**Modelo Canônico de Artigo científico**

André luis Pires Wenceslau Soares\* Bianca Andrade Galvão\* João Vicente\*Vinicius Lins de Faria\*

14/07/2022

**Resumo**

O projeto do teclado assistivo visa empregar acessibilidade para deficientes auditivos utilizando de um software online chamado tinkercard com componentes eletrônicos, sendo eles, um teclado matricial 4X4, um arduino uno r3, um display 32 LCD 16 x 2 (I2C), um piezo e um motor de vibração, os quais, foram utilizados para fabricação do teclado assistivo que vibra e transmite um som em cada tecla apertada. o projeto teve um protótipo funcional, porém com possibilidades de melhora dentro do árbitro do corpo estrutura do teclado e problemas eletrônicos, além de ter diversos desafios para sua confecção, principalmente na parte de impressão 3d e frequência sonora das teclas.

**Palavras-chave**: teclado, protótipo,

**Abstract**

The assistive keyboard project aims to employ accessibility for the hearing impaired using an online software called tinkercard with electronic components, namely, a 4X4 matrix keyboard, an arduino an r3, a 32 LCD 16 x 2 display (I2C), a piezo and a motor of vibration, which were used to manufacture the assistive keyboard that vibrates and transmits a sound with each pressed key. the project had a functional operation, but with improvements within the development of the possibilities of structure of various and electronic keyboards, of their challenges for their manufacture, mainly in the sound part of 3d printing and frequency of the keys.

**Keywords**: Keyboard, Prototype

**1 Introdução**

Segundo a Lei Nº 10.098 possibilidade e condição de alcance para utilização, com segurança e autonomia, de espaços, mobiliários, equipamentos urbanos, edificações, transportes, informação e comunicação, inclusive seus sistemas e tecnologias, bem como de outros serviços e instalações abertos ao público, de uso público ou privados de uso coletivo, tanto na zona urbana como na rural, por pessoa com deficiência ou com mobilidade reduzida. Com isso, entra o nosso projeto do teclado matricial com efeitos sonoros em suas teclas para diferenciá-las entre si, feito para garantir a acessibilidade para as pessoas com deficiência visual, que muitas vezes não ganham o suporte necessário da sociedade.

**2 Materiais e Métodos**

**2.1 Materiais**

A equipe utilizou um software online, Tinkercad (ferramenta de design de modelos 3D em CAD e também de simulação de circuitos elétricos analógicos e digitais, desenvolvida pela Autodesk), para prototipar e desenvolver o teclado. Por meio dele, através dos componentes listados abaixo, foi possível desenvolver o prototipo tanto virtual quanto físico.

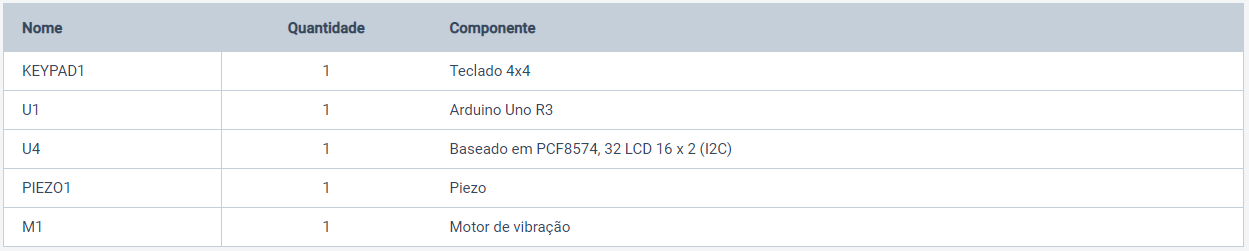


Tabela 1: Componentes utilizados no projeto

Fonte: Tinkercard

* Teclado Matricial 4×4

Trata-se de um componente do Arduino bastante utilizado para a realização da entrada de dados. Composto por 16 teclas dispostas em 4 linhas x 4 colunas, e um conector de 8 pinos para ligação.

No programa, primeiro definimos todos os pinos das linhas como saída (pinos 3, 4, 5 e 6), e os pinos de colunas como entrada (pinos 8,9,10 e 11). Um loop se encarrega de colocar cada pino de saída (linhas) em estado alto (HIGH), e verificar se alguma tecla foi pressionada, por meio de um comando IF para cada coluna. Caso isso aconteça, é gerada uma saída no serial monitor com a informação correspondente à qual tecla foi pressionada no teclado matricial.

Após a definição da posição das teclas, é possível configurar o teclado para exibir os caracteres que você desejar, ou até mesmo acionar outras portas do Arduino, já que o teclado nada mais é do que uma série de push-buttons dispostos em forma de matriz.

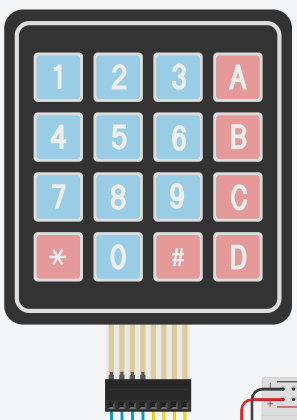


Imagem 1: Teclado Matricial 4x4 Tinkercad

Fonte: Tinkercard



Imagem 2: Teclado Matricial 4×4 real

Fonte: https://www.filipeflop.com/produto/teclado-matricial-de-membrana-16-teclas/

* Arduino UNO R3

O Arduino é uma plataforma open-hardware e possui seu próprio ambiente de desenvolvimento baseado na linguagem C. Trata-se de uma placa microcontroladora baseada no ATmega328P. Possui 14 pinos de entrada/saída digital (dos quais 6 podem ser usados ​​como saídas PWM), 6 entradas analógicas, um ressonador cerâmico de 16 MHz, uma conexão USB, um conector de alimentação, um conector ICSP e um botão de reset. Contém todo o suporte necessário para dar base ao microcontrolador; basta conectá-lo a um computador com um cabo USB ou alimentá-lo com um adaptador AC-to-DC ou bateria para começar.

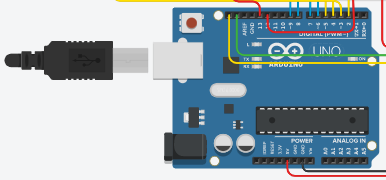


Imagem 3: Arduino UNO R3 Tinkercad

Fonte: Tinkercard

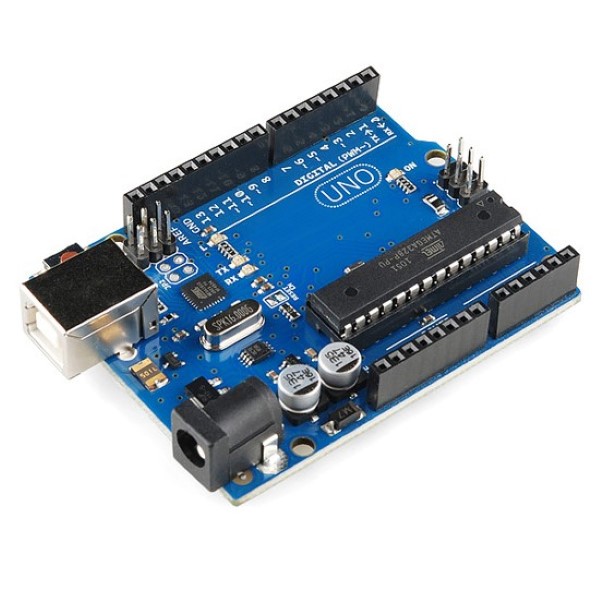


Imagem 4: Arduino UNO R3 Real

Fonte: https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino/

* Display 32 LCD 16 x 2 (I2C)

Possui 16 colunas e 2 linhas, com backlight (luz de fundo) azul e letras na cor branca. Para conexão, são 16 pinos, dos quais usamos 12 para uma conexão básica, já incluindo as conexões de alimentação (pinos 1 e 2), backlight (pinos 15 e 16) e contraste (pino 3). Para a conexão do display ao Arduino Uno a equipe utilizou apenas 4 pinos de dados (pinos digitais 2, 3, 4 e 5), e 2 pinos de controle (pinos digitais 11 e 12).

O controle desse display pode ser feito utilizando-se a biblioteca LiquidCrystal, já embutida na IDE do Arduino. No início do programa (linha 8), definiu-se os pinos que seriam utilizados pelo displays, nesse formato: LiquidCrystal lcd(<pino RS>, <pino enable>, <pino D4>, <pino D5>, <pino D6>, <pino D7>). No setup, foi inicializado o display definindo o número de colunas e linhas com o comando lcd.begin(16,2).

Possíveis erros podem surgir acarretando no não funcionamento do projeto tornando importante os seguintes passos: Verificar os jumpers, se estão na mesma coluna dos terminais dos componentes, fazendo assim a conexão; Verificar se os jumpers estão ligados nos pinos corretos no Arduino; Verificar se as conexões do display LCD estão corretas; Verificar se o código carregou na placa através da IDE Arduino.



Imagem 5: Display LCD 16 x 2 (I2C) Tinkercad

Fonte: Tinkercard

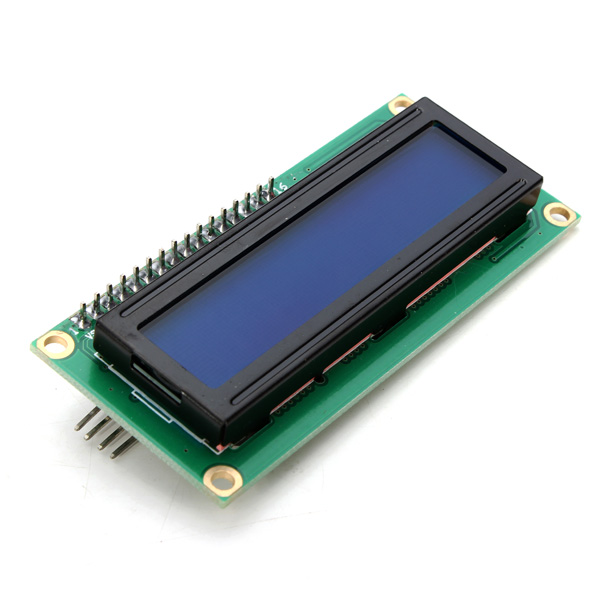


Imagem 6: Display LCD 16 x 2 (I2C) Real

Fonte: https://www.filipeflop.com/produto/display-lcd-16x2-i2c-backlight-azul/

* Piezo

Sensor Piezo elétrico. A palavra piezo é de origem grega e significa pressionar, torcer, portanto, piezos são elementos que utilizam o efeito piezoelétrico - capacidade que alguns cristais possuem de gerar tensão elétrica por resposta a uma pressão mecânica. Sendo neste caso utilizado para emitir uma frequência ao pressionar a tecla desejada, ou seja, após a aplicação de uma pressão.

Um disco piezo funciona quando uma corrente elétrica passa pelo material cerâmico do disco, fazendo com que ele mude de forma e produza um som (um clique). O disco também funciona de forma inversa: quando se bate nele ou ele sofre algum tipo de pressão, a força no material provoca a geração de uma corrente elétrica.

Por se tratar do uso associado a um arduino, utiliza-se um buzzer, disco piezo encapsulado em uma proteção plástica. O pino positivo do buzzer ficará no pino 9 e a negativa no GND.

O Buzzer utilizado neste projeto foi o do tipo passivo. Estrutura semelhante ao ativo, porém a diferença entre eles é que o passivo é ideal para melodias, pois é capaz de variar a frequência entregue aos terminais do componente, se apenas energizá-lo com uma única frequência ele emitirá apenas um estalo parecendo estar com defeito.

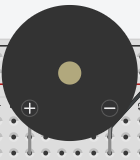


Imagem 7: Buzzer Tinkercad

Fonte: Tinkercard



Imagem 8: Buzzer passivo real

Fonte: https://repositorio.animaeducacao.com.br/bitstream/ANIMA/18843/1/TCC%20VERSÃO%20FINAL%20-%20LADY%20NARA%20E%20MAYARA\_Corrigido.pdf

* Motor de Vibração

O motor de vibração utilizado neste projeto foi o modelo 1027. É um componente semelhante ao usado para vibração de aparelhos celulares, que são executados assim que ocorre uma ação. Ideal para aplicações em robótica de pequeno porte que não sejam visuais e sonoras, apenas sensorial.



Imagem 9: Motor de Vibração Tinkercad

Fonte: Tinkercard



Imagem 10: Motor de Vibração Real

Fonte: https://www.institutodigital.com.br/produto/motor-de-vibracao-1027/

Para a acomodação do protótipo, a equipe realizou uma modelagem no SolidWorks, software de computação gráfica que permite a criação de objetos com modelagem tridimensional, de um case que comportasse todos os componentes. Após a finalização da modelagem, foi feita a impressão desse case através de uma impressora 3D.



Imagem 11: Simulação do Protótipo acoplado a case

Fonte: Própria

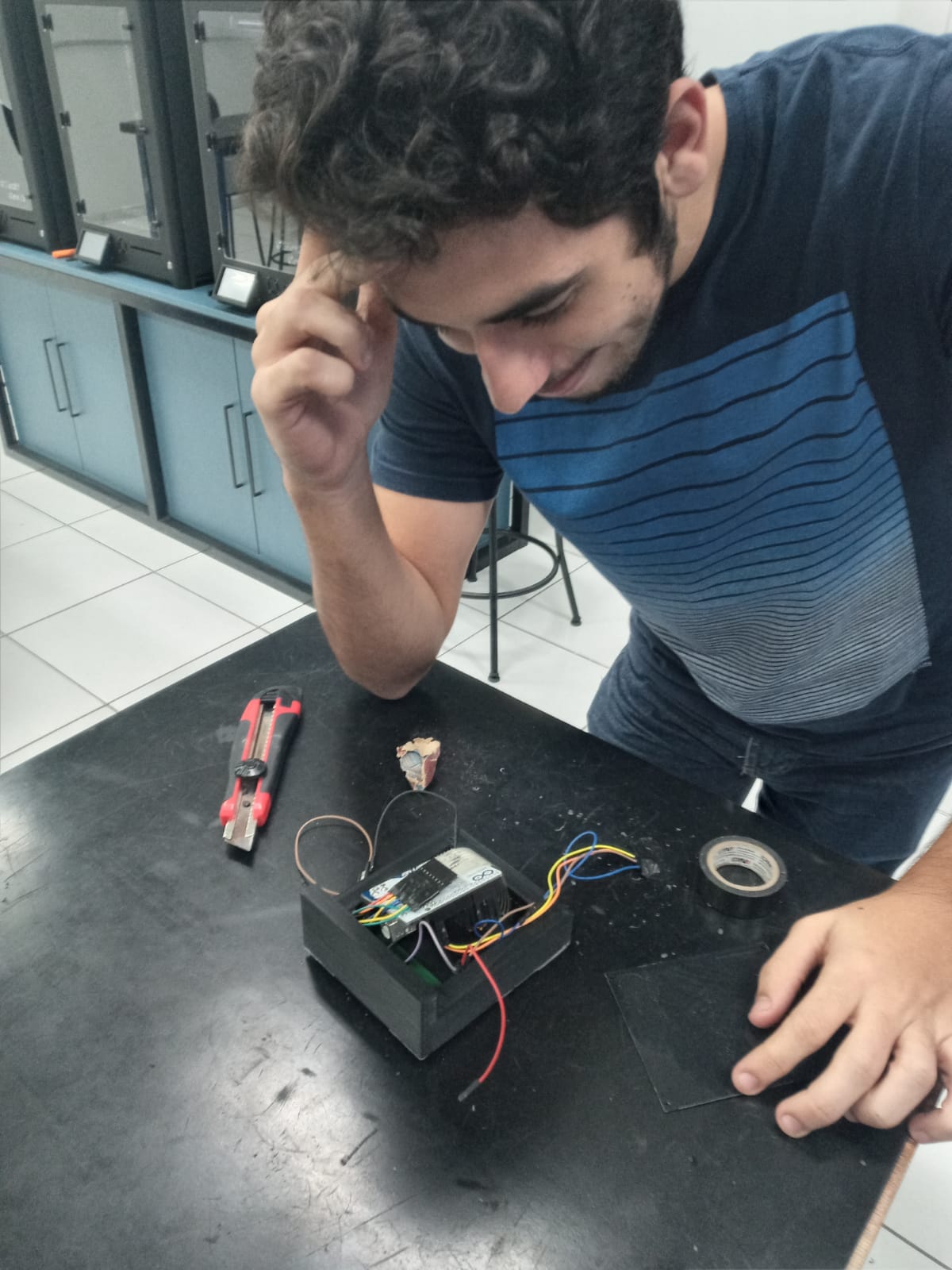


Imagem 12: Protótipo acoplado a case

Fonte: Própria

**2.2 Métodos**

Nesta seção serão expostos as metodologias de pesquisa utilizadas pela equipe ao longo do projeto compreendendo a conceituação teórica do tipo de trabalho, o desenvolvimento do protótipo e a funcionalidade/utilidade pretendida.

Este projeto foi dividido e desenvolvido em várias etapas, estas foram relatadas a cada reunião e salvas no diário de bordo pessoal da equipe. Sendo o seguinte exposto abaixo:

Primeiro houve um brainstorming, ou seja, foram coletadas todas as ideias de todos os componentes da equipe para futuramente, após pesquisas, serem filtradas deixando apenas as mais viáveis. Sugestões iniciais: código morse para representar som, frequências específicas (pesquisa para constatar a existência ou não de um padrão para portadores de deficiência auditiva), possibilidade de instalação de um som que representasse uma “voz” possuindo a frequência referente a cada tecla. Sobre os componentes e suas programações no Arduíno, foi definida para entrada de dados: Teclado, para o processamento: Bibliotecas ou Lógica no Arduíno e para saída de dados: Display e Som (buzzer). Ao fundir todas as ideias, foi possível desenvolver um protótipo inicial.

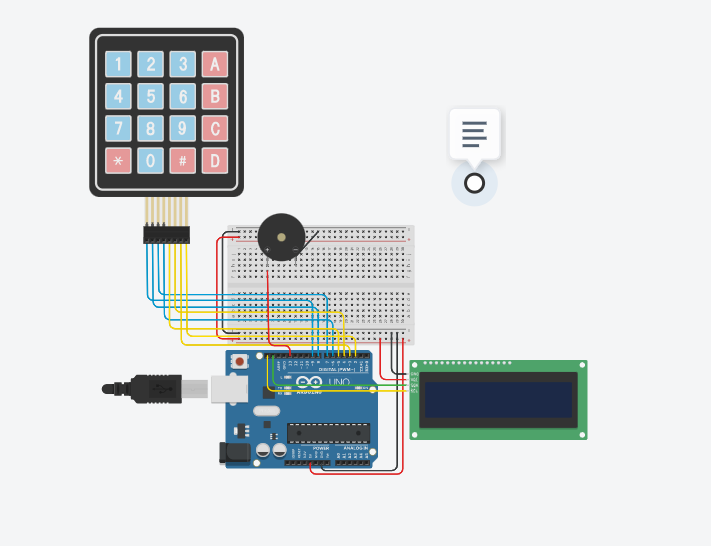


Imagem 13: Circuito base - Protótipo inicial - Tinkercad

Fonte: Tinkercard

Após testes e simulações embasados nas bibliotecas do arduino, utilização de exemplos de projetos já existentes na plataforma e fontes educacionais de ensino de eletrônica, foi possível finalizar a simulação. Código obtido:

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\*TRABALHO EMBS-CIMATEC|TRAINEEs: ANDRÉ;BIANCA;J.VICENTE;VINÍCIUS \**

*\*--------------------------------------------------------------- \**

*\* Teclado Assistivo: o projeto utiliza de um teclado Matricial, \**

*\* lógica em sistema embarcado, dispositivos de saída em LCD \**

*\* para verificação e em Buzzer para representação por frequências \**

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\*/*

*// biblioteca Keypad e LCD*

*#include <Keypad.h>*

*#include <Adafruit\_LiquidCrystal.h>*

*Adafruit\_LiquidCrystal lcd(0);*

*const int Buz =13; // Buzzer na porta 13*

*// Matriz 4x4*

*const byte ROWS = 4; // 4 linhas*

*const byte COLS = 4; // 4 colunas*

*char Keys[ROWS][COLS]= // La matrice del Keypad 4x4.*

*{*

*{'1','2','3','A'} ,*

*{'4','5','6','B'} ,*

*{'7','8','9','C'} ,*

*{'\*','0','#','D'}*

*};*

*byte colPins[4] = {5,4,3,2}; // Pin assegnati alle colonne*

*byte rowPins[4] = {9,8,7,6}; // Pin assegnati alle righe*

*// inicialização Keypad*

*Keypad keypad = Keypad( makeKeymap(Keys), rowPins, colPins, ROWS, COLS);*

*void setup(){*

*Serial.begin(9600); // inicializar monitor serial*

*lcd.begin(16, 2);// inicializar lcd*

*lcd.clear();// limpar lcd*

*pinMode(Buz,OUTPUT); // declarar Buzzer como saída*

*lcd.print("teclado assistivo");*

*lcd.setCursor(0, 1);*

*delay(1000); // espera 1s*

*}*

*void loop(){*

*char tecla = keypad.getKey();//capta dados do teclado 4x4*

*/\*lógica se alguma tecla for clicada entra em analise de casos*

*será imprimida no LCD e uma freqência é emetida para cada tecla*

*\*/*

*if (tecla != NO\_KEY) {*

*switch (tecla) {*

*case '0':*

*lcd.print(tecla);*

*tone(Buz,20,90);*

*break;*

*case '1':*

*lcd.print(tecla);*

*tone(Buz,50,90);*

*break;*

*case '2':*

*lcd.print(tecla);*

*tone(Buz,100,90);*

*break;*

*case '3':*

*lcd.print(tecla);*

*tone(Buz,150,90);*

*break;*

*case '4':*

*lcd.print(tecla);*

*tone(Buz,200,90);*

*break;*

*case '5':*

*lcd.print(tecla);*

*tone(Buz,250,90);*

*break;*

*case '6':*

*lcd.print(tecla);*

*tone(Buz,300,90);*

*break;*

*case '7':*

*lcd.print(tecla);*

*tone(Buz,350,90);*

*break;*

*case '8':*

*lcd.print(tecla);*

*tone(Buz,400,90);*

*break;*

*case '9':*

*lcd.print(tecla);*

*tone(Buz,450,90);*

*break;*

*case 'A':*

*lcd.print(tecla);*

*tone(Buz,500,90);*

*break;*

*case 'B':*

*lcd.print(tecla);*

*tone(Buz,550,90);*

*break;*

*case 'C':*

*lcd.print(tecla);*

*tone(Buz,600,90);*

*break;*

*case 'D':*

*lcd.print(tecla);*

*tone(Buz,650,50);*

*break;*

*case '\*':*

*lcd.print(tecla);*

*tone(Buz,700,90);*

*break;*

*case '#':*

*lcd.print(tecla);*

*tone(Buz,750,90);*

*}*

*tecla = NO\_KEY;//reseta váriável*

*}*

*}*

**3 Resultados e Discussões**

**3.1 Resultados**

Após uma série de pesquisas e muito esforço dos participantes tivemos a confecção de um protótipo de um teclado assistivo com um teclado matricial 4X4 com o objetivo de auxiliar deficientes auditivos.

**3.2 Discussões**

Inicialmente, é importante ressaltar todas as dificuldades que tivemos no projeto, começando com a dificuldade de achar uma frequência para as teclas, foi testado diversos modelos de frequência, mas no fim optamos por frequências semelhantes a teclas de teclado musical. Depois, teve-se a dificuldade de adequar o modelo 3d para a impressão 3d, para evitar menor uso de material, porém no fim conseguimos imprimir mesmo o modelo 3d estando um pouco errado. Partindo para o dia da montagem física, este foi um dia que tivemos maiores complicações, iniciou-se com a dificuldade de instalar a biblioteca para o lcd, em seguida, muito provavelmente por um problema elétrico o teclado possuia uma das colunas sem funcionar e o modelo 3d da case estava muito apertado e precisou-se aumentar a medidas com um estilete.

Para pontos de melhoria é importante ressaltar uma melhora no 3d para a possibilidade do encaixe de todos os componentes sem problemas. Poderia mudar o código para reduzir a quantidade de “switch case”, no entanto isso não é algo obrigatório.

**4 Conclusão**

Com tudo que vimos anteriormente, é nítido que esse projeto é de grande importância para o avanço da acessibilidade no nosso país, que mesmo com as complicações que ocorreram em meio ao projeto, com dificuldades na parte eletrônica, de programação e modelagem 3d, é possível notar que ele foi um sucesso e que é capaz de comprir sua função devidamente, emitindo sons e vibrações conforme o apertar das teclas, no entanto ainda deve ser aprimorado de diversas maneiras, logo deve-se continuar trabalhos no protótipo em questão.

**Referências**

<https://www.filipeflop.com/blog/teclado-matricial-4x4-arduino/>

[L10098 (planalto.gov.br)](https://www.planalto.gov.br/ccivil_03/LEIS/L10098.htm)

<https://docs.arduino.cc/hardware/uno-rev3>

<https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino/>

<https://www.filipeflop.com/blog/como-utilizar-o-display-lcd-16x2/>

<https://www.arduinoportugal.pt/usando-buzzer-arduino-sensor-piezo-eletrico/>

<https://www.tinkercad.com/things/fKC2iAuZdGy-projeto-13-sensor-de-batida-piezo>

https://repositorio.animaeducacao.com.br/bitstream/ANIMA/18843/1/TCC%20VERSÃO%20FINAL%20-%20LADY%20NARA%20E%20MAYARA\_Corrigido.pdf

**Agradecimentos**

Texto sucinto aprovado pelo periódico em que será publicado. Último elemento pós-textual.