

# 09 Colación

Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks.

You can specify collation for a collection or a view, an index, or specific operations that support collation.

## Collation Document

A collation document has the following fields:

```
{
  locale: <string>,
  caseLevel: <boolean>,
  caseFirst: <string>,
  strength: <int>,
  numericOrdering: <boolean>,
  alternate: <string>,
  maxVariable: <string>,
  backwards: <boolean>
}
```

Se puede establecer a nivel de colección.

```
db.productosSinColacion.insert({_id: 1, nombre: 'cafe'})
db.productosSinColacion.insert({_id: 2, nombre: 'café'})
db.productosSinColacion.insert({_id: 3, nombre: 'cafE'})
db.productosSinColacion.insert({_id: 4, nombre: 'cafÉ'})
```

```
db.productosSinColacion.find({}).sort({nombre: 1})
```

```
{ "_id" : 3, "nombre" : "cafE" }
{ "_id" : 1, "nombre" : "cafe" }
{ "_id" : 4, "nombre" : "cafÉ" }
{ "_id" : 2, "nombre" : "café" }
```

Ahora colección similar con colación:

```
db.createCollection( "productosConColacion", { collation: { locale: "es" } } )
```

```
db.productosConColacion.insert({_id: 1, nombre: 'cafe'})
db.productosConColacion.insert({_id: 2, nombre: 'café'})
db.productosConColacion.insert({_id: 3, nombre: 'cafE'})
```

```
db.productosConColacion.insert({_id: 4, nombre: 'café'})
```

```
db.productosConColacion.find({}).sort({nombre: 1})
```

```
{ "_id" : 1, "nombre" : "cafe" }  
{ "_id" : 3, "nombre" : "café" }  
{ "_id" : 2, "nombre" : "café" }  
{ "_id" : 4, "nombre" : "café" }
```

Sobre una colección sin colación se puede forzar el uso de colación con el método `collation(<documento-colacion>)`.

```
db.productosSinColacion.find({}).collation( { locale: "es" } )
```

```
{ "_id" : 1, "nombre" : "cafe" }  
{ "_id" : 2, "nombre" : "café" }  
{ "_id" : 3, "nombre" : "café" }  
{ "_id" : 4, "nombre" : "café" }
```

Opciones strength

Por defecto el valor es 3, es decir distingue mayúsculas/minúsculas y diacríticos

```
db.productosConColacion.find({nombre: "café"})  
{ "_id" : 2, "nombre" : "café" }
```

Podemos comprobar los valores actuales con:

```
db.getCollectionInfos({name:'productosConColacion'})
```

...

```
  "collation" : {  
    "locale" : "es",  
    "caseLevel" : false,  
    "caseFirst" : "off",  
    "strength" : 3,  
    "numericOrdering" : false,  
    "alternate" : "non-ignorable",  
    "maxVariable" : "punct",  
    "normalization" : false,  
    "backwards" : false,  
    "version" : "57.1"  
  }
```

...

Como la colección no se puede modificar vamos a modificar las opciones en la colección sin colación:

```
db.productosSinColacion.find({nombre: "café"}).collation({locale: "es"})
{ "_id" : 2, "nombre" : "café" }
```

Y ahora con nivel 2 que no distingue mayúsculas de minúsculas:

```
db.productosSinColacion.find({nombre: "café"}).collation({locale: "es", strength: 2})
{ "_id" : 2, "nombre" : "café" }
{ "_id" : 4, "nombre" : "café" }
```

Y con nivel 1 que no distingue ni mayúsculas, ni minúsculas ni diacríticos:

```
db.productosSinColacion.find({nombre: "café"}).collation({locale: "es", strength: 1})
{ "_id" : 1, "nombre" : "cafe" }
{ "_id" : 2, "nombre" : "café" }
{ "_id" : 3, "nombre" : "café" }
{ "_id" : 4, "nombre" : "café" }
```

Otra opción es caseLevel para determinar el orden con strength 1 para que no distinga diacríticos pero si mayúsculas minúsculas:

```
db.productosSinColacion.find({nombre: "café"}).collation({locale: "es", strength: 1,
caseLevel: true})
{ "_id" : 1, "nombre" : "cafe" }
{ "_id" : 2, "nombre" : "café" }
```

Otra opción es caseFirst "upper" o "lower" para ordenar

```
db.productosSinColacion.update({_id: 1},{ $set:{nombre: "Cafe"}})
db.productosSinColacion.update({_id: 3},{ $set:{nombre: "CafE"}})
```

```
db.productosSinColacion.find({}).sort({nombre: 1})
{ "_id" : 3, "nombre" : "CafE" }
{ "_id" : 1, "nombre" : "Cafe" }
{ "_id" : 4, "nombre" : "café" }
{ "_id" : 2, "nombre" : "café" }
```

```
db.productosSinColacion.find({}).collation({locale: "es"}).sort({nombre: 1})
{ "_id" : 1, "nombre" : "Cafe" }
{ "_id" : 3, "nombre" : "CafE" }
{ "_id" : 2, "nombre" : "café" }
{ "_id" : 4, "nombre" : "café" }
```

```
db.productosSinColacion.find({}).collation({locale: "es", caseFirst:
"upper"}).sort({nombre: 1})
```

```
{ "_id" : 3, "nombre" : "CafÉ" }
{ "_id" : 1, "nombre" : "Cafe" }
{ "_id" : 4, "nombre" : "café" }
{ "_id" : 2, "nombre" : "café" }
```

Y la última opción importante numericOrder para ordenar numéricamente en cadenas::

```
db.productosSinColacion.insert([
... { _id: 20, codigo: "A1"},
... { _id: 21, codigo: "A2"},
... { _id: 22, codigo: "A11"},
... { _id: 23, codigo: "A3"},
... { _id: 24, codigo: "A21"}
... ])
```

```
db.productosSinColacion.find({ _id: {$gte: 20}}).collation({locale: "es"}).sort({codigo: 1})
```

```
{ "_id" : 20, "codigo" : "A1" }
{ "_id" : 22, "codigo" : "A11" }
{ "_id" : 21, "codigo" : "A2" }
{ "_id" : 24, "codigo" : "A21" }
{ "_id" : 23, "codigo" : "A32" }
```

```
db.productosSinColacion.find({ _id: {$gte: 20}}).collation({locale: "es", numericOrdering:
true}).sort({codigo: 1})
```

```
{ "_id" : 20, "codigo" : "A1" }
{ "_id" : 21, "codigo" : "A2" }
{ "_id" : 22, "codigo" : "A11" }
{ "_id" : 24, "codigo" : "A21" }
{ "_id" : 23, "codigo" : "A32" }
```

