

lunes, 12 de mayo de 2025 11:57

Es un sistema que permite:

Es un sistema que permite:

- Pipelines de agregación (consultas como SUM, AVG, GROUP BY...)
- Transacciones

**Documento**

Es la forma nativa de organizar los datos como un conjunto de parejas de campo/clave-valor.

Es un conj

Es un conjunto de documentos que pueden o no, tener una estructura parecida; una bdd puede tener muchas colecciones y cada colección puede tener mucho documentos.

Es un grupo

Es un grupo de servidores que tienen copias de los datos, para garantizar disponibilidad.

Esa máqui

Esa máquina (local o remota) que sirve el servicio de MongoDB.

Es un co

Es un conjunto de instancias.

Aunque no es la forma más común de consultar en bdd no relaciones, existe la posibilidad de crear índices.

Por defecto existe un índice para `_id`.

Por ejemplo, si asignáramos un índice a "score" para que se ordene ascendentemente, la siguiente consulta es más eficiente porque ya está ordenado:

```
$ db.users.createIndex({"score": 1})
```



de darle permisos al script de instalación:

script de instalación ( `.sh` ) es:

```
chmod +x nombre_del_script.sh
```

📌 **Ejemplo típico.**

```
bash
```

En MongoDB c

regex).

- Los datos se van a usar juntos, deberían guardarse juntos

))))

```
db.contacts.insertOne({phone:91123123, name: 'Sofia' }) --> Error
```

```
--JSONArray
```

```
mongoimport --db <nombreBD> --collection <nombreColeccion> --file <ruta/al/archivo>
```

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

```
sample_airbnb> show collections
listingsAndReviews
```

```
test> use sample_airbnb
switched to db sample_airbnb
sample_airbnb> show collections
listingsAndReviews
sample_airbnb> db.list
db.listCommands          db.listingsAndReviews

sample_airbnb> db.listingsAndReviews
sample_airbnb.listingsAndReviews
sample_airbnb> db.listingsAndReviews.findOne()
{
  _id: '10000999',
  listing_url: 'https://www.airbnb.com/rooms/10000999',
  name: 'Horto Flat with small garden',
  summary: 'One bedroom + sofa-bed in quiet and bucolic neighbourhood right next to the Botanical Garden. Small garden, outside shower, well equipped kitchen and bathroom with shower and tub. Easy for transport with many restaurants and beautiful cafes in the area.',
  space: 'Lovely one bedroom + sofa-bed in the living room, perfect for two but fits up to four comfortably. There's a small outside garden with a shower. There's a well equipped open kitchen with both 110V / 220V wall plugs and one bathroom with shower, tub and even a sauna machine! All newly refurbished!',
  description: 'One bedroom + sofa-bed in quiet and bucolic neighbourhood right next to the Botanical Garden. Small garden, outside shower, well equipped kitchen and bathroom with shower and tub. Easy for transport with many restaurants and beautiful cafes in the area. Lovely one bedroom + sofa-bed in the living room, perfect for two but fits up to four comfortably. There's a small outside garden with a shower. There's a well equipped open kitchen with both 110V / 220V wall plugs and one bathroom with shower, tub and even a sauna machine! All newly refurbished! I'll be happy to help you with any doubts, tips or any other information needed during your stay. This charming ground floor flat is located in Horto, a quiet and bucolic neighborhood just next to the Botanical Garden, where most of the descendants of it's first gardeners still live. You'll be 30 minutes walk from waterfalls in the rainforest with easy hiking trails! There are nice bars and restaurants as well as basic facilities - pharmacy, b',
  neighborhood_overview: 'This charming ground floor flat is located in Horto, a quiet and bucolic neighborhood just next to the Botanical Garden, where most of the descendants of it's first gardeners still live. You'll be 30 minutes walk from waterfalls in the rainforest with easy hiking trails! There are nice bars and restaurants as well as basic facilities - pharmacy, bakery, small market - in the area.',
  amenities: 'There's a table in the living room now, that does not show in the photos.',
  transit: 'Easy access to transport (bus, taxi, car) and easy free parking around. Very close to Gávea, Leblon, Ipanema, Copacabana and Botafogo.',
  access: '',
  interaction: 'I'll be happy to help you with any doubts, tips or any other information needed during your stay.',
  house_rules: 'I just hope the guests treat the space as they're own, with respect to it as well as to my neighbours!',
  reviews_summary: 'que os hóspedes tratem o lugar com carinho e respeito aos vizinhos!',
  property_type: 'Apartment',
  room_type: 'Entire home/apt',
```

--> Devuelve UN elemento de la colección

```
sample_airbnb> db.listingsAndReviews.find()
```

... Devuelve todos los elementos de la colección... si hay más de 20, devuelve los primeros 20 y da la opción de seguir iterando en ellos de 20 en 20.

```
sample_airbnb>
```

```
sample_airbnb> db.listingsAndReviews.find().count()
5555
sample_airbnb>
```

```
sample_airbnb> db.listingsAndReviews.find({'property_type': 'Apartment'}).count()
3526
sample_airbnb>
```

--> db.<collection\_name>.find(query)

```
sample_training> db.createCollection("estudiantes_prueba_borrar")
{ ok: 1 }
sample_training> show collections
companies
estudiantes_prueba
estudiantes_prueba_borrar
grades
inspections
posts
routes
stories
trips
tweets
zips
sample_training> db.estudiantes_prueba_borrar.drop()
true
sample_training> show collections
companies
estudiantes_prueba
grades
inspections
posts
routes
stories
trips
tweets
zips
sample_training>
```

```
sample_training> db.estudiantes_prueba.insertOne({"nombre": "Orianna", "apellido": "Milone", "edad": 23, "carrera": "Ing. Informática", "promedio": 8.12})
{
  acknowledged: true,
  insertedId: ObjectId('6821fd82649697e019b71236')
}
sample_training> db.estudiantes_prueba.find()
[
  {
    id: ObjectId('6821fd82649697e019b71236'),
    nombre: 'Orianna',
    apellido: 'Milone',
    edad: 23,
    carrera: 'Ing. Informática',
    promedio: 8.12
  }
]
```

```

sample_training> db.estudiantes_prueba.updateOne({"nombre": "Orianna"}, {"$set": {"asignaturas": ["Sist. Web II", "RRHH"]}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
sample_training> db.estudiantes_prueba.find()
[
  {
    _id: ObjectId('6821fd82649697e019b71236'),
    nombre: 'Orianna',
    apellido: 'Milone',
    edad: 23,
    carrera: 'Ing. Informática',
    promedio: 8.12,
    asignaturas: [ 'Sist. Web II', 'RRHH' ]
  }
]
sample_training>

```

```

db.estudiantes_prueba.updateOne(
  { nombre: "Orianna" }, // filtro
  { $set: { asignaturas: ["Web", "Ciberseguridad"] } } // operación
)

```

```

sample_training> db.estudiantes_prueba.updateOne({"nombre": "Orianna"}, {"$inc": {"edad": 1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
sample_training> db.estudiantes_prueba.find()
[
  {
    _id: ObjectId('6821fd82649697e019b71236'),
    nombre: 'Orianna',
    apellido: 'Milone',
    edad: 24,
    carrera: 'Ing. Informática',
    promedio: 8.12,
    asignaturas: [ 'Sist. Web II', 'RRHH' ]
  }
]
sample_training>

```

```

sample_training> db.estudiantes_prueba.updateOne({"nombre": "Orianna"}, {"$push": {"asignaturas": "TFG"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
sample_training> db.estudiantes_prueba.find()
[
  {
    _id: ObjectId('6821fd82649697e019b71236'),
    nombre: 'Orianna',
    apellido: 'Milone',
    edad: 24,
    carrera: 'Ing. Informática',
    promedio: 8.12,
    asignaturas: [ 'Sist. Web II', 'RRHH', 'TFG' ]
  }
]

```

```

sample_training> db.estudiantes_pruebas.insertMany([
  {
    nombre: 'Luisa',
    apellido: 'Martínez',
    edad: 23,
    carrera: 'Ing. Informática',
    promedio: 7.85,
    asignaturas: [ 'Redes', 'Sist. Web II', 'Bases de Datos' ]
  },
  {
    nombre: 'Carla',
    apellido: 'Fernández',
    edad: 25,
    carrera: 'Ing. Industrial',
    promedio: 8.9,
    asignaturas: [ 'Física', 'Termodinámica', 'Proyecto' ]
  },
  {
    nombre: 'Tomás',
    apellido: 'Gómez',
    edad: 22,
    carrera: 'Ing. Informática',
    promedio: 6.7,
    asignaturas: [ 'Matemáticas', 'Compiladores', 'TFG' ]
  },
  {
    nombre: 'Lucía',
    apellido: 'Pérez',
    edad: 24,
    carrera: 'Ing. Informática',
    promedio: 9.3,
    asignaturas: [ 'Sist. Web II', 'IA', 'TFG' ]
  },
  {
    nombre: 'Diego',
    apellido: 'Ruiz',
    edad: 23,
    carrera: 'Ing. de Telecomunicaciones',
    promedio: 7.5,
    asignaturas: [ 'Comunicaciones', 'Electrónica', 'Antenas' ]
  }
])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68220126649697e019b71237'),
    '1': ObjectId('68220126649697e019b71238'),
    '2': ObjectId('68220126649697e019b71239'),
    '3': ObjectId('68220126649697e019b7123a'),
    '4': ObjectId('68220126649697e019b7123b')
  }
}

```

```
sample_training> db.estudiantes_prueba.updateMany({}, { '$push' :{'asignaturas': 'Prácticas'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 6,
  modifiedCount: 6,
  upsertedCount: 0
}
```

```
sample_training> db.estudiantes_prueba.find()
[
  {
    _id: ObjectId('6821fd82649697e019b71236'),
    nombre: 'Orianna',
    apellido: 'Milone',
    edad: 24,
    carrera: 'Ing. Informática',
    promedio: 8.12,
    asignaturas: [ 'Sist. Web II', 'RRHH', 'TFG', 'Prácticas' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123c'),
    nombre: 'Lucas',
    apellido: 'Martínez',
    edad: 23,
    carrera: 'Ing. Informática',
    promedio: 7.85,
    asignaturas: [ 'Redes', 'Sist. Web II', 'Bases de Datos', 'Prácticas' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123d'),
    nombre: 'Carla',
    apellido: 'Fernández',
    edad: 25,
    carrera: 'Ing. Industrial',
    promedio: 8.9,
    asignaturas: [ 'Física', 'Termodinámica', 'Proyecto', 'Prácticas' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123e'),
    nombre: 'Tomas',
    apellido: 'Gómez',
    edad: 22,
    carrera: 'Ing. Informática',
    promedio: 6.7,
    asignaturas: [ 'Matemáticas', 'Compiladores', 'TFG', 'Prácticas' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123f'),
    nombre: 'Lucia',
    apellido: 'Pérez',
    edad: 24,
    carrera: 'Ing. Informática',
    promedio: 9.3,
    asignaturas: [ 'Sist. Web II', 'IA', 'TFG', 'Prácticas' ]
  },
  {
    _id: ObjectId('68220155649697e019b71240'),
    nombre: 'Diego',
    apellido: 'Ruiz',
    edad: 23,
    carrera: 'Ing. de Telecomunicaciones',
    promedio: 7.5,
    asignaturas: [ 'Comunicaciones', 'Electrónica', 'Antenas', 'Prácticas' ]
  }
]
```

```
sample_training> db.estudiantes_prueba.updateMany({}, { '$push' :{'hobbies': { '$each': [ 'Lectura', 'Deportes' ] }}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 6,
  modifiedCount: 6,
  upsertedCount: 0
}
sample_training> db.estudiantes_prueba.find()
[
  {
    _id: ObjectId('6821fd82649697e019b71236'),
    nombre: 'Orianna',
    apellido: 'Milone',
    edad: 24,
    carrera: 'Ing. Informática',
    promedio: 8.12,
    asignaturas: [ 'Sist. Web II', 'RRHH', 'TFG', 'Prácticas' ],
    hobbies: [ 'Lectura', 'Deportes' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123c'),
    nombre: 'Lucas',
    apellido: 'Martínez',
    edad: 23,
    carrera: 'Ing. Informática',
    promedio: 7.85,
    asignaturas: [ 'Redes', 'Sist. Web II', 'Bases de Datos', 'Prácticas' ],
    hobbies: [ 'Lectura', 'Deportes' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123d'),
    nombre: 'Carla',
    apellido: 'Fernández',
    edad: 25,
    carrera: 'Ing. Industrial',
    promedio: 8.9,
    asignaturas: [ 'Física', 'Termodinámica', 'Proyecto', 'Prácticas' ],
    hobbies: [ 'Lectura', 'Deportes' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123e'),
    nombre: 'Tomas',
    apellido: 'Gómez',
    edad: 22,
    carrera: 'Ing. Informática',
    promedio: 6.7,
    asignaturas: [ 'Matemáticas', 'Compiladores', 'TFG', 'Prácticas' ],
    hobbies: [ 'Lectura', 'Deportes' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123f'),
    nombre: 'Lucia',
    apellido: 'Pérez',
    edad: 24,
    carrera: 'Ing. Informática',
    promedio: 9.3,
    asignaturas: [ 'Sist. Web II', 'IA', 'TFG', 'Prácticas' ],
    hobbies: [ 'Lectura', 'Deportes' ]
  },
  {
    _id: ObjectId('68220155649697e019b71240'),
    nombre: 'Diego',
    apellido: 'Ruiz',
    edad: 23,
    carrera: 'Ing. de Telecomunicaciones',
    promedio: 7.5,
    asignaturas: [ 'Comunicaciones', 'Electrónica', 'Antenas', 'Prácticas' ],
    hobbies: [ 'Lectura', 'Deportes' ]
  }
]
```

```
sample_training> db.estudiantes_prueba.find({}, {"promedio": 0, "hobbies": 0, "nombre": 0})
[
  {
    _id: ObjectId('6821fd82649697e019b71236'),
    apellido: 'Milone',
    edad: 24,
    carrera: 'Ing. Informática',
    asignaturas: [ 'Sist. Web II', 'RRHH', 'TFG', 'Prácticas' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123c'),
    apellido: 'Martínez',
    edad: 23,
    carrera: 'Ing. Informática',
    asignaturas: [ 'Redes', 'Sist. Web II', 'Bases de Datos', 'Prácticas' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123d'),
    apellido: 'Fernández',
    edad: 25,
    carrera: 'Ing. Industrial',
    asignaturas: [ 'Física', 'Termodinámica', 'Proyecto', 'Prácticas' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123e'),
    apellido: 'Gómez',
    edad: 22,
    carrera: 'Ing. Informática',
    asignaturas: [ 'Matemáticas', 'Compiladores', 'TFG', 'Prácticas' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123f'),
    apellido: 'Pérez',
    edad: 24,
    carrera: 'Ing. Informática',
    asignaturas: [ 'Sist. Web II', 'IA', 'TFG', 'Prácticas' ]
  },
  {
    _id: ObjectId('68220155649697e019b71240'),
    apellido: 'Ruiz',
    edad: 23,
    carrera: 'Ing. de Telecomunicaciones',
    asignaturas: [ 'Comunicaciones', 'Electrónica', 'Antenas', 'Prácticas' ]
  }
]
sample_training>
```

```
sample_training> db.estudiantes_prueba.find({}, {"promedio": 1, "nombre": 1, "apellido": 1})
[
  {
    _id: ObjectId('6821fd82649697e019b71236'),
    nombre: 'Orianna',
    apellido: 'Milone',
    promedio: 8.12
  },
  {
    _id: ObjectId('68220155649697e019b7123c'),
    nombre: 'Lucas',
    apellido: 'Martínez',
    promedio: 7.85
  },
  {
    _id: ObjectId('68220155649697e019b7123d'),
    nombre: 'Carla',
    apellido: 'Fernández',
    promedio: 8.9
  },
  {
    _id: ObjectId('68220155649697e019b7123e'),
    nombre: 'Tomás',
    apellido: 'Gómez',
    promedio: 6.7
  },
  {
    _id: ObjectId('68220155649697e019b7123f'),
    nombre: 'Lucía',
    apellido: 'Pérez',
    promedio: 9.3
  },
  {
    _id: ObjectId('68220155649697e019b71240'),
    nombre: 'Diego',
    apellido: 'Ruiz',
    promedio: 7.5
  }
]
sample_training>
```

```
sample_training> db.estudiantes_prueba.find({"apellido": {"$regex": "ez"}})
[
  {
    _id: ObjectId('68220155649697e019b7123c'),
    nombre: 'Lucas',
    apellido: 'Martínez',
    edad: 23,
    carrera: 'Ing. Informática',
    promedio: 7.85,
    asignaturas: [ 'Redes', 'Sist. Web II', 'Bases de Datos', 'Prácticas' ],
    hobbies: [ 'Lectura', 'Deportes' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123d'),
    nombre: 'Carla',
    apellido: 'Fernández',
    edad: 25,
    carrera: 'Ing. Industrial',
    promedio: 8.9,
    asignaturas: [ 'Física', 'Termodinámica', 'Proyecto', 'Prácticas' ],
    hobbies: [ 'Lectura', 'Deportes' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123e'),
    nombre: 'Tomás',
    apellido: 'Gómez',
    edad: 22,
    carrera: 'Ing. Informática',
    promedio: 6.7,
    asignaturas: [ 'Matemáticas', 'Compiladores', 'TFG', 'Prácticas' ],
    hobbies: [ 'Lectura', 'Deportes' ]
  },
  {
    _id: ObjectId('68220155649697e019b7123f'),
    nombre: 'Lucía',
    apellido: 'Pérez',
    edad: 24,
    carrera: 'Ing. Informática',
    promedio: 9.3,
    asignaturas: [ 'Sist. Web II', 'IA', 'TFG', 'Prácticas' ],
    hobbies: [ 'Lectura', 'Deportes' ]
  }
]
sample_training>
```