

Instituto Tecnológico de Costa Rica

Bases de Datos 2



Profesor: Erick Hernández B.

Investigación

Álvaro Moreira Villalobos

2018319327

Ronaldo Vindas Barboza

2018238697

S1 - 2022

Tabla de Contenidos

Tabla de Figuras	2
1. Introducción:	3
1. Objetivo de la Investigación	4
Objetivo General:	4
Objetivos Específicos:	4
2. Antecedentes	5
3. Descripción Detallada	6
Escalabilidad:	6
Disponibilidad:	6
Rendimiento:	6
4. Funcionamiento y Arquitectura	7
JSON:	7
BlitTable:	7
Nodos:	8
Lenguaje RQL (Raven Query Language) e Índices:	8
5. Construcción	8
6. Consistencia	9
7. Distribución	10
8. Usos posibles	11
9. Experiencias de Desarrollo Actuales	12
10. Conclusiones	13
11. Prototipo	14
12. Conclusiones del Prototipo	15
13. Referencias Bibliográficas	16

Tabla de Figuras

Ilustración 1 Modelo entidad relación del prototipo	14
---	----

1. Introducción:

En la actualidad, el estudio de las diferentes bases de datos existentes es un aspecto fundamental para todo desarrollador, dado que de esta manera puede llegar a modelar y crear una solución en concreto en base a los distintos requerimientos con los que se cuente. El correcto dominio y conocimiento de las diferentes capacidades de estas bases de datos es lo que determina para qué deberían ser utilizadas.

A lo largo de este documento se detallarán aspectos importantes de la historia, estructuras, características, composición y ejemplos de uso de Raven DB, así como también el uso de esta para la creación de un prototipo que destaque sus mayores cualidades para su utilización en la industria de esta base de datos. Raven DB es una base de datos no relacional, open source del tipo documental, tipo que le permite acoplarse a una gran variedad de usos en el campo de desarrollo.

1. Objetivo de la Investigación

Objetivo General:

Dar a conocer las distintas funcionalidades así como el rendimiento mediante el estudio de la base de datos no relacional Raven DB.

Objetivos Específicos:

- Detallar la historia de la creación de la base de datos RavenDB.
- Detallar el uso de las distintas herramientas que posee RavenDB.
- Analizar y comprender la estructura, así como arquitectura de la base de datos.
- Destacar características únicas de esta base de datos.
- Examinar de cerca las diferentes ventajas y desventajas del uso de esta.
- Hacer uso de lo aprendido para la creación de un prototipo que demuestre el funcionamiento y fortalezas de la base de datos.

2. Antecedentes

Para llegar a entender lo que es RavenDB hoy en día, debemos recordar primeramente cómo fue que todo este proceso comenzó. Tras el gran auge del internet a partir de los años 2000 y la gran cantidad de usuarios (alrededor de mil quinientos millones de usuarios para el año 2008 según la Unión Internacional de Telecomunicaciones) aunado a la gran cantidad de datos que se comenzaron a manejar con el establecimiento de redes sociales y otros sitios web basados en contenidos de usuario es que comenzaron a surgir problemas con datos no estructurados. Precisamente, en palabras del antiguo presidente de Google, Erich Schimdt (2010): *“La cantidad y volumen de datos generados sólo en internet cada dos días es equivalente a todos los datos generados por la humanidad hasta el año 2003 (aproximadamente cinco exabytes)”*.

Harto de esta situación y de los problemas generados al momento de trabajar en proyectos con múltiples bases relacionales es que Oren Eini en junio del 2008 decide crear una manera de procesar de mejor manera toda esta información para finalmente potenciar una aplicación del siglo XXI, una base de datos documental de nueva generación.

En mayo de 2010, RavenDB se convierte en la primera base de datos documental en apearse al principio de propiedades transaccionales de ACID (atomicidad, consistencia, aislamiento y durabilidad por sus siglas en inglés) sobre múltiples documentos y a lo largo de toda la base de datos. Consiguió ser la primera base de datos documental en estar disponible en la plataforma .NET. En años posteriores, se lanzaría una nueva versión de RavenDB que llevaría la replicación de la base de datos a un nivel completamente nuevo al lograr incorporar varias bases de datos nodo que a su vez le permitiría convertirse en una base de datos tolerante a fallos.

Asimismo, con el paso de los años se fueron agregando más funcionalidades a la base de datos ofreciendo la incorporación de características como ETL (Extracción, Transformación y Carga por sus siglas en inglés) permitiendo compatibilidad con otras bases de datos y la facilitación de integración de esta base de datos a diferentes proyectos y aplicaciones.

3. Descripción Detallada

La base de Datos RavenDB se encuentra catalogada como una base de datos NoSQL documental, de entre sus características se encuentran:

- Es de código libre (Open Source), ofreciendo alta disponibilidad y excelente rendimiento.
- Cumple con las garantías transaccionales ACID a lo largo de toda la base de datos incluyendo clusters.
- Sintaxis RQL basada en un 80% en SQL
- Integración con bases de datos SQL.
- Libre de uso de Esquemas(Schemas).
- Integración con Machine Learning.
- Es multiplataforma: C#, Node.js, Java, Python, Ruby, Go, entre otros.
- Es multisistema, siendo compatible con Windows, Linux, Mac OS, Docker, Raspberry Pi.
- Compatible con Cloud Computing.

Escalabilidad:

La escalabilidad horizontal de una base de datos NoSQL implica la división o partición del sistema en varios componentes de estructuras hospedadas en distintas máquinas físicas para así incrementar la cantidad de servicios que realizan la misma función en paralelo. Estos nodos de máquinas son sencillas de instalar y son idénticas, entre otras características se encuentran:

- Posibilidad de especificar los nodos exactos para ensamblar la base de datos distribuida.
- Cada nodo contiene una copia completa de la base de datos, tiene todos sus documentos y los indexa de manera local lo que simplifica la ejecución de consultas puesto que no necesita orquestar una agregación de datos de varios nodos.
- Cuando se añade un nuevo nodo a una base de datos ya existente se le asigna un nodo mentor que permite actualizar a este nuevo nodo al punto de quedar igual a los demás.
- Todos los nodos miembros funcionan mediante Replicación master-master con el fin de mantener los documentos en sincronía alrededor de todos los nodos.
- El método de distribución por Fragmentación (Sharding) se encuentra en desarrollo para una versión futura de la base de datos.

Disponibilidad:

- Debido a la consistencia de los datos, incluso si la gran mayoría del clúster se encuentra inactivo, siempre y cuando al menos un nodo se encuentre disponible, aún se podrán realizar operaciones de escritura y lectura.
- Las solicitudes de lectura pueden propagarse alrededor de los nodos del clúster para mayor rendimiento.

Rendimiento:

RavenDB es capaz de realizar un millón de lecturas por segundo y ciento cincuenta mil escrituras por segundo por cada nodo.

4. Funcionamiento y Arquitectura

La base de datos de RavenDB hace uso del almacenamiento de datos en formato JSON por su simplicidad de uso. Este último es un formato basado en texto que sigue la sintaxis de Javascript. Adicionalmente, en RavenDB no tiene la necesidad de definir un esquema para almacenar documentos, ya que almacena los datos en un formato binario conocido como BlitTable, el cual es un tipo de dato que se encuentra de manera predeterminada en la memoria. Seguidamente la base de datos puede ser implementadas en un clúster distribuido de servidores (nodos) donde para la realización de una operación requieren el acuerdo de la mayoría de estos nodos. Estos y otros aspectos importantes de la arquitectura de RavenDB se especifican a mayor detalle a continuación:

JSON:

De acuerdo a The JSON Data Interchange Standard, un JSON es un formato ligero para el intercambio de datos que permite de facilite la lectura y escritura para una persona, mientras que a su vez resulta sencillo de analizar y generar para una máquina. Estos documentos se encuentran contruidos por dos estructuras: La primera de ellas es una una colección de pares llave-valor (similar a como ocurre con diccionarios, objetos, struct en otros lenguajes) y una lista de valores (que equivalen a una matriz, vector, una lista o un arreglo en otros lenguajes).

BlitTable:

Es un tipo de datos que no requiere la atención de un contador de interoperabilidad (interop mashaler), esto es debido a que, por defecto, tiene una representación común en memoria. Además estos tipos de datos están fijados de manera tal que previene que el recolector de basura los remueva. Esto significa que el código cambiará consistentemente las ubicaciones de memoria de estos tipos de datos que consecuentemente logra que el serializador requiera de un esfuerzo mucho menor para mantener la integridad de los datos.

Algunos ejemplos de tipos de datos blittable en el framework .NET incluyen:

- System.Byte
- System.SByte
- System.Int16
- System.UInt16
- System.Int32
- System.Single
- System.Double

Nodos:

De acuerdo a la documentación de RavenDB, esta base de datos hace uso de la arquitectura distribuida shared-nothing en donde cada nodo que lo integra se comporta de forma independiente y autosuficiente y no tiene un único punto de contención a lo largo de todo el sistema. Todo esto debido al conocimiento de nodos distintos y la replicación entre nodos a través de la red. Las operaciones a nivel de clúster requieren del consenso de la mayoría de nodos, este consenso se logra mediante un algoritmo de tipo Raft llamado Rachis y distribuye las tareas a los diferentes nodos de forma equilibrada.

Lenguaje RQL (Raven Query Language) e Índices:

Es un lenguaje propietario de RavenDB para consulta, este cuenta con una sintaxis muy similar al SQL para la fácil comprensión y desarrollo por parte de los programadores. RavenDB tiene dos tipos de consultas:

- Consulta dinámica:
Le otorga al optimizador de consultas total libertad para escoger qué índice usará la consulta. Por ejemplo, en instrucciones como: “from Orders where ...”.
- Consulta indexada: Especifica qué índice usará para la consulta. Por ejemplo, en instrucciones como: “from index 'Orders/ByCompany' where ...”.

En casos en donde no se encuentre un índice para una consulta, el optimizador de consultas se encargará de crear un índice nuevo para dicha consulta.

5. Construcción

La base de datos RavenDB fue escrito en el lenguaje C# y desarrollado por el grupo Hibernating Rhinos Ltd. Como se ha mencionado anteriormente, es además una base de datos multiplataforma, siendo compatible con Microsoft, Linux, Mac Os.

6. Consistencia

Las bases de datos NoSQL están optimizadas para priorizar la disponibilidad sobre la consistencia. RavenDB no se diferencia de otras bases de datos NoSQL en este apartado, sin embargo, no deja completamente de lado la garantía de consistencia. Acciones como la recuperación de un solo documento, revisar un pedido o ver el perfil de un usuario son compatibles con las garantías transaccionales ACID.

algunas de las maneras que ravenDB maneja los datos serían:

- Se producen escrituras en el almacén de documentos.
- Las operaciones individuales van directamente al almacén de documentos.
- Las consultas se producen contra el almacén de índice
- A medida que se escriben los documentos, los datos se copian del almacén de documentos al almacén de índices, para aquellos índices que ya están definidos.
- En cualquier momento en que se consulte un índice, se obtendrá lo que ya se encuentre en ese índice, independientemente del estado de la copia que se esté realizando en segundo plano. Esta es la razón por la que en ocasiones se nos dice que los índices están "obsoletos".
- Si realiza una consulta sin especificar un índice y Raven necesita un nuevo índice para responder a su consulta, comenzará a crear un índice sobre la marcha y le devolverá algunos de esos resultados de inmediato. Solo bloquea el tiempo suficiente para darle una página de resultados. Luego continúa creando el índice en segundo plano, de modo que la próxima vez que realice una consulta tendrá más datos disponibles.

7. Distribución

El motivo por el cual se hace uso de sistemas distribuidos es para manejar mayor cantidad de datos. Los clústers en RavenDB no se limitan a ser simplemente una colección o conjunto de máquinas, sino que además son entidades auto-gestionables, comparten cargas y tareas a través de todos sus nodos.

El clúster distribuye las tareas entre los nodos de forma automática, permitiéndole manejar los errores, recuperación y asegurándose que todo funcione correctamente. Las operaciones que se realizan en RavenDB usualmente se encuentran divididas en operaciones disponibles para todo el clúster y en operaciones internas. Así, el hecho de crear una base de datos es una operación que se realiza para todo el clúster, mientras el escribir un documento a la base de datos solo afecta esa base de datos en particular. Esto ocurre ya que RavenDb opera dos capas distintas en sus sistemas distribuidos. Esto quiere decir que RavenDb utiliza un sistema de distribución dual que se añade a la robustez de la base de datos en caso de que ocurra algún fallo.

Luego, siguiendo el teorema de Brewer (o teorema CAP) la documentación de RavenDB revela que la base de datos ha optado por acoplarse a las garantías AP (Disponibilidad y Tolerancia a Particiones) y CP (Consistencia y Tolerancia a Particiones). Esto lo logra al no aplicar estas garantías a una única capa. Con RavenDB la capa de clúster sigue las garantías CP (Siempre consistente, pero podría no estar disponible en presencia de una partición), pero la capa de base de datos es AP (Siempre disponible, incluso si existe una partición, pero eventualmente consistente).

8. Usos posibles

De forma general, entre los posibles usos que se le pueden dar a esta base de datos se encuentran: sistemas relacionados a registro de eventos, análisis en tiempo real y perfiles de usuario en plataformas web. Adicionalmente encontramos su uso en sistemas como:

- Catálogos de productos.
- Administración de mercancía.
- Sistemas de gestión de contenido.
- Motores de recomendación.
- Detección de fraudes.
- Análisis de datos en tiempo real.

No se recomienda usar tanto esta como otras bases de datos no relacionales del tipo documentales cuando existen transacciones muy complejas que abarcan distintas operaciones.

9. Experiencias de Desarrollo Actuales

Muchos desarrolladores opinan que la base de datos RavenDB es una base de datos fácil de instalar y utilizar. aunque no se tenga experiencia previa en la utilización de bases de datos NoSQL y que va mejorando actualización tras actualización. con gran integración en código .NET, además de su disponibilidad para diferentes sistemas OS como: Windows, Linux, MacOS, Docker y Raspberry Pi. y compatibilidad con: C#, Node.js, Java, Python, Ruby y Go.

En algunos casos se menciona que al tratar de utilizar grandes cantidades de información puede llegar a tardar más de lo que se espera y para algunos la interfaz que brindan podría ser más clara para mejorar su entendimiento. un detrimento que muchos desarrolladores tan bien le asocian es la falta de una comunidad más grande a la cual acudir en caso de alguna traba. En contraste a esto se menciona que el servicio técnico brinda ayuda de gran manera. Con respecto a la documentación mencionan que en algunos aspectos está bien pero que en algunos casos específicos puede llegar a faltar partes importantes de la misma.

Jeremy Holt director ejecutivo de amberwood trading, decidio utilizar ravenDB como su plataforma de base de datos, luego de que aprendiera de manera simple, al utilizarla en la herramientas que brinda la página oficial de ravenDB, lo utilizo para crear un sistema de control para cada etapa de transición para brindar transparencia en todos los documentos de venta.

Win-tech es uno de los usuarios que también utiliza ravenDB aprovechando la opcion que brinda de administración en la nube y el modelo de base de datos documental para reducir la complejidad y maximizar los resultados.

10. Conclusiones

A lo largo de este documento hemos detallado cómo fue que RavenDB poco a poco se ha abierto paso en el mundo de las bases de datos no relaciones hasta la actualidad aprovechando ventajas como: código libre, garantías transaccionales ACID, integración con bases de datos SQL, libre de uso de esquemas, Cloud Computing, integración con machine learning, y su gran gama de apertura multiplataforma y multisistema. Creando una base de datos no relacional del tipo documental la cual utiliza archivos Json para almacenar la información en colecciones. Contando con integración nativa para generar clústers autogestionables que comparten cargas y tareas a través de los nodos.

RavenDB es una base de datos cada vez más popular que, si bien aún no es perfecta para todo tipo de trabajos, poco a poco se está labrando una comunidad y con grandes planes a futuro para mejorar el momento de este escrito se está planeando lanzar una versión 6.0 de RavenDB la cual incorporará aspectos como: Data Sharing, Corax y V8 engine para JavaScript lo cual hace que resulte interesante el seguir prestando atención a la esta base de datos.

11. Prototipo

Para esta investigación se realizó una pequeña aplicación que consiste en el registro de inventario de vehículos para agencias automotrices. Este caso de uso pretende demostrar algunas de las características más importantes de esta base de datos, así como las ventajas que esta aporta.

Para el desarrollo de esta aplicación, además del uso de RavenDB, se hizo uso del lenguaje de programación Java, el framework de Java Swing para la interfaz de usuario. RavenDB permitió sacar el máximo provecho esta aplicación al proveer de flexibilidad al momento de añadir cualquier clase de información a los documentos con suma facilidad.

En resumen, esta aplicación cuenta con:

- Registro de vehículos y agencias concesionarias.
- Tabla de consulta de vehículos y agencias.
- Posibilidad de edición y eliminación de vehículos o agencias.

Se adjunta un pequeño diagrama entidad-relación de la base de datos para comprender mejor el actual prototipo:

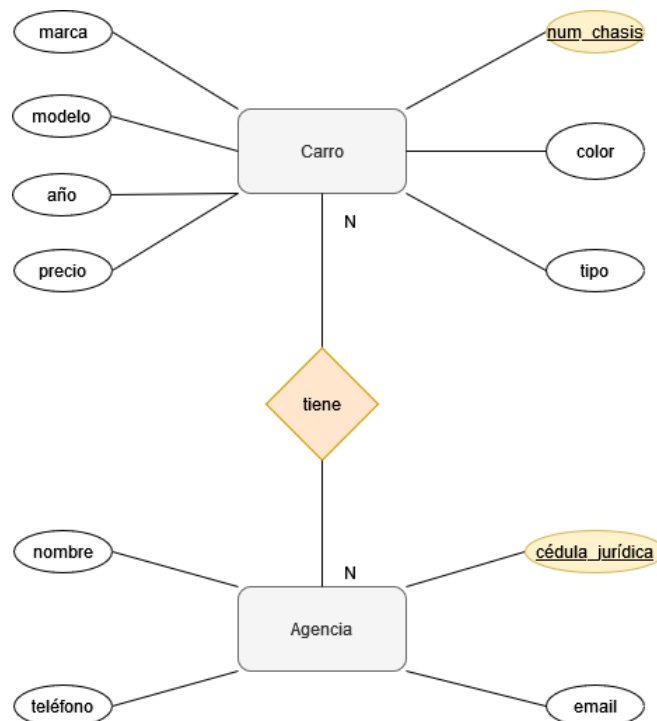


Ilustración 1 Modelo entidad relación del prototipo

12. Conclusiones del Prototipo

Con este prototipo se puede demostrar la funcionalidad y utilización de la base de datos Raven DB a través de Java, su conexión a una aplicación, almacenamiento y manejo de información de la misma.

RavenDB maneja una interfaz para el usuario llamada RavenDB Studio que brinda toda la información necesaria sobre los datos almacenados, monitoreo de rendimiento y telemetría. Asimismo, su compatibilidad con variedad de lenguajes de programación permitió que la conexión con la aplicación escrita en Java se llevara a cabo de manera sencilla.

Para la aplicación de operaciones y métodos como inserción, borrado, actualización y consulta de datos, se nos presentan varias opciones para realizarlas como: funciones dedicadas para la realización de estas operaciones o bien, la posibilidad de escribir una consulta de manera manual para que sea interpretado por RavenDB desde Java. De la misma manera, la búsqueda de documentación para analizar en detalle las funciones y ventajas que esta base de datos posee fueron de suma utilidad para la realización del actual prototipo.

El manejo de datos no es muy diferente a otras bases de datos documentales por lo que en este apartado no destaca de manera especial, haciendo que RavenDB sea una excelente opción para aquellas personas principiantes en Bases de Datos que quieran aprender acerca de bases de datos documentales. Tanto la instalación de la base de datos como la instanciación de nodos para el clúster resultó ser muy sencilla y cómoda, precisamente la herramienta de RavenDB Studio disponible desde el propio navegador brinda ayuda paso por paso para lograr cada configuración.

Como punto negativo, se ha de destacar el hecho de que la realización de consultas por medio de identificadores(id) creados por la base de datos son complejas dado que la base de datos los maneja de forma oculta como metadatos. Para solucionar este problema fue necesaria la creación de un atributo o campo adicional que cumpliera con dicha labor.

13. Referencias Bibliográficas

Acerca de RavenDB. (s/f). Componentsource.com. Recuperado de <https://www.componentsource.com/es/product/ravendb/about>

Akinshin, A. (s/f). Blittable types. Aakinshin.net. Recuperado de <https://aakinshin.net/posts/blittable/>

BillWagner. (s/f). Un paseo por C#: información general. Microsoft.com. Recuperado de <https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>

Cluster: Overview. (2021, mayo 10). RavenDB NoSQL Database. <https://ravendb.net/docs/article-page/4.2/csharp/server/clustering/overview>

Distributed database. (2021, mayo 10). RavenDB NoSQL Database. <https://ravendb.net/docs/article-page/5.3/csharp/server/clustering/distribution/distributed-database>

Eini, O. (2016, mayo 4). The design of RavenDB 4.0: The implications of the blittable format. Dzone.Com; DZone. <https://dzone.com/articles/the-design-of-ravendb-40-the-implications-of-the-b>

Enhancing NoSQL database performance. (2022, abril 14). RavenDB NoSQL Database. <https://ravendb.net/articles/enhancing-database-performance-and-simplifying-devops>

Gartner, Inc. (s/f). Reviews. Gartner. Recuperado de <https://www.gartner.com/reviews/market/data-and-analytics-others/vendor/hibernating-rhinos/product/ravendb/reviews?marketSeoName=data-and-analytics-others&vendorSeoName=hibernating-rhinos&productSeoName=ravendb>

gewarren. (s/f). Blittable and non-blittable types. Microsoft.Com. Recuperado de <https://docs.microsoft.com/en-us/dotnet/framework/interop/blittable-and-non-blittable-types?redirectedfrom=MSDN>

How Amberwood trading uses RavenDB to transform the African cashew market. (2021, mayo 11). RavenDB NoSQL Database. <https://ravendb.net/learn/webinars/amberwood-ravendb-nosql-document-database-use-case>

JSON. (s/f). Json.org. Recuperado el 25 de abril de 2022, de <https://www.json.org/json-en.html>

kexugit. (s/f). NoSQL documento base de datos: la incorporación de RavenDB en una aplicación ASP.NET MVC 3 aplicación. Microsoft.com. Recuperado de <https://docs.microsoft.com/es-es/archive/msdn-magazine/2011/november/nosql-document-database-embedding-ravendb-into-an-asp-net-mvc-3-application>

.NET: Qué es y cómo funciona. (2020, junio 15). aula21 | Formación para la Industria. <https://www.cursosaula21.com/que-es-net/>

NoSQL database scalability. (2021, abril 27). RavenDB NoSQL Database. <https://ravendb.net/articles/how-a-database-delivers-performance-under-heavy-load-ravendb>

RavenDB system properties. (s/f). Db-Engines.Com. Recuperado el 23 de abril de 2022, de <https://db-engines.com/en/system/RavenDB>

RavenDB: The Right Data Storage Solution for Today's information. (2021, agosto 30).

Resource query language — application programming interface. (s/f). Cloudblue.com. Recuperado el 25 de abril de 2022, de <https://docs.cloudblue.com/cbc/sdk/8.0/api/rest/rql/>

Sadalage, P. (s/f). NoSQL distilled: a brief guide to the emerging world of polyglot persistence. Addison-Wesley.

Siegler, M. G. (2010, agosto 4). Eric Schmidt: Every 2 days we create as much information as we did up to 2003. TechCrunch. <https://techcrunch.com/2010/08/04/schmidt-data/>

Statistics. (s/f). ITU. Recuperado de <https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx>

Trabajando con JSON. (s/f). Mozilla.org. Recuperado de <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>

Welcome to RavenDB. (2021, marzo 19). RavenDB NoSQL Database.
<https://ravendb.net/learn/inside-ravendb-book/reader/4.0/1-welcome-to-ravendb>