



# Videojuego portátil inspirado en consolas retro

Autor:

Lic. Jezabel Victoria Danon

Director:

Mg. Ing. Hanes Nahuel Sciarrone (FIUBA)

*Esta planificación fue realizada en el curso de Gestión de proyectos  
entre el 29 de abril de 2025 y el 18 de junio de 2025.*

## Índice

1. Descripción técnica-conceptual del proyecto a realizar . . . . .	5
2. Identificación y análisis de los interesados . . . . .	6
3. Propósito del proyecto . . . . .	7
4. Alcance del proyecto . . . . .	7
5. Supuestos del proyecto. . . . .	8
6. Requerimientos . . . . .	8
7. Historias de usuarios ( <i>Product backlog</i> ). . . . .	9
8. Entregables principales del proyecto . . . . .	13
9. Desglose del trabajo en tareas . . . . .	13
10. Diagrama de Activity On Node. . . . .	16
11. Diagrama de Gantt . . . . .	17
12. Presupuesto detallado del proyecto . . . . .	18
13. Gestión de riesgos . . . . .	19
14. Gestión de la calidad . . . . .	21
15. Procesos de cierre . . . . .	24

## Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	29 de abril de 2025
1	Se completa hasta el punto 5 inclusive	10 de mayo de 2025
2	Se completa hasta el punto 9 inclusive Agrega usuario final Cambio de cliente Cambio de título del proyecto	19 de mayo de 2025
3	Se completa hasta el punto 12 inclusive Modificación del formato de las historias de usuario Reducción de horas del proyecto	27 de mayo de 2025
4	Se completa el plan Se agrega el nombre del director Corrección de errores	3 de junio de 2025
5	Corrección del punto 14	6 de junio de 2025

## Acta de constitución del proyecto

Buenos Aires, 29 de abril de 2025

Por medio de la presente se acuerda con la Lic. Jezabel Victoria Danon que su Trabajo Final de la Carrera de Especialización en Sistemas Embebidos se titulará “Videojuego portátil inspirado en consolas retro” y consistirá en el desarrollo de un prototipo de consola de videojuegos portátil minimalista con un único juego integrado. El trabajo tendrá un presupuesto preliminar estimado de 640 horas y un costo estimado de € 13.086 (euros trece mil ochenta y seis), con fecha de inicio el 29 de abril de 2025 y fecha de presentación pública el 15 de diciembre de 2025.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg  
Director posgrado FIUBA

Lic. Jezabel Victoria Danon  
Proyecto personal

Mg. Ing. Hanes Nahuel Sciarrone  
Director del Trabajo Final

## 1. Descripción técnica-conceptual del proyecto a realizar

El proyecto responde a una temática de interés personal y tiene como objetivo principal acreditar los conocimientos obtenidos en el postgrado. Esto implica la utilización de diversos módulos de hardware y la implementación de técnicas de ingeniería de software específicas para sistemas embebidos. Como objetivo secundario, se planteó que el área de aplicación seleccionada no requiriera del asesoramiento experto de terceros, para maximizar el enfoque en el uso autónomo de los contenidos de la carrera de especialización. También se consideró que el proyecto fuera viable para alguien sin experiencia previa en estos temas.

La solución propuesta es un sistema embebido que articula los conocimientos del posgrado, manteniendo una complejidad técnica abordable. El desarrollo incluye el uso coordinado de periféricos variados y protocolos de comunicación comunes en sistemas reales. Además, requiere diseñar una arquitectura de software clara, con módulos separados para entrada, salida y lógica de control del sistema. La aplicación final consiste en una consola de videojuegos portátil inspirada en dispositivos retro, con algunas mejoras funcionales propias de plataformas actuales.

Las primeras consolas portátiles de videojuegos, conocidas como *handheld consoles*, establecieron una lógica de diseño centrada en la simplicidad, la portabilidad y el uso eficiente de recursos. Dispositivos como Mattel Auto Race (1976) o Electronic Football (1977) usaban pantallas de LED y una mecánica de juego muy básica, basada en puntos luminosos. Más adelante, la serie Game & Watch (Nintendo, 1980) introdujo pantallas LCD y juegos integrados en hardware dedicado. La aparición de consolas como la Game Boy (Nintendo), la Atari Lynx y la Sega Game Gear, entre 1989 y 1990, permitió expandir estas ideas mediante cartuchos intercambiables, mayor calidad gráfica y audio mejorado, sin abandonar el enfoque de sistema cerrado y orientado exclusivamente al juego. Estas plataformas funcionaban con microprocesadores de 4, 8 o hasta 16 bits, sin sistemas operativos ni procesamiento paralelo, y su interacción se limitaba a botones físicos y salidas visuales y sonoras básicas.

Con el tiempo, las consolas *handheld* evolucionaron hacia arquitecturas más complejas, con mejores capacidades gráficas, pantallas retroiluminadas a color, sonido estéreo y almacenamiento digital. También incorporaron nuevas formas de interacción, como pantallas táctiles, sensores de movimiento y motores de vibración. Estas incorporaciones permitieron enriquecer la experiencia de juego sin perder la portabilidad ni la simplicidad de uso. Aunque no todas estas tecnologías se consolidaron como estándar en el ámbito portátil, abrieron nuevas posibilidades para la interacción física y sensorial entre el usuario y el dispositivo.

El sistema a desarrollar adoptará una arquitectura cerrada y específica, centrada en la ejecución de un único videojuego implementado directamente en el firmware del dispositivo. Al iniciarse, el juego permitirá controlar la simulación del vuelo de una aeronave a partir de la interacción con botones y un joystick analógico, los cuales enviarán señales al sistema en tiempo real. La respuesta del sistema se manifestará a través de una interfaz visual, junto con retroalimentación sonora y háptica asociada a distintos eventos del juego.

El prototipo incluirá sensores de movimiento que permitirán detectar variaciones de posición del dispositivo. La información capturada del sensor, los botones y el joystick será procesada para determinar parámetros de vuelo tales como velocidad, altitud, dirección y posición de la aeronave dentro de un mapa predefinido. Dichos parámetros permitirán actualizar la representación gráfica, sonora y táctil de la simulación.

En la figura 1 se presenta el diagrama en bloques del sistema, en el que se observa el microcontrolador que coordinará el funcionamiento de los distintos módulos: entradas (botones,

joystick, sensores), salidas (pantalla, audio, vibración) y almacenamiento externo. El estado del juego deberá guardarse en memoria no volátil, utilizando una EEPROM externa o una tarjeta SD, permitiendo su recuperación tras un reinicio. El prototipo se conectará a una fuente de alimentación portátil.

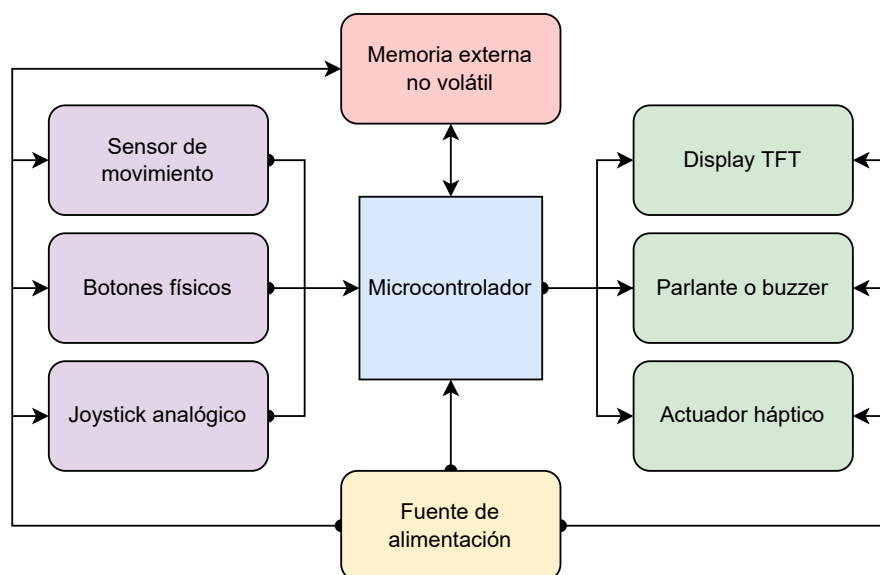


Figura 1. Diagrama en bloques del sistema.

El presente proyecto no se plantea como un producto de innovación con proyección comercial, pero se distingue por reinterpretar el concepto de consola portátil clásica desde una perspectiva actual. A partir de una arquitectura simple y dedicada, centrada en un único juego, se incorporan características técnicas poco comunes en dispositivos de este tipo, como el uso de un microcontrolador de 32 bits, pantalla a color, sensores de movimiento, retroalimentación háptica y almacenamiento persistente del estado del juego.

## 2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Cliente	Lic. Jezabel Victoria Danon	Proyecto personal	
Responsable	Lic. Jezabel Victoria Danon	FIUBA	Alumno
Orientador	Mg. Ing. Hanes Nahuel Sciarrone	FIUBA	Director del Trabajo Final
Usuario final	Entusiastas de sistemas embebidos o videojuegos retro		

- Cliente: al tratarse de un proyecto académico, se utilizará la figura de cliente a nombre del autor y responsable.

- Responsable: será el autor del presente documento, encargado del desarrollo del prototipo y del cumplimiento de los requerimientos pautados en la planificación.
- Usuario final: personas con conocimientos técnicos básicos o intermedios, interesados en sistemas embebidos y/o videojuegos retro.

### 3. Propósito del proyecto

Aplicar de forma integrada los conocimientos adquiridos durante el posgrado en una solución técnica concreta, desarrollada de manera autónoma. Se busca abordar un caso representativo de sistemas embebidos que permita ejercitar competencias clave de la carrera, como la gestión de periféricos, la programación en tiempo real y el diseño modular de software.

### 4. Alcance del proyecto

El proyecto incluye:

- El diseño y construcción de un prototipo funcional de consola portátil de videojuegos.
- El desarrollo de un juego de simulación de vuelo, implementado directamente en el firmware.
- La integración de periféricos de entrada:
  - Botones físicos.
  - Joystick analógico.
  - Sensor de movimiento (acelerómetro y/o giroscopio).
- La integración de periféricos de salida:
  - Pantalla TFT (a color).
  - Salida sonora mediante parlante o buzzer.
  - Motor de vibración para retroalimentación háptica.
- El almacenamiento del estado del juego en memoria no volátil.
- La documentación técnica requerida para su presentación como trabajo final de especialización.

El presente proyecto no incluye:

- El diseño y/o la fabricación de una placa de circuito impreso (PCB, por *Printed Circuit Board*).
- El desarrollo de un entorno ejecutable separado del juego: el firmware no estará diseñado de forma general para admitir juegos en formatos estándar.
- Conectividad externa, comunicación inalámbrica o funcionalidades multijugador.

## 5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- Se dispondrá de los materiales y componentes necesarios para el prototipo funcional.
- Las horas estimadas de trabajo serán suficientes para cumplir con los objetivos planteados.
- Los conocimientos requeridos para avanzar en las distintas etapas del proyecto se alcanzarán durante el transcurso del posgrado en tiempos compatibles con el desarrollo.
- La solución planteada es técnicamente viable dentro del alcance y recursos definidos.
- El responsable dispondrá de dedicación a tiempo completo hasta culminar el proyecto.

## 6. Requerimientos

Los requerimientos del proyecto son los siguientes:

### 1. Requerimientos de hardware:

- 1.1. El prototipo deberá montarse en *protoboard* o soldarse sobre placas de experimentación.
- 1.2. El prototipo deberá alimentarse a través de una batería.
- 1.3. El prototipo deberá integrar un sensor capaz de medir la inclinación en al menos dos ejes.
- 1.4. El prototipo deberá integrar un joystick analógico con capacidad de lectura en dos ejes.
- 1.5. El prototipo deberá incluir botones tipo switch para funciones definidas.
- 1.6. El prototipo deberá incorporar una pantalla capaz de mostrar información visual relevante para el juego.
- 1.7. El prototipo deberá incluir un mecanismo para emitir sonidos o alertas sonoras.
- 1.8. El prototipo deberá integrar un motor de vibración capaz de generar retroalimentación háptica.
- 1.9. El prototipo deberá contar con memoria no volátil para el almacenamiento persistente del estado del juego.

### 2. Requerimientos de firmware:

- 2.1. El firmware deberá implementar la lógica para interpretar los datos del sensor de inclinación y el joystick para controlar el modelo de vuelo (cabeceo, alabeo, giro, velocidad).
- 2.2. El firmware deberá incluir un demo de juego integrado que permita demostrar la lógica implementada y la funcionalidad de los periféricos de entrada y salida.
- 2.3. El firmware deberá gestionar la visualización de la información del juego en la pantalla.
- 2.4. El firmware deberá ser capaz de reproducir las alertas sonoras necesarias.



- 2.5. El firmware deberá controlar la activación y duración del motor de vibración en función de los eventos del juego.
  - 2.6. El firmware deberá implementar la lógica para guardar y cargar el estado del juego en la memoria no volátil.
  - 2.7. El firmware deberá detectar las pulsaciones de los botones y activar las funcionalidades correspondientes (inicio/fin, pausa, guardar, mapa, etc...).
3. Requerimientos de documentación:
- 3.1. Diagrama de conexión de módulos.
  - 3.2. Video demostrativo del uso de las funcionalidades requeridas.
  - 3.3. Informe de avance.
  - 3.4. Memoria técnica.
4. Requerimientos opcionales:
- 4.1. El firmware podrá incluir gráficos a color para representar la interfaz de usuario (UI por sus siglas en inglés).
  - 4.2. El sistema podrá incluir uno o varios selectores de niveles para aspectos como: el volumen de las señales audio, el nivel de vibración de la consola, modos de visualización diferentes para debug o gráficos, entre otras opciones posibles.

## 7. Historias de usuarios (*Product backlog*)

Escalas utilizadas para asignar *story points*:

1. Complejidad:
  - Baja complejidad: 1 punto
  - Media complejidad: 3 puntos
  - Alta complejidad: 5 puntos
2. Cantidad de trabajo:
  - Poco trabajo: 1 punto
  - Cantidad de trabajo media: 2 puntos
  - Mucho trabajo: 3 puntos
3. Incertidumbre asociada:
  - Baja incertidumbre: 2 puntos
  - Incertidumbre media: 5 puntos
  - Alta incertidumbre: 8 puntos

**HU1: control intuitivo por inclinación.** Como usuario, quiero controlar el cabeceo (*pitch*) y el alabeo (*roll*) de la aeronave inclinando la consola, para tener una experiencia de control más natural e inmersiva.

- Complejidad media (3 p) porque requiere integrar el sensor de inclinación y procesar sus datos.
- Trabajo medio (2 p) porque requiere trabajo de hardware y software.
- Incertidumbre media (5 p) porque requiere evaluar si se requerirán filtros y de qué tipo para el procesamiento de los datos del sensor.

*Story points:*  $3+2+5=10$ , redondeo Fibonacci: 13 puntos.

**HU2: maniobras precisas con el joystick.** Como usuario, quiero usar el joystick analógico para controlar el giro (*yaw*) y la velocidad de la aeronave, para poder realizar maniobras precisas y ajustar mi trayectoria.

- Complejidad media (3 p) porque requiere integrar el joystick y mapear sus rangos.
- Trabajo medio (2 p) porque requiere trabajo de hardware y software.
- Incertidumbre baja (2 p) porque el módulo de joystick es simple de utilizar.

*Story points:*  $3+2+2=7$ , redondeo Fibonacci: 8 puntos.

**HU3: inicio y fin de sesión sencillos.** Como usuario, quiero poder iniciar el juego rápidamente al encender la consola y finalizarla fácilmente cuando termine de jugar para gestionar eficientemente mi tiempo de juego.

- Complejidad baja (1 p) porque es una funcionalidad simple una vez que el sistema esta funcionando.
- Trabajo bajo (1 p) porque requiere trabajo mayormente de software.
- Incertidumbre baja (2 p) ya que no requiere tomar decisiones específicas para su implementación.

*Story points:*  $1+1+2=4$ , redondeo Fibonacci: 5 puntos.

**HU4: información clara del estado.** Como usuario, quiero ver en pantalla información clave como mi velocidad, dirección y posición, para estar siempre informado sobre el estado de mi juego.

- Complejidad alta (5 p) porque requiere la configuración apropiada de la pantalla con sus métodos de graficación y la integración de diversas variables de estado del juego.
- Trabajo medio (2 p) ya que, una vez definidas las variables y el módulo gráfico, debería ser bastante trivial graficar dichas variables.
- Incertidumbre alta (8 p) porque aún no se encuentra definido el método con el que se renderizarán los gráficos en la pantalla.

*Story points:*  $5+2+8=15$ , redondeo Fibonacci: 21 puntos.

**HU5: visualizar ubicación en el mapa.** Como usuario, quiero ver la posición actual de la aeronave en el mapa, para tener una referencia de navegación.

- Complejidad alta (5 p) porque requiere la configuración apropiada de la pantalla con sus métodos de graficación y el diseño de la interfaz del mapa.
- Trabajo alto (3 p) ya que requiere graficar elementos varios definidos en el mapa además de la posición de la aeronave.
- Incertidumbre alta (8 p) porque aún no se encuentra definido el método con el que se renderizarán los gráficos en la pantalla ni el diseño de la interfaz gráfica.

*Story points:*  $5+3+8=16$ , redondeo Fibonacci: 21 puntos.

**HU6: sonidos y alertas.** Como usuario, quiero que el sistema emita sonidos o señales acústicas distintivas y relevantes durante el juego, para recibir retroalimentación auditiva.

- Complejidad media (3 p) porque requiere investigación sobre la implementación del audio.
- Trabajo medio (2 p) porque requiere trabajo de hardware y software.
- Incertidumbre baja (2 p) ya que se asume que el módulo seleccionado para audio será simple de utilizar.

*Story points:*  $3+2+2=7$ , redondeo Fibonacci: 8 puntos.

**HU7: generar vibración durante el juego.** Como usuario, quiero que el dispositivo genere vibración en ciertos eventos del juego, para tener una experiencia más inmersiva y física.

- Complejidad baja (1 p) ya que se utilizará un módulo de driver que debería resolver la mayor complejidad de la implementación.
- Trabajo medio (2 p) porque requiere trabajo de hardware y software.
- Incertidumbre media (5 p) porque aún no se encuentran definidos los eventos que generarán las vibraciones y porque la implementación involucra un módulo que no se utilizó anteriormente.

*Story points:*  $1+2+5=8$ , redondeo Fibonacci: 8 puntos.

**HU8: guardar el estado actual de la partida.** Como usuario, quiero poder guardar el estado del juego en curso, para retomar el progreso más adelante.

- Complejidad media (3 p) porque requiere almacenar la totalidad de la información necesaria para definir el estado de la partida.
- Trabajo alto (3 p) porque requiere bastante trabajo de hardware y software.
- Incertidumbre media (5 p) ya que aún no se han definido las variables a preservar ni el formato de almacenamiento.

*Story points:*  $3+3+5=11$ , redondeo Fibonacci: 13 puntos.

**HU9: elegir cómo empezar.** Como usuario, quiero poder seleccionar entre continuar una partida previamente guardada o descartarla e iniciar una nueva, para tener control sobre cómo empezar cada sesión de juego.

- Complejidad baja (1 p) ya que la mayor complejidad será abordada al definir cómo guardar y recuperar el estado.
- Trabajo medio (2 p) porque requiere trabajo de hardware y software.
- Incertidumbre baja (2 p) porque solo resta definir cómo se realizará el descarte de la partida anterior.

*Story points:* 1+2+2=5, redondeo Fibonacci: 5 puntos.

**HU10: pausar y retomar el juego.** Como usuario, quiero poder pausar la partida en cualquier momento para tomar un descanso y luego reanudarla exactamente donde la dejé.

- Complejidad baja (1 p) ya que solo debería implicar un periodo en el que no se actualice el estado del juego.
- Trabajo medio (2 p) porque implica actualizar en pantalla un mensaje que indique las acciones de pausado y reanudación de la partida.
- Incertidumbre baja (2 p) ya que solo resta definir la forma en la que se comunicará el estado pausado al usuario.

*Story points:* 1+2+2=5, redondeo Fibonacci: 5 puntos.

**HU11: entender las conexiones del hardware.** Como cliente, quiero disponer de un diagrama de conexión de los módulos para comprender la arquitectura física del prototipo.

- Complejidad baja (1 p).
- Trabajo bajo (1 p).
- Incertidumbre baja (2 p).

*Story points:* 1+1+2=4, redondeo Fibonacci: 5 puntos.

**HU12: ver el prototipo en acción.** Como cliente, quiero ver un video que muestre las funcionalidades del prototipo para comprobar el cumplimiento de lo solicitado.

- Complejidad baja (1 p).
- Trabajo medio (2 p).
- Incertidumbre baja (2 p).

*Story points:* 1+2+2=5, redondeo Fibonacci: 5 puntos.

**HU13: seguir el proceso de desarrollo.** Como cliente, quiero recibir al menos un informe de avance para entender la evolución del proyecto.

- Complejidad media (3 p).

- Trabajo medio (2 p).
- Incertidumbre baja (2 p).

*Story points:*  $3+2+2=7$ , redondeo Fibonacci: 8 puntos.

**HU14: comprender las decisiones técnicas.** Como cliente, quiero tener acceso a una memoria técnica para entender en detalle el diseño y la implementación del prototipo.

- Complejidad media (3 p).
- Trabajo alto (3 p).
- Incertidumbre media (5 p).

*Story points:*  $3+3+5=11$ , redondeo Fibonacci: 13 puntos.

## 8. Entregables principales del proyecto

Los entregables del proyecto son:

- Plan de trabajo.
- Memoria del proyecto.
- Prototipo funcional.
- Diagrama de conexiones.
- Código fuente.

## 9. Desglose del trabajo en tareas

### 1. Desarrollo del hardware. (34 h)

#### 1.1. Selección y adquisición de componentes. (4 h)

- 1.1.1. Confirmación de los componentes existentes y verificación de su idoneidad. (2 h)
- 1.1.2. Adquisición de componentes faltantes o adicionales. (2 h)

#### 1.2. Montaje del prototipo en *protoboard*. (12 h)

- 1.2.1. Planificación del diseño de montaje. (3 h)
- 1.2.2. Montaje y cableado del microcontrolador y la fuente de alimentación. (3 h)
- 1.2.3. Montaje y cableado del sensor de inclinación. (1 h)
- 1.2.4. Montaje y cableado del joystick analógico. (0,5 h)
- 1.2.5. Montaje y cableado de los botones. (0,5 h)
- 1.2.6. Montaje y cableado de la pantalla. (1 h)
- 1.2.7. Montaje y cableado del sistema de audio. (1,5 h)
- 1.2.8. Montaje y cableado del motor de vibración. (1,5 h)

1.3. Pruebas por componente en *protoboard*. (18 h)

- 1.3.1. Prueba de conexión del sensor de inclinación. (5 h)
- 1.3.2. Prueba de conexión del joystick analógico. (1 h)
- 1.3.3. Prueba de conexión de los botones. (1 h)
- 1.3.4. Prueba de conexión de la pantalla. (6 h)
- 1.3.5. Prueba de conexión del sistema de audio. (2 h)
- 1.3.6. Prueba de conexión del motor de vibración. (3 h)

2. Desarrollo del firmware. (436 h)

2.1. Configuración Inicial del entorno de desarrollo. (1 h)

- 2.1.1. Configuración de la placa, pines y periféricos. (1 h)

2.2. Implementación de la lógica del sistema. (117 h)

- 2.2.1. Desarrollo del módulo de adquisición de señales de entrada. (30 h)

- 2.2.2. Desarrollo del módulo de generación de señales de salida. (37 h)

- 2.2.2.1. Módulo de salida: sub-módulo de audio. (4 h)

- 2.2.2.2. Módulo de salida: sub-módulo de vibraciones. (8 h)

- 2.2.2.3. Módulo de salida: sub-módulo de gráficos. (25 h)

- 2.2.3. Investigación sobre distintos sistemas operativos de tiempo real (RTOS por sus siglas en inglés). (5 h)

- 2.2.4. Implementación del RTOS seleccionado. (20 h)

- 2.2.5. Diseño e implementación de una máquina de estados para el control general del sistema. (25 h)

2.3. Implementación del demo de juego. (55 h)

- 2.3.1. Diseño de la lógica del demo de juego. (25h)

- 2.3.2. Desarrollo de la máquina de estados del juego. (10 h)

- 2.3.3. Integración de la lógica de control con el demo de juego y la máquina de estados del juego. (20 h)

2.4. Detección de entradas simples. (11 h)

- 2.4.1. Implementación de la detección de pulsaciones de los botones. (3 h)

- 2.4.2. Implementación de la lectura de los datos del joystick analógico. (3 h)

- 2.4.3. Integración de las entradas de botones y joystick con la máquina de estados del sistema. (5 h)

2.5. Detección de inclinación. (18 h)

- 2.5.1. Implementación de la lectura de los datos del sensor de inclinación. (5 h)

- 2.5.2. Procesamiento de los datos del acelerómetro para obtener la inclinación. (3 h)

- 2.5.3. Integración de la detección de inclinación con la lógica del sistema. (10 h)

2.6. Gestión de la interfaz de usuario. (140 h)

- 2.6.1. Investigación y aprendizaje de conceptos básicos de graficación en la pantalla. (25 h)

- 2.6.2. Inicialización y configuración de la pantalla para la visualización. (20 h)

- 2.6.3. Implementación de la visualización del estado de la aeronave. (40 h)

- 2.6.4. Implementación de la visualización del mapa y la posición. (35 h)

- 2.6.5. Implementación de la lógica para mostrar/ocultar elementos de la UI. (20 h)

2.7. Implementación de la reproducción de audio. (18 h)

- 2.7.1. Investigación sobre modos de generación y/o reproducción de audio en sistemas embebidos. (5 h)
- 2.7.2. Implementación de la capacidad de reproducir audio. (8 h)
- 2.7.3. Lógica para activar los sonidos en eventos específicos del juego. (5 h)
- 2.8. Implementación del control de vibración. (33 h)
  - 2.8.1. Investigación sobre generación de vibración con distintos patrones. (8 h)
  - 2.8.2. Implementación de la capacidad de controlar el motor de vibración. (15 h)
  - 2.8.3. Lógica para activar la vibración en eventos específicos del juego. (10 h)
- 2.9. Gestión del estado del juego. (43 h)
  - 2.9.1. Definición de las variables del estado del juego a persistir. (8 h)
  - 2.9.2. Implementación de la lógica para guardar el estado del juego en memoria no volátil. (15 h)
  - 2.9.3. Implementación de la lógica para cargar el estado del juego desde memoria no volátil. (15 h)
  - 2.9.4. Implementación de la lógica para pausar y reanudar el juego. (5 h)
- 2.10. Integración del demo de juego. (Tarea continua)
- 3. Pruebas y verificación. (55 h)
  - 3.1. Pruebas de control por inclinación. (8 h)
  - 3.2. Pruebas de control por joystick. (5 h)
  - 3.3. Pruebas de inicio y Fin de sesión. (8 h)
  - 3.4. Pruebas de visualización del estado de la aeronave. (8 h)
  - 3.5. Pruebas de visualización del mapa. (8 h)
  - 3.6. Pruebas de audio. (3 h)
  - 3.7. Pruebas de vibración. (5 h)
  - 3.8. Pruebas de guardado y carga del estado. (5 h)
  - 3.9. Pruebas de pausa y reanudación. (5 h)
  - 3.10. Pruebas de integración. (Tarea continua)
- 4. Documentación. (115 h)
  - 4.1. Creación del diagrama de conexión de módulos. (15 h)
  - 4.2. Grabación y edición del video demostrativo. (20 h)
  - 4.3. Elaboración del informe de avance. (30 h)
  - 4.4. Elaboración de la memoria técnica. (50 h)
- 5. Tareas opcionales (se abordarán solo si el tiempo y los recursos lo permiten). (141 h)
  - 5.1. Implementación de gráficos a color. (78 h)
    - 5.1.1. Investigación de la capacidad de la pantalla. (9 h)
    - 5.1.2. Diseño de gráficos y paleta de colores. (20 h)
    - 5.1.3. Modificación de las rutinas de dibujo. (20 h)
    - 5.1.4. Actualización de los elementos visuales. (20 h)
    - 5.1.5. Pruebas de la visualización a color. (9 h)
  - 5.2. Implementación de selectores (por selector). (28 h)
    - 5.2.1. Diseño de la interfaz del selector. (5 h)

- 5.2.2. Implementación de la lógica de navegación del selector. (10 h)
- 5.2.3. Implementación de la lógica de aplicación del valor seleccionado. (8 h)
- 5.2.4. Pruebas del selector y su correcta funcionalidad. (5 h)
- 5.3. Montaje en placas de experimentación. (35 h)
  - 5.3.1. Diseño del layout para placas de experimentación. (10 h)
  - 5.3.2. Soldadura de los componentes en las placas de experimentación. (15 h)
  - 5.3.3. Pruebas de alimentación de los componentes. (2 h)
  - 5.3.4. Pruebas de continuidad y cortocircuitos. (3 h)
  - 5.3.5. Pruebas de funcionamiento de cada componente. (5 h)

Cantidad total de horas (sin contemplar opcionales): 640.

Cantidad de horas de ingeniería (sin contemplar opcionales ni documentación): 525.

## 10. Diagrama de Activity On Node

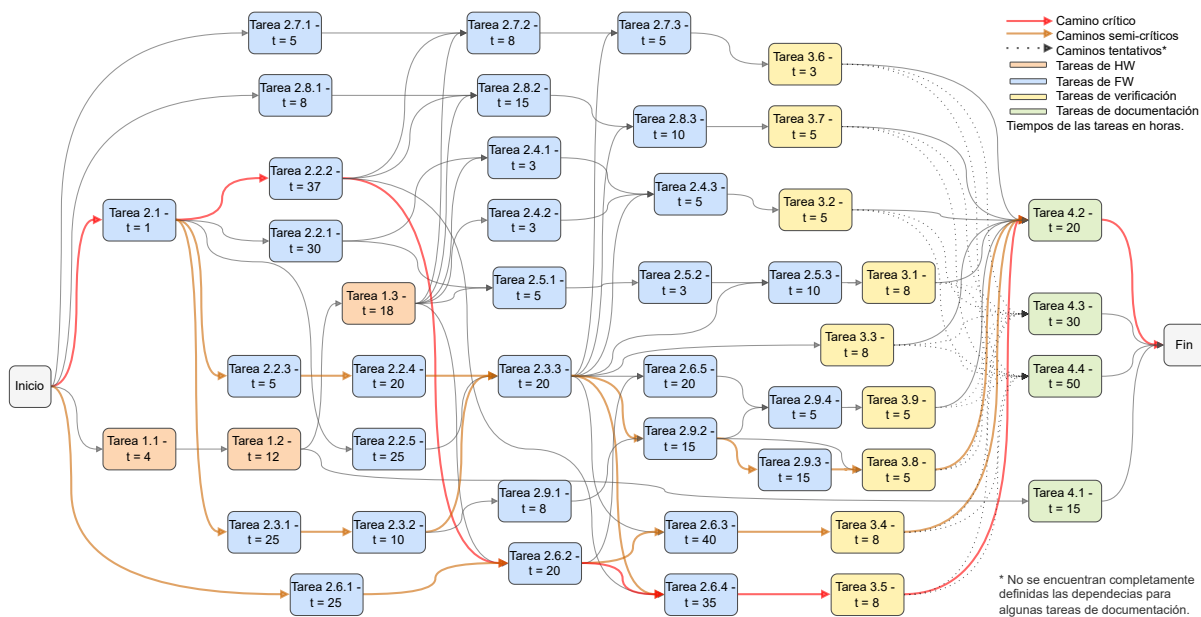


Figura 2. Diagrama de Activity on Node.

Camino Crítico:

- Inicio → Tarea 2.1 → Tarea 2.2.2 → Tarea 2.6.2 → Tarea 2.6.4 → Tarea 3.5 → Tarea 4.2 → Fin (121 h)

Caminos semi críticos:

- Inicio → Tarea 2.6.1 → Tarea 2.6.2 → Tarea 2.6.3 → Tarea 3.4 → Tarea 4.2 → Fin (113 h)
- Inicio → Tarea 2.1 → Tarea 2.3.1 → Tarea 2.3.2 → Tarea 2.3.3 → Tarea 2.9.2 → Tarea 2.9.3 → 3.8 → Tarea 4.2 → Fin (111 h)



- Inicio →Tarea 2.1 →Tarea 2.2.3 →Tarea 2.2.4 →Tarea 2.3.3 →Tarea 2.6.4 →Tarea 3.5 →4.2 →Fin (109 h)
- Inicio →Tarea 2.6.1 →Tarea 2.6.2 →2.6.4 →Tarea 3.5 →Tarea 4.2 →Fin (108 h)

## 11. Diagrama de Gantt

En la figura 3 se puede observar el diagrama de Gantt del proyecto. Se presentan las tareas de ingeniería hasta el segundo nivel de jerarquía de la estructura de desglose de trabajo (EDT) y las tareas de verificación y documentación hasta el primer nivel de jerarquía, a modo de facilitar la legibilidad del diagrama.

Para el cálculo de las fechas de inicio y fin se consideraron tanto las dependencias entre tareas como las restricciones de recursos necesarios. La restricción más relevante es la de recursos humanos, ya que se cuenta con un único responsable y desarrollador. Se estimó una carga de trabajo de 6 horas diarias durante 5 días a la semana. El detalle de las fechas, duración y descripción de las tareas puede observarse en el cuadro 1.

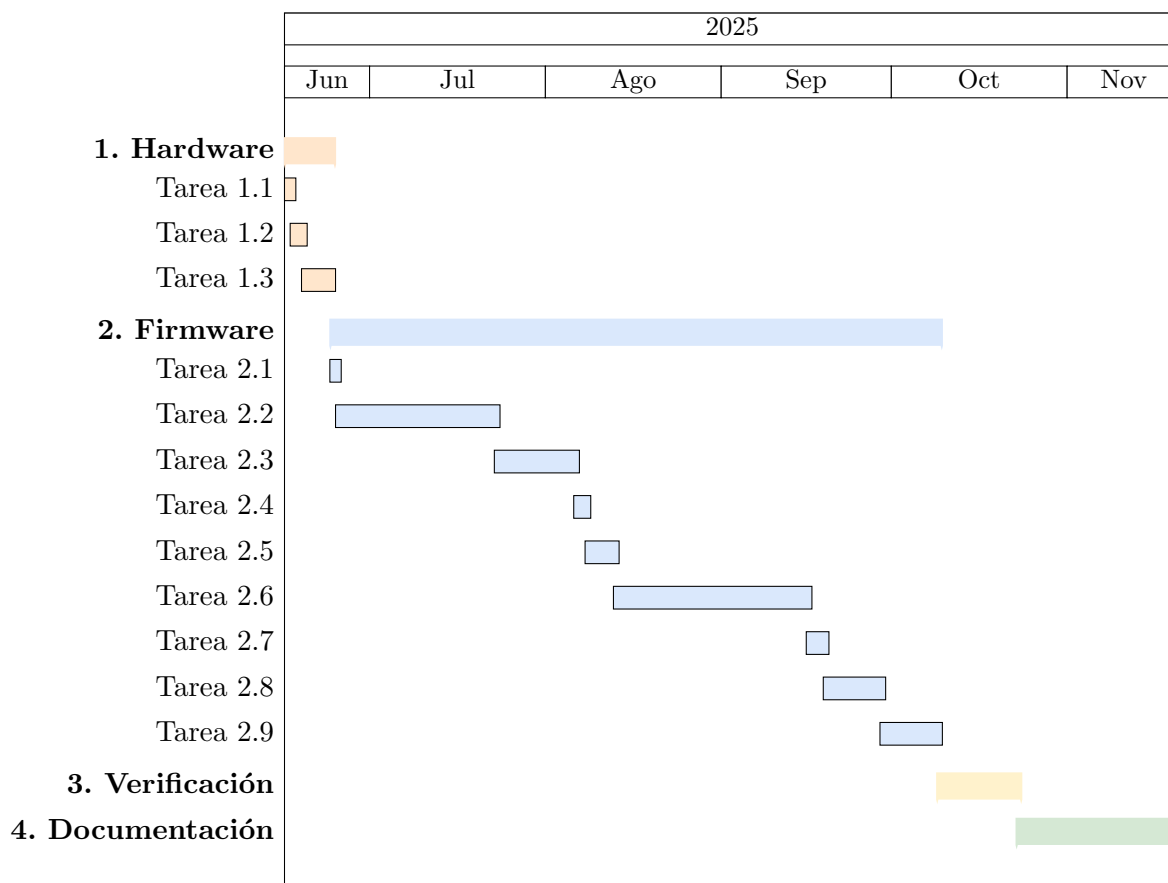


Figura 3. Diagrama de Gantt del proyecto.

Número de tarea	Descripción	Duración	Fecha inicio	Fecha finalización
1	Desarrollo del hardware	34 h	16/06/2025	24/06/2025
1.1	Selección y adquisición de componentes	4 h	16/06/2025	17/06/2025
1.2	Montaje del prototipo en <i>proto-board</i>	12 h	17/06/2025	19/06/2025
1.3	Pruebas por componente en <i>proto-board</i>	18 h	19/06/2025	24/06/2025
2	Desarrollo del firmware	436 h	24/06/2025	09/10/2025
2.1	Configuración inicial del entorno de desarrollo	1 h	24/06/2025	25/06/2025
2.2	Implementación de la lógica del sistema	117 h	25/06/2025	23/07/2025
2.3	Implementación del demo de juego	55 h	23/07/2025	06/08/2025
2.4	Detección de entradas simples	11 h	06/08/2025	08/08/2025
2.5	Detección de inclinación	18 h	08/08/2025	13/08/2025
2.6	Gestión de la interfaz de usuario	140 h	13/08/2025	16/09/2025
2.7	Implementación de la reproducción de audio	18 h	16/09/2025	19/09/2025
2.8	Implementación del control de vibración	33 h	19/09/2025	29/09/2025
2.9	Gestión del estado del juego	43 h	29/09/2025	09/10/2025
3	Pruebas y verificación	55 h	09/10/2025	23/10/2025
4	Documentación	115 h	23/10/2025	20/11/2025

Cuadro 1. Detalle de las tareas para el diagrama de Gantt del proyecto.

## 12. Presupuesto detallado del proyecto

En el cuadro 2 se listan todos los costos asociados al proyecto, expresados en euros (EUR). La cotización de otras monedas de interés al 1 de mayo de 2025 fue la siguiente:

- EUR 1 = dólares americanos (USD) 1,1343, presupuesto en USD: 14.843,45.
- EUR 1 = pesos argentinos (ARS) 1328,36, presupuesto en ARG: 17.382.918,96.

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Placa STM NUCLEO-F446RE	1	€ 33,99	€ 33,99
Pantalla TFT	1	€ 7,49	€ 7,49
Controlador háptico DRV2605L	1	€ 9,99	€ 9,99
Motor vibración 3V 44000rpm	5	€ 1,44	€ 7,21
Speaker 8 ohm 1W	2	€ 3,18	€ 6,35
Amplificadores para audio	5	€ 0,27	€ 1,33
Módulo power supply (9V → 5V/3.3V)	1	€ 2,00	€ 2,00
Protoboard	3	€ 2,83	€ 8,49
Cables de conexión (pack)	1	€ 8,59	€ 8,59
Joystick analógico	1	€ 2,00	€ 2,00
Botones tipo switch	5	€ 0,30	€ 1,50
Analizador lógico de 8 canales	1	€ 13,99	€ 13,99
EEPROM SPI 256K (25LC256)	1	€ 1,70	€ 1,70
Acelerómetro GY-521	1	€ 2,99	€ 2,99
Honorarios profesionales	640	€ 20,00	€ 12.800,00
SUBTOTAL			€ 12.906,80
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Servicios de electricidad e internet	640	€ 0,28	€ 179,20
SUBTOTAL			€ 179,20
TOTAL			€ 13.086,00

Cuadro 2. Presupuesto del proyecto.

### 13. Gestión de riesgos

En la presente sección se listan los riesgos detectados para el proyecto. Para estimar la severidad y la probabilidad de ocurrencia de cada riesgo se utilizó una escala de 1 a 10, donde 1 corresponde a la menor severidad/ocurrencia y 10 a la mayor.

Riesgo 1. Que las horas estimadas de trabajo no sean suficientes para cumplir con los objetivos planteados.

- Severidad (S): 7. Se retrasaría la fecha de presentación del proyecto y su defensa pública.
- Ocurrencia (O): 6. No se tiene experiencia en muchos de los aspectos técnicos a trabajar durante el proyecto, por lo que la incertidumbre asociada a la planificación de las horas de trabajo es grande.

Riesgo 2. Que el responsable no pueda dedicar su tiempo completo al proyecto.

- Severidad (S): 7. Se retrasaría la fecha de presentación del proyecto y su defensa pública.
- Ocurrencia (O): 3. El responsable no asumirá otros compromisos de largo plazo durante el transcurso del proyecto, pero eventos fuera de su control podrían requerir de su atención de todas maneras.

Riesgo 3. Que la solución planteada no sea técnicamente viable.

- Severidad (S): 9. Se debería cambiar de proyecto.
- Ocurrencia (O): 3. Existen dispositivos con elementos de hardware y software similares en el mercado, por lo que no es tan probable que no pueda desarrollarse el proyecto planteado.

Riesgo 4. Que los conocimientos del posgrado requeridos para avanzar en las distintas etapas del proyecto no se alcanzaran en tiempos compatibles con el desarrollo.

- Severidad (S): 8. Podría ser necesario el retrabajo sobre algún aspecto del proyecto, retrasando la fecha de presentación y su defensa pública.
- Ocurrencia (O): 4. El orden de las materias en el posgrado parece ser coherente con la progresión del trabajo necesario para el proyecto.

Riesgo 5. Que no se disponga de los materiales y componentes necesarios para el prototipo funcional.

- Severidad (S): 8. Podría ser necesario el rediseño de algún aspecto del proyecto y/o la búsqueda de componentes de reemplazo, retrasando la fecha de presentación y su defensa pública.
- Ocurrencia (O): 7. El mercado local de componentes electrónicos es escaso y la importación de componentes desde afuera del país suele ser bastante compleja.

Riesgo 6. Que la complejidad de graficar en la pantalla el tipo de información requerida sea mayor que la prevista en esta planificación.

- Severidad (S): 7. Se retrasaría la fecha de presentación del proyecto y su defensa pública.
- Ocurrencia (O): 7. Se desconocen por completo las técnicas de renderizado de imágenes para videojuegos.

Estos riesgos se ponderan de acuerdo a la siguiente fórmula:

$$RPN = S * O$$

Riesgo	S	O	RPN	S*	O*	RPN*
R1: Horas insuficientes para cumplir objetivos	7	6	42	7	4	28
R2: No disponer de tiempo completo	7	3	21			
R3: Solución técnicamente inviable	9	3	27			
R4: Avance del posgrado desfazado con el plan	8	4	32			
R5: Falta de disponibilidad de componentes	8	7	56	6	5	30
R6: Alta complejidad de renderizado de gráficos	7	7	49	6	5	30

Cuadro 3. Gestión de riesgos del proyecto.

Criterio adoptado:

Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a 35.

Nota: los valores marcados con (\*) en la tabla corresponden a los nuevos valores después de haber aplicado la mitigación.

Como puede observarse en el cuadro 3, existen 3 riesgos que exceden el límite de RPN adoptado, por lo que a continuación se presenta el plan de mitigación planteado para cada uno de ellos.

Riesgo 1. Este riesgo se mitigará solicitando asistencia al director del proyecto cuando sea necesario y revisando con frecuencia el avance de las tareas en el cronograma planteado.

- Severidad (S\*): 7. La severidad se conserva.
- Ocurrencia (O\*): 4. La ocurrencia disminuye por esta mitigación.

Riesgo 5. Se mitigará comprando los materiales al inicio del proyecto.

- Severidad (S\*): 6. La severidad disminuye comprando los materiales de forma anticipada al evitar la necesidad de reemplazar componentes con el proyecto ya avanzado.
- Ocurrencia (O\*): 5. La ocurrencia también disminuye ya que de haber retrasos en la entrega, ocurrirían solo al comienzo del proyecto.

Riesgo 6. Se mitigará planteando para el requerimiento de gráficos que éstos deben permitir observar el funcionamiento del prototipo, sin explicitar la generación de gráficos complejos.

- Severidad (S\*): 6. Se baja la severidad ya que permite simplificar la interfaz gráfica en caso de ser necesario.
- Ocurrencia (O\*): 5. Se baja la ocurrencia de que la complejidad exceda lo previsto al contemplar una alternativa de mayor libertad de decisión.

## 14. Gestión de la calidad

En esta sección se presentan los métodos que se utilizarán para gestionar la calidad del proyecto a partir de sus requerimientos más relevantes.

### 1. Requerimientos de hardware:

1.1. El prototipo deberá montarse en *protoboard* o soldarse sobre placas de experimentación.

- **Verificación:** se verificará que el prototipo coincide con el esquema de conexión realizado.
- **Validación:** se validará el esquema de conexión del prototipo.

1.2. El prototipo deberá alimentarse a través de una batería.

- **Verificación:**
  - Se comprobará que el módulo de alimentación utilizado permite el uso de una batería de 9 V para alimentar el microcontrolador y los periféricos.
  - Se realizará una prueba de encendido con todos los componentes conectados para confirmar que el sistema se inicializa correctamente usando dicho módulo.

- Se verificará que la fuente elegida sea funcional y no presente caídas de tensión visibles o reinicios inesperados durante el arranque del sistema.
  - **Validación:** se utilizará el prototipo en una sesión de juego de al menos 45 minutos alimentado por batería, confirmando que el sistema mantiene una experiencia de uso estable, sin cortes de energía ni reinicios.
- 1.3. El prototipo deberá integrar un sensor capaz de medir la inclinación en al menos dos ejes.
- **Verificación:**
    - Se revisará la hoja de datos del sensor para confirmar que posee al menos dos canales de aceleración.
    - Se implementará un código de prueba que envíe por UART los valores leídos de inclinación en los ejes X e Y.
  - **Validación:** se validará la estabilidad y coherencia de las lecturas al inclinar el prototipo en distintos ángulos.
- 1.4. El prototipo deberá integrar un joystick analógico con capacidad de lectura en dos ejes.
- **Verificación:**
    - Se verificará que el módulo utilizado dispone de dos salidas analógicas independientes, típicamente etiquetadas como X e Y.
    - Se ejecutará un código de prueba que envíe por UART los valores leídos en ambos ejes al mover el joystick en distintas direcciones.
  - **Validación:** se validará que los valores leídos de ambos ejes cambian de forma proporcional y continua a lo largo del rango físico del joystick, sin zonas muertas ni saturaciones inesperadas.
- 1.5. El prototipo deberá incluir botones tipo switch para funciones definidas.
- **Verificación:** se utilizará un código de prueba para confirmar la detección del estado de cada botón.
  - **Validación:** se validará que cada botón responde al presionarse y soltarse con normalidad, sin pérdida de pulsaciones ni retrasos aparentes.
- 1.6. El prototipo deberá incorporar una pantalla capaz de mostrar información visual relevante para el juego.
- **Verificación:**
    - Se comprobará que la pantalla se inicializa correctamente desde el microcontrolador y que responde a comandos de escritura.
    - Se ejecutará un código de prueba que muestre texto, formas geométricas y gráficos simples.
    - Se verificará la ausencia de parpadeos excesivos o problemas de sincronización.
  - **Validación:** se validará visualmente que la pantalla no presenta defectos de encendido, líneas muertas o problemas de contraste.
- 1.7. El prototipo deberá incluir un mecanismo para emitir sonidos o alertas sonoras.
- **Verificación:**
    - Se verificará que el dispositivo de audio utilizado (buzzer o parlante) responde a las señales aplicadas desde el microcontrolador.
    - Se utilizará un código de prueba para comprobar la generación de sonidos.
  - **Validación:** se validará que el sonido producido es claramente audible, sin distorsión, interferencias o comportamiento errático, al ser activado por una señal de prueba.

1.8. El prototipo deberá integrar un motor de vibración capaz de generar retroalimentación háptica.

■ **Verificación:**

- Se comprobará la activación del motor desde una salida digital del microcontrolador o mediante alimentación directa.
- Se verificará que el motor genera vibración mecánica perceptible al ser activado por el microcontrolador.

■ **Validación:** se validará que el motor puede ser activado repetidamente durante varios ciclos de encendido y apagado, sin calentamiento excesivo, fallos mecánicos ni desconexiones eléctricas.

1.9. El prototipo deberá contar con memoria no volátil para el almacenamiento persistente del estado del juego.

■ **Verificación:**

- Se verificará que la memoria seleccionada puede ser accedida desde el microcontrolador, escribiendo y leyendo un bloque de prueba.
- Se comprobará que los datos almacenados permanecen inalterados luego de un reinicio del sistema.

■ **Validación:** se validará que la memoria conserva datos grabados durante varias horas sin alimentación y que puede recuperarlos íntegramente sin errores de lectura.

2. Requerimientos de firmware:

2.1. El firmware deberá implementar la lógica para interpretar los datos del sensor de inclinación y el joystick para controlar el modelo de vuelo (cabeceo, alabeo, giro, velocidad).

■ **Verificación:**

- Se realizará revisión de código y pruebas unitarias de las funciones de conversión de lecturas en comandos de control.
- Se verificará que los datos simulados del sensor y el joystick se traducen correctamente en variaciones de cabeceo, alabeo, giro y velocidad.

■ **Validación:** se validará que al mover físicamente el prototipo y activar el joystick se observan cambios coherentes en la orientación o desplazamiento del objeto controlado en pantalla, sin saltos ni bloqueos.

2.2. El firmware deberá incluir un demo de juego integrado que permita demostrar la lógica implementada y la funcionalidad de los periféricos de entrada y salida.

■ **Verificación:**

- Se analizarán las cualidades necesarias del demo para demostrar las funcionalidades requeridas.
- Se revisará el código del demo para comprobar que accede efectivamente a cada periférico (entradas y salidas).
- Se verificará que se llaman correctamente las funciones de lectura y escritura asociadas a cada componente.

■ **Validación:** se validará que, al iniciar el demo, el usuario puede interactuar con el sistema usando todos los periféricos conectados y que se observa una respuesta esperable a esas interacciones.

2.3. El firmware deberá gestionar la visualización de la información del juego en la pantalla.

- **Verificación:**
    - Se realizarán pruebas unitarias o funcionales del módulo gráfico, utilizando funciones de dibujo sobre buffer o mock de pantalla.
    - Se inspeccionará que los datos visuales (como posición, dirección, velocidad) se corresponden con los estados internos del juego.
  - **Validación:** se validará visualmente que los cambios en el estado del sistema se reflejan correctamente en la pantalla, con gráficos que cambian en tiempo real y sin errores de visualización.
- 2.4. El firmware deberá ser capaz de reproducir las alertas sonoras necesarias.
- **Verificación:**
    - Se verificará que el firmware incluya rutinas de generación de tono o control PWM asociadas a eventos definidos.
    - Se realizarán pruebas unitarias sobre las funciones de generación de sonidos.
  - **Validación:** se validará que cada evento predefinido genera una señal audible correspondiente, y que estas señales se diferencian claramente entre sí.
- 2.5. El firmware deberá controlar la activación y duración del motor de vibración en función de los eventos del juego.
- **Verificación:**
    - Se analizarán los eventos que producen cada patrón de vibración implementado.
    - Se realizarán pruebas de generación y duración de la vibración producida.
  - **Validación:** se validará que la vibración producida sea perceptible, de duración apropiada y se corresponda con los eventos definidos para su generación.
- 2.6. El firmware deberá implementar la lógica para guardar y cargar el estado del juego en la memoria no volátil.
- **Verificación:**
    - Se realizarán pruebas unitarias de guardado y recuperación de la información de partidas simuladas.
    - Se verificará que los datos del estado se serializan correctamente y son leídos adecuadamente.
  - **Validación:** se validará que al reiniciar el sistema, el juego retoma desde un punto equivalente al guardado, sin pérdida de información.
- 2.7. El firmware deberá detectar las pulsaciones de los botones y activar las funcionalidades correspondientes (inicio/fin, pausa, guardar, mapa, etc...).
- **Verificación:**
    - Se verificará que cada evento de botón se encuentra correctamente mapeado a una acción en el código.
    - Se verificará que el tiempo y el método designado para el anti-rebote por software de los botones permite la detección de las pulsaciones de forma correcta.
  - **Validación:** se validará que, al presionar cada botón, se ejecuta de forma inmediata la acción esperada (inicio, pausa, guardar, etc.), sin necesidad de pulsaciones repetidas o excesiva presión.

## 15. Procesos de cierre

1. Análisis del seguimiento del plan de proyecto original:



- Responsables: Jezabel Victoria Danon junto al director del proyecto.
- Se analizará el cumplimiento del EDT con sus tiempos y dependencias.
- Se evaluará el cumplimiento de los objetivos planteados y los posibles cambios que hayan sido necesarios.

2. Análisis de las técnicas y procedimientos utilizados:

- Responsables: Jezabel Victoria Danon.
- Se analizarán las técnicas y herramientas seleccionadas para el proyecto y su grado de utilidad al momento de trabajar con ellas.
- Se identificarán aquellas que no cumplieron con las expectativas o aquellas que no fueron de utilidad para la resolución de problemas durante el proyecto.
- Se documentarán las conclusiones para futuras referencias.

3. Organización de presentación del proyecto:

- Responsables: Jezabel Victoria Danon.
- Se elaborará una memoria técnica y una presentación del proyecto para su defensa pública ante el jurado evaluador.
- Durante esta presentación se agradecerá al director por sus aportes, así como a todos los demás involucrados en el proyecto.