**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

Versión: 2020.11.12 20:30

*Observación: Cada vez que agregue nuevos elementos al documento, o que modifique algún componente del informe, revise la coherencia y consistencia con los otros elementos que hacen parte del mismo.*

## Responsables

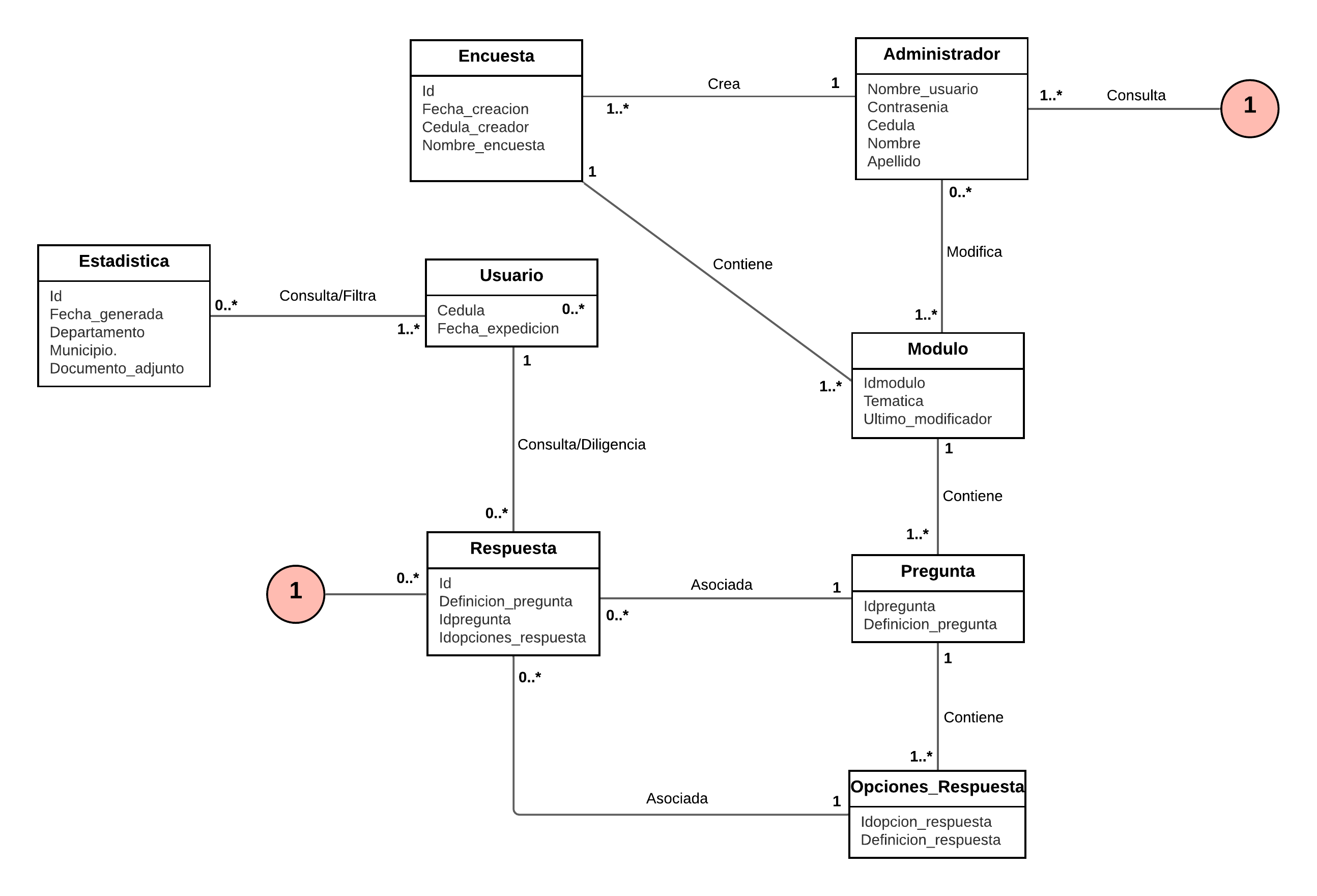
**Equipo ASIS (Análisis de situación de salud)**

|  |
| --- |
| Nombre Completo |
| Brayan Arias Davila |
| Esteban Gomez Benítez |
| Laura Katterine Zapata |
| Juan Manuel Vera Echeverri |
| Mateo Martínez Palacio |
| Geronimo Arboleda |
| Julian Orrego Martínez |

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.

# Modelo del dominio

***Observación****: Incluya el gráfico del modelo del dominio que representa la estructura de su problema.*

**

# Modelo Entidad-Relación

## 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.*

**Tabla Administrador:**

*CREATE TABLE "administrador" (*

*"cedula" INTEGER NOT NULL,*

*"nombre" TEXT NOT NULL,*

*"apellido" TEXT NOT NULL,*

*"nombre\_usuario" TEXT NOT NULL,*

*"contrasenia" TEXT NOT NULL,*

*PRIMARY KEY("cedula")*

*);*

## Sentencias para Insertar datos

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

**Insertar cinco administradores:**

*INSERT INTO administrador VALUES (1037656744, "Mateo", "Martinez", "matmar", "123"), (1193436739, "Julian", "Orrego", "juSKat", "jjo119"), (1001371395, "Juan", "Vera", "juan.vera", "1001371395"), (1000089545, "Esteban", "Gomez", "e.gomez", "egomez99"), (115478965, "Katterine", "Zapata", "katzapata", "ktekis478")*

## Sentencia de consulta

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

***Consultar la cedula, el usuario y la contraseña de un administrador***

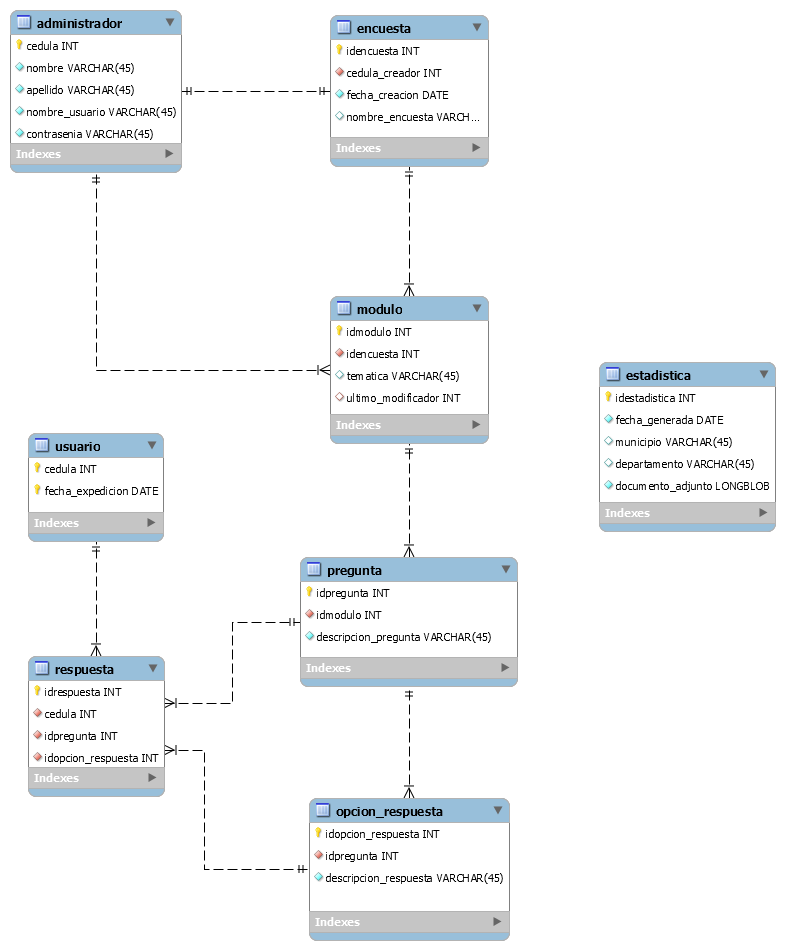
*SELECT cedula, nombre\_usuario, contrasenia FROM administrador;*

Siguiendo los pasos de inserción del apartado 2.2, esta consulta arroja como resultado lo siguiente:



## Toma de pantalla del modelo E-R

***Observación****: lo que se pide.*



# MongoDB

## Sentencia o consulta de creación del documento(s)

***Observación****: Escriba el código en MongoDB para crear al menos cuatro documentos que correspondan a su conjunto de datos específico.*

**Insertar cinco administradores:**

*db.administrador.insertMany([{cedula:1037656744, nombre:"Mateo", apellido: "Martinez", nombre\_usuario:"matmar", contrasenia:"123"},*

*{cedula:115478965, nombre:"Katerine", apellido: "Zapata", nombre\_usuario:"katzapata", contrasenia:"ktekis478"},*

*{cedula:1000089545, nombre:"Esteban", apellido: "Gomez", nombre\_usuario:"e.gomez", contrasenia:"egomez99"},*

*{cedula:1001371395, nombre:"Juan", apellido: "Vera ", nombre\_usuario:"juan.vera", contrasenia:"1001371395"},*

*{cedula:1193436739, nombre:"Julian", apellido: "Orrego", nombre\_usuario:"juSKat", contrasenia:"jjo119"}])*

## Sentencia de consulta

***Observación****: Escriba el código en mongo para consultar 2 documentos (use limit) y muestre dos atributos específicos que usted considere, de los creados en el punto 1.1.*

**Consultar el nombre de usuario y la contraseña de dos documentos de administrador**

*db.administrador.find({}, {nombre\_usuario:1, contrasenia:1}).limit(2)*

# Hypertable

## 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.*

**Crear la tabla administrador, en donde la cedula será utilizada como la clave de la fila**

*CREATE TABLE administrador (nombre, apellido, nombre\_usuario, contrasenia);*

## Sentencias para Insertar datos

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

**Insertar cinco administradores**

*INSERT INTO administrador VALUES ("1037656744", "nombre", "Mateo"), ("1037656744", "apellido","Martinez"), ("1037656744", "nombre\_usuario", "matmar"), ("1037656744", "contrasenia", "123");*

*INSERT INTO administrador VALUES ("115478965", "nombre", "Katterine"), ("115478965", "apellido","Zapata"), ("115478965", "nombre\_usuario", "katzapata"), ("115478965", "contrasenia", "ktekis478");*

*INSERT INTO administrador VALUES ("1000089545", "nombre", "Esteban"), ("1000089545", "apellido","Gomez"), ("1000089545", "nombre\_usuario", "e.gomez"), ("1000089545", "contrasenia", "egomez99");*

*INSERT INTO administrador VALUES ("1001371395", "nombre", "Juan"), ("1001371395", "apellido","Vera"), ("1001371395", "nombre\_usuario", "juan.vera"), ("1001371395", "contrasenia", "1001371395");*

*INSERT INTO administrador VALUES ("1193436739", "nombre", "Julian"), ("1193436739", "apellido","Orrego"), ("1193436739", "nombre\_usuario", "juSKat"), ("1193436739", "contrasenia", "jjo119");*

## Sentencia de consulta

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

**Consultar el nombre, nombre de usuario y contraseña de los administradores**

*SELECT nombre, nombre\_usuario, contrasenia FROM administrador;*

# Neo4j

## Sentencia o consulta de creación del Nodo(s) y Relaciones

***Observación****: Escriba el código en Neo4J para crear el (los) nodo(s) y relaciones que corresponda(n) con su conjunto de datos específico. Debe contener al menos 4 relaciones y 3 nodos.*

**Se creara un grafo que permita visualizar la creación de una encuesta por parte de un administrador y la modificación de uno de los módulos que contiene esta encuesta por dos administradores del sistema.**

*CREATE(Mateo:Administrador {cedula:"1037656744",nombre:"Mateo", apellido:"Martinez", nombre\_usuario:"matmar", contrasenia:"123"})*

*CREATE(Vera:Administrador {cedula:"1001371395",nombre:"Juan", apellido:"Vera", nombre\_usuario:"juan.vera", contrasenia:"1001371395"})*

*CREATE (Encuesta\_Drogas:Encuesta {nombre\_encuesta: "Encuesta de consumo de drogas"})*

*CREATE (Estratificador:Modulo {tematica: "Estratificador"})*

*CREATE*

*(Mateo)-[:CREO {fecha\_creación: "1-11-2020"}]->(Encuesta\_Drogas),*

*(Vera)-[:MODIFICO {fecha\_modificacion: "2-11-2020"}]->(Estratificador),*

*(Mateo)-[:MODIFICO {fecha\_modificacion: "3-11-2020"}]->(Estratificador),*

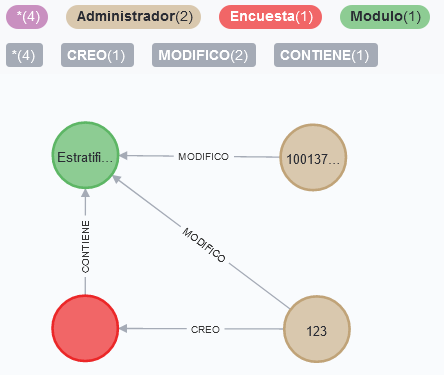
*(Encuesta\_Drogas)-[:CONTIENE]->(Estratificador)*

## Grafo generado

***Observación****: inserte la toma de pantalla del grafo generado.*

*En caso de que no se visualice, el grafo se puede consultar con la query:*

***MATCH (n) RETURN n***



**El nodo rojo representa la encuesta, el verde el modulo estratificador y los beige a los administradores.**

# 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?**

**Respuesta:**

* **Cambios en la estructura del guardado:** Los modelos relacionales suelen guardar la información en estructuras con esquemas bien definidos conocidas como tablas. Aquí, todos los registros deben seguir un mismo estandar de atributos al momento de ser almacenados, haciendo asi que se siga el patron de diseño propuesto para dicha tabla. Para las bases de datos no relacionales como pueden ser las orientadas a documentos, este esquema no es tan estricto y puede variar de documento a documento, apareciendo atributos en unos que no aparecen en otros. Esto no se ve como un problema en el caso del articulo puesto que se preserva la estructura con el framework utilizado pero es mencionado como una posible limitación.
* **Eficiencia de consultas:** Algunas consultas, ya sean de inserción, borrado, actualización o simple consulta de información, pueden tener unos costos computacionales mas bajos en cierto tipo de base de datos. Esto puede depender del tamaño de la información almacenada en la base de datos, redundancia de datos presente o estructura de almacenamiento dentro de las tablas-documentos-grafos.
* **Cambios en el código de la aplicación:** A pesar de que el framework mencionado en el artículo no requiere de cambios en el código utilizado en la aplicación para comunicarse con la base de datos, se infiere que al cambiar de un modelo relacional a uno NoSQL este cambio de código es necesario. Por ejemplo, cambiar en el código las consultas utilizadas. De un *SELECT* en SQL a un *DB.FIND* en mongo, líneas de conexión a la base de datos, permisos y demás.
* **Costos (Dinero):** En muchas ocasiones la migración de información de un tipo de base de datos a otra es un proceso costoso y que puede no brindar mucha mejoría a la empresa-organización. Es necesario realizar un análisis exhaustivo de estructura, tipos de consultas comunes y demás, antes de comenzar con la migración para así comprobar que verdaderamente aporta un beneficio significativo.
* **Grandes volúmenes de datos:** La mayoría de las compañías deciden migrar sus bases de datos relacionales a NoSQL cuando se dan cuenta de que su sistema de gestión base de datos relacional no satisface la escalabilidad y alta disponibilidad que necesitan, haciendo así que esto sea un limitante a considerar cuando a capacidad se refiere, a la hora de migrar grandes volúmenes de información.

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

**Respuesta:**

* **Datos:** Aspectos como las fuentes que proporcionan los datos, operaciones de depuración y limpieza sobre estos y la capacidad de almacenamiento de la que se disponga en el servidor, son limitaciones que también se deben tener en cuenta al momento de considerar realizar una migración efectiva.
* **Usuario**: También se deben considerar las necesidades del cliente final y los requisitos que planteo en un primer momento. Si bien es cierto que el proceso de migración es realizable, puede que no sea lo más adecuado para las condiciones del usuario final. En algunos casos, continuar con un modelo relacional y tener los datos almacenados sin ningún tipo de redundancia, puede que sea lo mas adecuado, mientras que en algunos otros para evitar costos computacionales que se traducen en tiempos de espera, se requiere almacenar información sin normalizar o desestructurada.
* **Tiempo:** El tiempo necesario para la migración puede ser una limitación importante por considerar. Es bien conocido en la industria que muchas empresas deciden no migrar sus sistemas a nuevas tecnologías o simplemente cambiar a una diferente. Estos procesos son bastante largos y costosos, e incluso puede que no reporten un beneficio para el usuario final. Es por esto por lo que este análisis de tiempo requerido y eficiencia debe ir acompañado junto con el de costos para así asegurar que el proceso sea adecuado.

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

**Respuesta:**

Son numerosos criterios que se deben tener en cuenta al momento de pensar en migrar una base de datos SQL a NoSQL que influyan en la mejora del Software que se está manejando. Como importantes se tienen los siguientes:

* Nuevos requerimientos del usuario
* Mejoras en el funcionamiento del sistema
* Interoperabilidad
* Aumento de volumen de datos
* Mejoras de seguridad y control de información
* Eficiencia
* Escalabilidad y flexibilidad
* Partición y replicación de datos.

Estos criterios pueden llevar a considerar la posible migración de una base de datos relacional a una NoSQL. En general, si se tiene un gran volumen de datos, que crece de manera rápida, en donde se necesita eficiencia en las consultas, una escalabilidad constante y una posible flexibilidad en el esquema de la estructura de la base de datos, es adecuado analizar la posibilidad de migrar de una base de datos a otra a pesar de que involucre recursos como tiempo y dinero, puesto que a la larga será un beneficio significativo para la empresa.