



## Práctica 3

Escuela de Ciencias y Sistemas

Sistemas de Bases de Datos 2

Ing. Luis Alberto Arias Solórzano

Aux. Erwin Fernando Vásquez Peñate

## OBJETIVOS

### Objetivo general

Proporcionar a los estudiantes una experiencia práctica en el diseño y desarrollo de una aplicación de comercio electrónico simulada, utilizando bases de datos no relacionales como MongoDB, Redis y DynamoDB. La aplicación debe permitir la gestión de productos de entretenimiento, incluyendo películas y series, a través de operaciones CRUD (Crear, Leer, Actualizar y Eliminar), además de múltiples consultas que reflejarán información importante.

### Objetivos específicos

- Diseñar el esquema de datos para productos de entretenimiento en las bases de datos no relacionales MongoDB, Redis y DynamoDB.
- Implementar operaciones CRUD para verificar el correcto funcionamiento de la manipulación de la información.
- Desarrollar una api que pueda ser consumida por un cliente, para poder verificar el correcto funcionamiento de las consultas.

## ENUNCIADO

### Descripción general

Actualmente, una empresa que provee servicios de streaming como películas y series ha tenido mucho auge en los últimos años, por lo que necesitan escalar de forma rápida.

Para almacenar información de sus películas y series planean utilizar bases de datos no relacionales, ya que esto les garantiza disponibilidad y escalabilidad de manera rápida.

Lamentablemente en la empresa de servicios de streaming no cuentan con un encargado especializado en bases de datos no relacionales, por lo que los contratan a ustedes como grupo para poder gestionar y solventar sus necesidades.

### Entidades con tipos de datos

- Película:
  - Year : dato numérico
  - Genre: string
  - Rating: dato numérico
  - Country : string
  - Principal\_Language: string
  - Director\_name: String
  - Clasification: String
  - Price: dato numérico
- Serie:
  - Year: dato numérico
  - Genre: String
  - Rating: dato numérico
  - Seasons: dato numérico
  - Current\_Season: dato numérico
  - Price: dato numérico

## Problema a resolver

Se le solicita a su grupo de trabajo, construir una API (El lenguaje queda a discreción del grupo de trabajo), para poder interactuar con la información almacenada en las bases de datos no relacionales.

Para esta ocasión tiene como alternativas el uso de 3 distintas bases de datos no relacionales, las cuales son:

- Redis
- Mongo DB
- Dynamo DB

Deberá escoger 2 bases de datos, una guardará la información de las películas y la otra guardará información de las series.

## Consultas requeridas (Películas)

La parte que se encuentra en negrita en cada endpoint representa algún parámetro que lleve la ruta, las rutas son solo de ejemplo, el nombre y la forma en que envíen los parámetros queda a discreción del estudiante.

1. Operaciones Crud básicas (Crear, eliminar, actualizar, obtener por id y listar todas)
2. Buscar películas por género (por ejemplo, Comedia)
  - [http://localhost:3000/movie/movies-by-genre/\*\*genero\*\*](http://localhost:3000/movie/movies-by-genre/genero)
3. Mostrar películas con una clasificación R (restringida) o superior
  - <http://localhost:3000/movie/movies-by-RClasification>
4. Encontrar películas dirigidas por un director específico
  - [http://localhost:3000/movie/movies-by-director/\*\*Director\*\*](http://localhost:3000/movie/movies-by-director/Director)
5. Buscar películas con un precio inferior a 10
  - <http://localhost:3000/movie/lower-price>
6. Recuperar películas lanzadas en un año específico (por ejemplo, 2022)
  - <http://localhost:3000/movie/movies-by-year/2009>
7. Promedio de precio de todas las películas
  - <http://localhost:3000/movie/movies-price-average>
8. Promedio de precio de todas las películas en base a un género
  - [http://localhost:3000/movie/movies-price-average-by-benre/\*\*genero\*\*](http://localhost:3000/movie/movies-price-average-by-benre/genero)
9. Buscar películas con una palabra clave en el título (Búsqueda parcial)
  - <http://localhost:3000/movie/keyword-seach/Tit> // Debería devolver titanic, titanes, etc.

## Consultas requeridas (Series)

La parte que se encuentra en negrita en cada endpoint representa algún parámetro que lleve la ruta, las rutas son solo de ejemplo, el nombre y la forma en que envíen los parámetros queda a discreción del estudiante.

1. Operaciones Crud básicas (Crear, eliminar, actualizar, obtener por id y listar todas)
2. Buscar series por género (por ejemplo, Comedia)
  - <http://localhost:3000/serie/series-by-genre/genero>
3. Mostrar series que tengan más de 2 temporadas (seasons)
  - <http://localhost:3000/serie/series-seasons>
4. Encontrar series que estén en una temporada específica (current\_season)
  - <http://localhost:3000/serie/current-season>
5. Buscar series con un precio inferior a 10
  - <http://localhost:3000/serie/lower-price>
6. Recuperar películas estrenadas en un año específico (por ejemplo, 2022)
  - <http://localhost:3000/serie/series-by-year/2009>
7. Promedio de precio de todas las series
  - <http://localhost:3000/serie/series-price-average>
8. Ordenar series por rating
  - <http://localhost:3000/serie/series-by-rating-sorted>
9. Buscar series con una palabra clave en el título (Búsqueda parcial)
  - <http://localhost:3000/serie/keyword-seach/Cob> // Debería devolver Cobra kai, Cobertura, etc

## Consideraciones importantes

- El lenguaje de la API queda a discreción del estudiante.
- Para el correcto funcionamiento de los endpoints se evaluarán desde un cliente, por ejemplo Postman.
- Las bases de datos no relacionales permitidas son: Redis, Mongo DB y Dynamo DB.
- Durante la calificación se pedirá que los estudiantes ejecuten sus consultas para verificar correcto funcionamiento.
- **La entrega deberá contener lo siguiente:**

### **Documento PDF con lo siguiente:**

- Carátula con los datos de los integrantes del grupo
  - Descripción de cada consulta realizada y explicada en forma de documentación.
- El código fuente de la API.

## ENTREGA

- ❖ La entrega será el 23 de abril, a más tardar a las 23:59 pm.
- ❖ Se debe utilizar el mismo repositorio de la práctica 1.
- ❖ Se debe crear una carpeta en dicho repositorio con el nombre “Practica3”.
- ❖ En el repositorio deben incluir el código fuente utilizado para su API.
- ❖ Para entregar el proyecto en UEDI se deberá subir un archivo de texto con el link del repositorio, nombre del archivo BD2\_Practica3\_#Carnet|Grupo.
- ❖ Copias totales o parciales entre integrantes del laboratorio o bajadas de internet obtendrán nota de 0 puntos.
- ❖ No se recibirán entregas tardías.
- ❖ Agregar al respectivo auxiliar al repositorio (@erwin14k).