btn4)) ){           //se botao esta desligado
        tocarNota(0x90, 0x39, 0x00);           //enviar comando desligar
65  anteriorBtn4=0;           //atualiza anterior para 0
    }
67  //se o btn5 foi pressionado e anterior era 0
    if(!digitalRead(btn5) && (anteriorBtn5 == 0)){
69  tocarNota(0x90, 0x3A, 0x60);           //chama funcao tocar nota
        anteriorBtn5=1;           //atualiza status anterior do botao
71
    }
73  else if((digitalRead(btn5)) ){           //se botao esta desligado
        tocarNota(0x90, 0x3A, 0x00);           //enviar comando desligar
75  anteriorBtn5=0;           //atualiza anterior para 0
    }
    //se o btn6 foi pressionado e anterior era 0
77  if(!digitalRead(btn6) && (anteriorBtn6 == 0)){
```

```

79     tocarNota(0x90, 0x3B, 0x60);    //chama funcao tocar nota
        anteriorBtn6=1;                //atualiza status anterior do botao
    }
81     else if((digitalRead(btn6)) ){    //se botao esta desligado
        tocarNota(0x90, 0x3B, 0x00);    //enviar comando desligar
83     anteriorBtn6=0;                //atualiza anterior para 0
    }
85     //se o btn7 foi pressionado e anterior era 0
    if((!digitalRead(btn7)) && (anteriorBtn7 == 0)){
87         tocarNota(0x90, 0x3C, 0x60);    //chama funcao tocar nota
            anteriorBtn7=1;                //atualiza status anterior do
            botao
89     }
    else if((digitalRead(btn7)) ){    //se botao esta desligado
91         tocarNota(0x90, 0x3C, 0x00);    //enviar comando desligar
            anteriorBtn7=0;                //atualiza anterior para 0
93     }

95     controlChange(0xB0,0X01,analogRead(A1)/8); //chama funcao control
        change 1
    controlChange(0xB0,0X01,analogRead(A2)/8); //chama funcao control
        change 2
97     controlChange(0xB0,0X07,analogRead(A3)/8); //chama funcao control
        change 3
    controlChange(0xB0,0X02,analogRead(A4)/8); //chama funcao control
        change 4
99

101    delay(2);                        //aguarda 2ms
    }
103
//*****
105 //funcao para enviar comando MIDI para tocar nota musical
//*****
107 void tocarNota(int command, int note, int velo) {
    Serial.write(command);    //enviar comando ligar ou desligar
109    Serial.write(note);        //enviar comando nota
    Serial.write(velo);        //enviar comando velocidade
111 }

113 //*****
//funcao para enviar comando MIDI para slide
115 //*****
void controlChange(byte command, byte ef, byte val){
117     Serial.write(command);    //enviar comando slide
    Serial.write(ef);        //enviar comando qual slide
119     Serial.write(val);        //enviar valor

```

```
121     delay(100);  
122 }  
123  
124 // *****  
125 //funcao para enviar comando MIDI para trocar o instrumento musical  
126 // *****  
127 void configInstrument(byte command, byte inst) {  
128     Serial.write(command); //enviar comando trocar instrumento  
129     Serial.write(inst);   //enviar comando qual instrumento  
130 }
```

Código Controlador MIDI