**Bases de Datos SQL y NoSQL**

**SQL**

* **Estructura:** Las bases de datos SQL siguen un modelo de datos relacional, donde la información se organiza en tablas con filas y columnas.
* **Lenguaje:** Utilizan SQL como lenguaje estándar para realizar consultas y manipular los datos.
* **Transacciones:** Son adecuadas para aplicaciones que requieren transacciones complejas y garantías ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).
* **Ejemplos de bases de datos SQL:** MySQL, PostgreSQL, SQL Server, Oracle.

**NoSQL**

* **Estructura:** Las bases de datos NoSQL ofrecen un modelo de datos más flexible y pueden almacenar datos no estructurados o semiestructurados (como texto plano, CSV, XSLX, etc.). No siguen el modelo relacional de las bases de datos SQL.
* **Tipos:** Hay varios tipos de bases de datos NoSQL, incluyendo las de tipo documental, de clave-valor, de columna amplia y de grafo.
* **Escalabilidad:** Suelen ser altamente escalables y pueden manejar grandes volúmenes de datos distribuidos.
* **Ejemplos de bases de datos NoSQL:** MongoDB, Cassandra.

**Implementación**

* **SQL:** Se pueden utilizar módulos como mysql y pg para PostgreSQL, mssql para SQL Server.
* **NoSQL:** Se pueden emplear librerías como mongoose en el caso de MongoDB para facilitar la interacción desde Node.js.

**Requerimientos (Servicio de Notificaciones)**

Dado que el servicio se dedicará exclusivamente a enviar notificaciones a través de diversos canales (SMS, Push, Email), es esencial considerar los siguientes puntos al elegir la base de datos:

* **Flexibilidad:** Para almacenar información sobre las notificaciones de diferentes tipos de canal, optaremos por una base de datos NoSQL que permita una estructura flexible y la capacidad de almacenar información diversa sin una rigidez en el esquema de datos.
* **Escalabilidad del Servicio:** Con la expectativa de manejar grandes volúmenes de datos debido a la naturaleza de envío masivo de notificaciones, una base de datos NoSQL ofrece una mejor capacidad de escalabilidad horizontal, lo que permite distribuir la carga de trabajo a través de múltiples servidores para manejar el aumento del tráfico.
* **Modelado de Datos:** En el caso de una base de datos NoSQL como MongoDB, un modelo de datos adecuado podría consistir en una colección principal para las notificaciones, donde cada documento representa una notificación enviada. Dentro de cada documento, podrías incluir campos para el tipo de notificación (SMS, Push, Email), el contenido del mensaje, el destinatario, la fecha de envío, entre otros detalles relevantes. Esto proporciona una estructura flexible que puede adaptarse a las diferentes necesidades de cada tipo de notificación, manteniendo la capacidad de consulta eficiente y escalabilidad.

**Principio del formulario**