



# Braille

---

CONVERSÃO PORTÁTIL DE TEXTO



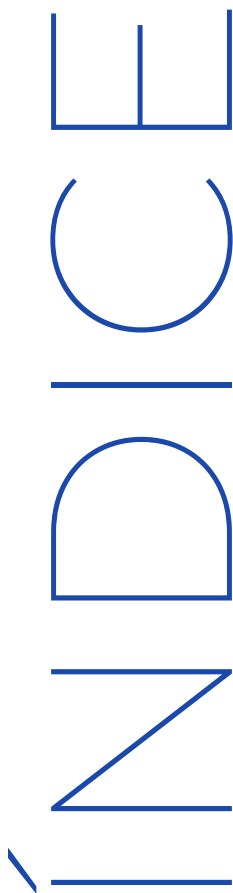
**Unidade Curricular:** Tecnologia da  
Informática

**Curso:** Engenharia Informática

**Data:** 05/12/2022

Diogo Ramos Barbosa  
Nº 2021234034

Luana Carolina Reis  
Nº 2022220606



01.

Problema a resolver

02.

Soluções existentes

03.

Limitações destas soluções

04.

Proposta de solução

05.

Materiais e custos  
associados

06.

Cuidados

07.

Conclusão

# PROBLEMA A RESOLVER

Tendo em consideração o aumento exponencial da utilização de dispositivos eletrônicos por parte da sociedade, e pensando nos casos de pessoas com deficiências visuais mais agravadas, decidimos criar um conversor de texto portátil, que o identifica e traduz para *Braille* físico.

"Segundo a OMS, cerca de 36 milhões de pessoas no mundo são cegas e outras 217 milhões tem baixa visão" (Fundação Dorina, 2016). Foi principalmente a pensar nestes 36 milhões que trabalhamos para a elaboração de um hipotético dispositivo portátil, que promete melhorar e aumentar a qualidade de muitas vidas, proporcionando-lhes alguma independência.

---

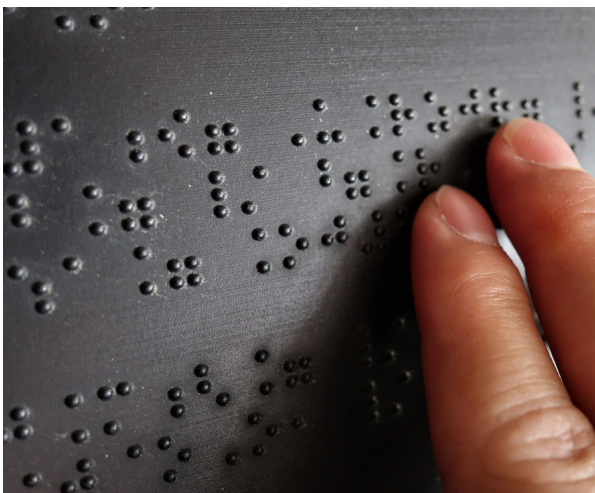


Fig. 1 - Leitura de *Braille*



Fig. 2 - *Technology can be our eyes*

# SOLUÇÕES EXISTENTES

Como seria de esperar, já existem inúmeras soluções, sendo a mais notável atualmente a função de "TalkBack", uma tecnologia de assistência disponível em qualquer dispositivo e que fornece *feedback* em formato de áudio, através do processamento de todas as informações de um ecrã. Este *feedback* pode passar pela descrição das funções disponíveis ou pela conversão exata de mensagens de texto recebidas, por exemplo.

Para além desta, existem inúmeras aplicações, desenvolvidas com o intuito de auxiliar o utilizador invisual no seu dia a dia, por exemplo, conectando-o a outros utilizadores com visão perfeita, que o poderão ajudar.

Durante a pesquisa por outras tecnologias, houve uma em particular que, ainda que em fase inicial de desenvolvimento, se mostrou promissora. Trata-se de uma aplicação para dispositivos móveis, baseada no conceito de assistência cognitiva, que é capaz de mapear um espaço interior e dar direções mais específicas (a quantos metros o utilizador está da porta, por quantas portas terá de passar até chegar à saída, ...). É também capaz de detetar a presença de outras pessoas, identificando-as e analisando as suas emoções, com recurso a tecnologias de reconhecimento facial. Aconselhamos a visualização da TED Talk "How New Technology Helps Blind People Explore the World", por Chieko Asakawa, para uma melhor compreensão da importância deste tipo de inovações e do seu funcionamento.

O nosso objetivo passa por acrescentar ao mundo dos cegos uma nova tecnologia alternativa e amplificar o espetro de escolha do utilizador, que pode decidir sobre a utilização das múltiplas ferramentas à sua disposição.

---

# LIMITAÇÕES DESTAS SOLUÇÕES

Apesar da versatilidade e utilidade destas soluções, tanto em espaços privados como em locais públicos, em algumas ocasiões, a privacidade do utilizador pode ficar exposta, visto que se tratam de alternativas baseadas numa interação auditiva com o mesmo. A leitura de toda a informação de um ecrã e respetiva conversão para áudio pode levar a algumas inconveniências, caso exista um certo grau de confidencialidade.

Poderíamos pensar que seria algo relativamente simples de resolver, com o uso de auriculares. No entanto, a pessoa pode não ter este tipo de dispositivos consigo ou pode ter também deficiências auditivas, que impossibilitam a utilização destas metodologias. E foi a pensar nestas limitações que desenvolvemos o conversor portátil de texto.

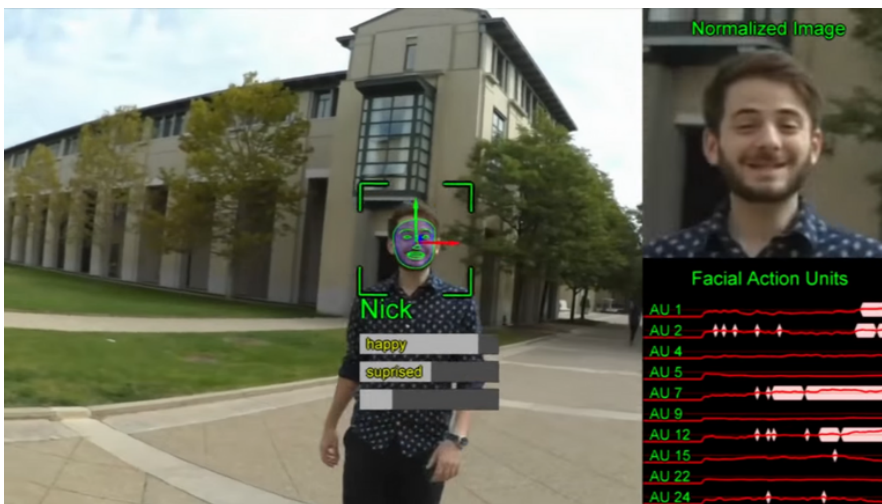


Fig. 3 - Demonstração do funcionamento da aplicação móvel de Chieko Asakawa (TED Talk)



Fig. 4 - Google TalkBack



Fig. 5 - Be My Eyes

# PROPOSTA DE SOLUÇÃO

Para a realização deste conversor portátil de texto, foi necessária a combinação de duas vertentes fundamentais: a leitura e tratamento de strings (*software*) e a respetiva representação física do resultado obtido (*hardware*).



## Software

- Responsável pela receção do texto, vindo do ecrã do telemóvel, e pela posterior conversão, caracter a caracter, para simbologia *Braille*;
- Para poder representar um método de conversão, de um modo mais prático, desenvolvemos um código que recebe *strings* e que as converte para *Braille*, através do piscar de LEDs (que fazem analogia com os solenoides utilizados no dispositivo).

## »» O código

```
String a;  
int controller_upper = 0;  
int controller_number = 0;  
  
void setup()  
{  
  for(int i = 8; i <= 13; i++){  
    pinMode(i, OUTPUT);  
  }  
  Serial.begin(9600);  
}
```

Fig. 6 - Variáveis globais e configurações iniciais

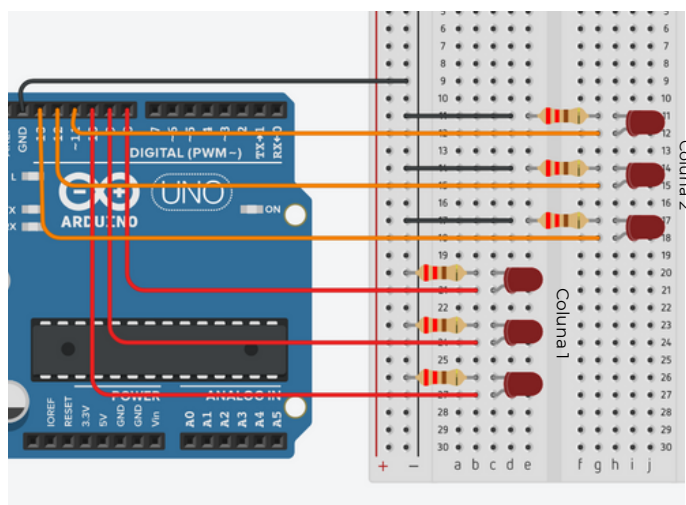


Fig. 7 - Circuito usado para a conversão (com LEDs)

## »» O código

```
void loop()
{
    // Character reception
    if(Serial.available() > 0){
        a = Serial.readStringUntil('\n');

        // Show entered text
        Serial.println(a);

        // Convert it
        for(int i = 0; i < a.length(); i++){

            if(isDigit(a[i]) == 0 and isDigit(a[i - 1]) == 1 and a[i] != ' '){
                letter(); // Change digits to letters (in a word)
            }
            if(isUpperCase(a[i]) == 1 and isUpperCase(a[i + 1]) == 0 and isUpperCase(a[i - 1]) == 0) {
                upper_case(); // In a word, if just one char is upper, call the upper_case function (just once)
            }
            if(isUpperCase(a[i]) == 1 and isUpperCase(a[i + 1]) == 1 and controller_upper == 0) {
                upper_case(); // In a word, if more than one char is upper, call the upper_case function (two times)
                upper_case();
                controller_upper = 1;
            }

            if(isDigit(a[i]) and controller_number == 0){
                numeric_digit(); // If the char is a digit, another symbol will appear before
                controller_number = 1;
            }
        }
        if(a[i] == ' '){
            controller_upper = 0; // If char is space, the word ended (reset the controllers)
            controller_number = 0;
        }
        if(a[i] == ' ' and a[i + 1] != ' ') delay(2000); // Waiting time between words...
        if(a[i] == 'A' or a[i] == 'a' or a[i] == '1') letter_a_or_1(); // A / a / 1
        if(a[i] == 'B' or a[i] == 'b' or a[i] == '2') letter_b_or_2(); // B / b / 2
        if(a[i] == 'C' or a[i] == 'c' or a[i] == '3') letter_c_or_3(); // C / c / 3
    }
    controller_upper = 0; // Reset the controller
    controller_number = 0; // Reset the controller
}
}
```

Fig. 8 - Código principal (void loop())

Desta secção, é importante reter a lógica utilizada pelo *Braille* e a maneira como foi implementada.

Todos os caracteres de uma *string* serão percorridos e convertidos para a respetiva **codificação em pontos** que, neste caso, são os LEDs.

Porém, é preciso ter em atenção alguns detalhes deste sistema de escrita:

[1] Quando o carácter é numérico, deve aparecer um símbolo a anteceder a conversão (fig. 9);

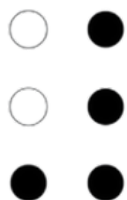


Fig. 9 - Identificação de caracter numérico (#)



Fig. 10 - Letra capitalizada

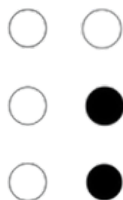


Fig. 11 - Identificação de letra

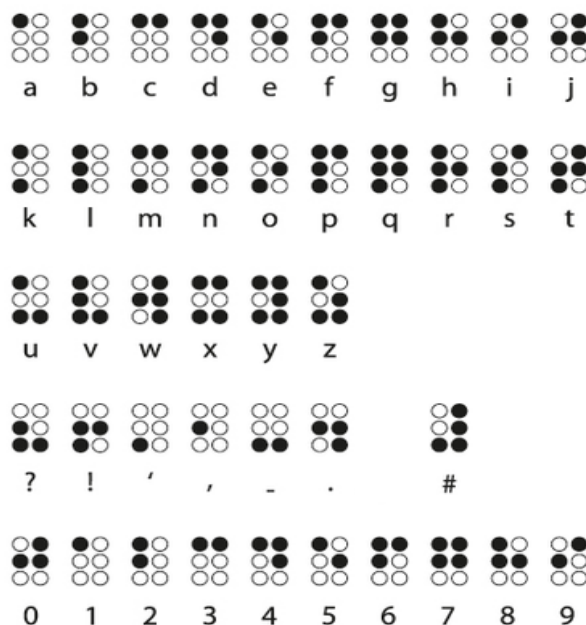


Fig. 12 - "Abecedário" do *Braille*

[ 2 ] Quando a palavra é iniciada por letra maiúscula, deve aparecer um símbolo a anteceder a conversão (fig. 10);

[ 3 ] Quando, numa palavra, mais do que uma letra está em maiúscula, deve aparecer o mesmo símbolo de [ 2 ] a anteceder a conversão, mas duas vezes seguidas (fig. 10);

[ 4 ] Numa palavra, na alternância de caracteres numéricos para letras, um outro símbolo deve anteceder a conversão (fig. 11);

[ 5 ] Existe uma correspondência direta entre os caracteres 'a' e '1', 'b' e '2', 'c' e '3', etc. Para haver distinção, segue-se o padrão definido em [ 1 ].

As *strings* que vão ser utilizadas para este programa são escolhidas pelo utilizador e são comparadas, caracter a caracter, a uma base de dados que contém o abecedário, os números e os caracteres especiais. Caso o caracter se encontre nessa base de dados, irá ser possível convertê-lo para simbologia *Braille*. Feita a sua tradução, aguarda-se um certo intervalo de tempo até passar ao caracter seguinte (1 segundo) e, na passagem de uma palavra para outra, aguarda-se o dobro do tempo (2 segundos).

Sendo mais realistas e abstraindo-nos do código exemplo anterior, as *strings* não vão ser recebidas diretamente do utilizador, mas sim através do conteúdo do ecrã do seu telemóvel, obtido de um modo semelhante àquele que a função "TalkBack" utiliza.

A informação necessária à obtenção e conversão de *strings* irá ser armazenada em memória interna, não volátil, cuja função é fornecer instruções sobre os dados que devem ou não ser temporariamente armazenados. Já as *strings*, serão armazenadas em memória RAM, uma memória volátil, isto é, uma memória temporária de acesso rápido.

Armazenados todos os dados necessários, é possível proceder à tradução das *strings*, que seria então o passo mais complexo, uma vez que é necessário ter em atenção os aspetos mencionados nos pontos [ 1 ] a [ 5 ]. É feita distinção entre números, letras e letras capitalizadas, sendo chamadas as funções adequadas ao carácter encontrado.

```
void letter_a_or_1() {  
    delay(1000);  
    digitalWrite(8, 1);  
    delay(1000);  
    digitalWrite(8, 0);  
}
```

NOTA: o código escrito está incompleto, fazendo apenas a conversão de uma amostra de caracteres ('A', 'B', 'C', 'a', 'b', 'c', '1', '2' e '3').

Fig. 13 - Função usada para representar a letra 'a', 'A' ou o dígito '1'

## Hardware

- Responsável pela representação física da conversão feita pelo software;
- Esta representação é possível através da utilização dos componentes descritos abaixo.

---

### »» Os componentes

Seleccionámos o *Arduino Nano Every* por ser um modelo compacto, versátil, económico e que satisfaz as necessidades deste projeto, que requer uma placa microcontroladora de fácil utilização.

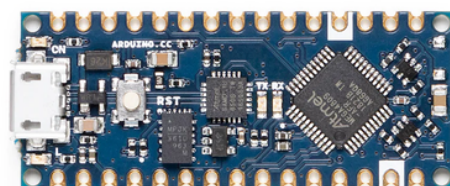


Fig. 14 - *Arduino Nano Every*

Um **solenóide** é constituído por um fio condutor (de cobre), enrolado em forma de hélice à volta de um suporte. Ao ser aplicada uma corrente elétrica neste fio, vai ser gerado um campo magnético no seu interior e nas suas proximidades. Assim, o solenóide passa a ter o comportamento magnético de um ímã. É este comportamento que vai permitir que os pinos que irão representar o *Braille* subam ou não, criando então o relevo desejado.



Fig. 15 - Solenóide *push pull*

A **placa PCB** ou "placa de circuito impresso" é uma placa fina, constituída por fibra de vidro e que funciona como componente conector, utilizado para a montagem de circuitos, com boas ligações.

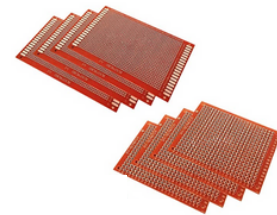


Fig. 16 - PCB Board



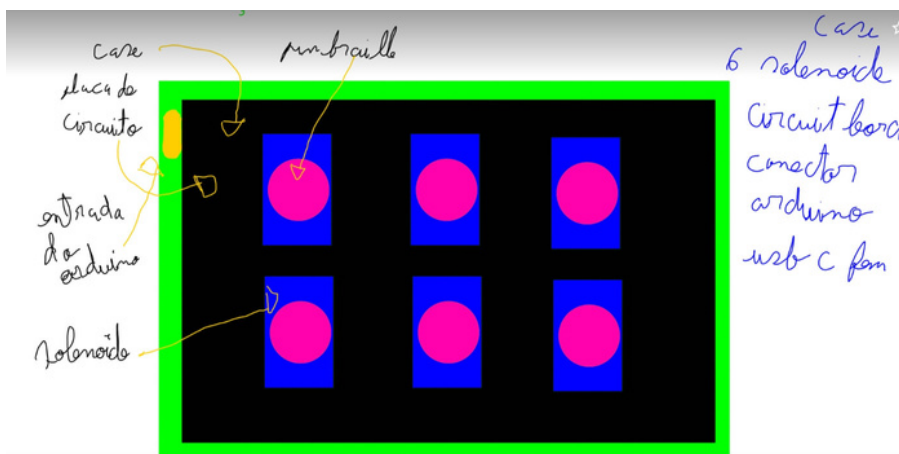
Fig. 17 - Fio de estanho

O **fio de estanho** é utilizado para efetuar as conexões elétricas.

O **Cabo USB Tipo A Macho para Micro Tipo B Macho** é usado para conectar o telemóvel ao Arduino (pode ser necessário um adaptador, caso o telemóvel não possua entrada micro USB).



Fig. 18 - Cabo USB para micro USB



É também utilizada uma **caixa protetora**, que protege todo o dispositivo de dano externo.

Fig. 19 - Esboço do dispositivo

# MATERIAIS E CUSTOS ASSOCIADOS

01

Arduino Nano Every

10,40 €

<https://store.arduino.cc/collections/boards/products/arduino-nano-every>

02

Solenoides Push Pull (6x)

52,14€ (8,69€ cada)

[https://www.amazon.es/dp/B07MJJB12M?ref\\_=cm\\_sw\\_r\\_cp\\_ud\\_dp\\_5Z18HEN3CXADM4CXRVA4](https://www.amazon.es/dp/B07MJJB12M?ref_=cm_sw_r_cp_ud_dp_5Z18HEN3CXADM4CXRVA4)

03

PCB Board

12,99€

[https://www.amazon.es/dp/B09XM4T3SG?ref\\_=cm\\_sw\\_r\\_cp\\_ud\\_dp\\_GBJ9NM5W5AQR8EBV01GY](https://www.amazon.es/dp/B09XM4T3SG?ref_=cm_sw_r_cp_ud_dp_GBJ9NM5W5AQR8EBV01GY)

04

Fio de estanho

6,61€

[https://www.amazon.es/dp/B07VHP9SZ4?ref\\_=cm\\_sw\\_r\\_cp\\_ud\\_dp\\_SFC5P6Y8WHA6DB381R04](https://www.amazon.es/dp/B07VHP9SZ4?ref_=cm_sw_r_cp_ud_dp_SFC5P6Y8WHA6DB381R04)

05

Cabo USB Tipo A Macho para Micro Tipo B Macho

5,90€

<https://store.arduino.cc/collections/boards/products/arduino-nano-every>

06

Caixa protetora

Preço variável, consoante a escolha do utilizador

**Custo total:** 88,04€ + preço da caixa protetora

**Nota:** Todos os produtos utilizados neste projeto podem ser obtidos através de outras lojas, reduzindo o custo total. Fica ao critério de cada utilizador, sendo o valor dado apenas uma estimativa.

# CUIDADOS

## »» Cuidados a ter na montagem

- Não deixar os componentes entrar em contacto com líquidos;
- Não deixar cair os componentes;
- Não dobrar os pinos do Arduino.

## »» Cuidados a ter com os dados

- Armazenar a informação que é obtida (do utilizador) em RAM e libertá-la no fim do processo ou, ao atingir o limite de memória, eliminar a informação mais antiga.

## »» Cuidados a ter no desenvolvimento do código

- Evitar repetição desnecessária de código, através da utilização de funções;
- Fazer as devidas configurações iniciais.

---

# CONCLUSÃO

Com a realização deste trabalho, esperamos ter conseguido alcançar os objetivos iniciais e ter construído um dispositivo portátil, de fácil utilização e que respeita a privacidade do utilizador. Este é mais um exemplo de que "a acessibilidade desperta a inovação" (ASAKAWA, 2016).



Fig. 5 - Frase de Chieko Asakawa (TED Talk)

- [ 1 ] Apple, "Be My Eyes – Helping blind see". [online]. Disponível em: <https://apps.apple.com/pt/app/be-my-eyes-helping-blind-see/id905177575>. [Consultado em 04/12/2022].
- [ 2 ] ASAKAWA, Chieko, "How New Technology Helps Blind People Explore the World," 2016. [online]. Disponível em: <https://youtu.be/f-mQIWnO3Ag>. [Consultado em 04/12/2022].
- [ 3 ] Canva, "Fotos". [online]. Disponível em: <https://www.canva.com/photos/MADesfysctA-close-up-of-male-hand-reading-braille-text/>. [Consultado em 03/12/2022].
- [ 4 ] Fundação Dorina, "Estatísticas sobre pessoas cegas e com baixa visão," 2016. [online]. Disponível em: <https://fundacaodorina.org.br/a-fundacao/pessoas-cegas-e-com-baixa-visao/estatisticas-da-deficiencia-visual/>. [Consultado em 03/12/2022].
- [ 5 ] InfoEscola, "Solenóide". [online]. Disponível em: <https://www.infoescola.com/fisica/solenoid/>. [Consultado em 05/12/2022].
- [ 6 ] In One Lesson, "How BRAILLE Works," 2010. [online]. Disponível em: <https://youtu.be/sqQ3gdE7ks0>. [Consultado em 03/12/2022].
- [ 7 ] PPLWare, "Curiosidade: Como surgiu o código braille?," 2013. [online]. Disponível em: <https://pplware.sapo.pt/informacao/curiosidade-como-surgiu-o-codigo-braille/>. [Consultado em 05/12/2022].
- [ 8 ] SOCO, Mary, "Google TalkBack 5.0.4, llega a dispositivos con versiones previas a Android Nougat," 2016. [online]. Disponível em: <https://www.xataka.com/celulares-y-smartphones/google-talkback-5-0-4-llega-a-dispositivos-con-versiones-previas-a-android-nougat>. [Consultado em 03/12/2022].
- [ 9 ] TechLib, "PCB". [online]. Disponível em: <https://techlib.wiki/definicion/pcb.html>. [Consultado em 05/12/2022].
- [ 10 ] VELASCO, Ariano, "5 aplicativos que podem ajudar deficientes visuais," 2020. [online]. Disponível em: <https://canaltech.com.br/apps/5-aplicativos-ajudar-deficientes-visuais/>. [Consultado em 03/12/2022].