

Realm

BASES DE DATOS ORIENTADAS A OBJETOS.





Agustín Jiménez polonio.

Santos Castillo PÉREZ.

ángel serrano garcía.

carlos aguilar alférez.

**ÍNDICE**

[1. INTRODUCCIÓN. BASES DE DATOS ORIENTADAS A OBJETOS. 1](#_Toc61971549)

[2. ¿Qué es Realm? 2](#_Toc61971550)

[3. HISTORIA. 2](#_Toc61971551)

[4. Ventajas. 3](#_Toc61971552)

[5. Desventajas. 4](#_Toc61971553)

# INTRODUCCIÓN. BASES DE DATOS ORIENTADAS A OBJETOS.

Se trata de un modelo de base de datos en la que los datos están almacenados como objetos en lugar de tablas relacionales o documentos, exactamente igual como en la programación orientada a objetos. Cuando se integra las características de una base de datos con las de un lenguaje de programación orientado a objetos, el resultado es un sistema gestor de base de datos orientada a objetos (ODBMS, Object Database Management System). Un ODBMS hace que los objetos de la base de datos aparezcan como objetos de un lenguaje de programación en uno o más lenguajes de programación a los que dé soporte. Un ODBMS extiende los lenguajes con datos persistentes de forma transparente, control de concurrencia, recuperación de datos, consultas asociativas y otras capacidades.

La iniciativa ODMG (Object Database Management Group) proporciona estándares sobre base de datos orientada a objetos.

Las bases de datos orientadas a objetos se diseñan para trabajar bien en conjunción con lenguajes de programación orientados a objetos como Java, C#, Visual Basic.NET, C++, JavaScript, ...

Los ODBMS son una buena elección para aquellos sistemas que necesitan un buen rendimiento en la manipulación de tipos de dato complejos.

Los ODBMS proporcionan los costes de desarrollo más bajos y el mejor rendimiento cuando se usan objetos gracias a que almacenan objetos en disco y tienen una integración transparente con el programa escrito en un lenguaje de programación orientado a objetos, al almacenar exactamente el modelo de objeto usado a nivel aplicativo, lo que reduce los costes de desarrollo y mantenimiento.

## ¿Qué es Realm?

Realm es un motor de base de datos pensado para ser utilizado en el desarrollo de aplicaciones móviles tanto para Android como para sistemas iOS, pero además con compatibilidad con distintos lenguajes; Java, Kotlin, Swift y Objective-C, React Native, la plataforma Xamarin (Microsoft) y NodeJS y con licencia Apache.

Este sistema nos permite crear bases de datos relacionales de forma sencilla además de tratarse de un sistema gratuito que puede ser descargado desde su propia web.

Realm usa un lenguaje de consulta avanzada que se basa en el cifrado AES256 (un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos, creado en Bélgica).

Aparte Realm incorpora el uso de transacciones sin olvidar que trabaja como un ORM (Object-Relational Mapping) , es decir, no habría que utilizar el lenguaje SQL para realizar las consultas a la base de datos, sino que todo se hace por medio de objetos, tanto la obtención de resultados de las consultas como a la hora de actualizar o insertar datos.

# HISTORIA.

El desarrollo de Realm comenzó a finales de 2010 por Alexander Stigsen, junto con Bjarne Christiansen, bajo el nombre de TightDB.

La empresa comenzó en 2011. Se promocionó como NoSQL (una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico de SGBDR) con durabilidad configurable y la capacidad de compartir los mismos grupos de datos en múltiples procesos, pero también en múltiples dispositivos y clústeres (Grupo de empresas interrelacionadas que trabajan en un mismo sector industrial y que colaboran estratégicamente para obtener beneficios comunes).

TightDB cambió el nombre de su producto a Realm en septiembre de 2014 y lo lanzó para pruebas públicas.

En marzo de 2015, se reveló una financiación de unos 20 millones de dólares. Realm fue mencionado en alguna prensa especializada, incluso por otras firmas como IBM (International Business Machines, una reconocida empresa multinacional estadounidense de tecnología y consultoría con sede en Armonk, Nueva York). Realm anunció la versión 1.0 en junio de 2016 y lanzó una plataforma para la sincronización bidireccional en tiempo real (versión beta en septiembre de 2016, lanzamiento en enero de 2017) y proporcionó un SDK de Node.js para aplicaciones del lado del servidor.

En mayo de 2017, se anunció la compatibilidad con UWP (Una plataforma informática creada por Microsoft y presentada por primera vez en Windows 10. El propósito de esta plataforma es ayudar a desarrollar aplicaciones universales que se ejecuten en Windows 10, Windows 10 Mobile , Xbox One sin la necesidad de reescribirlas para cada una Plataforma universal de Windows.

# Ventajas.

* No habría que utilizar el lenguaje SQL para realizar las consultas a la base de datos, sino que todo se hace por medio de objetos, tanto la obtención de resultados de las consultas como a la hora de actualizar o insertar datos, se haría por medio de objetos, por lo cual, destaca por su sencillez.
* Ofrece la adquisición de una API simple, mientras se mejora el rendimiento, el cual no se sacrifica por otras herramientas o acciones que trabaja el sistema.
* Su rendimiento se considera óptimo gracias a la asignación de memoria (enorme velocidad), a el motor de almacenamiento y a la carga lenta que hacen del trabajo algo fluido y rápido. Se le considera más rápido que un ORM (Asignación objeto-relacional), más fluido y veloz que SQLite (un sistema de gestión de bases de datos relacional contenida en una relativamente pequeña ​ biblioteca escrita en C. SQLite es un proyecto de dominio público​ creado por D. Richard Hipp), la base de datos móvil más famosa.
* Realm tiene un Listener que te permite saber cuándo ocurrió un cambio en cualquier objeto que cumpla las condiciones de una Query.
* Tiene un adapter para la RecyclerView que es mágico, en donde se podría decir que es como la versión local de lo que es Firebase.
* Documentación excelente, extensa y completa.
* Las tablas se crean solas.

# Desventajas.

* Algo incomodo de Realm es que hay que abrir una instancia y después limpiarla (generalmente en onStop).
* Ante el aprendizaje, no basta con leer el ejemplo para sacar provecho, hay que leer bien.