

23/24

MongoDB



Andrés Morales González

IES Gonzalo Nazareno

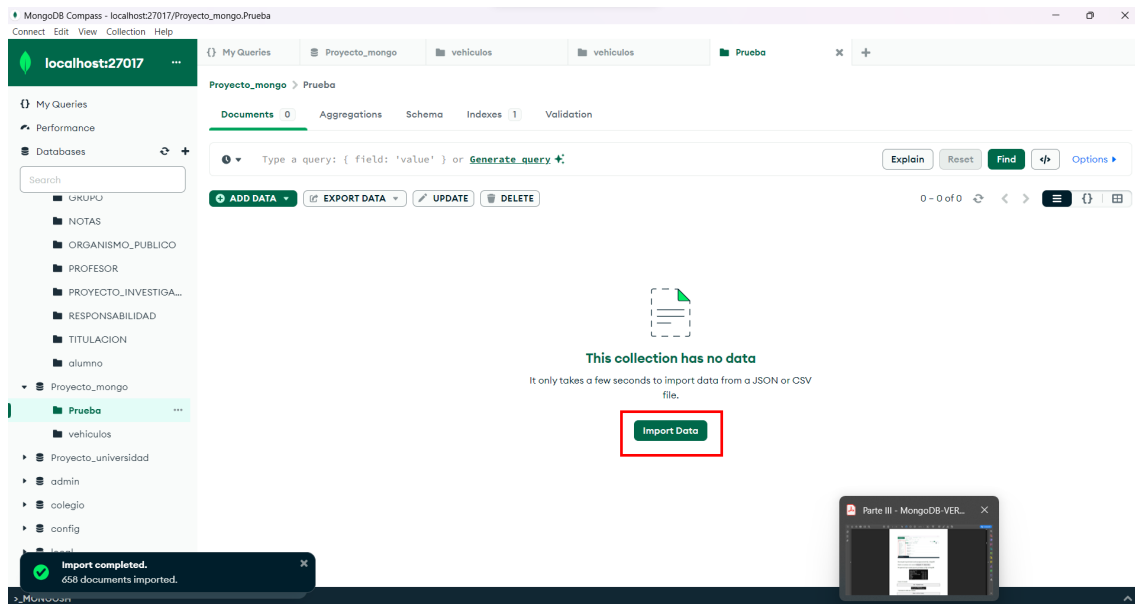
23/24

Contenido

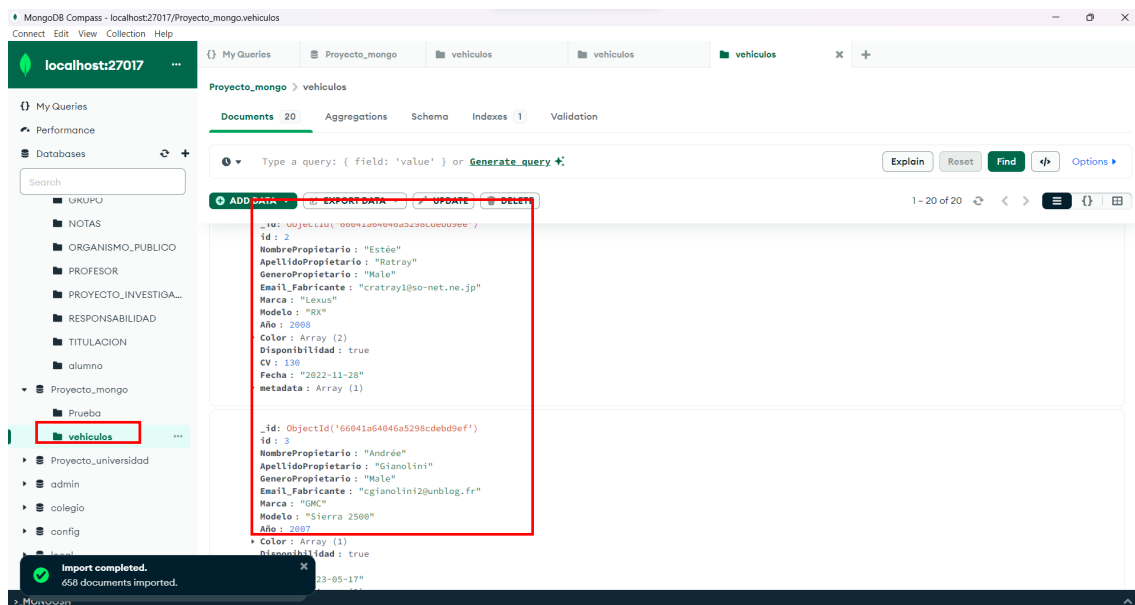
Usar JSON	2
Inserción de datos	3
Con insertOne	3
Con insertMany	4
Eliminación de datos	6
Con deleteOne	6
Con deleteMany	6
Actualizar datos	6
Con updateOne	6
Con updateMany	7
Con ReplaceOne	7
Consultas	8
Consultas simples	8
Consultas con Arrays	11
Consultas con documentos embebidos	15
Agrupación	17
Programa PYTHON	19

Usar JSON

Para esta practica lo que he hecho ha sido coger directamente el json y por entoprno grafico usarlo, tal que asi:



Haciendo un impor del json de vehículos, y si nos vamos a la colección del lateral izquierdo, el cual creamos, podemos observar que los tenemos en nuestra colección.



Inserción de datos

Con insertOne

Realizar una inserción de un **nuevo vehículo**.

```
db.vehiculos.insertOne({
  "id": 57,
  "NombrePropietario": "Edmée",
  "ApellidoPropietario": "Lister",
  "GeneroPropietario": "Male",
  "Email_Fabricante": "xlisterli@elpais.com",
  "Marca": "Subaru",
  "Modelo": "Legacy",
  "Año": 1996,
  "Color": ["Blue"],
  "Disponibilidad": false,
  "CV": 93,
  "metadata": {
    "sourceName": "Subaru",
    "sourceUrl": "https://www.subaru.es/"
  }
})
```

```
Proyecto_mongo> db.vehiculos.insertOne({
...   "id": 57,
...   "NombrePropietario": "Edmée",
...   "ApellidoPropietario": "Lister",
...   "GeneroPropietario": "Male",
...   "Email_Fabricante": "xlisterli@elpais.com",
...   "Marca": "Subaru",
...   "Modelo": "Legacy",
...   "Año": 1996,
...   "Color": ["Blue"],
...   "Disponibilidad": false,
...   "CV": 93,
...   "metadata": {
...     "sourceName": "Subaru",
...     "sourceUrl": "https://www.subaru.es/"
...   }
... })
{
  acknowledged: true,
  insertedId: ObjectId('66041c569e7448127246efe3')
}
Proyecto_mongo>
```

Con insertMany

Realizaremos la inserción de **varios vehículos**.

```
db.vehiculos.insertMany([
  {
    "id": 77,
    "NombrePropietario": "Carolina",
    "ApellidoPropietario": "González",
    "GeneroPropietario": "Female",
    "Email_Fabricante": "cgonzalez@example.com",
    "Marca": "Toyota",
    "Modelo": "Corolla",
    "Año": 2019,
    "Color": ["Azul"],
    "Disponibilidad": true,
    "CV": 150,
    "Fecha": "2024-03-27",
    "metadata": {
      "sourceName": "Toyota",
      "sourceUrl": "https://www.toyota.es/"
    }
  },
  {
    "id": 96,
    "NombrePropietario": "Juan",
    "ApellidoPropietario": "Martínez",
    "GeneroPropietario": "Male",
    "Email_Fabricante": "jmartinez@example.com",
    "Marca": "Honda",
    "Modelo": "Civic",
    "Año": 2020,
    "Color": ["Rojo"],
    "Disponibilidad": true,
    "CV": 180,
    "Fecha": "2024-03-27",
    "metadata": {
      "sourceName": "Honda",
      "sourceUrl": "https://www.honda.es"
    }
  }
]);
```

```
Proyecto_mongo> db.vehiculos.insertMany([
...   {
...     "id": 77,
...     "NombrePropietario": "Carolina",
...     "ApellidoPropietario": "González",
...     "GeneroPropietario": "Female",
...     "Email_Fabricante": "cgonzalez@example.com",
...     "Marca": "Toyota",
...     "Modelo": "Corolla",
...     "Año": 2019,
...     "Color": ["Azul"],
...     "Disponibilidad": true,
...     "CV": 150,
...     "Fecha": "2024-03-27",
...     "metadata": {
...       "sourceName": "Toyota",
...       "sourceUrl": "https://www.toyota.es/"
...     }
...   },
...   {
...     "id": 96,
...     "NombrePropietario": "Juan",
...     "ApellidoPropietario": "Martínez",
...     "GeneroPropietario": "Male",
...     "Email_Fabricante": "jmartinez@example.com",
...     "Marca": "Honda",
...     "Modelo": "Civic",
...     "Año": 2020,
...     "Color": ["Rojo"],
...     "Disponibilidad": true,
...     "CV": 180,
...     "Fecha": "2024-03-27",
...     "metadata": {
...       "sourceName": "Honda",
...       "sourceUrl": "https://www.honda.es"
...     }
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66041df39e7448127246efe4'),
    '1': ObjectId('66041df39e7448127246efe5')
  }
}
Proyecto_mongo> |
```

Eliminacion de datos

Eliminación de varios documentos utilizando los dos métodos de eliminación de MongoDB.

Con deleteOne:

Elimina el vehículo con modelo “Bonneville”

```
db.vehiculos.deleteOne({"Modelo": "Bonneville"})
```

```
Proyecto_mongo> db.vehiculos.deleteOne({"Modelo": "Bonneville"})
{ acknowledged: true, deletedCount: 1 }
Proyecto_mongo> |
```

Con deleteMany

Como en este caso el que más se repite es Suzuki, lo vamos a eliminar.

```
db.vehiculos.deleteMany({"Marca": "Suzuki"})
```

```
Proyecto_mongo> db.vehiculos.deleteMany({"Marca": "Suzuki"})
{ acknowledged: true, deletedCount: 2 }
Proyecto_mongo> |
```

Actualizar datos

Actualizar varios documentos usando los tres métodos de eliminación de MngoDB

Con updateOne

Actualizaremos el vehículo que cuente con 103 CV y cambiar la disponibilidad a true

```
db.vehiculos.updateOne({"CV": 103},{$set:{"Disponibilidad": true}})
```

```
{ acknowledged: true, deletedCount: 1 }
Proyecto_mongo> db.vehiculos.updateOne({"CV": 103},{$set:{"Disponibilidad": true}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Proyecto_mongo> |
```

Con updateMany

Vamos a actualizar en la que todo los vehículos pertenezcan de un año superior a 2008

Y que su potencia sea 120

```
db.vehiculos.updateMany({Año: { $gt: 2008 }}, {$set: {"CV": 120}})
```

```
Proyecto_mongo> db.vehiculos.updateMany({
...   Año: { $gt: 2008 }
... }, {
...   $set: {
...     "CV": 120
...   }
... })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0
}
Proyecto_mongo> |
```

Con ReplaceOne

Reemplaza el modelo por “Escalade” en el vehículo que tenga “Cadillac” en el nombre de la fuente de sus metadatos

```
db.vehiculos.replaceOne({"metadata.sourceName": "Cadillac"}, {"Modelo": "Escalade"})
```

```
Proyecto_mongo> db.vehiculos.replaceOne({"metadata.sourceName": "Cadillac"}, {"Modelo": "Escalade"})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Proyecto_mongo>
```


Consultas

Consultas simples

Muestra los vehículos que no están disponible

```
db.vehiculos.find({"Disponibilidad": false}).pretty()
```

```
proyecto_mongo> db.vehiculos.find({"Disponibilidad": false}).pretty()
{
  _id: ObjectId('66041a64046a5298cdebd9f2'),
  id: 6,
  NombrePropietario: 'Mélanie',
  ApellidoPropietario: 'Hedon',
  GeneroPropietario: 'Male',
  Email_Fabricante: 'fhedon5@goo.gl',
  Marca: 'Saturn',
  Modelo: 'S-Series',
  Año: 1993,
  Color: [ 'Verde', 'Azul', 'Naranja', 'Negro' ],
  Disponibilidad: false,
  CV: 125,
  Fecha: '2022-12-30',
  metadata: [
    {
      sourceName: 'Wikipedia',
      sourceUrl: 'https://es.wikipedia.org/wiki/Saturn_Corporation'
    }
  ]
},
{
  _id: ObjectId('66041a64046a5298cdebd9f3'),
  id: 7,
  NombrePropietario: 'Faites',
  ApellidoPropietario: 'Habin',
  GeneroPropietario: 'Male',
  Email_Fabricante: 'bhabin6@51.la',
  Marca: 'MINI',
  Modelo: 'Clubman',
  Año: 2010,
  Color: [ 'Negro', 'Azul', 'Naranja', 'Blanco' ],
  Disponibilidad: false,
  CV: 120,
  Fecha: '2023-07-21',
  metadata: [ { sourceName: 'MINI', sourceUrl: 'https://www.mini.es/' } ]
},
{
  _id: ObjectId('66041a64046a5298cdebd9f4'),
  id: 8,
  NombrePropietario: 'Léana',
  ApellidoPropietario: 'Aubrey',
  GeneroPropietario: 'Female',
  Email_Fabricante: 'daubrey7@hostgator.com',
  Marca: 'Mercury',
  Modelo: 'Sable',
  Año: 2008,
  Color: [ 'Negro', 'Verde', 'Gris' ],
  Disponibilidad: false,
  CV: 119,
  Fecha: '2023-03-09',
  metadata: [
    {
      sourceName: 'Wikipedia',
      sourceUrl: 'https://es.wikipedia.org/wiki/Mercury_(autom%C3%B3viles)'
    }
  ]
},
{
  _id: ObjectId('66041a64046a5298cdebd9f6'),
  id: 10,
  NombrePropietario: 'Nadège',
  ApellidoPropietario: 'Gatfield',
  GeneroPropietario: 'Male',
  Email_Fabricante: 'hgatfield9@tinyurl.com',
  Marca: 'Nissan',
  Modelo: 'Versa',
  Año: 2011,
  Color: [ 'Lima', 'Gris' ],
  Disponibilidad: false,
  CV: 120,
  Fecha: '2023-04-18',
  metadata: [ { sourceName: 'Nissan', sourceUrl: 'https://www.nissan.es/' } ]
},
{
  _id: ObjectId('66041a64046a5298cdebd9f7'),
  id: 11,
  NombrePropietario: 'Josée',
  ApellidoPropietario: 'Achrameev',
  GeneroPropietario: 'Male',
  Email_Fabricante: 'aachrameeva@4shared.com',
  Marca: 'Infiniti',
  Modelo: 'QX',
  Año: 2008,
  Color: [ 'Negro' ],
  Disponibilidad: false,
  CV: 85,
  Fecha: '2022-08-07',
  metadata: [
    {
      sourceName: 'Infiniti',
      sourceUrl: 'https://www.infiniti.com/'
    }
  ]
}
```

Muestra los vehículos de la marca Audi

```
db.vehiculos.find({"Marca": "Audi"}).pretty()
```

```
Proyecto_mongo> db.vehiculos.find({"Marca": "Audi"}).pretty()
[
  {
    _id: ObjectId('66041a64046a5298cdebd9fb'),
    id: 15,
    NombrePropietario: 'Maëlle',
    ApellidoPropietario: 'Rubel',
    GeneroPropietario: 'Male',
    Email_Fabricante: 'wrubela@columbia.edu',
    Marca: 'Audi',
    Modelo: 'RS6',
    Año: 2003,
    Color: [ 'Marrón', 'Azul', 'Gris', 'Plata' ],
    Disponibilidad: true,
    CV: 86,
    Fecha: '2023-11-27',
    metadata: [ { sourceName: 'Audi', sourceUrl: 'https://www.audi.es' } ]
  }
]
Proyecto_mongo> |
```

Muestra los vehículos con menos de 90 cv

```
db.vehiculos.find({"CV":{$lt:90}}).pretty()
```

```
Proyecto_mongo> db.vehiculos.find({"CV":{$lt:90}}).pretty()
[
  {
    _id: ObjectId('66041a64046a5298cdebd9fb'),
    id: 4,
    NombrePropietario: 'Judicaël',
    ApellidoPropietario: 'Folbigg',
    GeneroPropietario: 'Male',
    Email_Fabricante: 'kfolbigg3@tumblr.com',
    Marca: 'Lexus',
    Modelo: 'LX',
    Año: 1999,
    Color: [ 'Fuscia', 'Amarillo', 'Naranja', 'Rojo' ],
    Disponibilidad: true,
    CV: 83,
    Fecha: '2022-09-02',
    metadata: [ { sourceName: 'Lexus', sourceUrl: 'https://www.lexusauto.es/' } ]
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9fb'),
    id: 9,
    NombrePropietario: 'Céline',
    ApellidoPropietario: 'Pellew',
    GeneroPropietario: 'Female',
    Email_Fabricante: 'apellew8@google.pl',
    Marca: 'GMC',
    Modelo: '1500 Club Coupe',
    Año: 1995,
    Color: [ 'Negro', 'Gris' ],
    Disponibilidad: true,
    CV: 88,
    Fecha: '2022-10-11',
    metadata: [ { sourceName: 'GMC', sourceUrl: 'https://www.gmc.com/' } ]
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9fb'),
    id: 11,
    NombrePropietario: 'Josée',
    ApellidoPropietario: 'Achrameev',
    GeneroPropietario: 'Male',
    Email_Fabricante: 'aachrameeva@4shared.com',
    Marca: 'Infiniti',
    Modelo: 'QX',
    Año: 2008,
    Color: [ 'Negro' ],
    Disponibilidad: false,
    CV: 85,
    Fecha: '2022-08-07',
    metadata: [
      {
        sourceName: 'Infiniti',
        sourceUrl: 'https://www.infiniti.com/'
      }
    ]
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9fb'),
    id: 15,
    NombrePropietario: 'Maëlle',
    ApellidoPropietario: 'Rubel',
    GeneroPropietario: 'Male',
    Email_Fabricante: 'wrubela@columbia.edu',
    Marca: 'Audi',
    Modelo: 'RS6',
    Año: 2003,
    Color: [ 'Marrón', 'Azul', 'Gris', 'Plata' ],
    Disponibilidad: true,
    CV: 86,
    Fecha: '2023-11-27',
    metadata: [ { sourceName: 'Audi', sourceUrl: 'https://www.audi.es' } ]
  }
]
Proyecto_mongo> |
```

Muestra el número de vehículos de la marca GMC con menos de 120 cv

```
db.vehiculos.find({"Marca": "GMC", "CV": {"$lt": 120}}).count()
```

```
Proyecto_mongo> db.vehiculos.find({"Marca": "GMC", "CV": {"$lt": 120}}).count()
1
Proyecto_mongo> |
```

Ya que el total de vehículos que tengo de GMC esde tres, con la consulta:

```
db.vehiculos.find({"Marca": "GMC"}).count()
```

```
Proyecto_mongo> db.vehiculos.find({"Marca": "GMC"}).count()
3
Proyecto_mongo>
```

Muestra la marca y el modelo en los que los emails de fabricantes terminen por .com.

```
db.vehiculos.find({"Email_Fabricante": {"$regex": "\.com$"}}, {"Marca": 1,
"Modelo": 1, "_id": 0});
```

```
Proyecto_mongo> db.vehiculos.find({"Email_Fabricante": {"$regex": "\.com$"}}, {"Marca": 1, "Modelo": 1, "_id": 0});
[
  { Marca: 'GMC', Modelo: 'Yukon XL 2500' },
  { Marca: 'Lexus', Modelo: 'LX' },
  { Marca: 'Mercury', Modelo: 'Sable' },
  { Marca: 'Nissan', Modelo: 'Versa' },
  { Marca: 'Infiniti', Modelo: 'QX' },
  { Marca: 'Mercedes-Benz', Modelo: 'M-Class' },
  { Marca: 'Isuzu', Modelo: 'Ascender' },
  { Marca: 'Oldsmobile', Modelo: 'Bravada' },
  { Marca: 'Mazda', Modelo: 'Familia' },
  { Marca: 'Subaru', Modelo: 'Legacy' },
  { Marca: 'Toyota', Modelo: 'Corolla' },
  { Marca: 'Honda', Modelo: 'Civic' }
]
```

Consultas con Arrays

Muestra todos los vehiculos que tengan el color “Azul”

```
db.vehiculos.find({"Color":{"$in":["Azul"]}}).sort({"id":1}).pretty()
```

```
Proyecto_mongo> db.vehiculos.find({"Color": "Azul"}).sort({"id": 1}).pretty()
[
  {
    _id: ObjectId('66041a64046a5298cdebd9f2'),
    id: 6,
    NombrePropietario: 'Mélanie',
    ApellidoPropietario: 'Hedon',
    GeneroPropietario: 'Male',
    Email_Fabricante: 'fhedon5@goo.gl',
    Marca: 'Saturn',
    Modelo: 'S-Series',
    'Año': 1993,
    Color: [ 'Verde', 'Azul', 'Naranja', 'Negro' ],
    Disponibilidad: false,
    CV: 125,
    Fecha: '2022-12-30',
    metadata: [
      {
        sourceName: 'Wikipedia',
        sourceUrl: 'https://es.wikipedia.org/wiki/Saturn_Corporation'
      }
    ]
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9f3'),
    id: 7,
    NombrePropietario: 'Faïtes',
    ApellidoPropietario: 'Habin',
    GeneroPropietario: 'Male',
    Email_Fabricante: 'bhabin6@51.la',
    Marca: 'MINI',
    Modelo: 'Clubman',
    'Año': 2010,
    Color: [ 'Negro', 'Azul', 'Naranja', 'Blanco' ],
    Disponibilidad: false,
    CV: 120,
    Fecha: '2023-07-21',
    metadata: [ { sourceName: 'MINI', sourceUrl: 'https://www.mini.es/' } ]
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9f9'),
    id: 13,
    NombrePropietario: 'Régine',
    ApellidoPropietario: 'Troni',
    GeneroPropietario: 'Female',
    Email_Fabricante: 'utronic@miitbeian.gov.cn',
    Marca: 'Plymouth',
    Modelo: 'Colt',
    'Año': 1993,
    Color: [ 'Naranja', 'Azul', 'Amarillo', 'Gris' ],
    Disponibilidad: true,
    CV: 101,
    Fecha: '2023-09-14',
    metadata: [
      {
        sourceName: 'Wikipedia',
        sourceUrl: 'https://es.wikipedia.org/wiki/Plymouth_(autom%C3%B3viles)'
      }
    ]
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9fb'),
    id: 15,
    NombrePropietario: 'Maëlle',
    ApellidoPropietario: 'Rubel',
    GeneroPropietario: 'Male',
    Email_Fabricante: 'wrubele@columbia.edu',
    Marca: 'Audi',
  }
]
```

```

{
  _id: ObjectId('66041a64046a5298cdebd9fb'),
  id: 15,
  NombrePropietario: 'Maëlle',
  ApellidoPropietario: 'Rubel',
  GeneroPropietario: 'Male',
  Email_Fabricante: 'wrubele@columbia.edu',
  Marca: 'Audi',
  Modelo: 'RS6',
  'Año': 2003,
  Color: [ 'Marrón', 'Azul', 'Gris', 'Plata' ],
  Disponibilidad: true,
  CV: 86,
  Fecha: '2023-11-27',
  metadata: [ { sourceName: 'Audi', sourceUrl: 'https://www.audi.es' } ]
},
{
  _id: ObjectId('66041a64046a5298cdebda00'),
  id: 20,
  NombrePropietario: 'Mélinda',
  ApellidoPropietario: 'Pistol',
  GeneroPropietario: 'Female',
  Email_Fabricante: 'gpistolj@1688.com',
  Marca: 'Mazda',
  Modelo: 'Familia',
  'Año': 1988,
  Color: [ 'Azul', 'Negro', 'Blanco' ],
  Disponibilidad: false,
  CV: 126,
  Fecha: '2023-03-28',
  metadata: [ { sourceName: 'Mazda', sourceUrl: 'https://www.mazda.es/' } ]
},
{
  _id: ObjectId('66041df39e744812724efe4'),
  id: 77,
  NombrePropietario: 'Carolina',
  ApellidoPropietario: 'González',
  GeneroPropietario: 'Female',
  Email_Fabricante: 'cgonzalez@example.com',
  Marca: 'Toyota',
  Modelo: 'Corolla',
  'Año': 2019,
  Color: [ 'Azul' ],
  Disponibilidad: true,
  CV: 120,
  Fecha: '2024-03-27',
  metadata: { sourceName: 'Toyota', sourceUrl: 'https://www.toyota.es/' }
}

```

Si esto lo hiciéramos mediante una agrupación de marca y modelo, solamente, tendríamos:

```

db.vehiculos.aggregate([
  {
    $match: { "Color": "Azul" }
  },
  {
    $group: {
      _id: { Marca: "$Marca", Modelo: "$Modelo" },
      total: { $sum: 1 }
    }
  },
  {
    $project: {
      _id: 0,
      Marca: "$_id.Marca",
      Modelo: "$_id.Modelo",
      Total: "$total"
    }
  }
])

```

```

    }
  }
});

Proyecto_mongo> db.vehiculos.aggregate([
...   {
...     $match: { "Color": "Azul" }
...   },
...   {
...     $group: {
...       _id: { Marca: "$Marca", Modelo: "$Modelo" },
...       total: { $sum: 1 }
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       Marca: "$_id.Marca",
...       Modelo: "$_id.Modelo",
...       Total: "$total"
...     }
...   }
... ]);
[
  { Marca: 'MINI', Modelo: 'Clubman', Total: 1 },
  { Marca: 'Plymouth', Modelo: 'Colt', Total: 1 },
  { Marca: 'Audi', Modelo: 'RS6', Total: 1 },
  { Marca: 'Mazda', Modelo: 'Familia', Total: 1 },
  { Marca: 'Toyota', Modelo: 'Corolla', Total: 1 },
  { Marca: 'Saturn', Modelo: 'S-Series', Total: 1 }
]

```

Mostrar los vehículos que tengan exactamente los colores marron, azul, gris y plata

```
db.vehiculos.find({"Color": ["Marrón", "Azul", "Gris", "Plata"]}).pretty()
```

```

Proyecto_mongo> db.vehiculos.find({"Color": ["Marrón", "Azul", "Gris", "Plata"]}).pretty()
[
  {
    _id: ObjectId('66041a64046a5298cdebd9fb'),
    id: 15,
    NombrePropietario: 'Maëlle',
    ApellidoPropietario: 'Rubel',
    GeneroPropietario: 'Male',
    Email_Fabricante: 'wrubele@columbia.edu',
    Marca: 'Audi',
    Modelo: 'RS6',
    'Año': 2003,
    Color: [ 'Marrón', 'Azul', 'Gris', 'Plata' ],
    Disponibilidad: true,
    CV: 86,
    Fecha: '2023-11-27',
    metadata: [ { sourceName: 'Audi', sourceUrl: 'https://www.audi.es' } ]
  }
]
Proyecto_mongo> |

```

Muestra los vehículos que tenga los colores "Negro" y "Gris". Pueden tener adicionalmente más colores, pero esos 2 son obligatorios.

```
db.vehiculos.find({"Color":{"$all": ["Negro","Gris"]}}).pretty()
```

```

Proyecto_mongo> db.vehiculos.find({"Color":{"$all": ["Negro","Gris"]}}).pretty()
[
  {
    _id: ObjectId('66041a64046a5298cdebd9f4'),
    id: 8,
    NombrePropietario: 'Léana',
    ApellidoPropietario: 'Aubrey',
    GeneroPropietario: 'Female',
    Email_Fabricante: 'daubrey7@hostgator.com',
    Marca: 'Mercury',
    Modelo: 'Sable',
    'Año': 2008,
    Color: [ 'Negro', 'Verde', 'Gris' ],
    Disponibilidad: false,
    CV: 119,
    Fecha: '2023-03-09',
    metadata: [
      {
        sourceName: 'Wikipedia',
        sourceUrl: 'https://es.wikipedia.org/wiki/Mercury_(autom%C3%B3viles)'
      }
    ]
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9f5'),
    id: 9,
    NombrePropietario: 'Céline',
    ApellidoPropietario: 'Pellew',
    GeneroPropietario: 'Female',
    Email_Fabricante: 'apellew8@google.pl',
    Marca: 'GMC',
    Modelo: '1500 Club Coupe',
    'Año': 1995,
    Color: [ 'Negro', 'Gris' ],
    Disponibilidad: true,
    CV: 88,
    Fecha: '2022-10-11',
    metadata: [ { sourceName: 'GMC', sourceUrl: 'https://www.gmc.com/' } ]
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9ff'),
    id: 19,
    NombrePropietario: 'Jú',
    ApellidoPropietario: 'Peplaw',
    GeneroPropietario: 'Male',
    Email_Fabricante: 'mpeplawi@weebly.com',
    Marca: 'Oldsmobile',
    Modelo: 'Bravada',
    'Año': 2004,
    Color: [ 'Negro', 'Blanco', 'Gris' ],
    Disponibilidad: true,
    CV: 145,
    Fecha: '2023-06-07',
    metadata: [
      {
        sourceName: 'Wikipedia',
        sourceUrl: 'https://es.wikipedia.org/wiki/Oldsmobile'
      }
    ]
  }
]
Proyecto_mongo> |

```

Consultas con documentos embebidos

Mostrar los nombres de los propietarios de los vehículos cuyos metadatos incluyan fuentes que contengan la palabra 'Wikipedia' en cualquier parte de su nombre.

```
db.vehiculos.find({"metadata.sourceName": {$regex: /Wikipedia/i}},
                  {"NombrePropietario": 1}).pretty()
```

```
Proyecto_mongo> db.vehiculos.find({"metadata.sourceName": {$regex: /Wikipedia/i}}, {"NombrePropietario": 1}).pretty()
[
  {
    _id: ObjectId('66041a64046a5298cdebd9f2'),
    NombrePropietario: 'Mélanie'
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9f4'),
    NombrePropietario: 'Léana'
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9f9'),
    NombrePropietario: 'RéGINE'
  },
  {
    _id: ObjectId('66041a64046a5298cdebd9ff'),
    NombrePropietario: 'Jú'
  }
]
Proyecto_mongo> |
```

"Muestra el modelo y la disponibilidad de los vehículos de la marca 'GMC' que están disponibles en el inventario."

```
db.vehiculos.find({"Marca": "GMC", "Disponibilidad": true}, {"Modelo": 1,
                  "Disponibilidad": 1, "_id": 0}).pretty()
```

```
Proyecto_mongo> db.vehiculos.find({"Marca": "GMC", "Disponibilidad": true}, {"Modelo": 1, "Disponibilidad": 1, "_id": 0}).pretty()
[
  { Modelo: 'Yukon XL 2500', Disponibilidad: true },
  { Modelo: 'Sierra 2500', Disponibilidad: true },
  { Modelo: '1500 Club Coupe', Disponibilidad: true }
]
```

Muestra la marca, modelo y CV de los vehículos fabricados después de 2010 y registrados entre el 1 de enero de 2019 y el 31 de diciembre de 2022.

```
db.vehiculos.find({
  "Año": { $gt: 2010 },
  "Fecha": {
    $gte: "2019-01-01",
    $lte: "2022-12-31"
  }
}, {
  "Marca": 1,
  "Modelo": 1,
  "CV": 1,
  "Fecha": 1,
  "_id": 0
}).pretty()
```



```
Proyecto_mongo> db.vehiculos.find({
...   "Año": { $gt: 2010 },
...   "Fecha": {
...     $gte: "2019-01-01",
...     $lte: "2022-12-31"
...   }
... }, {
...   "Marca": 1,
...   "Modelo": 1,
...   "CV": 1,
...   "Fecha": 1,
...   "_id": 0
... }).pretty()
[ { Marca: 'Lexus', Modelo: 'RX', CV: 120, Fecha: '2022-11-28' } ]
Proyecto_mongo> |
```

Agrupación

Obtener el número de modelos por marca de vehículos fabricados después de 2010 y que estén disponibles para la venta. Además, calcular el total de modelos y el promedio de caballos de fuerza (CV) por marca. Los resultados deben estar ordenados por el número de modelos en orden descendente

```
db.vehiculos.aggregate([
  {
    $match: {
      "Año": { $gt: 2010 },
      "Disponibilidad": true
    }
  },
  {
    $group: {
      _id: "$Marca",
      "N° de modelos": { $sum: 1 },
      "CV total": { $sum: "$CV" }
    }
  },
  {
    $project: {
      _id: 0,
      Marca: "$_id",
      "N° de modelos": 1,
      "CV promedio": { $avg: "$CV total" }
    }
  },
  {
    $sort: { "N° de modelos": -1 }
  }
]).pretty()
```

```
Proyecto_mongo> db.vehiculos.aggregate([
...   {
...     $match: {
...       "Año": { $gt: 2010 },
...       "Disponibilidad": true
...     }
...   },
...   {
...     $group: {
...       _id: "$Marca",
...       "Nº de modelos": { $sum: 1 },
...       "CV total": { $sum: "$CV" }
...     }
...   },
...   {
...     $project: {
...       _id: 0,
...       Marca: "$_id",
...       "Nº de modelos": 1,
...       "CV promedio": { $avg: "$CV total" }
...     }
...   },
...   {
...     $sort: { "Nº de modelos": -1 }
...   }
... ]).pretty()
[
  { 'Nº de modelos': 1, Marca: 'Lexus', 'CV promedio': 120 },
  { 'Nº de modelos': 1, Marca: 'Toyota', 'CV promedio': 120 },
  { 'Nº de modelos': 1, Marca: 'Honda', 'CV promedio': 120 }
]
Proyecto_mongo> |
```

Programa PYTHON

En esta ocasión seguiremos con la temática de conexión desde Python hacia MongoDB, para ello lo que haremos será lo siguiente:

```
import pymongo
```

Y una vez hecho esto, crearemos lo que son las variables y procederemos a conectarnos:

```
MONGO_HOST = "localhost"
MONGO_PUERTO = "27017"
MONGO_TIEMPO_FUERA = 1000
MONGO_URI = "mongodb://" + MONGO_HOST + ":" + MONGO_PUERTO + "/"
NOMBRE_BASE_DE_DATOS = "Proyecto_mongo" # Nombre de tu base de datos
NOMBRE_COLECCION = "vehiculos"
```

En este programa contamos con un menú en el cual podemos ver las siguientes opciones:

```
--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Elegir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 2
```

Procederemos a ir desde el 1 hasta al 6.

Opción 1:

En el cual vamos a insertar un nuevo documento, y nos pide por pantalla todos los datos, así procederemos, y para ver si se ha hecho correctamente le daremos a la opción 4, y así es, lo tenemos como lo hemos puesto.

```

--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Elegir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 1
Introduce el ID del vehículo: 56
Introduce el nombre del propietario: Concha
Introduce el apellido del propietario: De la Rosa
Introduce el género del propietario: Female
Introduce el email del fabricante: www.bugatti.com
Introduce la marca del vehículo: Bugatti
Introduce el modelo del vehículo: Chiron Super Sport 300
Introduce el año del vehículo: 2016
Introduce el color del vehículo (separado por comas si hay más de uno): azul, negro
Introduce la disponibilidad del vehículo (True/False): True
Introduce los CV del vehículo: 1500
Introduce la fecha (YYYY-MM-DD): 2017-06-30
Documento insertado con éxito.

--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Elegir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 4
Introduce la marca del vehículo: Bugatti
Todos los modelos de la marca Bugatti:
Modelo: Chiron Super Sport 300, Año: 2016, Fecha: 2017-06-30, Color: azul, negro

```

Opción 2:

Si metemos un modelo que no existe, nos dice lo siguiente:

```

--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Elegir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 2
Introduce el modelo del vehículo a eliminar: La ferrari
El modelo 'La ferrari' no existe en la base de datos. ¿Deseas intentarlo de nuevo? (s/n): s
Introduce el modelo del vehículo a eliminar: Malvavisco
El modelo 'Malvavisco' no existe en la base de datos. ¿Deseas intentarlo de nuevo? (s/n): n

```

Si metemos uno que existe:

```

--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Elegir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 2
Introduce el modelo del vehículo a eliminar: Chiron Super Sport 300
Se han eliminado 1 documentos con el modelo 'Chiron Super Sport 300' de la colección.

```

Opción 3:

Si metemos un año que no existe:

```
--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Eleguir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 3
Introduce el año de los vehículos a actualizar: 1875
Este año no está en su base de datos. Introduce un año que sí esté en su base de datos.
Introduce el año de los vehículos a actualizar: |
```

Si metemos un año que si existe, nos muestra por pantallam la marca y el modelo que ha sido actualizado:

```
--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Eleguir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 3
Introduce el año de los vehículos a actualizar: 1875
Este año no está en su base de datos. Introduce un año que sí esté en su base de datos.
Introduce el año de los vehículos a actualizar: 2008
Introduce la nueva potencia: 130
Se han modificado 7 vehículos con éxito.
Información de los vehículos actualizados:
Marca: Lexus, Modelo: RX
Marca: MINI, Modelo: Clubman
Marca: Mercury, Modelo: Sable
Marca: Nissan, Modelo: Versa
Marca: Infiniti, Modelo: QX
Marca: Toyota, Modelo: Corolla
Marca: Honda, Modelo: Civic
```

Opción 4:

Si existe la marca nos mostrara esto:

```
--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Eleguir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 4
Introduce la marca del vehículo: Nissan
Todos los modelos de la marca Nissan:
Modelo: Versa, Año: 2008, Fecha: 2023-04-18, Color: Lima, Gris
```

Si no existe:

```
--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Eleguir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 4
Introduce la marca del vehículo: ferrari
No se encontraron vehículos de la marca 'ferrari'. ¿Deseas intentarlo de nuevo? (s/n):
```

Opción 5:

Si no existe no idra que lo intentemos de nuevo, y si existe no dará la mara y el modelo

```
--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Eleguir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 5
Introduce el color de los vehículos que deseas consultar: Negro, azul
No se encontraron vehículos con el color 'Negro, azul'. Inténtalo de nuevo.

Introduce el color de los vehículos que deseas consultar: Negro
Vehículos con el color 'Negro':
Marca: Lexus, Modelo: RX
Marca: Saturn, Modelo: S-Series
Marca: MINI, Modelo: Clubman
Marca: Mercury, Modelo: Sable
Marca: GMC, Modelo: 1500 Club Coupe
Marca: Infiniti, Modelo: QX
Marca: Oldsmobile, Modelo: Bravada
Marca: Mazda, Modelo: Familia
```

Opción 6:

SI existe los años nos dará esto, marca , modelo y potencia

```
--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Eleguir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 6
Introduce el año de fabricación de los vehículos: 2008
Introduce la fecha de inicio (YYYY-MM-DD): 2023-12-01
Introduce la fecha de fin (YYYY-MM-DD): 2025-12-31
Vehículos fabricados en el año 2008 y registrados entre 2023-12-01 y 2025-12-31:
Marca: Toyota, Modelo: Corolla, CV: 130
Marca: Honda, Modelo: Civic, CV: 130
```

Si no existe el intervalo de años nos dira lo siguiente:

```
--- MENÚ ---
1. Insertar documento
2. Eliminar por modelo
3. Modificar potencia
4. Mostrarme todos los modelos de una marca
5. Elegir color
6. Consultar marca y modelo entre fechas de fabricación
0. Salir
Selecciona una opción: 6
Introduce el año de fabricación de los vehículos: 1777
Introduce la fecha de inicio (YYYY-MM-DD): 2016-01-01
Introduce la fecha de fin (YYYY-MM-DD): 2169-03-52
No se encontraron vehículos que cumplan con los criterios especificados. Inténtalo de nuevo.
Introduce el año de fabricación de los vehículos: █
```