# MongoDB índices

Los índices en MongoDB al igual que en otras bases de datos permite mejorar la velocidad de acceso a los documentos en una colección.

Es importante comprender de que manera serán accedidos los documentos durante la etapa de diseño con el objetivo de generar los índices correctos para cada colección.

## Tipos de índices

* Índices de campo único: Como su nombre lo indica son índices que se aplican a un único campo de los documentos de una colección; bien sea a nivel superior o en subdocumentos.

#### Ejemplos:

* db.zips.createIndex( { "city": 1 } );
* db.posts.createIndex( { "author.name": 1 } );
* Índices compuestos: Un índice compuesto incluye mas de uno de los campos de los documentos de una colección.

#### Ejemplo:

* db.zips.createIndex( { "state": 1, "city": 1 } );
* Índice Multikey: Son índices generados para atributos de tipo arreglo de los documentos de una colección, en cuyo caso MongoDB agrega una llave al índice para cada elementos en el arreglo.

#### Ejemplo:

* db.survey.createIndex( { ratings: 1 } );
* db.inventory.createIndex( { "stock.size": 1, "stock.quantity": 1 } );
* Índices Geoespaciales: Son índices que soportan búsquedas sobre datos almacenados en formato GeoJSON o par de coordenadas.
  + Esféricos (2dsphere): Son usados para calcular geometría en una esfera similar a la tierra. Se debe almacenar la información como GeoJSON siguiendo como referencia el sistema de coordenadas de la especificación [EPSG:4326: WGS 84](http://spatialreference.org/ref/epsg/4326/)
  + Planos (2d): Son usados para calcular distancias en planos euclidianos.

#### Ejemplos:

* db.antioquia.createIndex( { geometria: "2dsphere" } );
* db.zips.createIndex( { loc : "2d" });
* Índices textuales: Soportan búsqueda de cadenas de texto en el contenido de los documentos.
* Índices de hash: Mantienen las llaves del índice a través de hash de los valores del campo que esta siendo indexado.

#### Ejemplo:

* db.zips.createIndex( { state: "hashed" } );

# MongoDB consultas Geospaciales

## $near:

db.getCollection('cartografia').find( { geometria : { $near :{$geometry : {
type : "Point" ,
coordinates : [-75.2764301754797884,5.8576303469698701]
},
$maxDistance : 20000 }}});

## $nearSphere:

db.getCollection('zips').find( { loc : { $nearSphere :[-75.2764301754797884,5.8576303469698701],
$maxDistance : 0.4 }});