

Práctica 02. Gestión de datos alojados en MongoAtlas

1. Obtener los datos (Dataset) [Mockaroo.com](https://mockaroo.com)

The screenshot shows the Mockaroo interface for generating test data. It displays a table of fields with their names, types, and options:

Field Name	Type	Options
<code>id</code>	Row Number	blank: 0% Σ X
<code>first_name</code>	First Name	blank: 0% Σ X
<code>last_name</code>	Last Name	blank: 0% Σ X
<code>email</code>	Email Address	blank: 0% Σ X
<code>gender</code>	Gender	blank: 0% Σ X
<code>ip_address</code>	IP Address v4	blank: 0% Σ X

Buttons at the bottom include '+ ADD ANOTHER FIELD', 'GENERATE FIELDS USING AI...', 'FORMAT', 'CSV', 'GENERATE DATA', 'PREVIEW', 'SAVE AS...', 'DERIVE FROM EXAMPLE...', and 'MORE'.

- a. Formato por defectos
 - b. En **Format**, selecciona **CSV**
 - c. Dale a **Download Data**. Tendrás un archivo llamado `MOCK_DATA.csv`.
2. Entrando en la colección podemos añadir el CSV en ADD DATA y vemos los datos como aparecen en la captura.

The screenshot shows the MongoDB Compass interface with the 'usuarios' collection selected. A red circle highlights the 'ADD DATA' button. The data grid shows several user documents imported from a CSV file. A message at the bottom indicates 'Import completed. 1000 documents imported.'

```
_id: ObjectId('606782ae7a991a09ffadicd82f')
id: 1
first_name: 'Niles'
last_name: 'Feltose'
email: 'nilesfeltose@google.com'
gender: 'Male'
ip_address: '233.116.183.187'

_id: ObjectId('606782ae7a991a09ffadicd830')
id: 2
first_name: 'Stanisla'
last_name: 'Orphicks'
email: 'stanislaorphicks@yandex.org'
gender: 'Male'
ip_address: '195.143.185.15'

_id: ObjectId('606782ae7a991a09ffadicd831')
id: 3
first_name: 'Loren'
last_name: 'Blessing'
email: 'lblessing@ops.gov'
gender: 'Female'
ip_address: '153.216.44.286'

_id: ObjectId('606782ae7a991a09ffadicd832')
id: 4
first_name: 'Vivian'
last_name: 'Baudinelli'
email: 'vbaudinelli@nature.com'
gender: 'Female'
ip_address: '77.93.32.111'

_id: ObjectId('606782ae7a991a09ffadicd833')
id: 5
name: 'Eric'
name: 'Bonden'
```

3. Abrir el terminal de mongosh
4. CONSULTAS
 - a. Listar las bases de datos que contiene el servidor

```
> show dbs
< ActividadMongo    8.00 KiB
  Practical      248.00 KiB
  admin          360.00 KiB
  local          4.98 GiB
> use Practical
< switched to db Practical
Atlas atlas-468z16-shard-0 [primary] Practical>
```

- b. Listar las colecciones de una de las bases de datos.

```
> show collections
< usuarios
```

- c. Crear y eliminar colecciones en una base de datos.

```
> db.createCollection("colección_test")
< { ok: 1 }

> db.colección_test.drop()
< true
```

- d. Consulta simple sobre una colección

```
> db.usuarios.find().limit(3)
< [
  {
    _id: ObjectId('696782e7a991a98fa01cd82f'),
    id: 1,
    first_name: 'Niles',
    last_name: 'Feltoe',
    email: 'nfeltoe0@google.com',
    gender: 'Male',
    ip_address: '253.116.143.167'
  },
  {
    _id: ObjectId('696782e7a991a98fa01cd830'),
    id: 2,
    first_name: 'Stanislaw',
    last_name: 'Phipps',
    email: 'sphipps1@dyndns.org',
    gender: 'Male',
    ip_address: '195.145.185.16'
  },
  {
    _id: ObjectId('696782e7a991a98fa01cd831'),
    id: 3,
    first_name: 'Leone',
    last_name: 'Bleesing',
    email: 'lbleesing2@nps.gov',
    gender: 'Female',
    ip_address: '153.216.44.206'
  }
]
```

e. Consulta con filtros sobre colección

```
> db.usuarios.find({ gender: "Female" })
< [
  {
    _id: ObjectId('696782e7a991a98fa01cd831'),
    id: 3,
    first_name: 'Leone',
    last_name: 'Bleesing',
    email: 'lbleesing2@nps.gov',
    gender: 'Female',
    ip_address: '153.216.44.206'
  },
  {
    _id: ObjectId('696782e7a991a98fa01cd837'),
    id: 9,
    first_name: 'Jerrine',
    last_name: 'Goodacre',
    email: 'jgoodacre8@icio.us',
    gender: 'Female',
    ip_address: '170.9.153.41'
  },
  {
    _id: ObjectId('696782e7a991a98fa01cd838'),
    id: 10,
    first_name: 'Hanni',
    last_name: 'Mallord',
    email: 'hmallord9@gmpg.org',
    gender: 'Female',
    ip_address: '91.159.154.221'
  }
]
```

f. Consulta con filtros y ordenación.

```
> db.usuarios.find({ gender: "Male" }).sort({ id: -1 }).limit(3)
< [
    {
        _id: ObjectId('696782e7a991a98fa01cdc12'),
        id: 996,
        first_name: 'Micheal',
        last_name: 'Wilshin',
        email: 'mwilshinrn@twitpic.com',
        gender: 'Male',
        ip_address: '171.43.29.156'
    },
    {
        _id: ObjectId('696782e7a991a98fa01cdc11'),
        id: 995,
        first_name: 'Verne',
        last_name: 'Josupeit',
        email: 'vjosupeitrm@marriott.com',
        gender: 'Male',
        ip_address: '19.167.136.4'
    },
    {
        _id: ObjectId('696782e7a991a98fa01cdc10'),
        id: 994,
        first_name: 'Tynan',
        last_name: 'Forkan',
        email: 'tforkanrl@sakura.ne.jp',
        gender: 'Male',
        ip_address: '56.145.173.154'
    }
]
```

- g. Consulta con filtros, ordenación y proyección.

```

> db.usuarios.find(
  { gender: "Female" },
  { first_name: 1, last_name: 1, _id: 0 }
).sort({ first_name: 1 }).limit(3)
< [
  {
    first_name: 'Abbi',
    last_name: 'McMurdo'
  },
  {
    first_name: 'Adelina',
    last_name: 'Cossington'
  },
  {
    first_name: 'Adelind',
    last_name: 'Veck'
  }
]

```

h. Investigación y consultas de agregación

```

> db.usuarios.aggregate([
  { $match: { id: { $gt: 10 } } },
  { $group: { _id: "$gender", total: { $sum: 1 } } },
  { $sort: { total: -1 } },
  { $limit: 3 } // Etapa de límite para agregación
])
< [
  {
    _id: 'Male',
    total: 444
  },
  {
    _id: 'Female',
    total: 441
  },
  {
    _id: 'Agender',
    total: 21
  }
]

```

i. Investigación sobre la integración de python con mongoDB.

```

# Ejemplo en el script de Python
cursor = colección.find({"gender": "Male"}).sort("id", -1).limit(3)
for doc in cursor:
    print(doc)

```