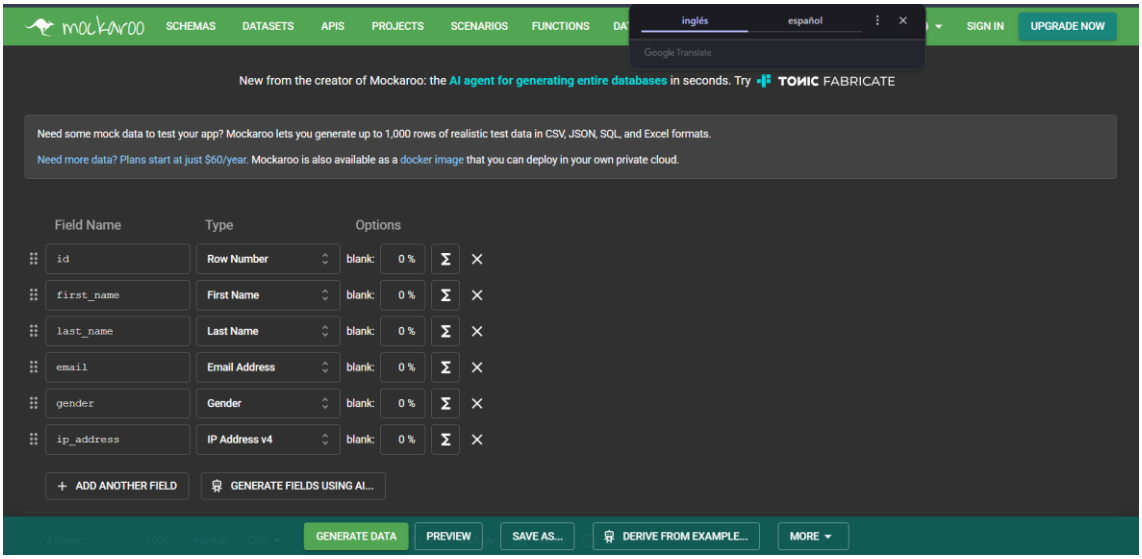
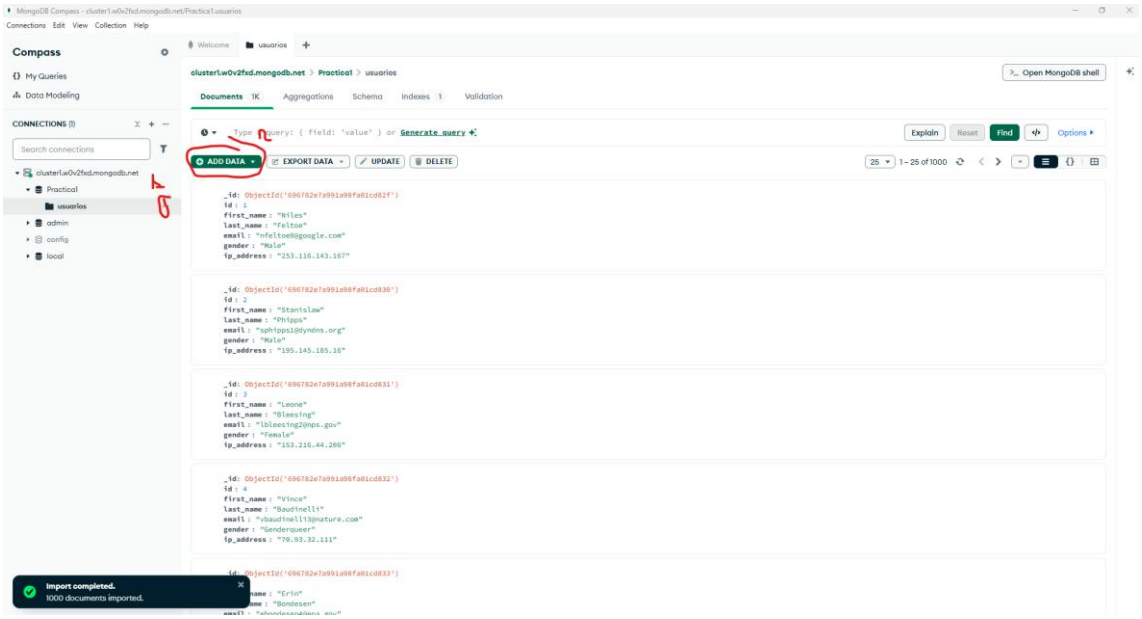
# Práctica 02. Gestión de datos alojados en MongoAtlas

1. Obtener los datos (Dataset) [Mockaroo.com](Mockaroo.com)




   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 captua.




3. Abrir el terminal de mongosh

4. CONSULTAS
   a. Listar las bases de datos que contiene el servidor

```
> show dbs
< ActividadMongo    8.00 KiB
  Practica1       248.00 KiB
  admin           360.00 KiB
  local             4.98 GiB
> use Practica1
< switched to db Practica1
Atlas atlas-468zl6-shard-0 [primary] Practica1>
```

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("coleccion_test")
< { ok: 1 }
```

```
> db.coleccion_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.


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 = coleccion.find({"gender": "Male"}).sort("id", -1).limit(3)
for doc in cursor:
    print(doc)
```