



**DataCode++**

## Índice

<b>Introducción .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Objetivos generales del proyecto en los distintos módulos formativos. ....</b>	<b>3</b>
Bases de datos.....	3
Programación. ....	3
Entornos de Desarrollo.....	3
<b>Enlaces del proyecto en la nube. ....</b>	<b>4</b>
<b>Historial del proyecto. ....</b>	<b>5</b>
Integrantes.....	5
Primera Fase.....	5
<b>Especificaciones del Hardware. ....</b>	<b>9</b>
Especificaciones para Eclipse. ....	9
Especificaciones para MySQL.....	10
<b>Especificaciones del Software. ....</b>	<b>11</b>
Características Generales. ....	11
Herramientas Utilizadas. ....	12
Eclipse.....	12
MySQL. ....	12
Trello. ....	12
GitHub. ....	12
Programas necesarios para la realización.....	12
Planificación detallada de los requisitos de la aplicación. ....	13
Requisitos mínimos. ....	13
<b>Modelo E-R. ....</b>	<b>15</b>
<b>Bibliografía.....</b>	<b>¡Error! Marcador no definido.</b>
<b>Conclusión .....</b>	<b>¡Error! Marcador no definido.</b>

## **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/ATTI9ba53f18fd762247954989021efac6c16A65D15E/  
datacode](https://trello.com/invite/b/uKghl6wk/ATTI9ba53f18fd762247954989021efac6c16A65D15E/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.

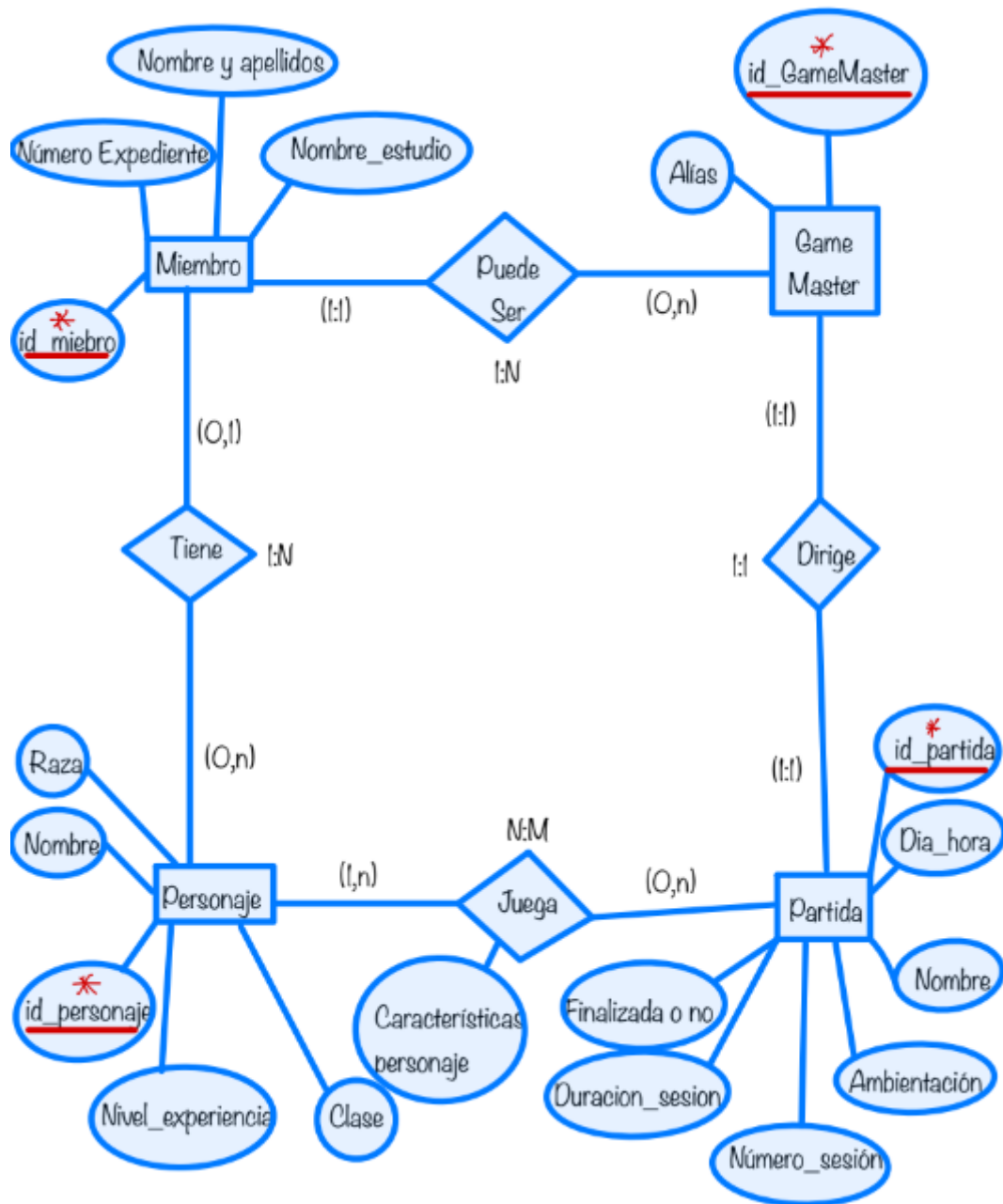
## 3. Requisitos de seguridad:

- 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.