

05 Indexación: Índices de texto

MongoDB provides text indexes to support text search queries on string content. text indexes can include any field whose value is a string or an array of string elements.

Create Text Index

A collection can have at most one text index.

Sintaxis

```
db.collection.createIndex( { <field1>: "text" } )
```

Práctica

```
> db.titulos.createIndex({titulo: "text"})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

```
> db.titulos.getIndexes()
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "biblioteca.titulos"
  },
  {
    "v" : 2,
    "key" : {
      "_fts" : "text",
      "_ftsx" : 1
    },
    "name" : "titulo_text",
    "ns" : "biblioteca.titulos",
  }
]
```

```

        "weights": {
            "titulo": 1
        },
        "default_language": "english",
        "language_override": "language",
        "textIndexVersion": 3
    }
]

```

No se pueden crear varios índices de texto en distintos campos.

```

> db.titulos.createIndex({autor: "text"})
{
  "ok": 0,
  "errmsg": "Index: { v: 2, key: { _fts: \"text\", _ftsx: 1 }, name: \"autor_text\", ns: \"biblioteca.titulos\", weights: { autor: 1 }, default_language: \"english\", language_override: \"language\", textIndexVersion: 3 } already exists with different options: { v: 2, key: { _fts: \"text\", _ftsx: 1 }, name: \"titulo_text\", ns: \"biblioteca.titulos\", weights: { titulo: 1 }, default_language: \"english\", language_override: \"language\", textIndexVersion: 3 }",
  "code": 85,
  "codeName": "IndexOptionsConflict"
}

```

Pero si un índice compuesto de texto con varios campos

```

> db.titulos.dropIndex("titulo_text")
{ "nIndexesWas": 2, "ok": 1 }

> db.titulos.createIndex({autor: "text", titulo: "text"})
{
  "createdCollectionAutomatically": false,
  "numIndexesBefore": 1,
  "numIndexesAfter": 2,
  "ok": 1
}

```

El índice no limita la entrada a tipo string:

```

> db.titulos.insert({autor: false})
WriteResult({ "nInserted": 1 })

```

Wildcard Text Indexes

```
db.collection.createIndex( { "$**": "text" } )
```

Práctica

```
> db.titulos.createIndex( { "$**": "text" } )
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

\$text

\$text performs a text search on the content of the fields indexed with a text index.

Sintaxis

```
{
  $text:
  {
    $search: <string>,
    $language: <string>, // Opcional, por defecto el lenguaje del índice
    $caseSensitive: <boolean>, // Opcional, por defecto false
    $diacriticSensitive: <boolean> // Opcional, por defecto false
  }
}
```

Search for a Single Word

Práctica

```
> db.titulos.find({$text: {$search: "señor"}})

> db.titulos.insert({titulo:"París era una Fiesta", autor: "Ernest Hemingway" })

> db.titulos.find({$text: {$search: "paRis"}})
```

```
{ "_id" : ObjectId("5e298c6f893b8b18709e50b7"), "titulo" : "París era una Fiesta",  
"autor" : "Ernest Hemingway" }  
> db.titulos.find({$text: {$search: "pa"}}) // devuelve vacío
```

Match Any of the Search Terms

Práctica

```
> db.titulos.insert({titulo:"París, La Guía Completa", autor: "vv.aa." })  
  
> db.titulos.find({$text: {$search: "Paris era"}})  
{ "_id" : ObjectId("5e298c6f893b8b18709e50b7"), "titulo" : "París era una Fiesta",  
"autor" : "Ernest Hemingway" }  
{ "_id" : ObjectId("5e298d66893b8b18709e50b8"), "titulo" : "París, La Guía  
Completa", "autor" : "vv.aa." }  
  
> db.titulos.find({$text: {$search: "La Ciudad"}})  
{ "_id" : ObjectId("5e280ddde5cb58e060e229fe"), "autor" : "Mario Vargas Llosa",  
"stock" : 10, "prestados" : 2, "titulo" : "La Ciudad y los Perros", "nacionalidad" : "Peru"  
}  
{ "_id" : ObjectId("5e298d66893b8b18709e50b8"), "titulo" : "París, La Guía  
Completa", "autor" : "vv.aa." }
```

Search for a Phrase

Práctica

```
> db.titulos.find({$text: {$search: "\"la ciudad\""}})  
{ "_id" : ObjectId("5e280ddde5cb58e060e229fe"), "autor" : "Mario Vargas Llosa",  
"stock" : 10, "prestados" : 2, "titulo" : "La Ciudad y los Perros", "nacionalidad" : "Peru"  
}
```

Exclude Documents That Contain a Term

Práctica

```
> db.titulos.find({$text: {$search: "paris -guia"}})
```

```
{ "_id" : ObjectId("5e298d76893b8b18709e50b9"), "titulo" : "París, La", "autor" :  
"vv.aa." }  
{ "_id" : ObjectId("5e298c6f893b8b18709e50b7"), "titulo" : "París era una Fiesta",  
"autor" : "Ernest Hemingway" }
```

Stop Words

```
> db.titulos.insert({titulo:"the", autor: "vv.aa." })  
WriteResult({ "nInserted" : 1 })  
> db.titulos.find({$text: {$search: "the"}}) // devuelve vacío  
  
> db.titulos.find({$text: {$search: "la"}}) // devuelve los textos que tengan "la"  
> db.titulos.find({$text: {$search: "la", $language: "es"}}) // devuelve vacío
```

Stemmed words

Práctica

```
> db.titulos.find({$text: {$search: "consult"}})  
{ "_id" : ObjectId("5e29911b893b8b18709e50ba"), "titulo" : "Agile Consultants",  
"autor" : "vv.aa." }  
{ "_id" : ObjectId("5e299130893b8b18709e50bb"), "titulo" : "Consulting for Global  
Markets", "autor" : "vv.aa." }
```

Perform Case Sensitive Search

Práctica

```
> db.titulos.find({$text: {$search: "la", $caseSensitive: true}})
```

Perform Diacritic Sensitive Search

Práctica

```
> db.titulos.find({$text: {$search: "Economía", $caseSensitive: true}})  
{ "_id" : ObjectId("5e29929a893b8b18709e50be"), "titulo" : "Claves de la Economia  
en el siglo XXI", "autor" : "vv.aa." }
```

```
{ "_id" : ObjectId("5e299190893b8b18709e50bd"), "titulo" : "Claves de la Economía en el siglo XXI", "autor" : "vv.aa." }
> db.titulos.find({$text: {$search: "Economía", $diacriticSensitive: true}})
{ "_id" : ObjectId("5e299190893b8b18709e50bd"), "titulo" : "Claves de la Economía en el siglo XXI", "autor" : "vv.aa." }
```

\$meta

The \$meta projection operator returns for each matching document the metadata (e.g. "textScore") associated with the query.

Sintaxis

```
{ $meta: <metaDataKeyword> }
```

Keyword	Description	Sort Order
"textScore"	Returns the score associated with the corresponding \$text query for each matching document. The text score signifies how well the document matched the search term or terms. If not used in conjunction with a \$text query, returns a score of 0.	Descending

Práctica

```
> db.titulos.insert({titulo: "París", autor: "vv.aa"})
WriteResult({ "nInserted" : 1 })
> db.titulos.insert({titulo: "París siempre será París", autor: "vv.aa"})
WriteResult({ "nInserted" : 1 })
> db.titulos.find({$text: {$search: "Paris"}}, {score: {$meta: "textScore"}})
{ "_id" : ObjectId("5e29a8c9893b8b18709e50c5"), "titulo" : "Viaje a París", "autor" : "vv.aa", "score" : 0.75 }
{ "_id" : ObjectId("5e298c6f893b8b18709e50b7"), "titulo" : "París era una Fiesta", "autor" : "Ernest Hemingway", "score" : 0.625 }
{ "_id" : ObjectId("5e29a8fc893b8b18709e50c6"), "titulo" : "París", "autor" : "vv.aa", "score" : 1 }
```

```

{ "_id" : ObjectId("5e298d66893b8b18709e50b8"), "titulo" : "París, La Guía Completa", "autor" : "vv.aa.", "score" : 0.625 }
{ "_id" : ObjectId("5e29a906893b8b18709e50c7"), "titulo" : "París siempre será París", "autor" : "vv.aa", "score" : 1.125 }

> db.titulos.find({$text: {$search: "Paris"}}, {titulo: 1, _id: 0, score: {$meta: "textScore"}})
{ "titulo" : "Viaje a París", "score" : 0.75 }
{ "titulo" : "París era una Fiesta", "score" : 0.625 }
{ "titulo" : "París", "score" : 1 }
{ "titulo" : "París, La Guía Completa", "score" : 0.625 }
{ "titulo" : "París siempre será París", "score" : 1.125 }

> db.titulos.find({$text: {$search: "Paris"}}, {score: {$meta: "textScore"}}).sort( { score:
{ $meta: "textScore" } } )
{ "_id" : ObjectId("5e29a906893b8b18709e50c7"), "titulo" : "París siempre será París", "autor" : "vv.aa", "score" : 1.125 }
{ "_id" : ObjectId("5e29a8fc893b8b18709e50c6"), "titulo" : "París", "autor" : "vv.aa", "score" : 1 }
{ "_id" : ObjectId("5e29a8c9893b8b18709e50c5"), "titulo" : "Viaje a París", "autor" : "vv.aa", "score" : 0.75 }
{ "_id" : ObjectId("5e298c6f893b8b18709e50b7"), "titulo" : "París era una Fiesta", "autor" : "Ernest Hemingway", "score" : 0.625 }
{ "_id" : ObjectId("5e298d66893b8b18709e50b8"), "titulo" : "París, La Guía Completa", "autor" : "vv.aa.", "score" : 0.625 }
> db.titulos.find({$text: {$search: "Paris"}}, {score: {$meta: "textScore"}}).sort( { score:
{ $meta: "textScore" } } ).limit(2)
{ "_id" : ObjectId("5e29a906893b8b18709e50c7"), "titulo" : "París siempre será París", "autor" : "vv.aa", "score" : 1.125 }
{ "_id" : ObjectId("5e29a8fc893b8b18709e50c6"), "titulo" : "París", "autor" : "vv.aa", "score" : 1 }

```

Specify a Language for Text Index

The default language associated with the indexed data determines the rules to parse word roots (i.e. stemming) and ignore stop words. The default language for the indexed data is english.

To specify a different language, use the `default_language` option when creating the text index. See [Text Search Languages](#) for the languages available for `default_language`.

Práctica

```
> db.titulos.createIndex({titulo: "text"},{ default_language: "spanish" })
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.titulos.find({$text: {$search: "La"}})
```

Create a text Index for a Collection in Multiple Languages

Specify the Index Language within the Document

Práctica

```
> db.titles.insert({language:"spanish", title: "El Quijote", translation: [{language:
"english", titleTr: "The Quijote"}, {language: "portuguese", titleTr: "O Quixote"}]})
WriteResult({ "nInserted" : 1 })
```

Los campos que estén en un documento con el campo language en el lenguaje en el que están escritos si que utilizarán el índice en ese idioma aunque sea distinto del lenguaje del índice.

```
> db.titles.createIndex({title: "text", "translation.titleTr": "text"},{ default_language:
"spanish" })
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}

> db.titles.find({$text: {$search: "el"}})
> db.titles.find({$text: {$search: "the"}})
> db.titles.find({$text: {$search: "o"}})

> db.titles.insert({language:"spanish", title: "El Quijote", translation: [{titleTr: "The
Quijote"}, {titleTr: "O Quixote"}]})
WriteResult({ "nInserted" : 1 })
```



```
> db.titles.find({$text: {$search: "the"}})
{ "_id" : ObjectId("5e29afc2893b8b18709e50ca"), "language" : "spanish", "title" : "El Quijote", "translation" : [ { "titleTr" : "The Quijote" }, { "titleTr" : "O Quixote" } ] }
```

Control Search Results with Weights

Text search assigns a score to each document that contains the search term in the indexed fields. The score determines the relevance of a document to a given search query.

For a text index, the weight of an indexed field denotes the significance of the field relative to the other indexed fields in terms of the text search score.

For each indexed field in the document, MongoDB multiplies the number of matches by the weight and sums the results. Using this sum, MongoDB then calculates the score for the document.

Práctica

```
> db.titulos.insert({titulo: "El coronel no tiene quien le escriba", categorias:
["amor","novela","ficción"]})
WriteResult({ "nInserted" : 1 })
> db.titulos.insert({titulo: "El amor en los tiempos del cólera", categorias:
["drama","novela","ficción"]})
WriteResult({ "nInserted" : 1 })
> db.titulos.createIndex({titulo: "text", categorias: "text"})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}

> db.titulos.find({$text: {$search: "amor"}},{score: {$meta: "textScore"}})
{ "_id" : ObjectId("5e29b2b4893b8b18709e50d0"), "titulo" : "El amor en los tiempos del cólera", "categorias" : [ "drama", "novela", "ficción" ], "score" : 0.5714285714285714 }
{ "_id" : ObjectId("5e29b29c893b8b18709e50cf"), "titulo" : "El coronel no tiene quien le escriba", "categorias" : [ "amor", "novela", "ficción" ], "score" : 1.1 }

> db.titulos.dropIndexes()
{
  "nIndexesWas" : 2,
```

```

    "msg" : "non-_id indexes dropped for collection",
    "ok" : 1
}
> db.titulos.createIndex({titulo: "text", categorias: "text"}, {weights: {titulo: 10,
categorias: 1}})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.titulos.find({$text: {$search: "amor"}},{score: {$meta: "textScore"}})
{ "_id" : ObjectId("5e29b29c893b8b18709e50cf"), "titulo" : "El coronel no tiene quien
le escriba", "categorias" : [ "amor", "novela", "ficción" ], "score" : 1.1 }
{ "_id" : ObjectId("5e29b2b4893b8b18709e50d0"), "titulo" : "El amor en los tiempos
del cólera", "categorias" : [ "drama", "novela", "ficción" ], "score" : 5.7142857142857135
}

```