



# TEMA-12-ACCESO-A-DATOS.pdf



**user\_4383848**



**Acceso a datos**



**1º Desarrollo de Aplicaciones Multiplataforma**



**Estudios España**

antes



**Descarga sin publi  
con 1 coin**



Después

**WUOLAH**



# TODOS <sup>LOS</sup> LADOS DE LA CAMA

14 NOVIEMBRE  
SOLO EN CINES

## TEMA 12 - ACCESO A DATOS OPERACIONES CON BD NO-SQL

### 1. BASE DE DATOS MONGODB:

Características:

- Base de datos NoSQL orientada a documentos.
- La estructura de los documentos es en formato JSON. Internamente los documentos son almacenados en formato BSON.
- Al tener documentos más ricos, se reduce el I/O (input/output) sobre la base de datos. No existe la necesidad de hacer Joins.
- Permite operaciones CRUD (Create, Read, Update, Delete) con una sintaxis parecida a JavaScript.
- Proporciona replicación y alta disponibilidad a través de Replica Sets.
- También dispone de balanceo y escalado horizontal usando Sharding. El balanceo de los datos se realiza automáticamente.
- Ofrece un mecanismo de procesamiento masivo de datos a través de operaciones de agregación.
- El grueso de los datos reside en memoria, por lo que las lecturas y escrituras son muy rápidas.
- Permite también crear colecciones de tipo circular de tamaño fijo, y mantiene el orden según se han ido insertando los datos (Capped Collections).
- Tienen múltiples motores de almacenamiento diferentes, mmap, WiredTiger, In-memory y Encrypted.
- Es enorme, forma parte de Big Data.

#### 1.1. MONGODB EL TEOREMA DEL CAP:

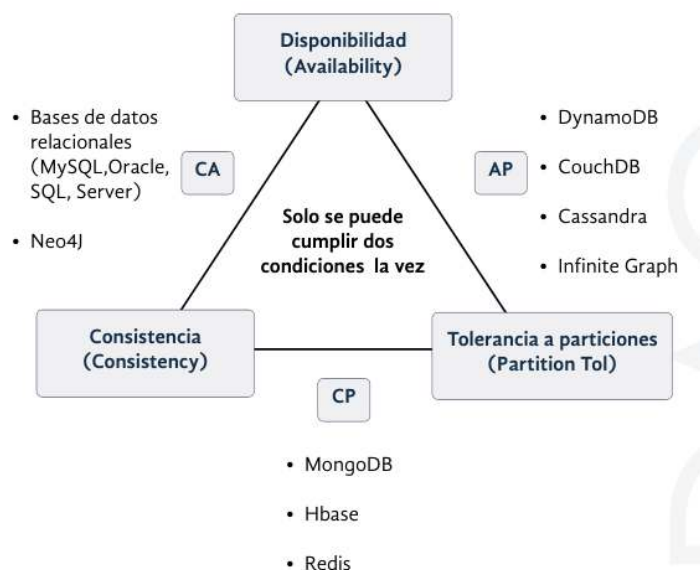
El teorema de CAP sostiene, que en sistemas distribuidos **no es posible garantizar al completo la consistencia, la disponibilidad y tolerancia.**

- **Consistencia:** La información que obtenemos a la hora de realizar una consulta debe ser siempre la misma.
- **Disponibilidad:** Todos los clientes deben de poder realizar las operaciones de lectura y escritura, aunque un nodo se haya caído.
- **Tolerancia a particiones:** Los sistemas pueden estar divididos en particiones distribuidas en distintos puntos. Esta propiedad consiste en, a pesar de esta división, asegurar el funcionamiento.

Según el teorema de CAP, podemos clasificar todas las bases de datos. No todas cumplen los mismos puntos del teorema.

- **Las bases de datos más cercanas al vértice AP:** aseguran la disponibilidad y tolerancia a particiones, pero no la consistencia, al menos en su totalidad. Algunas de ellas a través de la replicación y verificación, sí consiguen parte de consistencia.
- **Aquellas que su vértice esté más cercano a CP:** estarán del lado de la consistencia y tolerancia a particiones. Sacrifican la disponibilidad al replicar los datos a través de nodos.
- **Por último, las bases de datos más cercanas al vértice CA:** poseerán más consistencia y disponibilidad dejando un poco de lado la tolerancia a particiones. Este problema se solucionará replicando los datos.

WUOLAH



## 1.2. CASOS DE USO Y DOCUMENTOS:

### a. Casos de uso y términos

MongoDB es un producto de propósito general y es muy útil para múltiples casos de uso tales como:

- CMS, Aplicaciones móviles.
- Gaming.
- E-commerce.
- Business intelligence.
- Analytics.
- Proyectos Big Data.
- Web Caché.

Términos utilizados en MongoDB y su correspondencia en el mundo relacional:

Relacional	MongoDB
Base de datos	Base de datos
Tabla	Colección
Fila	Documento
Índice	Índice
<i>Insert</i>	<i>Insert</i>
<i>Select</i>	<i>Find</i>
<i>Update</i>	<i>Update</i>
<i>Delete</i>	<i>Remove</i>

## b. Documentos

- Cada entrada de una colección es un documento.
- Son estructuras de datos compuestas por campos de clave/valor.
- Los documentos tienen una estructura similar a objetos JSON.
- Los documentos se corresponden con tipos de datos nativos en los lenguajes de programación.
- En un documento es posible embeber otros documentos o arrays.
- Se tiene un esquema dinámico que permite polimorfismo de manera fluida.
- Las operaciones de escritura son solo atómicas a nivel de documento.

## 2. FICHEROS BINARIOS DE MONGODB:

Se usan para arrancar el servidor de la base de datos y para acceder al shell de la misma.

- **mongod**: Es el servicio principal de MongoDB. Maneja los accesos a los datos, las peticiones de datos, y ejecuta tareas de mantenimiento en background. Su fichero de configuración es “**mongod.conf**”.
- **mongo**: Es la shell interactiva de MongoDB. Aporta un entorno funcional completo para ser usado con la base de datos.
- **mongos**: Es un servicio propio del modo de despliegue Shard. Su función es la de enrutar las peticiones de la capa de aplicación y determinar la ubicación de los datos en los diferentes shards del despliegue.
- **mongodump**: Es una utilidad para crear un export binario del contenido de una base de datos. Podemos considerar MongoDB como una herramienta más para realizar copias de seguridad. Podemos usar esta herramienta contra “mongod” o “mongos” teniendo en cuenta que “mongod” podrá estar arrancado o parado.
- **mongorestore**: Se utiliza para restaurar los respaldos realizados con mongodump.
- **mongooplog**: Es una herramienta que permite hacer “polling” del oplog de un servidor remoto, y aplicarlo sobre el servidor local. Esta utilidad la podemos usar para realizar cierta clase de migraciones en tiempo real, donde se requiere que el servidor fuente se mantenga Online y en funcionamiento

## 3. HERRAMIENTAS:

### a. Herramientas exportación / importación

- **bsondump**: Convierte ficheros BSON a algún formato legible por humanos, incluido a JSON. Se trata de una herramienta de análisis, en ningún caso debe ser utilizada para otro tipo de actividades.
- **mongoexport**: Utilidad que permite exportar los datos de una instancia de MongoDB en formato JSON o CSV. En conjunción con mongoimport son útiles para hacer backup de una parte bien definida de los datos de la BBDD MongoDB o para casos concretos de inserción de datos.
- **mongoimport**: Utilidad que permite importar los datos de una instancia de MongoDB desde ficheros con formato JSON o CSV.
- **mongofiles**: Utilidad que permite manejar ficheros en una instancia de MongoDB con objetos GridFS, desde la línea de comandos. En Replica Set sólo podrá leer desde el primario.





#### b. Herramientas análisis

- **mongoperf**: Utilidad para comprobar el performance I/O de forma independiente a MongoDB.
- **mongostat**: Utilidad que proporciona una rápida visión del estado actual de los servicios mongod y mongos. Es similar a la utilidad vmstat.
- **mongotop**: Proporciona un método para trazar el tiempo que una instancia de MongoDB emplea en las operaciones de Lectura/Escritura de datos. Proporciona estadísticas a nivel de colección, por defecto, cada segundo.

#### 4. SHELL DE MONGODB:

Para interactuar con esta base de datos utilizamos la shell mongo

- Es una shell interactiva en JavaScript.
  - Permite ejecutar scripts escritos en JavaScript para manipulación de datos, ejecución de comandos en la base de datos, aplicación de índices, etc.
  - La shell puede ser utilizada tanto para la visualización de datos, como para la administración de la base de datos, sea Standalone, Replica Set o Sharding.
- Para trabajar con la shell, en primer lugar, nos conectamos a la instancia que hemos levantado anteriormente ejecutando el comando mongo

- `db.createCollection("holaMundo")`
- `db.holaMundo.insert({"Nombre" : "Ernesto", "Apellido" : "Perez", "Edad" : "45"})`
- `db.holaMundo.find()`

##### 4.1. COMANDOS:

Helper	Descripción
<code>help</code>	Muestra ayuda general.
<code>db.help()</code>	Muestra ayuda sobre los comandos de ejecutables sobre BBDD.
<code>db.&lt;collection&gt;.help()</code>	Muestra ayuda sobre comandos de ejecutables sobre colecciones.
<code>Show dbs</code>	Muestra las BBDD del servidor.
<code>db</code>	Devuelve el nombre de la BBDD donde nos encontramos posicionados.
<code>show collections</code>	Muestra las colecciones contenidas en la BBDD donde estamos posicionados.
<code>use &lt;db&gt;</code>	Nos posicionamos sobre la BBDD db.
<code>show users</code>	Muestra los usuarios sobre la BBDD actual.
<code>load("&lt;ruta_script&gt;")</code>	Carga en la sesión actual el script contenido en la ruta ruta_script.
<code>it</code>	Itera el cursor sobre el que se haya hecho una query.

Existen varias herramientas gráficas que también nos permiten acceder a nuestra BBDD y administrarla de manera más intuitiva. **Robomongo**.

## 5. OPERACIONES CRUD:

Operación	Mongo	SQL
Create	Insert / save	INSERT
Read	Find / findOne	SELECT
Update	Update / findAndModify / save	UPDATE
Delete	Remove / drop	DELETE

### INSERTAR DATOS:

```
db.users.insert (  ← collection
  {
    name: "sue",  ← field: value
    age: 26,      ← field: value
    status: "A"   ← field: value
  }              } document
)
```

### CONSULTAR DATOS:

```
db.users.find(
  { age: { $gt: 18 } },  ← collection
  { name: 1, address: 1 } ← query criteria
).limit(5)              ← projection
                        ← cursor modifier
```

### ACTUALIZAR DATOS:

```
db.users.update(
  { age: { $gt: 18 } },  ← collection
  { $set: { status: "A" } }, ← update criteria
  { multi: true }        ← update action
)                        ← update option
```

### BORRAR DATOS:

```
db.users.remove(
  { status: "D" }  ← collection
)                ← remove criteria
```