

# EJERCICIO

Se precisa diseñar un blog de noticias donde los usuarios registrados puedan publicar sus comentarios:

- Cada autor tiene un nombre, un nombre de usuario, una cuenta de Twitter y una descripción. Además, de forma opcional, los usuarios pueden proporcionar como datos su dirección postal (calle, número, puerta, C.P., ciudad) o sus teléfonos de contacto (pueden tener varios).
- Las noticias tienen un título, un cuerpo y una fecha de publicación. Son publicadas por un autor y pueden contener o no, una lista de tags.
- Las noticias reciben comentarios, quedando registrado la persona que lo escribió, el comentario escrito y el momento en el que lo hizo.

Realizar el diseño que se crea más conveniente y explicar por qué se ha elegido embebido o referenciado en cada relación.

*El diseño planteado para esta base de datos consiste en tres grandes colecciones: usuarios, noticias y comentarios. A continuación, se explica por apartado cómo están organizados los datos, puesto que se estima que un blog puede llegar a tener una gran cantidad de noticias con múltiples comentarios, y se estima, por tanto, un gran volumen de usuarios autores de comentarios y de noticias.*

*En primer lugar, los usuarios van a contar con un id asignado, que será el nombre de la cuenta de Twitter, para poder referenciarlos por el mismo y que sea mucho más rápida su identificación tanto en las noticias como en los comentarios, sin necesidad de mostrar toda de información personal de los mismos. Los teléfonos serán almacenados en un array, y la dirección será un subdocumento embebido.*

*En segundo lugar, las noticias tendrán un array de comentarios, puesto que siempre que consultemos las noticias necesitamos tener acceso a ellos aunque tengan muchas actualizaciones, y contarán con un autor que será el id de los usuarios, de esta manera controlamos que al crear una noticia se le asigna o no un autor y se identifica solo por su nombre de Twitter, puesto que es la “única” información que resulta relevante. Se ha planteado esta referencia así porque se entiende que un usuario puede escribir gran cantidad de noticias, y al mostrar en la noticia el nombre del autor directamente, hace más ágil la lectura y la consulta.*

*Y, en tercer lugar, los comentarios van a tener la referencia del autor con el id asignado y la referencia a la noticia sobre la que se ha escrito, puesto que entendemos que un comentario solo lo puede escribir un autor y solo pueden hacer referencia a una noticia. De esta manera, el crecimiento de cada documento es bastante medido, y tenemos los datos fundamentales a la hora de hacer consultas sobre comentarios.*

*Podemos observar entonces que la colección cuyos documentos van a sufrir más actualizaciones y crecimientos va a ser la de noticias, donde se añadirán continuamente los comentarios.*

Crear los siguientes índices:


```
blog> db.stats()
{
  db: 'blog',
  collections: 3,
  views: 0,
  objects: 23,
  avgObjSize: 224.95652173913044,
  dataSize: 5174,
  storageSize: 110592,
  indexes: 3,
  indexSize: 110592,
  totalSize: 221184,
  scaleFactor: 1,
  fsUsedSize: 104674930688,
  fsTotalSize: 250998095872,
  ok: 1
}
blog>
```

Consulta antes del índice

```

blog> db.usuarios.find().explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'blog.usuarios',
    indexFilterSet: false,
    parsedQuery: {},
    queryHash: '17830885',
    planCacheKey: '17830885',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: { stage: 'COLLSCAN', direction: 'forward' },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 4,
    executionTimeMillis: 21,
    totalKeysExamined: 0,
    totalDocsExamined: 4,
    executionStages: {

```




- Creación de los siguientes índices sobre usuario:  
Índice único para los campos nombre\_usuario y cuenta\_twitter.

```


blog> db.usuarios.createIndex({"cuenta_twitter": 1}, {unique: true})
cuenta_twitter_1
blog>

```

En el campo de nombre\_usuario no se puede crear un índice único. Resultado de la consulta con el índice:

```
blog> db.usuarios.find().explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'blog.usuarios',
    indexFilterSet: false,
    parsedQuery: {},
    queryHash: '17830885',
    planCacheKey: '17830885',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: { stage: 'COLLSCAN', direction: 'forward' },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 4,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 4,
    executionStages: {
      
```

- Índice para el campo cp (código postal), hay que tener en cuenta que el campo no es requerido.

```
blog> db.usuarios.find({}, {"direccion.cp":1}).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'blog.usuarios',
    indexFilterSet: false,
    parsedQuery: {},
    queryHash: '5C7E61FB',
    planCacheKey: '5C7E61FB',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'PROJECTION_DEFAULT',
      transformBy: { 'direccion.cp': 1 },
      inputStage: { stage: 'COLLSCAN', direction: 'forward' }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 4,
    executionTimeMillis: 10,
    totalKeysExamined: 0,
    totalDocsExamined: 4,
    executionStages: {
      
```

Tras añadir el índice (db.usuarios.createIndex({"cp":1},{sparse: true})):

```

blog> db.usuarios.find({}, {"direccion.cp":1}).explain("executionStats")
{p_1
  explainVersion: '1',
  queryPlanner: {
    namespace: 'blog.usuarios',
    indexFilterSet: false,
    parsedQuery: {},
    queryHash: '5C7E61FB',
    planCacheKey: '5C7E61FB',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'PROJECTION_DEFAULT',
      transformBy: { 'direccion.cp': 1 },
      inputStage: { stage: 'COLLSCAN', direction: 'forward' }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 4,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 4,
    executionStages: {
      stage: 'PROJECTION_DEFAULT',

```

Con el sparse se indica que el campo puede no estar.

- Varios índices para los teléfonos (tantos como campos haya).

```

blog> db.usuarios.createIndex({"telefono": 1})
telefono_1
blog>

```

- Creación de índice sobre noticia para facilitar las consultas requeridas (más frecuentes según enunciado).

Índice por antigüedad, de más reciente a más antigua, e índice por autor.

```


blog> db.noticias.createIndex({"fecha": -1})
fecha_-1
blog> db.noticias.createIndex({"autor": 1})
autor_1
blog>

```

```

blog> db.noticias.find().explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'blog.noticias',
    indexFilterSet: false,
    parsedQuery: {},
    queryHash: '17830885',
    planCacheKey: '17830885',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: { stage: 'COLLSCAN', direction: 'forward' },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 11,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 11,
    executionStages: {

```



Insertar los datos:

### EJEMPLO DE DATO CON TAGS, AUTOR Y COMENTARIOS

```

{
  _id: ObjectId("636f8fe1126ac1653cb947ff"),
  titulo: 'Fin de mundo',
  cuerpo: 'Los científicos anuncian el fin de la humanidad y la destrucción de la Tierra',
  fecha: 'new Date(2022, 11, 27)',
  autor: 'Frank_USA',
  tags: [ 'ciencia', 'Tierra' ],
  comentarios: [
    '636f939a126ac1653cb9480d',
    '636f93a3126ac1653cb9480e',
    '636f93a9126ac1653cb9480f'
  ]
}

```

Insertar varias noticias (al menos, 10 documentos) con diferentes valores en los campos y que, al menos, 3 documentos tengan varios valores en el array de tags, y otros que no tengan valor en este campo.

Realizar las siguientes consultas frecuentes sobre la BD

- Consulta de los datos del usuario por nombre de usuario y por cuenta de Twitter, ordenado por código postal.

```
blog> db.usuarios.find({}, {"nombre_usuario":1, "cuenta_twitter":1}).sort("direccion.cp")
[
  { _id: 'Pete', nombre_usuario: 'Peter_blog', cuenta_twitter: 'Pete' },
  {
    _id: 'Frank_USE',
    nombre_usuario: 'Frank_blog',
    cuenta_twitter: 'Frank_USE'
  },
  {
    _id: 'Frank_USA',
    nombre_usuario: 'Frank_blog',
    cuenta_twitter: 'Frank_USA'
  },
  {
    _id: 'dandreavh',
    nombre_usuario: 'Andrea_blog',
    cuenta_twitter: 'dandreavh'
  }
]
```

- Consulta las noticias ordenadas por fecha (de más reciente a más antigua), muestra solo las 5 últimas noticias publicadas.



```

blog> db.noticias.find().sort({"fecha":-1}).limit(5)
[
  {
    _id: ObjectId("636f8fe1126ac1653cb947ff"),
    titulo: 'Fin de mundo',
    cuerpo: 'Los científicos anuncian el fin de la humanidad y la destrucción de la Tierra',
    fecha: ISODate("2022-12-27T00:00:00.000Z"),
    autor: 'Frank_USA',
    tags: [ 'ciencia', 'Tierra' ],
    comentarios: [
      '636f939a126ac1653cb9480d',
      '636f93a3126ac1653cb9480e',
      '636f93a9126ac1653cb9480f'
    ]
  },
  {
    _id: ObjectId("636f8fec126ac1653cb94800"),
    titulo: 'Twitter en el punto de mira',
    cuerpo: 'Los empleados de Twitter temen despidos ante el nuevo cambio de dirección',
    fecha: ISODate("2022-11-27T00:00:00.000Z"),
    tags: [ 'empleo', 'tecnología', 'redes sociales' ],
    comentarios: [ '636f93b2126ac1653cb94810' ]
  },
  {
    _id: ObjectId("636f8ff7126ac1653cb94801"),
    titulo: 'Orestes gana Pasapalabra',
    cuerpo: 'El concursante ha ganado un bote extraordinario tras más de 200 programas',
    fecha: ISODate("2022-10-27T00:00:00.000Z"),
    autor: 'dandreavh',
    tags: [ 'concurso', 'sociedad', 'entretenimiento', 'television' ]
  },
  {
    _id: ObjectId("636f9000126ac1653cb94802"),
    titulo: 'Telegram imita a Whatsapp',
    cuerpo: 'Ahora Telegram imita la nueva función de Whatsapp para formar grupos',
    fecha: ISODate("2022-09-27T00:00:00.000Z"),
    autor: 'dandreavh',
    tags: [ 'tecnología', 'redes sociales' ],
    comentarios: [ '636f93b9126ac1653cb94811' ]
  }
]

```

- Noticias que no tienen el campo tag.

```

blog> db.noticias.find({"tags":{"$exists:false"}})
[
  {
    _id: ObjectId("636f901e126ac1653cb94805"),
    titulo: 'Elecciones en la RAE',
    cuerpo: 'Se celebrará para el próximo 2023 las elecciones de presidencia y los demás representantes',
    fecha: ISODate("2022-06-27T00:00:00.000Z")
  }
]
blog>

```

- Noticias publicadas en un periodo de fechas.

```

blog> db.noticias.find({$and: [{"fecha": {$gte: new Date('2022-01-01T00:00:00.000Z')}} ], {"fecha": {$lte: new Date('2022-05-10T13:35:03.835Z')}} ])
[
  {
    _id: ObjectId("636f9033126ac1653cb94807"),
    titulo: 'Nuevo tallaje en Zara',
    cuerpo: 'La empresa hace un primer acercamiento a la situación actual de la aceptación de diversidad de cuerpos y amplía el tallaje de sus productos',
    fecha: ISODate("2022-04-27T00:00:00.000Z"),
    autor: 'dandreavh',
    tags: [ 'moda' ],
    comentarios: [ '636f93c1126ac1653cb94812', [ '636f93cc126ac1653cb94813' ] ]
  },
  {
    _id: ObjectId("636f903c126ac1653cb94808"),
    titulo: 'El Ibex tiembla',
    cuerpo: 'Grandes cambios en la bolsa en los últimos días',
    fecha: ISODate("2022-03-27T00:00:00.000Z"),
    autor: 'Frank_USA',
    tags: [ 'economia' ]
  },
  {
    _id: ObjectId("636f9046126ac1653cb94809"),
    titulo: 'Sequia extrema',
    cuerpo: 'Andalucía se encuentra en una situación extrema de sequia ante la falta de lluvias, las altas temperaturas y el consumo excesivo',
    fecha: ISODate("2022-02-27T00:00:00.000Z"),
    tags: [ 'clima', 'Tierra' ]
  }
]
blog>

```

¿Se podrían realizar consultas por año, mes y día sobre el campo de tipo ISOdate?. Si se puede, haz la consulta.

*Sí, se podrían filtrar los años, meses y días, pero para ello necesitamos las agregaciones. Tal y como se indica en la documentación oficial de MongoDB:*  
<https://www.mongodb.com/docs/manual/reference/operator/aggregation/year/>

Ampliar la base de datos para almacenar los datos de los comentarios que los usuarios pueden dejar sobre las noticias, según la especificación indicada anteriormente.  
 Inserta 6 comentarios: un par de comentarios de distintos usuarios sobre tres noticias distintas.

```
blog> db.comentarios.find()
[
  {
    _id: ObjectId("636f939a126ac1653cb9480d"),
    fecha: ISODate("2022-12-27T00:00:00.000Z"),
    cuerpo: 'Fantasía pura',
    autor: 'Frank_USA',
    noticia: ObjectId("636f8fe1126ac1653cb947ff")
  },
  {
    _id: ObjectId("636f93a3126ac1653cb9480e"),
    fecha: ISODate("2022-12-27T00:00:00.000Z"),
    cuerpo: 'Forma parte del ciclo',
    autor: 'Pete',
    noticia: ObjectId("636f8fe1126ac1653cb947ff")
  },
  {
    _id: ObjectId("636f93a9126ac1653cb9480f"),
    fecha: ISODate("2022-12-27T00:00:00.000Z"),
    cuerpo: 'El ciclo de la vida',
    autor: 'Frank_USE',
    noticia: ObjectId("636f8fe1126ac1653cb947ff")
  },
  {
    _id: ObjectId("636f93b2126ac1653cb94810"),
    fecha: ISODate("2022-11-27T00:00:00.000Z"),
    cuerpo: 'Siempre bulos',
    autor: 'Frank_USA',
    noticia: ObjectId("636f8fec126ac1653cb94800")
  },
  {
    _id: ObjectId("636f93b9126ac1653cb94811"),
    fecha: ISODate("2022-09-27T00:00:00.000Z"),
    cuerpo: 'Me encanta',
    autor: 'dandreavh',
    noticia: ObjectId("636f9000126ac1653cb94802")
  },
  {
    _id: ObjectId("636f93c1126ac1653cb94812"),
    fecha: ISODate("2022-04-27T00:00:00.000Z"),
    cuerpo: 'Me encanta',
    autor: 'dandreavh',
    noticia: ObjectId("636f9033126ac1653cb94807")
  },
  {
    _id: ObjectId("636f93cc126ac1653cb94813"),
    fecha: ISODate("2022-04-27T00:00:00.000Z"),
    cuerpo: 'Muy necesario',
    autor: 'Pete',
    noticia: ObjectId("636f9033126ac1653cb94807")
  },
  {
    _id: ObjectId("636f93db126ac1653cb94814"),
    fecha: ISODate("2022-08-27T00:00:00.000Z"),
    cuerpo: 'Me preocupa',
    autor: 'Pete',
    noticia: ObjectId("636f9009126ac1653cb94803")
  }
]
blog>
```

¿Definirías algún índice? Consulta los comentarios, muestra del último comentario (el más actual), el título y fecha de publicación de la noticia y el nombre del usuario y cuenta de twitter

*Sí, definiría por fecha y por autor, puesto que pueden ser los filtros más demandados.*

*Los datos de dicha consulta no se pueden extraer así, puesto que se necesitan agregaciones.*

*De momento, solo puedo obtener esto:*

```
blog> db.comentarios.find({}, {"cuerpo":1, "noticia":1, "autor":1}).sort({"fecha":-1}).limit(1)
[
  {
    _id: ObjectId("636f939a126ac1653cb9480d"),
    cuerpo: 'Fantasía pura',
    autor: 'Frank_USA',
    noticia: ObjectId("636f8fe1126ac1653cb947ff")
  }
]
blog>
```

## CON AGREGACIÓN

```
db.comentarios.aggregate([{$lookup:{from: "usuarios", localField: "autor", foreignField: "_id",
as:"datos_autor"}},{$group:{}}])
```