



CASO PRÁCTICO:
BASES DE DATOS NOSQL - DATOS
GENERADOS DENTRO DE UNA
CIUDAD

Trabajo elaborado por:

• **Victor Hugo Pacheco Padilla**

Fecha de Entrega: 15 de octubre de 2023

OBJETIVO GENERAL

Diseñar un sistema o combinación de sistemas que permitan el adecuado almacenamiento de la información recabada dentro de diferentes eventos cotidianos que suceden en la actividad diaria de una ciudad.

OBJETIVOS PARTICULARES

- Poder almacenar en el sistema información sobre cualquier tipo de evento que se va a producir en la ciudad (deportivos, conciertos, manifestaciones, desfiles, etc).
- Recabar las emergencias que se reciben a través del sistema 112 (911)
- Almacenar las ubicaciones de los autobuses que se tienen monitorizados, mediante el uso de diferentes sensores.
- Analizar todos los Tweets (públicos) que se publiquen, que cuenten con determinadas etiquetas.
- Identificar problemas potenciales antes de que puedan llegar a ser problemas mucho más graves
- Conocer la opinión de la gente acerca del transporte, de los eventos de ocio, etc.
- Buscar mejorar la calidad de la ciudad y de sus habitantes.

INTRODUCCIÓN

Las ciudades representan un desafío enorme a la hora de realizar estimaciones o cuantificaciones de la información que se genera en las mismas, ya que, a diferencia de lo que ocurre en otro tipo de entornos (rurales, por ejemplo), las ciudades están inmersas en una dinámica donde existen cambios constantes de las situaciones y entornos que se tienen. Esto propicia, además, que los datos arrojados por cada situación particular se encuentren en constante cambio, se generen en gran cantidad, y además, sean de diferentes características.

El entorno citadino y los desarrollos urbanos pasan por una serie de retos constantes para poder mejorar la calidad de vida de sus habitantes, y dado que la mayoría de las ciudades no cuentan con una planificación adecuada, es importante conocer los desafíos que las ciudades enfrentan.

La dinámica urbana, y las diferentes actividades diarias que realiza la población propician la generación de una gran cantidad y variedad de datos, desde, por ejemplo, los avisos de tráfico que recibimos a través de las aplicaciones de mapas (como Google Maps o Waze), el concierto programado de algún artista internacional en alguna arena, o incluso las grabaciones de las cámaras de los sistemas de seguridad para poder atacar los problemas de seguridad existentes.

Duarte, F. y De Souza, P. (2020) mencionan que, a diferencia de otros fenómenos analizados por medio de la Ciencia de Datos, en el caso de las ciudades no solo se busca entender a la sociedad, sino también transformarla a través de la adecuada planificación, y examinar los límites que tiene el Big Data en la manera en la que

se toman decisiones y sus consecuencias a corto y largo plazo. Muchas veces, la manera en la que se cuantifica o se toman decisiones en las ciudades no siempre se hace de la manera en la que se debe, sino de la manera en la que se encuentra disponible, principalmente en el hecho de que los datos disponibles no son necesariamente los datos adecuados para realizar los estudios (ej. utilizando las conexiones de los usuarios a torres celulares para medir la exposición a contaminantes atmosféricos).

Es necesario que la información que fluye hacia los data lakes sea la necesaria para el procesamiento adecuado de la información, independientemente del tipo y cantidad, y encontrar la manera en que esta información sea veraz, pueda aportar un beneficio real y tenga un impacto social que funcione para mejorar la calidad de vida de sus habitantes, de la mano de nuevos métodos que puedan entender estos fenómenos, y con un proceso multidisciplinario junto con otros profesionales.

METODOLOGÍA Y RESULTADOS

Para poder lograr el objetivo de combinar diferentes fuentes y tipos de datos para el análisis y generación de alertas en una ciudad, se necesitará una plataforma de gestión de datos y análisis que permita manejar esta diferencia en la información a suministrar. Se propone la siguiente metodología para configurar el sistema:

A) Recopilación de datos:

- **Eventos de la Ciudad:** Se recopilarán datos sobre eventos programados en la ciudad a través de fuentes públicas, sitios web de eventos, aplicaciones de eventos y notificaciones de organizadores. Estos datos pueden ser generados mediante APIs de cada sitio web de organizadores, o ingresados mediante archivos CSV o Excel.
- **Emergencias 112 (911):** Los datos de emergencias se pueden recibir a través del sistema de gestión de emergencias 112 (911) de la ciudad. Se usará la información recopilada mediante archivos CSV.
- **Ubicaciones de Autobuses:** Los sensores instalados en los autobuses pueden enviar datos de ubicación en tiempo real a través de una red de comunicación, utilizando sistemas como GPS, y por lo tanto, generando coordenadas geográficas, que pueden ser interpretadas para ese fin por APIs como Google Maps y OpenStreetMap, o ingestadas como raw data numérico.
- **Tweets Públicos:** Puedes utilizar la API de X (antes Twitter) para recopilar tweets públicos que contengan etiquetas relevantes para su análisis.

B) Almacenamiento de Datos - Desarrollo de la Base de Datos:

Debido a que se manejarán diferentes tipos de datos provenientes de fuentes distintas, se propone como uso una base de datos no relacional, de tipo NoSQL,

en este caso la base de datos que se propone es **MongoDB**, de tipo documental, ya que cada fuente de datos **puede ser organizada como un “documento” diferente** de acuerdo con las características de la información que se quiera almacenar. Debido a que se van a almacenar diferentes tipos de datos, y con cambios constantes, MongoDB nos permite mantener flexibilidad al juntar los datos sin tener que mantener bases de datos separadas.

Los pasos a seguir para configurar nuestra bases de datos deseada (dentro de MongoDB) son los siguientes:

1. Crear la base de datos desde el shell de MongoDB:

```
use DBCiudad
```

2. Crear las colecciones respectivas de cada recopilación de datos a integrar en nuestra base “DBCiudad”:

```
db.createCollection("eventos")
db.createCollection("emergencias")
db.createCollection("ub_buses")
db.createCollection("tweets")
```

3. Insertar dentro de cada colección los documentos respectivos que usaremos para manejar los datos generados. Se propone la siguiente estructura para cada documento por colección:

```
db.eventos.insertOne(
{
  "nombre": "Informe de Gobierno - Alcalde",
  "fecha": "2023-11-15",
  "ubicacion": {
    "latitud": 123.456,
    "longitud": 789.012
  },
  "categoria": "Gobierno",
  // Se pueden agregar otros campos relevantes de acuerdo con el evento a realizar
}
)
```

```
db.emergencias.insertOne(
{
  "tipo_emergencia": "Incendio",
  "fecha": "2023-11-15",
  "ubicacion": {
    "latitud": 456.789,
    "longitud": 123.456
  },
  "pers_afectadas": 23,
  "heridos": 4,
  "fallecidos": 0,
  // Se pueden agregar otros campos relevantes de acuerdo con la incidencia
}
)
```

```

db.ub_buses.insertOne(
{
  "id_autobus": 12345,
  "ubicacion": {
    "latitud": 789.012,
    "longitud": 234.567
  },
  "fecha": "2023-11-15T10:30:00",
  // Se pueden agregar campos adicionales de acuerdo con necesidad
}
)

db.tweets.insertOne(
{
  "contenido": "El Alcalde dará su Informe de Gobierno 2023, ¡No te lo pierdas! #InformeAlcalde2023",
  "fecha": "2023-11-15T19:45:00",
  "autor": "usuario_twitter",
  "etiquetas": ["Informe", "Gobierno", "CDMX"],
  // Se pueden agregar campos adicionales de acuerdo con necesidad
}
)

```

Hay que considerar que la base de datos anterior (en MongoDB) únicamente está estructurada dejándola **lista para almacenar los datos**, sin realizar la conexión con las APIs respectivas o el ingreso de la información en formatos CSV o XLSX, pasos que se pueden realizar **por medio de Python** con el uso de librerías como **json** (para la lectura de las APIs de internet), o con el apoyo de frameworks como **Apache Spark o Apache Hadoop** (dependiendo la necesidad).

C) Ventajas y desventajas de este sistema vs otros sistemas:

El uso de este sistema de tipo documental presenta diversas ventajas, principalmente la flexibilidad del mismo, que permitirá a la ciudad y a sus responsables recopilar, analizar y actuar sobre datos generados en tiempo real, de tal manera que se podrían tomar mejores decisiones para mejorar la calidad de vida de los ciudadanos, además de que, en caso de requerirse por necesidad modificar las estructuras de datos, se puede hacer de manera rápida y sin modificar considerablemente la base, únicamente modificando las colecciones o los documentos respectivos, cosa diferente de lo que sucedería, por ejemplo, en una base de datos relacional.

Una desventaja grande de MongoDB es que no permite hacer joins para consultas, cosa que la diferencia notablemente de las bases de datos relacionales SQL, o de aquellas basadas en grafos, como Neo4j, cuya principal ventaja son este tipo de relaciones entre los datos que se presentan. Adicionalmente, para transacciones complejas donde se deba priorizar mayor estabilidad, MongoDB puede no ser idóneo debido a su estructura, y una base de datos relacional (como las basadas en SQL) puede ser una opción más acorde a lo requerido.

CONCLUSIONES

Al trabajar con situaciones reales de desarrollo de bases de datos, se deben de considerar diversos factores a la hora de decidir qué modelo y qué tipo de sistema es el más idóneo para cada desarrollo particular. En el caso de las ciudades, el hecho de poder trabajar con información en tiempo real (como la que se menciona en el caso práctico) y con diferentes tipos de datos provenientes de diferentes fuentes es un reto que requiere que una base de datos pueda mantenerse fiable y con un funcionamiento que reduzca al mínimo los posibles errores que puedan surgir. De esta manera, y estudiando la información generada, se podrá ayudar a prevenir problemas antes de que se conviertan en situaciones más complicadas o difíciles de atender para los habitantes, y reducir el impacto de los problemas que las ciudades tienen, y de la planificación a futuro que se pueda realizar.

BIBLIOGRAFÍA

- Duarte, F. & De Souza, P. (2020). *Data Science and Cities: A Critical Approach*. Harvard Data Science Review. Recuperado el 14/10/2023, de <https://hdsr.mitpress.mit.edu/pub/1um18ajd/release/2>
- *MongoDB Tutorial*. (2023). W3Schools. Recuperado el 10/10/2023, de <https://www.w3schools.com/mongodb/>
- Schelli, L. (2022). *Trabajar con MongoDB*. IMMUNE Technology Institute. Recuperado el 10/10/2023, de https://immune-content.learningcloud.me/api/content_managers/authoring_units/contents/review/app?id=30ee6b69-be53-4631-89f5-f510c8494edf&p_reviewToken=caa6ebf0a43cad7381afabf07f77fe82#
- Martín Gómez, P. (2020). *Ventajas y desventajas de MongoDB*. OpenWebinars. Recuperado el 14/10/2023, de <https://openwebinars.net/blog/ventajas-y-desventajas-de-mongodb/>