



FACULTAD  
DE INGENIERÍA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# **Dispositivo inalámbrico para ejecución y simulación de instrumentos MIDI (DIPESIM)**

Cuello Marina Giselle 02842/2

Forden Jones Ian 02543/3

Giacoa Pedro 02547/7

Amadori Franco 02545/5

Chavez Sánchez Máximo Nicolás 03076/1

**Departamento de Electrotecnia**

**Taller de Proyecto I**

**2 de octubre de 2025**

## 1. Introducción

La música digital ha avanzado de forma notable gracias al uso de microcontroladores y periféricos capaces de interpretar y reproducir señales en formato MIDI. Este proyecto propone el desarrollo de un sistema portátil que emule una guitarra mediante el uso de dispositivos interactivos, permitiendo al usuario realizar acordes y rasgueos en el aire para generar sonidos musicales reales.

El sistema integra sensores distribuidos en dos dispositivos: uno con pulsadores que simulan la presión sobre las cuerdas de una guitarra y otro con un sensor de movimiento que detecta el rasgueo. El objetivo es brindar una experiencia de interpretación musical sin necesidad de un instrumento físico, empleando tecnologías actuales de comunicación inalámbrica, microcontroladores y procesamiento digital de señales.

En el ámbito de la música digital existen diversos desarrollos orientados a la generación y control de sonidos, tales como sintetizadores, teclados electrónicos, controladores MIDI y guitarras digitales. Sin embargo, la mayoría de estas soluciones se basan en hardware especializado y de alto costo. La propuesta se destaca por su bajo costo, la utilización de hardware de libre acceso (incluyendo la EDU-CIAA) y la posibilidad de diseñar una PCB personalizada para la interconexión de los dispositivos.

El desafío principal del proyecto radica en la integración de los sensores con el sistema de procesamiento, en el diseño de la lógica para la correcta interpretación de los datos obtenidos y en la implementación de mecanismos que permitan filtrar movimientos no válidos, garantizando así una traducción precisa de las acciones físicas en comandos MIDI.

## 2. Objetivos del proyecto

### Objetivos primarios:

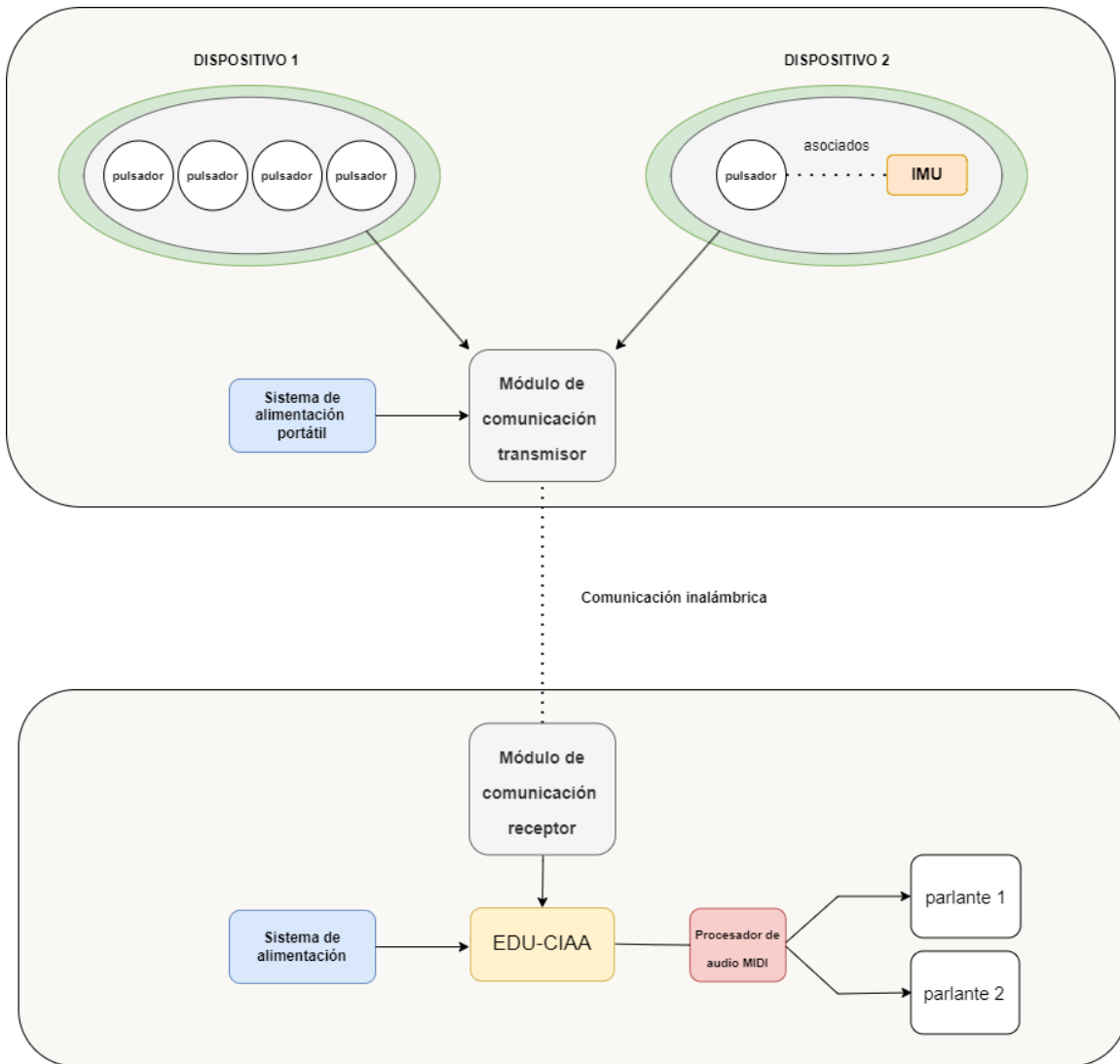
- Desarrollar un dispositivo capaz de emular acordes y otro capaz de emular rasgueos de guitarra.
- Transmitir los datos captados por los pulsadores y por la IMU a la EDU-CIAA-NXP para su procesamiento.

- Implementar un filtrado que descarte eventos inválidos (rasgueos débiles o sin pulsadores activos).
- Enviar los eventos válidos en formato MIDI hacia un periférico de reproducción de sonido conectado a un parlante.
- Diseñar e implementar una PCB que sirva de interfaz y soporte para los módulos del sistema.

### **Objetivos secundarios:**

- Optimizar la detección de rasgueos mediante algoritmos de análisis de aceleración angular.
- Implementar un sistema de grabación/reproducción de secuencias MIDI.
- Implementar y desarrollar la simulación de otros instrumentos.

A continuación, se presenta el diagrama en bloques del sistema estimado.



**Figura 1** - Diagrama en bloques del sistema a desarrollar

En la **Figura 1** se visualizan los distintos dispositivos que conforman el sistema completo, el cual se encuentra dividido en dos partes que, en conjunto, constituyen una única unidad funcional y se comunican entre sí mediante un enlace inalámbrico.

### **3. Análisis de requerimientos**

#### **Requerimientos de hardware**

1. El sistema debe contar con una unidad de procesamiento central (EDU-CIAA) capaz de recibir, interpretar y filtrar la información proveniente de los sensores externos.
2. Debe disponer de módulos de adquisición que permitan registrar la presión de pulsadores y los movimientos de un sensor inercial, garantizando la detección de acordes y rasgueos.
3. Debe incluir un mecanismo de comunicación inalámbrica que permita el envío confiable y de baja latencia de los datos entre los dispositivos portátiles y la unidad de procesamiento principal.
4. El hardware debe ser capaz de generar audio en formato MIDI y reproducirlo a través de un sistema de salida sonora externo.
5. El sistema debe contar con una fuente de alimentación portátil que asegure un funcionamiento autónomo durante sesiones continuas de uso.
6. El diseño físico debe contemplar una carcasa o estructura ergonómica que permita al usuario manipular el dispositivo de manera cómoda y segura.

#### **Requerimientos de software**

1. El software de los dispositivos portátiles debe encargarse de la captura de señales provenientes de pulsadores y del sensor de movimiento, y transmitirlos de forma eficiente hacia la unidad de procesamiento.
2. El software de la EDU-CIAA debe recibir los datos, procesarlos y aplicar un filtrado que descarte movimientos o eventos no válidos.
3. El sistema debe implementar un protocolo de comunicación que garantice la transmisión de datos en tiempo real con baja latencia.
4. Debe existir una capa de traducción de los eventos en mensajes MIDI estandarizados para su posterior reproducción sonora.
5. El software debe incluir algoritmos que permitan reconocer patrones válidos de acordes y rasgueos, diferenciando acciones intencionales de ruidos o falsos positivos.

## **Requerimientos no funcionales**

1. El tiempo de respuesta entre la acción del usuario y la generación del sonido no debe superar los 30 ms.
2. El sistema debe contar con una autonomía mínima de una hora de funcionamiento continuo con alimentación portátil.
3. El diseño general debe ser modular, de modo que cada componente (captura, transmisión, procesamiento, salida de audio) pueda modificarse o ampliarse sin afectar al resto.
4. El desarrollo debe utilizar como unidad de procesamiento central la EDU-CIAA.
5. El proyecto debe cumplir un cronograma de trabajo que asegure la finalización de todas las etapas para el mes de diciembre.
6. La documentación final debe describir de manera completa tanto el diseño de hardware como el de software, asegurando su posible reproducción y mejora futura.

## **4. Diseño de hardware**

El sistema propuesto se organiza en dos bloques principales: unidad portátil y unidad fija. Cada bloque integra los componentes necesarios para cubrir las funciones de captura de entradas del usuario, procesamiento de la información y generación de la salida de audio.

### **Bloque portátil**

La unidad portátil es la encargada de la adquisición de datos y la transmisión inalámbrica hacia la unidad fija. Sus principales componentes son:

- Unidad de control: ESP32-WROOM-32
- Dispositivo 1: matriz de pulsadores donde se seleccionan los acordes
- Dispositivo 2: sensor inercial MPU-6050 (acelerómetro + giroscopio) el cual detecta el rasgo en conjunto con un pulsador único que prende o apaga la detección.
- Comunicación: transmisión inalámbrica con ESP32-NOW hacia el bloque fijo.
- Alimentación: batería portátil 5V / 10000 mA

Considerando los consumos de los principales componentes (ESP32  $\approx 500$  mA y MPU-6050  $\approx 5$  mA), se obtiene un consumo total aproximado de 505 mA. En estas condiciones, la autonomía teórica supera las 20 horas de funcionamiento continuo, lo cual resulta adecuado para el uso previsto.

### **Bloque fijo**

La unidad fija es la encargada del procesamiento de los datos (podríamos poner algo mas). Sus principales componentes son:

- ESP32-WROOM-32(receptor): recibe los datos del bloque portátil vía ESP-NOW y los reenvía a la EDU-CIAA a través de una interfaz UART.
- EDU-CIAA-NXP: funciona como unidad de procesamiento central. Ejecuta las tareas de filtrado de eventos inválidos y la traducción de datos en mensajes MIDI. Posteriormente, transmite los comandos hacia el sintetizador de audio mediante interfaz SPI.
- Sintetizador de audio (VS1053B): recibe los comandos MIDI por SPI y genera la señal de audio correspondiente. Su salida se conecta a un parlante autoamplificado para la reproducción final.

Este bloque se alimenta con una fuente interna de 5V, dimensionada para abastecer al sintetizador y los periféricos asociados. El consumo estimado de la unidad fija se aproxima a 300mA.

### **Interconexiones eléctricas**

Las interfaces entre los distintos bloques y dispositivos se definen de la siguiente manera:

- Pulsadores  $\rightarrow$  ESP32 portátil: conexión directa a GPIOs configurados como entradas digitales con resistencias de pull-up internas.
- MPU-6050  $\rightarrow$  ESP32 portátil: interfaz I<sup>2</sup>C para la transmisión de datos inerciales.
- ESP32 portátil  $\leftrightarrow$  ESP32 fijo: comunicación inalámbrica mediante ESP-NOW (peer-to-peer).
- ESP32 fijo  $\leftrightarrow$  EDU-CIAA: comunicación serie a través de UART.
- EDU-CIAA  $\leftrightarrow$  VS1053B: comunicación síncrona mediante SPI (líneas MOSI, MISO, SCK).
- VS1053B  $\leftrightarrow$  parlante: salida analógica estéreo conectada a un módulo de audio autoamplificado.

El circuito esquemático completo, elaborado en Proteus, se incluye en los anexos junto con la lista de materiales correspondiente.

## **5. Diseño del firmware, simulación y depuración**

El diseño del firmware se plantea con un enfoque modular y orientado a eventos, de manera que cada componente del sistema actúe en función de las señales que recibe, ya sea desde sensores, pulsadores o interfaces de comunicación.

### **Bloque portátil - ESP32 con pulsadores + IMU**

- Lectura de estados: GPIO (pulsadores), I2C (IMU)
- Librerías: Adafruit MPU6050 para facilitar la gestión del acelerómetro y giroscopio.
- Servicios: filtrado de rebotes en pulsadores, detección de picos de aceleración para rasgeos.
- Aplicación: empaquetado de acordes + eventos de rasqueo → transmisión ESP-NOW.

### **Bloque fijo - ESP32 receptor**

- Drivers y bibliotecas: UART (hacia EDU-CIAA) y ESP-NOW (recepción inalámbrica)
- Aplicación: puente de comunicación que recibe la información del bloque portátil y la retransmite por UART hacia la EDU-CIAA.

### **Bloque fijo - EDU-CIAA**

- Drivers y bibliotecas:
  - UART para la recepción de datos desde el ESP32
  - SPI para la transmisión de comandos al sintetizador de audio
- Servicios
  - Decodificación de acordes recibidos
  - Filtrado de rasgueos inválidos para evitar activaciones erróneas
- Aplicación: conversión de eventos a mensajes MIDI estándar, posteriormente enviados al sintetizador.



## **Bloque fijo – Sintetizador VS1053B**

- Interfaz: configuración y control mediante protocolo SPI desde la EDU-CIAA
- Función: reproducción de audio a partir de los mensajes MIDI recibidos, con salida a un parlante autoamplificado.

## **Interfaz de usuario**

La interfaz de usuario se define en dos entradas principales:

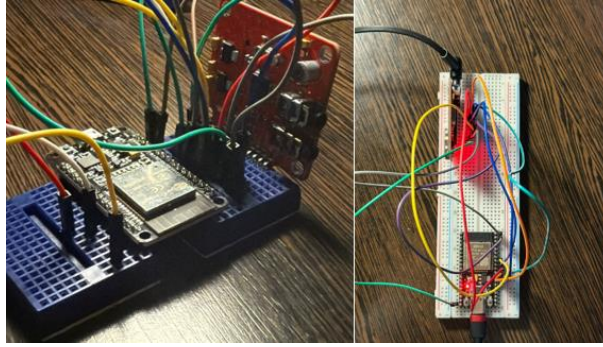
- **Selección de acordes:** mediante la matriz de pulsadores del bloque portátil.
- **Detección de rasgueos:** a partir del análisis de aceleraciones medidas por el sensor inercial.

La salida del sistema se materializa como una respuesta inmediata en forma de audio, lo que asegura una interacción natural y en tiempo real entre el usuario y el instrumento electrónico.

## **Ensayos preliminares**

En primera instancia se realizaron pruebas básicas de los módulos adquiridos para el proyecto con el objetivo de verificar su correcto funcionamiento y descartar fallas de fabricación.

En el caso del sintetizador de audio (VS1053B), se efectuó un conexionado inicial para comprobar su respuesta. El módulo encendió y recibió alimentación de manera adecuada. Sin embargo, durante esta etapa el microcontrolador ESP32 presentó fallas de software desconocidas que impidieron completar la primera serie de ensayos. Posteriormente se reemplazó el MCU y se pudo continuar con las pruebas.



**Figura 2.** A la izquierda, la prueba con el primer ESP32 adquirido. En la derecha, la prueba con el segundo ESP32.

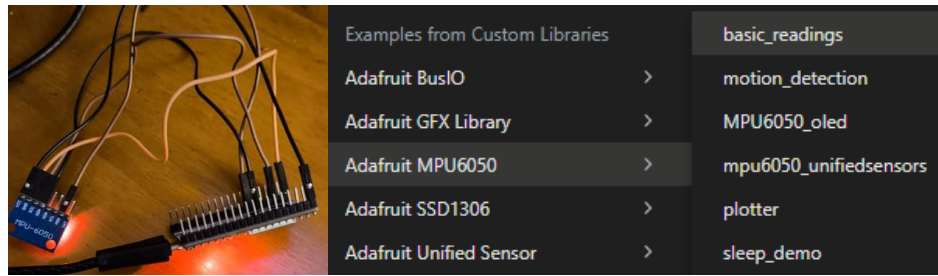
Las verificaciones realizadas incluyeron tanto el modo de reproducción de tonos de prueba (sine test<sup>1</sup>) como el modo sintetizador MIDI. La primera prueba resultó satisfactoria, ya que el sine test confirmó que el chip se encontraba correctamente alimentado, que la inicialización por SPI fue exitosa y que la etapa de salida de audio funcionaba sin interferencias. En la segunda prueba, se cargó el plugin MIDI en la memoria interna del VS1053B y se enviaron comandos de cambio de programa y notas. No obstante, en lugar de reproducir sonidos instrumentales (por ejemplo, piano acústico), únicamente se percibieron ruidos breves, pulsos aislados y ligeras interferencias. Esto indica que el chip recibe y procesa datos, pero aún no se encuentra configurado de manera estable para operar como sintetizador MIDI.

En conclusión, los ensayos permitieron comprobar el correcto funcionamiento del canal de audio y la comunicación SPI mediante el sine test, mientras que el modo MIDI requerirá ajustes adicionales para alcanzar una reproducción musical consistente.

Por otro lado, en el caso del acelerómetro (MPU6050), se llevó a cabo un ensayo inicial utilizando el programa de ejemplo Basic Readings de Arduino IDE. El dispositivo respondió de forma adecuada, registrando lecturas en los tres ejes, lo que permitió confirmar tanto el correcto funcionamiento de la interfaz I2C como la operatividad del sensor.

---

<sup>1</sup> Prueba del VS1053B que emite un tono senoidal de referencia para comprobar el funcionamiento del módulo de audio.



**Figura 3.** A la izquierda, el conexionado ESP32-MPU6050. A la derecha, la librería que contiene el programa de prueba.

```

21:29:13.570 -> Aceleración X:-2.32 m/s^2, Y:3.71 m/s^2, Z:-9.57 m/s^2
21:29:13.637 -> Aceleración X:-2.32 m/s^2, Y:3.65 m/s^2, Z:-9.51 m/s^2
21:29:13.670 -> Aceleración X:-2.37 m/s^2, Y:3.75 m/s^2, Z:-9.48 m/s^2
21:29:13.744 -> Aceleración X:-2.33 m/s^2, Y:3.67 m/s^2, Z:-9.53 m/s^2
21:29:13.775 -> Aceleración X:-2.34 m/s^2, Y:3.69 m/s^2, Z:-9.53 m/s^2
21:29:13.847 -> Aceleración X:-2.30 m/s^2, Y:3.87 m/s^2, Z:-9.55 m/s^2
21:29:13.881 -> Aceleración X:-2.23 m/s^2, Y:4.03 m/s^2, Z:-9.52 m/s^2
21:29:13.947 -> Aceleración X:-2.28 m/s^2, Y:3.68 m/s^2, Z:-9.56 m/s^2
21:29:13.980 -> Aceleración X:-2.29 m/s^2, Y:3.65 m/s^2, Z:-9.54 m/s^2
21:29:14.063 -> Aceleración X:-2.32 m/s^2, Y:3.66 m/s^2, Z:-9.52 m/s^2
21:29:14.097 -> Aceleración X:-2.38 m/s^2, Y:3.64 m/s^2, Z:-9.52 m/s^2
21:29:14.164 -> Aceleración X:-2.34 m/s^2, Y:3.64 m/s^2, Z:-9.52 m/s^2

```

**Figura 4.** Serial Monitor mostrando las mediciones realizadas por el MPU6050.

## 6. Cronograma preliminar

En la **Figura 2** se presenta el diagrama de Gantt, el cual detalla las actividades previstas para el desarrollo del sistema. Se proyecta que la finalización del proyecto tendrá lugar a mediados de diciembre.

Nro.	Actividades	Agosto				Septiembre				Octubre				Noviembre				Diciembre			
	Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Revisión bibliográfica																				
2	Diseño inicial y definición de requerimientos																				
3	Programación de los módulos de comunicación para lectura de pulsadores e IMU																				
4	Diseño de la pcb y pruebas de módulo vs1053																				
5	Fabricación pcb																				
6	Ensamble																				
7	Ensayos																				
8	Muestra - Informe final																				

**Figura 2** - Diagrama de Gantt

## 7. División de tareas del grupo

Integrante	Responsabilidad principal	Horas invertidas
Pedro Giacoia	Diseño y fabricación de módulo portátil (guantes, armazón, ubicación de módulos wi-fi)	12
Ian Enrique Walter Forden Jones	Programación EDU-CIAA-NXP (recepción, filtrado de eventos)	12
Máximo Nicolas Chavez Sanchez	Manejo de transferencia MIDI	12
Franco Amadori	Diseño y fabricación de PCB, montaje de hardware	12
Marina Giselle Cuello	Documentación, pruebas del sistema y validación final	12

## 8. Bibliografía

[1] Proyecto CIAA. Portal web:

<http://proyecto-ciaa.com.ar/devwiki/doku.php?id=start>

[2] Official MIDI specifications and resources. The MIDI Association.

[MIDI.org](http://MIDI.org)

[3] MIDIProjects. Instructables.

<https://www.instructables.com/search/?q=MIDI&projects=featured>

[4] ESP-NOW Tutorials. Random Nerd Tutorials.

<https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/>

[5] VS1053 Datasheet. VLSI Solution.

<https://www.vlsi.fi/fileadmin/datasheets/vs8053.pdf>

[6] Datasheet ESP32-WROOM-32. Espressif Systems

[https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)

[7] Datasheet MPU-6050. InvenSense

<https://cdn.sparkfun.com/datasheets/Sensors/Accelerometers/RM-MPU-6000A.pdf>

[8] Datasheet AMS1117. Advanced Monolithic Systems

<http://www.advanced-monolithic.com/pdf/ds1117.pdf>

## 9. Anexos

### 9.1 Tabla de materiales (BOM)

Componente	Cantidad	Descripción	Observaciones
ESP32-WROOM-32	2	Módulo ESP32, ESP-NOW	Uno para el bloque portátil, uno para el bloque fijo
Pulsadores	5	Pulsadores simples	Acordes y encendido de detección de rasgueo
MPU-6050	1	Sensor IMU 6-DoF	Detección de rasgueo, I2C
Batería Li-Po 5V/10000 mAh	1	Power bank o pack Li-Po	Alimentación de bloque portátil
USB C breakout board	1	Placa adaptadora USB-C para facilitar la conexión de alimentación	Permite tomar 5 V directamente de un cargador o power bank sin complicaciones
EDU-CIAA-NXP	1	Placa educativa CIAA	Procesamiento central
VS1053B	1	Codec/sintetizador audio	Interfaz SPI con CIAA
Parlante	1	5 V, baja potencia	Salida de audio
Fuente 5 V / 1 A	1	Alimentación bloque fijo	Para VS1053 + periféricos
Cables, conectores, headers	-	Varios	Interconexión eléctrica
PCB o perfboard	1	Placa prototipo	Montaje del sistema

## 9.2 Diagrama PCB (montaje)

