

MEMORIA: Modular Zork

Listado de scripts para las vistas

- **Vista inicio**

- ***index.php***

Consta de dos vistas diferentes:

- ***includes/contenidos/contenidoInicio.php***

Usuario sin registrar

- ***includes/FormularioSignin.php***

En esta vista tenemos en el bloque superior una breve descripción de la página web y su funcionamiento. Y en el bloque inferior tenemos un formulario para que el usuario pueda registrarse introduciendo su nombre de usuario, contraseña y email.

Usuario registrado

En esta vista disponemos de información que explica en qué consiste el juego.

- **Vista registrate**

- ***signin.php***

- ***includes/FormularioSignin.php***

Vista que muestra un formulario para registrarse en la web.

- **Vista juego**

- ***juego.php***

- ***includes/contenidos/contenidoJuego.php***

Desde este script se accede a game.js en donde se encuentra la lógica del juego, si el usuario ha iniciado sesión y tiene partidas empezadas se le proporciona un botón para que pueda cargarlas. Si el usuario no tiene partidas empezadas solo se le dará la opción de empezar una nueva partida.

Si el usuario no ha iniciado sesión se le proporcionará la posibilidad de que juegue a un mapa de demostración, con un personaje fijo y un mapa fijo.

- ***mapaContiene.php/mapaMazmorra.php/mazmorras.php***

Estos archivos se usan como conexión entre la base de datos y los archivos javascript a la hora de hacer la creación y el juego.

- ***javascripts:***

- ***loadGameFinal.js***

Realiza la carga desde la base de datos de un juego , cargando el mapa con las correspondientes mazmorras, consumibles y enemigos que contendrá, incluyendo si el mapa ha sido cargado como una partida anteriormente empezada.

- ***Objetos para la construcción del mapa:***

Se utiliza javascript con un comportamiento de “clases”, cada una de las clases proporcionadas corresponden a un elemento.

- ***Mazmorras.js***

Contiene la información necesaria para poder asignarle monstruos y consumibles a cada una de ellas que estén en el mapa.

- ***Monstruo.js***

Contiene toda la información necesaria para poder manejar la información de cada enemigo en el mapa..

- ***Personaje.js***

Contiene toda la información necesaria para poder manejar al personaje durante toda su aventura en el mapa.

- ***consumible.js***

Contiene la información necesaria para que el personaje pueda interactuar con ellos y se sumen a sus estadísticas.

- ***Mapa.js***

El mapa es el método de funcionamiento proporciona un medio para poder moverse entre las mazmorras, una lista de adyacencia, con un comportamiento muy parecido al de un grafo.

También realiza el guardado en la base de datos mediante el javascript ***guardarPartida.js***, en el cual se realizan consultas ajax de tipo post y get a PHP, a diferentes archivos alojados en la raíz del proyecto, cada uno de los llamara a su

correspondiente clase en includes que realizará la operación de la base de datos.

En este javascript, se lleva a cabo también la carga de partidas que haya empezado un determinado usuario, para ello se utiliza ***arrayMzUsadas.js*** e ***infoObjeto.js***.

- ***gameFinal.js***

En este javascript se lleva a cabo toda la lógica del juego, como: moverse entre mazmorras, atacar enemigos, guardar una partida ..etc.

Por último cuando el usuario finalice la partida solo y cuando haya ganado, se actualizaran sus datos en la Bd mediante ajax, para poder asignarle puntuación y monedas para las compras en el juego, tambien se borrarán en las tablas de la base de datos todos los datos correspondientes a esa partida terminada..

- ***Vista ranking***

- ***ranking.php***

En este apartado de la aplicación se muestran ordenados de mayor a menor los jugadores registrados con las mejores puntuaciones, conseguidos a partir de creación y de juego. Se realizará una consulta en la base de datos para obtener el listado.

- ***Vista tienda***

- ***tienda.php***

Página principal de tienda. Muestra los elementos que se venden divididos en tres secciones: Objetos, Enemigos, Mapas y Personajes.

Pulsando en el título de cada sección se accede a "SeccionTienda.php".

- ***tiendaSeccion.php***

Muestra la lista de todos los elementos de una sección (Objetos, Enemigos, Mapas o Personajes) mostrando su imagen, nombre y precio para cada uno de ellos.

- ***tiendaObjeto.php***

Muestra la información sobre un elemento de la tienda

Muestra la imagen del objeto y todas sus características, así como el precio y su nombre. Además incluye un botón para comprar el objeto. Este botón redirige a ***misCompras.php***

- ***infoTienda.php***
Informa al usuario de la cantidad de monedas que tiene y le permite conseguir más monedas mediante un enlace a añadirMonedas.php.
- ***añadirMonedas.php***
Añade a la cuenta del usuario el número de monedas(entre 1 y 10000) que desee.
- ***objetoTienda.php***
Muestra la información del objeto seleccionado en la tienda

Vista creación

Para la vista creación se han generado los siguientes scripts:

Pantalla 1:

- ***creacion.php***
 - ***includes/contenidos/contenidoCreacion1.php***
Tiene el contenido de la página.

Este script sirve para crear un nuevo juego a partir de ciertos contenidos.

Se da la opción de elegir un único mapa entre las múltiples opciones (los enlaces al final del bloque “Elige mapa”, nos permiten cargar más opciones), seleccionando la imagen a la izquierda de la breve descripción del mapa.

Tenemos también un bloque para elegir a los monstruos y objetos (“Elige monstruo” y “Elige objeto”, respectivamente), donde podremos seleccionar la imagen de los monstruos/objetos que deseemos usar en el proceso de creación de nuestro juego. Cada bloque tendrá enlaces para cargar más objetos y monstruos, relacionados con el mapa seleccionado.

Una vez seleccionado el mapa y el contenido de este, haríamos click en el botón crear, el cual llamaría a creacion.php para mostrarnos la siguiente pantalla.

Pantalla 2:

- ***creacion1.php***
 - ***includes/contenidos/contenidoCreacion2.php***
Tiene el contenido de la página.

Desde el sidebar izquierdo se podrán seleccionar con el ratón los objetos que se deseen añadir al mapa actual (esta funcionalidad se implementará más adelante).

Tendremos un bloque que contiene las 3 posibles preguntas relacionadas con el mapa y los objetos añadidos, de las cuales solo se podrá elegir 1. También tendremos un bloque de respuestas con 5 opciones de las cuales deberemos escoger 3.

Una vez escogida la pregunta y las respuestas, seleccionaremos el botón siguiente, para crear otra pregunta/respuesta o podremos volver a la pregunta anterior para rectificar algún dato.

Cuando se creen todas las preguntas/respuestas, podremos empezar a jugar.

- **Vista ayuda**

Pantalla 1:

- ***ayuda.php***

En este script se muestran distintas selecciones que enlazan con otros scripts que explicarán el apartado que se describe como título.

Pantalla 2:

- ***includes/contenidos/contenidoAyuda.php***

Se listan los contenidos que se mostraran a continuación:

- ***objetivo.php***

Aquí se explican el objetivo principal del juego, la dificultad de cada nivel y cómo se realizan los desplazamientos y acciones.

- ***ayudaRanking.php***

En este script se explica cómo funciona el ranking y cómo se puede subir en él además de las consecuencias que tendrá subir en el ranking.

- ***ayudaCompras.php***

En esta sección se explica cómo acceder a la tienda. A continuación se detalla cómo se seleccionarán los objetos y cómo se realizarán las compras además de explicar cómo se obtendrán las monedas para emplear aquí.

- ***ayudaCreacion.php***

En este script se detalla paso a paso cómo acceder a la parte de creación además de una explicación paso a paso para realizar un primer mapa dentro de la aplicación.

- **Visita mi perfil**

- ***contenidoMiPerfil.php***

Aquí encontramos información acerca del usuario que ha iniciado sesión permitiéndole cambiar datos de su perfil. se ha incluido un formulario de tipo file para que pueda disponer de una forma para cambiar su icono.

- ***cambiarNombre.php***

Aquí se proporciona al usuario un formulario para poder cambiar su nombre de usuario.

- ***includes/FormularioCambiarNombre.php***

Se realiza una consulta con la base de datos de tipo update cambiando el contenido de la tabla de usuarios, en concreto el nombre.

- ***cambiarCorreo.php***

Aquí se proporciona al usuario un formulario para poder cambiar su correo de inicio de sesión.

- ***includes/FormularioCambiarCorreo.php***

Se realiza una consulta con la base de datos de tipo update cambiando el contenido de la tabla de usuarios, en concreto el correo.

- ***cambiarContraseña.php***

Aquí se proporciona al usuario un formulario para poder cambiar su contraseña de inicio de sesión.

- ***includes/FormularioCambiarContraseña.php***

Se realiza una consulta con la base de datos de tipo update cambiando el contenido de la tabla de usuarios, en concreto la contraseña.

Todos los formularios son gestionados mediante la aplicación.php.

- ***Vista mis compras***

- ***misCompras.php***

En esta vista se proporciona al usuario una lista de los objetos que ha comprado en la tienda de la aplicación.

- ***includes/contenidos/contenidoMisCompras.php***

En este archivo realizamos una tabla con información acerca de los objetos que ha comprado el usuario.

- ***Vista mis Creaciones***

- ***misCreaciones.php***

- ***includes/contenidos/contenidoMisCreaciones.php***

En esta vista mostremos en una tabla al usuario los mapas que ha creado con información acerca de cuánto se ha jugado y su valoración media de los usuarios.

- **Vista administrador**

- **admin.php**

- En esta vista proporcionamos a los usuarios con rol de administrador, la posibilidad de añadir nuevos objetos al mercado, Enviar mensajes a usuarios identificados por su correo electrónico, bloquear usuarios y eliminar objetos.

- **añadirObjetos.php**

- Se proporciona al administrador un formulario con los campos necesarios para registrar un nuevo objeto en la base de datos.

- **enviarMensaje.php**

- Se proporciona al administrador un hilo de mensajes para poder comunicarse con un usuario.

- **bloquearUsuario.php**

- Se proporciona al administrador un formulario para indicar el correo del usuario al que desea bloquear.

- **includes/FormularioObjetos.php**

- Se realizará una consulta con la base de datos de tipo insert para añadir un nuevo objeto a la base de datos.

- **includes/FormularioMensaje.php**

- Formulario correspondiente para mantener un hilo de conversación entre miembros de la aplicación.

- **includes/FormularioBloquearUsuario.php**

- Se realiza una consulta con la base de datos de tipo update cambiando el contenido de la tabla de usuarios, eliminando o imposibilitando el acceso al usuario.

- **Vista sidebar izquierdo**

- Para la vista sidebar izquierdo se han generado los siguientes scripts:*

- **includes/comun/sidebarIzq.php**

- Este script llama siempre a menuSideBarIzq.php, para cargar el menú que nos permite ver dentro del sidebar el contenido de las noticias, los eventos o el cofre.

- **noticias.php**

- Este script mostrará noticias (principalmente texto o imágenes, más adelante se añadirán videos), acerca de temas relacionados con la web o con juegos similares a Zork.

- ***eventos.php***

Este script mostrará eventos que se produzcan en la página, así como competiciones, premios, etc.(En este caso solo se muestran imágenes).

- ***cofre.php***

El script del cofre mostrará dependiendo de la vista en la que se encuentre unos objetos u otros. Por ejemplo, en juego sólo nos mostrará los objetos de los que disponemos en nuestro perfil (en caso de estar registrado). En creación nos mostrará nuestros objetos y los objetos elegidos en la vista de creación. Será una lista que contiene los objetos de la BBDD.

- ***includes/comun/menuSideBarIzq.php***

Es un script auxiliar, que nos permite incluir un menú para el sidebar izquierdo por el cual accederemos a noticias, eventos o cofre, a través de los botones.

- ***incluirObjetos.php***

Script auxiliar para mostrar de forma temporal (hasta que accedemos a la BBDD) ítems dentro de la lista, que se muestra al llamar al contenido cofre.php .

- **Vista login/logout**

- ***login.php***

- ***includes/FormularioLogin.php***

Vista en la que aparece un formulario para iniciar sesión. Se deben introducir los campos: Email y contraseña.

- ***logout.php***

Vista en la que aparece un mensaje de despedida para el usuario, indicando que se ha cerrado la sesión.

- **Vista cabecera**

- ***includes/comun/cabecera.php***

Aquí encontraremos el nombre de la aplicación además de un barra de herramientas con la que podremos navegar por las distintas secciones de la aplicación.

- **Vista pie de página**

- ***includes/comun/pie.php***

- Este script contiene los enlaces a la sección de desarrolladores y contacto.

- ***contacto.php***

- En este apartado se le darán al usuario distintas opciones para marcar una de ellas y comunicarse con los administradores de la aplicación para reportar problemas, sugerencias o solicitar ayuda técnica.

- ***desarrolladores.php***

- En este script se encuentra la información de los distintos desarrolladores de la aplicación y una pequeña descripción además de un correo de contacto.

- ***includes/contenidos/contenidoContacto.php***

- ***includes/contenidos/contenidoDesarrolladores.php***

- Cada uno de estos php es el contenido de sus respectivas páginas.

Listado de scripts adicionales (Acceso a BBDD, definiciones de clases, lógica de la app)

- ***FormularioCambiarNombre.php***

- Se realiza una consulta con la base de datos de tipo update cambiando el contenido de la tabla de usuarios, en concreto el nombre.

- ***FormularioCambiarCorreo.php***

- Se realiza una consulta con la base de datos de tipo update cambiando el contenido de la tabla de usuarios, en concreto el correo.

- ***FormularioCambiarContraseña.php***

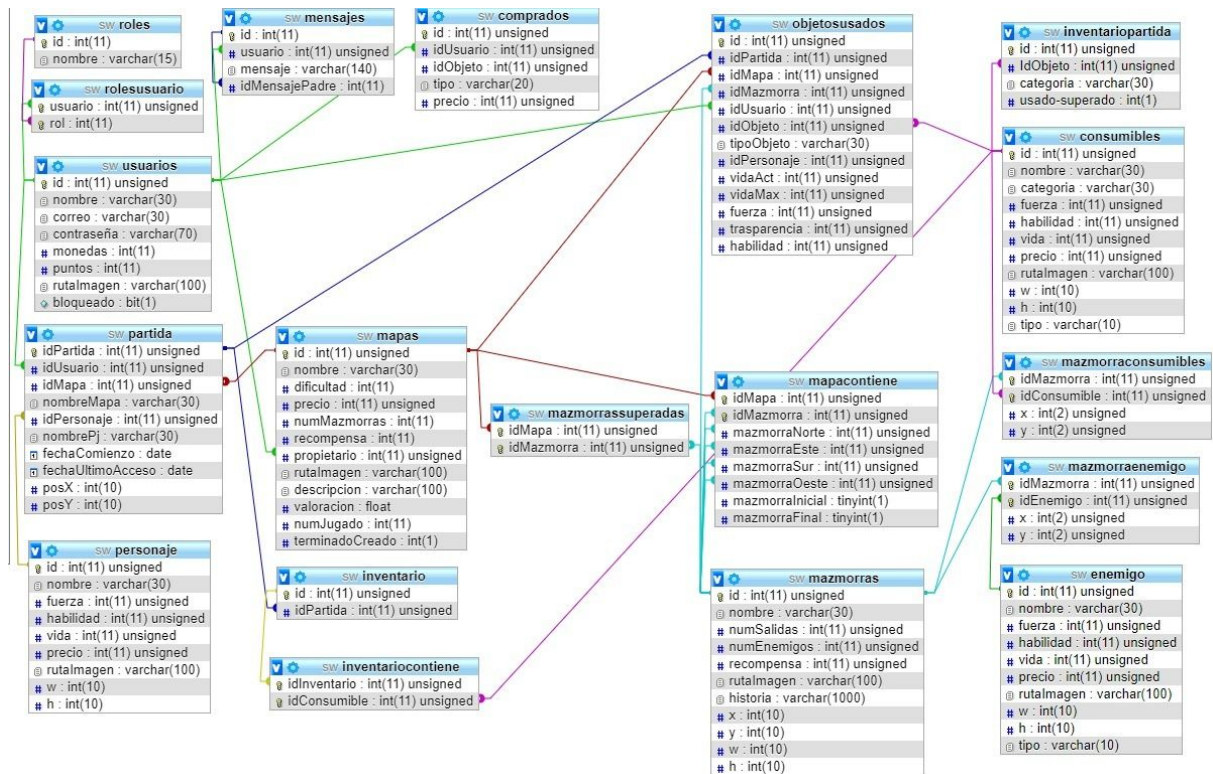
- Se realiza una consulta con la base de datos de tipo update cambiando el contenido de la tabla de usuarios, en concreto la contraseña.

- ***FormularioObjetosConsumibles.php***

- Se realizará una consulta con la base de datos de tipo insert para añadir un nuevo objeto a la base de datos.

- ***FormularioMensaje.php y FormularioRespuesta.php***
Formularios correspondientes para mantener un hilo de conversación entre miembros de la aplicación.
- ***FormularioBloquearUsuario.php***
Se realiza una consulta con la base de datos de tipo update cambiando el contenido de la tabla de usuarios, eliminando o imposibilitando el acceso al usuario.
- ***FormularioSignin.php***
Se realiza una consulta con la base de datos de tipo insert para llevar a cabo la incorporación de un usuario a la base de datos.
- ***FormularioLogin.php***
Se realiza una consulta con la base de datos de tipo select para comprobar la existencia del usuario y comprobar si los parámetros del formulario coinciden con los datos de la BBDD.
- ***Usuario.php***
Es el medio de comunicación entre los formularios, la aplicación y la base de datos.
- ***FormularioBotonComprar.php***
Si el usuario está conectado, realiza la compra del objeto de la tienda.
- ***FormularioBotonMonedas.php***
Si el usuario está conectado, realiza la compra del número de monedas introducido por el usuario.
- ***FormularioConexionMazmorras.php***
En la creación se encarga de indicar con qué mazmorras conecta por las distintas direcciones cada una de la mazmorra dada.
- ***FormularioCreacionMapa.php***
En la creación se encarga de definir los parámetros iniciales para la creación de un nuevo mapa

Estructura de la base de datos



ROLES

En esta tabla se almacenarán los posibles roles que se puedan dar a los usuarios junto con un id único.

-id: identifica el rol que tendrá el usuario.

-nombre: consta de un tipo varchar que identificará el tipo de rol, administrador, usuario, moderador.

ROLES_USUARIO

En esta tabla se guardarán el id de del usuario y el id del rol que tiene asignado.

-usuario: identifica el usuario que será el que tendrá el rol.

-rol: es la clave externa que señala al id de la tabla de roles.

USUARIOS

En esta entidad se almacenarán los datos del usuario, asociados a un id único.

-id: identifica al usuario de forma única.

-nombre: propio de cada usuario y no se podrá repetir dentro de la aplicación, pero no se guardarán en la base de datos como clave primaria.

-correo: propio de cada usuario, solo puede haber uno por usuario.

-contraseña: dará acceso al perfil de la aplicación del usuario.

-monedas: cantidad de monedas que tiene un usuario

- puntos:** cantidad de puntos que ha obtenido un usuario.
- rutaImagen:** ruta en la que se encuentra la foto de perfil del usuario.
- bloqueado:** controla si el usuario ha sido bloqueado por un administrador.

MAPAS

En esta entidad de la base de datos se guardarán todos los datos necesarios para identificar y que sea persistente un mapa.

- idMapa:** identifica al mapa de forma única, siendo clave primaria.
- nombre:** propio de cada mapa, el usuario/administrador podrá elegirlo durante su creación.
- dificultad:** dificultad asociada al mapa, el administrador y moderadores podrán rellenar este campo una vez el mapa haya sido probado.
- precio:** precio del mapa en la tienda que habrá en la aplicación, solo los administradores podrán tener acceso a rellenar este campo.
- numMazmorras:** será el número de mazmorras que dispone el mapa.
- recompensa:** la recompensa por superar el mapa será elegida por los moderadores antes de ponerse a disposición de los jugadores/usuarios.
- propietario:** usuario que ha creado el mapa, este atributo se usará para dar puntos al usuario cada vez que otro usuario juegue a su mapa.
- rutaImagen:** ruta de la imagen del mapa
- descripción:** descripción del mapa, dada por el propietario
- valoración:** una vez el jugador/usuario se haya pasado el mapa se pasará al mismo un formulario para evaluarlo.
- numJugado:** número de veces que ha sido jugado el mapa para mostrarse en la lista de misCreaciones.php.
- terminadoCreado:** estado del mapa si ha sido terminado por el usuario o creado.

MAZMORRAS

Para la entidad mazmorra se almacenarán el número de salidas de esa mazmorra junto con el número de objetos y de enemigos que hay en ella.

- id:** identifica a la mazmorra de forma única.
- nombre:** nombre de la mazmorra para ser anunciada posteriormente en la tienda y en el menu de creacion.
- numSalidas:** número de salidas de las que dispone la mazmorra, serán utilizadas para dar un número determinado de respuestas posibles al usuario.
- recompensa:** el progreso del usuario será guardado en forma de puntuación, así cada vez que juegue a un mapa, aunque no lo haya terminado, podrá disponer de una puntuación en función de las mazmorras que haya superado.
- numEnemigos:** número de enemigos que hay en esa mazmorra.
- rutaImagen:** ruta de la imagen que usa la mazmorra.
- historia:** texto breve que describe la mazmorra.

ENEMIGO

En la entidad enemigo se guardan el nombre y el identificador único para cada enemigo, junto con su habilidad, fuerza, precio y vida.

- id:** identifica al enemigo de forma única.
- nombre:** nombre del enemigo para ser anunciado posteriormente en la tienda y en el menu de creacion.
- precio:** cantidad de monedas necesarias para obtener el enemigo.
- fuerza:** fuerza con la que ataca el enemigo al personaje.
- habilidad:** resistencia que opondrá el enemigo para ser derrotado.
- vida:** número de puntos de daño que puede llegar a soportar el enemigo.
- rutaImagen:** ruta de la imagen que usa el enemigo.
- w:** anchura que tiene el enemigo en la mazmorra.
- h:** altura que tiene el enemigo en la mazmorra.

PERSONAJE

Cada personaje en la tabla de datos vendrá identificado de manera única con un identificador. Además tendrá asociado un nombre, un inventario, una fuerza, habilidad y vida determinadas.

- id:** identifica al personaje de forma única.
- nombre:** nombre del personaje para ser seleccionado posteriormente a la hora de elegir el personaje con en el que jugaremos.
- fuerza:** fuerza con la que ataca el personaje al enemigo.
- habilidad:** resistencia que opondrá el personaje para ser derrotado.
- vida:** número de puntos de daño que puede llegar a soportar el personaje.
- precio:** cantidad de monedas necesarias para obtener el enemigo.
- rutaImagen:** ruta de la imagen que usa el personaje.
- w:** anchura que tiene el personaje en la mazmorra.
- h:** altura que tiene el personaje en la mazmorra.

CONSUMIBLES

Serán los objetos que se pueden adquirir en el mapa y que pueden ser adquiridos en la tienda. No permanecerán en el inventario del usuario ni del personaje después de terminar la partida.

- id:** identificador único para cada objeto y el cual sirve de clave primaria.
- nombre:** nombre del objeto.
- categoría:** categoría a la que pertenece.
- fuerza, habilidad y vida:** estos son los 3 atributos de un personaje que se podrán mejorar mediante estos objetos, dependiendo del objeto y categoría mejorará los atributos del personaje u otros.
- precio:** es el precio por el cual se podrán adquirir en la tienda.

- rutaImagen**: ruta de la imagen que usa el consumible.
- w**: anchura que tiene el consumible en la mazmorra.
- h**: altura que tiene el consumible en la mazmorra.
- tipo**: define el tipo de consumible que es.

INVENTARIO

Es el inventario único de cada personaje en el que podremos almacenar objetos (armas, escudos, llave, etc..) los cuales el personaje obtendrá a lo largo de la partida y con estos superara las pruebas y enfrentamientos a monstruos.

- id**: identificador único de cada inventario que usaremos como clave primaria.
- idPartida**: identifica en qué partida se está usando ese inventario

INVENTARIOCONTIENE

Se guarda la relación entre un inventario de un personaje y los objetos (armas, escudos, llaves, etc..) que contiene este.

- idInventario**: identificador del inventario.
- idConsumible**: identificador del objeto que se encuentra en el inventario.

MAPACONTIENE

En esta tabla se almacena la relación entre el mapa y las diferentes mazmorras que lo componen.

- idMapa**: identificador del mapa.
- idMazmorra**: guarda el identificador de la mazmorra que forma parte del mapa.
- mazmorraNorte**: contiene el id de la mazmorra que se encuentra hacia el norte.
- mazmorraSur**: contiene el id de la mazmorra que se encuentra hacia el sur.
- mazmorraEste**: contiene el id de la mazmorra que se encuentra hacia el este.
- mazmorraOeste**: contiene el id de la mazmorra que se encuentra hacia el oeste.
- mazmorraInicial**: esta variable toma valor uno si la mazmorra en la que se encuentra es la inicial.
- mazmorraFinal**: esta variable toma valor uno si la mazmorra en la que se encuentra es la final.

MAZMORRASSUPERADAS

En esta tabla guardaremos el progreso del usuario para que pueda cargarse partida y pueda abandonar en mitad de la misma.

- idMapa**: se guarda el identificador del mapa en el que se encuentra el usuario antes de abandonar la partida, como clave externa.
- idMazmorra**: se guarda la mazmorra en la que se quedó, como clave externa.

MAZMORRAENEMIGO

Guardaremos la información de los enemigos/monstruos que nos encontraremos en la mazmorra y a los que deberemos enfrentarnos para poder superar esta misma.

-idMazmorra: se guarda el identificador único de la mazmorra.

-idEnemigo: se almacena el identificador del enemigo o enemigos que nos encontraremos.

-x: coordenada x en la que se encuentra la esquina superior izquierda de la imagen del enemigo.

-y: coordenada y en la que se encuentra la esquina superior izquierda de la imagen del enemigo.

MAZMORRACONSUMIBLES

Se guarda la relación de los consumibles que estarán disponibles en una mazmorra en concreto. Estos objetos se pueden usar para superar dicha mazmorra o para ayudar al personaje a realizar una tarea.

-idMazmorra: se guarda el identificador único de la mazmorra.

-idConsumible: se guarda el identificador único del consumible/objeto.

-x: coordenada x en la que se encuentra la esquina superior izquierda de la imagen del consumible.

-y: coordenada y en la que se encuentra la esquina superior izquierda de la imagen del consumible.

COMPRADOS

En esta tabla guardaremos la relación entre los objetos (personajes, mapas, mazmorras, consumibles y enemigos) que un usuario puede adquirir en la tienda para la creación de mapas y los personajes para poder jugar a dichos mapas.

-idUsuario: identifica a un usuario en concreto.

-idObjeto: identifica al objeto que tiene en posesión el usuario.

-precio: se almacenará el precio por el que el usuario ha adquirido dicho objeto.

-tipo: indica qué tipo de objeto se ha comprado.

PARTIDA

En esta tabla guardaremos los datos dinámicos que se generen a la hora de jugar una partida.

-idPartida: identifica de manera única la partida en la base de datos.

-idUsuario: identificador del usuario que está jugando la partida.

-idMapa: identificador del mapa que se está jugando en esa partida.

-idPersonaje: identificador del personaje que se está usando en la partida.

-fechaComienzo: fecha en la que se inicia la partida.

-fechaUltimoAcceso: ultima fecha en la que se accede a la partida.

INVENTARIO_PARTIDA

En esta tabla guardaremos la relación que existiría entre una partida en concreto y un objeto (consumible y enemigo) que se deben tener en cuenta en la partida.

-id: es un identificador único para cada inventario.

-idObjeto: guardaremos el id del objeto relacionado con este inventario y una partida en concreto.

-categoría: como podemos tener tanto consumibles como enemigos, se diferenciara entre unos y otros para un mejor manejo y rendimiento de consultas.

-usado_superado: se indicará con un verdadero o falso si el objeto en concreto ya se ha usado y no estaría en el inventario del personaje o si un monstruo ha sido vencido.