Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Bases de Datos No-SQL

Integrantes:
Brando Martinez
Erika Rosales

Informe: Proyecto MongoDB

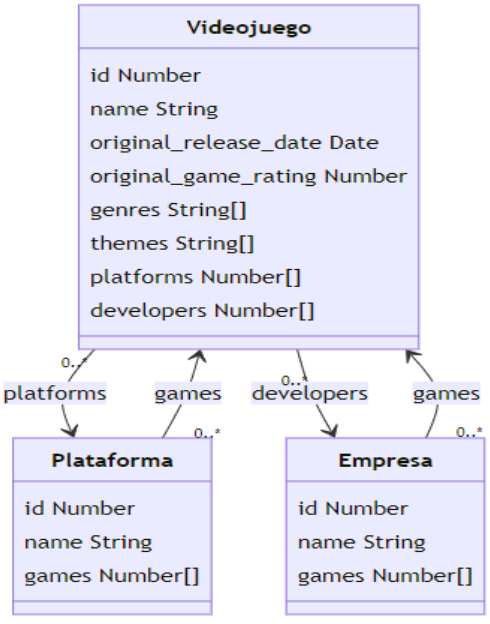
En este proyecto trabajamos con una base de datos MongoDB utilizando Mongoose y la biblioteca de Node.js que facilita la interacción con MongoDB a través de un esquema orientado a objetos. Se implementó el uso de un API para extraer los datos de videojuegos y datos relacionados y se cargaron los datos correspondientes en las diferentes colecciones que forman la estructura de la base de datos.

A continuación, detallamos la estructura y las tareas realizadas para llevar a cabo el proyecto.

Estructura de la base de datos

Siguiendo los lineamientos del proyecto, se crearon las siguientes colecciones:

- ★ Videojuego: Esta colección contiene la información principal de cada juego, incluyendo su nombre, fecha de lanzamiento, calificación, géneros, etiquetas, y los identificadores de sus desarrolladores y de las plataformas donde se encuentran.
- ★ Empresa: Almacena la información de las empresas desarrolladoras, y la lista de juegos desarrollados por cada empresa.
- ★ Plataforma: Contiene la información de las plataformas en las que están disponibles los juegos, incluyendo una lista de los Ids de juegos.



API

Queries

Para el desarrollo de las consultas se utilizó el lenguaje de programación JavaScript con Node.js como entorno de ejecución y Mongoose como biblioteca de Node.js que proporciona una solución basada en esquemas para modelar los datos en MongoDB.

Todas las consultas se desarrollaron siguiendo el esquema proporcionado en el archivo MongoDBClient.js. Cadá función será ejecutada al ser llamada desde el archivo index.js con los parámetros esperados devolviendo así los resultados.

Listado de consultas:

Consulta 1: Dado n géneros, buscar los juegos que contengan todos esos géneros.

Esta consulta a través del método find() nos permite recuperar los documentos de videojuegos que tienen asociados todos los géneros que se especifican en un array de elementos enviado como parámetro.

```
async consulta1(generos){
    try {
        const VideojuegoCollection = this.db.collection('Videojuego');
        // Buscar videojuegos que contengan todos los géneros especificados
        const resultados = await VideojuegoCollection.find({
            genres: { $all: generos }
        }).toArray();

        return resultados;
    } catch (error) {
        console.error('Error en la consulta 1:', error);
        return [];
    }

}
```

Consulta 2: Buscar juegos lanzados dentro de un rango de fechas (xx/xx/xxxx -yy/yy/yyyy) de n empresas.

Se buscan y muestran los juegos de las empresas especificadas y que hayan sido lanzados en el periodo de tiempo indicado. Para desarrollar esta consulta se requirió hacer una función para convertir el formato de las fechas ya que en Mongodb se almacenan en formatoISO y el formato que recibe la función es dd/mm/yyyy.

```
// Función para convertir una fecha en formato dd/mm/yyyy a un objeto Date
function convertirFecha(fechaStr) {
    const [dia, mes, anio] = fechaStr.split('/').map(Number);
    // Crear la fecha en UTC y ajustar la zona horaria local
    const fechaUTC = new Date(Date.UTC(anio, mes - 1, dia));
    // Crear una nueva fecha ajustada a la zona horaria local
    const fechaLocal = new Date(fechaUTC.getTime() - fechaUTC.getTimezoneOffset() * 60000);
    return fechaLocal;
}
```

Consulta 3: Buscar juegos que estén disponibles en más de n plataformas y a cuáles plataformas son.

Se implementó un pipeline de agregación que permitiera identificar los juegos que estuviesen en n+1 plataformas y a través de un lookup se obtuvieron los nombres de dichas plataformas en la colección correspondiente.

Consulta 4: Contar juegos por n empresas desarrolladoras con valoración mayor a x.

Se indican por parámetros las empresas desarrolladoras a través de un array y se envía también la valoración. Se utilizó el pipeline de agregación, se sumarizan los juegos que pertenecen a las empresas indicadas al igual que las valoraciones de los mismos y si superan el parámetro indicado son obtenidos estos documentos y mostrados. En el caso que los datos no cumplan los filtros requeridos la salida será `totalJuegos: 0, sumaValoraciones: 0`

Consulta 5: Buscar juegos con una calificación mayor al promedio y más de n géneros.

Para esta consulta se filtran los videojuegos que tienen n+1 géneros y luego se calcula el promedio de sus valoraciones y se muestran los documentos que tengan una valoración mayor al promedio total.

Consulta 6:Juegos con etiquetas específicas y ordenados por fecha de lanzamiento.

Utilizamos el método find() para filtrar los documentos que contienen todas las etiquetas especificadas y luego son ordenados descendentemente por su fecha de lanzamiento.

Consulta 7:Calificación promedio de juegos por género específico.

Esta consulta recibe un array de géneros que luego es descompuesto para agrupar por cada género y así poder calcular el promedio de la valoración por género.

Consulta 8: Buscar juegos por una palabra clave en el nombre

Utilizando nuevamente el método find() podemos a través de regex identificar los documentos que contienen una determinada palabra enviada por parámetro y devolverlos en la salida.

Consulta 9: Top 5 juegos mejor calificados por género específico y excluyendo ciertas empresas desarrolladoras.

En este caso, obtenemos los ids de las empresas indicadas por parámetro y luego poder filtrar a nivel de videojuegos excluyendo esas empresas y filtrando por los géneros. Para mostrar en forma de ranking se ordenan de mayor a menor valoración y se limita a solo 5 documentos que serían los primeros del ranking.

Consulta 10: Juegos por géneros y plataformas con proyección de campos.

A partir del listado de plataformas recibidas por parámetro se obtienen los ids desde la colección Plataforma para luego poder filtrar por esos ids en la colección de videojuego junto con los géneros. Luego en el pipeline de agregación también se realiza un lookup para obtener nuevamente el nombre de las plataformas para que finalmente sea proyectado junto con el nombre del juego.

Ambos integrantes realizaron aportes al desarrollo del proyecto, siendo Brando quien realizó mayor esfuerzo en la conexión con la API y carga de los datos y Erika concentró su participación en el desarrollo de las consultas.