



Proyecto Fase #II (Neo4j)

Autores:

Moisés Berenguer | CI: V24700980

Leonel Gonzalez | CI: V19175379

Eleazar Rojas | CI: V12715403

Materia: Bases de datos NoSQL

Referente a la carga de los datos 👍

1. Se realizó un pequeño proyecto en nodejs para agilizar la carga de datos y realización del ETL
2. Para correr el proyecto, es cuestión de crear un archivo llamado (.env) y copiar allí los datos de acceso de la DB que se encuentran en el archivo de ejemplo (.env.example)
3. La instancia de DB se encuentra en MLab y en Neo4j sandbox platform, por ende el servidor esta activo siempre y puede ser accedido desde Neo4j Browser a través de <http://35.153.19.215:33493/browser/> con las credenciales del archivo .env.example
4. Instalar los modulos de node (**npm install**)
5. Correr el script **npm start**
6. La operaciones utilizadas fueron: inserción común:

```
CREATE (category)-[:PRESENT_IN]->(award)
```

6. NOTA: Se insertaron todos los registros referentes a la década de 1980
7. NOTA: Se modificó el esquema para añadir la parte de los estados

Referente al modelo de datos

El modelo utilizado fue el siguiente:

```
//-- ACTOR
//-- Usando el esquema referenciado para relaciones N:N
actor: {
  _id: Number,
  name: String,
  acted_in: [Number],
```

```

    personifies:[Number]
}
//-- Logica usada: Saber en que peliculas ha actuado un actor y que papeles a
representado

//-- CHARACTER
//-- Usando el esquema referenciado para relaciones N:N
character: {
    _id:Number,
    name:String,
    appear_in:[Number],
    personified_by:[Number]
}
//-- Logica usada: Saber en en que peliculas ha aparecido un personaje y quien
a actuado a ese personaje

//-- DIRECTOR
//-- Usando el esquema referenciado para relaciones N:N
director: {
    _id:Number,
    name:String,
    directed_in:[Number]
}
//-- Logica usada: Saber que peliculas ha dirigido un director

//-- AWARD
//-- Usando el esquema referenciado para relaciones N:N
award: {
    _id:Number,
    name:String,
    has_category:[Number]
}
//-- Logica usada: Saber que categorias tiene un premio

//-- CATEGORY
//-- Usando el esquema referenciado para relaciones N:N
category: {
    _id:Number,
    name:String,
    present_in:[Number]
}
//-- Logica usada: Aber a que premios pertenece una categoria

//-- MOVIE
//-- Usando el esquema referenciado para relaciones N:N
movie: {

```

```

    _id: Number,
    name: {
      spanish: String,
      original: String,
    }
    clasification: String,
    rating: Number,
    studio: String,
    distributor: String,
    saga: String,
    release: Date,
    genre: [String],
    running_time: Number,
    gross: Number,
    preceded_by: Number,
    personified_by: [{
      actor_id: Number,
      character_id: Number,
      role: String
    }],
    directed_by: [Number],
    nominated_in: [{
      award_id: Number,
      category_id: Number,
      won: Boolean
    }]
  }
}
//-- Logica usada:
//-- Saber las peliculas que preceden
//-- Saber los personajes y actores
//-- Saber quien la ha dirigido
//-- Saber las nominaciones que tiene

```

Referente a los queries

Todas las consultas se encuentran en orden en el archivo **src/queries/index.js** puede ser ejecutado en mongo sin problema, Las salidas de cada uno se encuentra en los archivos adentro de la carpeta "images"

1. Actor, pelicula y personaje ![alt text][A1] ![alt text][A2]

2. Premio y categoria ![alt text][A3] ![alt text][A4]

3. Director y pelicula ![alt text][A5] ![alt text][A6]

4. Estado ![alt text][A7] ![alt text][A8]
