

B&B
+ data code +

Índice

Objetivos generales del proyecto en los distintos módulos formativos.	4
Bases de datos.	4
Programación.	4
Entornos de Desarrollo.	4
Enlaces del proyecto en la nube.	5
Historial del proyecto.	6
Integrantes.	6
Primera Fase.	6
Especificaciones del Hardware.	10
Especificaciones para Eclipse.	10
Especificaciones para MySQL.	11
Especificaciones del Software.	12
Características Generales.	12
Herramientas Utilizadas.	13
Eclipse.	13
MySQL.	13
Trello.	13
GitHub.	13
Programas necesarios para la realización.	13
Planificación detallada de los requisitos de la aplicación.	14
Requisitos mínimos.	15
Modelo E-R.	16
Diseño de la Interfaz de las ventanas.	17
Ventana “Inicio Sesión”.	17
Ventana GameMaster.	18
Menú Superior.	18
Ventana Menú Principal.	19
Ventana Mostrar Personajes.	19
Ventana Nuevo Personaje.	20
Ventana Modificar Personaje.	20
Ventana Crear Partida.	21
Ventana Editar Partida.	21
Ventana Consultar Partida.	22

Ventana Miembros del Club.....	22
Modelo Relacional y Normalización.....	23
Modelo Relacional.....	23
Normalización.....	24
Diseño del logo.....	26
Colores.....	26
Explicación del logo.....	27
Diagrama de casos de uso.....	28

Objetivos generales del proyecto en los distintos módulos formativos.

Bases de datos.

- Crear bases de datos definiendo su estructura y las características de sus elementos según el modelo relacional.
- Diseñar modelos lógicos normalizados interpretando diagramas entidad/relación.
- Realizar el diseño físico de bases de datos utilizando asistentes, herramientas gráficas y el lenguaje de definición de datos.
- Consultar y modificar la información almacenada utilizando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.

Programación.

- Realizar el diseño de clases necesarias para seguir el patrón MVC.
- Implementar cada una de las clases para lograr su objetivo: clases de interfaz gráfica pertenecientes a la vista, clases que representen el modelo y clases encargadas de la lógica de la aplicación pertenecientes al control.
- Desarrollar una aplicación que gestione información almacenada en bases de datos relacionales identificando y utilizando mecanismos de conexión.

Entornos de Desarrollo.

- Realizar el análisis y el diseño de cualquier aplicación empleando técnicas UML.
- Documentar aplicaciones.
- Gestionar las diferentes versiones de un software y el trabajo colaborativo.
- Realizar pruebas de testeo sobre los programas.
- Seguir metodologías ágiles para el desarrollo y planificación de programas.

Enlaces del proyecto en la nube.

En dicho apartado encontraras los enlaces que te llevaran al proyecto para el cual se utilizó GitHub, además se incluye la organización de los sprint en Trello (en el apartado de especificaciones del software podrás encontrar que es GitHub y Trello).

- **GitHub:**

<https://github.com/danielfariasmo/DataCode->

- **Trello:**

[https://trello.com/invite/b/uKghl6wk/ATTl9ba53f18fd762247954989021efac6c16A65D15E/
datacode](https://trello.com/invite/b/uKghl6wk/ATTl9ba53f18fd762247954989021efac6c16A65D15E/datacode)

Historial del proyecto.

En el año 2024, un grupo de estudiantes de la Universidad Europea de Madrid en el área de desarrolladores web se unieron para crear un proyecto con el interés educativo y en poder unir todos los conocimientos adquiridos en su primer año de la formación profesional, integrando los módulos de programación, bases de datos y entornos de desarrollo. En dicho proyecto se basa en simular una partida de rol en el entorno gráfico de Java, además de creación de las tablas correspondientes para almacenar la información de los usuarios en el entorno de MySQL y, por último, integrar la herramienta de Git para poder compartir entre los mismos integrantes los fragmentos de códigos y documentación necesaria. El nombre del proyecto fue denominado "DataCode++", en español significa "Código de Datos", de esta forma es fácil de intuir que se unirán lenguajes de programación junto a bases de datos externas.

En este apartado se encuentran las fases del proyecto, es decir, un historial de este. Podrás visualizar como se organizaron los integrantes del grupo para llevar a cabo todos los parámetros que se indicaron en cada módulo de la universidad. Cabe destacar, que lo primero que encontrarás, será quienes forman parte del equipo DataCode++, y luego cada fase del proyecto. Para mayor comprensión, cuando se mencione al cliente se hará referencia a los módulos de la universidad (Programación, Bases de Datos y Entornos de Desarrollo).

Integrantes.

1. Daniel Farías.
2. Ignacio Moreno.
3. Daniel González.

Primera Fase.

La primera fase del proyecto se centró en la comprensión de los parámetros impuestos por los clientes y ver las necesidades de estos.

Los puntos de esta primera fase fueron:

- Creación de grupos en redes sociales para poder compartir información o puntos de interés de cada integrante, se utilizó un grupo de WhatsApp y un grupo en Discord para compartir información y poder realizar reuniones en videollamada.
- Investigar las especificaciones del hardware como del software que se necesitan para desarrollar el proyecto, teniendo en cuenta lo mínimo que se debe tener en el ordenador.
- Desarrollo del estudio y estructura de modelo entidad-relación con dichos parámetros, con la finalidad de poder crear las tablas necesarias para el programa y no divagar en tablas sin sentido.
- Creación de un proyecto en la nube de GitHub para que dichos integrantes puedan trabajar en conjunto, pero a su vez individualmente en el proyecto.
- Realización de una estructura de objetivos que se deben alcanzar mediante un plazo de tiempo determinado, para ello se utilizó la aplicación de Trello para crear una organización adecuada. En el apartado de los sprint podrás visualizar dichos objetivos.

Sprint del Proyecto.

El presente proyecto de DataCode++ se realizará en base a cuatro (4) sprint y una fase final. En este apartado podrás visualizar de qué manera se han organizado los sprint y las fechas en las que se debe cumplir los objetivos establecidos en el sprint.

1. Sprint #1: 18 de marzo – 31 marzo.

- Análisis y diseño de los requisitos hardware y software que tendrá el nuevo sistema.
 - Planificación detallada de los requisitos de la aplicación.
- Análisis de las especificaciones del proyecto.
- Proyecto en GitHub. Planificación con Trello.
- Análisis de las especificaciones del proyecto. Diagrama E/R.

2. Sprint #2: 1 de abril – 14 abril.

- Creación del proyecto.
 - Diseño de la interfaz de las ventanas.
 - Creación de las clases pertenecientes a la Vista.
- Generación del Modelo Relacional.
 - Normalización
- Creación de la BBDD: tablas, índices, etc.
- Inserción de los datos necesarios para la aplicación.
- Diagrama de casos de uso.
- Diseño del logo.

3. Sprint #3: 15 de abril – 28 abril.

- Generación del diagrama de clases.
- A partir del diagrama de clases:
 - Creación de las clases pertenecientes al Modelo.
 - Creación de las clases pertenecientes al Control encargadas del acceso a BBDD y del manejo de la aplicación.

4. Sprint #4: 29 de abril – 12 mayo.

- Refinamiento del programa.
- Realización de pruebas JUnit.
 - Documentación de la aplicación JavaDoc.
- Integración de todos los elementos y solución de las incidencias de integración que se produzcan.
- Realización del manual de usuario en la wiki de GitHub.

5. Preparación release: 13 de mayo – 20 mayo

- Elaboración de la presentación.
- Revisión de la documentación, tanto JavaDoc como manual de usuario.

Especificaciones del Hardware.

Para el desarrollo del proyecto, se ha optado por utilizar las herramientas y aplicaciones integradas en el entorno Eclipse, que ofrece una plataforma sólida y versátil para el diseño y programación de la aplicación. Además, se ha empleado MySQL como sistema de gestión de bases de datos, aprovechando su eficiencia para el manejo de grandes volúmenes de datos.

Especificaciones para Eclipse.

Como se menciona en el apartado de especificaciones del software se utilizará la herramienta de Eclipse para la elaboración de la aplicación. Esta elección permite diseñar una interfaz dinámica, facilitando la interacción del usuario con la aplicación y mejorando la experiencia de usuario final.

Eclipse proporciona herramientas integradas para el desarrollo de código, depuración y prueba, optimizando así el flujo de trabajo y permitiendo la implementación de funcionalidades clave de la aplicación. Además, su amplia variedad de plugins ofrece la flexibilidad de incorporar funcionalidades adicionales y personalizar el entorno de desarrollo según nuestras necesidades específicas.

Los requisitos mínimos y recomendados para el uso de Eclipse son los siguientes:

Requisitos Mínimos

- SO: XP
- Procesador: Core 2 Duo.
- Memoria: 2 GB de RAM
- Gráficos: 9800 GT, o superior
- Almacenamiento: 1 GB de espacio disponible

Requisitos Recomendados

- SO: XP
- Procesador: Core i5
- Memoria: 4 GB de RAM
- Gráficos: 280 GTX
- Red: Conexión de banda ancha a Internet

- Almacenamiento: 2 GB de espacio disponible

Especificaciones para MySQL.

De igual forma, para el almacenamiento y gestión de datos de la aplicación, se ha integrado MySQL como sistema de gestión de bases de datos. Ya que ofrece una solución fiable y escalable para manejar grandes volúmenes de datos de manera eficiente, garantizando la integridad y seguridad de la información almacenada.

Los requerimientos básicos para el uso de MySQL son los siguientes:

- 512 Mb de memoria RAM.
- 1024 Mb máquina virtual.
- 1 GB de espacio de disco duro.
- Sistema operativo: Windows, Linux y Unix.
- Arquitectura del sistema 32/64 bit.
- Protocolo de red TCP/IP.

Es importante señalar que, si las especificaciones del equipo son superiores a las indicadas, se podrá hacer uso de MySQL sin ningún tipo de problema e incluso mejorando el rendimiento del sistema.

Especificaciones del Software.

Para el análisis y diseño del software del proyecto, así como su planificación detallada, es necesario seguir unas pautas y un proceso estructurado en referencia al software necesario:

Características Generales.

1. Gestión de usuarios:

- La aplicación debe permitir a los usuarios registrarse e iniciar sesión con un usuario y contraseña.
- Es importante destacar entre usuarios “jugadores” y “usuarios” GameMaster (GM), ya que cada rol, llevará unos permisos específicos y diferentes.

2. Gestión de Personajes:

- Crear, Leer, Actualizar, y Eliminar personajes.
- Cada personaje debe tener un identificador único, nombre, jugador asociado, raza, clase, nivel de experiencia y características como fuerza, destreza, constitución, inteligencia, sabiduría y carisma.

3. Gestión de partidas:

- Cada partida debe tener al igual que los personajes, un identificador único, nombre, Game Master asociado, ambientación, lista de personajes-participantes, día y hora del juego, duración de la sesión, número de sesión y estado de la partida (en curso/ terminada)

4. Interfaz de Usuario:

- Pantalla de inicio de sesión.
- Menú con opciones de consultas, altas, bajas y modificación para los usuarios.
- Permisos diferenciados para los distintos usuarios:
 - Los jugadores pueden editar y eliminar sus propios personajes, crear nuevos personajes y consultar cualquier personaje en el sistema.
 - Los jugadores pueden consultar cualquier partida, pero no editarla
 -

- Los GM pueden consultar cualquier partida y editar todas aquellas que esté dirigiendo
- Los GM pueden crear nuevas partidas

Herramientas Utilizadas.

Eclipse.

Es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores

MySQL.

Es un sistema de administración de bases de datos relacionales. Es un software de código abierto desarrollado por Oracle. Se considera como la base de datos de código abierto más utilizada en el mundo.

Trello.

Es una herramienta flexible para la gestión del trabajo, con la que los equipos pueden diseñar planes, colaborar en proyectos, organizar flujos de trabajo y hacer un seguimiento del progreso de una manera visual, productiva y gratificante

GitHub.

Es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc.

- -MySQL: Para la gestión de la base de datos del sistema
- -GitHub: Para la gestión de versiones del código fuente y colaboración en el desarrollo del software

Programas necesarios para la realización.

1. Entornos de desarrollo integrado (IDE):

Se utilizará Eclipse como el IDE principal para el desarrollo del proyecto (necesario tener instaladas las últimas versiones de JDK de Java y las herramientas necesarias para

el desarrollo en Java). Además, eclipse es ideal para el desarrollo del software, incluyendo la estructura del código y la gestión de versiones.

2. Gestión de proyectos:

Se utilizará Trello para la gestión de proyectos, seguimiento de tareas y colaboración en equipo. También como herramienta principal para la planificación y seguir con un orden en el desarrollo y progreso de los objetivos.

3. Gestión de versiones:

Git y GitHub serán usados para el control de versiones del código fuente y para la colaboración mediante un repositorio para el desarrollo grupal del proyecto. Además, de la colaboración entre los integrantes del grupo.

4. Gestión de base de datos:

Para la gestión de bases de datos se implementará la herramienta de MySQL como sistema para la gestión y procesamiento de datos, para el almacenamiento y para la manipulación de datos del sistema.

Planificación detallada de los requisitos de la aplicación.

Fase 1: Investigación y planificación.

- Definir el alcance del proyecto y los requisitos que este tiene.

Fase 2: Diseño y Arquitectura.

- Diseñar la arquitectura del sistema, incluyendo la estructura de la base de datos, la arquitectura del software y la interfaz del usuario.

Fase 3: Desarrollo.

- Desarrollar código fuente de la aplicación, utilizando Java en el entorno de Eclipse.
- Diseñar y crear la base de datos mediante MySQL.
- Utilizar Git para el control de versiones y colaboración en el desarrollo del código.

Fase 4: Pruebas.

- Realizar pruebas unitarias y de integración para garantizar el funcionamiento del sistema.

- Realizar pruebas de rendimiento y seguridad.

Requisitos mínimos.

1. Requisitos del sistema:

- Debe ser accesible desde cualquier dispositivo con conexión a internet.
- Compatible con los principales navegadores web como Chrome, Firefox...

2. Requisitos de interfaz de usuario:

- Necesitaremos una pantalla para el inicio de sesión con validación de usuario y contraseña.
- Menú adaptado al rol de cada usuario (jugador/GM).
- Formularios para ingresar información de nuevos personajes y partidas.
- Funcionalidad de búsqueda y visualización de personajes.

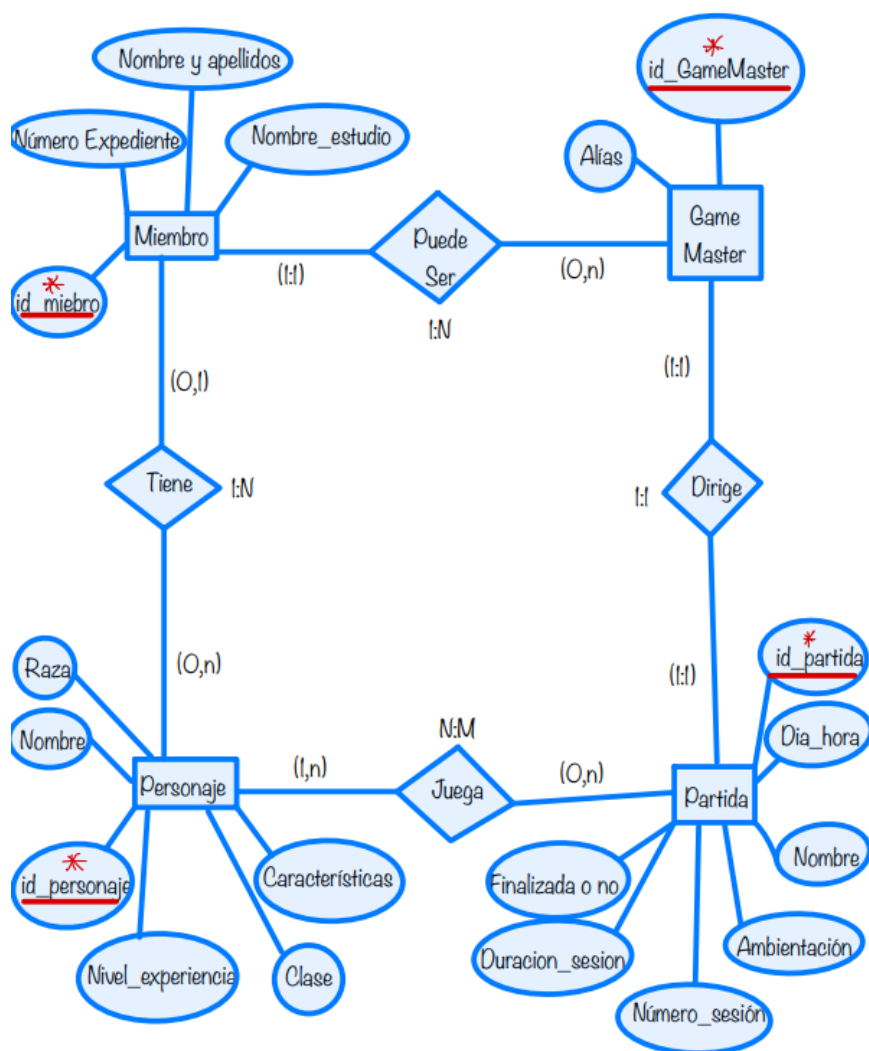
3. Requisitos de seguridad:

- Encriptación de contraseñas.
- Control de acceso basado en roles para que cada usuario sólo pueda acceder a sus funcionalidades permitidas en función de su rol.

4. Requisitos de funcionalidad:

- Validación de los datos de entrada para evitar errores y garantizar los datos.
- Notificaciones en los cambios de la partida o personajes.

Modelo E-R.



En base al juego de rol se ha realizado la creación de un modelo entidad relación para el comienzo del proyecto, en el cual tenemos cuatro (4) relaciones.

1. **Miembro:** el cual representa cual quiere individuo que participe en el juego.
2. **Personaje:** es la representación de un miembro en el juego.
3. **Game Máster:** es un miembro del juego y su rol es dirigir una partida de rol.
4. **Partida:** realizara la acción principal del juego, puede tener múltiples personajes.

Diseño de la Interfaz de las ventanas.

En la aplicación de DataCode++, se ha creado diez (10) interfaces gráficas fundamentales que facilitarán la ejecución del programa. Estas interfaces solamente son el primer diseño, es decir, puede que resulte no ser el resultado final. A continuación, se detallará el contenido y las funciones disponibles en cada una de estas ventanas.

Ventana “Inicio Sesión”.

Al ejecutar el programa, lo primero que se encuentra es el inicio de sesión, donde el usuario puede interactuar para ingresar con su nombre de usuario y contraseña. Además, puede seleccionar entre las opciones "Jugador" y "Game Master" para acceder a la aplicación, cuando el usuario pulse en el botón de “Entrar”. Dependiendo del tipo de usuario seleccionado, las opciones disponibles en las ventanas siguientes variarán.



Ventana GameMaster.

En la ventana de inicio de sesión la aplicación te preguntara como deseas entrar como usuario o como GameMaster. Pero si no tienes un alias creado como GameMaster el sistema te mostrara una ventana nueva en la cual podrás rellenar con tu alias favorito.



Menú Superior.

El menú exhibe todas las opciones disponibles con el que el usuario o GameMaster puede interactuar, en ella solo se verán tres (3) opciones, de las cuales son desplegables en la interfaz. Cada una de estas opciones dirige al usuario a la ventana correspondiente, permitiéndole ejecutar la función deseada.



Ventana Menú Principal.

En la ventana del menú principal, se encuentra una breve introducción al juego, además aparecerá el logo en la parte izquierda de la pantalla, y también el menú antes mencionadas anteriormente.




Ventana Mostrar Personajes.

En esta ventana el usuario o GameMaster podrá visualizar un listado con los personajes que tiene en el juego. Además se incluyen tres botones, eliminar, aceptar y consultar las stats, de los personajes.




Ventana Nuevo Personaje.

En ella, como su nombre lo dice, podrás crear nuevos personajes. Para ello la venta te indica que datos necesitara para crear el personaje. Simplemente se deberán rellenar cada campo con la información que el sistema indique y presionando el botón de aceptar, se crea el personaje.



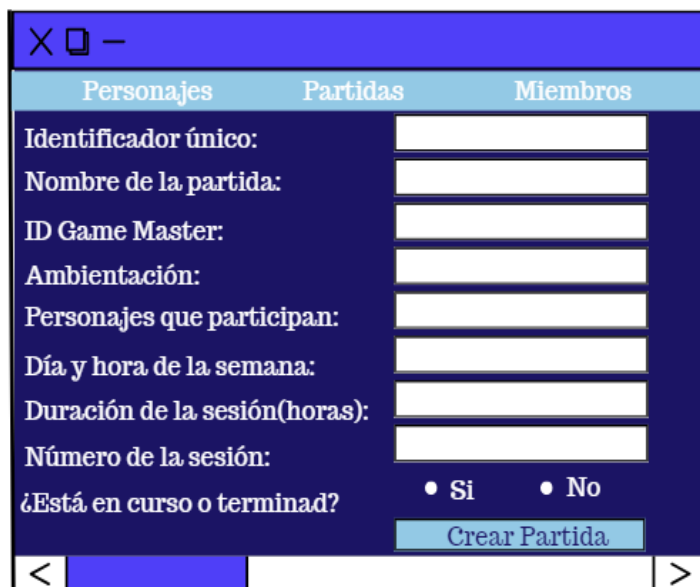
Ventana Modificar Personaje.

Para esta ventana, se podrá modificar las características de los personajes, si es que en algún momento el usuario se ha equivocado a la hora de introducir la información requerida para la creación de un personaje o, si el usuario desea modificar algún campo en específico.



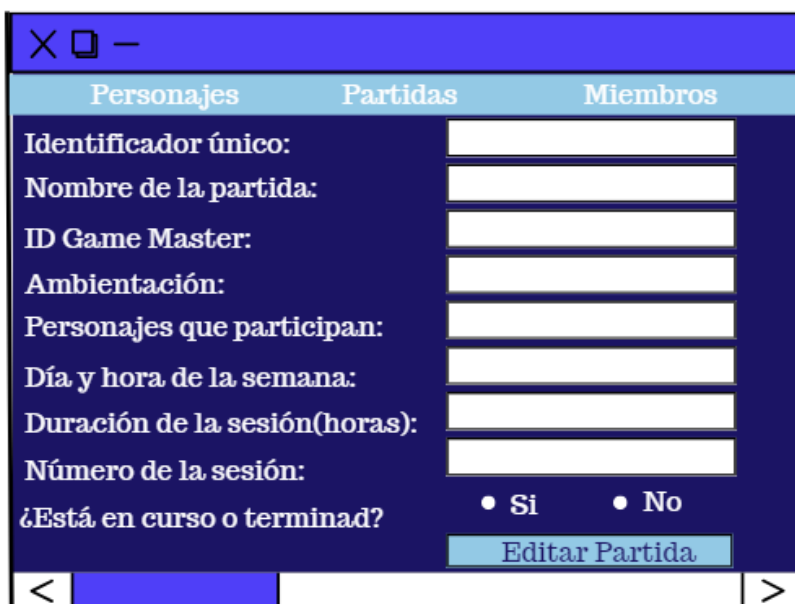
Ventana Crear Partida.

El usuario podrá ver la pantalla de “Crear Partida”, pero solamente el GameMaster será el encargado de rellenar los campos necesarios para que esa partida se cree. Simplemente es rellenar los datos y presionar el botón de “Crear Partida”.



Ventana Editar Partida.

De una manera similar a la ventana anterior, en esta ventana se podrá editar las partidas. Es importante recordar, que solo podrá rellenar los campos el GameMaster.



Venta Consultar Partida.

Y la última opción en partida es la de “Consultar Partida”, en esta ventana se visualizará un listado con todas las partidas, incluyendo toda la información relevante que lleva una partida. Para esta opción se puede visualizar tanto el usuario común como el GameMaster.

Identificador	Nombre	Día	Nº Sesión	Ambientación	Finalizada (Si/No)	Duración	Id Game Master
1	La búsqueda del anillo	2024-04-01	1	Tierra Media	No	4	99
2	Galaxia en guerra	2024-04-02	1	Galaxia	No	3	99
3	Las sombras de la montaña	2024-04-03	1	Montañas	No	5	99
4	La magia perdida	2024-04-04	1	Bosque encantado	No	4	99
5	El rescate de la princesa	2024-04-05	1	Castillo	No	3	99
6	Intriga en la corte	2024-04-06	1	Palacio real	No	5	99
7	La ciudad maldita	2024-04-07	1	Ciudad abandonada	No	4	99
8	El laberinto de la muerte	2024-04-08	1	Laberinto	No	6	99

Ventana Miembros del Club.

En ella se muestra en el lado derecho un listado con todos los miembros que hacen parte de la aplicación, y en la parte izquierda un cartel con famosos que también juegan rol.

Jugadores de Rol Famosos		Miembros
		Nacho Moreno
		Sara Villanueva
		Daniel Farías
		Daniel González

Modelo Relacional y Normalización.

Modelo Relacional.

En base al modelo entidad-relación se debe crear el modelo relacional para poder comenzar con la creación de las tablas de la base de datos. Para ello tendremos cinco (5) tablas, de las cuales cuatro (4) son de las entidades y una es de una relación. Las tablas con sus atributos son los siguientes:

Tabla Miembro.

- id_miembro. (Clave Primaria).
- Nombre y apellidos.
- Número_Expediente.
- Nombre_Estudio

Tabla GameMaster.

- id_GameMaster. (Clave Primaria).
- Alias.
- *id_miembro* (Clave foránea, referencia a la tabla Miembro).

Tabla Personaje.

- id_personaje. (Clave Primaria).
- Nombre.
- Raza
- Nivel_experiencia
- Clase.
- *id_miembro* (Clave foránea, referencia a la tabla Miembro).

Tabla Partida.

- id_partida. (Clave Primaria).
- Nombre.
- Dia_hora.
- Número_sesión.

- Ambientación.
- Finalizada.
- Duración_sesión.
- *id_GameMaster* (Clave foránea, referencia a la tabla GameMaster).

Tabla Juega.

- id_personaje. (Clave Primaria).
- id_partida. (Clave Primaria).
- Caracteristicas_personaje.

En esta última tabla obtenemos dos claves primarias, por lo cual hace referencia a distintas tablas, las cuales son la tabla personaje que alberga la clave foránea id_personaje, y la tabla partida que contiene la clave foránea id_partida.

Normalización.

En el modelo relacional que se ha descrito, existen distintas razones por las cuales se ha optado por no normalizar, esas razones son:

1. Normalizar estas tablas podría resultar un diseño más complejo y consultas más complicadas, lo que podría afectar el rendimiento del sistema y la facilidad de uso para los usuarios.
2. Una estructura de tablas menos normalizada puede ser más fácil de entender y mantener, especialmente para usuarios menos familiarizados con el modelo relacional de base de datos. Esto puede ser importante si la prioridad es la facilidad de uso.

La normalización es una técnica valiosa para garantizar la integridad y eficiencia de una base de datos. De igual manera explicaremos por que no es necesario realizar las tres (3) primeras formas de normalización.

1. Primera forma (1 FN): esta primera forma indica que cada columna de la tabla debe contener un único valor y no múltiples valores. En dichas tablas no hay columnas que puedan albergar más de un valor.
2. Segunda forma (2 FN): la segunda forma de normalización establece que una tabla debe estar en 1FN y que cada columna que no sea clave solo dependa de la clave primaria y no de otras columnas adicionales. Cada tabla tiene una clave primaria y

todas las columnas no clave están relacionadas funcionalmente con la clave primaria respectiva.

3. Tercera forma (3 FN): en ella establece que una tabla debe estar en 2FN y que no debe haber dependencias transitivas entre las columnas no clave, es decir, que, si una columna depende de otra que a su vez depende de la clave primaria, entonces la columna debería trasladarse a otra tabla.

Diseño del logo.



Colores.

1. [Azul Oscuro](#): Suele asociarse con la confianza, la profesionalidad y la seriedad. Puede transmitir estabilidad y fiabilidad, lo que es importante para muchas empresas.
2. [Azul Celeste](#): Este tono de azul más claro puede evocar calma, tranquilidad y frescura. También puede sugerir una sensación de apertura y claridad.
3. [Blanco](#): El blanco a menudo se asocia con la pureza, la limpieza y la simplicidad. También puede sugerir espacio, claridad y neutralidad.

En conjunto, esta combinación de colores podría transmitir un mensaje de confianza y profesionalismo (azul oscuro), junto con una sensación de calma y frescura (azul celeste) y una imagen de limpieza y claridad (blanco). Dependiendo del tipo de negocio y su público objetivo, esta combinación de colores podría ser adecuada para transmitir una variedad de mensajes, desde la fiabilidad hasta la modernidad o la frescura. Es importante considerar el contexto y la audiencia específica al interpretar el significado de los colores en un logo comercial.

Explicación del logo.

"DataCode++" como nombre del proyecto integrador, que abarca los módulos de programación, bases de datos y entornos de desarrollo, sugiere una combinación de dos elementos principales: "Data" y "Code".

1. Data (Datos): Este término hace referencia a la información que se maneja en el proyecto, lo que sugiere un enfoque en la gestión, manipulación y almacenamiento de datos. Esto podría implicar el uso de bases de datos para almacenar información de manera estructurada y accesible.
2. Code (Código): Este término implica una relación con la programación y el desarrollo de software. Sugiere que el proyecto involucra la creación y manipulación de código informático, lo que puede abarcar desde la escritura de aplicaciones hasta la implementación de algoritmos y la creación de interfaces de usuario.

El proyecto "DataCode" sugiere que se centra en la intersección entre la gestión de datos y el desarrollo de software. Esto podría implicar la creación de aplicaciones que manipulan datos de diversas fuentes, el diseño de sistemas de gestión de bases de datos, o el desarrollo de herramientas para análisis y visualización de datos.

En resumen, el nombre "DataCode" transmite la idea de un proyecto que combina la manipulación de datos con la programación y el desarrollo de software.

Diagrama de casos de uso.

Se ha realizado el diagrama de casos de uso, en función a las diferentes acciones que pueden realizar los personajes durante el transcurso del juego de rol. Existen dos formas de poder interactuar con la aplicación y son:

1. Como Jugador, que el mismo puede crear personajes, consultar miembros y consultar partidas.
2. Como GameMaster, que puede crear personajes, al igual que el Jugador, que, a su vez, puede mostrar personajes y modificar dichos personajes (indicado con relación "include" en el diagrama").

Tanto Jugador como GameMaster, además de poder crear personajes (lo que incluye mostrar y modificar dichos personajes) pueden iniciar sesión. Todas las acciones están incluidas en Iniciar sesión, ya que, para realizar cualquier acción, primero debemos iniciar sesión, es decir, si no iniciamos sesión en el juego, no podemos realizar nada más, como por ejemplo crear partidas o editarlas.

