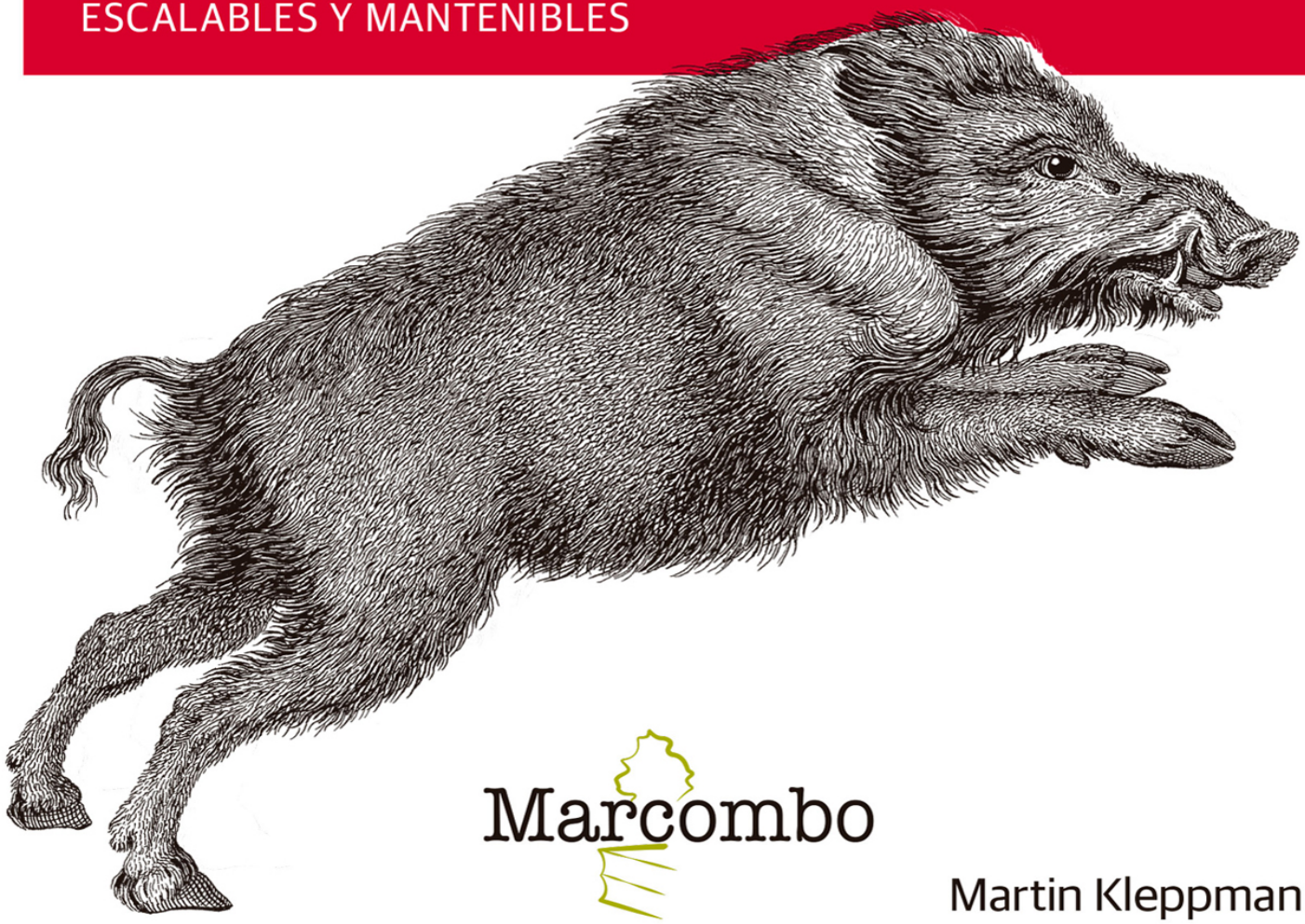
Diseño de aplicaciones mediante el uso intensivo de datos

LOS GRANDES CONCEPTOS SOBRE LOS SISTEMAS CONFIABLES, ESCALABLES Y MANTENIBLES



Marcombo

Martin Kleppmann

Diseño de aplicaciones mediante el uso intensivo de datos

Los grandes conceptos sobre los sistemas confiables, escalables y mantenibles

Martin Kleppmann

Marcombo

Primera edición original publicada en inglés por O'Reilly con el título *Designing Data-Intensive Applications,* ISBN 978-1-449-37332-0 ©Martin Kleppmann, 2017. Actualizada y corregida en enero de 2020.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Título de la edición en español:
Diseño de aplicaciones mediante el uso intensivo de datos

Primera edición en español, 2022

© 2022 MARCOMBO, S.L.
www.marcombo.com

Diseño de portada: Karen Montgomery
Ilustración: Rebecca Demarest
Traducción: Francisco Martínez Carreño
Corrección: Mónica Muñoz

Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra solo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. La presente publicación contiene la opinión del autor y tiene el objetivo de informar de manera precisa y concisa. La elaboración del contenido, aunque se ha trabajado de modo escrupuloso, no puede comportar una responsabilidad específica para el autor ni el editor de los posibles errores o imprecisiones que pudiera contener la presente obra.

ISBN: 978-84-267-3522-5
Producción del ePub: booqlab

La tecnología tiene una gran importancia en nuestra sociedad. Los datos, el software *y la comunicación pueden utilizarse para hacer el mal: afianzar estructuras de poder injustas, socavar los derechos humanos y proteger intereses creados. Pero también puede utilizarse para hacer el bien: para que se escuchen las voces de las personas que están insuficientemente representadas, para crear oportunidades para todos y para evitar desastres. Este libro está dedicado a todos quienes trabajan para hacer el bien.*

La informática es cultura pop […]. La cultura pop desprecia la historia. La cultura pop tiene que ver con la identidad y con sentirse partícipe. No tiene nada que ver con la cooperación, el pasado o el futuro: es vivir el presente. Creo que lo mismo ocurre con la mayoría de la gente que escribe código por dinero. No tienen ni idea de dónde viene [su cultura].

-Alan Kay, en una entrevista con el *Dr. Dobb's Journal* (2012)

Contenidos

3. Almacenamiento y recuperación

6. Particionado

7. Transacciones

8. El problema de los sistemas distribuidos

9. Coherencia y consenso

11. Procesamiento de flujos

12. El futuro de los sistemas de datos

Glosario

Prefacio

Si usted ha trabajado en ingeniería de *software* en los últimos años, especialmente en sistemas del lado del servidor y del *backend,* probablemente haya sido bombardeado con una plétora de palabras de moda relacionadas con el almacenamiento y el procesamiento de datos: «¡NoSQL!, *¡big data!,* ¡escala web!, ¡fragmentación!, ¡coherencia eventual!, ¡ACID!, ¡teorema CAP!, ¡servicios en la nube!, ¡MapReduce!, ¡tiempo real!».

En la última década, hemos asistido a muchos avances interesantes en bases de datos, en sistemas distribuidos y en la forma de construir aplicaciones sobre ellos. Hay varias fuerzas que impulsan estos desarrollos:

- Empresas de Internet como Google, Microsoft, Amazon, Facebook, LinkedIn, Netflix y Twitter manejan enormes volúmenes de datos y tráfico, lo que las obliga a crear nuevas herramientas que les permitan manejar eficientemente tal escala.
- Las empresas necesitan ser ágiles, probar hipótesis a bajo coste y responder con rapidez a los nuevos conocimientos del mercado, manteniendo ciclos de desarrollo cortos y modelos de datos flexibles.
- El *software* libre y de código abierto ha tenido mucho éxito y ahora, en muchos entornos, se prefiere al *software* comercial o al hecho a medida.
- Las velocidades de reloj de las *central processing units* (CPU) apenas aumentan, pero los procesadores multinúcleo son un estándar y las redes se presentan cada vez más rápidas. Esto significa que el paralelismo solo va a aumentar.barra

- Incluso si se trabaja en un equipo pequeño, ahora se pueden construir sistemas distribuidos en muchas máquinas e incluso en varias regiones geográficas, gracias a la infraestructura como servicio (IaaS), como es Amazon Web Services.
- Ahora se espera que muchos servicios tengan una alta disponibilidad; el tiempo de inactividad prolongado debido a las interrupciones o al mantenimiento resulta cada vez más inaceptable.

Las aplicaciones con uso intensivo de datos están superando los límites de lo posible, al hacer uso de estos desarrollos tecnológicos. Llamamos a una aplicación de *uso intensivo de datos* si estos son su principal reto: la cantidad de datos, su complejidad o la velocidad a la que cambian, en contraposición a la de *uso intensivo de computadores,* en la que los ciclos de CPU son el cuello de botella.

Las herramientas y tecnologías que ayudan a las aplicaciones con uso intensivo de datos a almacenarlos y procesarlos se han adaptado rápidamente a estos cambios. Los nuevos tipos de sistemas de bases de datos («NoSQL») han recibido mucha atención, pero las colas de mensajes, las cachés, los índices de búsqueda, los marcos de trabajo para el procesamiento por lotes y flujos y las tecnologías relacionadas son también muy importantes. Muchas aplicaciones utilizan alguna combinación de ellas.

Las palabras de moda que llenan este espacio constituyen una señal de entusiasmo por las nuevas posibilidades, lo cual es algo estupendo. Sin embargo, como ingenieros y arquitectos de *software,* también necesitamos tener una comprensión técnicamente exacta y precisa de las distintas tecnologías y sus contrapartidas, si queremos construir buenas aplicaciones. Para ello, hemos de profundizar más allá de las palabras de moda.

Afortunadamente, detrás de los rápidos cambios en la tecnología, existen principios perdurables que siguen siendo válidos, independientemente de la versión de la herramienta concreta que se utilice. Si entendemos esos principios, estaremos en condiciones de ver dónde encaja cada herramienta, cómo hacer un buen uso de ella y cómo evitar sus trampas. Ahí es donde entra en juego este libro.

El objetivo de esta obra radica en ayudarlo a navegar por el variado y cambiante panorama de las tecnologías de procesamiento y almacenamiento de datos. Este libro no es el tutorial de una herramienta concreta, ni un libro de texto lleno de árida teoría. En su lugar, veremos ejemplos de sistemas de datos que han alcanzado el éxito; tecnologías que forman la base de muchas aplicaciones populares y que han de cumplir requisitos de escalabilidad, rendimiento y fiabilidad en la producción del día a día.

Nos adentraremos en las entrañas de estos sistemas, esclareceremos sus algoritmos clave, discutiremos acerca de sus principios y las contrapartidas que tienen que cumplir. En este viaje, trataremos de encontrar formas útiles de *pensar en los sistemas de datos,* no solo cómo funcionan, sino también por qué lo hacen, y qué preguntas debemos hacer.

Después de leer este libro, se hallará en una excelente posición para decidir qué tipo de tecnología resulta apropiada para cada propósito, y entender cómo se pueden combinar las herramientas para formar la base de una buena arquitectura de aplicación. No estará preparado para construir su propio motor de almacenamiento de bases de datos desde cero, pero, afortunadamente, eso rara vez resulta necesario. Sin embargo, desarrollará una buena intuición sobre lo que hacen sus sistemas internamente, de modo que pueda razonar sobre su comportamiento, tomar buenas decisiones de diseño y localizar cualquier problema que pueda surgir.

¿Quién debería leer este libro?

Si desarrolla aplicaciones con algún tipo de servidor/*backend* para almacenar o procesar datos, y sus aplicaciones utilizan Internet (por ejemplo, aplicaciones web y móviles o sensores conectados a Internet), este libro es para usted.

La presente obra se dirige a ingenieros y arquitectos de *software,* así como a directores técnicos a quienes les gusta codificar. Es especialmente relevante si tiene que tomar decisiones sobre la arquitectura de los sistemas con los que trabaja; por ejemplo, si debe elegir herramientas para resolver un problema determinado y averiguar la mejor manera de aplicarlas. Pero,

incluso si no puede elegir sus herramientas, este libro lo ayudará a comprender mejor sus puntos fuertes y débiles.

Sería conveniente contar con alguna experiencia en la creación de aplicaciones basadas en la web o servicios de red, y debería estar familiarizado con las bases de datos relacionales y SQL. Las bases de datos no relacionales y otras herramientas vinculadas con los datos que pueda conocer constituyen una ventaja, pero no resultan necesarias. Será de utilidad un conocimiento general de los protocolos de red habituales, como TCP y HTTP. La elección que haga del lenguaje de programación o del marco de trabajo no supone ninguna diferencia a la hora de consultar este libro.

Si alguno de los siguientes puntos es cierto para usted, encontrará esta obra interesante:

- Quiere aprender a hacer que los sistemas de datos se presenten escalables; por ejemplo, para soportar aplicaciones web o móviles con millones de usuarios.
- Necesita hacer que las aplicaciones posean una alta disponibilidad (minimizando el tiempo de inactividad) y sean operativamente robustas.
- Busca formas de hacer que los sistemas resulten más fáciles de mantener a largo plazo, incluso a medida que crecen y cambian los requisitos y las tecnologías.
- Tiene una curiosidad natural acerca del funcionamiento de las cosas y quiere saber qué ocurre en los principales sitios web y servicios en línea. En este libro, se desglosan las interioridades de varias bases de datos y sistemas de procesamiento de datos, y resulta muy divertido explorar las brillantes ideas que se han utilizado en su diseño.

A veces, cuando se habla de sistemas de datos escalables, las personas hacen comentarios del tipo: «No eres Google o Amazon. Deja de preocuparte por la escala y utiliza una base de datos relacional». Esta afirmación resulta cierta: construir para una escala que no necesita supone un esfuerzo inútil y puede encerrarlo en un diseño inflexible. En efecto, es

una forma de optimización prematura. Sin embargo, también resulta importante elegir la herramienta adecuada para el trabajo, y las diferentes tecnologías cuentan con sus propios puntos fuertes y débiles. Como veremos, las bases de datos relacionales son importantes, pero no son la última palabra en el tratamiento de datos.

Alcance del libro

Con esta obra, no se pretende dar instrucciones detalladas sobre cómo instalar o utilizar paquetes de *software* o interfaces de programación de aplicaciones (*application programming interfaces,* API) específicos, ya que existe mucha documentación al respecto. En su lugar, se abarcan los diversos principios y contrapartidas fundamentales para los sistemas de datos, y se exploran las diferentes decisiones de diseño que se han adoptado en distintos productos.

En las ediciones de libros electrónicos, hemos incluido enlaces al texto completo de los recursos en línea. En el momento de la publicación, se verificaron todos los enlaces, pero, desgraciadamente, los vínculos tienden a romperse con frecuencia, debido a la naturaleza de la web. Si encuentra un enlace roto, o si está leyendo una copia impresa de este libro, puede buscar las referencias utilizando un motor de búsqueda. En el caso de los artículos académicos, puede buscar el título en Google Scholar para encontrar archivos pdf de libre acceso. También puede encontrar todas las referencias en *https://github.com/ept/ddia-references*, donde mantenemos los enlaces actualizados.

Nos centramos, principalmente, en la *arquitectura* de los sistemas de datos y en la forma en que se integran en las aplicaciones con uso intensivo de datos. En este libro no disponemos de espacio para tratar el despliegue, las operaciones, la seguridad, la gestión y otras áreas; son temas complejos e importantes, y no les haríamos justicia convirtiéndolos en notas secundarias y superficiales. Merecen libros propios.

Muchas de las tecnologías descritas entran dentro del ámbito de las palabras de moda *big data*. Sin embargo, la expresión *big data* está tan

excesivamente utilizada y tan pobremente definida que no es útil en una discusión seria de ingeniería. En este libro, se utilizan términos menos ambiguos, como «sistemas de un solo nodo» frente a «sistemas distribuidos», o «sistemas de procesamiento *online*/interactivo» frente a «sistemas fuera de línea/por lotes».

En esta obra, se manifiesta una inclinación hacia el *software* libre y de código abierto (*free and open-source software,* FOSS), porque leer, modificar y ejecutar el código fuente es una gran manera de entender cómo funciona algo en detalle. Las plataformas abiertas también reducen el riesgo de dependencia de un proveedor. Sin embargo, cuando se considera oportuno, también se habla de *software* propietario (*software* de código cerrado, *software* como servicio o *software* interno de las empresas, que solo se describe en la bibliografía, pero no se hace público).

Esquema del libro

El libro está organizado en tres partes:

1. En la parte I, analizamos las ideas fundamentales en las que se basa el diseño de aplicaciones con uso intensivo de datos. Comenzamos el capítulo 1 hablando de lo que realmente intentamos conseguir: fiabilidad, escalabilidad y mantenimiento; cómo debemos pensar sobre ello, y cómo podemos conseguirlo. En el capítulo 2, comparamos varios modelos de datos y lenguajes de consulta diferentes, y vemos cómo se muestran apropiados para diversas situaciones. En el capítulo 3, hablamos de los motores de almacenamiento: cómo las bases de datos organizan los datos en el disco, para que podamos encontrarlos de nuevo de forma eficiente. El capítulo 4 se centra en los formatos de codificación de datos (serialización) y en la evolución de los esquemas a lo largo del tiempo.
2. En la parte II, pasamos de los datos almacenados en una máquina a los distribuidos en varias. Esto se presenta a menudo necesario para la escalabilidad, pero trae consigo una variedad de desafíos

singulares. En primer lugar, tratamos la replicación (capítulo 5), la partición/repartición (capítulo 6) y las transacciones (capítulo 7). A continuación, profundizamos en los problemas de los sistemas distribuidos (capítulo 8) y en lo que significa conseguir coherencia y consenso en un sistema distribuido (capítulo 9).

3. En la parte III, se tratan los sistemas que derivan unos conjuntos de datos de otros conjuntos de datos. Los datos derivados suelen aparecer en sistemas heterogéneos: cuando no existe una base de datos que pueda hacerlo todo bien, las aplicaciones han de integrar varias bases de datos, cachés, índices, etc. En el capítulo 10, comenzamos con un enfoque de procesamiento por lotes para los datos derivados y lo ampliamos con el procesamiento de flujos en el capítulo 11. Por último, en el capítulo 12, lo ponemos todo junto y discutimos acerca de los enfoques para construir aplicaciones fiables, escalables y mantenibles en el futuro.

Referencias y lecturas complementarias

La mayor parte de la materia que tratamos en este libro ya se ha expuesto en otros lugares de una forma u otra: presentaciones de conferencias, artículos de investigación, publicaciones en blogs, código, rastreadores de errores, listas de correo y folclore de ingeniería. En este libro, se resumen las ideas más importantes de muchas fuentes diferentes e incluye referencias a la bibliografía original a lo largo del texto. Las referencias que aparecen al final de cada capítulo son un gran recurso si se quiere profundizar en un área, y la mayoría de ellas se encuentran disponibles gratuitamente en Internet.

Aprendizaje O'Reilly *online*

Durante más de cuarenta años, *O'Reilly Media* ha proporcionado formación en tecnología y negocios, conocimientos y perspectivas para ayudar a las empresas a tener éxito.

Nuestra red única de expertos y personal innovador comparte sus conocimientos y experiencia a través de libros, artículos, conferencias y

nuestra plataforma de aprendizaje en línea. La plataforma de aprendizaje en línea de O'Reilly le permite acceder a cursos de capacitación en vivo, rutas de aprendizaje en profundidad, entornos de codificación interactivos y una amplia colección de textos y vídeos de O'Reilly y de más de doscientos editores. Para más información, por favor, visite *http://oreilly.com*.

Cómo ponerse en contacto con nosotros

Tenemos una página web de este libro, en la que listamos las erratas, ejemplos y cualquier información adicional. Puede acceder a esta página en *http://bit.ly/designing-data-intensive-apps*.

Para comentar o hacer preguntas técnicas sobre el libro, envíe un correo electrónico a *bookquestions@oreilly.com*.

Reconocimientos

Este libro es una amalgama y sistematización de un gran número de ideas y conocimientos de otras personas, que combinan la experiencia tanto de la investigación académica como de la práctica industrial. En informática, tendemos a sentirnos atraídos por las cosas nuevas y brillantes, pero creo que tenemos mucho que aprender de las cosas que se han hecho antes. Este libro tiene más de ochocientas referencias a artículos, entradas de blog, charlas, documentación y mucho más, y han sido un recurso de aprendizaje inestimable para mí. Estoy muy agradecido a los autores de este material por compartir sus conocimientos.

También he aprendido mucho de las conversaciones personales, gracias a un gran número de personas que se han tomado el tiempo de discutir ideas o de explicarme pacientemente las cosas; en particular, me gustaría dar las gracias a Joe Adler, Ross Anderson, Peter Bailis, Márton Balassi, Alastair Beresford, Mark Callaghan, Mat Clayton, Patrick Collison, Sean Cribbs, Shirshanka Das, Niklas Ekström, Stephan Ewen, Alan Fekete, Gyula Fóra, Camille Fournier, Andres Freund, John Garbutt, Seth Gilbert, Tom Haggett, Pat Helland, Joe Hellerstein, Jakob Homan, Heidi Howard, John Hugg,

Julian Hyde, Conrad Irwin, Evan Jones, Flavio Junqueira, Jessica Kerr, Kyle Kingsbury, Jay Kreps, Carl Lerche, Nicolas Liochon, Steve Loughran, Lee Mallabone, Nathan Marz, CaitieMcCaffrey, Josie McLellan, Christopher Meiklejohn, Ian Meyers, Neha Narkhede, Neha Narula, Cathy O'Neil, Onora O'Neill, Ludovic Orban, Zoran Perkov, Julia Powles, Chris Riccomini, Henry Robinson, David Rosenthal, Jennifer Rullmann, Matthew Sackman, Martin Scholl, Amit Sela, Gwen Shapira, Greg Spurrier, Sam Stokes, Ben Stopford, Tom Stuart, Diana Vasile, Rahul Vohra, Pete Warden y Brett Wooldridge.

Muchas otras personas han contribuido, de forma inestimable, a la redacción de este libro revisando borradores y proporcionando comentarios. Por estas contribuciones estoy especialmente en deuda con Raul Agepati, Tyler Akidau, Mattias Andersson, Sasha Baranov, Veena Basavaraj, David Beyer, Jim Brikman, Paul Carey, Raúl Castro Fernández, Joseph Chow, Derek Elkins, Sam Elliott, Alexander Gallego, Mark Grover, Stu Halloway, Heidi Howard, Nicola Kleppmann, Stefan Kruppa, Bjorn Madsen, Sander Mak, Stefan Podkowinski, Phil Potter, Hamid Ramazani, Sam Stokes y Ben Summers. Por supuesto, asumo toda la responsabilidad por cualquier error u opinión desagradable que quede en este libro.

Por ayudar a que este proyecto se convierta en realidad, y por su paciencia con mi lentitud al escribir y mis inusuales peticiones, doy las gracias a mis editores Marie Beaugureau, Mike Loukides, Ann Spencer y a todo el equipo de O'Reilly. Por ayudarme a encontrar las palabras adecuadas, doy las gracias a Rachel Head. Por darme tiempo y libertad para escribir a pesar de otros compromisos laborales, doy las gracias a Alastair Beresford, Susan Goodhue, Neha Narkhede y Kevin Scott.

Hay que agradecer, muy especialmente, a Shabbir Diwan y Edie Freedman, que ilustraron con gran esmero los mapas que acompañan a los capítulos. Es maravilloso que hayan asumido la idea poco convencional de crear mapas, y que los hayan hecho tan bellos y convincentes.

Por último, mi cariño va para mi familia y mis amigos, sin los cuales no habría podido superar este proceso de escritura que ha durado casi cuatro años. Sois los mejores.

PARTE I
Fundamentos de los sistemas de datos

En los cuatro primeros capítulos, se repasan las ideas fundamentales que se aplican a todos los sistemas de datos, tanto si se ejecutan en una sola máquina como si se distribuyen en un *cluster* de máquinas:

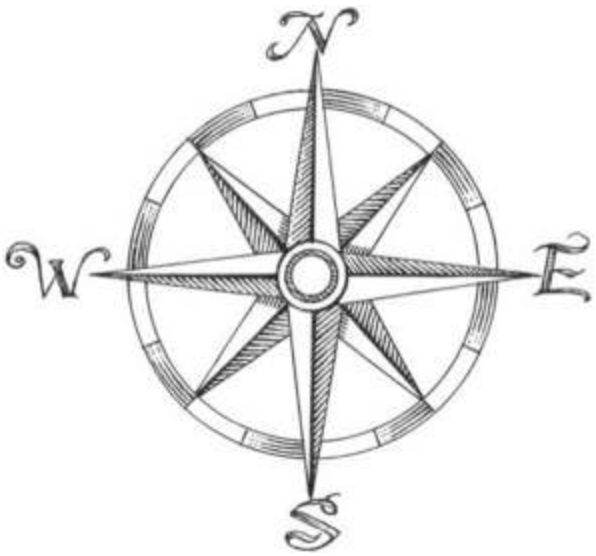
1. En el capítulo 1, se introduce la terminología y el enfoque que vamos a utilizar a lo largo del libro. Se examina lo que realmente queremos decir con palabras como *fiabilidad, escalabilidad* y *mantenimiento*, y cómo podemos intentar alcanzar estos objetivos.
2. En el capítulo 2, se comparan varios modelos de datos y lenguajes de consulta diferentes, el factor de distinción más visible entre las bases de datos desde el punto de vista del desarrollador. Veremos cómo los diferentes modelos son apropiados para distintas situaciones.
3. El capítulo 3 se centra en los aspectos internos de los motores de almacenamiento y se analiza también cómo las bases de datos distribuyen los datos en el disco. Los diferentes motores de almacenamiento están optimizados para diferentes cargas de trabajo, y la elección del correcto puede tener un gran efecto en el rendimiento.
4. En el capítulo 4, se comparan varios formatos de codificación de datos (serialización) y se examina especialmente su comportamiento, en un entorno en el que los requisitos de las aplicaciones cambian y los esquemas deben adaptarse con el tiempo.

Más tarde, en la parte II, se abordan los problemas particulares de los sistemas de datos distribuidos.

CAPÍTULO 1
Aplicaciones confiables, escalables y mantenibles

> Internet se hizo tan bien que la mayoría de la gente piensa que es un recurso natural como el océano Pacífico, en lugar de algo hecho por el hombre. ¿Cuándo fue la última vez que una tecnología de esta envergadura estuvo tan libre de errores?
>
> —Alan Kay, en una entrevista en *Dr. Dobb's Journal* (2012)


DISEÑO
de aplicaciones de uso intensivo de datos
Las grandes ideas sobre los sistemas confiables, escalables y mantenibles
CONFIABILIDAD
ESCALABILIDAD
MANTENIBILIDAD
CONFIABILIDAD
Tolerancia a fallos de hardware y software
Errores humanos
ESCALABILIDAD
Medición carga y rendimiento
Percentiles de latencia
MANTENIBILIDAD
Operatividad, sencillez y evolucionabilidad



Hoy día, muchas aplicaciones hacen un *uso intensivo de datos,* en lugar de uno intensivo de cálculo. La potencia bruta de la CPU rara vez es un factor limitante para estas aplicaciones: los problemas más importantes suelen estribar en la cantidad de datos, su complejidad y la velocidad a la que cambian.

Una aplicación con uso intensivo de datos se crea normalmente a partir de bloques de construcción estándar, que proporcionan la funcionalidad normalmente necesaria; por ejemplo, muchas aplicaciones precisan:

- Almacenar datos para que ellas, o cualquier otra aplicación, puedan volver a encontrarlos más tarde *(bases de datos)*
- Recordar el resultado de una operación onerosa, para acelerar las lecturas *(cachés)*
- Permitir a los usuarios buscar datos por palabras clave o filtrarlos de varias maneras *(índices de búsqueda)*
- Enviar un mensaje a otro proceso, para que lo gestione de forma asíncrona *(procesamiento de flujos)*

- Triturar periódicamente una gran cantidad de datos acumulados *(procesamiento por lotes)*

Que esto sea tan obvio se debe, precisamente, a que tales *sistemas de datos* representan una abstracción muy lograda: los usamos todo el tiempo, sin pensar demasiado. Cuando se construye una aplicación, a la mayoría de los ingenieros no se les ocurriría escribir un nuevo motor de almacenamiento de datos desde cero, porque las bases de datos son una herramienta perfectamente adecuada para realizar el trabajo.

Pero la realidad no es tan sencilla. Hay muchos sistemas de bases de datos con diferentes características, porque las distintas aplicaciones poseen diversos requisitos. Existen varios enfoques para el almacenamiento en caché y varias formas de construir índices de búsqueda. A la hora de crear una aplicación, tenemos que averiguar qué herramientas y qué enfoques son los más apropiados para la tarea en cuestión. Y puede resultar difícil combinar herramientas cuando se necesita hacer algo que una sola herramienta no puede hacer por sí sola.

Este libro constituye un viaje a través de los principios y los aspectos prácticos de los sistemas de datos, y de cómo se pueden utilizar para crear aplicaciones con uso intensivo de datos. Exploraremos qué poseen en común las distintas herramientas, qué las distingue y cómo consiguen sus características.

En este capítulo, empezaremos explorando los fundamentos de lo que estamos tratando de conseguir: sistemas de datos fiables, escalables y mantenibles. Aclararemos qué significan esas cosas, esbozaremos algunas formas de pensar sobre ellas y repasaremos los fundamentos que necesitaremos para abordar los capítulos posteriores. En los siguientes capítulos, continuaremos capa por capa, viendo las diferentes decisiones de diseño que se deben considerar cuando se trabaja en una aplicación intensiva de datos.

Reflexiones sobre los sistemas de datos

Normalmente pensamos que las bases de datos, las colas, las cachés, etc., son categorías de herramientas muy diferentes. Aunque una base de datos y una cola de mensajes presentan algunas similitudes superficiales, ambas almacenan datos durante algún tiempo (tienen patrones de acceso muy diferentes), lo que significa que poseen características de rendimiento distintas y, por tanto, implementaciones muy diferentes.

Entonces, ¿por qué debemos agruparlas todas bajo una expresión general como *sistemas de datos*?

En los últimos años, han surgido muchas herramientas nuevas para el almacenamiento y procesamiento de datos. Están optimizadas para una gran variedad de casos de uso y ya no se ajustan a las categorías tradicionales [1]; por ejemplo, algunos almacenes de datos también se utilizan como colas de mensajes (Redis), y hay colas de mensajes con garantías de durabilidad similares a las de las bases de datos (Apache Kafka). Los límites entre las categorías son cada vez más difusos.

En segundo lugar, cada vez son más las aplicaciones con requisitos tan exigentes o amplios que una sola herramienta ya no puede satisfacer todas sus necesidades de procesamiento y almacenamiento de datos. En su lugar, el trabajo se divide en tareas que *pueden* realizarse de forma eficiente con una sola herramienta, y esas diferentes herramientas se unen mediante el código de la aplicación.

Por ejemplo, si tenemos una capa de caché gestionada por la aplicación (usando Memcached o similar), o un servidor de búsqueda de texto completo (como Elasticsearch o Solr) separado de la base de datos principal, normalmente es responsabilidad del código de la aplicación mantener esas cachés e índices sincronizados con la base de datos principal. La figura 1.1 aporta una idea de cómo se puede conseguir esto (entraremos en detalle en capítulos posteriores).