**Fundamentos de Arquitectura de Software**

**Departamento de Ciencias de la Computación y la Decisión**

**Facultad de Minas**

**Universidad Nacional de Colombia**

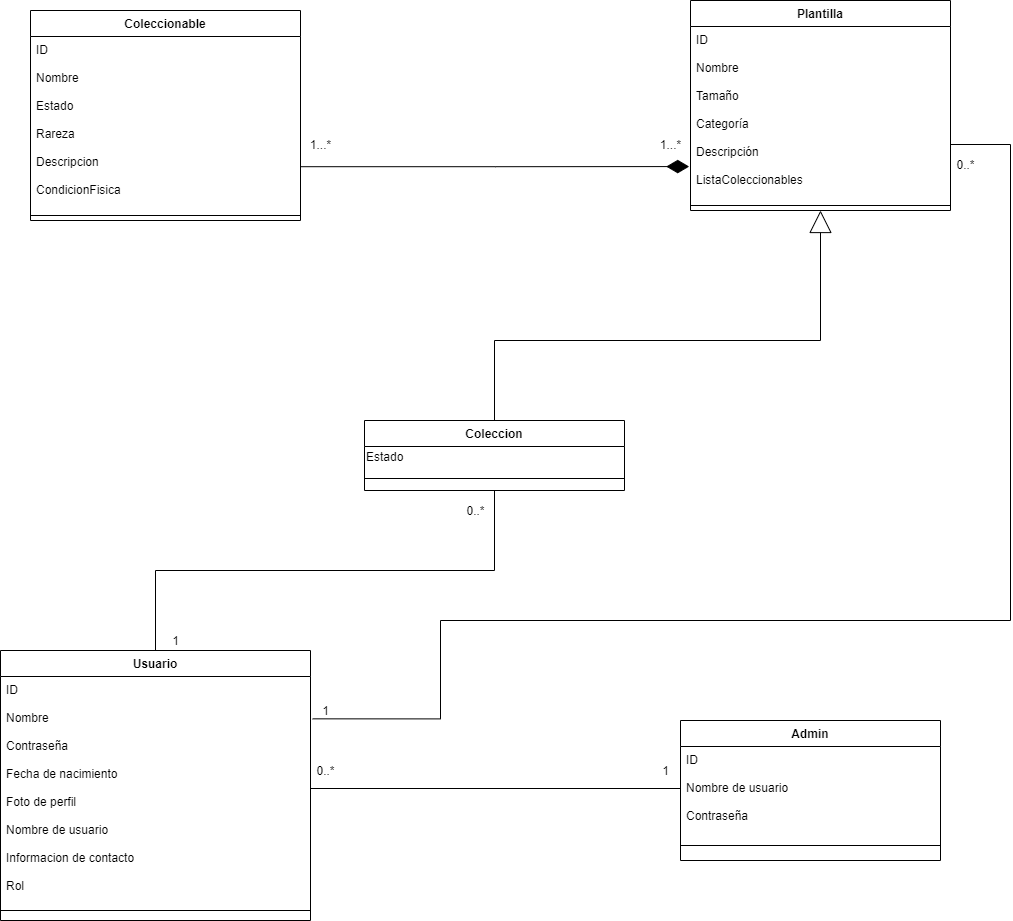
Taller Arquitectura de la Información

**responsables**

|  |
| --- |
| Nombre Completo |
| Sebastian Agudelo Osorio |
| Felipe Cortés Norena |
| Benjamin Cruz Álvarez |
| Daniel Giraldo Muñoz |
| Julián Ospina Murgueitio |
| Carlos Sebastian Zamora Rosero |

Los siguientes puntos se realizan considerando el conjunto de datos que se defina para la solución del proyecto específico, es decir, respecto al modelo del dominio.

1. **modelo de dominio**



1. **modelo entidad relacion**
   1. **sentencia o consulta de creación del tabla(s)**

***Observación****: Escriba el código en el Sistema de Gestión de Bases de Datos Relacionales de su elección para crear la tabla que corresponda con su conjunto de datos específico.*

***CREACION TABLA COLECCIONABLES:***

*CREATE TABLE coleccionables(*

*ID int PRIMARY KEY,*

*nombre VARCHAR(255) NOT NULL,*

*estado VARCHAR(255) NOT NULL,*

*rareza VARCHAR(255) NOT NULL,*

*descripcion VARCHAR(255),*

*condicion\_fisica VARCHAR(255)*

*);*

* 1. **sentencias para insertar datos**

***Observación****: Escriba el código para insertar al menos 5 elementos en la tabla(s) creadas en el punto 2.1.*

***INSERCIÓN DE 5 ITEMS:***

*INSERT INTO coleccionables (ID, nombre, estado, rareza, descripcion, condicion\_fisica) VALUES*

*(1, "Magikarp", "adquirido", "legendario", "Magikarp is a fish-like Pokémon with large, heavy reddish-orange scales.", "Nuevo"),*

*(2, "Pikachu", "adquirido", "común", "Pikachu es un Pokémon de tipo eléctrico introducido en la primera generación.", "Usado"),*

*(6, "Bidoof", "faltante", "legendario", "Bidoof es un Pokémon de tipo normal introducido en la cuarta generación.", "Nuevo"),*

*(99, "Mew", "faltante", "legendario", "Mew es un Pokémon singular de tipo psíquico introducido en la primera generación.", "Nuevo"),*

*(108, "Agumon", "adquirido", "raro", "Agumon es un pequeño y robusto dinosaurio bípedo de alrededor de 2 ½ pies de altura.", "Nuevo");*

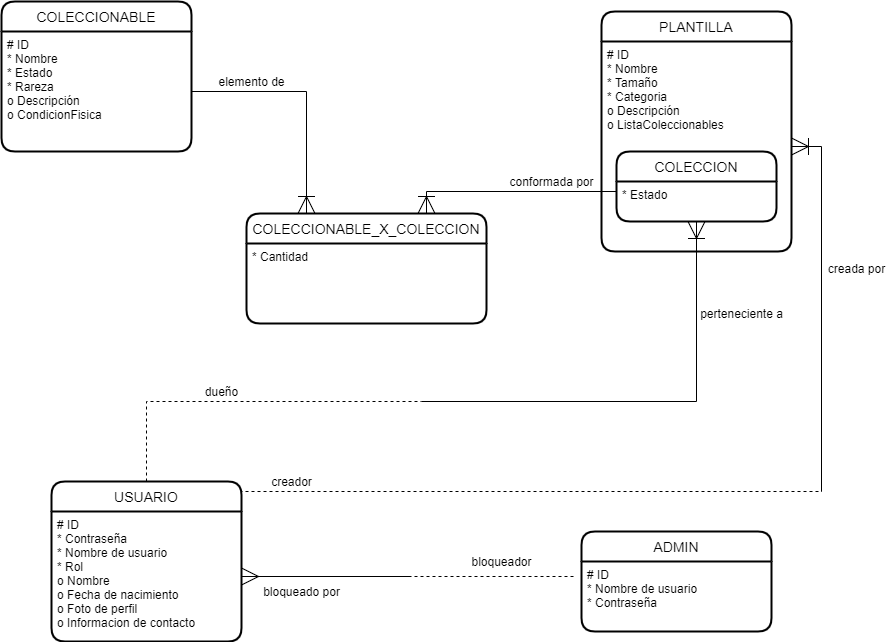
* 1. **sentencia de consulta**

***Observación****: Escriba el código para consultar solo tres atributos específicos de los insertados en la sección 2.2.*

***CONSULTA DE ID, NOMBRE Y DESCRIPCIÓN DE COLECCIONABLES QUE TENGAN EL ESTADO “ADQUIRIDO”:***

*SELECT ID, nombre, descripcion FROM coleccionables WHERE estado="adquirido";*

* 1. **toma de pantalla del modelo e r**

******

1. **mongodb**
   1. **sentencia o consulta de creación del documento(s)**

// Select the database to use.

use('colecciones');

// The drop() command destroys all data from a collection.

// Make sure you run it against proper database and collection.

db.coleccionable.drop();

db.coleccion.drop();

const coleccionable1 = {

\_id: 1,

nombre: 'Rafael Robayo',

estado: 'nuevo',

rareza: 'legendario',

descripcion: 'Luka Modric colombiano, pero con más talento',

condicionFisica: 'perfecta'

}

const coleccionable2 = {

\_id: 2,

nombre: 'Édizon Toloza',

estado: 'nuevo',

rareza: 'legendario',

descripcion: 'Mario Balotelli colombiano, pero con más sabor',

condicionFisica: 'perfecta'

}

const coleccionable3 = {

\_id: 3,

nombre: 'Marco Mbaperez',

estado: 'nuevo',

rareza: 'raro',

descripcion: 'Es algo que le puede pasar en el fútbol a cualquiera. Quizá quise hacer una jugada que no debí haber hecho y por ahí me resbalo y caigo',

condicionFisica: 'perfecta'

}

const coleccionable4 = {

\_id: 4,

nombre: 'Carlos Tevez',

estado: 'usado',

rareza: 'común',

descripcion: 'A medida que uno va ganando cosas, se hamburguesa',

condicionFisica: 'regular'

}

// Insert a few documents into the sales collection.

db.coleccionable.insertMany([

coleccionable1,

coleccionable2,

coleccionable3,

coleccionable4

]);

const coleccion1 = {

\_id: 1,

nombre: 'Láminas de fútbol',

tamano: 4, //se pueden usar las ñ?

categoria: [

'fútbol',

'leyendas',

'magos del balón'

],

descripcion: 'Colección de láminas de los mejores jugadores que ha conocido este planeta',

estado: 'incompleta',

listaColeccionables: [

{

id: 1,

cantidad: 2

},

{

id: 2,

cantidad: 0

},

{

id: 3,

cantidad: 3

},

{

id: 4,

cantidad: 1

}

]

};

db.coleccion.insertOne(coleccion1);

* 1. **sentencia de consulta**

db.coleccionable.find(

{ estado: 'nuevo'},

{ \_id: 0, nombre: 1, descripcion: 1}

).limit(2);

1. **hypertable**
   1. **sentencia o consulta de creación del tabla(s)**

***Observación****: Escriba el código en hypertable para crear la tabla que corresponda con su conjunto de datos específico.*

create namespace colecciones;

use colecciones;

CREATE TABLE coleccion (

nombre,

tamano,

categoria,

descripcion,

coleccionable,

ACCESS GROUP default (nombre, tamano, categoria, descripcion),

ACCESS GROUP coleccionables (coleccionable)

);

CREATE TABLE coleccionable (

nombre,

estado,

rareza,

descripcion,

condicionFisica

);

* 1. **sentencias para insertar datos**

***Observación****: Escriba el código en hypertable para insertar al menos 5 elementos en la tabla(s) creadas en el punto 4.1.*

INSERT INTO coleccion VALUES ("laminas", "nombre", "Láminas de fútbol"),

("laminas", "tamano", "3"),

("laminas", "categoria:1", "fútbol"),

("laminas", "categoria:2", "leyendas"),

("laminas", "categoria:3", "magos del balón"),

("laminas", "descripcion", "Colección de láminas de los mejores jugadores que ha conocido este planeta"),

("laminas", "coleccionable:1", "coleccionable1"),

("laminas", "coleccionable:2", "coleccionable2"),

("laminas", "coleccionable:3", "coleccionable3");

INSERT INTO coleccionable VALUES ("coleccionable1", "nombre", "Rafael Robayo"),

("coleccionable1", "estado", "nuevo"),

("coleccionable1", "rareza", "legendario"),

("coleccionable1", "descripcion", "Luka Modric colombiano, pero con más talento"),

("coleccionable1", "condicionFisica", "'perfecta'");

INSERT INTO coleccionable VALUES ("coleccionable2", "nombre", "Édizon Toloza"),

("coleccionable2", "estado", "nuevo"),

("coleccionable2", "rareza", "legendario"),

("coleccionable2", "descripcion", "Mario Balotelli colombiano, pero con más sabor"),

("coleccionable2", "condicionFisica", "'perfecta'");

INSERT INTO coleccionable VALUES ("coleccionable3", "nombre", "Marco Mbaperez"),

("coleccionable3", "estado", "nuevo"),

("coleccionable3", "rareza", "legendario"),

("coleccionable3", "descripcion", "Es algo que le puede pasar en el fútbol a cualquiera. Quizá quise hacer una jugada que no debí haber hecho y por ahí me resbalo y caigo"),

("coleccionable3", "condicionFisica", "'perfecta'");

* 1. **sentencia de consulta**

***Observación****: Escriba el código en hypertable consultar solo tres atributos específicos de los insertados en la sección 4.2.*

select nombre, tamano, categoria from coleccion where row = "laminas";

1. **neo4j**
   1. **sentencia o consulta de creación del nodo(s) y relaciones**

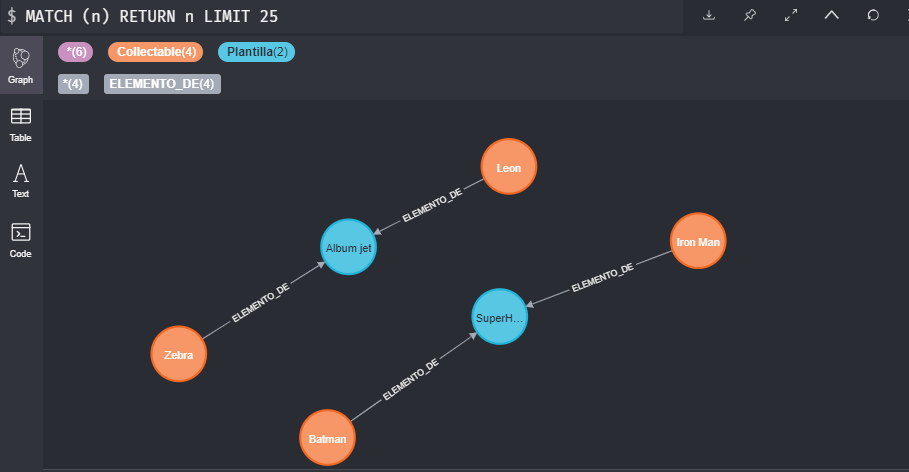
Para la base de datos en Neo4j se escogieron las entidades Coleccionable y Plantillas. Ya que el nombre en español por alguna razón generaba problemas, Coleccionable se cambió por Collectable.

CREATE (Batman:Collectable{ID:1,Nombre:'Batman',Estado:'Adquirido',rareza:'Comun',Descripcion:'Hombre murcielago de DC',CondicionFisica:'Usado'})-[:ELEMENTO\_DE]->(SuperHeroes:Plantilla {ID:345,Nombre:'SuperHeroes',Tamaño:20,Categoria:'Comics',Descripcion:'SuperHeroes de diferentes comics de Marvel y DC'}),(IronMan:Collectable{ID:5,Nombre:'Iron Man',Estado:'No Adquirido',Rareza:'Comun',Descripcion:'Genio, millonario, playboy, filantropo',CondicionFisica:'Nuevo'})-[:ELEMENTO\_DE]->(SuperHeroes)

CREATE

(Leon:Collectable{ID:40,Nombre:'Leon',Estado:'Adquirido',Rareza:'Comun',Descripcion:'Lamina del leon ubicada en el espacio numero 40 del album',CondicionFisica:'Doblado'})-[:ELEMENTO\_DE]->(AlbumJet:Plantilla {ID:123,Nombre:'Album jet',Tamaño:100,Categoria:'Albumes',Descripcion:'Coleccion con las laminas que salen en las chocolatinas jet'}),(Zebra:Collectable{ID:10,Nombre:'Zebra',Estado:'Adquirido',Rareza:'Legendario',Descripcion:'Lamina de la zebra ubicada en el espacio numero 10 del album',CondicionFisica:'En perfectas condiciones'})-[:ELEMENTO\_DE]->(AlbumJet)

* 1. **grafo generado**

******

1. **análisis de lectura**

***Observación****: Considerando el artículo: “A Framework for Migrating Relational Datasets to NoSQL” de Leonardo Rocha et al. (2015). Analice y responda concretamente:*

1. ¿Cuáles son las limitaciones, que se pueden inferir de la lectura, para migrar los conjuntos de datos relacionales a NoSQL? R//:

Una de las limitaciones para la migración de la información a NoSQL es el Volumen de datos que se migrarán, ya que las Organizaciones migran sus bases de datos cuando el volumen de datos almacenados es enorme y los RDBMS ya no satisfacen las expectativas de escalabilidad y alta disponibilidad, esto hace que decidan migrar los conjuntos de datos relacionales a NoSQL porque, NoSQL es una solución adecuada para migrar grandes volúmenes de datos.

Otra cosa que se puede analizar de la lectura es que, el modelo de bases de datos relacional evita la redundancia de datos, que es parte de los modelos de NoSQL, por esa razón se requiere mantener el nuevo modelo de datos semánticamente idéntico al original. Por lo tanto las relaciones existentes deben estar representadas adecuadamente sin pérdida ni distorsión de datos.

Además, existe un costo asociado con la adaptación de las aplicaciones de Software para comunicarse adecuadamente por el nuevo modelo de base de datos.

1. ¿Cuáles limitaciones considera que se deben considerar, a parte de las mencionadas en el artículo? R//:

La mayoría de las bases de datos NoSQL no admiten funciones de fiabilidad, que son soportadas por sistemas de bases de datos relacionales. Estas características de fiabilidad pueden resumirse en: “atomicidad, consistencia, aislamiento y durabilidad.” Esto también significa que las bases de datos NoSQL, que no soportan esas características, ofrecen consistencia para el rendimiento y la escalabilidad.

Con el fin de apoyar las características de fiabilidad y coherencia, los desarrolladores deben implementar su propio código, lo que agrega más complejidad al sistema.

Otras formas de complejidad encontradas en la mayoría de las bases de datos NoSQL, incluyen la incompatibilidad con consultas SQL. Esto significa que se necesita un lenguaje de consulta manual, haciendo los procesos mucho más lentos y complejos.

1. ¿Cuáles son las razones (criterios) que se deben considerar para migrar un conjunto de datos relacionados a NoSQL? R//:

Una de las razones que se deben considerar para migrar un conjunto de datos relacionados a NoSQL es que, NoSQLayer es una solución adecuada para manejar grandes volúmenes de datos.

A diferencia de las bases de datos relacionales, las bases de datos NoSQL están basadas en key-value.

Algunos tipos de almacén de bases de datos NoSQL incluyen diferentes tipos de almacenes como por ejemplo el almacén de columnas, de documentos, de key value store, de gráficos, de objetos, de XML y otros modos de almacén de datos.

También podemos observar que en las Evaluaciones que se realizaron en la lectura, como en la Evaluación Cualitativa, en la cual se tomaron dos escenarios, el primer escenario que es para Operaciones de Consulta, se realizó un almacenamiento de archivos de texto sin formato, después de esto se ejecutó una aplicación que determina diferencias entre ellos y ese resultado se almacenó en otro archivo, y para las consultas que hicieron, obtuvieron que no se produjo ninguna anomalía y con esto NoSQLayer había funcionado correctamente.

El segundo escenario son las Operaciones que cambian el estado de la base de datos, en el cual para cada ejecución de una operación que cambia el estado de la base de datos, se realizó un volcado de las tablas y se almacenaron los resultados en archivos de texto, nombrados según la operación realizada, también se evaluaron la diferencia entre esos archivos, y afectivamente no se encontraron diferencias, lo cual demuestra el funcionamiento adecuado de NoSQLayer.

En cuanto a la Evaluación Cuantitativa se evalúa la eficiencia de NoSQLayer, y si este sigue siendo una opción ventajosa en términos de rendimiento en comparación con el RDBMS, realizando un conjunto de experimentos para medir el tiempo de ejecución de varias transacciones SQL considerando varios escenarios. En el cual NoSQLayer es más eficiente que una base de datos relacional a medida que crece el volumen, obteniendo un buen resultado en ese criterio