

09 Indexación: Índices TTL

TTL indexes are special single-field indexes that MongoDB can use to automatically remove documents from a collection after a certain amount of time or at a specific clock time. Data expiration is useful for certain types of information like machine generated event data, logs, and session information that only need to persist in a database for a finite amount of time.

To create a TTL index, use the `db.collection.createIndex()` method with the `expireAfterSeconds` option on a field whose value is either a date or an array that contains date values.

Práctica

```
> db.logs.createIndex({startSesion: 1}, {expireAfterSeconds: 60})
{
  "createdCollectionAutomatically" : true,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.logs.insert({startSesion: new Date(), user: "hvkdfjh"})
WriteResult({ "nInserted" : 1 })
```

Expire Documents at a Specific Clock Time

To expire documents at a specific clock time, begin by creating a TTL index on a field that holds values of BSON date type or an array of BSON date-typed objects and specify an `expireAfterSeconds` value of 0. For each document in the collection, set the indexed date field to a value corresponding to the time the document should expire. If the indexed date field contains a date in the past, MongoDB considers the document expired.

```
> db.logs.dropIndexes()
{
  "nIndexesWas" : 2,
  "msg" : "non-_id indexes dropped for collection",
  "ok" : 1
}
> db.logs.createIndex({caducidad: 1}, {expireAfterSeconds: 0})
{
  "createdCollectionAutomatically" : false,
```

```
    "numIndexesBefore" : 1,  
    "numIndexesAfter" : 2,  
    "ok" : 1  
}  
> db.logs.insert({user: "hkjhkfd", logIn: new Date(), caducidad: new Date(2020, 0,  
23, 18, 55)})  
WriteResult({ "nInserted" : 1 })
```