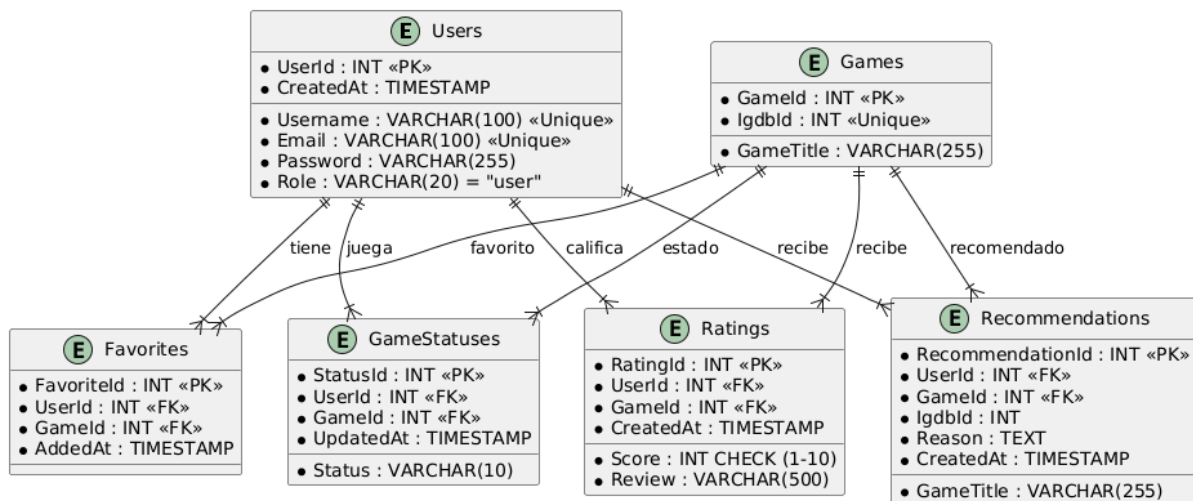


Documento del Proyecto: Catálogo Inteligente de Videojuegos con IA y Recomendaciones Personalizadas

1. Modelo de Datos

1.1 Diagrama Entidad-Relación (ER)

El diagrama entidad-relación representa la estructura de la base de datos, definiendo las relaciones entre usuarios, videojuegos y funcionalidades clave.



Clases y Atributos

Users (Usuarios)

- UserId: Identificador único del usuario.
- Username: Nombre único del usuario.
- Email: Correo electrónico del usuario.
- Password: Contraseña encriptada.
- Role: Rol del usuario (por defecto "user").
- CreatedAt: Fecha de creación del usuario.

Games (Juegos)

- GameId: Identificador único del juego en la base de datos.
- IgdbId: Identificador del juego en IGDB (fuente de datos externa).
- GameTitle: Nombre del juego.

Ratings (Calificaciones de juegos)

- RatingId: Identificador único de la calificación.
- UserId: Usuario que realizó la calificación.
- GameId: Juego calificado.
- Score: Puntuación del juego (entre 1 y 10).

- Review: Opinión del usuario sobre el juego.
- CreatedAt: Fecha de la calificación.

GameStatuses (Estados del juego)

- StatusId: Identificador único del estado.
- UserId: Usuario que registró el estado.
- GameId: Juego al que se le asignó un estado.
- Status: Estado del juego (Wishlist, Owned, Playing, Completed, Abandoned).
- UpdatedAt: Fecha de la última actualización del estado.

Favorites (Juegos favoritos)

- FavoriteId: Identificador único de favorito.
- UserId: Usuario que marcó el juego como favorito.
- GameId: Juego marcado como favorito.
- AddedAt: Fecha en que se agregó a favoritos.

Recommendations (Recomendaciones de juegos)

- RecommendationId: Identificador único de la recomendación.
- UserId: Usuario al que se le recomienda el juego.
- GameId: Juego recomendado.
- IgdbId: Identificador del juego en IGDB.
- GameTitle: Título del juego recomendado.
- Reason: Razón de la recomendación.
- CreatedAt: Fecha de la recomendación.
-

1.3 Estructura de la Base de Datos

La base de datos se basa en **PostgreSQL**, garantizando escalabilidad y eficiencia. Se utilizan claves foráneas para mantener integridad referencial y se implementan índices en los campos clave para optimizar las consultas.

```
-- ♦ Crear la base de datos

CREATE DATABASE SmartGameDB;

\c SmartGameDB;

-- ♦ Crear la tabla de Usuarios

CREATE TABLE Users (

    UserId SERIAL PRIMARY KEY,

    Username VARCHAR(100) NOT NULL UNIQUE,
```

```

    Email VARCHAR(100) UNIQUE NOT NULL,

    Password VARCHAR(255) NOT NULL,

    Role VARCHAR(20) DEFAULT 'user',

    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- ♦ Crear la tabla de Videojuegos (Referenciando IGDB)

CREATE TABLE Games (

    GameId SERIAL PRIMARY KEY,

    IgdbId INT NOT NULL UNIQUE,

    GameTitle VARCHAR(255) NOT NULL

);

-- ♦ Crear la tabla de Calificaciones

CREATE TABLE Ratings (

    RatingId SERIAL PRIMARY KEY,

    UserId INT NOT NULL,

    GameId INT NOT NULL,

    Score INT NOT NULL CHECK (Score >= 1 AND Score <= 10),

    Review VARCHAR(500),

    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (UserId) REFERENCES Users(UserId) ON DELETE CASCADE,

    FOREIGN KEY (GameId) REFERENCES Games(GameId) ON DELETE CASCADE

);

```

```
-- ♦ Crear la tabla de Estados del Juego (Wishlist, Owned, Playing,
Completed, Abandoned)

CREATE TABLE GameStatuses (

    StatusId SERIAL PRIMARY KEY,

    UserId INT NOT NULL,

    GameId INT NOT NULL,

    Status VARCHAR(10) NOT NULL CHECK (Status IN ('Wishlist', 'Owned',
'Playing', 'Completed', 'Abandoned')),

    UpdatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (UserId) REFERENCES Users(UserId) ON DELETE CASCADE,

    FOREIGN KEY (GameId) REFERENCES Games(GameId) ON DELETE CASCADE
);

-- ♦ Crear la tabla de Favoritos

CREATE TABLE Favorites (

    FavoriteId SERIAL PRIMARY KEY,

    UserId INT NOT NULL,

    GameId INT NOT NULL,

    AddedAt TIMESTAMP NOT NULL DEFAULT NOW(),

    FOREIGN KEY (UserId) REFERENCES Users(UserId) ON DELETE CASCADE,

    FOREIGN KEY (GameId) REFERENCES Games(GameId) ON DELETE CASCADE
);

-- ♦ Crear la tabla de Recomendaciones

CREATE TABLE Recommendations (

    RecommendationId SERIAL PRIMARY KEY,
```

```

    UserId INT NOT NULL,

    GameId INT NOT NULL,

    IgdbId INT NOT NULL,

    GameTitle VARCHAR(255) NOT NULL,

    Reason TEXT NOT NULL,

    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (UserId) REFERENCES Users(UserId) ON DELETE CASCADE,

    FOREIGN KEY (GameId) REFERENCES Games(GameId) ON DELETE CASCADE
);

-- ♦ Índices para mejorar la eficiencia

CREATE INDEX idx_users_email ON Users(Email);

CREATE INDEX idx_games_igdb ON Games(IgdbId);

CREATE INDEX idx_ratings_user ON Ratings(UserId);

CREATE INDEX idx_ratings_game ON Ratings(GameId);

CREATE INDEX idx_favorites_user ON Favorites(UserId);

CREATE INDEX idx_favorites_game ON Favorites(GameId);

CREATE INDEX idx_recommendations_user ON Recommendations(UserId);

CREATE INDEX idx_recommendations_game ON Recommendations(GameId);

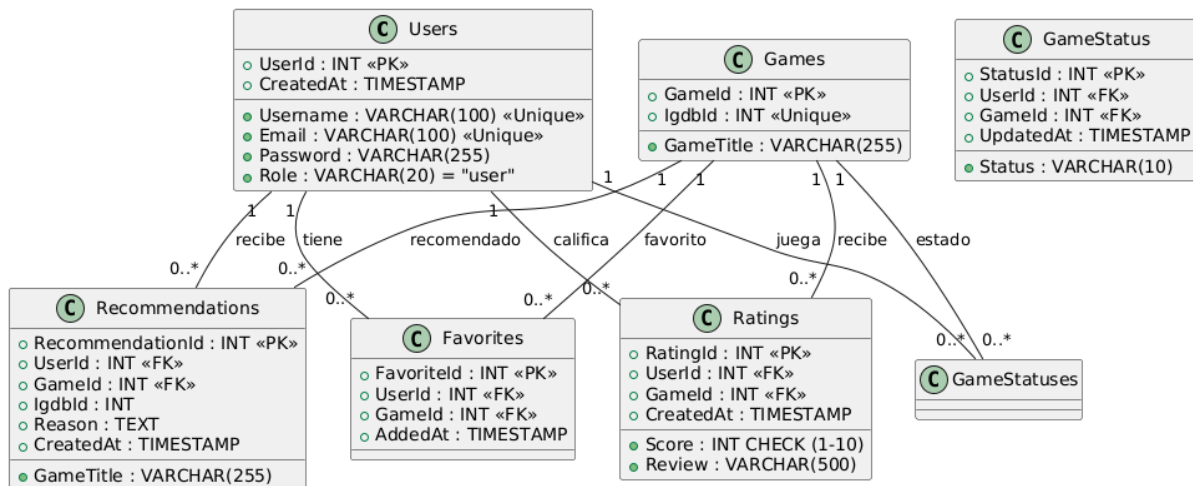
CREATE INDEX idx_gamestatus_user_game ON GameStatuses(UserId, GameId);

```

2. Diagramas UML

2.1 Diagrama de Clases

Este diagrama representa las principales entidades del sistema y sus relaciones, incluyendo atributos y métodos.



2.2 Diagrama de Casos de Uso

Este diagrama ilustra las interacciones de los usuarios con la plataforma, cubriendo acciones como:

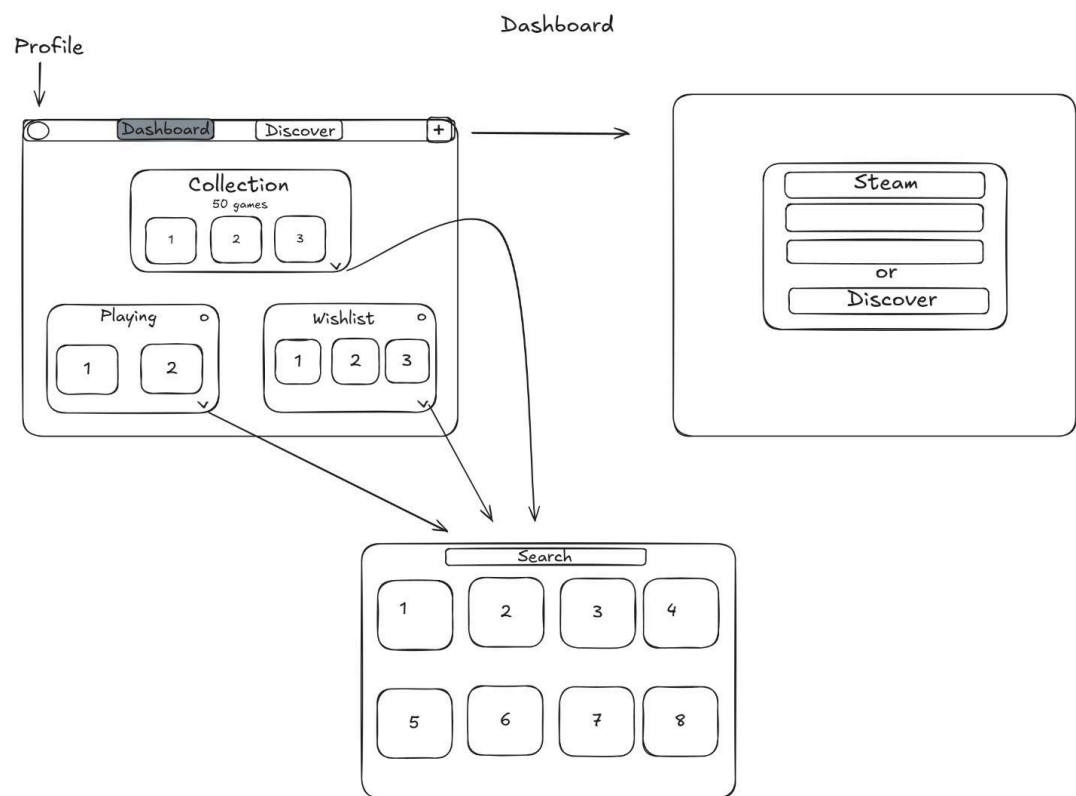
- Buscar juegos.
- Calificar juegos.
- Marcar juegos como "jugados" o "deseados".
- Recibir recomendaciones personalizadas.
- Administrar su perfil de usuario.

3. Diseño de Interfaces

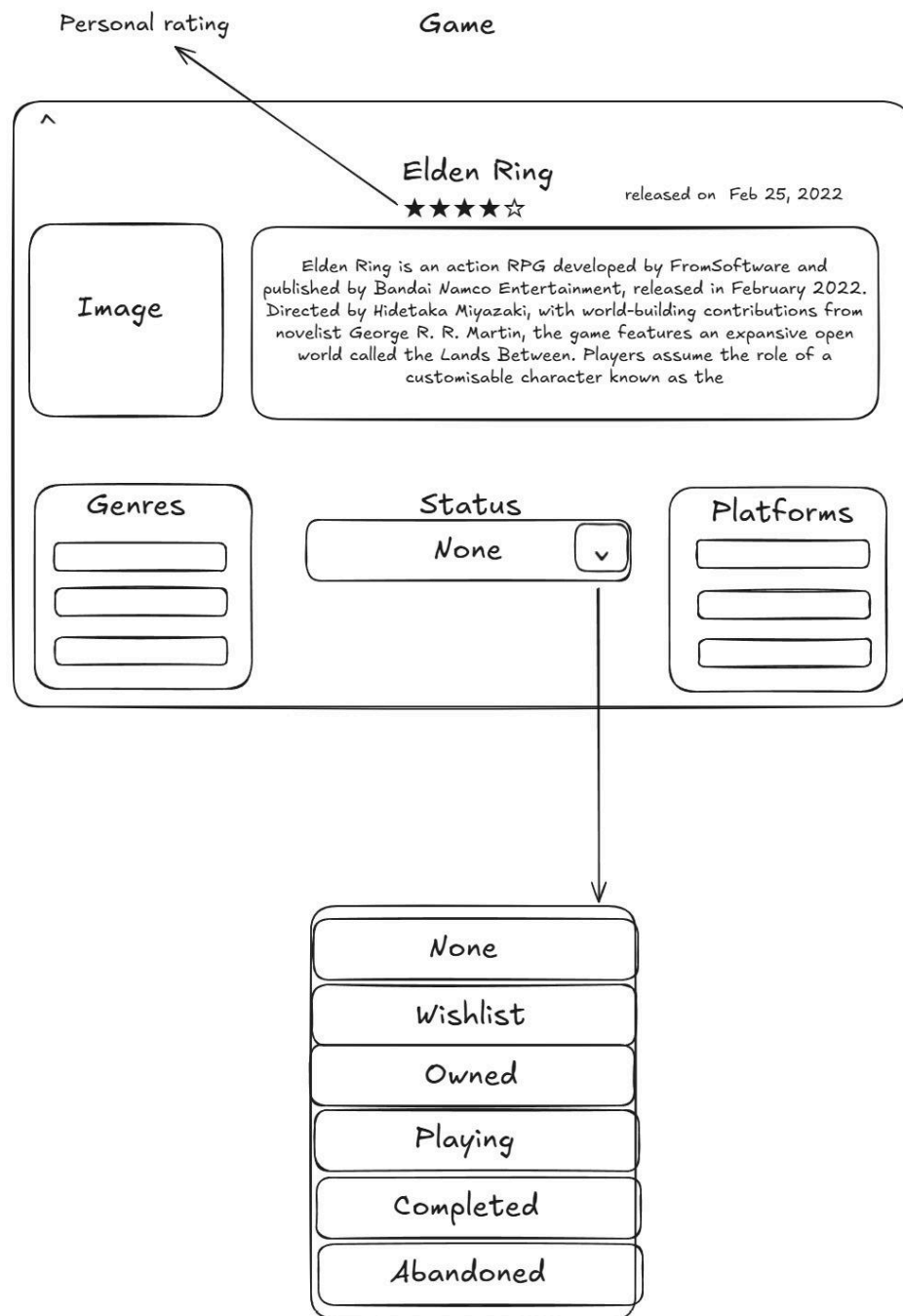
3.1 Bocetos de la Interfaz

Se incluyen diseños preliminares de la UI en Xcalidraw, mostrando:

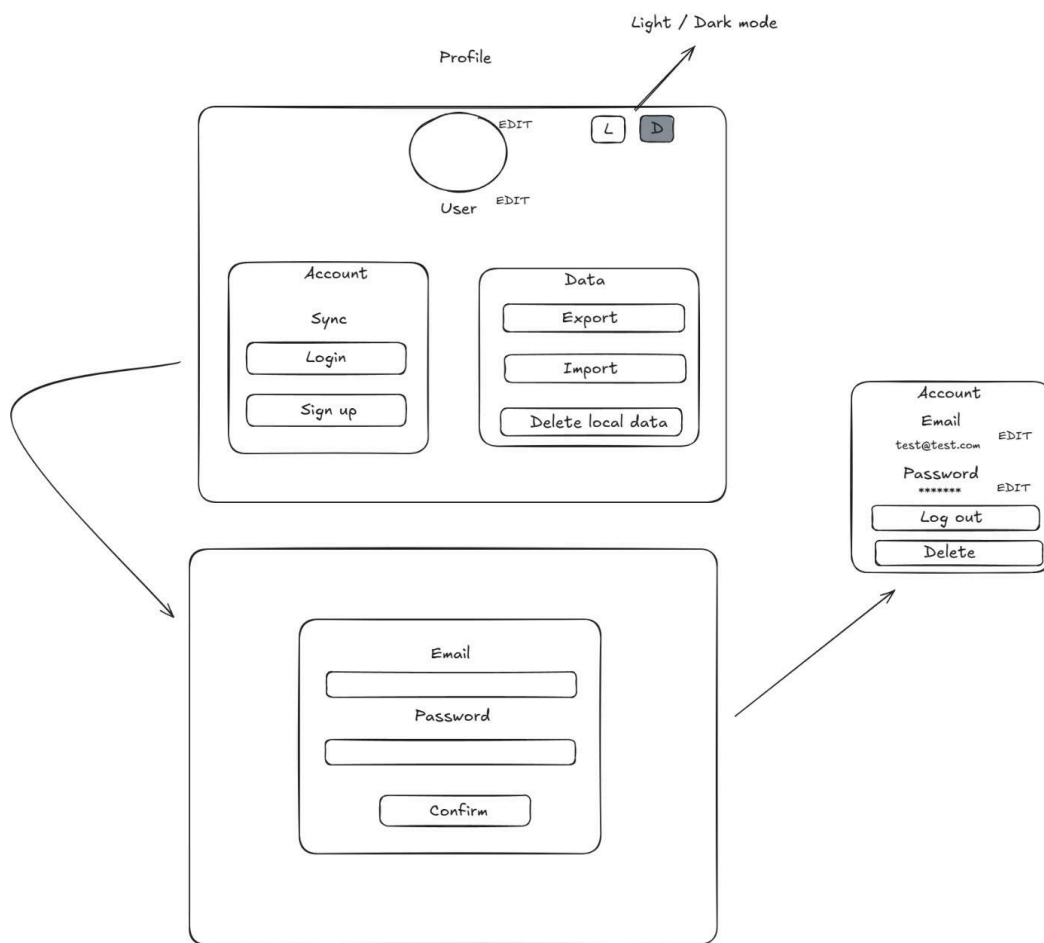
- Pantalla principal con buscador y recomendaciones.



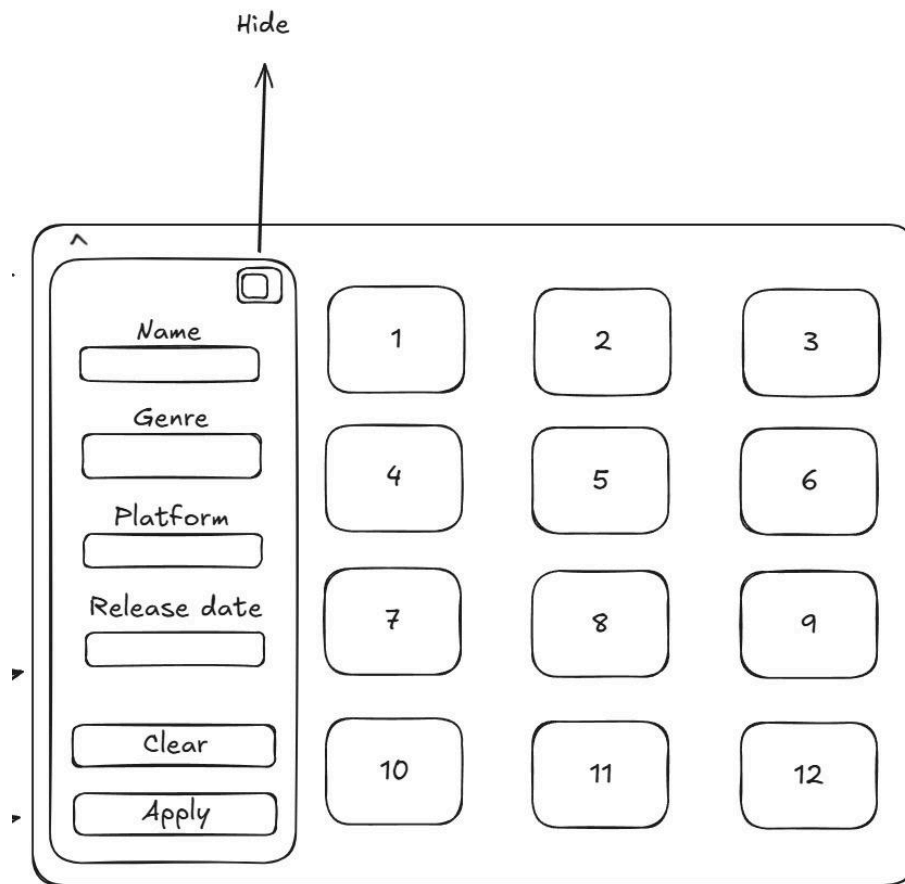
- Vista de detalles del juego con información relevante y opciones de interacción.



- Perfil del usuario con juegos favoritos y calificaciones.



- Discover con se con sugerencias de juegos por IA



3.2 Componentes Clave

- Búsqueda y filtros avanzados.
- Sección de recomendaciones basadas en IA.
- Página de detalles del juego con información de IGDB.
- Panel de usuario con historial de interacciones.

4. Planificación del Desarrollo

4.1 Hoja de Ruta

El desarrollo se divide en fases clave:

1. **Semana 1:** Diseño de arquitectura, ERD y UI.
2. **Semana 2-3:** Desarrollo del backend y conexión con PostgreSQL.
3. **Semana 4-5:** Desarrollo del frontend con Angular y Firebase Hosting.
4. **Semana 6:** Integración de IA con Weaviate.

5. **Semana 7:** Despliegue en Oracle Cloud VPS y pruebas finales.

4.2 Estado del Proyecto

- Creando estructura de datos DER y UML.
- Configuración de Docker Compose.
- Conexión con la base de datos PostgreSQL.
- Creacion preliminar del script de BBDD.
- Diseño preliminar de la imagen que tendra el Front.