

# DAM - Accés a dades

## Tema 3 - Bases de dades No-SQL

Roberto Sanz Requena

[rsanz@florida-uni.es](mailto:rsanz@florida-uni.es)

# Índex

1. Context
2. MongoDB
3. Firebase



# 1. Context

- Problemes de les BBDD relacionals en l'entorn Web 2.0: escalabilitat, rendiment, volum de transaccions, necessitat de distribució de recursos,...
- No-SQL: orientació a “objectes” en compte d'entrades en múltiples taules relacionades gestionat per un SGBD (DBMS).
- Magatzem estructurat: documents, clau/valor, grafo o columnes
- No-SQL és en realitat Not-Only-SQL, ja que pot admetre SQL
- Exemples: XML, JSON

<https://es.wikipedia.org/wiki/NoSQL>

# 1. Context

```
<employees>
  <employee>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName>
    <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName>
    <lastName>Jones</lastName>
  </employee>
</employees>
```

```
{ "employees": [
  { "firstName": "John", "lastName": "Doe" },
  { "firstName": "Anna", "lastName": "Smith" },
  { "firstName": "Peter", "lastName": "Jones" }
]}
```

**XML** (eXtensible Markup Language)

VS

**JSON** (Javascript Object Notation)

[https://www.w3schools.com/js/js\\_json\\_xml.asp](https://www.w3schools.com/js/js_json_xml.asp)

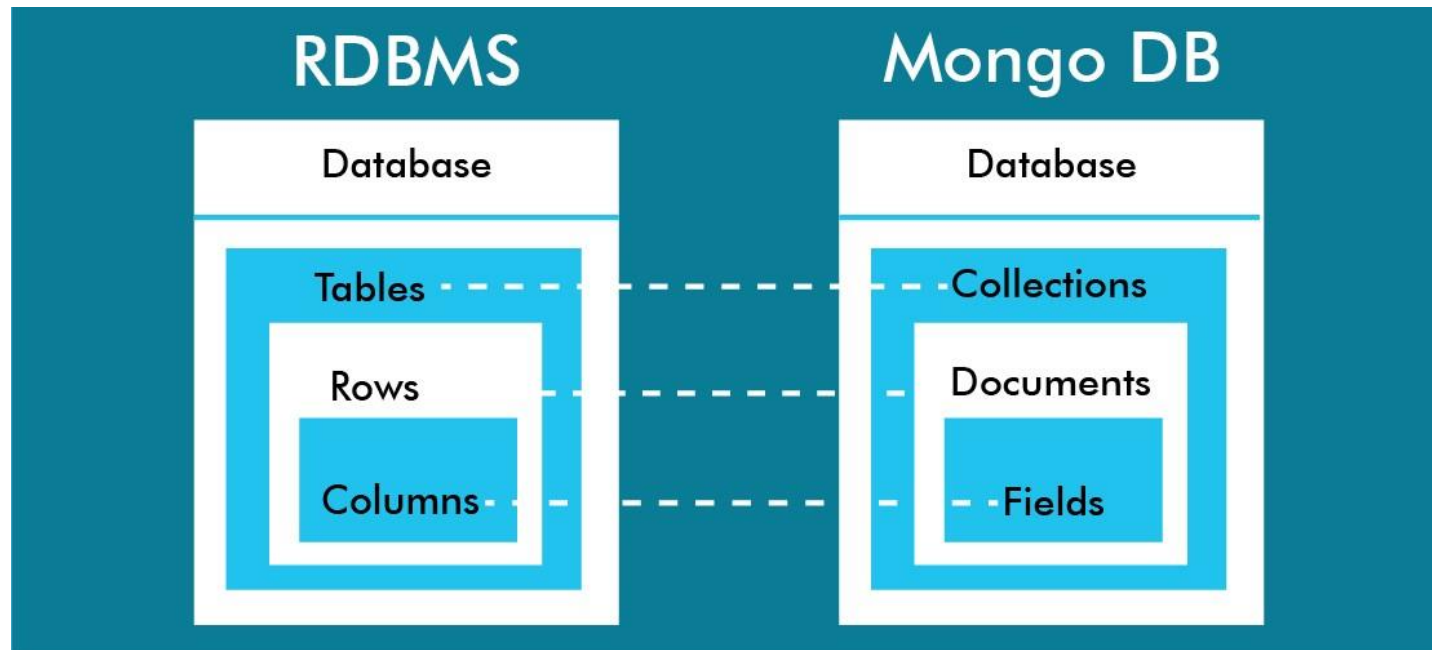
# 1. Context

- Aproximacions basades en JSON
  - MongoDB
    - Base de dades consolidada
    - Ús general
    - Projectes grans
    - Magatzem local i en cloud
    - Codi obert
  - Firebase
    - Emmagatzema i sincronitza dades en temps real
    - Magatzem en cloud
    - Orientada a facilitar la integració en apps mòbils
    - Comercial a partir de certa grandària

<https://www.mongodb.com/es/firebase-vs-mongodb>  
<https://qastack.mx/programming/29223835/mongodb-vs-firebase>

## 2. MongoDB

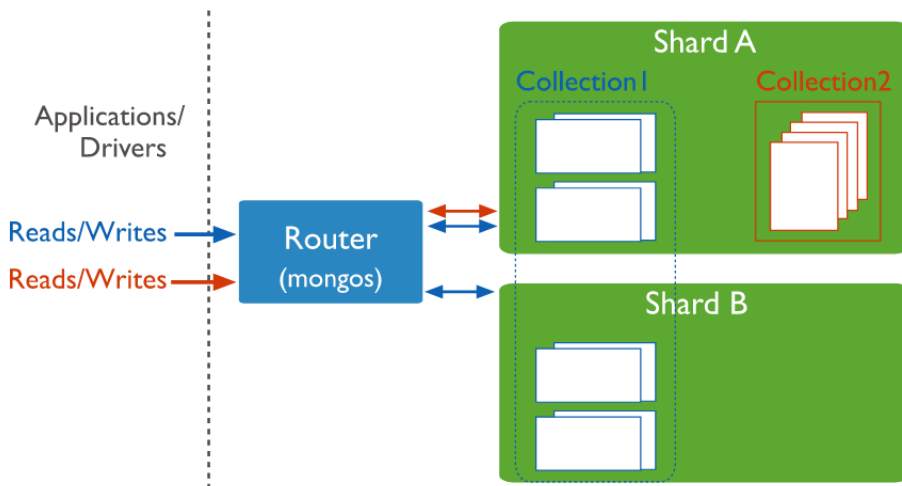
- Mongo DB vs Base de dades relacional



<https://blockchainsimplified.com/blog/mongodb-introduction/>

## 2. MongoDB

- Registres: documents (Binary JSON, BSON)
- Col·leccions: agrupacions de documents que tenen una clau única (`_id`)
- Magatzem distribuït de col·leccions: *shards* i *replica sets*



### **Sharding** automàtic (particionar):

- gestionar la càrrega dels servidors (A i B)
- Escalabilitat horitzontal
- sharded (distribuir la col·lecció entre màquines distintes, A i B) o unsharded (només en una màquina, A)

### **Replica sets:**

- redundància, accessibilitat, tolerància a errors

<https://docs.mongodb.com/manual/sharding/>

## 2. MongoDB



### Instal·lació MongoDB

MongoDB Community Server:

<https://www.mongodb.com/try/download/community>

Instal·lar també MongoDB Compass.



## 2. MongoDB

### MongoDB Compass

Ferramenta GUI per a gestionar MongoDB (equivalent a phpMyAdmin)

- ✓ Crear una BDD
- ✓ Connectar-se a una BDD
- ✓ Examinar una col·lecció
- ✓ Buscar en la col·lecció
- ✓ Insertar/editar/esborrar document
- ✓ Exportar/importar BDD com .json
- ✓ Agregació (operacions)
- ✓ Schema / Explain plan (estadístiques)
- ✓ Indexes (personalitzar índexos)
- ✓ Validació (regles i restriccions)

<https://www.mongodb.com/try/download/compass>

## 2. MongoDB

### Comparativa MySQL vs MongoDB

<https://docs.mongodb.com/manual/reference/sql-comparison/>

## 2. MongoDB

Operativa:

- Instal·lació de MongoDB i MongoDB Compass
- Descarregar i importar el .jar de MongoDB en CLASSPATH o definir el projecte com a Maven (dependències en pom.xml)
- Connexió a la BDD MongoDB
- Punter a la col·lecció
- Operacions CRUD (veure filtres i parser JSON)
- Tancar connexió

<https://examples.javacodegeeks.com/software-development/mongodb/java-mongodb-example/>  
<https://mongodb.github.io/mongo-java-driver/3.4/driver/getting-started/quick-start/>

## 2. MongoDB

Connexió

```
MongoClient mongoClient = new MongoClient("localhost", 27017);
MongoDatabase database = mongoClient.getDatabase("DAM_MongoDB");
MongoCollection<Document> coleccion = database.getCollection("canciones");
//CRUD operations
mongoClient.close();
```

Crear

```
Document doc = new Document();
doc.append("campo1", valor1);
doc.append("campo2", valor2);
coleccion.insertOne(doc);

...

coleccion.insertMany(listaDocs);
```

Query (filtres)

```
Bson query = eq("anyo", 1991);
```

Llegir

```
MongoCursor<Document> cursor = coleccion.find().iterator();
while (cursor.hasNext()) {
    System.out.println(cursor.next().toJson());
}
```

Parser JSON

```
JSONObject obj = new JSONObject(cursor.next().toJson());
System.out.println(obj.getString("titulo"));
```

**DAM - Accés a dades**

## 2. MongoDB

### Actualitzar

```
coleccion.updateOne(eq("formato", "WAV"), new Document("$set",  
    new Document("formato", "OGG")));  
  
coleccion.updateMany(eq("formato", "WAV"), new Document("$set",  
    new Document("formato", "OGG")));
```

### Eliminar

```
coleccion.deleteOne(eq("formato", "OGG"));  
  
coleccion.deleteMany(eq("formato", "OGG"));  
  
coleccion.drop();
```

<https://mongodb.github.io/mongo-java-driver/3.4/driver/getting-started/quick-start/>

## 2. MongoDB

### JARs (Importar en CLASSPATH) / Dependències (Maven):

- MongoDB: <https://mvnrepository.com/artifact/org.mongodb/mongo-java-driver/3.12.10>
- Parser JSON: <https://mvnrepository.com/artifact/org.json/json/20210307>

### Filtres (documentació i exemples):

<https://mongodb.github.io/mongo-java-driver/3.4/javadoc/?com/mongodb/client/model/Filters.html>  
<https://docs.mongodb.com/drivers/java/sync/v4.3/fundamentals/builders/filters/>

<https://examples.javacodegeeks.com/software-development/mongodb/java-mongodb-example/>  
<https://mongodb.github.io/mongo-java-driver/3.4/driver/getting-started/quick-start/>

# Activitat Entregable 6 - MongoDB

Presentació de l'Activitat Entregable 6 (AE06\_T3\_1\_MongoDB)

## 2. MongoDB

### MongoDB Atlas

Database-as-a-Service (DBaaS) → Desplegament en cloud (AWS, Azure)

<https://www.mongodb.com/basics/mongodb-atlas-tutorial>

512 MB gratuïts

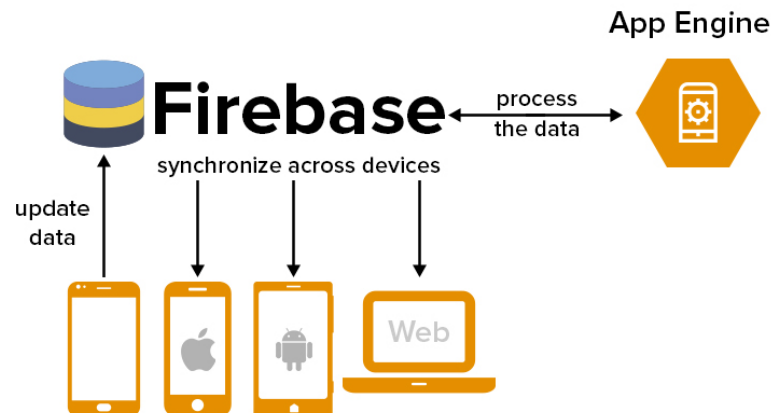


### 3. Firebase

SDK (Software Development Kit) o plataforma cloud per a desenvolupar aplicacions web i mòbils.

**Realtime Database / Cloud Firestore:** bases de dades en format JSON que es sincronitzen en temps real amb cada client connectat.

Persistència de les dades en el client en cas de perdre la connexió (sincronitzarà de nou quan es recupere).



<https://firebase.google.com>