

MongoDB

Una Base de Datos basada en
Documentos

DOCUMENT DATABASES

- La unidad básica de datos es el Documento. Comparable con un registro del modelo relacional, aunque con esquema flexible.
- Un conjunto de Documentos, es denominado Collection. Comparable con tabla del modelo relacional.
- Un conjunto de Collections es una Base de Datos de Documentos.

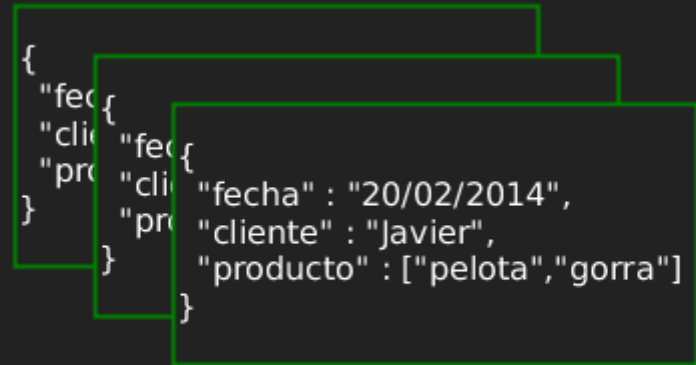
¿QUÉ ES UN DOCUMENTO?

```
{
  "_id" : "586fdc38fad477126124ba1b",
  "nombre": "José Garra Muñoz",
  "direccion" : {
    "calle" : "Juramento 1653",
    "ciudad" : "Capital Federal"
  },
  "telefono": ["123456", "7890123"],
  "fecha_nacimiento": "1978-01-06",
  "hijos" : [
    {
      "nombre" : "Javier",
      "fecha_nacimiento": "2010-01-06",
    },
    {
      "nombre" : "Josefina",
      "fecha_nacimiento": "2013-01-06",
    }
  ]
}
```

- El equivalente a un registro en una Base de Datos Relacional, es un documento en una Base de Datos basada en Documentos.
- Un documento es una estructura de datos compuesta por pares *atributo:valor*.
- Los documentos pueden contener otros documentos, arreglos y arreglos de documentos.
- En *MongoDB* un documento es un objeto JSON. Otras soportan XML.

¿QUÉ ES UNA COLLECTION?

- Los documentos se almacenan en collections.
- El equivalente a una tabla en una Base de Datos Relacional, es una collection en una Base de Datos basada en Documentos.
- A diferencia de una tabla en una Base de Datos Relacional, las collections no poseen un esquema. Una misma collection puede contener documentos con diferente esquema.



The diagram illustrates a MongoDB collection containing documents with varying schemas. It shows three nested green rectangles representing documents. The innermost rectangle contains a full document with fields: "fecha" (date), "cliente" (customer name), and "producto" (array of products). The middle rectangle shows a document with "fecha" and "cliente" fields, but the "producto" field is truncated. The outermost rectangle shows a document with only the "fecha" field visible. This visualizes how a single collection can hold documents with different structures.

```
{  
  "fecha": "20/02/2014",  
  "cliente": "Javier",  
  "producto": ["pelota", "gorra"]  
}
```

Collection

¿QUÉ ES UNA BD DE DOCUMENTOS?

Base De Datos Basada en Documentos

```
{
  "nombre": "Javier",
  "edad": "16",
  "estatura": "1.62",
  "hermanos": ["Jorge", "Julio"],
}
```

Collection 1

```
{
  "fecha": "20/02/2014",
  "cliente": "Javier",
  "producto": ["pelota", "gorra"],
}
```

Collection 2

```
{
  "producto": "Pelota",
  "previo": "100",
  "fecha_fab": "16/10/2015",
  "categoria": "deportes",
}
```

Collection N

MODELANDO DATOS EN UNA BD DE DOCUMENTOS

¿EMBEBER O NORMALIZAR?

MODELO EMBEBIDO

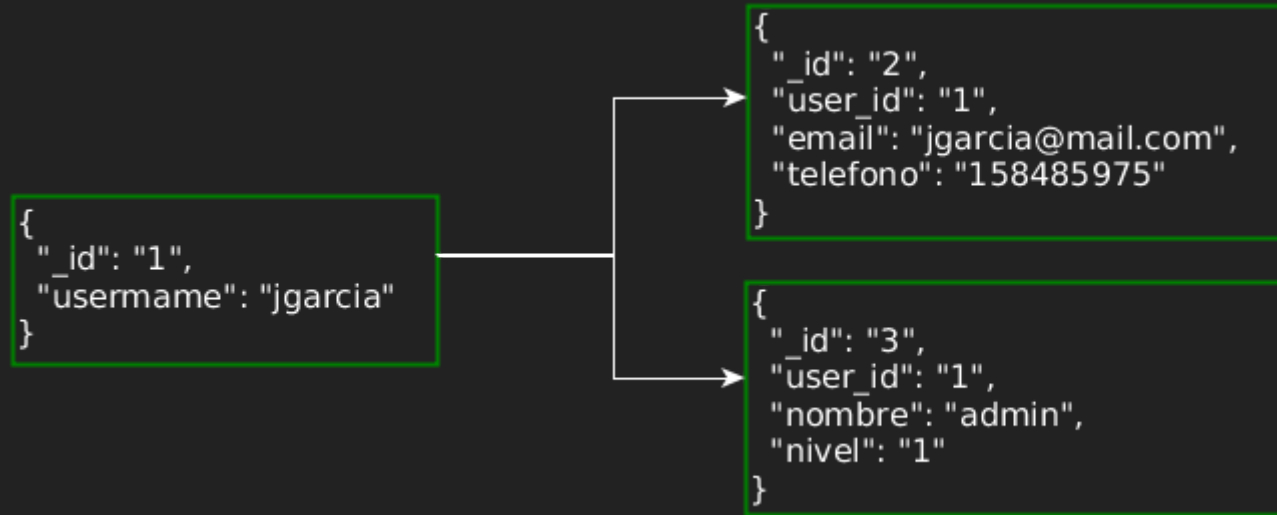
```
{
  "_id": "1",
  "username": "jgarcia",
  "contacto": {
    "email": "jgarcia@mail.com",
    "telefono": "158485975"
  },
  "rol": {
    "nombre": "admin",
    "nivel": "1"
  }
}
```

contacto
sub documento embebido

rol
sub documento embebido

- Modelo denormalizado. Optimizado para responder mejor ciertas consultas.
- Si en la aplicación siempre o en la mayoría de los casos de uso, cuando traigo un *Usuario*, necesito el *contacto* y su *rol*, entonces me conviene embeber.

MODELO NORMALIZADO



- Cuando se embebe se duplica información. Si no se obtienen ventajas de performance, entonces utilizamos referencias.
- Tiene mas flexibilidad, pero la aplicación debe realizar mas idas a la BD para traer las relaciones (si es que el caso de uso lo requiere).
- Si vas a necesitar utilizar muchas relaciones, utiliza mejor una BD relacional.

¿TRANSACCIONES?

- MongoDB es una Base de Datos NoSQL, distribuída, por lo que se rige por el teorema CAP.
- Las modificaciones son atómicas a nivel de Documento (incluyendo sub-documentos embebidos). O sea, si deseo modificar dos o mas propiedades de un documento, se modifican todas o ninguna.
- Una operación de modificación sobre múltiples documentos, no es atómica.

USOS ADECUADOS

- Logueo de Eventos: Las aplicaciones tienen diferentes necesidades de logueo y la estructura flexible de MongoDB permite agregar nuevos datos de diferentes aplicaciones sin cambiar el esquema.
- Web Analytics: Páginas Visitadas.
- Content Management Systems, Blogging Systems.

MONGODB

SHELL (COMANDOS BÁSICOS)

MONGODB - COMANDOS DEL SHELL

```
//Muestra todas las bases de datos de la instancia de MongoDB
> show dbs
//Cambia a utilizar la BD db_x, o la crea sino existe
> use db_x
//Muestra las collections existentes en la BD db_x
> show collections
//Inserta un documento en la collection movie, la crea sino existe
> db.movie.insert({"name":"tutorials point"})
//Lista todos los documentos de la collection movie
> db.movie.find()
//Lista los documentos de paginas donde el autor es Yo Mismo
> db.paginas.find({"autor":"Yo Mismo"})
```

MONGODB

JAVA DRIVER

MONGODB - JAVA DRIVER

SESSION SKELETON

```
try (MongoClient mongoClient = new MongoClient()) {  
    //Obtengo la instancia de Base de Datos  
    MongoDB database = mongoClient.getDatabase("unaBD");  
  
    //Obtengo la Colección con la que deseo trabajar  
    MongoCollection collection = database.getCollection("unaColección");  
  
    //Aca trabajo con la collection...  
}
```

Usamos try.with resources para que se cierre la session al finalizar.

MONGODB - JAVA DRIVER

INSERTAR UN DOCUMENTO

```
try (MongoClient mongoClient = new MongoClient()) {  
    //Obtengo la BD y la Colección  
    MongoDB database = mongoClient.getDatabase("unaBD");  
    MongoCollection collection = database.getCollection("unaColección");  
  
    //Creo una instancia de org.bson.Document  
    Document document = new Document("atributo1", "valor1")  
        .append("atributo2", "valor2")  
        .append("atributo-arreglo",  
            Arrays.asList("valor1", "valor2"))  
        .append("atributo-fecha", new Date());  
        .append("atributo-documento",  
            new Document("at1", "val1"));  
  
    //Inserto el Documento en la collection  
    collection.insertOne(document);  
}
```

MONGODB - JAVA DRIVER

BUSCAR DOCUMENTOS

```
try (MongoClient mongoClient = new MongoClient()) {  
    //Obtengo la BD y la Colección  
    MongoDB database = mongoClient.getDatabase("unaBD");  
    MongoCollection collection = database.getCollection("unaColección");  
  
    //Imprimo en consola el documento con id = "abc1".  
    collection.find(Filters.eq("_id", new ObjectId("abc1"))  
        .forEach((Document d) -> System.out.println(d));  
  
    //Imprimo en consola los documentos donde "atributo1" = "valorX",  
    //ordenados por atributo-fecha.  
    collection.find(Filters.eq("atributo1", "valorX"))  
        .sort(Sorts.descending("atributo-fecha"))  
        .forEach((Document d) -> System.out.println(d));  
}
```


MONGODB - JAVA DRIVER

BUSCAR DOCUMENTOS

```
try (MongoClient mongoClient = new MongoClient()) {
    //Obtengo la BD y la Colección
    MongoDB database = mongoClient.getDatabase("unaBD");
    MongoCollection collection = database.getCollection("unaColección");

    //Genero JSON y lo Imprimo en consola para el documento con id = "abc1".
    FindIterable d = collection.find(Filters.eq("_id", new ObjectId("abc1")));
    System.out.println(StreamSupport.stream(d.spliterator(), false)
        .map(Document::toJson)
        .collect(Collectors.joining(", ", "[", "]")));

    //Genero JSON y lo Imprimo en consola para los documentos donde
    //"atributo1" = "valorX", ordenados por atributo-fecha.
    FindIterable ds = collection.find(Filters.eq("atributo1", "valorX"))
        .sort(Sorts.descending("atributo-fecha"));
    System.out.println(StreamSupport.stream(ds.spliterator(), false)
        .map(Document::toJson)
        .collect(Collectors.joining(", ", "[", "]")));
}
```

MONGODB - DOCS

- Instalación de MongoDB
- Shell Docs
- Java Driver
- Java Driver Tutorial

