

UD3.T1 Entrega Ejercicios básicos CRUD

EJERCICIO 1

1. Partiendo del escritorio de tu ordenador, ejecuta los pasos necesarios para poder usar la consola de mongodb (abrir un terminal o consola, iniciar/parar servicios, ejecutar la shell de mongodb).

```
C:\Users\Ángel Alberto>mongosh
Current Mongosh Log ID: 6362aba9e3eecefe84dd1be5
Connecting to:  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.0
Using MongoDB:  6.0.2
Using Mongosh:  1.6.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
  2022-10-26T19:04:59.905+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

-----
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----
```

2. Crea una base de datos llamada concesionario. Recuerda insertar al menos un documento en una colección que se llame coches. Como propiedades del documento json, puedes usar matrícula, marca, modelo, versiones (sport, confort), kms y fecha de matriculación.

```
concesionario> db.createCollection("coches")
{ ok: 1 }
concesionario> db.coches.insertOne({marca:"ferrari", matricula:"12345AWS", modelo:"S", version:"sport", kms:"72", fechaMatriculacion:"19-02-2022"})
{ acknowledged: true, insertedId: ObjectId("6362ae7e8d19ce78d73f0eab") }
```

3. Crea una colección que se llame clientes. Como propiedades del documento json, puedes usar nombre, f_nacimiento, e_mail y telefono.

```
concesionario> db.createCollection("clientes")
{ ok: 1 }
concesionario> db.clientes.insertOne({nombre:"Angel", f_nacimiento:"17-04-2002", email:"angel.umbrete.pinorueda@gmail.com", telefono:"693814832"})
{ acknowledged: true, insertedId: ObjectId("6362ade68d19ce78d73f0eaa") }
```

4. Visualiza el listado de todas las colecciones de datos disponibles y su contenido.

```

concesionario> db.coches.find().pretty()
[
  {
    _id: ObjectId("6362ae7e8d19ce78d73f0eab"),
    marca: 'ferrari',
    matricula: '12345AWS',
    modelo: 'S',
    version: 'sport',
    kms: '72',
    fechaMatriculacion: '19-02-2022'
  }
]
concesionario> db.clientes.find().pretty()
[
  {
    _id: ObjectId("6362ade68d19ce78d73f0eaa"),
    nombre: 'Angel',
    f_nacimiento: '17-04-2002',
    email: 'angel.umbrete.pinorueda@gmail.com',
    telefono: '693814832'
  }
]

```

EJERCICIO 2

- Crea la base de datos Retail.

```

concesionario> use retail
switched to db retail

```

- Inserta en la colección productos con los siguientes documentos de uno en uno:

<pre> { "referencia": "P0001", "tipo": "camisa", "paraHombre": true, "talla": "XS", "precio": 20.99 } </pre>	<pre> { "referencia": "P0002", "tipo": "camisa", "paraHombre": true, "talla": "XL", "precio": 30.25 } </pre>	<pre> { "referencia": "P0003", "tipo": "pantalon", "paraMujer": true, "talla": "L", "precio": 20.99 } </pre>
--	--	--

```

retail> db.createCollection("productos")
{ ok: 1 }
retail> db.productos.insertOne({referencia:"P0001", tipo:"camisa", paraHombre:true, talla:"XS", precio:20.99})
{
  acknowledged: true,
  insertedId: ObjectId("6362b1098d19ce78d73f0eac")
}
retail> db.productos.insertOne({referencia:"P0002", tipo:"camisa", paraHombre:true, talla:"XL", precio:30.25})
{
  acknowledged: true,
  insertedId: ObjectId("6362b1458d19ce78d73f0ead")
}
retail> db.productos.insertOne({referencia:"P0003", tipo:"pantalon", paraMujer:true, talla:"L", precio:20.99})
{
  acknowledged: true,
  insertedId: ObjectId("6362b19d8d19ce78d73f0eae")
}

```

- Elimínalos todos.

```

retail> db.productos.deleteMany({})
{ acknowledged: true, deletedCount: 3 }

```

- Ahora créalos, pero todos a la vez.

```

retail> db.productos.insertMany([{"referencia": "P0001", "tipo": "camisa", "paraHombre": true, "talla": "XS", "precio": 20.99}, {"referencia": "P0002", "tipo": "camisa", "paraHombre": true, "talla": "XL", "precio": 30.25}, {"referencia": "P0003", "tipo": "pantalon", "paraMujer": true, "talla": "L", "precio": 20.99}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6363919df776ec18d193232e"),
    '1': ObjectId("6363919df776ec18d193232f"),
    '2': ObjectId("6363919df776ec18d1932330")
  }
}

```

- Actualiza todos los documentos con un precio inferior a 25 para que tenga un precio un 10% más caro.
- Reemplaza cada documento que sea para hombre con la misma estructura pero añadiendo una propiedad nueva: "paraMujer": false.
- Reemplaza cada documento que sea para mujer con la misma estructura pero añadiendo una propiedad nueva: "paraHombre": false.
- Actualiza cada documento, filtrando por sus referencias. Las propiedades cambiando el tipo camisa por chaqueta y la talla debe ser igual a M.

EJERCICIO 3

- Utilizando la base de datos Retail.

<pre> { "referencia": "P0001", "tipo": "camisa", "paraHombre": true, "talla": "XS", "precio": 20.99 } </pre>	<pre> { "referencia": "P0002", "tipo": "camisa", "paraHombre": true, "talla": "XL", "precio": 30.25 } </pre>	<pre> { "referencia": "P0003", "tipo": "pantalon", "paraMujer": true, "talla": "L", "precio": 20.99 } </pre>
--	--	--

- Actualiza la colección productos con las siguientes propiedades:
 - P0001, añade las siguientes propiedades:
tienda: ["Jerez", "Sevilla", "Cordoba"]
proveedor: {nombre: "Camiseros SA", nif: "B12345678", contacto: "Jose"}
 - P0002, añade las siguientes propiedades:
tienda: ["Jerez", "Chucena", "Cordoba"]
proveedor: {nombre: "Camiseros SA", nif: "B12345678", contacto: "Juan"}
 - P0003, añade las siguientes propiedades:
tienda: ["Rota", "Sevilla"]
proveedor: {nombre: "Pantaloneros SA", nif: "B87654321", contacto: "Jose"}
- Realiza las siguientes consultas sobre la colección productos:
 - De cada producto mostrar su referencia, las tiendas en las que se vende y el nombre del proveedor que lo suministra.
 - Mostrar la referencia y talla de todos los productos que se venden en la tienda de Jerez.
 - Mostrar los productos que se venden en las tiendas de Jerez y Córdoba.

- Mostrar la referencia, talla y precio de los productos suministrados por el proveedor cuyo nif es "B12345678".
- Mostrar los productos cuyo contacto del proveedor sea "Jose"

EJERCICIO 4