

# Data Model Examples and Patterns

Documentos embebidos frente a documentos referenciados. Tipos de relaciones.

## **Documentos embebidos o modelo denormalizado.**

Modelo por defecto de MongoDB y al que se debe recurrir siempre que se pueda.

Colección productos

```
{
  producto: "Nike FTV",
  marca: "Nike",
  distribuidores: [
    { nombre: "ServiZapas", contacto: "...", ... },
    { nombre: "Distribuciones Pérez", contacto: "...", ...}
  ],
  precio: ...
  ...
}
```

## **Documentos referenciados de otra colección o modelo normalizado.**

Modelo al que recurrir bajo determinadas circunstancias que impiden el anterior.

Colección productos

```
{
  producto: "Nike FTV",
  marca: "Nike",
  id_distribuidores: [ 1, 2, ...],
  precio: ...
  ...
}
```

Colección distribuidores

```
{
  _id: 1,
  nombre: "ServiZapas",
  contacto: "...",
  ...
}

{
  _id: 2,
  nombre: "Distribuciones Pérez",
  contacto: "...",
  ...
}
```

## Model One-to-One

Con modelo denormalizado

- Con una sola consulta se pueden conseguir los datos.

```
{
  _id: 3,
  nombre: "Jon Doe",
  direccion: {
    calle: "Gran Vía, 40",
    localidad: "Madrid",
    cp: "28001"
  },
  ...
}
```

# Modelo One-to-Many

## One-to-Few

Con modelo denormalizado

- El lado one es el que normalmente recibirá más consultas.
- Los documentos del lado “many” se suelen embeber en el lado one.
- No serán frecuentes las escrituras en el lado many.
- El índice se puede realizar sobre el array de embebidos para mejorar las consultas de éste.

Ejemplo Colección productos

```
{
  producto: "Nike FTV",
  marca: "Nike",
  imagenes: [
    { url: "https://dominio/img/nikeftv_1.jpg", textoAlt: "...", ... },
    { url: "https://dominio/img/nikeftv_2.jpg", textoAlt: "...", ... }
  ],
  precio: ...
  ...
}
```

## One-to-Many

En general con modelo denormalizado pero habría que estudiar las circunstancias. Por ejemplo, el tamaño de los documentos del lado many. Si son muy grandes podrían superar el límite de 16 megas por documento.

## One-to-skillions

Con modelo normalizado.

- El lado many es el que normalmente recibirá más consultas.
- Si se embebiera el lado many podría crecer hasta superar el límite de 16 megas por documentos.

Por ejemplo

Colección productos

```
{
  _id: 5648365846
  producto: "Nike FTV",
  marca: "Nike",
  id_opiniones: [ 2424242425, 6768768686, ..., n],
  precio: ...
  ...
}
```

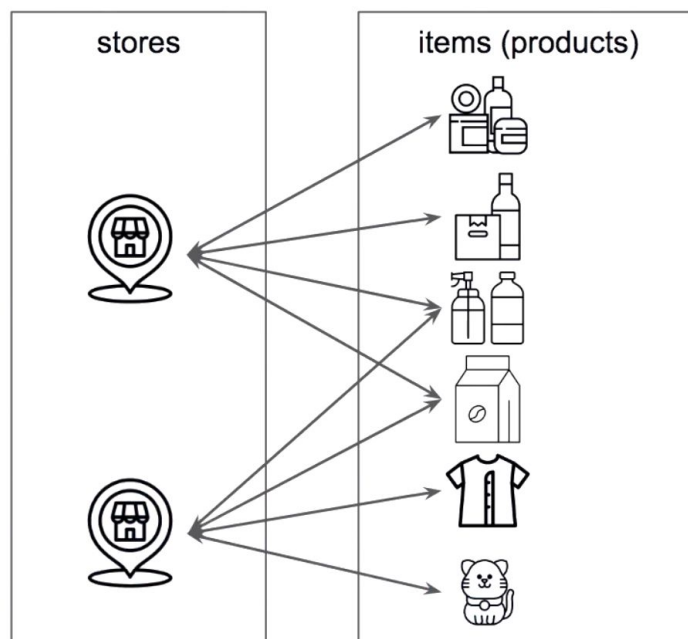
Colección opiniones

```
{
  _id: 2424242425,
  id_producto: 5648365846,
  texto: "Buen producto...",
  estrellas: 3,
  ...
}
```

...

```
{
  _id: 6768768686,
  id_producto: 5648365846,
  texto: "Buen producto...",
  estrellas: 3,
  ...
}
```

## Modelo Many-to-Many



Con modelo denormalizado.

- Mayor número de consultas en el lado de mayor número de registros.
- Redundancia.

Colección productos

```
{
  _id: 5648365846
  producto: "Nike FTV",
  marca: "Nike",
  tiendas: [
    { nombre: "Alcorcón Store", calle: "..."},
    { nombre: "Las Rozas Store", calle: "..."},
  ],
  precio: ...
  ...
}
```

```
{
  _id: 5648365847
  producto: "Adidas Tokyo",
  marca: "Adidas",
  tiendas: [
    { nombre: "Las Rozas Store", calle: "...."}
  ],
  precio: ...
  ...
}
```

Con modelo normalizado

- Necesita dos consultas para recuperar registros.

Colección productos.

```
{
  _id: 5648365846
  producto: "Nike FTV",
  marca: "Nike",
  id_tiendas: [

  ],
  precio: ...
  ...
}
```

```
{
  _id: 5648365847
  producto: "Adidas Tokyo",
  marca: "Adidas",
  tiendas: [
    { nombre: "Las Rozas Store", calle: "...."}
  ],
  precio: ...
  ...
}
```

## Colección tiendas

```
{
  _id: 1,
  nombre: "Alcorcón Store",
  id_productos: [5648365846, ... ]
  calle: "....
}
{
  _id: 2,
  nombre: "Las Rozas Store",
  id_productos: [5648365846, 5648365847, ... ]
  calle: "....
},
```