

# Introduccion a NoSQL

# Qué es NoSQL?

- ▶ Not Only SQL
- ▶ Proporciona un mecanismo para el almacenamiento y obtención de información que se modela en un formato distinto a las relaciones tabulares propias de las bases de datos relacionales.
- ▶ Esta nueva forma de trabajar responde a otra forma de organización de los datos que permiten una menor rigidez de los datos y formas novedosas de trabajo

# Motivación

- ▶ Existen aplicaciones para las que las bases de datos relacionales no son la mejor solución...
- ▶ O para las cuales no todo se resuelve mejor usando exclusivamente una base de datos relacional.

# Qué es NoSQL?

- No cumple con el esquema entidad-relación
- No impone una estructura de datos
- Almacena los datos en diferentes formatos

# Características

- ▶ **Consistencia:** No se implementan mecanismos rígidos de consistencia como los presentes en las bases de datos relacionales, donde la confirmación de un cambio implica una comunicación del mismo a todos los nodos que lo repliquen. Las bases de datos NoSQL son ACID, esto significa que una transacción cumple lo siguiente:
  - Atomicidad
  - Consistencia
  - Aislamiento
  - Durabilidad

# Características

- **Atomicidad:** es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- **Consistencia:** es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.

# Características

- **Aislamiento:** es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información sean independientes y no generen ningún tipo de error.
- **Durabilidad:** es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

# Características

- ▶ **Estructura distribuida:** Generalmente se distribuyen los datos mediante mecanismos distribuidos como las redes P2P.
- ▶ **Escalabilidad Horizontal:** La implementación típica se realiza en muchos nodos de capacidad de procesamiento limitado, en vez de utilizar grandes '*mainframes*'.
- ▶ **Tolerancia a fallos** (debido a la estructura ACID), redundancia y sin cuellos de botella.



# Ventajas

- ▶ Preparadas para grandes volúmenes de información estructurada, semi estructurada y sin estructurar.
- ▶ Facilitan iteración rápida, metodología ágiles y despliegue continuo.
- ▶ Coherente con la programación orientada a objetos
- ▶ Arquitectura eficiente y escalable.

# Desventajas

- ▶ No son suficientemente maduros para algunas empresas
- ▶ Falta de experiencia
- ▶ No trata con datos críticos que requieren ACID
- ▶ Problemas de compatibilidad

# Taxonomía de soluciones NoSQL

- ▶ Almacenes de clave – valor
  - Estas son las más sencillas de entender. Simplemente guardan tuplas que contienen una clave y su valor. Cuándo se quiere recuperar un dato, simplemente se busca por su clave y se recupera el valor.
  - En esta categoría encontramos:
    - DynamoDB: desarrollada por Amazon, es una opción de almacenaje que podemos usar desde los Amazon Web Services. La utilizan el Washington Post y Scopely.
    - Redis: desarrollada en C y de código abierto, es utilizada por Craigslist y Stack Overflow (a modo de caché).

# Taxonomía de soluciones NoSQL

## ▶ Almacenes de familia de columnas

- Este tipo de bases de datos están pensadas para realizar consultas y agregaciones sobre grandes cantidades de datos. Funcionan de forma parecida a las bases de datos relacionales, pero almacenando columnas de datos en lugar de registros.
- En esta categoría encontramos:
  - Cassandra: incluida en esta sección, aunque en realidad sigue un modelo híbrido entre orientada a columnas y clave-valor. Es utilizada por Facebook y Twitter (aunque dejaron de usarla para almacenar tweets).
  - HBase. Escrita en Java y mantenida por el Proyecto Hadoop de Apache, se utiliza para procesar grandes cantidades de datos. La utilizan Facebook, Twitter o Yahoo.

# Taxonomía de soluciones NoSQL

## ► Almacenes de documentos

- Son aquellas que gestionan datos semi estructurados. Es decir documentos. Estos datos son almacenados en algún formato estándar como puede ser XML, JSON o BSON. Para hacernos una idea un documento suele ser algo parecido a:

```
{
  Name: "Genbeta Dev",
  Tipo: "Bloggging",
  Categorías:
  [
    {
      Título: "Desarrollo",
      Artículos: 89
    },
    {
      Título: "Formación",
      Artículos: 45
    }
  ]
}
```

# Taxonomía de soluciones NoSQL

- Son las bases de datos NoSQL más versátiles. Se pueden utilizar en gran cantidad de proyectos, incluyendo muchos que tradicionalmente funcionarían sobre bases de datos relacionales.
- En esta categoría encontramos:
  - MongoDB: probablemente la base de datos NoSQL más famosa del momento. Algunas compañías que actualmente utilizan MongoDB son Foursquare o eBay.
  - CouchDB: es la base de datos orientada a documentos de Apache. Una de sus interesantes características es que los datos son accesibles a través de una API Rest. Este sistema es utilizado por compañías como Credit Suisse y la BBC.

# Taxonomía de soluciones NoSQL

## ▶ Almacenes de grafos

- Basadas en la teoría de grafos utilizan nodos y aristas para representar los datos almacenados. Son muy útiles para guardar información en modelos con muchas relaciones, como redes y conexiones sociales.
- En esta categoría encontramos:
  - [Infinite Graph](#): escrita en Java y C++ por la compañía Objectivity. Tiene dos modelos de licenciamiento: uno gratuito y otro de pago.
  - [Neo4j](#): base de datos de código abierto, escrita en Java por la compañía Neo Technology. Utilizada por compañías como HP, Infojobs o Cisco.

# Arquitectura NoSQL

- ▶ Las bases de datos relacionales tradicionales nos permiten definir la estructura de un esquema que demanda regla rígidas y garantizan ACID.
- ▶ Las aplicaciones web modernas presentan desafíos muy distintos a las que presentan los sistemas empresariales tradicionales (e.j. sistemas bancarios)
  - Datos a escala web
  - Alta frecuencia de lecturas y escrituras
  - Cambios de esquema de datos frecuentes
  - Las aplicaciones sociales (no bancarias) no necesitan el mismo nivel de ACID

Algunas de las opciones de NoSQL actualmente disponibles son: Cassandra, MongoDB, Jackrabbit, CouchDB, BigTable y Dynamo.



# Mongo DB

- ▶ MongoDB es un gestor de datos NoSQL distribuido de tipo documental que almacena documentos en un formato similar a JSON (para ser más exactos internamente usa BSON). Está escrita en C++ y es multi-plataforma, Open Source y gratuito.

# Mongo DB

- ▶ El proyecto nació a finales del año 2007 como un proyecto interno de una empresa llamada **10Gen** para usarlo en una aplicación de Internet que estaban desarrollando, pero en 2009 decidieron liberarlo como Open Source y dedicarse íntegramente a él, ofreciendo soporte comercial y servicios relacionados.

# Mongo DB

- ▶ Su nombre proviene de la palabra en inglés **Humongous**, que significa literalmente "algo realmente grande", y se refiere a su capacidad de gestionar cantidades enormes de datos.

# Mongo DB

- ▶ Sus principales características son:
  - Basado en el motor V8 de Google Chrome para JavaScript. Facilidad de aprendizaje por basarse en este lenguaje.
  - Almacenamiento flexible basado en JSON sin necesidad de definir esquemas previamente.
  - Soporte para creación de índices a partir de cualquier atributo, lo que facilita mucho su uso porque no es necesario definir procesos Map-Reduce.

# Mongo DB

Sus principales características son (cont):

- Alto rendimiento para consultas y actualizaciones.
- Consultas flexibles basadas en documentos.
- Alta capacidad de crecimiento, replicación y escalabilidad: puedes escalar horizontalmente simplemente añadiendo máquinas baratas sin ver afectado el rendimiento ni complicar la gestión.
- Soporte para almacenamiento independiente de archivos de cualquier tamaño basado en GridFS.
- Soporte para tareas Map-Reduce si es necesario.

# Mongo DB

- ▶ Aplicaciones

Este tipo de bases de datos no sustituyen a las bases de datos tradicionales, como SQL Server, Oracle o MySQL, sino que **las complementan para ciertos tipos de aplicaciones** especializadas. Las bases de datos documentales como MongoDB se utilizan para multitud de tareas, pero fundamentalmente cuando necesitamos **flexibilidad** en la definición de los datos, **sencillez** a la hora de acceder a éstos, **gran rendimiento** y posibilidad de **crecer muy rápido**.



# Mongo DB

- ▶ Principales componentes
  - Mongod (núcleo de la base de datos)
  - Mongos (controlador de la base de datos)
  - GridFS (Función de Almacenamiento)

# Mongo DB

- ▶ **Mongod**

Motor central de la BD, puede correr como programa o demonio y posee tres funcionalidades

- Standalone Server

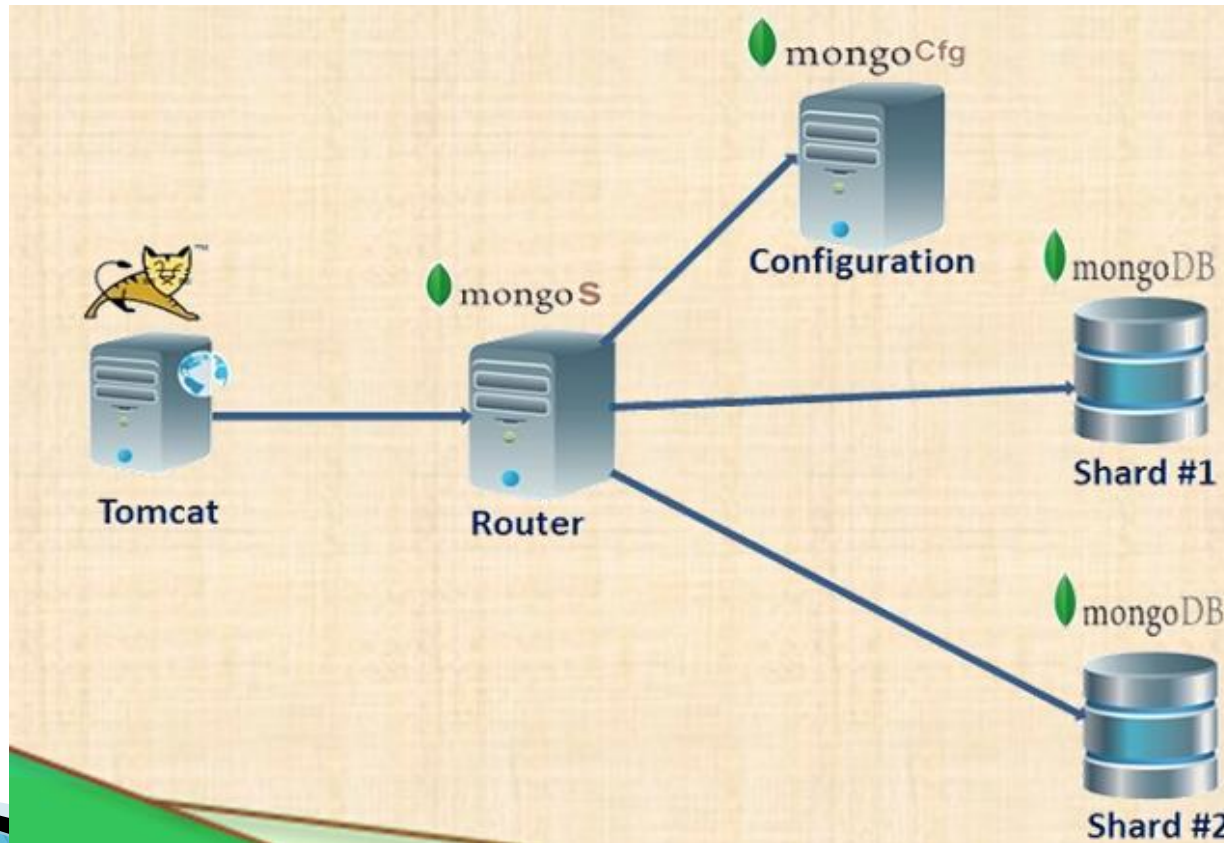
- Config Server

- Shard Partition



# Mongo DB

- ▶ Mongos
  - Brinda transparencia a las aplicaciones clientes



# Mongo DB

- ▶ **Mongod**

Motor central de la BD, puede correr como programa o demonio y posee tres funcionalidades

- Standalone Server

- Config Server

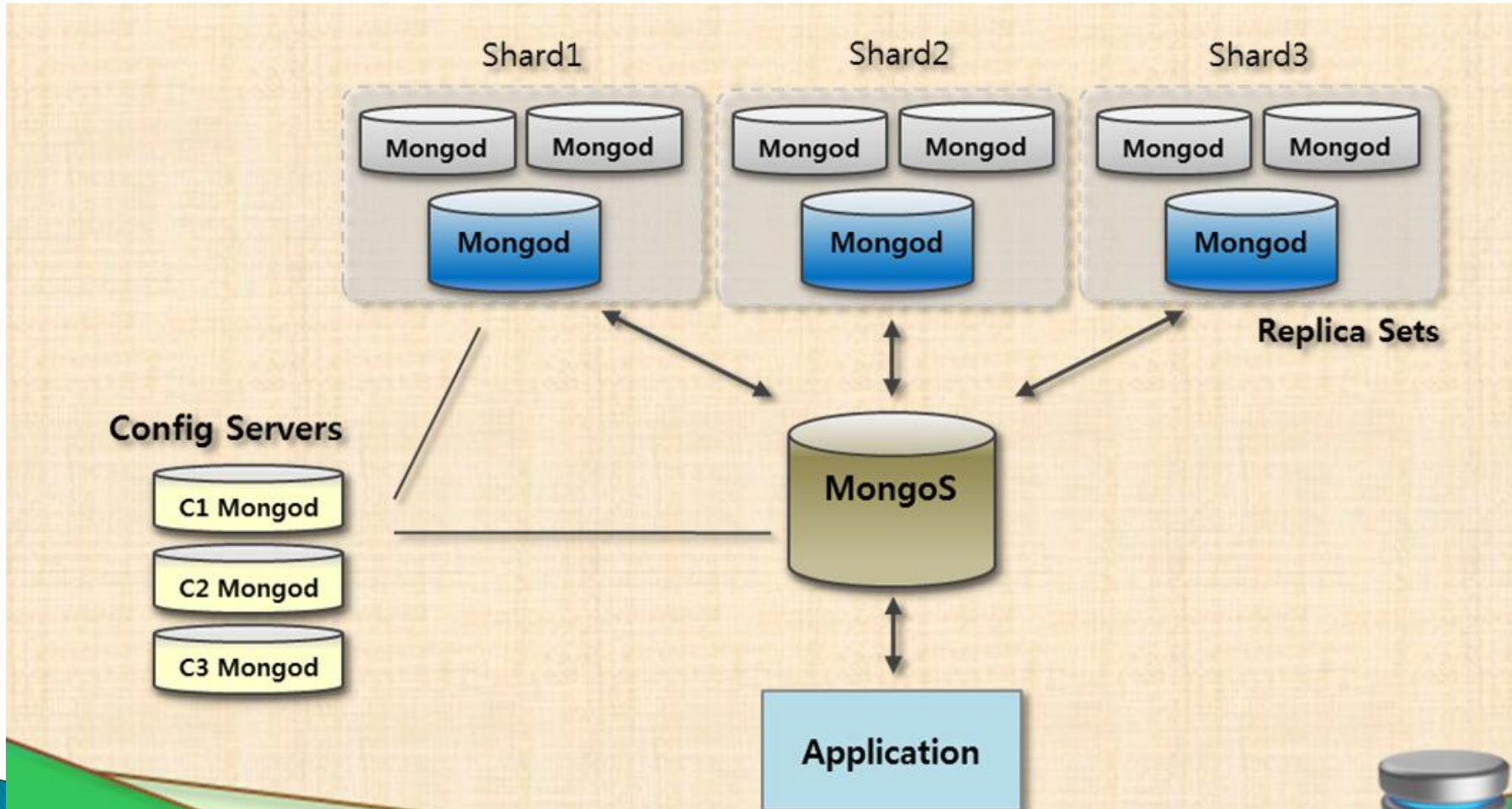
- Shard Partition

# Mongo DB

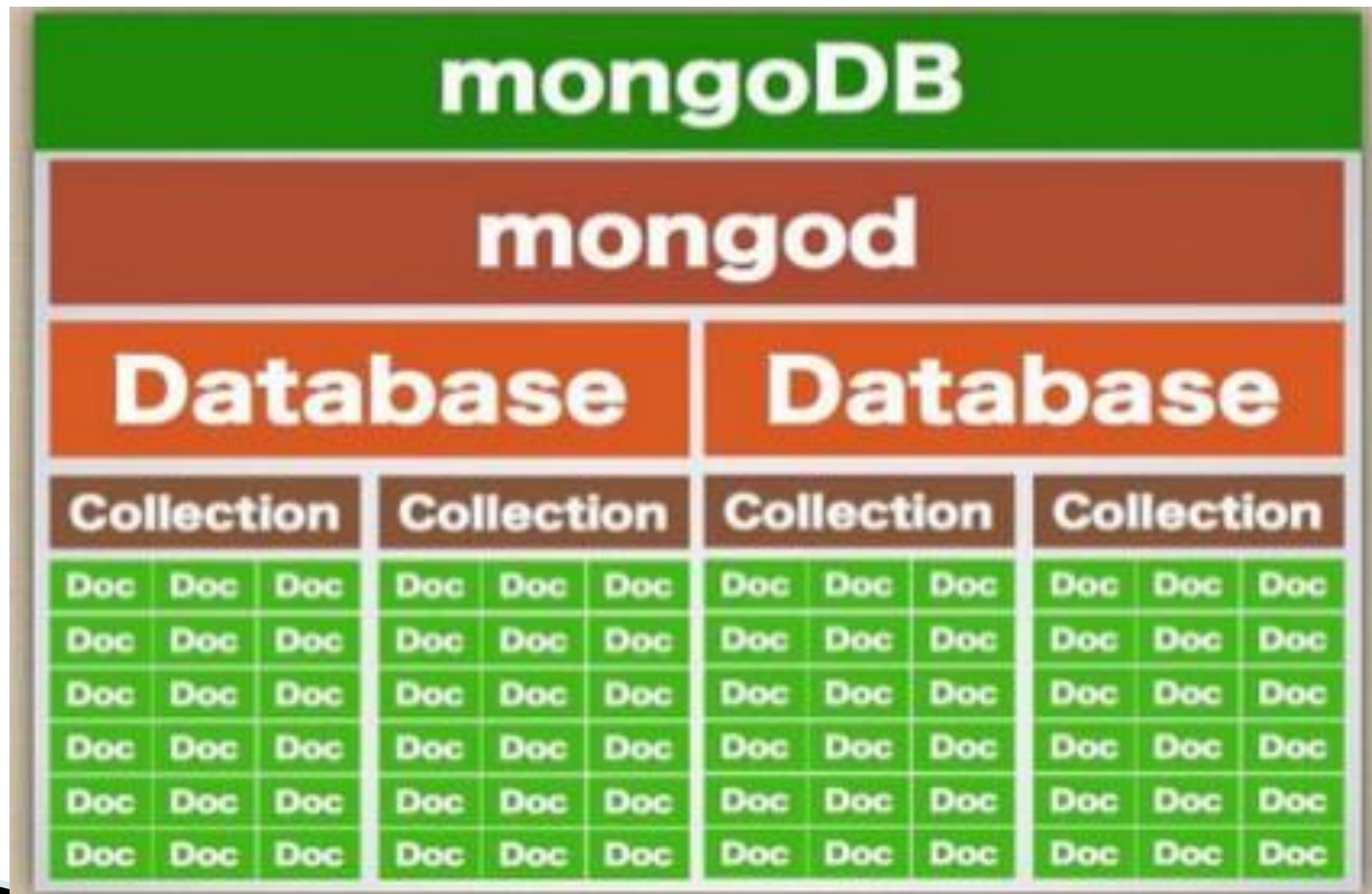
## ▶ GridFS

- Funcion que almacena y extrae ficheros de la BD
- Define dos colecciones files y chunks
- Ficheros mayores a 10MB se dividirán en partes menores, y cada bloque se guardará como un documento en la colección de bloques.

# Mongo DB. Arquitectura

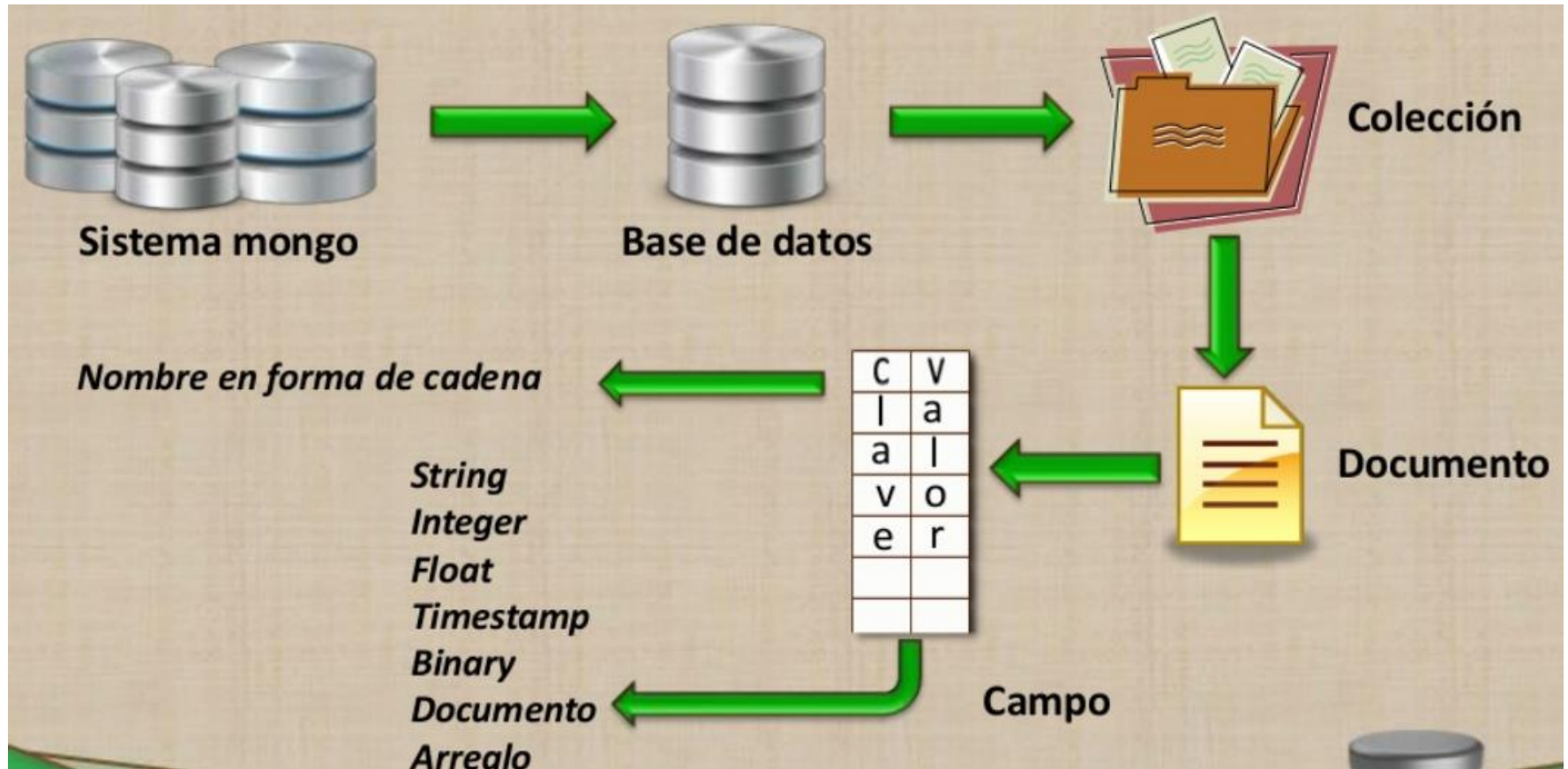


# Mongo DB. Modelo de Datos





# Mongo DB. Modelo de Datos



# Mongo DB

## ► Esquema Flexible

- Las colecciones no fuerzan una estructura idéntica para los documentos.
- Los documentos no necesitan la misma cantidad de campos, y aquellos comunes pueden contener diferentes tipos de datos
- Cada documento necesita un número relevante de campos de la entidad.

# Mongo DB

- ▶ Estructura de un documento
  - Guarda la estructura de datos en documentos tipo BSON, usando un esquema dinamico con la siguiente disposición:
  - { campo1: valor1,  
    campo2: valor2,  
    campo3: valor3,  
    ...  
    campoN: valorN  
}



# Mongo DB

- ▶ Respaldos y Restauración
  - Respaldos
    - Se utiliza el ejecutable mongodump
  - Restauración
    - Se utiliza mongorestore

# Mongo DB

SQL	MongoDB
CREATE TABLE users ( id MEDIUMINT NOT NULL AUTO_INCREMENT, user_id Varchar(30), age Number, status char(1), PRIMARY KEY (id));	db.users.insert({user_id: "abc123", age: 55, status: "A"}) odb.createCollection("users")
DROP TABLE users	db.users.drop()
INSERT INTO users(user_id,age, status) VALUES ("bcd001", 45, "A")	db.users.insert( { user_id: "bcd001", age: 45, status: "A" } )
SELECT * FROM users	db.users.find()
SELECT COUNT(*) FROM users	db.users.count() odb.users.find().count()
SELECT * FROM users WHERE status = "A" ORDER BY user_id DESC	db.users.find( { status: "A" } ). sort( { user_id: -1 } )
UPDATE users SET status = "C" WHERE age > 25	db.users.update( { age: { \$gt: 25 } }, { \$set: { status: "C" } }, { multi: true } )
DELETE FROM users WHERE status = "D"	db.users.remove( { status: "D" } )

# Mongo DB

## ► Aplicaciones

Es adecuada para crear **aplicaciones de Internet** que registren muchos datos o que simplemente queramos crear de manera muy flexible, pero también para **sistemas muy grandes** como registradores de datos de sensores, que pueden llegar a recibir decenas o cientos de miles de lecturas de datos por segundo, pasando por **gestores de datos de ventas**, infraestructura de almacenamiento para **redes sociales**, **juegos masivos online**, **gestores de contenidos**, aplicaciones de **análisis de datos** *y reporting...*

# Mongo DB

## ► Aplicaciones

Actualmente lo utilizan para empresas como eBay, Foursquare, SourceForge, The New York Times, The Guardian, SAP, pero también muchas empresas pequeñas que quieren poder desarrollar de manera ágil, barata, sencilla y sin miedo a poder crecer más adelante.

Al estar basada en JavaScript se lleva especialmente bien con todo tipo de aplicaciones web, especialmente las más modernas *Single Page Applications*.

En general **MongoDB puede utilizarse para casi cualquier cosa para la que utilizarías SQL Server o MySQL**, pero sin la rigidez que presentan este tipo de bases de datos.