DEPARTAMENTO DE
INGENIERÍA
MECÁNICA

UNIVERSIDAD DE SANTIAGO DE CHILE

Facultad de Ingeniería
Departamento de Ingeniería Mecánica
Carrera de Ingeniería Civil Mecatrónica

Proyecto Ingeniería de Software Interdisciplinaria

Sistema de validación de rendimiento de una bobina Tesla - Sprint 2

Integrantes:

Nombre	RUT	Correo electrónico
Matías Kohl	21.378.326-2	matias.kohl@usach.cl
Alejandro Alarcón	21.813.314-2	alejandro.alarcon.p@usach.cl
Pablo Canales	21.405.022-6	pablo.canales.p@usach.cl
Daniel González	20.239.382-9	daniel.gonzalez.i@usach.cl

Profesor: Daniel Gacitúa / Ricardo Hasbún

Asignatura: Ingeniería de Software Interdisciplinaria

Fecha de entrega: 7 de Enero de 2026

Índice

1. Introducción

El presente informe describe el estado de avance del desarrollo de un sistema hardware–software orientado a la validación experimental del funcionamiento de una bobina Tesla de baja potencia. El sistema combina una maqueta física instrumentada, un microcontrolador ESP32 encargado de la adquisición de datos y un software desarrollado en Python para el procesamiento y visualización de las variables medidas.

El objetivo de este documento es exponer el diseño general del sistema, la arquitectura de software adoptada y la forma en que se integra el hardware con el software de análisis. En particular, se presenta la organización del código bajo el patrón Modelo–Vista–Controlador (MVC), así como el flujo de datos desde los sensores hasta la interfaz de usuario.

El contenido del informe refleja un avance superior al 50 % del desarrollo comprometido, considerando módulos completamente implementados y otros definidos a nivel de arquitectura y diseño, los cuales serán desarrollados e integrados en las siguientes etapas del proyecto.

2. Objetivo

2.1. Objetivo general

Describir el diseño y el estado de avance del sistema hardware–software desarrollado para la validación experimental del funcionamiento de una bobina Tesla, abordando la arquitectura de software, la integración con el hardware de medición y la organización del código implementado.

2.2. Objetivos específicos

- Presentar la estructura general del sistema y su flujo de operación.
- Describir la arquitectura de software basada en el patrón Modelo–Vista–Controlador (MVC).
- Explicar la organización del repositorio y la responsabilidad de los principales módulos de software.
- Detallar la integración entre el hardware de adquisición de datos y el software de procesamiento y visualización.
- Evidenciar el estado actual del desarrollo, correspondiente a aproximadamente un 50 % del avance comprometido.

3. Descripción general del sistema

En esta sección se presenta una visión general del sistema desarrollado, describiendo su funcionamiento a alto nivel y la interacción entre los componentes de hardware y software. Se aborda el flujo de operación del sistema, los límites de su alcance y las principales consideraciones asociadas a su uso experimental.

3.1. Visión global del sistema

El sistema desarrollado corresponde a una plataforma experimental que integra hardware de medición, un microcontrolador y un software de análisis con el objetivo de validar el funcionamiento de una bobina Tesla bajo distintas condiciones de operación. Su funcionamiento general se estructura como una secuencia controlada de etapas, orientadas a garantizar la seguridad del sistema, la adquisición confiable de datos y la correcta ejecución del experimento.

En una primera etapa, el sistema se encuentra en estado de reposo, con la bobina Tesla desenergizada. Antes de iniciar cualquier medición, se realiza una validación eléctrica inicial que considera la verificación de conexiones, aislamiento, referencia a tierra y estado general de los componentes. Solo una vez superada esta etapa se habilita la alimentación del sistema de potencia, asegurando condiciones seguras de operación.

Posteriormente, se activa el sistema de monitoreo basado en un microcontrolador ESP32, el cual inicializa los sensores, el servomotor y la comunicación con el software. Durante esta fase se adquieren de forma sincronizada las variables de interés del experimento, tales como la intensidad relativa del campo electromagnético, la señal lumínica asociada a la descarga, el ángulo del servomotor y el voltaje de alimentación de la bobina.

El software recibe estos datos y gestiona el experimento según el modo de operación definido, ya sea automático o manual. El sistema permite ejecutar el barrido de medición durante un intervalo controlado, tras lo cual se detiene la adquisición de datos y se procede al apagado ordenado del sistema, retornando al estado seguro inicial.

Esta visión global del sistema se apoya en los diagramas de flujo presentados, los cuales ilustran la secuencia de validación, monitoreo y control que gobierna la operación completa del sistema experimental.

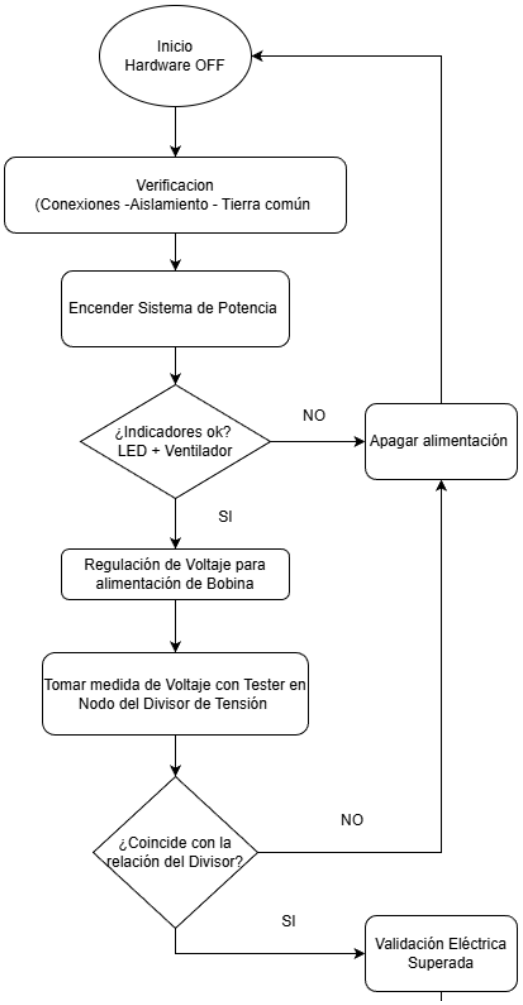

Figura 1: Diagrama de flujo general del sistema experimental.

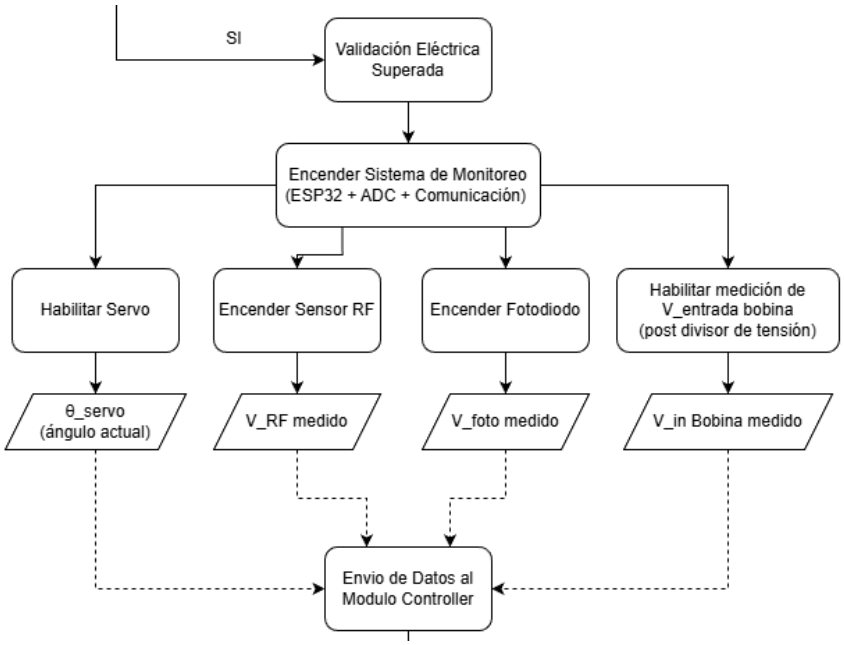

Figura 2: Diagrama de flujo del sistema de monitoreo y adquisición de datos.

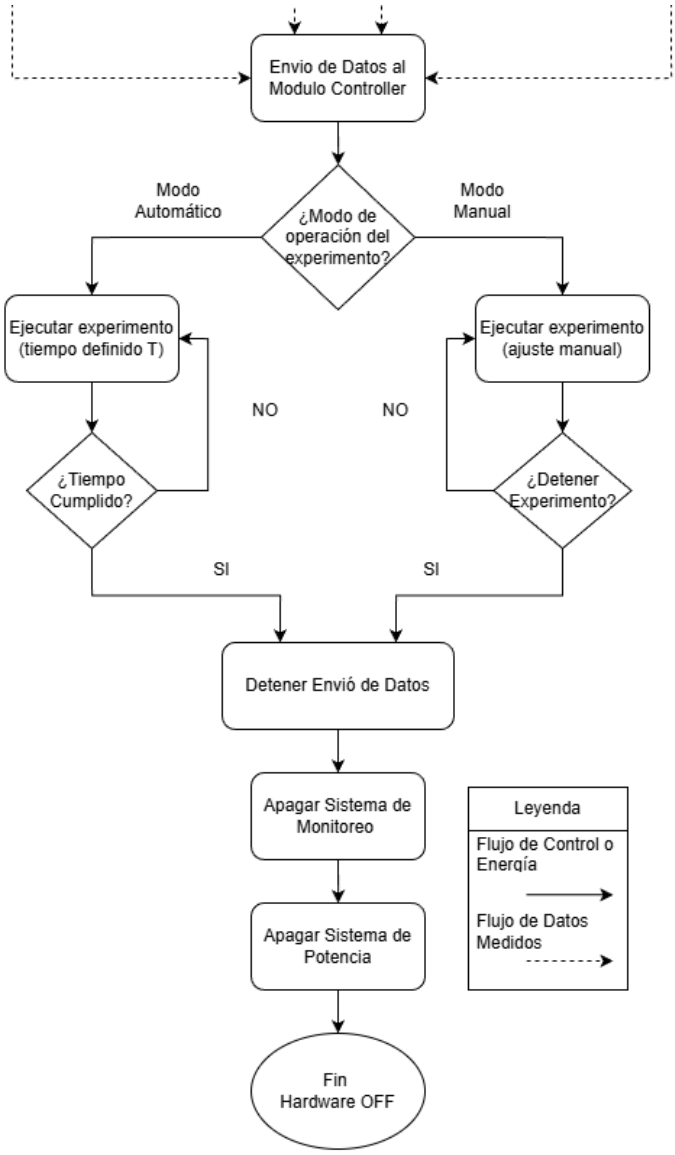

Figura 3: Diagrama de flujo del control del experimento y finalización segura.

3.2. Relación hardware–software

La relación entre el hardware y el software del sistema se basa en una separación clara de responsabilidades. El sistema físico se encarga de la generación, medición y transmisión de datos, mientras que el software realiza el procesamiento, análisis y visualización de la información adquirida.

El hardware está compuesto por una bobina Tesla instrumentada, sensores dedicados y un microcontrolador ESP32. Este último actúa como nodo de adquisición, leyendo las señales provenientes de los sensores, controlando el servomotor asociado al sistema de medición y enviando los datos al computador mediante comunicación serial USB. El microcontrolador no realiza procesamiento avanzado, limitándose a la lectura, empaquetado y transmisión de las variables medidas.

El software, desarrollado en Python, actúa como la capa de control y análisis del sistema. A través del módulo Controlador, los datos enviados por la ESP32 son recibidos y decodificados, para luego ser traspasados al Modelo, donde se aplican operaciones de procesamiento y cálculo de magnitudes derivadas. Finalmente, la información procesada es presentada al usuario mediante la capa de visualización.

Esta arquitectura permite desacoplar el hardware del software, facilitando la modificación, prueba y simulación del sistema sin requerir cambios en la maqueta física. Asimismo, asegura que la lógica experimental y la interpretación de los datos se centralicen en el software, mientras que el hardware mantiene una función robusta y acotada de adquisición y control básico.

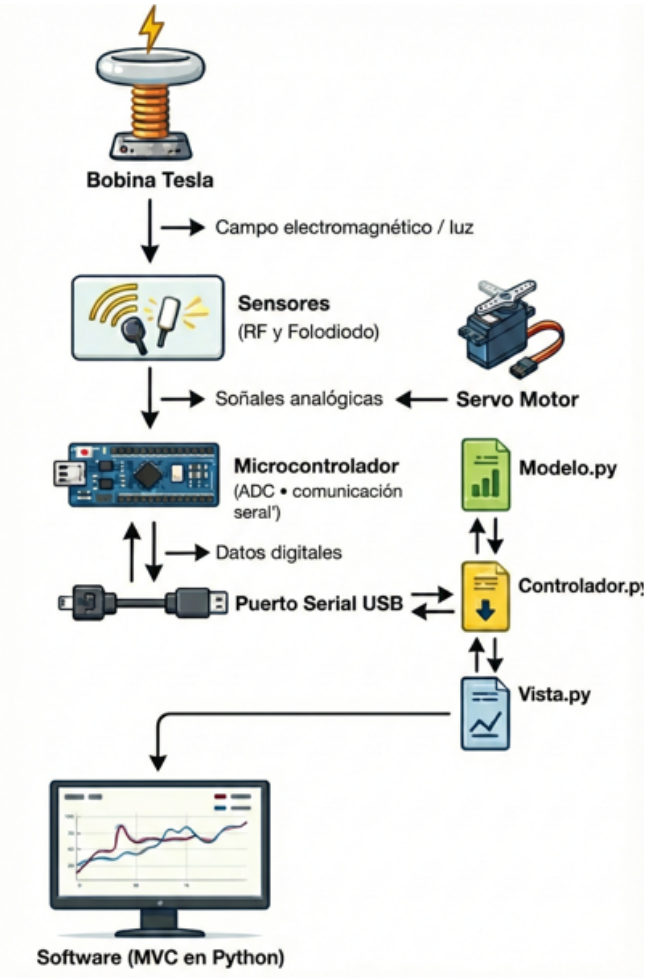

Figura 4: Arquitectura general del sistema y flujo de información entre hardware y software bajo el patrón MVC.

3.3. Alcances y limitaciones del sistema

El sistema desarrollado tiene como alcance principal la validación experimental del funcionamiento de una bobina Tesla de baja potencia mediante la adquisición y análisis de variables eléctricas, electromagnéticas y de control. El enfoque del proyecto está orientado a observar tendencias, comportamientos relativos y correlaciones entre variables, más que a la obtención de magnitudes absolutas con precisión metrológica.

Dentro de los alcances del sistema se contempla la modificación de parámetros relevantes del experimento. En particular, el ángulo del servomotor que posiciona el loop sensor puede ser ajustado desde el software, permitiendo variar de forma controlada la distancia efectiva entre el sensor de campo electromagnético y la bobina. Adicionalmente, el sistema considera la regulación del voltaje de alimentación de la bobina Tesla, lo que incide directamente en la potencia de operación del sistema.

La regulación del voltaje de alimentación se realiza de forma manual mediante un regulador DC–DC, ajustando el voltaje de entrada a la bobina dentro de un rango definido. Este voltaje es medido a través de un divisor de tensión y enviado al software como una variable de entrada, la cual es utilizada para el análisis, modelado teórico y validación del comportamiento experimental de la bobina, sin que el ajuste del voltaje sea realizado directamente por el software.

Entre las limitaciones del sistema se encuentra la ausencia de control automático de encendido y apagado de la potencia de la bobina y del sistema de monitoreo, los cuales se realizan de forma manual. El software no actúa como interruptor físico del sistema, sino que implementa un control lógico del experimento, habilitando o bloqueando la recepción y el procesamiento de datos en función de la interacción del usuario, por ejemplo, al iniciar o detener una medición.

Finalmente, debido al estado de avance del proyecto, algunas funcionalidades del controlador se encuentran definidas a nivel de arquitectura y flujo lógico, pero aún no han sido completamente implementadas. Estas serán abordadas en etapas posteriores del desarrollo conforme a la planificación establecida.

4. Descripción del hardware

En esta sección se describen los principales componentes que conforman el sistema físico desarrollado para la validación experimental de la bobina Tesla. Para cada componente se presentan su función dentro del sistema, sus especificaciones técnicas relevantes y las consideraciones asociadas a su uso. Finalmente, se describe la forma en que estos elementos se integran y conectan entre sí.

4.1. Bobina Tesla

La bobina Tesla corresponde al elemento central del sistema experimental y es la encargada de generar el campo electromagnético y la descarga eléctrica observable. Se trata de una bobina Tesla de baja potencia, diseñada para operar de forma segura en un entorno de laboratorio.

Cuadro 1: Especificaciones técnicas de la bobina Tesla

Parámetro	Valor aproximado
Tipo	Bobina Tesla de baja potencia
Potencia nominal	15 W
Voltaje de alimentación	15–24 V DC
Corriente máxima	2 A
Longitud de arco	5–10 mm

La bobina es alimentada mediante un regulador de voltaje externo y no es controlada directamente por el software. Su operación se supervisa a través de mediciones indirectas del campo electromagnético y de la descarga lumínica.

4.2. Fuente de alimentación y regulador de voltaje

La alimentación del sistema de potencia se realiza a partir de una fuente industrial conectada a la red eléctrica de 220 V AC, la cual entrega un voltaje continuo de 24 V DC. Este voltaje es posteriormente ajustado mediante un regulador DC–DC para alimentar la bobina Tesla.

Cuadro 2: Especificaciones del regulador de voltaje

Parámetro	Valor
Tipo	Regulador DC–DC reductor
Voltaje de entrada	24 V DC
Rango de salida	15–24 V DC
Corriente máxima	8 A
Ajuste de voltaje	Manual (potenciómetro)

El ajuste del voltaje se realiza de forma manual mediante un tornillo de regulación. El valor aplicado a la bobina es medido y enviado al software para su uso en el análisis teórico y experimental del sistema.

4.3. Microcontrolador ESP32

El microcontrolador ESP32 actúa como núcleo del sistema de monitoreo y adquisición de datos. Su función principal es leer las señales provenientes de los sensores, controlar el servomotor y transmitir la información al computador.

Cuadro 3: Especificaciones principales del ESP32

Parámetro	Descripción
Microcontrolador	ESP32
Voltaje de operación	3.3 V
Comunicación	USB (serial)
Entradas analógicas	ADC integrado

El ESP32 no realiza procesamiento avanzado de datos, limitándose a la adquisición, empaquetado y transmisión de las variables medidas hacia el software.

4.4. Sensor de campo electromagnético

La medición del campo electromagnético se realiza mediante un loop de cobre acoplado a un detector logarítmico de RF (AD8307). El loop capta el flujo magnético generado por la bobina, induciendo una señal eléctrica proporcional a la variación del campo.

Cuadro 4: Características del sensor de campo electromagnético

Parámetro	Valor
Tipo de sensor	Loop inductivo + AD8307
Frecuencia máxima	500 MHz
Tipo de salida	Voltaje analógico (logarítmico)
Uso principal	Medición relativa de campo RF

Este sensor permite evaluar variaciones relativas del campo electromagnético en función de la distancia y de las condiciones de operación de la bobina.

4.5. Fotodiodo para detección de descarga

La detección de la descarga eléctrica se realiza mediante un fotodiodo de silicio, encargado de medir la intensidad lumínica y el instante temporal en que ocurre la descarga visible de la bobina Tesla.

Cuadro 5: Especificaciones del fotodiodo

Parámetro	Valor
Modelo	BPW34
Tipo	Fotodiodo PIN de silicio
Rango espectral	Visible e infrarrojo cercano
Salida	Corriente/voltaje analógico

La señal obtenida permite correlacionar la actividad lumínica con el comportamiento electromagnético del sistema.

4.6. Servomotor

El sistema utiliza un servomotor para controlar el ángulo del brazo que sostiene el loop sensor. Este movimiento permite variar de forma controlada la distancia entre el sensor y la bobina Tesla.

Cuadro 6: Especificaciones del servomotor

Parámetro	Valor
Tipo	Servo micro 9G
Ángulo de operación	0–180°
Control	Señal PWM
Uso en el sistema	Barrido angular del sensor

El ángulo del servomotor es controlado desde el software, constituyendo uno de los parámetros manipulables del experimento.

4.7. Modelo matemático y módulo de ecuaciones

El análisis del comportamiento de la bobina Tesla se basa en un conjunto de relaciones matemáticas implementadas de forma centralizada en el módulo `ecuaciones.py`. Este módulo encapsula todas las expresiones necesarias para vincular las variables medidas experimentalmente con magnitudes físicas de interés, así como para realizar comparaciones entre modelos teóricos y datos experimentales.

La centralización de las ecuaciones permite mantener separada la formulación matemática del resto del sistema, facilitando la validación, reutilización y extensión del modelo sin afectar la adquisición de datos ni la interfaz de usuario.
El ángulo del servomotor, definido inicialmente en grados, es convertido a radianes para su uso en expresiones trigonométricas mediante:

$$\theta_{\text{rad}} = \theta_{\text{deg}} \cdot \frac{\pi}{180} \tag{1}$$

La distancia efectiva entre el loop del sensor y el centro de la bobina Tesla se calcula a partir de la geometría del sistema mecánico, según la expresión:

$$d(\theta) = \sqrt{L^2 + y_0^2 - 2\, y_0\, L\, \sin(\theta)} \tag{2}$$

donde L corresponde a la longitud efectiva del brazo del servomotor, y_0 es la distancia fija entre el eje de rotación y el centro de la bobina, y θ es el ángulo del servomotor en radianes.

El voltaje real aplicado a la bobina Tesla se reconstruye a partir del voltaje medido por el ADC del ESP32 mediante un divisor resistivo, de acuerdo con la relación:

$$V_{\text{bobina}} = V_{\text{div}} \cdot \frac{R_{\text{top}} + R_{\text{bot}}}{R_{\text{bot}}} \tag{3}$$

donde V_{div} corresponde al voltaje medido en el divisor, y R_{top} y R_{bot} representan las resistencias superior e inferior, respectivamente.

La potencia eléctrica teórica de entrada a la bobina se estima a partir del voltaje reconstruido y una resistencia equivalente del sistema, mediante:

$$P_{\text{in}} = \frac{V_{\text{bobina}}^2}{R_{\text{eq}}} \tag{4}$$

donde R_{eq} representa una resistencia equivalente utilizada para fines de análisis comparativo.

Para modelar la disminución de las magnitudes físicas con la distancia, se utiliza una relación de tipo potencia inversa:

$$f(r) = \frac{1}{r^n} \tag{5}$$

donde r corresponde a la distancia entre el sensor y la bobina, y n es un exponente que define la pendiente de la tendencia.

La tendencia teórica del campo electromagnético en función de la distancia se define como:

$$B_{\text{teo}}(r) = K_b \cdot \frac{1}{r^n} \tag{6}$$

donde K_b es un factor de escala teórico utilizado para comparar la forma de las curvas teóricas y experimentales.

De forma análoga, la intensidad lumínica teórica del arco eléctrico se modela como:

$$L_{\text{teo}}(r) = K_l \cdot \frac{1}{r^2} \tag{7}$$

donde K_l es un factor de escala teórico.

Las magnitudes experimentales del campo electromagnético y de la intensidad lumínica se definen directamente a partir de los voltajes medidos por los sensores:

$$B_{\text{exp}} = V_{\text{RF}} \tag{8}$$

$$L_{\text{exp}} = V_{\text{foto}} \tag{9}$$

El conjunto de ecuaciones descritas permite relacionar directamente las mediciones experimentales con modelos teóricos simplificados, proporcionando una base común para el análisis del comportamiento de la bobina Tesla y la validación del sistema de monitoreo desarrollado. En particular, los modelos teóricos se utilizan para describir la tendencia física esperada en función de la distancia, mientras que los factores de escala introducidos permiten ajustar dichas expresiones a las magnitudes medidas experimentalmente. De este modo, el análisis se centra en la comparación de comportamientos y tendencias relativas, más que en la obtención de valores absolutos calibrados, lo que resulta coherente con el alcance y objetivos del proyecto.

4.8. Montaje e integración del hardware

El sistema hardware se monta sobre una maqueta física diseñada para separar claramente las zonas de potencia y de monitoreo, minimizando interferencias y mejorando la seguridad durante la operación. La base de la maqueta corresponde a una superficie de madera, sobre la cual se definen dos áreas funcionales bien delimitadas.

La primera corresponde a una zona de seguridad asociada a la bobina Tesla, definida como un círculo de aproximadamente 60 cm de diámetro centrado en la posición de la bobina. Esta zona se mantiene libre de componentes electrónicos y elementos conductores, con el fin de evitar acoplamientos indeseados y reducir riesgos eléctricos durante la generación de la descarga.

La segunda área corresponde a una zona rectangular de control y monitoreo, donde se instalan el microcontrolador ESP32, los sensores, el servomotor, las mini protoboards y los circuitos auxiliares. Esta separación física permite un montaje ordenado y facilita el acceso a los componentes durante el ajuste y la medición.

El sistema utiliza tres mini protoboards independientes, cada una destinada a una función específica. La primera mini protoboard implementa el circuito del divisor de tensión utilizado para medir el voltaje de alimentación aplicado a la bobina Tesla. Este divisor reduce el voltaje a un rango compatible con las entradas analógicas del ESP32 e incluye resistencias de precisión, capacitores electrolíticos y cerámicos para filtrado de ruido, así como resistencias en serie para la protección del microcontrolador.

La segunda mini protoboard se utiliza para la alimentación del servomotor. El servo es alimentado mediante una batería externa independiente, con el objetivo de evitar caídas de tensión o reinicios no deseados del ESP32 durante los picos de corriente asociados al arranque del motor. Este circuito incorpora capacitores de desacoplo para absorber transitorios y asegurar un funcionamiento estable.

La tercera mini protoboard corresponde a los circuitos de adquisición de señales de los sensores, específicamente el detector de RF y el fotodiodo. Las señales analógicas provenientes de estos sensores son acondicionadas mediante resistencias en serie y capacitores de filtrado antes de ser enviadas a las entradas analógicas del ESP32, reduciendo el ruido y protegiendo la electrónica de medición.

4.9. Enlace de datos y consideraciones de instrumentación

La medición del campo electromagnético se realiza mediante un sensor de RF compuesto por un loop de cobre y un detector logarítmico AD8307. La señal de radiofrecuencia es captada por el loop de cobre, el cual se encuentra conectado al sensor mediante un cable coaxial, asegurando una transmisión adecuada de la señal y reduciendo la captación de ruido externo. El loop se posiciona de forma perpendicular al eje de la bobina Tesla, maximizando el acoplamiento con el campo magnético generado.

El módulo detector de RF se encuentra protegido por una jaula de Faraday, lo que permite aislarlo de interferencias electromagnéticas externas y mejorar la calidad de la medición. Asimismo, se establece una referencia de tierra común para todo el sistema, incluyendo la bobina, los sensores, el microcontrolador y la malla del cable coaxial, con el fin de minimizar diferencias de potencial y ruido eléctrico.

La detección de la descarga eléctrica se realiza mediante un fotodiodo, el cual se encuentra montado dentro de un cilindro opaco de color negro. Este encapsulado posee una ranura orientada hacia el punto de nacimiento de la descarga, permitiendo limitar el campo de visión del sensor y reducir la influencia de luz ambiente externa.

El microcontrolador ESP32 adquiere las señales analógicas provenientes del divisor de tensión, del sensor de RF y del fotodiodo, así como la información asociada al ángulo del servomotor. Estos datos son enviados al computador mediante comunicación serial USB. Aunque el sistema de monitoreo queda energizado al conectar el ESP32, el software implementa un control lógico que determina cuándo los datos deben ser recibidos y procesados, por ejemplo, al iniciar un experimento desde la interfaz de usuario.

De este modo, el sistema separa el control físico del hardware del control lógico del flujo de datos, permitiendo una operación segura, ordenada y coherente con la arquitectura de software definida.

5. Metodología de desarrollo

El proyecto se desarrolla utilizando un enfoque ágil basado en Scrum, definido en el primer informe del curso. En el presente documento, este marco se utiliza únicamente como referencia para contextualizar el sprint actual, los requisitos abordados y el estado de avance del sistema.

Durante este sprint, el trabajo se ha centrado en el montaje del hardware, la implementación progresiva de los módulos de software y la preparación del sistema para su posterior integración y validación.

5.1. Requisitos del sprint

Los requisitos definidos para el sprint correspondiente a este informe se agrupan en requisitos funcionales y no funcionales, los cuales delimitan el alcance del trabajo realizado.

Requisitos funcionales

- Medir el voltaje de alimentación aplicado a la bobina Tesla mediante un circuito divisor de tensión, obteniendo una señal compatible con el sistema de adquisición.
- Adquirir la señal de salida del sensor de campo electromagnético (RF) en forma de voltaje analógico, representativo de variaciones relativas del campo generado por la bobina.
- Adquirir el voltaje generado por el fotodiodo asociado a la detección de la descarga eléctrica, permitiendo identificar la ocurrencia y variaciones relativas de la emisión lumínica.
- Controlar el ángulo del servomotor para posicionar el loop sensor, habilitando la variación controlada de la distancia de medición respecto a la bobina Tesla.
- Transmitir los datos adquiridos desde el microcontrolador hacia el entorno de desarrollo (terminal serial), verificando la correcta adquisición y formato de las mediciones.
- Procesar y organizar las mediciones adquiridas en el módulo Modelo del software, aplicando las ecuaciones definidas para el análisis experimental.
- Almacenar conjuntos de datos en archivos CSV a partir de fuentes de datos simuladas, como base para pruebas y validación del flujo de procesamiento.
- Visualizar de forma gráfica las variables medidas y procesadas mediante la Vista del sistema, utilizando datos simulados representativos del experimento.

Requisitos no funcionales

- Garantizar la seguridad del sistema mediante la separación física entre los circuitos de potencia y los circuitos de monitoreo y control, reduciendo riesgos eléctricos y de interferencia electromagnética.

- Minimizar interferencias y perturbaciones en las señales de medición mediante el uso de filtrado pasivo, apantallamiento y una referencia de tierra común.
- Asegurar la estabilidad del microcontrolador durante la operación, evitando reinicios no deseados mediante una alimentación independiente del servomotor y una adecuada gestión de corrientes.
- Mantener una arquitectura de software modular y escalable, que permita la extensión del sistema y la incorporación de nuevas funcionalidades sin afectar los módulos existentes.
- Garantizar la legibilidad y mantenibilidad del código, mediante una organización clara de los módulos, nombres representativos y separación explícita de responsabilidades.
- Diseñar la interfaz de usuario de forma clara y comprensible, priorizando la correcta interpretación de las variables medidas y del estado del sistema por sobre aspectos estéticos.
- Priorizar la operación manual y controlada de los elementos de potencia del sistema, evitando automatizaciones que puedan comprometer la integridad del usuario o del equipo durante la operación experimental.

5.2. Sprint Backlog actual

El Sprint Backlog del presente sprint y su grado de ejecución se resumen en la Tabla 7.

Cuadro 7: Sprint Backlog y porcentaje de ejecución

Actividad del Sprint Backlog	**Ejecución**
Construcción física del hardware	80 %
Integración del microcontrolador con sensores	100 %
Implementación de circuitos de medición y acondicionamiento	100 %
Pruebas físicas del sistema de medición	85 %
Calibración de sensores RF y fotodiodo	100 %
Programación del módulo Modelo	90 %
Programación del módulo Vista	90 %
Programación del módulo Controlador	70 %
Definición del flujo lógico del experimento	80 %
Documentación técnica del sistema	80 %

5.3. Avance comprometido y estado actual del desarrollo

El avance alcanzado en el presente sprint corresponde a un progreso superior al 50 % del desarrollo total del proyecto, de acuerdo con la planificación establecida.

Desde el punto de vista del hardware, el sistema se encuentra prácticamente completo. La maqueta física, los circuitos de medición, el acondicionamiento de señales y las consideraciones de seguridad han sido implementados, quedando pendientes únicamente pruebas finales de validación y ajuste bajo condiciones reales de operación.

En cuanto al software, el desarrollo se encuentra en una etapa de construcción modular. El módulo Modelo presenta un alto grado de implementación y permite el procesamiento y organización de las mediciones adquiridas. La Vista se encuentra definida e implementada de manera parcial, a nivel de estructura y componentes gráficos, mientras que el módulo Controlador ha sido definido a nivel de arquitectura y se encuentra en una etapa de implementación parcial.

La integración completa entre los módulos de software, junto con la ejecución de pruebas funcionales del sistema en su conjunto, constituye el principal objetivo de los siguientes sprints.

6. Arquitectura y organización del software

En esta sección se describe la arquitectura de software adoptada para el sistema, así como la forma en que dicha arquitectura se materializa en la organización del repositorio. El software se estructura siguiendo el patrón Modelo–Vista–Controlador (MVC), lo que permite separar responsabilidades, facilitar el desarrollo modular y desacoplar la lógica de procesamiento de la interfaz de usuario y de la comunicación con el hardware.

A continuación, se presenta primero una visión general de la arquitectura MVC aplicada al sistema y luego se detallan los distintos módulos que la componen, indicando su rol, estructura y estado actual de implementación.

6.1. Descripción general de la arquitectura MVC

La arquitectura de software del sistema se basa en el patrón Modelo–Vista–Controlador (MVC), aplicado a un contexto de integración hardware–software. En esta arquitectura, el sistema físico y el microcontrolador ESP32 actúan como fuentes de datos, mientras que el software organiza la adquisición, el procesamiento y la visualización de la información de manera desacoplada.

El Controlador cumple el rol de intermediario entre el hardware y el software, recibiendo los datos enviados por el ESP32 a través de comunicación serial y coordinando su entrega al Modelo para su procesamiento. El Modelo se encarga de estructurar las mediciones, ejecutar los cálculos necesarios y mantener el estado interno del sistema. Finalmente, la Vista permite la interacción con el usuario y la visualización de los resultados, sin acceder directamente al hardware ni a la lógica de cálculo.

Esta separación de responsabilidades permite desarrollar y probar los distintos módulos de forma independiente, lo cual resulta especialmente relevante considerando el estado de avance actual del proyecto y la necesidad de integrar progresivamente el sistema completo.

6.2. Modelo: rol y estructura

El módulo `model/` implementa la capa **Modelo** del patrón MVC. Su responsabilidad es representar el estado interno del sistema, estructurar las mediciones y ejecutar el procesamiento/cálculo necesario para generar resultados derivados que serán consumidos por la Vista.

Elementos principales implementados (clases, métodos y atributos):

- **Clase** `Modelo` (`modelo.py`)
  - **Atributos relevantes:** `historial`, `Req`, `Rtop`, `Rbot`, `Kb`, `Kl`, `L_m`, `y0_m`.
  - **Métodos:**
    - `__init__()`: inicializa parámetros y estructuras internas.
    - `procesar_muestra()`: transforma una muestra cruda en resultados procesados.
    - `reset()`: reinicia el estado del modelo y/o su historial.
    - `get_historial()`: entrega el historial acumulado para visualización/análisis.
- **Clases de datos** (`muestra.py`)
  - `MuestraCruda`: estructura para agrupar variables medidas (entrada al modelo).
  - `MuestraProcesada`: estructura para resultados derivados (salida del modelo).
- **Funciones de cálculo** (`ecuaciones.py`)
  - Conversión/geométrica: `deg_rad()`, `distancia()`.
  - Entrada eléctrica: `vin_real()`, `potencia_in()`.
  - Tendencias/teoría/experimentales: `tendencia()`, `b_teo()`, `l_teo()`, `b_exp()`, `l_exp()`.
  - Métricas de error: `error_abs()`, `error_rel()`.
- **Persistencia de resultados** (`almacenamiento.py`)
  - `exportar_csv()`: exportación de mediciones/resultados a formato CSV.

En conjunto, el módulo Modelo concentra la representación interna del sistema y el procesamiento de la información adquirida durante el experimento. A través de estructuras de datos específicas, el Modelo organiza las mediciones provenientes del sistema físico y aplica las relaciones teóricas y experimentales necesarias para el análisis del comportamiento de la bobina Tesla, entregando resultados coherentes y listos para su posterior visualización.

Decisión de diseño: el Modelo se define como una capa independiente del hardware y de la interfaz de usuario, lo que permite validar, reutilizar y probar la lógica de cálculo del sistema utilizando distintas fuentes de datos, sin modificar su implementación.

6.3. Controlador: rol y estructura

El módulo `controller/` implementa la capa **Controlador** del patrón MVC. Su responsabilidad es coordinar la adquisición de datos desde el microcontrolador, transformar el formato de entrada a estructuras utilizables por el Modelo y orquestar el flujo lógico del experimento (inicio, adquisición, detención y distribución de resultados hacia la Vista).

Elementos principales implementados/definidos:

- **Definición del controlador** (`controller.py`)
  - Estado actual: definido a nivel de arquitectura (coordinación MVC) y en implementación parcial para la integración final.
  - Responsabilidades objetivo: gestión de flujo del experimento, envío de comandos a la ESP32 y distribución de datos hacia el Modelo/Vista.
- **Decodificación de tramas** (`decodificador.py`)
  - `decodificar_linea_data()`: interpreta líneas del tipo `DATA,...` y genera una `MuestraCruda` para el Modelo.
- **Fuentes de datos** (`fuentes.py`)
  - `FuenteDatos`: interfaz base con método `leer_muestra()`.
  - `FuenteSerialESP32`:
    - **Atributos:** `puerto`, `baudrate`, `timeout_s`, `_ser`.
    - **Métodos:** `conectar()`, `cerrar()`, `enviar_comando()`, `set_servo_deg()`, `start_stream()`, `stop_stream()`, `leer_muestra()`.
  - `FuenteSimulada`:
    - **Atributos:** `t_ms`, `dt_ms`, `servo_deg`, `v_div_base`, `v_rf_base`, `v_photo_base`, `ruido_v_div`, `ruido_v_rf`, `ruido_v_photo`.
    - **Métodos:** `set_servo_deg()`, `leer_muestra()`.
  - `FuenteCSV`:
    - **Atributos:** `ruta_csv`, `path`, `_archivo`, `_reader`.
    - **Métodos:** `leer_muestra()`, `cerrar()`.

El módulo Controlador constituye el eje central de la arquitectura MVC, ya que es el encargado de coordinar el flujo de información entre el hardware, el Modelo y la Vista. Su rol principal es recibir las mediciones provenientes del sistema físico, gestionar el ciclo lógico del experimento y distribuir los datos de forma ordenada hacia las demás capas del software. En el estado actual del proyecto, el Controlador se encuentra definido a nivel de arquitectura y con una implementación parcial, quedando pendiente su integración completa y la gestión final del flujo del experimento.

Decisión de diseño: se definen múltiples fuentes de datos (ESP32, simulación y CSV) para permitir el desarrollo y prueba del software sin dependencia directa del hardware, manteniendo un flujo de información consistente hacia el Modelo.

6.4. Vista: rol y estructura

El módulo `view/` implementa la capa **Vista** del patrón MVC. Su función es proporcionar la interfaz de usuario del sistema, permitiendo la configuración de parámetros del experimento y la visualización de los datos procesados, sin ejecutar lógica de adquisición ni procesamiento.

Estructura del módulo:

- `vista_streamlit.py`: implementación de la interfaz gráfica utilizando la librería Streamlit.

Elementos principales definidos:

- **Clase** `TeslaView`
  - **Responsabilidad general:** renderizar la interfaz gráfica, recibir interacciones del usuario y delegar acciones al Controlador.
  - **Métodos relevantes:**
    - `__init__()`: inicializa la configuración de la aplicación y gestiona la persistencia del controlador mediante el estado de sesión.
    - `mostrar_interfaz()`: define la estructura principal de la interfaz, incluyendo panel lateral, pestañas, controles y visualizaciones.
  - **Componentes de interfaz:**
    - Selección de fuente de datos (simulación o ESP32).
    - Configuración de puerto serial.
    - Control manual del ángulo del servomotor.
    - Botones de inicio, detención y reinicio del experimento.
    - Visualización de métricas instantáneas.
    - Gráficos comparativos teórico–experimental.
    - Tabla de historial y exportación de datos.

En su estado actual, la Vista se encuentra implementada a nivel de estructura y componentes gráficos, alineándose con los mockups definidos en el Sprint 1. Su funcionamiento completo depende de la integración con el módulo Controlador, razón por la cual esta capa se mantiene en desarrollo dentro del enfoque modular adoptado en el proyecto.

Decisión de diseño: la Vista se limita a la interacción con el usuario y a la visualización de resultados, evitando incorporar lógica de adquisición o procesamiento y manteniendo una separación clara de responsabilidades dentro de la arquitectura MVC.

7. Integración software–hardware

En esta sección se describe la forma en que el sistema de hardware se integra con el software desarrollado, detallando la arquitectura general de interacción, el rol del firmware cargado en la ESP32, el mecanismo de comunicación con el software en Python y el flujo completo de datos desde la adquisición hasta el almacenamiento y visualización.

7.1. Arquitectura Sensor–Controlador–Actuador

El sistema se estructura siguiendo una arquitectura del tipo Sensor–Controlador–Actuador. Los sensores sensor RF y fotodiodo capturan las variables físicas asociadas al funcionamiento de la bobina Tesla, generando señales analógicas que son acondicionadas y posteriormente adquiridas por el microcontrolador ESP32 mediante su conversor analógico–digital (ADC).

El ESP32 actúa como controlador embebido, encargándose de la lectura periódica de los sensores, la medición del voltaje de alimentación de la bobina y la adquisición del ángulo del servomotor. El actuador del sistema corresponde al servomotor, cuyo ángulo puede ser ajustado desde el software para modificar la posición relativa del sensor RF respecto de la bobina.

La lógica de control del experimento y la coordinación de los distintos estados del sistema se delegan al software en Python, manteniendo el firmware de la ESP32 orientado exclusivamente a la adquisición y envío de datos.

7.2. Comunicación y flujo de datos del sistema

La comunicación entre la ESP32 y el software se realiza mediante un enlace serial USB, utilizando un protocolo de texto simple basado en comandos. El firmware cargado en la ESP32 implementa un mecanismo de control del flujo de datos, que permite habilitar o detener la transmisión de mediciones según las órdenes enviadas desde el software.

Durante la operación del sistema, el software establece la conexión serial y envía los comandos de configuración e inicio del experimento. Solo una vez recibido el comando correspondiente, la ESP32 comienza a muestrear los sensores y a transmitir los datos de forma periódica. De manera análoga, al recibir un comando de detención, el firmware interrumpe el envío de información, manteniendo el sistema en un estado controlado.

Los datos transmitidos corresponden a paquetes de información estructurados que incluyen las variables medidas por los sensores, el estado del actuador y marcas temporales asociadas. Estas líneas de datos son recibidas por el software, donde son decodificadas por el módulo Controlador y distribuidas hacia el Modelo para su procesamiento y análisis.

Este flujo de comunicación y datos permite que el control del experimento se centralice en el software, mientras que la ESP32 se mantiene enfocada en la adquisición y transmisión confiable de información. La separación entre adquisición, procesamiento y visualización facilita la validación del sistema y la integración progresiva de los distintos módulos.

8. Evidencia de avance

En esta sección se presenta evidencia visual del estado actual del proyecto, con el objetivo de respaldar el avance alcanzado tanto en el desarrollo del hardware como del software. Las imágenes muestran el montaje físico del sistema, los circuitos de medición implementados y las interfaces desarrolladas para la visualización de datos.

8.1. Repositorio de código

El proyecto se desarrolla bajo control de versiones utilizando un repositorio Git, en el cual se mantiene separada la implementación del firmware, el software en Python y la documentación asociada. La estructura del repositorio refleja la arquitectura modular definida para el sistema.

8.2. Evidencia de hardware

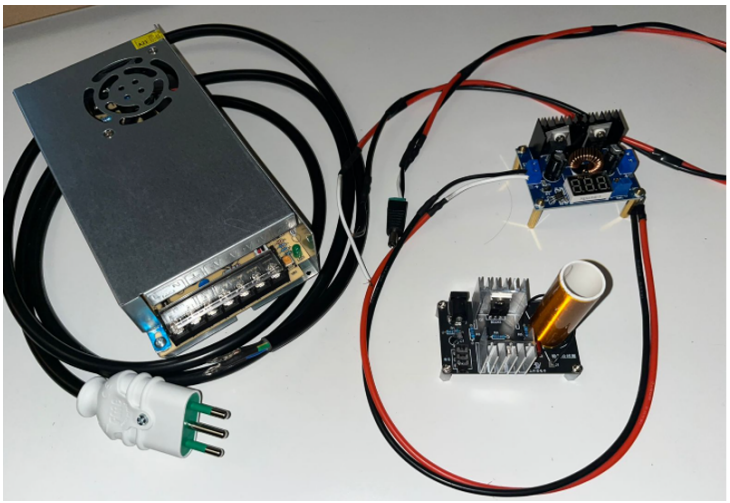

Figura 5: Sistema de alimentación y regulación de voltaje para la bobina Tesla.

La figura muestra la fuente de alimentación industrial y el regulador de voltaje utilizado para ajustar manualmente la tensión aplicada a la bobina Tesla, etapa clave para el control de la potencia del sistema.

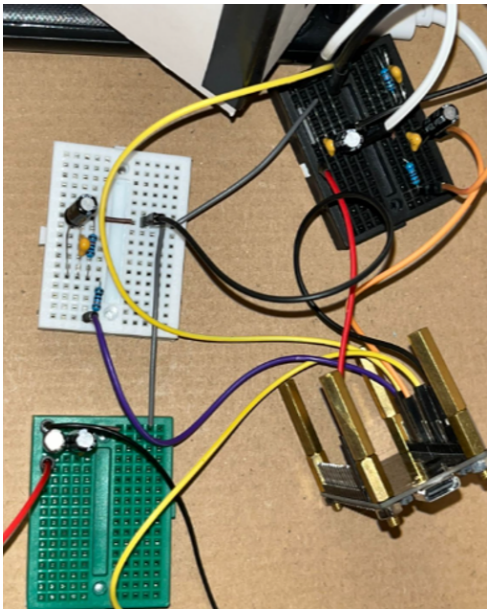

Figura 6: Circuitos de medición y acondicionamiento de señales implementados en protoboard.

Se presentan los circuitos de medición correspondientes al divisor de tensión, alimentación del servomotor y adquisición de señales del sensor RF y fotodiodo, incluyendo capacitores de desacople y resistencias de protección.

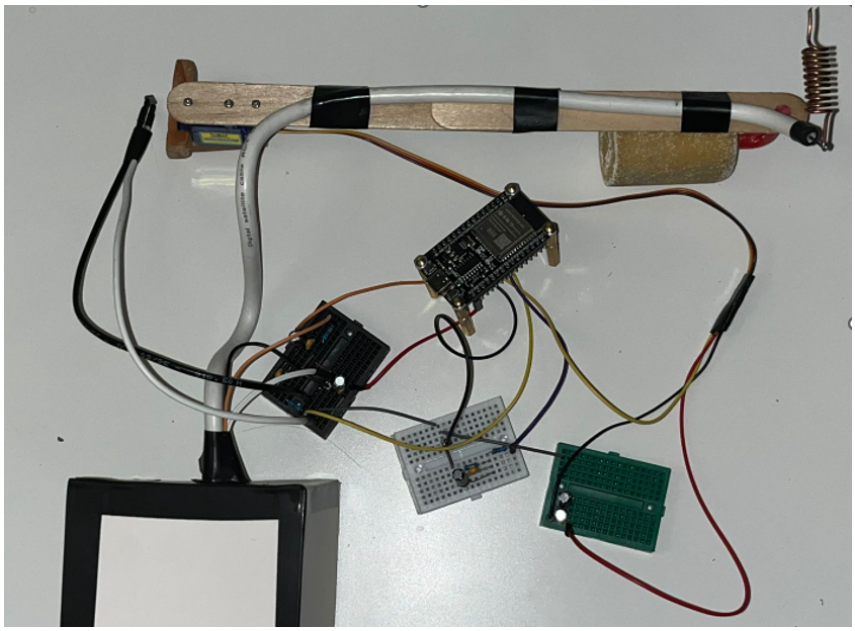

Figura 7: Montaje del sistema de monitoreo con ESP32 y módulos de adquisición.

La imagen evidencia el montaje físico del sistema de monitoreo, incluyendo el microcontrolador ESP32 y los módulos asociados a la adquisición de señales.

8.3. Capturas de ejecución del software

Figura 8: Interfaz gráfica del sistema: configuración y validaciones previas.

La interfaz permite configurar parámetros del experimento, verificar condiciones previas de seguridad y establecer los valores iniciales de operación.

Figura 9: Visualización gráfica de variables medidas durante el experimento.

Se muestra la visualización de variables como intensidad de campo en función del tiempo, generada a partir de los datos adquiridos por el sistema.

Figura 10: Historial de datos y exportación de resultados del experimento.

La interfaz permite acceder al historial de mediciones y exportar los resultados en formato CSV para su análisis posterior.

Adicionalmente, el código fuente y los archivos asociados al desarrollo del sistema se encuentran disponibles en un repositorio GitHub, el cual contiene la estructura completa del proyecto, versiones de implementación y material de soporte para su revisión y trazabilidad del avance [6].

9. Conclusiones

En el presente informe se ha descrito el estado de avance del sistema de validación experimental de una bobina Tesla de baja potencia, desarrollado como un proyecto interdisciplinario de integración hardware–software. Al cierre de este sprint, el proyecto presenta un avance global superior al 50 %, reflejando un desarrollo coherente con la planificación establecida y con el enfoque incremental adoptado.

Desde el punto de vista del hardware, la maqueta física, los circuitos de medición, el acondicionamiento de señales y las consideraciones de seguridad se encuentran prácticamente completos, quedando pendientes principalmente pruebas finales de validación bajo condiciones reales de operación. El sistema permite la adquisición de las variables relevantes del experimento, incluyendo el voltaje de alimentación, la señal del sensor de campo electromagnético, la intensidad lumínica de la descarga y la posición del servomotor.

En cuanto al software, se ha definido e implementado una arquitectura modular basada en el patrón Modelo–Vista–Controlador (MVC), que permite separar claramente las responsabilidades de adquisición, procesamiento y visualización de datos. El módulo Modelo presenta un alto grado de desarrollo e integra el conjunto de ecuaciones y relaciones matemáticas utilizadas para el análisis del comportamiento de la bobina Tesla. La Vista se encuentra implementada a nivel estructural y de componentes gráficos, mientras que el Controlador ha sido definido a nivel de arquitectura y se encuentra en una etapa de implementación parcial, a la espera de su integración completa con el hardware.

El desarrollo alcanzado en este sprint ha permitido consolidar una base sólida para la validación experimental del sistema, priorizando el análisis de tendencias físicas y el comportamiento relativo de las magnitudes medidas, más que la obtención de valores absolutos calibrados. Este enfoque resulta coherente con el alcance del proyecto y con las decisiones de diseño adoptadas. Las etapas futuras del trabajo se orientarán a la integración completa del sistema software–hardware, la ejecución de pruebas funcionales y experimentales, y la validación final del desempeño del sistema en su conjunto.

Referencias

[1] Sommerville, I., *Software Engineering*. Pearson, 10th ed., 2016.

[2] Espressif Systems, *ESP32 Series Datasheet*. Documento técnico del fabricante, s.f. Disponible en: `https://documentation.espressif.com/esp32-wroom-32_datasheet_en.pdf`.

[3] Analog Devices, *AD8307 Datasheet: DC to 500 MHz Logarithmic Amplifier/Detector*. Documento técnico del fabricante, s.f. Disponible en: `https://www.analog.com/media/en/technical-documentation/data-sheets/ad8307.pdf`.

[4] OSRAM Opto Semiconductors, *BPW34 Datasheet: Silicon PIN Photodiode*. Documento técnico del fabricante, s.f. Disponible en: `https://www.alldatasheet.com/datasheet-pdf/view/26251/VISHAY/BPW34.html`.

[5] Streamlit, *Streamlit Documentation*. Documentación oficial, s.f. Disponible en: `https://docs.streamlit.io/develop/api-reference/charts/st.line_chart`.

[6] Daanniieell2025 (GitHub), *Repositorio del proyecto: ING.-SOFTWARE*. Repositorio de código, s.f. Disponible en: `https://github.com/Daanniieell2025/ING.-SOFTWARE`. Consultado el: 07-01-2026.