



Proyecto Fin De Ciclo.

Slay the DAW

Descripción breve

Proyecto de un juego de construcción de mazos.
Tutor de proyecto Bautista Ramírez.

Diego Navas Alcobendas

Indice.

1. Introducción

- 1.a Presentación del proyecto.
- 1.b Objetivos del proyecto.
- 1.c Justificación del proyecto.

2. Análisis de requerimientos.

- 2.a Identificación de necesidades y requerimientos.
- 2.b Identificación de público.
- 2.c Estudio de mercado y competencia.

3. Diseño y planificación.

- 3.a Definición de la arquitectura del proyecto.
- 3.b Diseño de la interfaz de usuario.
- 3.c Planificación de las tareas y los recursos necesarios.

4. Implementación y pruebas

- 4.a Desarrollo de las funcionalidades del proyecto.
- 4.b Pruebas unitarias y de integración.

5. Documentación

- 5.a Documentación técnica.
- 5.b Documentación de usuario.
- 5.c Manual de instalación y configuración.

6. Mantenimiento y evolución.

- 6.a Plan de mantenimiento y soporte.
- 6.b Identificación de posibles mejoras y evolución del proyecto.
- 6.c Actualizaciones y mejoras futuras.

7. Conclusiones

- 7.a Evaluación del proyecto.
- 7.b Cumplimiento de objetivos y requisitos.
- 7.c Lecciones aprendidas y recomendaciones para futuros proyectos.

1. Introducción

1.a Presentación del proyecto.

El concepto del proyecto consiste en un juego de cartas en el cual el jugador juega varias partidas empezando con una serie de cartas que el mismo elige y obteniendo 1 aleatoria cada vez que gana un combate. El usuario obtendría la moneda del juego (psionita) al derrotar a jefes o pasarse una partida con éxito al completar 20 combates. Esta moneda permitirá al jugador comprar nuevas cartas con las que poder jugar o mejorar otras cosas como su vida máxima o el número de cartas con el que empieza cada partida.

1.b Objetivos del proyecto.

El proyecto busca crear un juego divertido de construcción de mazos al cual se puedan echar varias partidas muy rápido buscando una experiencia más casual que otros juegos del género. Además de esto se busca cumplir con todos los requisitos del proyecto y sobre todo mantener una experiencia muy fluida por parte del cliente que una vez se haya registrado podrá hacer todo con muy pocos clicks y abandonar las partidas en cualquier momento para luego continuar donde se abandonó.

1.c Justificación del proyecto.

Creo que como proyecto se ve justificado por la falta de juegos de este estilo que no sean slay the spire al cual intento imitar en algunas cosas, pero centrándome más en una experiencia rápida y casual dirigida a partidas muy cortas de unos 10 o 15 minutos.

2. Análisis de requerimientos.

2.a Identificación de necesidades y requerimientos.

El juego en si tan solo necesita de conexión a internet y el crear una cuenta. Se busca el crear un juego que todo el mundo pueda jugar en cualquier dispositivo. El proyecto además debe cumplir con todos los requisitos del proyecto. Por tanto, tiene una página llamada Sobre Mi con mi curriculum, un formulario de registro con validación en Ajax, un crud de usuarios donde se les puede hacer administradores y además la tabla de usuarios tiene fechas.

Por tanto, los requerimientos para el juego serian:

Debe tener un login y un registro de usuario validados por AJAX.

Cada usuario debe tener sus propias cartas, barajas, mazos, tiendas y personajes

Cada usuario debe poder ver en todo momento sus cartas ya desbloqueadas.

Cada usuario debe poder ver en todo momento las estadísticas de su personaje.

Cada usuario debe poder editar el contenido de sus mazos.

Cada usuario debe poder comprar nuevas cartas y mejoras para el personaje.

Se necesita un apartado "Sobre Mi" que contenga un currículum y una carta de presentación.

Se necesita un crud de manejo de usuarios para los administradores.

Se necesita una vista de combates que cargue con enemigos aleatorios dependiendo de la cantidad de partidas ya jugadas y con las cartas del mazo seleccionado por el usuario.

Se necesita una pantalla para elegir una de 3 cartas aleatorias al ganar un combate y un método que seleccione esas tres cartas aleatorias.

Se necesita una pantalla de victoria y otra de derrota.

2.b Identificación de público.

El público objetivo de este proyecto son personas a las que les gustan los juegos de construcción de mazos y que buscan una experiencia con toques roguelike y casual.

2.c Estudio de mercado y competencia.

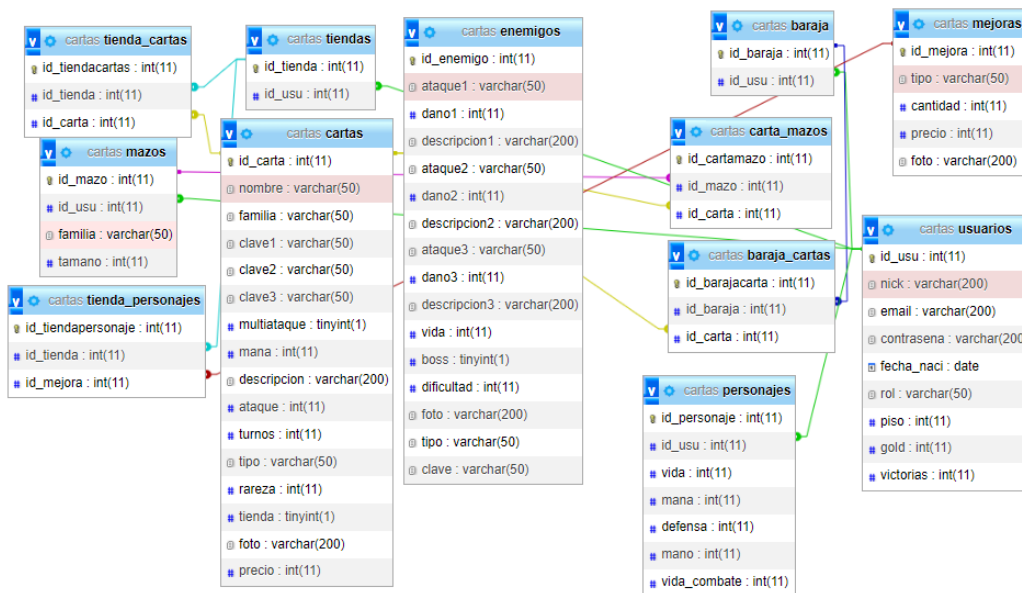
En el mercado actualmente existen muy pocos juegos de este estilo, la mayoría de los juegos de cartas están más orientados a la experiencia clásica multijugador donde compras cartas con dinero real y combates contra otras personas. En cuanto a juegos singleplayer como el proyecto, el principal es el famoso slay the spire pero ya está un poco viejo y todo el mundo al que le gustan este tipo de juegos ya lo ha jugado hasta el aburrimiento. Para diferenciarse de la competencia este juego busca ser una experiencia más rápida y casual centrándose únicamente en los combates y teniendo como objetivo el llenar pequeños ratos muertos.

3. Diseño y planificación.

3.a Definición de la arquitectura del proyecto.

Para el desarrollo del proyecto se utilizara una base de datos MySQL, PHP para el apartado del servidor(lo referente a consulta y modificación de datos en la base de datos) y JavaScript, HTML y CSS para las vistas del cliente(con el fin de dar sensación de fluidez):

El proyecto se compone de la siguiente base de datos de 12 tablas.

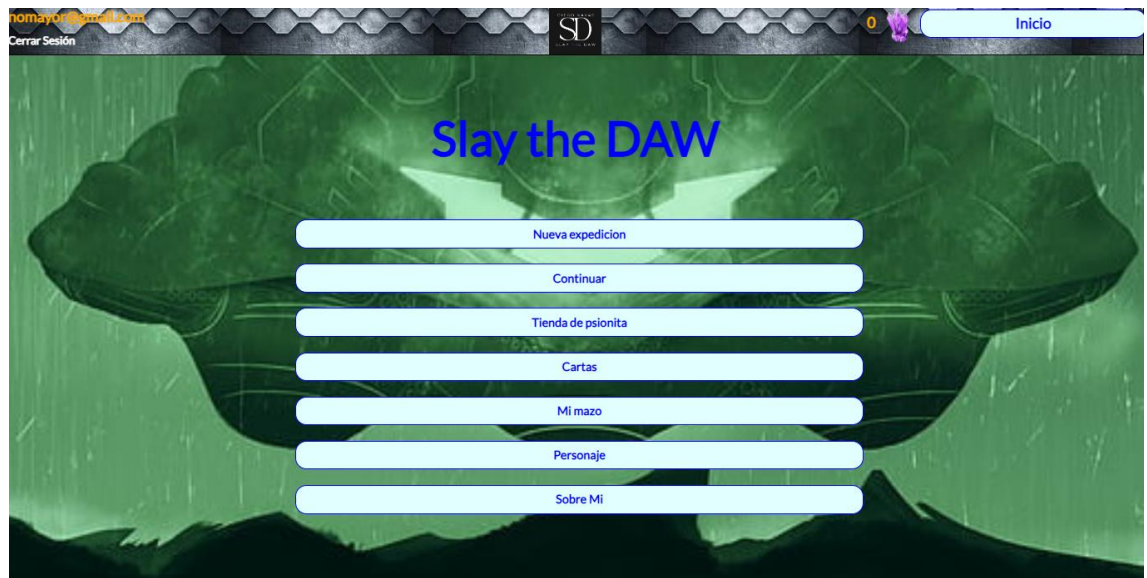


Después se compone de dos controladores (UsuariosControlador y CombatesControlador) que manejan lo referente a los usuarios (Su registro, login, que cartas tienen etc..) y a los combates respectivamente. Como se puede deducir por esto el proyecto usa MVC.

3.b Diseño de la interfaz de usuario.

La interfaz del usuario se compone normalmente (menos en los combates) de una barra de navegación superior que contiene el logo de la aplicación y que además permite a los usuarios cerrar sesión, saber su dinero y volver al inicio en todo momento. Además de esto la gama de colores se compone de azul para el texto y botones, tonos verdosos y negros para el fondo y utiliza la fuente Lato.

Como ejemplo este es el menú principal:



El único que es distinto es el menú de combate que se vería así:



3.c Planificación de las tareas y los recursos necesarios.

El proyecto debe cumplir las siguientes condiciones y tareas.

- Debe tener un login y un registro de usuario validados por AJAX.
 - Se necesitará una tabla de la base de datos para usuarios que tenga un campo fecha de nacimiento.
- Cada usuario debe tener sus propias cartas, barajas, mazos, tiendas y personajes.
 - Se necesitarán las siguientes cartas en la base de datos: Baraja, baraja_cartas, cartas, mazo, carta_mazo, tienda, tienda_carta y personaje.
- Cada usuario debe poder ver en todo momento sus cartas ya desbloqueadas.
 - Se necesitará una vista dedicada y un método para cargar sus datos.
- Cada usuario debe poder ver en todo momento las estadísticas de su personaje.
 - Se necesitará una vista dedicada y un método para cargar sus datos.
- Cada usuario debe poder editar el contenido de sus mazos.
 - Se necesitará una vista dedicada y un método para cargar sus datos.
 - Además, también se necesitará una vista previa para elegir los mazos.
- Cada usuario debe poder comprar nuevas cartas y mejoras para el personaje.
 - Se necesitará una vista dedicada a la tienda un método para cargarla, otro para comprar y las tablas de mejora y tienda_personajes.
- Se necesita un apartado "Sobre Mi" que contenga un currículum y una carta de presentación.
- Se necesita un crud de manejo de usuarios para los administradores.
 - Requerirá una vista dedicada un método para cargarla y otro para hacer administradores.
- Se necesita una vista de combates que cargue con enemigos aleatorios dependiendo de la cantidad de partidas ya jugadas y con las cartas del mazo seleccionado por el usuario.
 - Se requerirá una vista previa para elegir el mazo, un método que cargue el resultado de esa vista en el mazo de combate, una tabla de enemigos en la base de datos que serán cargados de forma aleatoria por el método.

- Se necesita un método que al terminar un combate actualice la vida del personaje y guarde los datos para que el usuario pueda salir en cualquier momento.
- Se necesita una pantalla para elegir una de 3 cartas aleatorias al ganar un combate y un método que seleccione esas tres cartas aleatorias.
- Se necesita una pantalla de victoria y otra de derrota.

4. Implementación y pruebas

4.a Desarrollo de las funcionalidades del proyecto.

- Debe tener un login y un registro de usuario validados por AJAX.
 - Se implementa el login en la vista inicio.php y su comprobación se realiza en el método login de UsuariosControlador.
 - El registro se implementa en la vista registro.php y su comprobación se realiza con el método comprobarRegistro de UsuariosControlador.
- Cada usuario debe poder ver en todo momento sus cartas ya desbloqueadas.
 - Se implementa en la vista Cartas.php y utiliza el método verCartas de UsuariosControlador para obtener las cartas del usuario.
- Cada usuario debe poder ver en todo momento las estadísticas de su personaje.
 - Se implementa en la vista Personaje.php y utiliza el método verPersonaje de UsuariosControlador.
- Cada usuario debe poder editar el contenido de sus mazos. Además, también se necesitará una vista previa para elegir los mazos.
 - Se implementa la vista previa con la vista SeleccionMazo.php y con el método selecMazo de UsuariosControlador.
 - La edición del contenido de los mazos se implementa en la vista Mazos.php y usa el método guardarMazo de UsuariosControlador para hacer las modificaciones a la base de datos.
- Cada usuario debe poder comprar nuevas cartas y mejoras para el personaje.
 - Se implementa en la vista Tienda.php y utiliza el método verTienda de UsuariosControlador.
- Se necesita un apartado "Sobre Mi" que contenga un currículum y una carta de presentación.
 - Se implementa en la vista SobreMi.php.
- Se necesita un crud de manejo de usuarios para los administradores.

-Se implementa en la vista `administraUsuarios.php` y utiliza los métodos `adminUsu` y `hacerAdmin` de `UsuariosControlador`.

-Se necesita una vista de combates que cargue con enemigos aleatorios dependiendo de la cantidad de partidas ya jugadas y con las cartas del mazo seleccionado por el usuario.

-Se implementa con la vista `Combate.php` y utiliza todos los métodos del controlador `CombateControlador`.

-Se necesita un método que al terminar un combate actualice la vida del personaje y guarde los datos para que el usuario pueda salir en cualquier momento.

-Se hace con el método `finalizarCombate` de `CombateControlador`.

-Se necesita una pantalla para elegir una de 3 cartas aleatorias al ganar un combate y un método que seleccione esas tres cartas aleatorias.

-Se implementa en la vista `SeleccionMazoVictoria.php` y usa el método `cartaVictoria` de `CombateControlador`

-Se necesita una pantalla de victoria y otra de derrota.

-Se implementa en las vistas `Victoria.php` y `Derrota.php` respectivamente.

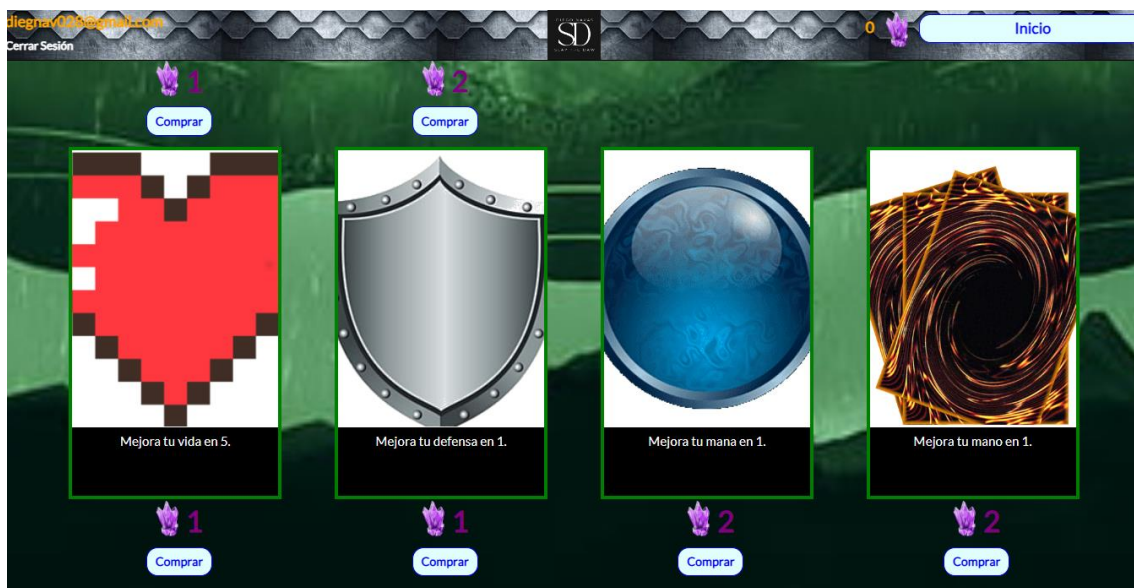
4.b Pruebas unitarias y de integración.

A modo de demostracion voy a realizar dos pruebas del correcto funcionamiento de la aplicación. Voy a comprar una mejora y ver como se suma al valor del personaje y voy a seleccionar unas cartas determinadas para hacer un mazo.

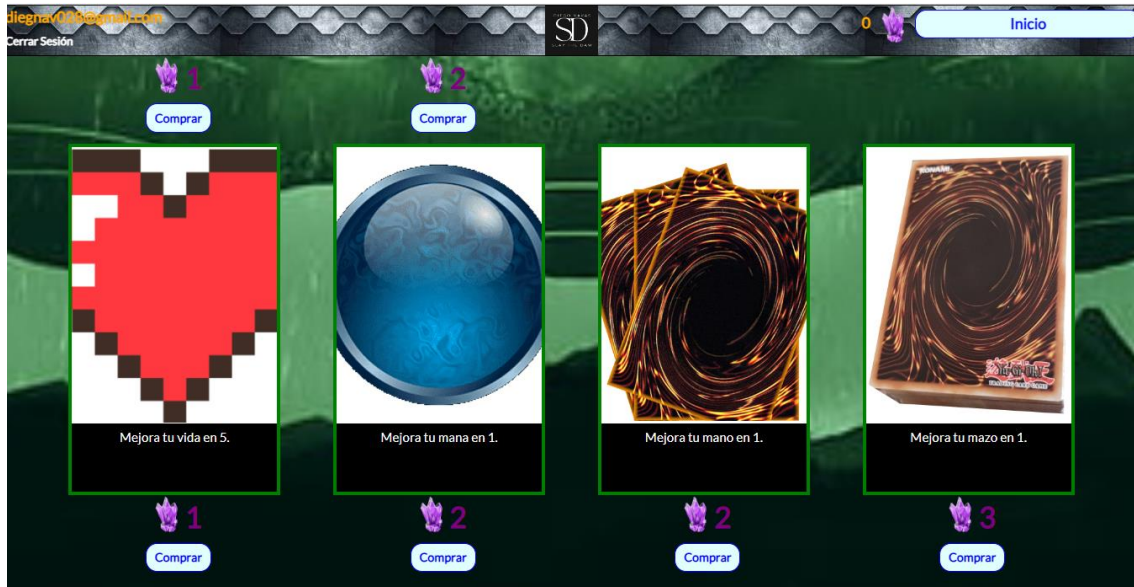
Comprar Mejora:



Aunque este usuario no tiene dinero como se puede observar en la parte superior derecha podrá comprar igualmente pues es administrador.



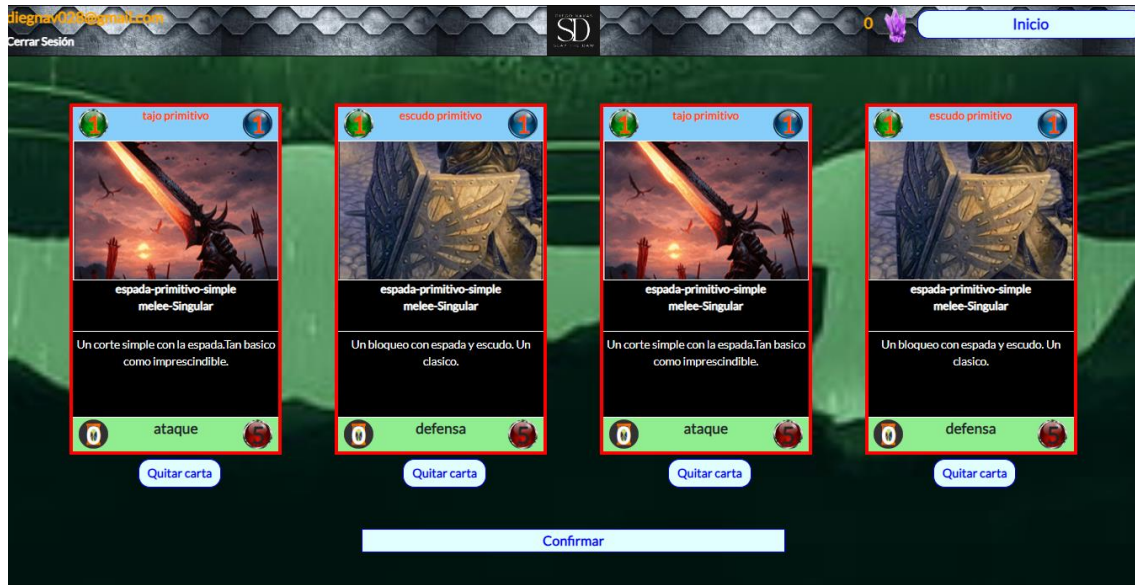
A continuación comprare la mejora de defensa



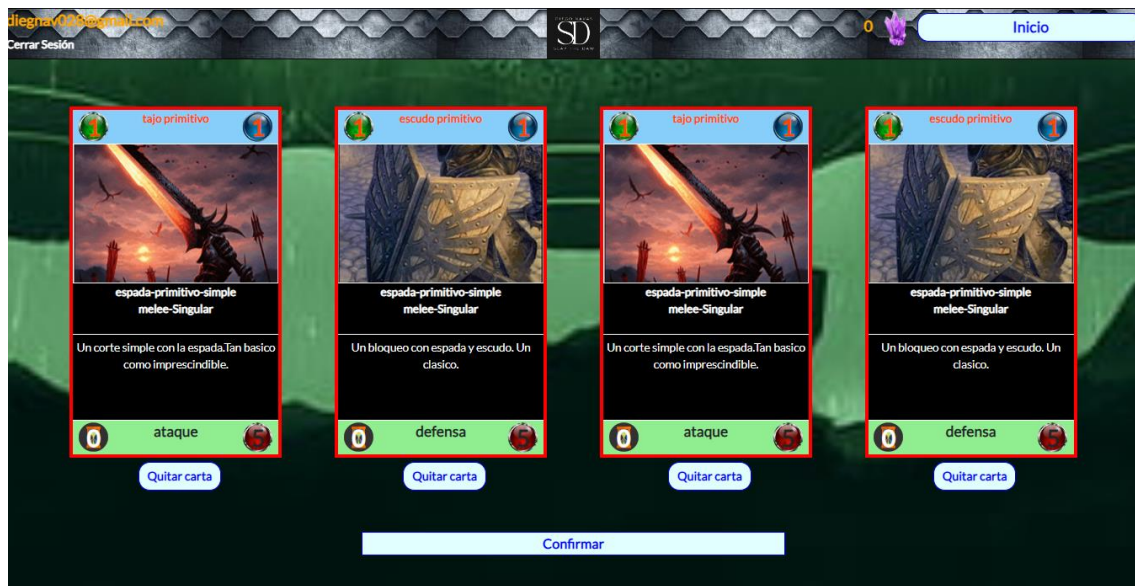
Como se puede ver funciona correctamente la mejora desaparece de la tienda y su valor es sumado al personaje.

Por último, vamos a hacer una comprobación de la selección de mazos:

Nos vamos a Mi mazo y creamos un mazo. El siguiente, por ejemplo:



Ahora si le damos a confirmar y volvemos a entrar el mazo debería ser el mismo.



Como se puede ver funciona correctamente.

5. Documentación

5.a Documentación técnica.

El proyecto esta desarrollado usando PHP para el servidor y JavaScript para el cliente con una base de datos MySQL.

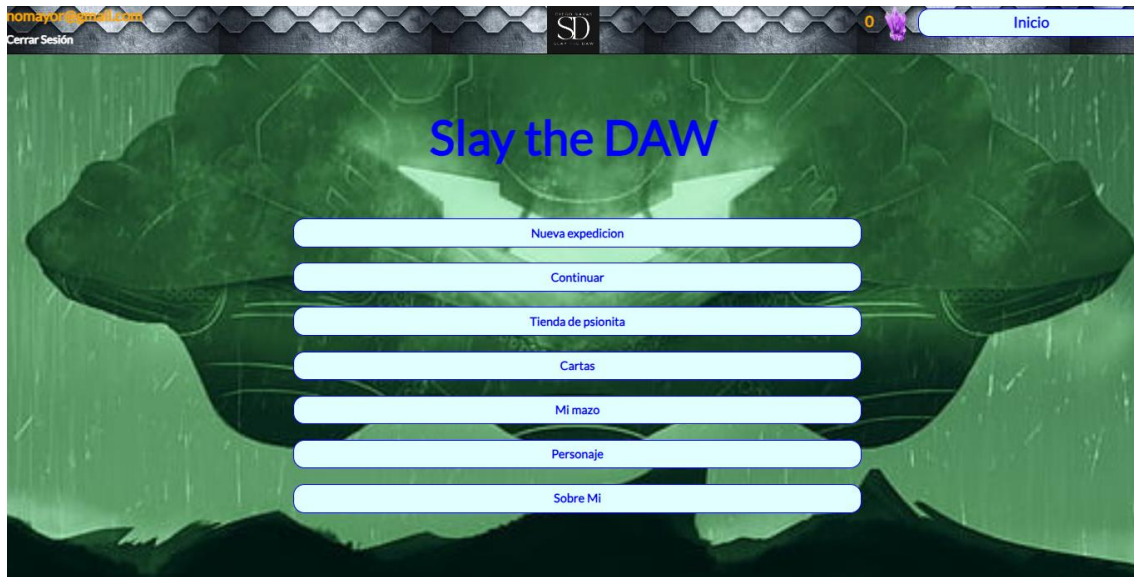
En la parte del servidor cuenta con dos controladores que manejan todo lo referente a la base de datos, además de las clases Baraja, Carta, CartaDAO, Conexión, Enemigo, EnemigoDAO, Mazo, MazoDAO, Mejora, MejoraDAO, Personaje, PersonajeDAO, Tienda, Usuario y UsuarioDAO que contienen la definición de los objetos y los métodos para obtenerlos y modificarlos en la base de datos.

Para la parte visual se tienen las paginas Cartas(que muestra las cartas desbloqueadas), Combate(que muestra los combates y contiene el javascript para hacerlos funcionar), Derrota(es la pantalla de derrota), Mazos(que muestra el contenido de un mazo y permite modificarlos), SeleccionMazo(que permite seleccionar el mazo a modificar en Mazos), SeleccionMazoCombate(que permite elegir con que mazo queremos jugar), SeleccionMazoVictoria(que nos permite elegir una carta entre 3 aleatorias), SobreMi (que contiene un curriculum), Tienda(que permite comprar nuevas cartas y mejoras),Victoria(es la página de victoria), administraUsuarios(es el CRUD de usuarios), inicio(que contiene el menú principal y el login), personaje(que muestra las estadísticas del personaje), plantilla(que contiene el menú superior de navegación y el css de toda la página) y registro(que contiene el formulario de registro).

Por último, tenemos index.php que contiene un mapa para enlazar los controladores y las funciones (Para el diseño MVC).

5.b Documentación de usuario.

Nada más iniciar la aplicación el usuario observara una pantalla de login que le pedirá que meta su email y contraseña o que si no está registrado que se registre. Si el usuario se va a registrar vera un formulario que le pedirá un Nick, un email, una contraseña y una fecha de nacimiento (a función de validar la fecha se pide que seas mayor de 18). Una vez registrado el usuario hace login y vera el menú principal.



Si se es admin también se verá una opción para manejar usuarios. Desde este momento el usuario ya podrá empezar a jugar (pues en el registro se asignan unas cartas por defecto) pero es bastante recomendable antes de eso abrir la pantalla de personaje para ver las estadísticas y luego darle a la pantalla de mi mazo y elegir las cartas con las que va a jugar. Como aun no tiene dinero la tienda es innecesaria así que vamos a inicializar un combate para ver cómo funcionan.



En la parte superior izquierda en la esfera azul podemos ver nuestro mana (el mana que regeneras por turno y tu mana actual). El mana es lo que nos cuesta el usar una carta. También tenemos un escudo que se regenera por turno para absorber algo de daño y nuestra vida que es permanente (a menos que se use una carta de curación). También podemos ver a nuestros enemigos a la derecha con su vida encima. Ellos actuarán al final de turno (si fuera un jefe estaría solo, pero atacaría 2 veces). Por último, tenemos tres iconos de cartas que son de izquierda a derecha nuestro cementerio, nuestra mano y nuestro mazo. El cementerio estará vacío a menos que usemos cartas instantáneas que solo se pueden usar una vez por combate y sirve únicamente para ver las instantáneas que hemos usado. La mano son las cartas que tenemos para jugar ese turno, al principio de cada turno robaremos del mazo cartas equivalentes a la estadística mano de nuestro personaje (o todas las que halla en el mazo si son menos que la mano). El mazo son las cartas que tenemos disponibles para robar al siguiente turno. Ahora veamos una carta:



En la esfera azul podemos ver su coste en mana. También tenemos una serie de palabras entre medias que nos indican por orden a que tipo pertenecen dentro de su familia (espadas, hachas o lanzas en melee, por ejemplo) esto sirve para cartas que aumentan el daño de tipos determinados. Después una carta puede ser primitivo, complejo o archeo y esto es común a todas las cartas y sirve para mejoras generales entre las mismas. Después una carta puede ser simple, esotéricas o alién. Las cartas simples no hacen nada especial pero las esotéricas aumentan tu defensa por turno en uno por tres turnos y las alién también aumentan tu ataque. A continuación, podemos ver si una carta es multiataque o singular (si ataca a todos o solo a un enemigo). Si es singular habrá que clicar a que enemigo queremos atacar después de darle a jugar carta. Por último, tenemos un reloj de arena que nos indica cuantos turnos durara el efecto y una esfera roja que nos dice por cuanto atacaremos, protegeremos, curaremos etc. La esfera verde es la rareza de la carta y sirve para cuando estamos haciendo mazos pues limita cuantas copias de la misma carta se pueden añadir, a menor rareza mas cartas (a 1 de rareza 5 y a 5 de rareza 1). Ahora bien, esto es todo lo que se necesita para poder jugar. Una vez hayamos matado a los enemigos se nos llevara a una pantalla donde podemos elegir entre 3 cartas para añadir a nuestro mazo para el siguiente combate. Las partidas terminan después de 20 combates con un jefe cada 5.

5.c Manual de instalación y configuración.

Para instalar el proyecto basta con descargarlo de github e inicializar la base de datos(debe llamarse "cartas"). Después de eso ya se puede acceder al juego y habría que registrarse. Para poder comprobar la funcionalidad de los administradores ya existe de forma previa el usuario adminsup de email adminsup@gmail.com y contraseña 1234. Lo único que distingue a un administrador de un usuario normal es que puede comprar cartas en la tienda sin pagar (para poder probarlas todas).

6. Mantenimiento y evolución.

6.a Plan de mantenimiento y soporte.

El juego se seguirá actualizando añadiendo nuevas cartas, mejoras y enemigos cada mes. También se crearán nuevos tipos de cartas, como cartas que le quiten daño a los enemigos o cartas que permitan robar del cementerio.

6.b Identificación de posibles mejoras y evolución del proyecto.

Los mayores fallos del juego son visuales en concreto el hecho de que las cartas y enemigos carecen de una temática y estilo de dibujo unificados. Por tanto, a futuro el objetivo serio crear el arte yo mismo para solucionar esto.

El otro problema del proyecto es que, aunque es divertido, se tarda muy poco en quemarlo pues tan solo tiene una mecánica principal. Para solucionar esto a futuro la idea sería crear una especie de mapa donde se puedan elegir los eventos y crear eventos que no sean de combate, como permitirte curarte o mejorar una carta para solo esa partida.

6.c Actualizaciones y mejoras futuras.

La próxima gran actualización incluirá un selector de dificultad (a más dificultad, más dinero) y la habilidad de ganar mejoras a las estadísticas de forma temporal tras matar un jefe. Aparte de eso también se creará un mapa para elegir eventos mediante grafos.

7. Conclusiones

7.a Evaluación del proyecto.

Creo que el proyecto cumple con todos los requisitos y además es bastante innovador y divertido. Aun así, sufre un poco en el apartado visual y su duración es bastante corta (alrededor de una semana de juego casual).

7.b Cumplimiento de objetivos y requisitos.

Tiene una página llamada Sobre Mi con mi curriculum, un formulario de registro con validación en Ajax, un crud de usuarios donde se les puede hacer administradores, la tabla de usuarios tiene fechas, las contraseñas están encriptadas y las consultas a la base de datos con claves foráneas se realizan en una sola consulta. Por tanto, el proyecto cumple con todos los requisitos del proyecto. Por otro lado, todos los objetivos funcionales también han sido cumplidos.

7.c Lecciones aprendidas y recomendaciones para futuros proyectos.

La gran lección que he aprendido en este proyecto es la importancia del css y la apariencia en un proyecto pues tuve que rehacer gran parte del mismo hacia la mitad del transcurso del tiempo asignado debido a que no podría mostrar toda la información y opciones de una forma adecuada así que tuve que crear nuevas vistas y métodos para poder hacerlo. Por eso como recomendación para futuros proyectos recomiendo diseñar una carcasa de la apariencia del proyecto antes de empezar pues es posible que luego no puedas mostrar todo lo que quieras o que lo hagas de una forma que parece rara.