



Videojuego portátil inspirado en consolas retro

Especificación de requerimientos de software

Autor:

Lic. Jezabel Danon (jezabel.danon@gmail.com)

07/07/2025

VERSIÓN A

Este documento fue creado en base al estándar IEEE-830 durante el curso de Ingeniería de Software entre el 26 de junio de 2025 y el 21 de agosto de 2025.

Historial de cambios

Versión	Fecha	Descripción	Autor	Revisores
A	07/07/2025	Creación del documento	Lic. Jezabel Danon	

Índice

1. Introducción	3
1.1. Propósito	3
1.2. Ámbito del sistema	3
1.3. Definiciones, Acrónimos y Abreviaturas	3
1.4. Referencias	4
1.5. Visión general del documento	4
2. Descripción general del documento	4
2.1. Perspectiva del producto	4
2.2. Funciones del producto	5
2.3. Características de los usuarios	6
2.4. Restricciones	6
2.5. Suposiciones y dependencias	6
2.6. Requisitos futuros	6
3. Requisitos específicos	6
3.1. Interfaces externas	6
3.2. Funciones	7
3.2.1. Flujo de encendido	7
3.2.2. Flujo de pausa y opciones	7
3.2.3. Bucle de juego	7
3.3. Requisitos de rendimiento	8
3.4. Restricciones de diseño	8
3.5. Atributos del sistema	8
3.6. Otros requisitos	8
4. Apéndices	8

1. Introducción

1.1. Propósito

1. Este documento representa una especificación de requerimientos de software para el sistema embebido *Videojuego portátil inspirado en consolas retro*.
2. Está dirigido a los desarrolladores que se ocupen del análisis, diseño e implementación del software, así como también a quienes desarrollen el testing, validaciones y/o verificaciones del mismo.

1.2. Ámbito del sistema

1. El software a desarrollar controlará la interfaz de usuario de la consola de juegos y procesará las entradas y salidas de los periféricos.
2. Las entradas del sistema serán:
 - Botones
 - Joystick analógico
 - Acelerómetro
3. Las salidas del sistema serán:
 - Pantalla
 - Salida de audio
 - Motor de vibraciones

1.3. Definiciones, Acrónimos y Abreviaturas

1. Acrónimos:
 - IEEE: Instituto de Ingenieros Eléctricos y Electrónicos.
 - ERS: Especificación de Requisitos de Software.
 - RTOS: Sistema Operativo de Tiempo Real (Real Time Operating System).
 - CMSIS: Arm's Common Microcontroller Software Interface Standard. Se compone por un conjunto de APIS, componentes de software, herramientas y flujos de trabajo provisto por el desarrollador de la arquitectura del microcontrolador a utilizar.
 - HAL: Hardware Abstraction Layer. Conjunto de bibliotecas de software provistas por STMicroelectronics para simplificar la interacción con el hardware de microcontroladores STM32.
 - UI: Interfaz de usuario (User Interface).
 - UART: Universal Asynchronous Receiver/Transmitter.
 - I2C: Inter-Integrated Circuit.
 - SPI: Serial Peripheral Interface.
 - ADC: Analogic to Digital Converter.
 - PWM: Pulse Width Modulation.
 - GPIO: General-purpose Input/Output.
 - ST-Link: programador y depurador integrado en placas STM32.
 - N/A: No Aplica.

2. Definiciones:

- SYS_SPLASH (estado del sistema): logo de bienvenida en pantalla.
- SYS_MAIN_MENU (estado del sistema): menú principal interactivo.
- SYS_IN_GAME (estado del sistema): juego activo.
- SYS_PAUSED (estado del sistema): juego pausado, menú de opciones de guardado y salida.

1.4. Referencias

1. Estándar IEEE Std. 830-1998.
2. [Plan de proyecto del trabajo práctico final](#) para la *Carrera de Especialización en Sistemas Embebidos* (RETRO_GAME-PP-v5).
3. Especificaciones de requisitos de hardware: RETRO_GAME-RH-vA.

1.5. Visión general del documento

1. Este documento se realiza siguiendo el estándar IEEE Std. 830-1998.

2. Descripción general del documento

2.1. Perspectiva del producto

1. El software especificado en este documento forma parte del sistema embebido *Videojuego portátil inspirado en consolas retro* a desarrollar como trabajo final de la *Carrera de Especialización en Sistemas Embebidos*.
2. Este software se encargará del control de entrada/salida, renderizado gráfico, retroalimentación háptica, reproducción de sonido y gestión del flujo de juego.
3. El software interactúa directamente con componentes de hardware como pantalla, buzzer o parlante, motor de vibraciones, botones físicos, joystick analógico, acelerómetro y memoria no volátil externa.
4. Este software está diseñado para operar de manera autónoma dentro del sistema embebido y no depende de otros sistemas externos para su funcionamiento.
5. La figura 1 presenta un diagrama de arquitectura en capas que contextualiza el software objeto de esta ERS dentro de la consola. Se distinguen:
 - a) Capas provistas por la plataforma
 - CMSIS + HAL: bibliotecas oficiales para manejo del hardware del microcontrolador.
 - Middlewares de terceros: entre ellos el RTOS a utilizar. Estos componentes se encuentran validados por la comunidad o por el proveedor y no forman parte del desarrollo a realizar, aunque serán configurados e integrados.
 - b) Capas a desarrollar
 - Drivers de hardware: para el manejo de los periféricos de entrada y salida. Se desarrollarán y/o adaptarán drivers de código abierto de la comunidad.
 - Lógica del sistema: se desarrollarán servicios comunes que coordinarán tareas entre drivers y el juego. Entre ellos:
 - Manejo de persistencia y configuraciones generales.

- Gestión de eventos (de entrada y otros generales).
 - Motores de generación de salidas (audio, gráficos y vibraciones).
 - Debug y logs.
- Lógica propia del juego: desarrollo e implementación del demo de juego mediante la gestión de estados internos, entradas y salidas específicas del juego.

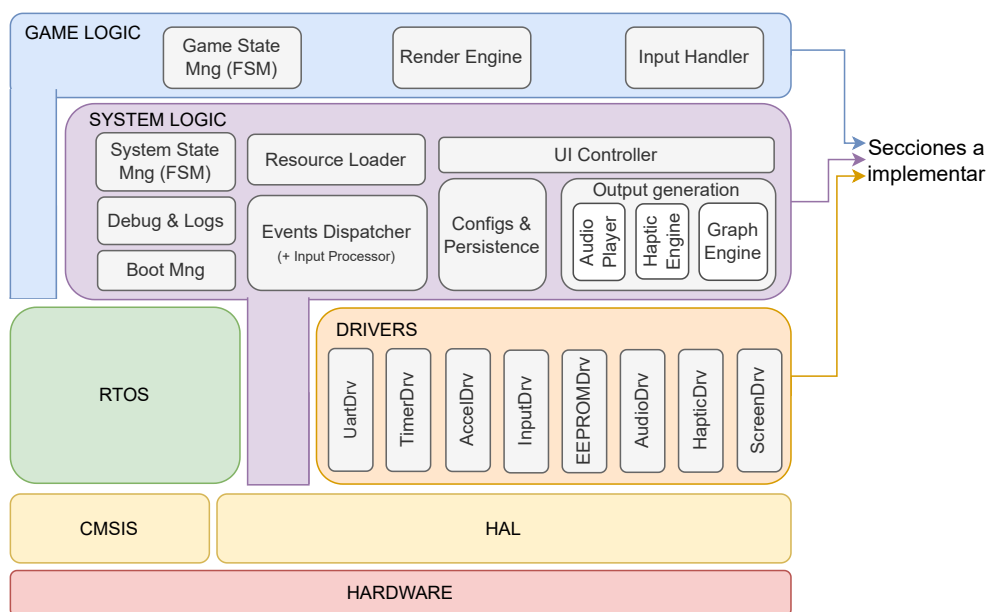


Figura 1: Diagrama de capas del sistema.

2.2. Funciones del producto

1. El software aquí especificado brindará las siguientes funcionalidades:
 - a) Gestión y procesamiento de las entradas mediante botones, joystick y acelerómetro.
 - b) Gestión y generación de salidas de imagen, audio y vibración.
 - c) Gestión de la persistencia del estado del juego en memoria no volátil.
 - d) Gestión de partidas guardadas (selección, eliminación).
 - e) Implementación de un demo de juego de simulación de vuelo en aeronave.
 - f) Gestión de comunicaciones mediante UART para debug.
2. El software aquí especificado no brindará los servicios de:
 - a) Conectividad externa con otros dispositivos más allá de los requeridos para programación del firmware y/o debug.
 - b) Funcionalidades multijugador.
 - c) Lectura o integración de otros juegos.
 - d) Gestión o integración con periféricos diferentes a los mencionados en el documento de requisitos de hardware RETRO_GAME-RH-vA.

2.3. Características de los usuarios

1. Los usuarios de este producto serán personas interesadas en consolas de juegos portátiles.
2. Se requiere la destreza básica para manejar botones, el joystick y la inclinación de la consola.

2.4. Restricciones

1. El software deberá mantenerse bajo control de versiones.
2. El software deberá integrar apropiadamente los periféricos de entrada y salida descritos en el documento RETRO_GAME-RH-vA.

2.5. Suposiciones y dependencias

1. Se supone que se dispondrá de todos los componentes y de hardware a utilizar desde el comienzo de la fase de análisis hasta la finalización del desarrollo del software para la defensa pública.

2.6. Requisitos futuros

1. Incorporación de funcionalidades de gestión de la energía, suspensión y/o lecturas del nivel de batería.
2. Ampliación del demo de juego para incorporar nuevos eventos, desafíos o niveles.
3. Incorporación de opciones de configuración (volumen de audio, brillo, intensidad de vibración).

3. Requisitos específicos

3.1. Interfaces externas

RETRO_GAME-RS-REQ0001: el software deberá comunicarse por I2C con una memoria EEPROM.

RETRO_GAME-RS-REQ0002: el software deberá comunicarse por SPI con una pantalla LED.

RETRO_GAME-RS-REQ0003: el software deberá comunicarse por I2C con un acelerómetro.

RETRO_GAME-RS-REQ0004: el software deberá generar una señal de audio mediante PWM hacia un amplificador externo.

RETRO_GAME-RS-REQ0005: el software deberá comunicarse por I2C con el módulo de vibración.

RETRO_GAME-RS-REQ0006: el software deberá adquirir entradas digitales mediante pines GPIO desde múltiples botones físicos.

RETRO_GAME-RS-REQ0007: el software deberá adquirir señales analógicas desde un joystick de dos ejes mediante entradas ADC.

RETRO_GAME-RS-REQ0008: el sistema deberá exponer, a través del conector USB del ST-Link de la placa, una interfaz UART para enviar logs de depuración a la PC. La consola de depuración deberá estar disponible en los estados SYS_SPLASH, SYS_MAIN_MENU, SYS_IN_GAME y SYS_PAUSED.

RETRO_GAME-RS-REQ0009: la consola de depuración deberá aceptar los comandos de texto «logon» y «logoff» para activar o silenciar la generación de trazas en tiempo de ejecución.

3.2. Funciones

3.2.1. Flujo de encendido

RETRO_GAME-RS-REQ0010: el sistema deberá entrar en estado `SYS_SPLASH` dentro de los 250 ms posteriores al encendido o reset.

RETRO_GAME-RS-REQ0011: el sistema deberá permanecer en el estado `SYS_SPLASH` durante no más de 1500 ms y luego pasar al estado `SYS_MAIN_MENU`, mostrando al menos las opciones «Iniciar nueva partida».

RETRO_GAME-RS-REQ0012: si se detecta una partida guardada válida, el estado `SYS_MAIN_MENU` deberá mostrar bajo la opción «Iniciar nueva partida» la aclaración de que se descartará la partida guardada.

RETRO_GAME-RS-REQ0013: si se detecta una partida guardada válida, el estado `SYS_MAIN_MENU` deberá mostrar además la opción «Continuar».

RETRO_GAME-RS-REQ0014: al seleccionar alguna de las opciones de inicio, se deberá pasar al estado `SYS_IN_GAME`.

RETRO_GAME-RS-REQ0015: al pasar a `SYS_IN_GAME` desde «Iniciar nueva partida», el sistema deberá descartar la partida guardada e iniciar una nueva partida.

RETRO_GAME-RS-REQ0016: al pasar a `SYS_IN_GAME` desde «Continuar», el sistema deberá cargar la partida guardada y restaurar el estado del juego antes de mostrar el primer fotograma.

3.2.2. Flujo de pausa y opciones

RETRO_GAME-RS-REQ0017: al pulsar el botón `START` durante `SYS_IN_GAME`, el sistema deberá entrar en el estado `SYS_PAUSED` en un tiempo no mayor a 500 ms y mostrar un menú con las opciones «Continuar», «Guardar» y «Salir».

RETRO_GAME-RS-REQ0018: mientras el sistema se encuentre en `SYS_PAUSED`, el juego permanece detenido, no debe correr el tiempo ni modificar ninguna variable de estado.

RETRO_GAME-RS-REQ0019: al seleccionar «Continuar», el sistema deberá regresar al juego en el estado `SYS_IN_GAME` en un tiempo no mayor a 500 ms y reanudar todas las acciones detenidas.

RETRO_GAME-RS-REQ0020: al seleccionar «Guardar», el sistema deberá guardar la partida en la EEPROM en un tiempo no mayor a 500 ms, mostrar en pantalla el mensaje «Guardado exitoso» durante al menos 1000 ms y permanecer en `SYS_PAUSED`.

RETRO_GAME-RS-REQ0021: si la operación de guardado falla, el sistema deberá mostrar «Error al guardar» durante al menos 1000 ms y permanecer en `SYS_PAUSED`.

RETRO_GAME-RS-REQ0022: al seleccionar la opción «Salir», el sistema deberá descartar los cambios no guardados y pasar al estado `SYS_MAIN_MENU` en un tiempo no mayor a 1000 ms.

3.2.3. Bucle de juego

RETRO_GAME-RS-REQ0023: el sistema deberá ejecutar el bucle de juego en el estado `SYS_IN_GAME` a una frecuencia mínima sostenida de 20 FPS (período no mayor a 50 ms) bajo la carga típica del demo.

RETRO_GAME-RS-REQ0024: en cada cuadro, los eventos de entrada (botones, joystick, acelerómetro) deberán despacharse a la lógica del juego en un tiempo no mayor a 10 ms desde su captura por los drivers.

RETRO_GAME-RS-REQ0025: la lógica del juego deberá generar la lista de comandos de render y entregarla al driver de pantalla en un tiempo acumulado no mayor a 10 ms dentro del mismo cuadro.

RETRO_GAME-RS-REQ0026: cuando la lógica del juego solicite la reproducción de un efecto de sonido, el sistema deberá iniciar dicho efecto en la salida de audio en un tiempo no mayor a 10 ms.

RETRO_GAME-RS-REQ0027: cuando la lógica del juego dispare un evento de vibración, el sistema deberá activar el patrón de vibración correspondiente en un tiempo no mayor a 10 ms.

3.3. Requisitos de rendimiento

RETRO_GAME-RS-REQ0028: el sistema deberá mantener la frecuencia de cuadro dentro de $\pm 5\%$ de 20 FPS durante al menos el 95 % del tiempo de ejecución continuo en SYS_IN_GAME.

3.4. Restricciones de diseño

RETRO_GAME-RS-REQ0029: el software deberá correr en la placa STM32 NUCLEO-F446RE.

RETRO_GAME-RS-REQ0030: se deberá utilizar el sistema operativo FreeRTOS para el manejo de las tareas del sistema.

3.5. Atributos del sistema

N/A.

3.6. Otros requisitos

N/A.

4. Apéndices

N/A.