 <small>Universidad Autónoma de Occidente</small>	UNIVERSIDAD AUTÓNOMA DE OCCIDENTE					Valoración
	FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS			BASES DE DATOS 2-NoSQL	GRUPO: 1	
	CÓDIGO:	2235953	NOMBRE:	CESAR ANDRES DIAZ JIMENEZ		
PARCIAL SEGUNDO CORTE						FECHA: 30-09-2025

CASO DE ESTUDIO

Ejercicio de Evaluación Parcial: MongoDB Atlas

Contexto del Caso de Estudio: Plataforma de Streaming de Películas "MongoFlix"

"MongoFlix" es una plataforma emergente de streaming de películas que necesita una base de datos robusta y escalable para gestionar su catálogo de películas, directores y usuarios. La plataforma requiere almacenar información detallada sobre cada película, incluyendo su título, año de lanzamiento, género, y los directores asociados a ella. Además, necesita llevar un registro de sus usuarios, sus planes de suscripción y las películas que han calificado.

El objetivo de este ejercicio es diseñar e implementar un modelo de datos en MongoDB Atlas que satisfaga las necesidades de "MongoFlix", poblar la base de datos con datos de ejemplo y realizar una serie de consultas que demuestren la capacidad del modelo para responder a las preguntas de negocio de la plataforma.

Fase 1: Modelado de Datos

Basado en el caso de estudio de "MongoFlix", tu primera tarea es diseñar un modelo de datos NoSQL. Deberás definir las colecciones necesarias y la estructura de los documentos para cada una.

Requisitos del modelado:

- Colecciones:** Debes proponer y justificar al menos tres colecciones principales para organizar los datos de películas, directores y usuarios.
- Relaciones:** El modelo debe implementar relaciones entre las colecciones. Específicamente:
 - Una película puede tener **uno o varios** directores.
 - Un director puede haber dirigido **una o varias** películas.
 - Un usuario puede calificar **varias** películas, y una película puede ser calificada por **varios** usuarios.
- Diseño de Esquemas:** Para cada colección, define la estructura de los documentos. Decide qué relaciones modelar mediante **documentos embebidos (embedding)** y cuáles mediante **referencias (referencing)**. Justifica tus decisiones de modelado basándote en los patrones de

acceso y la naturaleza de los datos.

Entregable de la Fase 1:

Un documento que describa la estructura de tus colecciones en formato JSON, junto con una explicación clara de por qué elegiste embeber o referenciar en cada caso.

R

Para el desarrollo de esta primera fase, decidí utilizar 3 colecciones principales en este caso serán: directores, películas y usuarios. Cuyas estructuras en formato JSON serían las siguientes:

Directores:

```
{
  "_id": ObjectId("..."),
  "nombre": "Cesar Diaz",
  "fechaNacimiento": ISODate("1970-07-30"),
  "nacionalidad": "Colombiano"
}
```

Películas:

```
{
  "_id": ObjectId("..."),
  "titulo": "Inception",
  "añoLanzamiento": 2010,
  "genero": ["Ciencia Ficción", "Thriller"],
  "directores_ids": [ObjectId("...")]
}
```

Usuarios:

```
{
  "_id": ObjectId("..."),
  "nombreUsuario": "pepito perez",
}
```

```

"email": "pepite@uao.edu.com",
"planSuscripcion": "Premium",
"calificaciones": [
  {
    "pelicula_id": ObjectId("..."),
    "puntuacion": 5,
    "fecha": ISODate("2025-09-30T00:00:00Z")
  },
  {
    "pelicula_id": ObjectId("..."),
    "puntuacion": 4,
    "fecha": ISODate("2025-09-30T00:00:00Z")
  }
]
}

```

Justificación del Modelado y Explicación

Este modelado de la base de datos contiene ambos tipos de documentos debido a sus relaciones. Se utiliza el referencing en la colección de movies mediante el `directores_ids`, y en calificaciones en usuarios mediante `pelicula_id`.

Este tipo de modelado se eligió debido a que los directores y las películas pueden estar relacionados con múltiples documentos, y de esta forma se evita la duplicidad de datos y permite generar actualizaciones independientes.

Los tipos de documentos embebidos se utilizan por ejemplo en calificaciones, en vez de crear una colección exclusivamente para calificaciones, se le atribuye un documento embebido a la colección usuarios con los atributos necesarios para la calificación, ¿Por qué se realiza de esta manera?, debido a que las calificaciones están directamente asociadas al usuario y sin la existencia de este ellas no existen, y de esta manera también se genera un mejor rendimiento en consultas centradas en el usuario

Esta última colección, “Usuarios” contiene modelado tanto de referencia como de embebido.

Fase 2: Implementación y Carga de Datos

Una vez aprobado tu modelo, deberás crear la base de datos y las colecciones en tu clúster de MongoDB Atlas. Luego, inserta los siguientes datos de ejemplo.

Instrucciones:

- Crea una base de datos llamada `mongo_flix_db`.
- Crea las colecciones que diseñaste en la Fase 1.
- Inserta los siguientes documentos. Asegúrate de mantener la consistencia de las referencias (`_id`) entre las colecciones.
- Capturas de pantalla y archivo con estructura y datos de la base de datos.

Datos para la colección `directors`:

```
[
  { "nombre": "Christopher Nolan", "fecha_nacimiento": new Date("1970-07-30"), "nacionalidad": "Británico" },
  { "nombre": "Quentin Tarantino", "fecha_nacimiento": new Date("1963-03-27"), "nacionalidad": "Estadounidense" },
  { "nombre": "Denis Villeneuve", "fecha_nacimiento": new Date("1967-10-03"), "nacionalidad": "Canadiense" }
]
```

Datos para la colección `movies` (usa los `_id` de los directores insertados):

```
[
  { "titulo": "Inception", "año_lanzamiento": 2010, "genero": ["Ciencia Ficción", "Thriller"], "directores_ids": [ObjectId("ID_DE_NOLAN")] },
  { "titulo": "Pulp Fiction", "año_lanzamiento": 1994, "genero": ["Crimen", "Drama"], "directores_ids": [ObjectId("ID_DE_TARANTINO")] },
  { "titulo": "Dune", "año_lanzamiento": 2021, "genero": ["Ciencia Ficción", "Aventura"], "directores_ids": [ObjectId("ID_DE_VILLENEUVE")] },
  { "titulo": "The Dark Knight", "año_lanzamiento": 2008, "genero": ["Acción", "Crimen", "Drama"], "directores_ids": [ObjectId("ID_DE_NOLAN")] }
]
```

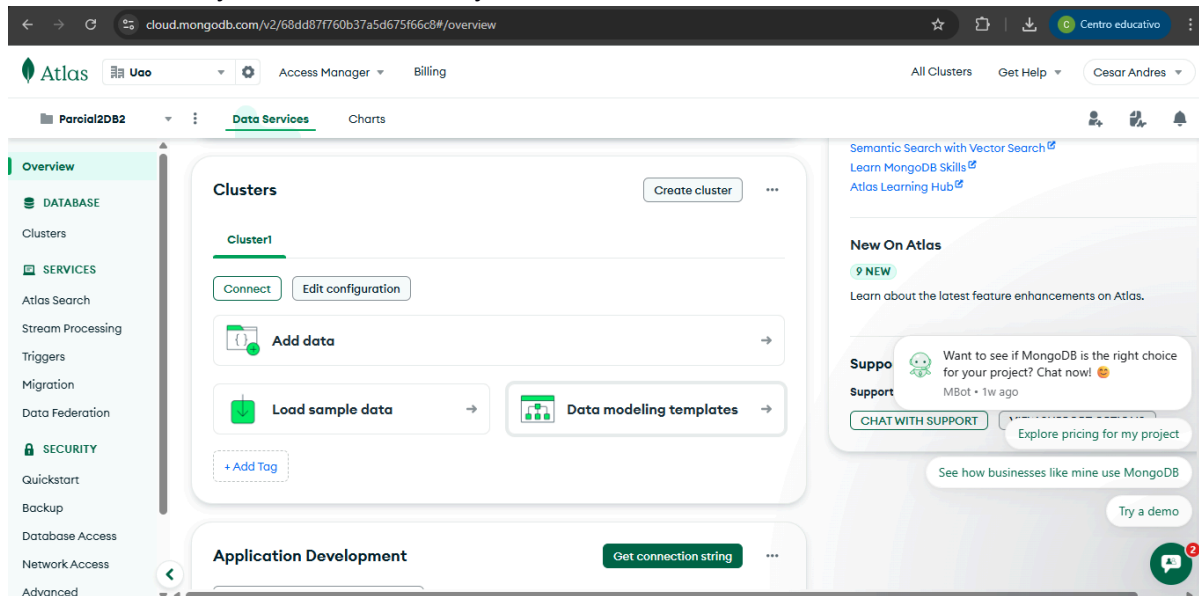
Datos para la colección `users` (usa los `_id` de las películas insertadas):

```
[
  { "nombre_usuario": "user_1", "email": "user1@example.com", "plan_suscripcion": "Premium", "calificaciones": [
    { "pelicula_id": ObjectId("ID_DE_INCEPTION"), "puntuacion": 5, "fecha": new Date() },
    { "pelicula_id": ObjectId("ID_DE_DUNE"), "puntuacion": 4, "fecha": new Date() }
  ] },
  { "nombre_usuario": "user_2", "email": "user2@example.com", "plan_suscripcion": "Básico", "calificaciones": [
    { "pelicula_id": ObjectId("ID_DE_PULP_FICTION"), "puntuacion": 5, "fecha": new Date() }
  ] },
  { "nombre_usuario": "user_3", "email": "user3@example.com", "plan_suscripcion": "Estándar", "calificaciones": [] }
]
```

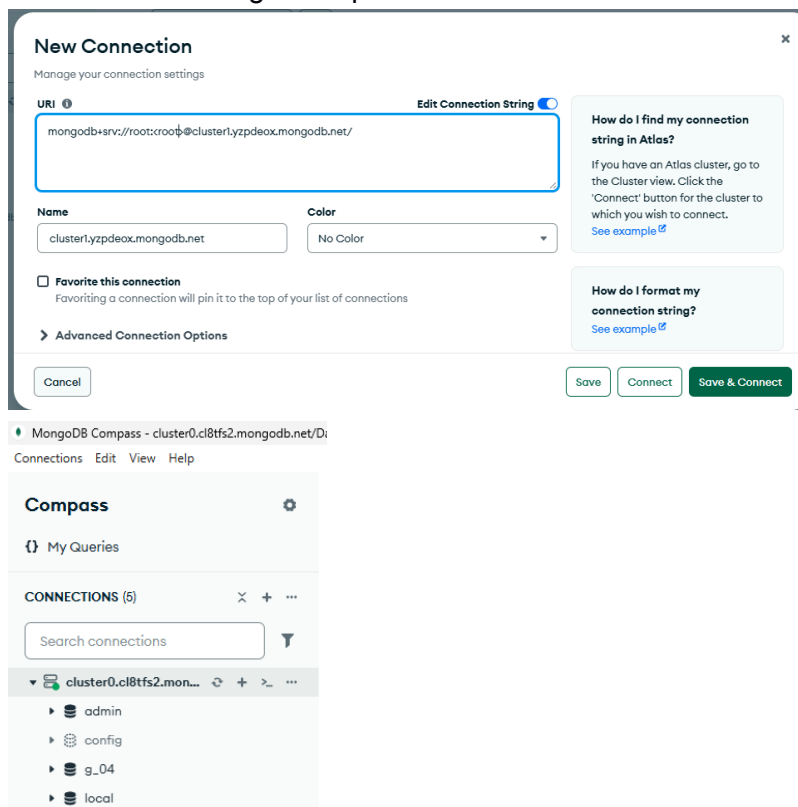
```
"calificaciones": [{ "pelicula_id": ObjectId("ID_DE_INCEPTION"), "puntuacion": 4, "fecha": new Date() }, { "pelicula_id": ObjectId("ID_DE_THE_DARK_KNIGHT"), "puntuacion": 5, "fecha": new Date() } ] }
```

Entrega de la Fase 2

Creación de Proyecto Parcial2DB2, y cluster1



Conexión con Mongo Compass



Creación de Colecciones desde Mongo Atlas

The screenshot shows the MongoDB Atlas interface for a cluster named 'Cluster1' in the 'AWS N. Virginia (us-east-1)' region. The 'Collections' tab is selected, showing the database 'mongo_flix_db' with a collection named 'directores'. The interface includes a sidebar with navigation options like Overview, DATABASE, Clusters, and SERVICES. The main panel shows the database 'mongo_flix_db' with a collection named 'directores'. A 'Loading Documents' bar is visible, indicating that documents are being loaded. The top navigation bar shows 'Uao' and 'Access Manager'.

Creación de Colecciones en Mongo Atlas

The screenshot shows the MongoDB Atlas interface for a cluster named 'Cluster1' in the 'AWS N. Virginia (us-east-1)' region. The 'Collections' tab is selected, showing the database 'mongo_flix_db' with a collection named 'directores'. The interface includes a sidebar with navigation options like Overview, DATABASE, Clusters, and SERVICES. The main panel shows the database 'mongo_flix_db' with a collection named 'directores'. A document is displayed with fields: '_id', 'nombre', 'fecha_nacimiento', and 'nacionalidad'. The top navigation bar shows 'Uao' and 'Access Manager'.

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 107B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

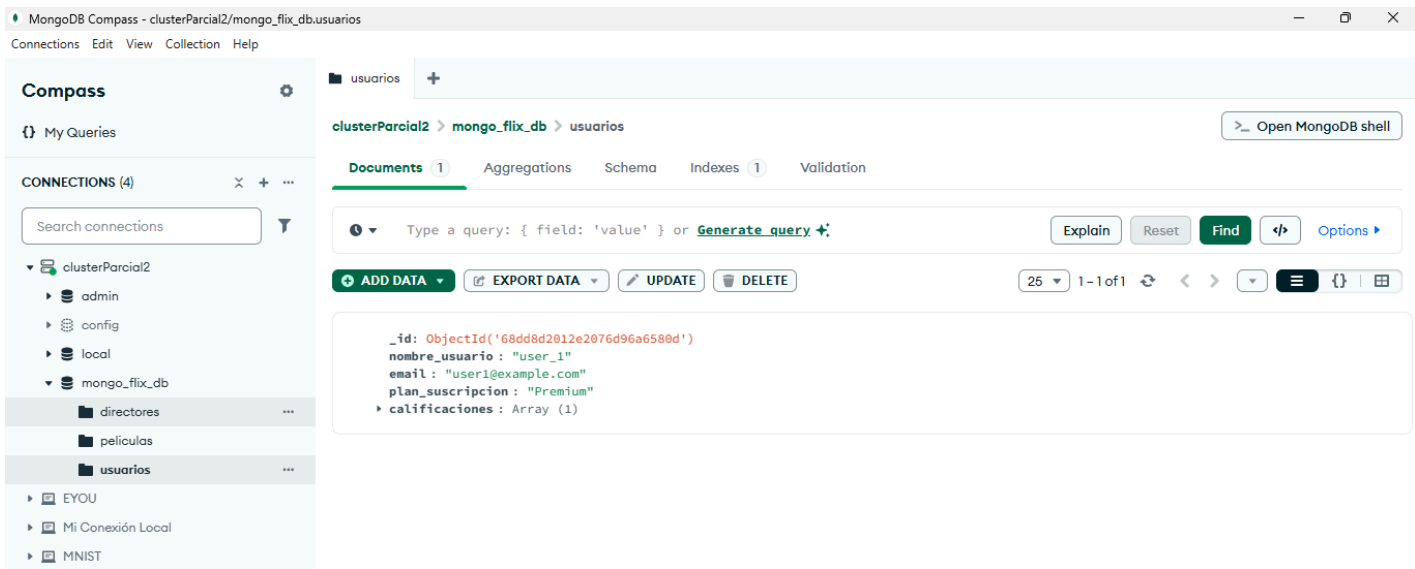
Generate queries from natural language in Compass

Filter Type a query: { field: 'value' } Reset Apply Options

QUERY RESULTS: 1-1 OF 1

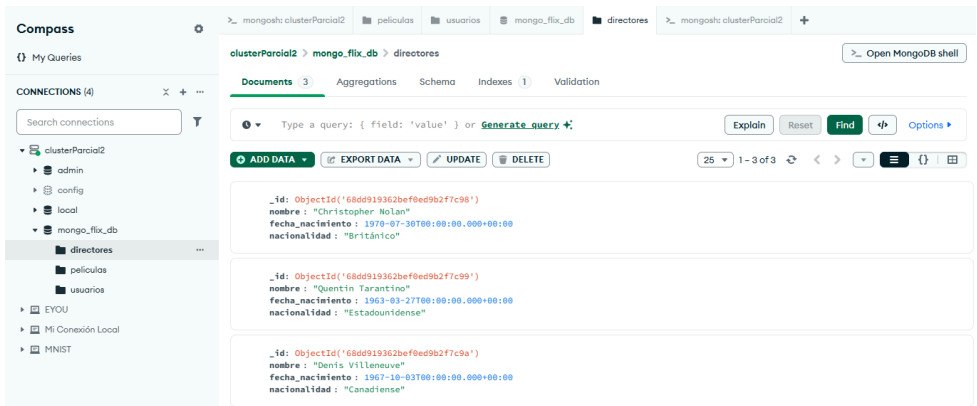
```
{
  "_id": ObjectId('68dd8c9112e2976d96a657f9'),
  "nombre": "Christopher Nolan",
  "fecha_nacimiento": "1970-07-30T00:00:00.000+00:00",
  "nacionalidad": "Británico"
}
```

Comprobación en Compass

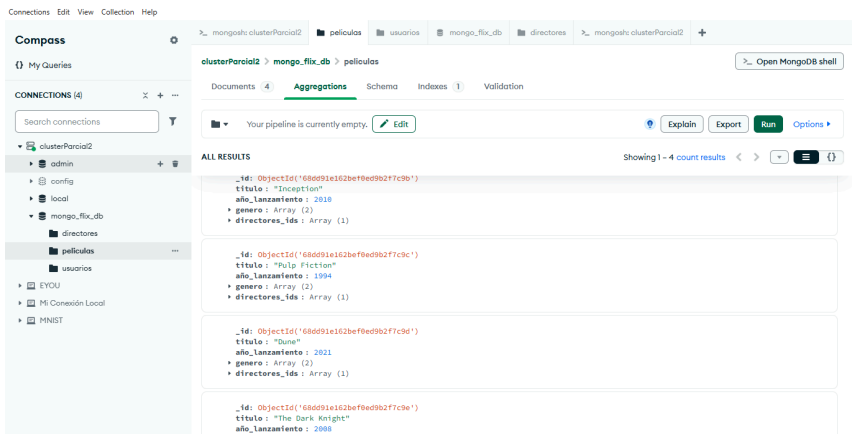


Creación de Datos

Colección Directores:



Colección Películas



Colección Usuarios

Compass

My Queries

CONNECTIONS (4)

Search connections

clusterParcial2

- admin
- config
- local
- mongo_fix_db
 - directores
 - peliculas
 - usuarios**
- EYOU
- Mi Conexión Local
- MNIST

clusterParcial2 > mongo_fix_db > usuarios

Documents 3 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

25 1-3 of 3

```
{
  "_id": ObjectId("68dd923b62bef0ed9b2f7c9f"),
  "nombre_usuario": "user_1",
  "email": "user1@example.com",
  "plan_suscripcion": "Premium",
  "calificaciones": Array (2)
    0: Object
      pelicula_id: ObjectId("68dd91e162bef0ed9b2f7c9b")
      puntuacion: 5
      fecha: 2025-10-01T20:42:35.813+00:00
    1: Object
      pelicula_id: ObjectId("68dd91e162bef0ed9b2f7c9d")
      puntuacion: 4
      fecha: 2025-10-01T20:42:35.813+00:00
}
```

```
{
  "_id": ObjectId("68dd923b62bef0ed9b2f7ca0"),
  "nombre_usuario": "user_2",
  "email": "user2@example.com",
  "plan_suscripcion": "Básico",
  "calificaciones": Array (1)
    0: Object
      pelicula_id: ObjectId("68dd91e162bef0ed9b2f7c9c")
      puntuacion: 5
      fecha: 2025-10-01T20:42:35.813+00:00
}
```

Fase 3: Consultas y Operaciones

Ahora, demuestra tu habilidad para extraer información valiosa de la base de datos. Escribe y ejecuta las siguientes consultas en MongoDB Shell (mongosh) o en el Data Explorer de Atlas.

Consultas requeridas:

- Búsqueda Simple Encuentra todas las películas del género "Ciencia Ficción".
- Filtro por Año: Lista las películas estrenadas después del año 2000.
- Actualización de un Documento: El usuario user_2 ha mejorado su plan a "Estándar". Realiza la actualización correspondiente en la base de datos.
- Consulta con \$lookup: Obtén una lista de todas las películas que incluya la información completa de sus directores (no solo sus _id). El resultado debe mostrar el título de la película y un array con los documentos de los directores.
- Pipeline de Agregación: Calcula la puntuación promedio para cada película que ha sido calificada. El resultado debe mostrar el título de la película y su calificación promedio. (Pista: necesitarás usar \$unwind, \$group y probablemente \$lookup).

Entregas de la Fase 3

a) Películas del Genero Ciencia Ficción

```
> db.peliculas.find({ genero: "Ciencia Ficción" })
< {
  _id: ObjectId('68dd91e162bef0ed9b2f7c9b'),
  titulo: 'Inception',
  'año_lanzamiento': 2010,
  genero: [
    'Ciencia Ficción',
    'Thriller'
  ],
  directores_ids: [
    ObjectId('68dd919362bef0ed9b2f7c98')
  ]
}
{
  _id: ObjectId('68dd91e162bef0ed9b2f7c9d'),
  titulo: 'Dune',
  'año_lanzamiento': 2021,
  genero: [
    'Ciencia Ficción',
    'Aventura'
  ],
  directores_ids: [
    ObjectId('68dd919362bef0ed9b2f7c9a')
  ]
}
```

b) Películas lanzadas después del año 2000

```
db.peliculas.find({ año_lanzamiento: { $gt: 2000 } })
{
  _id: ObjectId('68dd91e162bef0ed9b2f7c9b'),
  titulo: 'Inception',
  'año_lanzamiento': 2010,
  genero: [
    'Ciencia Ficción',
    'Thriller'
  ],
  directores_ids: [
    ObjectId('68dd919362bef0ed9b2f7c98')
  ]
}
{
  _id: ObjectId('68dd91e162bef0ed9b2f7c9d'),
  titulo: 'Dune',
  'año_lanzamiento': 2021,
  genero: [
    'Ciencia Ficción',
    'Aventura'
  ],
  directores_ids: [
    ObjectId('68dd919362bef0ed9b2f7c9a')
  ]
}
```

```
{
  _id: ObjectId('68dd91e162bef0ed9b2f7c9e'),
  titulo: 'The Dark Knight',
  'año_lanzamiento': 2008,
  genero: [
    'Acción',
    'Crimen',
    'Drama'
  ],
  directores_ids: [
    ObjectId('68dd919362bef0ed9b2f7c9a')
  ]
}
```

c) mejora de plan de usuario 2

```
> db.usuarios.updateOne(
  { nombre_usuario: "user_2" },
  { $set: { plan_suscripcion: "Estándar" } }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

d) películas con información completa de sus directores

```
db.películas.aggregate([
  {
    $lookup: {
      from: "directores",
      localField: "directores_ids",
      foreignField: "_id",
      as: "directores_info"
    }
  },
  {
    $project: {
      titulo: 1,
      directores_info: 1
    }
  }
])
{
  {
    _id: ObjectId('68dd91e162bef0ed9b2f7c9b'),
    titulo: 'Inception',
    directores_info: [
      {
        _id: ObjectId('68dd919362bef0ed9b2f7c98'),
        nombre: 'Christopher Nolan',
        fecha_nacimiento: 1970-07-30T00:00:00.000Z,
        nacionalidad: 'Británico'
      }
    ]
  },
  {
    _id: ObjectId('68dd91e162bef0ed9b2f7c9c'),
    titulo: 'Pulp Fiction',
    directores_info: [
      {
        _id: ObjectId('68dd919362bef0ed9b2f7c99'),
        nombre: 'Quentin Tarantino',
        fecha_nacimiento: 1963-03-27T00:00:00.000Z,
        nacionalidad: 'Estadounidense'
      }
    ]
  }
}
```

```
{
  _id: ObjectId('68dd91e162bef0ed9b2f7c9d'),
  titulo: 'Dune',
  directores_info: [
    {
      _id: ObjectId('68dd919362bef0ed9b2f7c9a'),
      nombre: 'Denis Villeneuve',
      fecha_nacimiento: 1967-10-03T00:00:00.000Z,
      nacionalidad: 'Canadiense'
    }
  ]
},
{
  _id: ObjectId('68dd91e162bef0ed9b2f7c9e'),
  titulo: 'The Dark Knight',
  directores_info: [
    {
      _id: ObjectId('68dd919362bef0ed9b2f7c9a'),
      nombre: 'Denis Villeneuve',
      fecha_nacimiento: 1967-10-03T00:00:00.000Z,
      nacionalidad: 'Canadiense'
    }
  ]
}
```

e) cálculo de puntuación promedio

```
> db.usuarios.aggregate([
  { $unwind: "$calificaciones" },
  {
    $group: {
      _id: "$calificaciones.pelicula_id",
      promedio: { $avg: "$calificaciones.puntuacion" }
    }
  },
  {
    $lookup: {
      from: "peliculas",
      localField: "_id",
      foreignField: "_id",
      as: "pelicula"
    }
  },
  {
    $project: {
      _id: 0,
      titulo: { $arrayElemAt: ["$pelicula.titulo", 0] },
      promedio: 1
    }
  }
])
```

```
< {
  promedio: 4.5,
  titulo: 'Pulp Fiction'
}
{
  promedio: 5,
  titulo: 'Inception'
}
{
  promedio: 4,
  titulo: 'Dune'
}
{
  promedio: 5,
  titulo: 'The Dark Knight'
}
```