

Tarea 1 – Investigación

López Sugahara Ernesto Danjiro

I. MODELO DE DATOS ORIENTADO A OBJETOS

En esencia, el modelo de datos orientado a objeto surge como respuesta al crecimiento de complejidad de proyectos de desarrollo de software, en los que -hoy en día- se utiliza un paradigma de programación orientado a objetos, es decir, la información del mundo real se abstrae en forma de un objeto definido a través de sus atributos (características del objeto: color, tamaño, peso, etc.) así como sus métodos (comportamientos del objeto). Estos bloques de información permiten modelar un sistema más grande y complejo.

Con esto, un modelo de este tipo busca representar los datos como objetos. Lo anterior permite expresar el comportamiento, estado y relación entre los objetos de un sistema. Estos objetos tienen una identidad y comportamiento propios, los cuales pueden ir variando a lo largo del funcionamiento de un sistema, por lo que es importante conocer el estado del objeto que se está utilizando; este es el principal punto de interés para este modelo de datos.

a. Sistema de manejador de bases de datos orientadas a objetos

Estos sistemas deberán de cumplir con las características básicas de cualquier manejador de base de datos, sin embargo, también será necesario la implementación de ciertos mecanismos que permitan modelar los datos en forma de objetos. Esto se puede visualizar a través del factor de que una DBMS orientada a objetos deberá de tener la posibilidad de resguardar y trabajar los métodos relacionados a ciertos objetos.

Algunos ejemplos de estos manejadores son:

- MongoDB
- Apache Cassandra
- ObjectDB
- Objective
- GemStone/S

b. Principales ventajas y desventajas

Ventajas	Desventajas
Permite manipular datos más complejos y de tipos variados.	No está bien definido, es decir, no hay un sustento teórico sólido para implementación.
Para lenguajes con paradigma orientado a objetos nos permite salvar y recuperar datos (objetos) de manera más sencilla y rápida.	Falta de estándares.
Al estar basado en un paradigma orientado a objetos, es más sencillo modelar información relacionada a problemas avanzados del mundo real	Poca gente especializada debido a la falta de estándares y teoría.
	Es mucho más complejo de implementar.

II. MODELOS NOSQL

Al igual que el modelo orientado a objetos, los modelos NoSQL surgen como una respuesta a la necesidad de desarrollar sistemas de datos más flexibles que permitieran capturar y ordenar mayor cantidad de información que no tiene una estructura o esquema definido (es decir, no es posible definir los datos a partir de tablas, filas y columnas – esto como producto de la alta demanda tecnológica de la sociedad en la actualidad -.

a. Sistema de manejador de bases de datos NoSQL

Deben de cumplir con las funcionalidades de un DBMS, sin embargo, deberá y de contemplar características de mayor flexibilidad de la información. Los sistemas más comunes hoy día son:

- MongoDB
- Apache Cassandra
- CouchDB
- Neo4j

Con base en esta información, se tienen tres tipos principales de bases de datos catalogadas como NoSQL:

b. Bases de datos clave-valor

En este modelo los datos se toman como un conjunto de pares “clave – valor”, algo muy similar a los diccionarios o mapas hash de ciertos lenguajes de programación. Por lo tanto, en este modelo las claves fungen como identificadores únicos que permiten acceder a ciertos valores de interés.

Algunos de los sistemas manejadores más comunes son:

- Cassandra
- Redis

Ventajas	Desventajas
Almacenar información de manera más flexible, es decir, sin una estructura de tabla.	No hay un esquema y estándar generalizado.
Debido a la propiedad de las claves, la modificación y acceso a la información es de las más eficientes y sencillas.	El lenguaje de consulta es muy básico (a través de la clave) por lo que no es posible una gran profundización.
Mayor velocidad y escalabilidad.	Existe mucho menos documentación al respecto.

c. Bases de datos documentales

Es un modelo que orienta su funcionamiento en datos semiestructurados que se encuentran alojados en documentos, es decir, almacena todos los registros y los datos relacionados en un solo documento que contiene el mismo tipo de registros. Por ejemplo, si se quiere almacenar la información de diversos alumnos, se tendrá un archivo que contenga diferentes registros de alumnos con su respectiva información, la cual no se encontrará estructurada en forma de tabla.

Los sistemas de manejo más reconocidos de este tipo de bases de datos son:

- MongoDB
- CouchDB

Ventajas	Desventajas
Almacenar información de manera más flexible, es decir, sin una estructura de tabla.	No hay un esquema y estándar generalizado.
Mayor facilidad de adición y actualización de datos.	No siempre se garantizan las propiedades de atomicidad, consistencia, integridad y durabilidad.
Útiles para el almacenamiento de grandes volúmenes de información.	Existe mucho menos documentación al respecto.

e. Bases de datos de grafos

Como el nombre lo indica, este tipo de modelo permite representar los datos en forma de grafos, lo cual brinda una estructura de información mucho más compleja. Estas son muy utilizadas cuando los datos tienen una gran interrelación entre elementos.

Los sistemas más comunes son:

- Neo4j
- HyperGraphDB
- InfoGrid

Ventajas	Desventajas
Permite analizar con mayor detalle y facilidad la relación entre datos.	No siempre es la mejor solución para aplicaciones, por lo que su implementación podría resultar deficiente.
Tienen una gran rapidez de consulta debido a la estructura de los grafos.	En términos de actualización de la información, puede llegar a ser poco eficiente debido a la fuerte interrelación de elementos.
Se puede relacionar y utilizar fácilmente con tecnologías emergentes relacionadas al Deep Learning.	Es necesario analizar de manera profunda si es necesario implementar un sistema de este tipo, analizando qué tipo de relaciones tendrán los datos de una aplicación.

III. BIBLIOGRAFÍA

[1] Hernández, E. D. K. (s.f.). Modelo orientado a objetos. [Online]. https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod_resource/content/8/contenido/index.html

[2] Kumar, L. (2019). Object-oriented Database Management System (OODBMS). [Online]. <https://www.techtarget.com/searchoracle/definition/object-oriented-database-management-system>

[3] AWS (s.f.). Bases de datos no relacionales. [Online]. <https://aws.amazon.com/es/nosql/>

[4] GraphEverywhere. (2019). Bases de datos NoSQL | Qué son, marcas, tipos y ventajas. [Online]. <https://www.grapheverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/>