



Bengoechea Patricio
Fortuna Juan





- Es un sistema base de datos NoSql (No relacional) orientado a documentos,
- De código abierto.
- Estructura de datos BSON
- Operaciones CRUD
- Pymongo



A diferencia de las bases de datos relacionales que se estructuran en tablas, MongoDB se estructura en documentos. Más específicamente en documentos BSON que es muy similar a JSON. Con la diferencia de que BSON es más eficiente para el almacenamiento y la transferencia.

Si están familiarizados con JSON, un documento de BSON es esencialmente un documento JSON con algunas características adicionales. Desde la perspectiva de Python es un diccionario.



Documento

Un registro en MongoDB es un documento, es un dato estructurado compuesto de un par campo-valor, similar a un diccionario de Python. Muy parecido a un objeto JSON. El valor de un campo puede incluir otros documentos, arreglos o arreglos de documentos.



Aspectos de MongoDB

Las principales diferencias entre MongoDB (orientado a documentos) y una base de datos tradicional (Relacional) son:


- 1- MongoDB no soporta Joins.
- 2- MongoDB no soporta transacciones. No se soportan operaciones atomizadas.
- 3- Los esquemas de MongoDB son flexibles. No todos los documentos en una colección deben cumplir con el mismo esquema.



Cuadro Comparativo SQL - MongoDB

Concept	SQL	MongoDB
One User	One Row	One Document
All Users	Users Table	Users Collection
One Username Per User (1-to-1)	Username Column	Username Property
Many Emails Per User (1-to-many)	SQL JOIN with Emails Table	Embed relevant email doc in User Document
Many Items Owned by Many Users (many-to-many)	SQL JOIN with Items Table	Programmatically Join with Items Collection

Tabla VS Documento



student_id	age	score
1	12	77
2	12	68
3	11	75



```
[
  {
    "student_id":1,
    "age":12,
    "score":77
  },
  {
    "student_id":2,
    "age":12,
    "score":68
  },
  {
    "student_id":3,
    "age":11,
    "score":75
  }
]
```



Operaciones CRUD en





Create - Crear

```
db.collection.insertOne()  
db.collection.insertMany()
```

```
db.users.insertOne(  ← collection  
  {  
    name: "sue",      ← field: value  
    age: 26,          ← field: value  
    status: "pending" ← field: value } document  
  }  
)
```



Read - Leer

```
db.collection.find()
```

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
) .limit(5)
```

← collection
← query criteria
← projection
← cursor modifier



Update - Actualizar

```
db.collection.updateOne()  
db.collection.updateMany()  
db.collection.replaceOne()
```

```
db.users.updateMany(  
  { age: { $lt: 18 } },  
  { $set: { status: "reject" } }  
)
```

← collection
← update filter
← update action



Delete - Borrar

```
db.collection.deleteOne()  
db.collection.deleteMany()
```

```
db.users.deleteMany(  
  { status: "reject" }  
)
```

← collection
← delete filter



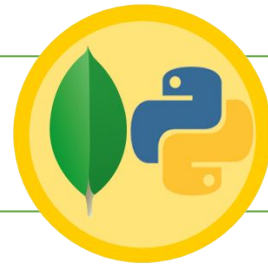
PyMongo



PyMongo es la librería de python que nos permite utilizar MongoDB como base de datos en nuestros proyectos.



Ejemplos PyMongo



```
import sys

from pymongo import MongoClient
from pymongo.errors import ConnectionFailure

def main():
    "Connect to MongoDB"
    try:
        c = MongoClient(
            host="localhost",
            port=27017
        )
        print("Connected successfully")
    except ConnectionFailure:
        sys.stderr.write("Could not connect to MongoDB")
        sys.exit(1)
```

```
db = c['mydb'] #<<<Crea la Base de Datos

#Crea el documento
user_doc = {
    'username': us,
    'nombre': name,
    'apellido': surname,
    'fechadenacimiento': dateofbith,
    'email': email,
}

db.users.insert(user_doc) #<<< Inserta el documento en la coleccion "Users"
print ('Carga exitosa en el documento: ', user_doc)
```



MongoDB Compass

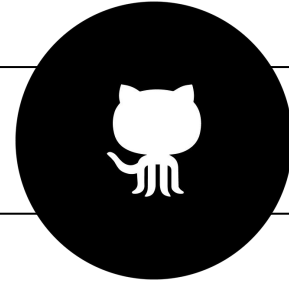
Interfaz gráfica de usuario de MongoDB donde se puede:

- Explorar los datos visualmente
- Ejecutar consultas ad hoc
- Interactúe con los datos gracias a todas las funciones CRUD.
- Visualizar y optimizar el rendimiento de las consultas.

Disponible para Linux, Mac o Windows. Compass le permite tomar mejores decisiones en cuanto a indexación, validación de documentos y mucho más.



Repo GIT con ejemplos



<https://github.com/junnchy/presentacionMongoDB2019>