



The MongoDB logo consists of the word "MongoDB" in a white, sans-serif font. Above the "M" is a white oval containing a black dot. Below the "D" is a horizontal line extending to the right, ending in a circle.

# CONEXIÓN Y CONSULTAS EN MONGODB

## OBJETIVO

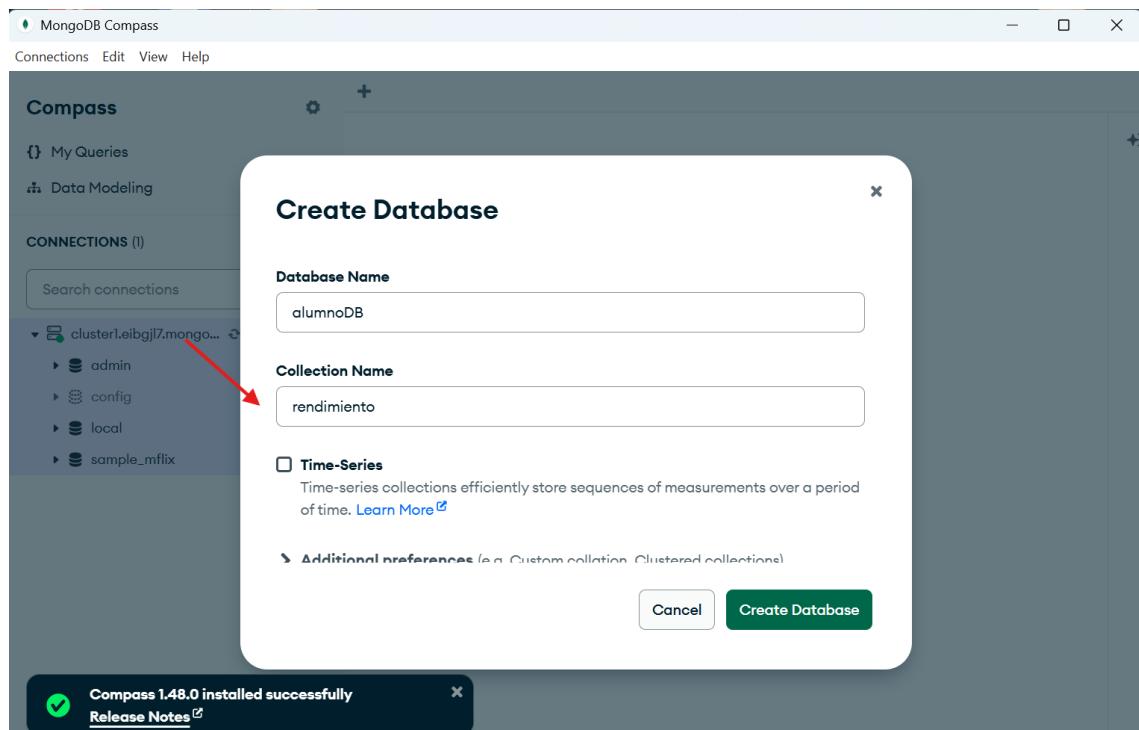
El objetivo de esta práctica es demostrar la capacidad de conectarse a un clúster de MongoDB, importar datos desde un CSV, y realizar consultas y agregaciones usando distintas herramientas: MongoDB Compass y Python.

## DESARROLLO DE LA TAREA

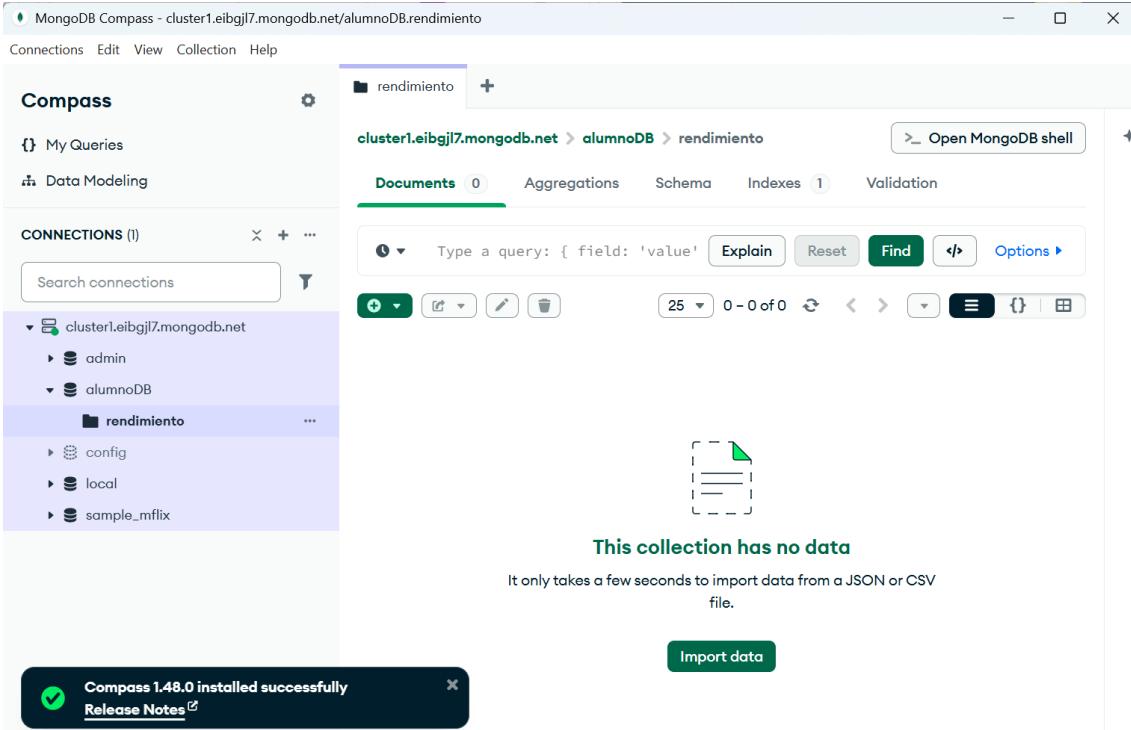
Para la tarea me descargo de Kaggle un archivo csv: Rendimiento de estudiantes, que contiene los siguientes campos:

- gender
- race/ethnicity
- parental level of education
- lunch
- test preparation course
- math score
- reading score
- writing score

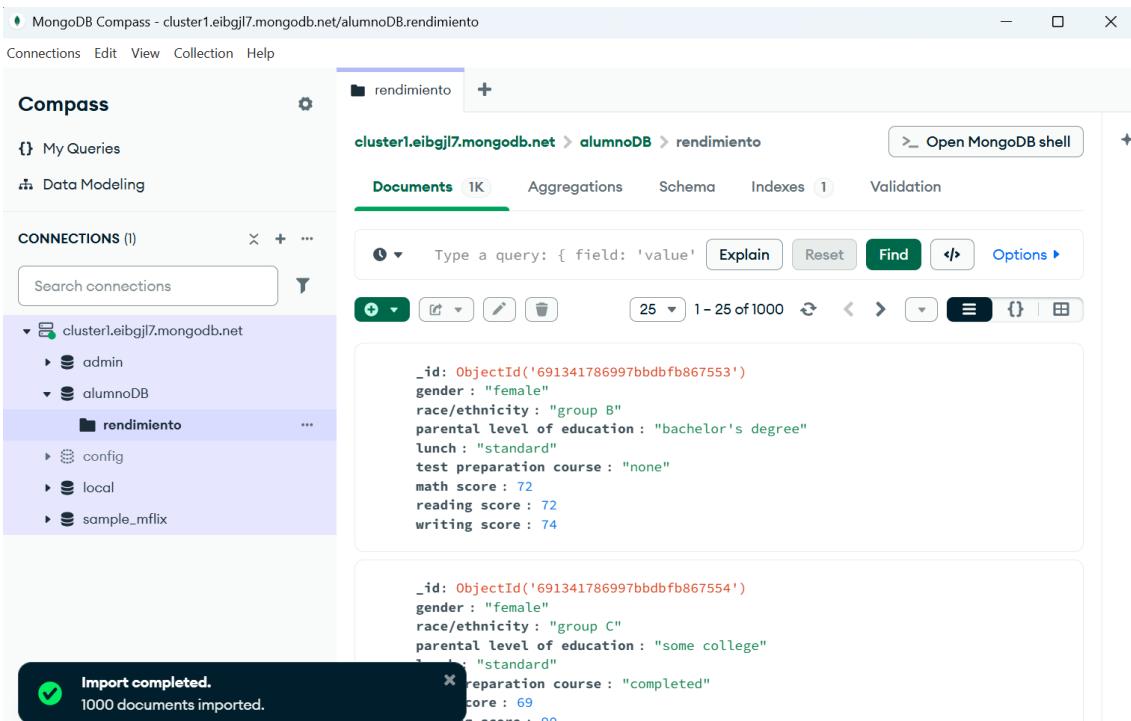
Se creó un clúster en MongoDB Atlas y se importó el CSV a la base de datos **alumnoDB** y la colección **rendimiento**.



## Importamos los datos



The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists a single connection to 'cluster1.eibgj17.mongodb.net'. Under this connection, the 'alumnoDB' database is selected, and the 'rendimiento' collection is highlighted. The main workspace displays a search bar with the query '{ field: 'value' }' and a button to 'Find'. Below the search bar, it says 'This collection has no data' and provides instructions to import data from a JSON or CSV file. A prominent green button at the bottom right says 'Import data'. A notification bar at the bottom left indicates 'Compass 1.48.0 installed successfully'.

The second screenshot shows the same MongoDB Compass interface after data has been imported. The 'Documents' tab now shows 1K documents. The results pane displays two document snippets. The first snippet is:

```
_id: ObjectId('691341786997bbdbfb867553')
gender : "female"
race/ethnicity : "group B"
parental level of education : "bachelor's degree"
lunch : "standard"
test preparation course : "none"
math score : 72
reading score : 72
writing score : 74
```

The second snippet is:

```
_id: ObjectId('691341786997bbdbfb867554')
gender : "female"
race/ethnicity : "group C"
parental level of education : "some college"
lunch : "standard"
test preparation course : "completed"
math score : 69
reading score : 90
```

A notification bar at the bottom left indicates 'Import completed. 1000 documents imported.'

## Explicación del proceso:

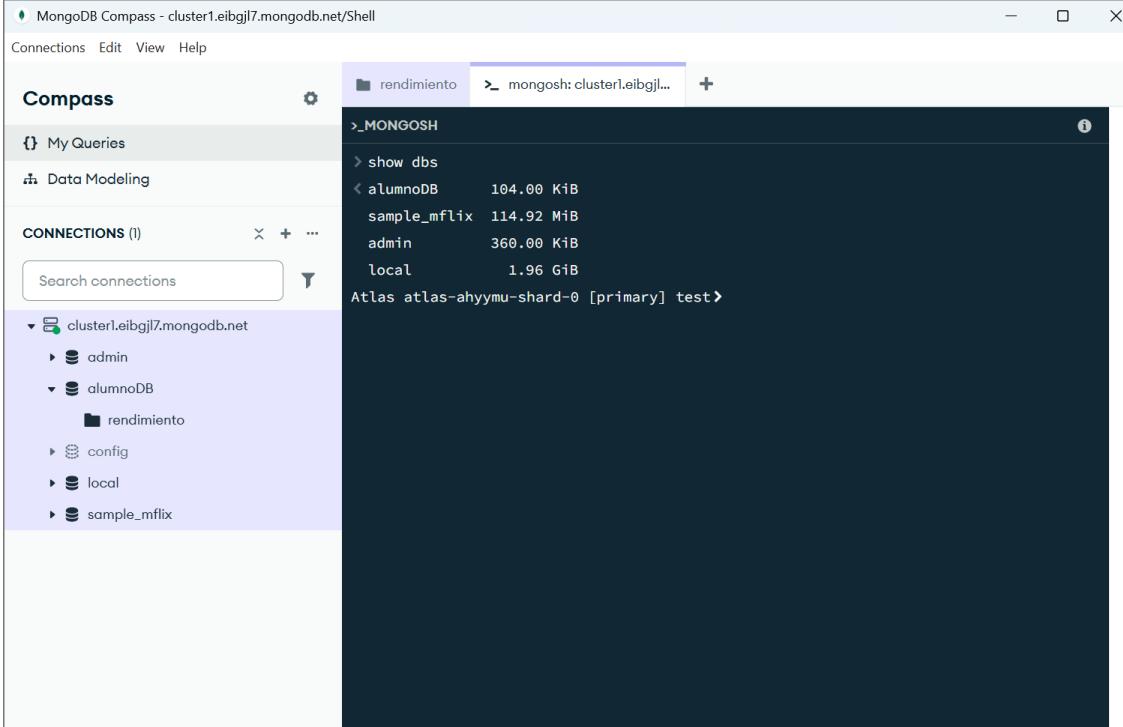
1. Abrir MongoDB Compass.
2. Crear la base de datos alumnoDB.
3. Crear la colección rendimiento.
4. Usar la opción **Import Data** y seleccionar el CSV.

La importación también se podría hacer de forma manual desde el terminal con el comando mongoimport:

```
mongoimport --uri "mongodb+srv://<user>:<pwd>@<cluster-url>/alumnoDB" \
--collection rendimiento \
--type csv --headerline \
--file /ruta/a/rendimiento.csv
```

## LISTAR BASES DE DATOS DEL SERVIDOR

Show dbs



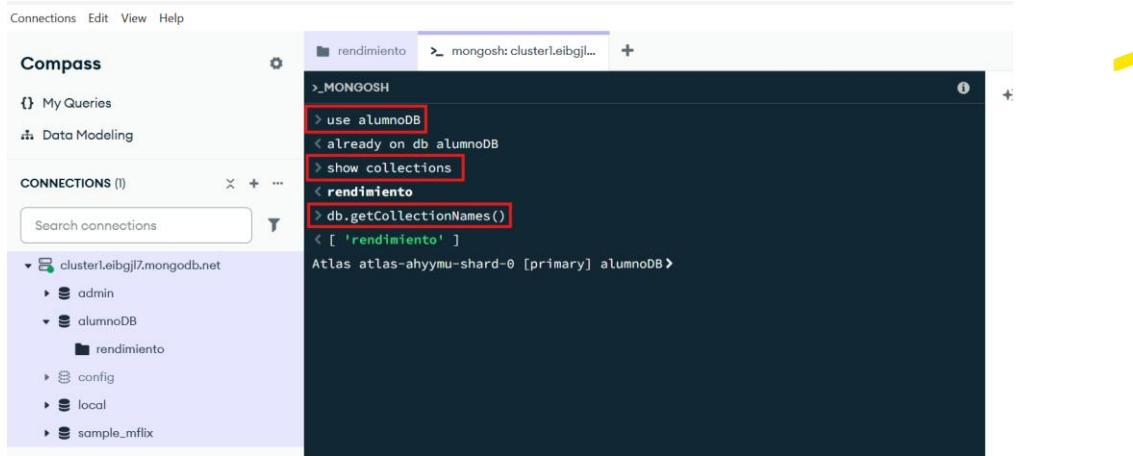
The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays connections and databases. A connection to 'cluster1.eibgjl7.mongodb.net' is selected, showing its databases: admin, alumnoDB, config, local, and sample\_mflix. The main panel shows the results of the 'show dbs' command in a monospaced font. The output includes the database name and size: alumnoDB (104.00 KiB), sample\_mflix (114.92 MiB), admin (360.00 KiB), local (1.96 GiB), and test (primary shard).

```
>_MONGOSH
> show dbs
< alumnoDB      104.00 KiB
  sample_mflix  114.92 MiB
  admin        360.00 KiB
  local        1.96 GiB
Atlas atlas-ahyymu-shard-0 [primary] test>
```

Para realizar consultas de una Base de Datos, lo primero que tendremos que hacer, será ubicarnos en ella →**use <nombre base de datos>**

## LISTAR COLECCIONES DE UNA BASE DE DATOS

Podremos utilizar **show collections** o con JS **db.getCollectionNames()**



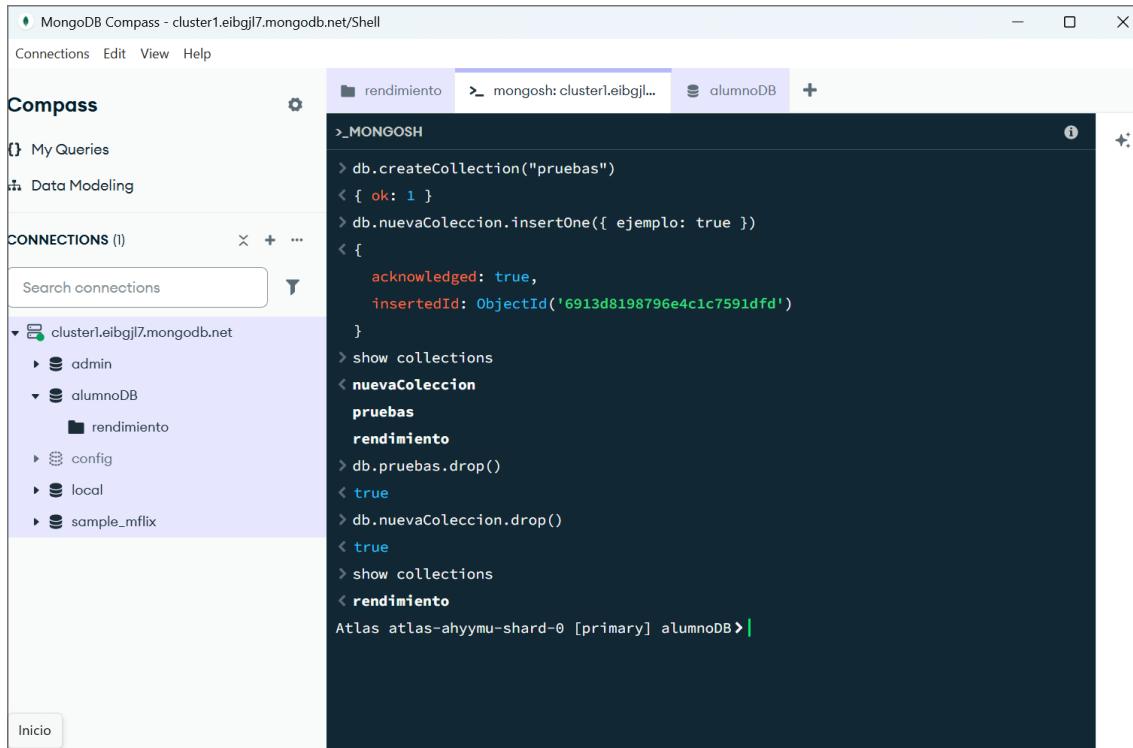
The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists a connection to 'cluster1.eibgjl7.mongodb.net'. The main panel displays a command history window with the following commands:

```
_MONGOSH
> use alumnoDB
< already on db alumnoDB
> show collections
< rendimiento
> db.getCollectionNames()
< [ 'rendimiento' ]
```

Below the command history, it says 'Atlas atlas-ahyymu-shard-0 [primary] alumnoDB'.

## CREAR Y ELIMINAR COLECCIONES

Para crear usamos el comando **db.createCollection** o con **insertOne**. Para eliminar, lo haremos con drop

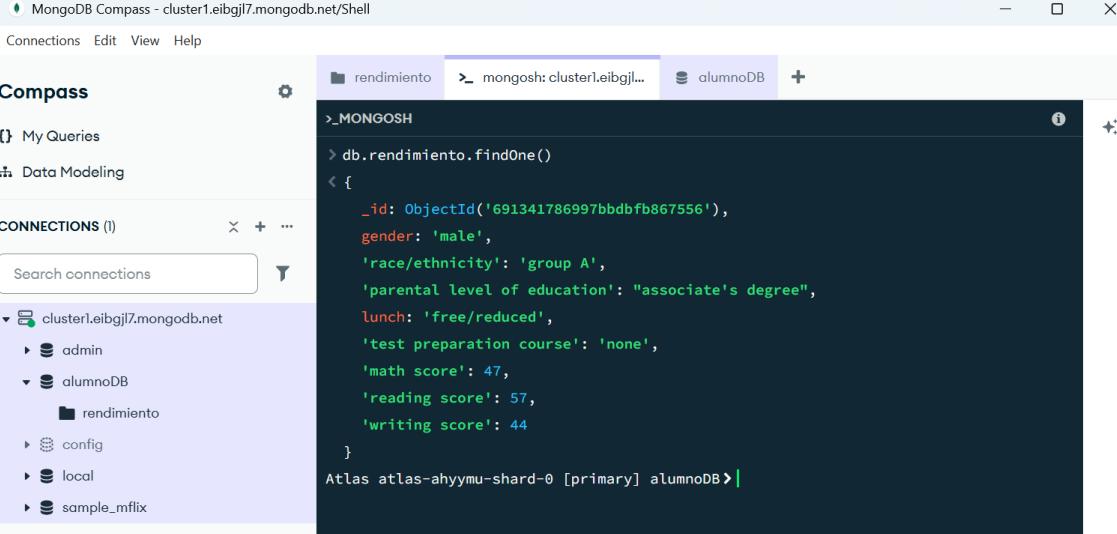


The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists a connection to 'cluster1.eibgjl7.mongodb.net'. The main panel displays a command history window with the following commands:

```
_MONGOSH
> db.createCollection("pruebas")
< { ok: 1 }
> db.nuevaColeccion.insertOne({ ejemplo: true })
< {
    acknowledged: true,
    insertedId: ObjectId('6913d8198796e4c1c7591dfd')
}
> show collections
< nuevaColeccion
  pruebas
  rendimiento
> db.pruebas.drop()
< true
> db.nuevaColeccion.drop()
< true
> show collections
< rendimiento
Atlas atlas-ahyymu-shard-0 [primary] alumnoDB
```

## CONSULTA SIMPLE SOBRE UNA COLECCIÓN

- Mostrar un documento

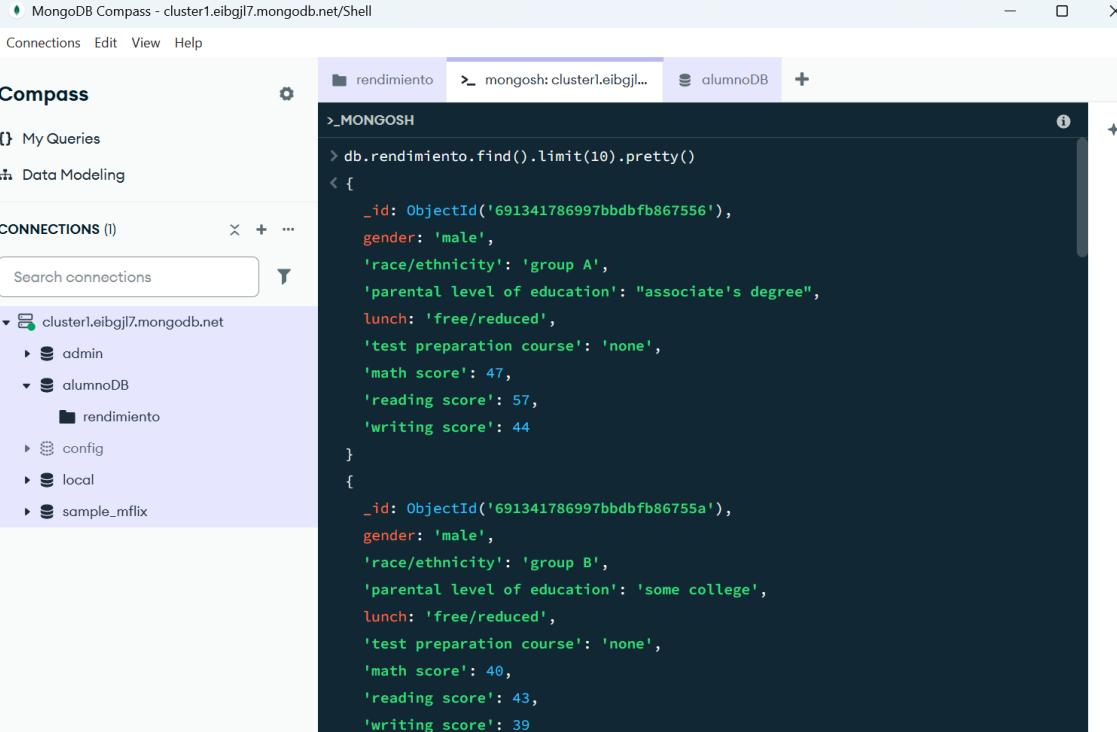


The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays connections and databases. A connection to 'cluster1.eibgjl7.mongodb.net' is selected, showing its contents: admin, alumnoDB (with rendimiento, config, local), and sample\_mflix. The main panel shows the MongoDB shell (MONGOSH) with the following command and result:

```
> db.rendimiento.findOne()
< {
  "_id": ObjectId('691341786997bbdbfb867556'),
  "gender": "male",
  "race/ethnicity": "group A",
  "parental level of education": "associate's degree",
  "lunch": "free/reduced",
  "test preparation course": "none",
  "math score": 47,
  "reading score": 57,
  "writing score": 44
}
```

The result is displayed in a light blue box. Below the result, the text 'Atlas atlas-ahyymu-shard-0 [primary] alumnoDB' is visible.

- Ver los primeros 10 documentos

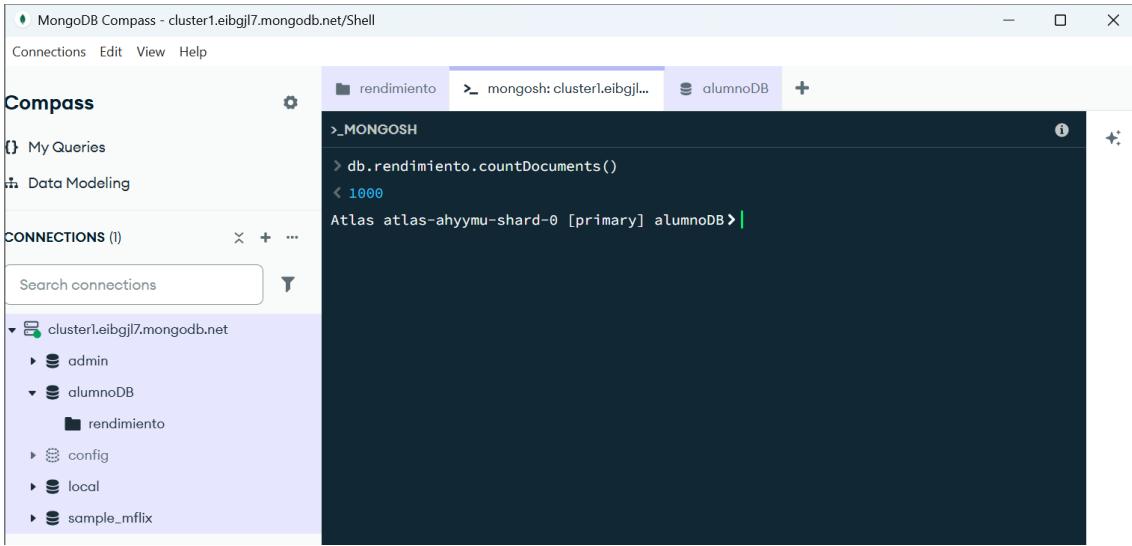


The screenshot shows the MongoDB Compass interface. The sidebar and connection selection are identical to the previous screenshot. The main panel shows the MongoDB shell (MONGOSH) with the following command and results:

```
> db.rendimiento.find().limit(10).pretty()
< [
  {
    "_id": ObjectId('691341786997bbdbfb867556'),
    "gender": "male",
    "race/ethnicity": "group A",
    "parental level of education": "associate's degree",
    "lunch": "free/reduced",
    "test preparation course": "none",
    "math score": 47,
    "reading score": 57,
    "writing score": 44
  },
  {
    "_id": ObjectId('691341786997bbdbfb86755a'),
    "gender": "male",
    "race/ethnicity": "group B",
    "parental level of education": "some college",
    "lunch": "free/reduced",
    "test preparation course": "none",
    "math score": 40,
    "reading score": 43,
    "writing score": 39
  }
]
```

The results are shown in a light blue box. The text 'Atlas atlas-ahyymu-shard-0 [primary] alumnoDB' is also present below the results.

- Contar documentos



MongoDB Compass - cluster1.eibgjl7.mongodb.net/Shell

Connections Edit View Help

**Compass**

- { My Queries
- ⋮ Data Modeling

**CONNECTIONS (1)**

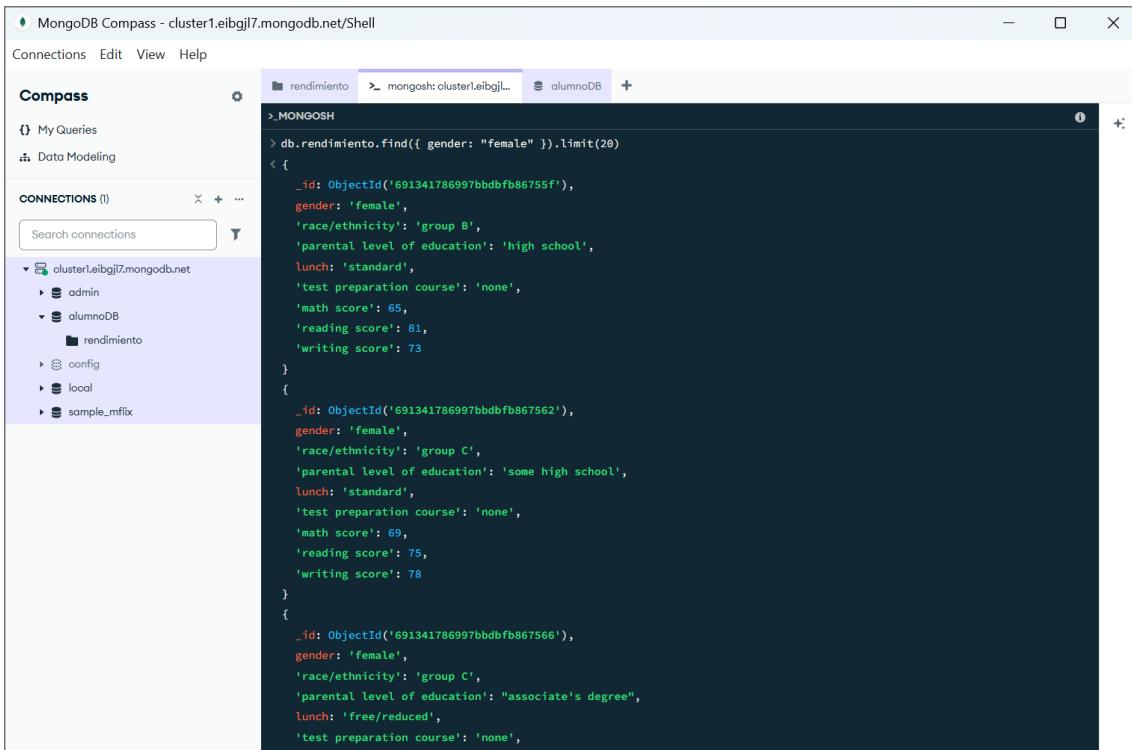
- cluster1.eibgjl7.mongodb.net
  - admin
  - alumnoDB
    - rendimiento
  - config
  - local
  - sample\_mflix

mongosh: cluster1.eibgjl... alumnoDB

```
>_MONGOSH
> db.rendimiento.countDocuments()
< 1000
Atlas atlas-ahyymu-shard-0 [primary] alumnoDB>
```

## CONSULTA CON FILTROS

- Todos los estudiantes femeninos



MongoDB Compass - cluster1.eibgjl7.mongodb.net/Shell

Connections Edit View Help

**Compass**

- { My Queries
- ⋮ Data Modeling

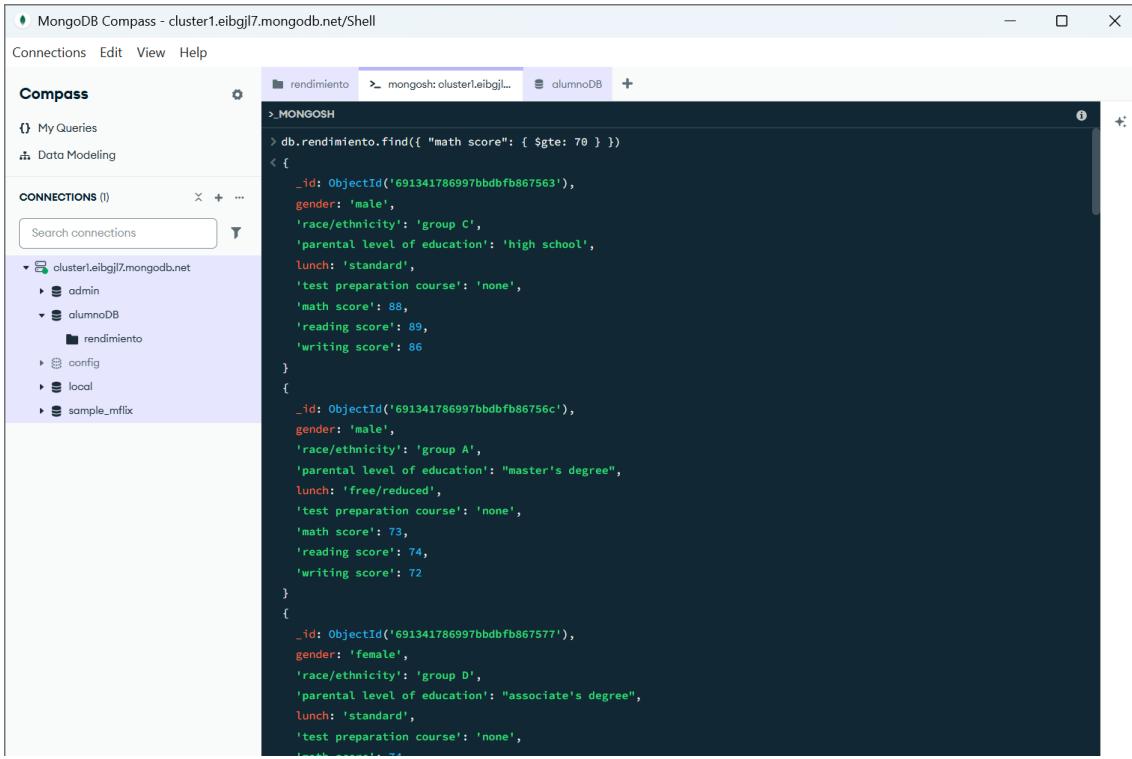
**CONNECTIONS (1)**

- cluster1.eibgjl7.mongodb.net
  - admin
  - alumnoDB
    - rendimiento
  - config
  - local
  - sample\_mflix

mongosh: cluster1.eibgjl... alumnoDB

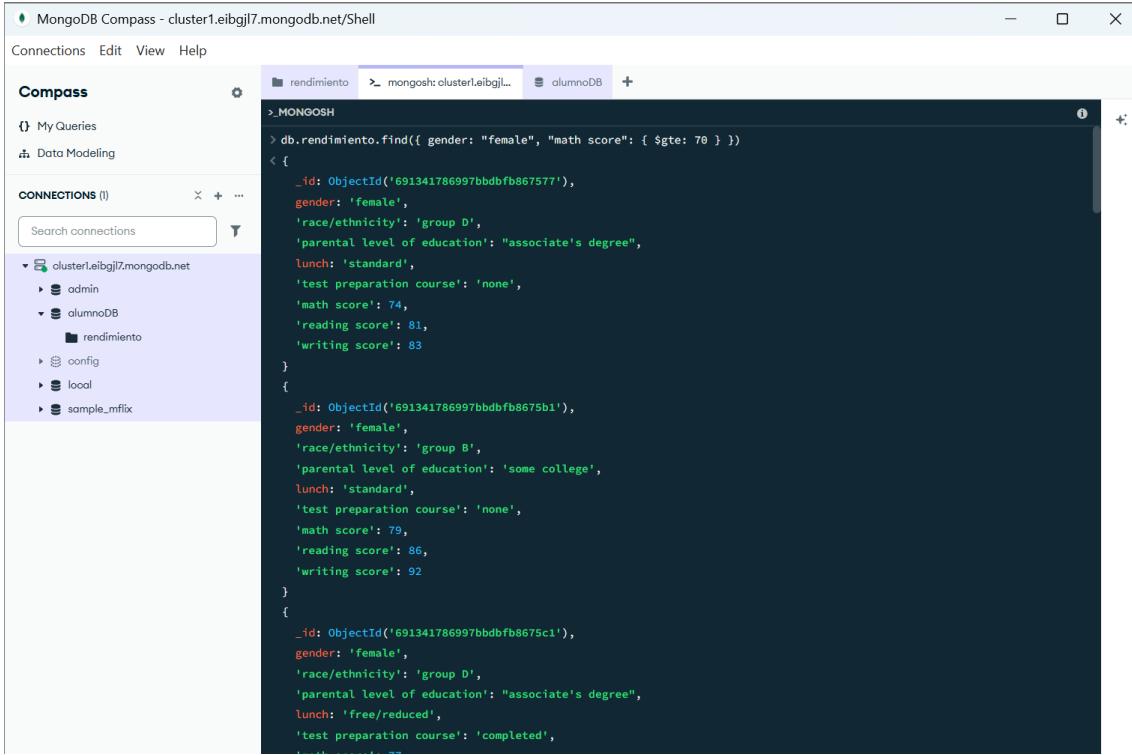
```
>_MONGOSH
> db.rendimiento.find({ gender: "female" }).limit(20)
< [
  {
    "_id": ObjectId('691341786997bbdbfb86755f'),
    "gender": "female",
    "race/ethnicity": "group B",
    "parental level of education": "high school",
    "lunch": "standard",
    "test preparation course": "none",
    "math score": 65,
    "reading score": 81,
    "writing score": 73
  },
  {
    "_id": ObjectId('691341786997bbdbfb867562'),
    "gender": "female",
    "race/ethnicity": "group C",
    "parental level of education": "some high school",
    "lunch": "standard",
    "test preparation course": "none",
    "math score": 69,
    "reading score": 75,
    "writing score": 78
  },
  {
    "_id": ObjectId('691341786997bbdbfb867566'),
    "gender": "female",
    "race/ethnicity": "group C",
    "parental level of education": "associate's degree",
    "lunch": "free/reduced",
    "test preparation course": "none",
    "math score": 64
  }
]
```

- Estudiantes con math score  $\geq 70$



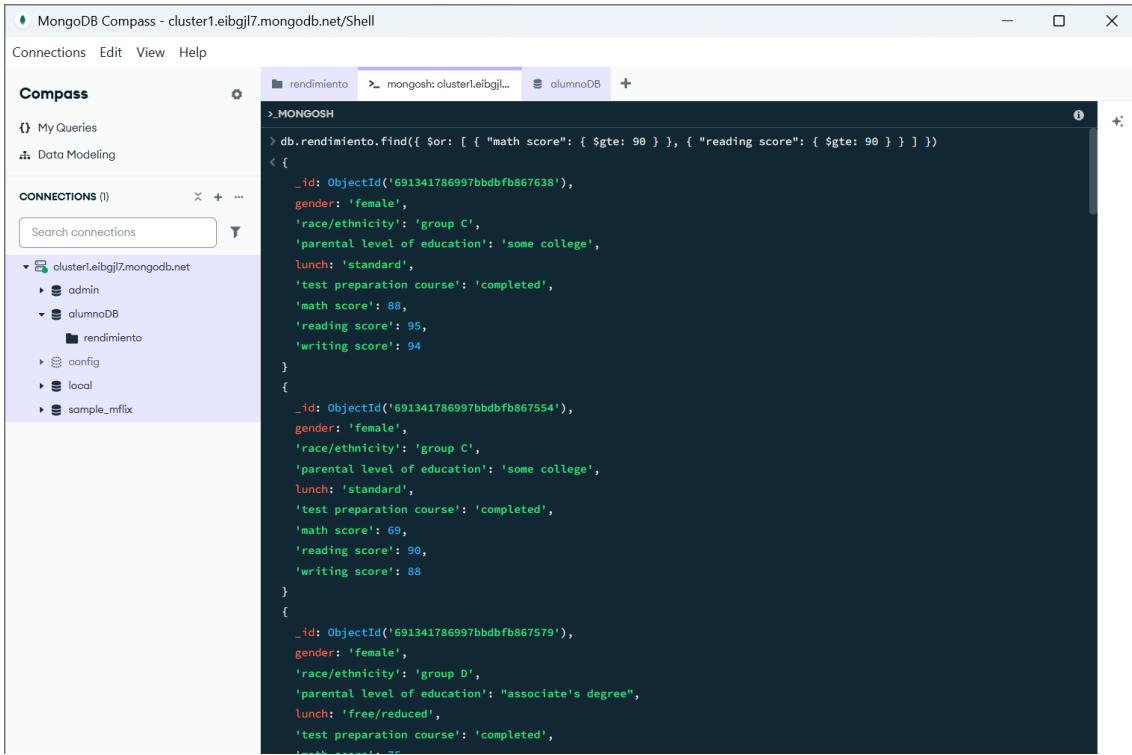
```
>_MONGOSH
> db.rendimiento.find({ "math score": { $gte: 70 } })
< [
  {
    "_id": ObjectId("691341786997bbdbfb867563"),
    "gender": "male",
    "race/ethnicity": "group C",
    "parental level of education": "high school",
    "lunch": "standard",
    "test preparation course": "none",
    "math score": 88,
    "reading score": 89,
    "writing score": 86
  },
  {
    "_id": ObjectId("691341786997bbdbfb86756c"),
    "gender": "male",
    "race/ethnicity": "group A",
    "parental level of education": "master's degree",
    "lunch": "free/reduced",
    "test preparation course": "none",
    "math score": 73,
    "reading score": 74,
    "writing score": 72
  },
  {
    "_id": ObjectId("691341786997bbdbfb867577"),
    "gender": "female",
    "race/ethnicity": "group D",
    "parental level of education": "associate's degree",
    "lunch": "standard",
    "test preparation course": "none",
    "math score": 74
  }
]
```

- Varios filtros: género y math



```
>_MONGOSH
> db.rendimiento.find({ gender: "female", "math score": { $gte: 70 } })
< [
  {
    "_id": ObjectId("691341786997bbdbfb867577"),
    "gender": "female",
    "race/ethnicity": "group D",
    "parental level of education": "associate's degree",
    "lunch": "standard",
    "test preparation course": "none",
    "math score": 74,
    "reading score": 81,
    "writing score": 83
  },
  {
    "_id": ObjectId("691341786997bbdbfb8675b1"),
    "gender": "female",
    "race/ethnicity": "group B",
    "parental level of education": "some college",
    "lunch": "standard",
    "test preparation course": "none",
    "math score": 79,
    "reading score": 86,
    "writing score": 92
  },
  {
    "_id": ObjectId("691341786997bbdbfb8675c1"),
    "gender": "female",
    "race/ethnicity": "group D",
    "parental level of education": "associate's degree",
    "lunch": "free/reduced",
    "test preparation course": "completed",
    "math score": 74
  }
]
```

## - Filtro con or



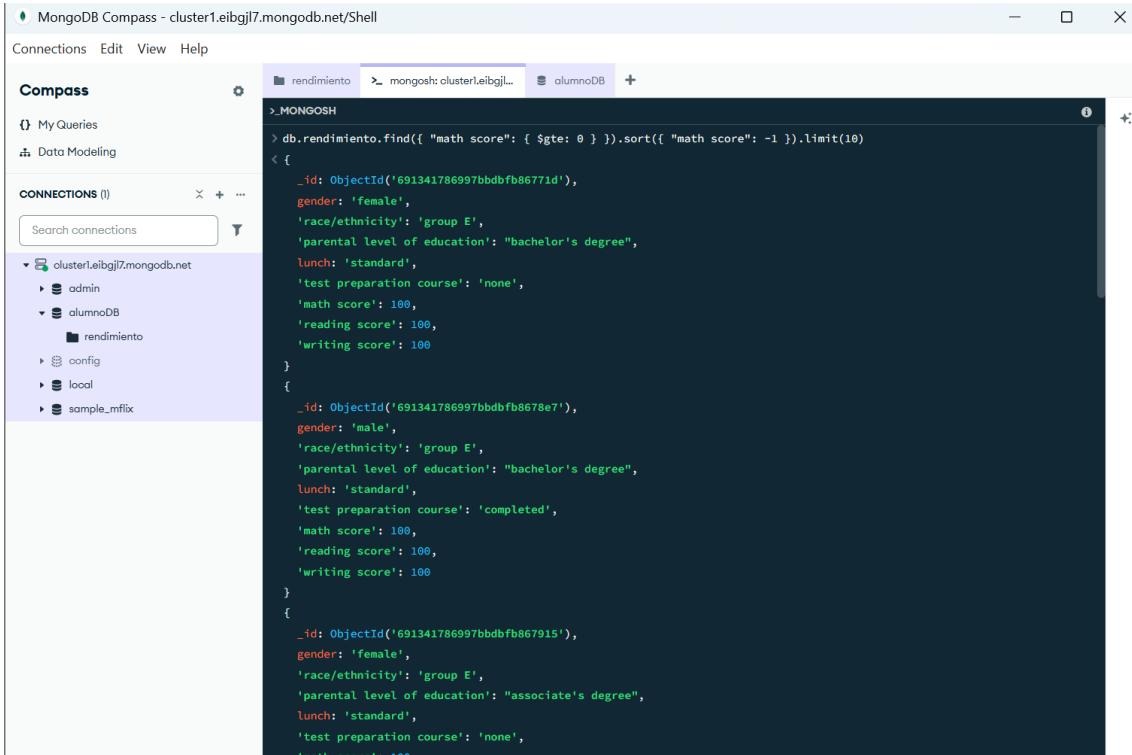
The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'cluster1.eibgj17.mongodb.net' with its databases: 'admin', 'alumnoDB', and 'rendimiento'. The 'alumnoDB' database is selected. In the main query editor window, the command is:

```
> db.rendimiento.find({ $or: [ { "math score": { $gte: 90 } }, { "reading score": { $gte: 90 } } ] })
```

This command finds documents where either the 'math score' or 'reading score' is greater than or equal to 90.

## CONSULTA CON FILTROS Y ORDENACIÓN

### - Orden descendente por math score

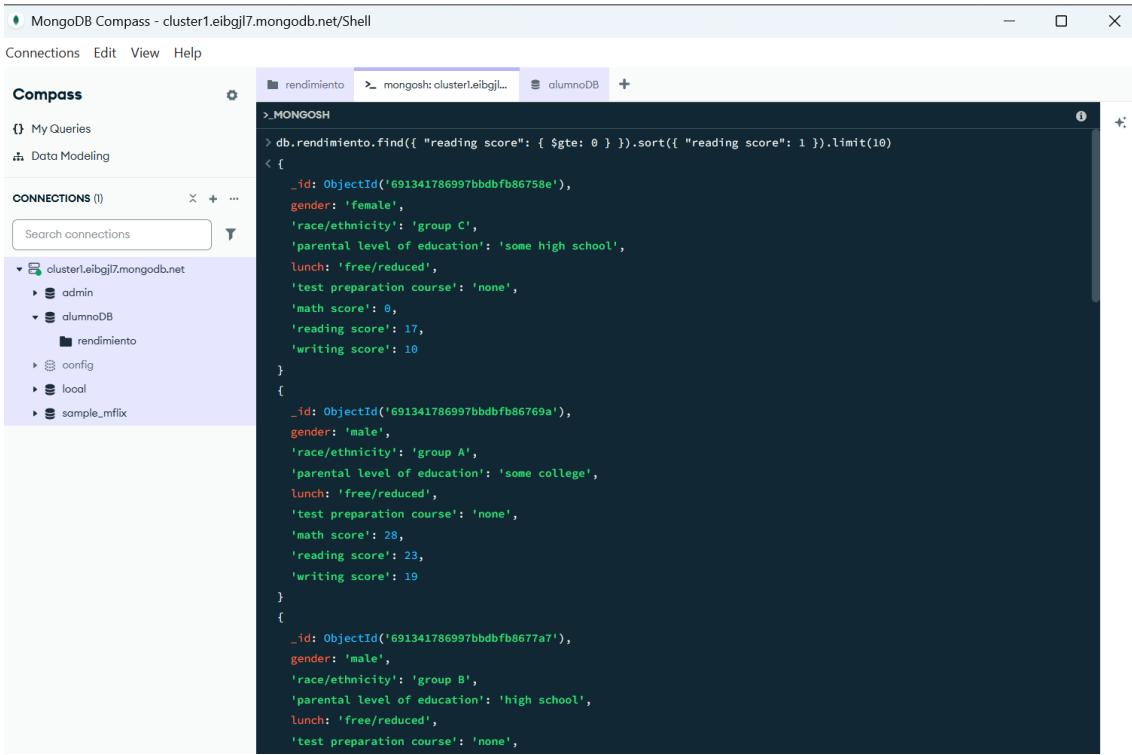


The screenshot shows the MongoDB Compass interface. The 'alumnoDB' database is selected in the sidebar. The query in the main window is:

```
> db.rendimiento.find({ "math score": { $gte: 0 } }).sort({ "math score": -1 }).limit(10)
```

This command finds all documents with a 'math score' greater than or equal to 0, sorts them by 'math score' in descending order, and limits the results to 10 documents. The output shows three female students from group E with a 'math score' of 100.

## - Orden ascendente por reading score



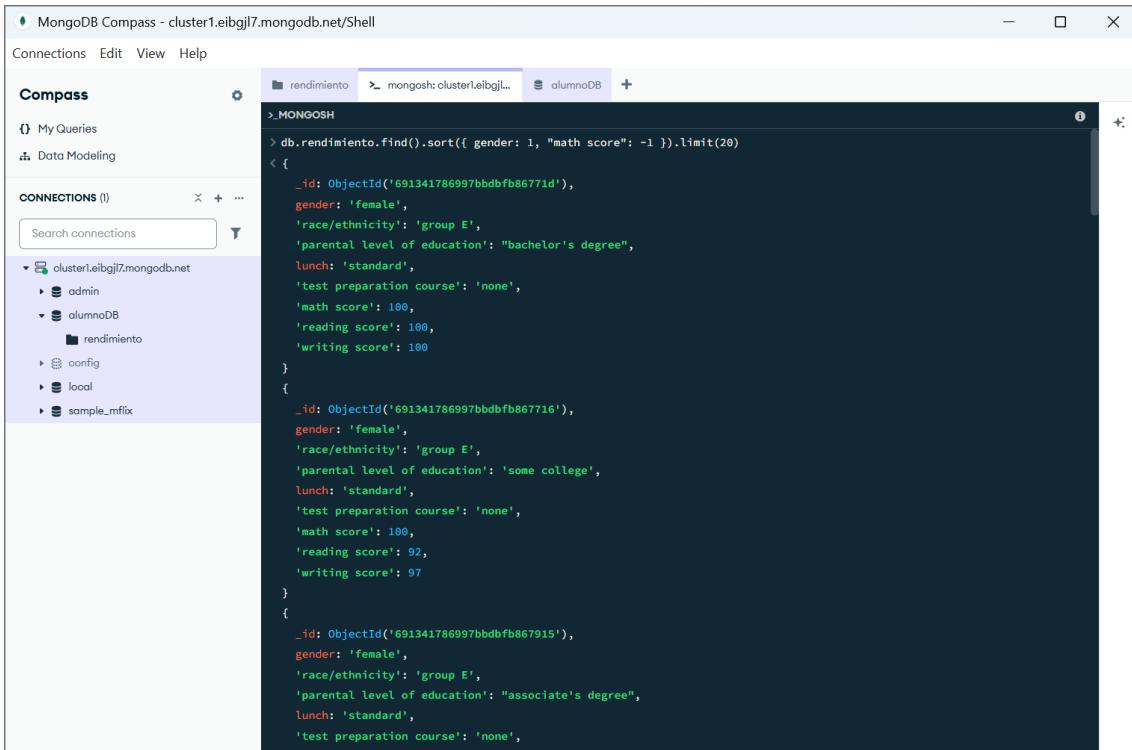
The screenshot shows the MongoDB Compass interface. On the left, the 'Compass' sidebar includes sections for 'My Queries', 'Data Modeling', and 'CONNECTIONS'. Under 'CONNECTIONS', there is a list of databases: 'cluster1.eibgjl7.mongodb.net' (selected), 'admin', 'alumnoDB', 'rendimiento' (selected), 'config', 'local', and 'sample\_mflix'. The main panel displays a MongoDB shell session titled 'MONGOSH'. The command entered is:

```
> db.rendimiento.find({ "reading score": { $gte: 0 } }).sort({ "reading score": 1 }).limit(10)
```

The results show 10 documents from the 'rendimiento' collection, ordered by 'reading score' in ascending order. The first few documents are:

```
< {
  "_id": ObjectId('691341786997bbdbfb86758e'),
  "gender": "female",
  "race/ethnicity": "group C",
  "parental level of education": "some high school",
  "lunch": "free/reduced",
  "test preparation course": "none",
  "math score": 0,
  "reading score": 17,
  "writing score": 10
}
{
  "_id": ObjectId('691341786997bbdbfb86769a'),
  "gender": "male",
  "race/ethnicity": "group A",
  "parental level of education": "some college",
  "lunch": "free/reduced",
  "test preparation course": "none",
  "math score": 28,
  "reading score": 23,
  "writing score": 19
}
{
  "_id": ObjectId('691341786997bbdbfb8677a7'),
  "gender": "male",
  "race/ethnicity": "group B",
  "parental level of education": "high school",
  "lunch": "free/reduced",
  "test preparation course": "none",
  "math score": 28,
  "reading score": 23,
  "writing score": 19
}
```

## - Orden por varias claves



The screenshot shows the MongoDB Compass interface, identical to the previous one but with a different query. The 'Compass' sidebar and 'CONNECTIONS' list are the same. The main panel displays a MongoDB shell session titled 'MONGOSH'. The command entered is:

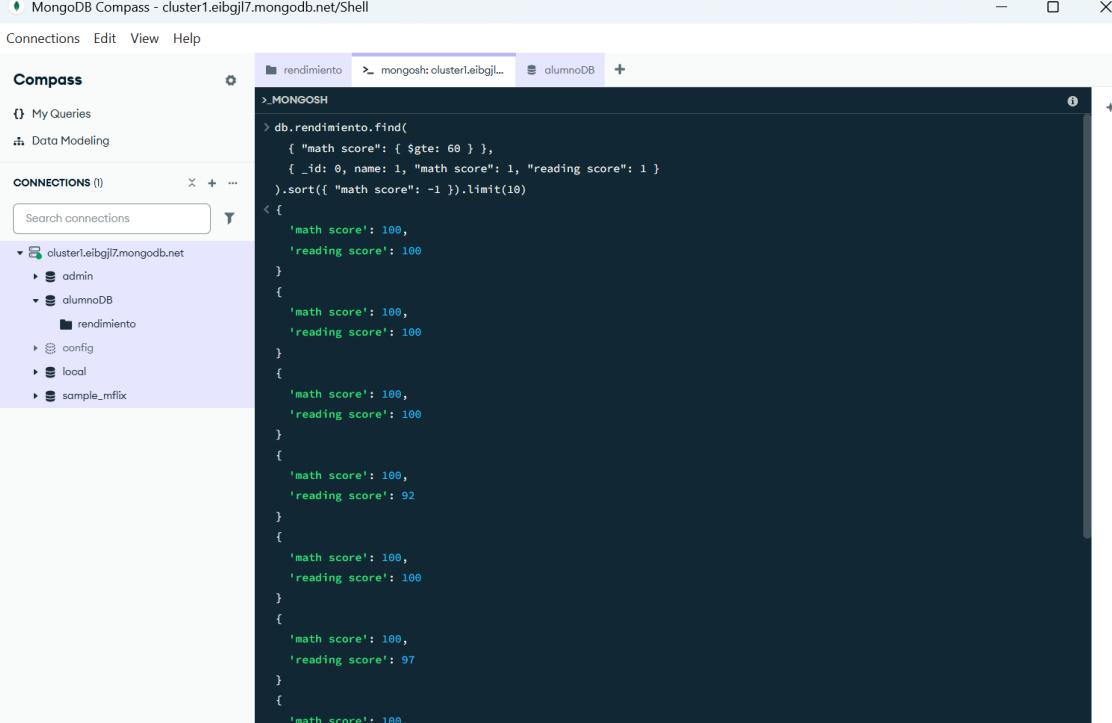
```
> db.rendimiento.find().sort({ "gender": 1, "math score": -1 }).limit(20)
```

The results show 20 documents from the 'rendimiento' collection, sorted by 'gender' in ascending order and 'math score' in descending order. The first few documents are:

```
< {
  "_id": ObjectId('691341786997bbdbfb86771d'),
  "gender": "female",
  "race/ethnicity": "group E",
  "parental level of education": "bachelor's degree",
  "lunch": "standard",
  "test preparation course": "none",
  "math score": 100,
  "reading score": 100,
  "writing score": 100
}
{
  "_id": ObjectId('691341786997bbdbfb867716'),
  "gender": "female",
  "race/ethnicity": "group E",
  "parental level of education": "some college",
  "lunch": "standard",
  "test preparation course": "none",
  "math score": 100,
  "reading score": 92,
  "writing score": 97
}
{
  "_id": ObjectId('691341786997bbdbfb867915'),
  "gender": "female",
  "race/ethnicity": "group E",
  "parental level of education": "associate's degree",
  "lunch": "standard",
  "test preparation course": "none",
  "math score": 100,
```

## CONSULTA CON FILTROS, ORDENACIÓN Y PROYECCIÓN

- Mostrar sólo campos relevantes sin id



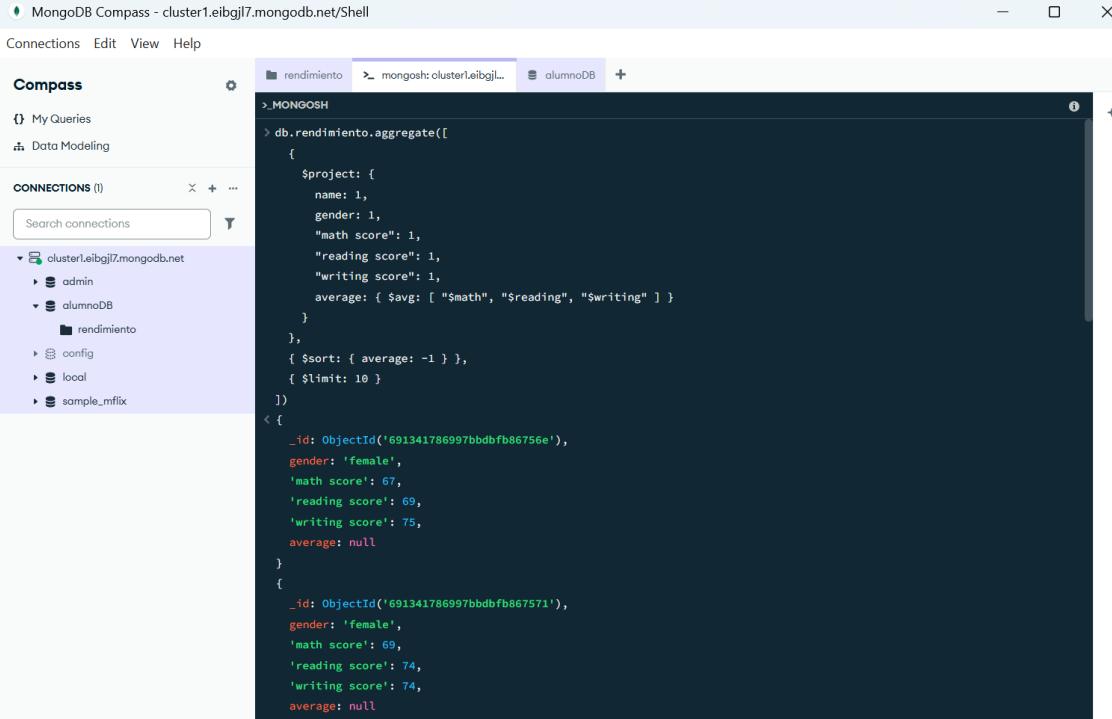
The screenshot shows the MongoDB Compass interface with the Shell tab active. The left sidebar displays connections to a cluster named 'cluster1.eibgjl7.mongodb.net'. The current database is 'alumnoDB' and the collection is 'rendimiento'. The query entered in the shell is:

```
> db.rendimiento.find(
  { "math score": { $gte: 60 } },
  { _id: 0, name: 1, "math score": 1, "reading score": 1 }
).sort({ "math score": -1 }).limit(10)

< [
  {
    'math score': 100,
    'reading score': 100
  },
  {
    'math score': 100,
    'reading score': 100
  },
  {
    'math score': 100,
    'reading score': 100
  },
  {
    'math score': 100,
    'reading score': 92
  },
  {
    'math score': 100,
    'reading score': 100
  },
  {
    'math score': 100,
    'reading score': 97
  },
  {
    'math score': 100,
```

## INVESTIGACIONES Y CONSULTAS DE AGREGACIÓN

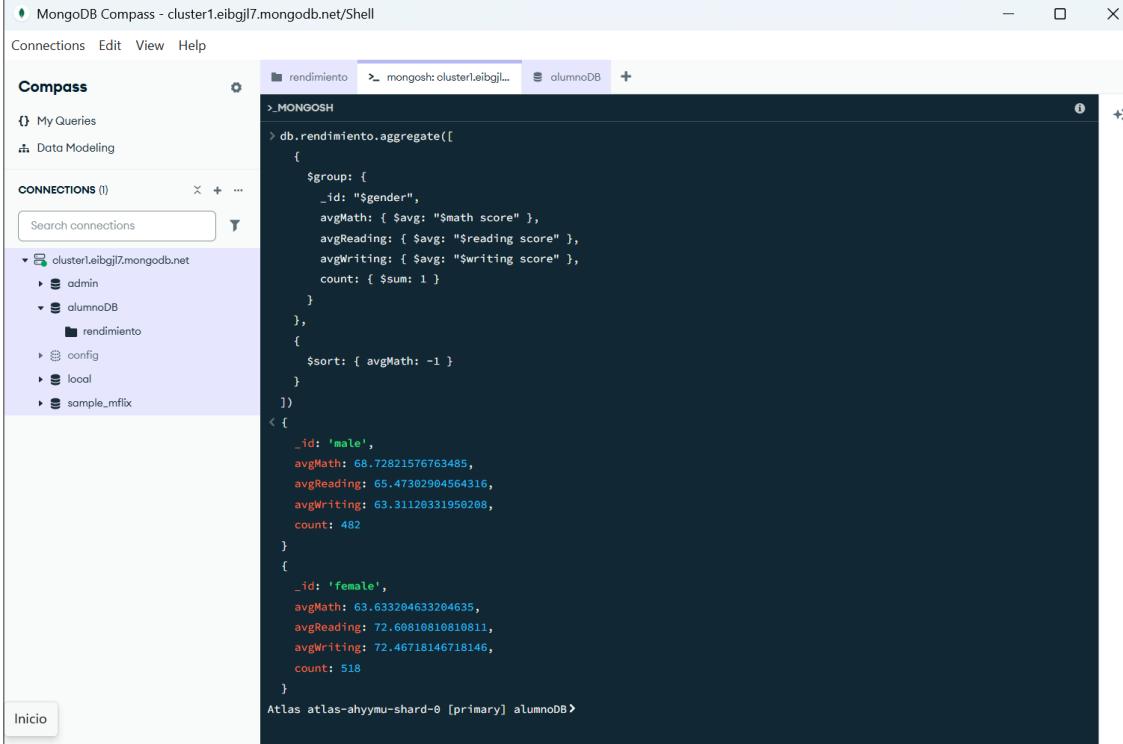
- Cálculo del promedio por estudiante



The screenshot shows the MongoDB Compass interface with the Shell tab active. The left sidebar displays connections to a cluster named 'cluster1.eibgjl7.mongodb.net'. The current database is 'alumnoDB' and the collection is 'rendimiento'. The query entered in the shell is an aggregation pipeline:

```
> MONGOSH
> db.rendimiento.aggregate([
  {
    $project: {
      name: 1,
      gender: 1,
      "math score": 1,
      "reading score": 1,
      "writing score": 1,
      average: { $avg: [ "$math", "$reading", "$writing" ] }
    }
  },
  {
    $sort: { average: -1 },
    { $limit: 10 }
  }
]
< [
  {
    _id: ObjectId('691341786997bbdbfb86756e'),
    gender: 'female',
    'math score': 67,
    'reading score': 69,
    'writing score': 75,
    average: null
  },
  {
    _id: ObjectId('691341786997bbdbfb867571'),
    gender: 'female',
    'math score': 69,
    'reading score': 74,
    'writing score': 74,
    average: null
  },
  {
    _id: ObjectId('691341786997bbdbfb867572'),
    gender: 'female',
    'math score': 74,
```

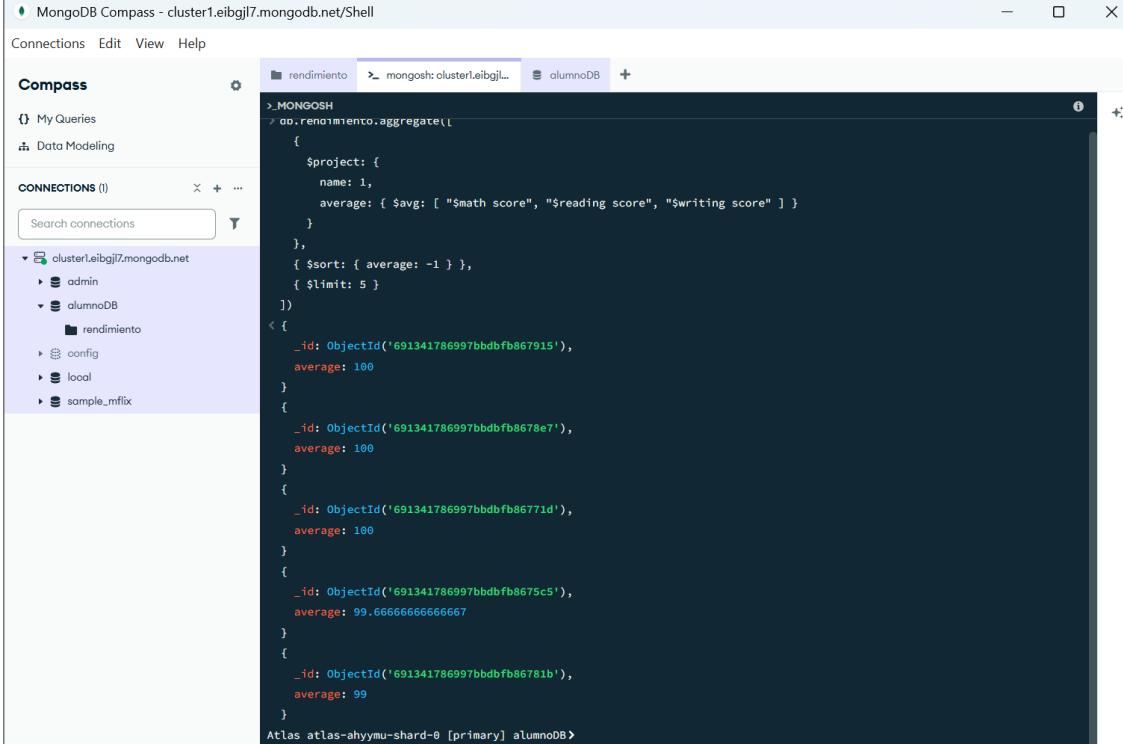
## - Media de puntuaciones por género



The screenshot shows the MongoDB Compass interface. The left sidebar displays connections to a cluster named 'cluster1.eibgjl7.mongodb.net'. The 'rendimiento' database is selected. In the main query editor, the following MongoDB aggregation pipeline is displayed:

```
>_MONGOSH
> db.rendimiento.aggregate([
  {
    $group: {
      _id: "$gender",
      avgMath: { $avg: "$math score" },
      avgReading: { $avg: "$reading score" },
      avgWriting: { $avg: "$writing score" },
      count: { $sum: 1 }
    }
  },
  {
    $sort: { avgMath: -1 }
  }
])
< [
  {
    _id: 'male',
    avgMath: 68.72821576763485,
    avgReading: 65.47362984564316,
    avgWriting: 63.31120331950208,
    count: 482
  },
  {
    _id: 'female',
    avgMath: 63.633204633284635,
    avgReading: 72.60810810810811,
    avgWriting: 72.46718146718146,
    count: 518
  }
]
Atlas atlas-ahyymu-shard-0 [primary] alumnoDB>
```

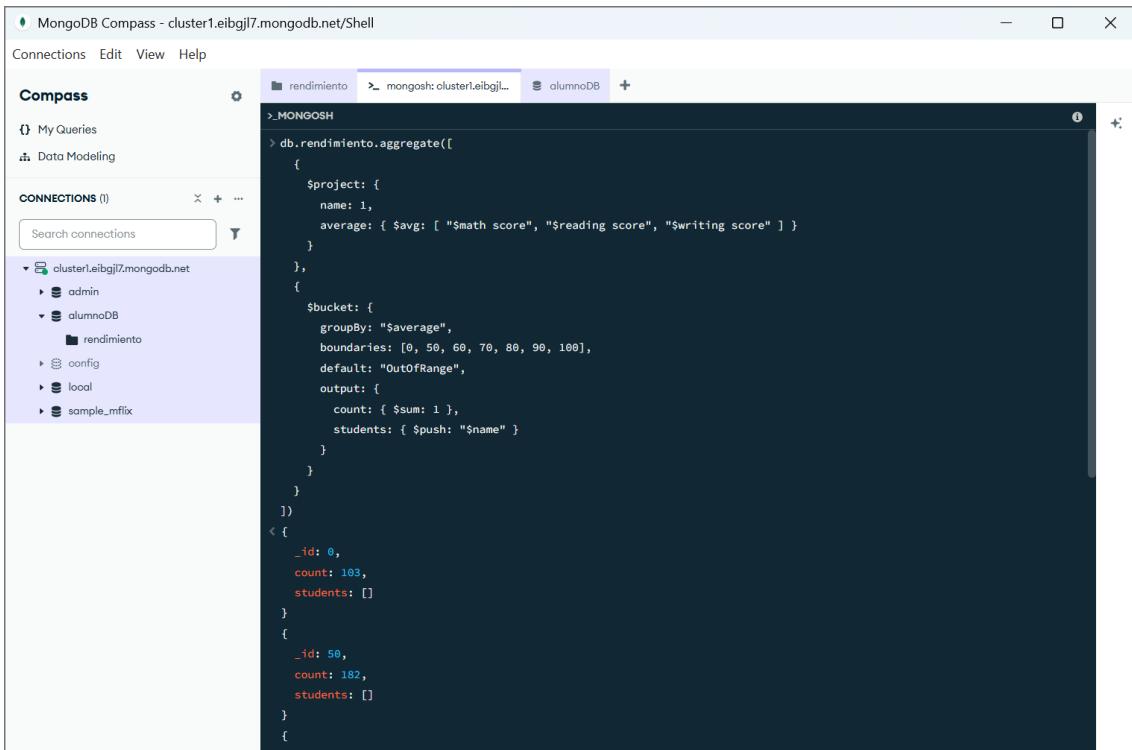
## - Top 5 alumnos por promedio



The screenshot shows the MongoDB Compass interface. The left sidebar displays connections to a cluster named 'cluster1.eibgjl7.mongodb.net'. The 'rendimiento' database is selected. In the main query editor, the following MongoDB aggregation pipeline is displayed:

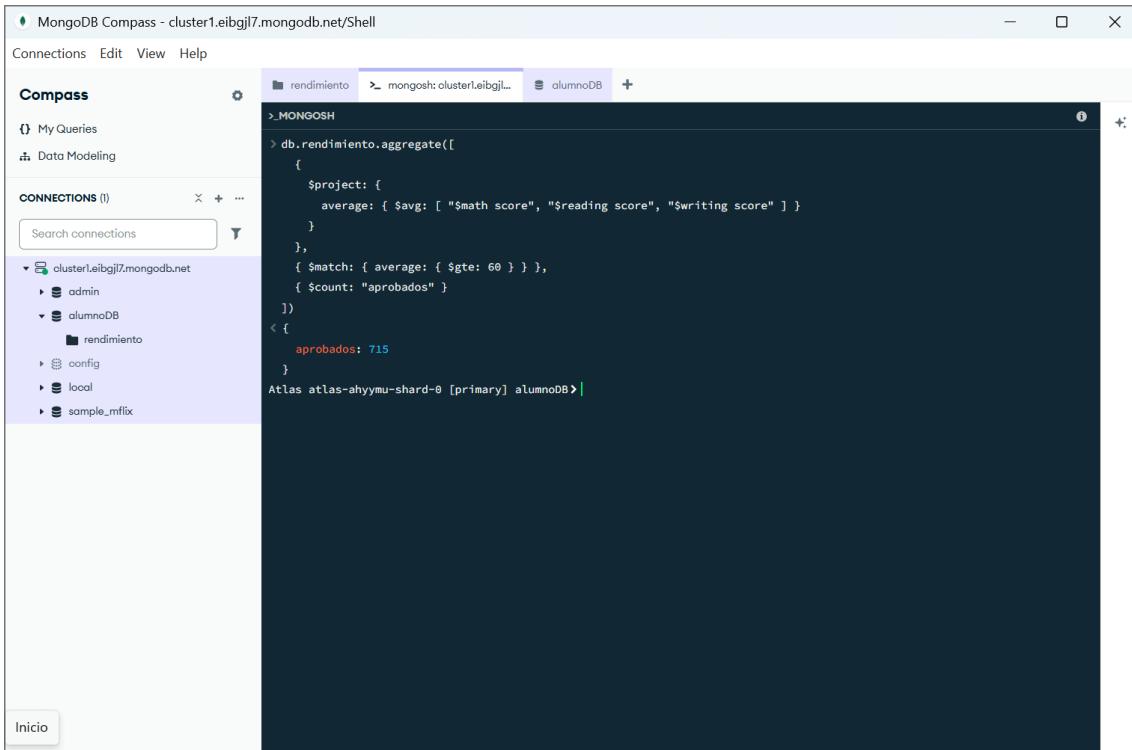
```
>_MONGOSH
> db.rendimiento.aggregate([
  {
    $project: {
      name: 1,
      average: { $avg: [ "$math score", "$reading score", "$writing score" ] }
    }
  },
  {
    $sort: { average: -1 },
    { $limit: 5 }
  }
])
< [
  {
    _id: ObjectId('691341786997bbdbfb867915'),
    average: 100
  },
  {
    _id: ObjectId('691341786997bbdbfb8678e7'),
    average: 100
  },
  {
    _id: ObjectId('691341786997bbdbfb86771d'),
    average: 100
  },
  {
    _id: ObjectId('691341786997bbdbfb8675c5'),
    average: 99.666666666666667
  },
  {
    _id: ObjectId('691341786997bbdbfb86781b'),
    average: 99
  }
]
Atlas atlas-ahyymu-shard-0 [primary] alumnoDB>
```

- Distribución por rangos de nota media



```
>_MONGOSH
> db.rendimiento.aggregate([
  {
    $project: {
      name: 1,
      average: { $avg: [ "$math score", "$reading score", "$writing score" ] }
    }
  },
  {
    $bucket: {
      groupBy: "$average",
      boundaries: [0, 50, 60, 70, 80, 90, 100],
      default: "OutOfRange",
      output: {
        count: { $sum: 1 },
        students: { $push: "$name" }
      }
    }
  }
])
< [
  {
    _id: 0,
    count: 103,
    students: []
  },
  {
    _id: 50,
    count: 182,
    students: []
  }
]
```

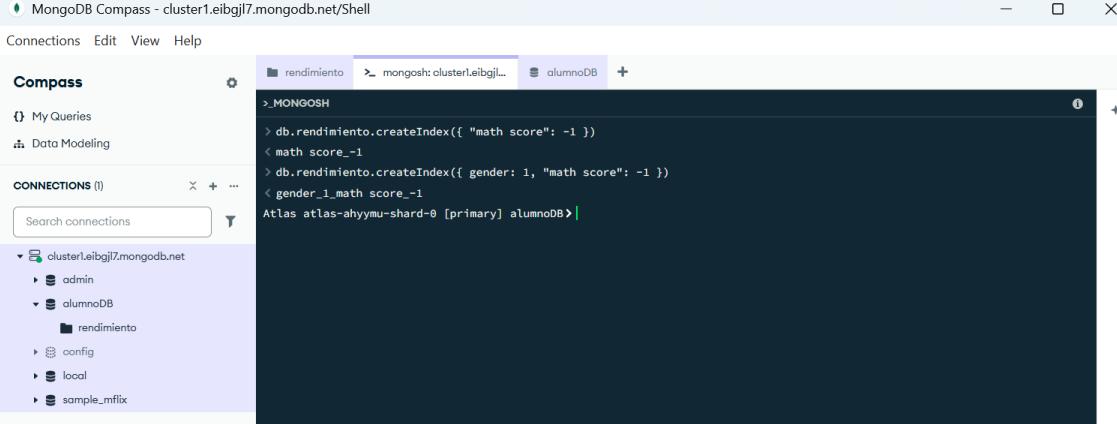
- Contar cuántos aprueban



```
>_MONGOSH
> db.rendimiento.aggregate([
  {
    $project: {
      average: { $avg: [ "$math score", "$reading score", "$writing score" ] }
    }
  },
  {
    $match: { average: { $gte: 60 } },
    $count: "aprobados"
  }
])
< [
  {
    aprobados: 715
  }
]
Atlas atlas-ahyymu-shard-0 [primary] alumnoDB>
```

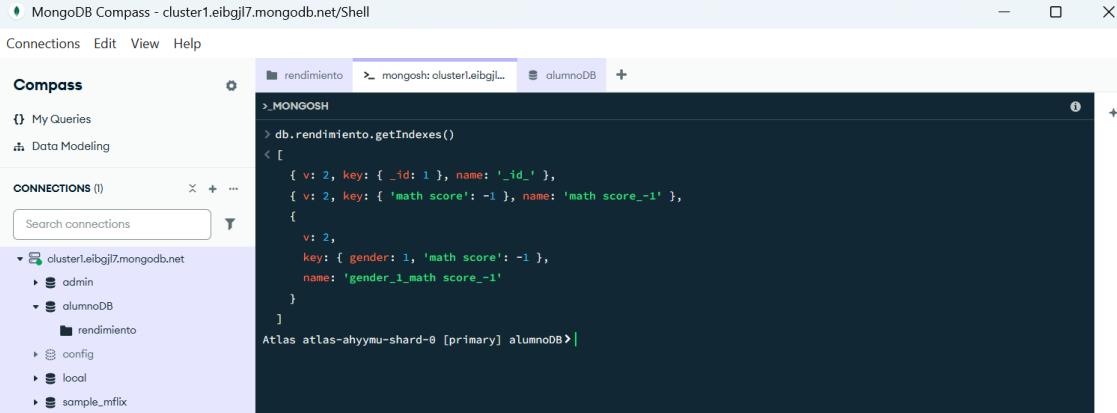
## ÍNDICES PARA OPTIMIZACIÓN

- Crear índices para mejorar el rendimiento de las consultas



```
>_MONGOSH
> db.rendimiento.createIndex({ "math score": -1 })
< math score_-1
> db.rendimiento.createIndex({ gender: 1, "math score": -1 })
< gender_1_math score_-1
Atlas atlas-ahyymu-shard-0 [primary] alumnoDB>
```

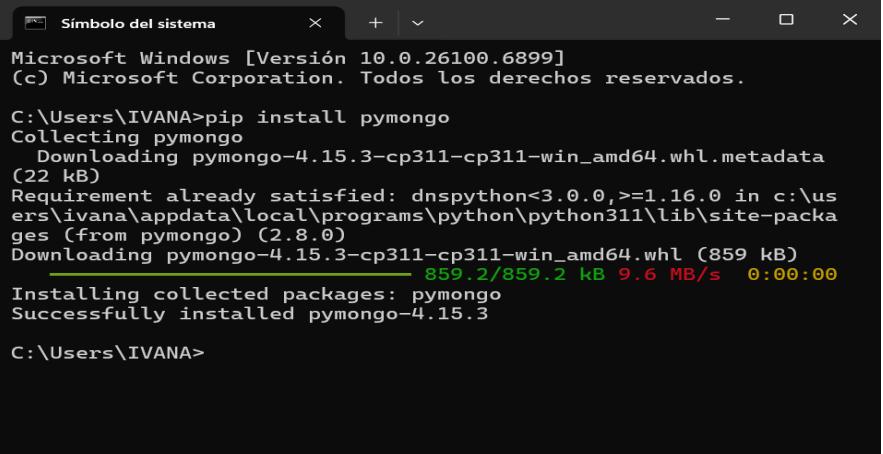
- Comprobar índices



```
>_MONGOSH
> db.rendimiento.getIndexes()
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    { v: 2, key: { 'math score': -1 }, name: 'math score_-1' },
    [
        {
            v: 2,
            key: { gender: 1, 'math score': -1 },
            name: 'gender_1_math score_-1'
        }
    ]
Atlas atlas-ahyymu-shard-0 [primary] alumnoDB>
```

## INTEGRACIÓN CON PYTHON (PYMONGO + PANDAS)

Para scripts y análisis en Python instalamos la librería pymongo



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.26100.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\IVANA>pip install pymongo
Collecting pymongo
  Downloading pymongo-4.15.3-cp311-cp311-win_amd64.whl.metadata
(22 kB)
Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in c:\users\ivana\appdata\local\programs\python\python311\lib\site-packages (from pymongo) (2.8.0)
  Downloading pymongo-4.15.3-cp311-cp311-win_amd64.whl (859 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 859.2/859.2 kB 9.6 MB/s 0:00:00
Installing collected packages: pymongo
Successfully installed pymongo-4.15.3

C:\Users\IVANA>
```

```
Símbolo del sistema Microsoft Windows [Versión 10.0.26100.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\IVANA>pip install pymongo pandas
Requirement already satisfied: pymongo in c:\users\ivana\appdata\local\programs\python\python311\lib\site-packages (4.15.3)
Requirement already satisfied: pandas in c:\users\ivana\appdata\local\programs\python\python311\lib\site-packages (2.3.3)
Requirement already satisfied: dns
```

## CÓDIGO Y RESULTADOS DEL SCRIPT DE PYTHON

```
# Título: Ejemplos de MongoDB
# Autor: Ivana Sánchez Pérez

# colores
azul = '\033[94m'
rosa = '\033[38;5;200m'
turquesa = '\033[38;5;44m'
reset = '\033[0m'

print(f"\n{rosa}===== Ejemplos de MongoDB ====={reset}")
print()

from pymongo import MongoClient
import pandas as pd

# Conexión
uri = "mongodb+srv://isanper504a:PeLuchingordi111%40@cluster1.eibgjl7.mongodb.net/alumnoDB"
client = MongoClient(uri)
db = client["alumnoDB"]
col = db["rendimiento"]

# 1) Ver algunos documentos (sin mostrar el _id)
docs = col.find({}, {"_id": 0}).limit(5)
df = pd.DataFrame(list(docs))
print(f"{azul}Primeros registros:{reset}")
print(f"{turquesa}{df}{reset}")

# 2) Agregación: promedio por género
pipeline = [
    {
        "$group": {
            "_id": "$gender",
            "avgMath": {"$avg": "$math score"},
            "avgReading": {"$avg": "$reading score"},
```

```

        "avgWriting": {"$avg": "$writing score"},  

        "count": {"$sum": 1}  

    }  

},  

{"$sort": {"avgMath": -1}}  

]  

result = list(col.aggregate(pipeline))  

print(f"\n{azul}\nPromedio de puntuaciones por género:{reset}")  

for r in result:  

    print(f"\n{turquesa}{r}{reset}")  

# 3) Consulta: top 10 por math  

top_math = list(col.find({}, {"_id":0, "name":1, "math score":1}).sort("mathscore", -1).limit(10))  

print(f"\n{azul}Top math:{reset}", top_math)  

# 4) Consulta con filtro y ordenación  

top_math = list(col.find({}, {"_id": 0, "gender": 1, "math score": 1})  

    .sort("math score", -1)  

    .limit(5))  

print(f"\n{azul}Top 5 alumnos con mejor nota en matemáticas:{reset}")  

print(pd.DataFrame(top_math))  

# 5) Estudiantes con math score >= 80  

filtro = {"math score": {"$gte": 80}}  

resultados = list(col.find(filtro, {"_id": 0, "gender": 1, "math score": 1}))  

print(f"\n{azul}Estudiantes con math score >= 80:{reset}")  

print(pd.DataFrame(resultados))  

pipeline = [  

    {  

        "$group": {  

            "_id": "$test preparation course",  

            "avgMath": {"$avg": "$math score"},  

            "avgReading": {"$avg": "$reading score"},  

            "avgWriting": {"$avg": "$writing score"},  

            "total": {"$sum": 1}  

        }  

    },  

    {"$sort": {"avgMath": -1}}  

]  

result = list(col.aggregate(pipeline))  

print(f"\n{azul}Promedio según haber hecho curso de preparación o no:{reset}")  

for r in result:  

    print(f"\n{turquesa}{r}{reset}")

```

## RESULTADO DE LA EJECUCIÓN

```
C:\Users\IVANA\Desktop>python ejemplo_python.py
===== Ejemplos de MongoDB =====

Primeros registros:
gender race/ethnicity parental level of education lunch test preparation courses math score reading score writing score
0 male group A associate's degree free/reduced none 47 87 84
1 male group B some college free/reduced none 40 43 39
2 female group B high school standard none 65 81 73
3 female group C some high school standard none 69 75 78
4 male group C high school standard none 88 89 86

Promedio de puntuaciones por género:
{"_id": "male", "avgMath": 68.72821576763485, "avgReading": 65.47382904564316, "avgWriting": 63.31120331950208, "count": 482}
{"_id": "female", "avgMath": 63.63320633249635, "avgReading": 72.66816161610811, "avgWriting": 72.46718146718146, "count": 518}

Top math: [{"math score": 67}, {"math score": 69}, {"math score": 88}, {"math score": 69}, {"math score": 54}, {"math score": 69}, {"math score": 40}, {"math score": 47}, {"math score": 65}, {"math score": 73}]

Top 5 alumnos con mejor nota en matemáticas:
gender math score
0 female 100
1 male 100
2 female 100
3 male 100
4 male 100

Estudiantes con math score >= 80:
gender math score
0 female 100
1 male 100
2 female 80
3 male 80
4 male 100
5 male ...
188 female 80
189 male 80
190 male 80
191 male 80
192 male 80

[193 rows x 2 columns]

Promedio según haber hecho curso de preparación o no:
{"_id": "completed", "avgMath": 69.6955397625699, "avgReading": 73.89385474869335, "avgWriting": 74.41899441340782, "total": 358}
{"_id": "none", "avgMath": 64.0778816199377, "avgReading": 66.53426791277259, "avgWriting": 64.56467289719626, "total": 602}
C:\Users\IVANA\Desktop>
```