MINI PIANO(TOUCH)



Integrantes del grupo: Jonas Pereyra, Máximo Cabral, Juan Martin Romero, Mateo Shinocca y Luciano Ojeda.

Profesor: Gonzalo Consorti

Alumno: Juan Martin Romero

Curso: 4°1 T.T

Proyecto: Mini instrumento electro-estático (touch)

Viernes 25/10/2024

**INTRODUCCIÓN AL PROYECTO**

El mini instrumento electroestático (touch) es un proyecto cuyo objetivo principal es la creación de un pequeño piano electrónico utilizando la plataforma de Arduino. Este proyecto utiliza varios componentes electrónicos, entre ellos, el DF Player Mini HW-247A, que se encarga de reproducir las notas musicales a través de un altavoz. La idea detrás de este mini piano es poder tocar diferentes notas mediante sensores táctiles (touch sensors), los cuales, al ser activados por el usuario, emiten diferentes frecuencias. Además, para enriquecer la funcionalidad del instrumento, se incorporó un módulo codificador rotatorio (Rotary Encoder), que permite al usuario cambiar la escala musical de las notas que se generan, dándole mayor versatilidad al dispositivo.

El proyecto busca, no solo construir un instrumento musical interactivo, sino también proporcionar a los integrantes del grupo una comprensión más profunda sobre cómo generar notas musicales utilizando código de programación y cómo gestionar entradas digitales en un microcontrolador como el Arduino Uno R3. Esto representa una introducción práctica al campo de la electrónica interactiva y la programación de dispositivos embebidos.

**MATERIALES Y COMPONENTES**

A continuación se detallan los materiales y componentes clave utilizados en este proyecto:

* **1x Arduino Uno R3**: El microcontrolador principal que actúa como el "cerebro" del proyecto. Recibe las señales de los sensores táctiles y del codificador rotatorio, y controla el **DF Player Mini** para emitir los sonidos correspondientes.
* **1x USB para programar**: Cable necesario para conectar el Arduino a un ordenador y cargar el código.
* **1 x Protoboard**: Plataforma de pruebas utilizada para conectar los componentes sin necesidad de soldaduras. Facilita las pruebas y ajustes del circuito.
* **7x Sensor Touch**: Sensores táctiles que funcionan como teclas de piano. Al tocarlos, envían una señal digital al Arduino, que luego activa la reproducción de la nota correspondiente en el **DF Player Mini**.
* **1x DF Player Mini HW-247A**: Módulo de audio que reproduce las notas musicales. Se utiliza para emitir las frecuencias correspondientes a las notas musicales cuando se presionan los sensores táctiles.
* **Cables**: Se utilizaron una gran cantidad de cables para conectar todos los componentes entre sí y al Arduino, permitiendo el flujo de señales y alimentación.
* **Rotary Encoder**: Un dispositivo mecánico-electrónico que convierte el movimiento rotatorio en señales digitales. En este proyecto, se utiliza para cambiar las notas y variar la escala musical que se está reproduciendo.
* **TRRS Audio Jack**: Conector estándar para transmitir señales de audio (y en este caso también de micrófono) entre el DF Player Mini y un dispositivo de salida, como un altavoz o audífono. Este tipo de conector es común en dispositivos de audio, como teléfonos o reproductores MP3.

#### **Descripción de los Componentes Clave:**

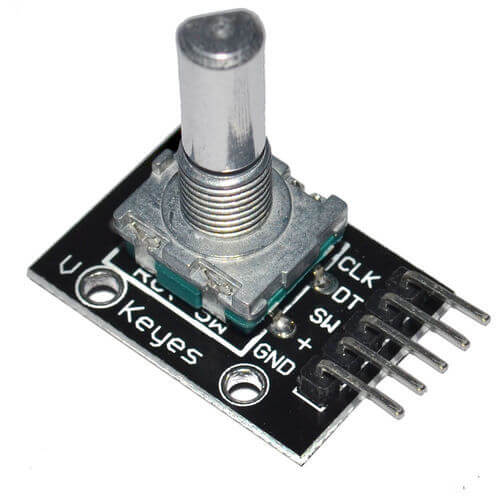
**Sensor Touch**: Este tipo de sensor es capaz de detectar la presencia de una mano u objeto conductor de electricidad sin necesidad de contacto físico directo. Utiliza la capacitancia para determinar cuando un dedo toca el área sensitiva, lo que lo convierte en un componente ideal para crear una interfaz de usuario sin necesidad de botones físicos.



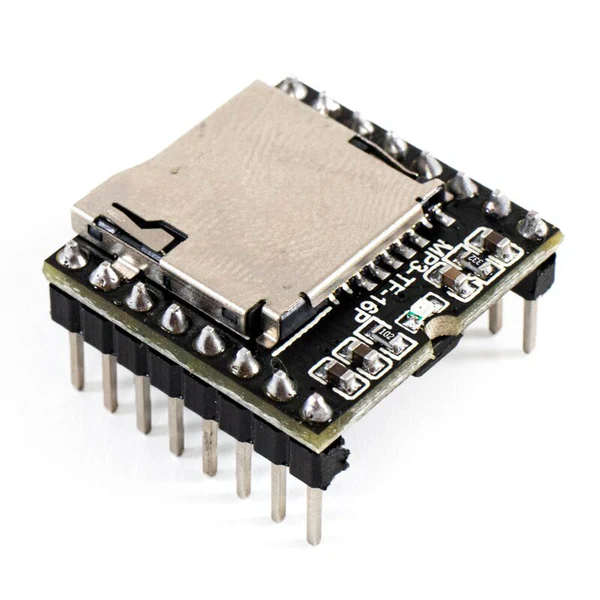
**TRRS Audio Jack**: Los conectores TRRS (Punta, Anillo, Anillo, Manga) son más comunes en dispositivos de audio y permiten la transmisión de señales de audio estéreo y micrófono en un solo cable. Este conector es esencial para transmitir el audio generado por el **DF Player Mini** al sistema de salida.



**Rotary Encoder**: El codificador rotatorio es una pieza clave para la interacción avanzada con el instrumento, ya que permite al usuario modificar parámetros como la frecuencia de las notas o la escala musical en tiempo real, brindando una experiencia musical dinámica.



**DF Player Mini HW-247A:**El DFPlayer Mini es un módulo compacto y versátil para reproducción de audio que soporta formatos comunes como MP3, WAV y WMA. Está diseñado para ser fácilmente integrable en proyectos de electrónica con o sin un microcontrolador gracias a su capacidad de trabajar con Arduino u otros sistemas de desarrollo. Además, cuenta con una ranura para tarjeta SD, un amplificador integrado para conectar directamente a un altavoz, y múltiples opciones de control incluyendo botones en el propio módulo para operación manual.

****

**CONFIGURACIÓN Y ESQUEMAS DE CONEXIONES**

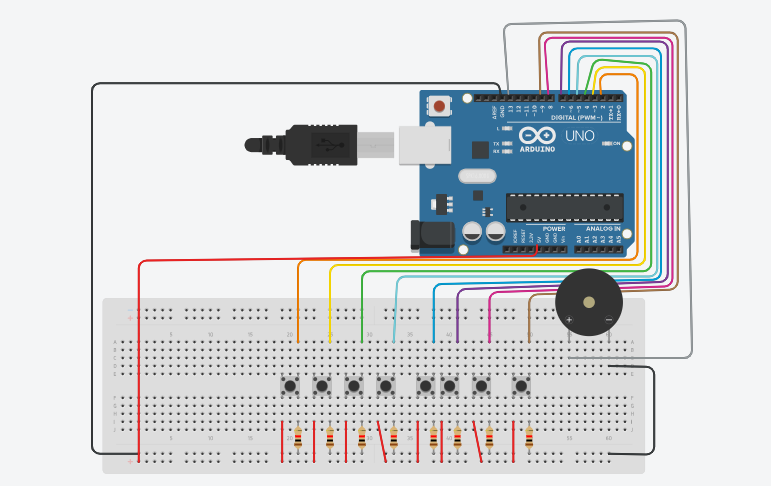
El esquema de conexiones de los componentes con el **Arduino Uno R3** es esencial para asegurar que cada parte del proyecto funcione de manera correcta. A continuación, se describe cómo se conectan los componentes:

1. **Sensor Touch**: Cada uno de los siete sensores táctiles se conecta a los pines digitales del Arduino. Estos sensores están conectados a través de purines digitales que permiten detectar el toque de manera eficiente.
2. **DF Player Mini HW-247A**: Este módulo se conecta a dos pines digitales del Arduino mediante resistencias de 1kΩ, que sirven para limitar la corriente que pasa a través de los pines y evitar daños al módulo. El **DF Player Mini** se conecta también al **TRRS Audio Jack**, permitiendo que el sonido se transmita al altavoz o audífono conectado al jack.
3. **Rotary Encoder**: Se conecta a los pines digitales del Arduino. Los pines de señal **Signal A** y **Signal B** del encoder se conectan a dos pines digitales separados en el Arduino para detectar el movimiento rotatorio. Además, se conectan los pines de **VCC** y **GND** para alimentar y dar referencia de tierra al encoder.

Al presionar cualquiera de los **Sensor Touch**, el Arduino lee la señal y activa la reproducción de la nota musical asociada a dicho sensor a través del **DF Player Mini**. El **Rotary Encoder** permite cambiar entre diferentes frecuencias o escalas, lo que permite al usuario modificar el tono de las notas según su preferencia.

**PROTOTIPOS Y MEJORAS FUTURA**

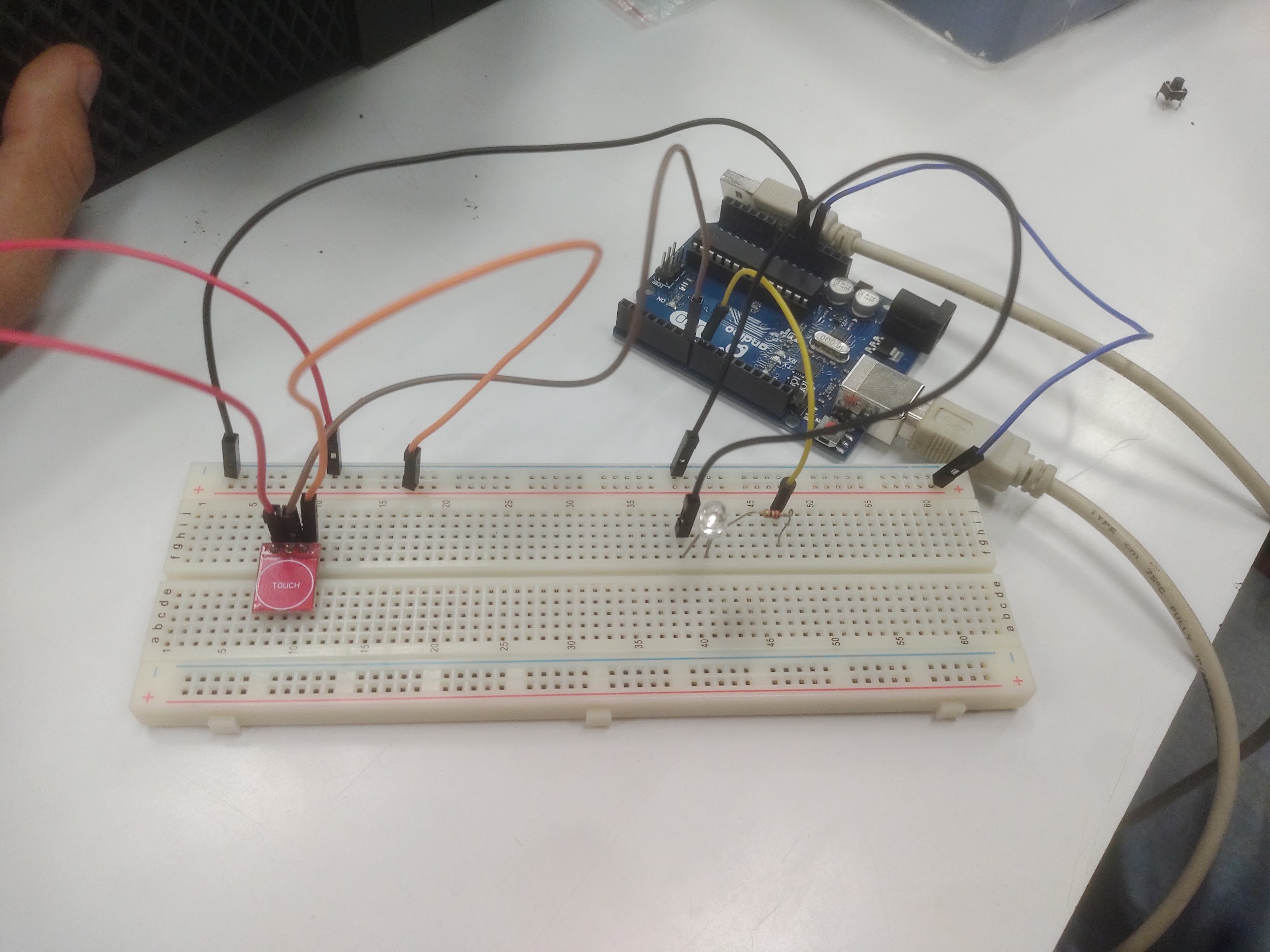
Antes de empezar siquiera con el proyecto teníamos pensado un modelo “Beta” de como podría ser el proyecto con sus respectivas conexiones.



Al comenzar con el proyecto nuestro profesor nos dijo que tendríamos que realizar el proyecto con otros componentes con los que no estábamos familiarizados, por lo que tuvimos la nueva tarea de buscar información de los componentes como los códigos, sus respectivas conexiones y funcionamientos de sus usos de cada componente.

Primeras conexiones del proyecto para el mini piano:

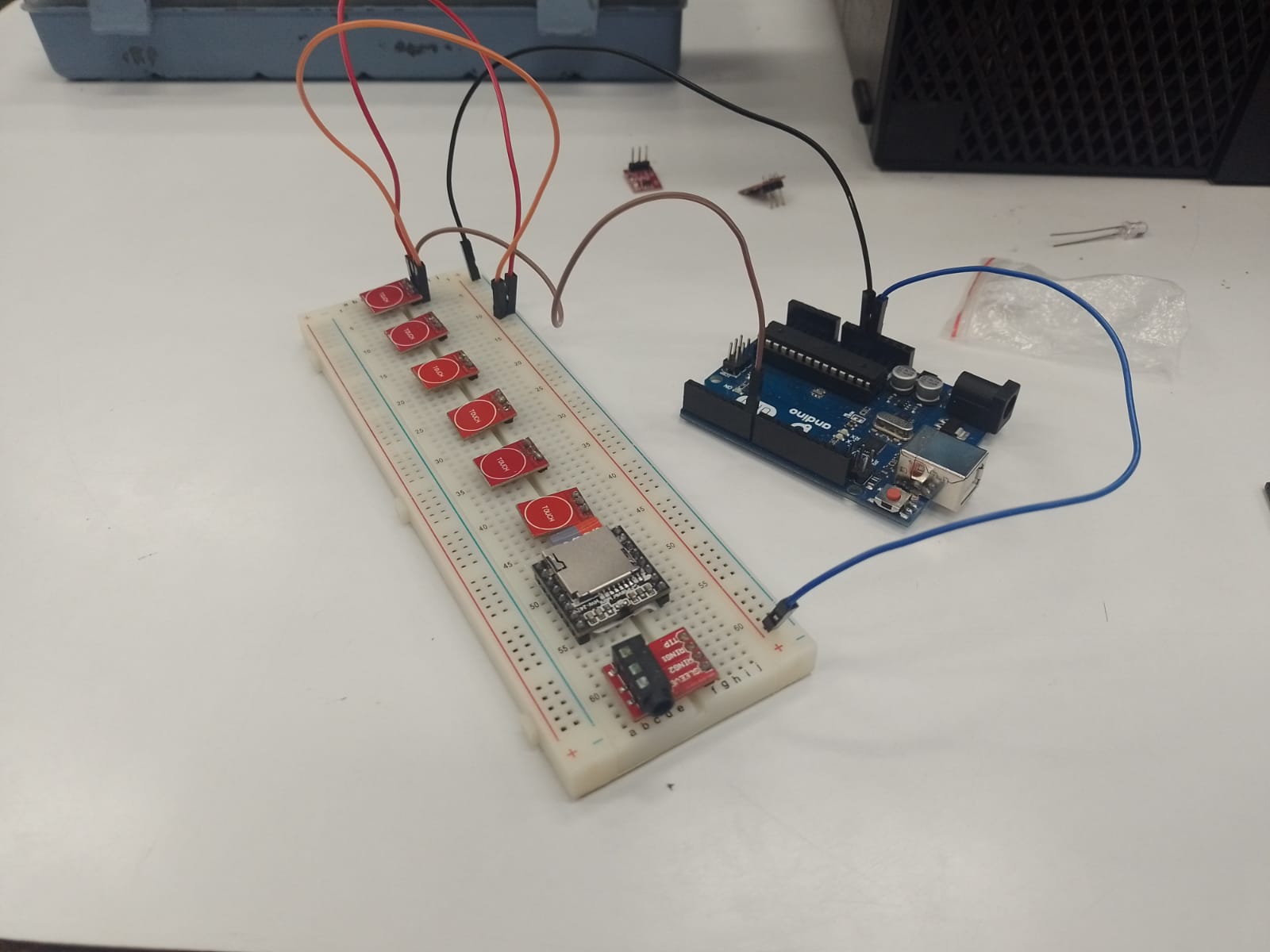
En esta imagen se muestran algunas conexiones simples las cuales hicimos para familiarizarnos con el nuevo componente que nos dio el profesor, buscando en internet encontramos una página la cual tenia un codigo, conexiones y informacion de los Sensor Touch el cual consiste en usar un Touch para prender un led.



Al momento de probarlo no nos funcionó debido a que el código era de un componente muy parecido al que estábamos usando y a las conexiones que conectamos mal al no ser el mismo por lo que tuvimos que modificarlo un poco.

En esta otra imagen fuimos avanzando con las conexiones ya que estamos más informados de cómo usar los componentes.

Fuimos poniendo todos los componentes como el DF Player Mini HW-247A, TRRS Audio Jack y pusimos el resto de los Sensor Touch.



**PRIMER CÓDIGO DE PRUEBA.**

Este fue el primer código que utilizamos para guiarnos de como usar los Touchs.

| #define led 8 #define sensorT 7  void setup() { pinMode (led, OUTPUT); pinMode (sensorT, INPUT); } void loop() { if (sensorT == HIGH) { digitalWrite(led, HIGH); } else{ digitalWrite(led, LOW); } } |
| --- |

**CÓDIGO PRELIMINAR AL RESULTADO DESEADO.**

DESCRIPCIÓN GENERAL DEL CÓDIGO:

Este código estaba diseñado para controlar un mini piano utilizando un módulo DFPlayer Mini, sensores táctiles y un Arduino. A continuación, lo explicaré en detalle, línea por línea y por bloques:

**Librerías y Configuración Inicial**

| #include "DFRobotDFPlayerMini.h" #include <SoftwareSerial.h> |
| --- |

DFRobotDFPlayerMini.h: Esta librería facilita el control del módulo DFPlayer Mini, un reproductor de MP3 de bajo costo que permite reproducir audio desde una tarjeta microSD.

SoftwareSerial.h: Permite crear puertos seriales adicionales en Arduino, ya que muchos modelos tienen un solo puerto hardware UART.

**Definición de Pines y Objetos**

| #define RX 10 #define TX 11 SoftwareSerial mySerial(RX, TX); DFRobotDFPlayerMini myDFPlayer; |
| --- |

#define RX 10 y #define TX 11: Define los pines digitales 10 y 11 como recepción (RX) y transmisión (TX) respectivamente, que se conectarán al DFPlayer Mini.

SoftwareSerial mySerial(RX, TX): Crea un puerto serial virtual para comunicarse con el módulo DFPlayer Mini.

DFRobotDFPlayerMini myDFPlayer: Instancia un objeto para interactuar con el reproductor MP3.

**Configuración de Pines para Sensores Táctiles**

| #define TOUCH\_PIN\_2 3 #define TOUCH\_PIN\_4 5 #define TOUCH\_PIN\_3 4 #define TOUCH\_PIN\_5 6 #define TOUCH\_PIN\_6 7 #define TOUCH\_PIN\_7 8 |
| --- |

#define: Define constantes para los pines de entrada asociados a los 7 sensores táctiles.

NUM\_KEYS 7: Indica el número total de teclas (o sensores táctiles) conectados.

Cada sensor táctil está asociado a un pin digital específico (pines 2 al 8).

| int touchPins[NUM\_KEYS] = {TOUCH\_PIN\_1, TOUCH\_PIN\_2, TOUCH\_PIN\_3, TOUCH\_PIN\_4, TOUCH\_PIN\_5, TOUCH\_PIN\_6, TOUCH\_PIN\_7}; |
| --- |

Se declara un array touchPins que almacena los números de los pines asignados a los sensores táctiles, facilitando su manejo en el código.

**setup() - Configuración Inicial**

| void setup() {  Serial.begin(9600);  mySerial.begin(9600); |
| --- |

Serial.begin(9600): Inicia la comunicación serial para depuración con la computadora.

mySerial.begin(9600): Configura el puerto serial virtual para comunicarse con el DFPlayer Mini.

| if (!myDFPlayer.begin(mySerial)) {  Serial.println("Error al iniciar DFPlayer.");  while (true); } |
| --- |

myDFPlayer.begin(mySerial): Inicializa el módulo DFPlayer Mini.

Si la inicialización falla, se imprime un mensaje de error y el programa queda en un bucle infinito (while (true);).

| for (int i = 0; i < NUM\_KEYS; i++) {  pinMode(touchPins[i], INPUT); } |
| --- |

Configura todos los pines asociados a los sensores táctiles como entradas (INPUT).

**loop() - Lógica Principal**

| void loop() {  for (int i = 0; i < NUM\_KEYS; i++) {  if (digitalRead(touchPins[i]) == HIGH) {  myDFPlayer.play(i + 1);  delay(200);  }  } } |
| --- |

**1. Bucle para Sensores Táctiles:**

El código recorre el array touchPins, leyendo el estado de cada pin asociado a los sensores táctiles.

digitalRead(touchPins[i]): Devuelve el estado del pin (HIGH o LOW). Si un sensor táctil detecta un toque, el pin correspondiente estará en estado HIGH.

**2. Reproducción de Archivos de Audio:**

myDFPlayer.play(i + 1): Reproduce el archivo de audio correspondiente al número de sensor.

Ejemplo: Si se activa el sensor táctil conectado a TOUCH\_PIN\_1 (índice 0 en el array), el archivo reproducido será track001.mp3. El DFPlayer numera los archivos en su tarjeta SD comenzando desde 1.

**3. Pausa para Evitar Lecturas Repetidas:**

delay(200): Introduce una pausa de 200 ms para evitar que el mismo toque se detecte varias veces consecutivas.

**Explicación del Flujo Completo**

**1. Inicialización:**

El Arduino configura la comunicación con el DFPlayer y prepara los pines de los sensores táctiles para leer señales digitales.

El DFPlayer se inicializa y el volumen se establece en un nivel moderado (15).

**2. Ejecución Continua:**

En cada iteración del loop(), el programa verifica si alguno de los sensores táctiles ha sido activado.

Si un sensor táctil detecta un toque, se reproduce un archivo MP3 asociado al número de tecla presionada.

Tras reproducir, se introduce un breve retraso para evitar activaciones repetidas.

### 

### INCONVENIENTES FUERA DEL CÓDIGO

Aunque el código estaba bien y debería de haber funcionado hubo algunos inconvenientes debido a algunas características fuera del código.

Al haber buscado y descargado las pistas de audio al guardarlo en la tarjeta microSD nos equivocamos el DF Player Mini HW-247A utiliza nombres específicos para las pistas de audio lo más común es poner tres dígitos en el nombre (001,002,etc) debido a las limitaciones y el diseño de su firmware. Estas restricciones hacen que sea más fácil para el módulo acceder rápidamente a los archivos en la tarjeta microSD sin requerir un sistema de archivo completo como los que usan los sistemas operativos modernos pero al poner el nombre a los audios por una confusión lo pusimos con cuatro dígitos esto fue un problema, esta habernos dado cuenta hicimos varios códigos pensados para usar tres dígitos por ende tuvimos que modificar el código para que pueda usar pistas de audio con esas características.

### PROGRAMACIÓN “FINAL” DEL MINI PIANO

El mini piano está diseñado para combinar sensores táctiles, un **DFPlayer Mini**, y un **rotary encoder**, permitiendo una experiencia musical dinámica. Este sistema funciona detectando la interacción del usuario a través de sensores táctiles y ajustando las pistas de audio reproducidas según un desplazamiento (offset) configurable con el encoder.

### Funcionalidad del Código

1. **Detección de Sensores Táctiles** Los sensores táctiles están conectados a pines digitales del Arduino y permiten detectar el contacto del usuario. Cada sensor está vinculado a una nota o pista de audio específica en la tarjeta microSD.
   * Cuando un sensor táctil es activado, se reproduce la pista correspondiente.
   * Los sensores son monitoreados continuamente para registrar cambios en su estado.
2. **Reproducción de Notas con el DFPlayer Mini** El **DFPlayer Mini** se encarga de reproducir las pistas de audio almacenadas en una tarjeta microSD. Cada pista corresponde a una nota del mini piano.
   * Los archivos de audio deben estar numerados de forma consecutiva (0001.mp3, 0002.mp3, etc.) para facilitar la selección.
   * La función play() del DFPlayer Mini reproduce la pista asignada al sensor activado.
3. **Ajuste de Notas con el Rotary Encoder** El rotary encoder permite ajustar un **offset** que desplaza las pistas asignadas a cada sensor táctil.
   * Girar el encoder hacia un lado incrementa el offset, mientras que girarlo hacia el otro lo decrementa.
   * Esto permite cambiar el rango de notas disponibles para los sensores, simulando un teclado con varias octavas.

### Componentes del Código

#### **Configuración Inicial**

* Se configuran los pines para los sensores táctiles y el rotary encoder.
* Se inicializa el DFPlayer Mini para gestionar la reproducción de audio.

#### **Gestión de Sensores Táctiles**

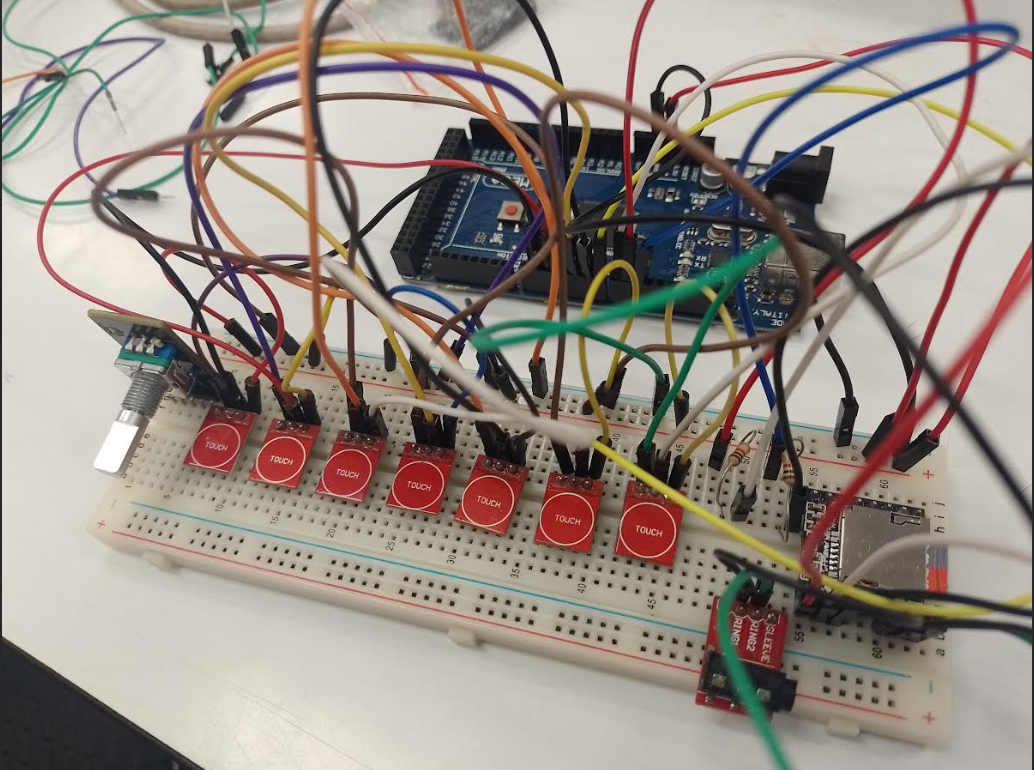
* Se detecta el estado de cada sensor táctil (presionado o no).
* Si un sensor es activado, se reproduce la pista correspondiente ajustada por el offset.

#### **Control del Rotary Encoder**

* Se monitorea el giro del encoder para ajustar dinámicamente el offset.
* El offset está limitado dentro de un rango para evitar desbordamientos.

### Requisitos de Hardware

* **DFPlayer Mini:** Reproduce las pistas de audio desde una tarjeta microSD.
* **Sensores táctiles:** Detectan la interacción del usuario, simulando las teclas del piano.
* **Rotary encoder:** Ajusta el rango de notas disponibles.
* **Arduino:** Coordina las interacciones entre los componentes y ejecuta el programa.



**Las conexiones al arduino quedaron así:**

Touch 1 al pin 2

Touch 2 al pin 3

Touch 3 al pin 4

Touch 4 al pin 5

Touch 5 al pin 6

Touch 6 al pin 7

Touch 7 al pin 8

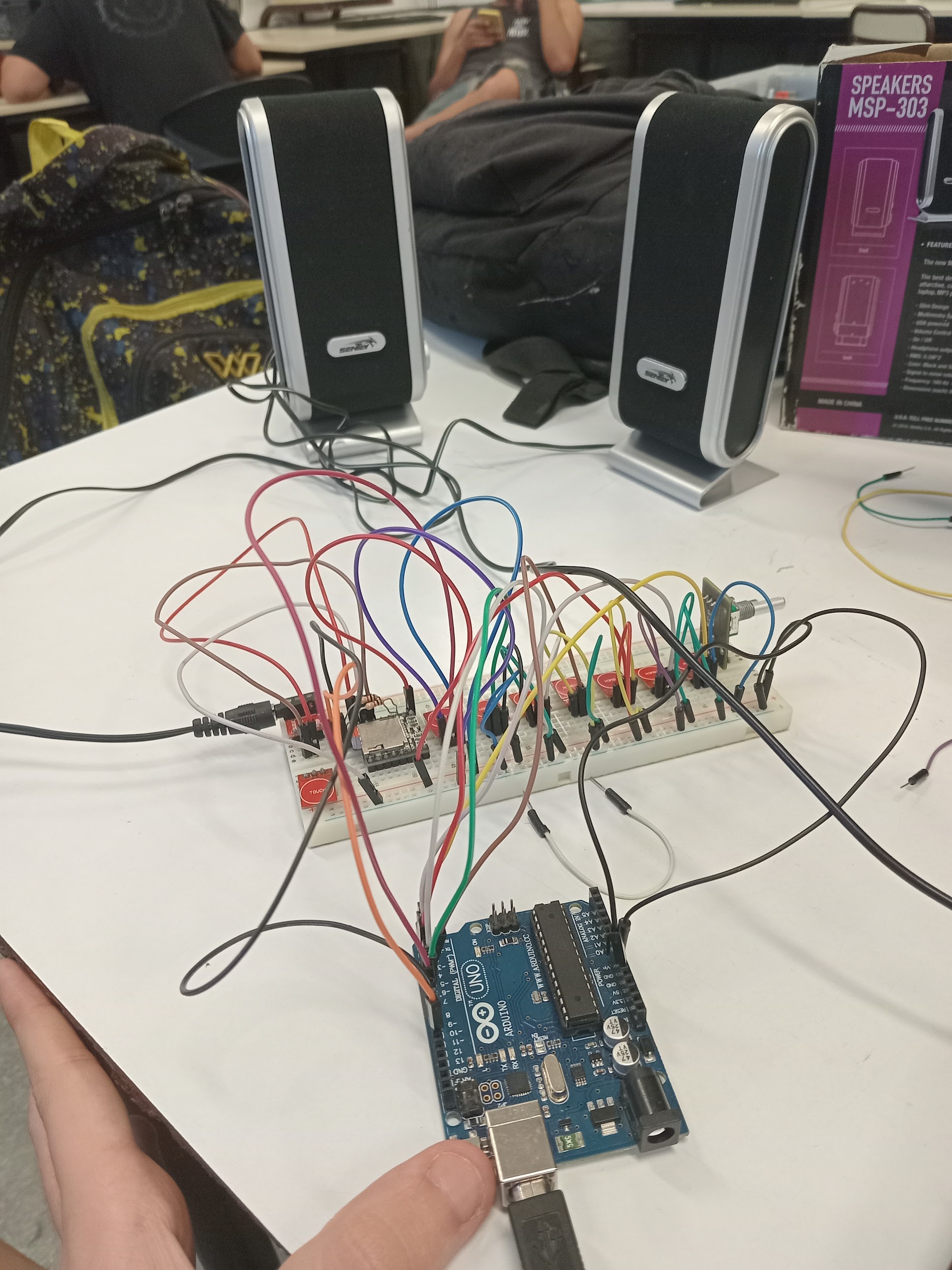
RX del display al pin 9 con una resistencia

TX del display al pin 10 con una resistencia

SW del rotary al pin 11

DT del rotary al pin 12

CLK del rotary al pin 13

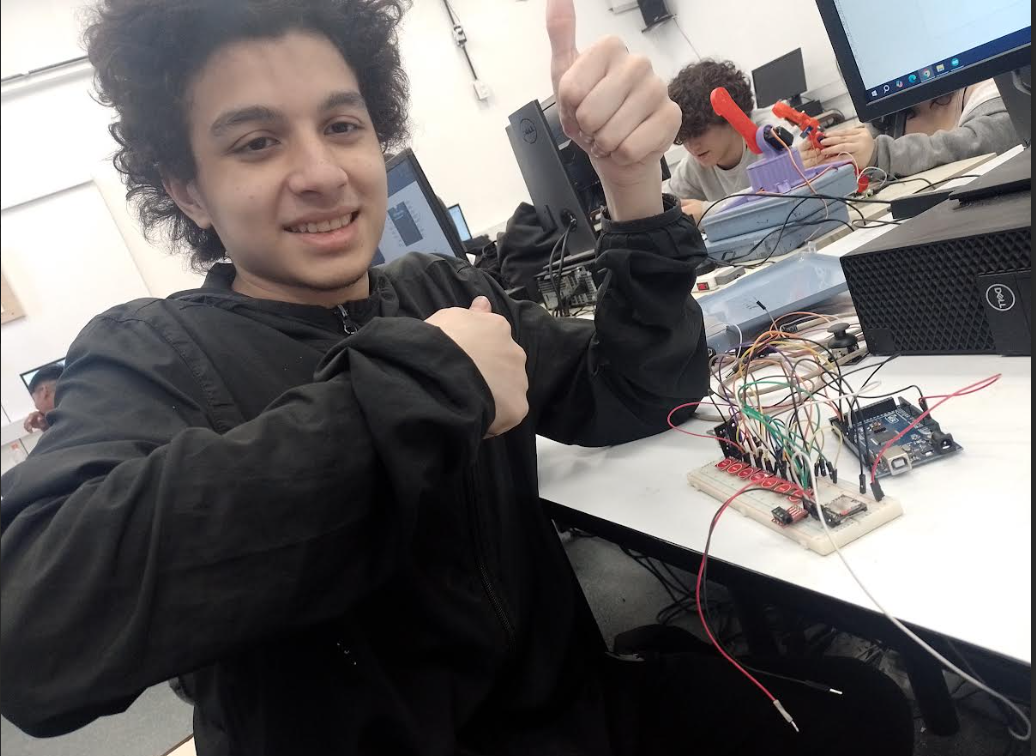


**PRUEBAS Y RESULTADOS**

A lo largo del desarrollo del proyecto, se realizaron varias pruebas para asegurar que cada componente funcionará como se esperaba. Las pruebas clave incluyen:

1. **Prueba de notas**: Cada uno de los **Sensor Touch** fue presionado para verificar que el **DF Player Mini** reprodujera el tono correcto. El resultado fue satisfactorio, ya que todas las notas fueron emitidas con precisión.
2. **Prueba de respuesta**: Se verificó que el **DF Player Mini** reaccionara de manera rápida y precisa a los toques de los sensores, sin retrasos perceptibles entre la acción del usuario y la reproducción del sonido.

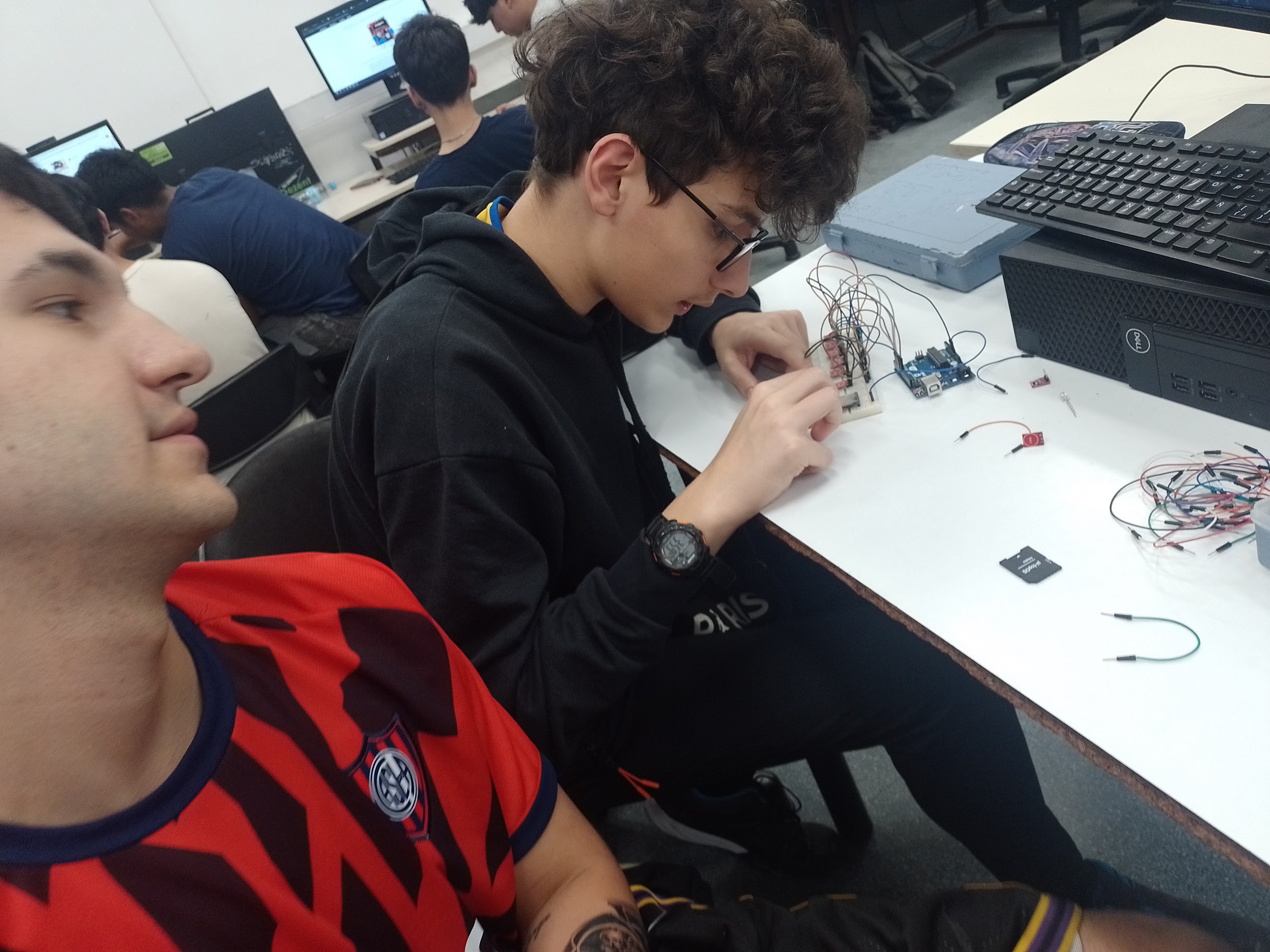
Ambas pruebas confirmaron que el sistema estaba funcionando correctamente y de manera eficiente.



**FALLOS Y CORRECCIONES**

A lo largo del proceso, se presentaron varios problemas, los cuales fueron solucionados de la siguiente manera:

1. **Problema : Sensor Touch no responde  
   Causa**: Las conexiones en el Sensor Touch estaban incorrectamente puestas.  
   **Solución**: Se revisó y ajustó la conexión de los sensores táctiles al Arduino, asegurando que todos los cables estuvieran correctamente conectados a los pines correctos.
2. **Problema : El DF Player Mini no tenía todas las notas necesarias  
   Causa**: Al cargar las notas en el DF Player Mini, algunas frecuencias se habían omitido.  
   **Solución**: Se subieron todas las notas musicales necesarias y se verificó que las frecuencias correctas estuvieran presentes en el módulo.
3. **Problema : Arduino incorrecto  
   Causa**: Se confundieron al seleccionar el Arduino y usaron uno que estaba suelto. Al intentar probar el código y pulsar los sensores táctiles, no se emitía ningún sonido. El problema no era el código, como inicialmente se pensó, sino el Arduino utilizado.  
   **Solución**: Se cambiaron los cables a otro Arduino y se probó nuevamente el código, pero aún no se emitía sonido. Se detectó que un sensor Touch no funcionaba correctamente debido a interferencias provenientes de un problema específico de otro proyecto del grupo 1. Se solucionó al reemplazar los cables defectuosos y verificar que el Touch funcionara correctamente.
4. **Problema : Interferencia en la señal de RX y TX  
   Causa**: Las resistencias de RX y TX estaban tocándose, causando interferencia que impedía la reproducción de sonido.  
   **Solución**: Se modificaron las conexiones de las resistencias y se separaron correctamente, eliminando la interferencia y permitiendo que el piano emitiera sonido.
5. **Problema : Problema de doble toque y reinicio  
   Causa**: Al presionar dos sensores táctiles al mismo tiempo, el sistema dejaba de responder y requería un reinicio del Arduino.  
   **Solución**: Se realizaron ajustes en el código y en la lógica de manejo de entradas para permitir que el sistema pudiera manejar múltiples toques simultáneamente sin que se bloquease.
6. **Problema : Problema de las notas agudas y graves  
   Causa**: Las dos primeras notas del piano eran excesivamente agudas y la tercera cambiaba repentinamente a una frecuencia grave.  
   **Solución**: Se revisó y ajustó el mapeo de notas en el código, corregiendo las frecuencias para que la transición entre notas fuera más suave y natural.
7. **Problema : Sensores táctiles demasiado juntos  
   Causa**: Los sensores táctiles estaban muy cerca entre sí, lo que provocaba que al intentar presionar un sensor, también se tocara otro, causando que el sistema se trabara.  
   **Solución**: Se reorganizaron los sensores táctiles para aumentar el espacio entre ellos y se implementaron ajustes en el código para filtrar las entradas y evitar activaciones simultáneas accidentales.



**CONCLUSIÓN**

El proyecto del mini piano con Arduino fue una excelente oportunidad para aprender sobre la integración de componentes electrónicos y la programación. A través de la resolución de problemas y la implementación de diversas soluciones técnicas, logramos construir un sistema interactivo funcional que genera notas musicales y permite variar la escala mediante el **Rotary Encoder**.

Este proyecto no solo ayudó a mejorar nuestra comprensión sobre cómo funciona el hardware y el software en conjunto, sino que también nos proporcionó una valiosa experiencia práctica en la creación de dispositivos interactivos. Los problemas surgidos durante el desarrollo fueron resueltos eficazmente, lo que nos permitió afianzar nuestros conocimientos sobre la electrónica y la programación de microcontroladores**.**

**REFERENCIAS**

Documentación oficial de Arduino: <https://www.arduino.cc>

Video Tutorial:

Como subir musica a un microsd en pc:

<https://www.youtube.com/watch?v=0BCpmH0ljeU>

Recursos de electrónica:

Para probar, buscamos un código con un touch y un led en esta página:

<https://techmake.com/blogs/tutoriales/empezando-con-arduino-2d-sensor-touch>

Para pasar audios mp3 de piano al DFplayer <https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299>

<https://www.tecneu.com/blogs/tutoriales-de-electronica/como-utilizar-el-dfplayer-mini-con-arduino-para-proyectos-de-audio>

<https://www.tecneu.com/blogs/tutoriales-de-electronica/como-utilizar-el-dfplayer-mini-con-arduino-para-proyectos-de-audio?srsltid=AfmBOoqOrAz9NmSPFa52e4i9wPOP93qdTdyRNu2RKlx-0xu2-0qHVTQs>

Esto es la información del Rotary Encoders

<https://arduinogetstarted.com/tutorials/arduino-rotary-encoder>

bibliotecas arduino

<https://www.youtube.com/watch?v=CK1THPvw77M>

<https://www.youtube.com/watch?v=Jum_6mMCGLA&t=632s>

<https://www.youtube.com/watch?v=pIFoOQJEFZ0>