**Documento de Especificación de Requerimientos para el Sistema de Base de Datos: Steam - Área de Juegos**

**Sistema**: Base de Datos para la Gestión de Juegos en Steam  
**Fecha**: 20/11/2024  
**Elaborado por**: Marco Montaner (Analista B.D)

### **1. Recopilación y Análisis de Requisitos**

**1.1 Introducción**El proyecto tiene como objetivo desarrollar una base de datos que centralice la gestión de juegos y sus datos asociados en Steam. Esto incluye información sobre títulos, desarrolladores, géneros, calificaciones de los usuarios, precios y logros. La base de datos facilitará la administración de datos, mejorará la experiencia del usuario final y permitirá generar informes relevantes para la toma de decisiones.

**1.2 Objetivos del Proyecto**

1. Registrar información detallada de cada juego disponible en Steam.
2. Facilitar la consulta y filtrado por género, desarrollador, calificación y precio.
3. Gestionar los datos de logros y reseñas de los usuarios.
4. Proveer herramientas para análisis y reportes de tendencias en la plataforma.

**1.3 Alcance del Sistema**La base de datos cubrirá los siguientes módulos principales:

* Gestión de información de los juegos: títulos, descripciones, precios, descuentos y géneros.
* Registro de desarrolladores y sus juegos publicados.
* Gestión de logros asociados a cada juego.
* Registro de calificaciones y reseñas de los usuarios.
* Módulo para generar reportes sobre tendencias, juegos más valorados, etc.

### **2. Diseño Conceptual**

**2.1 Identificación de Entidades y Relaciones**

* **Entidades principales**:
  + Juego
  + Desarrollador
  + Género
  + Reseña
  + Logro
  + Usuario

**2.2 Diagrama Entidad-Relación (ER)**

* El diagrama ER incluye las entidades principales mencionadas con relaciones como:
  + Un juego pertenece a un desarrollador.
  + Un juego puede estar asociado a múltiples géneros.
  + Un usuario puede realizar múltiples reseñas y desbloquear múltiples logros.
  + Las reseñas están asociadas a juegos específicos.

### **3. Diseño Lógico**

**3.1 Mapear Entidades a Tablas**

* **Tabla Juegos**:
  + id\_juego (PK)
  + nombre
  + descripcion
  + precio
  + descuento
  + id\_desarrollador (FK)
* **Tabla Desarrolladores**:
  + id\_desarrollador (PK)
  + nombre
  + pais\_origen
* **Tabla Generos**:
  + id\_genero (PK)
  + descripcion
* **Tabla Reseñas**:
  + id\_resena (PK)
  + id\_juego (FK)
  + id\_usuario (FK)
  + calificacion
  + comentario
* **Tabla Logros**:
  + id\_logro (PK)
  + id\_juego (FK)
  + descripcion
* **Tabla Usuarios**:
  + id\_usuario (PK)
  + nombre\_usuario
  + correo
  + fecha\_registro

**3.2 Relaciones entre Tablas**

* Los juegos pueden tener múltiples géneros mediante una tabla intermedia (juego\_genero).
* Las reseñas y logros están asociadas a juegos y usuarios.

### **4. Diseño Físico**

**4.1 Optimización del DBMS**

* Uso de índices en los campos más consultados, como nombre en Juegos y id\_juego en las tablas relacionadas.
* Elección de tipos de datos optimizados, como FLOAT para precios y VARCHAR para nombres y descripciones.
* Creación de particiones en tablas con alto volumen de datos, como Reseñas.

### **5. Implementación**

**5.1 Scripts SQL**

* Creación de las tablas mencionadas con claves primarias y foráneas.
* Inserción de datos iniciales (ejemplo: géneros predefinidos como Acción, Aventura, etc.).
* Configuración de restricciones como NOT NULL y UNIQUE para campos clave.

### **6. Pruebas**

**6.1 Tipos de Pruebas**

* **Pruebas de Integridad**: Verificar relaciones entre tablas (FK).
* **Pruebas de Rendimiento**: Evaluar tiempos de respuesta para consultas frecuentes.
* **Pruebas de Seguridad**: Validar que los usuarios no autorizados no puedan modificar datos sensibles.
* **Pruebas de Casos de Uso**:
  + Consulta de juegos por género.
  + Registro de reseñas por parte de usuarios.

### **7. Mantenimiento y Optimización Continua**

**7.1 Actividades**

* Monitorear el crecimiento de la base de datos para ajustar índices o particiones.
* Realizar migraciones cuando sea necesario para agregar nuevos campos o tablas.
* Implementar backups regulares para garantizar la recuperación en caso de falla.

### **Reglas de Negocio Adaptadas a la Base de Datos de Steam**

**RN-1:** Cada usuario debe tener un email único dentro de la base de datos.  
**Implementación:** La columna email en la tabla Usuarios se define como UNIQUE. Esto asegura que no pueda haber duplicados.

ALTER TABLE Usuarios ADD CONSTRAINT UNIQUE (email);

**RN-2:** Los usuarios deben mantener saldo positivo en su wallet para conservar su estatus activo en el sistema.  
**Implementación:** El saldo de la columna saldo\_wallet de la tabla Usuarios debe ser mayor o igual a 0. Además, podría ser necesario un proceso automatizado que evalúe este saldo y ajuste el nivel del usuario si no cumple con esta regla.

CREATE TRIGGER verificar\_saldo BEFORE UPDATE ON Usuarios

FOR EACH ROW

BEGIN

IF NEW.saldo\_wallet < 0 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'El saldo no puede ser negativo.';

END IF;

END;

**RN-3:** Las reseñas de juegos deben estar asociadas a un usuario registrado y especificar una puntuación válida entre 1 y 5.  
**Implementación:** Ya está implementado mediante las claves foráneas y la restricción CHECK en la tabla Reseñas.

**RN-4:** Cada juego puede estar asociado a varias categorías, pero una combinación específica de id\_juego e id\_categoria debe ser única.  
**Implementación:** Esto ya está asegurado mediante la clave primaria compuesta en la tabla Juego\_Categoría.

### **Resumen de Pasos**

1. Recopilación y análisis de requisitos.
2. Diseño conceptual mediante el diagrama ER.
3. Diseño lógico con la normalización y definición de tablas.
4. Diseño físico considerando la optimización del DBMS.
5. Implementación mediante scripts SQL.
6. Pruebas para garantizar integridad, rendimiento y seguridad.
7. Mantenimiento y optimización continua para asegurar la escalabilidad.

Este documento establece las bases para el desarrollo de una base de datos robusta y escalable para la gestión de juegos en Steam.