



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN SOFTWARE

Periodo Académico: 2025-B

Fecha de Entrega: 20/11/2025

Asignatura: Desarrollo de Juegos Interactivos

Grupo: GR1SW

Integrantes: Guerra Sebastián, Morales Anthony, Proaño Estéfano

Ficha de Análisis de GDD

1. Juego y Fuente:

- **Juego:** Silent Hill 2
- **Fuente:** [GDD – Silent Hill 2](#)

2. Visión Central:

El juego es una experiencia de terror psicológico en tercera persona donde el protagonista, James Sunderland, regresa al pueblo de Silent Hill tras recibir una carta de su esposa fallecida, debiendo sobrevivir a monstruos grotescos y resolver acertijos para desentrañar la verdad detrás de su "llamada". Se diferencia de otros títulos (como Resident Evil) enfocándose más en el terror psicológico, la soledad y una atmósfera opresiva que en la acción directa.

3. Resumen Estructural (Puntos Clave):

El documento se estructura en las siguientes secciones lógicas:

- **Mecánicas y Dinámicas (Core Game Play):** Define el ciclo de juego basado en la exploración, combate (evasivo vs ofensivo) y resolución de puzzles. Incluye detalles sobre la física del movimiento (caminar/correr/fatiga) y estadísticas.
- **Personajes y Entidades:** Fichas detalladas de los personajes (James, Maria, Eddie, Angela) y un bestiario completo con tablas de daño específicas por arma y dificultad.
- **Armamento e Ítems:** Propiedades técnicas de armas cuerpo a cuerpo y de fuego (rango, velocidad, daño) y gestión de ítems de salud.

- **Interfaz y Flujo (UI/UX):** Diagramas de flujo (flowcharts) detallados para los menús de inicio, opciones e inventario "in-game".
- **Mundo y Niveles:** Mapas detallados de las zonas principales (Apartamentos, Hospital, Prisión, Hotel) con la ubicación de enemigos y eventos clave.

4. Análisis de Ingeniería:

- **Lo más claro/útil:** La sección de "Monsters" y "Weapon Properties" es de gran utilidad para un ingeniero. Las tablas de daño ("Hits to Kill") y los datos de *frame data* implícitos (velocidad de ataque, rango) permiten programar las clases de combate y balancear el juego desde el día 1 sin adivinar valores. Los diagramas de flujo de la UI también facilitan enormemente la implementación del frontend.
- **Lo que falta/Ambigüedades:** Desde una perspectiva técnica moderna, la sección de "Artificial Intelligence" es un poco vaga. Describe comportamientos basados en "sentidos" (vista/oído), pero no define máquinas de estado claras para los enemigos (ej. ¿cuánto tiempo persiguen antes de desistir? ¿cómo navegan el *pathfinding* en pasillos estrechos?). También faltan Requerimientos No Funcionales críticos: presupuestos de memoria, tiempos de carga máximos entre las zonas "mundo real" y "otro mundo", o especificaciones del motor de niebla.

Lista de Épicas – Silent Hill 2

Nro. EP001	Título: Movimiento y Fatiga
Historia de Usuario: Como Jugador, quiero que James pueda moverse, y además, que su velocidad disminuya si corre demasiado tiempo, para añadir realismo y tensión a la huida.	

Nro. EP002	Título: Sistema de Combate Melee
Historia de Usuario: Como Jugador, quiero poder atacar con armas contundentes, para defenderme de enemigos a corto alcance.	

Nro. EP003	Título: Combate a Distancia
Historia de Usuario: Como Jugador, quiero apuntar y disparar armas, para eliminar amenazas desde una distancia segura.	

Nro. EP004	Título: Mecánica Iluminación en Zonas Oscuras
Historia de Usuario: Como Jugador, quiero iluminar zonas oscuras, para aumentar mi visibilidad del entorno.	

Nro. EP005	Título: Mecánica Advertencia Auditiva de Enemigos
Historia de Usuario: Como Jugador, quiero que suene una advertencia cuando hay monstruos cerca, para tener una advertencia auditiva de peligro inminente.	

Nro. EP006	Título: Gestión de Inventario
Historia de Usuario: Como Jugador, quiero gestionar mi inventario, para organizar mis recursos estratégicamente.	

Nro. EP007	Título: Interacción con Mapas
Historia de Usuario: Como Jugador, quiero ver un mapa de la zona en donde me encuentre, para orientarme a lo largo del juego.	

Nro. EP008	Título: Boss - Pyramid Head
Historia de Usuario: Como IA, quiero que 'Pyramid Head' sea invencible y persiga al jugador en encuentros específicos, para generar tensión extrema.	

Nro. EP009	Título: IA Enemigo Básico - Lying Figure
Historia de Usuario: Como IA, quiero que la 'Lying Figure' se arrastre rápido por el suelo o camine y escupa ácido, para obligar al jugador a usar estrategias de sigilo o combate.	

Nro. EP010	Título: Sistema de Puzzles
Historia de Usuario: Como Diseñador, quiero implementar puzzles que requieran encontrar ítems y descifrar acertijos, para desbloquear progreso en la historia.	

Nro. EP011	Título: Sistema de Salud
Historia de Usuario: Como Jugador, quiero que la salud de James se represente visualmente, para gestionar el estado de 'Game Over'.	

Nro. EP012	Título: Sistema de Daño
Historia de Usuario: Como Jugador, quiero que la salud de James disminuya al ser atacado por un enemigo, para aumentar dificultad al juego.	

Nro. EP013	Título: Transición de Mundos
Historia de Usuario: Como Sistema, quiero hacer transiciones de mundos, para agregar más suspenso y tensión al ambiente.	

Lista de Historias de Usuario

Nro. HU01	Título: Desplazamiento Estándar
Historia de Usuario: Como jugador, quiero controlar el movimiento de James usando el stick analógico sin presionar botones adicionales, para navegar por el entorno a un paso seguro que no consuma energía.	
Criterios de aceptación <ul style="list-style-type: none">• AC1: El personaje debe moverse a una velocidad constante predefinida como "Walk Speed" cuando el stick analógico se inclina más del 20% de su zona muerta.• AC2: El movimiento no debe reducir el valor actual de la variable de resistencia.	

Nro. HU02	Título: Activación de Carrera
Historia de Usuario: Como jugador, quiero aumentar la velocidad de James manteniendo presionado el botón "Cuadrado" mientras me muevo, para escapar de enemigos o cubrir distancias largas rápidamente.	
Criterios de aceptación <ul style="list-style-type: none">• AC1: Al mantener presionado el botón asignado a correr, la velocidad de desplazamiento debe cambiar a "Run Speed" inmediatamente.• AC2: La acción de correr debe consumir la variable de resistencia a una tasa constante por segundo.	

Nro. HU03	Título: Degradación de Velocidad por Cansancio
Historia de Usuario: Como sistema, quiero forzar a James a cambiar de "Correr" a "Trotar" automáticamente si ha corrido por un tiempo prolongado, para simular el agotamiento físico descrito en las reglas de física del juego.	
Criterios de aceptación	

- **AC1:** Cuando la variable de resistencia llega a cero, el estado de movimiento debe cambiar forzosamente de "Run" a "Jog" sin intervención del jugador.
- **AC2:** El sistema debe bloquear el retorno al estado "Run" hasta que la resistencia se recupere por encima de un umbral mínimo definido.

Nro. HU04	Título: Recuperación de Stamina
Historia de Usuario: Como jugador, quiero recuperar mi capacidad de correr más rápido si camino en lugar de trotar, para gestionar estratégicamente mi energía antes de un encuentro peligroso.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: La variable de resistencia debe regenerarse progresivamente cuando el personaje se encuentra en estado "Idle" o "Walk". • AC2: La tasa de recuperación debe ser mayor al estar quieto que al caminar. 	

Nro. HU05	Título: Movimientos Especiales
Historia de Usuario: Como jugador, quiero realizar pasos laterales con L1/R1 y un giro de 180 grados al presionar ambos, para esquivar ataques en pasillos estrechos o huir inmediatamente de una amenaza.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Presionar los botones L1 o R1 individualmente debe ejecutar un desplazamiento lateral sin rotar la cámara. • AC2: Presionar L1 y R1 simultáneamente debe ejecutar una animación de giro de 180 grados y rotar la transformación del personaje en la dirección opuesta. 	

Nro. HU06	Título: Reducción de Movilidad por Heridas
Historia de Usuario: Como Sistema, quiero reducir la velocidad máxima de movimiento si la salud de James cae por debajo de un umbral crítico, para comunicar visualmente el estado de peligro y urgencia al jugador.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe monitorear la variable de salud y aplicar un multiplicador de reducción de velocidad si el valor es inferior al umbral crítico. • AC2: La animación de locomoción debe sustituirse por la variante de "lesionado" al cumplirse la condición de salud baja. 	

Nro. HU07	Título: Postura de Combate
Historia de Usuario: Como Jugador, quiero mantener presionado un botón (R2) para entrar en "Modo Combate", para encarar automáticamente al enemigo más cercano y habilitar las acciones ofensivas.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Mantener presionado R2 debe activar el estado de guardia, fijando la rotación del personaje hacia el objetivo hostil más cercano. • AC2: Las acciones de ataque solo deben ser ejecutables mientras el estado de guardia esté activo. 	

Nro. HU08	Título: Distinción de Ataque Ligero y Pesado
Historia de Usuario: Como Jugador, quiero que el sistema distinga entre presionar brevemente o mantener presionado el botón de acción, para ejecutar ataques rápidos con poco daño o ataques lentos con alto daño según la situación.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Una pulsación del botón de acción menor a 0.2 segundos debe registrarse como "Ataque Ligero". • AC2: Una pulsación mayor a 0.2 segundos debe registrarse como "Ataque Pesado" y ejecutar la animación correspondiente. 	

Nro. HU09	Título: Atacar con tabla
Historia de Usuario: Como Jugador, quiero que la tabla permita realizar dos golpes en sucesión, para equilibrarla como el arma inicial de bajo daño.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: La animación de ataque con la tabla debe permitir encadenar un máximo de dos impactos consecutivos antes de reiniciar el ciclo. • AC2: El rango de colisión del ataque debe limitarse a la distancia corta definida en la tabla de propiedades de armas. 	

Nro. HU10	Título: Atacar con tubería de acero
Historia de Usuario: Como Sistema, quiero desplazar a James un pequeño paso hacia adelante con cada golpe, para simular el impulso del peso del arma.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Cada ejecución de la animación de ataque debe aplicar un vector de movimiento hacia adelante en la posición del personaje. • AC2: El daño base aplicado debe corresponder a los valores medios definidos para "Steel Pipe" en la tabla de daño. 	

Nro. HU11	Título: Restricción de Movimiento con el "Great Knife"
Historia de Usuario: Como Sistema, quiero reducir drásticamente la velocidad de movimiento del jugador cuando equipe el "Great Knife", para balancear su capacidad de eliminar enemigos de un solo golpe.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Equipar el Great Knife debe sobrescribir la velocidad de movimiento base del jugador a un valor muy lento. • AC2: El jugador debe quedar inmovilizado completamente durante la ejecución de la animación de ataque. 	

Nro. HU12	Título: Golpe de gracia
Historia de Usuario: Como Jugador, quiero poder pisotear a un enemigo cuando este ha sido derribado al suelo, para eliminarlo definitivamente antes de que recupere su estado.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: La acción debe estar disponible únicamente cuando el colisionador del enemigo intersecta con el jugador y el enemigo tiene el estado "Derribado". • AC2: La ejecución exitosa de esta acción debe cambiar el estado del enemigo a "Muerto" y desactivar su componente de IA. 	

Nro. HU13	Título: Mecánica de Apuntado y Movilidad
Historia de Usuario: Como Sistema, quiero bloquear el movimiento del jugador al apuntar con el "Rifle de Caza", para balancear el poder de fuego con la vulnerabilidad.	
Criterio de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe validar el tipo de arma equipada antes de procesar el input de movimiento durante el apuntado. • AC2: Si el arma es el Rifle de Caza, el input de movimiento debe ser ignorado mientras el botón de apuntado esté presionado. 	

Nro. HU14	Título: Control de Cadencia de Disparo
Historia de Usuario: Como Diseñador, quiero configurar intervalos de tiempo específicos entre disparos, para diferenciar el ritmo táctico de cada arma según la tabla de propiedades de fuego.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe impedir un nuevo disparo hasta que transcurra el tiempo de enfriamiento específico para el arma actual. • AC2: El tiempo de enfriamiento debe ser de 0.5 segundos para la Pistola y 2.0 segundos para la Escopeta. 	

Nro. HU15	Título: Lógica de Recarga y Vulnerabilidad
Historia de Usuario: Como Jugador, quiero que la acción de recargar me deje expuesto a ataques sin poder defenderme durante la animación, para añadir tensión estratégica al uso de armas de fuego.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: La animación de recarga debe activarse automáticamente al intentar disparar con el cargador vacío. • AC2: Todas las entradas de movimiento y ataque deben deshabilitarse hasta que la animación de recarga concluya. 	

Nro. HU16	Título: Consumo y Capacidad de Cargadores
Historia de Usuario: Como Sistema, quiero limitar la capacidad del cargador según el arma, para forzar al jugador a gestionar sus recursos.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Cada disparo debe reducir en una unidad el contador de munición del arma equipada. • AC2: El sistema debe impedir la recarga si el contador de munición actual es igual a la capacidad máxima del arma. 	

Nro. HU17	Título: Matriz de Movimiento al Disparar
Historia de Usuario: Como Sistema, quiero permitir caminar mientras se dispara solo con la Pistola, para respetar la tabla de diseño que diferencia armas ligeras de pesadas.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El personaje debe poder desplazarse mientras ejecuta la animación de disparo si porta la Pistola. • AC2: El personaje debe detenerse automáticamente al ejecutar la animación de disparo si porta la Escopeta. 	

Nro. HU18	Título: Activación/Desactivación de Linterna
Historia de Usuario: Como Jugador, quiero encender y apagar la linterna presionado el botón "Círculo", para decidir tácticamente entre tener visibilidad o mantener el sigilo según la situación.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Presionar el botón Círculo debe alternar el estado del componente de luz entre activado y desactivado. • AC2: El cambio de estado debe ser instantáneo y reflejarse en el modelo del personaje (linterna en el pecho). 	

Nro. HU19	Título: Cono de Luz Dinámico
Historia de Usuario: Como Jugador, quiero que el haz de luz revele ítems y monstruos que son invisibles o difíciles de ver en la oscuridad, para poder interactuar con el entorno y anticipar amenazas.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: La luz debe proyectar sombras dinámicas y revelar objetos marcados con la capa "Oculto en Oscuridad". • AC2: La dirección del cono de luz debe seguir la rotación del torso del personaje. 	

Nro. HU20	Título: Simulación de Adaptación a la Oscuridad
Historia de Usuario: Como Sistema, quiero aumentar gradualmente el brillo de la pantalla si el jugador mantiene la linterna apagada en una zona oscura por un tiempo, para imitar el efecto fisiológico de los ojos acostumbrándose a la oscuridad.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe incrementar el valor de exposición de la cámara progresivamente tras permanecer en una zona oscura sin linterna. • AC2: Activar la linterna debe restablecer el valor de exposición a su estado base inmediatamente. 	

Nro. HU21	Título: Impacto de Luz en Radio de Detección
Historia de Usuario: Como Sistema, quiero expandir drásticamente el radio de detección de los enemigos cuando la linterna está encendida, para penalizar al jugador haciéndolo "más fácil de ser visto por el peligro".	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El colisionador de detección de la IA enemiga debe aumentar su radio si el jugador tiene la linterna encendida. • AC2: El sistema de IA debe priorizar al jugador iluminado sobre otros objetivos o estados de patrulla. 	

Nro. HU22	Título: Modulación de Volumen por Distancia
Historia de Usuario: Como Sistema, quiero calcular la distancia entre James y los enemigos activos para reproducir un sonido de estática, para alertar al jugador de amenazas fuera de su campo de visión.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe calcular en cada frame la distancia lineal al enemigo activo más próximo. • AC2: El volumen del clip de audio de estática debe ajustarse dinámicamente en proporción inversa a la distancia calculada. 	

Nro. HU23	Título: Atracción de Enemigos por Ruido
Historia de Usuario: Como Sistema, quiero que el radio funcione como un "faro" auditivo que atraiga a los enemigos cercanos, para implementar la mecánica de riesgo donde el radio alerta al jugador pero también revela su posición.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Los enemigos deben cambiar a estado de alerta si el jugador entra en su radio auditivo con la radio encendida. • AC2: La IA debe trazar una ruta hacia la posición del jugador al detectar el ruido de la radio. 	

Nro. HU24	Título: Control de Encendido/Apagado
Historia de Usuario: Como Jugador, quiero poder apagar el radio desde el inventario, para entrar en modo sigilo y evitar atraer la atención de los monstruos.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El jugador debe poder cambiar el estado de la radio desde el menú de inventario. • AC2: Al apagar la radio, el efecto de sonido de estática debe detenerse y la atracción de enemigos por este factor debe cesar. 	

Nro. HU25	Título: Restricción de Señal
Historia de Usuario: Como Diseñador, quiero restringir el radio para que solo emita "ruido blanco" y no sintonice transmisiones normales.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: La radio no debe sintonizar ninguna estación de audio o música, limitándose a reproducir ruido blanco. • AC2: El audio de estática solo debe reproducirse si hay enemigos instanciados en el nivel actual. 	

Nro. HU26	Título: Acceso al Inventario y Pausa
Historia de Usuario: Como Jugador, quiero presionar el botón "Start" para abrir el menú de inventario y detener el flujo del juego.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Presionar Start debe renderizar la interfaz de usuario del inventario y establecer la escala de tiempo del juego a cero. • AC2: Cerrar el inventario debe reanudar la escala de tiempo y ocultar la interfaz. 	

Nro. HU27	Título: Clasificación de Recursos (Pestañas)
Historia de Usuario: Como Diseñador, quiero que el inventario separe visualmente los "Items" del "Equipamiento", para facilitar la búsqueda rápida.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe filtrar y mostrar los objetos en listas separadas según su tipo: Consumibles, Equipo y Objetos Clave. • AC2: La navegación entre pestañas debe realizarse con los botones direccionales laterales. 	

Nro. HU28	Título: Sistema de Acciones (Command Pane)
Historia de Usuario: Como Sistema, quiero desplegar una lista de acciones específicas al seleccionar un ítem, para interactuar con el objeto según su tipo.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Seleccionar un ítem debe abrir un submenú con las opciones aplicables a ese objeto específico. • AC2: Las opciones no válidas para el objeto seleccionado deben aparecer deshabilitadas o no mostrarse. 	

Nro. HU29	Título: Uso de Ítems de Curación
Historia de Usuario: Como Sistema, quiero aplicar valores exactos de recuperación al usar ítems, para cumplir con las reglas de balanceo de salud.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Usar una "Health Drink" debe incrementar la salud actual en un 25% del total. • AC2: Usar un "First Aid Kit" debe incrementar la salud actual en un 50% del total. 	

Nro. HU30	Título: Combinación de Objetos
Historia de Usuario: Como Jugador, quiero seleccionar la opción "Combinar" para fusionar dos objetos clave del inventario necesarios para resolver acertijos.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe permitir seleccionar un segundo objeto tras elegir la opción "Combinar" en el primero. • AC2: Si la combinación es válida según la tabla de datos, ambos objetos deben eliminarse y añadirse el objeto resultante al inventario. 	

Nro. HU31	Título: Inspección de Ítems
Historia de Usuario: Como Jugador, quiero usar la opción "Examinar" en cualquier objeto, para recibir una descripción sobre su utilidad.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: La opción "Examinar" debe mostrar un modelo 3D rotatable del objeto o una imagen detallada. • AC2: Debe mostrarse un texto descriptivo o pista asociada al objeto en la parte inferior de la pantalla. 	

Nro. HU32	Título: Validación de Posesión de Mapa
Historia de Usuario: Como Sistema, quiero verificar si el jugador posee el ítem "Mapa" del área actual, para cumplir con la lógica condicional especificada en el diagrama de flujo del menú.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe consultar el inventario del jugador al solicitar la vista del mapa. • AC2: Si el objeto mapa correspondiente a la zona actual no existe, debe mostrarse el mapa general del pueblo. 	

Nro. HU33	Título: Marcador de Ubicación en Tiempo Real
Historia de Usuario: Como Sistema, quiero renderizar un icono sobre el mapa que represente la posición actual de James en el mundo, para cumplir con el requerimiento funcional de visualizar la ubicación actual.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Se debe renderizar un icono de flecha sobre la interfaz del mapa. • AC2: Las coordenadas y rotación del icono deben actualizarse para coincidir con la posición relativa del jugador en el escenario 3D. 	

Nro. HU34	Título: Control de Interfaz de Mapa
Historia de Usuario: Como Jugador, quiero poder abrir el mapa con el botón "Triángulo", para respetar la configuración de "Player Controls" definida en el documento de diseño.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Presionar Triángulo debe abrir o cerrar la interfaz del mapa sin pausar completamente el juego si el diseño lo permite. • AC2: El mapa debe permitir desplazamiento y zoom mediante los controles analógicos. 	

Nro. HU35	Título: Cambio de Contexto de Mapa
Historia de Usuario: Como Sistema, quiero cambiar automáticamente la textura del mapa mostrada, para asegurar que la orientación corresponda siempre a la zona cargada en memoria.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe detectar el ID de la zona actual al cargar un nuevo nivel. • AC2: La textura del mapa mostrada debe actualizarse automáticamente para corresponder a la nueva zona cargada. 	

Nro. HU36	Título: Sistema de Inmunidad al Daño
Historia de Usuario: Como Sistema, quiero procesar las colisiones de los ataques del jugador sobre "Pyramid Head", para cumplir con la regla de diseño que establece que "James cannot defeat this monster".	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El controlador de salud del Boss debe ignorar cualquier valor de daño recibido de las armas del jugador. • AC2: El sistema debe reproducir efectos visuales y sonoros de impacto al recibir un ataque, a pesar de no restar salud. 	

Nro. HU37	Título: Física de Ataque Pesado (Great Knife)
Historia de Usuario: Como Diseñador, quiero que el ataque del Boss sea extremadamente lento, para reflejar las propiedades del "Great Knife" descritas en la tabla de monstruos.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El ataque del Boss debe tener un tiempo de preparación prolongado antes de activar el colisionador de daño. • AC2: El impacto debe causar un daño masivo capaz de dejar al jugador en estado crítico o muerto según la dificultad. 	

Nro. HU38	Título: Controlador de Encuentro por Tiempo
Historia de Usuario: Como Sistema, quiero activar un temporizador oculto al iniciar el combate, para permitir al jugador progresar sin haber derrotado al enemigo por fuerza bruta.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Al entrar en la zona del Boss, se debe iniciar una cuenta regresiva interna invisible para el jugador. • AC2: Al finalizar la cuenta, el Boss debe ejecutar su comportamiento de retirada y finalizar el encuentro. 	

Nro. HU39	Título: IA de Navegación en Pasillos Estrechos
Historia de Usuario: Como IA, quiero que Pyramid Head camine lentamente, para acorralar a James en espacios cerrados.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El agente de navegación debe calcular rutas en espacios reducidos sin atascarse en la geometría. • AC2: La velocidad de movimiento debe ser lenta pero constante, obligando al jugador a mantener la distancia. 	

Nro. HU40	Título: Ejecución de Eventos Narrativos (Maria)
Historia de Usuario: Como Diseñador, quiero triggerar una animación específica donde el Boss ataca a un NPC en lugar del jugador, para recrear la escena narrativa crítica descrita en el guion del juego.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe detectar si el NPC objetivo está dentro de un volumen de activación específico. • AC2: Se debe forzar una animación sincronizada de ataque letal sobre el NPC y transición a cinemática. 	

Nro. HU41	Título: Dualidad de Estados (De Pie vs. Rastrero)
Historia de Usuario: Como Sistema, quiero gestionar estados de navegación distintos, para cumplir con la descripción del monstruo que dicta comportamientos opuestos según su postura.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: La IA debe cambiar entre el estado de locomoción bípeda y el estado de arrastre basado en eventos de daño o distancia. • AC2: La velocidad de movimiento debe aumentar significativamente al entrar en el estado de arrastre. 	

Nro. HU42	Título: Sistema de Ataque a Distancia (Spray)
Historia de Usuario: Como Diseñador, quiero que la entidad, al estar de pie, detecte si el jugador está en rango medio para ejecutar un ataque de proyectil de ácido.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El enemigo debe realizar un ataque de proyectil si el jugador se encuentra fuera del rango melee pero dentro del rango de visión. • AC2: El proyectil debe tener una trayectoria definida y aplicar daño al colisionar con el jugador. 	

Nro. HU43	Título: Comportamiento de Huida Táctica
Historia de Usuario: Como IA, quiero que, al ser derribado, el enemigo se desplace rápidamente fuera del campo de visión de la cámara, para implementar la mecánica de tensión psicológica descrita en la sección de IA.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Al recibir daño que cause derribo, la IA debe identificar un punto de destino fuera del campo de visión de la cámara del jugador. • AC2: El enemigo debe desplazarse hacia ese punto para salir de la escena y reiniciar su ciclo de ataque. 	

Nro. HU44	Título: Umbrales de Aturdimiento y Derribo
Historia de Usuario: Como Sistema, quiero calcular si el daño recibido supera un umbral específico para decidir si el enemigo solo se detiene, retrocede o cae al suelo, para dar feedback visual del impacto del arma utilizada.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe acumular el daño recibido en una variable de estabilidad oculta. • AC2: Superar el umbral de estabilidad debe activar la animación de caída o retroceso en el enemigo. 	

Nro. HU45	Título: Percepción Sensorial (Radial System)
Historia de Usuario: Como IA, quiero detectar al jugador mediante un sistema radial, para iniciar la persecución solo cuando James entra en el radio de amenaza definido.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: La detección del jugador debe basarse en un sensor de rayo (visión) y una esfera de proximidad (oído). • AC2: El estado de persecución solo debe activarse si alguna de las validaciones sensoriales retorna positivo. 	

Nro. HU46	Título: Lógica Variable por Dificultad
Historia de Usuario: Como Sistema, quiero cargar diferentes pistas de texto y soluciones numéricas para el mismo puzzle basándome en el "Riddle Level" seleccionado al inicio, para cumplir con la arquitectura de dificultad.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe leer la configuración de "Riddle Level" al inicializar un puzzle. • AC2: Las pistas textuales y la solución lógica deben corresponder estrictamente al nivel de dificultad seleccionado. 	

Nro. HU47	Título: Uso e Inserción de Objetos
Historia de Usuario: Como Jugador, quiero seleccionar un objeto de mi inventario y usarlo frente a un elemento del entorno, para resolver acertijos basados en la colocación correcta de piezas faltantes.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El jugador debe poder abrir el inventario y seleccionar "Usar" frente a un objeto interactivo del puzzle. • AC2: El objeto debe ser removido del inventario y aparecer visualmente en el entorno si es el correcto. 	

Nro. HU48	Título: Mecánicas de Manipulación Física
Historia de Usuario: Como Jugador, quiero interactuar físicamente con objetos, para utilizar soluciones que modifiquen el estado del mundo.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe permitir al jugador rotar o mover elementos del puzzle mediante inputs direccionales. • AC2: El estado del puzzle debe actualizarse en tiempo real reflejando la manipulación física. 	

Nro. HU49	Título: Soluciones Procedurales (RNG)
Historia de Usuario: Como Sistema, quiero generar aleatoriamente el código de la caja fuerte en cada nueva partida, para evitar que el jugador resuelva el puzzle usando guías estáticas.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe generar una combinación numérica aleatoria al iniciar una nueva partida. • AC2: Esta combinación debe inyectarse tanto en la lógica de validación de la caja fuerte como en la textura o texto de la pista. 	

Nro. HU50	Título: Visualización de Pistas y Memos
Historia de Usuario: Como Jugador, quiero leer documentos que cambien su contenido según el nivel de dificultad, para obtener la información necesaria para deducir la solución lógica sin recurrir prueba-error.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Interactuar con un documento debe desplegar su contenido en pantalla completa o ventana superpuesta. • AC2: El texto mostrado debe cargarse dinámicamente desde una base de datos localizada según el idioma y dificultad. 	

Nro. HU51	Título: Gestión de Puntos de Vida y Estado de Muerte
Historia de Usuario: Como Sistema, quiero gestionar una variable de salud que, al llegar a cero, deshabilite los controles y dispare la secuencia de "Game Over", para establecer la condición de derrota fundamental del juego.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: La variable de salud debe disminuir al recibir eventos de daño desde fuentes enemigas o ambientales. • AC2: Si la salud llega a cero, se debe deshabilitar el control del jugador y cargar la pantalla de fin de juego. 	

Nro. HU52	Título: Penalización de Movimiento por Salud Baja
Historia de Usuario: Como Sistema, quiero reducir la velocidad máxima de caminar y correr cuando la salud cae por debajo de un umbral específico, para cumplir con la regla de físicas que dicta que las heridas reducen la movilidad.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El controlador de movimiento debe verificar el porcentaje de salud actual en cada frame. • AC2: Si la salud es crítica, se debe aplicar un factor de reducción a la velocidad de movimiento y cambiar la animación a "Herido". 	

Nro. HU53	Título: Visualización Cromática de Salud
Historia de Usuario: Como Jugador, quiero ver el estado de mi salud representado por el color de la imagen en el menú de "Status", para conocer mi estado vital sin usar una barra de vida intrusiva durante el juego.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El fondo o elemento indicador en el menú de estado debe cambiar de color según el rango de salud actual. • AC2: Los colores deben transicionar de verde (salud alta) a rojo (salud crítica). 	

Nro. HU54	Título: Cálculo de Restauración por Ítem
Historia de Usuario: Como Sistema, quiero procesar la curación basada en porcentajes exactos y limitar el valor máximo al 100%, para implementar las propiedades matemáticas de los ítems de salud definidas en el diseño.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe sumar el valor de curación del ítem a la salud actual. • AC2: El valor resultante no debe exceder el máximo de salud permitido (100%). 	

Nro. HU55	Título: Escalado de Daño por Nivel de Dificultad
Historia de Usuario: Como Sistema, quiero calcular el daño infligido a los enemigos aplicando un multiplicador basado en la dificultad seleccionada, para cumplir con las tablas de balanceo especificadas en el diseño.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El daño base recibido debe multiplicarse por un factor determinado por el nivel de dificultad de acción seleccionado. • AC2: Los valores resultantes deben coincidir con los rangos definidos en las tablas de daño del GDD. 	

Nro. HU56	Título: Máquina de Estados de Reacción
Historia de Usuario: Como Sistema, quiero determinar si un ataque causa que el enemigo se detenga, retroceda o caiga al suelo, para implementar la física de combate descrita.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe seleccionar la animación de reacción (retroceso, caída, muerte) basándose en la salud restante tras el impacto. • AC2: La animación seleccionada debe interrumpir cualquier acción que el 	

enemigo estuviera realizando.

Nro. HU57	Título: Estado de Indefensión tras Impacto
Historia de Usuario: Como Sistema, quiero deshabilitar los controles de defensa del jugador por un breve periodo tras recibir daño, para simular el desequilibrio físico.	
Criterios de aceptación <ul style="list-style-type: none">• AC1: Al recibir daño, el jugador debe entrar en un estado de aturdimiento durante un tiempo definido.• AC2: Durante este estado, no se deben procesar inputs de movimiento o ataque.	

Nro. HU58	Título: Aleatoriedad en el Daño (RNG)
Historia de Usuario: Como Sistema, quiero aplicar un rango aleatorio de daño en cada golpe, para evitar que el combate sea determinista.	
Criterios de aceptación <ul style="list-style-type: none">• AC1: Cada ataque debe calcular su daño final seleccionando un valor aleatorio dentro de un rango mínimo y máximo predefinido.• AC2: El cálculo debe realizarse en el momento de la colisión.	

Nro. HU59	Título: Modificador por Tipo de Input
Historia de Usuario: Como Sistema, quiero aplicar un valor de daño mayor si el jugador ejecutó un "Heavy Attack" en comparación con un "Light Attack", para diferenciar las mecánicas ofensivas de las armas contundentes.	
Criterios de aceptación <ul style="list-style-type: none">• AC1: El sistema debe detectar si el ataque fue iniciado como ligero o pesado.• AC2: Se debe aplicar el valor de daño correspondiente a la tabla de propiedades del arma para el tipo de ataque detectado.	

Nro. HU60	Título: Control Global de Estado (Normal vs. Otherworld)
Historia de Usuario: Como Sistema, quiero gestionar una variable global que determine el estado actual del nivel, para orquestar la carga simultánea de recursos visuales, sonoros y lógicos según la definición del documento de diseño.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El juego debe mantener una variable global que indique el estado actual del mundo. • AC2: Al modificar esta variable, se deben disparar eventos para actualizar los componentes dependientes del entorno. 	

Nro. HU61	Título: Reemplazo Dinámico de Assets
Historia de Usuario: Como Diseñador, quiero que, al activarse el "Otro Mundo", las texturas del entorno cambien instantáneamente, para transformar la atmósfera del nivel según la narrativa visual descrita.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El sistema debe sustituir los materiales y texturas de la geometría del nivel por sus variantes del "Otro Mundo" al activarse la transición. • AC2: El cambio debe realizarse sin descargar la geometría base si la topología se mantiene. 	

Nro. HU62	Título: Lógica de Acceso (Door Locking/Unlocking)
Historia de Usuario: Como Sistema, quiero bloquear o desbloquear puertas específicas automáticamente, para alterar el flujo de navegación del jugador.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Las puertas deben tener estados de bloqueo independientes para el mundo normal y el alterno. • AC2: Al transicionar, el sistema debe actualizar el estado de bloqueo de todas las puertas según la configuración del nuevo mundo. 	

Nro. HU63	Título: Capas de Audio Adaptativo
Historia de Usuario: Como Sistema, quiero sustituir el bucle de audio ambiental por nuevos efectos perturbadores al ocurrir la transición, para cumplir con los requerimientos de diseño sonoro del "Other World".	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: El gestor de audio debe realizar un fundido cruzado entre la pista ambiental normal y la pista del mundo alterno al ocurrir la transición. • AC2: Se deben activar emisores de sonido específicos asociados al nuevo estado. 	

Nro. HU64	Título: Disparadores de Cambio de Mundo
Historia de Usuario: Como Diseñador, quiero definir condiciones específicas que ejecuten la transición de mundo, para sincronizar la transformación del entorno con el progreso narrativo de James.	
Criterios de aceptación <ul style="list-style-type: none"> • AC1: Se deben colocar triggers invisibles o eventos de script en puntos narrativos clave. • AC2: Al cumplirse la condición del trigger, se debe ejecutar la secuencia de transición de mundo definida. 	

Mapa de Historias y Fases

Fase 1: Prototipo Jugable (MVP) El objetivo de esta fase es tener un "Vertical Slice" funcional donde el personaje pueda interactuar con el mundo y el sistema reconozca estados de éxito (sobrevivir) o fracaso (morir).

- **EP001 - Movimiento y Fatiga:** Es la base de la interacción. Sin desplazamiento y la mecánica de cansancio, no hay exploración ni tensión en la huida.
 - **HU01:** Desplazamiento Estándar
 - **HU02:** Activación de Carrera
 - **HU03:** Degradación de Velocidad por Cansancio
 - **HU04:** Recuperación de Stamina
 - **HU05:** Movimientos Especiales
 - **HU06:** Reducción de Movilidad por Heridas
- **EP011 - Sistema de Salud y Daño:** Esencial para establecer las reglas de juego (Win/Loss condition). El jugador necesita recibir feedback visual de que está perdiendo.
 - **HU51:** Gestión de Puntos de Vida y Estado de Muerte
 - **HU52:** Penalización de Movimiento por Salud Baja
 - **HU53:** Visualización Cromática de Salud

- **HU54:** Cálculo de Restauración por Ítem
- **HU55:** Escalado de Daño por Nivel de Dificultad
- **HU56:** Máquina de Estados de Reacción (Hit Reaction)
- **HU57:** Estado de Indefensión tras Impacto
- **HU58:** Aleatoriedad en el Daño (RNG)
- **HU59:** Modificador de Daño por Tipo de Input
- **EP002 - Sistema de Combate Melee:** El mecanismo de defensa primario. Antes de implementar armas de fuego o inventarios complejos, el jugador debe poder defenderse de forma básica.
 - **HU07:** Postura de Combate
 - **HU08:** Distinción de Ataque Ligero y Pesado
 - **HU09:** Atacar con tabla (Wooden Plank)
 - **HU10:** Atacar con tubería de acero (Steel Pipe)
 - **HU11:** Restricción de Movimiento con el "Great Knife"
 - **HU12:** Golpe de gracia
- **EP009 - IA Enemigo Básico (Lying Figure):** El prototipo necesita una amenaza activa para validar que el movimiento y el combate funcionan. Sin un enemigo, no hay "Survival".
 - **HU41:** Dualidad de Estados (De Pie vs. Rastrero)
 - **HU42:** Sistema de Ataque a Distancia (Spray)
 - **HU43:** Comportamiento de Huida Táctica
 - **HU44:** Umbrales de Aturdimiento y Derribo
 - **HU45:** Percepción Sensorial (Radial System)

Fase 2: Juego Completo (Features Centrales) Estas épicas transforman el prototipo mecánico en la experiencia narrativa y compleja que es Silent Hill 2.

- **EP006 - Gestión de Inventario:** Necesario para gestionar llaves, munición y curas, permitiendo una jugabilidad más larga y estratégica.
 - **HU26:** Acceso al Inventario y Pausa
 - **HU27:** Clasificación de Recursos (Pestañas)

- **HU28:** Sistema de Acciones (Command Pane)
- **HU29:** Uso de Ítems de Curación
- **HU30:** Combinación de Objetos
- **HU31:** Inspección de Ítems
- **EP003 - Combate a Distancia:** Añade la capa de gestión de recursos (munición) y variedad al combate, fundamental para la progresión del juego.
 - **HU13:** Mecánica de Apuntado y Movilidad
 - **HU14:** Control de Cadencia de Disparo
 - **HU15:** Lógica de Recarga y Vulnerabilidad
 - **HU16:** Consumo y Capacidad de Cargadores
 - **HU17:** Matriz de Movimiento al Disparar
- **EP010 - Sistema de Puzzles:** Es el bloqueo principal para el avance de la trama. Una vez que el combate funciona (Fase 1), se añade la capa intelectual/lógica.
 - **HU46:** Lógica Variable por Dificultad
 - **HU47:** Uso e Inserción de Objetos
 - **HU48:** Mecánicas de Manipulación Física
 - **HU49:** Soluciones Procedurales (RNG)
 - **HU50:** Visualización de Pistas y Memos
- **EP004 - Mecánica Iluminación en Zonas Oscuras:** Crucial para la atmósfera de *Silent Hill*. Se implementa aquí para definir la visibilidad y tensión en áreas diseñadas específicamente oscuras.
 - **HU18:** Activación/Desactivación de Linterna
 - **HU19:** Cono de Luz Dinámico
 - **HU20:** Simulación de Adaptación a la Oscuridad
 - **HU21:** Impacto de Luz en Radio de Detección
- **EP008 - Boss (Pyramid Head):** Se añade una vez que las mecánicas de IA básica están pulidas, funcionando como un evento "hito" en la narrativa.
 - **HU36:** Sistema de Inmunidad al Daño
 - **HU37:** Física de Ataque Pesado (Great Knife)

- **HU38:** Controlador de Encuentro por Tiempo
- **HU39:** IA de Navegación en Pasillos Estrechos
- **HU40:** Ejecución de Eventos Narrativos (Maria)

Fase 3: Pulido y Contenido Adicional (Nice-to-Have) Elementos que elevan la calidad del producto final, mejoran la experiencia de usuario (UX) o añaden "espectáculo" visual, pero no bloquean la jugabilidad principal.

- **EP005 - Mecánica Advertencia Auditiva de Enemigos (Radio):** Aunque icónica, el juego es funcional sin ella. Es una mejora de *feedback* para el usuario (UX auditivo).
 - **HU22:** Modulación de Volumen por Distancia
 - **HU23:** Atracción de Enemigos por Ruido
 - **HU24:** Control de Encendido/Apagado
 - **HU25:** Restricción de Señal
- **EP007 - Interacción con Mapas:** Es una herramienta de ayuda y accesibilidad para la orientación. El juego se puede jugar (aunque sea más difícil) sin un mapa interactivo.
 - **HU32:** Validación de Posesión de Mapa
 - **HU33:** Marcador de Ubicación en Tiempo Real
 - **HU34:** Control de Interfaz de Mapa
 - **HU35:** Cambio de Contexto de Mapa
- **EP012 - Transición de Mundos:** Es un efecto visual y de diseño de nivel complejo. Mecánicamente es igual a cargar otro nivel, pero el efecto visual de transformación es "pulido" de alta gama.
 - **HU60:** Control Global de Estado (Normal vs. Otherworld)
 - **HU61:** Reemplazo Dinámico de Assets
 - **HU62:** Lógica de Acceso (Door Locking/Unlocking)
 - **HU63:** Capas de Audio Adaptativo
 - **HU64:** Disparadores de Cambio de Mundo