¿Qué es MongoDB?

Mongo es una base de datos documental (NOSQL).

Trabaja con documentos los cuales forman colecciones; comparando con las bases de datos relacionales, las colecciones son tablas.

A diferencia de las tablas estas permiten guardar documentos con diferentes formatos, no están sometidos a un esquema fijo y los documentos son registros.

Los documentos en Mongo se asemejan mucho a los diccionarios de Python; estos se almacenan en BSON que es un formato modificado de JSON que agiliza la búsqueda de los datos.

Un documento en mongo podría verse así:

{
   "\_id": ObjectId("4efa8d2b744dap101esbc8"),
   "nombre": "Paula",
   "apellido": "Velez",
   "telefono": "12345667",
}

Mongo está escrito en C++, lo que hace que sea bastante rápida a  la hora de ejecutar tareas. Actualmente lidera las bases de datos NOSQL.

Ventajas al usar Mongo

-   Velocidad
-   Sencillo sistema de consultas
-   Balance entre rendimiento y funcionalidad
-   Escalabilidad

Comenzando con MONGO DB

**1.** Conectarse al servidor de la base de datos, desde la consola escribimos:

$ mongod

**Nota:** Por default mongo corre por el puerto 27017; pero se puede cambiar.

**2.** Una vez corriendo el servidor podremos entra al Shell de mongo

$ mongo

**Nota**: En el Shell de mongo se trabaja con javascript

**3.**Seleccionar la base de datos

> **use** \_nombreBaseDatos\_

Se usa db para saber qué base de datos estamos usando, si queremos listar todas las bases de datos usamos:

> show dbs

Con drop borramos todo lo que haya en la base de datos

> db.drop( )

**4.** Crear documentos e insertarlos en colecciones

Ej
Documento

> p = { name: paula }

Insertar a colección

> db.\_nombreColeccion\_.insert( p )

Ó

> db.\_nombreColeccion\_.save( { name: paula } )

Para ver las colecciones de nuestra base de datos usamos

> show collections

**Nota:**al insertar el primer documento a la colección mongo crea tanto la base de datos como la colección.

Para más información la página de MONGO DB ofrece una documentación muy completa [**http://docs.mongodb.org/manual/**](http://docs.mongodb.org/manual/)

Consultas en mongo

Las consultas en mongo son muy sencillas y de diferentes tipos, tamaños y sabores.

Ej.
Consultar todos los elementos de una colección

> db .\_nombreColeccion\_.find( { } )

Consultas con condiciones

> db .\_nombreColeccion\_.find( { <campo> : <valor> } )

Las consultas también pueden incluir operadores para especificar condiciones

> db .\_nombreColeccion\_.find ( {  <campo> : { $in: [  <valor> , <valor> ] } } )

Los operadores en las consultas no son solo de comparación también hay lógicos, de elementos, javascript, geoespaciales y  arreglos; algunos de estos son:

Comparación

-  **$all:** Compara todo el contenido especificado en la consulta
-  **$ne:** Compara que todos los valores no sean iguales al valor especificado en la consulta.
-  **$in:** Sirve para indicar que el valor de la consulta debe estar entre alguno de los proporcionados.
-  **$nin:** Hace lo contrario de $in.

Lógicos

-  **$or:** Indica que se debe cumplir al menos una de las condiciones pasadas en la consulta.
-  **$and:**  Indica que se deben cumplir todas las condiciones pasadas en la consulta.

Array

-  **$size:** Selecciona los documentos que cumplen con el tamaño especificado.

Javascript

**$where:** Se usa para pasar una cadena que contiene una expresión o una función de JavaScript, usar where proporciona mayor flexibilidad, pero requiere de mucho procesamiento de la base de datos.

Ej.

> db.users.find( { $where: **function**() { **return** (**this**.username == **this**.userlastname) } } )
> db.users.find( { $where:"this.username.length<=2" } )

**Nota:** Para saber más acerca de los operadores de Mongo DB en [**http://docs.mongodb.org/manual/reference/operator/**](http://docs.mongodb.org/manual/reference/operator/) encontras mas informacion al respecto.