

Del mismo modo que hicimos con los índices simples, restauramos la base de datos inventory para practicar con ella el tema de los índices compuestos ([https://github.com/pabSec/mongoDB\\_seeder/tree/master/inventory](https://github.com/pabSec/mongoDB_seeder/tree/master/inventory)).

Para practicar algunas queries complejas buscaremos elementos cuyo stock sea menor a 2 y proyectaremos la información relevante de los mismos:

```
> db.inventories.find({"stock.cantidad":{$lt:2}},{_id:0,item:1,stock:{$elemMatch:{cantidad:
{$lt:2}}}})
{ "item" : "Incredible Fresh Chips", "stock" : [ { "cantidad" : 1, "color" : "Rosa", "size" :
"xl" } ] }
{ "item" : "Rustic Concrete Bacon", "stock" : [ { "cantidad" : 1, "color" : "Azul", "size" : "x
l" } ] }
{ "item" : "Tasty Rubber Gloves", "stock" : [ { "cantidad" : 1, "color" : "Negro", "size" : "m"
} ] }
{ "item" : "Licensed Granite Cheese", "stock" : [ { "cantidad" : 1, "color" : "Blanco", "size"
: "xl" } ] }
{ "item" : "Rustic Frozen Cheese", "stock" : [ { "cantidad" : 1, "color" : "Azul", "size" : "x
l" } ] }
...
```

A continuación haremos una búsqueda para aquellos elementos de color rosa (\$elemMatch) de los que resten menos de 2 unidades en stock y comprobaremos la eficiencia de la query:

```
> db.inventories.find({stock:{$elemMatch:{cantidada":{$lt:2},"color":"Rosa"}}})
{ "_id" : ObjectId("58a366bd0831f80104df5abe"), "item" : "Incredible Fresh Chips", "stock" : [
{ "cantidad" : 1, "color" : "Rosa", "size" : "xl" }, { "cantidad" : 22, "color" : "Morado", "s
ize" : "l" }, { "cantidad" : 30, "color" : "Amarillo", "size" : "xl" }, { "cantidad" : 29, "col
or" : "Blanco", "size" : "m" }, { "cantidad" : 22, "color" : "Azul", "size" : "l" } ], "__v" :
0 }
{ "_id" : ObjectId("58a366bd0831f80104df5b3d"), "item" : "Fantastic Metal Shirt", "stock" : [ {
"cantidad" : 66, "color" : "Morado", "size" : "m" }, { "cantidad" : 1, "color" : "Rosa", "siz
e" : "l" }, { "cantidad" : 72, "color" : "Verde", "size" : "m" }, { "cantidad" : 76, "color" :
"Morado", "size" : "m" }, { "cantidad" : 10, "color" : "Rojo", "size" : "l" }, { "cantidad" : 5
3, "color" : "Blanco", "size" : "s" } ], "__v" : 0 }
...
```

La query tienen una duración de 63 milisegundos:

```
> db.inventories.find({stock:{$elemMatch:{cantidada":{$lt:2},"color":"Rosa"}}}).explain("execut
ionStats")
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "inventory.inventories",
    "indexFilterSet" : false,
    "parsedQuery" : {
```

```

        "stock" : {
            "$elemMatch" : {
                "$and" : [
                    {
                        "color" : {
                            "$eq" : "Rosa"
                        }
                    },
                    {
                        "cantidad" : {
                            "$lt" : 2
                        }
                    }
                ]
            }
        }
    },
    "winningPlan" : {
        "stage" : "COLLSCAN",
        "filter" : {
            "stock" : {
                "$elemMatch" : {
                    "$and" : [
                        {
                            "color" : {
                                "$eq" : "Rosa"
                            }
                        },
                        {
                            "cantidad" : {
                                "$lt" : 2
                            }
                        }
                    ]
                }
            }
        },
        "direction" : "forward"
    },
    "rejectedPlans" : [ ]
},
"executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 145,
    "executionTimeMillis" : 63,
    "totalKeysExamined" : 0,
    "totalDocsExamined" : 32000,
    "executionStages" : {
        "stage" : "COLLSCAN",
        "filter" : {
            "stock" : {
                "$elemMatch" : {
                    "$and" : [
                        {
                            "color" : {
                                "$eq" : "Rosa"
                            }
                        }
                    ]
                }
            }
        }
    },

```

```

    {
      "cantidad" : {
        "$lt" : 2
      }
    }
  ]
}
},
"serverInfo" : {
  "host" : "7ea9850856d3",
  "port" : 27017,
  "version" : "3.4.2",
  "gitVersion" : "3f76e40c105fc223b3e5aac3e20dcd026b83b38b"
},
"ok" : 1
}

```

Vamos a definir un índice compuesto para intentar optimizar la query y hacer nuestra aplicación de detección de stocks bajos más eficiente:

```

> db.inventories.createIndex({"stock.cantidad":1,"stock.color":1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.inventories.find({stock:{$elemMatch:{"cantidad":{$lt:2},"color":"Rosa"}}}).explain("executionStats")
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "inventory.inventories",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "stock" : {
        "$elemMatch" : {
          "$and" : [
            {
              "cantidad" : {
                "$lt" : 2
              }
            },
            {
              "color" : {
                "$eq" : "Rosa"
              }
            }
          ]
        }
      }
    }
  }
}

```





```

        "stock.cantidad" : 1,
        "stock.color" : 1
    },
    "indexName" : "stock.cantidad_1_stock.color_1",
    "isMultiKey" : true,
    "multiKeyPaths" : {
        "stock.cantidad" : [
            "stock"
        ],
        "stock.color" : [
            "stock"
        ]
    },
    "isUnique" : false,
    "isSparse" : false,
    "isPartial" : false,
    "indexVersion" : 2,
    "direction" : "forward",
    "indexBounds" : {
        "stock.cantidad" : [
            "[-inf.0, 2.0]"
        ],
        "stock.color" : [
            "[\"Rosa\", \"Rosa\"]"
        ]
    },
    "keysExamined" : 148,
    "seeks" : 3,
    "dupsTested" : 145,
    "dupsDropped" : 0,
    "seenInvalidated" : 0
    }
}
},
"serverInfo" : {
    "host" : "7ea9850856d3",
    "port" : 27017,
    "version" : "3.4.2",
    "gitVersion" : "3f76e40c105fc223b3e5aac3e20dcd026b83b38b"
},
"ok" : 1
}
> db.inventories.createIndex({"stock.color":1,"stock.cantidad":1})
{
    "createdCollectionAutomatically" : false,
    "numIndexesBefore" : 2,
    "numIndexesAfter" : 3,
    "ok" : 1
}
> db.inventories.find({stock:{$elemMatch:{"cantidad":{$lt:2},"color":"Rosa"}}}).hint({"stock.co
lor":1,"stock.cantidad":1}).explain("executionStats")
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "inventory.inventories",
        "indexFilterSet" : false,
        "parsedQuery" : {
            "stock" : {
                "$elemMatch" : {

```

```

        "$elemMatch" : {
          "$and" : [
            {
              "color" : {
                "$eq" : "Rosa"
              }
            },
            {
              "cantidad" : {
                "$lt" : 2
              }
            }
          ]
        }
      }
    },
    "winningPlan" : {
      "stage" : "FETCH",
      "filter" : {
        "stock" : {
          "$elemMatch" : {
            "$and" : [
              {
                "color" : {
                  "$eq" : "Rosa"
                }
              },
              {
                "cantidad" : {
                  "$lt" : 2
                }
              }
            ]
          }
        }
      },
      "inputStage" : {
        "stage" : "IXSCAN",
        "keyPattern" : {
          "stock.color" : 1,
          "stock.cantidad" : 1
        },
        "indexName" : "stock.color_1_stock.cantidad_1",
        "isMultiKey" : true,
        "multiKeyPaths" : {
          "stock.color" : [
            "stock"
          ],
          "stock.cantidad" : [
            "stock"
          ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
          "stock.color" : [

```





```

        "restoreState" : 1,
        "isEOF" : 1,
        "invalidates" : 0,
        "keyPattern" : {
            "stock.color" : 1,
            "stock.cantidad" : 1
        },
        "indexName" : "stock.color_1_stock.cantidad_1",
        "isMultiKey" : true,
        "multiKeyPaths" : {
            "stock.color" : [
                "stock"
            ],
            "stock.cantidad" : [
                "stock"
            ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
            "stock.color" : [
                ["\"Rosa\"", "\"Rosa\""]
            ],
            "stock.cantidad" : [
                "[-inf.0, 2.0]"
            ]
        },
        "keysExamined" : 145,
        "seeks" : 1,
        "dupsTested" : 145,
        "dupsDropped" : 0,
        "seenInvalidated" : 0
    }
},
"serverInfo" : {
    "host" : "7ea9850856d3",
    "port" : 27017,
    "version" : "3.4.2",
    "gitVersion" : "3f76e40c105fc223b3e5aac3e20dcd026b83b38b"
},
"ok" : 1
}

```

Si comprobamos los resultados para ambas combinaciones de índices compuestos, comprobamos que ambos reducen igual las queries pero que en el último caso se examinan menos keys y por tanto es más eficiente.