Modelado de datos

- En mongoDB, el esquema de los datos es flexible, correspondiendo, por tanto, al desarrollador decidir como implementarlo según los requisitos que se tengan.
- Se pueden definir 3 tipos de relaciones:
  - One-to-one (uno a uno): en este tipo de relación, uno de los documentos suele embeberse dentro de otro.
  - One-to-many embebido (uno-a-varios): los documentos de la relación se embeben dentro de otro en una estructura de tipo *array*.
  - One-to-many con referencias (uno-a-varios): se utilizan referencias al _id de los documentos relacionados, en vez de embeberlos por completo.

Más sobre relaciones en https://docs.mongodb.com/manual/applications/data-models-relationships/

Modelado de datos: *One-to-one*

- Las relaciones uno-a-uno pueden modelarse embebiendo uno de los documentos dentro de otro.
- Supongamos que almacenamos los datos de los socios de un gimnasio junto con su dirección, que se compone de sus propios campos. La solución pasaría por embeber la dirección como un subdocumento del documento principal:

```
{ _id: <ObjectId3>,
  nombre: "Julio",
  apellido1: "González",
        …
  dirección: {
            calle: "Avenida de la República",
            numero: 678
                …
        }}
```

Más sobre relaciones one-to-one en https://docs.mongodb.com/manual/tutorial/model-embedded-one-to-one-relationships-between-documents/

Modelado de datos: *One-to-many* embebido

- Las relaciones uno-a-varios pueden modelarse embebiendo documentos dentro de un campo *array*.
- Siguiendo con el ejemplo anterior, imaginemos que los socios pueden tener varias direcciones:

```
{ _id: <ObjectId3>,
  nombre: "Julio",
  apellido1: "González",
            …
  direcciones: [{
                  calle: "Avenida de la República",
                  numero: 678
                       …},
                 {calle: "Avenida de la República",
                  numero: 678
                       … }]}
```

Más sobre relaciones one-to-many en https://docs.mongodb.com/manual/tutorial/model-embedded-one-to-many-relationships-between-documents/

Modelado de datos: *One-to-many* con documentos embebidos en arrays

- En ocasiones, puede ser muy pesado almacenar arrays con todos los datos de los subdocumentos.
- Imaginemos que queremos almacenar libros junto con la referencia a los editores*. Siguiendo los ejemplo anteriores, podríamos almacenar los libros en un array como subdocumentos de los documentos de los editores:

```
{ _id: "edicionesSotileza",
  ciudad: "Santander",
          …
  libros: [{
            nombre: "Historia de Cantabria",
            ISBN: "678789789"
                    ...},
            nombre: "El pleito de los nueve valles",
            ISBN: "2671982093"
                    …
```

**Ejemplo extraído, traducido y adaptado de https://docs.mongodb.com/manual/tutorial/model-referenced-one-to-many-relationships-between-documents/*

Modelado de datos: *One-to-many* con documentos embebidos

- El problema de la anterior solución es que el array podría crecer en exceso, con lo que las búsquedas sería muy ineficientes.
- Podríamos pensar como solución el embeber a los editores en los libros:

```
{ _id: "HdC1",
  nombre: "Historia de Cantabria",
  ISBN: "678789789"
            ...
  editor: { nombre: "edicionesSotileza",
            ciudad: "Santander",
            ...}

}
```

**Ejemplo extraído, traducido y adaptado de https://docs.mongodb.com/manual/tutorial/model-referenced-one-to-many-relationships-between-documents/*

Modelado de datos: *One-to-many* con referencias

- El problema de la anterior solución es que se repetirían con frecuencia los datos de los editores, ocupando una gran cantidad de espacio.
- Podríamos pensar como solución el referenciar a los editores en los libros:

```
{ _id: "edicionesSotileza",
  ciudad: "Santander",
          …}
```

```
{ _id: "HdC1",
  nombre: "Historia de Cantabria",
  ISBN: "678789789"
          ...
  editor: "edicionesSotileza"}
```

**Ejemplo extraído, traducido y adaptado de https://docs.mongodb.com/manual/tutorial/model-referenced-one-to-many-relationships-between-documents/*