Exposición de un SMBD NoSQL

RavenDB

Arévalo Coronado María José
Hernández Morales Anahí
Román Ruiz María Celeste

Propósito y cómo sirve a los usuarios

Para RavenDB, la base de datos debe habilitar a los desarrolladores y no debe jugar un papel importante en el proceso de desarrollo y mantenimiento. Por lo tanto, su **propósito** consiste en realizar el trabajo duro del desarrollo mientras el usuario disfruta de la tranquilidad sin concesiones.

- **Para los desarrolladores:** RavenDB está diseñado para hacerles la vida más fácil. El esquema es flexible al usar documentos, además el auto-indexado reduce la carga de optimización.
- **Para las empresas:** Proporciona una solución robusta y escalable para gestionar datos. La alta disponibilidad asegura que la aplicación esté siempre en línea.
- **Para el usuario final:** Se beneficia de una experiencia de usuario rápida y fluida. La arquitectura de RavenDB permite que las aplicaciones web y móviles carguen información casi instantáneamente.

RavenDB

Principales Funcionalidades

- **Modelo de Datos Documental:** RavenDB se centra en documentos JSON. Esto permite almacenar datos de una manera muy flexible, similar a cómo los objetos se representan en el código. No se necesita un esquema rígido.
- **Transacciones ACID:** RavenDB garantiza la atomicidad, consistencia, aislamiento y durabilidad (ACID). Esto es fundamental para la integridad de los datos, ya que asegura que las transacciones sean confiables y seguras, incluso en operaciones distribuidas.
- **Indexación Automática e Inteligente:** RavenDB puede aprender de las consultas propias y crear y optimizar índices automáticamente. Esto hace que las búsquedas sean rápidas sin tenerque gestionar manualmente todos los índices, aunque también se pueden crear índices estáticos para tener un control total.

RavenDB

Principales Funcionalidades

- **Raven Query Language:** Para las consultas, utiliza un lenguaje similar a SQL llamado RQL. Es intuitivo y poderoso, lo que facilita a los desarrolladores con experiencia en SQL hacer consultas complejas sobre datos documentales.
- **Alta Disponibilidad y Escalabilidad:** RavenDB permite la replicación en tiempo real y la gestión de clústeres para asegurar la alta disponibilidad y la tolerancia a fallos.
- **Seguridad:** La documentación resalta el cifrado transparente de los datos, tanto en reposo (cuando están guardados) como en tránsito (mientras se mueven por la red), garantizando la protección de la información sin esfuerzo adicional del desarrollador.
- **Soporte Multi-modelo:** Es principalmente una base de datos de documentos, pero también soporta otros tipos de datos de forma nativa, como Series de Tiempo, Contadores y Adjuntos binarios.

Historia de RavenDB

¿Cuándo se Desarrolló?

RavenDB comenzó a desarrollarse alrededor de 2008, y su primera versión estable (1.0) fue lanzada en 2010. Al principio, el proyecto se llamaba "Rhino DivanDB".

Contribuyentes Clave:

El principal creador y desarrollador fue Oren Eini, más conocido en la comunidad tecnológica por su alias Ayende Rahien. La base de datos es desarrollada por su propia empresa, Hibernating Rhinos Ltd.

Objetivos Iniciales:

- Crear una base de datos de documentos NoSQL diseñada desde cero para .NET.
- Ofrecer transacciones ACID (un tipo de garantía de integridad de datos) en un modelo NoSQL.
- Hacerla fácil de usar para los desarrolladores.

Principales Características de RavenDB

Características Únicas que lo distinguen de otros SMBD:

- **Transacciones ACID por defecto:** RavenDB garantiza las propiedades en sus operaciones. Esto le da la confiabilidad y la integridad de datos de una base de datos relacional, pero con la flexibilidad de un modelo documental.
- **Indexado automático y dinámico:** RavenDB crea y actualiza los índices automáticamente en segundo plano a medida que haces consultas.
- **Diseñado para .NET:** Aunque tiene clientes para otros lenguajes, RavenDB fue construido desde cero para el ecosistema .NET.

Principales Características de RavenDB

Métricas de Rendimiento o Benchmarks que destaquen su Eficiencia:

- **Rendimiento en Escrituras:** Según RavenDB, un solo servidor con hardware estándar puede manejar más de 150,000 escrituras por segundo. Esto es gracias a la técnica "transaction merger", que agrupa múltiples transacciones en una sola operación de escritura en el disco.
- **Rendimiento en Lecturas:** En las mismas condiciones, puede alcanzar más de 1M de lecturas por segundo. La velocidad se debe en gran parte a la forma en que almacena los documentos en un formato binario optimizado para el acceso directo desde la memoria, sin necesidad de una deserialización costosa.
- **Comparaciones con otros SMBD:** En pruebas de rendimiento, RavenDB a menudo se destaca frente a otras bases de datos como MongoDB o bases de datos relacionales en escenarios complejos, lo que demuestra su eficiencia y su capacidad para trabajar con hardware de bajo costo.

Almacenamiento, recuperacion y actualizacion de datos

Mecanismos/tecnologías:
Se utiliza un motor de almacenamiento (como Voron en RavenDB) que maneja cómo se graban los documentos JSON en disco.

Procesos involucrados:
Utiliza estructuras como árboles B+ o almacenamiento basado en páginas para acceso rápido.

Impacto en usuario y rendimiento:

- Alta eficiencia en operaciones CRUD.
- El usuario no necesita preocuparse por la estructura interna.
- Mejora la escalabilidad y tiempos de respuesta.

Soporte de Transacciones

Mecanismos/tecnologías:
RavenDB usa un enfoque transaccional ACID, incluso en ambientes distribuidos.
Utiliza el algoritmo de consenso Raft (implementado como Rachis) para asegurar integridad entre nodos.

Procesos involucrados:
Cada operación transaccional debe ser validada por la mayoría de los nodos antes de ser confirmada.

Impacto en usuario y rendimiento:

- Brinda confiabilidad incluso en sistemas distribuidos.
- Puede introducir ligera latencia en sistemas replicados, pero se compensa con la robustez.

Servicios de recuperación

Mecanismos/tecnologías:
Journaling o WAL (Write-Ahead Logging): cada operación se guarda primero en un log.
Copias de respaldo automáticas y puntos de restauración.
Procesos involucrados:
Si hay una falla, RavenDB puede reconstruir el estado reciente leyendo el log y aplicando cambios pendientes.
Impacto en usuario y rendimiento:

- Mayor seguridad ante fallos.
- Ligeramente más uso de disco por los logs, pero transparente para el usuario.

Servicios de control de concurrencia

Mecanismos/tecnologías:
MVCC (Multi-Version Concurrency Control): permite múltiples versiones de documentos.
Bloqueos optimistas (Optimistic Concurrency Control).
Procesos involucrados:
Cada operación compara versiones del documento antes de guardar.
Si hay conflictos, se pueden resolver automáticamente o manualmente.
Impacto en usuario y rendimiento:

- Permite que varios usuarios trabajen al mismo tiempo sin interferencia.
- Reduce bloqueos, mejora la experiencia en apps colaborativas.

RavenDB

Servicios de autorización

Mecanismos/tecnologías:
Control basado en roles (RBAC).
Integración con sistemas de autenticación como OAuth o Active Directory.

Procesos involucrados:
Verificación de credenciales y asignación de permisos por operación (lectura, escritura, eliminación, etc.).

Impacto en usuario y rendimiento:

- Mayor seguridad.
- Posibilidad de administrar diferentes niveles de acceso (por usuario, por base, por documento).

Manejo de DDL/DML

-**¿Qué son DDL y DML?**

DDL (Data Definition Language)

Define la estructura de la base de datos: colecciones, índices, reglas, políticas.

DML (Data Manipulation Language)

Manipula los datos: inserta, consulta, actualiza o elimina documentos.

¿Por qué son importantes DDL y DML?

DDL: Define cómo se organiza y optimiza la base de datos.

DML: Gestiona los datos que dan vida a las aplicaciones.

Juntos permiten administrar estructura y contenido de forma eficiente.

Ventajas de RavenDB

- NoSQL moderno y orientado a documentos.
- Alta disponibilidad y transacciones ACID.
- Consultas poderosas (RQL, LINQ).
- Soporta documentos, archivos, series temporales y grafos.

DDL en RavenDB

- No requiere esquemas fijos, pero permite definir:
  - Índices personalizados
  - Configuración de colecciones
  - Políticas de compresión, expiración, etc.

```
index Users_ByCity {
  map 'from user in docs.Users select new { user.City }'
}
```

- Operaciones CRUD sobre documentos JSON.

Insertar:

csharp Copy code

```csharp
session.Store(new User { Name = "Carlos" });
session.SaveChanges();
```

Consultar:

rql Copy code

```rql
from Users where City = 'Hermosillo'
```

Actualizar:

csharp Copy code

```csharp
var user = session.Load<User>("users/1-A");
user.City = "Obregón";
session.SaveChanges();
```

Eliminar:

csharp Copy code

```csharp
session.Delete("users/1-A");
session.SaveChanges();
```

DML en RavenDB

Permite insertar, consultar, actualizar y eliminar documentos JSON dentro de colecciones, usando RQL o la API de cliente (C#, Java, JS).

Es el conjunto de operaciones que le dan vida a la base de datos, gestionando la información de forma flexible y eficiente.

Compatibilidad del sistema operativo

RavenDB es compatible con:

- Windows x64
- Windows x86
- Linux x64
- Linux arm64
- Mac OS (Intel x64)
- Mac OS (M1)
- Ubuntu x64
- Ubuntu arm64
- Ubuntu arm32

Requerimientos para instalar RavenDB

Generalmente requiere de mínimo:

- 512MB de RAM
- 1GHz CPU (x64, x86, ARM)
- 1GB de espacio en disco
- RavenDB puede operar en cualquier sistema operativo con .NET

Para una sola máquina (clúster de un solo nodo):

- 8GB de RAM
- 3GHz 4 núcleos CPU (x64, x86, ARM)
- 1GB de espacio en disco
- Windows Server 2022/ Ubuntu 22.04 LTS

Windows

Es la plataforma principal y más probada para .NET. No hay limitaciones significativas. Para entornos de producción, se recomienda encarecidamente el uso de una edición Server.

Linux

RavenDB es completamente compatible con Linux y se ejecuta de forma nativa. Es una plataforma de primera clase y muy popular para despliegues en producción, especialmente en entornos cloud y Docker. La principal consideración es asegurar que la distribución tenga soporte a largo plazo (LTS) para estabilidad.

macOS

La compatibilidad con macOS está dirigida principalmente al desarrollo y pruebas. No se recomienda para entornos de producción. Los desarrolladores pueden instalar y ejecutar RavenDB fácilmente en sus máquinas macOS para codificar y depurar aplicaciones.

Soporte de lenguajes de programación

- **C# (.NET) - Lenguaje Nativo y Principal**

La interacción con RavenDB se realiza mediante el cliente oficial RavenDB.Client, que ofrece una DocumentSession para operaciones CRUD sobre documentos JSON de forma transparente.

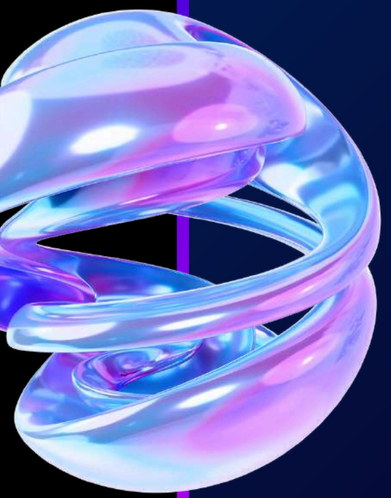
Bibliotecas:
El paquete NuGet **RavenDB.Client** es el núcleo. Para ASP.NET Core, la integración es sencilla y se suele registrar el cliente RavenDB en el contenedor de Inyección de Dependencias (DI).

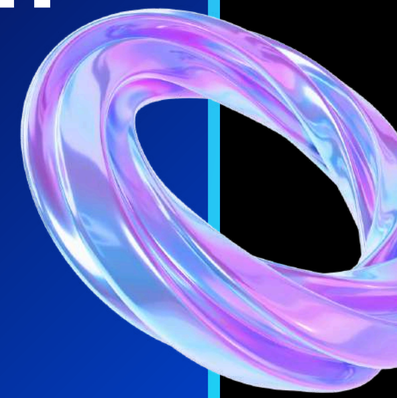
Soporte de lenguajes de programación

- **Java**

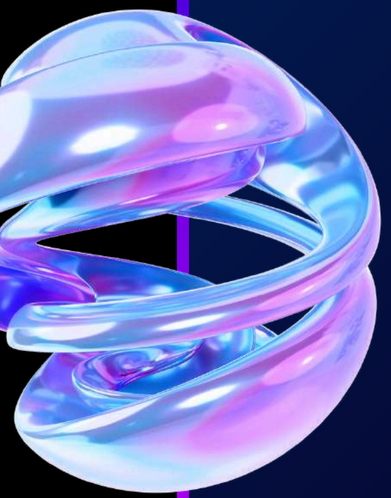
RavenDB ofrece un cliente oficial para Java. Sigue patrones similares al cliente .NET, permitiendo trabajar con objetos POJO (Plain Old Java Objects) y proporcionando una API de sesión para operaciones y consultas.

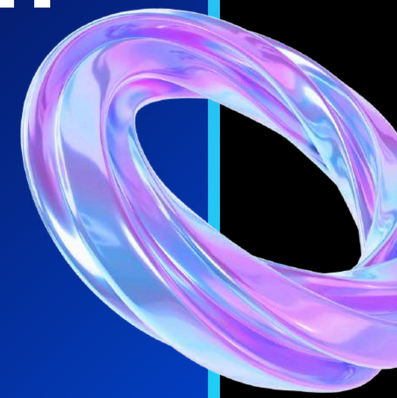
Bibliotecas:
El cliente oficial ravendb-jvm-client disponible en Maven Central.





Soporte de lenguajes de programación





Python:

Se puede interactuar con RavenDB a través de su API RESTful HTTP. Cualquier cliente HTTP estándar de Python (como requests) puede ser usado para realizar operaciones GET, POST, PUT, DELETE en los endpoints de documentos, consultas (Indexes), etc.

Bibliotecas

No existe un cliente oficial con la misma abstracción de alto nivel que .NET o Java, pero la biblioteca **pyravendb** (no oficial) puede facilitar algunas operaciones. La mayoría de las interacciones se hacen directamente con la API HTTP.

Aplicaciones del Mundo Real:

Empresas y Casos de Uso:

- **Empresas grandes y conocidas:**
  - The Home Depot: Usa RavenDB para manejar grandes volúmenes de datos.
  - Bosch: Utiliza RavenDB en sus sistemas.
  - NBCUniversal: Lo usa para gestionar su contenido.
  - Rakuten: Lo utiliza para el procesamiento de transacciones.

- **Empresas de software y servicios:**
  - DeskDirector: Usa RavenDB para su plataforma de gestión de tickets y proyectos. Se benefician de la capacidad de la base de datos para manejar arquitecturas multi-tenant, creando una base de datos separada y aislada para cada cliente.
  - Strife: Es un Sistema de Gestión de Contenido sin Interfaz Visual que considera la flexibilidad del modelo de documentos es clave, ya que les permite a sus clientes modelar contenido de cualquier forma sin un esquema rígido.

Aplicaciones del Mundo Real:

¿Cómo se benefician del uso de RavenDB?

- **Agilidad y desarrollo rápido:** La flexibilidad del modelo de documentos permite a los equipos de desarrollo iterar rápidamente y adaptar sus aplicaciones a nuevas necesidades sin tener que reestructurar la base de datos constantemente. Por ejemplo, Strife puede ofrecer a sus clientes un modelado de contenido ultra-flexible.

- **Escalabilidad y rendimiento:** La alta capacidad de escritura y lectura de RavenDB, junto con sus opciones de replicación y clústeres, permite a empresas como Home Depot manejar picos de tráfico y grandes cantidades de datos sin comprometer la velocidad.

- **Confiabilidad:** Las transacciones ACID son cruciales para sistemas que no pueden permitirse perder datos, como plataformas financieras (Rakuten) o sistemas de gestión (DeskDirector). Esto les da la seguridad de que sus operaciones de negocio son seguras y consistentes.

Referencias:

RavenDB Documentation. (s. f.). Recuperado el 31/08/2025, de https://docs.ravendb.net/7.1/

About us - RavenDB NoSQL Database. (2025, 7 agosto). RavenDB NoSQL Database. Recuperado el 31/08/2025, de https://ravendb.net/about

Indych, K. (2024, 26 junio). RavenDB vs MongoDB. RavenDB NoSQL Database. Recuperado el 31/08/2025, de https://ravendb.net/articles/ravendb-vs-mongodb

ACID Database Transactions | NoSQL Document Database. (2025, 30 abril). RavenDB NoSQL Database. Recuperado el 31/08/2025, de https://ravendb.net/why-ravendb/acid-transactions

RavenDB Performance Overview. (2025, 22 mayo). RavenDB NoSQL Database. Recuperado el 31/08/2025, de https://ravendb.net/performance

Benchmarking RavenDB on the Raspberry PI - RavenDB NoSQL Database. (2021, 11 junio). RavenDB NoSQL Database. Recuperado el 31/08/2025, de https://ravendb.net/articles/benchmarking-ravendb-on-the-raspberry-pi

Use Case: RavenDB at DeskDirector - RavenDB NoSQL Database. (2024, 31 julio). RavenDB NoSQL Database. Recuperado el 31/08/2025, de https://ravendb.net/articles/use-case-ravendb-at-deskdirector

Muchas Gracias!!