

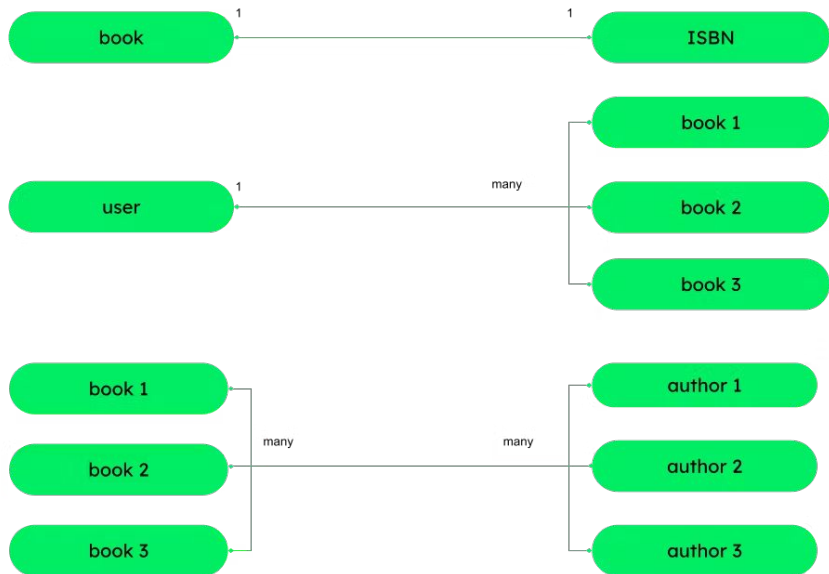
Estrategias de Relación de Colecciones en MongoDB

DEV.FX
DESARROLLAMOS(PERSONAS);

Elaborado por César Guerra

www.cesarquerra.mx

dev



<https://www.mongodb.com/basics/data-modeling>

Tipos de Relaciones

1:1 En esta relación, un campo está asociado con solo un documento.

1:N Aquí, un valor puede estar asociado con más de un documento o valor.

N:M En este tipo de modelo, múltiples documentos pueden ser asociados entre ellos.



Estrategias de Relación de Documentos

En MongoDB, tenemos dos formas de modelar las relaciones entre datos: **relaciones embebidas (incrustadas)** y **relaciones referenciadas**. La elección entre estas dos depende de la cardinalidad de la relación y otros factores.

Embebida: Dentro de una colección registramos un documento que tenga otras subcolecciones.

Referenciada: Se realiza relación con varias colecciones agregando un identificador único.



<https://www.mongodb.com/docs/manual/data-modeling/concepts/embedding-vs-references/>

Embeber

Usar sub-documentos para realizar las relaciones, que está un documento dentro de otro.

Características:

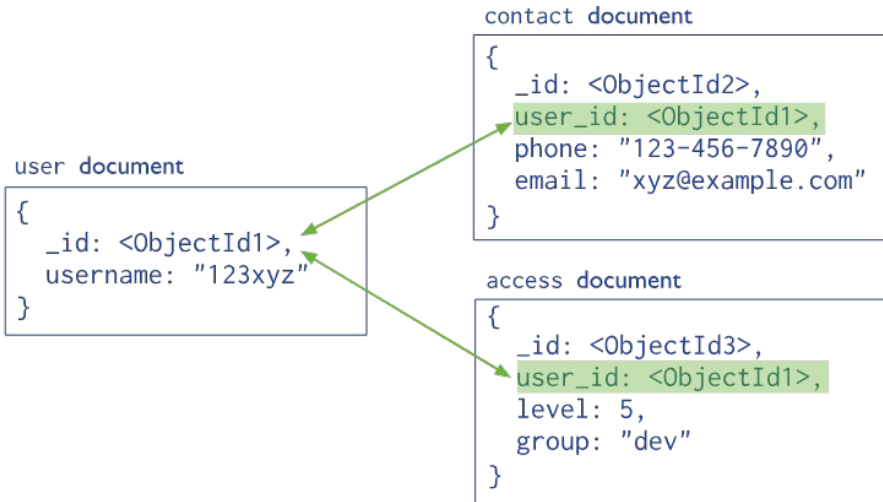
- Mejora rendimiento en búsqueda de muchos datos relacionados.
- Limitado a 1-1 y 1-N en una dirección.
- Cambios contenidos.
- Facilita las consultas y actualizaciones atómicas, ya que los datos están juntos.
- Los datos embebidos no se pueden compartir entre documentos.

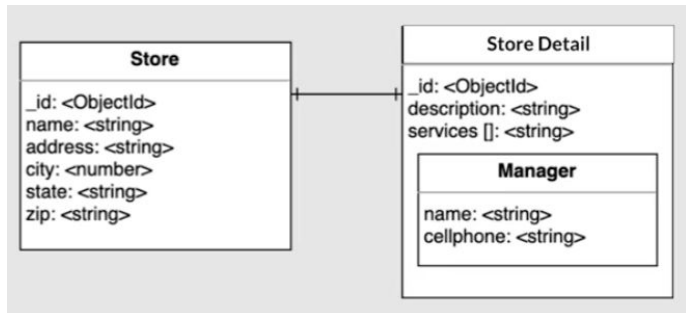
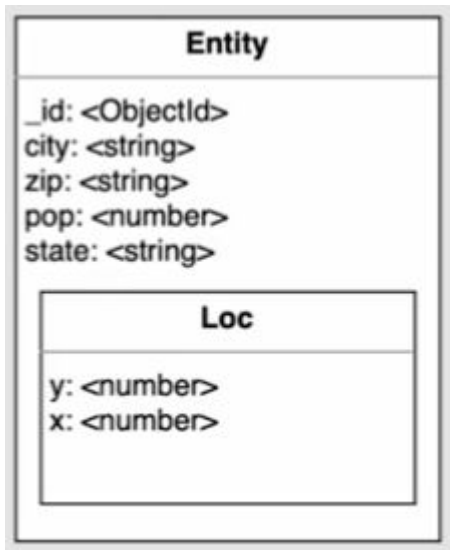
Referenciar

Los datos se almacenan en documentos separados, y se utilizan referencias (normalmente ***_id***) para establecer la relación.

Características:

- Reduce el tamaño de la base de datos.
- No repetir datos.
- Los datos se pueden compartir y reutilizar entre múltiples documentos.
- Permite 1-1, 1-N, N-1 y N-N.
- Requiere consultas adicionales para obtener los datos relacionados (join manual).





Modelado de Datos: Embebidos vs Referenciados

Visualmente, cuando realizamos el modelado de la base de datos, y la relación es embebida, solemos colocar la colección dentro de otra.

A diferencia de cuando relacionamos, que se mantienen de forma independiente.

Relaciones 1:1 (uno a uno)

DEV.F
DESARROLLAMOS(PERSONAS);

```
{
  "_id": "ObjectId('662078a6c15b1a247c71691f')",
  "name": "César Guerra",
  "company": "DevF",
  "email": "cesar@cesarguerra.mx",
  "twitter": "@warderer",
  "instagram": "cesarguerra.mx",
  "website": "www.cesarguerra.mx"
}
```

<https://www.mongodb.com/docs/manual/tutorial/model-embedded-one-to-one-relationships-between-documents/>

Relaciones 1:1 Embebidas

En la gran mayoría de los casos las **relaciones 1-1** en mongodb son embebidas, ya que estas se manejan como si fueran atributos directos de la entidad principal dentro de subdocumentos y por lo general la información se consulta en conjunto (existe una interdependencia directa de los datos).

Es recomendable embeber en una relación 1:1 cuando:

- Cuando la información se consulta en conjunto.
- Los subdocumentos sean una dependencia directa.

Relaciones 1:1 Referenciadas

Es poco común usar relaciones 1:1 referenciadas en MongoDB, sin embargo podrían ser necesaria en los siguientes casos:

Superan el peso máximo de 16MB: Esto podría pasar si se llena con muchas propiedades y/o muy pesadas, como pueden ser binarios, imágenes, videos, etc.

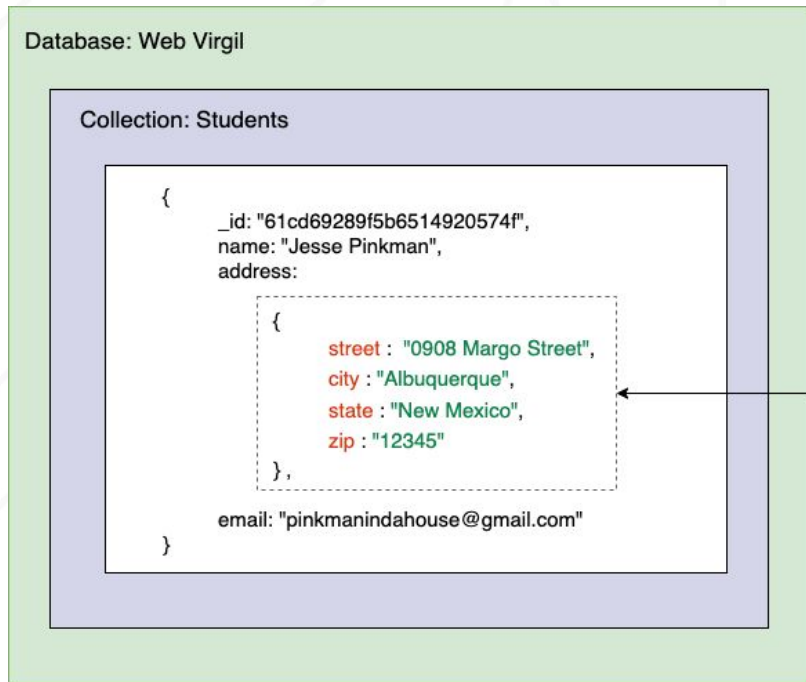
Consultas lentas: Al realizar una petición, puede que se procese demasiada información innecesaria por lo que hace más lenta la consulta. Sobre todo cuando son grandes cantidades de datos.

```
// Colección "persona"
{
  "_id": ObjectId("662078a6c15b1a247c71691f"),
  "name": "César Guerra",
  "company": "DevF",
  "email": "cesar@cesarguerra.mx",
  "socialMedia": ObjectId("123"),
  "website": "https://www.cesarguerra.mx"
}

// Colección "social_media"
{
  "_id": ObjectId("123"),
  "twitter": "@warderer",
  "instagram": "cesarguerra.mx",
}
```

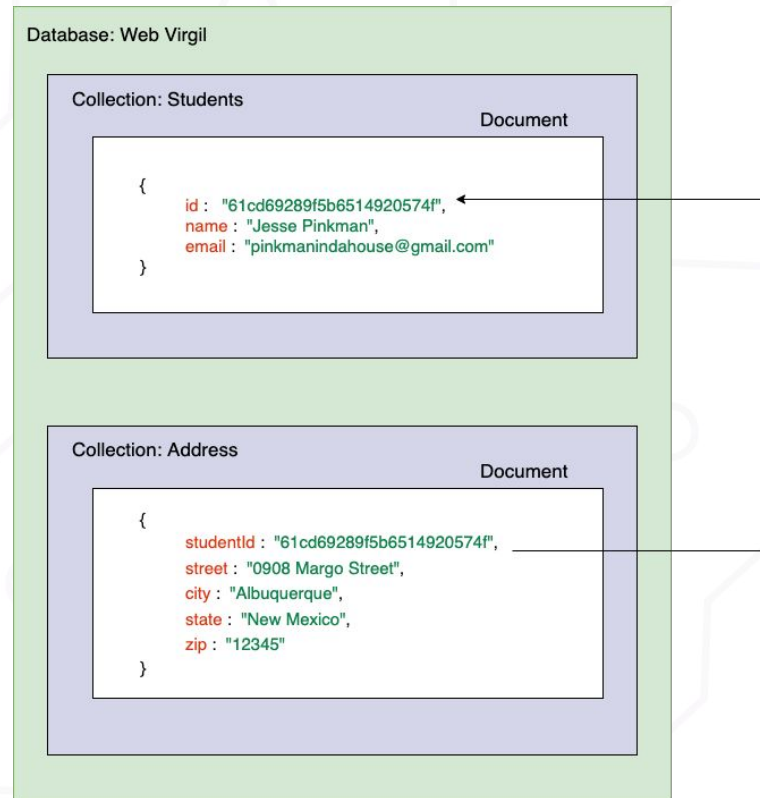
<https://www.mongodb.com/docs/manual/tutorial/model-embedded-one-to-one-relationships-between-documents/>

Relación 1:1 Embebida vs Referenciada



Embebida

Embedded Document



Referenciada

Relaciones 1:N (uno a muchos)

DEV.F
DESARROLLAMOS(PERSONAS);

```
// Colección "autores"
{
  "_id": 1,
  "nombre": "Juan Pérez",
  "libros": [
    { "titulo": "Libro 1", "año": 2010 },
    { "titulo": "Libro 2", "año": 2015 }
  ]
}
```

<https://www.mongodb.com/docs/manual/tutorial/model-embedded-one-to-many-relationships-between-documents/>

Relaciones 1:N Embebidas

Si la cardinalidad del "muchos" es realmente pequeña (por ejemplo, hasta 100 aproximadamente), es común embeber la colección "muchos" en el lado "uno". Esto permite consultas y actualizaciones más eficientes.

¿Cuándo usar 1:N Embebido?

- Cuando la información es consultada en conjunto.
- El arreglo de subdocumentos no alcance el límite de 16MB.
- Cuando los subdocumentos son una dependencia.
- Cuando los sub-docs son 1:few (1 a pocos)

```
// Colección "autores"
{
  "_id": 1,
  "nombre": "Juan Pérez",
  "librosIds": [5, 7, 9]
}

// Colección "libros"
{ "_id": 5, "titulo": "Libro 1", "año": 2010, "autorId": 1 }
{ "_id": 7, "titulo": "Libro 2", "año": 2015, "autorId": 1 }
{ "_id": 9, "titulo": "Libro 3", "año": 2020, "autorId": 1 }
```

<https://www.mongodb.com/docs/manual/tutorial/model-referenced-one-to-many-relationships-between-documents/>

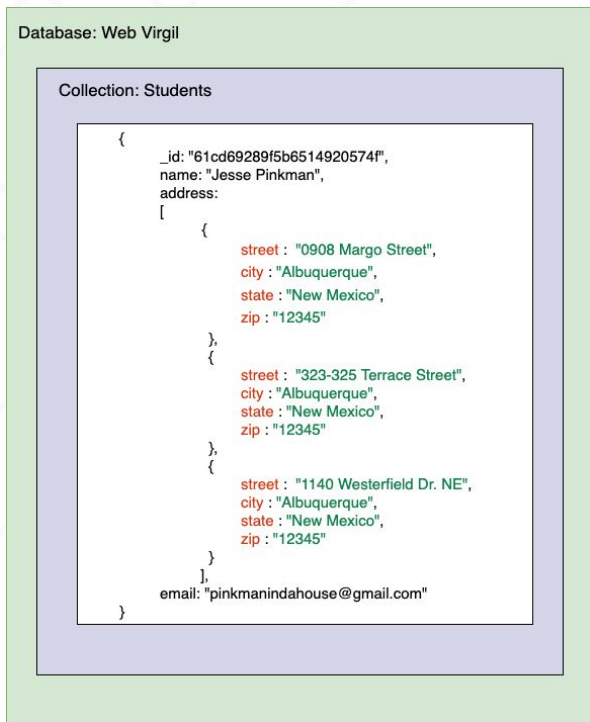
Relaciones 1:N Referenciadas

Si la cardinalidad del "muchos" es grande o crece ilimitadamente, es mejor usar referencias. En este caso, se crea una colección separada para el lado "muchos" y se almacenan las referencias (normalmente el `_id`) en el lado "uno".

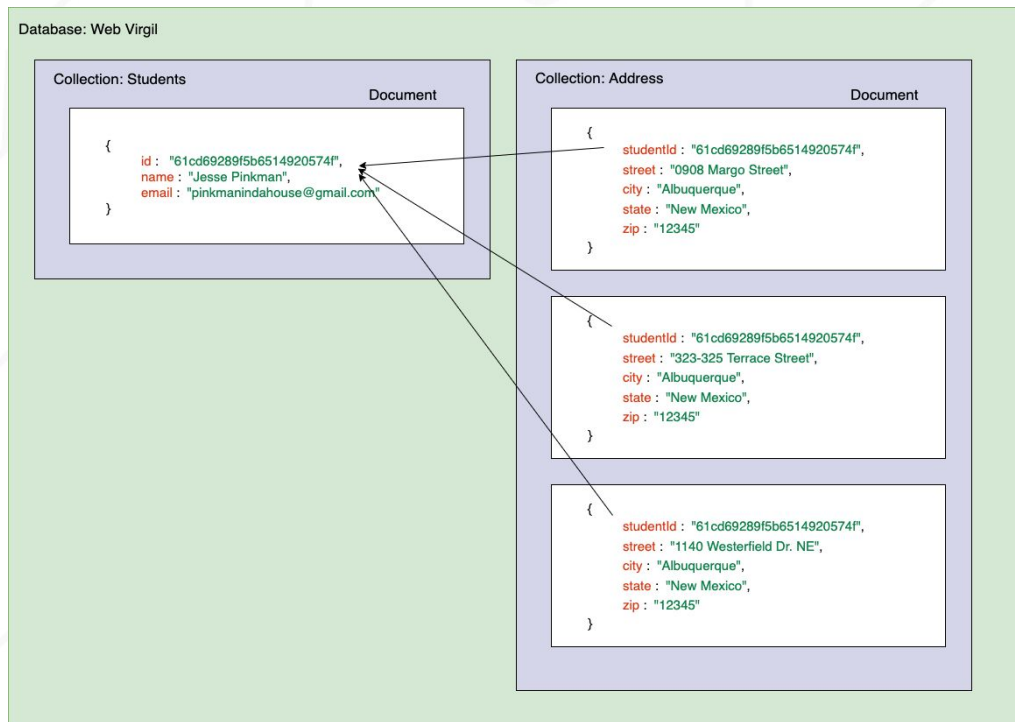
¿Cuándo usar 1:N Embebido?

- Cuando la relación puede incrementar considerablemente.
- Cuando la entidad se actualiza de forma constante.
- Cuando la entidad es usada por muchos documentos.

Relación 1:N Embebida vs Referenciada



Embebida



Referenciada

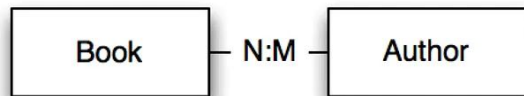
RESUMEN 1:N

Las relaciones 1-N embebidas son relaciones estáticas, por ser características propias del elemento principal, mientras que las relaciones 1-N referenciadas son relaciones dinámicas, es decir, cambian en el tiempo.

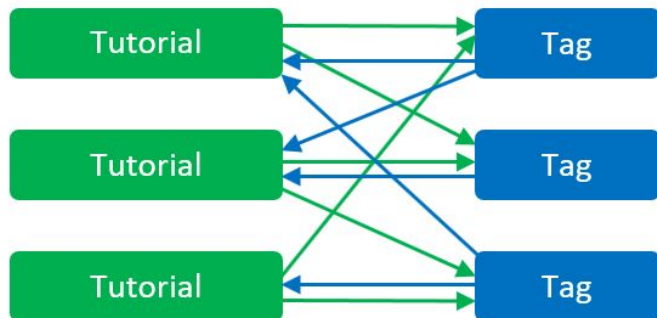
Relaciones N:M (muchos a muchos)

DEV.F
DESARROLLAMOS(PERSONAS);

dev



bezkoder.com



<https://www.mongodb.com/docs/manual/tutorial/model-embedded-one-to-many-relationships-between-documents/>

Relaciones N:M

Es un tipo de relación donde múltiples documentos de una colección están asociados con múltiples documentos de otra colección.

Esta relación es siempre Referencial debido a las restricciones del tamaño del documento de Mongo.

¿Cuándo usar N:N Embebido?

- Siempre que la relación es muchos a muchos.

En MongoDB, **hay dos formas** principales de modelar una relación muchos a muchos (N:M):
Colección de unión y matriz de referencias.

Relaciones N:M

Join Collection

Esta es la forma más común y recomendada de representar una relación N:M en MongoDB.

Se crea una colección separada que actúa como una tabla de unión, donde cada documento contiene un par de referencias (normalmente ***_id***) a los documentos involucrados en la relación.

```
// Colección "productos"
{ "_id": 1, "nombre": "Producto 1" }
{ "_id": 2, "nombre": "Producto 2" }

// Colección "categorias"
{ "_id": 101, "nombre": "Categoría A" }
{ "_id": 102, "nombre": "Categoría B" }

// Colección "producto_categoria" (tabla de unión)
{ "_id": 1, "productoId": 1, "categoriaId": 101 }
{ "_id": 2, "productoId": 1, "categoriaId": 102 }
{ "_id": 3, "productoId": 2, "categoriaId": 102 }
```

Relaciones N:M

Array of References

En este enfoque, se crea una matriz de referencias en uno de los lados de la relación, apuntando a los documentos relacionados en la otra colección.

```
// Colección "productos"
{
  "_id": 1,
  "nombre": "Producto 1",
  "categoriasIds": [101, 102]
}
{
  "_id": 2,
  "nombre": "Producto 2",
  "categoriasIds": [102]
}

// Colección "categorias"
{ "_id": 101, "nombre": "Categoría A" }
{ "_id": 102, "nombre": "Categoría B" }
```

Populate

DEV.F
DESARROLLAMOS(PERSONAS);

dev



```
const books = await Book
  .find({ isActive: true })
  .populate('authors')
```

<https://mongoosejs.com/docs/populate.html>

Populate

Populate en **Mongoose** es una función que permite automáticamente reemplazar los campos especificados en un documento con documentos de otra colección. Esto se usa para resolver las referencias de documentos, similar a cómo se hacen las operaciones JOIN en las bases de datos relacionales. Es similar al operador **\$lookup** de agregaciones en **MongoDB**.



¡GRACIAS!



César Guerra



www.cesarguerra.mx