

Proyecto **FPMAD***digital*

Recursos digitales y multimedia para Formación Profesional



**Comunidad
de Madrid**

Dirección General
de Educación Secundaria,
Formación Profesional
y Régimen Especial

CONSEJERÍA DE EDUCACIÓN,
UNIVERSIDADES, CIENCIA
Y PORTAVOCÍA



Unión Europea

Fondo Social Europeo

“El FSE invierte en tu futuro”

**Financiado como parte de la respuesta
de la Unión a la pandemia de COVID-19**

CFGS Desarrollo de Aplicaciones Multiplataforma

módulo profesional

0486 - Acceso a Datos

unidad didáctica

04 Uso de BBDD NoSQL

resultados de aprendizaje

04 Desarrolla aplicaciones que gestionan la información almacenada en bases de datos objeto relacionales y orientadas a objetos valorando sus características y utilizando los mecanismos de acceso incorporados.



Comunidad
de Madrid

Dirección General
de Educación Secundaria,
Formación Profesional
y Régimen Especial

CONSEJERÍA DE EDUCACIÓN,
UNIVERSIDADES, CIENCIA
Y PORTAVOCÍA



Unión Europea

Fondo Social Europeo

“El FSE invierte en tu futuro”

Financiado como parte de la respuesta
de la Unión a la pandemia de COVID-19

Resultados de aprendizaje y unidades didácticas

Resultados de aprendizaje y unidades didácticas

RESULTADOS DE APRENDIZAJE						UNIDAD DIDÁCTICA
1	2	3	4	5	6	
X						1.- Ficheros, colecciones y data frames
				X		2.- Manejo de XML
	X	X				3.- Procesamiento de BBDD Relacionales
			X			4.- Uso de BBDD NoSQL
		X	X		X	5.- Programación de componentes de acceso a datos

Unidades didácticas y materiales asociados

Unidades didácticas y materiales multimedia

RRAA						UDD	Material Multimedia
1	2	3	4	5	6		
X						1.- Ficheros, colecciones y data frames	1.1 Contenidos básicos 1.2 Ejemplos aplicados
				X		2.- Manejo de XML	2.1 Contenidos básicos 2.2 Ejemplos aplicados
	X	X				3.- Procesamiento de BBDD Relacionales	3.1 Contenidos básicos 3.2 Ejemplos aplicados
			X			4.- Uso de BBDD NoSQL	4.1 Contenidos básicos 4.2 Ejemplos aplicados
		X	X		X	5.- Programación de componentes de acceso a datos	5.1 Contenidos básicos 5.2 Ejemplos aplicados

Repositorios de materiales y prácticas

Repositorio de materiales y prácticas

Todos los proyectos mostrados, así como otros materiales utilizados en las unidades didácticas los podrás encontrar completos en:

<https://github.com/joseluisgs/FP-NextGen-AccesoDatos>

Cualquier error o propuestas de mejora se publicarán en el repositorio indicado.
Gracias por tu colaboración.

Contenidos

1. **JSON**
2. **Bases de Datos NoSQL**
3. **MongoDB**
4. **JPA en NoSQL**



JSON

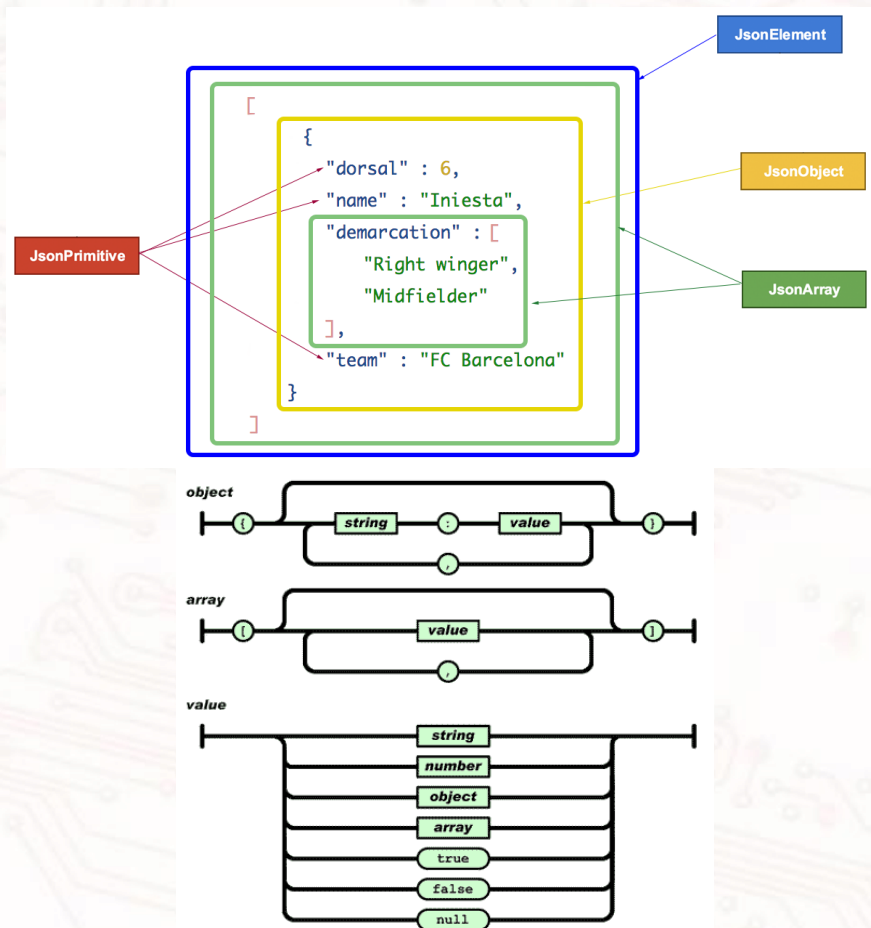


JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos.

JSON es un formato de texto que es completamente independiente del lenguaje. Un elemento JSON (JsonElement) está constituido por:

- Una colección de pares de nombre/valor: en varios lenguajes esto es conocido como un objeto o registro (JsonObject), cada elemento par/valor se le conoce como elemento primitivo (JsonPrimitive)
- Una lista de valores o elementos. En la mayoría de los lenguajes, esto se implementa como arrays o listas (JsonArray)



Bases de Datos NoSQL

Bases de Datos NoSQL

Se caracterizan por **no seguir obligatoriamente el modelo relacional** (Not Only SQL)

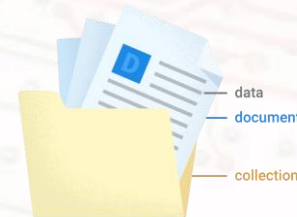
Intentan resolver problemas de almacenamiento masivo, alto desempeño, procesamiento masivo de transacciones (sitios con alto tránsito) y, en términos generales, ser alternativas NoSQL a problemas de persistencia y almacenamiento masivo (voluminoso) de información para las organizaciones.

No utilizan SQL como lenguaje de consultas. La mayoría de las bases de datos NoSQL evitan utilizar este tipo de lenguaje o lo utilizan como un lenguaje de apoyo. Por poner algunos ejemplos: Cassandra utiliza el lenguaje CQL, MongoDB utiliza MQL basado en JavaScript o BigTable hace uso de GQL.

No utilizan estructuras fijas como tablas para el almacenamiento de los datos. Permiten hacer uso de otros tipos de modelos de almacenamiento de información como sistemas de **clave—valor, documentos, objetos o grafos**.

No suelen permitir operaciones JOIN ya que al disponer de un volumen de datos tan extremadamente grande suele resultar deseable evitar los JOIN. Esto se debe a que, cuando la operación no es la búsqueda de una clave, la sobrecarga puede llegar a ser muy costosa. Las soluciones más directas consisten en desnormalizar los datos, o bien, realizar el JOIN mediante software en la capa de aplicación.

Bases de Datos NoSQL



A la hora de **diseñarlas**, tendremos en cuenta desde nuestro diagrama de clases:

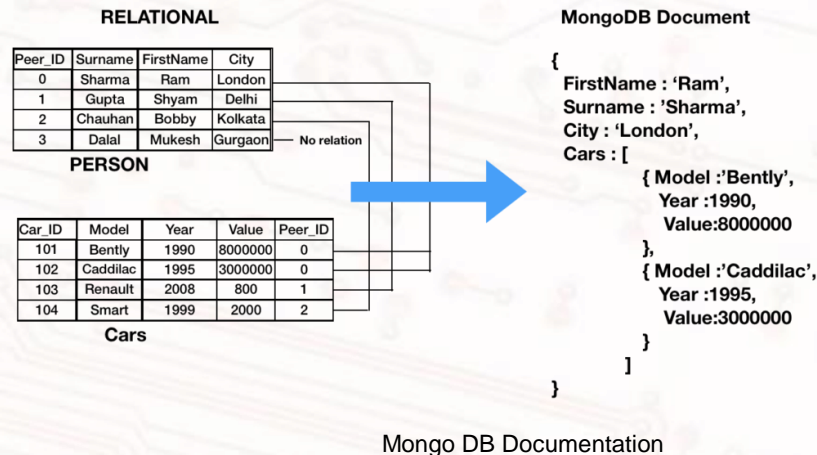
Relaciones 1-1 y 1-M. Puedes usar **referencias propagando objetos o claves y haciendo joins (es una operación costosa)**. También puedes usar **conjunto de documentos y objetos embebidos** pudiendo o no elegir entre replicar elementos repetidos dependiendo si primas la lectura o lectura/escritura. Tiene la ventaja de que **la información es accesible con una única lectura de la base de datos, pero se pierde escalabilidad**.

Relaciones basadas en árboles/grafos con referencia al padre. Cada nodo se almacena con referencia al padre.

Relaciones basadas en árboles/grafos con referencia a los hijos. Por cada padre habrá una lista a los hijos que tiene.

Relaciones basadas en árboles/grafos con referencia a los ancestros. Por cada nodo habrá una relación a los ancestros y padre.

Relaciones basadas en árboles/grafos con conjuntos embebidos. Se almacenan subárboles o subgrafos.



The background of the slide features a faint, repeating pattern of a circuit board. It includes various lines, pads, and components in a light gray and reddish-pink color scheme, creating a technical and digital aesthetic.

MongoDB

A large, solid red rectangular block occupies the bottom half of the slide, positioned below the MongoDB text. It is a uniform, vibrant red color with no internal details or patterns.

MongoDB

MongoDB (del inglés humongous, "enorme") es un sistema de base de datos NoSQL, orientado a documentos y de código abierto que nació en 2007.

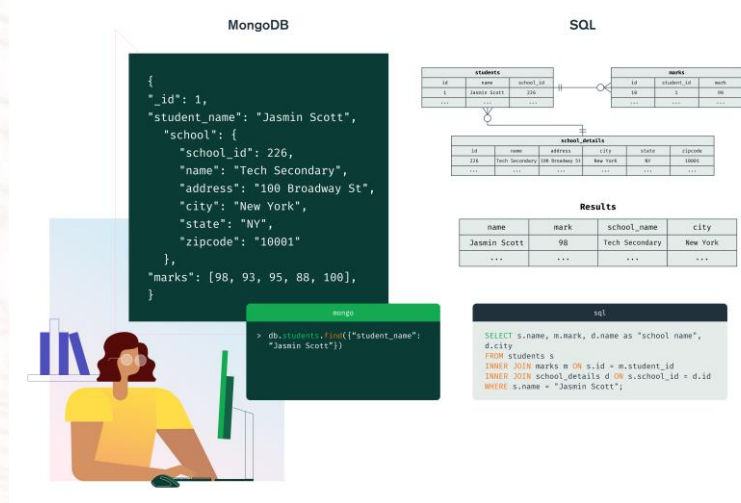
MongoDB guarda estructuras de datos **BSON** (una especificación similar a JSON) con un esquema dinámico.

MongoDB soporta la búsqueda por campos, consultas de rangos y expresiones regulares. Las consultas pueden devolver un campo específico del documento pero también puede ser una función definida por el usuario para su mejor ocupación usando **MQL** y **JavaScript** como lenguaje de referencia.

Permite trabajar con **agregados**, se basa en someter una colección a un conjunto de operaciones o etapas, estas etapas irán convirtiendo y transformando el conjunto de documentos pertenecientes a la colección, hasta obtener un conjunto de documentos con el resultado deseado.

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

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



Mongo DB Documentation

JPA en NoSQL

JPA en NoSQL

Podemos usar JPA en NoSQL con:

- **MongoDB** usando Spring Data para Mongo o Hibernate OGM. Para muchas consultas es mejor usar **Native Query con las consultas MongoDB propias de agregados que JPQL**
- **ObjectDB** base de datos orientada a objetos. **Por lo que nuestro diseño OO es totalmente compatible**
- **Neo4j** base de datos basada en **grafos**.

Todo lo que sabemos de JPA nos sirve, así como nuevas anotaciones para crear estructuras embebidas o jerarquizadas, como hemos visto en puntos anteriores.

JPA en NoSQL

@Embeddable: indicamos que la clase puede ser «integrable» dentro de una entidad. No es necesario clave primaria.

@Embedded: indicamos que el campo o la propiedad de una entidad es una instancia de una clase que puede ser integrable. Es decir, para que funcione, el campo que hayamos anotado como `@Embedded`, debe corresponderse con una clase que tenga la anotación `@Embeddable`.

@AttributeOverrides: podemos sobrescribir o redefinir el mapeo de campos o propiedades de tipo básico. Esta anotación recibe un array de `@AttributeOverride`.

@AttributeOverride: podemos redefinir el mapeo de tipos básicos. Su uso es `@AttributeOverride(name=atributoDeLaClaseEmbebida, column = @Column(name=nombreColumnaEnLaTabla))`.

@AssociationOverrides: podemos sobrescribir o redefinir el mapeo de campos o propiedades complejos. Esta anotación recibe un array de `@AssociationOverride`.

@AssociationOverride: podemos redefinir el mapeo de tipos básicos. Su uso es `@AssociationOverride(name=atributoDeLaClaseEmbebida, joinColumns=@JoinColumn(name=columna_id))`, donde el primer parámetro es el nombre del atributo de la clase embebida que estamos redefiniendo, y el segundo parámetro es un `@JoinColumn`, en donde podemos especificar el campo al que hacemos el join.

JPA en NoSQL

Si queremos embeber una colección tenemos dos opciones:

Embeber una colección completa de objetos:

@ElementCollection

```
private List<String> authors = new ArrayList<>();
```

Embeber una colección de identificadores o índices a cada elemento embebido. Siempre que se marque como @Indexed

@IndexedEmbedded

```
private List<Author> authors = new ArrayList<>();
```

Esta última opción también es aplicable para distintas relaciones: 1-1, 1-M, N-1, M-N (@OneToOne, @ManyToOne, @OneToMany, @ManyToMany)

JPA en NoSQL

Ademas en SpringData para MongoDB tenemos:

@DBRef para almacenar referencias por ejemplo en el contenido de una coleccion de manera automática.

@DocumentReference para enlazar a referencias de manera manual, debe ir acompañada en el otro lado de la relación con la anotación **@DocumentReference(lazy=true)**.

Un ejemplo lo podemos ver en:

```
class Libro {  
    // ...  
    @DocumentReference(lazy=true)  
    private Editor editor;  
}
```

```
class Editor {  
    // ...  
    @DocumentReference  
    List<Libro> libros;  
}
```

<https://spring.io/blog/2021/11/29/spring-data-mongodb-relation-modelling>

Conclusiones



¡Vamos con la práctica!

"Lo realmente necesario es saberlo todo sobre los cambios en la información. Nadie quiere o necesita que le recuerden 16 horas al día que tiene sus zapatos puestos"

- David Hubel



**Comunidad
de Madrid**

Dirección General
de Educación Secundaria,
Formación Profesional
y Régimen Especial

CONSEJERÍA DE EDUCACIÓN,
UNIVERSIDADES, CIENCIA
Y PORTAVOCÍA



Unión Europea

Fondo Social Europeo

“El FSE invierte en tu futuro”

**Financiado como parte de la respuesta
de la Unión a la pandemia de COVID-19**