# **Diagrama Descripción generada automáticamente**

Contenido

[1](#_Toc162287440)

[1. Introducción 2](#_Toc162287441)

[**Proyecto:** Desarrollo de aplicación de Gestión de Personajes de Rol en Java y SQLite. 2](#_Toc162287442)

[**Fecha:** 18 de marzo – 12 de mayo de 2024 2](#_Toc162287443)

[**Equipo del Proyecto:** Gonzalo Pozo, Daniel Simón, Manuel Gómez y Héctor Chango. 2](#_Toc162287444)

[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 sus características, clase y raza. Para la realización del proyecto utilizaremos Java y SQLite, y metodologías para gestionar eficazmente el desarrollo del software, como Scrum. 2](#_Toc162287445)

[2. Requisitos Hardware y Software 3](#_Toc162287446)

[3. DIAGRAMA ENTIDAD – RELACIÓN 3](#_Toc162287447)

[4. Cronograma de Actividades SPRINT 1 🡪 18/03/2024 – 31/03/2024 4](#_Toc162287448)

## Introducción

# **Proyecto:** Desarrollo de aplicación de Gestión de Personajes de Rol en Java y SQLite.

# **Fecha:** 18 de marzo – 12 de mayo de 2024

# **Equipo del Proyecto:** Gonzalo Pozo, Daniel Simón, Manuel Gómez y Héctor Chango.

# 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 sus características, clase y raza. Para la realización del proyecto utilizaremos Java y SQLite, y metodologías para gestionar eficazmente el desarrollo del software, como Scrum.

## Requisitos Hardware y Software

**REQUISITOS PARA WINDOWS**

|  |  |  |
| --- | --- | --- |
| **Requisitos** | **Mínimo** | **Recomendado** |
| SO | Windows Vista o superior | Windows 10 |
| Procesador | 2 GHz | 4 GHz |
| Memoria | 2 GB RAM | 4 GB RAM |
| Gráficos | 256 MB de memoria de video, shader model 3.0+ | 512 MB de memoria de video, shader model 5.0+ |
| DirectX | Versión 10 | Versión 11 o superior |
| Almacenamiento | 500 MB de espacio disponible | 1 GB de espacio disponible |

**REQUISITOS PARA macOS**

|  |  |  |
| --- | --- | --- |
| **Requisitos** | **Mínimo** | **Recomendado** |
| SO | Mac OSX 10.10+ | macOS 10.14+ |
| Procesador | 2 GHz | 4 GHz |
| Memoria | 2 GB RAM | 4 GB RAM |
| Gráficos | 256 MB de memoria de video, OpenGL 2 | 512 MB de memoria de video, OpenGL 3.3 |
| Almacenamiento | 500 MB de espacio disponible | 1 GB de espacio disponible |

**REQUISITOS PARA LINUX**

|  |  |  |
| --- | --- | --- |
| **Requisitos** | **Mínimo** | **Recomendado** |
| SO | Ubuntu 12.04 LTS | Ubuntu 20.04 LTS |
| Procesador | 2 GHz | 4 GHz |
| Memoria | 2 GB RAM | 4 GB RAM |
| Gráficos | 256 MB de memoria de video, OpenGL 2 | 512 MB de memoria de video, OpenGL 3.3 |
| Almacenamiento | 500 MB de espacio disponible | 1 GB de espacio disponible |

## 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.

// INSERTAR FOTOOOOOO

## Cronograma de Actividades SPRINT 1 🡪 18/03/2024 – 31/03/2024

**18/03/2024**

* Creamos el repositorio en GitHub.
* Realizamos el tablero de tareas en Trello.
* Comenzamos con el diagrama de entidad-relación.

**19/03/2024**

* Creamos el proyecto de JAVA y lo subimos a la GitHub, con un primer commit.
* Terminamos la versión 0.1 del diagrama de entidad relación.
* Añadimos un segundo commit con la primera versión del diagrama terminado.

**22/03/2024**

* Actualizamos el diagrama entidad relación e hicimos un commit con la versión 1.0.
* Iniciamos la creación de los requisitos de la aplicación.

**25/03/2024**

* Actualizamos el diagrama entidad relación e hicimos un nuevo commit con la versión 1.1.
* Finalizamos la creación de una primera fase de requisitos de la aplicación, que puede actualizarse a medida que avance el proyecto.