



# SISTEMA DE VALIDACIÓN DE RENDIMIENTO DE UNA BOBINA TESLA

Ingeniería de Software Interdisciplinaria - Sprint 2

Profesores : Daniel Gacitúa / Ricardo Hasbún

Fecha: 07/01/2026

Integrantes: Alejandro Alarcón

Pablo Canales

Daniel González

Matias Kohl





# Índice

1. Introducción
2. Objetivos
3. Arquitectura del sistema
4. Funcionamiento del Hardware
5. Flujo de datos
6. Estructura modular del Software
7. Ecuaciones del sistema
8. Demostración funcional
9. Especificaciones de requisitos
10. Sprint Backlog
11. Prototipo Maqueta
12. Hardware real
13. Estructura del Repositorio
14. Conclusión



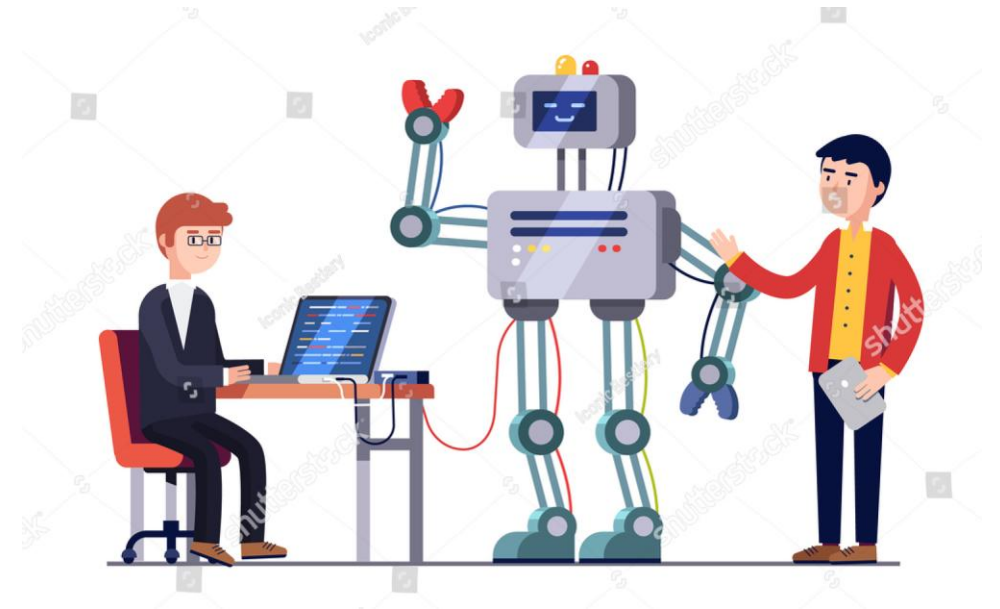
# 1. Introducción

## Contexto

El proyecto se enmarca en el desarrollo de un sistema mecatrónico basado en el esquema Sensor–Controlador–Actuador, integrando hardware y software, y aplicando Programación Orientada a Objetos en Python bajo el patrón MVC, con el apoyo de metodologías ágiles.

## Enfoque del sprint 2

- En el Sprint 1 se diseñó la arquitectura, backlog y SRS.
- En este Sprint 2 se comenzó con la implementación real del sistema.
- Se trabajó en el código, implementación del Hardware y pruebas iniciales.
- Se implementó cada módulo del software basado en arquitectura MVC





## 2. Objetivos del sprint 2

### Objetivo general

Implementar y validar las primeras funciones operativas del sistema de medición de la bobina.

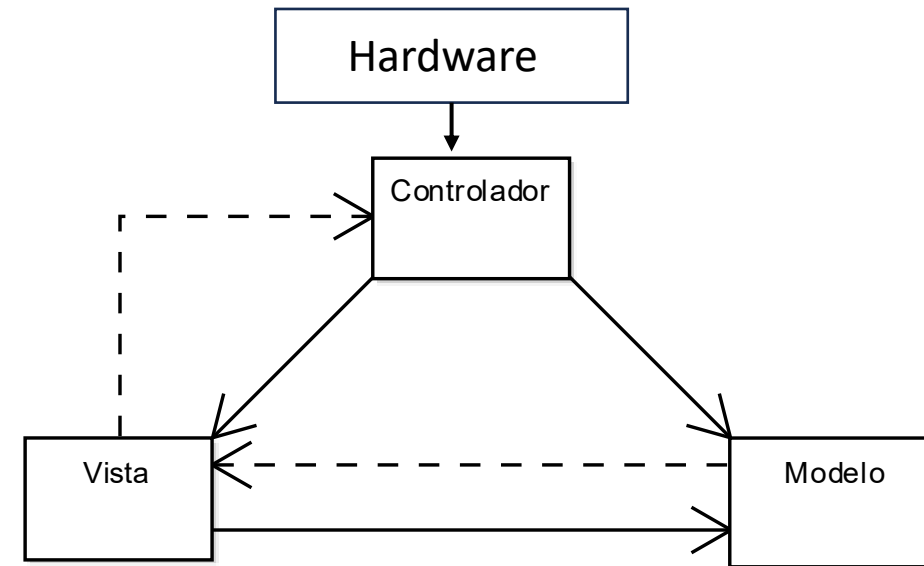
### Objetivos específicos

- Integrar sensores al microcontrolador.
- Desarrollar comunicación hardware-software.
- Documentar avances y dificultades.
- Validar la arquitectura MVC en código real.

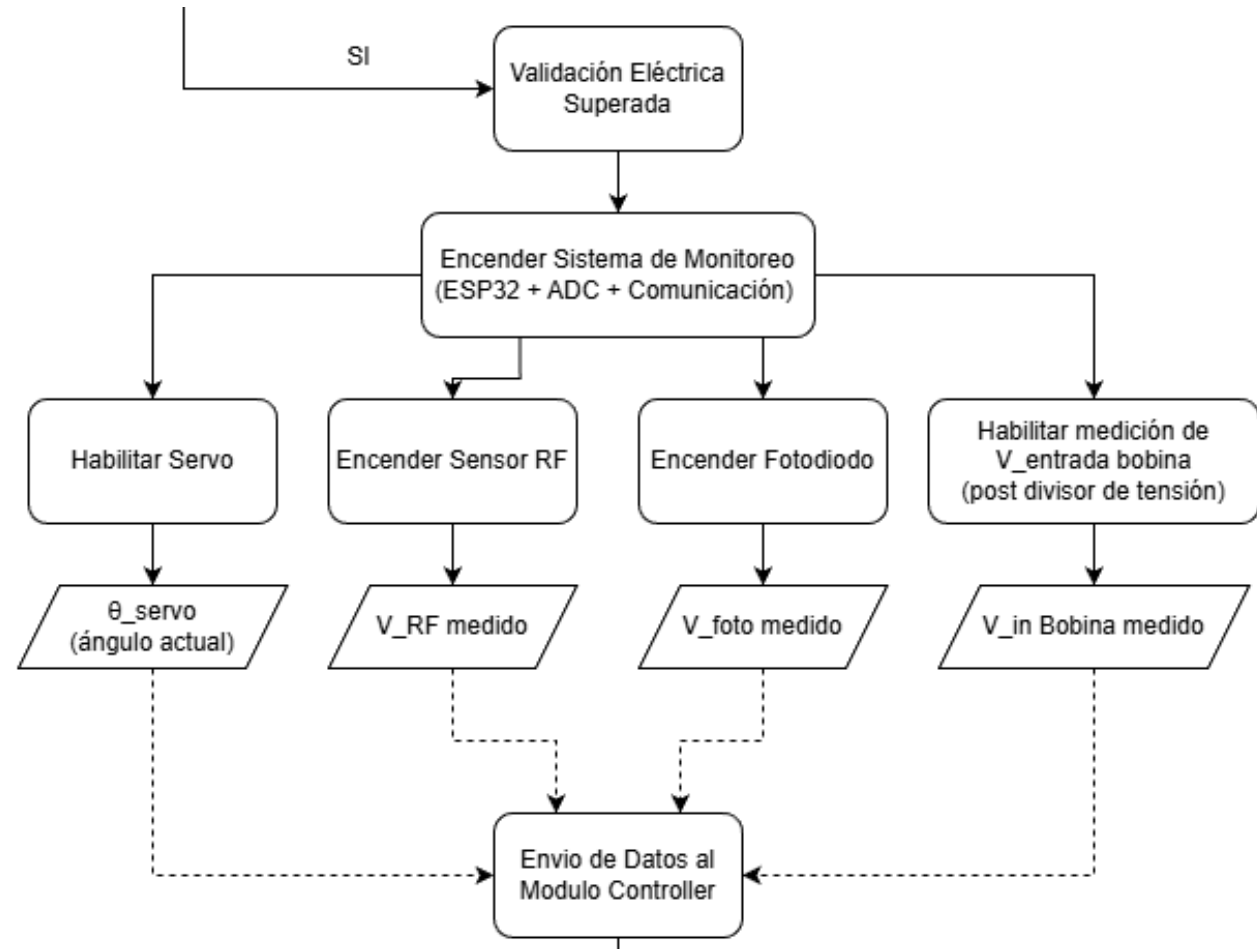
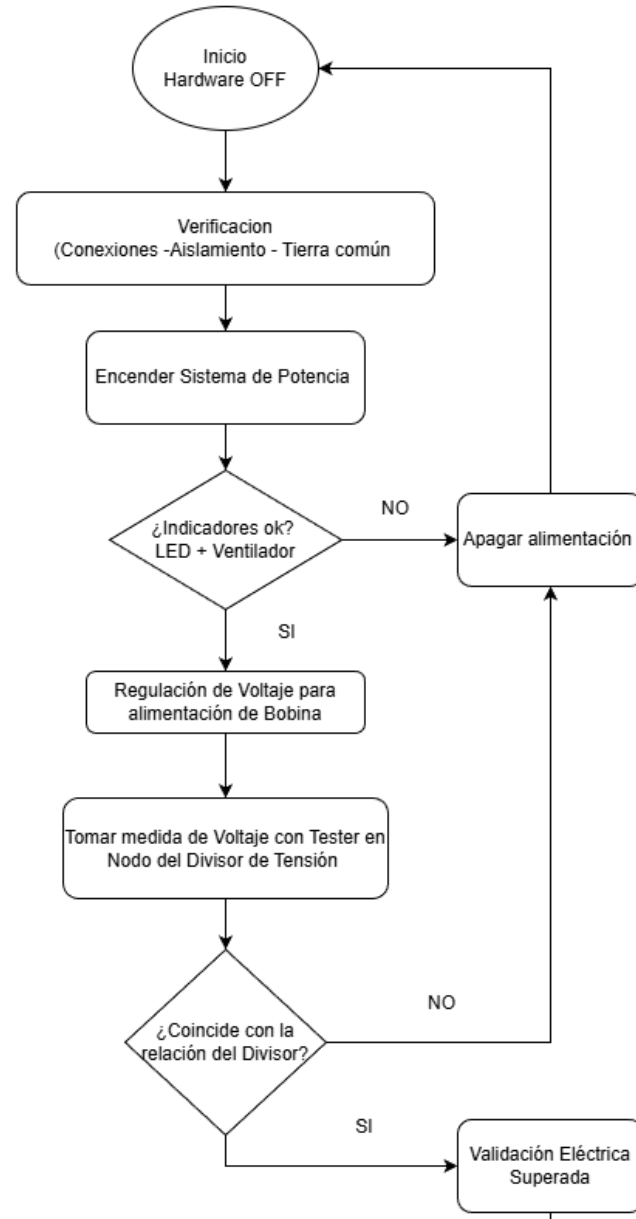


# 3. Arquitectura del sistema

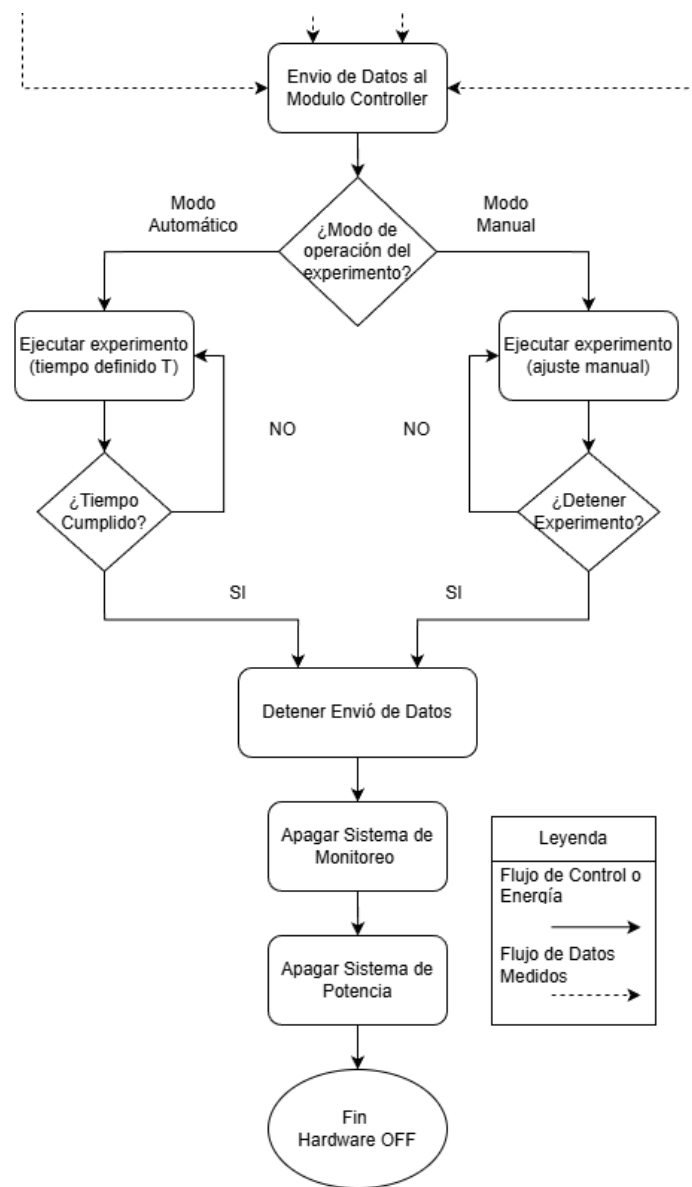
- **Modelo:** Procesa los datos crudos de los sensores.
- **Vista:** Muestra gráficos en tiempo real.
- **Controlador:** Gestiona la comunicación entre el usuario y el sistema.



# 4. Funcionamiento del hardware

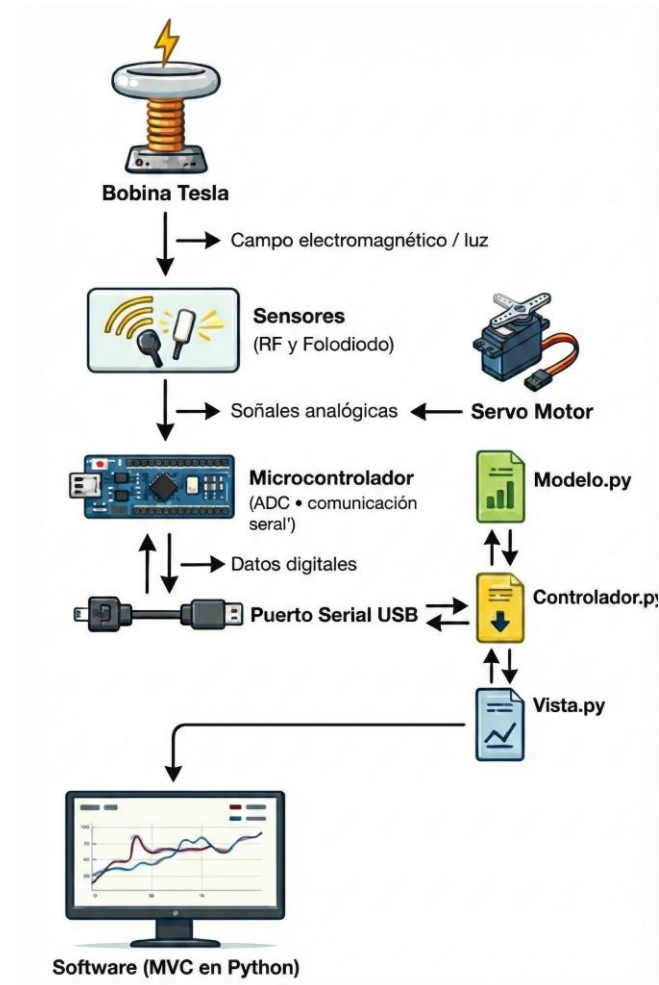


# Funcionamiento del hardware



## 5. Flujo de datos: Hardware → Software

1. Sensores → señales analógicas
2. Microcontrolador (ESP) las convierte a digitales
3. Se envían por puerto serial (USB)
4. Software en Python lee los datos, los procesa y los grafica







## 6. Estructura modular del software

### Vista – Interfaz del sistema

La vista se encarga de mostrar los datos procesados a través de gráficos y valores numéricos.

- Implementada en Python utilizando Streamlit
- Permite visualizar gráficos
- Incluye controles de usuario
- Diseñada para mantener una interfaz clara, funcional y centrada en el usuario técnico

Class: Vista

- Metodos:
  - init
  - mostrar\_interfaz





# Estructura modular del software

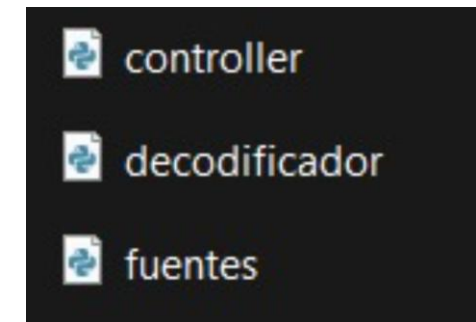
## Controlador – Lógica de conexión

El controlador actúa como puente entre los sensores físicos, el modelo de datos y la vista

- Recibe los datos desde el microcontrolador
- Interpreta comandos desde la vista
- Envía instrucciones al microcontrolador vía puerto serial

Class: FuenteSerialESP32

- Metodos:
  - Init
  - conectar
  - cerrar
  - enviar\_comando





# Estructura modular del software

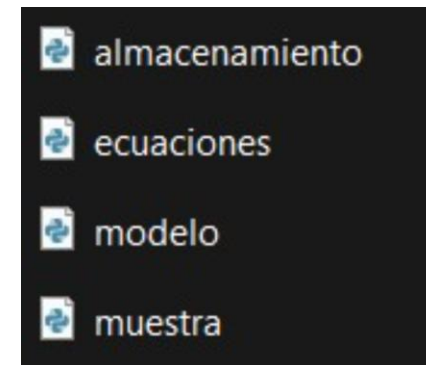
## Modelo – Procesamiento de datos

El modelo es el responsable del procesamiento de las mediciones

- Convierte los voltajes recibidos a intensidades físicas reales
- Calcula errores teóricos vs medidos
- Guarda los datos en archivos
- Opera de forma independiente a la interfaz

Class: Modelo

- Metodos:
  - Init
  - procesar\_muestra
  - reset
  - get\_historial





## 7. Ecuaciones del sistema

Distancia del loop a la bobina en función del ángulo theta en radianes

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

Voltaje de alimentación de la bobina después de la transformación del divisor

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

Intensidad de campo electromagnético e Intensidad Lumínica

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

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

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

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

Potencia de bobina

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





## 8. Demostración funcional

Configuraciones del sistema

Validacion previa (manual)

☒ Conexiones / Aislamiento / Tierra comun (OK)

☒ Indicadores OK (LED + Ventilador)

☒ Divisor de tension verificado con tester

Tiempo

Paso de tiempo por muestra [s]

0,50 - +

Duracion del experimento [s]

20,00 - +

Voltaje de alimentacion

Voltaje de alimentacion inicial [V]

15,00 - +

Deploy

### Simulacion Bobina Tesla - Modo continuo

Campo = intensidad de campo. Luz = intensidad luminica. Se guardan voltajes de sensores y potencia relativa.

Estado del sistema ☒ Grafico Historial de datos

Entrada (se puede cambiar mientras corre)

Angulo del servo [deg]

30,00 - +

	Métrica	Valor
0	Muestra #	40
1	Tiempo	19.5 s
2	Ángulo	30.0 °
3	Distancia	0.312 m
4	Voltaje Alimentación	23.77 V
5	Voltaje RF	1.728 V
6	Voltaje Fotodiodo	0.453 V
7	Potencia Relativa	0.97
8	Intensidad Campo	0.524





# Demostración funcional





# Demostración funcional

Configuraciones del sistema

Validacion previa (manual)

☒

 Conexiones / Aislamiento / Tierra comun (OK)

☒

 Indicadores OK (LED + Ventilador)

☒

 Divisor de tension verificado con tester

Tiempo

Paso de tiempo por muestra [s]

0,50 - +

Duracion del experimento [s]

20,00 - +

Voltaje de alimentacion

Voltaje de alimentacion inicial [V]

15,00 - +

Deploy

Demo finalizada. Guardado en: C:\Users\ptoma\OneDrive\Escritorio\2-2025\Ingeniería de Software\PROYECTO\Tesla\src\datos\_experimento.csv

## Simulacion Bobina Tesla - Modo continuo

Campo = intensidad de campo. Luz = intensidad luminica. Se guardan voltajes de sensores y potencia relativa.

Estado del sistema ☒ Grafico Historial de datos

Descargar CSV

ultimo\_guardado=C:\Users\ptoma\OneDrive\Escritorio\2-2025\Ingeniería de Software\PROYECTO\Tesla\src\datos\_experimento.csv





## 9. Especificación de requisitos (SRS)

### Requisitos funcionales

ID	Descripción
RF-01	El sistema debe leer datos del sensor.
RF-02	Visualización grafica en tiempo real (V vs t).
RF-03	Almacenamiento de mediciones en archivo CSV.
RF-04	Control manual del ángulo del servomotor.

### Requisitos no funcionales

ID	Descripción
RNF-01	Legibilidad del código.
RNF-02	Escalabilidad del sistema.
RNF-03	Interfaz intuitiva para usuarios.





# 10. Sprint backlog

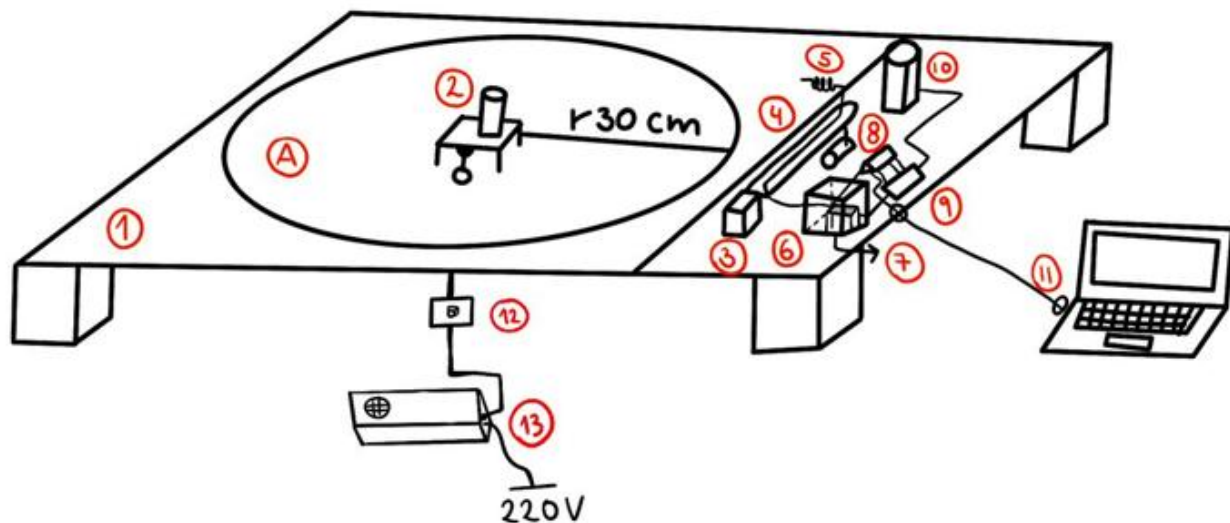
## Sprint 2:

• Construcción física del hardware	→	80%
• Integración del microcontrolador con sensores	→	100%
• Pruebas físicas del sistema de medición	→	85%
• Calibración de sensores RF y fotodiodo	→	100%
• Programación de Vista	→	70%
• Programación de Modelo	→	90%
• Programación de Controlador	→	50%

## Porcentaje de avance



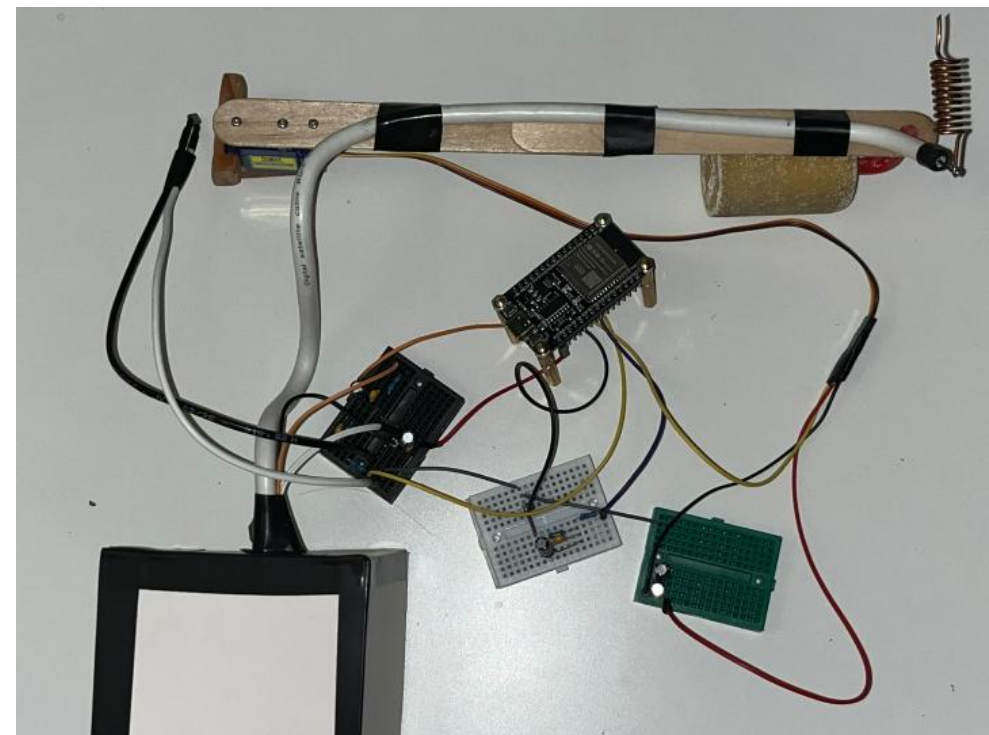
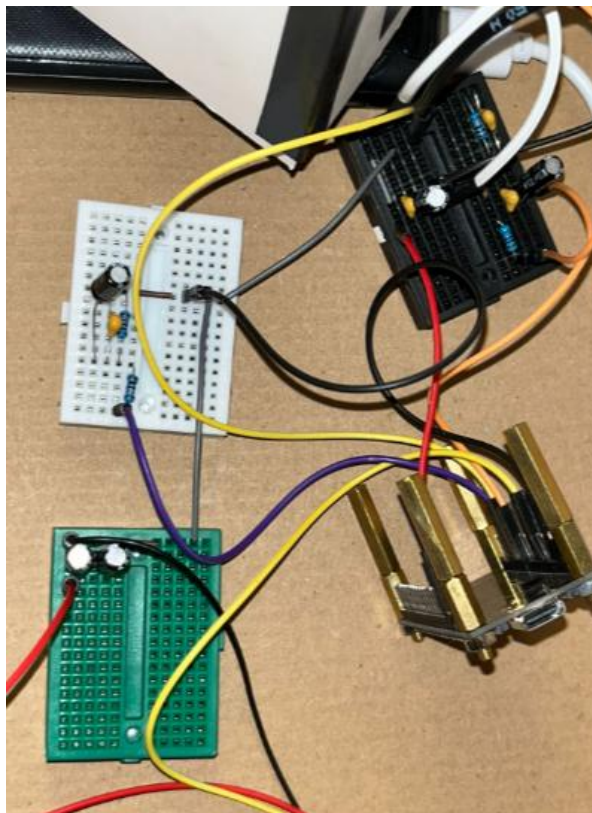
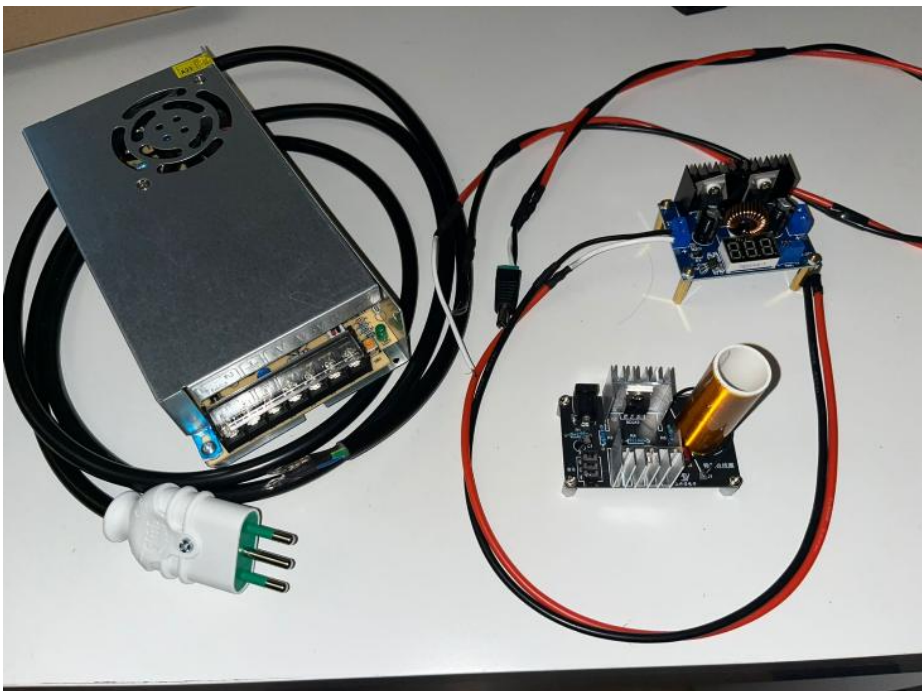
# 11. Prototipo Maqueta



1. Base de madera
  2. Bobina Tesla
  3. Servo motor
  4. Brazo móvil
  5. Loop
  6. Jaula de Faraday
  7. Sensor RF
  8. ESP32
  9. Circuitos
  10. Fotodiodo
  11. Puerto serial
  12. Regulador de voltaje
  13. Fuente de alimentación
- A. Zona de seguridad eléctrica



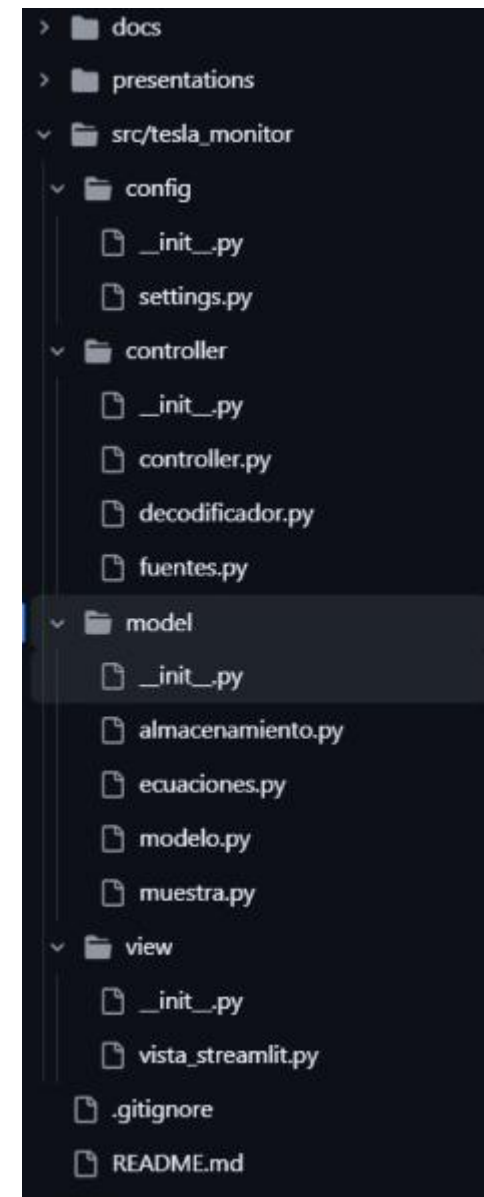
## 12. Hardware real





# 13. Estructura del Repositorio

<https://github.com/Daanniieell2025/ING.-SOFTWARE>



# Conclusiones

- Se realizó la estructura de funcionamiento de módulos
- Se llegó a un avance en la programación de aprox un 60%
- Hardware modular listo
- Se logró llegar hasta el 50% del proyecto





# Gracias por su atención



## SISTEMA DE VALIDACIÓN DE RENDIMIENTO DE UNA BOBINA TESLA

Ingeniería de Software Interdisciplinaria - Sprint 2

Profesores : Daniel Gacitúa / Ricardo Hasbún

Fecha: 07/01/2026

Integrantes: Alejandro Alarcón

Pablo Canales

Daniel González

Matias Kohl

