

# Proyecto Integrado

Desarrollo de Aplicaciones Multiplataforma

- José Jalado Viedma -

## 1. Índice.

(Implementado en la finalización del proyecto)

## 2. Análisis del problema.

### 2.1. Descripción del proyecto.

- Nombre del producto: The Past
- Ambientación:
  - **Mundo:**  
Mundo basado en el ideal de fantasía que corresponde con castillos, caballeros, criaturas, etc. El avance entre niveles torna el ambiente más oscuro y disonante con el canon de fantasía clásica.
  - **Tiempo:**  
Desconocido. Es importante la ausencia de un momento temporal al que poder asociar el mundo presentado ya que se intenta transmitir un mensaje atemporal sobre circunstancias que se han dado y darán siempre.

### 2.2. Objetivos.

Mi objetivo al desarrollar un videojuego como proyecto es transmitir una serie de sentimientos específicos al público, utilizando la interacción, la narrativa y el ambiente visual y sonoro como medios de expresión. El fin último de este proyecto es que, al terminar, el juego deje una huella emocional en ellas.

## 3. Apartado gráfico.

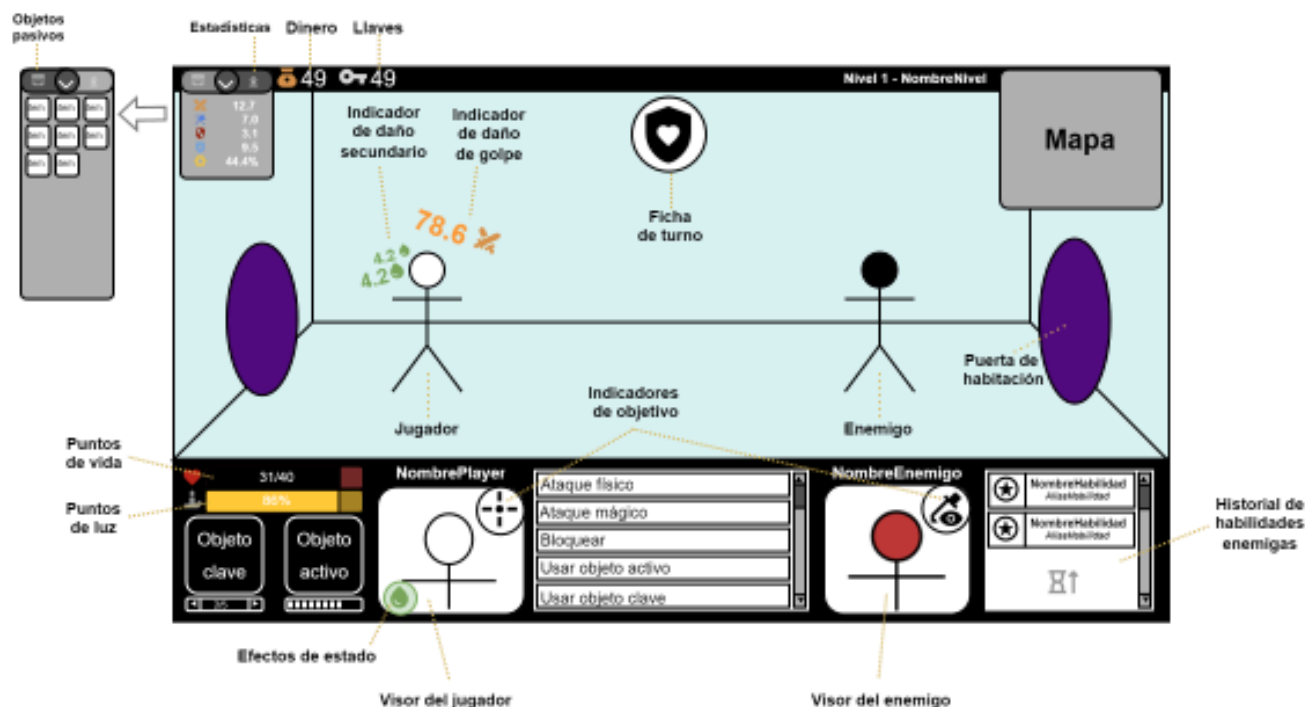
### 3.1. Diseño previo (mockups).

La interfaz está basada en tres estados principales:

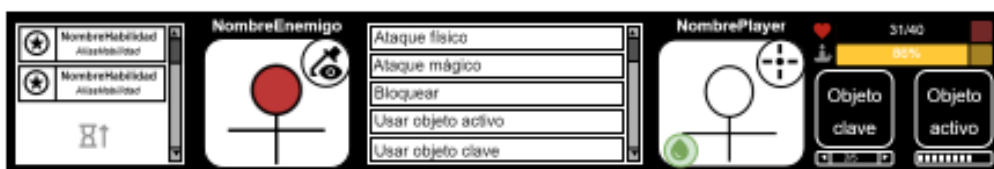
- **Menús out of game:**  
Interfaz con botones para jugar, ajustes, estadísticas de partida, salir...

- Interfaz in game (combate y viaje):

# Vista general de la interfaz



## Interfaz de combate:



## Interfaz de viaje:



## Características principales

### Parte central:

Se caracteriza por mostrar los principales elementos del combate, entre ellos, ambos personajes presentes en la escena. Conforme el combate suceda, la ficha de turno irá indicando cuando le toca atacar o defender al jugador. Además, muestra las cifras de daño producidas tanto al jugador como al enemigo acompañadas de un icono y color identificativo para cada tipo de daño.

### Barra superior:

Muestra la cantidad de bienes que posee el jugador y tiene anclados el visor de estadísticas y objetos; y el mapa.

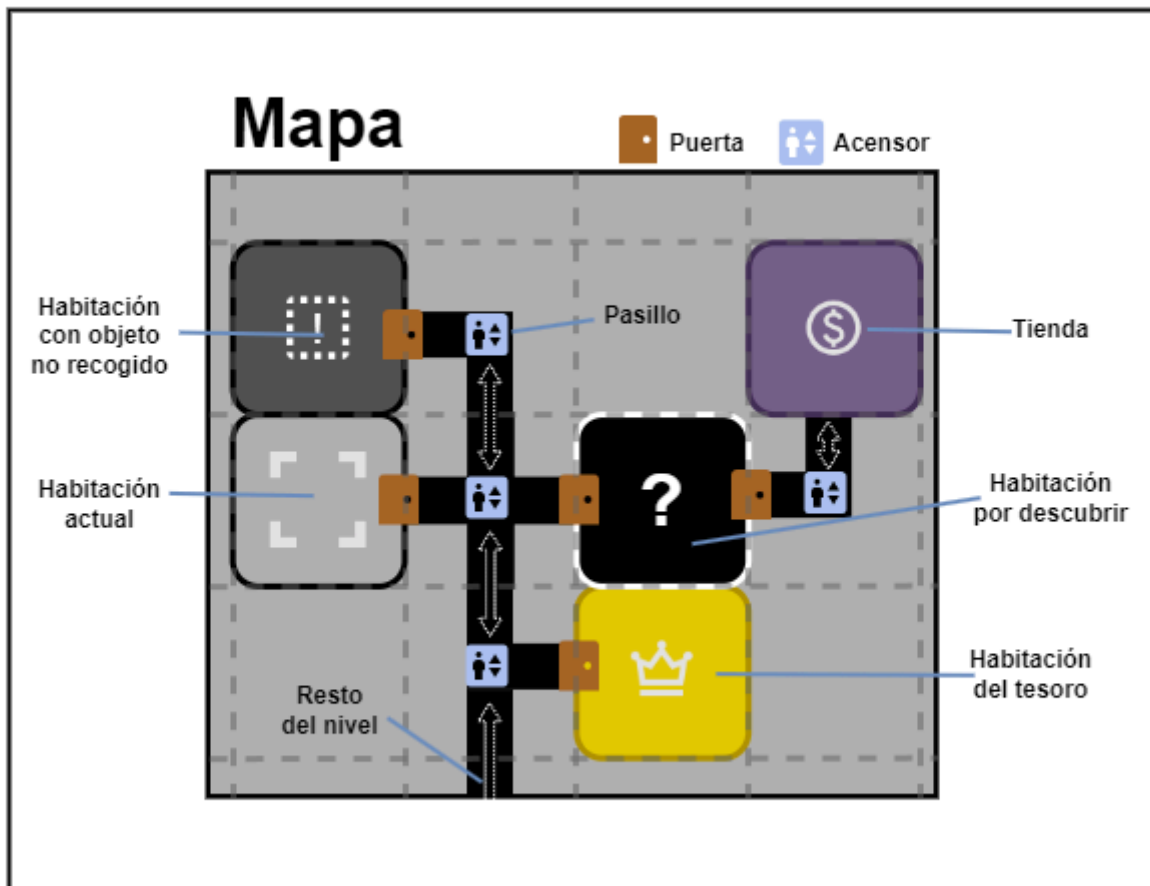
**Barra inferior:**

- Visores de personajes (jugador y enemigo).
- Efectos de estado aplicados a los personajes (veneno, sangrado, etc).
- Objetos activables (activo y clave).
- Estadísticas de estado del jugador (vida y luz).
- Marcadores de objetivo en combate.
- Menú de acciones global.

**Interfaz en estado de viaje:**

La barra inferior toma el estado de viaje cuando el jugador está fuera de combate, de esta manera es posible la utilización de objetos activables, interacción mediante el menú de acciones global para multitud de acciones:

- Abrir cofres.
- Hablar con npc.
- Accionar mecanismos.
- Comprar objetos.
- ...

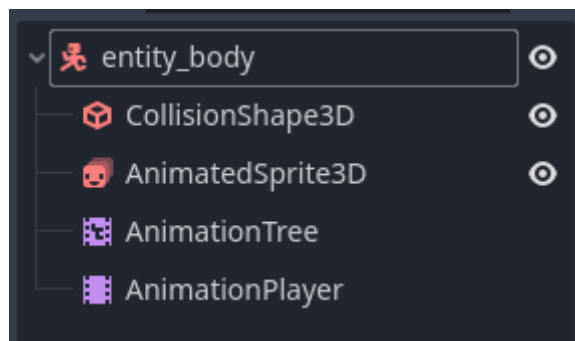
**- Minimapa:**

El mapa muestra en todo momento el estado de las habitaciones, puertas, ascensores conocidos y la ubicación del jugador en todo momento.

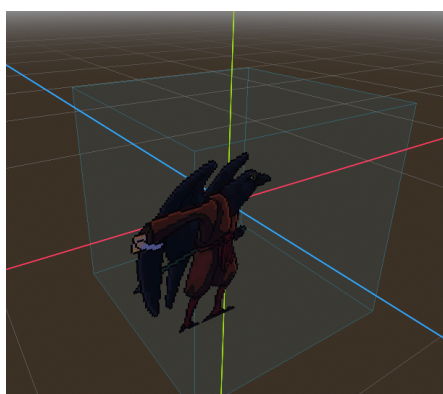
### 3.2. Entidades.

Cada entidad está compuesta por un objeto propio llamado Entity del tipo correspondiente (EntityPlayer, EntityEnemy...). Este objeto guarda información sobre la entidad, su comportamiento y el Body. Este body es la parte física de la entidad en el juego, este almacena y controla la colisión y animaciones. Hay algunos “body” que se comparten entre entidades.

#### Estructura del Body



#### CollisionShape3D



Proporciona una forma simple para controlar las colisiones de la entidad.

#### AnimatedSprite3D



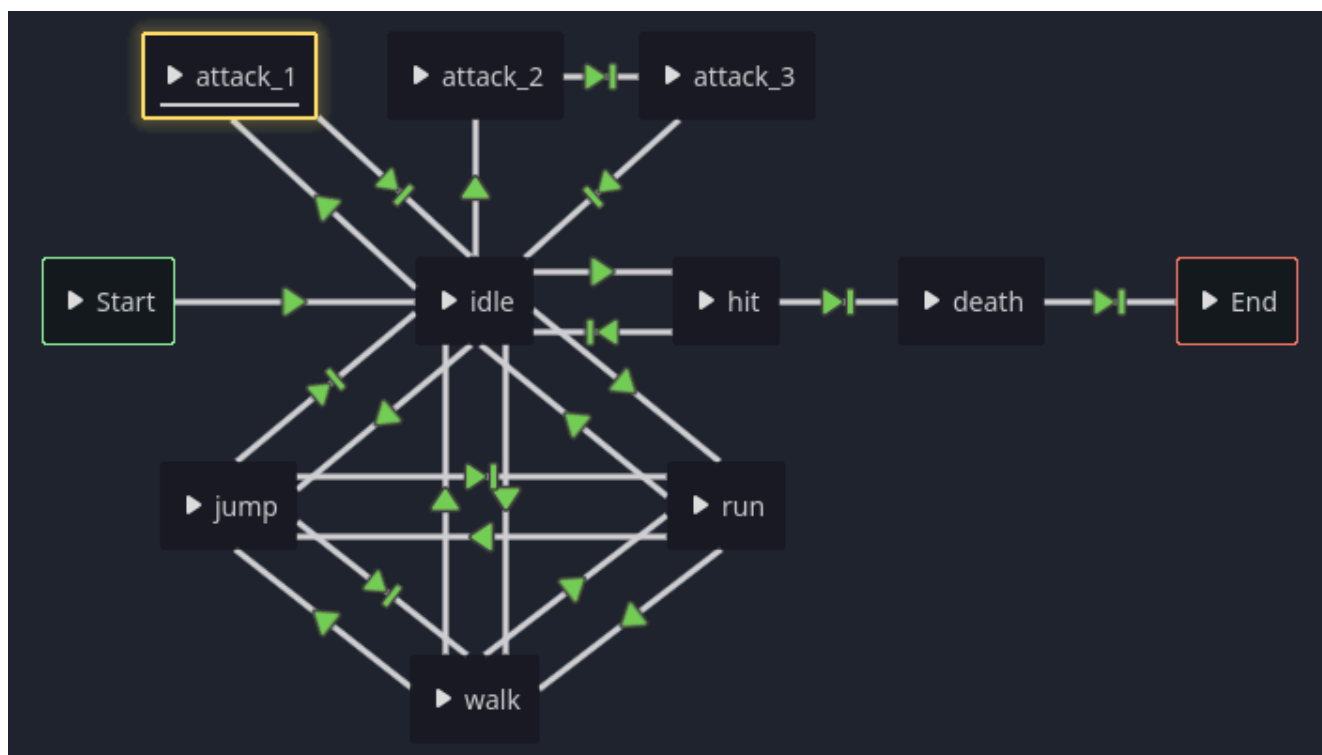
Agrupar conjuntos de regiones del spritesheet para generar animaciones básicas y primitivas. Se importan las animaciones en AnimationPlayer gracias a un plugin de la comunidad de Godot llamado *AnimatedSprite to AnimationPlayer Converter*.

#### AnimationPlayer



Utiliza animaciones importadas del AnimatedSprite3D y se permiten establecer multitud de configuraciones en la animación, tales como cambiar archivos de recursos, reproducir audio, progresión lineal entre frames...

### AnimationTree



Este componente es el que orchestra los cambios entre animaciones del AnimationPlayer mediante condiciones y transiciones estableciendo qué animaciones han de repetirse, detenerse, sincronizarse o esperar a que termine otra animación.

### 3.3. Habitaciones.

### 3.4. Interfaz gráfica.

#### 4. Elementos y dinámicas principales.

##### 4.1. Personaje principal.

###### - Estadísticas del personaje:

Estadística	Descripción
Vida	Puntos de salud del jugador. Se reduce al recibir impactos o efectos de daño. Al llegar a 0 el jugador muere.
Luz (%)	Al llegar a 0 habilita la aparición de “amigo de las sombras” (***).
Daño físico	Aumenta el daño de ataques y efectos físicos.
Daño mágico	Aumenta el daño de ataques y efectos mágicos.
Daño verdadero	Aumenta el daño de ataques y efectos de daño verdadero. No puede ser reducido con resistencias.
Armadura física	Reduce parte del daño físico recibido, puede ser ignorada con penetración física.
Armadura mágica	Reduce parte del daño mágico recibido, puede ser ignorada con penetración mágica.
Probabilidad de crítico (%)	Probabilidad de cada golpe o efecto de asestar un impacto crítico (+ 75% daño).
Velocidad de ataque	Número de fichas de acción por cada turno de ataque.

###### - Habilidades del personaje: (No implementado)

##### 4.2. Enemigos.

###### - Características:

Los enemigos están determinados por su apariencia, estadísticas, habilidades y efectos adicionales.

###### - Estadísticas enemigas:

Los enemigos comunes y jefes tienen una cantidad fija de estadísticas y un incremento que viene determinado por el nivel en el que se encuentra el jugador. Las estadísticas no son visibles por el jugador por lo que ha de deducir y recordar en qué estadística destaca cada enemigo.

##### 4.3. Objetos e inventario.

###### - Inventario:

Es el espacio donde se almacenan los objetos que recoge el jugador, el inventario tiene espacio ilimitado para objetos excepto para los objetos activos que sólo se puede portar uno (excluyendo efectos pasivos que permitan llevar más de un objeto activo).

- **Calidad de objeto:**

Cada objeto tiene asignado un valor (tier), este va asociado con la calidad del objeto y el ratio de aparición del mismo (cuan más alto el tier, mayor calidad y menor ratio de aparición). El tier más bajo es el 0 y el más alto es 4(\*\*\*)

- **Variación de estadísticas:**

Los objetos activos y pasivos comúnmente proporcionan una serie de modificaciones en la estadísticas que se aplican en el momento de recoger el objeto. En el caso de que se intercambie el objeto activo por otro se conservan las estadísticas (no vuelve a modificar estadísticas si se vuelve a recoger uno que ya modificó estadísticas).

- **Objetos pasivos:**

Son objetos que actúan sin necesidad de accionarlos manual e intencionadamente. Estos proporcionan multitud de estadísticas, habilidades y *efectos de impacto* diferentes.

- **Objetos activos:**

Son objetos que actúan al realizar la acción de “Usar objeto activo” en el menú de acción. Existen algunos objetos activos que tienen la capacidad de ser utilizados fuera de combate. La mayoría de los objetos activos necesitan el máximo de cargas para poder ser usado, se irán cargando mientras el jugador limpia habitaciones de enemigos. Una vez usado el objeto, este se queda sin cargas. Si el objeto activo no tiene indicador de cargas, significa que se trata de un objeto de un sólo uso y una vez se utilice, desaparecerá.

- **Objetos clave:**

Son objetos que actúan al realizar la acción de “Usar objeto clave” en el menú de acción. Existen algunos objetos clave que tienen la capacidad de ser utilizados fuera de combate. Excepto efectos que especifiquen lo contrario, los objetos clave se consumen al utilizarse.

- **Lista de objetos:**

ID	NOMBRE	ICONO	DESCRIPCIÓN
000	Inicio primigéneo		- Los inicios nunca fueron limpios -  <b>Efecto pasivo:</b> Este objeto hace cosas cuando <b>enterRoom</b> y los enemigos hacen volteretas y aplica <b>25 de Veneno (3 hits)</b> por las siguientes 3 rondas.
001	Objeto bbbb		- Doble recompensa -  <b>Estadísticas:</b> + 200 Vida + 1 Velocidad de ataque  <b>Efecto pasivo (pre-golpe):</b> Este objeto cuando el jugador <b>golpear</b> reduce <b>33 de Armadura Física</b> al enemigo.

Enlace: (\*\*\*)

- **Efectos de objetos.**

Los objetos recogidos por el personaje brindan diferentes efectos que actúan en momentos determinados tales como entrar en la habitación, al golpear, al asestar un impacto crítico, al bloquear, al recoger un objeto, al comprar en la tienda, al abrir un cofre...

#### 4.4. Combate.

- **Características generales del combate:**

Tras entrar en una habitación con un enemigo, se bloquea la capacidad de movimiento del jugador y comienza el **estado de combate**.

Quién inicia el combate y qué orden de ataque se sigue vienen determinados por la estadística de **velocidad de ataque** de los integrantes, el que mayor velocidad tenga, comienza con el turno de ataque. Si no existe diferencia comienza el jugador. En cada turno se pueden realizar tantas acciones como puntos de **velocidad de ataque** se poseen.

- **Combate del Jugador:**

El jugador tiene permitido realizar tantas acciones como puntos de velocidad de ataque tenga y se habilitan una serie de opciones en el menú de acción:

**Acciones del jugador en combate:**

- **Atacar:**

Permite seleccionar una de las habilidades disponibles del jugador. Al lanzar una habilidad se disparan sus efectos aplicando todos los efectos pasivos y/o activos en función del momento de aplicación (antes de golpear, al golpear, tras el golpe). No termina el turno de ataque (a menos que un efecto o habilidad lo indique).

- **Bloqueo:**

El bloqueo normal fuerza al enemigo a atacar y proporciona un 25% (\*\*\*) de reducción de daño en los siguientes golpe recibidos (existen modificaciones de esta habilidad). Termina el turno de ataque.

- **Utilizar objeto:**

El jugador puede hacer uso de su objeto activo y de sus objetos clave. No termina turno de ataque (a menos que un efecto o habilidad lo indique). En algunos casos, puede desatar un ataque rival (termina turno de ataque) en función del objeto utilizado y el enemigo presente.

- **No hacer nada:**

Opción anti intuitiva pero útil si se poseen las herramientas necesarias. Como su nombre indica, el jugador no hace nada y termina el turno de ataque. Existen efectos de objetos que aprovechan esta arriesgada acción para brindar efectos especialmente fuertes.



- **Opciones especiales:**

En determinadas situaciones aparecerán opciones en el menú diseñadas especialmente para el enemigo o entorno en el que el jugador se encuentra.

- **Combate del Enemigo:**

Cuando el turno de ataque del jugador termina pasa al enemigo y el jugador entra en el turno de defensa.

El enemigo tiene la posibilidad de seleccionar diferentes acciones:

**Acciones del enemigo en combate:**

- **Atacar:**

El enemigo lanza una habilidad del repertorio que posee y esta se muestra en el historial de habilidades enemigas de la interfaz de combate. No termina el turno de ataque.

- **Bloqueo:**

El bloqueo normal fuerza al jugador a atacar y proporciona un 25% (\*\*\*) de reducción de daño en los siguientes golpe recibidos (existen modificaciones de esta habilidad). Termina el turno de ataque.

- **Opciones especiales:**

En situaciones especiales los enemigos pueden realizar ciertas acciones poco comunes.

#### 4.5. Personajes no jugables (NPC) y misiones secundarias.

##### 4.5.1. Personajes no jugables (NPC).

Nombre	Aparición	Descripción
<b>El Tendero</b>	<b>Lugar:</b> <a href="#">Tienda</a>	~ <i>Gatito no te metas con el jaguar</i> ~  Va acompañado de un cánido, es inocente e influenciable. Su mirada revela su pasión por el dinero.
	<b>Probabilidad:</b> 75 %	Pedirá una cuantía de dinero o llaves para hablar con él. Tras hacerlo, su compañera canina te dará una recompensa aleatoria en función del valor pedido.
<b>El Minero</b>	<b>Lugar:</b> <a href="#">Habitación inicial</a>	~ <i>Vida antes que muerte. Fuerza antes que debilidad. Viaje antes que destino</i> ~  Es trabajador, persistente y resiliente. Posee una mirada noble.
	<b>Probabilidad:</b> 25 %	Tras hablar con él, el Minero se irá a rebuscar elementos de valor que te entregará al final del nivel. El jugador tiene la posibilidad de perder una mascota. (***)

<b>El Herald</b>	<b>Lugar:</b> <a href="#">Habitación del tesoro.</a>	~ ¡Catapún! ~  Siempre va arreglado y nunca le faltan sus poco fiables consumibles.
	<b>Probabilidad:</b> No siempre	Bloquea la recompensa de la sala y pedirá un porcentaje de la salud máxima del jugador para poder abrirlo. Es posible “probar suerte” para evitarlo, si sale bien, el tesoro es accesible, si sale mal, el jugador pierde la vida máxima y la recompensa desaparece.
<b>El Herrero</b>	<b>Lugar:</b> <a href="#">Pasillo</a> aleatorio.	~ ¿Aguja e hilo? Sólo conozco mi martillo y mi yunque. ~
	<b>Probabilidad:</b> 14 % (Máximo una vez por nivel)	Interactuar con él permitirá al jugador intercambiar un objeto aleatorio de la calidad más baja que tenga por otro de una calidad mayor.
<b>Grem corazón de lobo</b>	<b>APARICIONES ESPONTÁNEAS:</b>	~ La bella y cruda naturaleza en sí misma. ~
	<b>Lugar:</b> <a href="#">Habitación común</a> aleatoria.	Es una loba, verla nunca es buena señal... Pero, ¿para quién?
	<b>Probabilidad:</b> Máximo una vez por nivel.	<b>Apariciones espontáneas:</b> Durante el combate ejecuta al enemigo si tiene menos de 25% de vida, aumentando las presas cobradas en uno.
	<b>APARICIÓN FINAL:</b>	<b>Aparición final:</b> Grem crea una habitación efímera antes de acceder a una <a href="#">habitación de jefe</a> . En el Limbo interactuar con Grem le dará al jugador un número de cofres con recompensas en función de la cantidad de presas cobradas.
	<b>Lugar:</b> <a href="#">El Limbo</a>	Interactuar con ella tiene un 1 % de probabilidades de matar al jugador.
	<b>Probabilidad:</b> Máximo una vez por partida.	
<b>La Gata</b>	<a href="#">Habitación común</a> despejada. Máximo 3 veces por nivel.	~ Vale, me levanto ~  Algunos la consideran adorable.
		Al aparecer, la rata tiene una probabilidad de 1 % de duplicar la recompensa de la sala recién limpiada y otro 1 % de eliminar la recompensa. Al hablar con ella, repetirá las palabras del jugador (**).
<b>Eras el ermitaño</b>	<b>Lugar:</b> <a href="#">Las Escaleras.</a> <b>Frecuencia:</b> Siempre que aparezca la habitación.	~ La vida es de lo bueno lo mejor de lo mejor lo superior. ~  No le gustan los pájaros. Es uno de los hermanos de <a href="#">Las Escaleras</a> .
<b>Smol el viajero</b>	<b>Habitación especial:</b> <a href="#">Las Escaleras.</a> Siempre que aparezca la habitación.	~ Nadie es merecedor. ~  Despacito pero va. Es uno de los hermanos de <a href="#">Las Escaleras</a> .
<b>Equipo de Apricot</b>	<b>Habitación especial:</b> <a href="#">El Campus.</a>	~ La madre de todos los amigos. ~  Al hablar con Apricot se ofrecerán varias opciones al jugador en función de su cometido en el lugar pudiendo elegir tan solo una: - <b>Calmar la mente:</b> Restaura un 50 % de puntos de luz del jugador.

		<ul style="list-style-type: none"> <li>- <b>Cuidar:</b> El sol brilla. Cura toda la vida.</li> <li>- <b>Creecer:</b> El jugador tiene un 50 % de probabilidades de obtener un objeto aleatorio</li> <li>- <b>Pulir:</b> El jugador obtiene una cantidad aleatoria de dinero y una llave.</li> <li>- <b>Curtir:</b> El jugador pierde un 50 % de vida actual a cambio de aumentar en 25% su vida máxima.</li> </ul>
--	--	--

#### 4.6. Mundo:

##### 4.6.1. Niveles.

Las habitaciones y pasillos se agrupan en niveles. Cada nivel comienza en la habitación inicial y termina comúnmente en la habitación del jefe. Hay 4 (\*\*\*) niveles en total, existiendo dos alternativas para el último nivel.

##### 4.6.2. Pasillo.

Sirve como enlace entre habitaciones principales, permitiendo la navegación horizontal o vertical. Puede formar cadenas de conexión entre salas comunes, especiales y jefe. Estructura base que da forma al nivel.

##### 4.6.3. Habitación inicial.

Es el punto de partida del jugador en cada nivel. Siempre se ubica en la celda (0,0). Es única por nivel.

##### 4.6.4. Habitación común.

Esta habitación tiene dos estados posibles:

- Por descubrir:  
La habitación posee uno o varios enemigos que han de ser derrotados para cambiar al estado “despejada”. El movimiento del jugador se bloquea y comienza el estado de [combate](#).
- Despejada:  
Tras el cambio de estado, la habitación revela la recompensa por el combate y los npc que decidan aparecer en dicha habitación (cómo [la Rata](#)).

##### 4.6.5. Habitación del tesoro.

Esta habitación posee una o varias recompensas (excepto en casos especiales). Aparece una en cada nivel.  
No conecta con más habitaciones.

##### 4.6.6. Tienda.

###### - Descripción:

Esta habitación puede contener entre 0 y 3 mostradores con objetos pasivos, activos y clave (en función del nivel actual). Dichos objetos pueden ser obtenidos a cambio de una determinada cantidad de oro.

Aparece una en cada nivel.  
No conecta con más habitaciones.  
Es necesaria una llave para acceder.

#### 4.6.7. Habitación del jefe.

#### 4.6.8. El Limbo.

- **Descripción:**

A esta habitación se accede de forma involuntaria cuando Grem la loba hace su aparición final. Es una instancia entre el nivel y la habitación del jefe.

#### 4.6.9. Las Escaleras.

- **Descripción:**

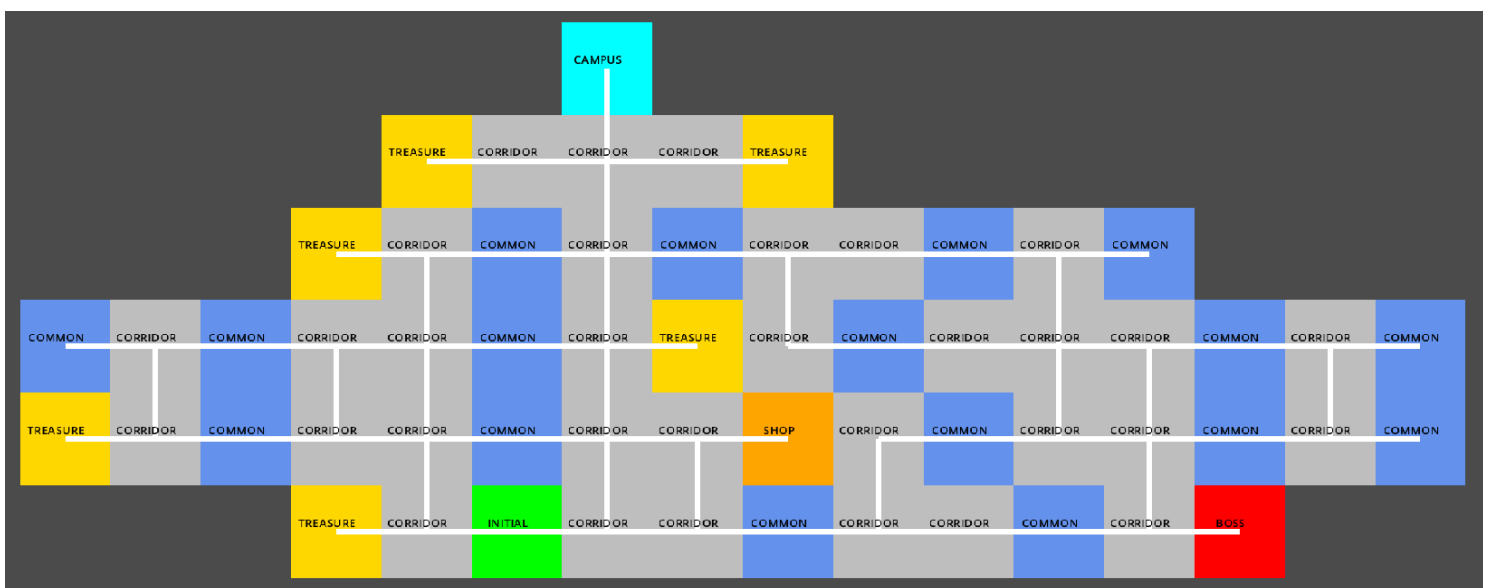
A esta habitación se puede acceder al viajar entre niveles (después de un jefe).

#### 4.6.10. El Campus.

- **Descripción:**

A esta habitación se accede siempre desde abajo (con un ascensor de subida).

### 5. Mecanismos y dinámicas secundarias.



Generación del primer nivel del juego con cuatro habitaciones del tesoro, una tienda, un campus y un jefe.

#### 5.1. Flujo de generación.

The Past cuenta con un sistema de generación procedural de mazmorras implementado mediante la clase **WorldGenerator**, responsable de construir la disposición lógica de habitaciones, pasillos y elementos especiales para cada nivel. Este sistema garantiza variabilidad entre partidas manteniendo una estructura coherente, utilizando una semilla numérica (seed) que asegura la repetibilidad del contenido generado.

El nivel se modela sobre una cuadrícula lógica representada por un **Dictionary<Vector2i, RoomData>**, donde cada celda contiene la información sobre la sala correspondiente: tipo, conexiones y distancia desde el origen. Este modelo es liviano y desacoplado de la instancia real en la escena, permitiendo un proceso eficiente que no depende de nodos físicos hasta que se requiere.

## 5.2. Inicialización del nivel.

El proceso comienza llamando a `generate_level`, que recibe como parámetros la semilla, el nivel actual, y múltiples valores configurables (como población máxima, cantidad de salas especiales, etc.). Este método se encarga de coordinar el resto de procesos internos en el siguiente orden:

- **\_generate\_path**: construye la rama principal del nivel.
- **\_ensure\_minimum\_common\_rooms**: garantiza una cantidad mínima de habitaciones comunes.
- **\_ensure\_instances\_range**: valida que el nivel no exceda el tamaño permitido.
- **\_place\_boss\_room**: intenta colocar la sala del jefe en el punto más alejado.
- **\_place\_special\_rooms**: añade salas especiales como tienda, tesoro o campus.
- **\_connect\_corridors\_to\_adjacent\_rooms**: reconecta verticalmente pasillos adyacentes.
- **\_clean\_isolated\_corridors**: elimina pasillos aislados sin utilidad.
- **\_expand\_dead\_end\_corridors\_with\_common\_rooms**: intenta extender pasillos muertos agregando habitaciones comunes.

## 5.3. Procesos de generación.

### 5.3.1. Rama principal (**\_generate\_main\_branch**).

Desde la celda (0,0) se construye una rama lineal hacia adelante, alternando pasillos y habitaciones comunes, con bifurcaciones verticales limitadas. Se intenta alcanzar una distancia lógica máxima, ajustada según el nivel actual. Si el generador se bloquea sin rutas disponibles, activa un sistema de backtracking que permite retroceder a posiciones anteriores para buscar caminos alternativos. La variable `last_direction` se usa para evitar repetir direcciones. Durante esta fase también se establece la primera habitación (INITIAL) mediante **\_create\_initial\_room**, que inicia la construcción y se añade a `map_data`.

### 5.3.2. Expansión mediante ramas laterales (**\_populate\_branches**).

A partir de los pasillos creados en la rama principal, el sistema intenta generar nuevas habitaciones en direcciones adyacentes válidas, respetando la regla de colocación sobre que las habitaciones comunes sólo pueden expandirse horizontalmente. Cada rama lateral se limita por un número máximo configurable de instancias.

### 5.3.3. Ramas secundarias completas (**\_generate\_secondary\_branches**).

Para alcanzar la población deseada de salas, se generan ramas completas partiendo de celdas ya existentes. Estas ramas combinan pasillos y habitaciones comunes, adaptando la dirección según las restricciones del sistema (como la necesidad de colocar habitaciones comunes solo en horizontal).

### 5.3.4. Colocación de salas especiales (**\_place\_special\_rooms**).

El sistema busca habitaciones comunes que cumplan criterios específicos (una sola entrada horizontal) y las reemplaza por:

- **Tienda (**\_place\_shop**)**: solo se ubica una por nivel.
- **Tesoro (**\_place\_treasure\_rooms**)**: se colocan varias según la configuración.
- **Campus (**\_place\_campus**)**: aparece con una probabilidad configurada, y siempre sobre un pasillo vertical.

Cada una de estas reemplazará la celda manteniendo las conexiones originales.

### 5.3.5. Colocación de la sala del jefe (**\_place\_boss\_room**).

Se intenta ubicar la sala del jefe en la habitación común más alejada del inicio que cumpla con las restricciones de entrada. Se permite un número limitado de intentos; si todos fallan, se reinicia la generación del nivel desde cero.

### 5.3.6. Limpieza y correcciones.

Una vez generado el mapa, se aplican procesos de corrección:

- **Conexión vertical de pasillos (**\_connect\_corridors\_to\_adjacent\_rooms**)**: permite que los pasillos se conecten entre sí en vertical, si están alineados y no se han conectado aún.
- **Eliminación de pasillos aislados (**\_clean\_isolated\_corridors**)**: cualquier pasillo conectado con una sola celda se elimina, limpiando también las conexiones de celdas adyacentes.
- **Expansión de pasillos muertos (**\_expand\_dead\_end\_corridors\_with\_common\_rooms**)**: si un pasillo no conduce a ninguna otra sala, se intenta extender horizontalmente agregando una habitación común al final.

### 5.3.7. Garantías de diseño.

Para evitar inconsistencias o diseños no deseados, el sistema incorpora controles adicionales:

- **\_ensure\_minimum\_common\_rooms**: asegura una cantidad mínima de habitaciones comunes, intentando generarlas si no se alcanzó el número mínimo.
- **\_ensure\_instances\_range**: fuerza la regeneración completa del nivel si el número de habitaciones supera el máximo permitido.

Todos estos mecanismos se adaptan dinámicamente al nivel actual, aumentando la dificultad y complejidad de forma progresiva.

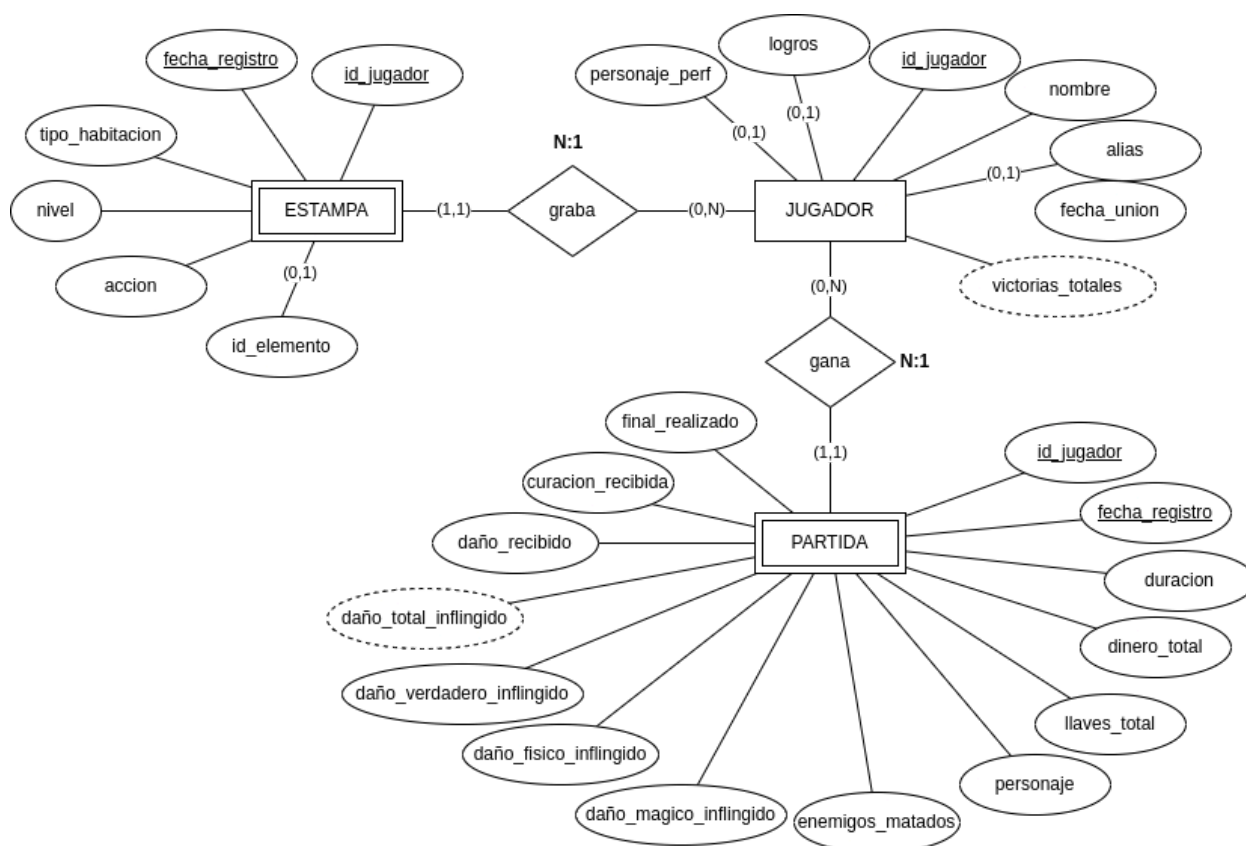
### 5.3.8. Carga dinámica de instancias.

Una vez finalizado el modelo lógico, el sistema está listo para instanciar nodos reales mediante un instanciador, encargado de convertir cada **RoomData** en su representación visual cuando el jugador accede a dicha celda. Esta separación permite mantener un rendimiento alto, ya que solo se cargan las habitaciones necesarias en cada momento.

## 6. Acceso y gestión de datos remotos.

### 6.1. Base de datos (MariaDB).

#### - Entidad-Relación:



#### - Tablas de las entidades:

JUGADOR			
Atributo	Descripción	Detalle	Dato
<u>id_jugador</u>	Identificador del jugador.	Clave primaria y Autoincrement	INT
nombre	Nombre en línea del jugador.		VARCHAR(100)
alias	Pequeña expresión asociada a un jugador.		VARCHAR(100)
fecha_union	Fecha del registro del jugador en la BD.		DATE

victorias_totales	Número de registros de partidas ganadas por el jugador.		BIGINT
logros	Cadena de texto JSON con los datos de estado de los logros	Opcional	LONGTEXT
personaje_pref	Identificador del personaje principal con más victorias usado por el jugador.	Opcional	INT

PARTIDA			
Atributo	Descripción	Detalle	Dato
<u>id_jugador</u>	Identificador del jugador.	Clave foránea de JUGADOR y clave primaria compuesta.	INT
<u>fecha_registro</u>	Fecha y hora del registro de la victoria en la BD.	Clave primaria compuesta.	DATETIME
duracion	Tiempo en segundos que tomó la partida.		BIGINT
dinero_total	Dinero total recogido a lo largo de la partida.		INT
llaves_total	Llaves totales recogidas a lo largo de la partida.		INT
personaje	Identificador del personaje principal con quien se ganó la partida.		INT
enemigos_matados	Cadena de texto JSON con los datos de los enemigos matados durante la partida.		LONGTEXT
daño_magico_inflingido	Daño mágico total infligido.		DECIMAL(15,3)
daño_fisico_inflingido	Daño físico total infligido.		DECIMAL(15,3)
daño_verdadero_inflingido	Daño verdadero total infligido.		DECIMAL(15,3)
daño_total_inflingido	Daño total infligido.	Campo calculado.	DECIMAL(15,3)
daño_recibido	Daño total recibido por el jugador.		DECIMAL(15,3)
curacion_recibida	Puntos de vida recuperados en total por el jugador.		DECIMAL(15,3)
final_realizado	Identificador de final realizado para la victoria.		INT



ESTAMPA			
Atributo	Descripción	Detalle	Dato
<u>id_jugador</u>	Identificador del jugador.	Clave foránea de JUGADOR y clave primaria compuesta.	INT
<u>fecha_registro</u>	Fecha y hora del registro de la estampa en la BD.		DATETIME
tipo_habitación	Identificador del tipo de habitación donde se grabó la estampa.		INT
nivel	Identificador del nivel donde se grabó la estampa.		VARCHAR(100)
accion	Identificador de la acción que se grabó en la estampa.		VARCHAR(50)
id_elemento	Identificador del elemento relacionado con la estampa (enemigo, npc u objeto)		INT

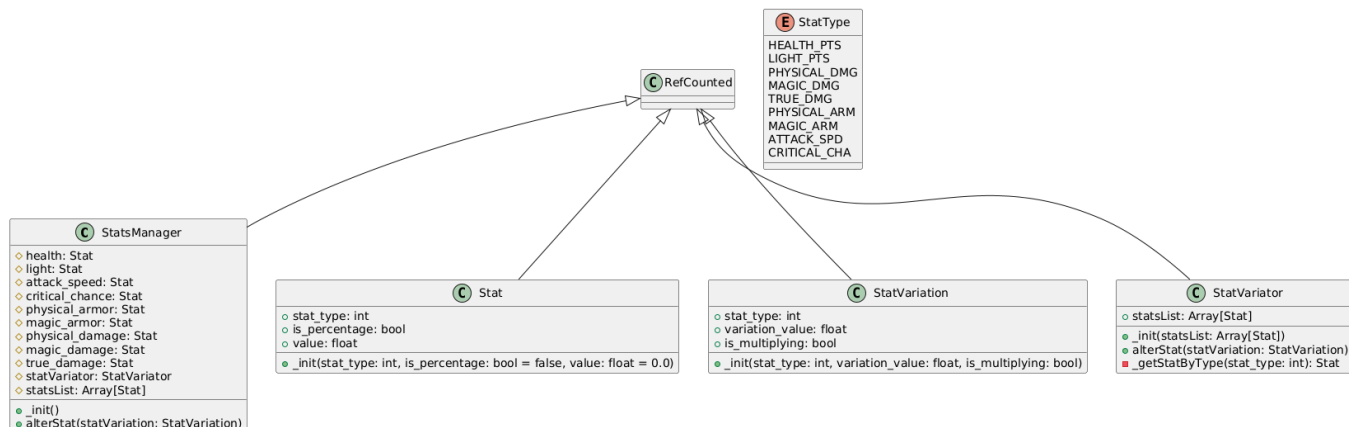
## 6.2. Despliegue de BD.

(No implementado)

## 7. Estructura del proyecto.

### 7.1. Modelo de datos.

#### Estadísticas y variación por objetos.



Una entidad como un jugador o un enemigo posee un administrador (**StatManager**) con una serie de estadísticas (**Stats**) que varían en función de la variación de estadísticas (**StatVariation**) que brindan los objetos al recogerlos mediante un intermediario (**StatVariator**).

### 7.2. Generación de mundo.

#### Generación procedural de la estructura lógica de niveles.



Un nivel se genera mediante un generador (**WorldGenerator**) que construye la estructura lógica de habitaciones (**RoomData**) en una cuadrícula. Esta estructura incluye una rama principal, ramas secundarias y salas especiales, respetando reglas de conexión y tamaño. La generación se basa en una semilla y se controla mediante constantes globales. El generador no instancia salas directamente, sino que prepara los datos para que un sistema externo los cargue dinámicamente según se necesiten.

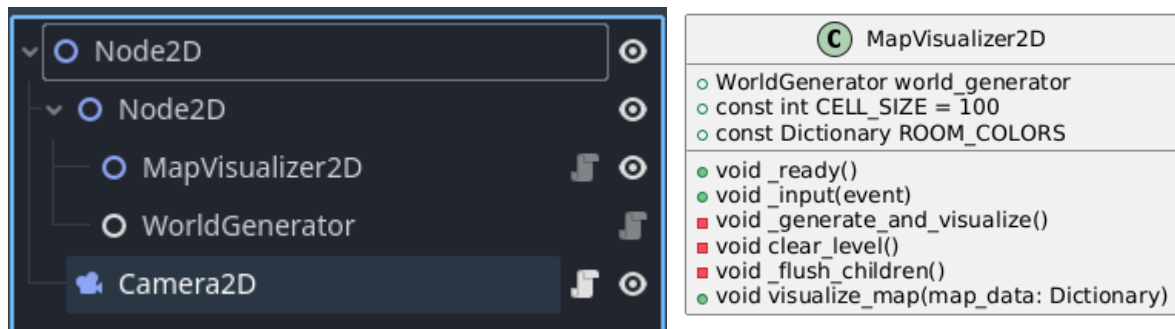
### **Parámetros constantes de la generación procedural.**

**GlobalConstants** actúa como un contenedor centralizado de constantes de configuración que controlan el comportamiento de la generación procedural de niveles. Define rutas de datos, parámetros de diseño, reglas de generación y opciones de depuración, permitiendo ajustar fácilmente el equilibrio y la complejidad del juego.

## 8. Testing.

### 8.1. Pruebas de generación procedural.

#### Visualización de la estructura lógica de niveles.



**MapVisualizer2D** es una herramienta de depuración visual que representa gráficamente el contenido generado por **WorldGenerator**. Dibuja habitaciones y conexiones sobre una cuadrícula utilizando **ColorRect** y **Label**, facilitando la inspección del mapa lógico generado. Permite regenerar y visualizar nuevos mapas en tiempo real al pulsar una tecla ("ENTER"). Además se añade una **Camera2D** personalizada para explorar el esquema generado.

## 9. Incisos:

- Los elementos terminados en (\*\*\*) se encuentran en valoración para ser implementados o desechados.

## 10. Referencias.

(No implementado)