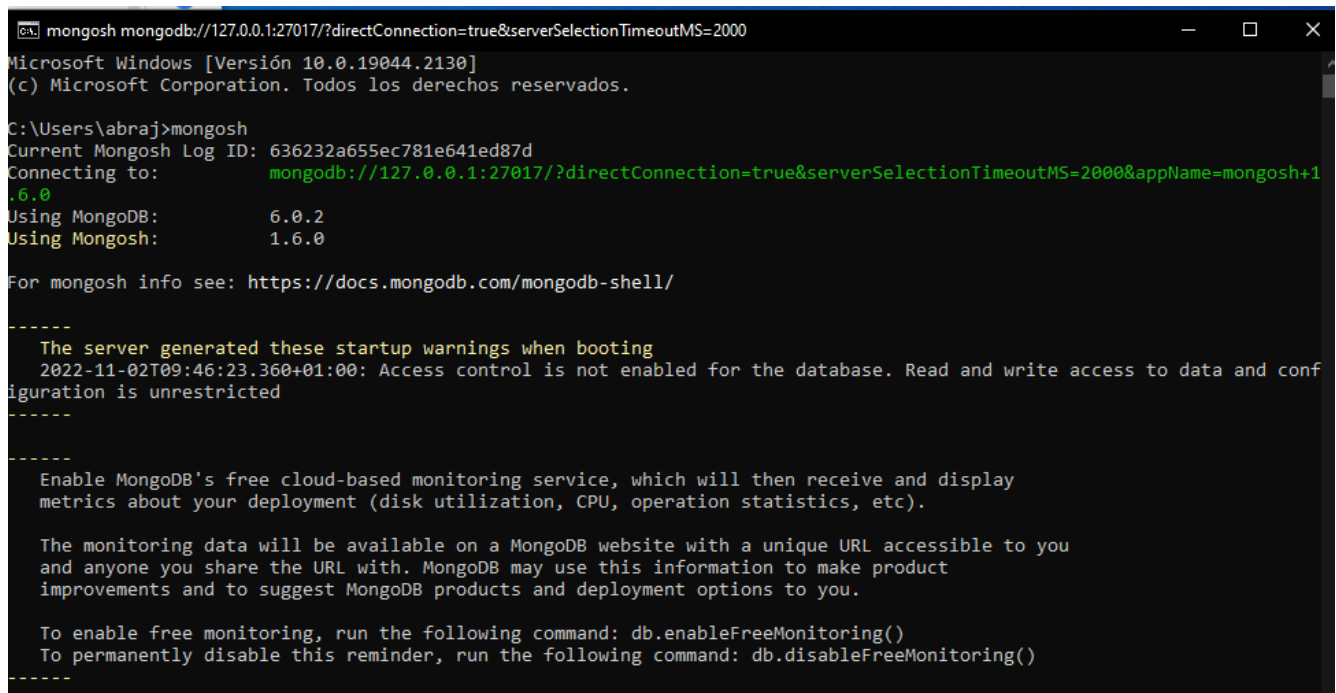


EJERCICIOS

EJERCICIO1

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)



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [Versión 10.0.19044.2130]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\abraj>mongosh
Current Mongosh Log ID: 636232a655ec781e641ed87d
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/mongosh-shell/

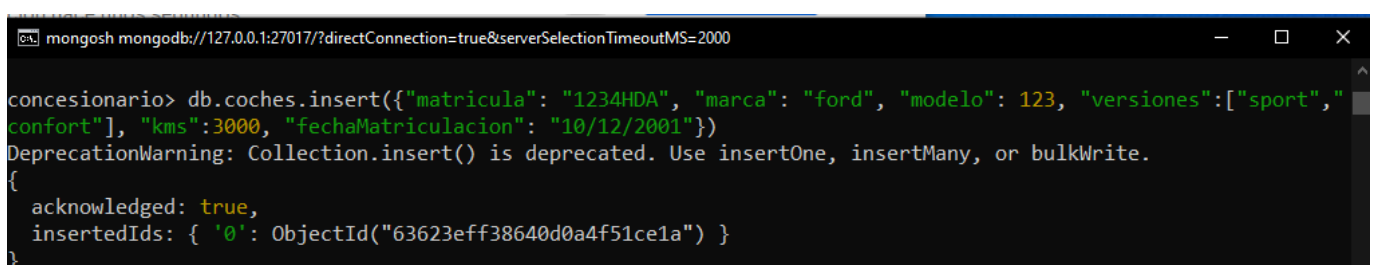
-----
  The server generated these startup warnings when booting
  2022-11-02T09:46:23.360+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

-----
  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.coches.insert({"matricula": "1234HDA", "marca": "ford", "modelo": "123", "versiones": ["sport", "confort"], "kms": 3000, "fechaMatriculacion": "10/12/2001"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63623eff38640d0a4f51ce1a") }
}
```

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

```
Seleccinar mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
concesionario> db.clientes.insert({"nombre": "Abraham", "f_nacimiento": "10/01/2001", "e_mail": "abraham@gmail.com", "telefono": 123456789})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("636242b438640d0a4f51ce1c") }
}
```

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
concesionario> show collections
clientes
coches
concesionario> db.clientes.find()
[
  {
    _id: ObjectId("636242b438640d0a4f51ce1c"),
    nombre: 'Abraham',
    f_nacimiento: '10/01/2001',
    e_mail: 'abraham@gmail.com',
    telefono: 123456789
  }
]
concesionario> db.coches.find()
[
  {
    _id: ObjectId("63623eff38640d0a4f51ce1a"),
    matricula: '1234HDA',
    marca: 'ford',
    modelo: 123,
    versiones: [ 'sport', 'confort' ],
    kms: 3000,
    fechaMatriculacion: '10/12/2001'
  }
]
```

EJERCICIO2

- Crea la base de datos Retail.

```
Seleccionar mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
concesionario> use Retail
switched to db Retail
Retail> _
```

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

```
{
  "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
}
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
concesionario> use Retail
switched to db Retail
Retail> db.productos.insert({"referencia": "P0001", "tipo": "camisa", "paraHombre": true, "talla": "XS", "precio": 20.99})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63624a1238640d0a4f51ce1d") }
}
Retail> db.productos.insert({"referencia": "P0002", "tipo": "camisa", "paraHombre": true, "talla": "XL", "precio": 30.25})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63624a3738640d0a4f51ce1e") }
}
Retail> db.productos.insert({"referencia": "P0003", "tipo": "pantalon", "paraMujer": true, "talla": "L", "precio": 20.99})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63624acc38640d0a4f51ce1f") }
}
Retail> show collections
productos
```

```

C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Retail> db.productos.find().pretty()
[
  {
    _id: ObjectId("63624a1238640d0a4f51ce1d"),
    referencia: 'P0001',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XS',
    precio: 20.99
  },
  {
    _id: ObjectId("63624a3738640d0a4f51ce1e"),
    referencia: 'P0002',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XL',
    precio: 30.25
  },
  {
    _id: ObjectId("63624acc38640d0a4f51ce1f"),
    referencia: 'P0003',
    tipo: 'pantalon',
    paraMujer: true,
    talla: 'L',
    precio: 20.99
  }
]

```

- Elimínelos todos.

```

C:\> Seleccionar mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Retail> db.productos.drop();
true
Retail> show collections
Retail>

```

- Ahora créalos, pero todos a la vez.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
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("6362500b460e5c0a91653cb7"),
    '1': ObjectId("6362500b460e5c0a91653cb8"),
    '2': ObjectId("6362500b460e5c0a91653cb9")
  }
}
Retail> show collections
productos
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Retail> db.productos.find().pretty()
[
  {
    _id: ObjectId("6362500b460e5c0a91653cb7"),
    referencia: 'P0001',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XS',
    precio: 20.99
  },
  {
    _id: ObjectId("6362500b460e5c0a91653cb8"),
    referencia: 'P0002',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XL',
    precio: 30.25
  },
  {
    _id: ObjectId("6362500b460e5c0a91653cb9"),
    referencia: 'P0003',
    tipo: 'pantalon',
    paraMujer: true,
    talla: 'L',
    precio: 20.99
  }
]
Retail>
```

- Actualiza todos los documentos con un precio inferior a 25 para que tenga un precio un 10% más caro.

```
Seleccionar mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Retail> db.productos.updateMany({precio: {$lt:25}},{$mul: {precio: 1.10}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
Retail> db.productos.find().pretty()
[
  {
    _id: ObjectId("63626631460e5c0a91653cba"),
    referencia: 'P0001',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XS',
    precio: 23.089
  },
  {
    _id: ObjectId("63626631460e5c0a91653cbb"),
    referencia: 'P0002',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XL',
    precio: 30.25
  },
  {
    _id: ObjectId("63626631460e5c0a91653cbc"),
    referencia: 'P0003',
    tipo: 'pantalon',
    paraMujer: true,
    talla: 'L',
    precio: 23.089
  }
]
Retail> _
```

- Reemplaza cada documento que sea para hombre con la misma estructura pero añadiendo una propiedad nueva: “paraMujer”: false.

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
test> db.productos.updateMany({paraHombre: true},{ $set: {"paraMujer": false}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
test> db.productos.find().pretty()
[
  {
    _id: ObjectId("6362ace5d1438fdf044cc6e2"),
    referencia: 'P0001',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XS',
    precio: 20.99,
    paraMujer: false
  },
  {
    _id: ObjectId("6362ace5d1438fdf044cc6e3"),
    referencia: 'P0002',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XL',
    precio: 30.25,
    paraMujer: false
  },
  {
    _id: ObjectId("6362ace5d1438fdf044cc6e4"),
    referencia: 'P0003',
    tipo: 'pantalon',
    paraMujer: true,
    talla: 'L',
    precio: 20.99
  }
]
test> 

```

- Reemplaza cada documento que sea para mujer con la misma estructura pero añadiendo una propiedad nueva: "paraHombre": false.

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
test> db.productos.updateMany({paraMujer: true},{ $set: {"paraHombre": false}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
test> db.productos.find().pretty()
[
  {
    _id: ObjectId("6362ace5d1438fdf044cc6e2"),
    referencia: 'P0001',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XS',
    precio: 20.99,
    paraMujer: false
  },
  {
    _id: ObjectId("6362ace5d1438fdf044cc6e3"),
    referencia: 'P0002',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XL',
    precio: 30.25,
    paraMujer: false
  },
  {
    _id: ObjectId("6362ace5d1438fdf044cc6e4"),
    referencia: 'P0003',
    tipo: 'pantalon',
    paraMujer: true,
    talla: 'L',
    precio: 20.99,
    paraHombre: false
  }
]

```

- Actualiza cada documento, filtrando por sus referencia. Las propiedades cambiando el tipo camisa por chaqueta y la talla debe ser igual a M.


```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
test> db.productos.updateMany({tipo: "camisa"},{$set: {tipo: "chaqueta", talla: "M"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
test> db.productos.find().pretty()
[
  {
    _id: ObjectId("6362b15bd1438fdf044cc6e5"),
    referencia: 'P0001',
    tipo: 'chaqueta',
    paraHombre: true,
    talla: 'M',
    precio: 20.99,
    paraMujer: false
  },
  {
    _id: ObjectId("6362b15bd1438fdf044cc6e6"),
    referencia: 'P0002',
    tipo: 'chaqueta',
    paraHombre: true,
    talla: 'M',
    precio: 30.25,
    paraMujer: false
  },
  {
    _id: ObjectId("6362b15bd1438fdf044cc6e7"),
    referencia: 'P0003',
    tipo: 'pantalon',
    paraMujer: true,
    talla: 'L',
    precio: 20.99,
    paraHombre: false
  }
]

```

EJERCICIO3

- Actualiza la colección productos con las siguientes propiedades:
- Utilizando la base de datos Retail.
- 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"}

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Retail> db.productos.updateMany({referencia: "P0001"},{$set: {tienda: ["Jerez", "Sevilla", "Cordoba"], proveedor:{nombre: "Camiseros SA", nif:"B12345678", contacto:"Jose"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Retail> db.productos.find().pretty()
[
  {
    _id: ObjectId("63626bf1460e5c0a91653cbd"),
    referencia: 'P0001',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XS',
    precio: 23.089,
    proveedor: { nombre: 'Camiseros SA', nif: 'B12345678', contacto: 'Jose' },
    tienda: [ 'Jerez', 'Sevilla', 'Cordoba' ]
  },
  {

```

- P0002, añade las siguientes propiedades:
 tienda: ["Jerez", "Chucena", "Cordoba"]
 proveedor: {nombre: "Camiseros SA", nif:"B12345678",
 contacto:"Juan"}

```

Seleccionar mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Retail> db.productos.updateMany({referencia: "P0002"},{$set: {tienda: ["Jerez", "Chucena", "Cordoba"], proveedor:{nombre: "Camiseros SA", nif:"B12345678", contacto:"Juan"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Retail> db.productos.find().pretty()
[
  {
    _id: ObjectId("63626bf1460e5c0a91653cbd"),
    referencia: 'P0001',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XS',
    precio: 23.089,
    proveedor: { nombre: 'Camiseros SA', nif: 'B12345678', contacto: 'Jose' },
    tienda: [ 'Jerez', 'Sevilla', 'Cordoba' ]
  },
  {
    _id: ObjectId("63626bf1460e5c0a91653cbe"),
    referencia: 'P0002',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XL',
    precio: 30.25,
    proveedor: { nombre: 'Camiseros SA', nif: 'B12345678', contacto: 'Juan' },
    tienda: [ 'Jerez', 'Chucena', 'Cordoba' ]
  },
  {

```

- P0003, añade las siguientes propiedades:
 tienda: ["Rota", "Sevilla"]
 proveedor: {nombre: "Pantaloneros SA", nif:"B87654321",
 contacto:"Jose"}

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Retail> db.productos.updateMany({referencia: "P0003"},{$set: {tienda: ["Rota", "Sevilla"], proveedor:{nombre: "Panta
loneros SA", nif:"B87654321", contacto:"Jose"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Retail> db.productos.find().pretty()
[
  {
    _id: ObjectId("63626bf1460e5c0a91653cbd"),
    referencia: 'P0001',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XS',
    precio: 23.089,
    proveedor: { nombre: 'Camiseros SA', nif: 'B12345678', contacto: 'Jose' },
    tienda: [ 'Jerez', 'Sevilla', 'Cordoba' ]
  },
  {
    _id: ObjectId("63626bf1460e5c0a91653cbe"),
    referencia: 'P0002',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XL',
    precio: 30.25,
    proveedor: { nombre: 'Camiseros SA', nif: 'B12345678', contacto: 'Juan' },
    tienda: [ 'Jerez', 'Chucena', 'Cordoba' ]
  },
  {
    _id: ObjectId("63626bf1460e5c0a91653cbf"),
    referencia: 'P0003',
    tipo: 'pantalón',
    paraMujer: true,
    talla: 'L',
    precio: 23.089,
    proveedor: { nombre: 'Pantaloneros SA', nif: 'B87654321', contacto: 'Jose' },
    tienda: [ 'Rota', 'Sevilla' ]
  }
]
Retail>

```

- 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.

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Retail> db.productos.find({}, {"referencia":true, "tienda":true, "proveedor":true})
[
  {
    _id: ObjectId("63626bf1460e5c0a91653cbd"),
    referencia: 'P0001',
    proveedor: { nombre: 'Camiseros SA', nif: 'B12345678', contacto: 'Jose' },
    tienda: [ 'Jerez', 'Sevilla', 'Cordoba' ]
  },
  {
    _id: ObjectId("63626bf1460e5c0a91653cbe"),
    referencia: 'P0002',
    proveedor: { nombre: 'Camiseros SA', nif: 'B12345678', contacto: 'Juan' },
    tienda: [ 'Jerez', 'Chucena', 'Cordoba' ]
  },
  {
    _id: ObjectId("63626bf1460e5c0a91653cbf"),
    referencia: 'P0003',
    proveedor: { nombre: 'Pantaloneros SA', nif: 'B87654321', contacto: 'Jose' },
    tienda: [ 'Rota', 'Sevilla' ]
  }
]
Retail>

```

- Mostrar la referencia y talla de todos los productos que se venden en la tienda de Jerez.

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Retail> db.productos.find({"tienda":"Jerez"}, {"referencia":true, "talla":true})
[
  {
    _id: ObjectId("63626bf1460e5c0a91653cbd"),
    referencia: 'P0001',
    talla: 'XS'
  },
  {
    _id: ObjectId("63626bf1460e5c0a91653cbe"),
    referencia: 'P0002',
    talla: 'XL'
  }
]
Retail>

```

- Mostrar los productos que se venden en las tiendas de Jerez y Cordoba.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Retail> db.productos.find({tienda: {$all: ["Jerez", "Cordoba"]}})
[
  {
    _id: ObjectId("63626bf1460e5c0a91653cbd"),
    referencia: 'P0001',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XS',
    precio: 23.089,
    proveedor: { nombre: 'Camiseros SA', nif: 'B12345678', contacto: 'Jose' },
    tienda: [ 'Jerez', 'Sevilla', 'Cordoba' ]
  },
  {
    _id: ObjectId("63626bf1460e5c0a91653cbe"),
    referencia: 'P0002',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XL',
    precio: 30.25,
    proveedor: { nombre: 'Camiseros SA', nif: 'B12345678', contacto: 'Juan' },
    tienda: [ 'Jerez', 'Chucena', 'Cordoba' ]
  }
]
Retail> _
```

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Retail> db.productos.find({"proveedor.nif": "B12345678"}, {"referencia":true, "talla":true, "precio":true})
[
  {
    _id: ObjectId("63626bf1460e5c0a91653cbd"),
    referencia: 'P0001',
    talla: 'XS',
    precio: 23.089
  },
  {
    _id: ObjectId("63626bf1460e5c0a91653cbe"),
    referencia: 'P0002',
    talla: 'XL',
    precio: 30.25
  }
]
Retail> _
```

- Mostrar los productos cuyo contacto del proveedor sea “Jose”.

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Retail> db.productos.find({"proveedor.contacto": "Jose"})
[
  {
    _id: ObjectId("63626bf1460e5c0a91653cbd"),
    referencia: 'P0001',
    tipo: 'camisa',
    paraHombre: true,
    talla: 'XS',
    precio: 23.089,
    proveedor: { nombre: 'Camiseros SA', nif: 'B12345678', contacto: 'Jose' },
    tienda: [ 'Jerez', 'Sevilla', 'Cordoba' ]
  },
  {
    _id: ObjectId("63626bf1460e5c0a91653cbf"),
    referencia: 'P0003',
    tipo: 'pantalon',
    paraMujer: true,
    talla: 'L',
    precio: 23.089,
    proveedor: { nombre: 'Pantaloneros SA', nif: 'B87654321', contacto: 'Jose' },
    tienda: [ 'Rota', 'Sevilla' ]
  }
]
Retail>

```

EJERCICIO4

Trabajando con Profesores

```

{
  "nombre": "María",
  "apellidos": "Suárez Manrique",
  "especialidad": [
    "biología"
  ],
  "esTitular": true,
  "esAsociado": false,
  "edad": 51
},
{
  "nombre": "Jose Luis",
  "apellidos": "López Pérez",
  "especialidad": [
    "matemáticas",
    "física"
  ],
  "esTitular": false,
  "esAsociado": true,
  "edad": 39
},
{
  "nombre": "Antonio",
  "apellidos": "Munigua Arteché",
  "especialidad": [
    "física, química"
  ],
  "esTitular": true,
  "esAsociado": false,
  "edad": 54
}
]

```

- Obtener todos los profesores.
- Obtener todos los profesores asociados.
- Obtener todos los profesores titulares y mayores de 50.
- Obtener todos los profesores con edades entre 50 y 60.
- Número total de profesores.
- Número total de profesores asociados.
- Obtener solo los 2 primeros resultados.

- Obtener todos los profesores.

```
C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find()
[
  {
    _id: ObjectId("6363879c2e434ce09a427992"),
    nombre: 'Maria',
    apellidos: 'Suarez Manrique',
    especialidad: [ 'biologia' ],
    esTitular: true,
    esAsociado: false,
    edad: 51
  },
  {
    _id: ObjectId("636387fb2e434ce09a427993"),
    nombre: 'Jose Luis',
    apellidos: 'Lopez Perez',
    especialidad: [ 'matematicas', 'fisica' ],
    esTitular: false,
    esAsociado: true,
    edad: 39
  },
  {
    _id: ObjectId("636388532e434ce09a427994"),
    nombre: 'Antonio',
    apellidos: 'Munguia Arteche',
    especialidad: [ 'fisica', 'quimica' ],
    esTitular: true,
    esAsociado: false,
    edad: 54
  }
]
Profesores>
```

- Obtener todos los profesores asociados.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({"esAsociado":true})
[
  {
    _id: ObjectId("636387fb2e434ce09a427993"),
    nombre: 'Jose Luis',
    apellidos: 'Lopez Perez',
    especialidad: [ 'matematicas', 'fisica' ],
    esTitular: false,
    esAsociado: true,
    edad: 39
  }
]
Profesores>
```

- Obtener todos los profesores titulares y mayores de 50.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({"esTitular":true, edad:{$gt:50}})
[
  {
    _id: ObjectId("6363879c2e434ce09a427992"),
    nombre: 'Maria',
    apellidos: 'Suarez Manrique',
    especialidad: [ 'biologia' ],
    esTitular: true,
    esAsociado: false,
    edad: 51
  },
  {
    _id: ObjectId("636388532e434ce09a427994"),
    nombre: 'Antonio',
    apellidos: 'Munguia Arteche',
    especialidad: [ 'fisica', 'quimica' ],
    esTitular: true,
    esAsociado: false,
    edad: 54
  }
]
Profesores> █
```


- Obtener todos los profesores con edades entre 50 y 60.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({edad:{$gte:50 , $lte:60}})
[
  {
    _id: ObjectId("6363879c2e434ce09a427992"),
    nombre: 'Maria',
    apellidos: 'Suarez Manrique',
    especialidad: [ 'biologia' ],
    esTitular: true,
    esAsociado: false,
    edad: 51
  },
  {
    _id: ObjectId("636388532e434ce09a427994"),
    nombre: 'Antonio',
    apellidos: 'Munguia Arteche',
    especialidad: [ 'fisica', 'quimica' ],
    esTitular: true,
    esAsociado: false,
    edad: 54
  }
]
Profesores>
```

- Numero total de profesores.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.count()
3
Profesores> _
```

- Numero total de profesores asociados.

```
C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({"esAsociado":true}).count()
1
```

- Obtener solo los 2 primeros resultados.

```
C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find().limit(2)
[
  {
    _id: ObjectId("6363879c2e434ce09a427992"),
    nombre: 'Maria',
    apellidos: 'Suarez Manrique',
    especialidad: [ 'biologia' ],
    esTitular: true,
    esAsociado: false,
    edad: 51
  },
  {
    _id: ObjectId("636387fb2e434ce09a427993"),
    nombre: 'Jose Luis',
    apellidos: 'Lopez Perez',
    especialidad: [ 'matematicas', 'fisica' ],
    esTitular: false,
    esAsociado: true,
    edad: 39
  }
]
Profesores> _
```

- Obtener todos los profesores con especialidad física o biología.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({$or:[{"especialidad":"fisica"}, {"especialidad":"biologia"}]})
[
  {
    _id: ObjectId("6363879c2e434ce09a427992"),
    nombre: 'Maria',
    apellidos: 'Suarez Manrique',
    especialidad: [ 'biologia' ],
    esTitular: true,
    esAsociado: false,
    edad: 51
  },
  {
    _id: ObjectId("636387fb2e434ce09a427993"),
    nombre: 'Jose Luis',
    apellidos: 'Lopez Perez',
    especialidad: [ 'matematicas', 'fisica' ],
    esTitular: false,
    esAsociado: true,
    edad: 39
  },
  {
    _id: ObjectId("636388532e434ce09a427994"),
    nombre: 'Antonio',
    apellidos: 'Munguia Arteche',
    especialidad: [ 'fisica', 'quimica' ],
    esTitular: true,
    esAsociado: false,
    edad: 54
  }
]
Profesores> _
```

- Obtener todos los profesores con especialidad biología o que sea asociado.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({"especialidad":"biologia", "esAsociado":true})

Profesores> _
```

- Obtener todos los profesores que imparten exactamente matemáticas y física (exactly en ese orden).

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Profesores> db.Profesores.find({"especialidad.0":"matematicas", "especialidad.1":"fisica"})
[
  {
    _id: ObjectId("636387fb2e434ce09a427993"),
    nombre: 'Jose Luis',
    apellidos: 'Lopez Perez',
    especialidad: [ 'matematicas', 'fisica' ],
    esTitular: false,
    esAsociado: true,
    edad: 39
  }
]
Profesores> _
```

- Obtener todos los profesores que imparten matemáticas y física (sin orden).

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Profesores> db.Profesores.find({"especialidad":"matematicas", "especialidad":"fisica"})
[
  {
    _id: ObjectId("636387fb2e434ce09a427993"),
    nombre: 'Jose Luis',
    apellidos: 'Lopez Perez',
    especialidad: [ 'matematicas', 'fisica' ],
    esTitular: false,
    esAsociado: true,
    edad: 39
  },
  {
    _id: ObjectId("636388532e434ce09a427994"),
    nombre: 'Antonio',
    apellidos: 'Munguia Arteche',
    especialidad: [ 'fisica', 'quimica' ],
    esTitular: true,
    esAsociado: false,
    edad: 54
  }
]
Profesores>
```

- Número total de profesores que imparten biología.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
```

```
Profesores> db.Profesores.find({"especialidad":"biologia"}).count()
1
```

Trabajando con Profesores y asignaturas

<pre>{ "nombre": "María", "apellidos": "Suárez Manrique", "especialidad": ["biología"], "esTitular": true, "esAsociado": false, "edad": 51, "asignatura": { "id": "A0001", "nombre": "Biología molecular", "creditos": 6 } }</pre>	<pre>{ "nombre": "Jose Luis", "apellidos": "López Pérez", "especialidad": ["matemáticas", "física"], "esTitular": false, "esAsociado": true, "edad": 39, "asignatura": { "id": "A0002", "nombre": "Termodinámica", "creditos": 9 } }</pre>	<pre>{ "nombre": "Antonio", "apellidos": "Munguía Arteche", "especialidad": ["física, química"], "esTitular": true, "esAsociado": false, "edad": 55, "asignatura": { "id": "A0003", "nombre": "Termodinámica", "creditos": 9 } }</pre>
--	--	--

Actualiza la colección Profesores con los datos de las asignaturas que imparten.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
```

```
Profesores> db.Profesores.update({nombre:"María"},{$set: { asignatura:{id:"A0001", "nombre":"Biología molecular", "creditos":6}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Profesores> db.Profesores.find()
[
  {
    _id: ObjectId("6363879c2e434ce09a427992"),
    nombre: 'María',
    apellidos: 'Suarez Manrique',
    especialidad: [ 'biología' ],
    esTitular: true,
    esAsociado: false,
    edad: 51,
    asignatura: { id: 'A0001', nombre: 'Biología molecular', creditos: 6 }
  }
]
```

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Profesores> db.Profesores.update({nombre:"Jose Luis"},{$set: { asignatura:{id:"A0002", "nombre":"Termodinamica", "creditos":9}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Profesores> db.Profesores.update({nombre:"Antonio"},{$set: { asignatura:{id:"A0002", "nombre":"Termodinamica", "creditos":9}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

Obtener todos los profesores que imparten la asignatura A0002.

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Profesores> db.Profesores.find({"asignatura.id":"A0002"})
[
  {
    _id: ObjectId("636387fb2e434ce09a427993"),
    nombre: 'Jose Luis',
    apellidos: 'Lopez Perez',
    especialidad: [ 'matematicas', 'fisica' ],
    esTitular: false,
    esAsociado: true,
    edad: 39,
    asignatura: { id: 'A0002', nombre: 'Termodinamica', creditos: 9 }
  },
  {
    _id: ObjectId("636388532e434ce09a427994"),
    nombre: 'Antonio',
    apellidos: 'Munguia Arteche',
    especialidad: [ 'fisica', 'quimica' ],
    esTitular: true,
    esAsociado: false,
    edad: 54,
    asignatura: { id: 'A0002', nombre: 'Termodinamica', creditos: 9 }
  }
]
Profesores>

```

Obtener todos los profesores que imparten termodinámica.

```
C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({"asignatura.nombre":"Termodinamica"})
[
  {
    _id: ObjectId("636387fb2e434ce09a427993"),
    nombre: 'Jose Luis',
    apellidos: 'Lopez Perez',
    especialidad: [ 'matematicas', 'fisica' ],
    esTitular: false,
    esAsociado: true,
    edad: 39,
    asignatura: { id: 'A0002', nombre: 'Termodinamica', credits: 9 }
  },
  {
    _id: ObjectId("636388532e434ce09a427994"),
    nombre: 'Antonio',
    apellidos: 'Munguia Arteche',
    especialidad: [ 'fisica', 'quimica' ],
    esTitular: true,
    esAsociado: false,
    edad: 54,
    asignatura: { id: 'A0002', nombre: 'Termodinamica', credits: 9 }
  }
]
Profesores> _
```

Obtener todos los profesores que den asignaturas de más de 6 créditos.

```

C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({"asignatura.creditos": {$gt: 6}})
[
  {
    _id: ObjectId("636387fb2e434ce09a427993"),
    nombre: 'Jose Luis',
    apellidos: 'Lopez Perez',
    especialidad: [ 'matematicas', 'fisica' ],
    esTitular: false,
    esAsociado: true,
    edad: 39,
    asignatura: { id: 'A0002', nombre: 'Termodinamica', creditos: 9 }
  },
  {
    _id: ObjectId("636388532e434ce09a427994"),
    nombre: 'Antonio',
    apellidos: 'Munguia Arteche',
    especialidad: [ 'fisica', 'quimica' ],
    esTitular: true,
    esAsociado: false,
    edad: 54,
    asignatura: { id: 'A0002', nombre: 'Termodinamica', creditos: 9 }
  }
]
Profesores>

```

Obtener todos los que den asignaturas de más de 6 créditos y mayores de 40.

```

C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({"asignatura.creditos": {$gt: 6}, "edad": {$gt: 40}})
[
  {
    _id: ObjectId("636388532e434ce09a427994"),
    nombre: 'Antonio',
    apellidos: 'Munguia Arteche',
    especialidad: [ 'fisica', 'quimica' ],
    esTitular: true,
    esAsociado: false,
    edad: 54,
    asignatura: { id: 'A0002', nombre: 'Termodinamica', creditos: 9 }
  }
]
Profesores>

```


Trabajando con Profesores y vehículos

Añadir propiedades (vehículo) y registros a la colección Profesores

```
{ "nombre": "María", "apellidos": "Suárez Manrique", "especialidad":  
[ "biología" ],
```

```
"esTitular": true, "esAsociado": false, "vehiculo": { "matricula":  
"1111ABC", "tipo": "SUV",
```

```
"marca": "Lexus", "caballos": 190, "uso": 3 } },
```

```
{ "nombre": "Jose Luis", "apellidos": "López Pérez", "especialidad": [ "matemáticas",
```

```
"física" ], "esTitular": true, "esAsociado": false, "vehiculo": {  
"matricula": "2222DEF",
```

```
"tipo": "SUV", "marca": "Toyota", "caballos": 120, "uso": 13 } },
```

```
{ "nombre": "Antonio", "apellidos": "Munguía Arteche",
```

```
"especialidad": [ "física, química"
```

```
], "esTitular": true, "esAsociado": false, "vehiculo": { "matricula":  
"3333HIJ", "tipo":
```

```
"turismo", "marca": "volvo", "caballos": 176, "uso": 6 } },
```

```
{ "nombre": "Fernando", "apellidos": "Delgado De La Fuente",  
"especialidad": [ "física" ],
```

```
"esTitular": false, "esAsociado": true, "vehiculo": { "matricula":  
"4444KLM", "tipo": "moto",
```

```
"marca": "Yamaha", "caballos": 75, "uso": 8 } },
```

```
{ "nombre": "Elena", "apellidos": "Hérendez Serafín",
```

```
"especialidad": [ "matemáticas",
```

```
"física" ], "esTitular": true, "esAsociado": false, "vehiculo": {  
"matricula": "5555NOP",
```

```
"tipo": "moto", "marca": "honda", "caballos": 170, "uso": 4 } }
```

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Profesores> db.Profesores.insert({ "nombre": "María", "apellidos": "Suárez Manrique", "especialidad": [ "biología" ], "esTitular": true, "esAsociado": false, "vehículo": { "matricula": "1111ABC", "tipo": "SUV", "marca": "Lexus", "caballos": 190, "uso": 3 } })
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63639e6e2e434ce09a42799a") }
}
Profesores> db.Profesores.insert({ "nombre": "Jose Luis", "apellidos": "López Pérez", "especialidad": [ "matemáticas", "física" ], "esTitular": true, "esAsociado": false, "vehículo": { "matricula": "2222DEF", "tipo": "SUV", "marca": "Toyota", "caballos": 120, "uso": 13 } })
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63639e7f2e434ce09a42799b") }
}
Profesores> db.Profesores.insert({ "nombre": "Antonio", "apellidos": "Munguía Arteché", "especialidad": [ "física", "química" ], "esTitular": true, "esAsociado": false, "vehículo": { "matricula": "3333HIJ", "tipo": "turismo", "marca": "volvo", "caballos": 176, "uso": 6 } })
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63639e922e434ce09a42799c") }
}
Profesores> db.Profesores.insert({ "nombre": "Fernando", "apellidos": "Delgado De La Fuente", "especialidad": [ "física" ], "esTitular": false, "esAsociado": true, "vehículo": { "matricula": "4444KLM", "tipo": "moto", "marca": "Yamaha", "caballos": 75, "uso": 8 } })
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63639ea02e434ce09a42799d") }
}
Profesores> db.Profesores.insert({ "nombre": "Elena", "apellidos": "Hérendez Serafín", "especialidad": [ "matemáticas", "física" ], "esTitular": true, "esAsociado": false, "vehículo": { "matricula": "5555NOP", "tipo": "moto", "marca": "honda", "caballos": 170, "uso": 4 } })
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63639eac2e434ce09a42799e") }
}

```

Consultas a realizar:

- Obtener todos los profesores que tienen un SUV.

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
Profesores> db.Profesores.find({"vehiculo.tipo":"SUV"})
[
  {
    _id: ObjectId("63639e6e2e434ce09a42799a"),
    nombre: 'María',
    apellidos: 'Suárez Manrique',
    especialidad: [ 'biología' ],
    esTitular: true,
    esAsociado: false,
    vehiculo: {
      matricula: '1111ABC',
      tipo: 'SUV',
      marca: 'Lexus',
      caballos: 190,
      uso: 3
    }
  },
  {
    _id: ObjectId("63639e7f2e434ce09a42799b"),
    nombre: 'Jose Luis',
    apellidos: 'López Pérez',
    especialidad: [ 'matemáticas', 'fisica' ],
    esTitular: true,
    esAsociado: false,
    vehiculo: {
      matricula: '2222DEF',
      tipo: 'SUV',
      marca: 'Toyota',
      caballos: 120,
      uso: 13
    }
  }
]
Profesores> █
```

- Obtener todos los profesores que tienen una moto y un uso de menos de 5 años.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({"vehiculo.tipo":"moto", "vehiculo.uso":{"$lt: 5}})
[
  {
    _id: ObjectId("63639eac2e434ce09a42799e"),
    nombre: 'Elena',
    apellidos: 'Hé Hernández Serafín',
    especialidad: [ 'matemáticas', 'física' ],
    esTitular: true,
    esAsociado: false,
    vehiculo: {
      matricula: '5555NOP',
      tipo: 'moto',
      marca: 'honda',
      caballos: 170,
      uso: 4
    }
  }
]
Profesores> _
```

- Obtener todos los profesores con vehículo de más de 150 caballos y que impartan física.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Profesores> db.Profesores.find({"vehiculo.caballos":{"$gt: 150}, "especialidad":"física"})
[
  {
    _id: ObjectId("63639eac2e434ce09a42799e"),
    nombre: 'Elena',
    apellidos: 'Hé Hernández Serafín',
    especialidad: [ 'matemáticas', 'física' ],
    esTitular: true,
    esAsociado: false,
    vehiculo: {
      matricula: '5555NOP',
      tipo: 'moto',
      marca: 'honda',
      caballos: 170,
      uso: 4
    }
  }
]
Profesores> _
```

- Obtener el nº de profesores con más de 50 años, que tienen moto.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Profesores> db.Profesores.find({"edad": {$gt: 50}, "vehiculo.tipo":"moto"})
Profesores>
```

- Obtener el nº de profesores titulares que tienen un SUV.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Profesores> db.Profesores.find({"vehiculo.tipo":"SUV", "esTitular":true }).count()
2
Profesores>
```