



ESTUDIANTES

**Juan Camilo Mendieta Hernández**  
**Camilo Hernández Guerrero**

PROFESOR

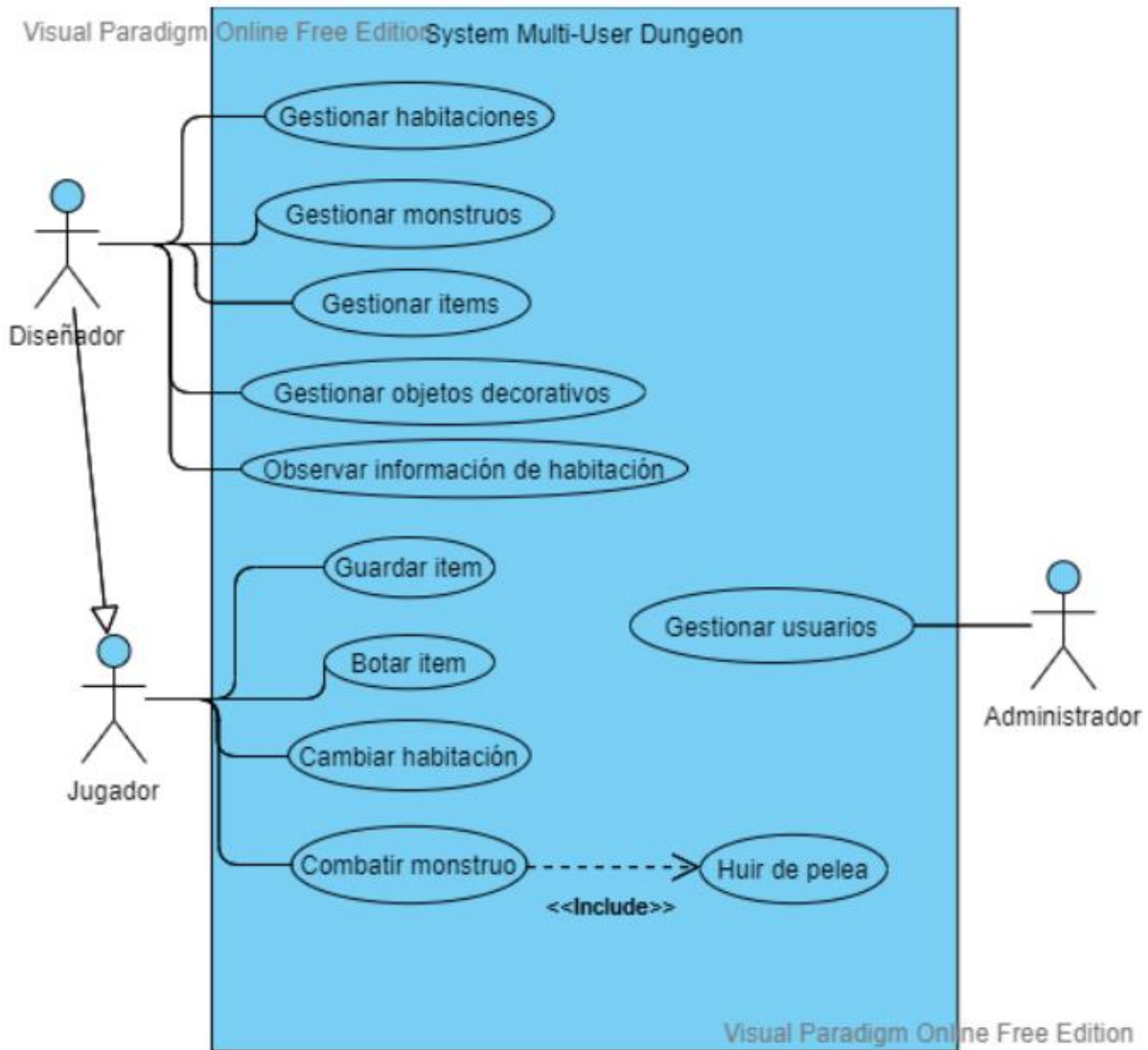
**Jaime Andrés Pavlich Mariscal**

ASIGNATURA

**Desarrollo web**

PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS  
BOGOTÁ, D.C.  
2022-I

**Diagrama de casos de uso:**



**Descripción de casos de uso:**

<b>Caso de uso</b>	Gestionar habitaciones.
<b>Actores</b>	Usuario diseñador.
<b>Resumen</b>	El usuario diseñador tiene permisos para modificar la entidad habitación del juego y su contenido.
<b>CU asociados</b>	Gestionar monstruo, gestionar objetos decorativos, gestionar ítems.
<b>Precondiciones</b>	Estar registrado como usuario diseñador y tener iniciada la sesión.
<b>Flujo básico</b>	
1. Se ingresa al listado de habitaciones de la mazmorra.	

2. Se escoge alguna opción de edición de las habitaciones listadas (creación, eliminación, modificación). <ul style="list-style-type: none"> <li>a. Se elige opción de modificación haciendo clic en el id de la habitación. <ul style="list-style-type: none"> <li>i. El diseñador modifica los datos deseados.</li> <li>ii. Se guardan los cambios.</li> </ul> </li> <li>b. Se elige opción de eliminación. <ul style="list-style-type: none"> <li>i. Se confirma la acción.</li> </ul> </li> <li>c. Se elige opción de creación. <ul style="list-style-type: none"> <li>i. Se llenan las casillas de datos necesarias para crear la nueva habitación.</li> <li>ii. Se confirma la acción.</li> </ul> </li> </ul>	
<b>Flujo alterno</b>	
2. Se ingresan datos incorrectos en la modificación o creación de una habitación. <ul style="list-style-type: none"> <li>a. Se redirige a página de error.</li> <li>b. No se realizan cambios en la base de datos.</li> </ul>	
<b>Requisitos no funcionales</b>	1. <b>Eficiencia:</b> El sistema debe ser capaz de permitir la gestión de varias habitaciones al mismo tiempo. 2. <b>Usabilidad:</b> La función debe tener un tiempo de aprendizaje menor a 5 minutos. 3. <b>Seguridad:</b> Se debe validar rol de diseñador.
<b>Postcondiciones</b>	Los cambios realizados por el diseñador quedan guardados y se reflejan en la base de datos.

<b>Caso de uso</b>	Gestionar monstruos.
<b>Actores</b>	Usuario diseñador.
<b>Resumen</b>	El usuario diseñador tiene permisos para modificar la entidad monstruo del juego y sus datos.
<b>CU asociados</b>	Gestionar habitación.
<b>Precondiciones</b>	Estar registrado como usuario diseñador y tener iniciada la sesión.
<b>Flujo básico</b>	
1. Se ingresa al listado de monstruos de la mazmorra. 2. Se escoge alguna opción de edición de los monstruos listados (creación, eliminación, modificación). <ul style="list-style-type: none"> <li>a. Se elige opción de modificación haciendo clic en el nombre del monstruo. <ul style="list-style-type: none"> <li>i. El diseñador modifica los datos deseados.</li> <li>ii. Se guardan los cambios.</li> </ul> </li> <li>b. Se elige opción de eliminación. <ul style="list-style-type: none"> <li>i. Se confirma la acción.</li> </ul> </li> <li>c. Se elige opción de creación. <ul style="list-style-type: none"> <li>i. Se llenan las casillas de datos necesarias para crear el nuevo monstruo.</li> <li>ii. Se confirma la acción.</li> </ul> </li> </ul>	
<b>Flujo alterno</b>	
2. Se ingresan datos incorrectos en la modificación o creación de un monstruo. <ul style="list-style-type: none"> <li>a. Se redirige a página de error.</li> <li>b. No se realizan cambios en la base de datos.</li> </ul> 2. Se asocia monstruo a una habitación inexistente. <ul style="list-style-type: none"> <li>a. Se muestra mensaje indicando el error.</li> </ul>	

b. No se realizan cambios en la base de datos. 2. Se asocia monstruo a una habitación con un monstruo asociado previamente. a. Se muestra mensaje indicando el error. b. No se realizan cambios en la base de datos.	
<b>Requisitos no funcionales</b>	1. <b>Eficiencia:</b> El sistema debe ser capaz de permitir la gestión de varias habitaciones al mismo tiempo. 2. <b>Usabilidad:</b> La función debe tener un tiempo de aprendizaje menor a 5 minutos. 3. <b>Seguridad:</b> Se debe validar rol de diseñador.
<b>Postcondiciones</b>	Los cambios realizados por el diseñador quedan guardados y se reflejan en la base de datos.

<b>Caso de uso</b>	Gestionar ítems.
<b>Actores</b>	Usuario diseñador.
<b>Resumen</b>	El usuario diseñador tiene permisos para modificar la entidad item del juego y su información.
<b>CU asociados</b>	Gestionar habitación.
<b>Precondiciones</b>	Estar registrado como usuario diseñador y tener iniciada la sesión.
<b>Flujo básico</b>	
1. Se ingresa al listado de ítems de la mazmorra. 2. Se escoge alguna opción de edición de los ítems listadas (creación, eliminación, modificación). a. Se elige opción de modificación haciendo clic en el nombre del ítem. i. El diseñador modifica los datos deseados. ii. Se guardan los cambios. b. Se elige opción de eliminación. i. Se confirma la acción. c. Se elige opción de creación. i. Se llenan las casillas de datos necesarias para crear el nuevo ítem. ii. Se confirma la acción.	
<b>Flujo alterno</b>	
2. Se ingresan datos incorrectos en la modificación o creación de un ítem. a. Se redirige a página de error. b. No se realizan cambios en la base de datos. 2. Se asocia ítem a una habitación inexistente. a. Se muestra mensaje indicando el error. b. No se realizan cambios en la base de datos.	
<b>Requisitos no funcionales</b>	1. <b>Eficiencia:</b> El sistema debe ser capaz de permitir la gestión de varios ítems al mismo tiempo. 2. <b>Usabilidad:</b> La función debe tener un tiempo de aprendizaje menor a 5 minutos. 3. <b>Seguridad:</b> Se debe validar rol de diseñador.
<b>Postcondiciones</b>	Los cambios realizados por el diseñador quedan guardados y se reflejan en la base de datos.

<b>Caso de uso</b>	Gestionar objetos decorativos.
--------------------	--------------------------------

<b>Actores</b>	Usuario diseñador.
<b>Resumen</b>	El usuario diseñador tiene permisos para modificar la entidad objetos decorativo del juego y su información.
<b>CU asociados</b>	Gestionar habitación.
<b>Precondiciones</b>	Estar registrado como usuario diseñador y tener iniciada la sesión.
<b>Flujo básico</b>	
<ol style="list-style-type: none"> <li>1. Se ingresa al listado de objetos decorativos de la mazmorra.</li> <li>2. Se escoge alguna opción de edición de los objetos decorativos listados (creación, eliminación, modificación). <ol style="list-style-type: none"> <li>a. Se elige opción de modificación haciendo clic en el nombre del objeto decorativo. <ol style="list-style-type: none"> <li>i. El diseñador modifica los datos deseados.</li> <li>ii. Se guardan los cambios.</li> </ol> </li> <li>b. Se elige opción de eliminación. <ol style="list-style-type: none"> <li>i. Se confirma la acción.</li> </ol> </li> <li>c. Se elige opción de creación. <ol style="list-style-type: none"> <li>i. Se llenan las casillas de datos necesarias para crear el nuevo objeto decorativo.</li> <li>ii. Se confirma la acción.</li> </ol> </li> </ol> </li> </ol>	
<b>Flujo alternativo</b>	
<ol style="list-style-type: none"> <li>2. Se ingresan datos incorrectos en la modificación o creación de un objeto decorativo. <ol style="list-style-type: none"> <li>a. Se redirige a página de error.</li> <li>b. No se realizan cambios en la base de datos.</li> </ol> </li> <li>2. Se asocia objeto decorativo a una habitación inexistente. <ol style="list-style-type: none"> <li>a. Se muestra mensaje indicando el error.</li> <li>b. No se realizan cambios en la base de datos.</li> </ol> </li> </ol>	
<b>Requisitos no funcionales</b>	<ol style="list-style-type: none"> <li>1. <b>Eficiencia:</b> El sistema debe ser capaz de permitir la gestión de varios objetos decorativos al mismo tiempo.</li> <li>2. <b>Usabilidad:</b> La función debe tener un tiempo de aprendizaje menor a 5 minutos.</li> <li>3. <b>Seguridad:</b> Se debe validar rol de diseñador.</li> </ol>
<b>Postcondiciones</b>	Los cambios realizados por el diseñador quedan guardados y se reflejan en la base de datos.

<b>Caso de uso</b>	Observar información de habitación.
<b>Actores</b>	Usuario diseñador.
<b>Resumen</b>	El usuario diseñador tiene permisos para observar los datos de las habitaciones de la mazmorra.
<b>CU asociados</b>	Gestionar habitación.
<b>Precondiciones</b>	Estar registrado como usuario diseñador y tener iniciada la sesión.
<b>Flujo básico</b>	
<ol style="list-style-type: none"> <li>1. Se ingresa al listado de habitaciones de la mazmorra.</li> <li>2. Se hace clic sobre el id de una habitación.</li> <li>3. Se redirige a la página de edición donde se pueden visualizar los datos actuales de la habitación seleccionada.</li> </ol>	
<b>Flujo alternativo</b>	
---	

<b>Requisitos no funcionales</b>	<ol style="list-style-type: none"> <li>1. <b>Eficiencia:</b> El sistema debe ser capaz de permitir la visualización de varias habitaciones al mismo tiempo desde diferentes cuentas de usuarios diseñadores.</li> <li>2. <b>Usabilidad:</b> La función debe tener un tiempo de aprendizaje menor a 5 minutos.</li> </ol> <p><b>Seguridad:</b> Se debe validar rol de diseñador.</p>
<b>Postcondiciones</b>	---

<b>Caso de uso</b>	Guardar ítem.
<b>Actores</b>	Jugador, diseñador.
<b>Resumen</b>	Durante la partida el jugador puede tomar ítems de la habitación y guardarlos en su inventario.
<b>CU asociados</b>	Botar ítem.
<b>Precondiciones</b>	Estar registrado como usuario diseñador o jugador y tener iniciada la sesión y estar jugando una partida. Encontrarse en una habitación con ítems.
<b>Flujo básico</b>	
1. El jugador selecciona un ítem de la habitación y lo almacena.	
<b>Flujo alterno</b>	
1. El jugador no cuenta con espacio de almacenamiento en el inventario. <ol style="list-style-type: none"> <li>a. Se muestra mensaje indicando el error.</li> <li>b. No se realiza la acción.</li> </ol>	
<b>Requisitos no funcionales</b>	<ol style="list-style-type: none"> <li>1. <b>Concurrencia:</b> El sistema debe ser capaz de permitir la interacción de varios jugadores con el objeto y reflejar estos cambios sin producir problemas de sincronización.</li> <li>2. <b>Usabilidad:</b> La función debe tener un tiempo de aprendizaje menor a 1 minuto.</li> </ol>
<b>Postcondiciones</b>	El ítem se guarda en el inventario del jugador removiéndose de la habitación y este cambio se ve reflejado en la información del ítem y la habitación en la base de datos.

<b>Caso de uso</b>	Botar ítem.
<b>Actores</b>	Jugador, diseñador.
<b>Resumen</b>	Durante la partida el jugador puede seleccionar un ítem de su inventario y botarlo en la habitación en la que se encuentre.
<b>CU asociados</b>	Guardar ítem.
<b>Precondiciones</b>	Estar registrado como usuario diseñador o jugador y tener iniciada la sesión y estar jugando una partida. Tener ítems en el inventario.
<b>Flujo básico</b>	
1. El jugador selecciona un ítem de su inventario y lo desecha en la habitación.	
<b>Flujo alterno</b>	
1. El jugador no tiene ítems en su inventario.	

a. Se muestra mensaje indicando que no hay ítems en el inventario.	
<b>Requisitos no funcionales</b>	<ol style="list-style-type: none"> <li>1. <b>Concurrencia:</b> El sistema debe ser capaz de permitir la interacción de varios jugadores con el objeto y reflejar estos cambios sin producir problemas de sincronización.</li> <li>2. <b>Usabilidad:</b> La función debe tener un tiempo de aprendizaje menor a 1 minuto.</li> </ol>
<b>Postcondiciones</b>	El ítem queda en la habitación donde se encuentre el jugador eliminándose de su inventario y este cambio se ve reflejado en la información del ítem y el jugador en la base de datos.

<b>Caso de uso</b>	Cambiar habitación.
<b>Actores</b>	Jugador, diseñador.
<b>Resumen</b>	Durante la partida el jugador puede desplazarse a las habitaciones contiguas a la habitación en la que se encuentre en ese momento a través de una salida.
<b>CU asociados</b>	---
<b>Precondiciones</b>	Estar registrado como usuario diseñador o jugador y tener iniciada la sesión y estar jugando una partida.
<b>Flujo básico</b>	
1. El jugador selecciona una salida para cambiar de habitación.	
<b>Flujo alterno</b>	
---	
<b>Requisitos no funcionales</b>	<ol style="list-style-type: none"> <li>1. <b>Concurrencia:</b> El sistema debe ser capaz de permitir el movimiento de varios jugadores entre habitaciones y reflejar estos cambios sin producir problemas de sincronización ni errores.</li> <li>2. <b>Usabilidad:</b> La función debe tener un tiempo de aprendizaje menor a 1 minuto.</li> </ol>
<b>Postcondiciones</b>	El jugador cambia de habitación, esta información se ve reflejada en la información del jugador y la habitación en la base de datos.

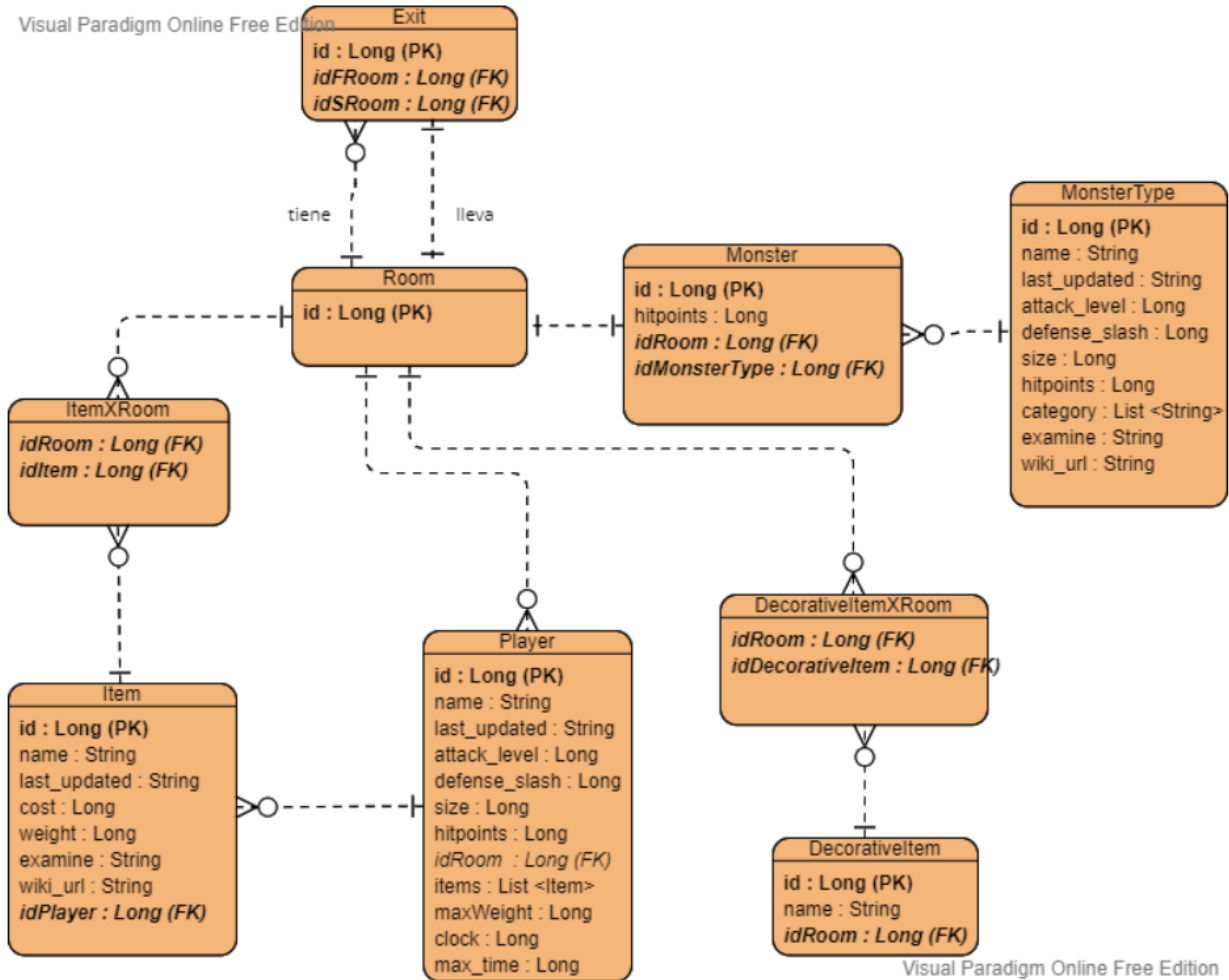
<b>Caso de uso</b>	Combatir monstruo.
<b>Actores</b>	Jugador, diseñador.
<b>Resumen</b>	Durante la partida el jugador puede combatir con el monstruo de esa habitación si hay uno.
<b>CU asociados</b>	---
<b>Precondiciones</b>	Estar registrado como usuario diseñador o jugador y tener iniciada la sesión y estar jugando una partida. Encontrarse en una habitación con un monstruo presente.
<b>Flujo básico</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de combatir un monstruo presente en la habitación.</li> <li>2. El usuario selecciona la opción de atacar o huir.</li> </ol>	
<b>Flujo alterno</b>	
---	

<b>Requisitos no funcionales</b>	<ol style="list-style-type: none"> <li>1. <b>Concurrencia:</b> El sistema debe ser capaz de permitir la interacción de varios jugadores con el monstruo y reflejar estos cambios sin producir problemas de sincronización.</li> <li>2. <b>Usabilidad:</b> La función debe tener un tiempo de aprendizaje menor a 1 minuto.</li> </ol>
<b>Postcondiciones</b>	Dependiendo de la acción elegida el monstruo recibe daño el cual se refleja en la información de este en la base de datos o el jugador escapa de la interacción de combate recibiendo un ataque de parte del monstruo.

<b>Caso de uso</b>	Gestionar usuarios.
<b>Actores</b>	Administrador.
<b>Resumen</b>	El usuario administrador tiene permisos para modificar la entidad jugador del juego y su información.
<b>CU asociados</b>	---
<b>Precondiciones</b>	Estar registrado como usuario administrador y tener iniciada la sesión.
<b>Flujo básico</b>	
<ol style="list-style-type: none"> <li>1. Se ingresa al listado de jugadores del juego.</li> <li>2. Se escoge alguna opción de edición de los jugadores listados (creación, eliminación, modificación). <ol style="list-style-type: none"> <li>a. Se elige opción de modificación haciendo clic en el nombre del jugador. <ol style="list-style-type: none"> <li>1. El diseñador modifica los datos deseados.</li> <li>2. Se guardan los cambios.</li> </ol> </li> <li>b. Se elige opción de eliminación. <ol style="list-style-type: none"> <li>1. Se confirma la acción.</li> </ol> </li> <li>c. Se elige opción de creación. <ol style="list-style-type: none"> <li>1. Se llenan las casillas de datos necesarias para crear el nuevo jugador.</li> <li>2. Se confirma la acción.</li> </ol> </li> </ol> </li> </ol>	
<b>Flujo alterno</b>	
<ol style="list-style-type: none"> <li>2. Se ingresan datos incorrectos en la modificación o creación de un jugador. <ol style="list-style-type: none"> <li>a. Se redirige a página de error.</li> <li>b. No se realizan cambios en la base de datos.</li> </ol> </li> </ol>	
<b>Requisitos no funcionales</b>	<ol style="list-style-type: none"> <li>1. <b>Eficiencia:</b> El sistema debe ser capaz de permitir la gestión de varios jugadores al mismo tiempo.</li> <li>2. <b>Usabilidad:</b> La función debe tener un tiempo de aprendizaje menor a 5 minutos.</li> <li>3. <b>Seguridad:</b> Se debe validar rol de administrador.</li> </ol>
<b>Postcondiciones</b>	Los cambios realizados por el administrador quedan guardados y se reflejan en la base de datos.

**Diagrama lógico de datos:**





### Descripción de relaciones de entidades:

- Exit:** Indica la conexión entre dos habitaciones, una habitación puede tener varias salidas las cuales conectan habitaciones contiguas, el jugador puede usarlas para desplazarse entre habitaciones.
  - Room:** Lleva a una habitación.
- Room:** Una habitación es el espacio a través del cual se desplaza el jugador en la mazmorra, estas habitaciones pueden contener un monstruo y varios jugadores, ítems y objetos decorativos. Están conectadas con otras habitaciones a través de salidas.
  - Exit:** Puede tener varias salidas.
  - Monster:** Puede tener un monstruo.
  - Ítem:** Puede contener varios ítems.
  - Decorative Item:** Puede contener varios objetos decorativos.
  - Player:** Puede contener varios jugadores presentes.
- Ítem:** Objetos que se encuentran en diferentes habitaciones de la mazmorra, tienen un peso y un precio asociado, pueden ser recolectados por jugadores y almacenados en sus inventarios.

- **Player:** Estos recorren las habitaciones de la mazmorra recogiendo objetos para acumular puntos. Tienen un tiempo limitado el cual se va agotando por cada habitación que recorran. Pueden enfrentar al monstruo presente en las habitaciones.
  - **Ítem:** Puede almacenar en su inventario varios objetos.
- **Decorative ítem:** Objetos que se encuentran en las diferentes habitaciones e la mazmorra, no tienen un valor monetario asociado.
- **Monster:** Una entidad que puede estar presente en las habitaciones, su presencia impide que un jugador pueda recoger ítems de la habitación por lo que debe ser derrotado primero para que el jugador pueda realizar esta acción.
- **Monster type:** Indica los tipos de monstruo que pueden estar presentes en las habitaciones de la mazmorra.
  - **Monster:** Varios monstruos ubicados en diferentes habitaciones de la mazmorra pueden tener asociado el mismo tipo, sus hp son independientes.