

# Índices

---

El objetivo del uso de índices es el de mejorar el rendimiento **de las consultas** sobre los datos de una colección.

Los índices son estructuras de datos que se definen sobre uno o más campos de una colección (almacenando internamente sus valores) y serán manejadas internamente por MongoDB. Los índices están ordenados en función de alguno de los valores de sus campos.

El funcionamiento de un índice es el siguiente. Cuando realicemos una consulta sobre una colección, MongoDB, como paso previo, hará la consulta sobre el índice. Puesto que el índice es una estructura mucho más ligera que la colección real la consulta será más rápida. Esto le permitirá a MongoDB determinar qué documentos de la colección real han de ser devueltos como respuesta a la consulta. De este modo se evitará realizar la consulta sobre el conjunto total de documentos de la colección, que sería mucho más lento.

**En resumen:** los índices son un intermediario rápido para evitar tener que realizar una consulta sobre el conjunto total de los datos de una colección.

Por defecto MongoDB, si no lo hacemos nosotros, crea un índice sobre el campo `_id` de cada colección.

## Buenas prácticas

---

Es conveniente crear los índices antes de que la colección contenga una gran cantidad de datos. Si no lo hacemos así la creación de los índices será más lenta. Por lo tanto se recomienda crear los índices que estimemos oportunos en el momento en que creamos la colección.

Si esperamos a tener muchos datos el proceso de creación será más lento además de que habremos perdido la oportunidad de que MongoDB haya ido creando los índices de forma progresiva y, de este modo, acelerado las consultas.

## Consultar los índices de una colección

---

Para comprobar qué índices hay definidos en una colección podremos usar el siguiente comando:

```
db.alumnos.getIndexes()
```

## Tipos de índices

---

- Índices simples (un solo campo).
- Índices compuestos (varios campos)
- Índices únicos.
- Índices *sparse*.
- Otros: *sparse*, geoespaciales, TTL, hash, texto.

## Índices simples

---

Son los índices más sencillos. Se definen sobre un solo campo de la colección.

# Creación de un índice simple

## Sintaxis del comando `db.collection.createIndex()`

El comando `createIndex` admite tres argumentos, el primero obligatorio y los otros dos opcionales. En el primer argumento hemos de indicar sobre qué campos de la colección vamos a crear el índice si su *tipo* (orden ascendente o descendente). En el segundo documento podremos indicar opciones sobre cómo han de crearse los índices. Finalmente el último argumento indica a mongo cuantos nodos han de confirmar que han creado el índice para que la operación de creación de índice que considere exitosa.

Por ejemplo, si queremos crear un índice sobre el campo `nombre` de la colección `alum` lo haremos de la siguiente manera:

```
db.alumnos.createIndex( { nombre: 1 } )
```

El valor `1` indica que el índice se ordenará de forma ascendente. Si quisiéramos que se ordenara de forma descendente usaríamos el valor `-1`.

Si deseamos eliminar un índice lo haremos de la siguiente manera:

```
db.alum.dropIndex( { nombre: 1 } )
```

o bien usando el *nombre* del índice:

```
db.alum.dropIndex('nombre_1')
```

Por ejemplo, en nuestra base de datos de ejemplo `airbnb_bar` podía ser interesante crear un índice sobre el campo `listing_id` de la colección `det_reviews`:

```
db.airbnb_bar.det_reviews.createIndex( { price: 1 } )
```

## Índices compuestos

Son índices que se definen sobre varios campos de una colección. Se definen de la siguiente manera:

```
db.alum.createIndex( { nombre: 1, edad: -1 } )
```

En este caso estamos indicando a mongo que cree un índice sobre dos campos: `nombre` y `edad`. El índice se ordenará de forma ascendente en el campo `nombre` y de forma descendente en el campo `edad`.

## Índices únicos

Los índices únicos son aquellos que no permiten que se repitan los valores de los campos sobre los que se definen. Por ejemplo, si queremos que el campo `DNI` de la colección `alum` no contenga valores repetidos crearemos un índice único sobre este campo:

```
db.alum.createIndex( { DNI: 1 }, { unique: true } )
```

Como podemos ver se incluye un segundo parámetro en el que se indica que el índice es único. Este documento contendrá las opciones que deseemos aplicar al índice y es opcional. En las opciones podemos indicar, por ejemplo, que el índice sea único, el nombre del índice, etc.

Si intentamos crear un índice único sobre un campo que ya contiene valores repetidos obtendremos un error y el índice no se creará. Lo mismo sucederá si, para **más de un documento** de la colección, no existe el campo sobre el que queremos crear el índice o si su valor fuese `null`.

## Índices parciales (*sparse*)

---

Los índices parciales son aquellos se sólo contienen entradas para aquellos documentos de la colección que contienen el campo sobre el que se ha creado el índice (aunque su valor fuese `null`). Por ejemplo, si queremos crear un índice sobre el campo `nombre` de la colección `alum` pero sólo queremos que se cree para aquellos documentos que contienen el campo `nombre` lo haremos de la siguiente manera:

```
db.alum.createIndex( { nombre: 1 }, { sparse: true })
```

Hasta aquí llega el contenido de este documento. En el siguiente veremos cómo podemos obtener información sobre el rendimiento de las consultas que realizamos sobre MongoDB y así podremos comprobar si los índices que hemos creado son efectivos o no.