



Videojuego portátil inspirado en consolas retro

Especificación de casos de uso

Autor:

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

13/07/2025

VERSIÓN A

*Este documento fue creado 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	13/07/2025	Creación del documento	Lic. Jezabel Danon	

Índice

1. Introducción	3
1.1. Propósito	3
1.2. Definiciones, Acrónimos y Abreviaturas	3
1.3. Referencias	3
1.4. Visión general del documento	3
2. Descripción general del documento	3
2.1. Perspectiva del producto	3
2.2. Funciones del producto	3
2.3. Características de los usuarios	4
3. Casos de uso	4
3.1. Caso de uso 1	5
3.2. Caso de uso 2	6
3.3. Caso de uso 3	7
4. Apéndices	7

1. Introducción

1.1. Propósito

1. Este documento representa una especificación de casos de uso 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. Definiciones, Acrónimos y Abreviaturas

1. UART: Universal Asynchronous Receiver/Transmitter.
2. N/A: No Aplica.

1.3. Referencias

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

1.4. Visión general del documento

1. Este documento se estructura siguiendo guías y buenas prácticas para las definiciones de casos de uso como *Writing Effective Use Cases* (Alistair Cockburn, Addison-Wesley, 2001) y *Use Cases 2.0* (Jacobson, Spence & Bittner, 2011).

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.

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.

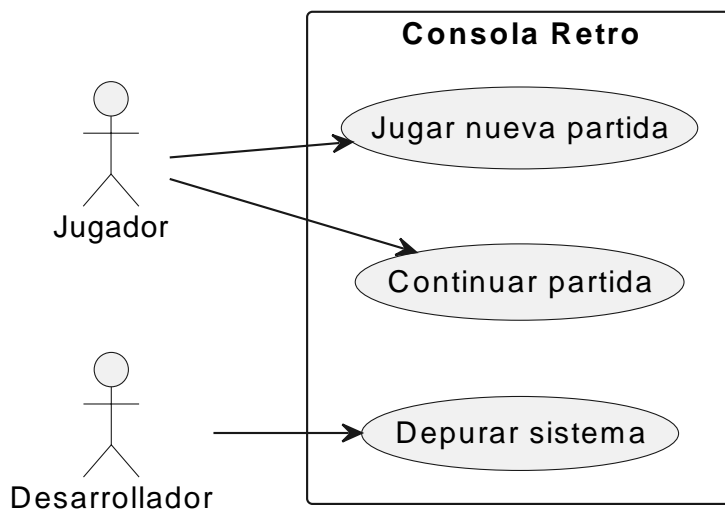
2. El software aquí especificado no brindará los servicios de:

- 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. El usuario principal de este producto será cualquier persona interesada en consolas de juegos portátiles.
2. Un usuario adicional será el desarrollador del software que requiera habilitar o deshabilitar las trazas para depuración desde la consola UART.

3. Casos de uso



3.1. Caso de uso 1

Jugar nueva partida	
1. Nombre	Jugar nueva partida
1.1 Breve descripción	El jugador inicia el sistema y juega hasta que decide salir, pudiendo guardar la partida o no.
1.2 Actor principal	Jugador
1.3 Disparador	Encendido de la consola desde el switch de encendido.
2. Flujo de eventos	
2.1 Flujo básico	<ol style="list-style-type: none">1. El sistema muestra la pantalla de bienvenida.2. Luego de un tiempo máximo de 1,5 segundos, el sistema muestra el menú principal.3. El jugador selecciona la opción «Iniciar nueva partida».4. El sistema inicia el juego.5. El jugador juega el demo de simulación de vuelo.6. El jugador pausa el juego presionando el botón START en cualquier momento de la partida.7. El sistema muestra el menú de pausa con las opciones «Continuar», «Guardar» y «Salir».8. El jugador selecciona la opción «Guardar».9. El sistema muestra el mensaje «Guardado exitoso» y luego de un segundo muestra nuevamente el menú de pausa.10. El jugador selecciona la opción «Salir».11. El sistema sale del juego y retorna al menú principal.12. El jugador puede apagar la consola desde el switch de encendido.
2.2 Flujos alternativos	<ol style="list-style-type: none">1. Seguir jugando luego de la pausa.<ol style="list-style-type: none">a) Desde el menú de pausa el jugador presiona la opción «Continuar».b) El juego retoma desde donde fué pausado.2. Guardado fallido.<ol style="list-style-type: none">a) Al presionar la opción «Guardar» se muestra el mensaje «Error al guardar» y luego de un segundo se muestra nuevamente el menú de pausa.
3. Requerimientos especiales	N/A
4. Pre-condiciones	Consola apagada.
5. Post-condiciones	Consola apagada.

3.2. Caso de uso 2

Continuar partida guardada	
1. Nombre	Continuar partida guardada
1.1 Breve descripción	El jugador retoma una partida previamente guardada.
1.2 Actor principal	Jugador
1.3 Disparador	Encendido de la consola desde el switch de encendido.
2. Flujo de eventos	
2.1 Flujo básico	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla de bienvenida. 2. Luego de un tiempo máximo de 1,5 segundos, el sistema muestra el menú principal con las opciones «Iniciar nueva partida» y «Continuar». 3. El jugador selecciona la opción «Continuar». 4. El sistema inicia el juego cargando el estado de la partida guardada. 5. El jugador continúa jugando su partida hasta que decide finalizar. 6. El jugador pausa el juego presionando el botón START en cualquier momento de la partida. 7. El sistema muestra el menú de pausa con las opciones «Continuar», «Guardar» y «Salir». 8. El jugador selecciona la opción «Guardar». 9. El sistema muestra el mensaje «Guardado exitoso» y luego de un segundo muestra nuevamente el menú de pausa. 10. El jugador selecciona la opción «Salir». 11. El sistema sale del juego y retorna al menú principal. 12. El jugador puede apagar la consola desde el switch de encendido.
2.2 Flujos alternativos	<ol style="list-style-type: none"> 1. Seguir jugando luego de la pausa. <ol style="list-style-type: none"> a) Desde el menú de pausa el jugador presiona la opción «Continuar». b) El juego retoma desde donde fué pausado. 2. Guardado fallido. <ol style="list-style-type: none"> a) Al presionar la opción «Guardar» se muestra el mensaje «Error al guardar» y luego de un segundo se muestra nuevamente el menú de pausa.
3. Requerimientos especiales	Partida previamente guardada.
4. Pre-condiciones	Consola apagada.
5. Post-condiciones	Consola apagada.

3.3. Caso de uso 3

Depurar el sistema	
1. Nombre	Depurar el sistema
1.1 Breve descripción	El desarrollador habilita la salida de logs por la consola UART/USB, permitiendo observar el comportamiento interno del sistema en tiempo real.
1.2 Actor principal	Desarrollador
1.3 Disparador	El desarrollador envía el comando <code>logon</code> en la consola de depuración.
2. Flujo de eventos	
2.1 Flujo básico	<ol style="list-style-type: none">1. La consola UART está disponible en cualquier estado del sistema.2. El desarrollador envía el comando <code>logon</code> por la consola.3. El sistema reconoce el comando, establece <code>logEnabled = true</code> y responde <code>LOG ON</code>.4. El sistema comienza a transmitir las trazas mientras <code>logEnabled</code> sea <code>true</code>.5. El desarrollador envía el comando <code>logoff</code>.6. El sistema reconoce el comando, establece <code>logEnabled = false</code> y responde <code>LOG OFF</code>.7. El sistema no envía trazas por la consola.
2.2 Flujos alternativos	<ol style="list-style-type: none">1. <i>Comando desconocido</i>: el sistema responde <code>ERR CMD</code> y mantiene el estado previo.
3. Requerimientos especiales	La consola está conectada a 115 200 bps con configuración 8N1.
4. Pre-condiciones	Al iniciar el sistema las trazas se encuentran deshabilitadas.
5. Post-condiciones	N/A

4. Apéndices

N/A.