



**Universidad
Europea** MADRID

PROYECTO INTEGRADOR

1DAW

EL GRUPO



Héctor Augusto Chango Tapia

Manuel Gómez Navarro

Gonzalo Pozo Sánchez

Daniel Simón del Olmo

Índice

| | |
|--|----|
| Resumen..... | 3 |
| 1. Introducción | 4 |
| 2. Objetivos | 6 |
| 3. Tecnologías utilizadas..... | 9 |
| 4. Desarrollo e implementación..... | 11 |
| ESPECIFICACIONES | 11 |
| DIAGRAMA ENTIDAD – RELACIÓN | 13 |
| Modelo Relacional..... | 15 |
| Normalización del Modelo Relacional | 16 |
| Diagrama de Casos de Uso | 16 |
| Diagrama de Clases | 17 |
| Pruebas con Junit | 18 |
| Base de Datos..... | 19 |
| Diseño de la Interfaz..... | 20 |
| Creación del Proyecto y Vistas | 20 |
| Diseño del Logo | 21 |
| 5. Metodología | 21 |
| 1. CRONOGRAMA DE ACTIVIDADES SPRINT 1 ➔ 18/03/2024 – 31/03/2024..... | 27 |
| 2. CRONOGRAMA DE ACTIVIDADES SPRINT 2 ➔ 01/04/2024 – 14/04/2024..... | 28 |
| 3. CRONOGRAMA DE ACTIVIDADES SPRINT 3 ➔ 15/04/2024 – 05/05/2024..... | 29 |
| 4. CRONOGRAMA DE ACTIVIDADES SPRINT 3 ➔ 05/05/2024 – 19/05/2024..... | 30 |
| 6. Resultados y conclusiones..... | 31 |
| 7. Trabajos futuros | 32 |

| | |
|--|--------------------------------------|
| Anexos | 34 |
| Anexo I – Listado de requisitos de la aplicación | 34 |
| Anexo II – Guía de uso de la aplicación | 35 |
| Anexo III..... | ¡Error! Marcador no definido. |

Resumen

Palabras clave: Black Temple; aplicación; gestor; GM; jugador.

La aplicación de Black Temple, es un gestor de partidas de un juego de rol que permite a los usuarios registrarse y almacenar sus cuentas en una base de datos. Una vez creada la cuenta, los usuarios pueden iniciar sesión como jugadores o como Game Masters(GM). Los jugadores pueden crear personajes eligiendo su sexo, raza y clase y después podrán unirse a una partida. Los GM, por su parte, tienen la capacidad de administrar y supervisar las partidas, lo que incluye la modificación de atributos y otros aspectos del juego para los personajes de los jugadores, además, los GM pueden diseñar y dirigir la trama del juego, asegurando una experiencia dinámica y envolvente para los jugadores. Esta funcionalidad permite a los GM controlar y dirigir la narrativa y mecánica del juego de manera efectiva, mientras que los jugadores pueden sumergirse en el rol de sus personajes y participar en la historia del juego. La aplicación no solo facilita la creación de personajes y la gestión de partidas, sino que también mejora la interacción y colaboración entre jugadores y GM. Con una interfaz intuitiva y funcionalidad robustas, Black Temple se convierte en una plataforma esencial para cualquier entusiasta de los juegos de rol, ofreciendo una experiencia de juego como nunca se había visto.

1. Introducción

El proyecto se centra en el desarrollo de una aplicación con la que se va a gestionar la creación de los personajes de un juego de rol, pudiendo crear desde el nombre del personaje hasta su sexo, clase y raza.

En primer lugar, el usuario accederá a una pantalla de “Login” donde, si es la primera vez que visita la aplicación, podrá registrarse por medio de una serie de vistas donde deberá rellenar varios campos con sus datos. Una vez rellenados todos los campos, cuando el usuario haga clic en “Crear cuenta”, dicha información será almacenada en la base de datos de “Miembros”. Gracias a esta base de datos, se podrá acceder con normalidad a la aplicación por medio del “Login”.

Una vez dentro, aparecerá una ventana de selección de rol, donde el usuario deberá escoger entre entrar a la aplicación como Game Master, para poder gestionar una partida de rol, o como jugador, para crear un personaje o empezar a jugar con uno ya creado.

Si el usuario escoge acceder como Game Master, deberá elegir entre continuar con una partida ya existente, si es que ya la había creado, o si crear una nueva partida. Si desea crear una partida nueva, le redirigirá a otra ventana donde ingresará el nombre de la partida y escogerá la ambientación que desea. Por otro lado, si se desea continuar con una partida ya creada, al seleccionar “Continuar”, se abrirá una nueva ventana donde se mostrarán las partidas disponibles. Cuando se elija una de las partidas y se presione en “Entrar a partida”, se abrirá la pantalla de gestión de personajes de esa partida.

Si el usuario desea ingresar como jugador, se mostrará una pantalla donde aparecerá el primer personaje creado por el usuario seleccionado por defecto. En dicha pantalla, el usuario podrá moverse entre sus personajes creados y presionar el botón “Ver partidas” para ver en qué partida está jugando cada uno y qué estadísticas tiene. Además, podrá crear nuevos personajes, seleccionando su nombre, género, raza y clase, y serán añadidos a la base de datos de los Personajes de ese miembro en cuestión.

Cuando el usuario seleccione el personaje con el que desea jugar, podrá presionar en dicho botón y aparecerá una tabla con el nombre de las partidas disponibles, con su ambientación, nivel medio y un mensaje que indicará al jugador si ese personaje se encuentra ya en esa partida o no. Además de esto, hay un botón “Detalles” que, al pulsarlo, muestra la

información del personaje para la partida seleccionada, es decir, su nivel, raza, clase, y estadísticas. Una vez seleccionada la partida, el usuario deberá hacer clic en “Unirse”, esto hará que, si el personaje ya se encontraba en la partida ingrese a esta directamente y sino, automáticamente se meterá al personaje en la base de datos de esa partida e ingresará igualmente.

Para la realización del proyecto se utilizará Java y SQLite, y metodologías ágiles para gestionar eficazmente el desarrollo del software, como Kanban y SCRUM.

Dicho proyecto se basa en la implementación de las distintas asignaturas de este curso, combinando los conocimientos adquiridos en Programación, Bases de Datos y Entornos de Desarrollo. La aplicación ha sido creada con lenguaje de programación Java, en Eclipse, y apoyándose también en el uso de Visual Studio Code para el trabajo en equipo. Dicho proyecto está organizado con el modelo-vista-controlador.

En el paquete model se encuentran las clases de personaje, miembro y partida. Además de estas tres, también se encuentra dentro de este paquete la clase AccesoBD, la cual será la encargada de los métodos de acceso a la base de datos.

En el paquete views se encuentra la descripción general de las vistas, que son las responsables de la representación visual de los datos de la aplicación. Las vistas son las encargadas de la interacción con el usuario.

El paquete control funciona como intermediario entre las vistas y el modelo y contiene los listeners utilizados para poder navegar entre ventanas, así como funcionalidades añadidas en botones o el acceso, ingreso o modificación de información en la base de datos.

El paquete main contiene el código necesario con las vistas y algunos de los métodos que permiten ejecutar la aplicación.

Por último, el paquete components tiene una clase que contiene el código con una de las funcionalidades del programa.

2. Objetivos

Bases de Datos

- **Objetivos Generales:**

1. Comprender la importancia de las bases de datos en el almacenamiento y gestión de información.
 - Asegurar que los datos de usuarios, personajes y partidas se almacenan de manera segura y organizada.
2. Aprender a diseñar, crear y gestionar bases de datos relacionales.
 - Establecer una estructura de base de datos eficiente que soporte las operaciones de creación, lectura, actualización y eliminación de información relacionada con las partidas.
3. Aplicar técnicas de modelado de datos y normalización.
 - Optimizar la base de datos para evitar redundancias y asegurar la integridad de los datos.

- **Objetivos Específicos:**

1. Identificar los componentes y arquitecturas de un SGBD.
 - Conocer cómo interactuar con SQLite para gestionar los datos de la aplicación.
2. Diseñar esquemas de bases de datos utilizando modelos entidad-relación.
 - Crear diagramas de ER para planificar cómo se relacionarán usuarios, personajes y partidas.
3. Crear y gestionar bases de datos mediante SQLite.
 - a. Implementar la base de datos en SQLite para almacenar información de usuarios, personajes y partidas.
4. Implementar consultas, actualizaciones y eliminaciones de datos utilizando SQLite.
 - Escribir consultas SQL para manipular y recuperar datos, como listar personajes o verificar credenciales de Login.
5. Aplicar reglas de integridad y restricciones para asegurar la calidad de los datos.
 - Implementar restricciones de integridad para garantizar que los datos ingresados sean válidos y coherentes.

Programación

Objetivos Generales:

1. Desarrollar competencias en la lógica de programación y resolución de problemas.
 - Aplicar técnicas de programación para resolver problemas como el registro de usuarios y la gestión de personajes.
2. Aprender a programar en el lenguaje de programación Java.
 - Utilizar Java para desarrollar la lógica de la aplicación, incluyendo la interacción con la base de datos y la interfaz de usuario.

3. Comprender los conceptos básicos de programación y estructuras de control.
 - Usar bucles, condicionales y otros controles de flujo para manejar la lógica de la aplicación.

Objetivos Específicos:

1. Comprender los principios de programación estructurada y orientada a objetos.
 - Diseñar la aplicación utilizando clases y objetos para representar usuarios, personajes y partidas.
2. Desarrollar algoritmos para resolver problemas específicos.
 - Crear algoritmos para tareas como la validación de usuarios.
3. Implementar programas utilizando estructuras de control como bucles y condicionales.
 - Manejar la navegación entre diferentes vistas.
4. Utilizar estructuras de datos básicas como arrays y listas.
 - Almacenar y manipular colecciones de personajes o partidas dentro del programa.
5. Crear y manipular objetos y clases en un lenguaje de programación orientado a objetos.
 - Definir clases y sus interacciones.
6. Gestionar excepciones y errores en los programas.
 - Implementar manejo de excepciones para asegurar que la aplicación se comporte de manera robusta ante los errores.
7. Realizar pruebas y depuración de programas para asegurar su correcto funcionamiento.
 - Probar y depurar el código para asegurar que todas las funcionalidades trabajen correctamente y sin errores.

Entornos de Desarrollo

Objetivos Generales:

1. Desarrollar habilidades en la gestión de proyectos de desarrollo de software por medio de metodologías ágiles.
 - Planificar y ejecutar el proyecto de desarrollo de la aplicación utilizando metodologías ágiles como Scrum o Kanban.
2. Comprender el ciclo de vida del desarrollo de software y las metodologías asociadas.
 - Seguir el ciclo de vida de desarrollo de software desde la planificación hasta el “lanzamiento” de la aplicación.

Objetivos Específicos:

1. Identificar y utilizar entornos de desarrollo integrados (IDEs) para la programación.
 - Utilizar IDEs como Eclipse para desarrollar, depurar y ejecutar el código Java de la aplicación.
2. Gestionar proyectos de desarrollo utilizando sistemas de control de versiones como Git.
 - Utilizar Git para llevar control de versiones del código, facilitando el trabajo colaborativo y la gestión de cambios.
3. Integrar herramientas de prueba y depuración en el proceso de desarrollo.
 - Implementar pruebas unitarias y usar herramientas de depuración para asegurar la calidad y funcionalidad del código.

3. Tecnologías utilizadas

Durante el desarrollo de la aplicación Black Temple, se han utilizado las siguientes herramientas:

- Eclipse
- Visual Studio Code
- Extensiones de Visual Studio Code
- SQLite
- Trello
- Adobe Illustrator

Para el desarrollo de la aplicación, se ha utilizado el lenguaje de programación Java mediante el IDE Eclipse, que tiene una funcionalidad que permite el uso de Window Builder para el realizar el diseño de interfaces gráficas de usuario de una forma visual y sencilla. Además, tiene un sistema de depuración avanzada que permite la ejecución del código paso a paso o el análisis del flujo del programa.

A la hora de la construcción del proyecto, como entorno de desarrollo se usará Visual Studio Code junto al pack de extensiones de Java que habilitarán que Visual Studio Code se convierta un IDE muy completo para programar en este lenguaje de programación, para la realización del diagrama E/R se ha utilizado la extensión “Draw.io Integration” que ha permitido realizar el trabajo de una forma más sencilla desde el propio Visual Studio Code y por último se ha utilizado la extensión “Live Share” para poder trabajar todos al mismo tiempo y de forma conjunta.

Se decidió utilizar el programa Visual Studio Code porque permite trabajar de manera colaborativa y visualizar en tiempo real los cambios aplicados tanto en el diagrama E/R como en el proyecto en sí. Esta herramienta ha facilitado la participación de todos en cada etapa del proyecto en lugar de dividir el trabajo en partes. De esta manera, todos los miembros del grupo han contribuido y participado activamente en todos los procesos del proyecto, permitiendo compartir ideas y mantener al grupo al tanto de los avances de forma simultánea. Esta dinámica ha resultado beneficiosa para el aprendizaje, ya que ha permitido aprender de todas las fases del proyecto al mismo tiempo.

Se ha elegido SQLite como base de datos para la aplicación debido a sus numerosas ventajas y características, que se adecuaban a nuestra forma de trabajar.

En primer lugar, la integración ligera de SQLite facilita su incorporación directa en la aplicación sin necesidad de configurar un servidor de base de datos independiente. Esto es ideal para este proyecto, ya que se necesitaba una solución sencilla y eficiente que no requiera de configuraciones complejas.

Además, la independencia de servidor de SQLite permite que la base de datos se contenga en un solo archivo, simplificando su gestión y distribución. Esta característica es muy útil para Black Temple, ya que era necesaria una base de datos fácil de transportar y desplegar en diferentes entornos.

Otra de las razones claves era que no necesita configuración ni instalación, lo que permite conectarse a la base de datos de forma inmediata tras incluir la biblioteca en el proyecto, lo que ahorra tiempo y reduce problemas de una posible configuración.

La compatibilidad con SQL permite usar consultas SQL estándar para interactuar con la base de datos, facilitando la escritura de código y el manejo de datos.

Todo ello unido a la velocidad y eficiencia de SQLite han sido factores clave para elegirlo como la base de datos del proyecto.

Como software de control de versiones se decidió usar Git, ya que se integra a la perfección con GitHub. Git proporciona las herramientas para crear un repositorio, actualizar, hacer rollback (volver a una versión anterior del proyecto), trabajar de forma cooperativa y guardar los cambios en un servidor remoto (en este caso GitHub). Como aplicación para gestionar el repositorio de Git se ha utilizado Git Bash, ya que ofrece una forma rápida y efectiva para trabajar en Git permitiendo usar todos sus comandos al completo.

Para la realización de este proyecto se usará un repositorio en GitHub, que se actualizará a medida que vayamos avanzando en nuestro trabajo. Pulsa en la primera foto para dirigirte al repositorio del proyecto y pulsa en cada logo de GitHub para ir al perfil de cada integrante del proyecto.

Como herramienta de gestión de proyectos se decidió utilizar Trello ya que, en primer lugar, Trello ofrece una interfaz visual intuitiva basada en tableros, listas y tarjetas, que facilita la organización y seguimiento de tareas de manera clara y visual. Esta estructura permite que todos los miembros vean de un simple vistazo el estado del proyecto y las tareas pendientes.

Además, Trello es flexible y personalizable, permitiendo agregar etiquetas, fechas de vencimiento y checklist colaborando a tiempo real con el resto del equipo comentando tarjetas o adjuntando archivos o recibiendo notificaciones, lo que mejora la comunicación y coordinación.

Por otro lado, Adobe Illustrator, gracias a su precisión y calidad vectorial, ha permitido la creación del logo escalable en varios tamaños sin perder la calidad, esto es de vital importancia para que el logo se vea bien, aunque las ventanas cambien de tamaño.

Además, Illustrator ofrece la opción de exportar imágenes en png, conservando el estilo vectorial de la imagen.

Adobe ofrece una amplia gama de herramientas y funcionalidades avanzadas de diseño, desde formas básicas hasta complejos gráficos vectoriales, permitiendo un control total sobre cada aspecto del diseño, facilitando la creación de un logo único.

4. Desarrollo e implementación

ESPECIFICACIONES

Al iniciar el programa, se mostrará un “Login” para que cada miembro inicie sesión, dicho menú está formado por el logo en la parte central superior, la parte de abajo contiene dos campos de texto en el que se introduce el usuario y la contraseña. También contiene un botón para confirmar el inicio de sesión y otro para crear una cuenta en el caso de no tener una.

Si el usuario no tiene una cuenta, deberá hacer clic en el botón “Crear cuenta”, donde recorrerá una serie de 4 ventanas (que tendrán una estructura similar al login) para crearla. Primero, se dirigirá a una ventana que contendrá el logo en la parte superior, un título debajo del logo con el texto “Crear cuenta” y un subtítulo debajo con el texto “Introducir tu nombre”, además esta ventana contendrá tres campos de texto los cuales serán para el nombre, los apellidos y el nombre de usuario tras rellenar los campos de texto el usuario deberá clicar en un botón “Continuar” situado debajo de los campos de texto.

Tras realizar este primer paso, el usuario deberá introducir sus datos de estudiante, que son su número de expediente y los estudios que está cursando, siendo cada uno un campo de texto. Debajo de estos dos campos, estará situado el botón “Continuar” que llevará a la siguiente ventana. Una vez realizado este paso, el usuario deberá establecer su contraseña en una nueva ventana, que estará formada por dos nuevos campos de texto, el primero será para introducir la contraseña y el segundo para confirmarla. Debajo de estos dos campos, habrá un “checkbox” para aceptar los términos y condiciones de uso, seguido de un nuevo botón para crear la cuenta. Una vez creada la cuenta, se mostrará una ventana de

confirmación con un icono indicador de confirmación junto al nombre de la cuenta y un botón “Continuar” que, al pulsarlo, llevará al usuario a la pantalla principal de la aplicación con el usuario ya logeado.

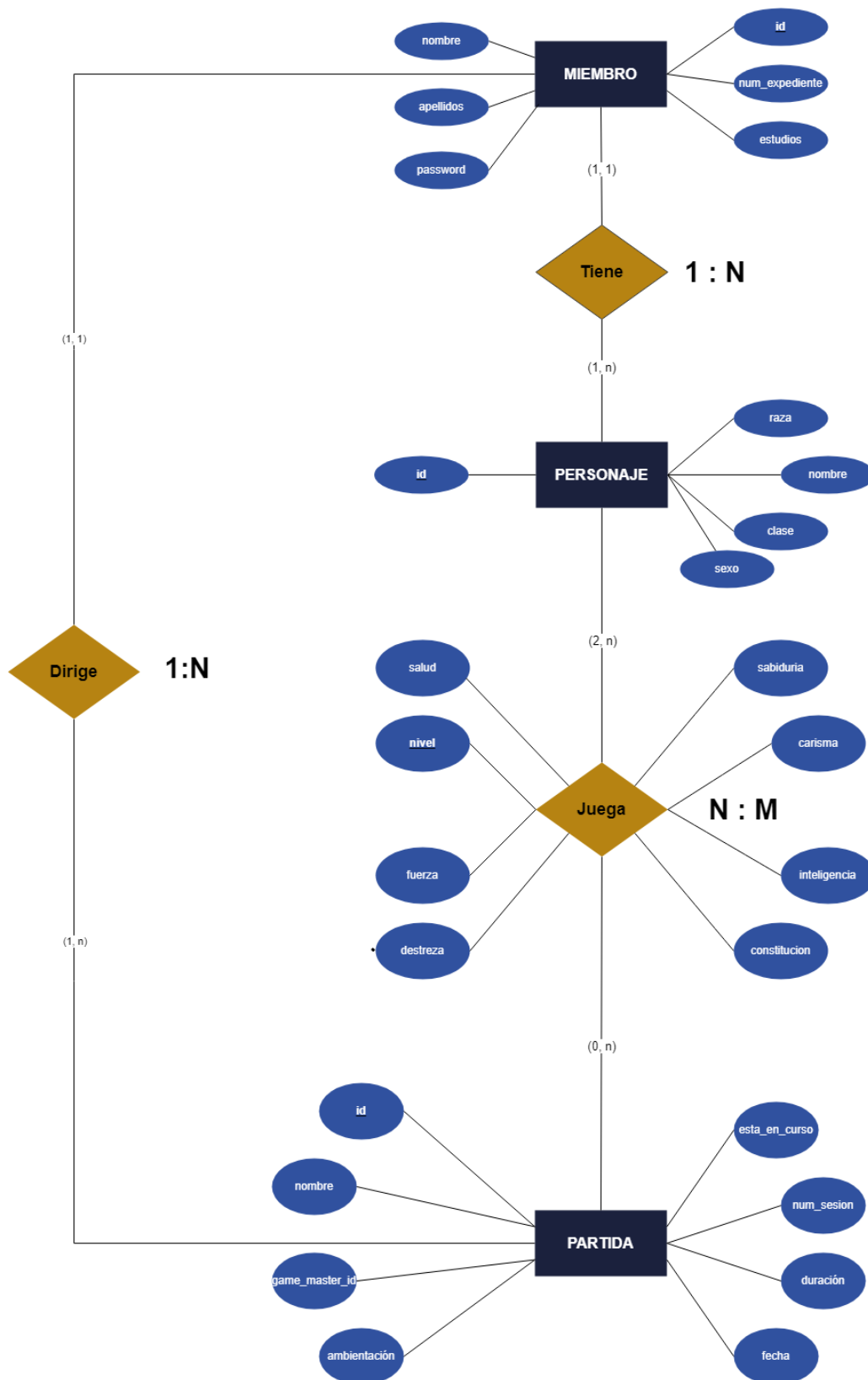
La estructura de la pantalla principal estará formada por el logo en la esquina superior izquierda y debajo del mismo un botón “Crear partida” (este botón estará deshabilitado en el caso de no tener ningún personaje), en el centro de la pantalla el usuario podrá visualizar su personaje seleccionado (si todavía no tiene creado ningún personaje, se mostrará el fondo de la pantalla principal), y debajo de este se mostrará su nombre y el botón “Jugar” (este botón estará deshabilitado en el caso de no tener ningún personaje y/o no tener ninguna partida). La parte derecha de la pantalla estará formada por una columna que contendrá los personajes creados por el usuario, siendo cada personaje un bloque que contendrá la clase, la raza o un icono de la raza junto al nombre del personaje. En la parte inferior de la columna se ubicará el botón “Crear nuevo personaje”.

Para la creación de personaje, primero es necesario saber que en el juego hay dos bandos, alianza y horda. Esto se verá reflejado en la pantalla de creación, ya que, a la izquierda, el usuario podrá seleccionar las razas de la alianza, y en la derecha, las de la horda. Estará formado por un bloque de botones con forma circular que nos mostrará el icono de cada raza. Cuando el usuario haga clic en la raza, se mostrará en la parte central el personaje y encima, dos botones donde se podrá seleccionar el género. Debajo del personaje, habrá un campo de texto donde deberá escribir el nombre de su personaje y, debajo de este, una serie de botones con un icono representativo de la clase que podrá elegir. Al pulsar el icono de cada clase, el personaje cambiará su apariencia según su clase (el guerrero con un hacha, el mago con un bastón...). Por último, se ubicará en la esquina inferior derecha un botón de "Crear personaje".

Cuando se pulse el botón de jugar, el usuario entrará a la partida y podrá gestionar la partida (ver sus estadísticas, ver las estadísticas de los demás personajes que están en esa partida, subir de nivel...). En el caso de que el personaje seleccionado esté en más de una partida, aparecerá una ventana emergente que mostrará las partidas en las que se encuentra dicho personaje junto al nivel.

Al pulsar el botón de crear partida, el usuario entrará en una ventana de creación de partida, que contendrá dos campos de textos donde insertará el nombre de la partida y la ambientación, además aquí se gestionarán los personajes que jugarán la partida. En la parte inferior derecha se ubicará un botón para la creación de partida, al pulsar este botón el usuario ingresará en la partida.

DIAGRAMA ENTIDAD – RELACIÓN



En primer lugar, hemos realizado el diagrama con 3 entidades, que son Miembro, Personaje y Partida.

La relación Miembro - Personaje tiene cardinalidad 1: N, ya que un miembro tiene como mínimo un personaje, porque sino no sería miembro, y como máximo N, ya que puede crear tantos personajes como quiera. Un personaje únicamente puede pertenecer a un miembro.

La relación Miembro – Partida tiene una cardinalidad de 1: N porque la partida es dirigida sólo por un miembro (que es el GameMaster), y al mismo tiempo el GM puede dirigir varias partidas.

La relación Personaje - Partida tiene una cardinalidad N: M porque el personaje puede jugar en varias partidas y a su vez, en cada partida hay varios personajes. Al ser una relación N: M, se crea una tabla intermedia, donde la clave primaria será la concatenación de partida_id, personaje_id y el nivel del personaje. Cada registro de esta tabla contendrá las estadísticas de los personajes de cada partida. Estos registros serán únicos de cada nivel y se conservarán dichos registros a medida que se avance en la partida, es decir, cada miembro podrá acceder y ver las estadísticas de su personaje a medida que vaya avanzando en la partida.

Modelo Relacional

TABLA MIEMBROS

| ID | NOMBRE | APELLIDOS | PASSWORD | NUM_EXPEDIENTE | ESTUDIOS |
|----|-----------|---------------------------|----------------|----------------|-------------|
| 1 | Gonzalo | Pozo Sánchez | lebron | 10001 | SMIR |
| 2 | Daniel | Simón del Olmo | cochecitos | 10002 | Superiores |
| 3 | Manuel | Gómez Navarro | 1234 | 10003 | Ed Infantil |
| 4 | Hector | Chango Tapia | 1111 | 10004 | Superiores |
| 5 | Rubén | Peña González | fernandoalonso | 10005 | SMIR |
| 6 | Cristiano | Ronaldo dos Santos Aveiro | siuuu | 10006 | Escuela |

| |
|---|
| Miembro (<u>ID</u> , Nombre, Apellidos, Num_Expediente, Estudios): |
| PK: id |

TABLA PERSONAJES

| ID | RAZA | NOMBRE | SEXO | CLASE | MIEMBRO_ID* |
|----|------------------|--------------------|--------|------------------------|-------------|
| 1 | Enano | Gimli | Hombre | Guerrero | 1 |
| 2 | Elfo de la noche | Atreides | Mujer | Bruja | 2 |
| 3 | Elfo de la noche | El Emisario | Hombre | Cazador de demonios | 3 |
| 4 | Tauren | Legolas | Hombre | Cazador | 4 |
| 5 | Humana | Padme | Mujer | Sacerdotisa | 5 |
| 6 | Humano | Anakin | Hombre | Paladin | 6 |
| 7 | No muerta | Ashoka | Mujer | Picara | 1 |
| 8 | Huargen | Darth Maul | Hombre | Mago | 2 |
| 9 | Pandaren | Joselito | Hombre | Chamán | 3 |
| 10 | Tauren | LeBron James | Hombre | Caballero de la muerte | 4 |
| 11 | Troll | Pepe el del Madrid | Hombre | Druida | 5 |
| 12 | Dracthyr | D'Angelo Russell | Hombre | Evocador | 6 |

| |
|--|
| Personaje (<u>ID</u> , Raza, Nombre, Clase, <i>Miembro_ID*</i>): |
| PK: id |
| FK: miembro_id → tabla miembro |

TABLA PARTIDAS

| ID | NOMBRE | GAME_MASTER_ID* | AMBIENTACIÓN | ESTA_EN_CURSO | NUM_SESIÓN | DURACIÓN | FECHA |
|----|---|-----------------|------------------------|---------------|------------|----------|------------|
| 1 | La batalla por el templo del Tigre Blanco | 1 | Estepas de Tong Long | No | 7 | 315 | 02/08/2023 |
| 2 | Liberación de los elfos | 2 | Suramar | Sí | 2 | 90 | 25/03/2024 |
| 3 | Lucha contra el Rey Exánime | 3 | Cementerio de Dragones | Sí | 3 | 90 | 01/04/2024 |
| 4 | Nuevas Alianzas | 4 | Zul'Drak | No | 4 | 90 | 23/09/2024 |
| 5 | El despertar | 5 | Suello Esmeralda | No | 10 | 120 | 06/01/2024 |
| 6 | La conquista demoníaca | 6 | Argus | Sí | 8 | 180 | 22/02/2024 |

| |
|---|
| Partida (<u>ID</u> , Nombre, Game_Master_ID, Está_En_Curso, Num_Sesión, Duración, Fecha, <i>Miembro_ID*</i>): |
| PK: id |
| FK: game_master_id → tabla miembro |

TABLA JUEGA

| NIVEL | PERSONAJE_ID* | PARTIDA_ID* | SALUD | FUERZA | DESTREZA | SABIDURIA | CARISMA | INTELIGENCIA | CONSTITUCIÓN |
|-------|---------------|-------------|-------|--------|----------|-----------|---------|--------------|--------------|
| 70 | 1 | 2 | 50000 | 10000 | 1000 | 1000 | 3000 | 1000 | 450 |
| 50 | 4 | 2 | 15000 | 300 | 10000 | 1800 | 800 | 1750 | 100 |
| 2 | 3 | 2 | 150 | 30 | 100 | 150 | 300 | 300 | 450 |
| 7 | 2 | 4 | 5000 | 1500 | 4500 | 3300 | 3700 | 4500 | 150 |
| 35 | 5 | 3 | 15000 | 5000 | 7000 | 5000 | 7000 | 5000 | 400 |
| 40 | 6 | 3 | 20000 | 7000 | 5000 | 10000 | 5000 | 3000 | 500 |
| 15 | 7 | 3 | 7500 | 2500 | 3000 | 3600 | 4000 | 7000 | 150 |
| 23 | 8 | 103 | 10000 | 3000 | 4000 | 1500 | 8000 | 3500 | 300 |

| |
|--|
| Juega (<u>nivel</u> , <u>personaje_id*</u> , <u>partida_id*</u> , salud, fuerza, destreza, sabiduría, carisma, inteligencia, constitución): |
| PK: nivel + personaje_id + partida_id |
| FK: personaje_id → tabla personaje |
| FK: partida_id → tabla partida |

Para la realización del Modelo Relacional se ha realizado una tabla por entidad, y otra de la relación Juega, la cual existe entre personaje y partida. Se hace esta tabla al ser una relación N:M.

Normalización del Modelo Relacional

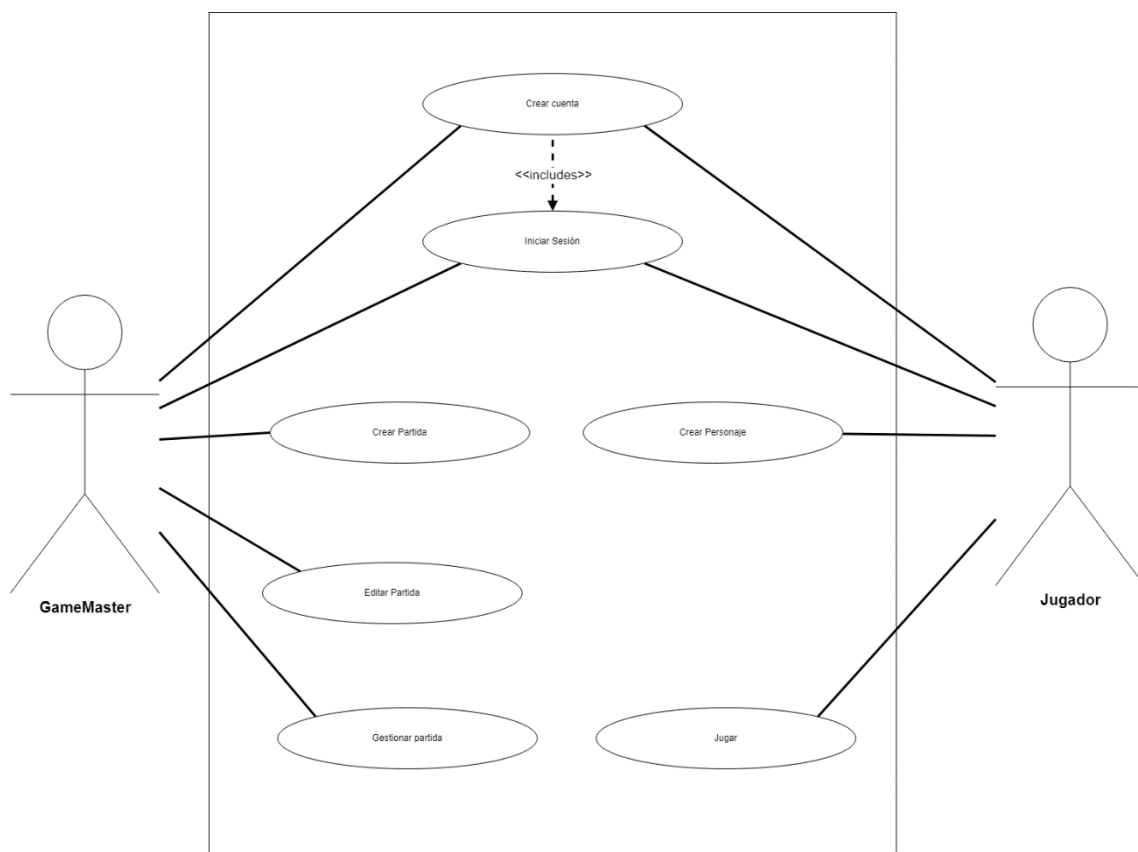
Para realizar la normalización de un modelo relacional, es necesario tener en cuenta las tres formas normales, e ir comprobando cada una de ellas en orden.

En primer lugar, la base de datos ya se encuentra en la primera forma normal, debido a que esta no tiene atributos multivaluados.

La base de datos ya está en segunda forma normal, ya que todos los atributos dependen en su totalidad de la clave.

Finalmente, la base de datos ya se encuentra en tercera forma normal, debido a que no hay dependencias transitivas.

Diagrama de Casos de Uso

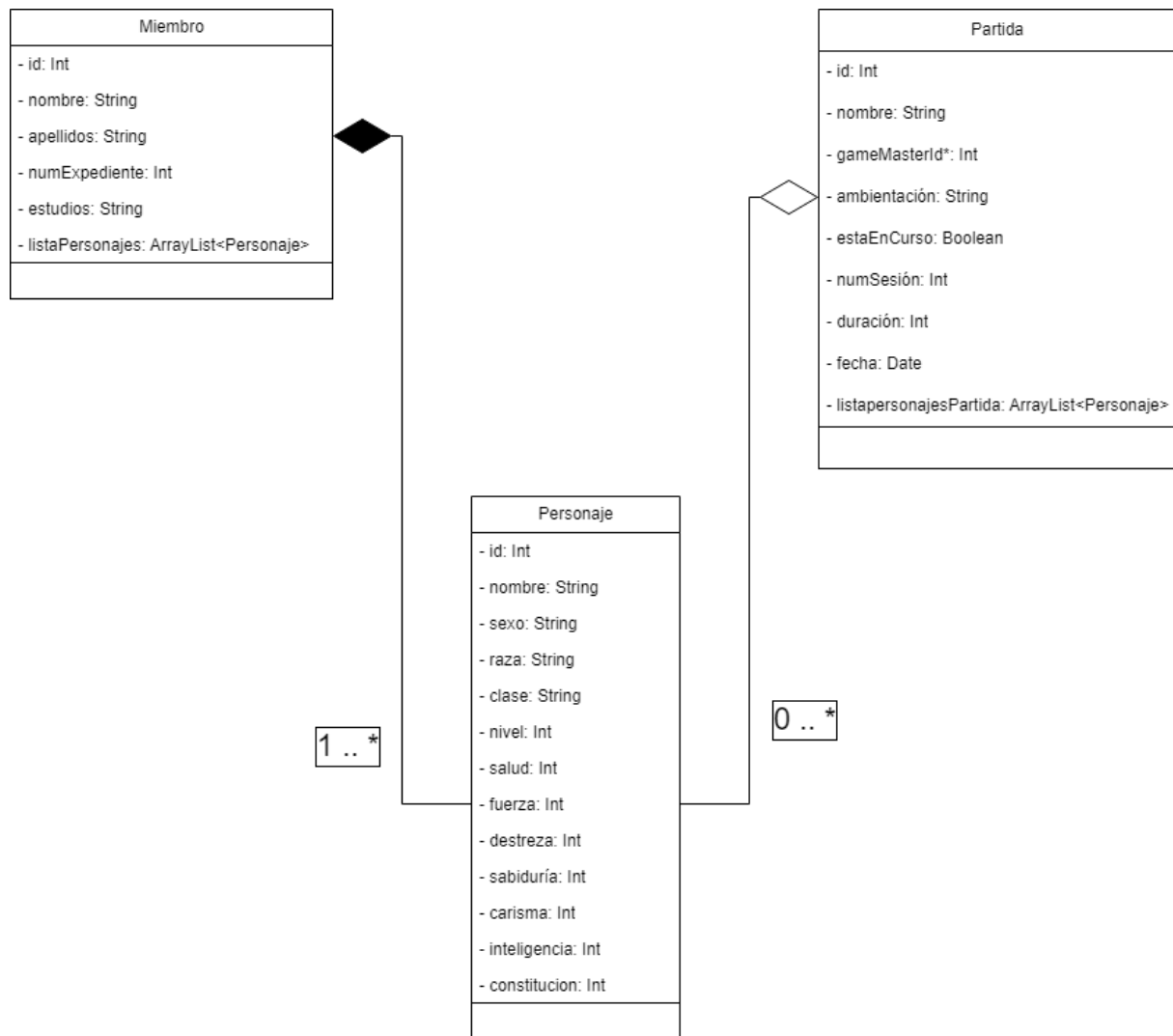


El diagrama de casos de uso está formado por dos actores, que serán el GameMaster y el Jugador. Hay tres casos de uso que coinciden para ambos actores, que son Crear Cuenta,

Iniciar Sesión y seleccionar partida. Iniciar sesión incluye de Crear Cuenta, ya que, al crear sesión, está se inicia de manera obligatoria.

El GameMaster puede de manera exclusiva crear partida, editarla y gestionarla. Por otro lado, el jugador puede crear un personaje, o jugar una partida.

Diagrama de Clases



El diagrama de clases está formado por un total de tres tablas. En primer lugar, se encuentra la tabla Miembro, la cual está formada por los siguientes atributos: id, nombre, apellidos, numExpediente, estudios y el ArrayList de listaPersonajes. Esta tabla tiene relación con la tabla de personaje, mediante una relación de composición. Esto se debe a que para que el personaje pueda existir, ha de existir el miembro, que será el que creará al personaje. Por el contrario, el miembro puede existir sin el personaje, lo que hace que la relación sea de composición. La relación es de 1 .. *, debido a que mínimo hay un miembro, y puede haber un máximo indefinido de personajes. La tabla de Personaje está formada por los siguientes atributos: id, nombre, raza, clase, nivel, salud, fuerza, destreza, sabiduría, carisma, inteligencia y constitución.

Finalmente, la tabla Personaje tiene relación con la tabla Partida, dicha relación es de agregación. Esto se debe a que el personaje pertenece a la partida, pero al contrario que la anterior relación, el personaje si que puede existir sin la partida. Por el otro lado, lo mismo ocurre con la partida, ya que está puede existir sin el personaje. La relación es de 0 .. *, ya que puede haber personajes sin ninguna partida, pero el máximo es indefinido. La tabla Partida está formada por los siguientes atributos: id, nombre, gameMasterId, ambientación, estaEnCurso, numSesión, duración, fecha y el Arraylist de listaPersonajesPartida.

Pruebas con JUnit

JUnit es un marco de trabajo de pruebas unitarias utilizado en el desarrollo de software Java, Su principal objetivo es facilitar la escritura, ejecución y análisis de pruebas unitarias de manera eficiente y estructurada. Algunos de los aspectos clave del uso de JUnit son:

- JUnit proporciona una estructura clara y consistente para escribir pruebas unitarias. Esto incluye la definición de métodos de prueba utilizando anotaciones como `@Test`, que indican qué métodos son pruebas unitarias.
- Aserciones: JUnit ofrece un conjunto de métodos de aserción que permiten verificar que los resultados de las pruebas sean los esperados. Esto garantiza que las pruebas se ejecuten en un contexto controlado y predecible.

Pruebas realizadas en JUnit

1. Pruebas de comprobación de Login en la base de datos

Propósito: verificar que el método `comprobarLogin` de la clase `AccesoBD` funciona correctamente al comprobar las credenciales de un usuario.

- Resultados esperados:
 - El resultado no debe ser nulo.
 - El arreglo resultado debe tener una longitud de 2.
 - El primer elemento debe ser 1.
 - El segundo elemento debe ser "lebron".

2. Prueba de devolución de ID de miembro

Propósito: comprobar que el método `devolverIdMiembro` devuelve el ID correcto para un miembro específico.

- Resultados esperados:
 - El resultado no debe ser nulo.
 - El ID devuelto debe ser 2.

3. Prueba de botón seleccionado en grupo de botones

Propósito: verificar que el método `getSelectedButtonText` identifica correctamente el botón seleccionado dentro de un `ButtonGroup`.

- Resultados esperados:
 - Cuando el botón 2 está seleccionado, el texto devuelto debe ser `Boton2`.
 - Si no hay ningún botón seleccionado, el resultado debe ser `null`.

4. Prueba de devolución de personaje por ID

Propósito: asegurar que el método `devolverPersonajeId` devuelve el ID correcto de un personaje específico.

- Resultados esperados:
 - El resultado no debe ser nulo.
 - El ID devuelto debe ser 9.

5. Prueba de devolución de partidas por nombre

Propósito: comprobar que el método `devolverPartidasNombre` devuelve correctamente una lista de nombres de partidas.

- Resultados esperados:
 - El resultado no debe ser nulo.
 - El resultado debe ser una instancia de `ArrayList`.

Base de Datos

La base de datos se ha creado basándose en los modelos y diagramas vistos anteriormente y consta de cuatro tablas: miembros, partidas, personajes y juega.

Para crear cada una de estas tablas se ha realizado un script de creación de las tablas. Cada una de ellas se ha creado definiendo sus columnas, estableciendo las claves primarias, foráneas y el resto de los datos de cada columna.

Por otro lado, se ha creado otro script para la inserción de datos en cada una de las tablas. Inicialmente se han añadido varias filas para tener una base establecida, ya que el resto de los datos e inserciones los irá añadiendo el usuario a través de la propia aplicación con las distintas funcionalidades que esta ofrece.

Todo esto queda reflejado de una forma más visual en el apartado anexos, donde se adjuntan las capturas de la creación de la base de datos.

Diseño de la Interfaz



Para la realización del diseño de la interfaz de las clases, se decidió optar por una paleta de colores con tonos azules, y tonos dorados y blancos para hacer contraste con el fondo de la interfaz.

Hay un total de once ventanas que se explicarán brevemente a continuación y cuyas imágenes estarán en el apartado anexos.

Creación del Proyecto y Vistas

En este proyecto, se ha aprovechado Maven para incluir y gestionar FlatLaf, un moderno y versátil Look and Feel de código abierto multiplataforma para aplicaciones Java. FlatLaf proporciona una apariencia visual moderna y consistente similar a las interfaces gráficas de aplicaciones profesionales actuales, lo que mejora significativamente la experiencia del usuario final.

FlatLaf ofrece una apariencia limpia y moderna, inspirada en los estilos de interfaces más recientes como las de macOS y Windows. Esto ayuda a que la aplicación tenga un aspecto profesional y atractivo, independientemente de la plataforma en que se ejecute. Además, FlatLaf soporta pantallas de alta resolución y permite personalizar fácilmente colores, fuentes

y otros aspectos visuales a través de archivos de configuración o directamente en el código. Esto permite ajustar la interfaz gráfica para que se alinee con la identidad visual del proyecto.

Diseño del Logo



Para el diseño del logo, se decidió optar por un aspecto bélico para mostrar la identidad del juego. Para conseguir transmitir esta identidad, se hace uso de ciertos elementos que aparecerán en el juego, como puede ser el caso de un hacha, una espada y un escudo. Finalmente, debido a que el nombre es el de Black Temple, se optó por colocar un templo en el centro del escudo.

5. Metodología

Para la gestión del proyecto, se han utilizado dos metodologías ágiles muy populares: Kanban, implementada a través de Trello, y Scrum, estructurada con sprints de 2 semanas. Ambas metodologías ofrecen características valiosas que han mejorado significativamente la eficiencia y efectividad del trabajo.

Utilizar Kanban con Trello ha permitido visualizar claramente el flujo del trabajo del equipo mediante la creación de tableros visuales que muestran el estado de cada tarea mediante tarjetas, organizando las tareas a realizar por sprints. Esto facilita la identificación rápida de tareas pendientes, en progreso y completadas, proporcionando una transparencia total en el avance del proyecto.

La flexibilidad es una de las mayores ventajas de Kanban, sin plazos ni sprints, permite que las tareas puedan cambiar de prioridad rápidamente. Kanban ha permitido establecer límites de trabajo en curso, evitando la sobrecarga del equipo y asegurando que se enfoquen en

completar tareas antes de comenzar nuevas, lo que mejora tanto la eficiencia como la calidad del trabajo.

Además, con Trello se pueden recopilar datos sobre el flujo de trabajo, identificar cuellos de botella y hacer ajustes en tiempo real. La visibilidad y transparencia proporcionadas por Trello facilitan la colaboración entre los miembros, comentando las tarjetas, adjuntando archivos o asignando tareas a personas específicas, mejorando la comunicación y coordinación.

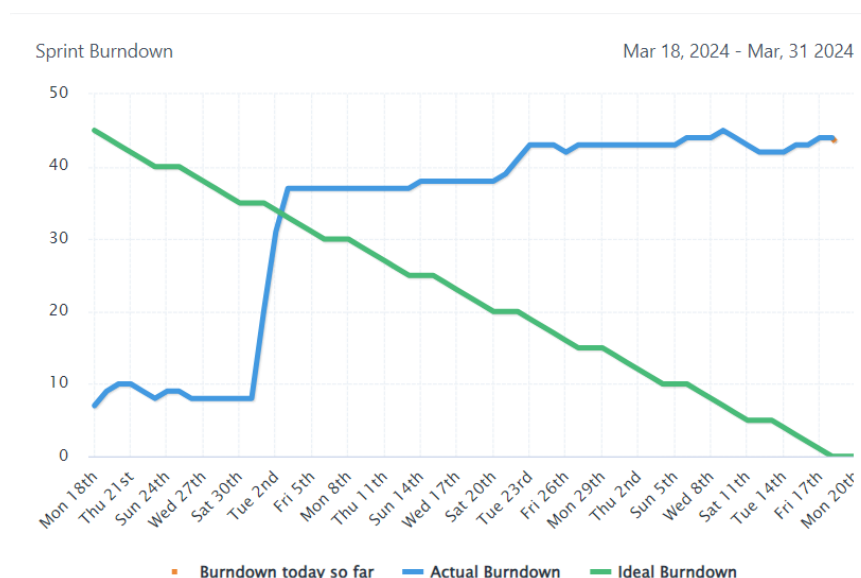
Por otro lado, se ha adoptado la metodología Scrum con sprints de 2 semanas para proporcionar una estructura y un ritmo predecible del trabajo.

Scrum incluye reuniones estructuradas como la planificación del sprint, las reuniones diarias, revisión del sprint y la retrospectiva. Estas reuniones fomentan la comunicación constante, aseguran que todos los miembros del equipo estén alineados y promuevan la mejora continua al reflexionar sobre el trabajo realizado y planificar mejoras para el siguiente sprint.

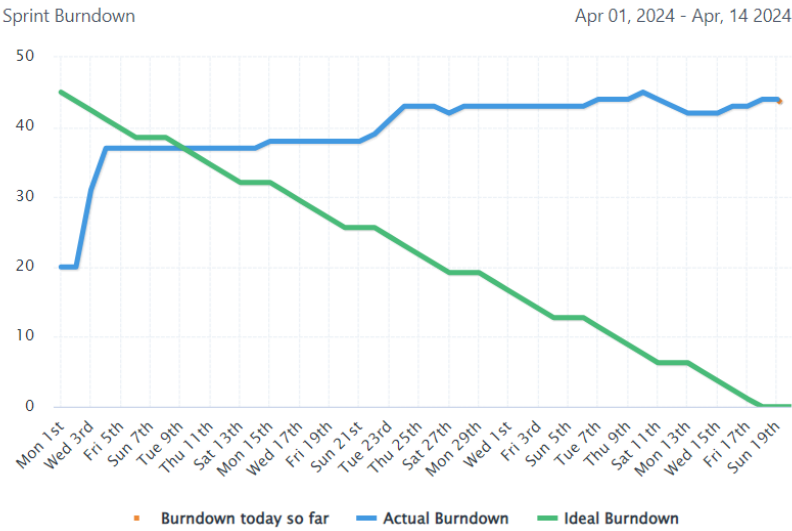
Una de las ventajas de Scrum es la entrega incremental de valor. Al final de cada sprint, se entrega un incremento de producto funcional, lo que nos permite obtener feedback temprano y frecuente. Esto aumenta la capacidad de adaptación a los cambios en los requisitos y garantiza que se está trabajando en lo más importante primero, maximizando el valor para el cliente.

Los Burndown Chart de cada uno de los sprints se pueden ver a continuación:

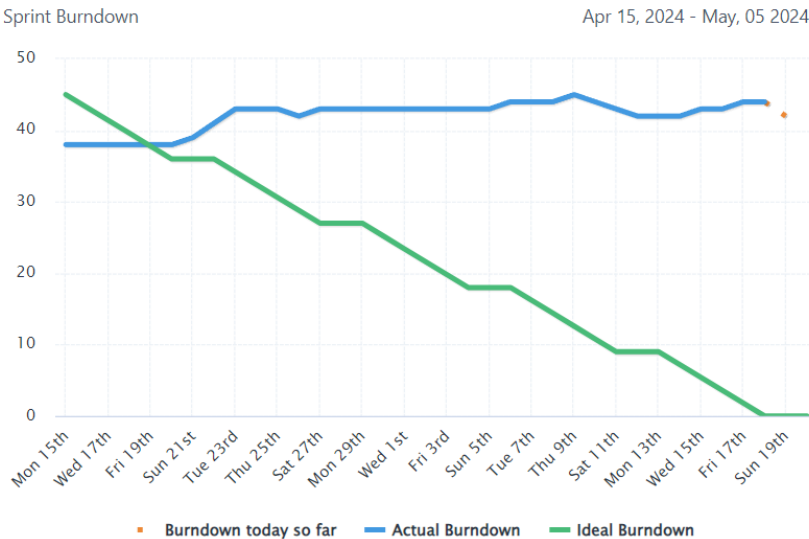
- 1er Sprint:



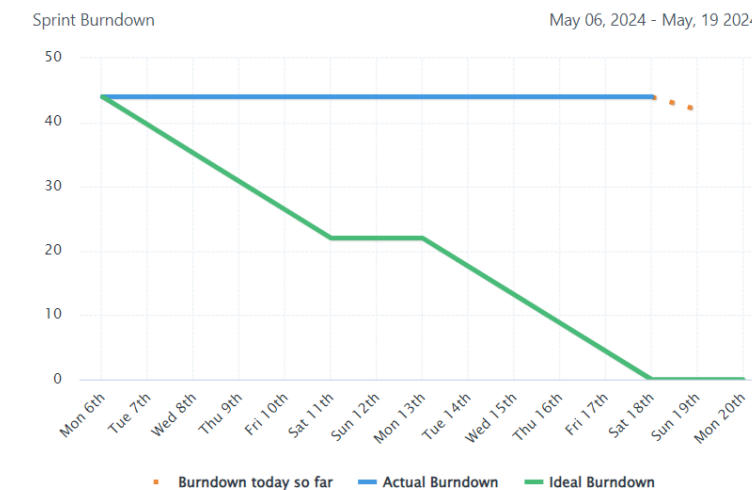
- 2do Sprint:



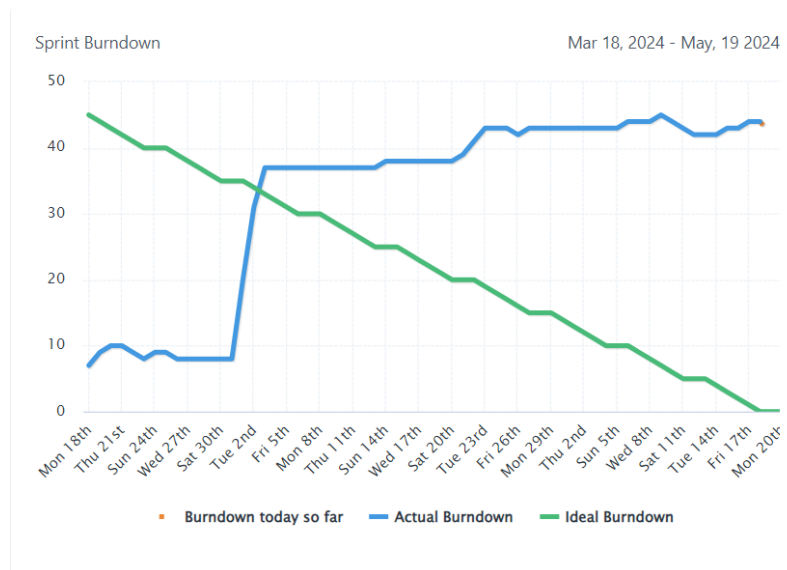
- 3er Sprint



- 4to Sprint

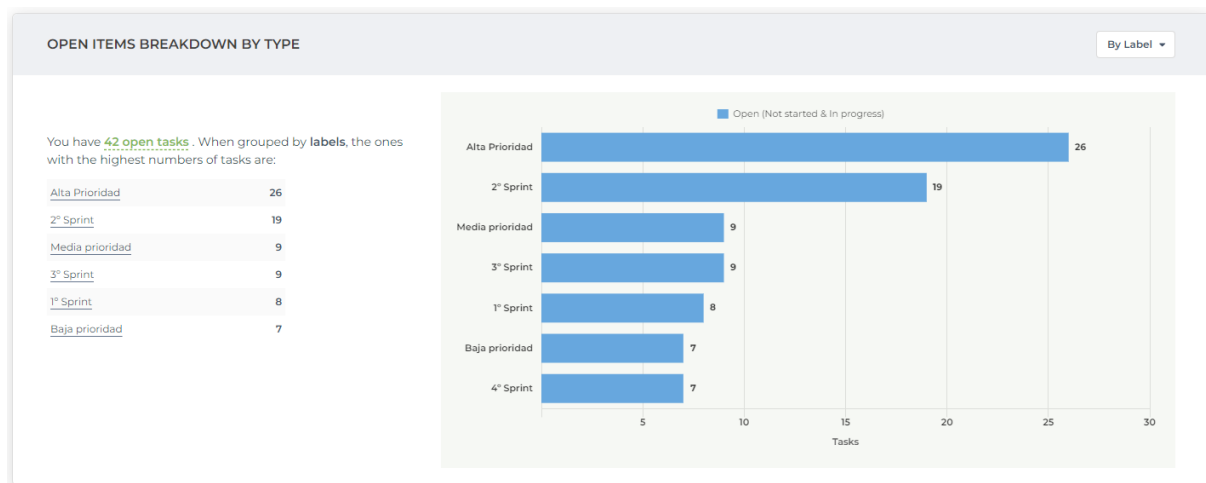
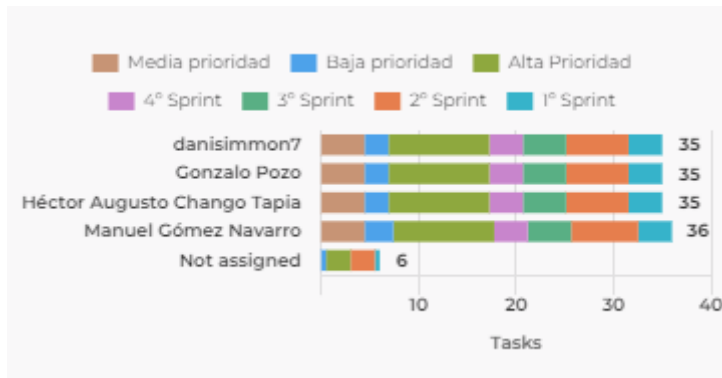


- Burndown Chart del proyecto completo



El enlace al GitHub del proyecto: <https://github.com/gonzalopozo/proyecto-integrador-1-daw-gonzalo-manuel-daniel-hector>

SprintBackLog



Las tareas completadas en las primeras semanas muestran horas restantes como 0 en las siguientes semanas.

Las tareas completadas en las dos últimas semanas se reflejan con horas en las semanas 6, 7 y 8.

Las tareas no iniciadas tienen una disminución de horas prevista conforme avanzan las semanas para reflejar una planificación progresiva.

Esta tabla proporciona una visión clara del progreso semanal de cada tarea a lo largo de un periodo de 8 semanas, incluyendo la finalización de la documentación JavaDoc, el manual de usuario en GitHub y el refinamiento del programa en las dos últimas semanas.

| Requisito | Tarea | Quién | Estado | Semana 1 | Semana 2 | Semana 3 | Semana 4 | Semana 5 | Semana 6 | Semana 7 | Semana 8 |
|-------------|----------------------------------|----------------------------------|------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Requisito 1 | Creación del proyecto de Java | Daniel, Héctor, Manuel y Gonzalo | Completada | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Requisito 1 | Creación proyecto GitHub | Daniel, Héctor, Manuel y Gonzalo | Completada | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Requisito 1 | Diagrama E/R | Daniel, Héctor, Manuel y Gonzalo | Completada | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Requisito 1 | Preguntas Sara | Daniel, Héctor, Manuel y Gonzalo | Completada | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Requisito 2 | Crear ventana partida | Daniel, Héctor, Manuel y Gonzalo | Completada | 32 | 24 | 16 | 8 | 0 | 0 | 0 | 0 |
| Requisito 2 | Crear ventana crear partida | Daniel, Héctor, Manuel y Gonzalo | Completada | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Requisito 2 | Crear ventana crear personaje | Daniel, Héctor, Manuel y Gonzalo | Completada | 56 | 48 | 40 | 32 | 24 | 16 | 8 | 0 |
| Requisito 2 | Crear ventana pantalla principal | Daniel, Héctor, Manuel y Gonzalo | Completada | 32 | 24 | 16 | 8 | 0 | 0 | 0 | 0 |
| Requisito 2 | Crear ventana registro miembro | Daniel, Héctor, Manuel y Gonzalo | Completada | 56 | 48 | 40 | 32 | 24 | 16 | 8 | 0 |
| Requisito 3 | Clases para Control BBDD | Daniel, Héctor, Manuel y Gonzalo | Completada | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Requisito 3 | Añadir tareas realizadas al Word | Daniel, Héctor, Manuel y Gonzalo | Completada | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Requisito 3 | Clases para Modelo | Daniel, Héctor, Manuel y Gonzalo | Completada | 56 | 48 | 40 | 32 | 24 | 16 | 8 | 0 |
| Requisito 3 | Introducir función JPassword | Daniel, Héctor, Manuel y Gonzalo | Completada | 48 | 40 | 32 | 24 | 16 | 8 | 0 | 0 |
| Requisito 4 | Realización de pruebas JUnit | Daniel, Héctor, Manuel y Gonzalo | Completada | 48 | 40 | 32 | 24 | 16 | 8 | 0 | 0 |
| Requisito 4 | Documentación JavaDoc | Daniel, Héctor, Manuel y Gonzalo | Completada | 0 | 0 | 0 | 0 | 0 | 8 | 16 | 24 |
| Requisito 4 | Manual de usuario en GitHub | Daniel, Héctor, Manuel y Gonzalo | Completada | 0 | 0 | 0 | 0 | 0 | 8 | 16 | 24 |
| Requisito 4 | Refinamiento del programa | Daniel, Héctor, Manuel y Gonzalo | Completada | 0 | 0 | 0 | 0 | 0 | 8 | 16 | 24 |

La forma en la que se ha dividido el trabajo a lo largo de cada uno de los Sprints ha sido de la siguiente manera:

1. CRONOGRAMA DE ACTIVIDADES SPRINT 1 → 18/03/2024 – 31/03/2024

18/03/2024

- Creación del repositorio en GitHub.
- Realización del tablero de tareas en Trello.
- Inicio del diagrama de entidad-relación.

19/03/2024

- Creación del proyecto de JAVA y subida a GitHub, con un primer commit.
- Finalización de la versión 0.1 del diagrama de entidad relación.
- Se añade un segundo commit con la primera versión del diagrama terminado.

22/03/2024

- Actualización del diagrama entidad relación y realización de un nuevo commit con la versión 1.0.
- Inicio de la creación de los requisitos de la aplicación.

25/03/2024

- Actualización del diagrama entidad relación Y realización de un nuevo commit con la versión 1.1.
- Creación de una primera fase de requisitos de la aplicación, que puede actualizarse a medida que avance el proyecto.

30/03/2024

- Modificación de la licencia del proyecto.
- Actualización de la memoria añadiendo las especificaciones del proyecto, el repositorio de GitHub y el tablero de Trello.
- Modificación del README del repositorio de GitHub.

03/04/2024

- Actualización de la memoria del proyecto modificando la disposición de algunos apartados, añadiendo un apartado sobre SCRUM y modificando el apartado de “ANÁLISIS DE ESPECIFICACIONES DEL PROYECTO”.

2. CRONOGRAMA DE ACTIVIDADES SPRINT 2 → 01/04/2024 – 14/04/2024

01/04/2024

- Creación del modelo relacional.

02/04/2024

- Inicio diseño de interfaz a mano.

03/04/2024

- Inicio del diagrama de casos de uso.

04/04/2024

- Normalización del modelo relacional.
- Diseño de la interfaz en Illustrator.

09/04/2024

- Finalización del diagrama de casos de uso.

11/04/2024

- Creación de la Base de Datos.
- Creación de tablas, índices, etc.
- Creación de las clases pertenecientes a la Vista.

12/04/2024

- Finalización de la Base de Datos.
- Inserción de datos necesarios para la app.

13/04/2024

- Modificación de algunas ventanas.

14/04/2024

- Finalización del logo.
- Se implementa el diseño de la interfaz a las ventanas de las clases.

3. CRONOGRAMA DE ACTIVIDADES SPRINT 3 → 15/04/2024 – 05/05/2024

16/04/2024

- Creación del diagrama de clases.
- Actualización de diagrama de casos de uso.
- Actualización de las vistas de la app.
- Actualización del .gitignore.

24/04/2024

- Solución de errores: dependencia FlatLaf en Maven.
- Configuración en VSCode, transferencia de clases de Java a VSCode.

01/05/2024

- Actualización de las vistas de la app.
- Actualización del diagrama de clases.

02/05/2024

- Actualización de las vistas de la app.

03/05/2024

- Solución de errores: imágenes corrompidas.

04/05/2024

- Actualización de las vistas de la App y creación de Listeners.

05/05/2024

- Actualización de las vistas de la App, implementación de la conexión con la base de datos.

4. CRONOGRAMA DE ACTIVIDADES SPRINT 3 → 05/05/2024 – 19/05/2024

05/05/2024

- Actualización de las vistas de la app.

11/05/2024

- Actualización de las vistas de la app.
- Modificación listeners.
- Modificación diagramas.

15/05/2024

- Actualización de las vistas de la app.
- Actualización listeners y acceso base de datos.

16/05/2024

- Crear cuenta.

17/05/2024

- Solución de errores.
- Integración botones personaje.
- Actualización Listeners.

18/05/2024

- Actualización de las vistas de la App y de Listeners.
- Pruebas JUnit.

19/05/2024

- Actualización de las vistas de la App.
- Creación de unirse a partida listener.
- Seleccionar Game Master botones.
- Pruebas JUnit.

6. Resultados y conclusiones

Una vez finalizado el proyecto, algo en lo que estamos todos los miembros de acuerdo es que el proyecto ha resultado un gran desafío para todos nosotros, especialmente a la hora de coordinarnos y trabajar con metodologías ágiles, algo que no habíamos hecho hasta el momento.

La primera adversidad a la que nos enfrentamos fue decidir la forma en la íbamos a trabajar, llegando a el acuerdo de trabajar cada miembro del grupo en todos los distintos aspectos de este proyecto. Tomamos esta decisión ya que consideramos que sería la mejor forma para aprender de todos los aspectos diferentes que se iban a tratar a lo largo del proyecto.

En cuanto a los resultados obtenidos, destacan los siguientes puntos clave:

- Funciones implementadas: se han desarrollado y probado con éxito las funcionalidades básicas de la aplicación, permitiendo a los usuarios acceder y gestionar las partidas de manera eficiente.
- Interfaz de usuario intuitiva, inspirada en otras interfaces de juegos y páginas conocidas. Se ha diseñado una interfaz de usuario intuitiva y fácil de usar, que facilita la navegación e interacción con la aplicación tanto para los Game Master como para los jugadores.
- Estabilidad y rendimiento: se ha prestado especial atención en la estabilidad y rendimiento de la aplicación, realizando infinitas pruebas para garantizar una experiencia fluida y sin problemas a los usuarios.

Como conclusiones, el equipo ha demostrado habilidades sólidas en el desarrollo de software, implementando los conocimientos adquiridos a lo largo del curso para lograr alcanzar los objetivos del proyecto dentro del tiempo establecido para cada entrega. A pesar de los desafíos encontrados durante el proceso, como la gestión de recursos y la resolución de problemas técnicos y corrección de errores, el equipo ha sabido enfrentarlos de manera efectiva y solvente, encontrando soluciones adecuadas en cada momento.

Por otro lado, el proyecto ha sido una oportunidad de trabajar y aprender en equipo. Se han adquirido nuevos conocimientos y habilidades en el desarrollo de aplicaciones, así como en la gestión de proyectos y colaboración en equipo. Se han identificado áreas de mejora para futuros proyectos, como la implementación de nuevas funcionalidades.

En resumen, el proyecto de desarrollo de la aplicación Black Temple ha sido una experiencia enriquecedora que ha permitido al equipo alcanzar sus objetivos, aprender de sus experiencias y sentar las bases para futuros proyectos. Estamos orgullosos del trabajo realizado y esperamos seguir innovando y creciendo en el campo del desarrollo de software.

7. Trabajos futuros

La aplicación dedicada a la gestión de partidas de juegos de rol Black Temple cuenta ya con su versión inicial, que incorpora características fundamentales como el Login, la creación de cuentas, selección de roles y gestión de partidas. Sin embargo, este es solo el comienzo, ya que a medida que se desarrolla Black Temple, surgen nuevas ideas de mejoras y cambios que enriquecerán aún más la experiencia para ofrecer a sus usuarios la mejor experiencia posible.

Algunos de estos cambios y mejoras que se desean implementar en el futuro son:

- En lugar de utilizar imágenes estáticas para representar personajes, introducir GIFs animados. Esto añadiría dinamismo a la interfaz y haría que los personajes y escenas fueran más vivos e interesantes para los usuarios.
- Implementar imágenes y modelos que cambien en función de las clases seleccionadas al crear un personaje. Cada clase (guerrero, mago, paladín, etc.) tendría su propio conjunto de imágenes o GIFs únicos, proporcionando una representación visual acorde a las características del rol escogido.
- Integrar un sistema de imágenes que cambien a medida que el personaje sube de nivel. Esto mostraría visualmente el progreso y la evolución del personaje, motivando al jugador al ver reflejado el avance en la apariencia de su personaje con nuevos trajes y sets de armaduras o armas.
- En un paso más avanzado, se consideraría la inclusión de modelos en 3D que se puedan rotar y visualizar en diferentes ángulos, lo que brindaría una experiencia más inmersiva.
- Incorporar animaciones de combate que podrían previsualizarse en la pantalla de creación de personaje y en la pantalla de gestión de partidas, dando una mejor idea e inmersión de la experiencia del entorno de juego.
- Ampliar las opciones de personalización, permitiendo a los usuarios modificar aspectos detallados de la apariencia de los personajes, como el peinado, los colores de la vestimenta o los rasgos faciales.

- Por último, agregar mayor personalización para las diferentes clases en el momento de subir de nivel, de forma que cuando sea un guerrero o un paladín, los atributos que más estadísticas suban sean los de fuerza o salud, y los de otros personajes como el mago o el brujo, sean la inteligencia o la sabiduría.

Anexos

Anexo I – Listado de requisitos de la aplicación

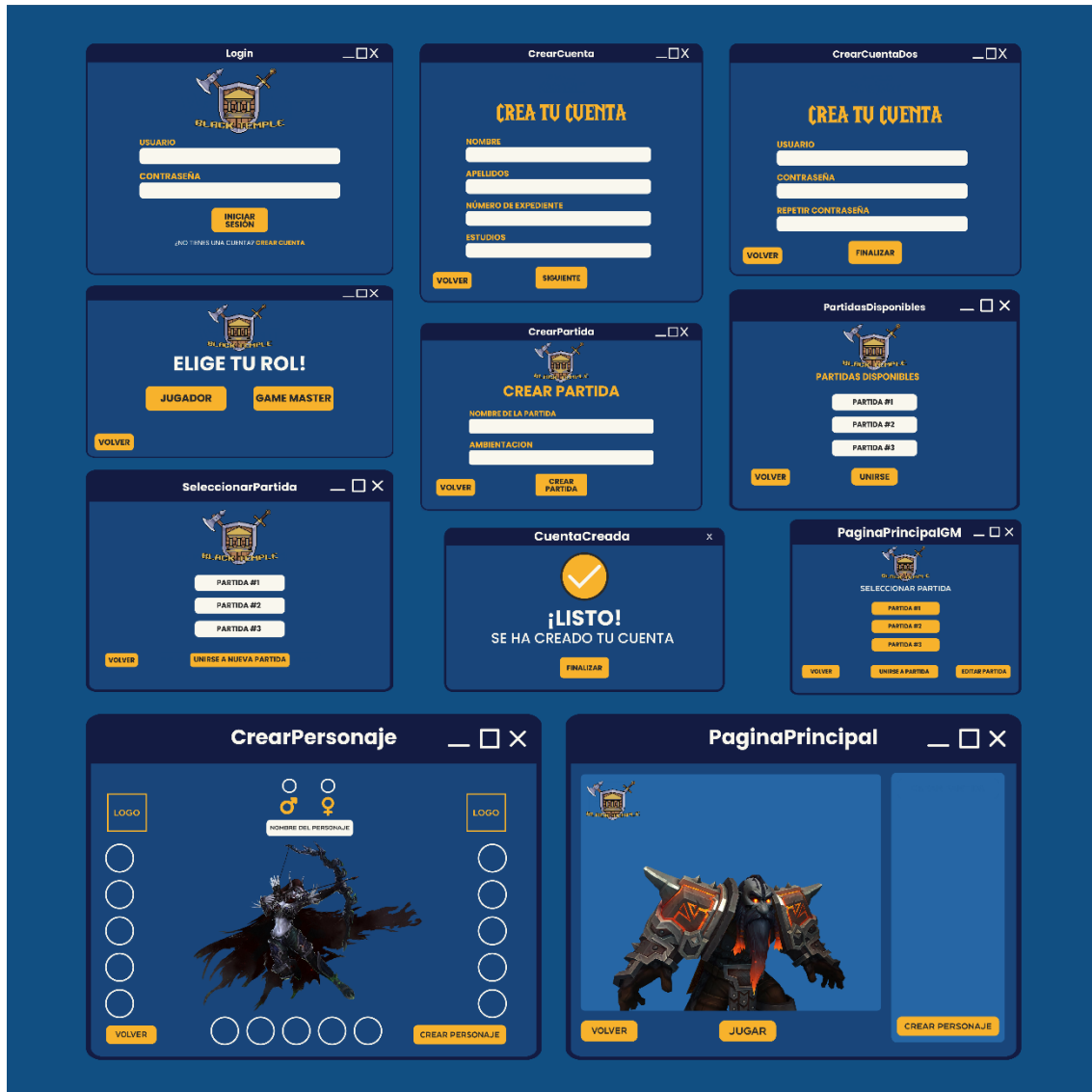
Para garantizar el correcto funcionamiento de la aplicación Black Temple en el entorno de desarrollo con Eclipse y MySQLite, se han identificado los siguientes requisitos:

Requisitos de Programación:

- Entorno de Desarrollo Integrado (IDE):
 - Gestión de Dependencias: Integración de herramientas de gestión de dependencias como Maven para la administración eficiente de las librerías y recursos del proyecto.
- Requisitos de Base de Datos:
 - Gestor de Base de Datos: Utilización de MySQLite como gestor de base de datos relacional para almacenar y gestionar la información de usuarios, partidas y personajes.
 - Conector JDBC: Integración del conector JDBC (Java Database Connectivity) para establecer la conexión entre la aplicación Java y la base de datos MySQLite.
 - Driver JDBC para MySQLite: Descarga e instalación del driver JDBC compatible con MySQLite para permitir la comunicación entre la aplicación Java y la base de datos.
 - Creación de la Base de Datos: Ejecución de scripts de creación de la base de datos y de las tablas necesarias para almacenar los datos relevantes para el funcionamiento de la aplicación.

Anexo II – Guía de uso de la aplicación

Idea principal de cómo se iban a ver las ventanas, este fue uno de los primeros bocetos sobre los que posteriormente se realizaron cambios.

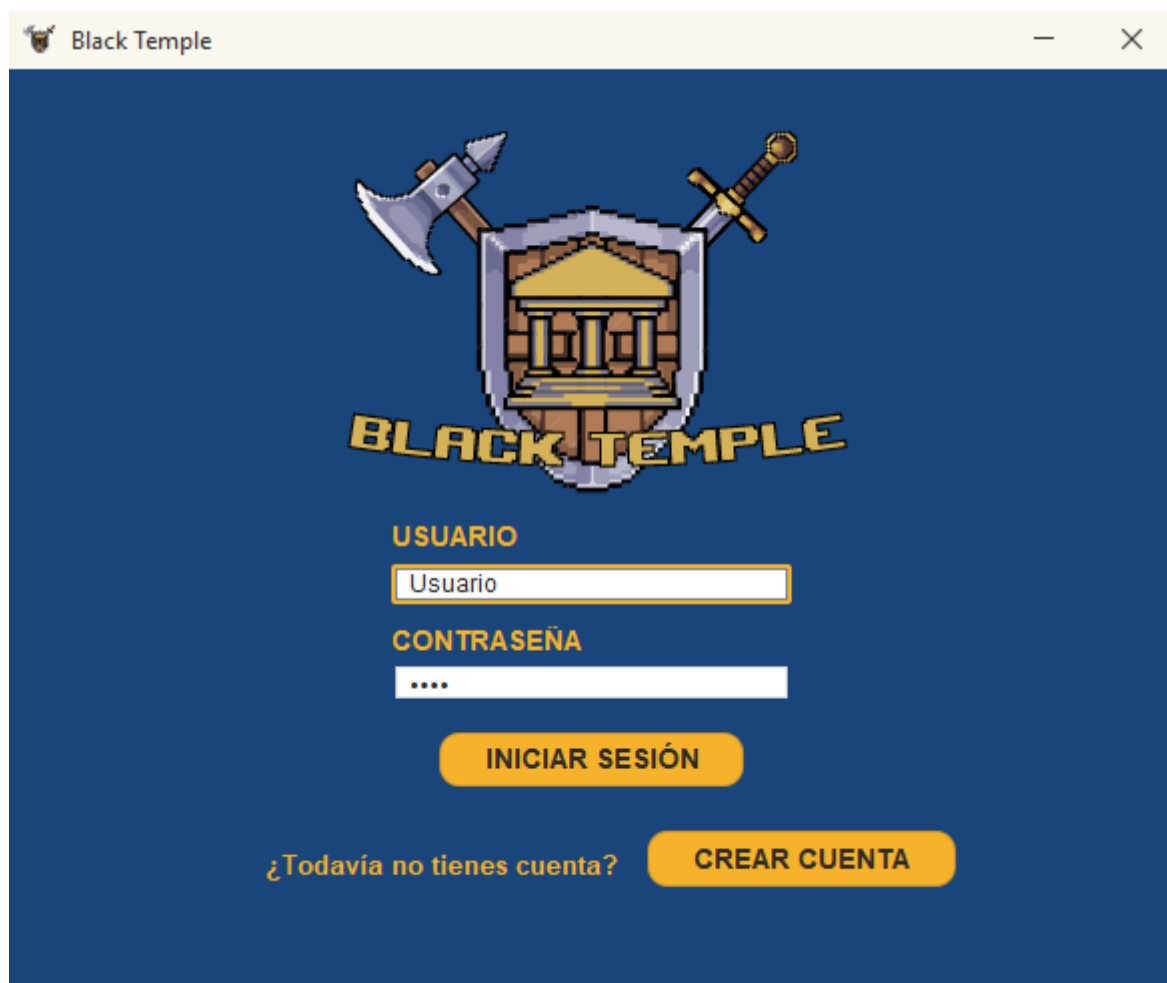


La primera ventana es la del Login (1), a través de la cual el usuario podrá o bien acceder a la aplicación, o crearse una cuenta. En caso de querer crear la cuenta, cuando se haga clic en el botón, aparecerá la ventana de crear cuenta (2), en la que introducirá los datos pertinentes. Una vez hecho esto, aparecerá la siguiente ventana de creación de cuenta (3), en la que terminará de configurar su cuenta con usuario y contraseña. Al haberse creado la cuenta, aparecerá el mensaje indicativo de que se ha creado la cuenta (4).

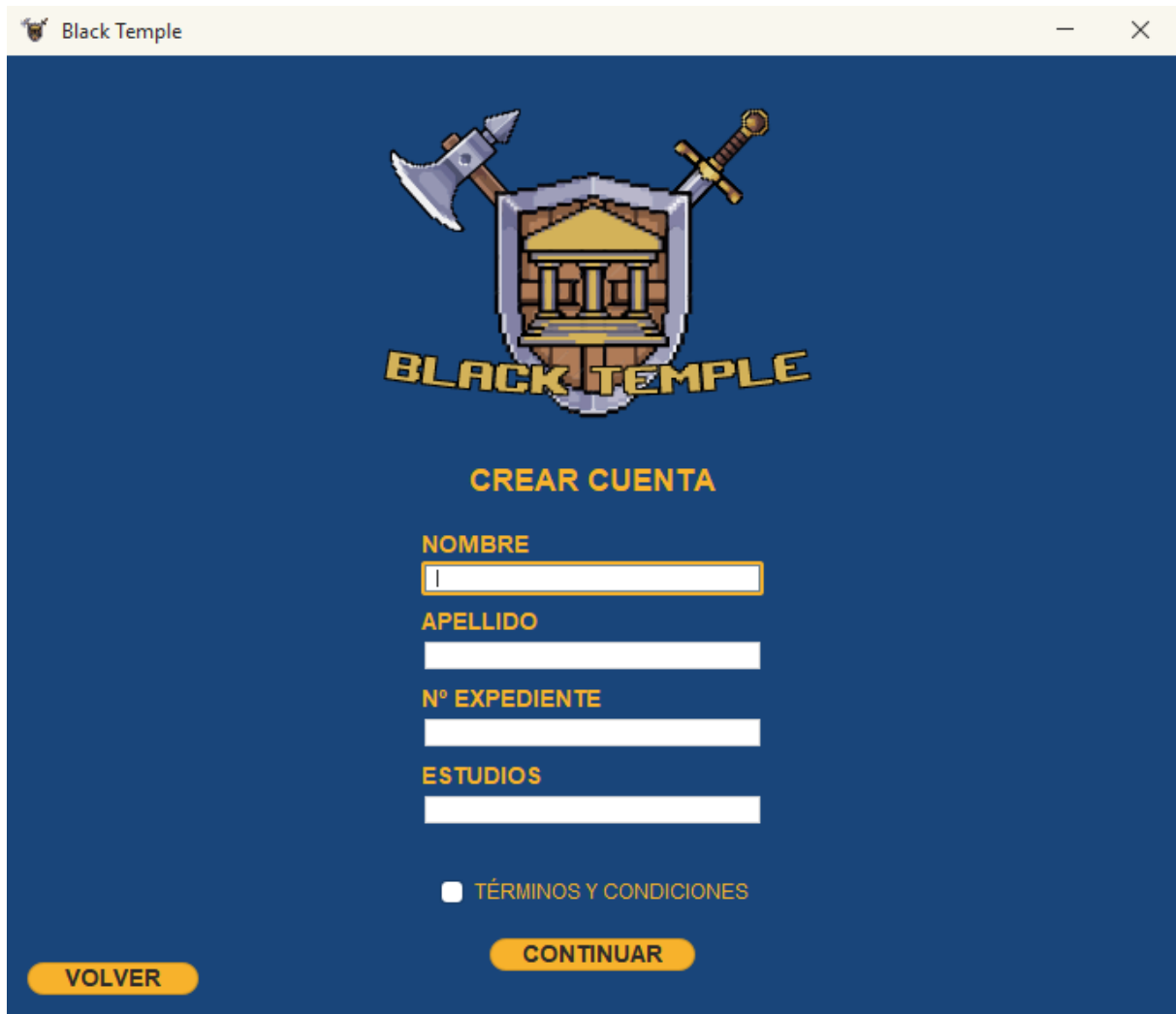
Lo siguiente que le aparecerá al usuario, será la pantalla de selección de rol (5), ya que este podrá seleccionar si desea acceder como Game Master, o como jugador. Si se selecciona ser Game Master, al usuario le aparecerá una nueva ventana de selección, donde deberá elegir si continuar una partida o empezar una nueva (6). Para crear una nueva partida (7) se deberá

elegir el nombre y la ambientación, si se presiona en continuar (8), aparecerán las partidas ya comenzadas y dirigidas por este usuario, donde podrá editarlas para cambiar su nombre (9), o unirse a ellas (10) para dirigirlas.


Por otro lado, si el usuario accede como jugador, aparecerá una ventana principal (11) donde el jugador podrá crear un personaje (12) eligiendo su nombre, sexo, raza y clase, y donde, una vez creados los personajes, podrá seleccionarlo y ver las partidas en las que se encuentra (13), borrarlo, o presionar en jugar (14), donde aparecerá una nueva ventana que indicará al usuario las partidas disponibles tanto para unirse a ellas por primera vez, como aquellas partidas en las que el usuario ya está jugando, para que elija a qué partida quiere acceder. Además, este usuario podrá consultar los detalles del resto de participantes de cada partida pulsando en detalles (15), para ver quiénes están participando y todos sus rasgos y atributos. Por último, si se selecciona una partida y se presiona el botón unirse (16), el personaje entrará a la partida. Dentro de la partida hay un botón de detalles niveles (17), el cual permite ver los niveles de los personajes.



(1) Pantalla correspondiente al Login



Black Temple



CREAR CUENTA

NOMBRE

APELLIDO

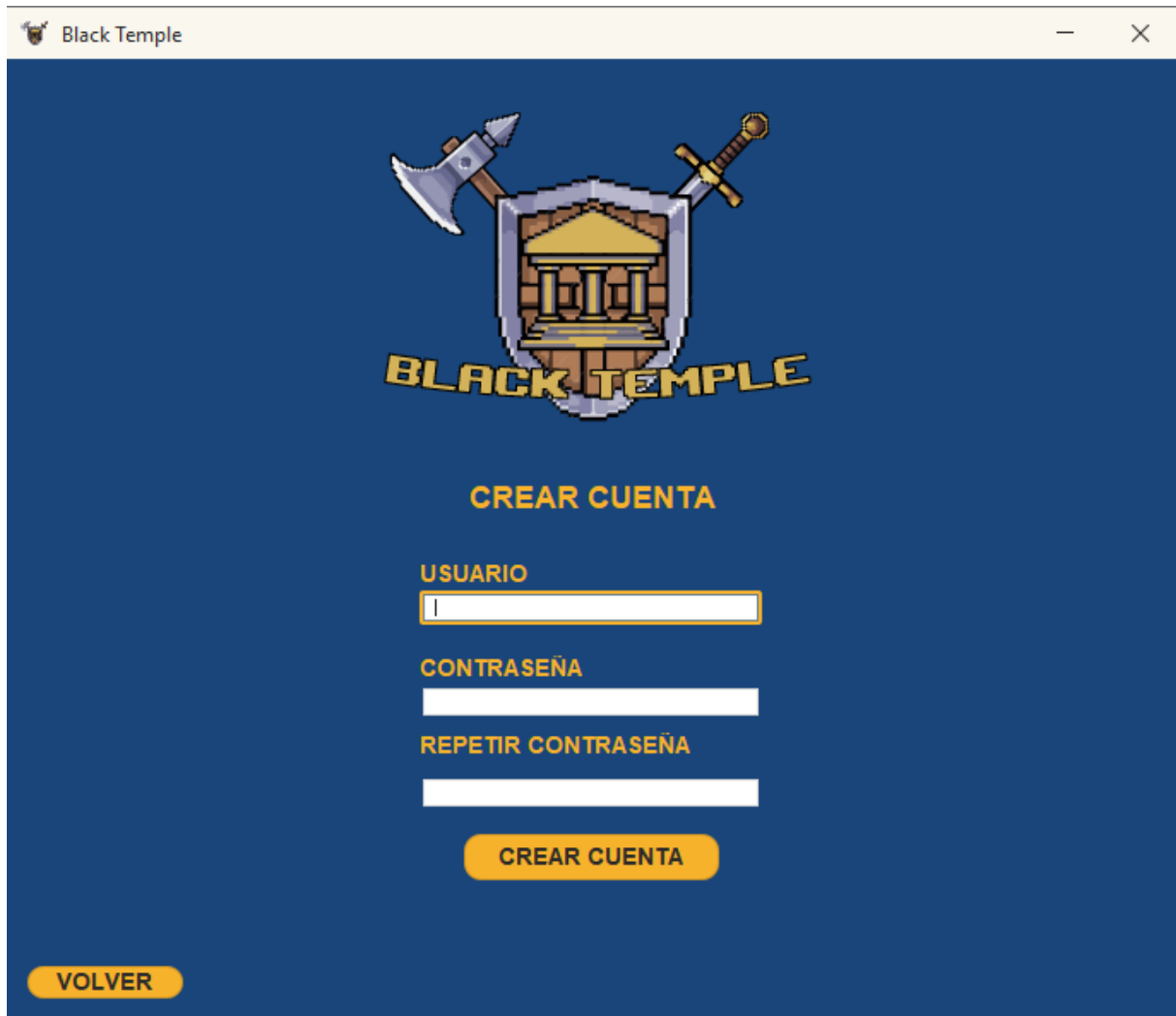
Nº EXPEDIENTE

ESTUDIOS


☐ TÉRMINOS Y CONDICIONES

VOLVER **CONTINUAR**

(2) Pantalla de crear cuenta



Black Temple



CREAR CUENTA

USUARIO

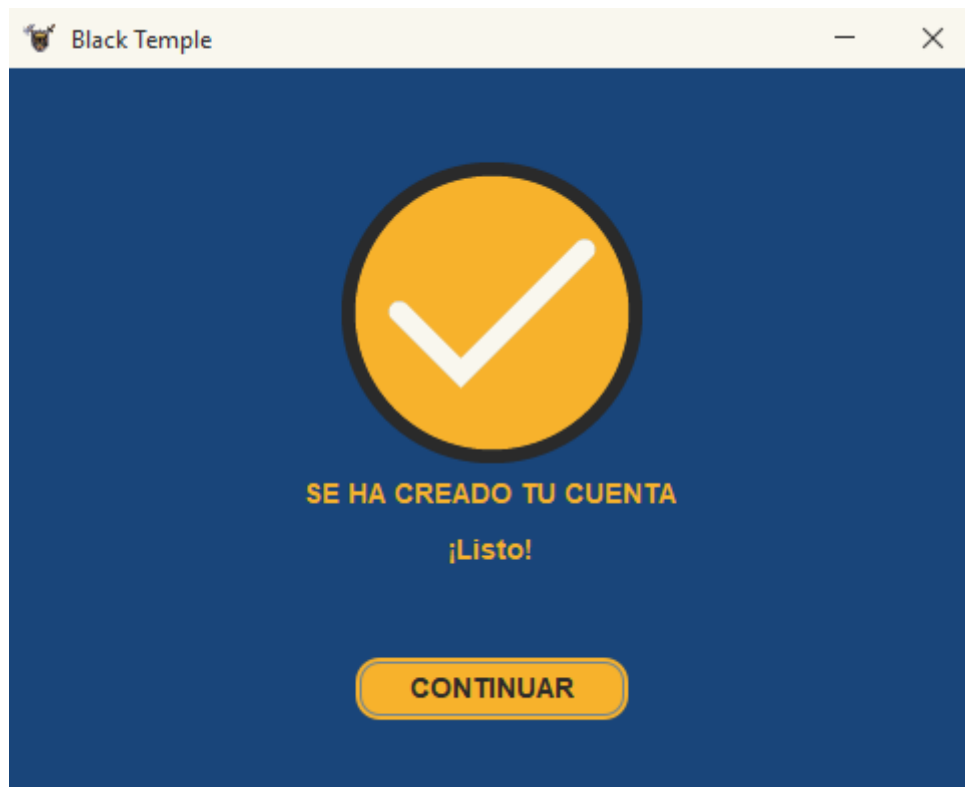
CONTRASEÑA

REPETIR CONTRASEÑA

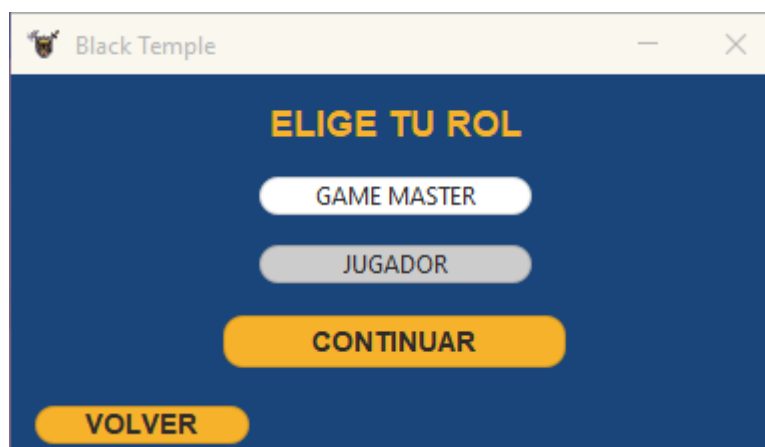
CREAR CUENTA

VOLVER

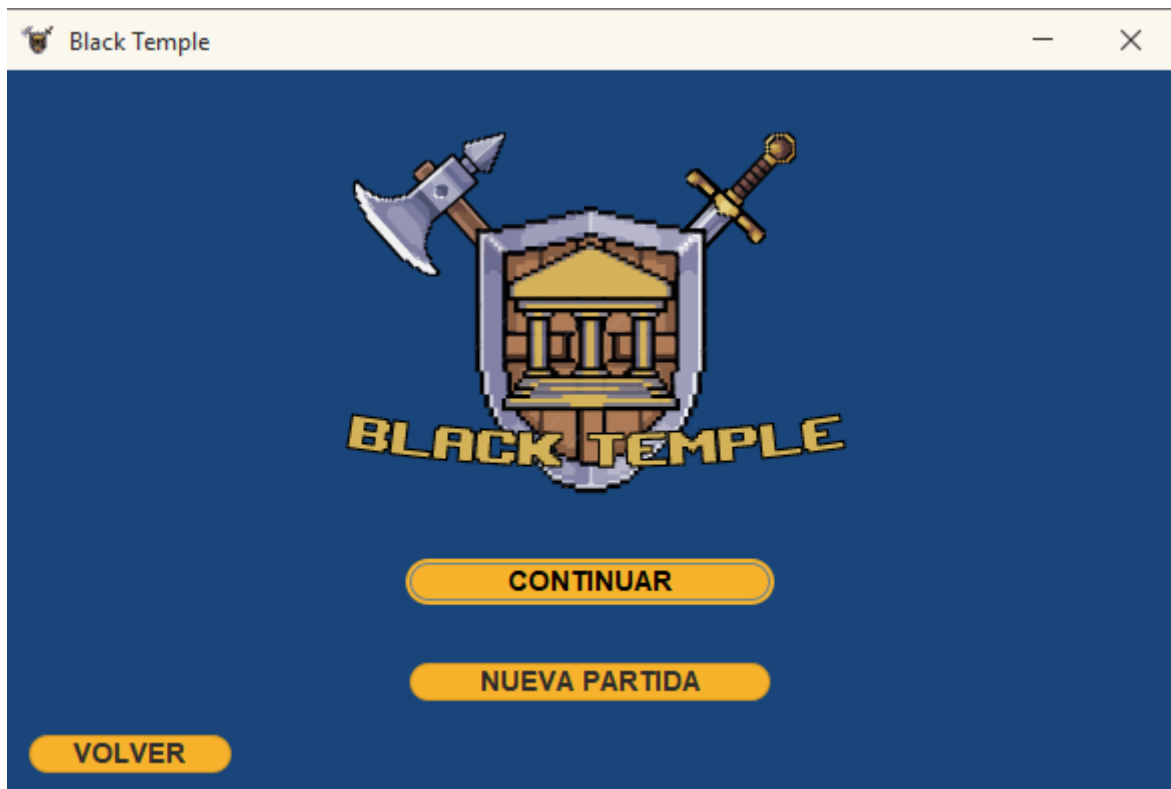
(3) Pantalla de crear cuenta dos



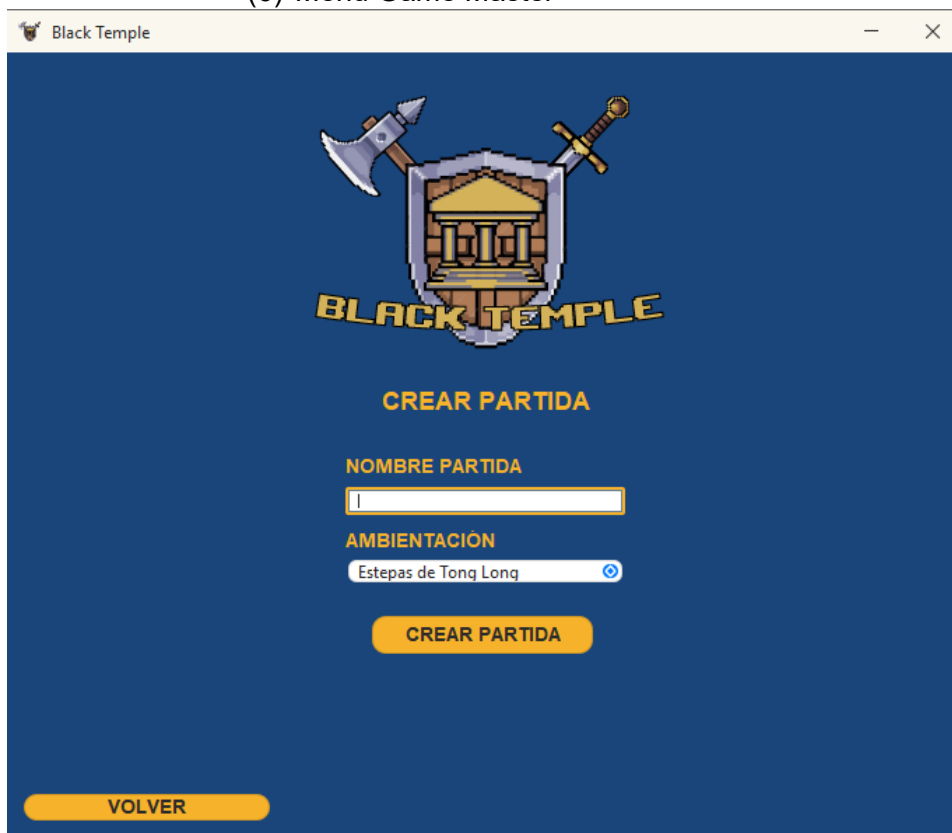
(4) Pantalla de cuenta ha sido creada



(5) Selección de rol



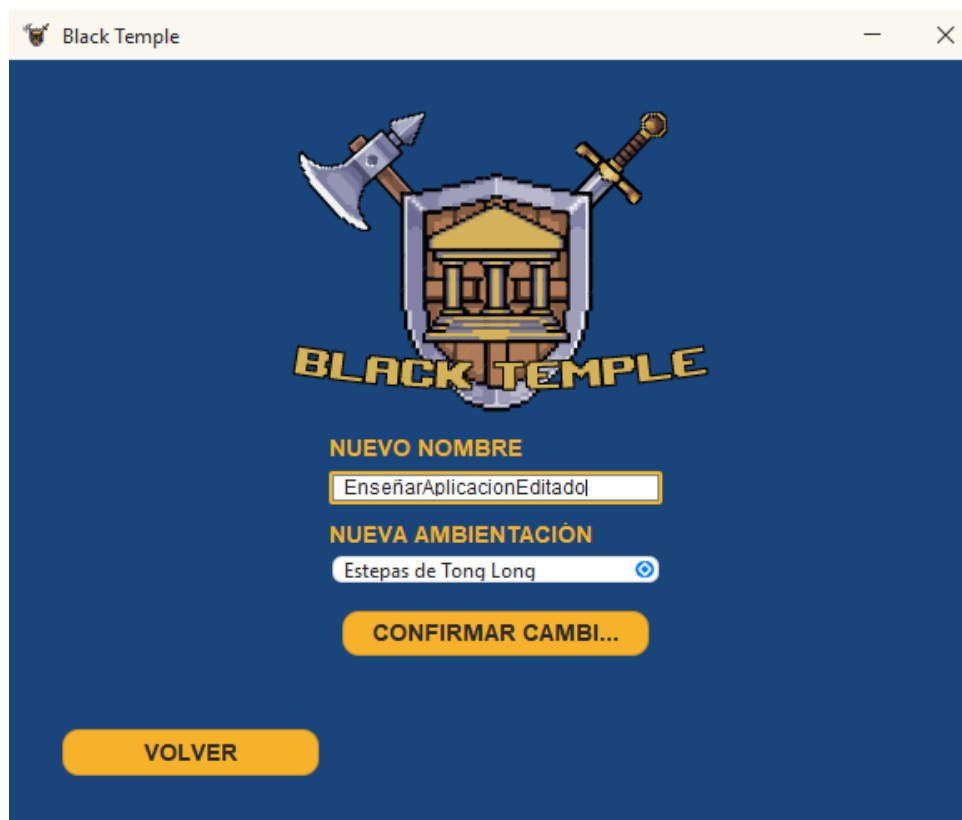
(6) Menú Game Master



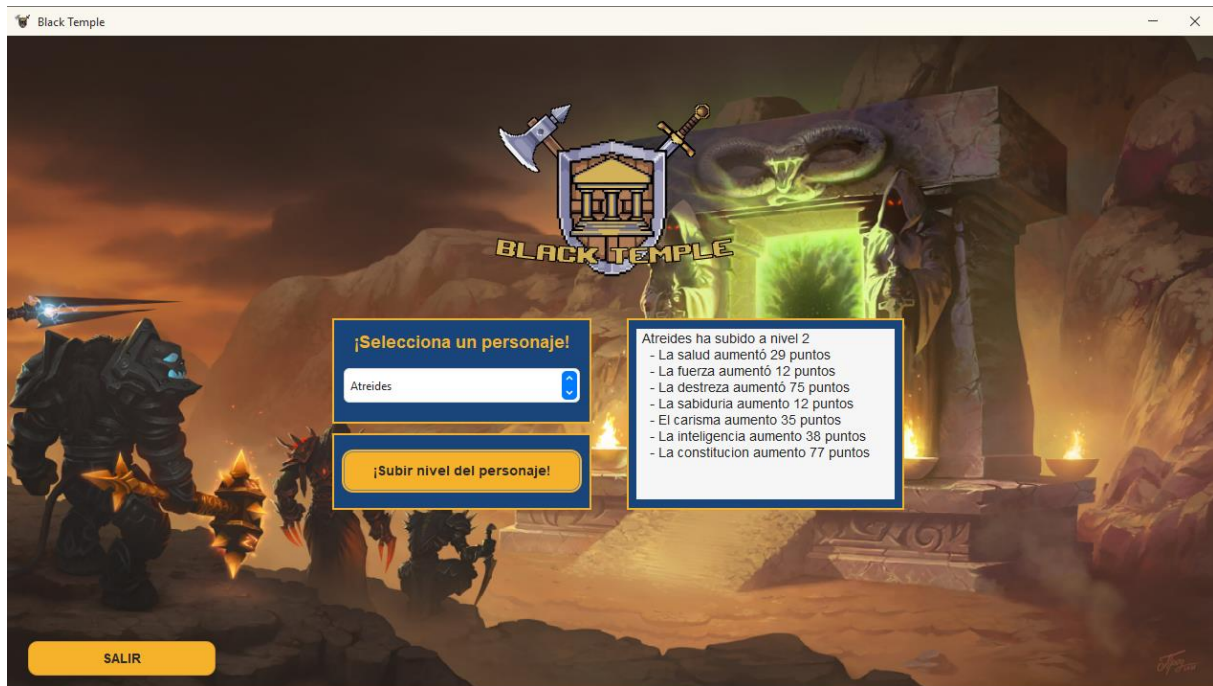
(7) Pantalla Nueva Partida



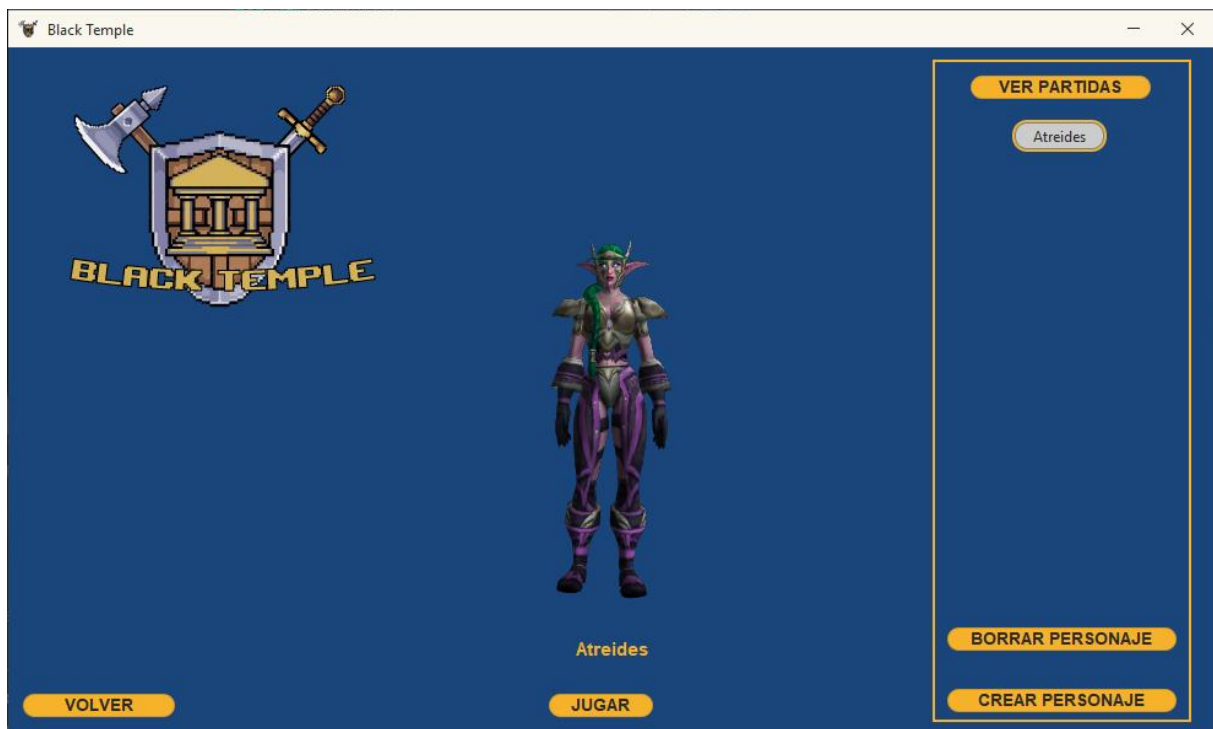
(8) Pantalla Seleccionar Partida



(9) Pantalla Editar Partida



(10) Pantalla Iniciar Partida



(11) Pantalla Principal Jugador



(12) Pantalla creación de personaje

| Nom... | Raza | Clase | Partida | Nivel | Salud | Fuerza | Destr... | Sabid... | Caris... | Inteli... | Const... |
|-----------|-----------|-------|-----------|-------|-------|--------|----------|----------|----------|-----------|----------|
| Atreid... | Elfo d... | Bruja | Nuev... | 7 | 5000 | 1500 | 4500 | 3300 | 3700 | 4500 | 150 |
| Atreid... | Elfo d... | Bruja | Partid... | 1 | 124 | 108 | 102 | 131 | 103 | 112 | 125 |
| Atreid... | Elfo d... | Bruja | Prueb... | 2 | 146 | 135 | 180 | 198 | 205 | 207 | 172 |
| Atreid... | Elfo d... | Bruja | NoEs... | 1 | 119 | 108 | 118 | 101 | 107 | 126 | 110 |
| Atreid... | Elfo d... | Bruja | Luch... | 1 | 131 | 104 | 128 | 129 | 110 | 114 | 126 |

VOLVER

(13) Pantalla Partidas Disponibles

| Black Temple | | | |
|--------------|----------------------------|------------------------|--------------------|
| Jugando | Partida | Ambientación | Nivel medio |
| X | noesunprueba | Cementerio de Dragones | 3.5 |
| X | Liberación de los elfos | Suramar | 40.666666666666664 |
| X | Lucha contra el Rey Exá... | Cementerio de Dragones | 19.166666666666668 |
| ✓ | Nuevas Alianzas | Zul'Drak | 4.0 |
| X | El despertar | Sueño Esmeralda | 1.0 |
| X | La conquista demoníaca | Argus | 5.5 |
| ✓ | Partida Prueba | Suramar | 1.0 |
| X | Test2 | Estepas de Tong Long | 2.4285714285714284 |
| ✓ | NoEsUnaPrueba | Zul'Drak | 12.15625 |
| X | Pochapartida | Suramar | 14.442307692307692 |
| ✓ | Prueba3 | Sueño Esmeralda | 1.5 |

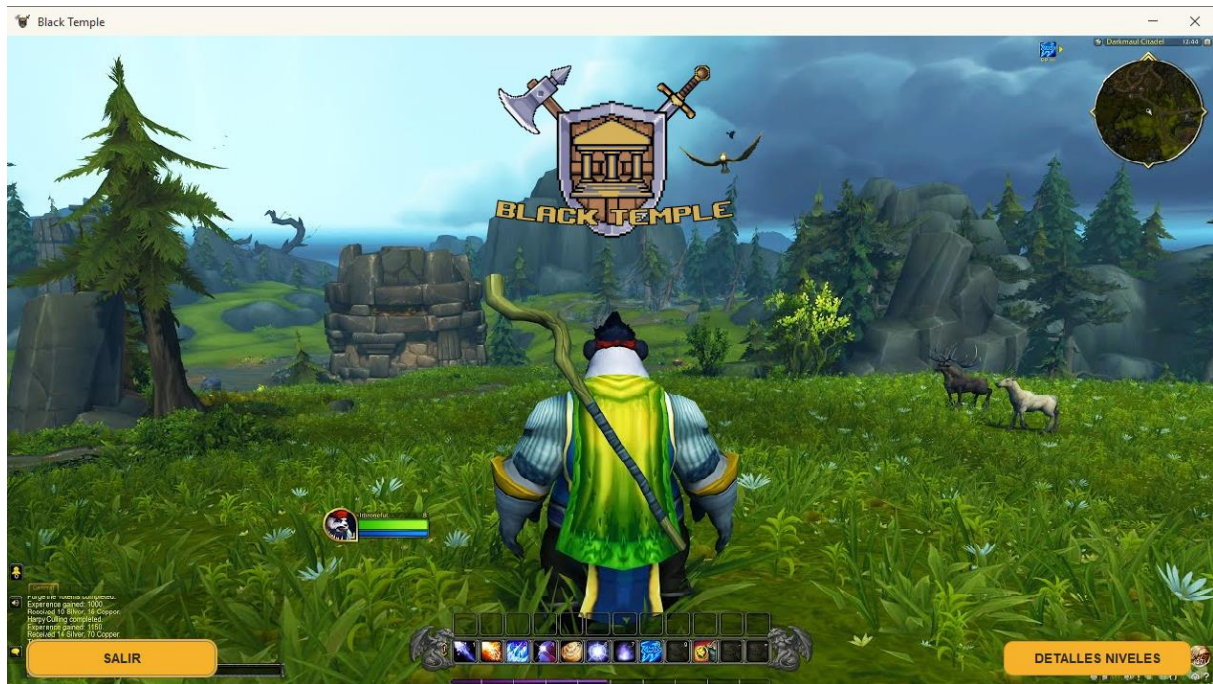
VOLVER **DETALLES** **UNIRSE**

(14) Pantalla selección de partida

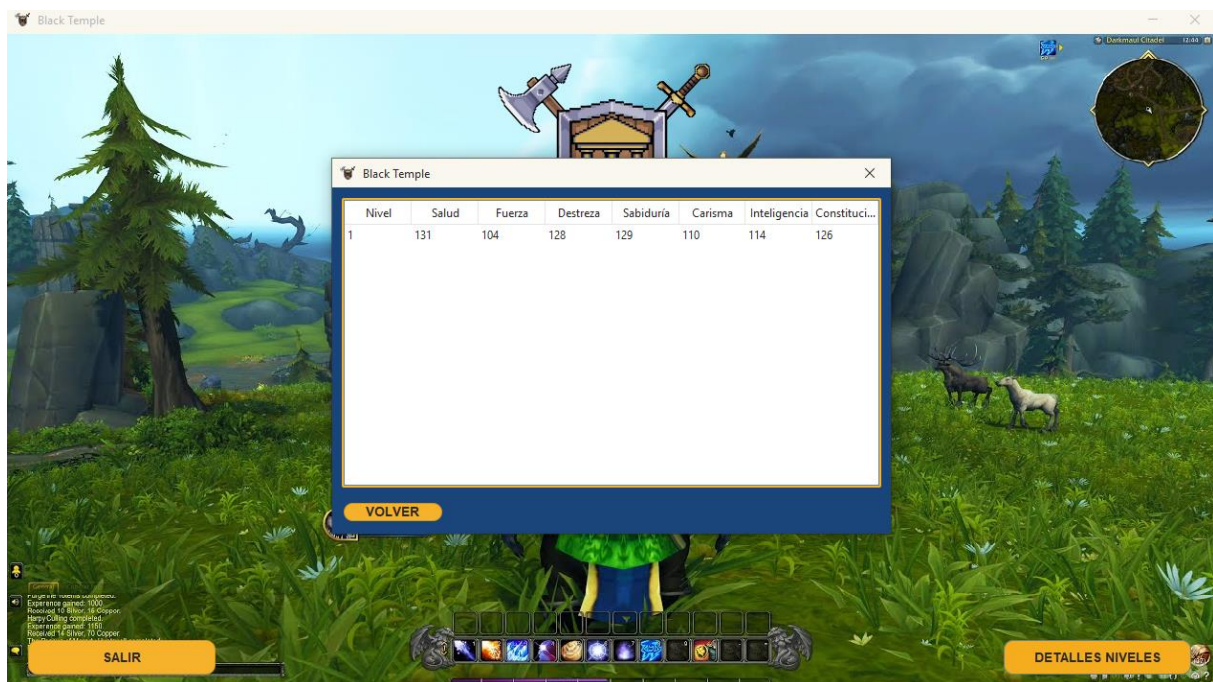
| Black Temple | | | | | | | | | | |
|--------------|----------|-----------|-------|-------|--------|----------|-----------|---------|------------|-----------|
| Nombre | Raza | Clase | Nivel | Salud | Fuerza | Destreza | Sabidu... | Carisma | Intelig... | Consti... |
| Padme | Huma... | Sacerd... | 35 | 15000 | 5000 | 7000 | 5000 | 7000 | 5000 | 400 |
| Anakin | Enano | Paladín | 40 | 20000 | 7000 | 5000 | 10000 | 5000 | 3000 | 500 |
| Ashoka | No m... | Pícara | 15 | 7500 | 2500 | 3000 | 3600 | 4000 | 7000 | 150 |
| Darth ... | Huarg... | Mago | 23 | 10000 | 3000 | 4000 | 1500 | 8000 | 3500 | 300 |
| Lobo | Huarg... | Paladín | 1 | 125 | 120 | 112 | 112 | 119 | 121 | 109 |

VOLVER

(15) Pantalla detalles de partidas



(16) Pantalla Unirse Partida



(17) Pantalla Detalles Niveles

Anexo Base de datos

Creación de los scripts de las tablas

```
1 DROP TABLE IF EXISTS juega;
2 DROP TABLE IF EXISTS partidas;
3 DROP TABLE IF EXISTS personajes;
4 DROP TABLE IF EXISTS miembros;
5
6 CREATE TABLE miembros (
7     id INTEGER PRIMARY KEY AUTOINCREMENT,
8     nombre TEXT NOT NULL,
9     apellidos TEXT NOT NULL,
10    password TEXT NOT NULL,
11    num_expediente INTEGER NOT NULL,
12    estudios TEXT NOT NULL
13 );
14
15 CREATE TABLE personajes (
16     id INTEGER PRIMARY KEY AUTOINCREMENT,
17     nombre TEXT NOT NULL,
18     sexo TEXT NOT NULL,
19     raza TEXT NOT NULL,
20     clase TEXT NOT NULL,
21     miembro_id INTEGER,
22     FOREIGN KEY (miembro_id) REFERENCES miembros(id)
23 );
24
25 CREATE TABLE partidas (
26     id INTEGER PRIMARY KEY AUTOINCREMENT,
27     nombre TEXT NOT NULL,
28     game_master_id INTEGER,
29     ambientacion TEXT NOT NULL,
30     esta_en_curso BOOLEAN NOT NULL,
31     num_sesion INTEGER NOT NULL,
32     duración INTEGER NOT NULL,
33     fecha TEXT NOT NULL,
34     FOREIGN KEY (game_master_id) REFERENCES miembros(id)
35 );
36
```

```
38 CREATE TABLE juega (  
39     nivel INTEGER NOT NULL,  
40     personaje_id INTEGER NOT NULL,  
41     partida_id INTEGER NOT NULL,  
42     salud INTEGER NOT NULL,  
43     fuerza INTEGER NOT NULL,  
44     destreza INTEGER NOT NULL,  
45     sabiduria INTEGER NOT NULL,  
46     carisma INTEGER NOT NULL,  
47     inteligencia INTEGER NOT NULL,  
48     constitucion INTEGER NOT NULL,  
49     PRIMARY KEY (nivel, personaje_id, partida_id),  
50     FOREIGN KEY (personaje_id) REFERENCES personajes(id),  
51     FOREIGN KEY (partida_id) REFERENCES partidas(id)  
52 );
```


Inserción de los datos para cada tabla

```

1  INSERT INTO miembros (nombre, apellidos, password, num_expediente, estudios)
2  VALUES
3  ('Gonzalo', 'Pozo Sánchez', 'lebron', '10001', 'SMIR'),
4  ('Daniel', 'Simón del Olmo', 'cochecitos', '10002', 'Superiores'),
5  ('Manuel', 'Gómez Navarro', '1234', '10003', 'ED. Infantil'),
6  ('Héctor', 'Chango Tapia', '1111', '10004', 'Superiores'),
7  ('Rubén', 'Peña González', 'fernandoalonso', '10005', 'SMIR'),
8  ('Cristiano', 'Ronaldo dos Santos Aveiro', 'siuuu', '10006', 'Escola');
9
10 INSERT INTO personajes (nombre, sexo, raza, clase, miembro_id)
11 VALUES
12 ('Gimli', 'Hombre', 'Enano', 'Guerrero', 1),
13 ('Atreides', 'Mujer', 'Elfo de la noche', 'Bruja', 2),
14 ('El Emisario', 'Hombre', 'Elfo de la noche', 'Cazador de demonios', 3),
15 ('Legolas', 'Hombre', 'Tauren', 'Cazador', 4),
16 ('Padme', 'Mujer', 'Humano', 'Sacerdotisa', 5),
17 ('Anakin', 'Hombre', 'Enano', 'Paladín', 6),
18 ('Ashoka', 'Mujer', 'No muerto', 'Pícara', 7),
19 ('Darth Maul', 'Hombre', 'Huargen', 'Mago', 8),
20 ('Joselito', 'Hombre', 'Pandaren', 'Chamán', 9),
21 ('LeBron James', 'Hombre', 'Tauren', 'Caballero de la muerte', 10),
22 ('Pepe el del Madrid', 'Hombre', 'Troll', 'Druida', 11),
23 ('D`Angelo Russell', 'Hombre', 'Dracthyr', 'Evocador', 12);

```

```

INSERT INTO partidas (nombre, game_master_id, ambientacion, esta_en_curso, num_sesion, duración, fecha)
VALUES
('La batalla por el templo del Tigre Blanco', 1, 'Estepas de Tong Long', false, 7, 315, '2023-08-02'),
('Liberación de los elfos', 2, 'Suramar', true, 2, 90, '2024-03-25'),
('Lucha contra el Rey Exánime', 3, 'Cementerio de Dragones', true, 3, 90, '2024-04-01'),
('Nuevas Alianzas', 4, 'Zul`Drak', false, 4, 90, '2024-09-23'),
('El despertar', 5, 'Sueño Esmeralda', false, 10, 120, '2024-06-01'),
('La conquista demoníaca', 6, 'Argus', true, 8, 180, '2024-02-22');

```

```

INSERT INTO juega
VALUES
(70, 1, 2, 50000, 10000, 1000, 1000, 3000, 1000, 450),
(50, 4, 2, 15000, 300, 10000, 1800, 800, 1750, 100),
(2, 3, 2, 150, 30, 100, 150, 300, 300, 450),
(7, 2, 4, 5000, 1500, 4500, 3300, 3700, 4500, 150),
(35, 5, 3, 15000, 5000, 7000, 5000, 7000, 5000, 400),
(40, 6, 3, 20000, 7000, 5000, 10000, 5000, 3000, 500),
(15, 7, 3, 7500, 2500, 3000, 3600, 4000, 7000, 150),
(23, 8, 3, 10000, 3000, 4000, 1500, 8000, 3500, 300);

```

Representación visual de cómo quedan las diferentes tablas

Tabla juega

| Rows: 16 | | | | | | | | | | | |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|---------------|--------------|-----|
| | nivel | per... | part... | salud | fuerza | destreza | sabiduria | carisma | intelligen... | constituc... | |
| | Filter... | Filter... | Filter... | Filter... | Filter... | Filter... | Filter... | Filter... | Filter... | Filter... | |
| 7 | 1 | 70 | 1 | 2 | 50000 | 10000 | 1000 | 1000 | 3000 | 1000 | 450 |
| | 2 | 50 | 4 | 2 | 15000 | 300 | 10000 | 1800 | 800 | 1750 | 100 |
| | 3 | 2 | 3 | 2 | 150 | 30 | 100 | 150 | 300 | 300 | 450 |
| | 4 | 7 | 2 | 4 | 5000 | 1500 | 4500 | 3300 | 3700 | 4500 | 150 |
| | 5 | 35 | 5 | 3 | 15000 | 5000 | 7000 | 5000 | 7000 | 5000 | 400 |
| | 6 | 40 | 6 | 3 | 20000 | 7000 | 5000 | 10000 | 5000 | 3000 | 500 |
| | 7 | 15 | 7 | 3 | 7500 | 2500 | 3000 | 3600 | 4000 | 7800 | 150 |
| | 8 | 23 | 8 | 3 | 10000 | 3000 | 4000 | 1500 | 8000 | 3500 | 300 |
| | 9 | 1 | 13 | 0 | 109 | 124 | 128 | 125 | 107 | 116 | 133 |
| | 10 | 1 | 14 | 0 | 105 | 117 | 108 | 137 | 122 | 136 | 116 |
| | 11 | 1 | 15 | 5 | 108 | 117 | 117 | 130 | 107 | 109 | 114 |
| | 12 | 1 | 15 | 4 | 131 | 105 | 124 | 100 | 105 | 134 | 124 |
| | 13 | 1 | 15 | 3 | 125 | 120 | 112 | 112 | 119 | 121 | 109 |
| | 14 | 1 | 15 | 1 | 103 | 120 | 102 | 127 | 115 | 114 | 128 |
| | 15 | 1 | 17 | 0 | 116 | 102 | 137 | 125 | 113 | 107 | 128 |
| | 16 | 1 | 15 | 6 | 131 | 104 | 104 | 127 | 132 | 127 | 129 |

Tabla Miembros

| Rows: 6 | | | | | | |
|---------|-----------|-----------|---------------------------|----------------|------------|--------------|
| | id | nombre | apellidos | password | num_exp... | estudios |
| | Filter... | Filter... | Filter... | Filter... | Filter... | Filter... |
| 7 | 1 | Gonzalo | Pozo Sánchez | lebron | 10001 | SMIR |
| | 2 | Daniel | Simón del Olmo | cochecitos | 10002 | Superiores |
| | 3 | Manuel | Gómez Navarro | 1234 | 10003 | ED. Infantil |
| | 4 | Héctor | Chango Tapia | 1111 | 10004 | Superiores |
| | 5 | Rubén | Peña González | fernandoalonso | 10005 | SMIR |
| | 6 | Cristiano | Ronaldo dos Santos Aveiro | siuuu | 10006 | Escola |

Tabla Personajes

Rows: 17

| | id | nombre | sexo | raza | clase | miem... |
|--|-----------|-----------------------|-----------|------------------|------------------------|-----------|
| | Filter... | Filter... | Filter... | Filter... | Filter... | Filter... |
| | 1 | 1 Gimli | Hombre | Enano | Guerrero | 1 |
| | 2 | 2 Atreides | Mujer | Elfo de la noche | Bruja | 2 |
| | 3 | 3 El Emisario | Hombre | Elfo de la noche | Cazador de demonios | 3 |
| | 4 | 4 Legolas | Hombre | Tauren | Cazador | 4 |
| | 5 | 5 Padme | Mujer | Humano | Sacerdotisa | 5 |
| | 6 | 6 Anakin | Hombre | Enano | Paladín | 6 |
| | 7 | 7 Ashoka | Mujer | No muerto | Pícara | 7 |
| | 8 | 8 Darth Maul | Hombre | Huargen | Mago | 8 |
| | 9 | 9 Joselito | Hombre | Pandaren | Chamán | 9 |
| | 10 | 10 LeBron James | Hombre | Tauren | Caballero de la muerte | 10 |
| | 11 | 11 Pepe el del Madrid | Hombre | Troll | Druida | 11 |
| | 12 | 12 D'Angelo Russell | Hombre | Dracthyr | Evocador | 12 |
| | 13 | 13 Ant | Hombre | No muerto | Mago | 1 |
| | 14 | 14 eeee | Hombre | Orco | Caballero de la muerte | 1 |
| | 15 | 15 Lobo | Hombre | Huargen | Paladín | 1 |
| | 16 | 16 eeee | Hombre | Orco | Caballero de la muerte | 1 |
| | 17 | 17 eeeeehji | Hombre | Orco | Chaman | 1 |

Tabla Partidas

Rows: 8

| | id | nombre | game... | ambientacion | esta_en_c... | num_sesi... | duración | fecha |
|--|-----------|---|-----------|------------------------|--------------|-------------|-----------|---------------------|
| | Filter... | Filter... | Filter... | Filter... | Filter... | Filter... | Filter... | Filter... |
| | 1 | 1 La batalla por el templo del Tigre Blanco | 1 | Estepas de Tong Long | 0 | 7 | 315 | 2023-08-02 |
| | 2 | 2 Liberación de los elfos | 2 | Suramar | 1 | 2 | 90 | 2024-03-25 |
| | 3 | 3 Lucha contra el Rey Exánime | 3 | Cementerio de Dragones | 1 | 3 | 90 | 2024-04-01 |
| | 4 | 4 Nuevas Alianzas | 4 | Zul'Drak | 0 | 4 | 90 | 2024-09-23 |
| | 5 | 5 El despertar | 5 | Sueño Esmeralda | 0 | 10 | 120 | 2024-06-01 |
| | 6 | 6 La conquista demoníaca | 6 | Argus | 1 | 8 | 180 | 2024-02-22 |
| | 7 | 7 Partida Prueba | 2 | Suramar | 0 | 0 | 0 | 2024-05-19 10:47:26 |
| | 8 | 8 Test2 | 2 | Estepas de Tong Long | 0 | 0 | 0 | 2024-05-19 10:50:50 |