

# Flexibilidad de datos estratégicos

Tanto los proveedores de bases de datos de la vieja escuela como los proveedores de servicios en la nube exigen modelos comprometidos de la capa de datos. Afortunadamente, Redis Enterprise ofrece una tercera vía que combina lo mejor de ambos mundos.

# Resumen ejecutivo

Elegir la base de datos correcta es una decisión fundamental, ya que puede darse una cadena de consecuencias en toda la organización en todos los niveles del entramado tecnológico durante años e incluso décadas. Como aspecto negativo, la elección de una base de datos puede resultar difícil y compleja, y las consecuencias no se ven plenamente hasta que se amplía la base de datos o se sobrepasa el modelo de datos inicial.

Lo irónico es que cuanto más limitada es la gama de opciones más complejo resulta elegir una base de datos. A grandes rasgos, el sector de las bases de datos se divide en dos frentes. Un frente defiende el modelo monolítico de bases de datos relacionales, mientras que el otro impulsa a los clientes a utilizar distintas bases de datos diseñadas en función de su finalidad para cada aplicación. Piense en el primer frente como un carrito de hot dogs: ofrece una cosa, que son hot dogs. El segundo frente es como un área de alimentación, en la que se selecciona de una gama de opciones que pueden ir juntas o no.

## Bases de datos relacionales = hot dogs

Los vendedores de hot dogs confían en un solo modelo de datos (relacional) y un solo tipo de software (DBMS). No importa de qué tipo sean realmente los datos que se almacenan y consultan: hay que adaptarlos y moldearlos de forma que se adapten al modelo relacional. El equipo de desarrollo tiene que escribir una capa de software adicional para adaptar distintos problemas de datos a problemas relacionales/tabulares. Desde el punto de vista operativo, esto suele provocar problemas de rendimiento o de escala, ya que obliga a la base de datos a hacer cosas que, en realidad, no estaba diseñada para gestionar. Da igual la cantidad de condimentos que les pongan encima: estos vendedores solamente ofrecen hot dogs.

“ Este carro vende perritos calientes. Aunque se les ponga ketchup, mostaza o incluso pepinillos, no dejan de ser hot dogs. Que nadie le diga lo contrario. ”



“ Redis Enterprise ofrece una tercera vía que combina lo mejor de ambos mundos. ”



## Bases de datos en la nube = áreas de alimentación

La perspectiva del frente del área de alimentación es que se necesitan muchos modelos de datos, incluso en una sola aplicación, lo cual implica que todo el mundo tiene que emplear varias bases de datos diseñadas en función de su finalidad. Estas bases de datos están unidas en una red, con lo que se obtiene un nido de ratas lento y frágil de dispositivos interconectados. Los desarrolladores tienen que aprender una gama desconcertante de lenguajes de consulta, métodos de conexión y bibliotecas de clientes. Y tienen que preocuparse acerca de los problemas de sincronización y las tolerancias a fallos cuando gestionan datos en varios servicios. Los equipos de operaciones tienen que lidiar con la configuración, la ampliación, la aplicación de medidas de seguridad y el mantenimiento no solo de una plataforma de bases de datos, sino de varias.

Ambas opciones están comprometidas. Implican conflictos y fragilidades, así como también problemas de rendimiento y capacidad de mantenimiento. Lo que es más importante aún: es una elección falsa y forzada. Hay otra vía, una que puede ayudarle a evitar el nido de ratas y tener que incorporar todo en un solo modelo de datos.

## Redis Enterprise = la tercera vía

**Redis Enterprise** presenta la tercera vía: una sola interfaz operativa en la que puede implementar varias bases de datos. Las bases de datos de Redis Enterprise ofrecen módulos que amplían la funcionalidad básica fundamental de Redis a modelos de datos más específicos como gráficos, búsquedas, documentos y series temporales, así como también capacidades de servicio de IA que se activan según los eventos. Los desarrolladores entran en contacto con la base de datos mediante las bibliotecas de clientes estándar de Redis. En vez de un carro de hot dogs o un área de alimentación, Redis Enterprise ofrece un menú completo de opciones bien elaboradas y pensadas en un solo restaurante.

# Introducción

El núcleo del software más moderno son los datos. Las bases de datos son una de las opciones tecnológicas fundamentales para los líderes de tecnología. Esta elección se basó en el desarrollo del almacenamiento en sí. Antes, el almacenamiento de datos era físico y constaba literalmente de orificios en tarjetas o cinta de papel. Con los años, pasamos a utilizar medios magnéticos y terminamos por utilizar dispositivos de almacenamiento de estado sólido. Hoy, la sofisticación está evolucionando, de las características físicas de la forma de almacenamiento de los bits en los medios a la forma en que se modelan los datos y se traducen en información que se puede utilizar.

No obstante, gran parte del software de bases de datos de la vieja escuela se diseñó en una era en la que los niveles de abstracción disponibles hoy básicamente no existían; ese software solamente se podía facilitar en medios físicos.

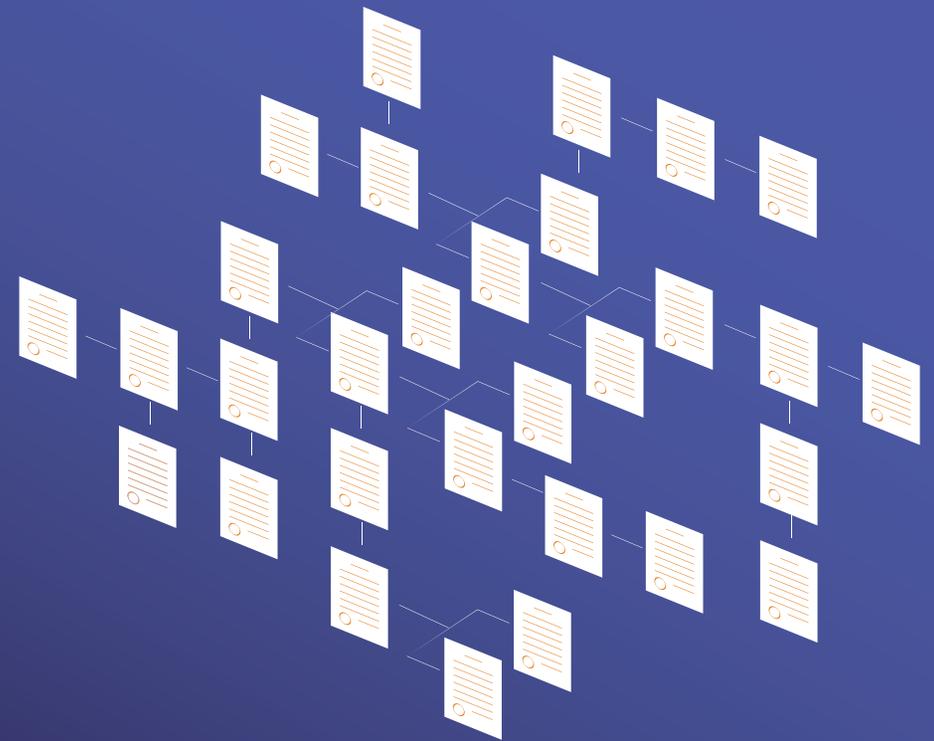
Todavía existe una base enorme de instalaciones de bases de datos de la vieja escuela que iguala a los sistemas no heredados. Estas bases de datos se basan en un solo modelo relacional y monolítico. Si bien ese modelo sigue siendo útil para modelar determinados problemas de datos, representa un motor de almacenamiento generalizado (ahora) que se ha utilizado con demasiada frecuencia, tanto para bien como para mal, en vez de las técnicas adecuadas de modelado de datos.

Mientras tanto, los proveedores de servicios en la nube están abstrayendo por completo la noción de las máquinas: ahora ofrecen bases de datos diseñadas en función de su finalidad disponibles exclusivamente en su servicio. Lo "único" que tiene que hacer es conectarlas con su aplicación. ¿Necesita una base de datos con un modelo diferente? Lo "único" que tiene que hacer es generar una base de datos nueva y conectarla con ella.

Lo fundamental es que las bases de datos, tanto en la nube como heredadas, dan lugar a riesgos fundamentales. En muchos casos, dan lugar a la adopción cíclica de estrategias tecnológicas diferentes en un intento recurrente de aliviar las molestias que provoca una serie de componentes que acaba por costar otros componentes.

Hablemos claro: los compromisos en su estrategia de bases de datos pueden provocar problemas internos (satisfacción y logro de objetivos del personal, longevidad de un proyecto) y problemas externos (rendimiento y estabilidad). Afortunadamente, la tercera opción ofrece una vía intermedia que evita las dificultades que provocan las bases de datos heredadas y los proveedores de servicios en la nube. Una opción estratégica lo bastante flexible como para adaptar modelos de datos nuevos de forma nativa y ofrecer sencillez y estabilidad operativas.

“ Gran parte del software de bases de datos de la vieja escuela se diseñó en una época en la que los niveles de abstracción disponibles hoy básicamente no existían. ”





# Posiciones del sector de las bases de datos

A la vista de las dos estrategias enfrentadas en el diseño de bases de datos, no sorprende que gran parte del sector de las bases de datos se encuadre en dos frentes: los proveedores de bases de datos de la vieja escuela y los proveedores de servicios en la nube. Cada frente ve de forma muy distinta el panorama y defiende soluciones muy diferentes.

## Proveedores de bases de datos de la vieja escuela

En los inicios de la informática, la gente empezó a pensar en formas de almacenar los datos en un formato estandarizado. Charles W. Bachman había creado el Integrated Data Store (Almacenamiento de datos integrados) en General Electric en 1963, el cual representa lo primero que hoy podríamos identificar como una base de datos. Unos años después, en 1970, Edgar Codd escribió "A Relational Model of Data for Large Shared Data Banks" (Modelo relacional de datos para grandes bancos de datos compartidos), que sirvió como anteproyecto del modelo de las bases de datos relacionales. Larry Ellison adoptó las ideas de este artículo como base de lo que sería la base de datos de Oracle, que se lanzó en 1979.

El modelo relacional resultó ser muy flexible (para aquella época) y terminó por convertirse en el patrón de facto en bases de datos. El modelo y las bases de datos relacionales se volvieron funcionalmente equivalentes en la mayor parte de los debates. De hecho, este modelo tan potente le permite conseguir muchas cosas con una sola interfaz. Independientemente de que se tuviera un proyecto sencillo para unos pocos usuarios o un proyecto de datos complejo para toda una organización, el modelo relacional era la respuesta. Por supuesto, se derrocharon incontables vatios de energía a medida que se lanzaban y mantenían bases de datos relacionales de forma masiva para no hacer nunca nada más complejo que consultar elementos mediante una clave primaria en una sola tabla. La intención no era malgastar recursos, pero durante varias décadas las bases de datos fueron la única opción viable que la gente conocía y entendía bien, por lo que estas se pusieron en servicio para todo.

También se introdujeron datos modelados de formas completamente distintas, lo cual provocó dolores de cabeza en el modelado. Por ejemplo, asignar datos de gráficos a una base de datos relacional genera problemas de indexación y problemas de rendimiento masivos. De igual forma, los datos de series temporales constituyen una carga de trabajo optimizada para la escritura con un fuerte componente de

“ Se derrocharon incontables vatios de energía a medida que se lanzaban y mantenían bases de datos relacionales de forma masiva para nunca hacer nada más complejo que consultar elementos mediante una clave primaria en una única tabla. ”

consolidación que muchas bases de datos relacionales no podían gestionar en tiempo real de ninguna manera.

El almacenamiento de documentos en una base de datos relacional trata de incorporar a la fuerza estos tipos de datos de forma libre y sin esquemas en un formato tabular muy tipificado y estructurado. Todos estos modelos de datos son completamente legítimos, pero la ubicuidad de las bases de datos relacionales tendían a introducir la complejidad y la adaptación de modelos en la aplicación. En muchos casos, el resultado era una base de datos relacional que contenía datos a los que se podía acceder solamente desde una aplicación concreta, y una base de datos sobrecargada que hacía lo que podía para funcionar según un patrón de uso para el que no se había diseñado.

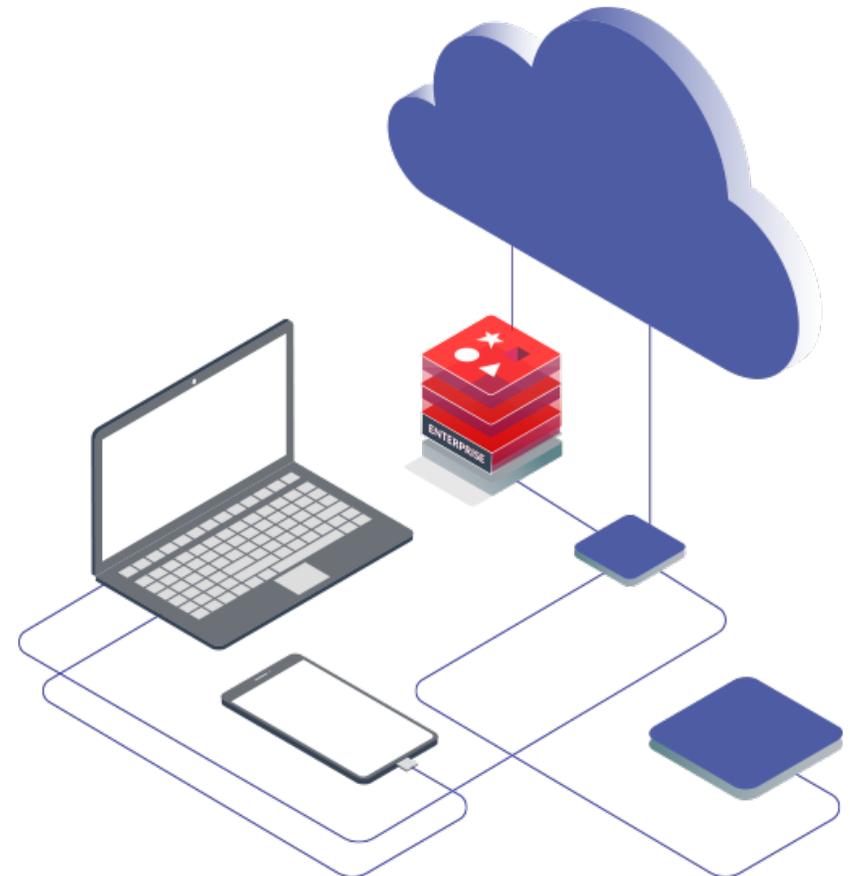
Una vez más, esto no quiere decir que el modelo relacional sea del todo inadecuado. Al igual que con cualquier otro modelo de datos, hay problemas para los que las bases de datos relacionales son la opción óptima. Las aplicaciones que tienen una gran cantidad de datos relacionales o que no se pueden desnormalizar de forma eficaz siguen adaptándose muy bien en estos tipos de bases de datos. Ahora bien, igual que cuando todo lo que uno tiene es un martillo, todo parece un clavo, si lo único de lo que dispone es de bases de datos relacionales, todos los problemas parecen ser relacionales.

Para ser justos, establecer una sola plataforma de bases de datos ofrece algunas ventajas operativas. Los profesionales que operan estas plataformas han aprendido con rapidez y profundidad la forma de mantenerlas estables. Dado que son programas informáticos grandes y complejos, los sistemas de bases de datos relacionales se pueden operar a un nivel organizativo en lugar de implementarlos específicamente para un determinado software, con lo que se centralizan la complejidad y los conocimientos expertos. Si bien la estrategia de unificación puede provocar problemas de desarrollo, también ha dado lugar a un solo vector conocido que poner en funcionamiento.

No obstante, desde el punto de vista de la arquitectura, las raíces de estos sistemas se remontan a una época anterior a que nacieran muchas de las personas que ahora los operan: antes del Internet moderno y mucho antes de la informática en la nube. Por supuesto, existe muy poco (si acaso existe) código fuente cuyo origen se remonte a finales de los años 70 del pasado siglo y que siga activo en este tipo de software, pero los conceptos subyacentes emergieron en una época diferente en los aspectos más básicos.

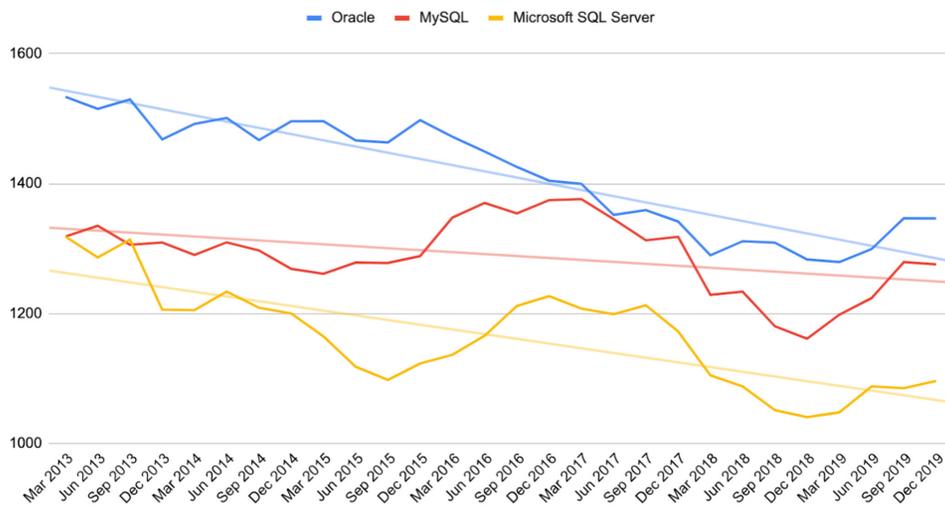
## Adéntrese en la nube

Cuando las empresas empezaron a trasladar su software a la nube, muchas intentaron hacerlo sin remodelar sus aplicaciones, tomándolas y llevándolas tal cual a ese entorno. Lo negativo es que las premisas adoptadas a nivel arquitectónico de muchos programas informáticos no encajan con el entorno en la nube, que es radicalmente diferente; y los SGBDR no son una excepción. Durante las décadas de optimización a nivel bajo y micro se produjo software que operaba con un rendimiento aceptable en entornos de las propias instalaciones. Ahora bien, cuando se trasladó a un entorno virtualizado, aparecieron problemas nuevos; un rendimiento menor cuando no había conexión con el almacenamiento acoplado a la red, los recursos informáticos efímeros y la ausencia de automatización hicieron que el trayecto hasta la nube estuviera repleto de obstáculos. En resumen, los ejercicios de tomar una aplicación y llevarla a ese entorno demostraron que los profesionales operativos no solo tienen que ser expertos en el software de bases de datos masivas, sino también en la nube que están empleando. El mundo cambió.



“ Los proveedores de bases de datos de la vieja escuela buscaban ofrecer una base de datos a una organización, mientras que es posible que los proveedores de servicios en la nube busquen ofrecer muchas bases de datos para una sola aplicación. ”

Puntuaciones en DB-Engines de los tres principales bases de datos relacionales, por trimestres entre 2013 y 2019



La época moderna de la informática en la nube ha afectado la popularidad de las bases de datos de la vieja escuela. Si bien las bases de datos relacionales ocupan las tres posiciones principales en las clasificaciones de DB-Engines.com, con el tiempo la tendencia es claramente descendente. Desde que DB-Engines empezó a hacer un seguimiento de las puntuaciones en 2013, ha habido un descenso general de popularidad de los principales modelos relacionales, según el método de puntuación de DB-Engines.

# Sombras de la nube: Las bases de datos relacionales siguen siendo populares... por ahora

## Proveedores de servicios en la nube

En el otro extremo del espectro se sitúan los proveedores de servicios en la nube. En función de sus líneas de productos, los proveedores de servicios en la nube parecen rechazar la idea de que el modelo relacional sea la forma estándar de almacenar y recuperar datos. Los distintos modelos de datos representados en los productos de bases de datos de los proveedores de servicios en la nube indican que los datos de su aplicación no se deberían moldear conforme a la base de datos disponible, sino que usted debería seleccionar la base o las bases de datos adecuadas para su aplicación.

Los proveedores de servicios en la nube se pueden ver como defensores de NoSQL, término que se acuñó nada menos que en 1998 pero que tiene un significado contemporáneo que data del 2009. Las bases de datos NoSQL se centraron, en su origen, en ofrecer un solo modelo de datos aplicable a una lista más breve de casos de uso. Se diseñan en función de su finalidad para un solo modelo de datos. No obstante, toda aplicación con una complejidad suficiente probablemente tenga una amplia gama de datos que se deban modelar de formas distintas, por lo que una sola aplicación puede precisar varias bases de datos diferentes. De hecho, en los casos en que los proveedores de bases de datos de la vieja escuela buscaban ofrecer una base de datos a una organización, es posible que los proveedores de servicios en la nube busquen ofrecer muchas bases de datos para una sola aplicación.

Varias bases de datos en un solo proyecto pueden ayudar a resolver los problemas de los desarrolladores. Las bases de datos relacionales generan resultados que exigen que los desarrolladores confíen en las bibliotecas o en su propio código para reunir o desunir datos de forma que se ajusten al modelo relacional (el problema de la falta de concordancia con la impedancia relacional de objetos). Las entradas y salidas de las bases de datos NoSQL tienden a parecerse a la forma en que el código de la aplicación modela los datos a nivel interno. Por ello, cuando un desarrollador trabaja con bases de datos diseñadas en función de un propósito, la proporción de tiempo

asignada a escribir códigos relevantes de resolución de problemas es mayor que cuando en las bases de datos relacionales existe el requisito oculto de darles a los datos una forma en que se puedan utilizar.

No obstante, el uso de varias bases de datos diseñadas en función de su finalidad en una sola aplicación tiene sus propios inconvenientes. Al igual que con cualquier otra herramienta, esto implica capacidades y técnicas de uso; con cada base de datos, los desarrolladores tienen que aprender no solo la base conceptual y el modelo de la base de datos, sino también las bibliotecas que la acompañan. Por ejemplo, entender los parámetros de conexión de las bases de datos no suele ser una tarea simple, por lo que, si no se gestiona adecuadamente, puede provocar problemas de rendimiento o de estabilidad.

Aun cuando formen parte de la nube, el uso de varias bases de datos suele presentar problemas operativos. Uno de esos problemas está asociado a la gestión de un número elevado de servicios en la nube en su organización. Dado que numerosas bases de datos atienden a cada aplicación, los equipos de operaciones tienen que manejar más programas informáticos distintos. Los servicios en la nube son más fáciles de mantener que los servicios en las instalaciones físicas, pero el mantenimiento operativo sigue siendo variable, nunca nulo. Por ejemplo, algunas bases de datos se actualizan sin que se note mientras que en otras hay que intervenir. Algunas bases de datos necesitan ajustes específicos del contexto, mientras que otras pueden funcionar perfectamente con los valores predeterminados.

## Latencia y ampliación

Además de la repercusión global en cuanto a la complejidad que implica disponer de varias bases de datos por aplicación, también la latencia se ve afectada. Asimismo, puede costar manejar la latencia global de un sistema complejo. Si una sola operación de una aplicación tiene que tocar varias bases de datos, la latencia acumulativa de todas las bases de datos y la aplicación afectará al rendimiento global de la aplicación. Mientras que una sola plataforma de bases de datos puede realizar muchas operaciones con una sola solicitud, en la nube, cada operación podría ser en una base de datos independiente, lo cual aumenta la latencia del trayecto a la aplicación y en sentido inverso. Es más: las operaciones síncronas pueden precisar varios saltos de red de ida y vuelta (latencia agravante), mientras que otras se pueden ejecutar en paralelo, de modo que todos los saltos de red ocurran a la vez (en este caso, la latencia global sería la mayor latencia del grupo).

La ampliación es otro aspecto a tener en cuenta. Cada una de las bases de datos que sustentan la aplicación se pueden ampliar de forma diferente, pero la aplicación solo rendirá tanto como su eslabón más débil. Incluso con una infraestructura optimizada para cada base de datos, la estrategia operativa es compleja, ya que el rendimiento alcanza mesetas en una base de datos u otra a medida que las amplía.

“ Con cada base de datos, los desarrolladores tienen que aprender no solo la base conceptual y el modelo de la base de datos, sino también las bibliotecas que la acompañan. ”



# Cuidado con las mantas cortas

Piense en intentar tender la cama con una manta que no es lo bastante larga como para cubrir el colchón. Si se pliega la manta a los pies de la cama, la cabeza queda destapada. Si se alinea al área de la cabeza, los pies quedan descubiertos. Si se deja la manta en el área central, tanto la cabeza como los pies quedan descubiertos. Todas las opciones son inadecuadas.

De igual forma, los proveedores de bases de datos de la vieja escuela ofrecen un despliegue y una ampliación organizacionales unificados, pero sus modelos inadecuados se tienen que compensar mediante complejidad adicional (consulte *el segmento inferior izquierdo*). Los proveedores de servicios en la nube ofrecen una gama de tipos de bases de datos diseñadas en función de la finalidad, lo que, por desgracia, fomenta la complejidad operativa a la hora de conectar, ampliar y operar varias bases de datos por aplicación (consulte *el segmento inferior derecho*).

En última instancia, las dos estrategias son inadecuadas. Trasladar la complejidad a los equipos de desarrollo u operativo no reduce la complejidad general. El gran esfuerzo de ingeniería necesario para trasladar una aplicación de un modelo de datos sencillo en una base de datos de la vieja escuela a muchas bases de datos diseñadas en función de una finalidad en la nube podría ser una pérdida de tiempo si no reduce realmente la complejidad sino que se limita a reasignar la complejidad a un equipo diferente.

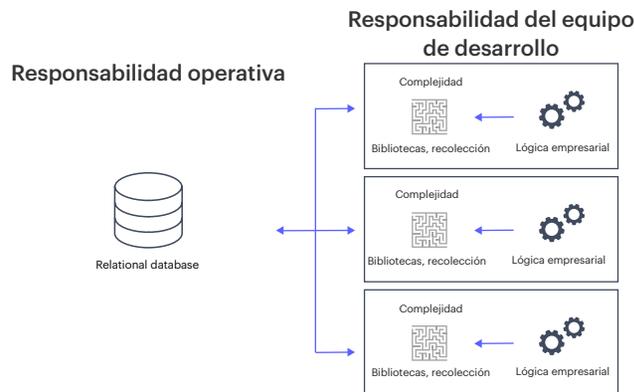
Aún peor: migrar una aplicación a la nube puede marcar el comienzo de un ciclo interminable de refactorización. Por ejemplo: piense en una aplicación que funcione bien y que sea estable con distintos modelos de datos, pero que esté adaptada a una base de datos relacional. Aunque la aplicación funcione bien, el código subyacente es largo



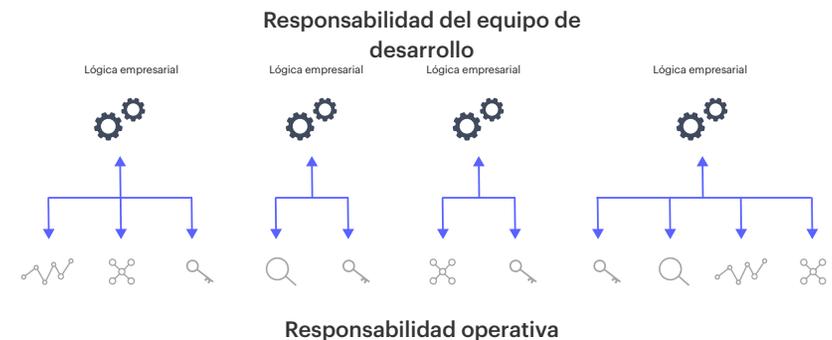
y difícil de mantener; gran parte de este no se atiene a la lógica empresarial, sino que busca resolver los problemas de falta de concordancia con la impedancia relacional de objetos. El peso del código dificulta añadir características nuevas y parches.

Para resolver este problema, el equipo de desarrollo contempla, digamos, una base de datos de gráficos, una base de datos de serie temporal y una base de datos de búsquedas que hagan innecesarias grandes partes del código base existente. Parece razonable cambiar a estas bases de datos diseñadas en función de su finalidad. Tiene lugar el proceso de reingeniería y la siguiente versión de la aplicación se escribe de forma tal que se comunique con estas tres nuevas bases de datos.

El equipo de desarrollo introduce rápidamente características nuevas... pero el entorno de producción ahora es lento e inestable. El equipo de operaciones observa problemas con el tiempo de implantación y tiene dificultades para resolver los problemas de tener varias bases de datos nuevas. Cada vez resulta más difícil diagnosticar el problema exacto debido a que las soluciones facilitadas en la nube son cajas negras con una visibilidad limitada a las bases de datos problemáticas. El equipo de operaciones empieza a defender el traslado a otra estrategia de bases de datos (lo más probable es el retorno a la base de datos relacional) que sea más estable y rinda mejor. El ciclo de molestias y gastos vuelve a empezar al desplazar la manta por una cama en la que simplemente no encaja.



La complejidad sigue dependiendo del equipo de desarrollo, ya que son ellos quienes tienen que asignar los datos relacionales de tal forma que reflejen el verdadero modelo de la aplicación (lógica empresarial).



La complejidad de la aplicación es operativa en lo que respecta a la conexión y el mantenimiento de varias bases de datos diseñadas en función de su finalidad, como defienden los proveedores de servicios en la nube.

“ La opción estratégica es buscar una vía que incorpore lo mejor de ambos mundos. Una vía que permita la coexistencia de varios modelos de datos nativos y que haga predecible la ampliación. ”



## Flexibilidad estratégica con Redis Enterprise

Hasta ahora, este documento solo ha abordado dos opciones, por inadecuadas que sean. Tanto la estrategia de SGBDR para todo como la de los modelos de varias bases de datos en la nube funcionan solamente a nivel táctico, ya que resuelven un problema de forma reglamentada y trasladan las dificultades de un sitio a otro de la organización, con lo que, en última instancia, se lleva a la organización a establecer proyectos de remodelación técnica costosos y cíclicos.

Una opción más estratégica es buscar una vía que incorpore lo mejor de ambos mundos. Una vía que permita la coexistencia de varios modelos de datos nativos y que haga predecible la ampliación. Una vía que permita a los desarrolladores emplear una biblioteca y conectarse con un extremo y disponer de acceso a varios modelos. Esta vía flexible y estratégica existe, y lleva a Redis Enterprise.

Redis Enterprise es una plataforma de bases de datos diseñadas en torno a un núcleo de Redis. Redis, sin extensiones, ofrece un modelo de acceso a datos de valor clave gracias a la adición de estructuras de datos como mapas de Hash, Lists, Sorted Sets, Streams y otros. Redis se puede ampliar con módulos que incorporen funciones totalmente nuevas a la base de datos. Estos módulos implantan modelos de datos nativos como los gráficos, la búsqueda de texto completo, el almacenamiento de documentos y las series temporales, así como también otras funciones en torno a la inteligencia artificial y la creación de scripts en función de los acontecimientos.

Las ventajas de la estrategia de Redis Enterprise se puede resumir en tres factores clave:

1. Una interfaz operativa, con varios modelos de datos
2. Una interfaz sencilla y unificada
3. Nacida y criada en la nube

## Una interfaz operativa, con varios modelos de datos

Redis Enterprise ofrece lo mejor de ambos mundos, ya que brinda una sola interfaz operativa en la que pueden coexistir muchos modelos de datos en un solo plano de datos. Redis Enterprise se ha desarrollado partiendo de la necesidad de los usuarios empresariales de poner en funcionamiento sus instalaciones de Redis. El ecosistema de Redis ha evolucionado hasta convertirse en un sistema sólido de módulos conectables que se asienta en lo profundo del corazón de Redis, lo cual permite a Redis ir más allá del modelo de valor clave integrado y alojar otros modelos de datos. Esta estrategia multimodelo le permite heredar las características operativas de Redis al tiempo que incorpora capacidades nuevas y potentes. A medida que amplíe, solamente tendrá que añadir más infraestructura a Redis Enterprise.

Activar un módulo en Redis Enterprise es tan sencillo como pulsar un botón en una IU; ahora bien, cambia drásticamente la forma en que se guardan y consultan los datos. Redis Labs ha producido varios módulos que ofrecen distintos modelos de datos, entre ellos:



### RedisSearch

Búsqueda de texto completo e indexación secundaria



### RedisGraph

Motor de gráficos compatible con cifrado



### RedisJSON

Almacenamiento de documentos JSON



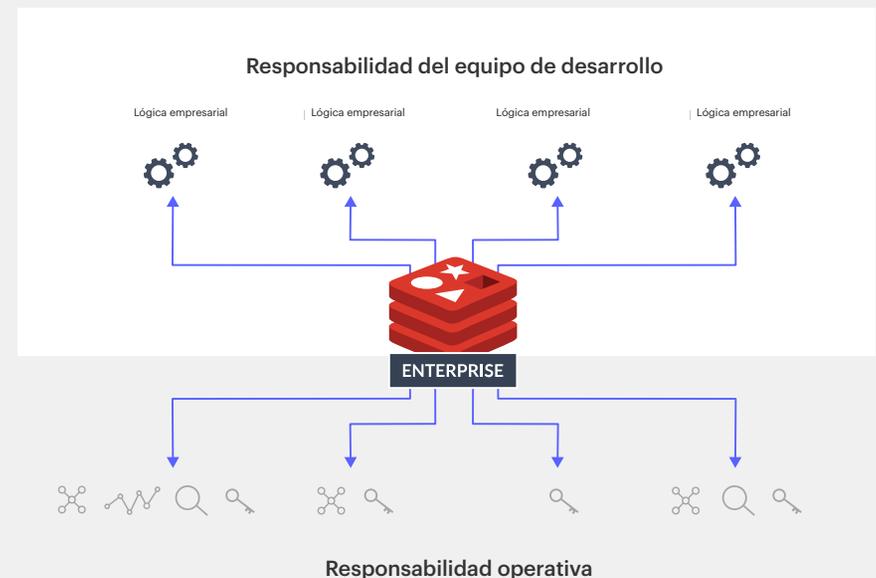
### RedisTimeSeries

Registro de muestras de datos en volúmenes elevados y consolidación, así como también reducciones de muestras sobre la marcha

Los módulos son opcionales: puede activar solamente aquellos que necesite para un proyecto concreto. Redis Enterprise es multiusuario, de forma que puede crear bases de datos para necesidades individuales con módulos específicos o bien puede crear bases de datos en las que muchos módulos operen de forma coordinada, lo cual le permitirá ampliar las capacidades específicas que necesita para su aplicación.

Estos módulos, junto con la estructura de datos de valores clave integrados de Redis, hacen posible una gran variedad de usos. Es importante entender que estos módulos no son capacidades añadidas al modelo de valor clave, sino que más bien modelos que acceden al interior de Redis para almacenar datos en la memoria y conservarlos en discos. Muchos de estos modelos se pueden habilitar en la misma base de datos, por lo que es posible, por ejemplo, emplear una búsqueda de texto completo en nodos de

“ Redis Enterprise ofrece lo mejor de ambos mundos, ya que brinda una sola interfaz operativa en la que pueden coexistir muchos modelos de datos en un solo plano de datos. ”



Redis Enterprise ofrece una separación elegante de los problemas de desarrollo y los operativos sin poner en riesgo la flexibilidad del modelo de datos.

“ Redis Enterprise es independiente en cuanto a la nube. Redis Labs no es proveedor de servicios en la nube, por lo que podrá elegir las opciones de nube e implementación que funcionen mejor para su organización y sus aplicaciones. ”



gráficos o almacenar documentos JSON junto con datos de series temporales. Como los distintos modelos de datos se dan en el mismo plano de datos, hasta puede haber transacciones entre modelos.

### Una interfaz sencilla y unificada

Como la base de datos de Redis puede atender a varios modelos del mismo plano de datos, se convierte en un punto de conexión único. Todas las propiedades de conexión se reducen a una y no hay penalizaciones de rendimiento o latencia extra por emplear varios modelos de datos nativos. Además, los desarrolladores tienen que entender solamente una interfaz programática para todos los distintos modelos de datos: todo se reduce al protocolo de Redis. De hecho, obtiene las ventajas de emplear bases de datos separadas en un solo paquete operativo.

### Nacida y criada en la nube

Redis siempre ha sido una solución para la nube. Redis Enterprise lo amplía con un paquete integral que tiene presente el entorno de la nube gracias a características como la conciencia de zonas y bastidores, monitorización y datos bifurcados y planos administrativos.

Como Redis emplea la memoria del sistema para el almacenamiento de datos primario, genera una dinámica operativa bien adaptada para ejecutarla en cualquier nube o en varias nubes. Redis Enterprise también es compatible con Redis on Flash, que posibilita el uso de instalaciones en la nube de almacenamiento en SSD como extensión de DRAM, gracias a lo cual los conjuntos de datos de gran tamaño pueden tener un alto rendimiento a un costo asequible. Como base de datos integrada en la memoria, Redis está listo para las nuevas generaciones de almacenamiento de datos a medida que el sector evoluciona desde discos que giran y del almacenamiento en flash por bloques al almacenamiento duradero al nivel de los bytes.

Es importante el hecho de que Redis Enterprise sea independiente de la nube. Redis Labs no es proveedor de servicios en la nube, por lo que podrá elegir la nube que funcione mejor para su organización y aplicaciones. Esto reduce el confinamiento a una sola nube y posibilita las implementaciones en nubes híbridas y en varias nubes. Además, Redis Labs ofrece distintos métodos de implementación: las opciones van desde un entorno en la nube de gestión total hasta despliegues en instalaciones físicas bajo licencia.

# Redis Enterprise: una solución estratégica

Resolver un solo problema suele ser relativamente fácil. Lo que suele ser más difícil es elegir opciones que resuelvan varios problemas. A esto se añade la necesidad de resolver varios problemas actuales al tiempo que se resuelven varios problemas de cara al futuro, y las cosas se pueden volver sumamente complejas. Este es el núcleo de la resolución estratégica de problemas. A veces se adoptan a la ligera las opciones de bases de datos, mientras que otras veces vienen predeterminadas desde antes del comienzo mismo de un proyecto. De cualquier forma, no se trata de una opción estratégica.

Redis Enterprise ofrece una nueva vía que puede resolver estratégicamente tanto los problemas de desarrollo como los operativos no solo para hoy, sino también en un futuro cercano. Atenúa las molestias que generan los modelos de datos en cuanto al desarrollo, el rendimiento y la estabilidad de las operaciones al mismo tiempo, lo cual ayuda a garantizar una decisión que superará la prueba del tiempo.



## Libertad de incorporar su propio modelo de datos

**Las aplicaciones diseñadas con Redis Enterprise: funcionan mejor gracias a que almacenan los datos de la forma en que su aplicación los usa.** Redis Enterprise admite distintos modelos integrados con gran cantidad de datos que se pueden utilizar para ajustarlo a la tarea que se esté realizando. En lo que respecta a las implementaciones completas, Redis Enterprise no intenta amoldar los datos a metáforas de datos inadecuadas, lo cual le ayuda a reducir el código de la aplicación y la complejidad. Reducir el código de la aplicación y la complejidad da lugar a que los desarrolladores sean más productivos y estén centrados en los problemas empresariales, no en intentar embutir datos en un modelo de datos inadecuado.



## Flexibilidad operativa integrada

Dado que Redis Enterprise permite que coexistan muchos modelos de datos en el mismo clúster en incluso en el mismo espacio de datos, elimina las conexiones entre bases de datos diferentes e independientes.

“ Redis Enterprise ofrece una nueva vía que puede resolver estratégicamente tanto los problemas de desarrollo como los operativos no solo para hoy, sino también en un futuro cercano. ”



“ El CTP de una base de datos debe tener presente no solo el precio por unidad de tiempo, sino también la repercusión organizativa. Su base de datos debe ser lo bastante flexible como para admitir de forma nativa más de un solo modelo de datos, con lo que optimizará y protegerá la capa de datos a futuro. ”



La consolidación de varios modelos en una conexión reduce la fragilidad y la inestabilidad que traen consigo varias conexiones. En Redis Enterprise, incluso puede ejecutar transacciones entre modelos de datos, algo que simplemente no se puede hacer con un conjunto de bases de datos independientes.



### Sin confinamiento a una sola nube

Como no procede de un proveedor de servicios en la nube, **Redis Enterprise no está confinado a una nube concreta: puede llevarse sus datos a una nube completamente nueva si surge la necesidad, o incluso desplegar una configuración híbrida o multinube.** Tiene pleno control sobre la infraestructura en la que opera Redis Enterprise y puede incorporar fácilmente más infraestructura para disponer de un clúster. Dentro del clúster, puede crear bases de datos aisladas o combinadas que tengan modelos de datos diseñados en función de la finalidad.



### Flexibilidad de la infraestructura

El costo total de propiedad de una base de datos debe tener presente no solo el precio por unidad de tiempo, sino también la repercusión organizativa. La solución que elija debe ser lo bastante flexible como para admitir de forma nativa más de un solo modelo de datos, con lo que optimizará y protegerá la capa de datos frente a variaciones y cambios imprevistos a futuro. **Redis Enterprise simplifica los problemas derivados de las capas de datos al otorgar a los desarrolladores y arquitectos la libertad y flexibilidad de diseñar aplicaciones que rindan sin sacrificar la eficiencia operativa.** Este equilibrio lleva adelante los intereses de quienes escriben software y de quienes tienen la responsabilidad de ejecutarlo, lo cual reduce las cargas indebidas sobre cualquier parte de la organización.

Tanto los proveedores de bases de datos de la vieja escuela como los proveedores de servicios en la nube disponen de plataformas desde las que proclaman a toda voz que ellos tienen la vía verdadera y que los demás están totalmente equivocados. Ambas partes tienen críticas razonables de los puntos débiles de los modelos de datos y la complejidad operativa de una parte frente a otra. Menos mal que la opción no es binaria: tiene opciones fuera de esta falsa dicotomía. Puede aprender más de las ventajas de Redis Enterprise en [redislabs.com/modules](https://redislabs.com/modules).



---

## Acerca de Redis Labs

Las empresas modernas dependen de la potencia de los datos en tiempo real. Con Redis Labs, las organizaciones ofrecen experiencias instantáneas de forma totalmente fiable y ampliable.

Redis Labs alberga Redis, la base de datos en una memoria más popular del mundo, y es el proveedor comercial de Redis Enterprise, que ofrece un rendimiento superior, una fiabilidad sin comparación y una flexibilidad imbatible en la personalización, el aprendizaje automático, el IoT, las búsquedas, el comercio electrónico, las redes sociales y las soluciones de cuantificación en todo el mundo.

Redis Labs, que de forma constante se sitúa como líder en informes de analistas de NoSQL, en bases de datos en la memoria, bases de datos operativas y bases de datos como servicio (DBaaS), cuenta con la confianza de más de 7400 clientes empresariales, entre ellos cinco de las 10 principales empresas según la revista Fortune, tres de los cuatro emisores de tarjetas de crédito, tres de las cinco principales empresas de comunicación, tres de las cinco principales empresas sanitarias, seis de las ocho principales empresas tecnológicas y cinco de los principales vendedores al por menor.

Redis Enterprise, disponible como servicio en nubes públicas y privadas, en forma de software descargable, en contenedores y en despliegues en nubes híbridas o en instalaciones físicas, impulsa las situaciones de uso populares de Redis como son las transacciones de alta velocidad, la gestión de trabajos y colas, los almacenamientos de sesiones de usuario, la ingesta de datos en tiempo real, el almacenamiento en caché del contenido y los datos de series temporales.

Síganos:



[redislabs.com](https://redislabs.com)