



# Videojuego portátil inspirado en consolas retro

Autor:

Lic. Jezabel Victoria Danon

Director:

Título y Nombre del director (pertenencia)

*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 . . . . .	14
10. Diagrama de Activity On Node. . . . .	16
11. Diagrama de Gantt . . . . .	17
12. Presupuesto detallado del proyecto . . . . .	20
13. Gestión de riesgos . . . . .	20
14. Gestión de la calidad . . . . .	21
15. Procesos de cierre . . . . .	22

## 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

## 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 720 horas y un costo estimado de € 120 (ciento veinte euros), 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

Título y Nombre del director  
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.

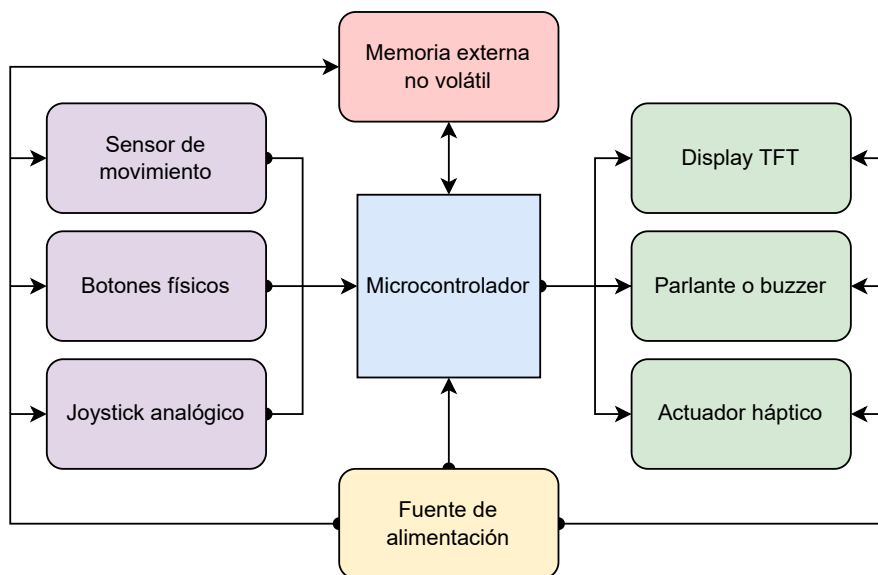


Figura 1. Diagrama en bloques del sistema.

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	Título y Nombre del director	pertenencia	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

**Épica 1:** Control e interacción del usuario con el demo de juego

**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.

Criterios de aceptación:

- El sistema interpreta la inclinación de la consola en los ejes X e Y como comandos de alabeo y cabeceo respectivamente.
- La respuesta del modelo de vuelo a la inclinación es inmediata y fluida, sin latencia perceptible.
- El control debe estar calibrado en un rango usable (sin saturar los extremos).
- Se pueden realizar movimientos suaves y precisos con pequeñas inclinaciones.

*Story points:* 13 (complejidad: 3, trabajo: 2, incertidumbre: 5)

**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.

Criterios de aceptación:

- El movimiento horizontal del joystick controla el ángulo de giro de la aeronave de forma proporcional.
- El movimiento vertical del joystick controla la velocidad de avance de la aeronave.
- El control de giro es suave y permite realizar ajustes finos.

*Story points:* 8 (complejidad: 3, trabajo: 2, incertidumbre: 2)

**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.

Criterios de aceptación:

- Al presionar el botón de “Inicio”, la demostración del juego comienza en un estado predefinido.
- Al presionar el botón de “Finalizar”, el juego se detiene y la consola vuelve a un menú principal o estado de espera.

*Story points:* 5 (complejidad: 1, trabajo: 1, incertidumbre: 2)

**Épica 2:** Visualización del entorno y mapa

**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.

Criterios de aceptación:

- La velocidad, dirección y posición actual de la aeronave se representan visualmente en la pantalla.
- Los cambios en la velocidad, dirección y posición se reflejan en la pantalla en tiempo real (sin lags perceptibles).

*Story points:* 21 (complejidad: 5, trabajo: 2, incertidumbre: 8)

**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.

Criterios de aceptación:

- Al presionar el botón de “Mapa”, se muestra una representación del entorno de vuelo.
- La posición actual de la aeronave se indica claramente en el mapa.
- El mapa es legible y proporciona una orientación básica del entorno.

*Story points:* 21 (complejidad: 5, trabajo: 3, incertidumbre: 8)

**Épica 3:** Audio y retroalimentación háptica

**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.

Criterios de aceptación:

- Se generan al menos dos sonidos distintos para diferentes eventos del juego.
- Los sonidos son claramente audibles.
- La emisión de sonido está sincronizada con el evento que la provoca.

*Story points:* 8 (complejidad: 3, trabajo: 2, incertidumbre: 2)

**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.

Criterios de aceptación:

- El motor de vibración se activa ante al menos un evento definido.
- La duración de la vibración es perceptible y coherente con el evento.

*Story points:* 8 (complejidad: 1, trabajo: 2, incertidumbre: 5)

**Épica 4:** Gestión del estado del juego

**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.

Criterios de aceptación:

- El sistema almacena las variables clave del estado del juego en memoria no volátil.
- Se proporciona una señal de confirmación visual y/o sonora al guardar.
- El estado guardado se puede cargar al reiniciar el entorno.

*Story points:* 13 (complejidad: 3, trabajo: 3, incertidumbre: 5)

**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.

Criterios de aceptación:

- Si existe un estado guardado, el sistema ofrece la opción de continuar o iniciar una nueva partida.
- Si no hay estado guardado, el juego comienza automáticamente.
- Se indica claramente la opción seleccionada por el usuario.

*Story points:* 5 (complejidad: 1, trabajo: 2, incertidumbre: 2)

**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.

Criterios de aceptación:

- Al presionar el botón de “Pausa” por primera vez, el juego se detiene.
- Al presionar el botón de “Pausa” nuevamente, el juego se restablece en el mismo estado que estaba antes de pausar.
- Se indica claramente en pantalla cuando el juego está pausado.

*Story points:* 5 (complejidad: 1, trabajo: 2, incertidumbre: 2)

**Épica 5:** Documentación del proyecto

**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.

Criterios de aceptación:

- El diagrama muestra todos los módulos de hardware principales.
- Las conexiones entre los módulos son claras y fáciles de seguir.
- Se incluyen etiquetas descriptivas para cada componente y conexión.

*Story points:* 5 (complejidad: 1, trabajo: 1, incertidumbre: 2)

**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.

Criterios de aceptación:

- El video demuestra las funcionalidades clave de las historias de usuario implementadas.

- El video es claro, bien enfocado y con una duración adecuada.
- Se pueden observar los criterios de aceptación de las funcionalidades en el video.

*Story points:* 5 (complejidad: 1, trabajo: 2, incertidumbre: 2)

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

Criterios de aceptación:

- Se entrega al menos un informe de avance.
- El informe documentan los avances realizados y los cambios respecto al plan (si se presentan).
- El informe es claro y proporciona una visión del progreso del proyecto.

*Story points:* 8 (complejidad: 3, trabajo: 2, incertidumbre: 2)

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

Criterios de aceptación:

- La memoria técnica describe la arquitectura del hardware y del software.
- Se explican las decisiones de diseño clave y sus justificaciones.
- La documentación es clara, organizada y proporciona suficiente detalle técnico.

*Story points:* 13 (complejidad: 3, trabajo: 3, incertidumbre: 5)

## 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. (42 hs)

#### 1.1. Selección y adquisición de componentes. (5 hs)

1.1.1. Confirmación de los componentes existentes y verificación de su idoneidad. (2 h)

1.1.2. Adquisición de componentes faltantes o adicionales. (3 h)

#### 1.2. Montaje del prototipo en *protoboard*. (18 hs)

1.2.1. Planificación del diseño de montaje. (5 h)

1.2.2. Montaje y cableado del microcontrolador y la fuente de alimentación. (5 h)

1.2.3. Montaje y cableado del sensor de inclinación. (1 h)

1.2.4. Montaje y cableado del joystick analógico. (1 h)

1.2.5. Montaje y cableado de los botones. (1 h)

1.2.6. Montaje y cableado de la pantalla. (1 h)

1.2.7. Montaje y cableado del sistema de audio. (2 h)

1.2.8. Montaje y cableado del motor de vibración. (2 h)

#### 1.3. Pruebas por componente en *protoboard*. (18 hs)

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. (509 hs)

#### 2.1. Configuración Inicial del entorno de desarrollo . (2 hs)

2.1.1. Configuración de la placa, pines y periféricos. (2 h)

#### 2.2. Implementación de la lógica del sistema. (135 hs)

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 . (40 h)

2.2.2.1. Módulo de salida: sub-módulo de audio.. (5 h)

2.2.2.2. Módulo de salida: sub-módulo de vibraciones.. (10 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). (10 h)

2.2.4. Implementación del RTOS seleccionado. (30 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. (60 hs)

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

2.3.2. Desarrollo de la máquina de estados del juego. (15 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. (20 hs)

2.4.1. Implementación de la detección de pulsaciones de los botones. (5 h)

- 2.4.2. Implementación de la lectura de los datos del joystick analógico. (5 h)
- 2.4.3. Integración de las entradas de botones y joystick con la máquina de estados del sistema. (10 h)
- 2.5. Detección de inclinación. (30 hs)
  - 2.5.1. Implementación de la lectura de los datos del sensor de inclinación . (10 h)
  - 2.5.2. Procesamiento de los datos del acelerómetro para obtener la inclinación. (10 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 hs)
  - 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. (24 hs)
  - 2.7.1. Investigación sobre modos de generación y/o reproducción de audio en sistemas embebidos. (8 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. (8 h)
- 2.8. Implementación del control de vibración. (40 hs)
  - 2.8.1. Investigación sobre generación de vibración con distintos patrones. (10 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. (15 h)
- 2.9. Gestión del estado del juego. (58 hs)
  - 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. (20 h)
  - 2.9.3. Implementación de la lógica para cargar el estado del juego desde memoria no volátil. (20 h)
  - 2.9.4. Implementación de la lógica para pausar y reanudar el juego. (10 h)
- 2.10. Integración del Demo de Juego. (Tarea continua)
- 3. Pruebas y verificación. (55 hs)
  - 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 hs)
  - 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 hs)
  - 5.1. Implementación de gráficos a color. (78 hs)
    - 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 hs)
    - 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 hs)
    - 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): 720.

Cantidad de horas de ingeniería (sin contemplar opcionales ni documentación): 605.

## 10. Diagrama de Activity On Node

Armar el AoN a partir del WBS definido en la etapa anterior.

Una herramienta simple para desarrollar los diagramas es el Draw.io (<https://app.diagrams.net/>). Draw.io

Indicar claramente en qué unidades están expresados los tiempos. De ser necesario indicar los caminos semi críticos y analizar sus tiempos mediante un cuadro. Es recomendable usar colores y un cuadro indicativo describiendo qué representa cada color.



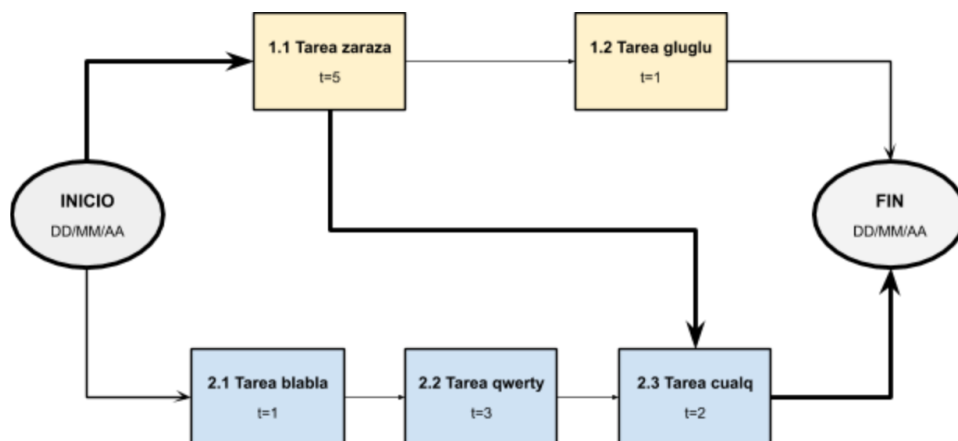


Figura 2. Diagrama de *Activity on Node*.

## 11. Diagrama de Gantt

Existen muchos programas y recursos *online* para hacer diagramas de Gantt, entre los cuales destacamos:

- Planner
- GanttProject
- Trello + *plugins*. En el siguiente link hay un tutorial oficial:  
<https://blog.trello.com/es/diagrama-de-gantt-de-un-proyecto>
- Creately, herramienta online colaborativa.  
<https://creately.com/diagram/example/ieb3p3ml/LaTeX>
- Se puede hacer en latex con el paquete *pgfgantt*  
<http://ctan.dcc.uchile.cl/graphics/pgf/contrib/pgfgantt/pgfgantt.pdf>

Pegar acá una captura de pantalla del diagrama de Gantt, cuidando que la letra sea suficientemente grande como para ser legible. Si el diagrama queda demasiado ancho, se puede pegar primero la “tabla” del Gantt y luego pegar la parte del diagrama de barras del diagrama de Gantt.

Configurar el software para que en la parte de la tabla muestre los códigos del EDT (WBS).  
Configurar el software para que al lado de cada barra muestre el nombre de cada tarea.  
Revisar que la fecha de finalización coincida con lo indicado en el Acta Constitutiva.

En la figura 3, se muestra un ejemplo de diagrama de gantt realizado con el paquete de *pgfgantt*. En la plantilla pueden ver el código que lo genera y usarlo de base para construir el propio.

Las fechas pueden ser calculadas utilizando alguna de las herramientas antes citadas. Sin embargo, el siguiente ejemplo fue elaborado utilizando [esta hoja de cálculo](#).

Es importante destacar que el ancho del diagrama estará dado por la longitud del texto utilizado para las tareas (Ejemplo: tarea 1, tarea 2, etcétera) y el valor *x unit*. Para mejorar la apariencia del diagrama, es necesario ajustar este valor y, quizás, acortar los nombres de las tareas.

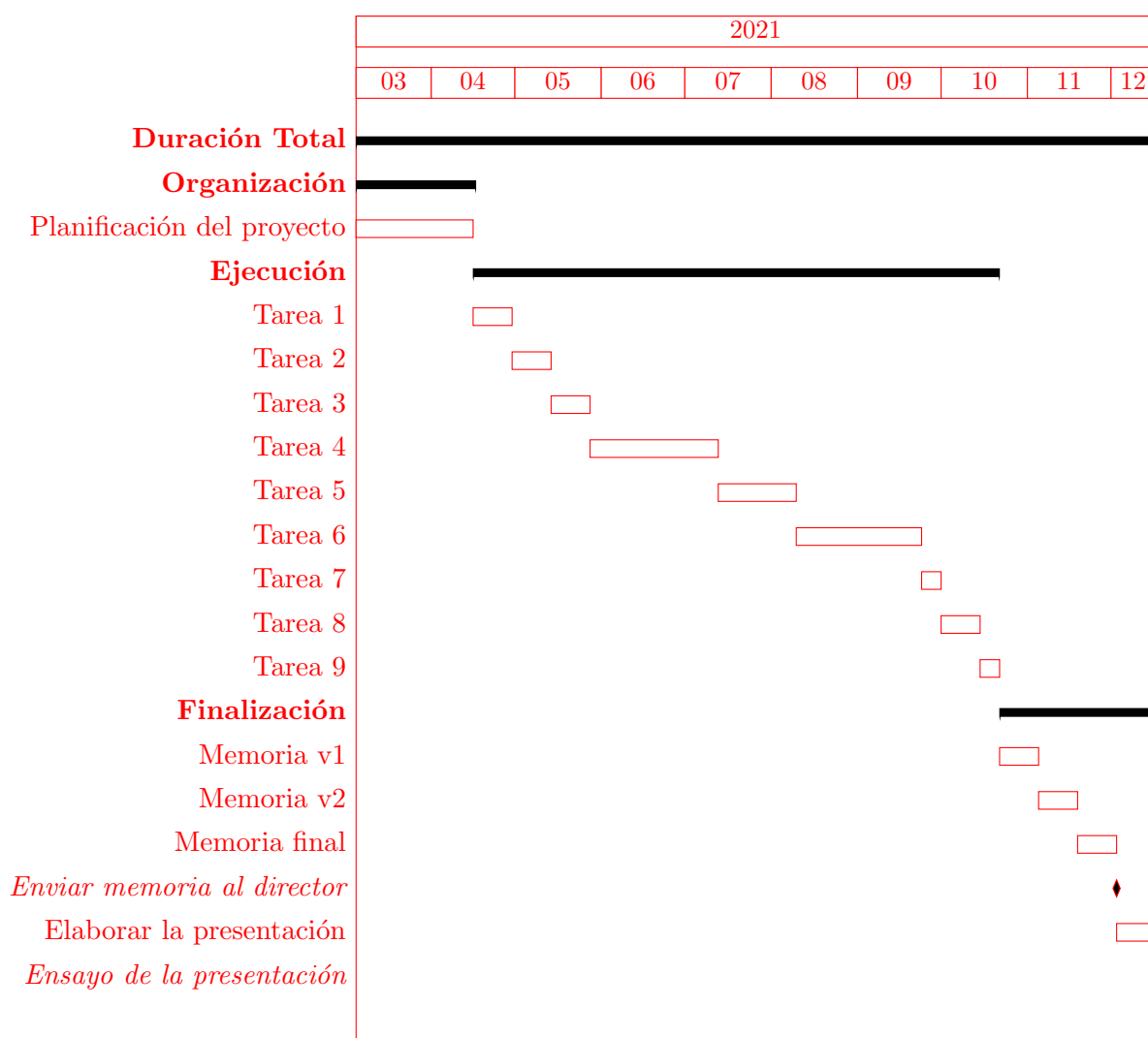


Figura 3. Diagrama de gantt de ejemplo

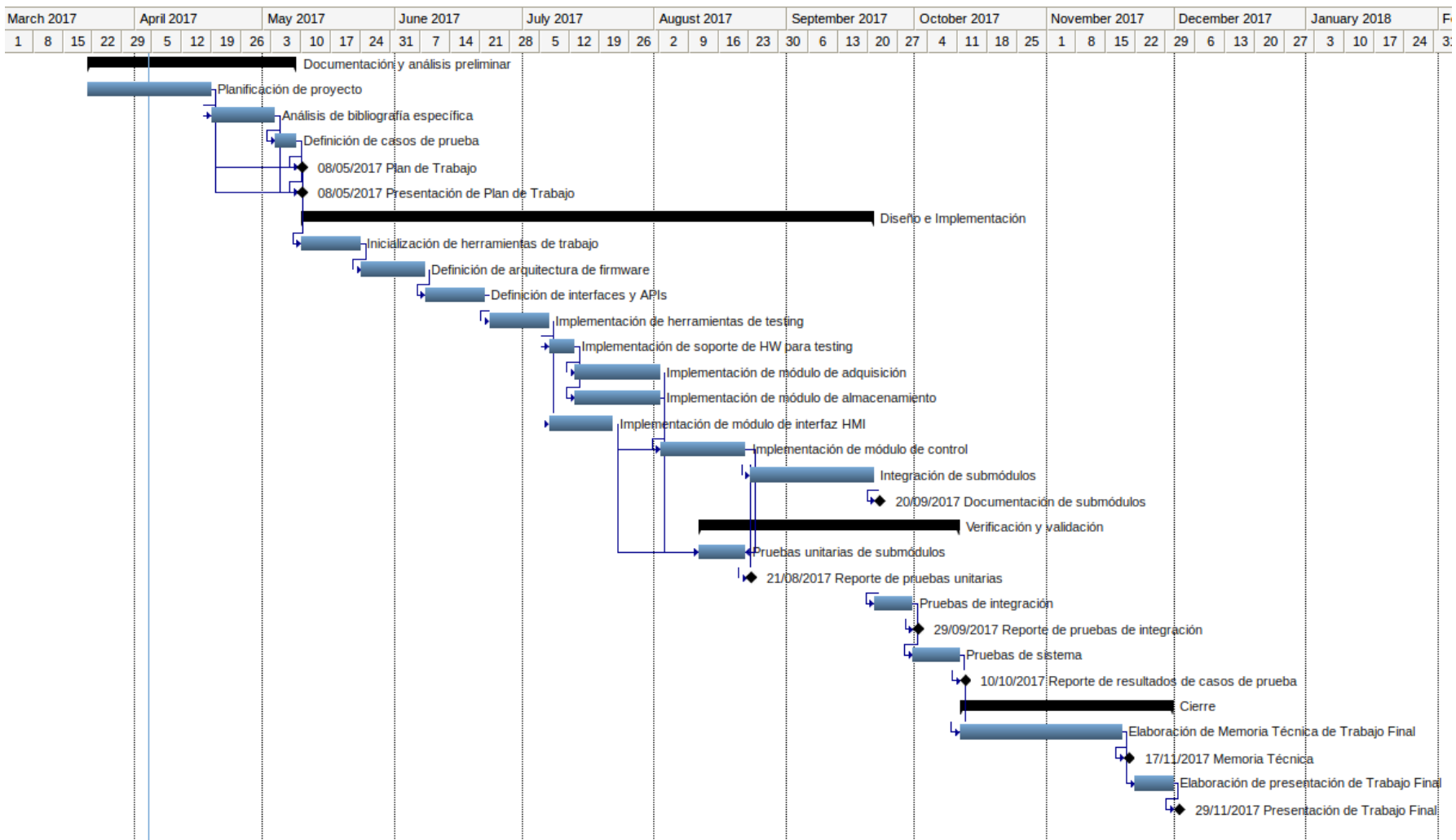


Figura 4. Ejemplo de diagrama de Gantt (apaisado).

## 12. Presupuesto detallado del proyecto

Si el proyecto es complejo entonces separarlo en partes:

- Un total global, indicando el subtotal acumulado por cada una de las áreas.
- El desglose detallado del subtotal de cada una de las áreas.

**IMPORTANTE:** No olvidarse de considerar los **COSTOS INDIRECTOS**.

Incluir la aclaración de si se emplea como moneda el peso argentino (ARS) o si se usa moneda extranjera (USD, EUR, etc). Si es en moneda extranjera se debe indicar la tasa de conversión respecto a la moneda local en una fecha dada.

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
SUBTOTAL			
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
SUBTOTAL			
TOTAL			

## 13. Gestión de riesgos

a) Identificación de los riesgos (al menos cinco) y estimación de sus consecuencias:

Riesgo 1: detallar el riesgo (riesgo es algo que si ocurre altera los planes previstos de forma negativa)

- Severidad (S): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2:

- Severidad (S): X.  
Justificación...

- Ocurrecia (O): Y.  
Justificación...

Riesgo 3:

- Severidad (S): X.  
Justificación...
- Ocurrecia (O): Y.  
Justificación...

b) Tabla de gestión de riesgos: (El RPN se calcula como  $RPN=S \times O$ )

Riesgo	S	O	RPN	S*	O*	RPN*

Criterio adoptado:

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

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

c) Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).  
Nueva asignación de S y O, con su respectiva justificación:

- Severidad (S\*): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O\*): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

Riesgo 3: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

## 14. Gestión de la calidad

Elija al menos diez requerimientos que a su criterio sean los más importantes/críticos/que aportan más valor y para cada uno de ellos indique las acciones de verificación y validación que permitan asegurar su cumplimiento.

- Req #1: copiar acá el requerimiento con su correspondiente número.

- Verificación para confirmar si se cumplió con lo requerido antes de mostrar el sistema al cliente. Detallar.
- Validación con el cliente para confirmar que está de acuerdo en que se cumplió con lo requerido. Detallar.

Tener en cuenta que en este contexto se pueden mencionar simulaciones, cálculos, revisión de hojas de datos, consulta con expertos, mediciones, etc.

Las acciones de verificación suelen considerar al entregable como “caja blanca”, es decir se conoce en profundidad su funcionamiento interno.

En cambio, las acciones de validación suelen considerar al entregable como “caja negra”, es decir, que no se conocen los detalles de su funcionamiento interno.

## 15. Procesos de cierre

Establecer las pautas de trabajo para realizar una reunión final de evaluación del proyecto, tal que contemple las siguientes actividades:

- Pautas de trabajo que se seguirán para analizar si se respetó el Plan de Proyecto original:
  - Indicar quién se ocupará de hacer esto y cuál será el procedimiento a aplicar.
- Identificación de las técnicas y procedimientos útiles e inútiles que se emplearon, los problemas que surgieron y cómo se solucionaron:
  - Indicar quién se ocupará de hacer esto y cuál será el procedimiento para dejar registro.
- Indicar quién organizará el acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores:
  - Indicar esto y quién financiará los gastos correspondientes.