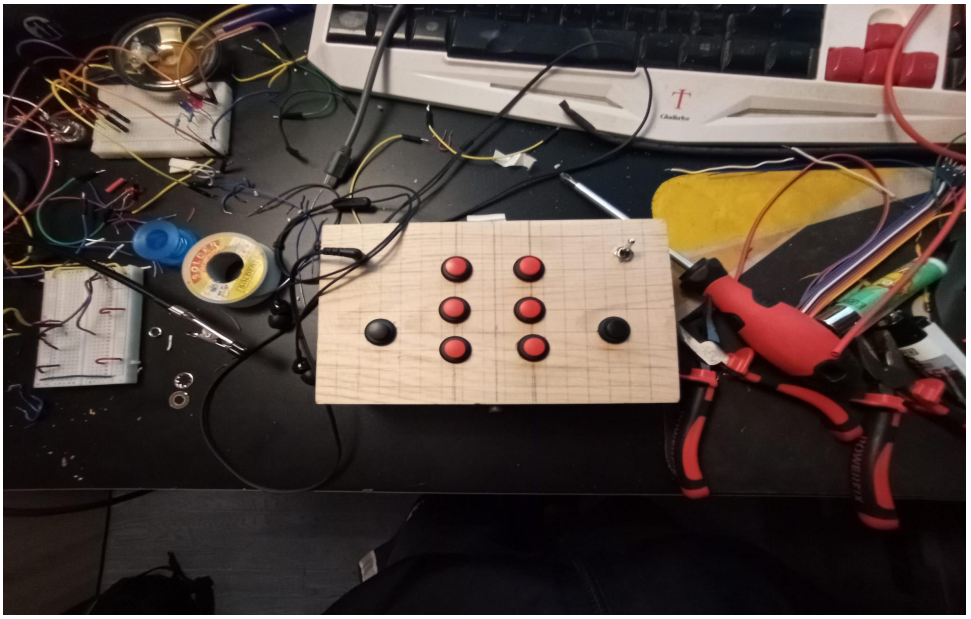
BRAILLER



Igor Dumbrava
S21ME
TFG

Índice_____Página 1

1- **Resumen** —---------- Página 2
2- **Justificación del trabajo** ---------- Página 3
3- **Objetivos** ---------- Página 3
3.1 - Usos diarios ---------- Página 3

4- **Diseño** ---------- Página 3
4.1 - Diseño de carcasa ---------- Página 3
4.2 - Elección de material ---------- Página 3

5- **Componentes y Arduino(desarrollar)**---------- Página 4
6- **Presupuesto** ---------- Página 10
6.1 - Coste de materiales necesarios---------- Página 10
6.2 - Precio final del prototipo---------- Página 10

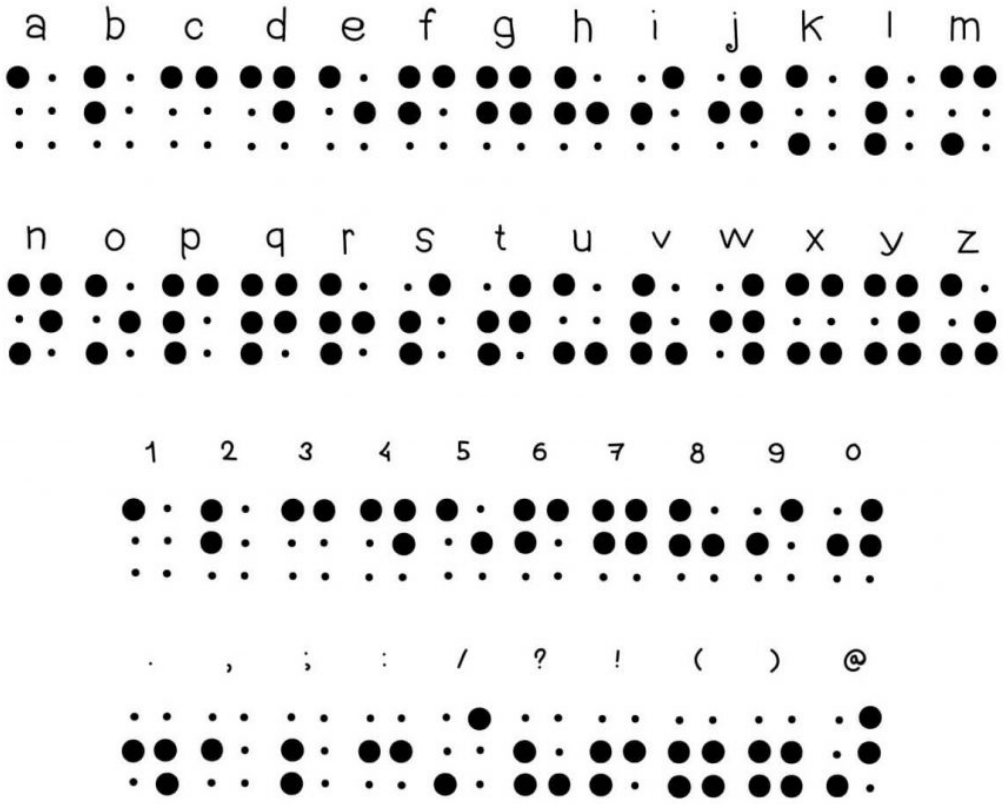
7- **Esquema lógico y diseño de Circuito**---------- Página 11
7.1 - Bloque de Alimentación---------- Página 11
7.2 - Bloque de control---------- Página 11

8- **Diagrama de flujo** ---------- Página 12
6.1 - Reproducción de letra---------- Página 12
6.2 - Ejecución modo juego---------- Página 12

9- **Código de programación (**lenguaje C**)** ---------- Página 13
10- **AUDIOS UTILIZADOS Y TRATAMIENTOS** ---------- Página 24
11- **Desarrollo y proceso de fabricación**---------- Página 25
11.1 - Pruebas de arduino con botones, led, altavoz, etc…-------- Página 25
11.2 - Pruebas con reproductor de tarjeta SD---------- Página 25
11.3 - Ensamble de simulación con botones---------- Página 26
11.4 - Primeras pruebas con letras---------- Página 26
11.5 - Pruebas con números---------- Página 26
11.6 - Ejecución modo juego y Reproducción de letras---------- Página 26
11.7 - Carcasa y diseño ---------- Página 27
11.8 - Implementación de código morse ---------- Página 28
11.7 - Ajuste de la carcasa para mejorar el agarre----------Página 28

12- **Conclusiones** ---------- Página 29
13- **Líneas de investigación futuras** ---------- Página 29
14- **Webgrafía** —----------Página 29

1. Hola, quiero presentarte una idea y un proyecto impulsado por el propósito de ayudar a personas. Puede sonar típico pero considero que es una fuente de inspiración para mi.
La idea es tener una herramienta para ayudar a aprender BRAILLE y hacerlo de manera entretenida para los niños. La manera que se decide implementar es un juego que he llamado Brailler, el inicio de la idea viene inspirado de un muñeco “Braillin” en el año 2004 por una profesora Argentina que ayuda a niños con discapacidad a aprender y gracias a su idea los niños que padecen enfermedades relacionadas con la vista pueden ir aprendiendo las letras .
El braille es un código de 6 puntos para expresar letras y permitir leer, y transmitir información de manera que pueda interpretarse por personas con discapacidad visual.
Tal y como lo he visto en una visita a mi clase por parte de unas maestras que se dedican a ayudar a personas con discapacidad visual y ya desde ese momento supe que mi Trabajo final de grado sería algo relacionado con este tema en particular.
Mi idea era que este juega sea lo más simple posible en lo que a botones respecta y que se pueda usar sin ningún tipo de conocimientos previos, para que se pueda usar, transportar, que sea cómodo, ligero y divertido.
Para entender un poco como funciona el braille voy a presentar a continuación la codificación y cómo se organiza cada letra y número .



2. Este trabajo es solo un paso más para concluir nuestro paso por el grado superior de mantenimiento electrónico, y se que a más de uno se le ha ocurrido llevarse algo de recuerdo de esas aulas, algunos pensaron en crear algo que podrán quedarse después como uso personal o de recuerdo, yo personalmente quería llevarme un buen recuerdo de este último año estudiado, por eso la oportunidad vista relacionada con hacer este proyecto no se me ha podido escapar de ninguna manera, realizo mi trabajo final y de paso ayudo a personas a aprender, es un buen recuerdo, y yo hago por fin algo relacionado con el tema de lenguaje.

3. Los principales objetivos de este proyecto es que mediante los conocimientos adquiridos durante el grado de mantenimiento electrónico se desarrolle un prototipo que funcione y que realice algo con alguna finalidad, y los principales objetivos de esta idea es primero facilitar el trabajo de las maestras de estos niños, y segundo para que las personas con discapacidad visual puedan aprender por su cuenta y de manera muy dinámica el BRAILLE. Lo mejor de este juego es que el propio jugador sabe cuando ha acertado y cuando ha fallado sin necesidad de ninguna ayuda y gracias a la elección del diseño lo más minimalista posible es muy intuitivo, el jugador tiende a probar los botones de manera que entiende el funcionamiento al instante.

Uno de los principales objetivos de este prototipo es que sea cómodo, simple y de un diseño resistente, que sea fácil de usar y fácil de transportar. El diseño debe ir enfocado principalmente para las manos pequeñas de personas de poca edad, por eso se revisan maneras de adaptar el diseño a esas necesidades.

4. El diseño del juego tiene que tener en cuenta varias cosas. Panel de botones para las combinaciones de 6 puntos, botón de juego y de carga, espacio para circuito, arduino y espacio para un altavoz.

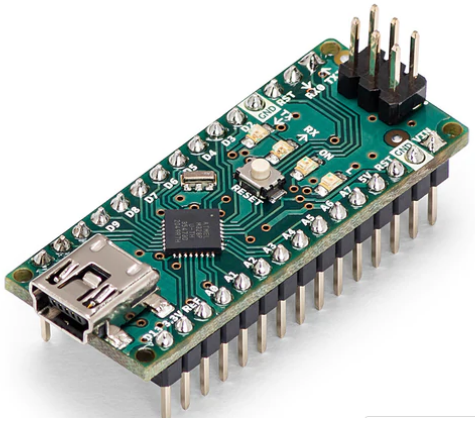
Como podemos ver hay bastantes cosas a tener en cuenta, lo primero que se ha barajado era la posibilidad de diseñar una carcasa con una impresora 3D pero para eso necesitaba el diseño ya definitivo cosa que aún no tenía, además de los conocimientos necesarios para diseñar eso. Tras ver diferentes posibilidades he decidido ir por lo económico y he comprado una caja de madera para usarla de carcasa y adaptar el circuito y los bloques del juego a ese espacio. Mi conclusión fue que ha sido una muy buena decisión principalmente porque yo no estoy haciendo algo que se exactamente como voy a llevarlo a cabo, lo que hago es ir avanzando y mejorando el proyecto a base de probar y equivocarme, y tener una carcasa moldeable me permite introducir mejoras como nuevos agujeros para añadirle un conector miniJack hembra o poder adaptar el diseño para poder cargar externamente la batería.

Como elección para el material las opciones que tenía son plástico o madera y la verdad es que el tacto de la madera es bastante agradable por lo que es la opción elegida.

5. *La lista de los componentes necesarios para la realización de este proyecto son:*

-Arduino nano

Arduino Nano es una placa pequeña, completa y compatible con placas de prueba basada en ATmega328 (Arduino Nano 3.x). Tiene más o menos la misma funcionalidad del Arduino Duemilanove, pero en un paquete diferente. Solo carece de un conector de alimentación de CC y funciona con un cable USB Mini-B en lugar de uno estándar.



-Altavoz 8 Ohmnios

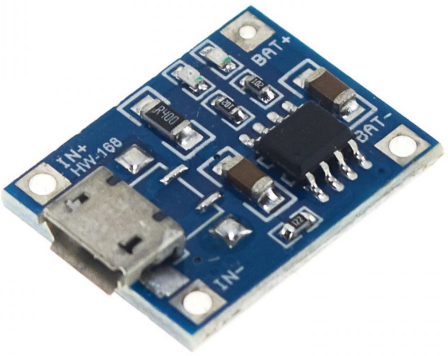
Se trata de unos altavoces de muy baja potencia y de muy bajo consumo, se suelen utilizar para aplicaciones del ámbito de la electrónica y circuitos donde se necesita audio

-Convertidor DC/DC (5V)

Módulo regulador de voltaje de 5V, 3A, USB, convertidor de DC-DC, cargador de teléfono, módulo de fuente de alimentación.

-Cargador de batería TP4056

El Módulo Cargador USB de batería litio 18650 1A permite cargar de forma sencilla baterías LiPo/LiOn de una celda. Funciona con baterías con voltajes de carga nominal de 3.7V y de carga completa a 4.2V. Ideal para baterías LiPo 18650 o baterías de LiOn de smartphones. La corriente de carga es de 1A, este valor de corriente puede ser modificado al cambiar la resistencia R3 según se muestra en el datasheet.

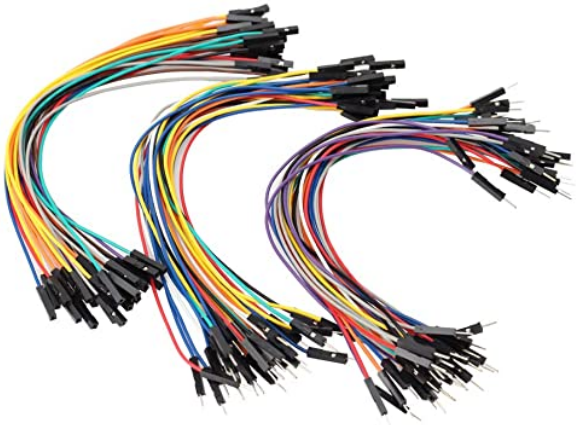


-Interruptor

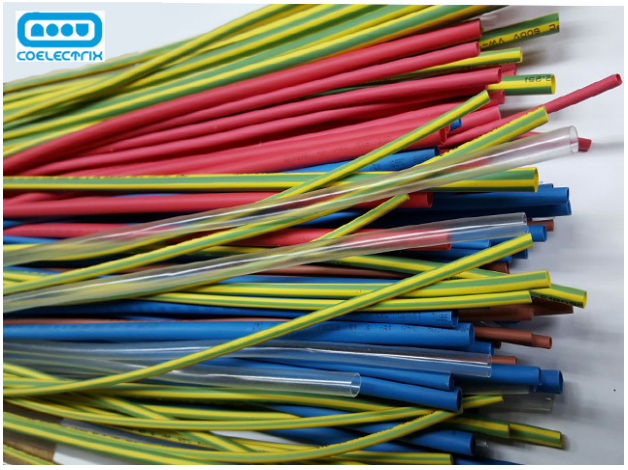
Lo que nos permitirá encender y apagar el equipo en el momento que queramos.

-Cable micro-usb

-Cables



-Termoretractil



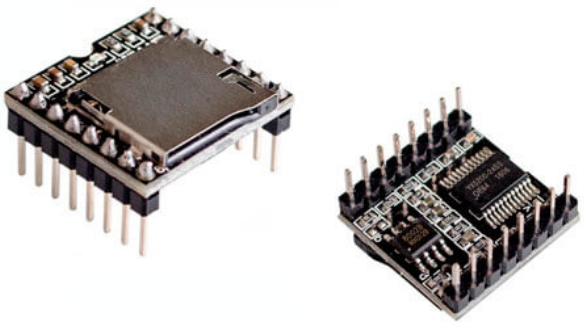
El termoretractil es un muy buen aliado de un técnico a la hora de trabajar, permite aislar los cables que consideremos precisos y además se usa de un modo muy fácil, se pasa el cable que se va a soldar y después de realizar la soldadura se posiciona el termoretractil encima de la soldadura, se aplica calor con un mechero y el termoretractil se contrae y se reduce hasta llegar al grosor del cable o de la soldadura que aisla.

-Resistencias 1K 5 unidades

Para este proyecto se necesitan una serie de resistencias y las de 1k son las apropiadas para el trabajo a realizar. Se usarán para configuraciones pull-up y para los leds que se usarán.

-DFPlayer Mini

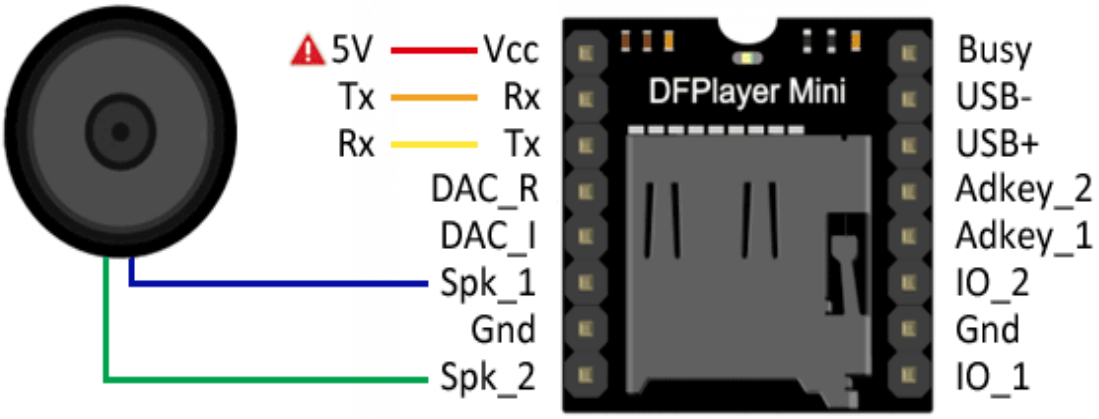
El DFPlayer Mini es un reproductor de audio de bajo coste y pequeño que podemos conectar a un procesador como Arduino para reproducir audio en formato MP3.



Arduino no tiene potencia potente para reproducir un archivo comprimido como un MP3. Por mucho que lo intentéis, o incluso si empleáis un procesador más potente, no es función de un autómata realizar estas tareas. Es mucho mejor delegar en un subprocesador específico.

Existen distintas opciones para reproducir ficheros MP3 desde Arduino, como diversos shields y placas más o menos caras. Entre las distintas opciones disponibles los módulos DFPlayer Mini han adquirido una gran popularidad por su bajo precio y grandes características.

Su esquema de conexión es el siguiente:



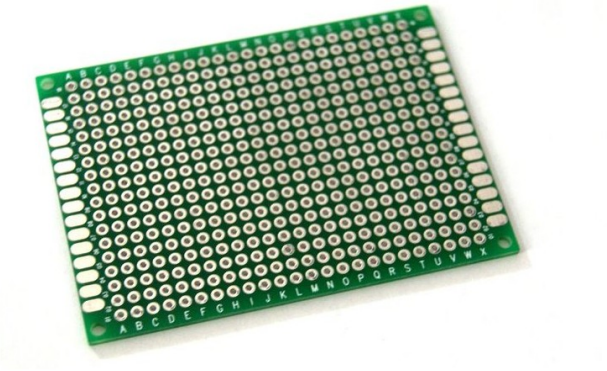
-Botones y pulsadores

-Batería de litio 1200mAh

Batería de Litio recargable de reducido tamaño ideal para este tipo de configuraciones con arduino.

-Placa perforada

Esta placa perforada es la herramienta perfecta para crear circuitos propios sin necesidad de usar empalmes o protoboard, estas placas se pueden cortar para la medida necesaria, se pueden realizar soldaduras y es una buena alternativa a usar placas de circuito impreso.



-Conector miniJack hembra

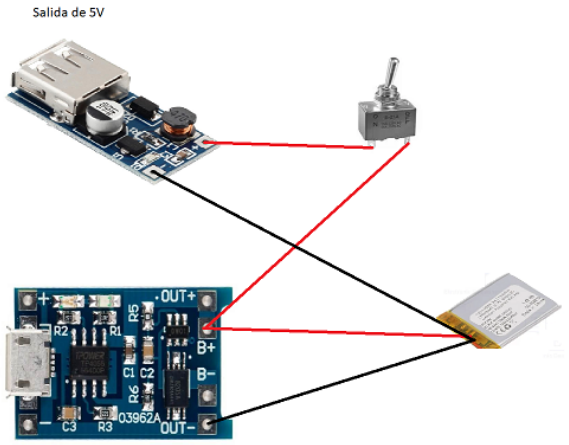
-Tarjeta SD

6. Presupuesto.

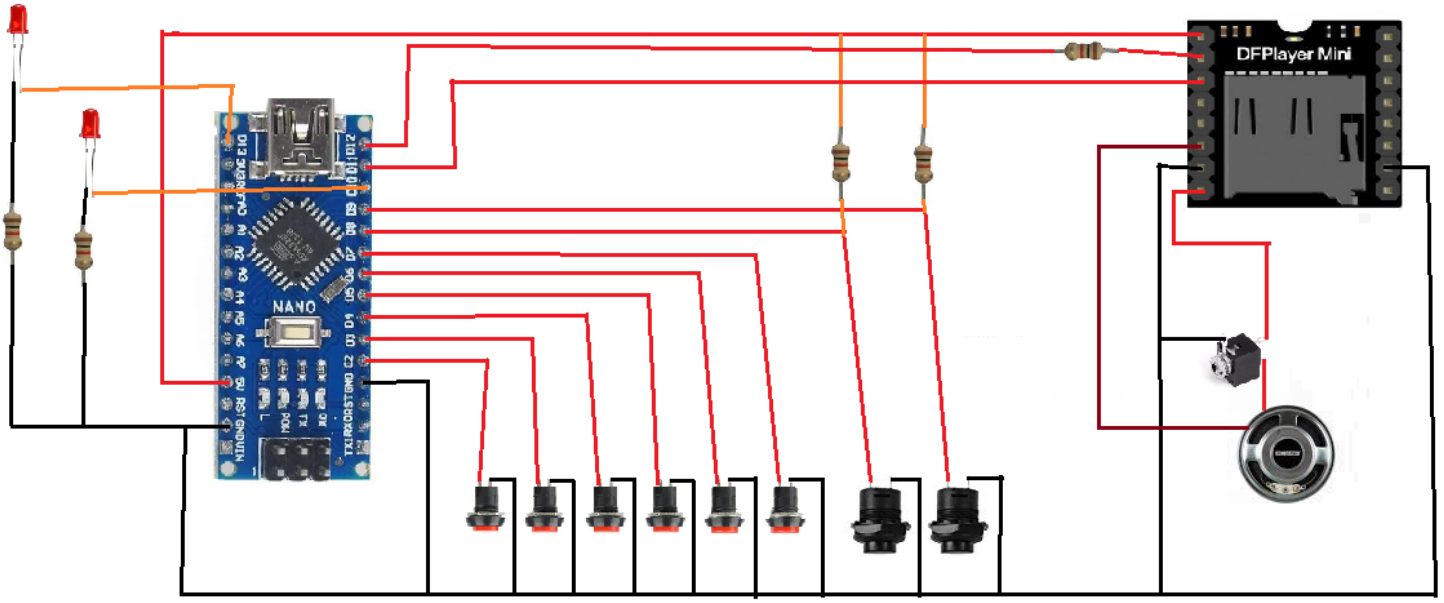
Materiales	Precio aproximado	Coste final
Arduino nano	9€	suministrado por el centro
Altavoz 8 Ohm	2€	suministrado por el centro
Convertidor DC/DC	2€	suministrado por el centro
Cargador de batería TP4056	1€	suministrado por el centro
Interruptor	1€	1€
Cable micro-usb	3€	suministrado por el centro
Cables	3€	3€
Termoretractil	1€	1€
Resistencias 1K 5 unidades	1€	1€
DFPlayer Mini	3€	suministrado por el centro
Botones	4€	suministrado por el centro
Batería de litio 1200mAh	5€	suministrado por el centro
Placa perforada	1€	suministrado por el centro
Conector miniJack hembra (mono)	2€	2€
Tarjeta SD	5€	suministrado por el centro
Carcasa	3€	3€
TOTAL	45€	11€

7.

Bloque de Alimentación



Bloque de control



8. Diagramas de flujo

Diagrama de ejecución

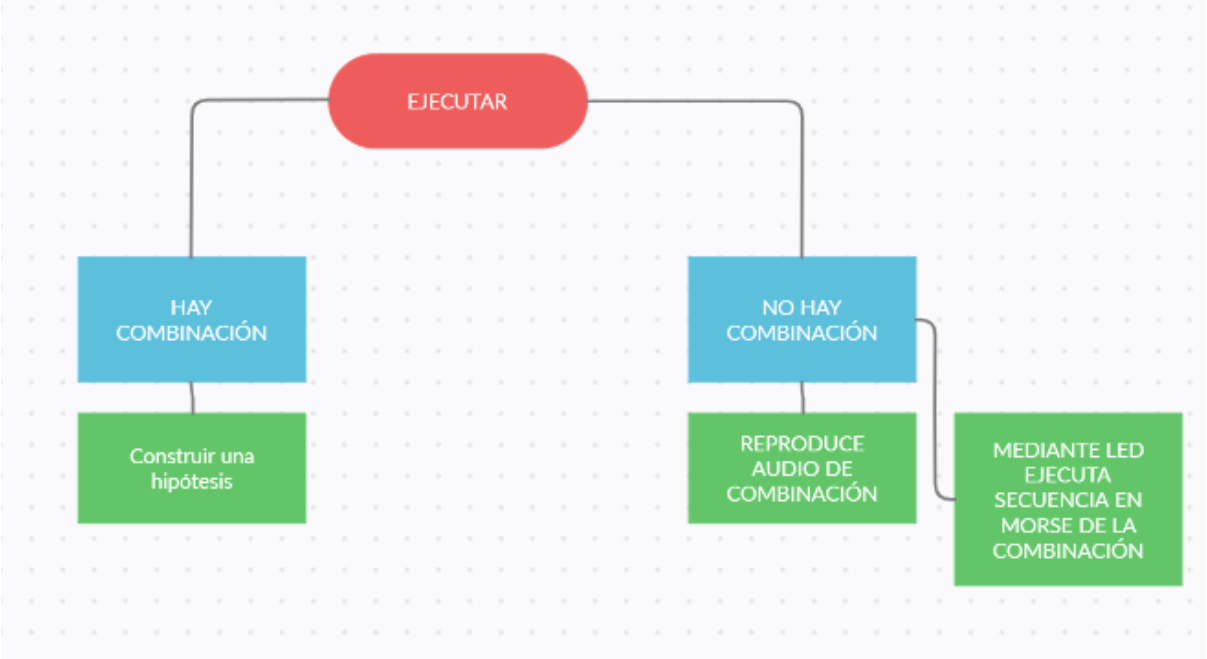
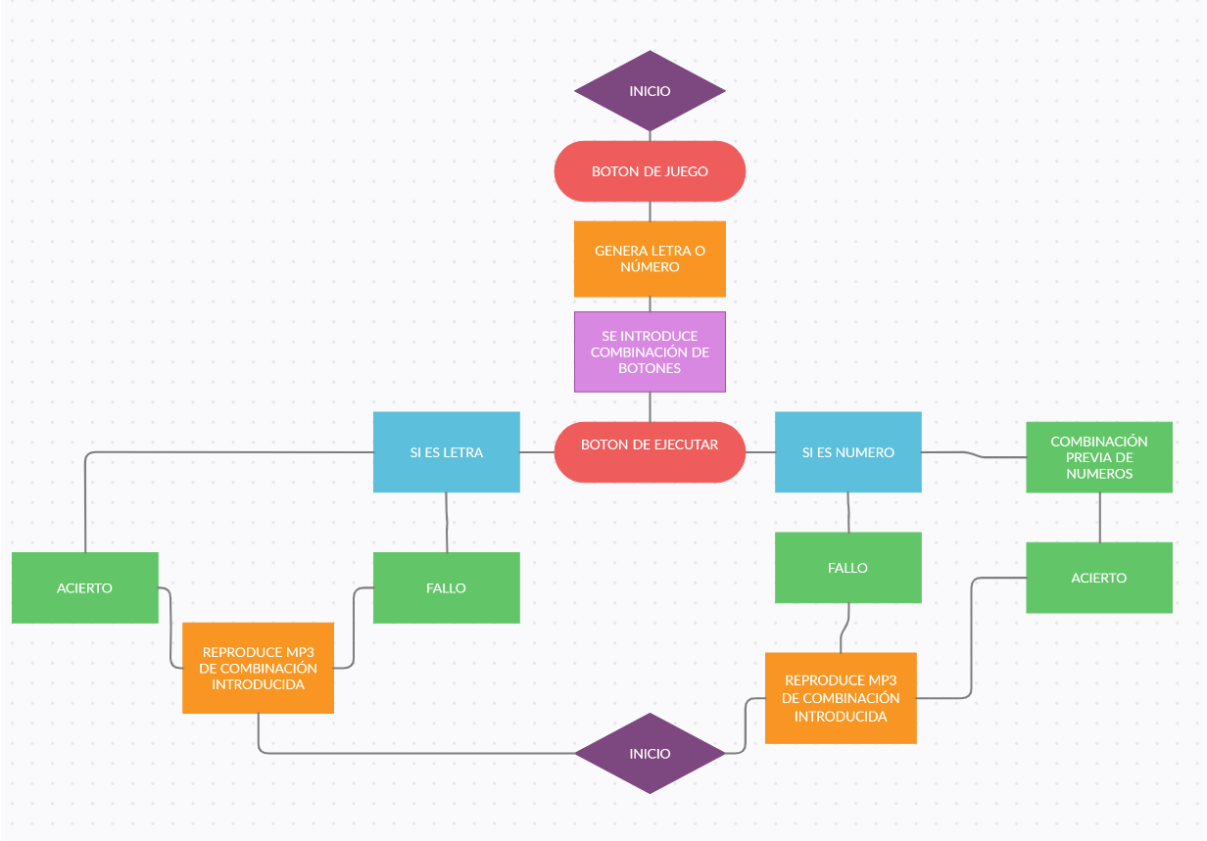


Diagrama de modo juego



9. Código de programación

```
#include "Arduino.h "
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"
```

**SE INTRODUCEN LAS LIBRERÍAS NECESARIAS PARA EL PROGRAMA

```
//Pulsadores
const int s1 = 2; // pulsadores
const int s2 = 3;
const int s3 = 4;
const int s4 = 5;
const int s5 = 6;
const int s6 = 7;
const int resuelve = 8;
const int aleatorio = 9;

const int led1 =  10;
const int led2 =  13;

int boton_aleatorio = 0;
int resolviendo = 0;
int boton1 = 0;
int boton2 = 0;
int boton3 = 0;
int boton4 = 0;
int boton5 = 0;
int boton6 = 0;
int numL = 0;
int digito = 0;
int randomNumber = 0;
```

**SE DECLARAN TODAS LAS VARIABLES NECESARIAS EN EL PROGRAMA

```
SoftwareSerial mySoftwareSerial(10, 11); // RX, TX
DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);

boolean Juego = false;
boolean Num = false;
boolean Pal = false;
```

**SE DECLARAN VARIABLES TIPO BOOLEANO PARA INTRODUCIR DIFERENTES MODOS DE FUNCIONAMINTO

```
void setup()
```

**VOID SETUP DONDE SE REALIZAN TODAS LAC CONFIGURACIONES DEL FUNCIONAMIENTO DEL PROGRAMA

```
{
 mySoftwareSerial.begin(9600);
 Serial.begin(115200);
 if (!myDFPlayer.begin(mySoftwareSerial)) { //Use softwareSerial to communicate with
mp3.
 }
 myDFPlayer.setTimeOut(500); //Set serial communictaion time out 500ms
 myDFPlayer.volume(15); //Set volume value (0~30).  //----Set volume----
 myDFPlayer.volumeUp(); //Volume Up
 myDFPlayer.volumeDown(); //Volume Down
 //----Set different EQ----
 myDFPlayer.EQ(DFPLAYER_EQ_NORMAL);
 //----Set device we use SD as default----
```

**SE ACABA DE REALIZAR LA CONFIGURACIÓN DEL REPRODUCTOR DE LA TARJETA SD.

```
 pinMode(s1, INPUT_PULLUP);
 pinMode(s2, INPUT_PULLUP);
 pinMode(s3, INPUT_PULLUP);
 pinMode(s4, INPUT_PULLUP);
 pinMode(s5, INPUT_PULLUP);
 pinMode(s6, INPUT_PULLUP);
```

**SE DECLARAN LAS ENTRADAS DIGITALES USADAS EN EL JUEGO, GRACIAS A ESTA CONFIGURACIÓN PULL-UP NO HACE FALTA QUE CADA UNO DE LOS BOTONES TENGA UNA RESISTENCIA.

```
 void punto()  **FUNCIÓN PARA REPRESENTAR EL PUNTO EN EL MORSE.
 {
  digitalWrite(led2, HIGH);
  delay(150);
  digitalWrite(led2, LOW);
  delay(150);
}
```

```
void raya() **FUNCIÓN PARA REPRESENTAR LA RAYA EN MORSE
{
 digitalWrite(led2, HIGH);
 delay(450);
 digitalWrite(led2, LOW);
 delay(150);
}


void morse(int num) ** FUNCIÓN QUE EJECUTA EL CÓDIGO MORSE
{
 if (num == 1)
 {punto(); raya();} //A
 if (num == 2)
 {raya(); punto(); punto(); punto();} //B
 if (num == 3)
 {raya(); punto(); raya(); punto();} //C
 if (num == 4)
 {raya(); punto(); punto(); } //D
 if (num == 5)
 {punto();} //E
 if (num == 6)
 {punto(); punto(); raya(); punto();} //F
 if (num == 7)
 {raya(); raya(); punto();} //G
 if (num == 8)
 {punto(); punto(); punto(); punto();} //H
 if (num == 9)
 {punto(); punto();} //I
 if (num == 10)
 {punto(); raya(); raya(); raya();} //J
 if (num == 11)
 {raya(); punto(); raya();} //K
 if (num == 12)
 {punto(); raya(); punto(); punto();} //L
 if (num == 13)
 {raya(); raya();} //M
 if (num == 14)
 {raya(); punto();} //N
 if (num == 15)
 {raya(); raya(); raya(); } //O
 if (num == 16)
 {punto(); raya(); raya(); punto();} //P
 if (num == 17)
```

```
 {raya(); raya(); punto(); raya(); } //Q
 if (num == 18)
 {punto(); raya(); punto();} //R
 if (num == 19)
 {punto(); punto(); punto();} //S
 if (num == 20)
 {raya(); } //T
 if (num == 21)
 {punto(); punto(); raya(); } //U
 if (num == 22)
 {punto(); punto(); punto(); raya(); } //V
 if (num == 23)
 {punto(); punto(); raya(); } //W
 if (num == 24)
 {raya(); punto(); punto(); raya(); } //X
 if (num == 25)
 {raya(); punto(); raya(); raya(); } //Y
 if (num == 26)
 {raya(); raya(); punto(); punto();} //Z
 if (num == 27)
 {punto(); raya(); raya(); raya(); raya(); } //1
 if (num == 28)
 {punto(); punto(); raya(); raya(); raya(); } //2
 if (num == 29)
 {punto(); punto(); punto(); raya(); raya(); } //3
 if (num == 30)
 {punto(); punto(); punto(); punto(); raya(); } //4
 if (num == 31)
 {punto(); punto(); punto(); punto(); punto(); } //5
 if (num == 32)
 {raya(); punto(); punto(); punto(); punto(); } //6
 if (num == 33)
 {raya(); raya(); punto(); punto(); punto(); } //7
 if (num == 34)
 {raya(); raya(); raya(); punto(); punto(); } //8
 if (num == 35)
 {raya(); raya(); raya(); raya(); punto(); } //9

}

 int numeroLetra() **FUNCIÓN QUE LE ASIGNA UN NÚMERO A CADA UNA DE LAS
COMBINACIONES
{
```

```
 if (Num == 0 && boton1 == HIGH && boton2 == LOW && boton3 == LOW && boton4
== LOW && boton5 == LOW && boton6 == LOW)
 {
 return 1; //A
 }
 else if (Num == 0 && boton1 == HIGH && boton2 == HIGH && boton3 == LOW &&
boton4 == LOW && boton5 == LOW && boton6 == LOW)
 {
 return 2; //B
 }
 else if (Num == 0 && boton1 == HIGH && boton2 == LOW && boton3 == LOW &&
boton4 == HIGH && boton5 == LOW && boton6 == LOW)
 {
 return 3; //C
 }
 else if (Num == 0 && boton1 == HIGH && boton2 == LOW && boton3 == LOW &&
boton4 == HIGH && boton5 == HIGH && boton6 == LOW)
 {
 return 4; //D
 }
 else if (Num == 0 && boton1 == HIGH && boton2 == LOW && boton3 == LOW &&
boton4 == LOW && boton5 == HIGH && boton6 == LOW)
 {
 return 5; //E
 }
 else if (Num == 0 && boton1 == HIGH && boton2 == HIGH && boton3 == LOW &&
boton4 == HIGH && boton5 ==LOW && boton6 == LOW)
 {
 return 6; //F
 }
 else if (Num == 0 && boton1 == HIGH && boton2 == HIGH && boton3 == LOW &&
boton4 == HIGH && boton5 ==HIGH && boton6 == LOW)
 {
 return 7; //G
 }
 else if (Num == 0 && boton1 == HIGH && boton2 == HIGH && boton3 == LOW &&
boton4 == LOW && boton5 == HIGH && boton6 == LOW)
 {
 return 8; //H
 }
 else if (Num == 0 && boton1 == LOW && boton2 == HIGH && boton3 == LOW &&
boton4 == HIGH && boton5 == LOW && boton6 == LOW)
 {
 return 9; //I
```

```
 }
 else if (boton1 == LOW && boton2 == HIGH && boton3 == LOW && boton4 == HIGH
&& boton5 == HIGH && boton6 == LOW)
 {
 return 10; //J
 }
 else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == LOW
&& boton5 == LOW && boton6 == LOW)
 {
 return 11; //K
 }
 else if (boton1 == HIGH && boton2 == HIGH && boton3 == HIGH && boton4 == LOW
&& boton5 ==LOW && boton6 == LOW)
 {
 return 12; //L
 }
 else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == HIGH
&& boton5 ==LOW && boton6 == LOW)
 {
 return 13; //M
 }
 else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == HIGH
&& boton5 ==HIGH && boton6 == LOW)
 {
 return 14; //N
 }
 else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == LOW
&& boton5 == HIGH && boton6 == LOW)
 {
 return 15; //O
 }
 else if (boton1 == HIGH && boton2 == HIGH && boton3 == HIGH && boton4 == HIGH
&& boton5 == LOW && boton6 == LOW)
 {
 return 16; //P
 }
 else if (boton1 == HIGH && boton2 == HIGH && boton3 == HIGH && boton4 == HIGH
&& boton5 == HIGH && boton6 == LOW)
 {
 return 17; //Q
 }
 else if (boton1 == HIGH && boton2 == HIGH && boton3 == HIGH && boton4 == LOW
&& boton5 ==HIGH && boton6 == LOW)
 {
```

```
 return 18; //R
 }
 else if (boton1 == LOW && boton2 == HIGH && boton3 == HIGH && boton4 == HIGH
&& boton5 ==LOW && boton6 == LOW)
 {
 return 19; //S
 }
 else if (boton1 == LOW && boton2 == HIGH && boton3 == HIGH && boton4 == HIGH
&& boton5 ==HIGH && boton6 == LOW)
 {
 return 20; //T
 }
 else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == LOW
&& boton5 == LOW && boton6 == HIGH)
 {
 return 21; //U
 }
 else if (boton1 == HIGH && boton2 == HIGH && boton3 == HIGH && boton4 == LOW
&& boton5 ==LOW && boton6 == HIGH)
 {
 return 22; //V
 }
 else if (boton1 == LOW && boton2 == HIGH && boton3 == LOW && boton4 == HIGH
&& boton5 == HIGH && boton6 == HIGH)
 {
 return 23; //W
 }
 else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == HIGH
&& boton5 ==LOW && boton6 == HIGH)
 {
 return 24; //X
 }
 else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == HIGH
&& boton5 ==HIGH && boton6 == HIGH)
 {
 return 25; //Y
 }
 else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == LOW
&& boton5 == HIGH && boton6 == HIGH )
 {
 return 26; //Z
 }
 else if (Num == 1 && boton1 == HIGH && boton2 == LOW && boton3 == LOW &&
boton4 == LOW && boton5 == LOW && boton6 == LOW)
```

```
 {
 return 27; //1
 }
 else if (Num == 1 && boton1 == HIGH && boton2 == HIGH && boton3 == LOW &&
boton4 == LOW && boton5 == LOW && boton6 == LOW)
 {
 return 28; //2
 }
 else if (Num == 1 && boton1 == HIGH && boton2 == LOW && boton3 == LOW &&
boton4 == HIGH && boton5 == LOW && boton6 == LOW)
 {
 return 29; //3
 }
 else if (Num == 1 && boton1 == HIGH && boton2 == LOW && boton3 == LOW &&
boton4 == HIGH && boton5 == HIGH && boton6 == LOW)
 {
 return 30; //4
 }
 else if (Num == 1 && boton1 == HIGH && boton2 == LOW && boton3 == LOW &&
boton4 == LOW && boton5 == HIGH && boton6 == LOW)
 {
 return 31; //5
 }
 else if (Num == 1 && boton1 == HIGH && boton2 == HIGH && boton3 == LOW &&
boton4 == HIGH && boton5 == LOW && boton6 == LOW)
 {
 return 32; //6
 }
 else if (Num == 1 && boton1 == HIGH && boton2 == HIGH && boton3 == LOW &&
boton4 == HIGH && boton5 == HIGH && boton6 == LOW)
 {
 return 33; //7
 }
 else if (Num == 1 && boton1 == HIGH && boton2 == HIGH && boton3 == LOW &&
boton4 == LOW && boton5 == HIGH && boton6 == LOW)
 {
 return 34; //8
 }
 else if (Num == 1 && boton1 == LOW && boton2 == HIGH && boton3 == LOW &&
boton4 == HIGH && boton5 == LOW && boton6 == LOW)
 {
 return 35; //9
 }
  else
```

```
{
return 0;
}

}

void loop() **EJECUCIÓN DEL PROGRAMA
{
boton1 = digitalRead(s1);
boton2 = digitalRead(s2);
boton3 = digitalRead(s3);
boton4 = digitalRead(s4);
boton5 = digitalRead(s5);
boton6 = digitalRead(s6);
resolviendo = digitalRead(resuelve);
boton_aleatorio = digitalRead(aleatorio);

**SE ACABA DE DECLARAR QUE LOS BOTONES SON IGUALES A LAS
VARIABLES "s1, s2, s3…"

if (resolviendo == LOW && Juego == 0) {

**FUNCIÓN DE PULSACIÓN DE BOTÓN DE RESOLVER

Pal = 0;
Juego = 0;
numL = numeroLetra();
if ( numL != 0 ) {
myDFPlayer.play(numL); // Suena Letra indicada
morse(numL);
delay(1000); // descansa 1 segundo.
}
}

if( boton_aleatorio == LOW) {

**FUNCIÓN QUE EJECUTA EL MODO JUEGO PIDIENDO UN VALOR ALEATORIO

Juego = 1;
  digitalWrite(led1, HIGH);
  delay(500);
  digitalWrite(led1, LOW);

randomNumber = random(1,35);  **ESCOGE UN VALOR ALEATORIO
```

```
myDFPlayer.play(randomNumber);
delay(1000); // descansa 1 segundo.
}


//EMPIEZA EL BUCLE
```

** EMPIEZA EL BUCLE QUE COMPRUEBA SI LA COMBINACIÓN ES IGUAL AL VALOR DE randomNumber

```
if (resolviendo == LOW && Juego == 1){
numL = numeroLetra();

if( numL == randomNumber ){
 myDFPlayer.play(numL);
 morse(numL);
 digitalWrite(led1, HIGH);
 delay(1000);

 Pal = 0;
 Juego = 0;
 digitalWrite(led1, LOW);
}
```

**EMPIEZA EL BUCLE QUE SIRVE PARA ENTRAR A MODO NUMERO DADO QUE PARA REPRESENTAR NUMEROS HAY QUE USAR COMBINACIONES DE ALGUNAS LETRAS, POR LO TANTO PRIMERO SE INTRODUCE LA COMBINACIÓN QUE AVISA QUE LA SIGUIENTE SERÁ UN NUMERO.

```
else if (boton1 == LOW && boton2 == LOW && boton3 == HIGH && boton4 == HIGH
&& boton5 == HIGH && boton6 == HIGH){
 Num = 1;
 Pal = 0;
 digitalWrite(led1, HIGH);
 delay(300);
 digitalWrite(led1, LOW);
 delay(300);
```

**A CONTINUACIÓN EMPIEZA EL BUCLE QUE VERIFICA SI EL NÚMERO DE LA COMBINACIÓN CONCUERDA CON EL VALOR DADO POR EL JUEGO,

```
if (resolviendo == LOW && Juego == 1){
numL = numeroLetra();

if( numL == randomNumber ){
 myDFPlayer.play(numL);
 digitalWrite(led1, HIGH);
 delay(1000);

 Num = 0;
 Juego = 0;
 numL = 0;
 digitalWrite(led2, LOW);
 }
 else{ Juego = 0;}
 }}
```

**SI NO CONCUERDAN LOS VALORES ENTONCES EL PROGRAMA MANDA UNA SEÑAL DE AUDIO Y VUELVE AL ESTADO INICIAL.

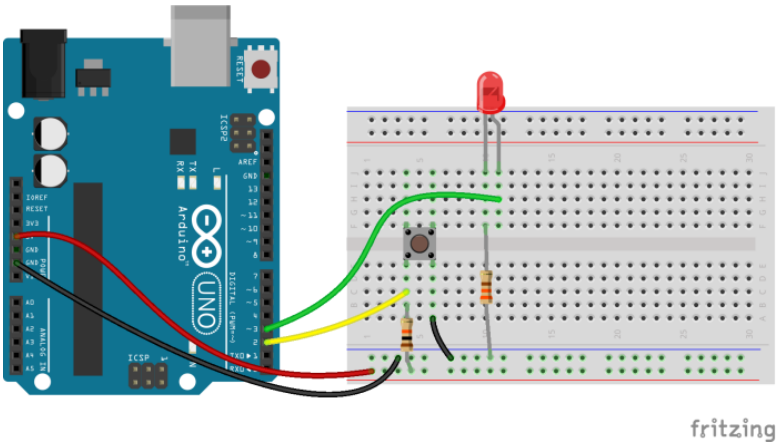
```
else {
Pal = 0;
myDFPlayer.play(49);
delay(1000);
myDFPlayer.play(numL);

delay(1000);
 numL = 0;
Num = 0;
Juego = 0;
  digitalWrite(led2, LOW);
}
}
}
```

10.

- Los audios que se han utilizado son de formato MP3.
- Tanto las letras como los números tienen un audio que le corresponde.
- Los audios se han extraído de youtube y de reproductores de textos como loquendo.
- Los audios se han tratado en el DAW FL Studio 20 y se le ha aplicado una compresión en -9 dB para asegurar la seguridad de los usuarios del BRAILLER.

- 11. Proceso de desarrollo

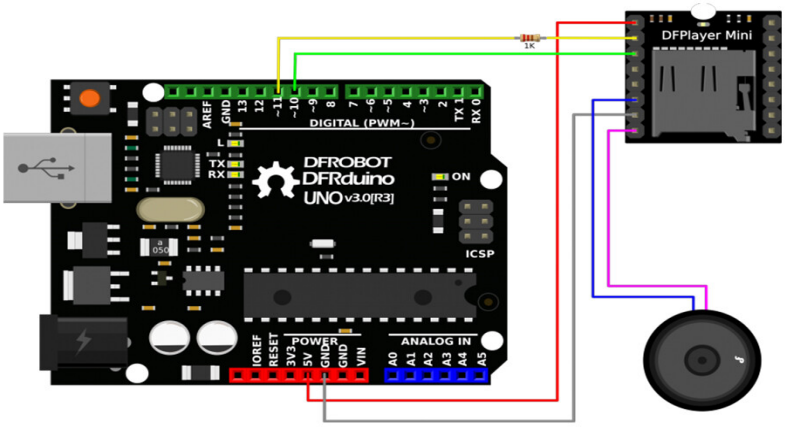
Dado que no sabía muy bien por dónde empezar decidí empezar rompiendo un poco el hielo, cogí el arduino y empecé a probar códigos sencillos con botones y leds, y poco a poco iba incrementando el número de botones y maneras de combinarlos.

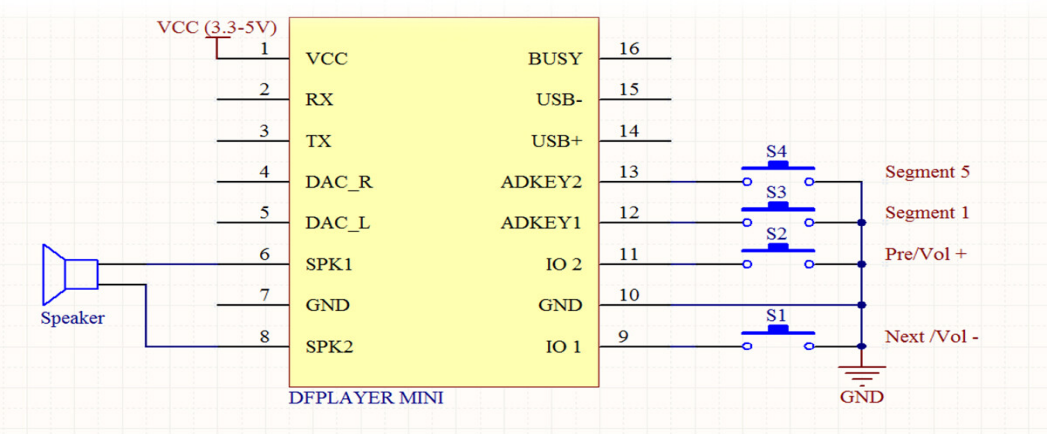


Como segundo paso he probado configuraciones para usar un DFPlayer Mini, usando una SD en la cual introducido audios, estos audios llevan su nombre pero para que se puedan ejecutar necesita añadirle al principio del nombre un código de 4 dígitos.
Ejemplo:

Nombre de archivo: “Nombre1, Nombre 2”

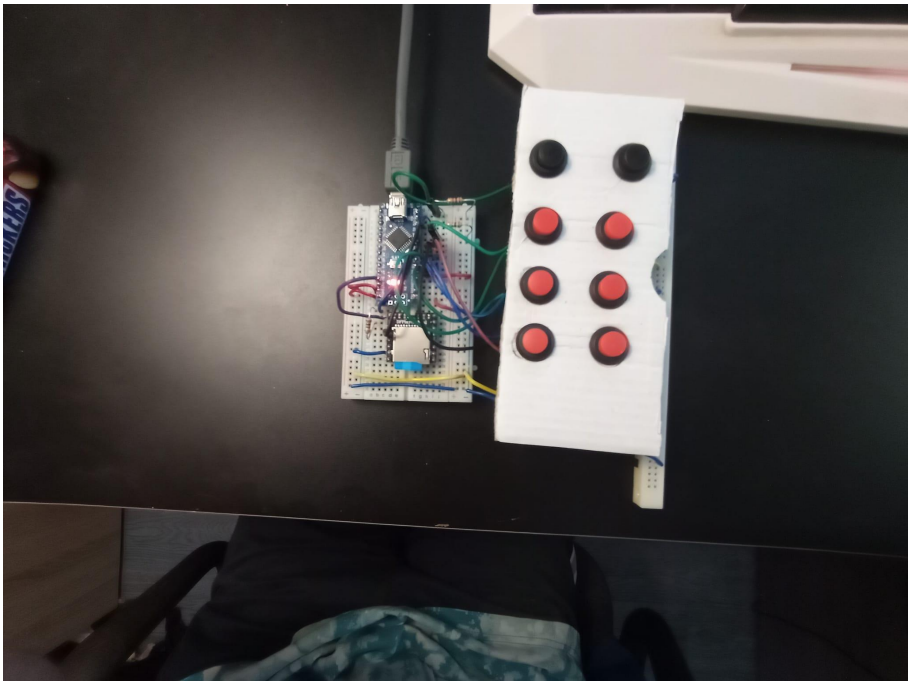
Nombre de archivos en la SD: “0001Nombre1, 0002Nombre 2” (y así sucesivamente)



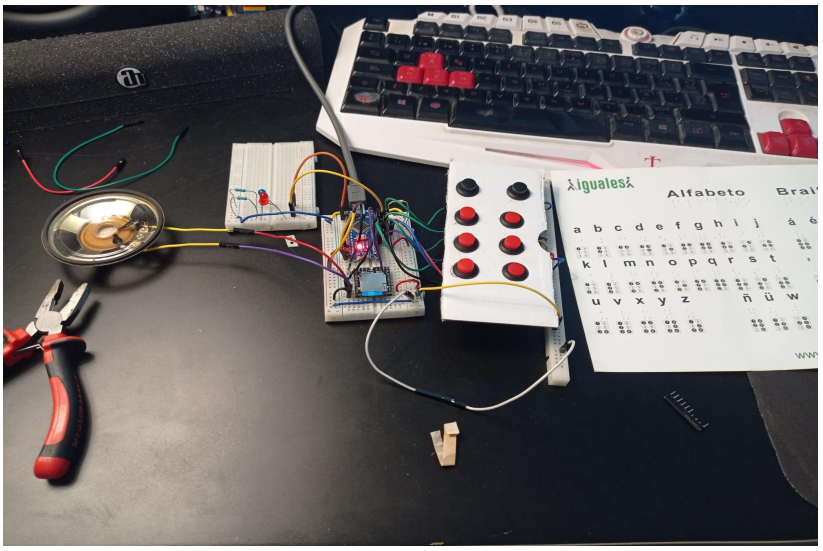


En el siguiente paso se ha montado la simulación de lo que al final sería el panel de botones, esto se hace para tener ya una especie de interfaz y tener las cosas de manera un poco más ordenada

Una vez montado el panel se han hecho pruebas de programación con combinaciones de botones, y su ejecución. Se prueban las primeras letras para asegurar que el diseño es adecuado.

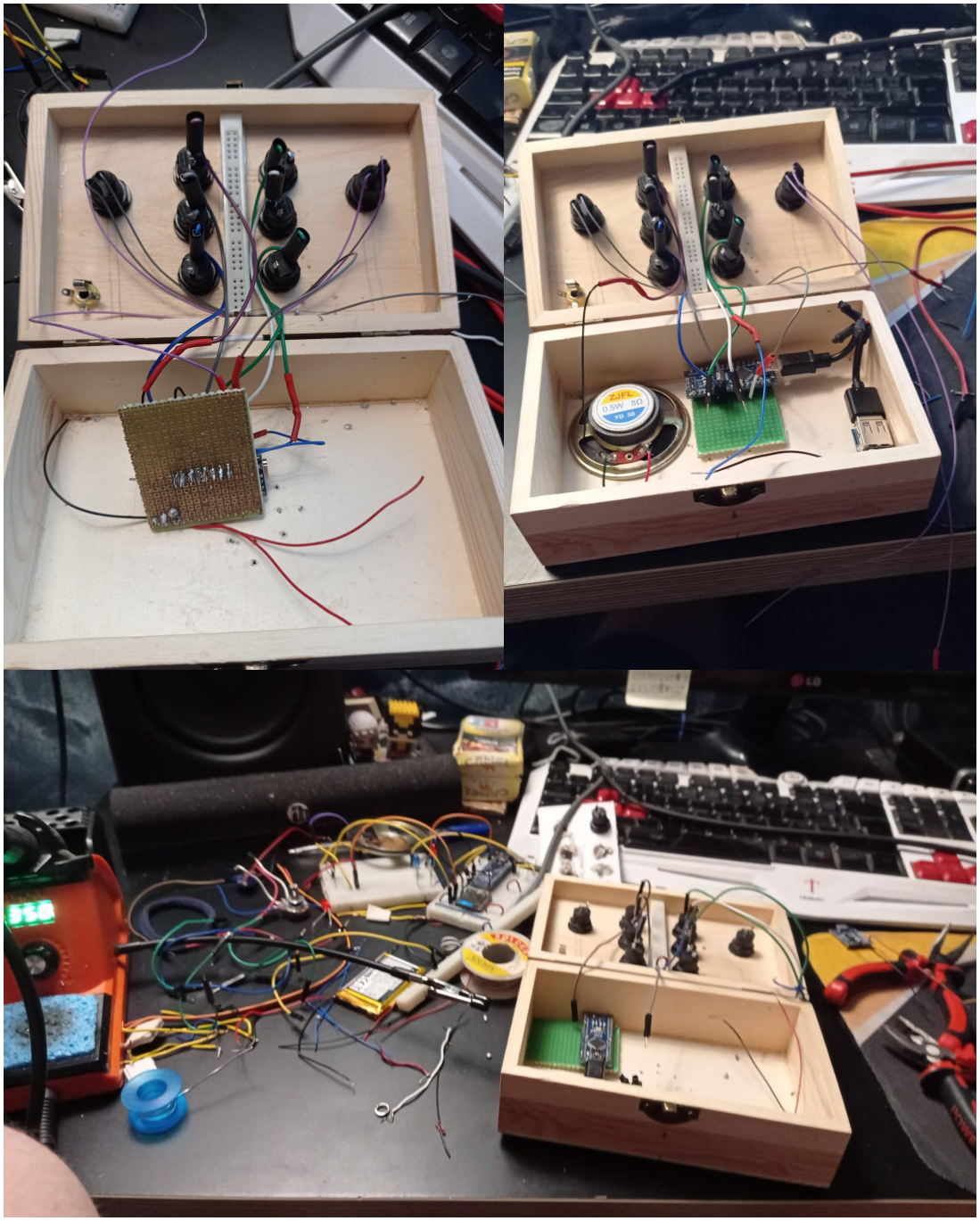


Lo siguiente que se ha hecho es introducir y probar todas las combinaciones de las letras, y posteriormente de los números, el objetivo es asegurar que todas las combinaciones corresponden a un número para una variable y que cada variable se corresponde a su respectivo audio.



A continuación se termina de trabajar en el código y se hacen exhaustivas pruebas de funcionamiento del programa para corregir los posibles fallos y errores. Se reproduce el juego en múltiples ocasiones y durante extensos periodos de tiempo para asegurar la calidad del prototipo, además se hacen pruebas de unas últimas configuraciones para el audio.

Dado que ya tenía casi todo listo llegó el momento de trabajar en la carcasa. Se han hecho cálculos aproximados de donde quedarían cada uno de los botones y se realiza el trabajo, se ha adaptado el circuito y los componentes a las medidas de la carcasa y la carcasa y los botones se han adaptado acorde a las necesidades que se consideran relevantes por su uso.

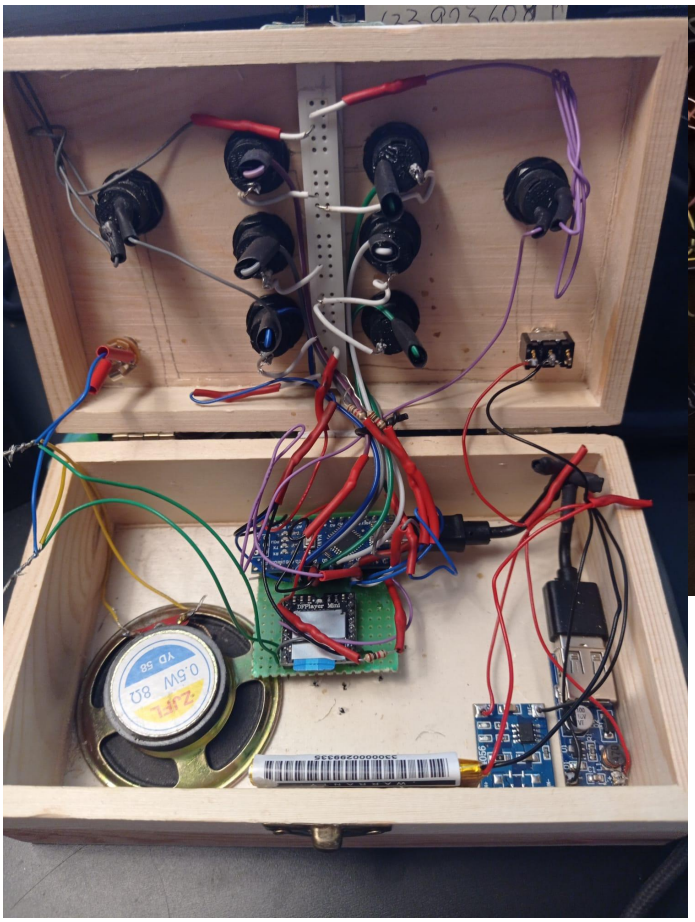


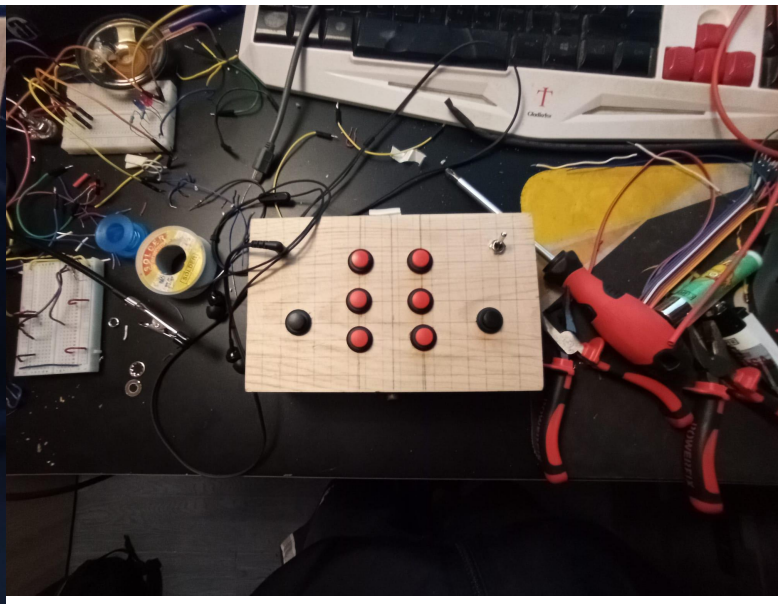
Una vez que estaba todo prácticamente acabado se me ha ocurrido implementar algo mas a mi trabajo, que sea original e interesante, he encontrado la manera de hacer posible aprender morse, si el usuario o alguien del entorno tiene capacidad visual entonces podrá identificar el código morse reflejado en uno de los led usados durante este trabajo.

International Morse Code

A	•–	N	–•	1	•–––––
B	–•••	O	–––	2	••–––––
C	–•–•	P	•––•	3	•••–––
D	–••	Q	––•–	4	••••–
E	•	R	•–•	5	•••••
F	••–•	S	•••	6	–••••
G	––•	T	–	7	––•••
H	••••	U	••–	8	–––••
I	••	V	•••–	9	––––•
J	•–––	W	•––	0	–––––
K	–•–	X	–••–	.	•–•–•–
L	•–••	Y	–•–––	,	––••–––
M	––	Z	––••	?	••––••

Por último pero no menos importante se ha trabajado un poco la carcasa para que sea más cómoda y un poco más ergonómica y facilitar así el agarre de los más pequeños.
El resultado final se ve de la siguiente manera:





12. La única conclusión a la que he llegado es que ayudar a otros es maravilloso, si se le ponen suficientes ganas a algo eso se vuelve en una posibilidad que se pueda hacer. Lo que más me ha gustado en este proyecto es darme cuenta realmente de los conocimientos que adquirí en estos dos años y he podido realizar tareas con técnicas aprendidas en clase, trabajar de manera correcta y segura y lo que es más importante es que ahora estoy aprendiendo las cosas sabiendo.
Durante este proyecto he cogido un poco más de soltura en el lenguaje de programación C++ y la verdad es que aporta bastante satisfacción.

13. Las líneas de investigación futuras podrían ser implementaciones de palabras, más símbolos en braille y a lo mejor un nuevo interfaz para el juego.

14. WEBGRAFÍA:

Arduino Nano: todo lo que necesitas saber sobre esta placa de desarrollo | Hardware libre (hwlibre.com)
boolean - Guía de Referencia de Arduino
Reproducir MP3 en Arduino con DFPlayer Mini (luisllamas.es)
https://www.makerelectronico.com/producto/dfplayer-mini-reproductor-mp3/
https://ferretronica.com/products/modulo-cargador-baterias-litio-tp4056-micro-usb-5v-1a
https://www.luisllamas.es/arduino-mp3-dfplayer-mini/
https://es.wikipedia.org/wiki/Braille_(lectura)
https://morsecode.world/international/translator.html
https://morsedecoder.com/