

# PianoGraph!

Sistema virtual de aprendizaje de piano.



Autores: Daniele, Fernando.  
Velazquez, Gonzalo.

Asignatura: Proyecto Final

Profesor: Ing. Daniel Musso

Institución: U.T.N. Facultad Regional  
San Francisco.

Lugar: San Francisco, Córdoba

Daniele, Fernando – Velazquez, Gonzalo .....  
PianoGraph! .....II

## RESUMEN

Aprender a tocar un instrumento musical es una actividad que requiere de un gran esfuerzo mental, similar al de aprender un nuevo idioma. Esto desde siempre ha fascinado a gran parte de la población, sin importar su edad o ubicación geográfica. Según datos del Instituto Nacional de Estadística y Geografía (México), cerca del 30% de la población mexicana toca algún instrumento musical. Este número asciende hasta casi un 50% en Austria, ubicándose el piano como el segundo instrumento más tocado (después de la guitarra eléctrica).

En este proyecto, hemos desarrollado una herramienta para facilitar el proceso de aprendizaje de un instrumento en específico, el piano. Mediante una pantalla/cortina de ledes ubicada sobre las teclas del mismo, y haciendo uso de señales lumínicas, el dispositivo le indicará al usuario cuales teclas debe presionar en el instante preciso en que corresponda. Mediante un microcontrolador se decodifica cualquier canción en formato MIDI que el usuario desee aprender, y se muestra en el dispositivo. De esta forma el usuario puede ejecutar perfectamente cualquier canción siguiendo las instrucciones. Al finalizar la canción el dispositivo otorga un puntaje de éxito/error obtenido, con el fin de tener una medida para auto superarse o competir con amigos.

**Palabras claves propuestas:** CORTINA LED, PANTALLA LED, MIDI, PIANO, INSTRUMENTO MUSICAL, APRENDIZAJE.

Daniele, Fernando – Velazquez, Gonzalo .....

PianoGraph! .....III

## ÍNDICE

RESUMEN.....	III
ÍNDICE.....	IV
INTRODUCCIÓN.....	1
OBJETIVOS.....	4
DESCRIPCIÓN .....	5
METAS .....	9
JUSTIFICACIÓN.....	11
ASPECTOS DE MERCADO .....	12
ASPECTOS INSTITUCIONALES .....	15
CONTENIDO DEL PROYECTO .....	14
CONCLUSIONES.....	55
APENDICE .....	56
BIBLIOGRAFÍA .....	57
ANEXO I.....	58
ANEXO II.....	60

Daniele, Fernando – Velazquez, Gonzalo .....

PianoGraph! .....IV

## INTRODUCCIÓN

Este proyecto nace de la necesidad personal de uno de nosotros de aprender a tocar el piano, quien, al momento de hacerlo, se encontró con muchas dificultades, principalmente por los actuales métodos de aprendizaje.

A grandes rasgos, existen dos métodos de aprender a ejecutar un instrumento musical:

- Un primer método consiste en aprender la teoría musical, cómo están compuestos los acordes, dónde se ubica cada nota en el teclado, y comenzar a leer partituras simples, hasta eventualmente poder ejecutar partituras más complejas. Este es un método que requiere gran dedicación y esfuerzo, ya que se aprende verdaderamente música, se comprende lo que se está tocando y se puede llegar a dominar el instrumento profesionalmente.



Fig.1

- Un segundo método consiste en observar a otra persona tocando la canción deseada, memorizar cuales teclas presiona y copiar los movimientos. Esta es una solución que no requiere de estudios ni aprendizaje de la teoría musical. Presenta la ventaja de que en poco tiempo (quizás menos de un día), la persona puede estar ejecutando perfectamente alguna pieza. La desventaja es que se requiere de una solución específica para cada canción diferente que la persona quiera aprender, ya que en cada canción se tocan diferentes notas, la persona debe copiar diferentes movimientos.

Este es un método ampliamente difundido en internet, donde se pueden observar videos para aprender diferentes canciones, e incluso existen aplicaciones dedicadas a este método, como Synthesia ®. Algunas personas sólo desean poder tocar el piano sin dedicarse profesionalmente a ello, y ésta es una solución adecuada para aprender a ejecutar canciones sin tener que pasar quizás años estudiando música. Es por eso que optamos por desarrollar un dispositivo que utilice este método para enseñar a los usuarios.

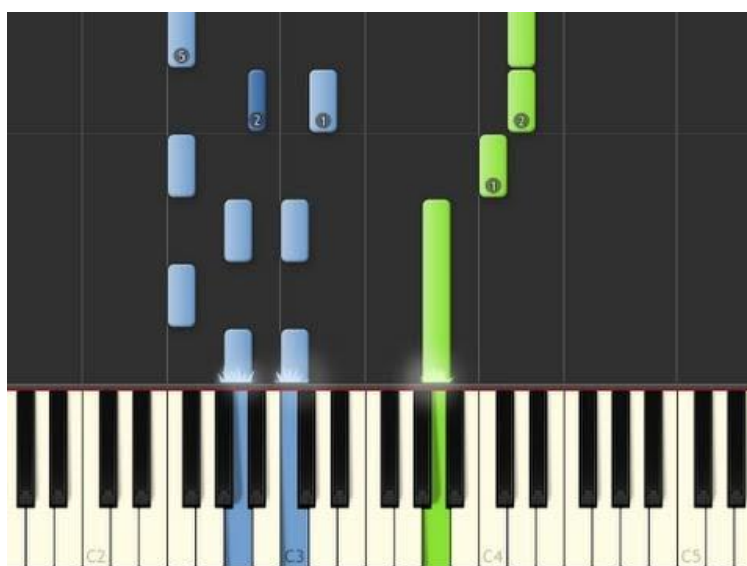


Fig.2 - Synthesia

El dispositivo a desarrollar constaría de una pantalla o cortina de ledes, con tantas columnas de ledes como teclas tenga el piano/teclado. Estos ledes se encenderán unos instantes antes del momento de que el usuario deba presionar la nota, desde la primer fila (ubicada más alejada del piano), secuencialmente hasta la última fila (ubicada lo más cerca de las teclas posible), que será el momento en que se debe presionar dicha tecla.

Gracias a una interfaz y una metodología de juego sencilla cualquier persona, de cualquier edad o educación podrá aprender a tocar el piano. Además, este proyecto presenta un potencial enorme ya que no existe nada similar en el mercado. Los videojuegos que utilizan el mismo principio, como Guitar Hero ® (en el que se toca una guitarra con seis botones), o Pump It Up ® (una alfombra donde se presionan unos indicadores luminosos con los pies al ritmo de la música), tuvieron

gran aceptación y éxito, lo que indica que para las personas es fácil y divertido entender la dinámica del juego.



Fig 3 y 4 – Videojuegos que utilizan el mismo principio

Una ventaja fundamental, es que el usuario no debería comprar un piano específico para utilizar este sistema, sino que se puede adaptar a cualquier piano/teclado existente.

## OBJETIVOS

La finalidad del proyecto es crear un dispositivo gracias al cual cualquier persona, sin importar su edad o educación, sea capaz de aprender de manera sencilla y divertida a tocar el piano, o instrumentos similares tales como el órgano o teclado electrónico. El usuario debe ser capaz de elegir la melodía que desea tocar mediante una computadora o celular. Los mismos deben estar conectados física o inalámbricamente al dispositivo, el cual a través de una pantalla de diodos ledes indicará en que momento presionar o soltar cada tecla del instrumento musical.

Podemos dividir el objetivo general en tres objetivos específicos:

- 1° Comprender el protocolo de comunicación MIDI, y a través de una conexión por puerto MIDI mostrar las notas correspondientes en la cortina de ledes.
- 2° Realizar una retroalimentación desde el teclado electrónico hacia el dispositivo para comparar lo tocado por el usuario con la melodía original.
- 3° Desarrollar un aplicación para celulares que realice una comunicación bluetooth con el dispositivo para poder cargar las canciones a través del mismo y visualizar el puntaje obtenido.



## DESCRIPCIÓN

El protocolo MIDI (Musical Instrument Digital Interface), es un tipo de comunicación serial utilizado ampliamente por casi todos los instrumentos musicales que posean la necesidad de intercambiar información entre ellos, (sintetizadores, controladores, teclados, etc). Si tomamos como ejemplo un teclado electrónico que tenga la posibilidad de recibir mensajes MIDI provenientes de otro instrumento o computadora, y pueda reproducirlos por sus parlantes, lo que sucede para que ello ocurra es lo siguiente:

1. El dispositivo emisor del mensaje envía mediante el cable MIDI una serie de bytes, donde se encuentra especificado, por ejemplo, si es un mensaje para encender una nota: el número de esa nota (del 0 al 127, correspondiendo el 60 al do central de un piano), y la intensidad con la que esa nota debe sonar (el volumen).
2. El teclado electrónico que reciba esos mensajes, tiene un banco de sonidos almacenados, con un sonido diferente para cada una de las 128 notas posibles, y reproducirá según indique el mensaje MIDI, una de estas notas, hasta que llegue otro mensaje MIDI, que indique que el sonido debe cesar.

Cabe aclarar que en un mensaje MIDI no está especificado el tipo de sonido que debe reproducirse, es decir, el dispositivo que reproduzca el sonido puede asignar el timbre de una guitarra, de una flauta, o de cualquier instrumento que tenga cargado en su base de datos al mensaje MIDI recibido.

De la misma manera que un teclado recibe y transforma esos mensajes MIDI a sonido, **PianoGraph!** posee un microcontrolador que recibe y transforma los mensajes MIDI en estímulos visuales (luces ledes) y asigna una columna de ledes a cada número de nota recibido.

Cada columna se encuentra posicionada exactamente sobre cada tecla de un piano estándar. Entonces al encenderse el primer Led de una columna, este comienza a descender fila por fila, hasta llegar a la última fila, donde será el momento de presionar la tecla indicada.

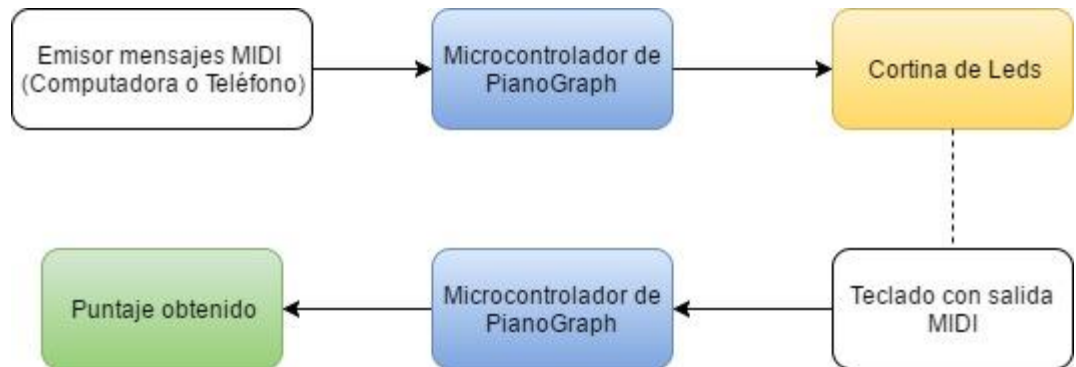


Fig. 5 - Descripción

Las luces descenderán desde la primera fila hacia la última, desplazándose a intervalos de tiempo iguales. Este intervalo puede ser ajustado por el usuario, cuanto menor sea el tiempo de desplazamiento, más rápida será la velocidad con la que las luces llegan hasta la última fila, con lo que el tiempo de reacción del usuario deberá ser también menor. Al disminuir el tiempo de desplazamiento lo que se logra es tener una mayor resolución de las notas de muy corta duración, a costo de disminuir el tiempo total en que la nota se muestra en pantalla.

Por ejemplo, si suponemos que una nota dada tiene una duración tal que logra encender 4 filas de ledes simultáneamente mientras se desplaza, al disminuir el tiempo de desplazamiento a la mitad, lo que lograremos es que esa nota ahora encienda 8 filas de ledes, pero el tiempo total en que la tecla del piano deberá ser presionada permanece exactamente igual.

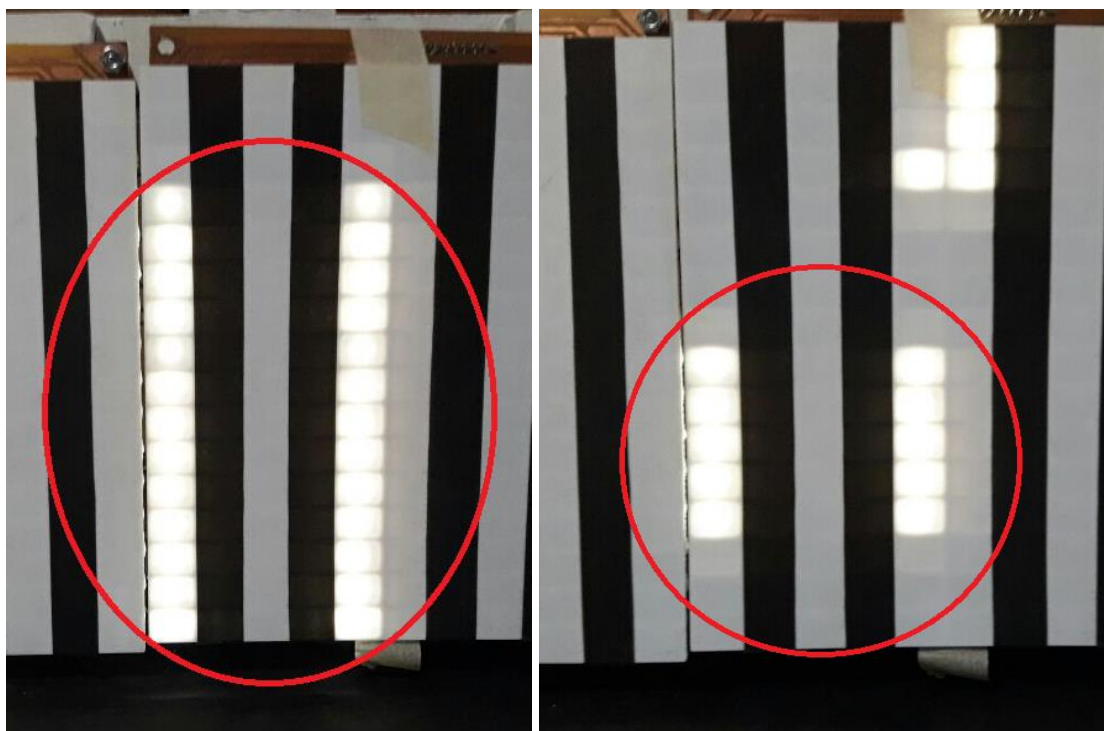


Fig.6 – Notas iguales a distinta velocidad de desplazamiento

Una vez que las notas han sido indicadas por **PianoGraph!**, el microcontrolador detectará si existe un teclado enviando datos mediante una conexión MIDI, para lograr entonces realizar la comparación entre la nota mostrada y la tecla presionada, obteniendo así mediante un algoritmo un porcentaje que representará la fidelidad con la que fueron presionadas las notas en el teclado, como consecuencia final entonces, la fidelidad con la que la canción fue ejecutada por el usuario.

Este puntaje será enviado mediante la conexión Bluetooth al teléfono desde donde se estaban enviando las instrucciones MIDI (utilizando la apk desarrollada para el mismo propósito), y será visualizada en pantalla inmediatamente al finalizar la canción.

El envío de los mensajes MIDI que contienen las notas pertenecientes a las distintas canciones que el usuario desee ejecutar se realiza mediante la aplicación para dispositivos Android llamada también **PianoGraph!**. Esta aplicación establece una conexión Bluetooth entre el celular y el dispositivo para poder enviar las instrucciones MIDI.

## METAS

Para evaluar el avance de nuestro proyecto, los puntos a cubrir deberían ser:

1. **Lograr controlar una alta cantidad de ledes con una única unidad microcontroladora** (podrían llegar a ser más de 1000 ledes que operen de manera independiente cada uno).

Una vez que se haya logrado el multiplexado y desarrollados los algoritmos pertinentes para controlar eficazmente esa cantidad de diodos, se podría proceder a la segunda parte del proyecto.

2. **Desarrollar un algoritmo capaz de interpretar los mensajes MIDI y transformarlos en instrucciones para encender los ledes correspondientes.**

Cuando el microcontrolador sea capaz de entender un mensaje MIDI (inicialmente enviado a través de cables, directamente desde un instrumento MIDI), y asignar a cada columna de ledes, una nota recibida, el próximo paso sería automatizar el envío de estas instrucciones.

3. **Desarrollar una aplicación móvil que sea capaz de interpretar archivos MIDI** (el formato en que deberían encontrarse las canciones a aprender), y enviarlos al microcontrolador.

Consiste en programar mediante el entorno de programación “**Android Studio**”, una aplicación capaz de comunicarse con **PianoGraph!**, e interpretar archivos MIDI descargados desde internet.

4. **Establecer una comunicación inalámbrica entre el teléfono y la cortina de ledes.**

Este paso estará finalizado cuando podamos conectarnos y desconectarnos a voluntad, desde un dispositivo móvil a **PianoGraph!**, y poder repetir el proceso desde cualquier otro dispositivo.

5. **Realimentar el microcontrolador de la cortina con las notas del piano,** compararlas con las notas que debían ser presionadas, y enviar al teléfono un valor de error/acierto.

## JUSTIFICACIÓN

Actualmente, una de las aplicaciones más utilizadas para aprender el manejo del piano es **Synthesia** (con más de un millón de descargas en **Google PlayStore**). Posee una interfaz sencilla, rápida y amigable para el usuario. Además de proporcionar un método de enseñanza altamente gráfico y efectivo.

Fue por eso, que al ver un producto tan ampliamente aceptado como lo fue **Synthesia**, sentimos que podíamos seguir el mismo camino y mejorar aún más este método de enseñanza. No existe actualmente ningún producto con características similares a **PianoGraph!**. Los softwares como **Synthesia** presentan la gran desventaja que dependen del tamaño del dispositivo en donde sean ejecutadas. Al ver las notas presionadas en un teléfono de quizás 5 pulgadas, se pierde la perspectiva al tratar de ubicarlas rápidamente en un piano de cerca de un metro de ancho.

Por eso consideramos que **PianoGraph!** presenta una ventaja fundamental sobre todas estas aplicaciones, y que su comercialización es posible internacionalmente.

No es ningún secreto que el piano es uno de los instrumentos más tocados y más difundidos a lo largo del planeta, por lo que estamos frente a un mercado considerablemente amplio.

## ASPECTOS DE MERCADO

Al ser un producto que requiere un bajo costo de producción, poco instrumental, y poco tiempo de fabricación, sería posible manufacturarlo de manera personal, en un espacio relativamente reducido, tal vez tercerizar los elementos que requieran una calidad visual especial (como el soporte o la cortina), ensamblarlo y comercializarlo por internet, sin la necesidad de un local físico donde se presente al público.

Esperamos que la demanda crezca exponencialmente con el paso del tiempo, a medida que las personas conozcan la existencia del producto.

Creemos que este producto tendrá un impacto mayor sobre los jóvenes, es por eso que nuestra publicidad estaría fuertemente enfocada a las redes sociales, cuyo alcance trasciende fronteras y puede ser visualizado por personas de cualquier parte del mundo en instantes. Además, al ser los costos de producción relativamente bajos, sería posible su comercialización a un precio razonable, posibilitando la aún más rápida difusión del producto (mientras más personas lo posean, más personas conocerán la existencia del producto).

## ASPECTOS FINANCIEROS

Para la realización de este proyecto se necesitó la compra de diversos materiales, obtenidos en mercados locales, que están detallados a continuación:

### Componentes para placa PCB:

CANTIDAD	TIPO	VALOR
1	CONECTOR TRANSFORMADOR	
1	LED VERDE 5MM	
1	OPTOACOPLADOR	6N136
1	DIODO PROTECCION	1N4001
1	C1	100pF
1	C2	100pF
1	RD2	1K
1	REGULADOR	7805
1	R MIDI1	220
1	R MIDI 2	470
1	R MIDI 3	10K
1	DIODO MIDI 1	1N4148
1	TRANSISTOR MIDI Q1	BC547
1	ATMEGA 2560	
1	CM1	22p
1	CM2	100n
1	CM3	100n
1	CM4	100n
1	CM5	100n
1	CM6	100n
1	RM1	10K
1	XTAL	SPK16.000G
2	CXTAL	22 pF
1	R XTAL	1M
1	RXTAL	27 $\Omega$
1	SWITCH BOTON	
1	DIODO M 1	1N4001
6	CONECTORES HEMBRA	2X8
12	CONECTORES MACHO	2x8
2	TIRA DE 40 PINES	

Fig. 7

**Componentes para ensamblado y desarrollo:**

Placa desarrolladora Arduino Mega	390
Circuitos Optoacopladores	60
Madera	40
Plástico para impresora 3D	20
Conversor MIDI-USB	200
Placas para PCB	410
Separadores	60
Vinilo Polarizado	26
Módulo Bluetooth	150
Caños	189
Bulones y tornillos	100
LED	1050
Cables + conectores	300
Manufacturación de prototipos de PCB	2000
TOTAL	4995

Fig. 8

El diseño del primer prototipo tuvo un costo aproximado de \$5000 pesos, aunque confiamos en que, una vez iniciada la producción, los costos se reducirán considerablemente debido a la compra en cantidad de insumos, y a una mejor selección de proveedores.



## **ASPECTOS INSTITUCIONALES**

No existe una institución que realice o financie este proyecto, los únicos responsables del mismo somos los autores del presente trabajo.

## CONTENIDO DEL PROYECTO

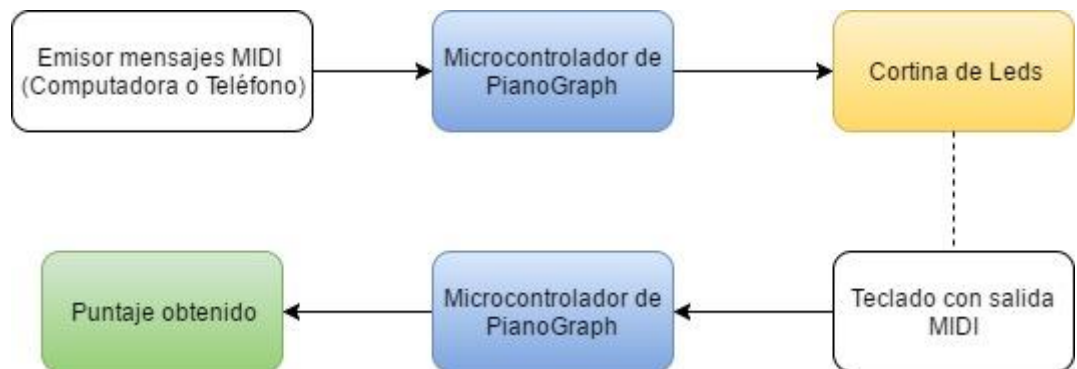


Fig. 9

Para poder utilizar este dispositivo, en primer lugar, debemos saber qué canción deseamos aprender a tocar en el piano. A modo de ejemplo, vamos a suponer que deseamos aprender a tocar la canción “Moonlight Sonata” de “Beethoven”. Entonces debemos conseguir el archivo en formato MIDI que contenga las instrucciones a interpretar por **PianoGraph!**.

En internet se puede encontrar prácticamente cualquier canción en este formato simplemente ingresando en un buscador web el título de la canción deseada acompañado de la palabra “MIDI”.



Fig. 10 – Búsqueda en Google.

En este caso, clickeamos en la primera página web ([www.8notes.com](http://www.8notes.com)) y seguimos las instrucciones para descargar el archivo.

Entonces ya tenemos el archivo, pero...

- **¿Qué es un archivo MIDI?:**

Las siglas MIDI son una abreviatura de Musical Instrument Digital Interface. Se trata de un protocolo de comunicación que apareció en el año 1982, fecha en la que distintos fabricantes de instrumentos musicales electrónicos se pusieron de acuerdo en su implementación. Aunque originalmente se concibió como un medio para poder interconectar distintos sintetizadores, el protocolo MIDI se utiliza actualmente en una gran variedad de aplicaciones: grabación musical, cine, TV, ordenadores domésticos, presentaciones multimedia, etc.

Dado que este protocolo es bastante eficiente en cuanto a enviar cantidades de datos relativamente grandes a una velocidad respetable, se ha convertido en un elemento de gran utilidad para compositores, educadores, programadores y gente que intenta crear música con varios instrumentos. Con la ayuda de un ordenador o un secuenciador hardware, permite crear arreglos multipistas, líneas o partes instrumentales, etc.

Veamos de forma un poco más precisa algunas de las ventajas que proporciona:

Generar sonido a partir de un sintetizador MIDI en vez de hacerlo partiendo de un sampler tiene algunas ventajas. La primera de ellas es que se necesita una gran cantidad de espacio de almacenamiento para guardar el audio muestreado (p.ej., en forma de archivos .WAV o AIFF).

Se necesitan unos 10 Mb de espacio en disco para almacenar 1 minuto de audio estéreo muestreado en calidad CD (16 bits y 44,1kHz). En comparación, los archivos de datos MIDI tienen un tamaño insignificante. Una secuencia MIDI típica utiliza sólo unos 10 Kb por minuto.

**¿Y cómo funciona?**

El archivo MIDI no contiene datos de audio muestreado, sino más bien una serie de instrucciones que el sintetizador u otro generador de sonido utiliza para reproducir el

sonido en tiempo real. Estas instrucciones son mensajes MIDI que indican al instrumento qué sonidos hay que utilizar, qué notas hay que tocar, el volumen de cada una de ellas, etc.

### Otras ventajas:

Ofrece la posibilidad de editar la música con facilidad.

Permite alterar la velocidad de reproducción y la altura tonal de los sonidos de forma independiente.

### ¿Cómo está compuesto un mensaje MIDI?

Un mensaje MIDI es una secuencia de bytes que contienen información sobre qué tipo de mensaje se está transmitiendo (ya sea un comando o un dato), y datos referentes a la nota que se debe ejecutar. Estos datos se transmiten entre los instrumentos musicales, con una tasa de baudios de 31.250.



Fig. 11 – Mensaje MIDI

Entonces podemos observar que por ejemplo para un mensaje de “Nota ON” (que empiece a reproducirse una nota dada en el instrumento objetivo), se utilizarán 3 bytes. El primer byte indicará que es una instrucción, utilizando los primeros 4 bits para definir de cuál de todas las instrucciones se trata, y los últimos 4 bits indicarán en cuál de los 16 canales disponibles (se pueden manejar hasta 16 instrumentos diferentes) debe reproducirse dicha instrucción.

El byte siguiente indicará qué nota se debe encender, y el último indicará la velocidad con la que debe presionarse (intensidad o volumen). Con el protocolo MIDI se pueden manejar hasta 128 diferentes notas, más que suficiente para cualquier instrumento (el piano con mayor cantidad de notas actualmente posee 96 teclas). En la siguiente tabla se pueden observar los números asignados a cada nota. Cabe

aclarar que en la tabla las notas se encuentran designadas con cifrado americano.

Do	Re	Mi	Fa	Sol	La	Si
C	D	E	F	G	A	B

Fig. 12 - (Traducción del cifrado latino a cifrado americano)

Octava	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

Fig. 13 – Número de notas

Otra aclaración es que según la teoría musical, existen 12 notas en la escala cromática, ya que 5 de las 7 notas naturales presentan una nota intermedia, llamada sostenido, o bemol, dependiendo de la escala que se esté utilizando. Cada grupo de 12 notas se denomina "Octava", nuestro dispositivo fue diseñado para contener 5

octavas (60 notas), existiendo la posibilidad de anexas más octavas si se deseease.

Command	# Parameters	Parameter 1	Parameter 2
Note Off	2	note	velocity
Note On	2	note	velocity
Aftertouch	2	note	aftertouch
Continuous Controller	2	controller #	controller value
Patch Change	1	instrument #	
Channel Pressure	1	pressure	
Pitch bend	2	lsb (7 bits)	msb (7bits)
Non-musical Commands			

Fig. 14 – Parámetros MIDI

En la tabla anterior podemos observar distintos tipos de comandos que pueden ser enviados en formato MIDI. Y en la tabla siguiente podemos observar con qué código se identifica cada comando (el primer byte). Para el uso de nuestro dispositivo, sólo nos interesan los comandos “Nota ON” y “Nota OFF”, es por eso que la aplicación diseñada para interpretar los archivos MIDI, está diseñada para ignorar el resto de los comandos, junto a sus parámetros respectivos.

1000XXXX = Note off
1001XXXX = Note on
1010XXXX = Aftertouch
1011XXXX = Continuous controller
1100XXXX = Patch change
1101XXXX = Channel pressure
1110XXXX = Pitch bend
1111XXXX = Non-musical commands

Fig. 15 – Instrucciones MIDI

El protocolo de comunicación MIDI se basa en el **protocolo Serial Asíncrono**.

Cuando se opera en modo asíncrono no existe una línea de reloj común que establezca la duración de un bit y el carácter puede ser enviado en cualquier momento. Esto conlleva que cada dispositivo tiene su propio reloj y que previamente se ha acordado que ambos dispositivos transmitirán datos a la misma velocidad.

No obstante, en un sistema digital un reloj es normalmente utilizado para sincronizar la transferencia de datos entre las diferentes partes del sistema. El reloj definirá el inicio y fin de cada unidad de información, así como la velocidad de transmisión. Si no existe reloj común, algún modo debe ser utilizado para sincronizar el mensaje.

En realidad, la frecuencia con que el reloj muestrea la línea de comunicación es mucho mayor que la cadencia con que llegan los datos. Por ejemplo, si los datos están llegando a una cadencia de 2400 bps, el reloj examinará la línea unas 19200 veces por segundo, es decir, ocho veces la cadencia binaria. La gran rapidez con que el reloj muestrea la línea permite al dispositivo receptor detectar una transmisión de 1 a 0 o de 0 a 1 muy rápidamente, y mantener así la mejor sincronización entre los dispositivos emisor y receptor.

El tiempo por bit en una línea en que se transfiere la información a 2400 bps es de unos 416 microsegundos ( $1 \text{ seg}/2400$ ). Una frecuencia de muestreo de 2400 veces por segundo nos permitirá muestrear el principio o el final del bit. En ambos casos detectaremos el bit, sin embargo, no es extraño que la señal cambie ligeramente, y permanezca la línea con una duración un poco más larga o más corta de lo normal. Por todo ello, una frecuencia de muestreo lenta no sería capaz de detectar el cambio de estado de la señal a su debido tiempo, y esto daría lugar a que la estación terminal no recibiera los bits correctamente.

### **Bit de inicio y bit de parada**

En la transmisión asíncrona un carácter a transmitir es encuadrado con un indicador de inicio y fin de carácter, de la misma forma que se separa una palabra con una letra mayúscula y un espacio en una oración. La forma estándar de encuadrar un carácter es a través de un bit de inicio y un bit de parada.

Durante el intervalo de tiempo en que no son transferidos caracteres, el canal debe poseer un "1" lógico. Al bit de parada se le asigna también un "1". Al bit de inicio del carácter a transmitir se le asigna un "0". Por todo lo anterior, un cambio de nivel de "1" a "0" lógico le indicará al receptor que un nuevo carácter será transmitido.

### **Reglas de transmisión asíncrona**

La transmisión asíncrona que vamos a ver es la definida por la norma RS232, que se basa en las siguientes reglas:

- a. Cuando no se envían datos por la línea, ésta se mantiene en estado alto (1).
- b. Cuando se desea transmitir un carácter, se envía primero un bit de inicio que pone la línea a estado bajo (0) durante el tiempo de un bit.
- c. Durante la transmisión, si la línea está a nivel bajo, se envía un 0 y si está a nivel alto se envía un 1.
- d. A continuación, se envían todos los bits del mensaje a transmitir con los intervalos que marca el reloj de transmisión. Por convenio se transmiten entre 5 y 8 bits.
- e. Se envía primero el bit menos significativo, siendo el más significativo el último en enviarse.
- f. A continuación del último bit del mensaje se envía el bit (o los bits) del final que hace que la línea se ponga a 1 por lo menos durante el tiempo mínimo de un bit. Estos bits pueden ser un bit de paridad para detectar errores y el bit o bits de stop, que indican el fin de la transmisión de un carácter.

Los datos codificados por esta regla pueden ser recibidos siguiendo los pasos siguientes:

- a. Esperar la transición 1 a 0 en la señal recibida.
- b. Activar el reloj con una frecuencia igual a la del transmisor.
- c. Muestrear la señal recibida al ritmo de ese reloj para formar el mensaje.
- d. Leer un bit más de la línea y comprobar si es 1 para confirmar que no ha habido error en la sincronización.



## Velocidad de transmisión

En la transmisión asíncrona por cada carácter se envía al menos 1 bit de inicio y 1 bit de parada así como opcionalmente 1 bit de paridad. Esta es la razón de que los baudios no se correspondan con el número de bits de datos que son transmitidos.

La característica fundamental del formato de transmisión asíncrono es su capacidad de manejar datos en tiempo real, con un intervalo de longitud arbitraria entre caracteres sucesivos. Al final de cada carácter, la línea va a 1 en el bit de parada y permanece en ese estado durante un número arbitrario de bits ociosos. El inicio del nuevo carácter estará definido por la transición a 0 del bit de inicio.

### • Optoacoplador:

Si bien el protocolo MIDI está basado en la comunicación Serial asíncrona, también presenta algunas diferencias que hacen imposible (o poco recomendado) acoplarlo directamente a un dispositivo que utilice el protocolo Serial TTL.

La diferencia fundamental, es que un dispositivo que utiliza el protocolo TTL, utiliza diferentes niveles de tensión para identificar los 1 y los 0 lógicos (0V y 5V respectivamente), mientras que un dispositivo MIDI utiliza valores de corriente para el mismo objetivo (0mA y 20mA).

Es por eso que se recomienda aislar eléctricamente ambos dispositivos mediante el uso de un optoacoplador.

Un optoacoplador, también llamado *optoaislador* o aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un diodo led que satura un componente optoelectrónico, normalmente en forma de fototransistor o fototriac. De este modo se combinan en un solo dispositivo semiconductor, un fotoemisor y un fotorreceptor cuya conexión entre ambos es óptica. Estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP. Se suelen utilizar para aislar eléctricamente a dispositivos muy sensibles.

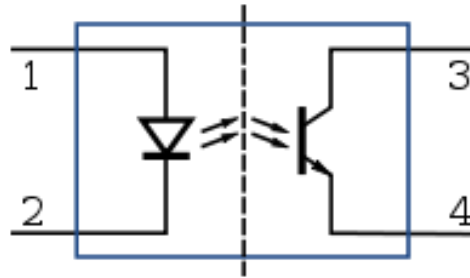


Fig. 16 – Esquema Optoacoplador

Como la velocidad de transmisión de 31250 baudios requiere una buena respuesta en frecuencia del transistor interno del optoacoplador, hemos utilizado el circuito integrado **6N136** para este proyecto, ya que presenta buenas prestaciones en alta frecuencia.

En las siguientes imágenes podemos observar la diferencia en el mensaje recibido por 3 diferentes optoacopladores (dos de ellos con mala respuesta en alta frecuencia, y otro, el dispositivo elegido).

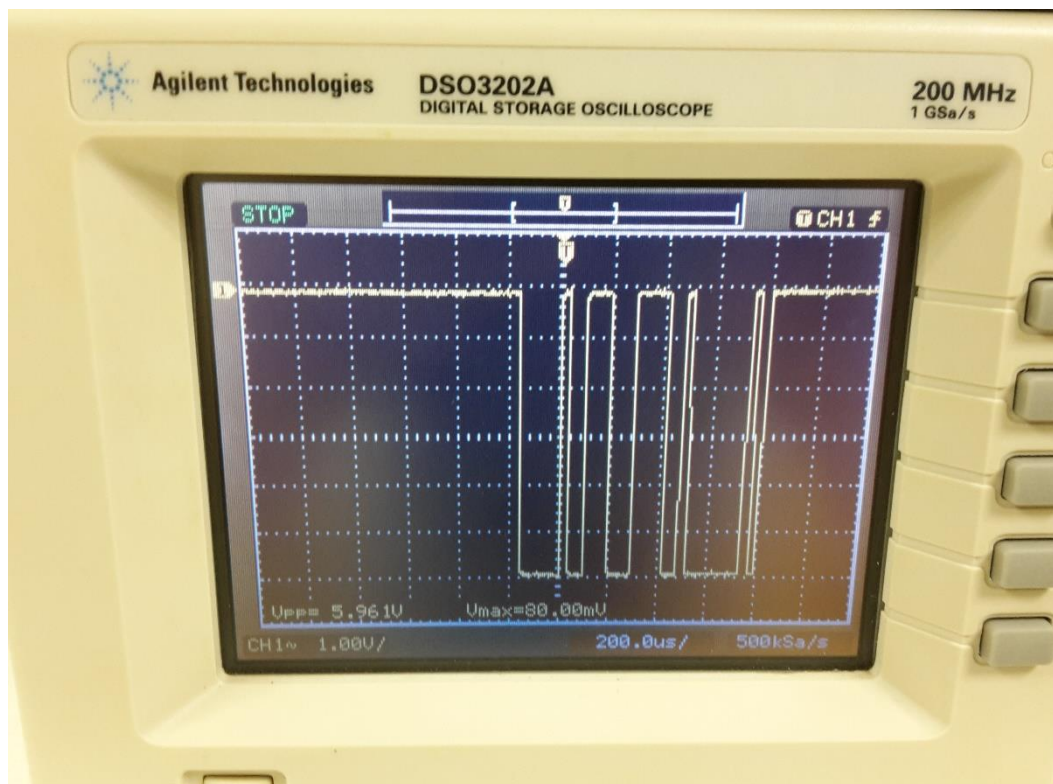


Fig. 17 – Mensaje original, utilizando el optoacoplador 6N136

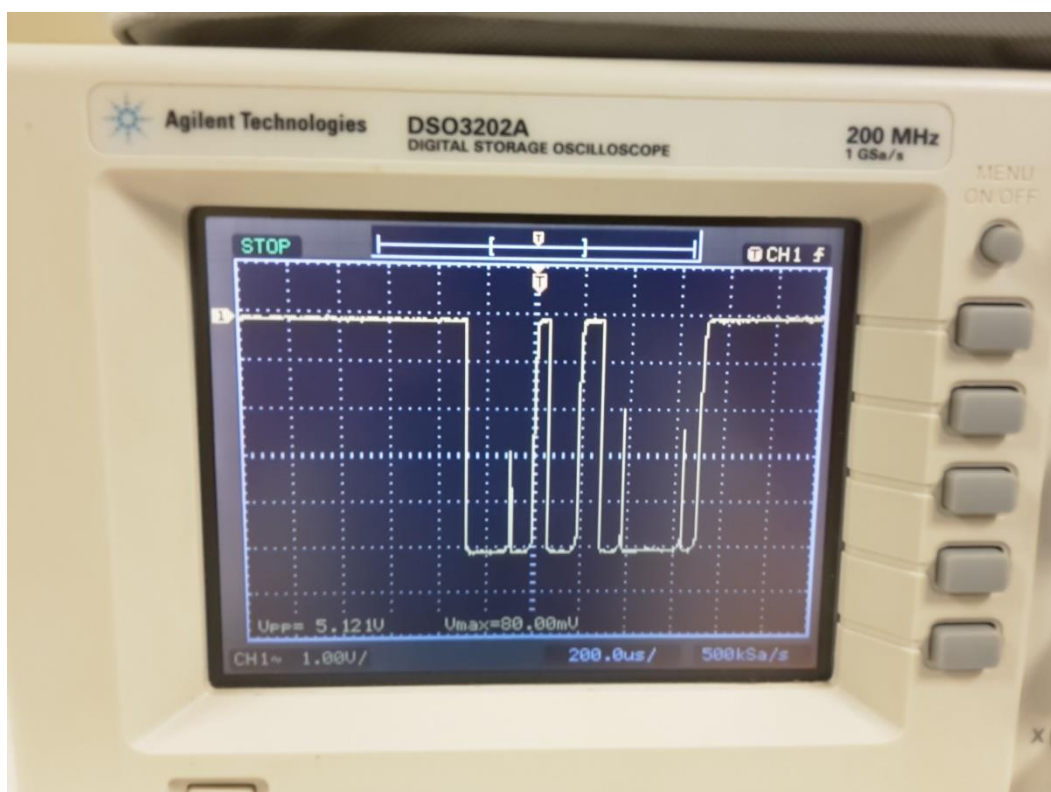


Fig. 18 – Mensaje distorsionado, utilizando el optoacoplador 4N26

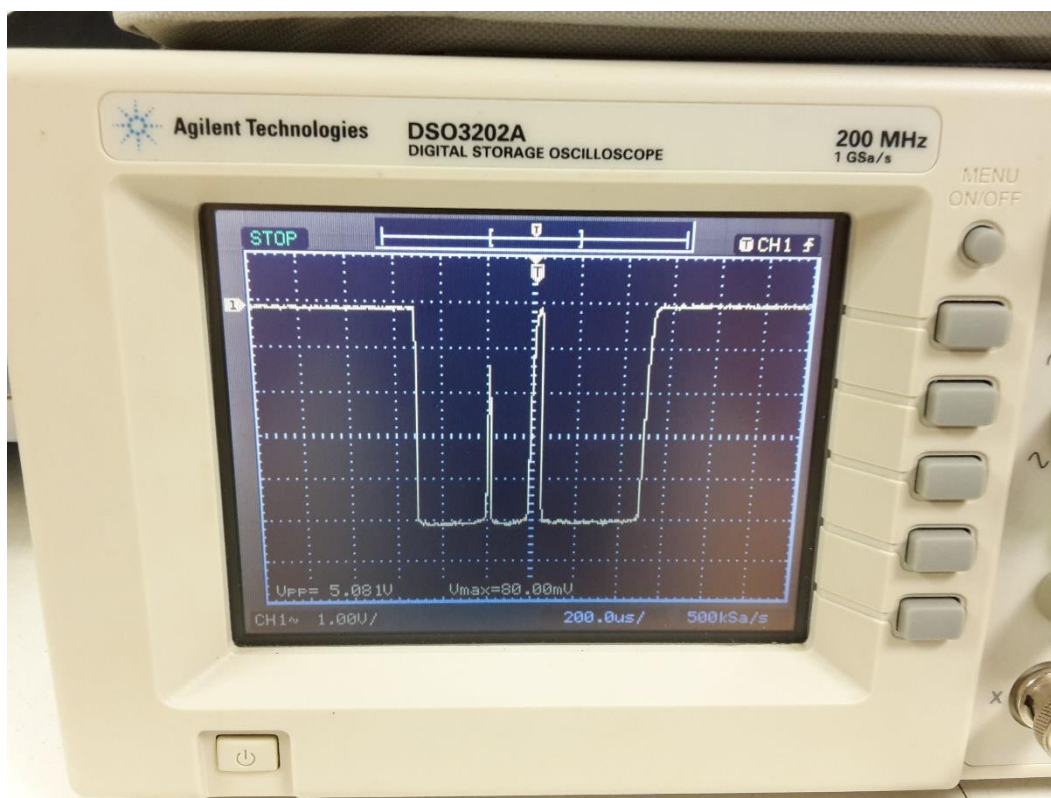


Fig. 19 – Mensaje distorsionado, utilizando el optoacoplador PC817

Una vez hechas las pruebas pertinentes respecto a la velocidad de respuesta del dispositivo elegido, diseñamos el circuito esquemático y luego, la placa PCB para integrar dentro de la placa final donde montaremos también el microcontrolador.

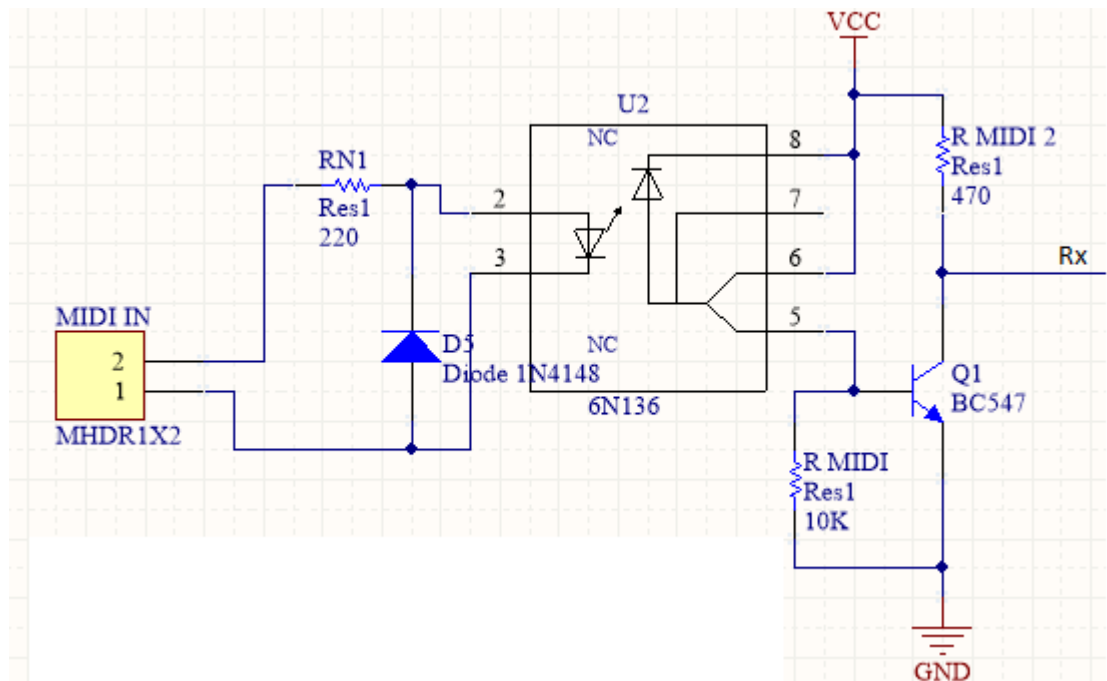


Fig. 20 – Circuito esquemático Acoplador MIDI-TTL

### • Pantalla Led

Para que el usuario comprenda qué teclas del piano debe presionar, diseñamos un arreglo de ledes de tal manera que sobre cada tecla del instrumento se sitúe una columna de 16 ledes separados 1cm uno de otro.

Estos ledes se encenderán momentos antes que el usuario deba presionar dicha tecla, comenzando por el diodo más alejado de la nota a tocar, y encendiéndose uno a uno, hasta llegar a la luz más cercana a la tecla, que será el momento en que el usuario debe presionar dicha tecla.

Para la construcción de esta pantalla, construimos 5 módulos idénticos de 12x16 ledes. Cada módulo representa una octava entera (las 12 notas de la escala).



Decidimos utilizar este método modular ya que existen teclados conformados desde 2 octavas hasta 8, entonces para evitar tener que construir un dispositivo para cada opción, simplemente el usuario debe decidir cuántas octavas desea y anexar tantos módulos como octavas precise.

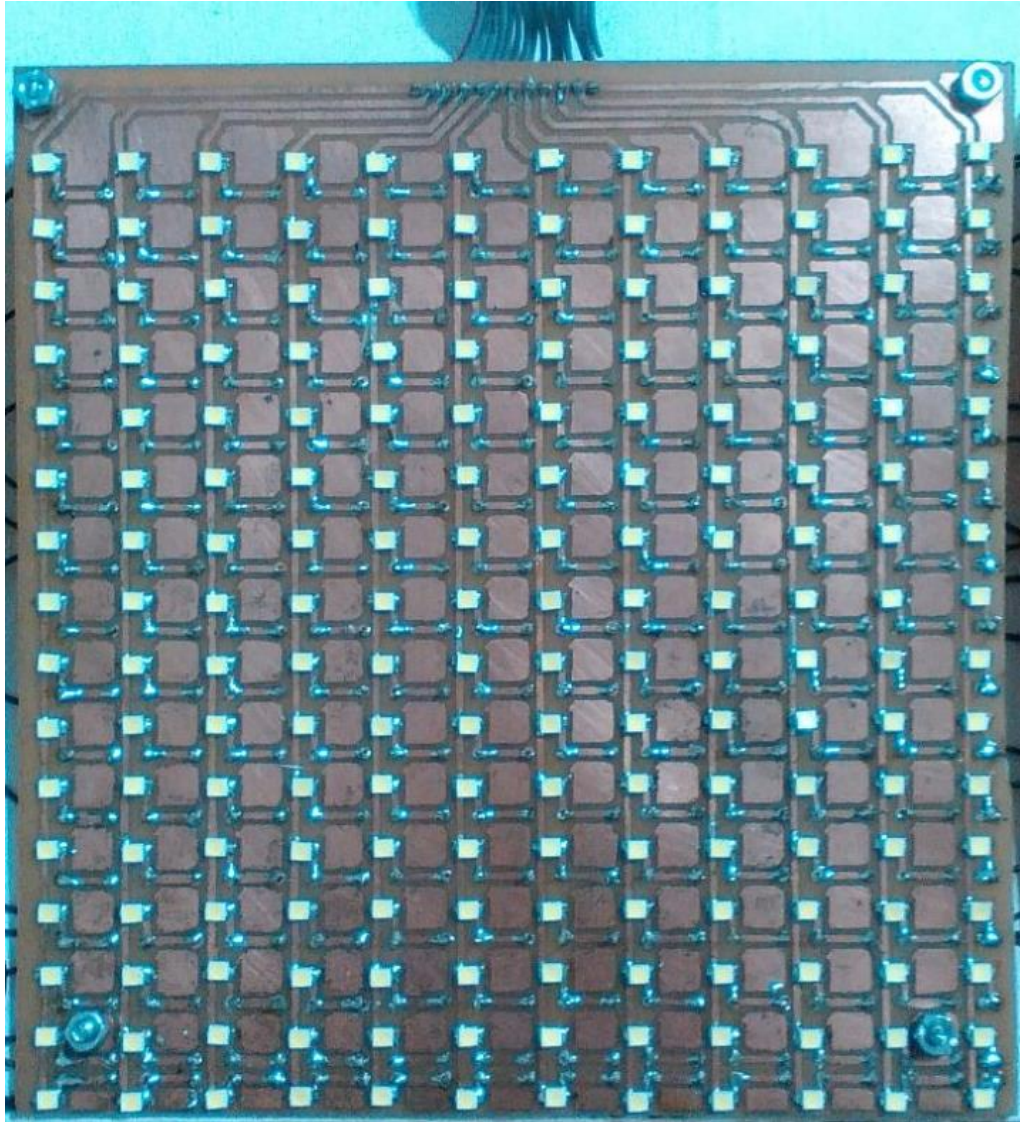


Fig. 21 – Módulo de una octava, 192 ledes

Cada módulo presenta un total de 28 entradas, 12 superiores, y 16 laterales, con las que con sólo 28 cables se puede controlar independientemente un total de 192 ledes diferentes.

Además, al anexar cada módulo junto al otro, los 16 cables laterales son compartidos, es decir que, por cada octava añadida, el microcontrolador sólo debe disponer de 12 nuevas salidas.

Otra ventaja de este método es que, para encender los ledes, hacemos uso de un multiplexado, por lo que sólo una fila está encendida a la vez, reduciendo el consumo hasta 16 veces lo que consumiría si se encendieran todas las luces simultáneamente. Al utilizar un circuito integrado multiplexador, no es necesario entonces que los 16 cables laterales provengan del microcontrolador, sino que podemos utilizar un número en binario del 0 al 16, que indique secuencialmente qué fila debe ser encendida. Esto fue de hecho lo que hemos realizado, reduciendo la cantidad de salidas necesarias de parte del microcontrolador de 16 a sólo 4.

Finalmente podemos decir que se necesitan 4 salidas del microcontrolador, y 12 más por cada octava que contenga la pantalla.

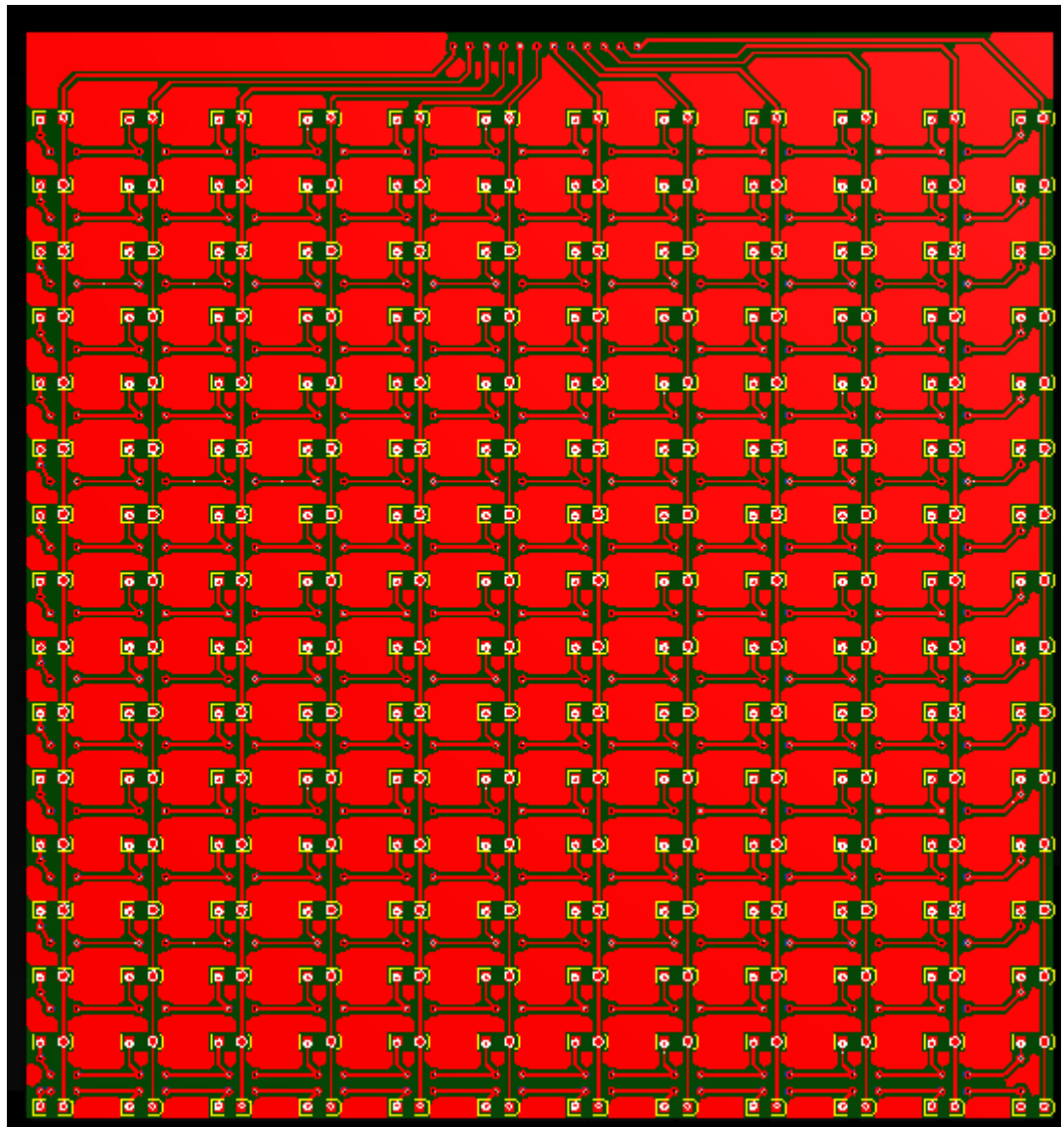


Fig. 22 – Circuito final de un módulo de octava

Una vez conectados todos los módulos necesarios (en nuestro caso utilizamos 5 octavas), procedemos a los cálculos electrónicos para diseñar los circuitos que manipulen esa gran cantidad de ledes (960 diodos) de la manera más eficiente posible.



Fig. 23 – Conexión de 5 módulos montados sobre soporte de madera

Para manejar la corriente de los ledes utilizamos dos integrados **ULN2803** que son drivers de corriente para 8 salidas. Utilizando dos de ellos podemos alimentar a las 16 filas de luces.

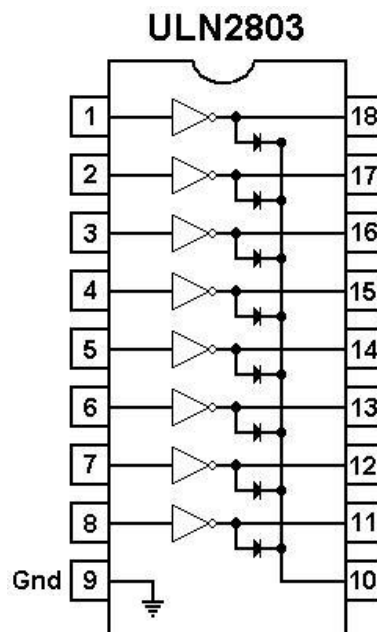


Fig. 24 – Esquema de ULN2803

Según la hoja de datos del led, la condición normal de funcionamiento se da con una tensión directa de 3 V y consume 20 mA. Como alimentamos la placa con 5V, es necesario utilizar una resistencia de 100 Ohms para limitar la corriente en cada fila de ledes.



$$R = \frac{5V - 3V}{20mA} = 100\Omega$$

Fig. 25

Estas resistencias estarían ubicadas en serie con cada fila, inmediatamente después de los drivers de corriente.

Para manipular las 16 entradas de los drivers y, por ende las 16 filas de ledes, hemos utilizado un decodificador **CD74HC4514** de 4 a 16 líneas, manejado por un microcontrolador. De esta forma cada fila se habilitará una a la vez, durante un intervalo de tiempo pequeño, para lograr una velocidad que supere el tiempo de respuesta de la vista humana y lograr el efecto de visual de que todos los ledes estén prendidos al mismo tiempo.

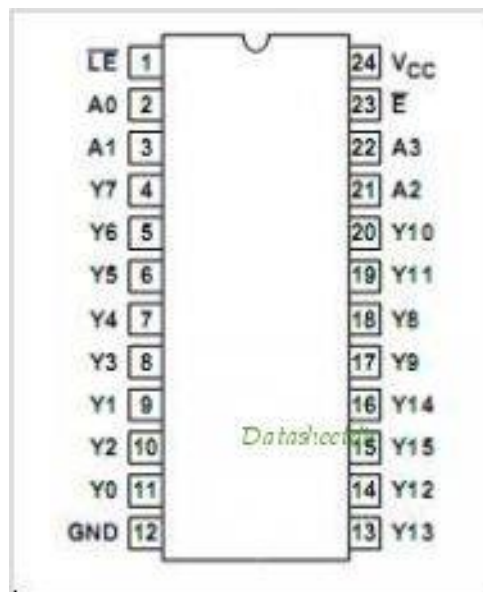


Fig. 26 – Pinout CD74HC4514

El circuito final entonces, utilizando todos los componentes antes dichos, quedaría construido de la siguiente manera.

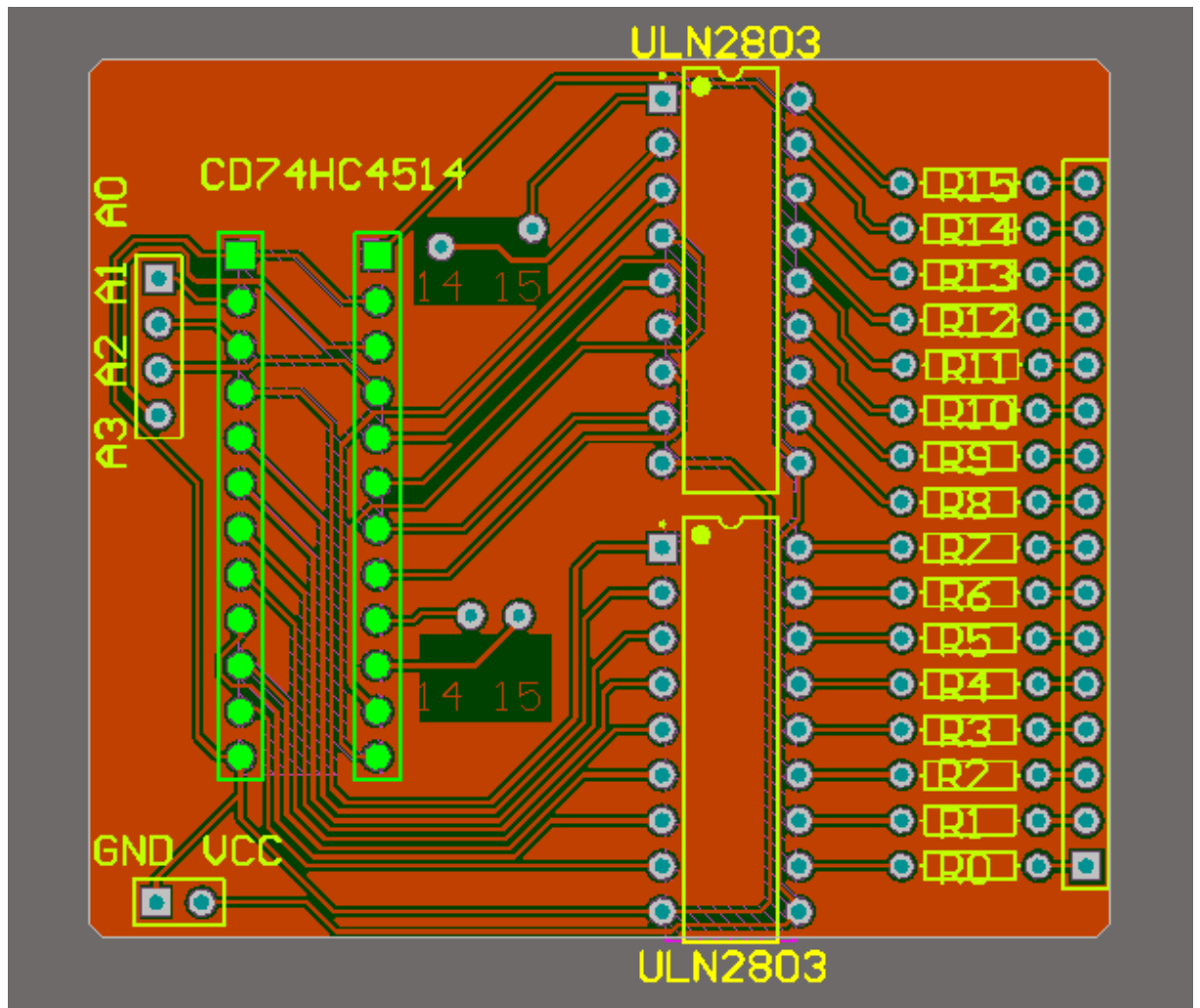


Fig. 27 – PCB circuito de control de filas.

Para controlar el encendido de cada Led, el sistema funciona de la siguiente manera:

- a. Supongamos que enumeramos las 60 columnas de ledes, siendo la primera la columna número 1, y la última la número 60.
- b. Supongamos también, que enumeramos las 16 filas, utilizando la letra “a” para la fila superior, y la letra “ñ” para la última.

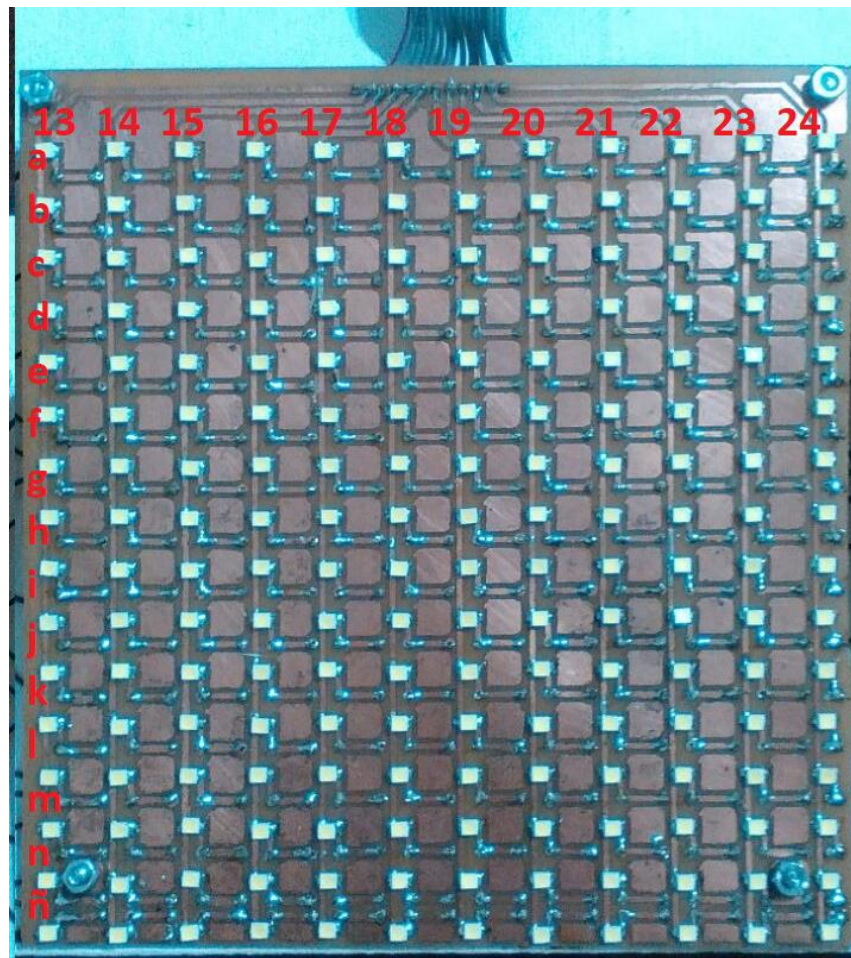


Fig. 28 – Matriz de ledes

c. Sabemos que en condiciones normales todos los ledes se encuentran apagados. Esto se logra aplicando tensión inversa a los terminales, es decir, las salidas del microcontrolador correspondientes a los cables superiores (que controlan las columnas) se encuentran en un estado lógico bajo, mientras que las salidas del decodificador, (y conjuntamente de los drivers de corriente) se encuentran en estado lógico alto. En estas condiciones es imposible que ningún Led se encuentre encendido.

d. Si deseamos entonces, encender el diodo correspondiente a la posición “24f” lo que debemos hacer es en primer lugar poner en un estado lógico Alto (5V) la salida del microcontrolador correspondiente a la columna 24.

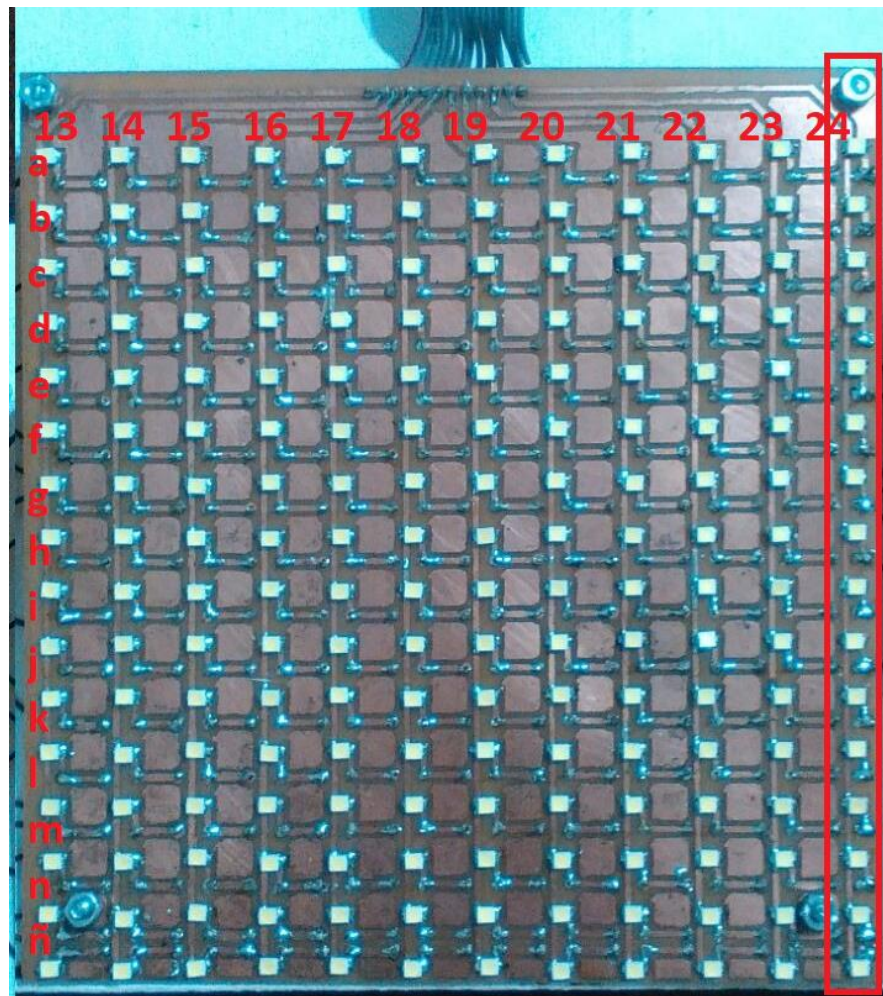


Fig. 29 – Matriz de ledes

Este proceso aún no encenderá ningún Led, ya que ahora ambos terminales de los diodos de toda la columna **24**, se encuentran al mismo potencial. A diferencia de los diodos de las demás columnas, que se encuentran polarizados inversamente.

**e.** A continuación, debemos cambiar el estado de la fila “**f**” a 0V, logrando entonces que sólo el diodo correspondiente a la posición “**24f**” esté polarizado directamente y se encienda.





Fig. 30 – Matriz de ledes

Cabe aclarar que todos los diodos de la columna 24 **excepto el 24f** poseen sus dos terminales a 5V (haciendo imposible su encendido), y todos los diodos de la fila “f” poseen ambas terminales a 0V, impidiendo también su encendido, con excepción del diodo correspondiente a la columna 24.

Este proceso se repetirá de la misma manera para todos los ledes que sea necesario encender.

Debido a que el microcontrolador puede cambiar el estado de todas sus salidas simultáneamente, es posible encender todos los ledes de una fila sin necesidad de multiplexar uno por uno. Por el contrario, el decodificador sólo puede activar una salida a la vez, por lo que es obligatorio el multiplexado de las filas una a una. Al hacerlo a una velocidad menor a una décima de

segundo, el parpadeo de los ledes se vuelve imperceptible para el ojo, creando un efecto óptico en el que pareciese que todas las luces están encendidas a la vez. Esto se llama “**Persistencia de la visión**” (ver anexo).

### • Microcontrolador

Una vez enviados los mensajes MIDI, estos deben ingresar a un microcontrolador que los reciba, interprete y ejecute las acciones correspondientes para iluminar los ledes indicados en la pantalla en el momento exacto que esto deba suceder. Para ello hemos utilizado el microcontrolador **ATMega2560**.

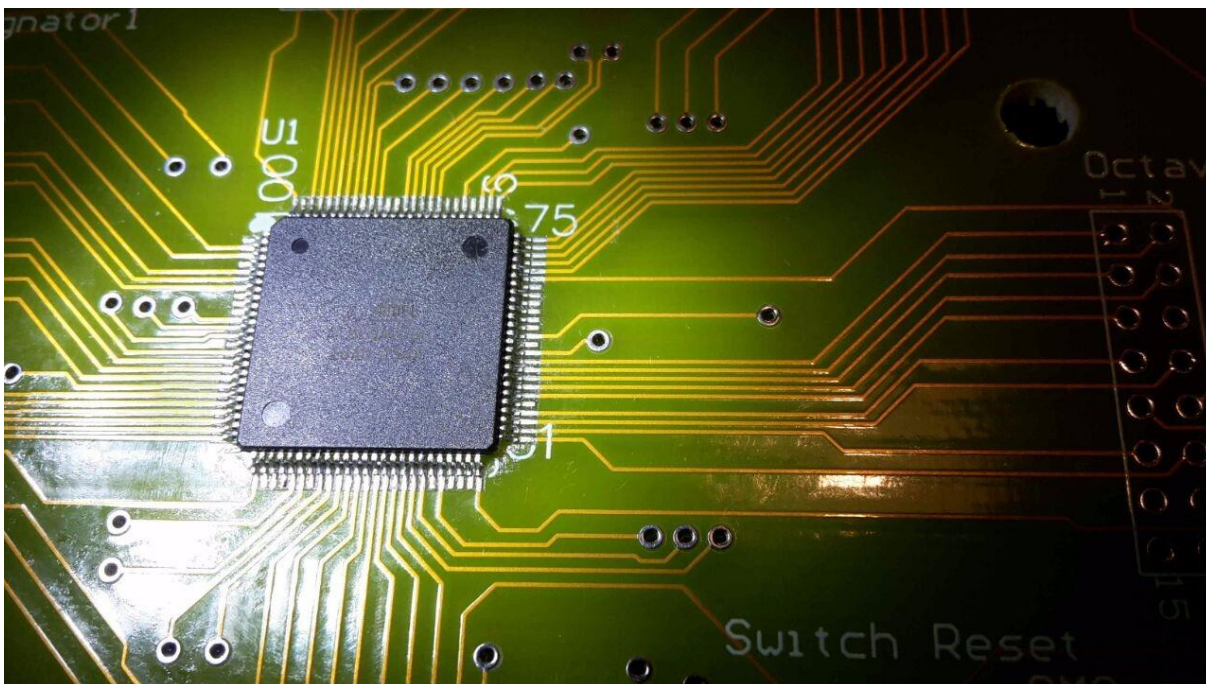


Fig. 31 – Microcontrolador ATMega2560 en **PianoGraph!**

Este microcontrolador de 8 bits posee 256KBytes de memoria Flash programable, y 87 salidas/entradas digitales, lo que lo hace ideal para nuestro propósito, ya que necesitamos 68 en total: 12 salidas por cada octava (lo que da un total de 60), 4 pines para la comunicación serial con los dispositivos MIDI (tanto el bluetooth que envía las instrucciones de la canción, como la realimentación opcional del teclado, para comparar las notas presionadas y otorgar un resultado de éxito/error), y 4 salidas más para manipular un decodificador de 4 a 16 salidas, para realizar el

multiplexado de la pantalla y dar la sensación de que todos los ledes están encendidos a la vez.

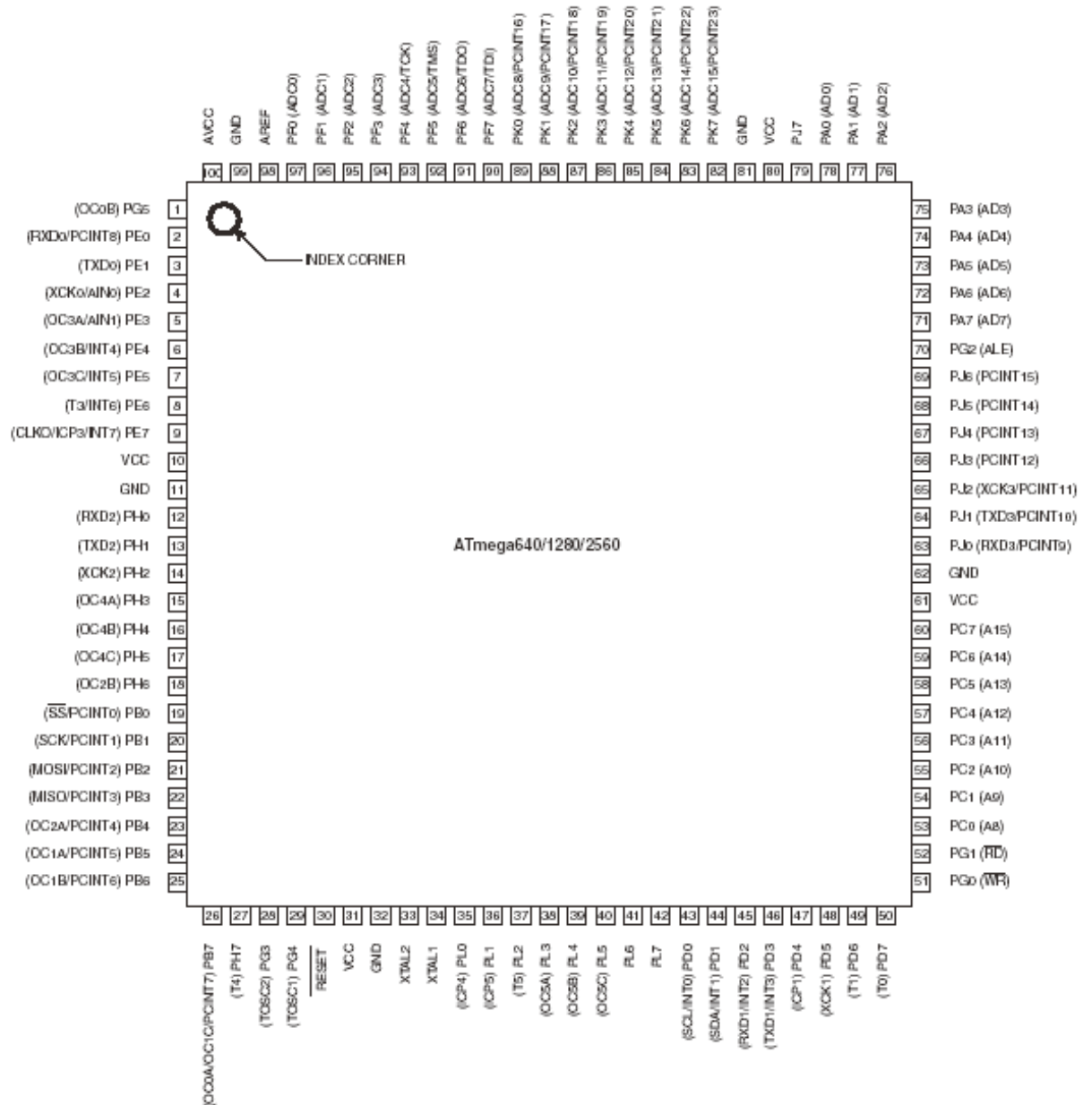


Fig. 32 – Pinout ATmega2560

Este microcontrolador tendrá que ejecutar diversas tareas simultáneamente. Debido a su costo, velocidad, y capacidad de procesamiento fue el elegido para darle vida a nuestro proyecto.

El microcontrolador deberá establecer una comunicación Bluetooth con el dispositivo que enviará el archivo MIDI a través de la aplicación desarrollada específicamente para este proyecto.

Para ello hemos asignado uno de los 3 puertos Seriales que ATmega2560 exclusivamente para “escuchar” los datos provenientes del dispositivo Bluetooth.

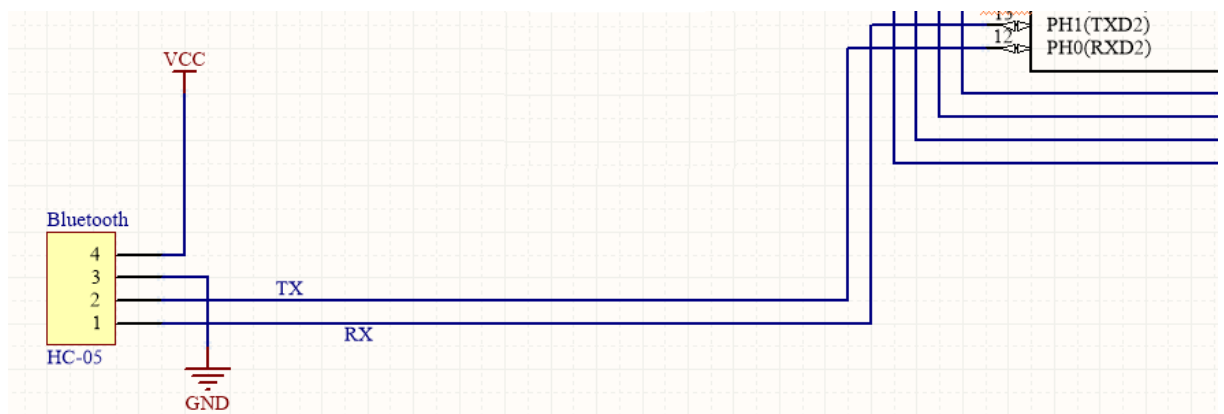


Fig. 33 – Bluetooth HC-06

Los pines de Transmisión y Recepción serial (Tx y Rx) se encuentran ubicados en la posición 12 y 13 respectivamente.

En toda comunicación serial entre dos dispositivos que presenten estos pines (Tx y Rx) es necesario comunicarlos inversamente, es decir, unir mediante cables el pin de transmisión de un dispositivo, con el de recepción del otro, y viceversa.

Para este proyecto, decidimos utilizar el módulo Bluetooth **HC-06**, debido a que es un dispositivo simple de utilizar, sólo requiere alimentación y establecer la comunicación serial mediante 2 pines (Tx y Rx). Además, generalmente no es necesaria ninguna configuración, sólo se conecta y comienza a funcionar. No obstante, el mismo fue configurado para trabajar a una velocidad de 115200 baudios y le modificamos su nombre para que muestre “**PianoGraph!**” al ser encontrado por otro dispositivo Bluetooth.





Fig. 34 – Bluetooth HC-06

Para establecer la comunicación Bluetooth con el módulo HC-06 desde un dispositivo Android simplemente es necesario encender el Bluetooth desde el dispositivo, abrir la aplicación “**PianoGraph!**”, y seleccionarlo entre la lista de dispositivos disponibles. Una vez realizado este paso, el módulo está listo para recibir información y transmitirla al microcontrolador.



Fig. 35

Una segunda función que deberá desempeñar el microcontrolador es la de interpretar y discriminar entre los mensajes recibidos que correspondan a mensajes

MIDI para ejecutar en la pantalla Led, y los mensajes que no se apliquen a este fin y deban ser ignorados.

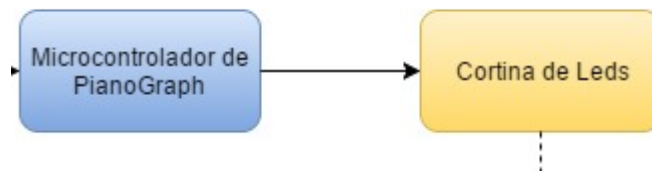


Fig. 36

Como tercera función, el microcontrolador deberá desempeñar la tarea de encender las luces de la pantalla led, alternando el estado lógico de 16 salidas digitales correspondientes a las filas y activar o desctivar las columnas correspondientes a cada fila encendida en ese momento.

Como cuarta función, el microcontrolador también presenta una entrada MIDI, mediante la cual es posible conectar la salida MIDI (si es que posee) del teclado/piano que se esté tocando.

El microcontrolador “escuchará” también las instrucciones MIDI enviadas por el teclado, y comparará estas instrucciones con el estado de la última fila de la pantalla led, si dicha fila indica que una nota “x” debe estar encendida, y del teclado llega la instrucción que esa nota está siendo presionada, se añadirá puntaje a una variable de éxito.

Si el caso inverso sucede, (hay que presionar una tecla y no está siendo presionada, o hay una tecla presionada que no había que presionar), se incrementará una variable de error.

El puntaje final será obtenido como porcentaje de puntos correctos entre puntos totales.



Fig. 37

Al finalizar la canción, la aplicación Android está diseñada para enviar una instrucción no utilizada por los comandos MIDI, para indicarle al microcontrolador que la melodía finalizó y que debe devolver el puntaje final.

El microcontrolador devolverá entonces a través del módulo Bluetooth, un dato indicando el puntaje obtenido, calculado mediante un algoritmo que otorgará un número comprendido entre el 0% de aciertos y el 100%.

La programación del microcontrolador se realizó en lenguaje C y a continuación se muestran los diagramas de flujo del código correspondiente:

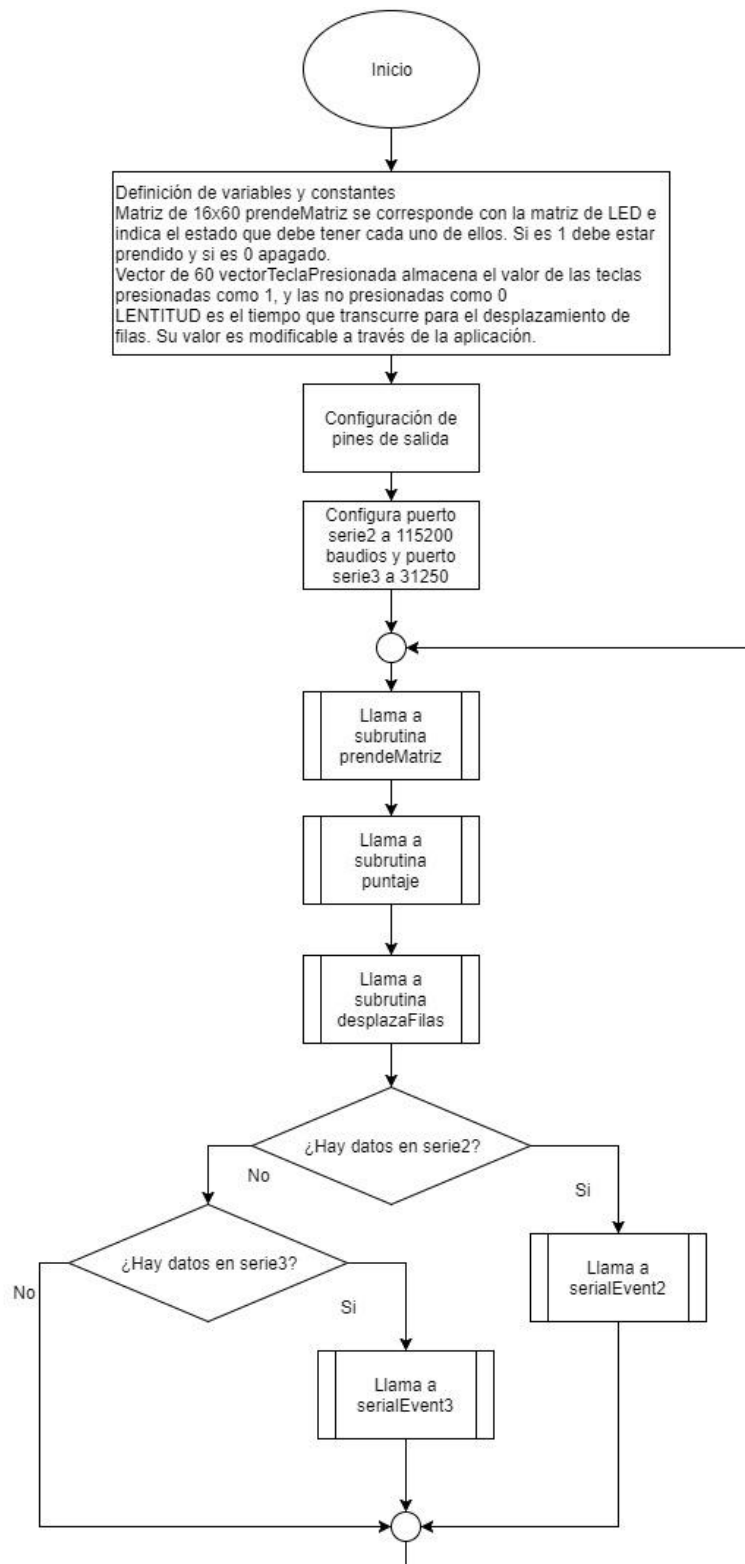


Fig. 38 – Rutina principal

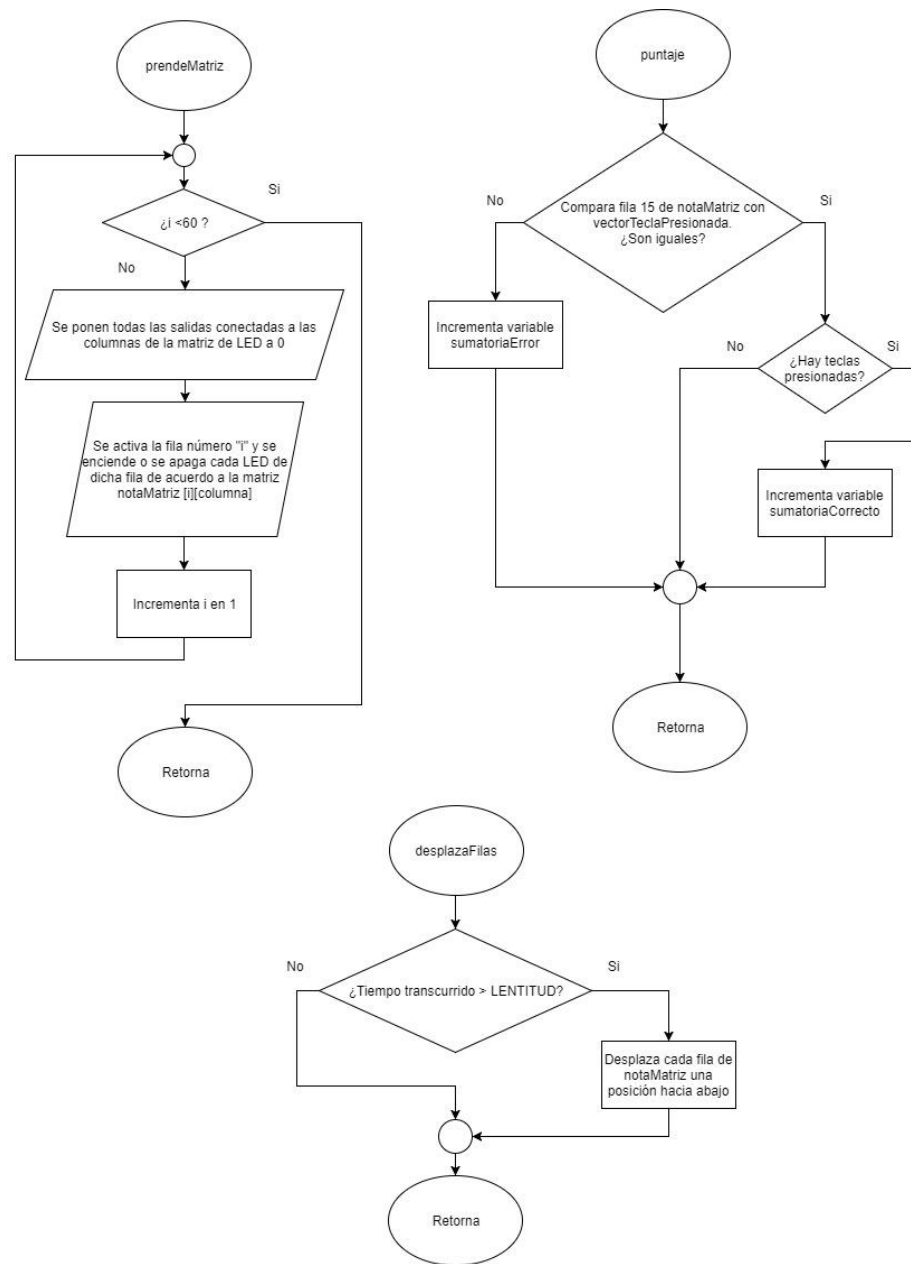


Fig. 39 - Subrutinas

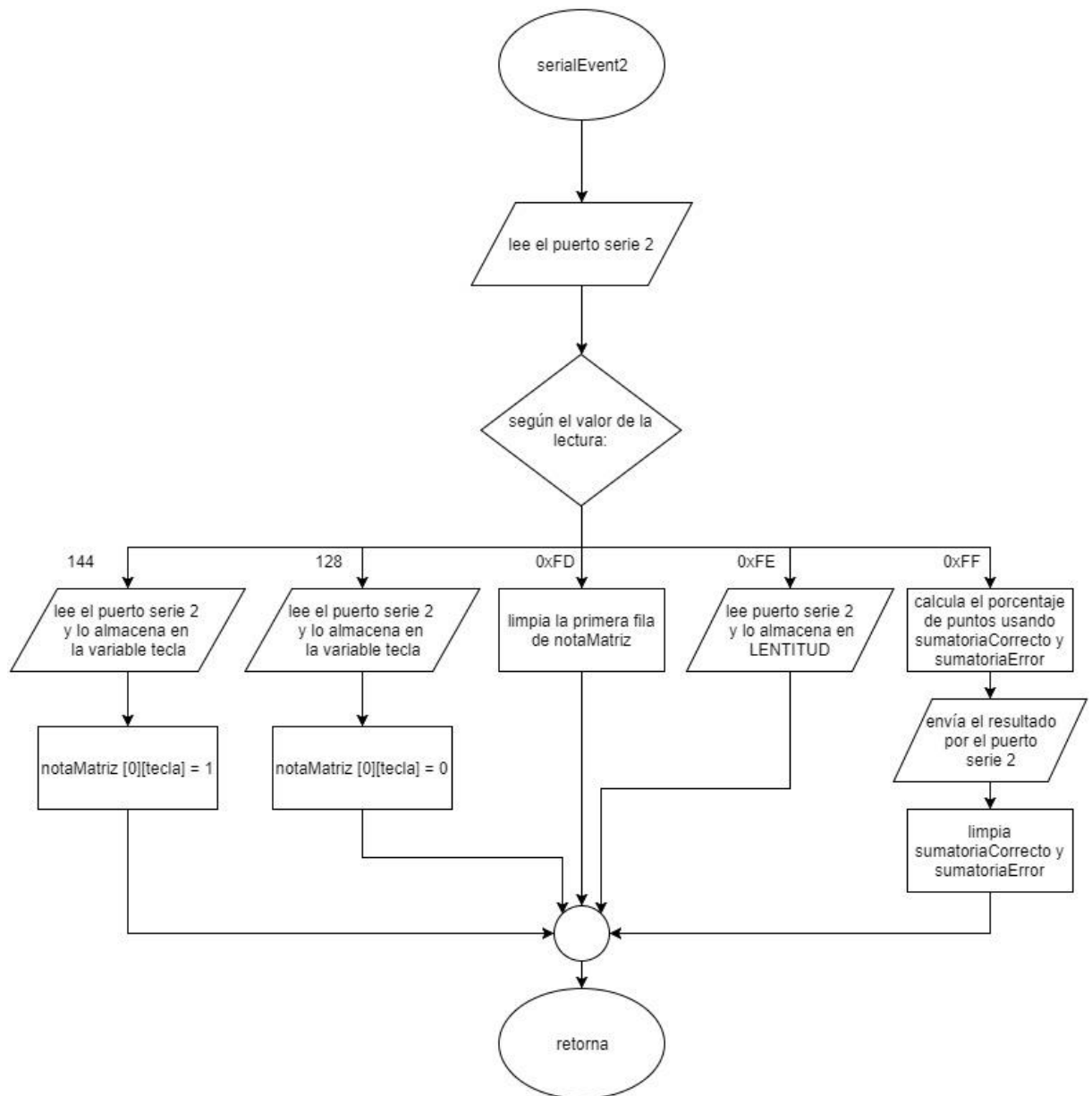


Fig. 40 – Atención del puerto serial de parte del Bluetooth

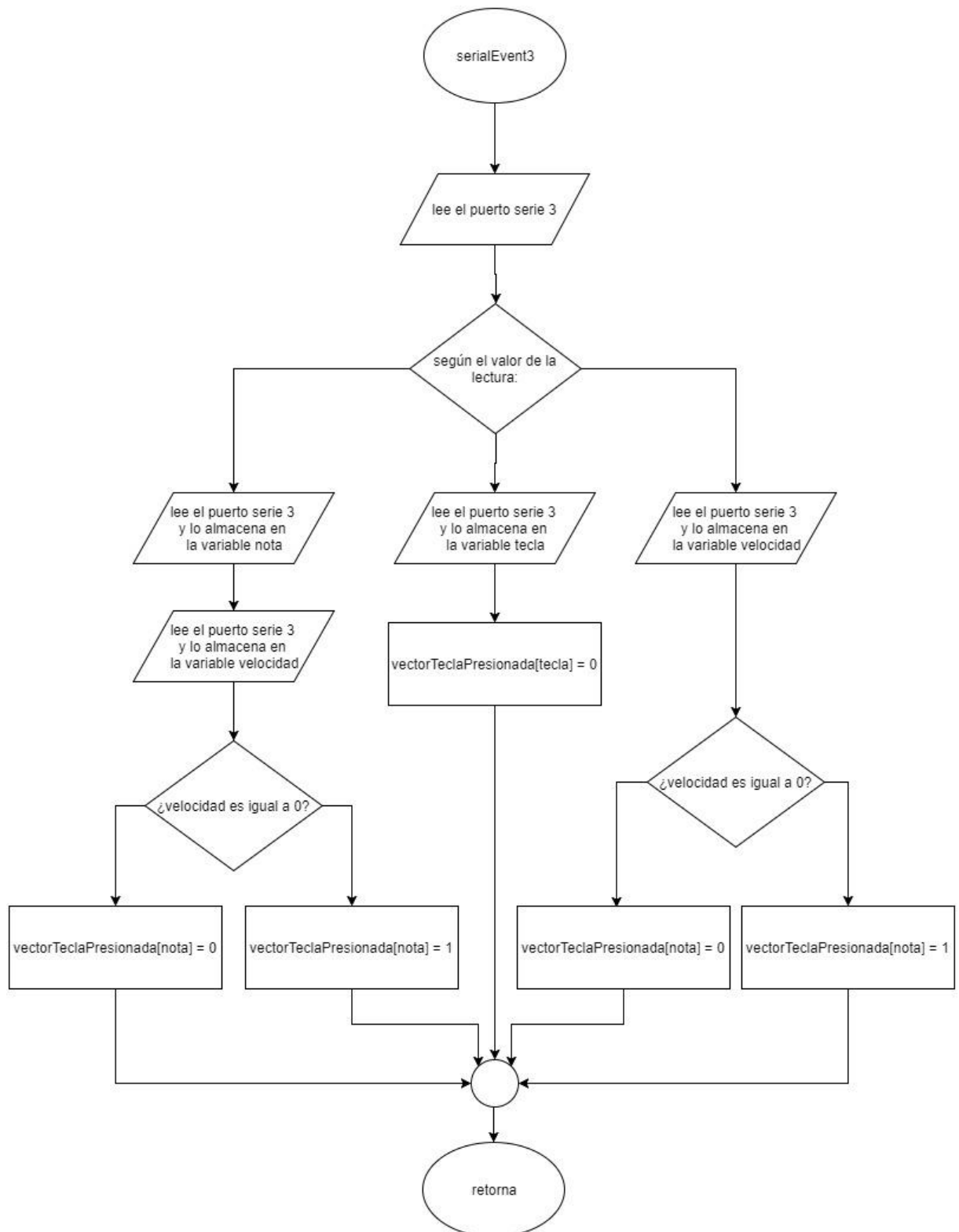


Fig. 41 – Atención del puerto serial de parte del Teclado

La placa final fue diseñada utilizando el programa Altium, y el PCB se envió a fabricar por la empresa **Ernesto Mayer S.A**

Esquema eléctrico:

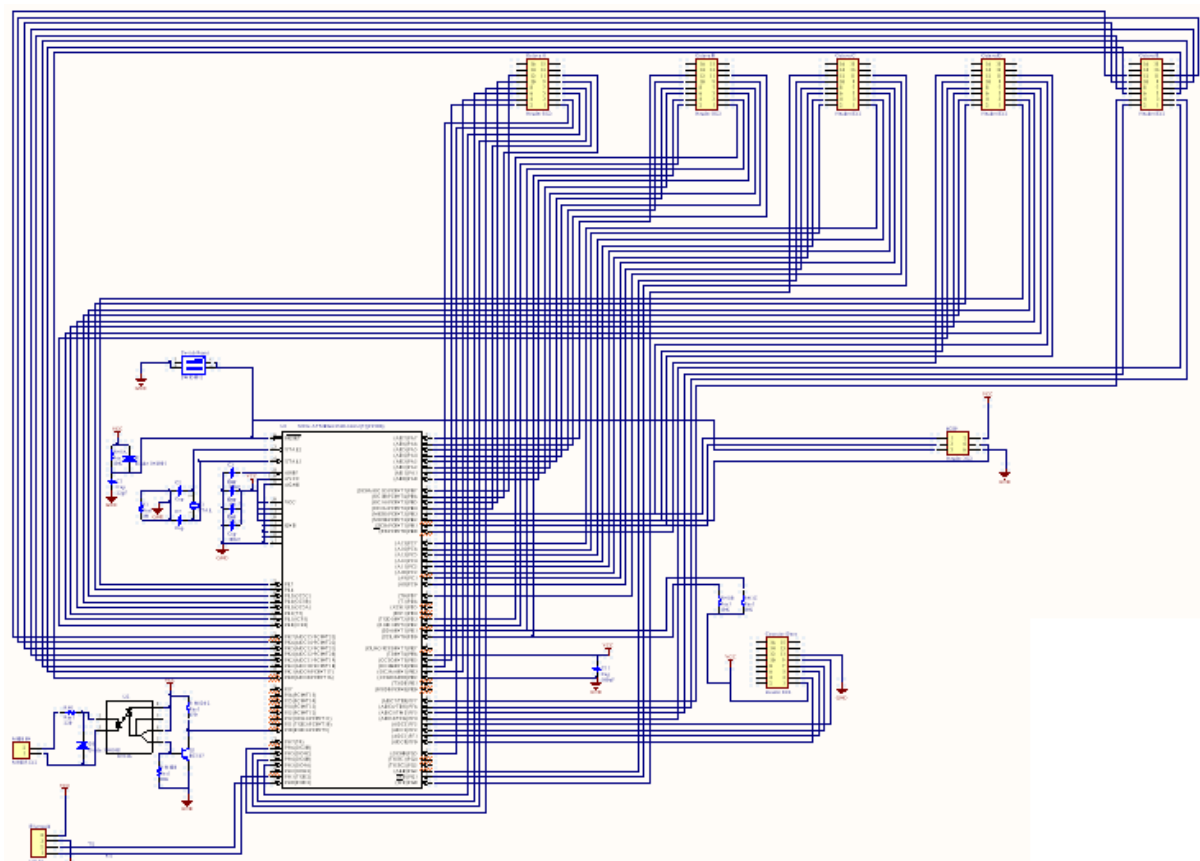
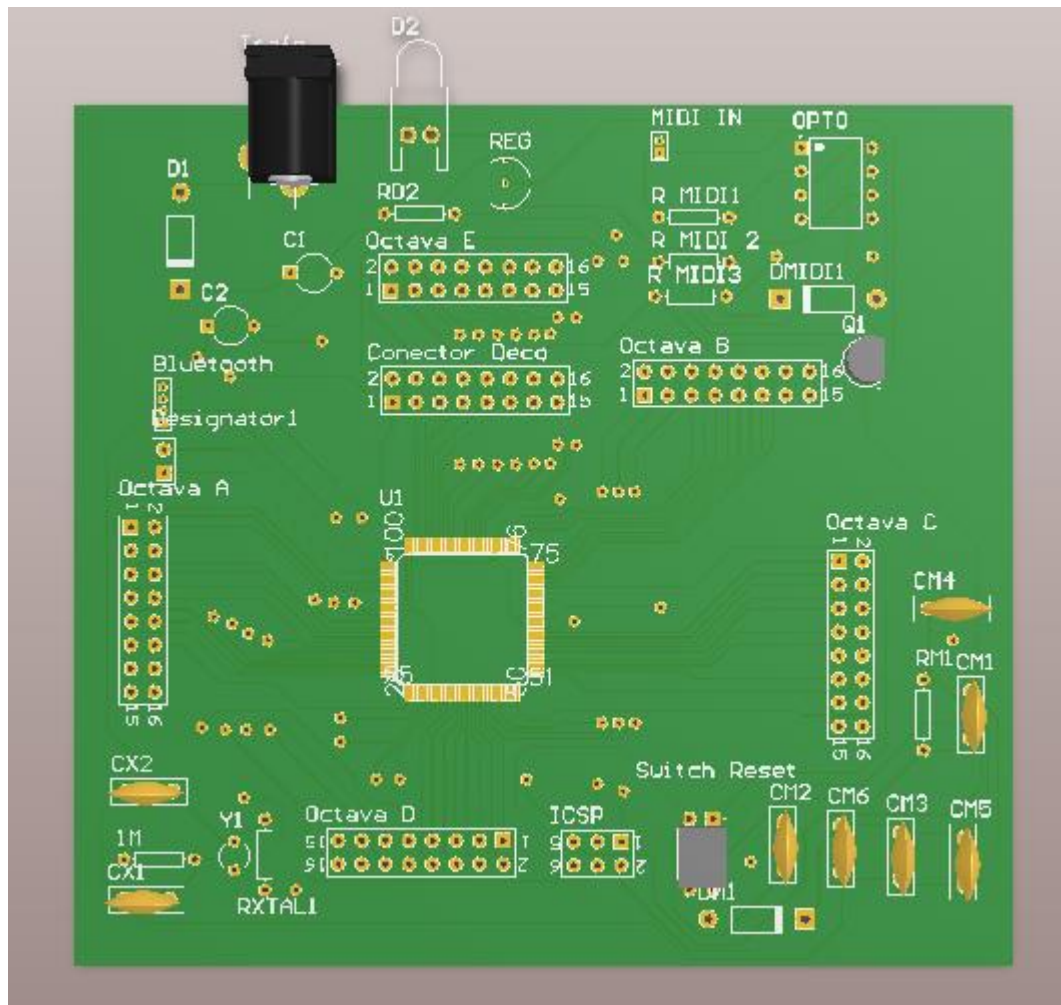


Fig. 42 – Esquemático Placa controladora





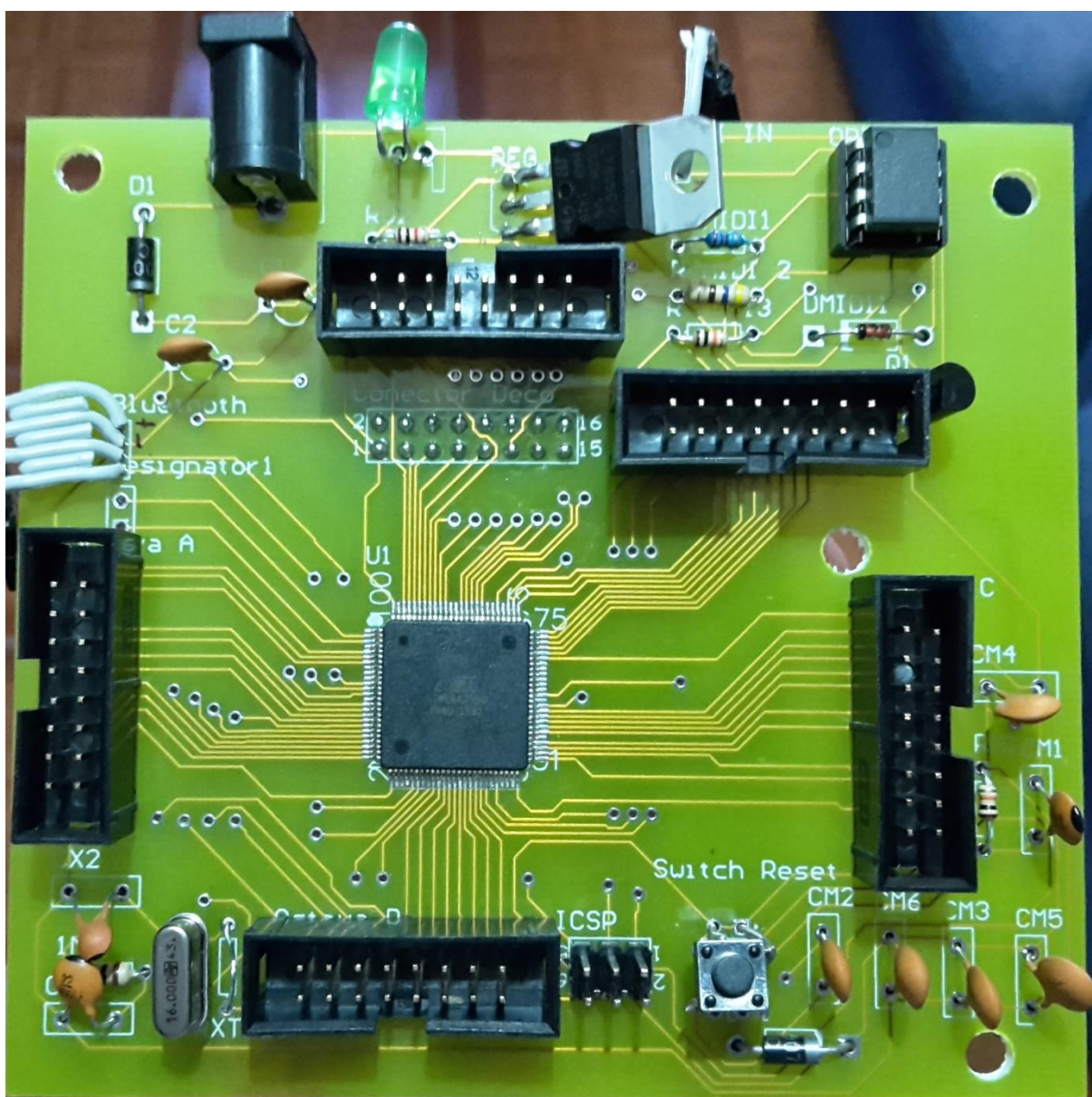


Fig. 44 – Prototipo PCB

## • Aplicación Android

La aplicación desarrollada para Android se concibió utilizando la plataforma gratuita “**Android Studio**” para lo cual fue necesario incorporar una gran cantidad de conceptos nuevos y conocimientos nunca vistos en la carrera. Dicha aplicación se encuentra en etapa “beta” aún, pero se comporta de manera operativa, y es posible utilizarla para manejar “**PianoGraph!**”.

La aplicación realiza las siguientes tareas:

1. Conexión inalámbrica con el dispositivo.

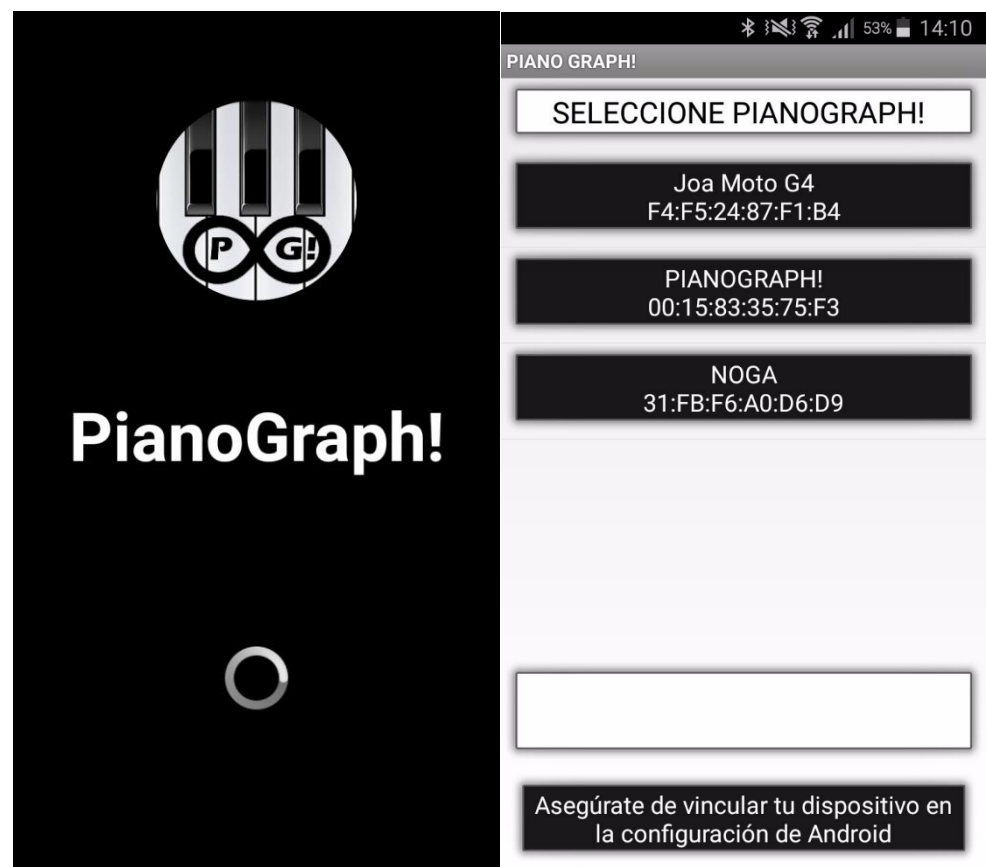


Fig. 45 – PianoGraph!

2. Selección del archivo MIDI descargado, y comprensión del mismo



Fig. 46 y 47 – PianoGraph!

3. Envío de dicha canción a “**PianoGraph!**”, pudiendo controlar variables como la velocidad de muestreo del dispositivo, y visualizar inmediatamente el puntaje obtenido una vez finalizada la canción.

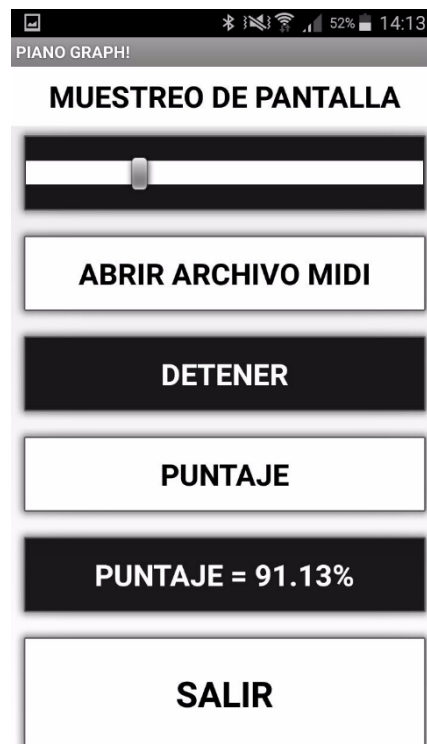


Fig. 48 – PianoGraph!

## • Ensamblado y Carcasa

Para la realización de la carcasa, hemos utilizado piezas diseñadas por nosotros y producidas gracias a la tecnología de impresión 3D, utilizando una impresora “Geeetech Prusa i3”



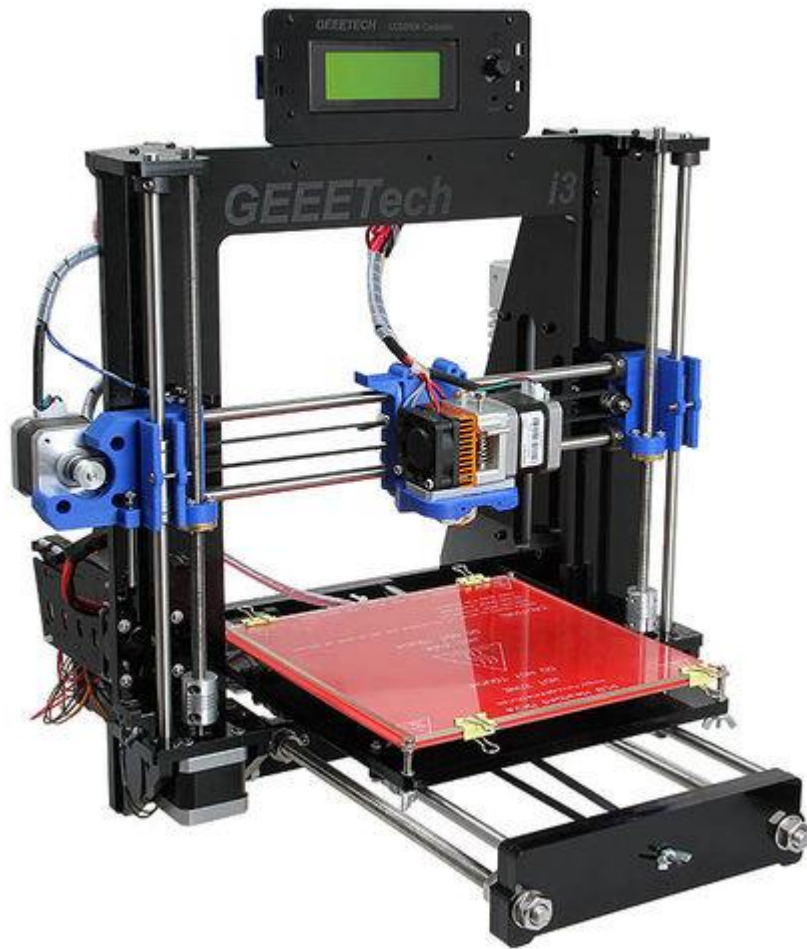


Fig. 49 – Geeetech Prusa i3

Una **impresora 3D** es una máquina capaz de realizar réplicas de diseños en 3D, creando piezas o maquetas volumétricas a partir de un diseño hecho por ordenador, descargado de internet o recogido a partir de un escáner 3D. Surgen con la idea de convertir archivos de 2D en prototipos reales o 3D. Comúnmente se ha utilizado en la prefabricación de piezas o componentes, en sectores como la arquitectura y el diseño industrial. En la actualidad se está extendiendo su uso en la fabricación de todo tipo de objetos, modelos para vaciado, piezas complicadas, alimentos, prótesis médicas (ya que la impresión 3D permite adaptar cada pieza fabricada a las características exactas de cada paciente), etc.

La impresión 3D en el sentido original del término se refiere a los procesos en los que secuencialmente se acumula material en una cama o plataforma por diferentes

métodos de fabricación, tales como polimerización, inyección de aporte, inyección de aglutinante, extrusión de material, cama de polvo, laminación de metal, depósito metálico.

Las piezas diseñadas consisten en una lámina de aproximadamente 100mm x 130mm, que se encuentra subdividida en celdas de 10x13.5mm, que corresponden a un cobertor de un led. Teniendo entonces 192 celdas de plástico traslúcido, cubriendo cada una un led diferente.

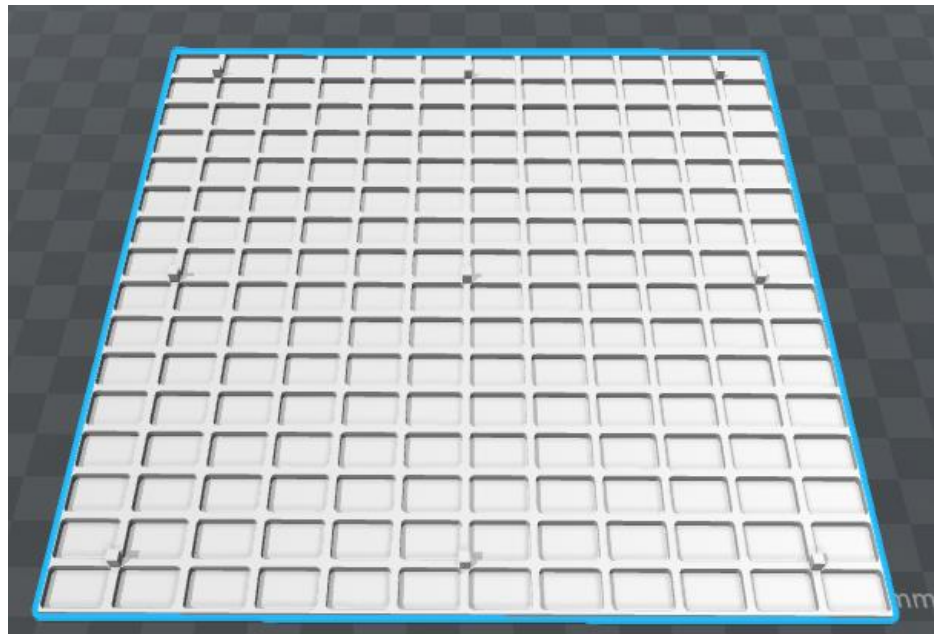


Fig. 50 – Cobertor de 192 ledes

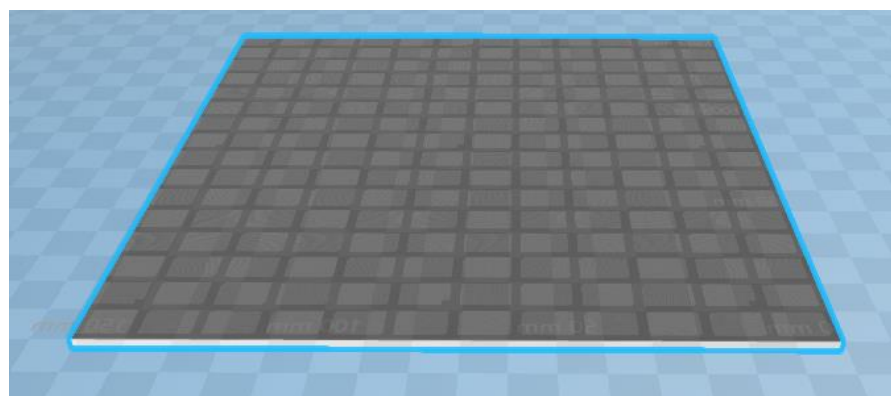


Fig. 51 – Cobertor de 192 ledes

Una vez impresas las 5 piezas correspondientes una a cada octava del dispositivo, procedimos a diferenciar las columnas correspondientes a las notas “sostenidas” o “bemoles” (notas identificadas con el color negro en los pianos), cubriendo éstas con papel vinilo semitransparente de color negro. Siendo el resultado final el siguiente:

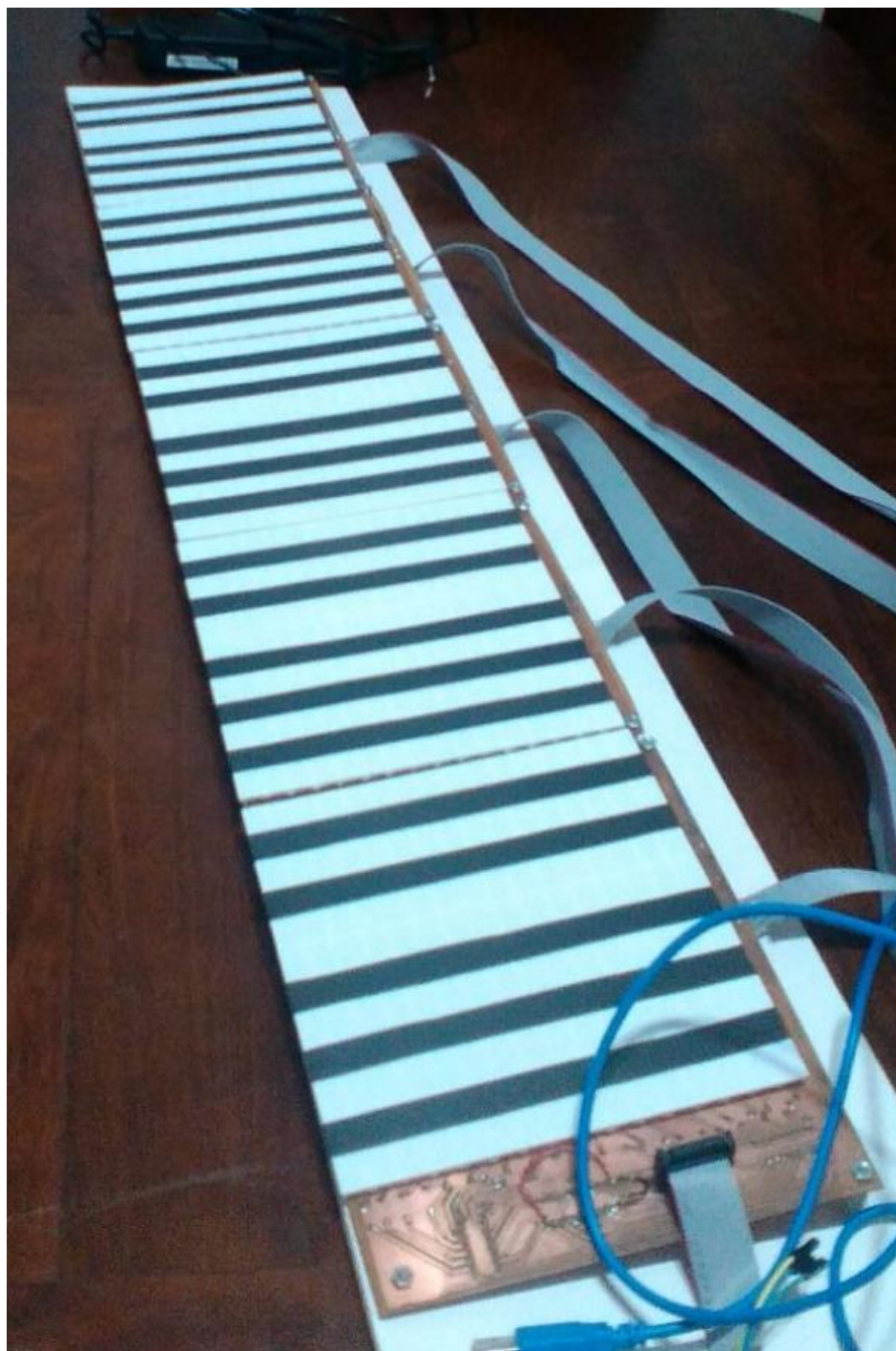


Fig. 52 – Primer montaje



Luego de comprobar que el dispositivo funcionaba correctamente, hemos decidido mejorar la estética del mismo, y volver a diseñar las piezas que componen la carcasa del mismo, además reduciendo los tiempos de ensamblaje mediante la fabricación de piezas más fáciles de montar. Todas estas piezas fueron diseñadas utilizando el software “**SolidWorks**”

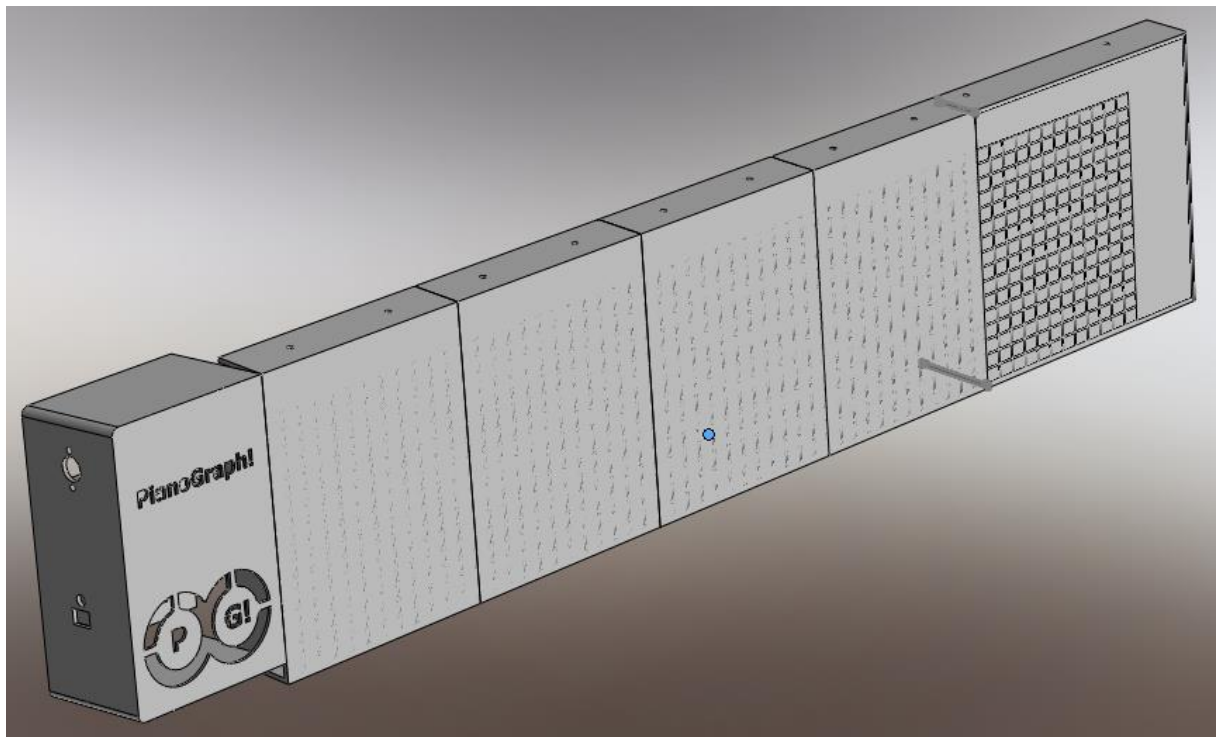


Fig. 53 – Ensamblaje final



Fig. 54 – Dispositivo listo

## CONCLUSIONES

Este proyecto presentó un gran desafío personal y profesional para nosotros, en el cual debimos aplicar una amplia gama de conocimientos adquiridos durante nuestra formación académica, y además, incorporar gran cantidad de nueva información quizás lejos de estar relacionada con la electrónica, pero a su vez igual o aún más necesaria, ya sean aspectos mecánicos, de diseño, económicos, de organización de un proyecto, y demás elementos siempre presentes a la hora de encarar un emprendimiento (algo que deberemos enfrentar a partir de ahora y a lo largo de nuestra carrera profesional).

La concepción inicial de la idea no incluía el desarrollo de una aplicación Android exclusiva para el dispositivo ni la obtención de un puntaje, pero gracias a la cooperación de ambos, la investigación, la asistencia siempre disponible de parte de la UTN, las herramientas ofrecidas por nuestros mentores a lo largo de estos casi seis años de formación y la ayuda de compañeros, hemos conseguido superar los primeros objetivos y avanzar para conseguir un prototipo completo del producto final deseado.

Además, este proyecto fue realizado con gran empeño, ya que consideramos real la posibilidad de producción y comercialización. Tenemos grandes expectativas para comenzar su fabricación en el transcurso del año 2018. No queremos, como tantos otros trabajos finales, que éste sea sólo una calificación más en nuestro certificado analítico, y que sea guardado en un archivo, en el olvido.

No nos quedan más palabras que de agradecimiento para todos aquellos que nos acompañaron en este camino.

## APÉNDICE

Descargas MIDI: [www.8notes.com](http://www.8notes.com)

Información útil: <http://www.css-audiovisual.com/areas/guias/midi.htm>

Datasheet Mega2560 : <http://pdf1.alldatasheet.com/datasheet-pdf/view/107092/ATMEL/ATMEGA2560.html>

Datasheet optoacopladores:

PC817: <http://pdf1.alldatasheet.com/datasheet-pdf/view/43368/SHARP/PC817.html>

4N26: <http://pdf1.alldatasheet.com/datasheet-pdf/view/2847/MOTOROLA/4N26.html>

6N136: <http://pdf1.alldatasheet.com/datasheet-pdf/view/35184/QT/6N136.html>

Datasheet CD74HC4514: <http://pdf1.alldatasheet.com/datasheet-pdf/view/26940/TI/CD74HC4514.html>

Datasheet ULN2803: <http://pdf1.alldatasheet.com/datasheet-pdf/view/12687/ONSEMI/ULN2803.html>

Web Arduino (soporte y comunidad): <https://www.arduino.cc/>

Datasheet ledes: <https://www.dled.com.ar/producto/download-documento/?id=53&pr=280>

## BIBLIOGRAFÍA

- Instituto Nacional de Estadística y Geografía
- Instituto de Sociología Musical de la Universidad de Viena de Música y Arte Dramático.
- Sergi Jordà Puig, Audio digital y MIDI, Guías Monográficas Anaya Multimedia, Madrid 1997
- The MIDI Manual, Second Edition, Libro de David Miles Huber
- El Sistema Midi, Libro de Antonio Mínguez Olivares, Antonio Pedrero González, Fernando Garcia Moreno y Francisco J. Tabernero Gil

## ANEXO I

### PERSISTENCIA DE LA VISIÓN

La persistencia de la visión fue un supuesto fenómeno visual descubierto por Peter Mark Roget que demostraría como una imagen permanece en la retina humana una décima de segundo antes de desaparecer por completo.

Según sus estudios, esto permitiría que veamos la realidad como una secuencia de imágenes ininterrumpidas y que podamos calcular fácilmente la velocidad y dirección de un objeto que se desplaza, si no existiese, veríamos pasar la realidad como sucesión de imágenes independientes y estáticas.

Joseph-Antoine Ferdinand Plateau creyó descubrir que nuestro ojo ve con una cadencia de diez imágenes por segundo. En virtud de dicho fenómeno, las imágenes se superponen en la retina y el cerebro las "enlaza" como una sola imagen visual, móvil y continua.

Se supuso que el cine aprovechaba este efecto y provoca ese "enlace" proyectando a más de 10 imágenes por segundo (generalmente 24), lo que genera en nuestro cerebro la ilusión de movimiento (la televisión se da a 29,97 fotogramas por segundo, y 25 en Europa).

#### Ley de Ferry-Porter

Hay un amplio margen de frecuencias desde la absoluta evidencia de parpadeo hasta la sensación de continuidad. Este fenómeno se conoce con el término inglés flicker. La frecuencia en la que las imágenes se funden y desaparece el parpadeo lo llamamos frecuencia crítica de flicker y, como ya hemos apuntado, depende del nivel de iluminación de la pantalla. Es una dependencia de tipo logarítmico y se conoce como ley de Ferry-Porter:

$$f_{lindar} = 37 + 12,6 \log(B)$$

B = nivel de brillo de la pantalla medido en foot-lambert.

Se considera que la frecuencia umbral o crítica es del orden de 40 Hz (algo más baja en un cine por ejemplo donde las condiciones de iluminación se adecuan al visionado). Sin embargo, esta frecuencia podrá variar entre los 60 Hz (fuerte iluminación diurna) hasta 4 Hz (iluminación nocturna y visión fotópica). Esto se explica gracias a la naturaleza de los conos y bastones. Los primeros (conos) captan las imágenes en entornos de alta iluminación y lo hacen con una respuesta bastante elevada. Por el contrario, los bastones son mucho más sensibles, de ahí que se encarguen de captar en entornos de iluminación precaria. Ahora bien, necesitan mucho más tiempo para excitarse y volver a la situación inicial de reposo (hasta 0,25 segundos).

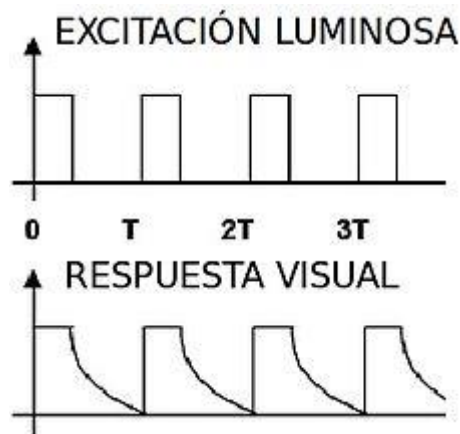


Fig. 55



## Declaración Jurada de Originalidad del

### Proyecto Final

Los autores del Proyecto Final titulado: Sistema de aprendizaje virtual para piano,

#### Declaramos bajo Juramento:

1. Que el Proyecto Final no ha sido plagiado ni total ni parcialmente, para la cual se han respetado las normas internacionales de citas y referencias para las fuentes consultadas.
2. Que el Proyecto Final presentado no atenta contra derechos de terceros.
3. Que el Proyecto Final no ha sido publicado ni presentado anteriormente para obtener algún grado académico previo o título profesional.
4. Que los datos presentados en los resultados son reales, no han sido falsificados, ni duplicados, ni copiados.
5. Que se adjuntará al presente un acuerdo formal de la parte involucrada, si el proyecto tuviese información relacionada sobre una empresa o institución.

Por lo expuesto, mediante la presente asumo frente LA UNIVERSIDAD cualquier responsabilidad que pudiera derivarse por la autoría, originalidad y veracidad del contenido del Proyecto Final, así como por los derechos sobre la obra y/o invención presentada. En consecuencia, me hago responsable frente a LA UNIVERSIDAD y frente a terceros, de cualquier daño que pudiera ocasionar a LA UNIVERSIDAD o a terceros, por el incumplimiento de lo declarado o que pudiera encontrar causa en El Proyecto Final presentado, asumiendo todas las cargas que pudieran derivarse de ello.

De identificarse fraude, piratería, plagio, falsificación o que el Proyecto Final haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, sometiéndome a la normatividad vigente de la UTN.

Lugar y fecha: San Francisco, Córdoba, 27 de noviembre de 2017.

**Integrantes del grupo:**

Nombre: Daniele, Fernando Ezequiel .....

Firma.: \_\_\_\_\_

Nombre: Velazquez, Gonzalo Martín .....

Firma.: \_\_\_\_\_