



Universidad Veracruzana
Facultad de Estadística e Informática



Tema

Documentos JSON: Formato y consulta

Base de datos MongoDB

Elaborado por:

Mtra. Lorena Alonso Ramírez

lalonso@uv.mx



Última actualización: mayo 2022

Documentos JSON

- JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos.
- Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo
- JSON está constituido por una **colección de pares de nombre/valor**.

Ejemplo 1

- El objeto Json está formado por los campos: nombre, edad, aficiones y residencia.
- Los valores de los campos son: cadenas de caracteres, valores numéricos, arreglos.

```
{
  Nombre : "Juan",
  Edad: 28,
  Aficiones: ["Música", "Cine", "Tenis"],
  Residencia: "Madrid"
}
```

Ejemplo 2

- En este ejemplo el valor del campo responsable es a su vez un objeto json.
- El responsable, está definido por nombre, edad, aficiones y Residencia.
- Los elementos de un arreglo pueden ser objetos json

```
{
  "responsable":
  {
    "Nombre": "Juan",
    "Edad": 28,
    "Aficiones": ["Música", "Cine", "Tenis"],
    "Residencia": "Madrid"
  },
  "empleados":
  [
    { "Nombre" : "Elena",
      "Edad": 26,
      "Aficiones": ["Música", "Cine"],
      "Residencia": "Madrid"
    },
    { "Nombre" : "Luis",
      "Edad": 31,
      "Residencia": "Madrid"
    }
  ]
}
```

MongoDB

Consulta y actualización de datos

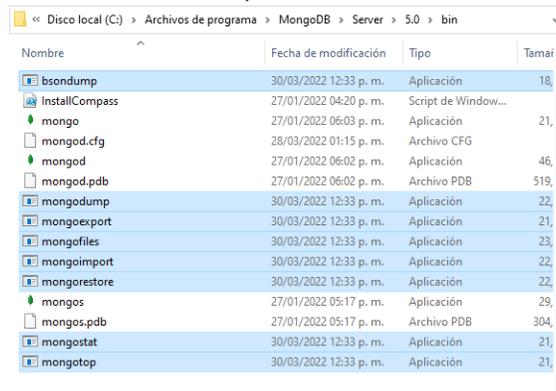
En esta práctica se trabaja con una colección de documentos donde se almacena información sobre diferentes restaurantes.

1. Los documentos tienen la siguiente estructura. **Identifica** los elementos del documento, los subelementos, y los arreglos:

```
db.restaurants.insert( {
  "address" : { "street" : "2 Avenue",
               "zipcode" : "10075",
               "building" : "1480",
               "coord" : [ -73.9557413, 40.7720266 ]
            },
  "borough" : "Manhattan",
  "cuisine" : "Italian",
  "grades" : [
    { "date" : ISODate("2014-10-01T00:00:00Z"), "grade" : "A", "score" : 11 },
    { "date" : ISODate("2014-01-16T00:00:00Z"), "grade" : "B", "score" : 17 }
  ],
  "name" : "Vella",
  "restaurant_id" : "41704620"
})
```

2. Descargar archivo con datos:
<https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json>
3. Descargar la herramienta para poder importar datos
(MongoDB Database Tools - https://www.mongodb.com/try/download/database-tools?tck=docs_databasetools)

Descomprimir y copiar los archivos en la carpeta 'bin' de la instalación de mongodb



Nombre	Fecha de modificación	Tipo	Tamaño
bsondump	30/03/2022 12:33 p. m.	Aplicación	18,
InstallCompass	27/01/2022 04:20 p. m.	Script de Window...	
mongo	27/01/2022 06:03 p. m.	Aplicación	21,
mongod.cfg	28/03/2022 01:15 p. m.	Archivo CFG	
mongod	27/01/2022 06:02 p. m.	Aplicación	46,
mongod.pdb	27/01/2022 06:02 p. m.	Archivo PDB	519,
mongodump	30/03/2022 12:33 p. m.	Aplicación	22,
mongoexport	30/03/2022 12:33 p. m.	Aplicación	21,
mongofiles	30/03/2022 12:33 p. m.	Aplicación	23,
mongoimport	30/03/2022 12:33 p. m.	Aplicación	22,
mongorestore	30/03/2022 12:33 p. m.	Aplicación	22,
mongos	27/01/2022 05:17 p. m.	Aplicación	29,
mongos.pdb	27/01/2022 05:17 p. m.	Archivo PDB	304,
mongostat	30/03/2022 12:33 p. m.	Aplicación	21,
mongotop	30/03/2022 12:33 p. m.	Aplicación	21,

4. Importar los datos en MongoDB. **Desde la línea de comandos** ejecutar el comando mongoimport (modifica la ruta de tus datos y la ruta donde está instalado Mongo):

```
"C:\Program Files\MongoDB\Server\5.0\bin\mongoimport" --db ejemplo --collection restaurants --drop --file c:\mongoData\primer-dataset.json
```

Con éste comando se crea una base de datos llamada 'ejemplo', con una colección llamada 'restaurants'

A continuación se listan una serie de consultas (ejecutadas desde el cliente de mongo), algunas ya incluyen la respuesta con la finalidad de que sirva de ejemplo, y así aprender la sintaxis de consulta en mongoDB.

Escribir las consultas faltantes y capturar la pantalla de respuesta (solo una pantalla por consulta, aunque no se visualice la totalidad de la respuesta arrojada)

5. Consultar en base a un campo del documento
 - a. Mostrar los restaurantes del barrio (borough) de Manhattan
`db.restaurants.find({ "borough": "Manhattan" })`
 - b. Mostrar los restaurantes de cocina Italiana
 - c. Mostrar los datos del restaurant "Regina Caterers"
6. Consulta en base a un campo de un documento anidado
 - a. Mostrar los restaurantes que tengan el código postal "10075"
`db.restaurants.find({ "address.zipcode": "10075" })`
 - b. Mostrar los restaurantes ubicados en la calle "Nostrand Avenue"
7. Consultar por un campo dentro de un arreglo
 - a. Mostrar los restaurantes que tengan un calificación (grade)"B"
`db.restaurants.find({ "grades.grade": "B" })`
8. Uso de operadores relacionales
 - a. Mostrar los restaurantes que tengan un puntaje mayor a 30
`db.restaurants.find({ "grades.score": { $gt: 30 } })`
 - b. Mostrar los restaurantes que tengan un puntaje menor a 10
 - c. Mostrar los restaurantes que tengan un puntaje mayor a 20
9. Uso del AND lógico en los criterios de búsqueda (condiciones compuestas)
 - a. Mostrar los restaurantes de cocina italiana que se ubique en el código postal 10075

```
db.restaurants.find( { "cuisine": "Italian", "address.zipcode":  
"10075" } )
```

- b. Mostrar los restaurantes de cocina "American" ubicados en "Brooklyn"
- c. Mostrar los restaurantes de "Queens" que tengan un puntaje mayor a 20

10. Uso del OR lógico

- a. Mostrar los restaurantes de cocina italiana, o los que se encuentren en el código postal 10075

```
db.restaurants.find( { $or: [ { "cuisine": "Italian" }, {  
"address.zipcode": "10075" } ] } )
```

- b. Mostrar los restaurantes de cocina italiana o cocina American
- c. Mostrar los restaurantes con calificación A ó B
- d. Mostrar los restaurantes de cocina italiana con calificación A ó B

11. Contar

- a. Contar los restaurantes que tengan comida italiana

```
db.restaurants.find({"cuisine":"Italian"}).count()
```

12. Ordenar resultados (Sort)

- a. Ordenar los restaurantes por barrio (borough, y código postal)

```
db.restaurants.find().sort( { "borough": 1, "address.zipcode":1 } )
```

- b. Ordenar los restaurantes por tipo de cocina

13. Modificar datos. **Comenta** cómo funcionan cada una de las siguientes instrucciones.

- a. Actualizar campos

```
db.restaurants.update(  
{ "name" : "Juni" },  
{ $set: { "cuisine": "American (New)" },  
$currentDate: { "lastModified": true }  
} )
```

- b. Actualizar campos de un documento anidado

```
db.restaurants.update(  
{ "restaurant_id" : "41156888" },  
{ $set: { "address.street": "East 31st Street" }  
} )
```

- c. Actualizar múltiples documentos

```
db.restaurants.update(  
{ "cuisine": "Italian" },  
{ $set: { "rating": 20 } },  
{ "multi": true } )
```

```
{ "address.zipcode": "10016", cuisine: "Other" },
{ $set: { cuisine: "Category To Be Determined" },
$currentDate: { "lastModified": true } },
{ multi: true}
)
```

- d. Reemplazar un documento

```
db.restaurants.update(
{ "restaurant_id" : "41704620" },
{ "name" : "Vella 2",
"address" : { "coord" : [ -73.9557413, 40.7720266 ] ,
"building" : "1480",
"street" : "2 Avenue",
"zipcode" : "10075" }
} )
```
- e. Borrar todos los documentos que cumplan con una condición

```
db.restaurants.remove( { "borough": "Manhattan" } )
```
- f. Borrar un documento que cumpla con una condición

```
db.restaurants.remove(
{"borough": "Queens" },
{ justOne: true } )
```

14. Otras consultas

- a. Mostrar el nombre y el tipo de cocina de los restaurantes cuyo tipo de cocina sea donas o repostería
- b. Mostrar el nombre y calificación de los restaurantes cuya calificación sea menor a 5.
- c. Mostrar los restaurantes de cocina "American" ubicados en "Queens"
- d. Mostrar el nombre de los restaurantes de cocina Delicatessen, o los que se encuentren en el código postal 11106.
- e. Listar todos los restaurantes de clasificación A

15. Define 5 nuevas consultas. Escribe el texto y la consulta en mongo