



DESARROLLO DE UN DISPOSITIVO PARA EMULAR LAS VIBRACIONES EN LA MANO DE PERSONAS CON LA ENFERMEDAD DE PARKINSON

DIEGO ANDERSSON RAMOS TORO
CÓDIGO: 1927722

ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
FACULTAD DE INGENIERÍA
UNIVERSIDAD DEL VALLE

SANTIAGO DE CALI
2025

**DESARROLLO DE UN DISPOSITIVO PARA EMULAR LAS VIBRACIONES EN LA
MANO DE PERSONAS CON LA ENFERMEDAD DE PARKINSON**

DIEGO ANDERSSON RAMOS TORO

TRABAJO PRESENTADO PARA OPTAR POR EL TÍTULO DE INGENIERO ELECTRÓNICO

**DIRECTOR
ING. JOSÉ ISIDRO GARCÍA MELO, PhD.**

Grupo de Investigación BIONOVO

**ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
FACULTAD DE INGENIERÍA
UNIVERSIDAD DEL VALLE**

**SANTIAGO DE CALI
2025**

Nota de aceptación:

Firma del director del trabajo

Firma del evaluador

Firma del evaluador

Santiago de Cali, 23 de junio del 2025.

DEDICATORIA

Quiero expresar mi más profundo agradecimiento a todas las personas que han sido pilares fundamentales en la culminación de este trabajo de grado.

A mi familia, por ser mi raíz y mi refugio. Por su apoyo incondicional, su aliento constante en los momentos de dificultad y por enseñarme, con su ejemplo, el verdadero significado del esfuerzo, la constancia y el amor.

A mis amigos cercanos, por acompañarme en este camino con su amistad sincera, sus consejos, su apoyo desinteresado y su capacidad de estar presentes, incluso en la distancia. Gracias por las risas, las conversaciones y por ayudarme a seguir adelante cuando más lo necesitaba.

A mi novia y compañera de vida, por su amor inquebrantable, su paciencia infinita y su fe en mí. Por compartir conmigo cada logro y cada desafío, y por recordarme siempre la importancia de seguir soñando juntos.

A mí mismo, por no rendirme. Por siempre estar dispuesto a mejorar como persona, por cada obstáculo superado, por creer en este proyecto incluso en los momentos más inciertos. Por demostrarme que los sueños, con trabajo y pasión, se construyen paso a paso.

Y, de manera muy especial, agradezco al profesor José Isidro García, director de este trabajo de grado, por su guía experta, su dedicación y paciencia a lo largo de todo el proceso. Sus conocimientos, su acompañamiento constante y su compromiso con mi formación fueron fundamentales para alcanzar los resultados obtenidos.

Tabla de contenido

GLOSARIO	10
RESUMEN	11
ABSTRACT	13
CAPÍTULO 1.	14
GENERALIDADES	14
1.0 INTRODUCCIÓN	14
1.1 PLANTEAMIENTO DEL PROBLEMA	15
1.2 JUSTIFICACIÓN	16
1.3 ANTECEDENTES	17
1.4 OBJETIVOS	19
1.4.1 Objetivo general	19
1.4.2 Objetivos específicos	19
1.5.1 Enfermedades neurodegenerativas	19
1.5.2 Anatomía de la mano	19
1.5.3 Enfermedad de Parkinson	19
1.5.4 Estadios de la EP	20
1.5.5 Sistemas automáticos y de control	20
1.5.6 Protocolos de prueba	20
1.6 ESTRUCTURA DEL DOCUMENTO	20
CAPÍTULO 2	22
MARCO METODOLÓGICO	22
2.1. MARCO METODOLÓGICO	23
CAPÍTULO 3	25
DISEÑO Y ARQUITECTURA DEL SISTEMA PARKIMOTION	25
3.1 DEFINICIÓN DE REQUISITOS	26
3.1.1 Requerimientos Funcionales y No Funcionales	26
3.1.2 Diagrama de casos de uso	27
3.1.2.1 ACTOR: USUARIO	28
3.2 DESCRIPCIÓN FUNCIONAL DEL SISTEMA	30
3.3 FLUJO FUNCIONAL DEL SISTEMA	31
3.4 MÓDULOS FUNCIONALES	33
3.4.1 MÓDULO DE INTERFAZ	33
3.4.2 MÓDULO DE SENSADO	34
3.4.3 MÓDULO DE PROCESAMIENTO	35
3.4.4 MÓDULO DE ACTUACIÓN	36
3.4.4.1 Componentes principales:	36
3.4.5 MÓDULO DE COMUNICACIÓN	37
3.4.5.1 Funciones principales:	37
3.4.5.2 Componentes principales:	37

3.4.5.3 Justificación técnica:	37
3.4.6 MÓDULO MECÁNICO	38
3.5 DISEÑO DEL SISTEMA PARKIMOTION	39
3.5.1 DISEÑO MECÁNICO	39
3.5.1.1 Subsistemas mecánicos y componentes principales	39
3.5.1.1.1 Subsistema de oscilación transversal	39
3.5.1.1.2 Subsistema de oscilación sagital	39
3.5.1.1.3 Estructura soporte	39
3.5.2 DISEÑO DE LA INTERFAZ	42
3.5.2.2 Componentes principales	43
3.5.2.3 Interfaces	43
3.5.3 MODELADO DE LA PLANTA Y DISEÑO DEL SISTEMA DE CONTROL	43
3.5.3.1 Modelado matemático de la planta	43
3.5.3.2 Identificación experimental del modelo	44
3.5.3.3 Planteamiento del Problema de Control	46
3.5.3.3.1 Objetivo técnico del sistema	46
3.5.3.3.2 Descripción funcional del sistema	46
3.5.3.3.3 Necesidad de control	47
3.5.3.4 Diseño del controlador	47
3.5.3.4.1 Modelo de Control en Lazo Cerrado	47
3.5.3.4.2 Modelo matemático del controlador PID	49
3.5.3.4.3 Discretización de la planta y diseño del controlador PID	49
3.5.3.4.4 Diseño del controlador PID a partir de la planta discretizada	50
3.5.4 DISEÑO DEL SISTEMA EMBEBIDO	52
3.5.4.1 Flujo funcional de tareas	54
3.5.4.2 Diseño conceptual del controlador	55
3.5.5 DISEÑO ELECTRÓNICO	55
3.5.5.2 Diseño de la PCB y distribución de componentes	57
3.5.5.3 Consideraciones sobre componentes no implementados	57
Capítulo 4	58
PROTOTIPO EXPERIMENTAL DE PARKYMOTION.	58
4.0 INTRODUCCIÓN	59
4.1 IMPLEMENTACIÓN DEL SISTEMA MECÁNICO	59
4.1.1 Base del sistema	59
4.1.2 Subsistema de oscilación sagital	60
4.1.3 Subsistema de oscilación transversal	60
4.1.4 Marco estructural principal y diseño final	61
4.2 IMPLEMENTACIÓN DEL SISTEMA EMBEBIDO	62
4.2.1 Pseudocódigo general del módulo de procesamiento	63
4.2.2. Adquisición de datos	64

4.2.3. Aplicación del PID discreto	64
4.2.4. Transmisión UART	65
4.3 IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE ACTUACIÓN	66
4.4 IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE SENSADO	67
4.5 IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE COMUNICACIÓN SERIAL	70
4.6 IMPLEMENTACIÓN DE LA INTERFAZ	72
4.6.1 Selector de etapa del temblor	73
4.6.2 Estado de conexión y control de adquisición	75
4.6.3.1 Panel de indicadores	77
4.6.3.2 Análisis espectral (FFT)	77
4.6.4 Pseudocódigo del sistema de interfaz gráfica	80
4.7 IMPLEMENTACIÓN DEL SISTEMA ELECTRÓNICO	81
4.8 MONTAJE FINAL DEL SISTEMA	82
4.9 DESARROLLO DE PROTOCOLO DE EVALUACIÓN USANDO EL PROTOTIPO FUNCIONAL IMPLEMENTADO.	83
4.9.2 PROCESAMIENTO COMPUTACIONAL DE LOS DATOS	84
4.9.3 RESULTADOS OBTENIDOS	84
Capítulo 5	86
CONCLUSIONES Y TRABAJOS FUTUROS	86
5.1 CONCLUSIONES	87
5.2 LIMITACIONES DEL DESARROLLO	88
5.3 TRABAJOS FUTUROS	89
6. ANEXOS	91

LISTA DE TABLAS

TABLA 1.2. TRABAJOS PREVIOS CLASIFICADOS SEGÚN SUS CARACTERÍSTICAS	19
2.1 Tabla marco metodológico , autoría propia.	24
Tabla 3.1.1 Requerimientos funcionales, autoría propia	27
Tabla 3.1.2. Requerimientos NO funcionales	28
Tabla 3.1.1. Resumen de casos de uso del sistema de emulación de vibraciones	30
Tabla 3.2 Funcionalidades principales del sistema	31
Tabla 3.4.6.3 Comparación entre arquitecturas mecánicas para emulación de vibraciones	39
.Tabla 3.5.1.3 Identificación de conjuntos funcionales en el sistema mecánico	43
Tabla 3.5.3.1. Parámetros del modelo dinámico identificado para la planta vibratoria..	46
Tabla 3.5.3.4.4 Principales indicadores de desempeño dinámico ante una entrada escalón en lazo cerrado.	53

Tabla 4.2.2.1. Resumen de tareas del sistema embebido en ESP32 bajo FreeRTOS	64
Tabla 4.3 Prueba módulo de actuación, autoría: propia.	68
Tabla 4.4. Prueba módulo de sensado, autoría: propia.	69
Tabla 4.5.1 Prueba módulo de comunicación serial, autoría: propia.	72
Tabla 4.6.1 Rango de frecuencia y características asociadas a cada etapa del temblor. Autoría: propia.	75
Tabla 4.6.2 Prueba del selector de etapa del temblor, autoría: propia.	76
Tabla 4.6.2 Prueba de visualización de frecuencia detectada, amplitud pico y desplazamiento estimado. Autoría: propia.	77
Tabla 4.6.3.1 Prueba de la visualización de la FFT , autoría: propia.	80
Tabla 4.9.3 Medidas observadas por tipo de paciente autoría: propia	85
Tabla 4.9.4. Resumen de errores y amplitudes autoría: propia	86

LISTA DE FIGURAS

Fig 1.1. Árbol del problema, autoría: propia.	15
Fig 3.1.1 Diagrama de casos de uso, autoría: propia.	26
Fig 3.3. Diagrama Interno de Bloques (IBD) del sistema, autoría: propia.	30
Fig 3.3.1 Diagrama de secuencia funcional del sistema, autoría: propia.	30
Fig 3.3.2 Esquema general de interacción entre módulos del sistema ParkiMotion. Autoría: propia.	31
Fig 3.4.1. Representación funcional del módulo de interfaz del sistema ParkiMotion y su interacción con los módulos de comunicación, sensado y procesamiento. Autoría propia.	32
Fig 3.4.2.2 MPU6050: Acelerómetro y giroscopio de 6 ejes, autoría: [29]	32
Fig 3.4.3.2. Sistema Operativo en Tiempo Real (RTOS), autoría: [31]	33
Fig 3.4.4.2. Motor paso a paso NEMA 17, autoría: [32]	34
Fig 3.4.5.2 Interfacing to UART with Computer, autoria [34]	35
Fig 3.5.1. Modelo CAD final del módulo mecánico con arquitectura tipo gimbal adaptada y subsistema de ajuste de amplitud.	39
Fig 3.5.3.1. Comparación entre la respuesta del modelo identificado y los datos reales de validación.	44
Fig 3.5.3.4.1 Bucla típica, autoría: propia.	46
Fig 3.5.3.4.4 Respuesta del sistema en lazo cerrado con controlador PID sintonizado, ante una entrada escalón de 3 Hz aplicada en $t=0,05$	49
Fig 3.5.4.1. Relación entre submódulos del módulo de procesamiento y control.	52
Fig 3.5.5.1. Esquemático eléctrico del sistema desarrollado en EasyEDA. Autoría: propia.	54
Fig 3.5.5.2. Diseño 2D de la PCB con distribución de componentes. Autoría: propia.	55
Fig 3.5.5.3. Vista 3D del diseño final de la PCB generada en EasyEDA. Autoría: propia.	55

Fig 4.1.2 Sistema de rotación en el eje Y, autoría: propia.	58
Fig 4.1.3 Sistema de rotación en el eje X, autoría: propia.	59
Fig 4.1.4. Vista frontal y lateral del marco estructural principal, autoría: propia.	60
Fig 4.4.1 Señales de aceleración (7HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia.	67
Fig 4.4.2 Gráfica FFT (7 HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia.	67
Fig 4.4.3 Señales de aceleración (5 HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia.	67
Fig 4.4.4 Gráfica FFT (5 HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia	68
Fig 4.4.5 Señales de aceleración (3 HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia.	68
Fig 4.4.6 Gráfica FFT (3 HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia	68
Fig 4.5.1 Captura del monitor serial mostrando los datos de aceleración transmitidos por UART desde el ESP32. Autoría: propia.	69
Fig 4.6.Vista general de la interfaz gráfica autoría propia	71
Fig 4.6.1 Prueba del selector de etapa del temblor en la interfaz Dash. Autoría: propia.	73
Fig 4.6.2. Estado de conexión del sistema y controles de adquisición de datos en la interfaz . Autoría: propia	75
Fig 4.6.3.1Espectro de frecuencia(3 Hz) generado mediante transformada rápida de Fourier (FFT). Autoría: propia	77
Fig 4.6.3.2 Registro temporal de amplitud pico. Autoría: propia	78
Fig 4.6.3.2 Panel de indicadores en la interfaz gráfica. Autoría: propia	78
Fig 4.7.1. Vista superior de la PCB . Autoría: propia.	80
Fig 4.7.2. Vista inferior de la PCB ensamblada. Autoría: propia.	80
Fig 4.8.1 Prototipo funcional ensamblado con los componentes accesibles en el contexto regional. Autoría: propia.	81

LISTA DE ALGORITMOS

Algoritmo 4.2.1. Pseudocódigo flujo de ejecución principal, autoría: propia.	62
Algoritmo 4.2.2. Pseudocódigo del módulo de adquisición de datos desde la IMU, autoría: propia.	63
Algoritmo 4.2.3. Pseudocódigo del módulo de control PID discreto, autoría propia.	64
Algoritmo 4.2.4. Pseudocódigo del módulo de control PID discreto, autoría propia.	65
Algoritmo 4.9.2 Pseudocódigo del análisis estadístico en Python, autoría propia.	83

LISTA DE ECUACIONES

Ecuación 3.5.3.1. Modelo masa-resorte-amortiguador.	45
Ecuación 3.5.3.2. Función de transferencia de la planta.	45
Ecuación 3.5.3.3. Identificación experimental del modelo.	46
Ecuación 3.5.3.4.1. Representación en tiempo continuo de un controlador PID clásico.	50
Ecuación 3.5.3.4.4 Representación en tiempo discreto del controlador PID.	51
Ecuación 3.5.3.4.5. Estructura general del controlador PID en tiempo discreto.	51
Ecuación 4.2.2. Conversión de aceleración a variable física para filtro complementario.	65
Ecuación 4.6.3.2 Cálculo del desplazamiento estimado a partir de la aceleración pico	79

GLOSARIO

EP: Enfermedad de Parkinson. Trastorno neurodegenerativo progresivo que afecta el sistema nervioso central, generando temblores, rigidez y bradicinesia.

ParkiMotion: Prototipo experimental diseñado para emular temblores típicos del Parkinson mediante un sistema mecatrónico controlado por interfaz gráfica.

IMU: Unidad de Medición Inercial. Sensor que registra aceleraciones y giros. En este proyecto se emplea el MPU6050.

Firmware: Conjunto de instrucciones en lenguaje de bajo nivel grabado en el microcontrolador, encargado del control del hardware.

Gimbal: Mecanismo de suspensión rotacional que permite generar vibraciones controladas en dos ejes, replicando el temblor en la mano.

RESUMEN

La enfermedad de Parkinson (EP) es uno de los trastornos neurológicos de mayor frecuencia en adultos mayores, la cual es caracterizada por temblores en reposo, rigidez muscular y alteraciones del movimiento voluntario. Estas alteraciones afectan la autonomía del paciente en actividades cotidianas, como la alimentación, escritura o higiene personal. Motivado en aportar al desarrollo de soluciones a esta problemática, el presente trabajo de grado propone el desarrollo de *ParkiMotion*: un prototipo funcional de plataforma experimental diseñado para emular los temblores en la mano de personas afectadas por EP, con el objetivo de facilitar el desarrollo de tecnologías asistivas y sistemas de compensación.

Para esto se planteó una metodología que integra teoría y herramientas que permitieron definir una arquitectura del sistema que se basa en una estructura tipo *gimbal* de dos grados de libertad, la cual permite la generación de vibraciones controladas mediante el accionamiento de motores paso a paso. Por su parte, el sistema de control se realiza a través de un microcontrolador ESP32 programado bajo FreeRTOS, que ejecuta tareas concurrentes para el control de los actuadores, lectura de sensores inerciales (MPU6050) y comunicación serial. Adicionalmente, la plataforma incorpora una interfaz gráfica desarrollada en Dash (Python), desde la cual el usuario puede seleccionar la etapa de temblor a emular (leve, moderada o severa), visualizar señales en tiempo real y exportar los datos adquiridos para su análisis.

Para validar las propuestas tecnológicas planteadas se desarrolló una validación experimental estableciendo un protocolo estructurado en tres escenarios clínicos, correspondientes a frecuencias de 3, 5 y 7 Hz. Se realizó un registro de la información por escenario que permitió analizar diferentes parámetros, tales como: frecuencia, amplitud pico y desplazamiento. El procesamiento de los datos incluyó técnicas de análisis espectral (FFT), filtrado digital y análisis estadístico. Los resultados demostraron que *ParkiMotion* tiene la capacidad de producir vibraciones con un error medio menor a ± 0.3 Hz, amplitudes coherentes con el perfil clínico simulado, y desplazamientos acordes al comportamiento físico esperado para cada tipo de paciente.

De esta forma, *ParkiMotion* se constituye como una herramienta experimental de bajo costo, configurable y robusta, con potencial de aplicación tanto en investigación biomédica como en el desarrollo de tecnologías de asistencia para pacientes con Parkinson. Por otra parte, su arquitectura modular permite futuras extensiones para incorporar control adaptativo, aprendizaje automático o integración con dispositivos de estimulación neuromuscular.

Palabras clave: Enfermedad de Parkinson, sistema embebido, control de vibraciones, interfaz gráfica, análisis de señales.

ABSTRACT

Parkinson's disease (PD) is one of the most common neurodegenerative disorders in older adults, characterized primarily by resting tremors, muscular rigidity, and impaired voluntary movement. These tremors, especially in the early stages, significantly affect the patient's autonomy in daily activities such as eating, writing, or personal hygiene. To address this issue, this thesis presents the development of *ParkiMotion*: a functional prototype of an experimental platform designed to accurately emulate the typical tremors associated with PD, with the aim of supporting the validation of assistive technologies and compensation systems.

The system architecture is based on a two-degree-of-freedom gimbal structure that enables controlled vibrations through the actuation of stepper motors. These motors are controlled by an ESP32 microcontroller running under FreeRTOS, which manages concurrent tasks for motor control, inertial sensor (MPU6050) reading, and serial communication. The platform integrates a user interface developed with Dash (Python), allowing users to select the tremor stage to emulate (mild, moderate, or severe), visualize real-time signals, and export acquired data for further analysis.

Experimental validation was performed through a structured protocol across three clinical scenarios, corresponding to vibration frequencies of 3, 5, and 7 Hz. A total of 217 samples per scenario were collected, analyzing parameters such as detected frequency, peak acceleration, and estimated displacement. Data processing included spectral analysis (FFT), digital filtering, and statistical evaluation. The results demonstrated that *ParkiMotion* can reproduce tremors with a mean error below ± 0.3 Hz, peak amplitudes consistent with clinical expectations, and displacements in line with the physical behavior expected for each patient profile.

In conclusion, *ParkiMotion* represents a low-cost, configurable, and robust experimental tool with potential applications in biomedical research and the development of assistive technologies for individuals with Parkinson's disease. Its modular design allows for future enhancements, including adaptive control, machine learning integration, or neuromuscular stimulation support.

Keywords: Parkinson's disease, embedded system, vibration control, graphical interface, signal analysis.

CAPÍTULO 1.

GENERALIDADES

1.0 INTRODUCCIÓN

La Enfermedad de Parkinson (EP) es una afección neurodegenerativa progresiva que impacta gravemente la calidad de vida de millones de personas en el mundo. Entre sus síntomas más debilitantes, el temblor involuntario en las extremidades superiores dificulta tareas cotidianas esenciales como la alimentación, reduciendo la autonomía y afectando el bienestar emocional de los pacientes [1]. Los temblores característicos de la EP, con oscilaciones involuntarias en un rango de 3 a 7 Hz, afectan directamente la precisión y estabilidad de los movimientos. Esta falta de control motor genera dificultades en actividades cotidianas, como la alimentación y el manejo de objetos, impactando negativamente la autonomía de los pacientes [2].

A pesar de los avances en tratamientos médicos y dispositivos asistivos, las soluciones actuales son limitadas en accesibilidad y costo, dejando a muchos pacientes sin alternativas efectivas para mejorar su independencia [3]. Los dispositivos estabilizadores han sido una estrategia empleada para mitigar el impacto del temblor, sin embargo, un área poco explorada es la capacidad de emular de manera controlada estos movimientos involuntarios. Contar con un dispositivo que reproduzca los temblores característicos del Parkinson de forma precisa y ajustable permitiría evaluar nuevas estrategias de mitigación, mejorar el diseño de tecnologías asistivas.

En respuesta a esta necesidad, este proyecto tiene como objetivo el diseño y construcción de un dispositivo automático que emule las vibraciones propias del Parkinson. Este prototipo permite generar temblores artificiales controlados para su aplicación en pruebas experimentales. Su implementación proporciona una plataforma confiable para el desarrollo de estrategias innovadoras de asistencia, contribuyendo a mejorar la calidad de vida de los pacientes con EP y facilita el desarrollo de dispositivos que ayudan a controlar o atenuar el temblor, mejorando la calidad de vida de las personas con Parkinson.

1.1 PLANTEAMIENTO DEL PROBLEMA

La enfermedad de Parkinson (EP) es un trastorno neurodegenerativo complejo que afecta múltiples áreas del sistema nervioso central y periférico. Se manifiesta con síntomas motores y no motores, incluyendo temblores, rigidez muscular, problemas de coordinación y, en algunos casos, dificultades cognitivas y emocionales [1]-[3]. Su prevalencia sólo es superada por el Alzheimer, y a nivel mundial se estima que 6.1 millones de personas la padecen, con una proyección de duplicación para 2040 [4].

En Colombia, la Asociación Colombiana de Neurología reporta más de 220.000 personas diagnosticadas con EP, quienes presentan afectaciones significativas en su movilidad y calidad de vida [3]. A nivel regional, el Departamento Administrativo Nacional de Estadística (DANE) ha registrado un incremento notable de casos en Cali [5], lo que evidencia la necesidad urgente de intervenciones tecnológicas que permitan mejorar la independencia y el bienestar de los afectados. Uno de los síntomas más limitantes de la EP es el temblor involuntario de los miembros superiores, lo que interfiere con la ejecución de tareas esenciales como la alimentación. La falta de precisión y estabilidad en los movimientos dificulta el manejo de utensilios, provocando derrames y reduciendo la autonomía del paciente. Esta condición no solo

afecta la capacidad funcional, sino que también impacta la autoestima y el bienestar emocional de quienes la padecen [6][7].

Para abordar este problema, se han desarrollado prototipos de cucharas inteligentes estabilizadoras, como Liftware Spoon [8] y Steady Spoon [9], que utilizan tecnología para compensar los temblores y facilitar la alimentación. Sin embargo, su elevado costo (entre 195 y 300 dólares) y la falta de cobertura por parte de los seguros médicos limitan su accesibilidad, especialmente en regiones con menor disponibilidad tecnológica [7]. Además, aunque existen dispositivos diseñados para mitigar los efectos del temblor, la investigación sobre la emulación controlada de estos movimientos es aún limitada y requiere un mayor desarrollo. Contar con un sistema que reproduzca el temblor de forma precisa y ajustable permite la evaluación de nuevas estrategias para atenuar y facilitar el desarrollo de tecnologías de asistencia más accesibles y efectivas. En este contexto, surge la siguiente pregunta de investigación:

¿Cómo se puede desarrollar un dispositivo automático que permita emular los temblores presentes en miembros superiores durante la alimentación en personas con Parkinson?

El presente proyecto se apoya en la revisión de soluciones tecnológicas previas, con el fin de desarrollar un prototipo que contribuya al estudio y diseño de dispositivos de asistencia más eficientes y asequibles.

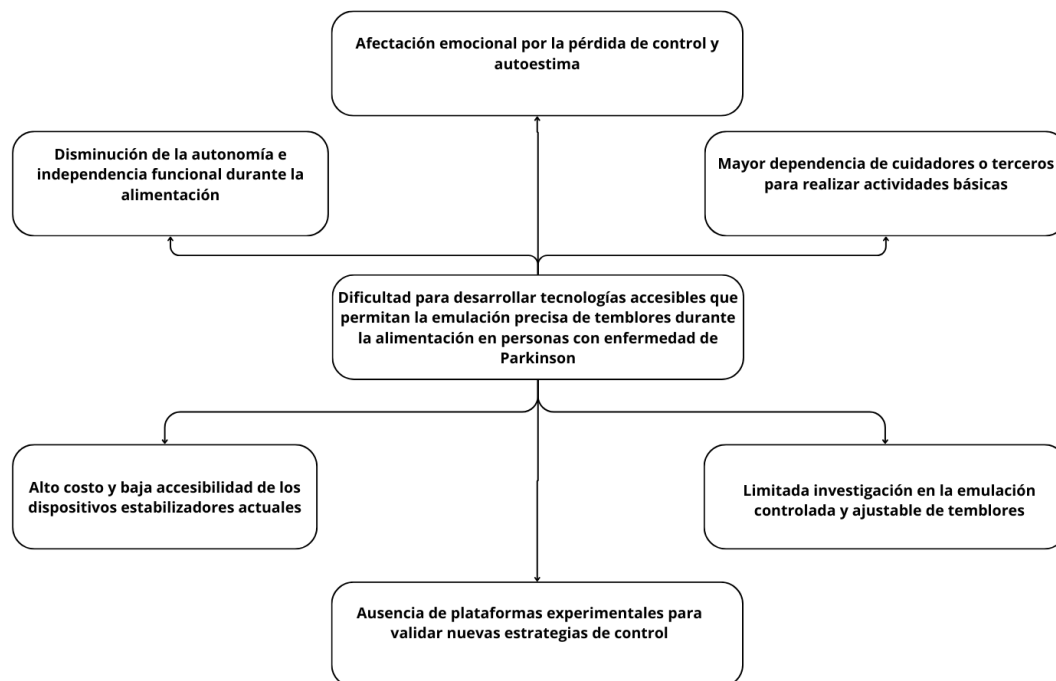


Fig 1.1. Árbol del problema, autoría: propia.

1.2 JUSTIFICACIÓN

La EP compromete la autonomía de los pacientes, especialmente durante la alimentación, al dificultar la manipulación de utensilios. En Colombia no se encuentran soluciones tecnológicas accesibles y adaptadas a la población, lo que resalta la necesidad de alternativas locales que mejoren la independencia y calidad de vida de los afectados.

Este proyecto propone el diseño y construcción de una **planta experimental automática** que emule los temblores característicos del Parkinson. Dicha plataforma permite replicar patrones de vibración mediante motores y sensores inerciales, facilitando el ajuste y validación de algoritmos de compensación en un entorno controlado. De esta manera, se establece una base tecnológica para el desarrollo de dispositivos de asistencia innovadores, eficaces y asequibles.

1.3 ANTECEDENTES

Teniendo en cuenta lo mencionado anteriormente, en este apartado se presenta el análisis de trabajos relevantes que desarrollan prototipos funcionales y metodologías para la caracterización, detección y mitigación de temblores en personas con la enfermedad de Parkinson, enfocándose en su aplicación en dispositivos experimentales y de asistencia. Estos trabajos se presentan resumidos en la **Tabla 1.2**, la cual incluye información sobre los componentes principales, los métodos de análisis o emulación del temblor, las tecnologías utilizadas y los resultados obtenidos después de su evaluación. Cabe mencionar que no se encontraron antecedentes específicos del desarrollo de una planta experimental para la emulación de temblores en el contexto local. A continuación, se realiza una descripción funcional de los trabajos representativos relacionados con el diseño y validación de sistemas para la medición y compensación de temblores, los cuales sirven como base para el desarrollo de la presente propuesta

TABLA 1.2. TRABAJOS PREVIOS CLASIFICADOS SEGÚN SUS CARACTERÍSTICAS

Título del Estudio	Componentes Principales	Método de Análisis o Emulación	Resultados y Evaluación	Tecnologías Utilizadas
Three-Axis Gimbal Design and Its Application [8]	IMU, motores brushless servo, PID controller	Compensación de movimiento con motores en ejes Pitch, Yaw y Roll	Mejora de estabilidad de imagen en video, resultados simulados en MATLAB	IMU de 6 ejes, control PID, filtro complementario, motores brushless servo
Design and Simulation of a Customize Three-axis Gimbal Structure [9]	Gimbal de tres ejes, análisis de elementos finitos (FEA), material PLA	Simulación de cargas y resistencia estructural en SolidWorks	Capacidad de soportar hasta 23.54 kg antes de fractura	Software SolidWorks para simulación FEA, impresión 3D con PLA
Evaluación de Dispositivos de Mitigación de Temblor para Parkinson [10]	IMU, motores, algoritmos de control adaptativo	Medición de temblores y aplicación de contramedidas	Dispositivos reducen temblores hasta en un 50% en pruebas de laboratorio	Sensores inerciales, algoritmos de control, hardware embebido
Quantification of Hand Motor Symptoms in Parkinson's Disease [11]	Sensores de movimiento	Análisis de síntomas motores	Cuantificación precisa de temblores	IMU, procesamiento de señales
Detection of Hand Tremor in Patients with Parkinson's Disease [12]	Sensores inerciales	Algoritmos de detección de temblor	Identificación precisa de patrones de temblor	IMU, machine learning
Hand-Tremor Frequency Estimation in Videos [13]	Cámaras de alta velocidad	Análisis en video	Estimación precisa de la frecuencia del temblor	Procesamiento de imágenes
Hand Rhythmic Tapping and Timing in Parkinson's Disease [14]	Dispositivos de medición rítmica	Análisis del movimiento secuencial	Evaluación del control motor	Sensores de presión, captura de movimiento
Hand Tremor Characterization from a Spatiotemporal Representation [15]	Modelado espaciotemporal	Representación de señales de temblor	Mayor comprensión de la variabilidad del temblor	Algoritmos de procesamiento de señales

Los estudios revisados aportan parámetros cuantitativos (frecuencia entre 3–7 Hz, amplitud de 1–12 mm, reducción del temblor hasta en un 50 %, cargas soportadas hasta 23 kg) y cualitativos (arquitecturas tipo gimbal, control PID, algoritmos adaptativos, análisis por visión artificial) que orientaron el diseño de la planta experimental. Esta información permitió establecer rangos realistas de emulación y seleccionar tecnologías accesibles y replicables en el contexto regional.

En este marco, se definieron tres frecuencias de referencia 3 Hz, 5 Hz y 7 Hz que representan escenarios leves, moderados y severos, respectivamente, dentro del rango característico de los temblores parkinsonianos.

1.4 OBJETIVOS

1.4.1 Objetivo general

Desarrollar un dispositivo para emular las vibraciones en la mano en personas con la enfermedad de Parkinson.

1.4.2 Objetivos específicos

- Definir los requerimientos del dispositivo basado en análisis de literatura.
- Concebir la arquitectura del dispositivo mediante una orientación modular.
- Verificar los componentes de la arquitectura propuesta basado en modelos analíticos, computacionales o experimentales.
- Desarrollar un prototipo funcional considerando la tecnología regional.
- Desarrollar un protocolo de evaluación del prototipo funcional implementado.

El marco teórico presenta los fundamentos clínicos y tecnológicos necesarios para comprender la problemática de la Enfermedad de Parkinson (EP) y orientar el diseño del prototipo de emulación de temblores.

1.5.1 Enfermedades neurodegenerativas

Las enfermedades neurodegenerativas implican la pérdida progresiva de neuronas, afectando funciones como el movimiento y la coordinación. En este grupo se encuentra la EP, segunda en prevalencia después del Alzheimer, cuya manifestación motora más característica son los temblores [16].

1.5.2 Anatomía de la mano

Los temblores del Parkinson afectan principalmente los músculos flexores y extensores de la mano y el antebrazo, comprometiendo tareas finas como la alimentación [17]. Esta afectación motriz es el foco del dispositivo diseñado, el cual busca emular dichas vibraciones.

1.5.3 Enfermedad de Parkinson

La EP se debe a la pérdida de neuronas dopaminérgicas en la sustancia negra, lo que altera los circuitos de los ganglios basales y produce oscilaciones patológicas que se manifiestan como temblores [18][20].

Los principales síntomas motores incluyen:

- **Temblor en reposo:** oscilaciones rítmicas de 3–7 Hz en manos, brazos o mandíbula.
- **Rigidez muscular:** resistencia irregular al movimiento, tipo “cogwheel”.
- **Bradicinesia:** lentitud y reducción en la amplitud de los movimientos [21][23].
Estos parámetros (frecuencia y amplitud) sirven como referencia directa para calibrar el prototipo.

1.5.4 Estadios de la EP

La escala de Hoehn y Yahr describe la progresión desde síntomas leves unilaterales (estadio 1) hasta dependencia total (estadio 5) [25]. El dispositivo propuesto se enfoca en **estadios iniciales**, donde los temblores predominan y afectan la autonomía en la alimentación.

1.5.5 Sistemas automáticos y de control

Un sistema automático integra sensores, controlador y actuadores para regular un proceso sin intervención humana [26][27]. En este proyecto:

- **Sensores:** IMU MPU6050 para medir aceleraciones.
- **Controlador:** PID digital implementado en ESP32 con FreeRTOS.
- **Actuadores:** motores paso a paso NEMA 17 con drivers DRV8825.
Este esquema permite emular temblores con parámetros ajustables y controlados

1.5.6 Protocolos de prueba

Un protocolo de pruebas define procedimientos estandarizados para evaluar variables relevantes [28]. En este proyecto se estableció un protocolo experimental basado en frecuencias de 3, 5 y 7 Hz, representativas del rango clínico del Parkinson, con el fin de validar el desempeño del prototipo.

1.6 ESTRUCTURA DEL DOCUMENTO

Este documento se encuentra estructurado en cinco capítulos, organizados en correspondencia directa con los objetivos específicos definidos para el desarrollo del sistema de emulación de temblores en personas con enfermedad de

Parkinson. La organización propuesta permite una lectura progresiva, facilitando la comprensión de la problemática abordada, los fundamentos técnicos, la arquitectura funcional, el desarrollo del prototipo, el diseño del sistema de control y la validación experimental. A continuación, se describe la estructura del documento:

- **Capítulo 1: Planteamiento del problema, antecedentes y objetivos del proyecto**

Se presenta el contexto clínico y tecnológico de la enfermedad de Parkinson, la revisión de antecedentes, la formulación del problema, la justificación, los objetivos generales y específicos, así como el alcance del proyecto.

- **Capítulo 2: Marco metodológico**

Se describe el enfoque metodológico adoptado para alcanzar los objetivos planteados. Se detallan las etapas de desarrollo, las actividades correspondientes a cada objetivo específico, los recursos requeridos y los indicadores de cumplimiento.

- **Capítulo 3: Diseño del sistema ParkiMotion**

Se desarrolla la arquitectura modular del sistema, desglosada en seis módulos funcionales. Se presenta el análisis de requerimientos funcionales y no funcionales, los diagramas de bloques, las justificaciones técnicas de diseño mecánico, electrónico y embebido, y se detalla la implementación conceptual del controlador PID.

- **Capítulo 4: Prototipo experimental y validación del sistema**

Se documenta la implementación física del sistema completo, incluyendo el ensamblaje mecánico, diseño de PCB, firmware embebido con FreeRTOS y la interfaz gráfica en Dash. Se describen las pruebas realizadas sobre los distintos módulos y se presentan resultados obtenidos a partir de ensayos controlados.

- **Capítulo 5: Conclusiones, limitaciones y trabajos futuros**

Se presentan las conclusiones obtenidas con base en los resultados experimentales, se identifican las principales limitaciones encontradas durante el desarrollo del prototipo, y se proponen líneas de trabajo orientadas a futuras mejoras y aplicaciones potenciales del sistema ParkiMotion.

CAPÍTULO 2

MARCO METODOLÓGICO

2.1. MARCO METODOLÓGICO

El presente proyecto se desarrolló bajo un enfoque experimental y tecnológico, estructurado en etapas que permitieron cumplir con los objetivos planteados. A continuación, se presenta la tabla metodológica que resume las actividades realizadas, los recursos utilizados y los resultados esperados para cada fase del proceso.

2.1 Tabla marco metodológico, autoría propia.

Objetivo específico	Actividades	Recursos	Indicador de cumplimiento
1. Definir los requerimientos del dispositivo basado en análisis de literatura y consultas con expertos.	1.1. Se revisó literatura científica sobre emulación de temblores y tecnologías aplicadas a la enfermedad de Parkinson (EP). 1.2. Se analizaron estudios previos y modelos matemáticos para caracterizar los temblores asociados al proceso de alimentación. 1.3. Se documentaron los hallazgos obtenidos, organizando y analizando la información más relevante.	- Acceso a bibliografía científica. - Acceso a información de médicos especializados en el tema.	Documento de requerimientos elaborado.
2. Especificar la arquitectura del dispositivo mediante una orientación modular.	2.1. Se estructuró el sistema en módulos funcionales enfocados en procesamiento, estabilización, detección y actuación. 2.2. Se asignó jerarquía entre los módulos, detallando funciones y responsabilidades. 2.3. Se creó un diagrama de flujo de datos representando la interacción entre módulos. 2.4. Se documentaron las funciones, interfaces y relaciones de cada módulo.	- Modelos para definir la arquitectura.	Modelo de arquitectura completo.

3. Verificar los componentes de la arquitectura propuesta basado en modelos analíticos, computacionales o experimentales.	3.1. Se identificó y seleccionó el sistema de control más adecuado. 3.2. Se formularon modelos analíticos del comportamiento del sistema. 3.3. Se realizaron pruebas de laboratorio y simulaciones. 3.4. Se compararon los resultados obtenidos con las especificaciones definidas. 3.5. Se documentaron los resultados de las pruebas y simulaciones.	- Equipo de laboratorio. - Software de simulación. - Componentes del prototipo.	Resultados de pruebas verificados.
4. Desarrollar un prototipo funcional considerando la tecnología regional.	4.1. Se codificaron los algoritmos para la implementación del sistema de control. 4.2. Se definió el esquema de las placas electrónicas. 4.3. Se manufacturaron las placas electrónicas. 4.4. Se diseñaron los componentes mecánicos. 4.5. Se fabricaron los componentes mecánicos. 4.6. Se ensambló el prototipo funcional. 4.7. Se documentaron los códigos, placas y piezas elaboradas y ensambladas.	- Equipo de desarrollo y diseño. - Espacio para pruebas. - Materiales para prototipado.	Espacio para pruebas con la planta emuladora.
5. Desarrollar un protocolo de evaluación usando el prototipo funcional implementado.	5.1. Se verificó experimentalmente un prototipo funcional para emular las vibraciones asociadas a la enfermedad de Parkinson. 5.2. Se compararon los resultados experimentales con los valores esperados. 5.3. Se documentaron los resultados y conclusiones del ensayo experimental.	- Espacio para pruebas con la planta emuladora.	Protocolo de evaluación completado. Prototipo funcional fabricado.

CAPÍTULO 3

DISEÑO Y ARQUITECTURA DEL SISTEMA PARKIMOTION

3.1 DEFINICIÓN DE REQUISITOS

Para estructurar la arquitectura funcional del sistema se definieron requisitos que delimitan su propósito y comportamiento esperado. Estos se clasificaron en funcionales y no funcionales, constituyendo la base del diseño modular del prototipo.

3.1.1 Requerimientos Funcionales y No Funcionales

En la Tabla 3.1.1 se presentan los requerimientos funcionales para el desarrollo del sistema. Esta tabla se compone de tres columnas donde se tiene:

- **Numeración de Referencia:** Ref#, de cada requerimiento funcional.
- **Funciones:** Descripción corta del requerimiento funcional.
- **Categoría:** Puede ser E (Esencial) u O (Opcional)

Los requerimientos funcionales se formulan con base en las capacidades esperadas del sistema en su versión final.

Tabla 3.1.1 Requerimientos funcionales, autoría propia

Ref #	Funciones	Categoría
1.0	Interfaz de usuario	
1.1	Permitir la selección de la etapa del temblor a emular (leve, moderado o severo)	E
1.2	Visualizar las señales inerciales provenientes del sensor	E
1.3	Mostrar la frecuencia dominante del temblor mediante análisis FFT	E
1.4	Enviar comandos de control (iniciar, pausar, seleccionar etapa clínica) al microcontrolador	E
1.5	Visualizar la amplitud pico y desplazamiento estimado a partir de la señal de aceleración	E
1.6	Permitir el registro de datos en archivos CSV desde la interfaz gráfica	E
2.0	Microcontrolador	
2.1	Activar y controlar dos motores paso a paso independientes para generar oscilaciones angulares controladas. Cada motor debe producir una frecuencia de oscilación entre 3 Hz y 7 Hz, con una resolución ajustable mediante microstepping.	E
2.3	Detectar la frecuencia de la vibración a partir de los datos de aceleración.	E
2.4	Clasificar automáticamente la etapa del temblor en leve, moderado o severo según los valores medidos.	E
2.5	Coordinar las tareas de lectura de sensores, control de motores y comunicación serial bajo un esquema multitarea (FreeRTOS)	E
2.6	Aplicar un controlador PID digital para que la frecuencia deseada sea seguida por la frecuencia vibratoria generada	E
3.0	Sistema de emulación	
3.1	Generar oscilaciones angulares horizontales (eje X) mediante un motor paso a paso montado en la parte superior	E
3.2	Generar oscilaciones angulares verticales (eje Y) mediante un segundo motor montado lateralmente	E

En la Tabla 3.1.2 se presentan los requerimientos no funcionales, estos son consideraciones necesarias para el desarrollo del sistema, sin embargo, no afectan la lógica del sistema.

Tabla 3.1.2. Requerimientos NO funcionales

Ref #	Funciones	Categoría
1.0	Software	
1.1	El software del sistema fue desarrollado en lenguaje de programación C	E
1.2	Se utilizó el kernel FreeRTOS v10.4.3	E
1.3	Se hizo uso del editor de código Visual Studio Code V1.74.2	O
1.4	Se estructuró el código en módulos para facilitar la escalabilidad y el mantenimiento	E
1.5	Se utilizó Dash en Python para el desarrollo de la interfaz	E
2.0	Hardware	
2.1	Se utilizó el microcontrolador ESP32 DevKit V1	E
2.2	Se utilizó el sensor inercial MPU6050 para capturar aceleraciones y velocidades angulares	E
2.4	Se utilizaron dos drivers DRV8825 para el control de los motores paso a paso	E
2.7	Se utilizó un sistema mecánico modular con perfiles de aluminio y placas acrílicas	E
2.8	Se utilizó un sistema dual UART para separar lectura y escritura entre el ESP32 y la interfaz Dash	E

3.1.2 Diagrama de casos de uso

El diagrama de casos de uso (Fig. 3.1.1) representa las interacciones entre el usuario y el sistema, identificando funciones como seleccionar etapa del temblor, iniciar/pausar la emulación y visualizar parámetros vibratorios.

La **Tabla 3.1.3** resume los casos de uso principales (CU01–CU07), los cuales se describen en detalle en el anexo *arquitectura_sistema*.

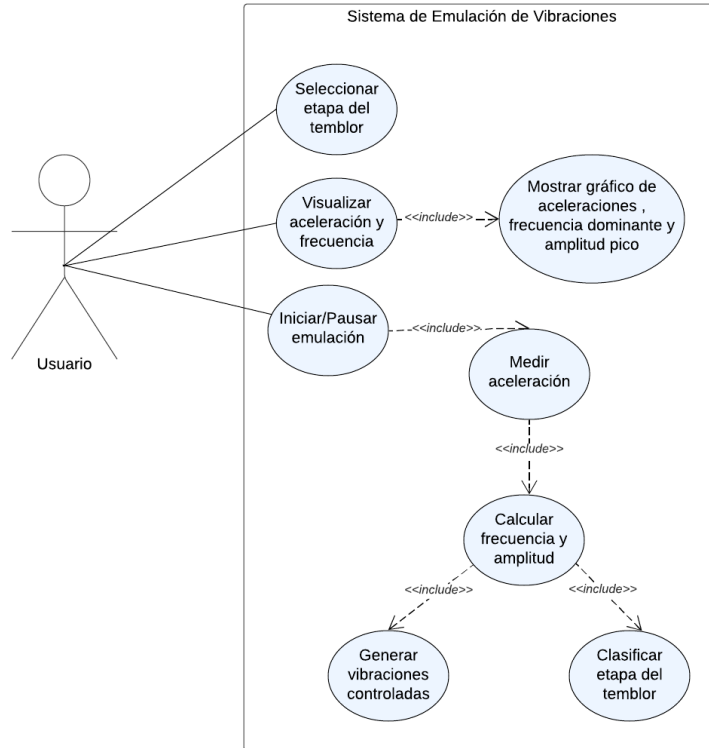


Fig 3.1.1 Diagrama de casos de uso, autoría: propia.

En el diagrama de casos de uso se identifican los actores clave que interactúan con el sistema, así como las funciones principales que este debe proporcionar. A continuación, se describen:

3.1.2.1 ACTOR: USUARIO

Representa a la persona que opera directamente el sistema a través de la interfaz gráfica. Este actor puede ser un investigador, profesional de la salud, estudiante o técnico que requiera emular temblores específicos para fines de evaluación, pruebas o investigación.

1. El usuario tiene acceso a las siguientes funciones del sistema:

- **Seleccionar etapa del temblor:** permite definir si la emulación corresponde a un temblor leve, moderado o severo.
- **Iniciar o pausar la emulación:** habilita el control del ciclo de prueba, activando o deteniendo el sistema.
- **Visualizar aceleración, frecuencia y amplitud pico:** muestra en tiempo real los datos recolectados por los sensores, incluyendo gráficas e indicadores numéricos.

2. Funciones internas del sistema (casos de uso automáticos)

Las siguientes funciones no son activadas directamente por el usuario, sino que son ejecutadas por el sistema como parte del flujo interno tras recibir un comando como "iniciar":

- **Medir aceleración:** adquisición de datos desde los sensores (inercial y ultrasónico).
- **Calcular frecuencia y amplitud:** procesamiento de los datos sensados para caracterizar la vibración.
- **Clasificar etapa del temblor:** evaluación automática de los parámetros medidos para validar o ajustar la etapa emulada.
- **Generar vibraciones controladas:** activación del sistema de motores para replicar el temblor correspondiente.

La siguiente tabla presenta un resumen de los principales casos de uso identificados en el sistema de emulación de vibraciones. Cada caso refleja una funcionalidad clave dentro del flujo de operación del sistema, estableciendo la relación entre el usuario y los distintos módulos internos, así como las acciones automatizadas. Esta tabla permite tener una visión general de las interacciones más relevantes.

Tabla 3.1.1. Resumen de casos de uso del sistema de emulación de vibraciones

ID	Nombre del Caso de Uso	Actor	Descripción breve
CU01	Seleccionar etapa del temblor	Usuario	El usuario elige la etapa de temblor (leve, moderada o avanzada) a emular.
CU02	Visualizar aceleración y frecuencia	Usuario	Se muestran en tiempo real las señales inerciales y su análisis espectral (FFT).
CU03	Iniciar/Pausar emulación	Usuario	Permite al usuario activar o detener la generación de vibraciones.
CU04	Medir aceleración	Sistema	Se adquieren datos del sensor IMU para análisis.
CU05	Calcular frecuencia y amplitud	Sistema	El sistema procesa las señales para determinar parámetros vibratorios.
CU06	Clasificar etapa del temblor	Sistema	Se compara la frecuencia y amplitud con umbrales para clasificar la etapa actual.
CU07	Generar vibraciones controladas	Sistema	El sistema envía señales PWM para controlar los motores que generan oscilaciones.

Para una descripción detallada de cada caso de uso, incluyendo actores, precondiciones, flujo de eventos, postcondiciones y excepciones, se remite al anexo: **arquitectura_sistema**, donde se encuentran las tablas de caso de uso mencionadas anteriormente.

3.2 DESCRIPCIÓN FUNCIONAL DEL SISTEMA

La identificación funcional se realizó con un enfoque Top-Down, descomponiendo el dispositivo en módulos interdependientes. La Tabla 3.2 organiza estas funciones jerárquicamente en seis bloques: interfaz, sensado, procesamiento, actuación, comunicación y mecánico.

Cada módulo cumple un rol específico: la interfaz gestiona la interacción con el usuario; el sensado mide aceleraciones en los tres ejes; el procesamiento calcula frecuencia y amplitud; la actuación genera vibraciones con motores paso a paso; la comunicación enlaza al ESP32 con la interfaz; y el módulo mecánico provee la estructura tipo gimbal para el montaje.

El diagrama de bloques (Fig. 3.2) muestra la organización general del sistema.

Tabla 3.2 Funcionalidades principales del sistema

Ref #	Funciones	Categoría
1.0	Módulo de Interfaz	
1.1	Mostrar en pantalla los datos de aceleración y frecuencia de vibración y amplitud pico en tiempo real	E
1.2	Permitir la selección de la etapa del temblor a emular (leve, moderado o severo)	E
1.3	Permite guardar los resultados en CSV, para posterior análisis	E
2.0	Módulo de Sensado	
2.1	Medir vibraciones en un rango de 3 Hz a 7 Hz	E
2.2	Detectar aceleraciones en los tres ejes espaciales (X, Y, Z)	E
3.0	Módulo de Procesamiento	
3.1	Procesar los datos de aceleración para calcular amplitud y frecuencia de vibraciones	E
3.2	Determinar la etapa del temblor con base en los parámetros vibratorios calculados	E
4.0	Módulo de Actuación	
4.1	Controlar de forma independiente dos motores paso a paso para generar vibraciones controladas	E
4.2	Controlar la amplitud de vibración mediante la modulación de la velocidad angular de los motores paso a paso.	E
5.0	Módulo De Comunicación	
5.1	Transmisión de datos hacia una PC o plataforma externa mediante UART	E
5.2	Permitir la activación y desactivación remota del sistema de vibraciones mediante comandos externos	O
6.0	Módulo mecánico	

6.1	Proporcionar una estructura modular y estable para el montaje de los componentes del sistema	E
6.2	Permitir el movimiento oscilatorio en dos grados de libertad mediante un diseño tipo gimbal	E
6.3	Asegurar el aislamiento de vibraciones externas a través de materiales y elementos mecánicos	O

Con el fin de representar de manera clara y estructurada la organización funcional del sistema, se elaboró un diagrama de definición de bloques. Este diagrama permite visualizar los principales módulos que conforman el dispositivo, así como las interacciones entre ellos. La Figura 3.2 muestra esta representación, que sirvió como base para el diseño e implementación de la arquitectura general del sistema.

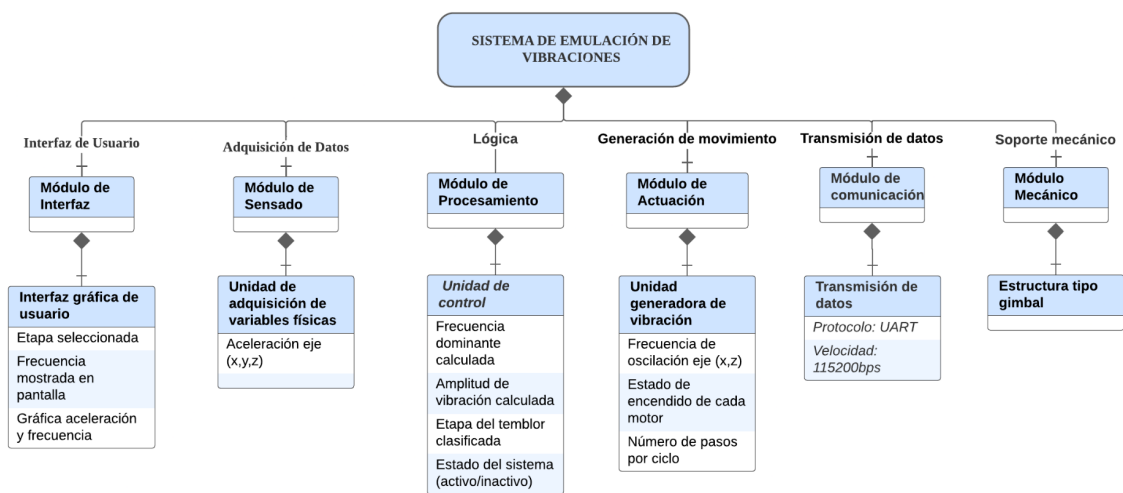


Fig 3.2. Diagrama de definición de bloques, autoría: propia.

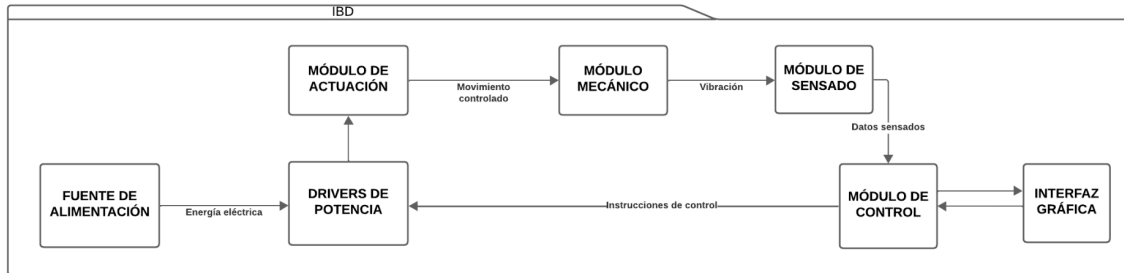
El sistema se estructuró en seis módulos funcionales: **interfaz**, encargada de la interacción con el usuario; **sensado**, responsable de medir aceleraciones en los tres ejes dentro del rango de 3–7 Hz; **procesamiento**, núcleo lógico que calcula frecuencia, amplitud y clasifica la etapa del temblor; **actuación**, que controla los motores paso a paso para generar oscilaciones; **comunicación**, que enlaza al microcontrolador con la interfaz mediante UART; y **mecánico**, que proporciona la estructura tipo gimbal y asegura la estabilidad del conjunto. Esta organización modular facilita la comprensión, integración y escalabilidad del prototipo.

3.3 FLUJO FUNCIONAL DEL SISTEMA

El flujo funcional inicia cuando el usuario selecciona la etapa de temblor en la interfaz gráfica, que transmite los comandos al ESP32 mediante UART. El microcontrolador coordina la lectura de datos del sensor MPU6050, activa los motores paso a paso y aplica control PID para seguir la frecuencia deseada.

En paralelo, los datos adquiridos son enviados a la interfaz, donde se realiza el procesamiento (FFT, amplitud pico, desplazamiento estimado) y se registran automáticamente en archivos

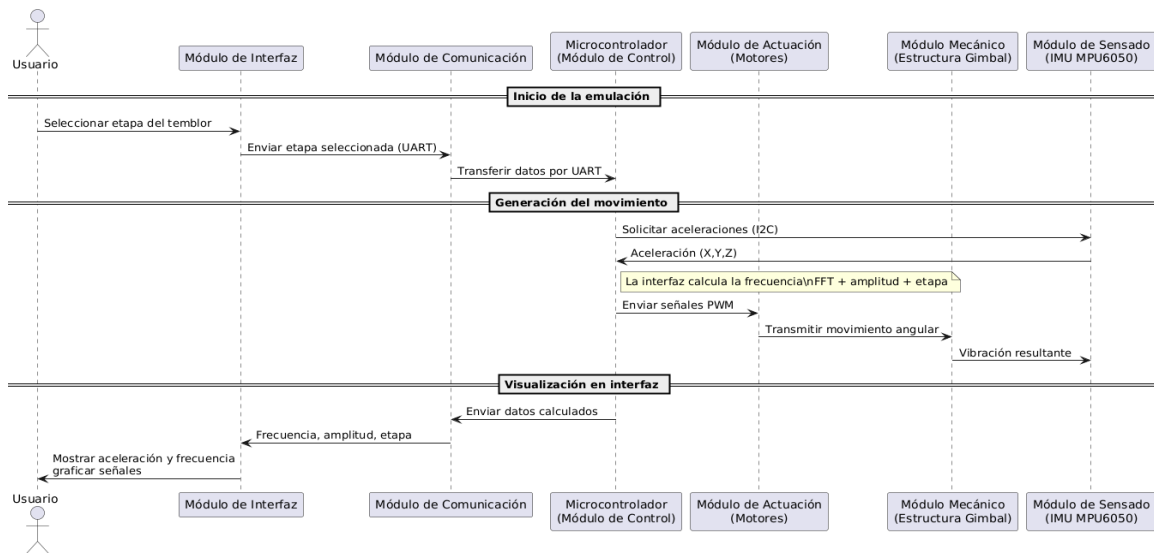
CSV. La estructura mecánica tipo gimbal permite la generación de oscilaciones angulares en dos grados de libertad, replicando los patrones de temblor característicos. El flujo de datos, control y energía se ilustra en el diagrama IBD (Fig. 3.3) y en el diagrama de secuencia funcional (Fig. 3.3.1).



Fig

3.3. Diagrama Interno de Bloques (IBD) del sistema, autoría: propia.

El flujo funcional integra tres niveles principales: **energía**, proveniente de la fuente de 12 V hacia los drivers DRV8825 que accionan los motores paso a paso; **control**, iniciado en la interfaz gráfica (Dash) que envía comandos al ESP32 mediante UART, el cual ejecuta la señal a los actuadores; y **datos**, adquiridos por el sensor MPU6050 y transmitidos al microcontrolador vía I2C para su envío y procesamiento en la interfaz (FFT, amplitud, desplazamiento y clasificación del temblor). Estas interacciones se ilustran en el diagrama interno de bloques (Fig. 3.3) y en el diagrama de secuencia funcional (Fig. 3.3.1), que representan la dinámica completa del



sistema.

Fig 3.3.1 Diagrama de secuencia funcional del sistema, autoría: propia.

La Fig. 3.3.2 resume de forma esquemática el flujo de interacción entre los distintos módulos que conforman ParkiMotion. En ella se observa cómo el usuario, a través de la interfaz gráfica desarrollada en Dash, envía parámetros de referencia al sistema embebido (ESP32 con FreeRTOS). Este procesa la información y genera señales de control hacia los drivers DRV8825 y motores paso a paso, los cuales inducen las vibraciones en la estructura mecánica tipo gimbal. De manera simultánea, la IMU MPU6050 registra las aceleraciones producidas y retroalimenta el lazo de control mediante la transmisión de datos hacia la interfaz. Este esquema integra en una sola vista los bloques de usuario, comunicación, procesamiento, actuación y sensado, facilitando la comprensión global del sistema.

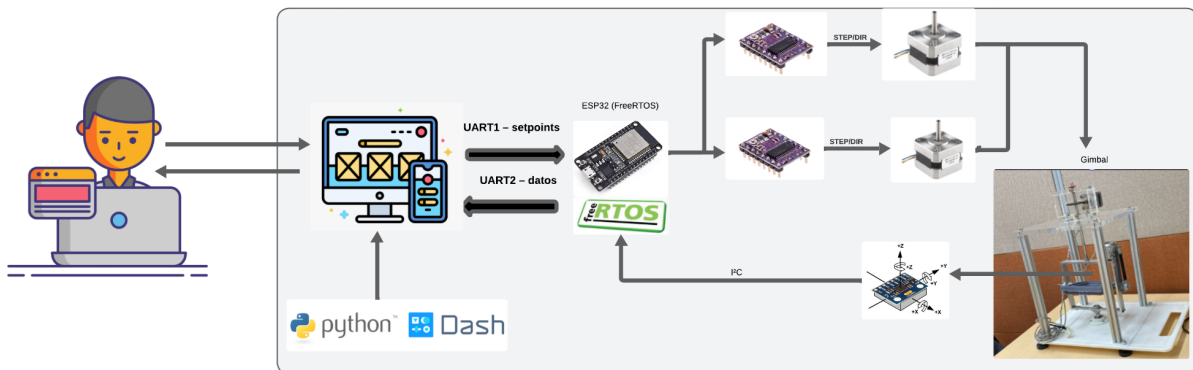


Fig 3.3.2 Esquema general de interacción entre módulos del sistema ParkiMotion. Autoría: propia.

3.4 MÓDULOS FUNCIONALES

Esta sección presenta la descripción detallada de los módulos funcionales que componen el sistema de emulación de vibraciones diseñado en este proyecto. El enfoque se centró en la arquitectura electrónica y de control, estructurada de forma modular para facilitar la integración, el análisis funcional y la escalabilidad del sistema.

Cada módulo se documentó considerando tres aspectos fundamentales:

- Funciones principales que desempeña dentro del sistema.
- Componentes electrónicos que lo conforman, con su respectiva justificación técnica.
- Interfaces de comunicación y relaciones con los demás módulos, tanto a nivel físico como lógico.

3.4.1 MÓDULO DE INTERFAZ

El módulo de interfaz constituye el punto de contacto entre el usuario y el sistema ParkiMotion. Permite configurar parámetros experimentales, monitorear en tiempo real el comportamiento vibratorio y gestionar los datos generados, actuando como capa de supervisión y control dentro de la arquitectura.

Funciones principales:

- Selección del tipo de paciente (leve, moderado o severo)
- Definición de la frecuencia deseada como referencia de control.
- Visualización de aceleración, frecuencia detectada, amplitud pico y desplazamiento.
- Activación/detención del proceso de adquisición y almacenamiento de datos.
- Monitoreo del estado de conexión con el sistema embebido.

Interacción con otros módulos:

El módulo se comunica con el microcontrolador vía UART, recibe datos de la IMU para análisis y muestra al usuario los resultados procesados (filtrado y FFT). La Fig. 3.4.1 resume gráficamente estas interacciones.

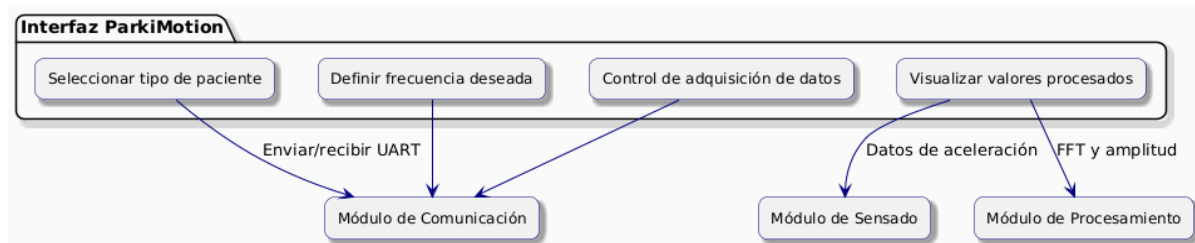


Fig 3.4.1. Representación funcional del módulo de interfaz del sistema ParkiMotion y su interacción con los módulos de comunicación, sensado y procesamiento. Autoría propia.

3.4.2 MÓDULO DE SENSADO

Este módulo emplea un sensor inercial MPU6050 para registrar aceleraciones en los tres ejes (X, Y, Z), variables que permiten estimar frecuencia dominante y amplitud pico de vibración. La información adquirida se transmite al ESP32 mediante I2C y constituye la base para el análisis espectral y la validación del comportamiento dinámico.

Funciones principales:

- Captar aceleraciones lineales en los ejes X, Y y Z.
- Transmitir datos en tiempo real al microcontrolador.
- Proveer insumos para estimación de frecuencia y amplitud.
- Servir como retroalimentación para validar la salida del módulo de actuación.

Componente principal: MPU6050 (IMU de 6 DoF con acelerómetro y giroscopio). En este proyecto se emplea únicamente el acelerómetro por ser el más representativo del fenómeno vibratorio.

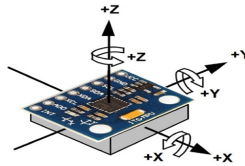


Fig 3.4.2.2 MPU6050: Acelerómetro y giroscopio de 6 ejes, autoría: [29]

3.4.3 MÓDULO DE PROCESAMIENTO

El núcleo lógico del sistema está constituido por un microcontrolador **ESP32-WROOM-32**, seleccionado por su capacidad de procesamiento concurrente bajo **FreeRTOS** y su compatibilidad con múltiples interfaces (UART, I2C, PWM). Esta unidad gestiona en tiempo real las tareas de sensado, control y comunicación con la interfaz de usuario.

Funciones principales:

- Controlar los motores paso a paso mediante señales PWM.
- Adquirir datos de la IMU (MPU6050) vía I2C.
- Mantener comunicación serial bidireccional con la interfaz gráfica.
- Recibir comandos del usuario (inicio, pausa, selección de etapa clínica).

El uso de **FreeRTOS** asegura ejecución multitarea determinista, permitiendo controlar motores, leer sensores y comunicar datos de forma sincronizada sin bloqueos.

Componentes principales:

- ESP32 -WROOM-32:

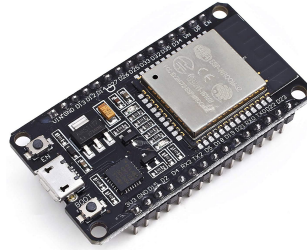


Fig 3.4.3. Vista superior del módulo ESP32-WROOM-32. Fuente: [30]

- Sistema Operativo en Tiempo Real (RTOS).

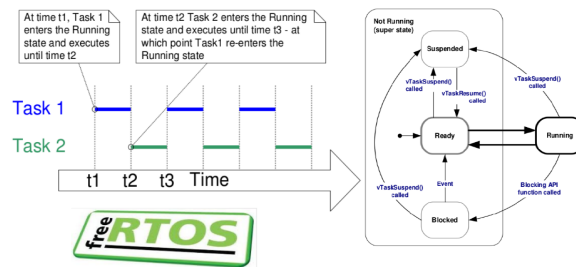


Fig 3.4.3.2. Sistema Operativo en Tiempo Real (RTOS), autoría: [31]

3.4.4 MÓDULO DE ACTUACIÓN

Este módulo genera las oscilaciones físicas que simulan los temblores de Parkinson a partir de señales digitales recibidas desde el ESP32. Para ello emplea dos motores paso a paso **NEMA 17**, dispuestos en ejes ortogonales, accionados por drivers **DRV8825**.

Funciones principales:

- Generar vibraciones controladas mediante dos motores paso a paso independientes.
- Ejecutar las instrucciones de frecuencia y amplitud definidas por el usuario.
- Proporcionar una base física reproducible para validar el sistema de sensado.

3.4.4.1 Componentes principales:

1. Motores paso a paso NEMA 17 (x3)



Fig 3.4.4.2. Motor paso a paso NEMA 17, autoría: [32]

2. Drivers DRV8825 (x3)

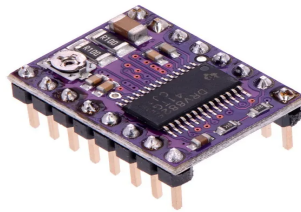


Fig 3.4.4.3. DRV8825 Módulo driver para motor paso a paso, autoría: [33]

3.4.5 MÓDULO DE COMUNICACIÓN

Este módulo gestionó la conexión de datos entre el sistema embebido y la interfaz de usuario, permitiendo una interacción fluida entre el entorno físico y el entorno digital.

3.4.5.1 Funciones principales:

1. Transmitir datos sensados desde el ESP32 hacia la interfaz.
2. Recibir comandos de control desde la interfaz.

3.4.5.2 Componentes principales:

1. Interfaz UART a 115200 baudios

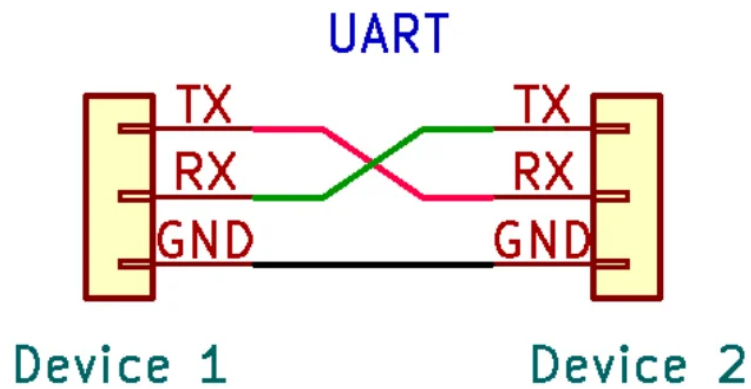


Fig 3.4.5.2 Interfacing to UART with Computer, autoria [34]

3.4.5.3 Justificación técnica:

1. UART es un protocolo de comunicación robusto, simple y eficiente para transmisión continua.
2. Es ideal para sistemas embebidos con recursos limitados.

3.4.6 MÓDULO MECÁNICO

El módulo mecánico constituye la base estructural del sistema, diseñado para reproducir temblores mediante oscilaciones en dos ejes ortogonales y permitir el ajuste de amplitud mediante la posición de la IMU. Para ello se adoptó una arquitectura tipo **gimbal**, seleccionada por su modularidad, estabilidad y fidelidad biomecánica.

Funciones principales:

- Emular físicamente temblores en ejes X y Y mediante movimientos oscilatorios controlados.
- Ajustar la amplitud vibratoria mediante el desplazamiento de la IMU.
- Brindar soporte rígido y confiable para componentes electrónicos y mecánicos.
- Facilitar integración con sensores, drivers y cableado.

3.4.6.1 Justificación de la arquitectura:

El gimbal permite desacoplar movimientos en ejes ortogonales, asegurando control independiente y alta resolución angular. Su uso es frecuente en robótica biomédica y dispositivos de rehabilitación por su capacidad para replicar trayectorias humanas con precisión. Frente a configuraciones planas o suspendidas, ofrece mayor fidelidad biomecánica, modularidad y compatibilidad con esquemas de control avanzados. La **Tabla 3.4.6.3** resume la comparación entre arquitecturas, evidenciando las ventajas del gimbal sobre alternativas convencionales.

Tabla 3.4.6.3 Comparación entre arquitecturas mecánicas para emulación de vibraciones

CRITERIO TÉCNICO	ARQUITECTURA TIPO GIMBAL	PLATAFORMA PLANA FIJA	ESTRUCTURA SUSPENDIDA
Desacoplamiento de ejes	Total (rotaciones independientes por eje)	Parcial o nulo	Generalmente acoplados
Fidelidad biomecánica	Alta: permite replicar trayectorias humanas complejas	Baja: limitada a movimientos en un solo plano	Media: buena amplitud pero sin control preciso
Modularidad y mantenimiento	Alta: cada subsistema es desmontable	Alta	Baja: requiere desmontaje completo
Compatibilidad con sistemas de control	Excelente: desacoplado, ideal para control por software	Limitada a control por vibradores o actuadores simples	Difícil de modelar y controlar con precisión
Estabilidad estructural	Alta: permite aislar vibraciones no deseadas	Alta, si está bien fijada	Media: susceptible a resonancias
Aplicación en sistemas biomédicos	Frecuente en robótica de rehabilitación, prótesis y simulación	Esporádica	A veces en simuladores físicos, pero con restricciones técnicas

3.5 DISEÑO DEL SISTEMA PARKIMOTION

3.5.1 DISEÑO MECÁNICO

3.5.1.1 Subsistemas mecánicos y componentes principales

El módulo mecánico del sistema ParkiMotion está compuesto por varios subsistemas que interactúan para generar oscilaciones controladas, emulando de manera precisa el temblor característico de la enfermedad de Parkinson. Cada uno de estos conjuntos cumple una función específica dentro de la arquitectura tipo gimbal, permitiendo controlar la dirección y la amplitud del movimiento. A continuación, se describen los principales subsistemas y sus componentes estructurales:

3.5.1.1.1 Subsistema de oscilación transversal

Este mecanismo permite el movimiento lateral de todo el conjunto interno, similar a un vaivén de izquierda a derecha. Es el primer subsistema en la cadena cinemática y aporta variaciones complementarias al patrón vibratorio general.

Componentes clave:

- Motor paso a paso NEMA 17 montado lateralmente.
- Eje vertical de rotación con acoplamiento mecánico.
- Soporte giratorio conectado a la estructura superior.

3.5.1.1.2 Subsistema de oscilación sagital

Este conjunto proporciona el movimiento hacia adelante y atrás de la plataforma de la IMU, equivalente a una oscilación tipo péndulo. Es el subsistema más representativo, ya que induce la señal dominante registrada en el eje Z de la IMU, constituyendo la referencia principal para el análisis de frecuencia.

Componentes clave:

- Motor paso a paso NEMA 17.
- Engranaje recto acoplado a eje de acero rígido.
- Soporte acrílico superior con rodamientos mecanizados.

3.5.1.1.3 Estructura soporte

La estructura principal proporciona rigidez y estabilidad a todo el sistema. Su diseño modular permite integrar fácilmente los motores, la electrónica y los elementos móviles, asegurando una base sólida para la generación de vibraciones controladas.

Componentes clave:

- Cuatro columnas verticales de perfil ranurado en aluminio.

- Placa base rígida y placa superior acrílica.
- Soportes intermedios para montaje de motores y mecanismos.

El diseño final del módulo mecánico fue el resultado de integrar los subsistemas previamente definidos en una estructura tridimensional coherente, funcional y modular. Esta configuración permitió representar con precisión la disposición espacial de cada componente, asegurar el acoplamiento adecuado entre piezas móviles y estáticas, y prever posibles interferencias físicas antes de la fabricación del prototipo.

Durante el proceso de modelado se priorizó la compatibilidad con componentes comerciales, la facilidad de ensamblaje y el equilibrio estructural. Cada elemento fue diseñado considerando tanto su función mecánica como su interacción con el sistema electrónico y sensorial, garantizando así la viabilidad funcional del conjunto.

Entre los aspectos clave del diseño destacan:

- Una **configuración tipo gimbal**, que permite el control desacoplado de los movimientos en los ejes X y Y mediante la orientación ortogonal de los motores paso a paso.
- Un **espacio reservado para el ajuste vertical de la IMU**, lo cual posibilita la modificación de la amplitud vibratoria mediante desplazamiento físico controlado.
- Una **estructura de soporte** construida con perfiles de aluminio y placas acrílicas, que proporciona rigidez, estabilidad y una plataforma segura para el montaje modular.
- Una **disposición central de la IMU**, que garantiza una captación precisa de las vibraciones generadas, sin influencias asimétricas.
- Un **diseño compacto y simétrico**, ideal para pruebas repetibles en entorno de laboratorio, y fácilmente replicable para futuras versiones del sistema.

Adicionalmente, la ubicación de la IMU en el centro de la plataforma asegura que las oscilaciones inducidas por los subsistemas mecánicos se proyecten principalmente sobre el eje Z del sensor. Esta condición consolidó al eje Z como la referencia principal para el análisis espectral y la estimación de frecuencia, aspecto clave para la validación experimental del sistema.

La **Fig. 3.5.1 muestra** el modelo CAD definitivo del sistema ParkiMotion, donde se aprecian claramente los subsistemas integrados y la disposición final adoptada para su fabricación.

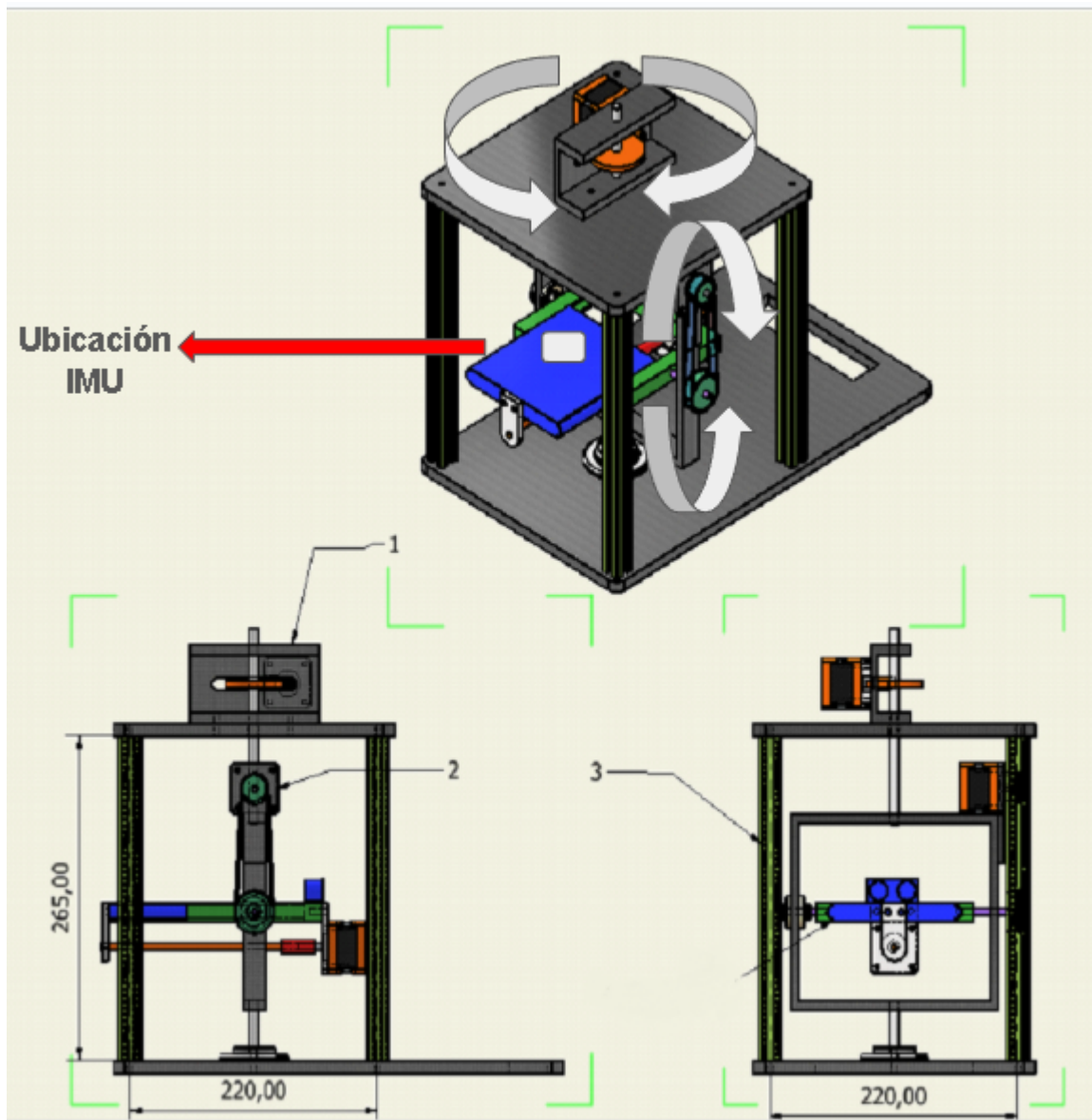


Fig 3.5.1. Modelo CAD final del módulo mecánico con arquitectura tipo gimbal adaptada y subsistema de ajuste de amplitud.

En la **Tabla 3.5.1.3**, se resumen los principales subsistemas que componen el módulo mecánico del sistema ParkiMotion. Cada uno de ellos cumple una función específica dentro de la arquitectura gimbal, contribuyendo de forma articulada a la generación y control del movimiento oscilatorio. La tabla presenta su numeración, nombre y descripción funcional:

Tabla 3.5.1.3 Identificación de conjuntos funcionales en el sistema mecánico

N°	Subsistema	Descripción
1	Subsistema de oscilación transversal	Permite el movimiento de vaivén lateral del marco completo.
2	Subsistema de oscilación sagital	Produce el movimiento hacia adelante y atrás de la plataforma. Es el eje más representativo, ya que genera la señal dominante registrada en el eje Z de la IMU y constituye la referencia principal para el análisis de frecuencia.
3	Estructura soporte	Conformada por perfiles de aluminio y placas acrílicas. Proporciona rigidez, modularidad y aislamiento estructural para reducir interferencias.

3.5.2 DISEÑO DE LA INTERFAZ

La interfaz gráfica de ParkiMotion fue diseñada para ofrecer una interacción clara e intuitiva con el sistema. Se organiza en tres secciones principales:

- **Panel superior de control:** incluye la selección de tipo de paciente (leve, moderado, severo), ajuste de frecuencia mediante slider, indicadores en tiempo real (frecuencia, amplitud, desplazamiento) y botones de inicio/pausa de adquisición.
- **Zona de visualización de señales:** muestra dinámicamente las aceleraciones registradas por la IMU en los ejes X, Y y Z, tras un preprocesamiento por filtrado digital.
- **Sección de análisis de frecuencia y amplitud:** contiene la gráfica de FFT del eje Z para identificar la frecuencia dominante y la gráfica de amplitud pico temporal para evaluar la intensidad vibratoria.

La **Fig. 3.5.2** resume esta distribución modular y jerárquica, que prioriza la usabilidad y la escalabilidad hacia futuras versiones del sistema.

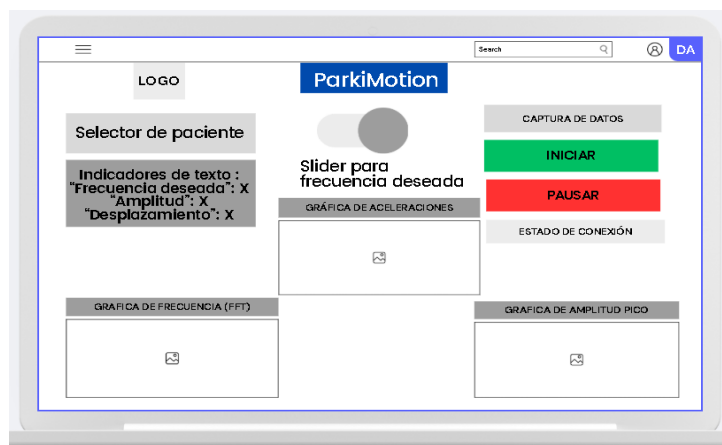


Fig 3.5.2. Definición estructural de la interfaz para Patkimotion, autoría propia.

3.5.2.1 Funciones principales

- Seleccionar el tipo de paciente: leve, moderado o severo.
- Ajustar y visualizar la frecuencia de emulación.
- Mostrar las señales inerciales (aceleraciones X, Y, Z) en tiempo real.
- Realizar análisis espectral mediante FFT sobre el eje de interés.
- Mostrar la evolución temporal de la amplitud pico.
- Iniciar y detener la captura de datos en CSV.
- Observar el estado del sistema en tiempo real (conectado/desconectado).

3.5.2.2 Componentes principales

- Framework Dash (Python) con estructura modular HTML y callbacks asincrónicos.
- Gráficos interactivos para visualización de aceleración, frecuencia y amplitud.
- Elementos de control: sliders, dropdowns, botones de acción, indicadores dinámicos.

3.5.2.3 Interfaces

- Comunicación UART bidireccional con el módulo de procesamiento (ESP32), permitiendo:
 - Envío de frecuencia deseada desde la interfaz.
 - Recepción continua de datos de aceleración y frecuencia detectada.
- Visualización integrada en tiempo real de las variables sensadas y los parámetros procesados.
- Exportación automática de resultados en archivos (.csv) durante la captura activa.

La representación detallada de cada componente visual y su función dentro del flujo de interacción con el usuario se aborda con mayor profundidad en el Capítulo 4.6, donde se expone la estructura completa de la interfaz, incluyendo ilustraciones representativas de cada módulo funcional.

3.5.3 MODELADO DE LA PLANTA Y DISEÑO DEL SISTEMA DE CONTROL

3.5.3.1 Modelado matemático de la planta

El sistema de emulación de temblores desarrollado presenta un comportamiento oscilatorio inducido por la acción de motores paso a paso sobre una estructura mecánica. Desde el punto de vista del modelado dinámico, este tipo de comportamiento puede representarse mediante un sistema **masa-resorte-amortiguador** clásico de segundo orden. Este modelo ha sido ampliamente utilizado para describir sistemas vibratorios en ingeniería de control y mecatrónica [32]. La ecuación diferencial que rige este sistema es:

$$m \frac{d^2 y(t)}{dt^2} + c \frac{dy(t)}{dt} + ky(t) = F(t)$$

Ecuación 3.5.3.1. Modelo masa-resorte-amortiguador.

Donde:

- $y(t)$: desplazamiento o aceleración de la estructura (variable de salida),
- m : masa equivalente de la estructura vibrante,
- c : coeficiente de amortiguamiento del sistema,
- k : constante de rigidez mecánica de la estructura,
- $F(t)$: fuerza externa aplicada por los motores (proporcional a la entrada de control).

Esta ecuación describe cómo la estructura responde ante una entrada oscilatoria. Se asume que el sistema es lineal y de parámetros constantes dentro del rango de operación, y que la vibración ocurre en un solo grado de libertad, bajo el eje de mayor sensibilidad del sensor.

Al aplicar la transformada de Laplace, se obtiene la función de transferencia de la planta:

$$G(s) = \frac{Y(s)}{F(s)} = \frac{1}{ms^2 + cs + k}$$

Ecuación 3.5.3.2. Función de transferencia de la planta.

Esta función representa la relación entre la entrada del sistema (fuerza aplicada por los motores) y la salida (respuesta vibratoria, medida en aceleración o desplazamiento). Su análisis permite determinar parámetros clave como la frecuencia natural, el amortiguamiento y la respuesta en frecuencia del sistema.

3.5.3.2 Identificación experimental del modelo

Dado que los valores exactos de los parámetros físicos (m , c y k) no son directamente medibles debido a la complejidad estructural de la planta vibratoria, se optó por una metodología de identificación experimental a partir de datos reales obtenidos en la plataforma. Esta estrategia permitió capturar el comportamiento dinámico efectivo del sistema sin necesidad de modelar detalladamente cada componente físico.

El procedimiento seguido fue el siguiente:

1. Se introdujo una señal escalonada en la frecuencia deseada enviada al sistema de actuación, representando un cambio repentino en las condiciones vibratorias para evaluar la respuesta transitoria del sistema.
2. La respuesta dinámica del sistema fue registrada utilizando el sensor inercial MPU6050, tomando como referencia principal el eje Z, por su correspondencia directa con la dirección dominante del movimiento vibratorio.

3. A partir de la señal de aceleración registrada, se estimó la frecuencia dominante del sistema mediante el análisis espectral con una Transformada Rápida de Fourier (FFT), aplicando una ventana de Hann para atenuar el efecto de fugas espectrales.
4. Se generaron las series temporales correspondientes a la entrada $u(t)$, asociada a la frecuencia deseada enviada al sistema, y a la salida $y(t)$, correspondiente a la frecuencia estimada a partir de los datos sensoriales.
5. Utilizando el comando (*tfest*) del *System Identification Toolbox* de MATLAB, se ajustó un modelo de segundo orden con un cero.

Como resultado, se obtuvo el siguiente modelo dinámico:

$$G(s) = \frac{918s + 137.8}{s^2 + 16.59s + 3122}$$

Ecuación 3.5.3.3. Identificación experimental del modelo.

Los parámetros identificados se resumen en la Tabla 4.2.2.1. Aunque la calidad del ajuste es moderada, se considera aceptable para los fines de control y emulación, dada la complejidad estructural y la presencia de perturbaciones mecánicas.

Tabla 3.5.3.1. Parámetros del modelo dinámico identificado para la planta vibratoria.

Parámetro	Valor estimado	Interpretación
Ganancia estática	0.044	Amplificación del sistema ante entradas
Cero real	$s = -0.15$	Introduce adelanto en la respuesta
Frecuencia natural W_n	8.89 Hz	Comportamiento oscilatorio inherente
Tipo de respuesta	Subamortiguada	Respuesta oscilante con amortiguamiento
Coefficiente de amortiguamiento	0.15	Forma de la respuesta; sobreimpulso
Porcentaje de ajuste (validación)	59.22 %	Nivel de coincidencia con datos reales

Aunque el modelo dinámico identificado presentó un ajuste del **59,22 %** frente a los datos experimentales de validación, este valor se considera adecuado dadas las particularidades de la planta. El sistema vibratorio posee elementos mecánicos acoplados y no lineales (soportes, holguras, fricción variable) que dificultan un modelado preciso. Además, el sensor MPU6050 introduce limitaciones de resolución y ruido que afectan la repetibilidad de las mediciones.

Se optó por un **modelo de segundo orden con un cero**, obtenido mediante *tfest* de MATLAB a partir de datos experimentales de barrido de frecuencias. Esta aproximación prioriza la

simplicidad computacional y permite una implementación eficiente en el ESP32, asegurando estabilidad en el rango clínico de interés (3–7 Hz).

La **Fig. 3.5.3.1** compara la salida del modelo con los datos experimentales, mostrando correspondencia aceptable en la dinámica general, lo que confirma su utilidad como base para el diseño del controlador PID.

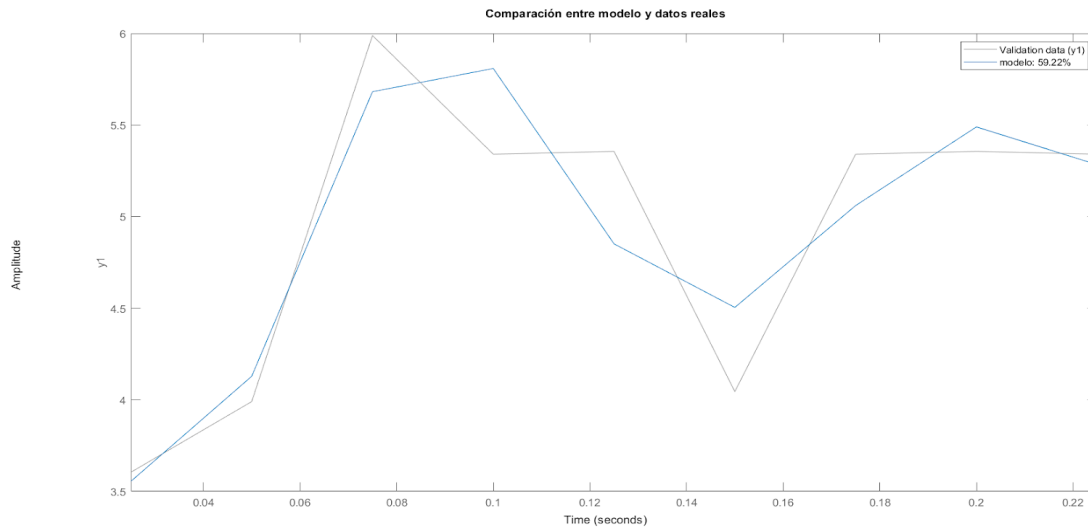


Fig 3.5.3.1. Comparación entre la respuesta del modelo identificado y los datos reales de validación.

3.5.3.3 Planteamiento del Problema de Control

3.5.3.3.1 Objetivo técnico del sistema

El objetivo de esta sección es diseñar un sistema de control que permita generar vibraciones mecánicas controladas con características similares a los temblores en personas con enfermedad de Parkinson. Esto se logrará mediante el ajuste dinámico de la señal de control enviada a motores paso a paso que generan la oscilación deseada en una plataforma experimental.

3.5.3.3.2 Descripción funcional del sistema

El sistema cuenta con:

- Dos motores paso a paso NEMA 17, controlados mediante drivers DRV8825 configurados en modo 1/32 de microstepping.
- Un ESP32, programado con FreeRTOS, encargado de generar la señal de pasos (STEP) con frecuencia variable.
- Una IMU MPU6050, que mide las aceleraciones inducidas por los motores en los ejes X, Y y Z.

- El objetivo de control es regular la frecuencia de vibración, a través de la modulación de la señal de entrada (frecuencia de pasos) en función del comportamiento medido.

3.5.3.3 Necesidad de control

Aunque es posible generar vibraciones utilizando señales de entrada fijas hacia los motores paso a paso, este enfoque no garantiza una respuesta precisa ni coherente con las características clínicas del temblor asociado a la enfermedad de Parkinson. La ausencia de regulación puede ocasionar comportamientos no deseados como:

- Aparición de resonancias mecánicas que alteran la respuesta del sistema.
- Generación de vibraciones no lineales o distorsionadas respecto a la señal deseada.
- Limitaciones para adaptar el estímulo vibratorio a diferentes perfiles vibración (leve, moderado o avanzado).

Ante estas limitaciones, se plantea la necesidad de implementar un sistema de control en lazo cerrado, que permita ajustar dinámicamente la señal de control en función de las condiciones reales del sistema. Este enfoque garantiza un comportamiento más estable, repetible y representativo de las frecuencias características del temblor, permitiendo además una mayor flexibilidad en pruebas clínicas o de laboratorio.

3.5.3.4 Diseño del controlador

3.5.3.4.1 Modelo de Control en Lazo Cerrado

Para garantizar la emulación precisa de temblores, el sistema implementa una arquitectura de control en lazo cerrado, basada en la estructura clásica de sistemas realimentados. En esta configuración, el objetivo es regular la vibración inducida por los motores paso a paso de forma que reproduzca un perfil similar al temblor patológico del Parkinson.

El funcionamiento del sistema puede representarse mediante un **diagrama de bloques típico**, en el que se identifican claramente los componentes fundamentales del lazo de control: la entrada o referencia, el comparador, el controlador (PID), la planta (estructura vibratoria), el sensor (IMU) y los efectos de perturbaciones y ruido. Cada uno de estos elementos desempeña un papel clave en la regulación del sistema y su análisis resulta fundamental para comprender el comportamiento dinámico general del prototipo desarrollado.

La **fig 3.5.4.1** representa la vulva típica del sistema.

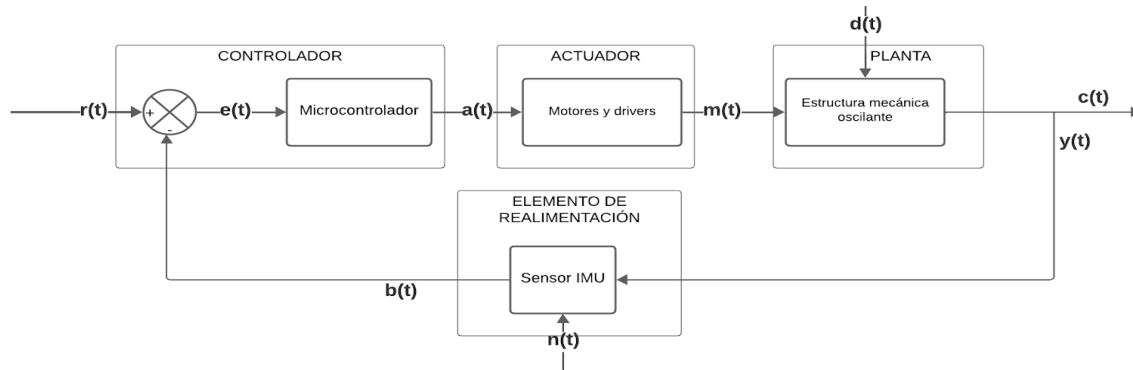


Fig 3.5.3.4.1 Bucla típica, autoría: propia.

- **Entrada o referencia $r(t)$:** Es la señal deseada de vibración que se aplica al sistema. Representa la etapa del temblor a emular (leve, moderada o severa), definida en términos de frecuencia.
- **Salida o variable controlada $c(t)$, $y(t)$:** Es la vibración mecánica real inducida por la plataforma experimental. Esta cantidad es medida y controlada, y representa el resultado del sistema ante la referencia y las perturbaciones.
- **Señal de realimentación $b(t)$:** Es la señal generada por el sensor IMU (MPU6050), que entrega una imagen de la salida controlada, adecuada en tipo y magnitud, para su comparación con la señal de referencia.
- **Error $e(t)$:** Representa la diferencia entre la señal de referencia $r(t)$ y la señal de realimentación $b(t)$. Este valor indica cuánto se desvía el sistema del comportamiento deseado y es utilizado como entrada del controlador.
- **Señal de control $a(t)$, $u(t)$:** Es la señal calculada por el controlador (microcontrolador). Esta señal se aplica a los actuadores para corregir el error en tiempo real.
- **Variable manipulada $m(t)$:** Es la acción de vibración mecánica inducida por los motores paso a paso a través de los drivers DRV8825. Esta acción modifica el comportamiento de la estructura oscilante para afectar el valor de la salida controlada.
- **Disturbio o perturbación $d(t)$:** Son señales externas que afectan de forma adversa el comportamiento de la planta, como fricción, resonancia o deformaciones estructurales que interfieren con la vibración deseada.
- **Ruido $n(t)$:** Es una señal indeseable introducida durante el proceso de medición del sensor. Afecta la realimentación, dificultando la precisión del sistema de control.

Para regular la frecuencia de vibración generada por la plataforma experimental y lograr que siga una referencia definida por el usuario, se diseñó un controlador digital de tipo PID (Proporcional–Integral–Derivativo). Este tipo de controlador fue seleccionado por su capacidad de mejorar el comportamiento dinámico del sistema, su implementación sencilla en sistemas

embebidos y su eficacia probada en sistemas de segundo orden como el modelado dinámico de la planta desarrollado en este trabajo.

3.5.3.4.2 Modelo matemático del controlador PID

En el dominio continuo, la expresión general del controlador PID es:

$$u(t) = K_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}$$

Ecuación 3.5.3.4.1. Representación en tiempo continuo de un controlador PID clásico.

Donde:

- $u(t)$ es la salida del controlador, que define la frecuencia de pasos hacia los motores.
- $e(t) = r(t) - y(t)$ es el error entre la frecuencia deseada y la frecuencia detectada.
- K_p, K_i, K_d son las constantes del controlador proporcional, integral y derivativo respectivamente.

3.5.3.4.3 Discretización de la planta y diseño del controlador PID

Para llevar a cabo la implementación práctica del controlador diseñado, se requiere expresar su forma continua en un equivalente discreto. Esta transformación es necesaria debido a que el sistema de control opera sobre una plataforma embebida basada en microcontrolador (ESP32), la cual trabaja en tiempo discreto. La discretización permite traducir la acción de control a un algoritmo que se ejecuta de forma secuencial con un intervalo de muestreo definido.

Para la discretización del controlador y planta en continuo, se seleccionó un tiempo de muestreo $T_s = 0,025s$ (25 milisegundos). Este valor fue elegido teniendo en cuenta dos criterios fundamentales:

- **Criterio de Nyquist aplicado al rango de frecuencias objetivo:** el sistema busca controlar vibraciones dentro del rango de 3–7 Hz, correspondiente a los temblores de la enfermedad de Parkinson. Para representar adecuadamente estas señales sin aliasing, el muestreo debe ser al menos el doble de la máxima frecuencia. Así,

$$f_s \geq 2 \times 7 = 14 \text{ Hz.}$$

Lo que implica:

$$T_s \leq 1/14 \approx 0,071s.$$

Se obtiene una frecuencia de muestreo de 40 Hz, casi seis veces superior a la frecuencia máxima de interés, lo que garantiza:

- Una captura suficientemente detallada de las variaciones de la señal.
- Un seguimiento más preciso de los transitorios en el lazo cerrado.
- **Respuesta dinámica de la planta:** el modelo obtenido presenta una dinámica relativamente rápida, por lo que un tiempo de muestreo más fino permite capturar mejor el comportamiento transitorio y mejora la estabilidad y precisión del lazo cerrado.

Con el objetivo de implementar el sistema de control en una plataforma embebida (ESP32), fue necesario transformar el modelo dinámico continuo identificado en su equivalente discreto. Esta transformación se realizó mediante el método de retención de orden cero (**ZOH**, *Zero-Order Hold*), una técnica ampliamente utilizada en control digital que asume que la señal de entrada permanece constante durante cada intervalo de muestreo.

Aplicando la discretización al modelo continuo de la ecuación (4.2.2.3), se obtuvo el modelo discreto $G_z(z)$:

$$G(z) = \frac{13.29z + 13.23}{z^2 + 0.306z + 0.6605}$$

Ecuación 3.5.3.4.4 Representación en tiempo discreto del controlador PID.

Este modelo discretizado representa la dinámica del sistema en el dominio de tiempo discreto, y fue utilizado como base para el diseño del controlador.

3.5.3.4.4 Diseño del controlador PID a partir de la planta discretizada

Una vez obtenido el modelo discreto de la planta $G_z(z)$, se procedió al diseño del controlador digital de tipo PID. Para ello, se utilizó inicialmente una herramienta de sintonización automática del entorno MATLAB, que permitió calcular una primera configuración del controlador directamente en el dominio discreto, con un tiempo de muestreo de $T_s = 0,025S$. La estructura general del controlador discreto generado fue:

$$C(z) = K_p + k_i \frac{T_s}{z-1} + k_d \frac{z-1}{T_s}$$

Ecuación 3.5.3.4.5. Estructura general del controlador PID en tiempo discreto.

Inicialmente, se empleó una herramienta de sintonización automática en MATLAB para obtener un conjunto preliminar de parámetros del controlador PID, los cuales ofrecieron un desempeño aceptable en lazo cerrado. Posteriormente, se realizó un ajuste manual fino mediante

simulaciones con entradas escalón, variando progresivamente las ganancias K_p, K_i, K_d hasta alcanzar un balance adecuado entre rapidez de respuesta, estabilidad y error en estado estacionario.

Como resultado, se definieron los parámetros finales del controlador:

$$K_p = 1,5, \quad K_i = 40, \quad K_d = 0,00851$$

Ecuación 3.5.3.4.6. Valores iniciales del controlador PID discreto obtenidos mediante sintonización automática.

A continuación, se presenta la respuesta del sistema en lazo cerrado ante una entrada escalón de referencia, utilizando los parámetros finales obtenidos tras el proceso de sintonización manual del controlador PID.

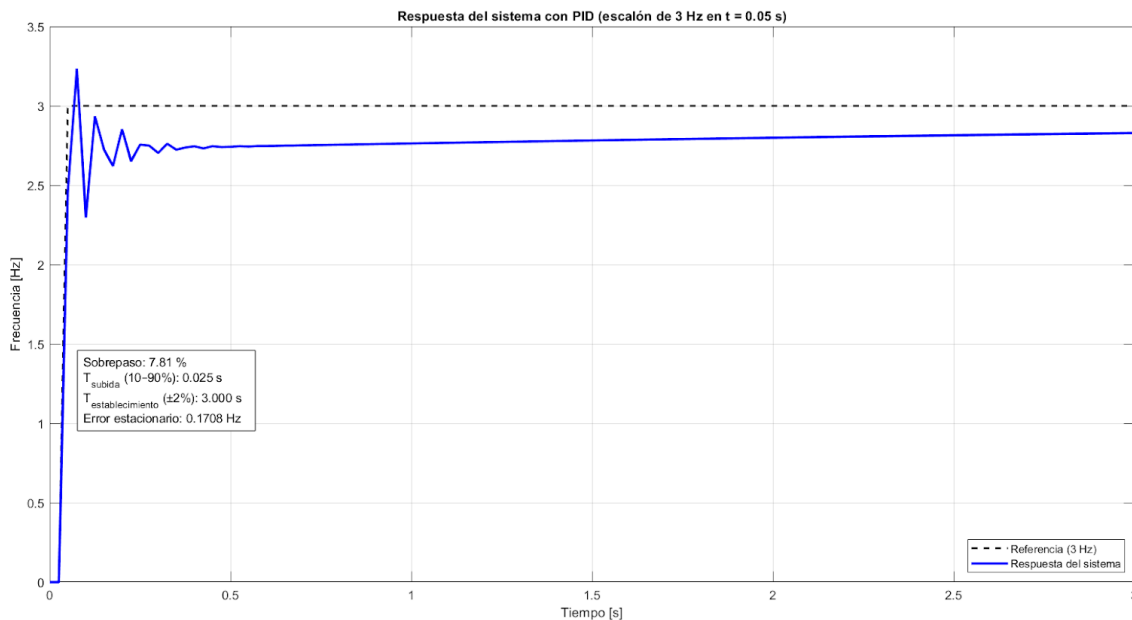


Fig 3.5.3.4.4 Respuesta del sistema en lazo cerrado con controlador PID sintonizado, ante una entrada escalón de 3 Hz aplicada en $t=0,05$

Como parte del análisis del comportamiento del sistema en lazo cerrado, se evaluaron los principales indicadores dinámicos ante una entrada escalón de frecuencia. Estos parámetros permiten cuantificar el desempeño del controlador en términos de estabilidad, rapidez de respuesta y precisión. **La Tabla 3.5.3.4.4** presenta los resultados obtenidos.

Tabla 3.5.3.4.4 Principales indicadores de desempeño dinámico ante una entrada escalón en lazo cerrado.

Indicador	Valor obtenido
Valor obtenido Frecuencia de referencia	3 Hz
Sobrepaso	7.81 %
Tiempo de subida (10–90 %)	0.025 s
Tiempo de establecimiento (± 2 %)	3.000 s
Error en estado estacionario	0.1708 Hz

Los resultados obtenidos con el controlador PID sintonizado presentados en la Tabla 3.5.3.4.4, donde se detallan los principales indicadores de desempeño dinámico ante una entrada escalón de 3 Hz. Como puede observarse, la respuesta obtenida presenta un sobrepaso moderado y un tiempo de subida breve, lo que indica una respuesta rápida sin comprometer la estabilidad del sistema. Si bien se evidencia un pequeño error en estado estacionario (0.1708 Hz), este se considera aceptable para la aplicación propuesta, dado que la precisión requerida en la emulación de temblores no demanda un seguimiento exacto milimétrico, sino la capacidad de replicar el comportamiento oscilatorio general dentro de un rango según los antecedentes representativo. Además, el sistema logra estabilizarse dentro de un margen de ± 2 % en un tiempo razonable, lo cual resulta funcional para una plataforma experimental embebida con recursos limitados.

3.5.4 DISEÑO DEL SISTEMA EMBEBIDO

El sistema embebido constituye el núcleo operativo del prototipo ParkiMotion, encargado de coordinar la adquisición de datos, el procesamiento de señales y la activación de los actuadores en tiempo real. Para ello, se emplea un microcontrolador ESP32, el cual ofrece capacidades de procesamiento paralelo, comunicación serial y control de periféricos con bajo consumo energético.

Con el objetivo de asegurar un funcionamiento estable y concurrente entre los diferentes procesos, el sistema fue diseñado bajo una arquitectura multitarea implementada mediante el sistema operativo en tiempo real FreeRTOS. Esta decisión permite separar funcionalmente las tareas críticas del sistema, asignando prioridades y recursos de forma eficiente.

A nivel conceptual, el diseño embebido se estructura en cinco bloques funcionales:

- **Adquisición de datos:** Captura continua de aceleraciones del sensor IMU MPU6050 mediante comunicación I2C.

- **Procesamiento de señales:** Estimación de frecuencia dominante y cálculo de amplitud pico a partir de los datos adquiridos.
- **Control del sistema:** Comparación entre la frecuencia deseada y la detectada, cálculo del error y generación de la señal de control mediante un controlador PID.
- **Actuación:** Traducción de la señal de control en pasos para los motores, generando así la vibración deseada.
- **Comunicación UART:** Canal bidireccional para recibir comandos desde la interfaz gráfica y enviar resultados de análisis en tiempo real.

La **Fig 3.5.4.1** muestra el diagrama conceptual de esta arquitectura, en el que se pueden identificar los bloques funcionales principales del sistema embebido.

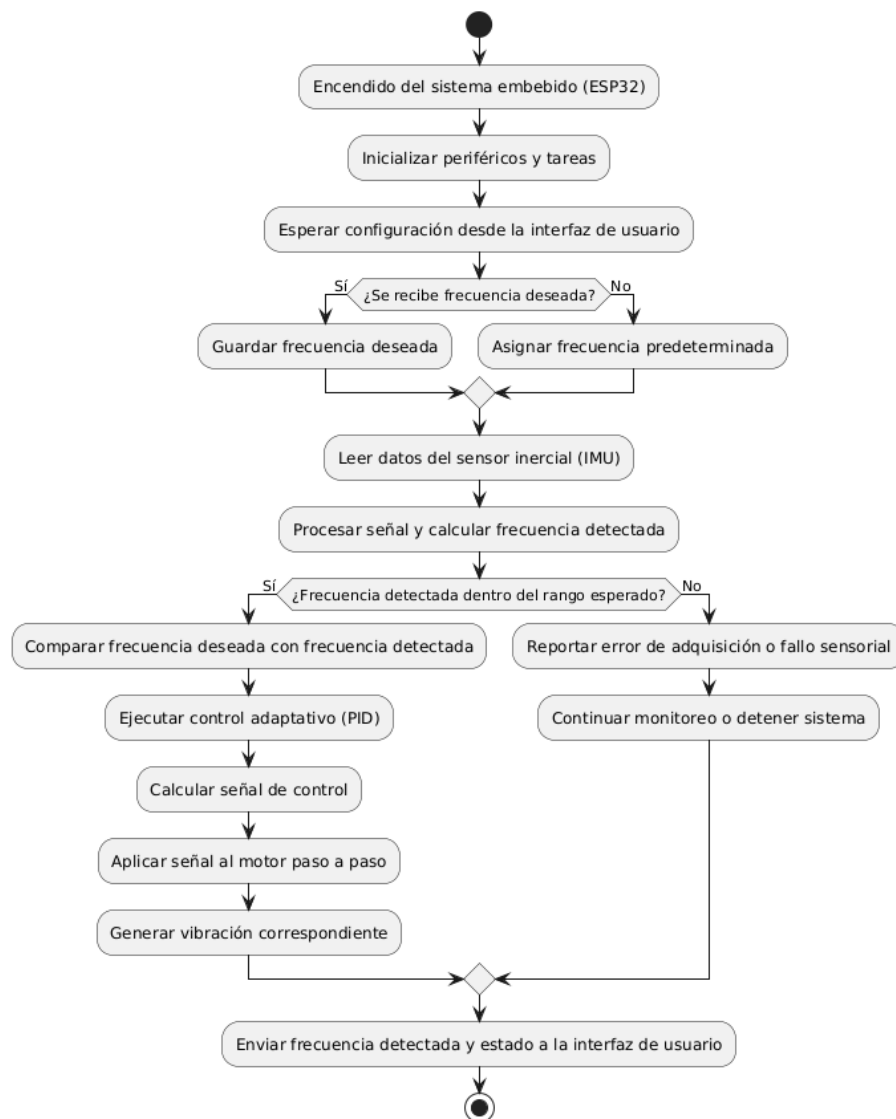


Fig 3.5.4.1. Diagrama conceptual del sistema embebido.

3.5.4.1 Flujo funcional de tareas

El sistema embebido fue diseñado con una arquitectura multitarea en **FreeRTOS**, donde cada bloque funcional se ejecuta como una tarea independiente con prioridad definida.

- **Adquisición de datos:** lectura continua de aceleraciones (X, Y, Z) desde el sensor MPU6050 vía I2C.
- **Procesamiento:** aplicación de filtrado digital pasa banda (3–7 Hz) para obtener una señal limpia representativa del movimiento vibratorio.
- **Control y actuación:** recepción de la frecuencia deseada desde la interfaz gráfica, cálculo de la señal de control y generación de pasos PWM para los motores paso a paso mediante drivers DRV8825.
- **Comunicación UART:** enlace bidireccional con la interfaz, recibiendo la frecuencia deseada y transmitiendo los datos sensados filtrados en tiempo real.

El intercambio entre tareas se implementó mediante colas, asegurando sincronización eficiente sin pérdida de datos. La **Fig. 3.5.4.1** muestra la organización modular de estas tareas dentro del ESP32.

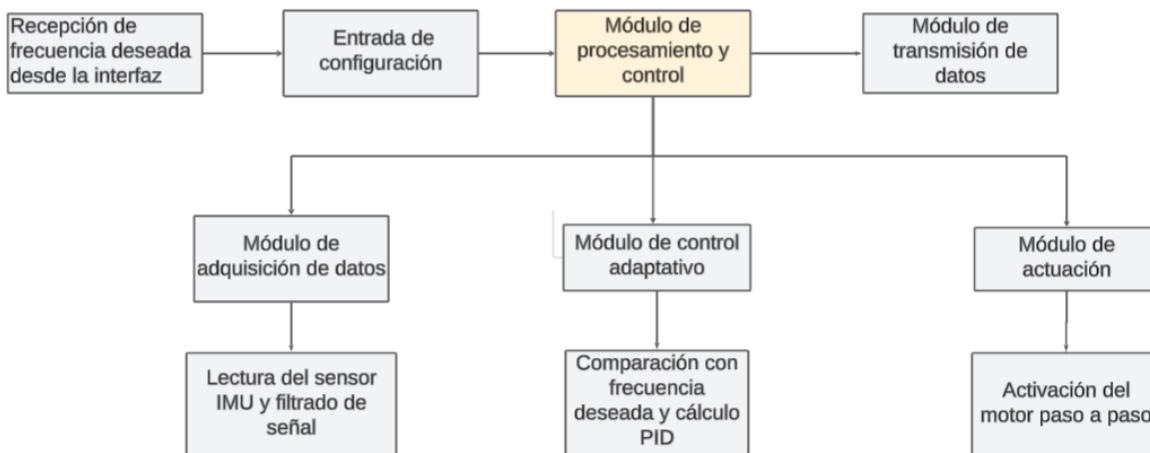


Fig 3.5.4.1. Relación entre submódulos del módulo de procesamiento y control.

3.5.4.2 Diseño conceptual del controlador

El controlador utilizado en el sistema embebido de ParkiMotion corresponde a una estructura PID (Proporcional–Integral–Derivativo), cuyo diseño y ajuste fueron desarrollados previamente en la etapa de diseño del sistema de control. Los parámetros K_p , K_i y K_d fueron determinados mediante simulaciones en MATLAB, considerando la dinámica de la planta vibratoria y los objetivos de desempeño definidos para cada tipo de temblor a emular.

En esta sección no se rediseña el controlador, sino que se implementa el modelo obtenido directamente dentro de la arquitectura multitarea de FreeRTOS. El algoritmo PID fue codificado como parte de la tarea de control, ejecutándose a una frecuencia constante, y tomando como entrada la diferencia entre la frecuencia deseada (recibida desde la interfaz gráfica) y la frecuencia detectada (calculada por la propia GUI). El resultado del PID es convertido en una frecuencia de paso para los motores paso a paso, generando así una vibración proporcional al error.

El control se realiza en tiempo real, utilizando aritmética de punto flotante y evitando retardos excesivos para no afectar la estabilidad del sistema. La salida del controlador se traduce directamente en señales de paso que activan los drivers DRV8825, responsables de excitar la estructura mecánica.

3.5.5 DISEÑO ELECTRÓNICO

El sistema fue integrado en una placa de circuito impreso (PCB) diseñada en EasyEDA, con el fin de organizar eficientemente los componentes electrónicos y facilitar futuras modificaciones. El diseño se basó en criterios de zonificación, agrupando cada función en áreas específicas para simplificar conexiones y depuración.

Las principales zonas son:

- **Control:** microcontrolador ESP32 para gestión de sensores, motores y comunicación UART.
- **Drivers:** dos módulos DRV8825 para el control de los motores paso a paso.
- **Alimentación:** conectores de voltaje y pads para un regulador LM2596 (no implementado en esta versión).
- **Sensores:** conexión de la IMU MPU6050 por I2C y pads para un sensor ultrasónico HC-SR04 (futuras versiones).
- **Indicadores:** LEDs con resistencias limitadoras para estado del sistema y operación de motores.

3.5.5.1 Esquemático eléctrico del sistema

El esquemático eléctrico del prototipo, elaborado en EasyEDA, organiza de forma lógica los componentes del sistema de control de vibraciones (Fig. 3.5.5.1).

- Control: microcontrolador ESP32 DevKit, núcleo del sistema embebido.
- Actuación: dos drivers DRV8825 conectados a los motores paso a paso, con señales DIR y STEP para control de frecuencia y dirección.
- Sensado: sensor MPU6050, enlazado al ESP32 mediante bus I2C y alimentado a 3.3 V.
- Alimentación: fuente externa de 12 V distribuida a motores y drivers, con pads reservados para un regulador LM2596 y conexión futura de un sensor ultrasónico HC-SR04.
- Indicadores: LEDs con resistencias limitadoras para visualizar estados de funcionamiento y control.

El diseño considera tanto la implementación actual como posibles expansiones, garantizando modularidad y facilidad de mejora.

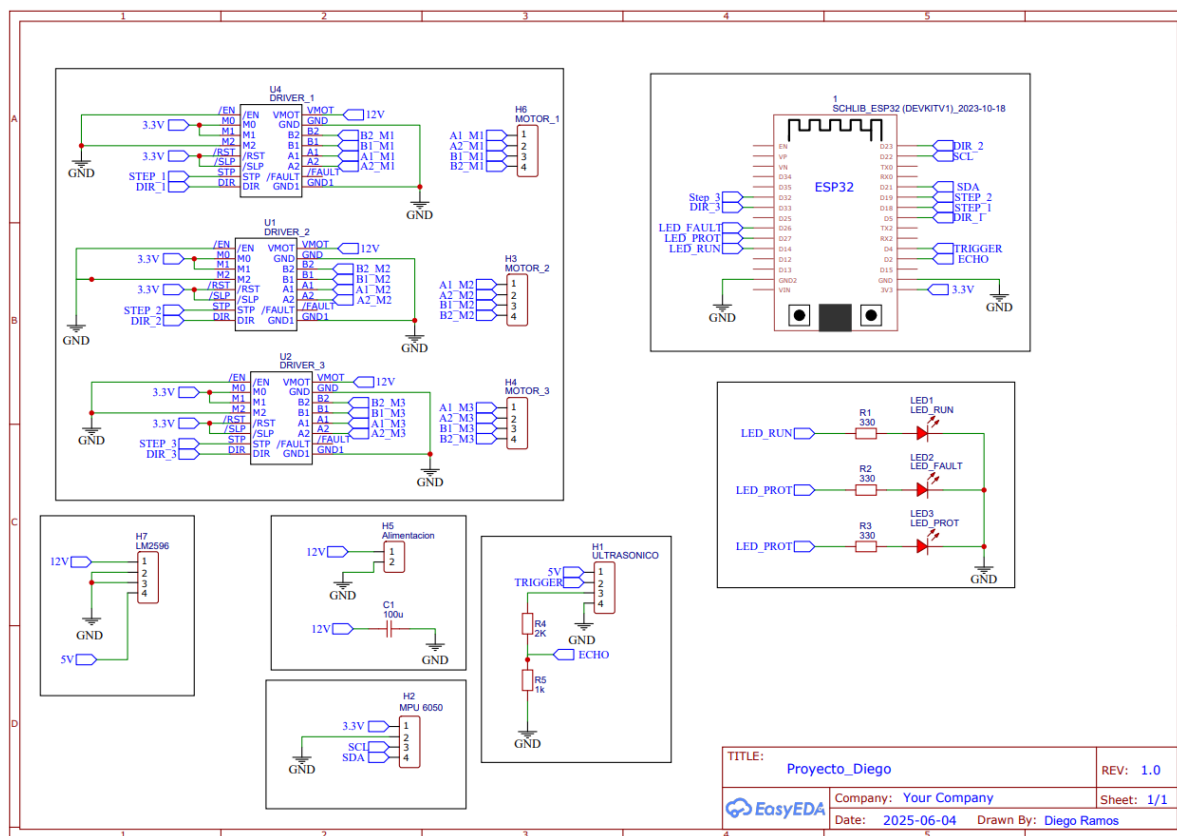


Fig 3.5.5.1. Esquemático eléctrico del sistema desarrollado en EasyEDA. Autoría: propia.

3.5.5.2 Diseño de la PCB y distribución de componentes

A partir del esquemático eléctrico, se diseñó la PCB en EasyEDA con el fin de integrar todos los módulos en una tarjeta compacta y organizada. Se optimizó el uso del espacio, se reforzaron las pistas de potencia para soportar la corriente de los motores y se mantuvo un diseño modular que facilita diagnóstico y mantenimiento.

La Fig. 3.5.5.2 muestra la distribución 2D de los componentes y la Fig. 3.5.5.3 la vista 3D del diseño final, utilizada para validar la disposición física y anticipar interferencias antes de la fabricación.

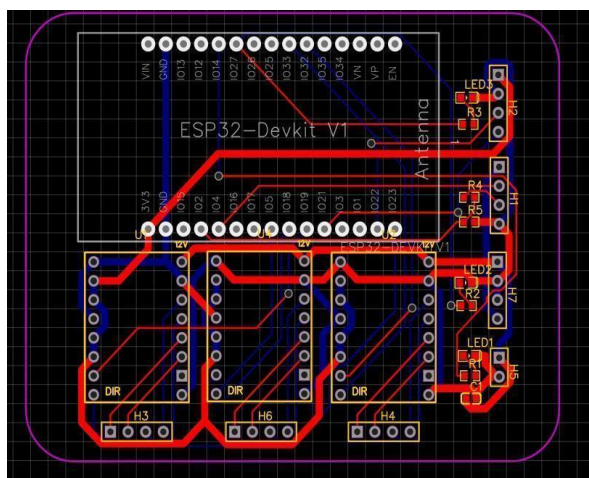


Fig 3.5.5.2. Diseño 2D de la PCB con distribución de componentes. Autoría: propia.

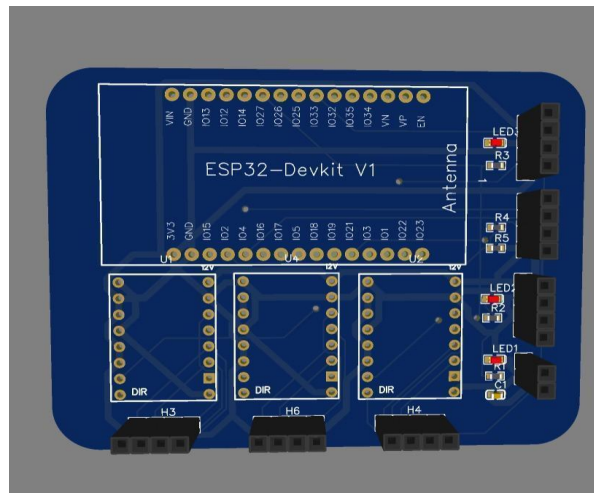


Fig 3.5.5.3. Vista 3D del diseño final de la PCB generada en EasyEDA. Autoría: propia.

3.5.5.3 Consideraciones sobre componentes no implementados

El diseño de la PCB incluye ciertas conexiones y footprints que no fueron utilizados en la versión final del prototipo, como el módulo reductor de voltaje LM2596, un sensor ultrasónico HC-SR04 y un tercer driver DRV8825. Estos elementos se incorporaron como parte de una planificación para futuras expansiones del sistema, orientadas al control adaptativo de amplitud en función de la posición del sensor. Aunque no se implementaron en esta etapa, su inclusión facilita futuras mejoras sin necesidad de rediseñar la placa.

Capítulo 4

PROTOTIPO EXPERIMENTAL DE PARKYMOTION.

4.0 INTRODUCCIÓN

Este capítulo documenta la implementación integral del sistema ParkiMotion, desde la estructura mecánica tipo gimbal hasta la validación del sistema integrado. Se presentan inicialmente la arquitectura mecánica y los ensambles, seguidos de las pruebas de actuación, sensado y comunicación realizadas de manera independiente. Posteriormente se aborda el desarrollo del sistema de control embebido en ESP32 con FreeRTOS, la interfaz gráfica en Dash y, finalmente, la validación experimental del prototipo bajo condiciones controladas.

4.1 IMPLEMENTACIÓN DEL SISTEMA MECÁNICO

4.1.1 Base del sistema

La base constituye el soporte inferior del sistema y es el punto de anclaje sobre el cual se monta toda la estructura mecánica. Su diseño rígido y plano garantiza una estabilidad estructural adecuada, resistencia frente a vibraciones externas y un plano de referencia confiable para el ensamblaje de los ejes superiores.

Además, incluye orificios estratégicamente distribuidos para facilitar su fijación sobre superficies de prueba, mesas de laboratorio o plataformas antivibración. Su construcción en material de alta rigidez (por ejemplo, acrílico o polímero técnico) permite también una distribución uniforme de la carga y estabilidad durante el funcionamiento del sistema.

- **Características clave**
 - Superficie de apoyo rígida
 - Orificios de montaje para fijación segura al banco de pruebas
 - Patas niveladores en las esquinas

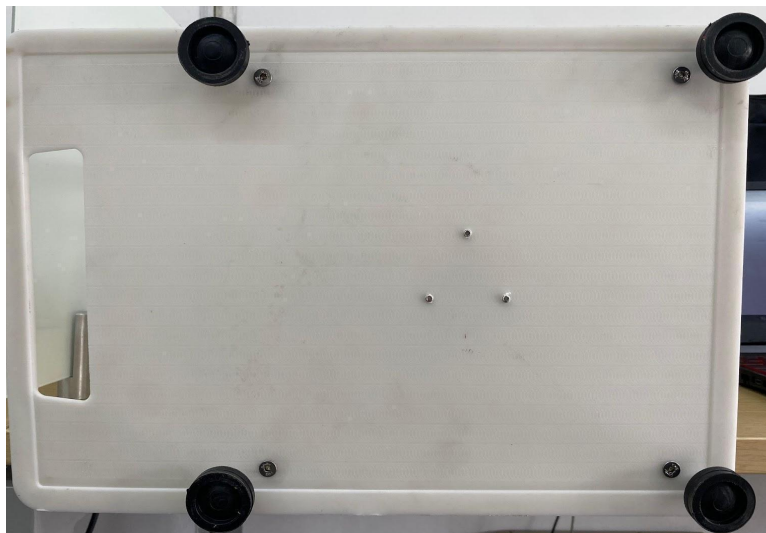


Fig 4.1.1 Vista inferior de la base del sistema. autoría propia

4.1.2 Subsistema de oscilación sagital

Este subsistema permite el movimiento de vaivén hacia adelante y atrás de la plataforma (tipo péndulo). Es el más representativo, ya que concentra la energía vibratoria dominante y genera la señal principal detectada en el eje Z de la IMU. De este modo, constituye la referencia central para el análisis de frecuencia y amplitud.

Componentes principales:

- Motor paso a paso NEMA 17.
- Soporte giratorio montado sobre la placa superior.
- Acoplamiento mecánico al eje vertical de giro.

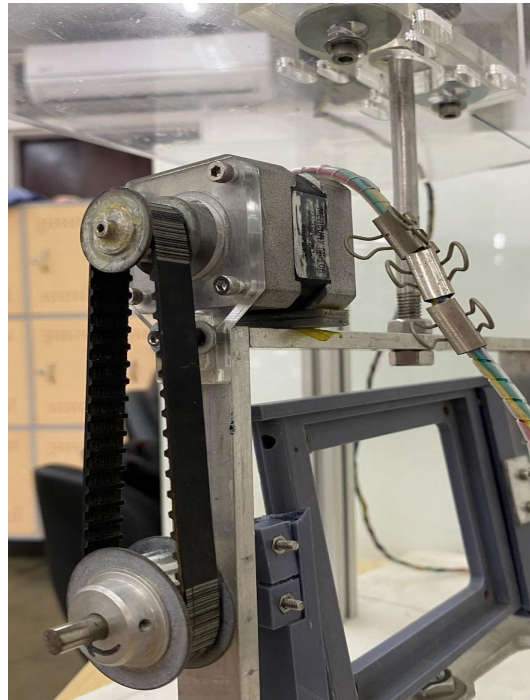


Fig 4.1.2 Sistema de rotación en el eje Y, autoría: propia.

4.1.3 Subsistema de oscilación transversal

Montado sobre la estructura giratoria, este subsistema produce oscilaciones laterales (izquierda-derecha), complementando el patrón vibratorio general. Para lograrlo, emplea una transmisión indirecta por engranajes acoplados a un motor paso a paso, que transmiten el movimiento a un eje rígido conectado a la plataforma de la IMU.

Componentes principales:

- Motor paso a paso NEMA 17.
- Engranaje recto y eje de acero rígido.
- Soporte superior de acrílico con rodamientos mecanizados.
- Acoplamiento inferior con la plataforma móvil de la IMU.

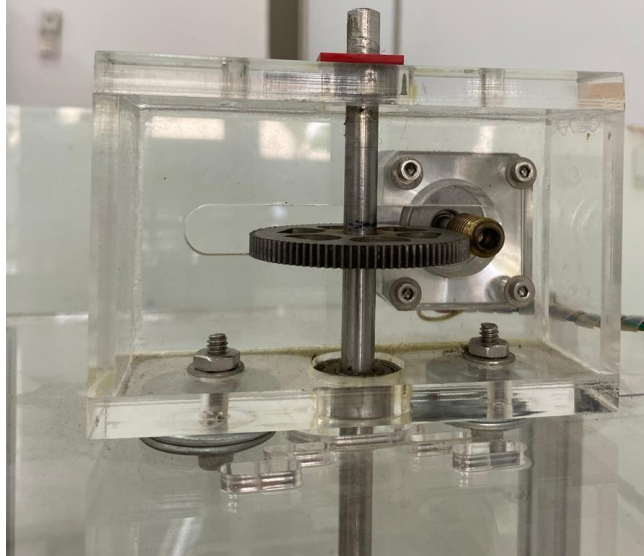


Fig 4.1.3 Sistema de rotación en el eje X, autoría: propia.

4.1.4 Marco estructural principal y diseño final

El marco estructural representa la columna vertebral del sistema ParkiMotion, proporcionando soporte, alineación y rigidez a todos los subsistemas mecánicos y electrónicos integrados. Su función es garantizar una base estable y modular sobre la cual se ensamblan los ejes de rotación, el sistema de transmisión, los motores paso a paso y la unidad de medición inercial (IMU).

El diseño definitivo, mostrado en las figuras de esta sección, está compuesto por una estructura tipo pórtico, conformada por columnas verticales de aluminio ranurado, placas acrílicas superior e inferior, y elementos de transmisión mecánica de precisión. Esta configuración permite una integración ordenada de todos los elementos funcionales, optimizando el espacio y reduciendo posibles interferencias mecánicas.

- **Componentes principales observados:**
 - 4 columnas de perfil ranurado en aluminio.
 - Placa superior de acrílico con soporte para motor del subsistema sagital.
 - Placa base rígida con fijación del eje central.
 - Varilla roscada central y acoplamientos mecánicos.
 - Polea y banda dentada para transmisión.
 - Soportes metálicos transversales para el subsistema transversal.

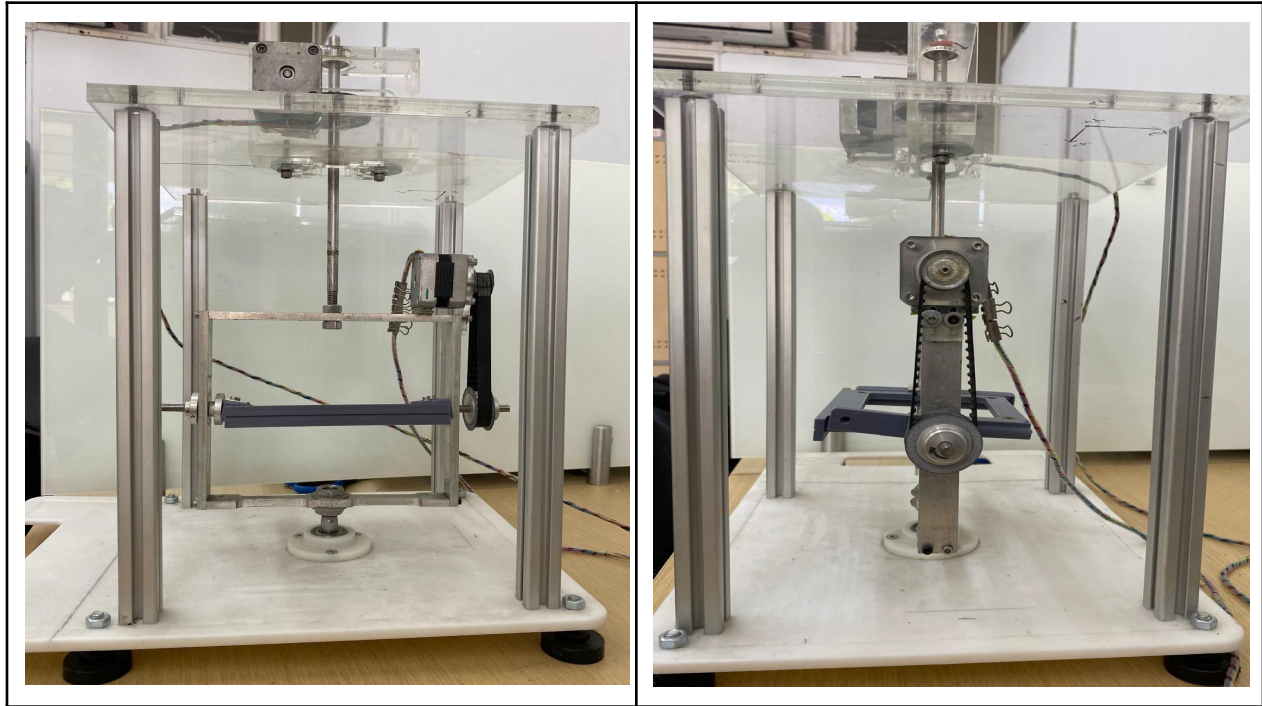


Fig 4.1.4. Vista frontal y lateral del marco estructural principal, autoría: propia.

4.2 IMPLEMENTACIÓN DEL SISTEMA EMBEBIDO

Una vez definida la arquitectura conceptual, se implementó el firmware en el microcontrolador ESP32 bajo **FreeRTOS**, adoptando una arquitectura multitarea. Cada bloque funcional se programó como una tarea independiente con prioridades definidas, garantizando concurrencia, sincronización y respuesta en tiempo real.

La **Tabla 4.2.2.1** resume las tareas principales implementadas:

Tabla 4.2.2.1. Resumen de tareas del sistema embebido en ESP32 bajo FreeRTOS

Nombre de la tarea	Descripción funcional	Frecuencia de ejecución
<i>imu_reader_task</i>	Adquiere datos de aceleración desde la IMU y aplica filtrado digital.	40 Hz (cada 25 ms)
<i>motor_step_task</i>	Genera los pulsos necesarios para accionar los motores paso a paso.	Variable según señal de control
<i>motor_pid_control_task</i>	Calcula la señal de control PID basada en el error de frecuencia.	40 Hz (cada 25 ms)
<i>uart_receive_task</i>	Recibe los comandos de frecuencia deseada desde la interfaz UART.	Revisión continua
<i>freq_updater_task</i>	Actualiza el valor de frecuencia deseada en el sistema de control.	Revisión continua

4.2.1 Pseudocódigo general del módulo de procesamiento

Con base en la arquitectura conceptual del sistema embebido presentada anteriormente, se establece a continuación la estructura general del módulo de procesamiento y control. El siguiente pseudocódigo resume el flujo de ejecución principal del sistema embebido, destacando la inicialización de periféricos, la creación de tareas concurrentes, y la estructura jerárquica del programa implementado sobre el microcontrolador ESP32.

Algoritmo 4.2.1. Flujo de ejecución principal.

- 1 Inclusión de librerías;
- 2 Declaración de variables globales y configuración de FreeRTOS;
- 3 Inicialización de sensores (IMU) y comunicación UART;
- 4 Configuración del módulo de control (PID y parámetros);
- 5 Tarea principal:
 - 6 Inicializar módulos físicos (IMU, motor, UART);
 - 7 Crear tarea de adquisición de datos IMU;
 - 8 Crear tarea de recepción de frecuencia desde la interfaz (UART RX);
 - 9 Crear tarea de control PID;
 - 10 Crear tarea de activación del motor paso a paso;
 - 11 Crear tarea de transmisión UART hacia la interfaz;
 - 12 Iniciar el planificador de tareas;
 - 13 Retorna;
- 14 Fin Tarea principal;

Algoritmo 4.2.1. Pseudocódigo flujo de ejecución principal, autoría: propia.

Una vez finalizada la configuración e inicialización mostrada en el algoritmo anterior, el control del sistema pasa al planificador de FreeRTOS. A partir de este momento, cada tarea definida comienza a ejecutarse de forma concurrente y autónoma, según la prioridad asignada y la

disponibilidad de recursos del sistema. Esta ejecución paralela es gestionada íntegramente por el sistema operativo en tiempo real, lo cual permite una operación estable y escalable.

4.2.2. Adquisición de datos

La tarea de adquisición de datos se encarga de obtener en tiempo real los valores de aceleración provenientes del sensor inercial MPU6050, utilizando el bus I2C. Una vez adquiridas las señales crudas, estas se convierten a unidades físicas (m/s²) aplicando la relación mostrada en la **Ecuación 4.2.2**.

$$acc_{xyz} = \left(\frac{acc_{xyz}}{16384.0} \right) * 9.81 \left[\frac{m}{s^2} \right]$$

Ecuación 4.2.2. Conversión de aceleración a variable física para filtro complementario.

Posteriormente, las aceleraciones se filtran con un Butterworth pasa banda (3–7 Hz) para reducir ruido y aislar el contenido útil. Los datos filtrados se envían vía UART a la interfaz gráfica, donde se estiman la frecuencia dominante y la amplitud del temblor. Esta tarea, ejecutada de forma continua en FreeRTOS a 40 Hz (25 ms), asegura la calidad de las señales procesadas.

Algoritmo 4.2.2. Tarea de adquisición de datos desde la IMU

```

1  Inclusión de librerías;
2  Definición del flujo de información (IMU → UART);
3  Declaración de variables globales y estructuras de filtrado;
4  Tarea principal:
5      Inicializar bus I2C;
6      Configurar sensor MPU6050;
7  Fin Tarea principal;
8  Ciclo de tareas:
9      Llamar lectura de aceleraciones;
10     Aplicar filtrado digital;
11     Transmitir datos filtrados por UART;
12 Fin ciclo de tareas
13 Lectura de aceleraciones:
14     Leer ejes(x,y,z)
15     Escalar valores a unidades físicas (g);
16 Fin lectura de aceleraciones
17 Filtrado digital:
18     Aplicar filtro Butterworth pasa banda;
19 Fin filtrado digital;
20 Transmisión de datos:
21     Enviar datos filtrados a la interfaz vía UART;
22 Fin transmisión de datos
    
```

Algoritmo 4.2.2. Pseudocódigo del módulo de adquisición de datos desde la IMU, autoría: propia.

4.2.3. Aplicación del PID discreto

El sistema de control emplea un PID discreto que ajusta la señal de salida según la diferencia entre la frecuencia deseada (definida en la interfaz) y una referencia interna en el sistema

embebido. Con los parámetros previamente sintonizados, el controlador garantiza una respuesta estable en lazo cerrado. La señal de mando resultante genera los pulsos de los motores paso a paso, cuya frecuencia y dirección se adaptan dinámicamente para reproducir los patrones vibratorios. Ambas tareas se ejecutan de forma concurrente en FreeRTOS, asegurando un desempeño coherente y adaptativo.

Algoritmo 4.2.3. Control PID discreto y activación de actuadores

```

1  Inclusión de librerías y definición de pines;
2  Declaración de parámetros PID (Kp, Ki, Kd, Ts);
3  Inicialización de variables del controlador (errores, salidas);
4  Tarea principal:
5      Configurar pines de control (STEP, DIR, ENABLE);
6      Inicializar motor y sentido de giro;
7  Fin Tarea principal;
8  Tarea de control PID:
9      Leer frecuencia deseada desde interfaz;
10     Leer frecuencia estimada del sistema;
11     Calcular error entre referencia y real;
12     Calcular incremento PID discreto ;
13     Actualizar salida de control
14     Limitar salida de control entre 3 Hz y 7 Hz;
15     Esperar intervalo de muestreo Ts;
16 Fin tarea de control
17 Tarea de activación del motor:
18     Calcular frecuencia de pasos;
19     Determinar periodo de pulsos;
20     Alternar señal STEP a los dos motores;
21     Contar micropasos y cambiar dirección al completar ciclo;
22     Esperar según periodo calculado;
```

Algoritmo 4.2.3. Pseudocódigo del módulo de control PID discreto, autoría propia.

4.2.4. Transmisión UART

Esta etapa permite la interacción entre el sistema embebido y la interfaz externa, a través de un canal de comunicación serial. Su objetivo principal es recibir, desde una interfaz gráfica, el valor deseado de frecuencia para ajustar dinámicamente la respuesta del sistema de control.

Para ello, se establece un mecanismo que escucha constantemente la llegada de datos provenientes del exterior. Una vez que se detecta un valor válido dentro del rango establecido, este se transfiere de forma segura al sistema de control, donde es interpretado como el nuevo valor de referencia que debe alcanzarse.

El sistema también cuenta con una estructura de comunicación interna que facilita el traspaso ordenado de los datos recibidos. Esto permite una actualización fluida y sin interferencias en el comportamiento del sistema, asegurando una adecuada sincronización con el lazo de control. La transmisión se realiza de forma continua durante toda la operación del dispositivo, habilitando una interacción constante y en tiempo real con el usuario final.

Algoritmo 4.2.4. Comunicación con la interfaz gráfica por canal serial

- 1 Inclusión de librerías y configuración inicial;
 - 2 Declaración de variables internas para datos recibidos;
 - 3 Tarea principal:
 - 4 Configurar parámetros del canal serial (velocidad, bits, pines);
 - 5 Activar el canal de comunicación y dejarlo listo para lectura;
 - 6 Fin Tarea principal;
 - 7 Tarea de recepción de datos:
 - 8 Convertir mensaje en valor numérico;
 - 9 Verificar que el valor esté dentro del rango válido (3 Hz a 7 Hz);
 - 10 Esperar intervalo breve antes de la siguiente lectura;
 - 11 Fin Tarea de recepción
 - 12 Tarea de actualización del valor de referencia:
 - 13 Revisar si hay un nuevo dato recibido para actualizar;
 - 14 Fin Tarea de actualización
-

Algoritmo 4.2.4. Pseudocódigo del módulo de control PID discreto, autoría propia.

Con la implementación completa del módulo de procesamiento y control se consolida la lógica operativa central del sistema embebido. A través de una arquitectura multitarea y modular, el sistema permite adquirir, procesar, controlar y comunicar información en tiempo real, garantizando una respuesta coherente ante las variaciones configuradas por el usuario. Esta sección ha detallado cada uno de los bloques funcionales que intervienen en este proceso, desde la captura sensorial hasta la generación del estímulo vibratorio. La correcta interacción entre estos componentes asegura el cumplimiento de los objetivos funcionales del prototipo y sienta las bases para su validación experimental.

4.3 IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE ACTUACIÓN

El módulo de actuación, descrito previamente en la sección 3.4.4. está compuesto por dos motores paso a paso NEMA 17, controlados mediante drivers DRV8825 configurados en modo microstepping 1/32. Esta configuración permite generar movimientos rotacionales suaves y precisos, necesarios para simular el patrón oscilatorio característico de los temblores asociados al Parkinson.

Durante esta fase, se buscó validar el funcionamiento de los motores implementados en el diseño mecánico del sistema, comprobando su capacidad para inducir oscilaciones estables en la plataforma física sin necesidad de un controlador activo. Para ello, se desarrolló un firmware básico sobre el microcontrolador ESP32, ejecutado bajo FreeRTOS, el cual genera señales de pulsos STEP con frecuencias fijas entre 3 Hz y 7 Hz. Estos valores se seleccionaron con base en rangos típicos reportados en literatura especializada para temblores leves y severos.

La respuesta del sistema fue monitoreada con un sensor inercial MPU6050 ubicado en la plataforma oscilante. Las señales de aceleración registradas permitieron verificar que las vibraciones inducidas eran periódicas y consistentes, cumpliendo así con el propósito de prueba

funcional. Las **Figs 4.4.1, 4.4.3 y 4.4.5** ilustran los resultados obtenidos, evidenciando la efectividad del módulo de actuación.

A continuación, se resumen los parámetros y resultados clave de la prueba realizada:

Tabla 4.3 Prueba módulo de actuación, autoría: propia.

Módulo de actuación				
Tipo de prueba	Integración funcional			
Hardware requerido	ESP32 DevKit V1, DRV8825, motores NEMA 17, fuente 12 V, protoboard, jumpers, IMU MPU6050			
software requerido	FreeRTOS, firmware de vibración, Visual Studio Code con ESP-IDF			
Actor	Usuario			
Objetivo	Validar el funcionamiento de los motores paso a paso NEMA 17 en configuración de vibración controlada mediante señales STEP			
Procedimiento de la prueba	Entrada	Salida esperada	Salida obtenida	Evaluación
Actividades	Señal de pulsos STEP generada desde el ESP32 con una frecuencia de 3 Hz a 7 Hz, configurada para realizar 20 micropasos por ciclo con la configuración de microstepping 1/32 en el driver DRV8825.	Movimiento vibratorio estable y simétrico	Aceleración periódica registrada por la IMU, coherente con la frecuencia programada (ver Fig 3.2)	Éxito

4.4 IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE SENSADO

El módulo de sensado del sistema ParkiMotion emplea una unidad inercial MPU6050 para la medición de aceleraciones lineales en los ejes X, Y y Z. Este sensor se comunica con el microcontrolador ESP32 mediante el protocolo I2C, permitiendo la adquisición continua de datos en tiempo real. Para validar su funcionamiento, se desarrolló una rutina en FreeRTOS encargada de realizar lecturas periódicas, con una frecuencia de muestreo de 40 Hz, enfocada en registrar el comportamiento dinámico inducido por los motores paso a paso.

Durante los ensayos iniciales se identificó que el eje Z, perpendicular al plano principal de la estructura, presenta la mayor respuesta vibratoria, especialmente como resultado del movimiento rotacional inducido por el motor que opera en el eje Y. En consecuencia, tanto el análisis espectral mediante Transformada Rápida de Fourier (FFT) como el monitoreo de la amplitud pico se centraron en dicho eje para la validación experimental.

Con el fin de evaluar la operatividad completa del módulo, se diseñó una prueba de integración funcional en condiciones controladas. Esta prueba buscó verificar que el sensor MPU6050

captura de forma aceptable las aceleraciones generadas por el sistema, utilizando tanto el firmware embebido como la interfaz de visualización desarrollada para el prototipo.

La **Tabla 4.4** presenta un resumen detallado de esta prueba, incluyendo los elementos técnicos involucrados, el procedimiento seguido, los resultados esperados y obtenidos, así como su evaluación final.

Asimismo, en las **Figs 4.4.1 a 4.4.6** se presentan las gráficas correspondientes a la aceleración medida y su análisis espectral para tres frecuencias representativas del sistema (3 Hz, 5 Hz y 7 Hz). Estas evidencias gráficas permiten validar la correcta captación del fenómeno vibratorio por parte del módulo de sensado, y su adecuada integración con el resto del sistema.

Tabla 4.4. Prueba módulo de sensado, autoría: propia.

Módulo de sensado				
Tipo de prueba	Integración funcional			
Hardware requerido	ESP32 DevKit V1, IMU MPU6050, jumpers, PC			
software requerido	FreeRTOS, firmware de adquisición I2C, interfaz de visualización			
Actor	Usuario			
Objetivo	Validar que la IMU MPU6050 mide adecuadamente la aceleración generada por la vibración mecánica del sistema			
Procedimiento de la prueba	Entrada	Salida esperada	Salida obtenida	Evaluación
Actividades	Vibración inducida en la estructura por los motores paso a paso, con rango de 3 Hz a 7 Hz y 30 micropasos por ciclo.	Señales periódicas de aceleración registradas en los ejes X,Y,Z,	Gráficas periódicas de aceleración en los ejes X, Y y Z obtenidas durante la vibración	Éxito

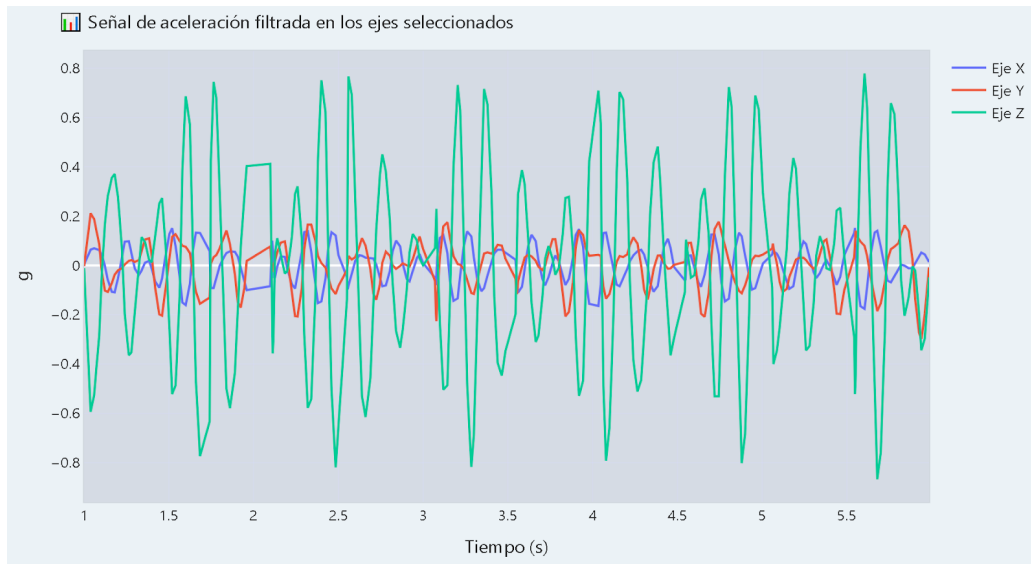


Fig 4.4.1 Señales de aceleración (7HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia.

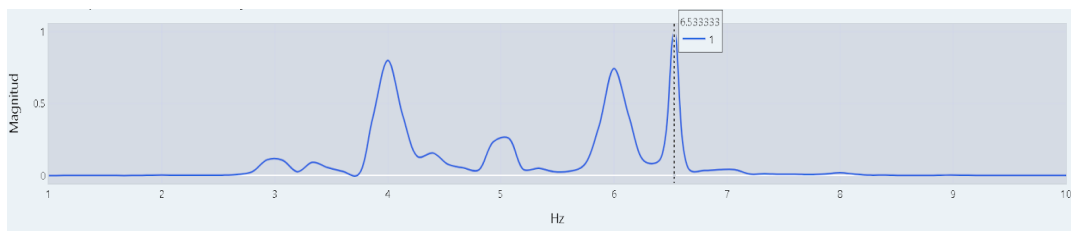


Fig 4.4.2 Gráfica FFT (7 HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia.

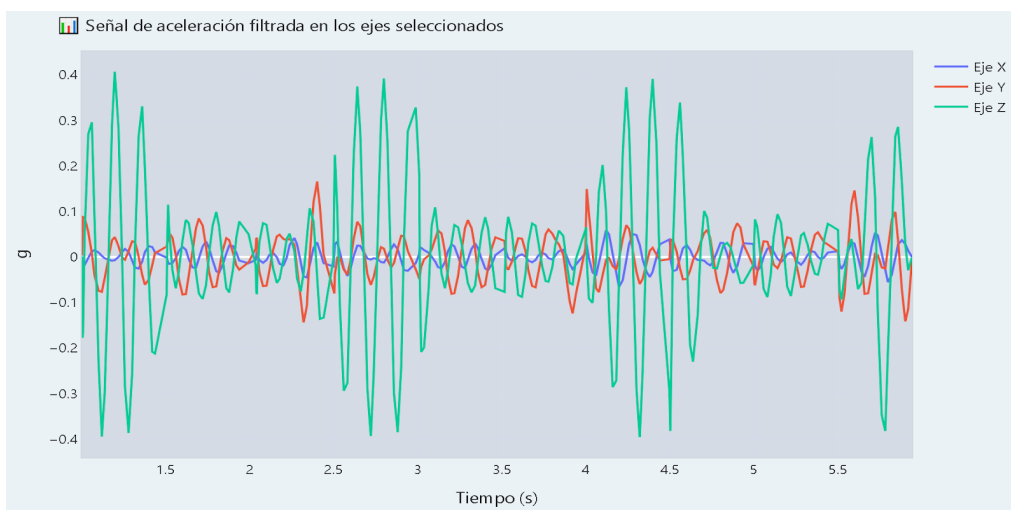


Fig 4.4.3 Señales de aceleración (5 HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia.

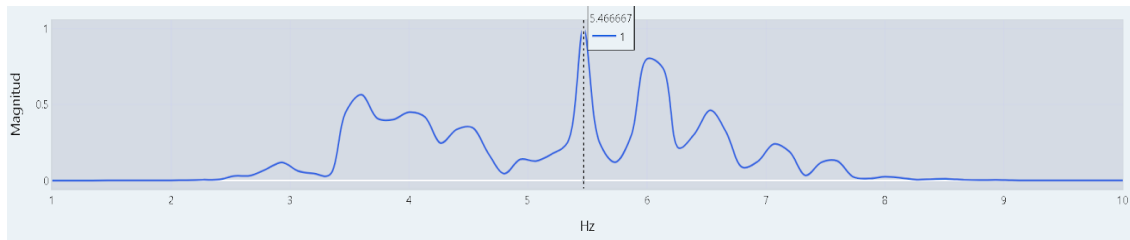


Fig 4.4.4 Gráfica FFT (5 HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia

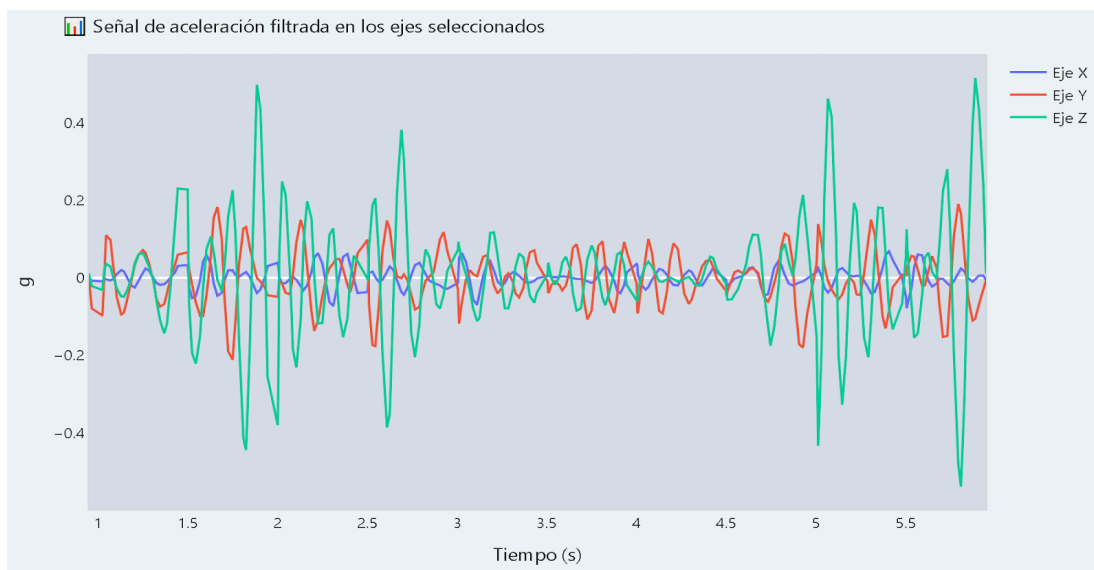


Fig 4.4.5 Señales de aceleración (3 HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia.

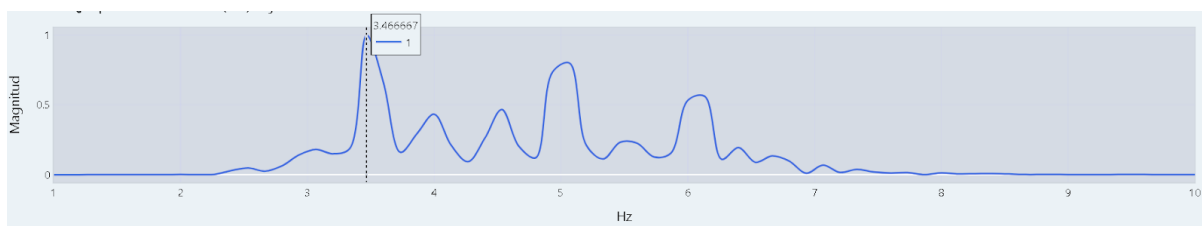


Fig 4.4.6 Gráfica FFT (3 HZ) registrada con la IMU MPU6050 durante la prueba de vibración. Autoría: propia

4.5 IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO DE COMUNICACIÓN SERIAL

El **módulo de comunicación serial** conecta el ESP32 con la interfaz externa mediante **UART**, permitiendo transmitir datos procesados (aceleraciones y frecuencias) y recibir parámetros de

control (frecuencia deseada). Para validar su funcionamiento, se realizó una prueba de transmisión, observando en el monitor serial la llegada continua y estructurada de datos del módulo de sensado. La **Fig. 4.5.1** muestra esta evidencia, confirmando la consistencia temporal y confiabilidad del canal bidireccional. La **Tabla 4.5.1** resume el protocolo de prueba, incluyendo hardware, software, objetivo y resultados obtenidos.

Tabla 4.5.1 Prueba módulo de comunicación serial, autoría: propia.

Módulo de comunicación serial				
Tipo de prueba	Integración funcional			
Hardware requerido	ESP32 DevKit V1, cable USB tipo C, computador, terminal serial o interfaz personalizada			
software requerido	FreeRTOS, UART driver de ESP-IDF, Visual Studio Code, interfaz de monitoreo (Docklight / PuTTY / Dash)			
Actor	Usuario			
Objetivo	Capturar datos de aceleración con la IMU y enviarlos en tiempo real por UART al computador; visualizar los datos en formato legible			
Procedimiento de la prueba	Entrada	Salida esperada	Salida obtenida	Evaluación
Actividades	Señales de aceleración leídas desde la IMU (X, Y, Z)	Cadenas seriales con los datos formateados correctamente y recibidas sin error en el monitor serial	Transmisión continua de datos en tiempo real con recepción estable y correcta de los valores esperados en la interfaz (ver Fig 3.3)	Éxito

```

ACC_X: 0.026, 0.512, ACC_Z: -
0.217
ACC_Y: 0.217, 0.494, ACC_Z: -
0.186
ACC_X: 0.069, 0.475, ACC_Z: -
0.234
ACC_Y: 0.020, 0.154, ACC_Z: -
0.020
ACC_X: 0.226, 0.922, ACC_Z: -
0.114
ACC_Y: 0.258, 0.707, ACC_Z: -
0.014

```

Fig 4.5.1 Captura del monitor serial mostrando los datos de aceleración transmitidos por UART. Autoría: propia.

4.6 IMPLEMENTACIÓN DE LA INTERFAZ

La interfaz gráfica, desarrollada en **Dash (Plotly)**, se organiza en bloques visuales modulares que facilitan la interacción, el monitoreo en tiempo real y el control de parámetros clave del sistema.

Bloques funcionales principales:

- **Encabezado institucional:** logotipos, nombre del sistema *ParkiMotion* y breve descripción.
- **Paneles de control e indicadores:**
 - Tipo de paciente y rango sugerido (leve, moderado, avanzado).
 - Slider de frecuencia deseada.
 - Indicadores de frecuencia aplicada, frecuencia detectada, amplitud pico y desplazamiento estimado.
 - Selección de ejes (X, Y, Z).
 - Estado de conexión con el microcontrolador.
- **Controles de adquisición:** botones verde y rojo para iniciar/detener captura y almacenamiento en CSV.
- **Paneles de visualización gráfica:**
 - Aceleración filtrada en tiempo real.
 - FFT en eje Z.
 - Evolución de la amplitud pico.

Diseño visual: paleta de colores contrastante para legibilidad, distribución en columnas y filas para flujo lógico, y elementos interactivos que aseguran una experiencia **intuitiva y profesional** para usuarios técnicos y no técnicos.

La **Fig. 4.6** muestra la vista general de la interfaz, donde se integran los módulos descritos y su disposición en pantalla.



Fig 4.6. Vista general de la interfaz gráfica autoría propia

En las siguientes subsecciones se realiza una descomposición detallada por módulos, ilustrando y explicando el propósito y funcionamiento de cada uno de los componentes clave de la interfaz.

4.6.1 Selector de etapa del temblor

Este módulo permite configurar la señal vibratoria generada por el sistema, combinando dos elementos principales: un **selector de etapa del temblor** y un **control deslizante para ajustar la frecuencia deseada**. Ambos se encuentran en la parte superior izquierda de la interfaz.

El **selector de etapa** ofrece tres opciones: leve, moderada y avanzada. Cada una representa un rango de frecuencias establecido a partir de los antecedentes revisados sobre temblores en personas con enfermedad de Parkinson. Si bien esta selección no altera automáticamente los

parámetros internos del sistema, sí proporciona un marco de referencia para orientar las pruebas dentro de un contexto emulativo realista.

Una vez seleccionada la etapa, el usuario puede usar el **control deslizable** para configurar la frecuencia deseada dentro del rango sugerido. Este valor se envía al ESP32, que lo utiliza como entrada de referencia para ajustar la frecuencia de vibración. La interfaz muestra de manera clara el valor seleccionado, junto con indicadores que permiten comparar la frecuencia aplicada y la detectada.

La **Tabla 4.6.1** presenta un resumen de estos rangos de frecuencia y su correspondencia con la configuración interna del sistema, incluyendo observaciones técnicas relevantes sobre la intensidad vibratoria observada en cada caso.

Tabla 4.6.1 Rango de frecuencia y características asociadas a cada etapa del temblor. Autoría: propia.

Etapas del temblor	Rango frecuencia configurada (Hz)	Micropasos por ciclo	Observación técnica
Leve	3.0-4.0	20	Vibración suave y baja amplitud
Moderada	4.1-5.9	20	Vibración media y perceptible
Avanzada	6.0-7.0	20	Vibración intensa y de mayor amplitud

Para validar la correcta implementación de esta funcionalidad, se diseñó una prueba funcional enfocada en el selector de etapa de la interfaz. En esta prueba se evaluó si, al seleccionar una etapa determinada desde la GUI, el sistema ajusta automáticamente los parámetros de vibración en el microcontrolador. Los resultados obtenidos se presentan en la **Tabla 4.6.2**, que documenta el objetivo, los elementos técnicos utilizados, el procedimiento seguido y la evaluación final de la prueba.

Tabla 4.6.2 Prueba del selector de etapa del temblor, autoría: propia.

Módulo de interfaz web	
Tipo de prueba	Funcional
Hardware requerido	ESP32, computador
software requerido	Dash, código de backend con envío de parámetros
Actor	Usuario
Objetivo	Validar que la selección de la etapa modifica los parámetros de vibración enviados al ESP32

Procedimiento de la prueba	Entrada	Salida esperada	Salida obtenida	Evaluación
Actividades	Selección del modo: leve, moderado o avanzado	Modificación dinámica de la frecuencia y micropasos por ciclo, generando variaciones perceptibles en la vibración	La selección es interpretada por el sistema y genera la respuesta vibratoria correspondiente	Éxito

La Fig 4.6.1 ilustra el módulo visual correspondiente al selector de etapa del temblor y al deslizador de frecuencia, resaltando su ubicación en la interfaz y los indicadores relacionados con la respuesta del sistema.

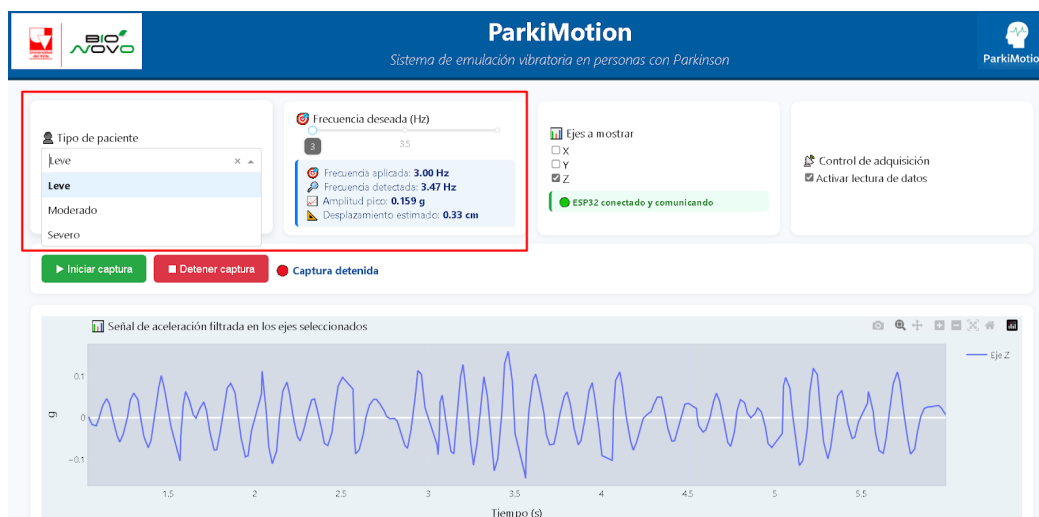


Fig 4.6.1 Prueba del selector de etapa del temblor en la interfaz Dash. Autoría: propia.

4.6.2 Estado de conexión y control de adquisición

Esta sección de la interfaz permite al usuario verificar la conexión activa entre el sistema embebido (ESP32) y la aplicación gráfica, así como iniciar o detener la captura de datos durante una sesión de prueba. El estado de conexión se representa mediante un indicador visual ubicado en el panel lateral derecho. Este cambia dinámicamente entre los estados “ESP32 conectado y comunicando” (color verde) y “Sin conexión” (color rojo), dependiendo de si se ha establecido correctamente la comunicación por puerto serial. Esta retroalimentación inmediata facilita la validación previa al inicio de cualquier ensayo.

Junto al estado de conexión, se encuentra el control de adquisición, que incluye una casilla para activar o desactivar la lectura de datos y dos botones codificados por color para controlar la captura:

- Iniciar captura (botón verde), que activa el registro y almacenamiento de datos en un archivo (csv).
- Detener captura (botón rojo), que finaliza el proceso de captura actual.

Durante el estado inactivo, se muestra una etiqueta informativa en color rojo con el mensaje “Captura detenida”, como confirmación visual adicional. Esta lógica de control garantiza que los datos solo se almacenen cuando el usuario así lo desee, manteniendo la integridad del análisis posterior.

La **Tabla 4.6.2** resume la prueba funcional realizada para verificar la correcta visualización de estos parámetros a través de la interfaz gráfica.

Tabla 4.6.2 Prueba de visualización de frecuencia detectada, amplitud pico y desplazamiento estimado. Autoría: propia.

Módulo de interfaz web				
Tipo de prueba	Funcional			
Hardware requerido	ESP32, computador			
software requerido	Dash, código de backend con envío de parámetros			
Actor	Usuario			
Objetivo	Verificar que el estado de conexión y los controles de captura responden adecuadamente a la interacción del usuario			
Procedimiento de la prueba	Entrada	Salida esperada	Salida obtenida	Evaluación
Actividades	Activación de conexión y pulsación de botones de captura	Indicador verde al conectar, etiqueta roja si está inactivo. CSV generado al iniciar.	El sistema muestra correctamente el estado. La captura almacena datos como se espera.	Éxito

La Fig 4.6.2 muestra esta sección de la interfaz, destacando los elementos de control y el indicador de estado del sistema.



Fig 4.6.2. Estado de conexión del sistema y controles de adquisición de datos en la interfaz . Autoría: propia

La interfaz **ParkiMotion** incorpora herramientas de análisis que permiten validar el comportamiento dinámico del sistema a partir de los datos de aceleración registrados. Estos resultados se presentan en forma gráfica y en un **panel numérico** ubicado en la parte superior, facilitando la verificación rápida del desempeño del sistema.

4.6.3.1 Panel de indicadores

El sistema calcula y actualiza en tiempo real:

- **Frecuencia aplicada:** valor definido por el usuario mediante el deslizador.
- **Frecuencia detectada:** frecuencia dominante obtenida del eje Z mediante FFT.
- **Amplitud pico:** máximo valor de aceleración en g en el eje Z.
- **Desplazamiento estimado:** cálculo aproximado en cm, derivado de la amplitud pico bajo el supuesto de movimiento armónico simple.

Este panel resume los resultados clave en una sola vista, complementando las gráficas y resultando especialmente útil en pruebas rápidas o validaciones experimentales. La **Fig. 4.6.3.3** muestra su disposición en la interfaz.

4.6.3.2 Análisis espectral (FFT)

La **Transformada Rápida de Fourier (FFT)** constituye la herramienta principal para evaluar el desempeño vibratorio del prototipo. A partir de la señal del MPU6050, se identifica la frecuencia dominante inducida por los motores paso a paso, la cual debe corresponder con la frecuencia configurada desde la interfaz, confirmando así la precisión del sistema.

Matemáticamente, la FFT se basa en la Transformada Discreta de Fourier (DFT), definida como:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j \frac{2\pi}{N} kn}, \quad k = 0, 1, \dots, N - 1$$

Ecuación 4.6.3.1 Transformada Discreta de Fourier

donde:

- $x(n)$ es la señal temporal,
- N es el número total de muestras,
- $x(k)$ representa la componente espectral de la frecuencia k ,
- j es la unidad imaginaria.

Este proceso transforma la señal desde el dominio del tiempo hacia el dominio de la frecuencia, facilitando la identificación de componentes sinusoidales. En particular, se aplica una ventana de Hann para reducir los efectos de fuga espectral, y la transformación se implementa utilizando las librerías NumPy y SciPy. Aunque en versiones anteriores se contempló permitir la selección del eje de análisis (X, Y o Z), se optó por trabajar exclusivamente con el eje Z, dado que, según las gráficas generadas en la interfaz, este presenta de forma consistente las aceleraciones de mayor magnitud durante la operación del sistema.

El resultado se presenta como un espectro de frecuencia, donde se identifica un pico dominante dentro del rango de frecuencia esperado (3–7 Hz), lo que valida el correcto funcionamiento del sistema de vibración.

De manera complementaria, la interfaz calcula la amplitud pico de la señal de aceleración en el eje Z, expresada en términos de aceleración gravitacional (g). Esta magnitud se convierte en desplazamiento lineal estimado, bajo el supuesto de un movimiento armónico simple. La relación utilizada para convertir una aceleración pico A_{pico} a desplazamiento es la siguiente [20]:

$$x_{pico} = \frac{A_{pico} \cdot 9.81}{(2\pi f)^2}$$

Ecuación 4.6.3.2 Cálculo del desplazamiento estimado a partir de la aceleración pico

donde:

- x_{pico} Es el desplazamiento estimado en metros,
- f Es la frecuencia dominante en Hz,
- A_{pico} Es la aceleración pico convertida a m/s².

Esta estimación permite comparar el comportamiento del sistema con los valores clínicamente reportados en pacientes con enfermedad de Parkinson, cuyas amplitudes típicas oscilan entre 1 mm y 10mm [6].

La interfaz, desarrollada con la librería Dash (Plotly), visualiza tanto la señal de aceleración filtrada como su espectro de frecuencia. La amplitud pico se presenta en una gráfica temporal para observar su evolución (ver Fig 4.6.3.2), mientras que la FFT se muestra en forma de gráfico de barras con resolución adecuada en el eje de frecuencias (ver Fig 4.6.3.1). Este

conjunto de visualizaciones convierte la interfaz en una herramienta práctica para validar experimentalmente los patrones vibratorios generados por el sistema.

Tabla 4.6.3.1 Prueba de la visualización de la FFT , autoría: propia.

Módulo de comunicación remota				
Tipo de prueba	Análisis espectral			
Hardware requerido	ESP32, IMU MPU6050, computador			
software requerido	Dash (Python), NumPy, SciPy, Plotly			
Actor	Usuario			
Objetivo	Verificar que el sistema representa correctamente la frecuencia, amplitud y desplazamiento estimado tanto en gráficas como en indicadores visuales			
Procedimiento de la prueba	Entrada	Salida esperada	Salida obtenida	Evaluación
Actividades	Muestras de aceleración del eje Z	Espectro FFT con pico dominante entre 3–7 Hz, amplitud pico coherente, desplazamiento estimado correcto	Los gráficos y el panel informativo reflejan los valores esperados de forma clara	Éxito

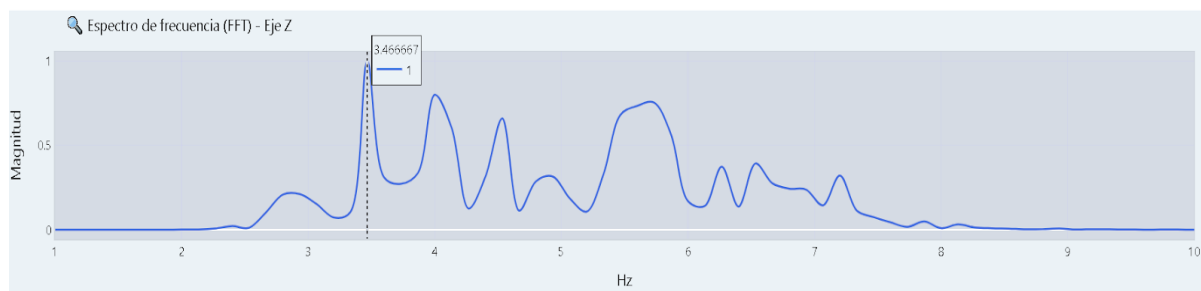
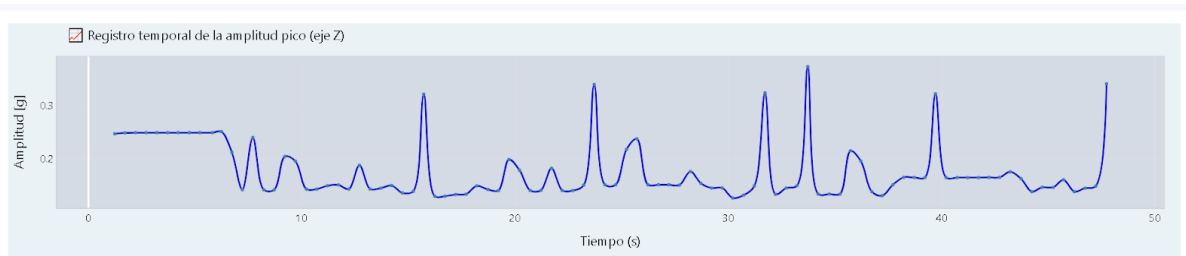


Fig 4.6.3.1 Espectro de frecuencia (3 Hz) generado mediante transformada rápida de Fourier (FFT). Autoría: propia



Fig

4.6.3.2 Registro temporal de amplitud pico. Autoría: propia



Fig 4.6.3.2 Panel de indicadores en la interfaz gráfica. Autoría: propia

4.6.4 Pseudocódigo del sistema de interfaz gráfica

El funcionamiento de la interfaz gráfica ParkiMotion se estructuró en dos archivos principales desarrollados en Python: **app.py**, encargado de definir la arquitectura visual y las interacciones del usuario, y **utils.py**, responsable de la adquisición y procesamiento de los datos recibidos desde el ESP32.

A continuación, se presentan los algoritmos en pseudocódigo que sintetizan la lógica operativa de cada uno de los módulos que componen la interfaz gráfica ParkiMotion. Estos algoritmos permiten comprender de forma estructurada cómo se gestionan la adquisición de datos, el procesamiento de señales y la interacción con el usuario.

Algoritmo 4.6.4.1 Adquisición y análisis de datos (utils.py)

- 1 Importar librerías de lectura serial, filtrado y procesamiento;
- 2 Configurar puerto UART para recepción desde ESP32;
- 3 Leer datos de aceleración (ejes X, Y, Z);
- 4 Aplicar filtro pasa banda;
- 5 Calcular FFT y frecuencia dominante (eje Z);
- 6 Calcular amplitud pico y desplazamiento estimado;
- 7 Actualizar variables compartidas con app.py;
- 8 Si la captura está activa → guardar datos en archivo CSV

Algoritmo 4.6.4.1 Pseudocódigo del módulo de control PID discreto, autoría propia.

Algoritmo 4.6.4.2 Construcción de la interfaz gráfica (app.py)

```

1  Importar librerías Dash, Plotly y estilos personalizados;
2  Definir layout: encabezado, panel de controles, botones e indicadores;
3  Añadir gráficas: aceleración, FFT, amplitud pico;
4  Definir callbacks para:
5      Actualizar frecuencia deseada desde el slider;
6      Cambiar etapa del temblor;
7      Iniciar o detener captura de datos;
8      Actualizar visualización de gráficas e indicadores;
9  Ejecutar servidor local Dash;

```

Algoritmo 4.6.4.2. Pseudocódigo del módulo de control PID discreto, autoría propia.

4.7 IMPLEMENTACIÓN DEL SISTEMA ELECTRÓNICO

La implementación electrónica de ParkiMotion integró los módulos de sensado, control embebido, actuación y comunicación, materializando la arquitectura previamente diseñada. El núcleo está compuesto por un **ESP32 DevKit V1**, seleccionado por su capacidad dual core, compatibilidad con FreeRTOS y protocolos UART/I2C, que gestiona en paralelo las tareas críticas del sistema.

Todos los elementos se organizaron en una **PCB personalizada** diseñada en EasyEDA, bajo criterios de orden lógico, claridad estructural y facilidad de montaje, en la cual se integraron los principales módulos.

- **Módulo de Sensado:** IMU MPU6050 conectada vía I2C, registra aceleraciones en los tres ejes.
- **Módulo de Actuación:** dos drivers DRV8825 que controlan motores NEMA 17 para generar vibraciones en los ejes X y Y.
- **Módulo de Comunicación:** transmisión y recepción de datos entre ESP32 e interfaz Dash mediante dos UART independientes.
- **Módulo de Alimentación:** diseñado para incluir un regulador buck LM2596, que convierte 12 V DC a 5 V.
- **Indicadores LED:** retroalimentación visual del estado del sistema, motores y comunicación.

La **disposición final garantiza modularidad, estabilidad y escalabilidad**. Las Figs. 4.7.1 y 4.7.2 muestran la vista inferior y superior de la PCB ensamblada, donde se distinguen zonas reservadas para futuras expansiones (LM2596, sensor ultrasónico, tercer DRV8825). Estas previsiones, aunque no implementadas en la versión actual, facilitarán mejoras como el control adaptativo de amplitud o nuevas funcionalidades sin necesidad de rediseño físico.

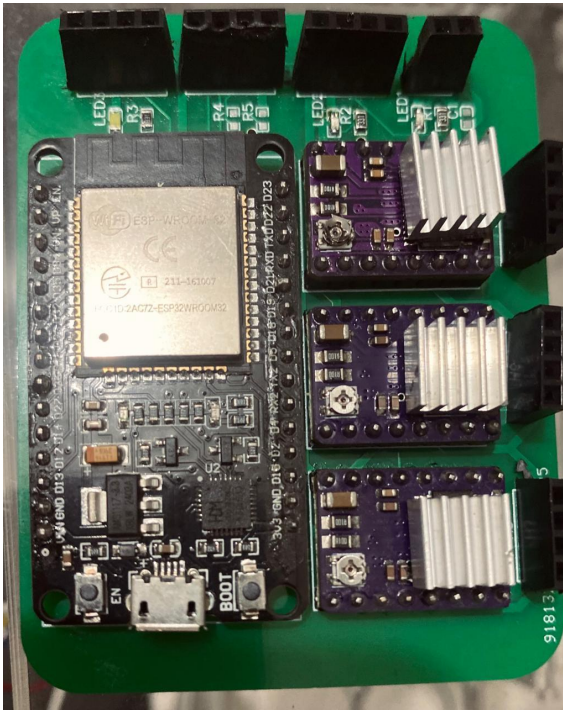


Fig 4.7.1. Vista superior de la PCB . Autoría: propia.

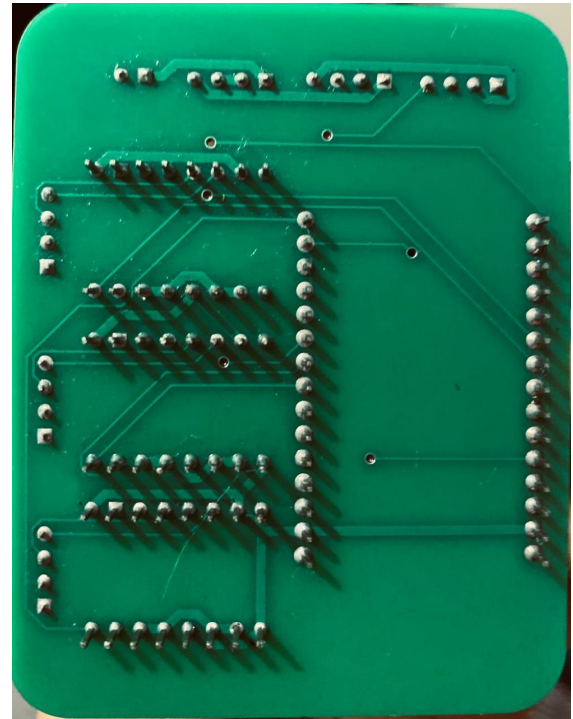


Fig 4.7.2. Vista inferior de la PCB ensamblada. Autoría: propia.

4.8 MONTAJE FINAL DEL SISTEMA

El montaje final del prototipo ParkiMotion consolida la integración de todos los componentes desarrollados a lo largo del proyecto, incluyendo la estructura mecánica, los motores paso a paso, el sistema de control embebido en ESP32 y la electrónica de sensado y actuación. Esta fase permitió verificar el funcionamiento coordinado del sistema en condiciones reales, y sentó las bases para la validación funcional en pruebas controladas.

El sistema fue ensamblado sobre una estructura rígida de perfiles de aluminio y placas acrílicas, que garantiza estabilidad, modularidad y accesibilidad a los componentes. En la base se ubicó la PCB personalizada, sobre la cual se montaron el microcontrolador ESP32 DevKit V1 y los drivers DRV8825. Desde allí se distribuyen las conexiones hacia los motores y sensores, permitiendo un control eficiente y seguro.

En la **Fig 4.8.1** se presenta una vista general del prototipo ensamblado, en la que se observan claramente los subsistemas físicos y electrónicos que conforman ParkiMotion. Adicionalmente, en el **Anexo : Video_Funcionamiento** se incluye un registro audiovisual que muestra el sistema en operación, evidenciando la correcta integración de los módulos y el funcionamiento del lazo de control basado en la interfaz gráfica.

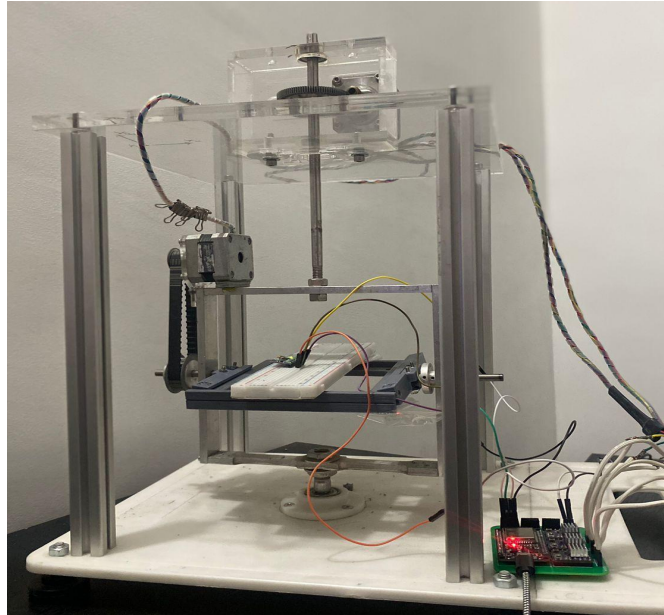


Fig 4.8.1 Prototipo funcional ensamblado con los componentes accesibles en el contexto regional. Autoría: propia.

4.9 DESARROLLO DE PROTOCOLO DE EVALUACIÓN USANDO EL PROTOTIPO FUNCIONAL IMPLEMENTADO.

Este capítulo ítem presenta el protocolo experimental diseñado para evaluar cuantitativamente el desempeño del sistema ParkiMotion, basado en la comparación entre las señales deseadas y las señales medidas en condiciones simuladas de temblores asociados a la enfermedad de Parkinson. Se consideran tres escenarios representativos: paciente leve, moderado y severo.

El sistema fue configurado para generar vibraciones a estas frecuencias mediante el control de motores paso a paso, y las señales resultantes fueron capturadas usando el sensor inercial (IMU MPU6050) ubicado sobre la superficie vibratoria.

Los datos de aceleración y frecuencia fueron adquiridos en tiempo real a través de la interfaz gráfica desarrollada en Dash (Python), que también permitió la exportación automática a archivos .csv para su posterior análisis.

Para garantizar la comparabilidad entre grupos, se seleccionaron **217 muestras por tipo de paciente**, constituyendo un conjunto balanceado de observaciones. Cada muestra incluyó:

- Frecuencia deseada enviada al sistema
- Frecuencia detectada mediante análisis FFT
- Amplitud pico de aceleración (en g)
- Desplazamiento estimado (en cm)

4.9.2 PROCESAMIENTO COMPUTACIONAL DE LOS DATOS

El procedimiento computacional automatizó las tareas necesarias para procesar, analizar y visualizar los datos obtenidos en las pruebas experimentales. Este procesamiento incluyó la carga y depuración de datos, la clasificación por grupos según la frecuencia deseada, el cálculo de métricas de desempeño y la exportación de resultados gráficos y tabulares.

Algoritmo 4.9.2. Pseudocódigo del procesamiento estadístico

```

1  Importar librerías y configurar rutas de entrada/salida;
2  Cargar todos los archivos CSV y unirlos en un único DataFrame;
3  Asignar a cada grupo su frecuencia deseada esperada (3, 5, 7 Hz);
4  Balancear el número de muestras: seleccionar 217 observaciones por grupo;
5  Calcular:
6      Frecuencia detectada promedio;
7      Error medio absoluto;
8      Amplitud pico y desplazamiento promedio;
9  Exportar:
10 Tablas como imágenes PNG y archivos Excel;
11 Gráficas de barras con errores medios.
```

Algoritmo 4.9.2 Pseudocódigo del análisis estadístico en Python, autoría propia.

4.9.3 RESULTADOS OBTENIDOS

El análisis estadístico de las pruebas de validación permitió obtener métricas clave para las etapas de temblor (leve, moderado y severo), evaluando frecuencia deseada, frecuencia detectada, amplitud pico y desplazamiento estimado. Estas métricas se contrastaron con los parámetros de referencia establecidos en los antecedentes clínicos y experimentales (3–7 Hz y amplitud 1–12 mm), asegurando coherencia entre las pruebas realizadas y los rangos reportados en la literatura.

Tabla 4.9.3 Medidas observadas por tipo de paciente autoría: propia

Paciente	Frecuencia deseada (Hz)	Frecuencia detectada promedio (Hz)	Amplitud pico promedio (g)	Error promedio (Hz)
leve	3	3.182	0.268	0.182
moderado	5	4.928	0.271	0.072
severo	7	6.756	0.292	0.244

La **Tabla 4.9.3** presenta los valores promedio por tipo de paciente, evidenciando que la frecuencia detectada se aproxima a la deseada con un error inferior a ± 0.25 Hz en todos los casos. Asimismo, la amplitud pico muestra un incremento progresivo a medida que aumenta la severidad del temblor, validando el comportamiento escalonado de la plataforma vibratoria.

La **Tabla 4.9.4** incluye métricas complementarias como el error medio absoluto, la desviación estándar y el error máximo registrado, además del desplazamiento promedio. Estos resultados refuerzan la capacidad del sistema para reproducir patrones dinámicos diferenciados según el escenario clínico simulado.

Tabla 4.9.4. Resumen de errores y amplitudes autoría: propia

Paciente	Frecuencia deseada esperada (Hz)	Error Medio (Hz)	Desviación Std (Hz)	Error Máximo (Hz)	Desplazamiento Medio (cm)
leve	3	0.209	0.19	0.47	0.677
moderado	5	0.278	0.206	0.9	0.241
severo	7	0.215	0.184	0.5	0.162

La **Tabla 4.9.4** complementa el análisis mostrando que, en todos los escenarios, el error medio absoluto se mantuvo cercano a 0.2 Hz, con desviaciones estándar por debajo de 0.21 Hz y errores máximos inferiores a 1 Hz. Estos resultados confirman la estabilidad del sistema en lazo cerrado, con un desempeño suficientemente preciso para las frecuencias de interés (3–7 Hz). Asimismo, se observa que el desplazamiento medio disminuye progresivamente con el incremento de frecuencia, comportamiento esperado en sistemas vibratorios, mientras que la amplitud media de aceleración se mantiene en un rango consistente entre 0.24 g y 0.29 g.

La **Fig 4.9.3** ilustra el error medio absoluto agrupado por tipo de paciente. En todos los casos, este error se mantuvo por debajo de los 0.3 Hz, indicando un desempeño aceptable del sistema en lazo cerrado. El grupo moderado presentó la mayor desviación estándar y el mayor error máximo, lo que podría deberse a resonancias o transitorios no lineales en esa frecuencia intermedia. En contraste, los grupos leve y severo mostraron un comportamiento más estable y predecible.

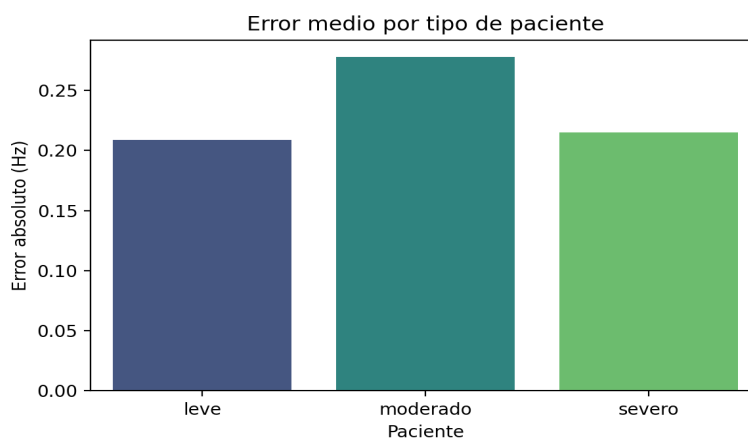


Fig 4.9.3. Error medio por tipo de paciente, autoría propia

Capítulo 5

CONCLUSIONES Y TRABAJOS FUTUROS

5.1 CONCLUSIONES

Los resultados obtenidos permiten afirmar que se cumplió satisfactoriamente con el desarrollo de un sistema funcional para la emulación de temblores mecánicos tipo Parkinson en entornos controlados. A continuación, se detalla el alcance logrado con respecto a los objetivos específicos planteados:

1. Definir los requerimientos del dispositivo basado en análisis de literatura.

El análisis de literatura técnica y clínica permitió establecer requerimientos funcionales y no funcionales precisos, fundamentados en parámetros realistas del temblor parkinsoniano (frecuencia de 3–7 Hz y amplitud de 1–12 mm). Estos criterios garantizaron que el diseño del sistema fuera modular, de bajo costo, escalable y adaptable, sirviendo como base sólida para la concepción de la arquitectura y la validación experimental posterior.

2. Concebir la arquitectura del dispositivo mediante una orientación modular.

Se estableció una arquitectura modular compuesta por los subsistemas de sensado, control, actuación, comunicación e interfaz de usuario, la cual demostró ser efectiva para organizar el diseño del sistema en bloques independientes y complementarios. Esta estructuración facilitó el análisis, la implementación y la validación de cada módulo, asegurando además la coherencia funcional del conjunto y la posibilidad de escalar o reemplazar componentes en futuras versiones del prototipo.

3. Verificar los componentes de la arquitectura propuesta basado en modelos analíticos, computacionales o experimentales.

Los módulos principales del sistema fueron verificados mediante pruebas experimentales en la planta vibratoria, lo que permitió validar su funcionamiento frente a distintas frecuencias de entrada. La implementación del controlador PID discreto y de la interfaz gráfica facilitó tanto la modulación de la salida como la observación en tiempo real de las respuestas obtenidas. Estos ensayos confirmaron no solo el cumplimiento de las funcionalidades previstas para cada módulo bajo condiciones de laboratorio, sino también la pertinencia de la arquitectura propuesta como base confiable para el control y análisis del temblor simulado.

4. Desarrollar un prototipo funcional considerando la tecnología regional.

Se construyó un prototipo funcional del sistema ParkiMotion empleando componentes de amplia disponibilidad en el entorno local, lo cual garantizó la viabilidad técnica y económica del proyecto. El prototipo integró un microcontrolador ESP32, sensores MPU6050, drivers DRV8825 y motores paso a paso, montados sobre una PCB personalizada que optimizó el orden y la robustez de las conexiones. A nivel de software, se implementó un sistema embebido multitarea basado en FreeRTOS y una interfaz gráfica desarrollada en Python (Dash), que permitió controlar parámetros y visualizar resultados en tiempo real. La elección de componentes

accesibles y la organización modular del sistema no solo facilitaron el montaje y la verificación de su operatividad, sino que además sentaron las bases para futuras ampliaciones y adaptaciones del prototipo a escenarios más complejos.

5. Desarrollar un protocolo de evaluación del prototipo funcional implementado.

Se diseñó y ejecutó un protocolo experimental para evaluar el desempeño del prototipo bajo tres escenarios representativos de temblor parkinsoniano: leve, moderado y severo. Este protocolo incluyó la adquisición de datos en tiempo real mediante la IMU, su procesamiento en la interfaz gráfica y un análisis estadístico sistemático que garantizó la comparabilidad entre grupos. Los resultados mostraron una correspondencia clara entre la frecuencia deseada y la detectada, así como una evolución coherente de la amplitud y el desplazamiento, validando cuantitativamente la capacidad del sistema para emular patrones vibratorios característicos del temblor. Además, los ensayos realizados sirvieron como base para identificar posibles mejoras y ajustes en futuras versiones del dispositivo, consolidando a ParkiMotion como una plataforma experimental confiable y replicable.

5.2 LIMITACIONES DEL DESARROLLO

A pesar de que el prototipo ParkiMotion cumplió con los objetivos específicos propuestos y demostró un funcionamiento sólido en las pruebas experimentales, se identificaron varias limitaciones técnicas que podrían ser abordadas en futuras fases de desarrollo:

- **Estimación indirecta del desplazamiento:** La magnitud de desplazamiento lineal fue estimada a partir de la integración numérica de las señales de aceleración. Este proceso es inherentemente sensible al ruido y a los errores acumulativos típicos de la integración, particularmente en sistemas de medición sin referencias absolutas. La precisión obtenida depende del filtrado aplicado y de la estabilidad de la señal registrada, lo que introduce una fuente de incertidumbre en los resultados.
- **Ruido e incertidumbre en el sensor inercial:** El sensor MPU6050 presenta un nivel de ruido característico que afecta la calidad de los datos, especialmente en contextos donde se requieren mediciones precisas de pequeñas oscilaciones. Esto impacta tanto la estimación de la frecuencia dominante como el cálculo de la amplitud pico, limitando la resolución con la que se pueden discriminar cambios sutiles en la señal vibratoria.
- **Validación restringida a entorno experimental:** La verificación del sistema se realizó exclusivamente bajo condiciones simuladas en laboratorio. Las frecuencias y amplitudes utilizadas para representar las etapas leve, moderada y severa del temblor se definieron con base en literatura especializada, sin contar con datos reales adquiridos directamente de pacientes con enfermedad de Parkinson. Esto limita la extrapolación directa de los resultados a un contexto clínico real.

- **Interacción unidireccional usuario-sistema:** El sistema depende de la selección manual de la etapa del temblor por parte del usuario. No se cuenta con una retroalimentación autónoma que permita adaptar la emulación o sugerir cambios en la etapa de simulación basándose en las mediciones actuales, reduciendo la autonomía del sistema.
- **Análisis centrado únicamente en el eje Z:** El procesamiento de las señales inerciales y la estimación de la frecuencia dominante se realizó únicamente sobre el eje transversal Z. Esta decisión técnica se fundamentó en que el diseño mecánico del sistema (basado en una estructura tipo gimbal) concentra la mayor parte del movimiento oscilatorio relevante sobre dicho eje, al ser el que refleja directamente la vibración generada por el motor principal de rotación. Sin embargo, esta limitación excluye posibles análisis de oscilaciones en los otros ejes, que podrían aportar información adicional en sistemas más complejos.

Estas limitaciones no comprometen el objetivo fundamental del proyecto, que era demostrar la factibilidad técnica del sistema ParkiMotion como plataforma experimental para emular temblores. No obstante, representan oportunidades de mejora claras para futuras iteraciones, orientadas a aumentar la precisión, adaptabilidad y realismo clínico del sistema.

5.3 TRABAJOS FUTUROS

Los resultados obtenidos con el prototipo ParkiMotion abren el camino para nuevas líneas de desarrollo e investigación aplicada en el campo de la asistencia motora para personas con enfermedad de Parkinson. A continuación, se plantean posibles extensiones y mejoras al sistema, orientadas tanto a su perfeccionamiento como a la creación de nuevos dispositivos derivados:

- **Desarrollo de utensilios funcionales con control activo de temblores**
Una de las proyecciones más relevantes es el diseño de dispositivos portables inspirados en el sistema desarrollado, como una cuchara inteligente tipo *Steady Spoon*, que utilice sensores inerciales y actuadores miniaturizados para atenuar temblores en tiempo real durante la alimentación. El conocimiento adquirido sobre detección de frecuencia, control en lazo cerrado y diseño mecánico desacoplado puede ser reutilizado y adaptado para este fin.
- **Miniaturización del sistema y transición a formato wearable**
A partir del modelo de banco validado, se propone una etapa futura de rediseño orientada a miniaturizar el sistema de control y los actuadores, para su incorporación en dispositivos portátiles o vestibles. Esto permitiría probar soluciones más cercanas al entorno real de uso, incluyendo pruebas de ergonomía, autonomía energética y usabilidad.

- **Implementación de algoritmos adaptativos basados en aprendizaje automático**

Se sugiere explorar técnicas de aprendizaje automático que permitan ajustar automáticamente los parámetros del controlador en función del perfil vibratorio detectado, sin requerir intervención manual del usuario. Esto posibilitará una adaptación más precisa a diferentes tipos de temblor y evolución del paciente.

- **Evaluación con señales reales y validación clínica colaborativa**

Un paso esencial será realizar una validación experimental utilizando datos provenientes de pacientes reales (en colaboración con instituciones de salud), para comparar el rendimiento del sistema ante condiciones clínicas auténticas. Esto permitirá establecer métricas de efectividad, robustez y precisión en escenarios reales.

- **Diseño de nuevos módulos de asistencia sensoriomotriz**

El enfoque modular de ParkiMotion puede extenderse para desarrollar sistemas que no solo emulen, sino también compensen el movimiento, aplicando fuerzas opuestas o patrones de estimulación háptica. Esto abre la puerta a prototipos de rehabilitación interactiva y entrenamiento neuromotor.

En conjunto, estas proyecciones apuntan a la consolidación de una plataforma tecnológica versátil, capaz de evolucionar desde la etapa experimental hacia soluciones clínicas reales, económicas y adaptadas a las necesidades de personas con Parkinson en etapas tempranas o avanzadas.

6. ANEXOS

En la **Fig 6.1** se presenta la estructura de carpetas organizada durante el desarrollo del proyecto ParkiMotion. Estos archivos pueden consultarse en el siguiente enlace: [ANEXOS](#)

Mi unidad > Proyecto de grado_par... > ANEXOS ▾

Tipo ▾ Personas ▾ Modificado ▾ Fuente ▾

Nombre ↑	Propietario
 Diseño_PCB	 yo
 ARQUITECTURA_SISTEMA	 yo
 CODIGOS_FUENTE Y PRUEBAS	 yo
 MANUAL_DE_USUARIO	 yo
 VERSIONES_TG	 yo
 VIDEO_FUNCIONAMIENTO_PARKIMOTION	 yo

Fig 6.1 Anexos

A continuación, se describen los contenidos mas relevantes:

1. **Manual_Usuario:** Instructivo detallado para la manipulación e interpretación de resultados del sistema ParkiMotion.
2. **ARQUITECTURA_SISTEMA:** Incluye diagramas, tablas de casos de uso y el diseño CAD de la estructura mecánica.
3. **Diseño_PCB:** Contiene los archivos de diseño esquemático y trazado de la PCB desarrollada.
4. **Codigos_fuente_y_pruebas:** Incluye todo el código del proyecto:
 - **Parkimotion (ESP32):** Código en lenguaje C bajo FreeRTOS, organizado por módulos funcionales.
 - **Interfaz_Dash:** Aplicación en Python para visualización, control del sistema y procesamiento de datos CSV (aceleraciones, amplitud, frecuencia).
 - **MATLAB_DESIGN:** Scripts para modelado, identificación de planta y diseño del controlador PID.
 - **Resultados:** CSV y scripts de análisis empleados en la validación experimental.
5. **Video_funcionamiento:** Registro audiovisual del sistema en operación, mostrando la interacción entre el ESP32, la interfaz Dash y la respuesta física del sistema mecánico. Este material evidencia la validación funcional del prototipo.
6. **Versiones_TG:** Integra algunas de las versiones anteriores de este trabajo.

REFERENCIAS

- [1] Organización Mundial de la Salud, *Neurological Disorders: Public Health Challenges*. Ginebra, Suiza: OMS, 2006.
- [2] M. Jankovic, "Parkinson's disease: clinical features and diagnosis," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 79, no. 4, pp. 368–376, 2008.
- [3] E. R. Dorsey et al., "Global, regional, and national burden of Parkinson's disease, 1990–2016: a systematic analysis," *The Lancet Neurology*, vol. 17, no. 11, pp. 939–953, Nov. 2018.
- [4] GBD 2016 Parkinson's Disease Collaborators, "Global, regional, and national burden of Parkinson's disease, 1990–2016: a systematic analysis," *The Lancet Neurology*, vol. 17, no. 11, pp. 939–953, 2018.
- [5] Departamento Administrativo Nacional de Estadística (DANE), "Indicadores de salud y discapacidad en Colombia 2023," Bogotá, Colombia, 2023. [En línea]. Disponible en: <https://www.dane.gov.co/>
- [6] C. A. Lowry, "Motor and non-motor symptoms in Parkinson's disease: impact on quality of life," *Parkinsonism & Related Disorders*, vol. 14, no. S2, pp. S49–S52, 2008.
- [7] S. W. Anderson, "Cost-effectiveness of assistive devices for Parkinson's tremor," *NeuroRehabilitation*, vol. 45, no. 2, pp. 247–252, 2019.
- [8] Liftware, "Liftware Steady: Handle and Spoon for hand tremor," [En línea]. Disponible en: <https://www.liftware.com/steady/>
- [9] Gyenno Technologies, "Gyenno Spoon: Smart spoon for tremor," [En línea]. Disponible en: <https://www.gyenno.com/>
- [8] W. Liu, J. Li, Y. Wang, X. Zhao and J. Zhang, "Three-axis gimbal design and its application," *2016 IEEE International Conference on Information and Automation (ICIA)*, Ningbo, 2016, pp. 1200–1205, doi: 10.1109/ICInfA.2016.7832014.
- [9] P. Teerapabkajornet, M. Nitta and H. Kobayashi, "Design and simulation of a customize three-axis gimbal structure," *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, Takamatsu, 2021, pp. 553–558, doi: 10.1109/ICMA52553.2021.9535857.
- [10] I. de la Torre et al., "Evaluation of active tremor compensation devices for Parkinson's disease: a systematic review," *Sensors*, vol. 20, no. 3, p. 736, Feb. 2020, doi: 10.3390/s20030736.
- [11] A. Heldman, D. Espay, R. LeWitt and M. Giuffrida, "Clinician versus machine: Reliability and responsiveness of motor endpoints in Parkinson's disease," *Parkinsonism & Related Disorders*, vol. 17, no. 6, pp. 467–472, Jul. 2011, doi: 10.1016/j.parkreldis.2011.04.002.

- [12] G. Giuffrida et al., "Detection of hand tremor in patients with Parkinson's disease using a wearable system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 4, pp. 664–673, Jul. 2013, doi: 10.1109/TNSRE.2013.2258916.
- [13] F. Nunes, D. Folgado, R. P. Paiva and H. Gamboa, "Hand-tremor frequency estimation in videos," *Journal of Biomedical and Health Informatics*, vol. 25, no. 4, pp. 1201–1210, Apr. 2021, doi: 10.1109/JBHI.2020.3037576.
- [14] J. Hove and J. Keller, "Hand rhythmic tapping and timing in Parkinson's disease," *Brain and Cognition*, vol. 58, no. 2, pp. 145–151, Mar. 2005, doi: 10.1016/j.bandc.2004.11.005.
- [15] D. Díaz et al., "Hand tremor characterization from a spatiotemporal representation using wearable sensors," *Sensors*, vol. 21, no. 17, p. 5873, Sep. 2021, doi: 10.3390/s21175873.
- [16] MedlinePlus, "Enfermedades neurodegenerativas," Biblioteca Nacional de Medicina de EE. UU., 2022. [En línea]. Disponible en: <https://medlineplus.gov/spanish/>
- [17] J. Kim, J. Chung, y Y. Kim, "Muscle activation patterns in Parkinsonian tremor," *Clinical Neurophysiology*, vol. 130, no. 6, pp. 945–953, 2019.
- [18] NINDS, "Parkinson's Disease: Hope Through Research," National Institute of Neurological Disorders and Stroke, NIH, 2020.
- [19] A. Carlsson, "Presence of dopamine in brain," *Science*, vol. 130, no. 3368, pp. 219–220, 1959.
- [20] C. Hammond, H. Bergman, y P. Brown, "Pathological synchronization in Parkinson's disease: networks, models and treatments," *Trends in Neurosciences*, vol. 30, no. 7, pp. 357–364, 2007.
- [21] M. M. Hoehn y M. D. Yahr, "Parkinsonism: Onset, progression and mortality," *Neurology*, vol. 17, pp. 427–442, 1967.
- [22] M. Fasano y A. Daniele, "Bradykinesia in Parkinson's disease," *Movement Disorders*, vol. 30, no. 4, pp. 473–481, 2015.
- [23] Mayo Clinic, "Enfermedad de Parkinson – Síntomas y causas," 2022. [En línea]. Disponible en: <https://www.mayoclinic.org/es/diseases-conditions/parkinsons-disease/symptoms-causes>
- [24] L. Smith, R. Johnson y A. Nguyen, "Standardized Testing Protocols for Parkinson's Motor Evaluation," *Neurorehabilitation and Neural Repair*, vol. 35, no. 5, pp. 389–397, 2021.
- [25] Parkinson's Foundation, "Hoehn and Yahr Staging of Parkinson's Disease," 2023. [En línea]. Disponible en: <https://www.parkinson.org>
- [26] K. Ogata, *Sistemas de control en ingeniería*, 5.^a ed., México: Pearson Educación, 2010.
- [27] R. Dorf y R. Bishop, *Modern Control Systems*, 13th ed., Pearson, 2016.

- [28] L. G. McDonald y S. D. Block, "Clinical trial designs in Parkinson's disease: Methodological considerations," *Parkinsonism & Related Disorders*, vol. 48, pp. 51–57, 2018.
- [29] InvenSense, *MPU-6000 and MPU-6050 Register Map and Descriptions*, 2013. [En línea]. Disponible en: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf>
- [30] Espressif Systems, *ESP32-WROOM-32 Datasheet*, 2022. [En línea]. Disponible en: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
- [31] FreeRTOS, *FreeRTOS Kernel Documentation*, Real Time Engineers Ltd., 2023. [En línea]. Disponible en: <https://www.freertos.org>
- [32] Wantai Motor Co., Ltd., *NEMA17 Stepper Motor Specifications*, 2020. [En línea]. Disponible en: <https://www.wantmotor.com/ProductsView.asp?id=155&pid=78>
- [33] Texas Instruments, *DRV8825 Stepper Motor Controller Datasheet*, 2021. [En línea]. Disponible en: <https://www.ti.com/lit/ds/symlink/drv8825.pdf>
- [34] R. Tocci, N. Widmer y G. Moss, *Sistemas Electrónicos Digitales*, 10.^a ed., Pearson, 2013.
- [35] B. Wilson, *Stabilization Platforms: A Guide to Gimbals and Motion Control Systems*, 2nd ed., New York, NY, USA: McGraw-Hill, 2019.
- [36] F. J. Espinosa Ramírez, "Control de orientación en sistemas embarcados usando arquitectura tipo gimbal y modelo Newton-Euler," Tesis de pregrado, Univ. Autónoma de Nuevo León, México, 2021.