

DB4O – Database for objects

DB4O, es un motor de base de datos orientada a objetos, sus siglas se corresponden con la expresión “DataBase 4 (for) Objects” este es un producto con un alto rendimiento más aún en modo embebido.

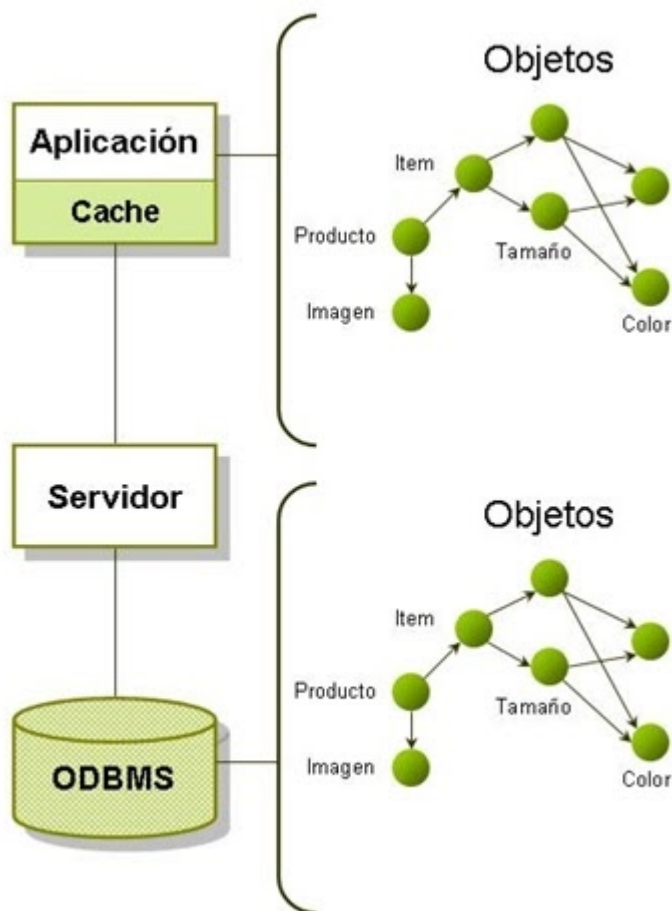
DB4O propone un modelo de desarrollo que proporciona a las aplicaciones en su capa de acceso a datos un abandono completo del paradigma relacional de las bases de datos tradicionales. En efecto, deja de existir un lenguaje SQL de consultas o modificaciones para pasar a crearse sistemas de consulta por métodos delegados y actualización, creación, borrado automático de entidades mediante código compilable.

No obstante, este producto está impulsando a algunas de las aplicaciones más exigentes del mundo para compañías de diversos sectores como se muestra a continuación:

- **Telecomunicaciones:** Alcatel-Lucent, Deutsche Telecom, France Telecom, Ericsson, NEC, Nortel, Orange, Samsung, etc.
- **Finanzas:** Financial Times, New York Stock Exchange, Dow Jones, Reuters, Londres Clearing House, Bank of America, etc.
- **Transporte:** Sabre, GE ferrocarriles, ferrocarriles BNS, Galileo, etc.
- **Defensa:** Raytheon, Northrop Grumman, Lockheed, etc.
- **Bioinformática:** May Clinic, St. Jude medical, Eidogen, etc.

DB4O, tiene una **licencia dual GPL/comercial** que permite desarrollar software de código abierto y libre, si se desea aplicar a un software privativo, se aplica otro modelo de licenciamiento concreto.

Los usuarios reducen obtienen muchas ventajas en la eficiencia en el acceso a datos, menos datos de indexación, diseño simple, evitando el mapeo como en un ORM.



En tal sentido, el esquema de datos del dominio viene representado por la implementación que se realice del diagrama de clases. Por ello, se consigue evitar el problema del DESFASE OBJETO RELACIONAL sin sacrificar el rendimiento que los ORM (Mapeado Objeto Relacional) sufren actualmente para llevar a cabo el mismo objetivo.

CARACTERÍSTICAS DB4O

La principal característica de DB4O es ser una base de objetos nativa de alto rendimiento para .Net o Java con función embebida y remota. En efecto, podemos citar ventajas y desventajas propias de este gestor:

VENTAJAS

Alto rendimiento, velocidad de desarrollo gracias a la transparencia que ofrece, bajo consumo de recursos, lo hace ideal para dispositivos móviles y entornos Clientes/Servidor.

- Doble licencia GPL (Open Source) y licencia comercial que incluye soporte.
- Documentación Clara, amplia y ordenada. Orientada a ejemplos y de fácil lectura.
- No existe mapeo entre objetos y tablas.
- Características ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) que garantizan el éxito de las transacciones.
- Modos de trabajo Embebido y Cliente/Servidor.
- Código de acceso a la base sencillo, entendible acorde a paradigma orientado a objetos.
- Mejor rendimiento con objetos de negocio complejos (árboles, estructuras anidadas, relaciones N a N, relaciones recursivas)
- Es posible realizar de forma fácil el Backup de la base de datos debido a que esta se encuentra en un solo archivo.
- Sistema de versionado automático
- Las búsquedas se hacen directamente usando objetos, de la misma forma efectuar consultas es posible utilizando QBE, (búsquedas comparando un objeto plantilla), Native Query (búsquedas nativas) SODA Query API (búsquedas de bajo nivel).
- Cambios en la estructura de base de datos orientada a objetos como agregar o quitar atributos a una clase se aplican directamente en la base, sin tener que migrar datos ni realizar reconfiguraciones.
- Multiplataforma, además de poder manejarse nativamente en Java y .Net
- Ofrece un motor de base de datos embebida para dispositivos móviles en entornos OO

DESVENTAJAS

No existe un lenguaje de consultas similar a SQL estas tareas deben realizarse programáticamente.

No existen restricciones, deben programarse debido a que el gestor no implementa integridad referencial.

El tamaño de la base de datos es teóricamente limitado a ficheros de 264GB, no obstante se garantiza un acceso fluido y eficiente hasta con ficheros de 10GB.

- No permite clustering.
 - No existe una gestión de permisos.
 - No tiene aplicaciones para data-mining ni informes u otras herramientas que están presentes en gestores relacionales.
 - La explotación de los datos existentes se debe realizar mediante otros medios.
 - La concurrencia de usuarios realizando la misma petición o ejecutando el mismo proceso es limitada a comparación de otros gestores comerciales.
-