

# Tarea UD2: MongoDB



Juan Manuel García Moyano  
IABD 24/25  
Informática y comunicaciones

## Índice

Caso práctico.....	3
Parte 1.....	4
Parte 2.....	6
Práctica 1. MongoDB Atlas. Creación de un Clúster.....	6
Práctica 2. Conexión al Clúster de MongoDB Atlas y lenguaje de consultas de MongoDB.....	8

## Caso práctico

Andrés estudió el Grado Superior “Desarrollo de Aplicaciones Web” y desde hace años trabaja en el equipo de desarrollo de la empresa Aseguradora Zaidín-Vergeles que tiene una alta implantación en todo el territorio español.

Hasta el momento su trabajo ha consistido en participar en proyectos de desarrollo de aplicaciones web sobre un RDBMS de Oracle.

A principios de año asistió a una presentación del *Product owner* de la empresa en la que se anunciaron nuevos cambios para poder gestionar el crecimiento acelerado del volumen de datos que se viene produciendo en los últimos años.

A muy largo plazo se plantea una migración del core tradicional a una nueva arquitectura basada en microservicios, eventos y NoSQL, aunque esta migración es muy compleja y se requerirán unos 3 años para completarla. Durante este periodo ambos sistemas tendrán que convivir.

A corto plazo, y dada la necesidad de responder al desafío del volumen de datos generados, se adoptará una solución intermedia desarrollando las aplicaciones sobre una base de datos NoSQL. Esta base de datos se nutrirá de los datos necesarios del sistema relacional y los nuevos proyectos de desarrollo se realizarán y se pondrán en producción sobre esta nueva base de datos. El nuevo enfoque de la empresa será Data Driven.

La empresa ha elegido MongoDB Atlas priorizando SaaS sobre PaaS y sobre IaaS. Esto permitirá a los equipos centrarse en la programación abstrayéndose de la tecnología subyacente y de la administración de esta. También se ha elegido por ser una tecnología Open Source con un buen soporte empresarial.

A Andrés le gustaría participar en este nuevo proceso por lo que, tras la presentación, empezó a investigar sobre la solución MongoDB Atlas y está haciendo también unos cursos del lenguaje de consultas de MongoDB.

¿Qué te pedimos que hagas?

Genera un documento PDF que responda a las preguntas y prácticas que se te plantean a continuación.

## Parte 1.

Responde a las siguientes preguntas. Para tus respuestas apóyate en la documentación de MongoDB, [MongoDB Manual](#).

1.1 ¿Tiene MongoDB un esquema flexible? Si la respuesta es afirmativa, explica qué significa y pon un ejemplo.

Si, MongoDB no necesita que sus documentos sigan una estructura fija para los datos almacenados en sus documentos. De esta forma permite que los documentos de MongoDB tengan una estructura sin necesidad de cumplir con un esquema fijo. Esto implica las siguientes características:

- Variabilidad de estructura: una misma colección puede tener diferentes tipos de datos.
- Evolución del esquema: el diseño de los datos pueden ir variando fácilmente sin afectar a la evolución de estos en un futuro.
- Facilidad almacenar datos semiestructurados y no estructurados.

Para comprobar la facilidad que nos proporciona MongoDB, supongamos que tenemos un colección llamada clientes, en la que almacenamos información de estos (documentos). Este ejemplo se realiza en formato JSON:

```
{  
  "nombre": "Pepe",  
  "email": "pepigrillo@gmail.com"  
}
```

```
{  
  "nombre" : "Juan",  
  "teléfono" : "699999999"  
}
```

Como se puede observar en el ejemplo, los documentos no tienen la necesidad de compartir obligatoriamente los mismo tipos de datos. Ambos tienen el nombre pero uno tiene el email y otro el teléfono y no te obliga a que ambos tenga teléfono y email.

1.2 ¿MongoDB soporta escalado vertical u horizontal? Explica como lo consigue y con qué componentes. Detállalo.

Soporta ambos, pero principalmente está basado en el escalado horizontal. Esto lo consigue mediante la fragmentación (sharding). Divide los datos entre los diferentes servidores (nodos) y un shard es una réplica de los datos en una parte del clúster. De esta forma conseguimos el escalado horizontal porque los datos se van replicando entre los diferentes servidores y una vez que haga falta más almacenamiento lo único que tenemos que hacer es añadir más servidores, sin la necesidad de tener que parar el servidor.

1.3 ¿La estrategia de replicación en MongoDB es maestro-esclavo o peer-to-peer? Explícalo siguiendo la documentación oficial.

Es maestro-esclavo pero a un nivel superior, gracias al replica set. El nodo primario recibe todas las operaciones de escritura. Un conjunto de réplicas puede tener solo un nodo primario capaz de confirmar escrituras. Sin embargo, en ciertas circunstancias, otra instancia de mongodb puede transitoriamente creer que también es primaria. El nodo primario registra todos los cambios en sus conjuntos de datos en su registro de operaciones, es decir, el oplog.

Los nodos secundarios replican el registro de operaciones (oplog) del nodo primario y aplican las operaciones a sus propios conjuntos de datos, de manera que los conjuntos de datos de los secundarios reflejan el conjunto de datos primario. Si el nodo primario no está disponible, un nodo secundario elegible llevará a cabo una elección para convertirse en el nuevo primario.

En algunas circunstancias (como cuando tienes un nodo primario y uno secundario, pero las restricciones de costos impiden agregar otro nodo secundario), puedes optar por añadir una instancia de mongodb a un conjunto de réplicas como un árbitro. Un árbitro participa en las elecciones, pero no almacena datos (es decir, no proporciona redundancia de datos).

Un árbitro siempre será un árbitro, mientras que un nodo primario puede renunciar y convertirse en un nodo secundario, y un nodo secundario puede convertirse en el primario durante una elección.

1.4 Durante el proceso de creación de un clúster podemos seleccionar como proveedor AWS, Google Cloud o Azure. Detalla qué repercusión o qué influencia tiene esto de cara a una puesta en producción futura y por cual te decantarías de esas tres opciones. Justifica tu elección.

Para elegir un proveedor u otro, principalmente debemos fijarnos en los factores clave que tienen en común pero cada factor cambia según el proveedor. Una vez elegido, nos va a repercutir según los siguientes factores:

- Rendimiento.
- Precios.
- Escalabilidad.
- Integración del ecosistema.
- Conjunto de características.

Al elegir uno en concreto nos podría influenciar en un futuro principalmente en el coste de sacar los datos. Por ello, se debe de tener especial cuidado en la selección del proveedor, ya que, el cambiar los datos a otro proveedor o pasarlo a una solución local nos va a generar unos costos bastantes altos. Por supuesto, los costes dependerán de la cantidad de datos.

Me decantaría por AWS. AWS normalmente es la opción más robusta y flexible para la mayoría de las aplicaciones. Además, lo he elegido por su escalabilidad y variedad de servicios que se pueden integrar para mejorar el rendimiento y la fiabilidad de la infraestructura, los cuales estos servicios vamos a poder unir con nuestra base de datos en MongoDB.

## **Parte 2.**

### **Práctica 1. MongoDB Atlas. Creación de un Clúster.**

Antes de analizar la práctica lee la página principal de MongoDB Atlas para hacerte una idea de las características y el potencial de la herramienta que vamos a probar. Documenta los pasos dados y las distintas características seleccionadas incluyendo imágenes/capturas de todo proceso.

En esta práctica debes crear un **clúster de prueba** con MongoDB Atlas. En la web [MongoDB Atlas](#) pulsa el botón *Try Free*. A continuación puedes registrarte con la cuenta corporativa del instituto.

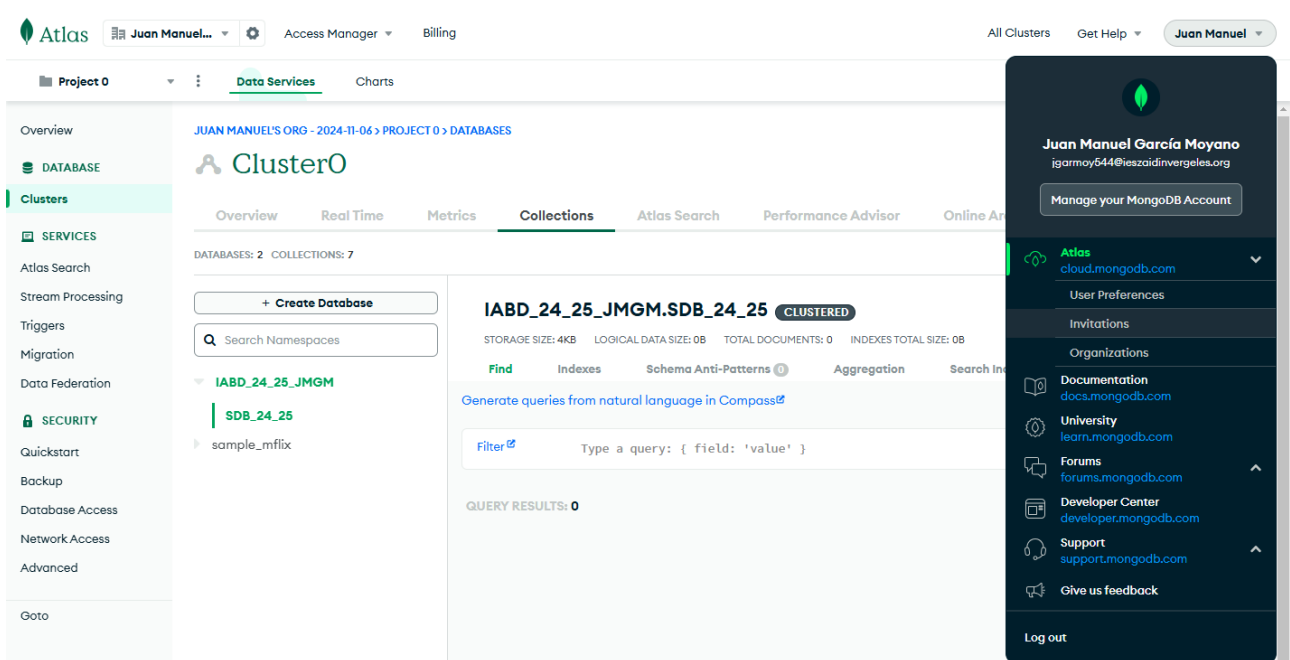
Para crear el cluster sigue las instrucciones de la documentación *Get Started with Atlas*, [MongoDB Atlas](#) o las instrucciones del siguiente [vídeo](#). Puedes seleccionar el modo de configuración avanzando o no, según desees (queda abierto a tu elección).

Ten en cuenta que para conectarte al clúster desde la máquina local hay diferentes estrategias. Nosotros utilizaremos **Shell** seleccionando el botón *I don't have the MongoDB Shell installed* y descargando mongosh en nuestro equipo. Previamente deberás crear un usuario. Lee atentamente todos los pasos de las distintas pantallas de la creación del Clúster. Durante el proceso, copia el string de conexión y habilitamos la carga de la base de datos de prueba.

Una vez desplegado el clúster y cargado el dataset de ejemplos, pulsa *Browse Collections*. En esta nueva pantalla podrás observar la lista de bases de datos de ejemplo y realizar operaciones de inserción, búsqueda o agregaciones en ellas, así como crear nuevas bases de datos y colecciones.

Ahora, crea una base de datos denominada **IABD\_24\_25\_<iniciales nombre, apellido1, apellido2>** y una colección denominada **SDB\_24\_25**. Si quieres saber más sobre como operar desde la interfaz de usuario de Atlas visita el siguiente enlace [Interactuar con sus datos](#).

Como resultado de esta práctica debes añadir al documento una captura de pantalla de esta última pantalla en la que se vea claramente tu nombre y tu correo (desplegando el usuario en la esquina superior derecha) y la lista de las bases de datos en la que observe claramente la base de datos y la colección. Se han de documentar todos los pasos, tanto los establecidos en esta explicación como aquellos que parezcan triviales.



## Práctica 2. Conexión al Clúster de MongoDB Atlas y lenguaje de consultas de MongoDB.

En la práctica 1 hemos instalado mongosh en nuestro equipo. Si todo ha ido bien bastará ejecutar desde la consola la cadena de conexión. Puedes volver a obtener el string de conexión desde el menú Cluster > Cmd Line Tools.

Ahora abrimos un terminal y pegamos la string de conexión copiada en la práctica anterior.

Para la búsqueda de las operaciones que tienes que realizar utiliza, en primera instancia, el [manual](#) de MongoDB.

La práctica consiste en realizar una serie de consultas, desde la shell de comandos, sobre la base de datos de ejemplo sample\_mflix y su colección movies. Deberás crear esta base de datos si no lo has hecho antes.



Algunos de los comando más utilizados son:

- Mostrar bases de datos: **show dbs**
- Mostrar base de datos en uso: **db**
- Cambiar la base de datos: **use sample\_mflix**
- Mostrar colecciones: **show collections**
- Elimina la base de datos actual: **db.dropDatabase()**
- Muestra los comandos disponibles: **db.help()**
- Muestra la versión actual del servidor: **db.version()**

En el documento PDF de la tarea debes incluir las siguientes preguntas. En la respuesta incluye el **comando ejecutado, explícalo e incluye el resultado** de éste capturando la imagen del resultado.

Una vez conectado mediante el string que nos proporciona MongoDB, uso la base de datos de sample\_mflix.

```
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> use sample_mflix
switched to db sample_mflix
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> |
```

1. Mostrar sólo el número de películas con una duración superior a 500 minutos.

De la base de datos, escojo la colección de movies y cuento los documentos cuya propiedad “runtime” sea mayor a 500.

```
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> db.movies.countDocuments(
{ runtime: { $gt: 500 } })
21
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> |
```

2. Mostrar sólo el título de las 10 primeras películas por orden alfabético de 1992.

De la colección movies, encuentro los documentos que sean del año 1992 y me quedo solo con los títulos. Después ordeno por título de forma ascendente. Para finalizar el resultado lo limito a 10 películas.

NOTA: el `_id` lo pongo a 0 para que no me lo muestre, porque aunque le indiques que solo quieres el título también te muestra el `_id`.

```
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> db.movies.find({year : 1992}, {title : 1, _id : 0}).sort({title : 1}).limit(10)
[
  { title: '1492: Conquest of Paradise' },
  { title: '3 Ninjas' },
  { title: 'A Demon in My View' },
  { title: 'A Few Good Men' },
  { title: 'A Is for Autism' },
  { title: 'A League of Their Own' },
  { title: 'A Midnight Clear' },
  { title: 'A Place in the World' },
  { title: 'A River Runs Through It' },
  { title: 'A Sense of History' }
]
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> |
```

3. Mostrar el número de películas que hayan ganado más de 70 premios en USA.

Para mostrar el número de películas, hago una consulta a la base de datos y a la colección “movies”. Los documentos tienen la propiedad “award”, que a su vez tiene “wins”, para esta propiedad busco las que tengan más de 70 premios y en la segunda parte le indico que se quede con las películas donde la propiedad “countries” sea “USA”.

```
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> db.movies.countDocuments(
{"award.wins" : {$gt : 70}, "countries" : "USA"})
0
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> |
```

4. Mostrar el número total de películas anteriores a 1930 y posteriores a 2016 (lógicamente han de cumplir una u otra condición).

Cuento los documentos cuyas películas tenga el año menor a 1930 o mayor a 2016. Ambas expresiones utilizo el \$or y las expresiones se las paso por una lista.

```
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> db.movies.countDocuments(
{$or: [{year: {$lt : 1930}}, {year: {$gt: 2016}}]})
66
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> |
```

5. Mostrar el título y año de aquellas películas con una puntuación superior a 7.8 en imdb y que incluya en el reparto a Brad Pitt.

De la colección “movies”, cuento los documentos que devuelva la consulta. Esta tiene una operación lógica y le paso las expresiones por una lista. La primera expresión indica que el “rating” tiene que ser mayor a 7.8 y el “cast” tiene que ser “Brad Pitt”.

```
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> db.movies.countDocuments(
{$and: [{imdb.rating: {$gt : 7.8}}, {cast: "Brad Pitt"}]})
4
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> |
```

6. Mostrar el título de aquellas películas cuyo nombre contenga la palabra “ocean” en mayúscula o minúscula, ordenadas alfabéticamente de manera ascendente.

De la colección “movies”, encuentro los documentos cuyo título siga la expresión regular. Esta expresión indica que debe de tener la palabra “ocean” y acaba con i para indicarle que ignore mayúsculas y minúsculas. El siguiente parámetro que le paso indica las propiedades que debe mostrar de los documentos encontrados que en este caso es título. Una vez realizado el find, ordeno el resultado por el título de forma ascendente.

NOTA: el \_id lo pongo a 0 para que no me lo muestre, porque aunque le indiques que solo quieres el título también te muestra el \_id.

```
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> db.movies.find({title : {
$regex : /ocean/i}}, {title : 1, _id: 0}).sort({title : 1})
[
  { title: 'Color of the Ocean' },
  { title: 'Ocean Heaven' },
  { title: 'Ocean of Pearls' },
  { title: 'Ocean's Eleven' },
  { title: 'Ocean's Thirteen' },
  { title: 'Ocean's Twelve' },
  { title: 'Oceans' },
  { title: 'The Deep End of the Ocean' }
]
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> |
```

7. Cambia a la otra base de datos creada con anterioridad, IABD\_24\_25\_<iniciales nombre, apellido1, apellido2> y sobre la colección SDB\_24\_25, inserta (de uno en uno) tantos documentos como módulos existan en este curso de especialización. Indicando en cada uno de ellos: nombre del módulo (etiqueta “nombre”), horas del módulo (etiquetas “horas”), nombre del profesor (etiqueta “profesor”) y una breve descripción de la materia (etiqueta “descripción”). Lista el contenido actual de la colección SDB\_24\_25.

Cambio y utilizo la base de datos IABD\_24\_25\_JMGM

```
Atlas atlas-4tjcup-shard-0 [primary] sample_mflix> use IABD_24_25_JMGM
switched to db IABD_24_25_JMGM
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |
```

A la colección SGB\_24\_25 le inserto 5 documentos. Cada uno va a tener las siguientes etiquetas nombre, horas, profesor y una descripción. Al insertOne le paso estas etiquetas mediante un objeto. Cada uno nos va a devolver otro objeto indicando que se a creado correctamente y el id que se le ha puesto.

NOTA: Se puede hacer con el método insert pero está desfasado. En caso de querer insertar varios documentos a la vez, se puede utilizar con el insertMany. Para ello le tenemos que pasar los diferentes objetos dentro de una lista.

```
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.SDB_24_25.insertOne
({nombre : "Big Data Aplicado", horas : 75, profesor: "José Antonio Serrano
García", descripcion: "Parte práctica de Big Data."})
{
  acknowledged: true,
  insertedId: ObjectId("6734f15163830cdb3b695061")
}
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |
```

```
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.SDB_24_25.insertOne
({nombre : "Modelos de Inteligencia Artificial", horas : 40, profesor: "Manu
ela Ruiz Peña", descripcion: "Ver y probar los diferentes modelos de IA."})
{
  acknowledged: true,
  insertedId: ObjectId("6734f21e63830cdb3b695062")
}
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |
```

```
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.SDB_24_25.insertOne({nombre : "Programación de Inteligencia Art
ificial", horas : 110, profesor: "Miguel Ángel López Montero", descripcion: "Desarrollar aplicaciones de Inteligencia Ar
tificial utilizando lenguajes de programación y entornos de modelado."})
{
  acknowledged: true,
  insertedId: ObjectId("67361eac3b92123868ee4451")
}
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |
```

```
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.SDB_24_25.insertOne({nombre : "Sistemas de Aprendiz
aje Automático", horas : 50, profesor: "Gerardo González Pérez", descripcion: "Desempeñar las funciones de a
nalizar y relacionar las técnicas de aprendizaje automático con la predicción de comportamientos futuros que
permitan a las organizaciones y empresas la eficiencia operativa."})
{
  acknowledged: true,
  insertedId: ObjectId("67361fb63b92123868ee4452")
}
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |
```

```
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.SDB_24_25.insertOne({nombre : "Sistemas de Big Data", horas : 55, profesor: "Jesús Beas Arco", descripcion: "Desempeñar las funciones de integrar, procesar y analizar la información de grandes volúmenes de datos, aplicando herramientas y técnicas de Big Data para su visualización, valoración y presentación de resultados en la toma de decisiones de las empresas y organizaciones."})
{
  acknowledged: true,
  insertedId: ObjectId("673621063b92123868ee4453")
}
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |
```

Con el método find sin pasarle ningún parámetro, lista todos los documentos que tiene la colección SDB\_24\_25.

```
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.SDB_24_25.find()
[
  {
    _id: ObjectId("6734f15163830cdb3b695061"),
    nombre: 'Big Data Aplicado',
    horas: 75,
    profesor: 'José Antonio Serrano García',
    descripcion: 'Parte práctica de Big Data.'
  },
  {
    _id: ObjectId("6734f21e63830cdb3b695062"),
    nombre: 'Modelos de Inteligencia Artificial',
    horas: 40,
    profesor: 'Manuela Ruiz Peña',
    descripcion: 'Ver y probar los diferentes modelos de IA.'
  },
  {
    _id: ObjectId("67361eac3b92123868ee4451"),
    nombre: 'Programación de Inteligencia Artificial',
    horas: 110,
    profesor: 'Miguel Ángel López Montero',
    descripcion: 'Desarrollar aplicaciones de Inteligencia Artificial utilizando lenguajes de programación y entornos de modelado.'
  },
]
```

```

  {
    _id: ObjectId("67361fb63b92123868ee4452"),
    nombre: 'Sistemas de Aprendizaje Automático',
    horas: 50,
    profesor: 'Gerardo González Pérez',
    descripcion: 'Desempeñar las funciones de analizar y relacionar las técnicas de aprendizaje automático con la predicción de comportamientos futuros que permitan a las organizaciones y empresas la eficiencia operativa.'
  },
  {
    _id: ObjectId("673621063b92123868ee4453"),
    nombre: 'Sistemas de Big Data',
    horas: 55,
    profesor: 'Jesús Beas Arco',
    descripcion: 'Desempeñar las funciones de integrar, procesar y analizar la información de grandes volúmenes de datos, aplicando herramientas y técnicas de Big Data para su visualización, valoración y presentación de resultados en la toma de decisiones de las empresas y organizaciones.'
  }
]
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |
```

8. En esa misma colección, ahora añade otro documento indicando tus datos personales: DNI, dirección, fecha de nacimiento (en formato Date) y fecha (fecha actual de creación).

Al igual que en el ejercicio anterior le paso un objeto con las etiquetas que convenga (DNI, dirección, fecha de nacimiento y fecha actual). Para la fecha de nacimiento creo un objeto de tipo Date y le paso por parámetro un string con mi fecha de nacimiento y para la fecha\_insertado creo un objeto de tipo Date sin pasarle un parámetro, de esta forma pone la fecha actual. Devuelve que se ha insertado correctamente y el id que ha puesto a la inserción.

```
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.SDB_24_25.insertOne({dni : "23234545Z", direccion : {calle : "Calle San Jerónimo", numero : 5}, fecha_nacimiento : new Date("1999-07-23"), fecha_insertado : new Date()})
{
  acknowledged: true,
  insertedId: ObjectId("673623b23b92123868ee4454")
}
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |
```

9. Actualiza el documento anterior por su DNI, añadiendo un nuevo campo afición (indicando varias aficiones dentro de este en forma de array).

Con el método updateOne actualizamos un documento. Para ello le pasamos dos objetos. El primero para indicarle por que tiene que buscar (en este caso el dni) y el segundo con \$set que es el operador de modificación que le ponemos un objeto con la etiqueta que queremos modificar o añadir. En este caso añadimo la etiqueta “aficcion” con una lista y 3 aficiones.

Devuelve que se ha actualizado correctamente, no se ha insertado una id (null), cantidad de documentos que coinciden, cantidad de líneas modificadas y el número de documentos insertados.

NOTA: Con modificar o añadir significa a que si existe lo modifica y si no existe lo añade.

```
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.SDB_24_25.updateOne({dni : "23234545Z"}, {$set : {aficcion : ["Videojuegos", "Leer", "Padel"]}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |
```

10. Lista el contenido de la colección. ¿Todos los documentos tienen la misma estructura? ¿A qué se debe? Detalla si sería posible en una base de datos relacional.

Como se observa en las capturas, los documentos no siguen la misma estructura. Se debe a que mongoDB no sigue un esqueleto fijo, es decir, cada documento puede tener las etiquetas que necesite cada uno. Aunque mongoDB nos permite que cada documento tenga sus propias etiquetas, es una buena práctica que estas no cambien mucho entre documentos para que no se vaya de las manos.

En una base de datos esto sería inviable, porque tendríamos que crear cada característica posible y definir las de tal forma que puedan tener valores nulos porque algunas de las características no van a existir. Esta es una ventaja que nos proporciona las no relacionales frente a las relacionales, no tienen que seguir un esquema predefinido.

```
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.SDB_24_25.find()
[
  {
    _id: ObjectId("6734f15163830cdb3b695061"),
    nombre: 'Big Data Aplicado',
    horas: 75,
    profesor: 'José Antonio Serrano García',
    descripcion: 'Parte práctica de Big Data.'
  },
  {
    _id: ObjectId("6734f21e63830cdb3b695062"),
    nombre: 'Modelos de Inteligencia Artificial',
    horas: 40,
    profesor: 'Manuela Ruiz Peña',
    descripcion: 'Ver y probar los diferentes modelos de IA.'
  },
  {
    _id: ObjectId("67361eac3b92123868ee4451"),
    nombre: 'Programación de Inteligencia Artificial',
    horas: 110,
    profesor: 'Miguel Ángel López Montero',
    descripcion: 'Desarrollar aplicaciones de Inteligencia Artificial utilizando lenguajes de programación y entornos de modelado.'
  },
]
```



```

{
  _id: ObjectId("67361fb63b92123868ee4452"),
  nombre: 'Sistemas de Aprendizaje Automático',
  horas: 50,
  profesor: 'Gerardo González Pérez',
  descripcion: 'Desempeñar las funciones de analizar y relacionar las técnicas de aprendizaje automático con la predicción de comportamientos futuros que permitan a las organizaciones y empresas la eficiencia operativa.'
},
{
  _id: ObjectId("673621063b92123868ee4453"),
  nombre: 'Sistemas de Big Data',
  horas: 55,
  profesor: 'Jesús Beas Arco',
  descripcion: 'Desempeñar las funciones de integrar, procesar y analizar la información de grandes volúmenes de datos, aplicando herramientas y técnicas de Big Data para su visualización, valoración y presentación de resultados en la toma de decisiones de las empresas y organizaciones.'
},
{
  _id: ObjectId("673623b23b92123868ee4454"),
  dni: '23234545Z',
  direccion: { calle: 'Calle San Jerónimo', numero: 5 },
  fecha_nacimiento: ISODate("1999-07-23T00:00:00.000Z"),
  fecha_insertado: ISODate("2024-11-14T16:22:10.956Z"),
  aficcion: [ 'Videojuegos', 'Leer', 'Padel' ]
}
]
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |

```

11. Elimina la colección SDB\_24\_25 y la base de datos IABD\_24\_25<iniciales nombre, apellido1, apellidos2>.

Elimino la colección SDB\_24\_25 de la base de datos que esté seleccionada con el método drop. Devuelve true si ha eliminado correctamente, por el contrario, no se realiza la operación.

```

Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.SDB_24_25.drop()
true

```

dropDatabase elimina la base de datos que esté seleccionada. La BD seleccionada es IABD\_24\_25\_JMGM como se puede ver en la captura. Una vez que ejecutamos el comando, devuelve ok : 1 si se ha eliminado correctamente y el nombre de la base de datos eliminada.

```

Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> db.dropDatabase()
{ ok: 1, dropped: 'IABD_24_25_JMGM' }
Atlas atlas-4tjcup-shard-0 [primary] IABD_24_25_JMGM> |

```

