

# **BD2\_C1B**

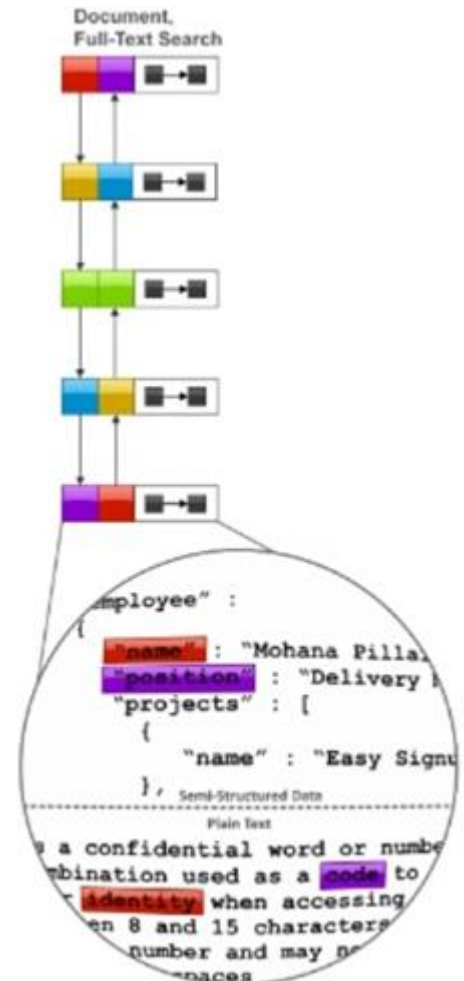
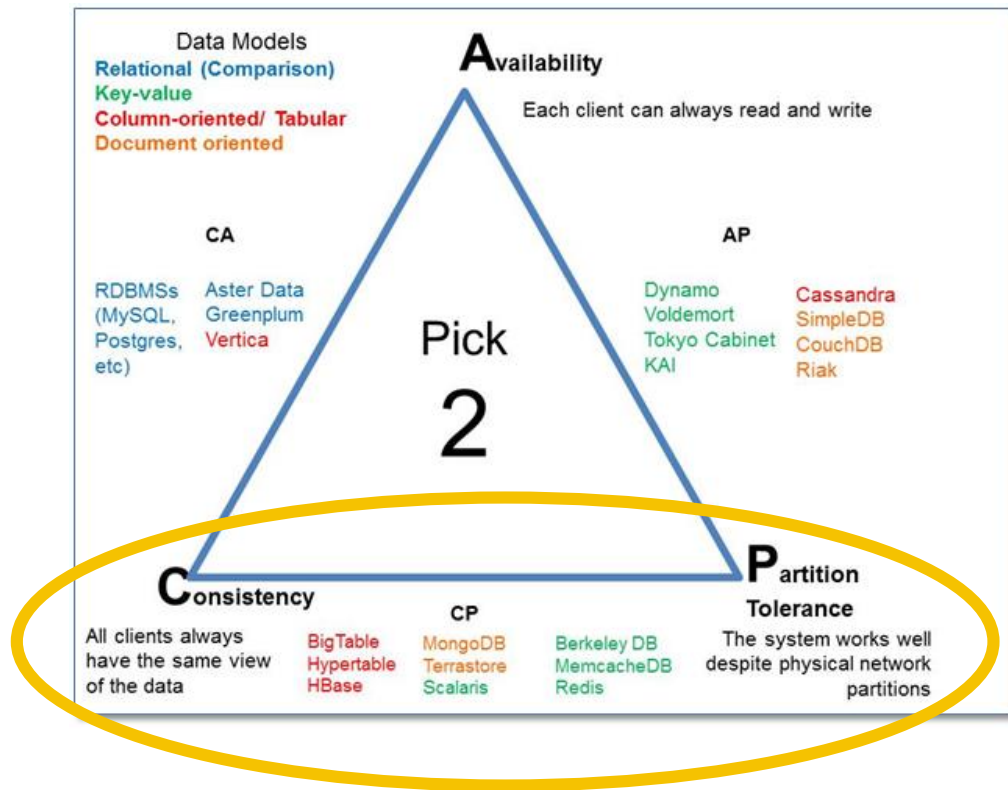
# **MONGO**

# **MODELANDO**

**SERGIO ÁLVAREZ**

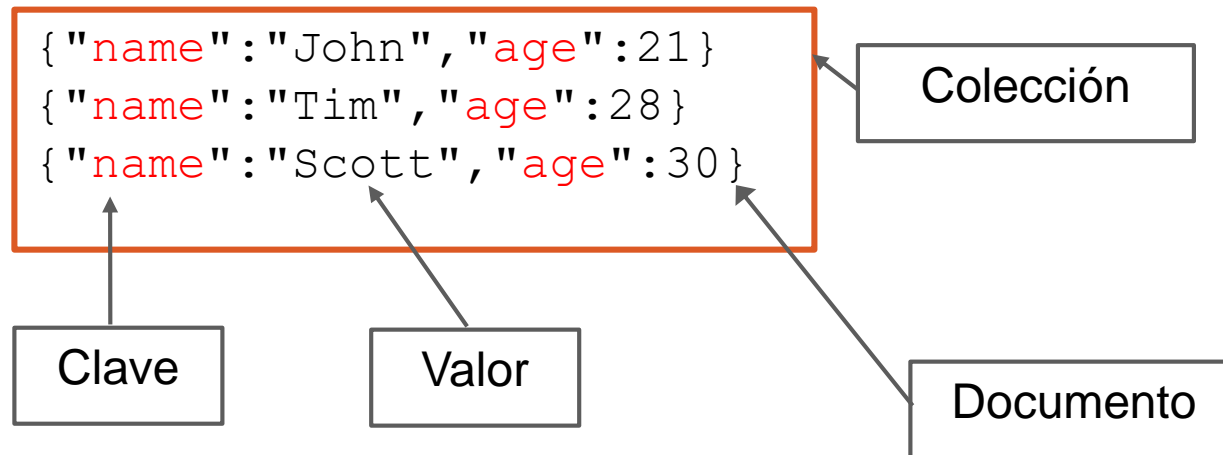
**VERSIÓN 2.0**

# CONSISTENCIA/PARTICIONES DOCUMENTOS



# DOCUMENTOS JSON

JSON (acrónimo de JavaScript Object Notation, «notación de objeto de JavaScript») es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera (año 2019) un formato independiente del lenguaje.



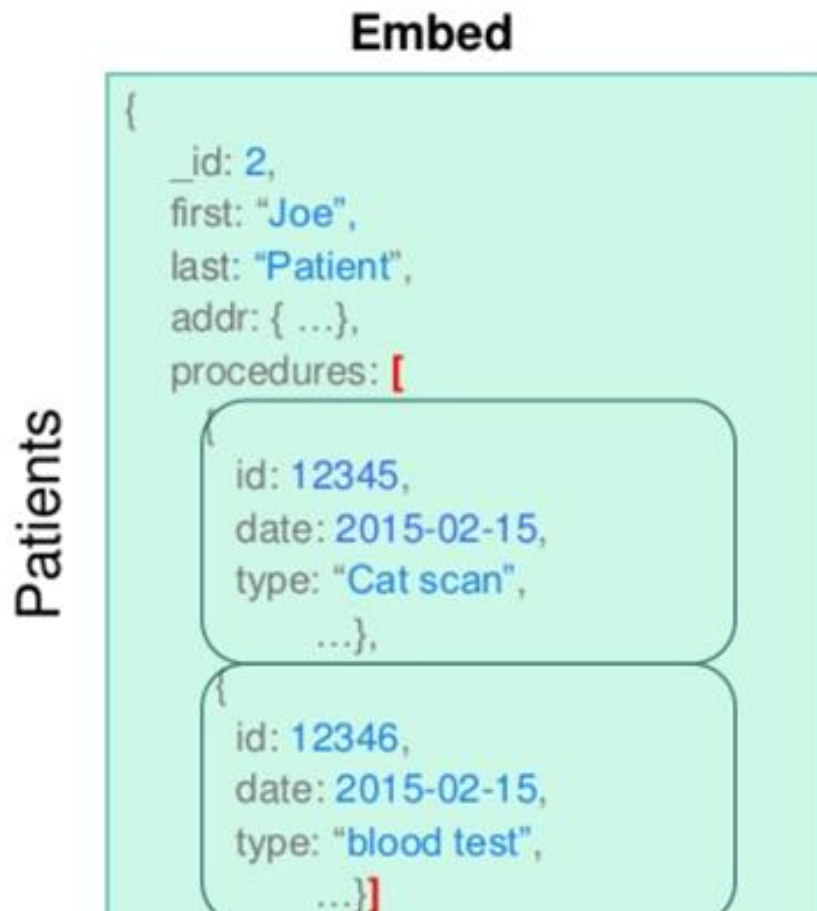
# DOCUMENTOS JSON

```
{
  "Key": "Value"
  "string": "John",
  "number": 123.45,
  "boolean": true,
  "array": [ "a", "b", "c" ],
  "object": { "str": "Miller", "num": 711 },
  "value": NULL,
  "date": ISODate("2013-10-01T00:33:14.000Z")
}
```

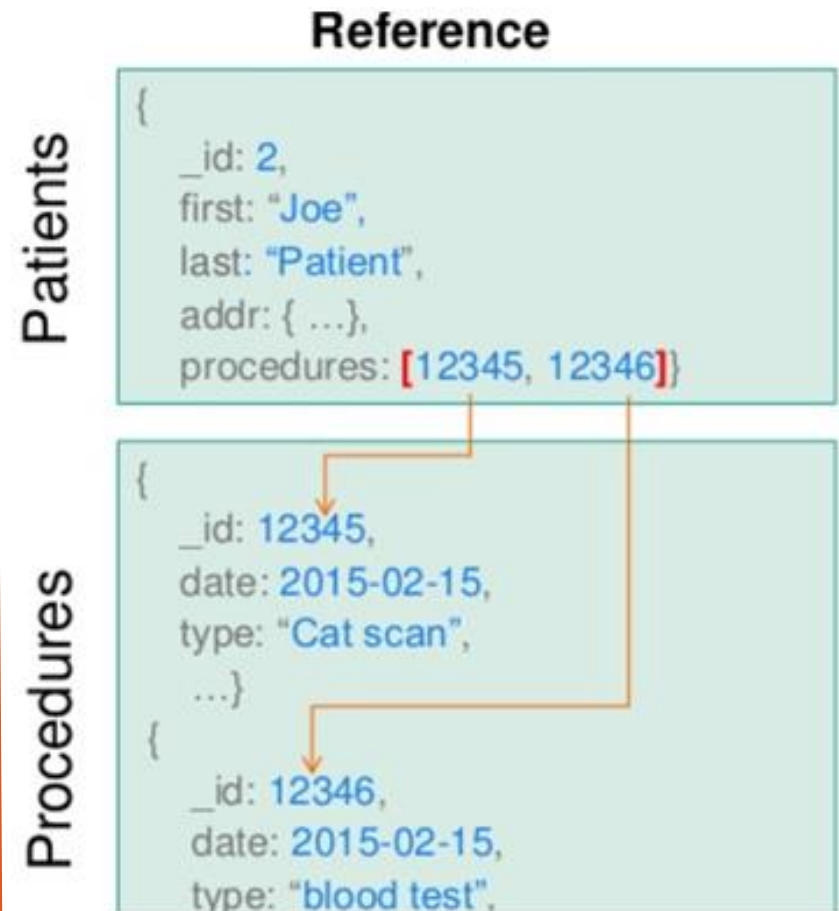
```
[
  {
    Empleado: {
      apellido: "Laban",
      nombre: "Elias"
    }
  },
  {
    Empleado: {
      apellido: "Taylor",
      nombre: "Bradley"
    }
  },
  {
    Empleado: {
      apellido: "Curry",
      nombre: "Rhonda"
    }
  }
]
```

- Números: Se permiten números negativos y opcionalmente pueden contener parte fraccional separada por puntos.
- Cadenas: Representan secuencias de cero o más caracteres. Se ponen entre doble comilla y se permiten cadenas de escape.
- Booleanos: Representan valores booleanos y pueden tener dos valores: true y false
- null: Representan el valor nulo.
- Array: Representa una lista ordenada de cero o más valores los cuales pueden ser de cualquier tipo. Los valores se separan por comas y el vector se mete entre corchetes.
- Objetos: Son colecciones no ordenadas de pares de la forma <nombre>:<valor> separados por comas y puestas entre llaves. El nombre tiene que ser una cadena y entre ellas.

# DOS POSIBLES OPCIONES O PATRONES DE DISEÑO



Una Colección



Dos Colección

# EMBEDDING VS REFERENCING

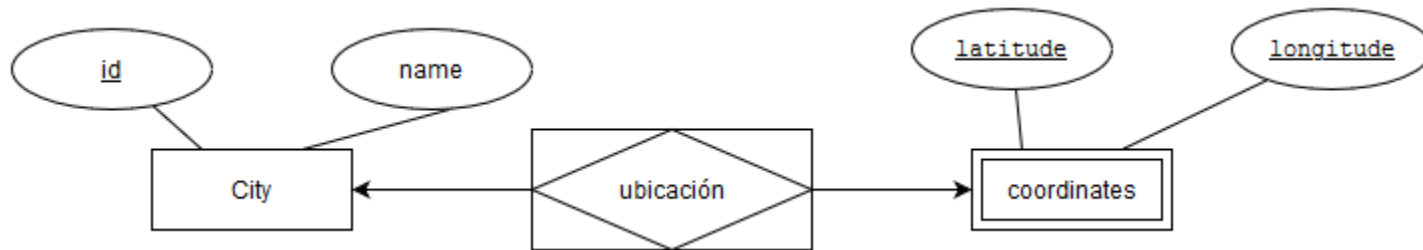
## EMBEBIDO

- Pequeños subdocumentos
- Datos que no cambian regularmente
- La consistencia eventual es aceptable
- Documento que crece por una pequeña cantidad
- Datos que a menudo necesitará realizar una segunda consulta para obtenerlos
- **Lecturas rápidas**

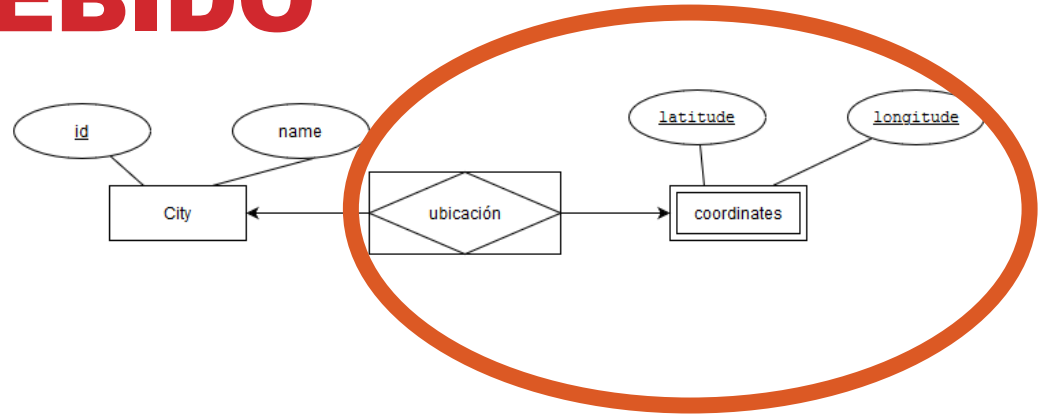
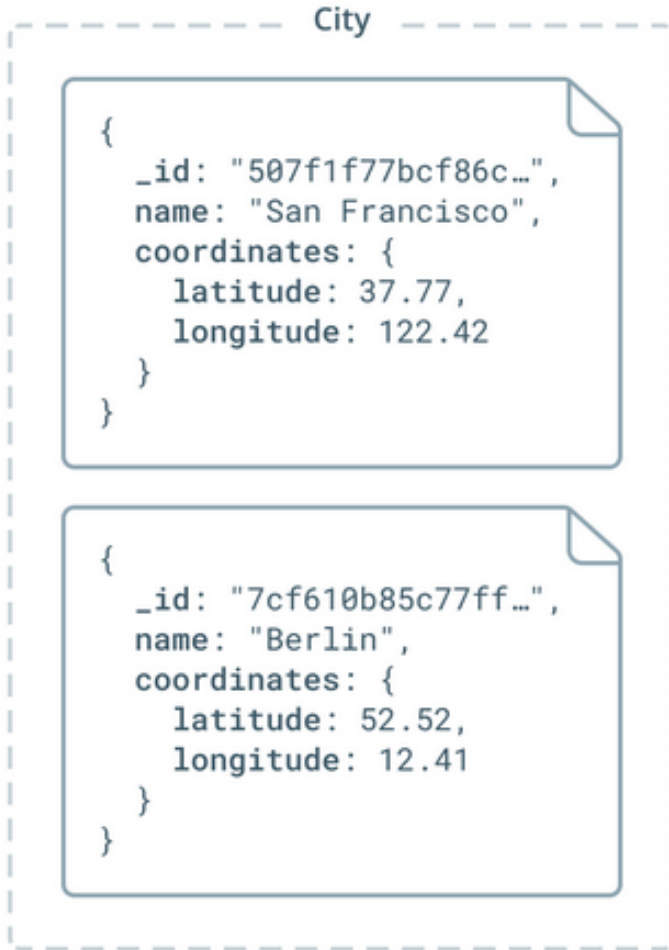
## REFERENCIA

- Grandes subdocumentos
- Datos volátiles
- Es necesaria una consistencia inmediata
- Documento que crece mucho
- Datos que a menudo excluirá de los resultados
- **Escrituras rápidas**

# MODELO ER - EMBEBIDO



# MODELO EMBEBIDO



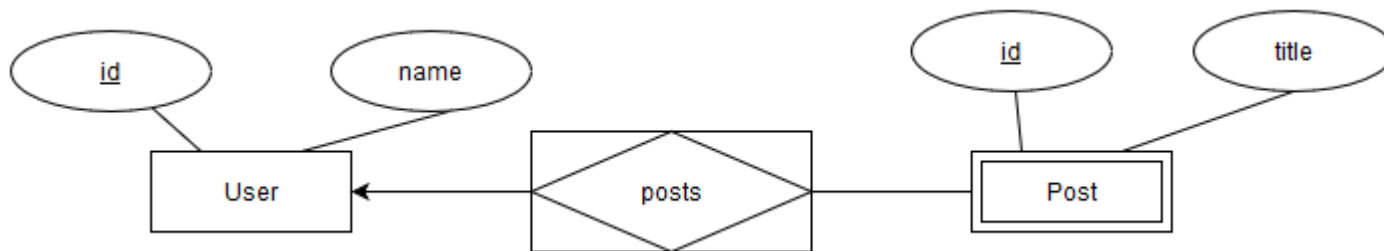
Se diseña como si fuera un atributo o “array de objetos”



# MODELO EMBEBIDO

```
type City {  
  id: ID! @id  
  name: String!  
  coordinates: Coordinates  
}  
  
type Coordinates @embedded {  
  latitude: Float!  
  longitude: Float!  
}
```

# MODELO ER - REFERENCIA



## Opciones

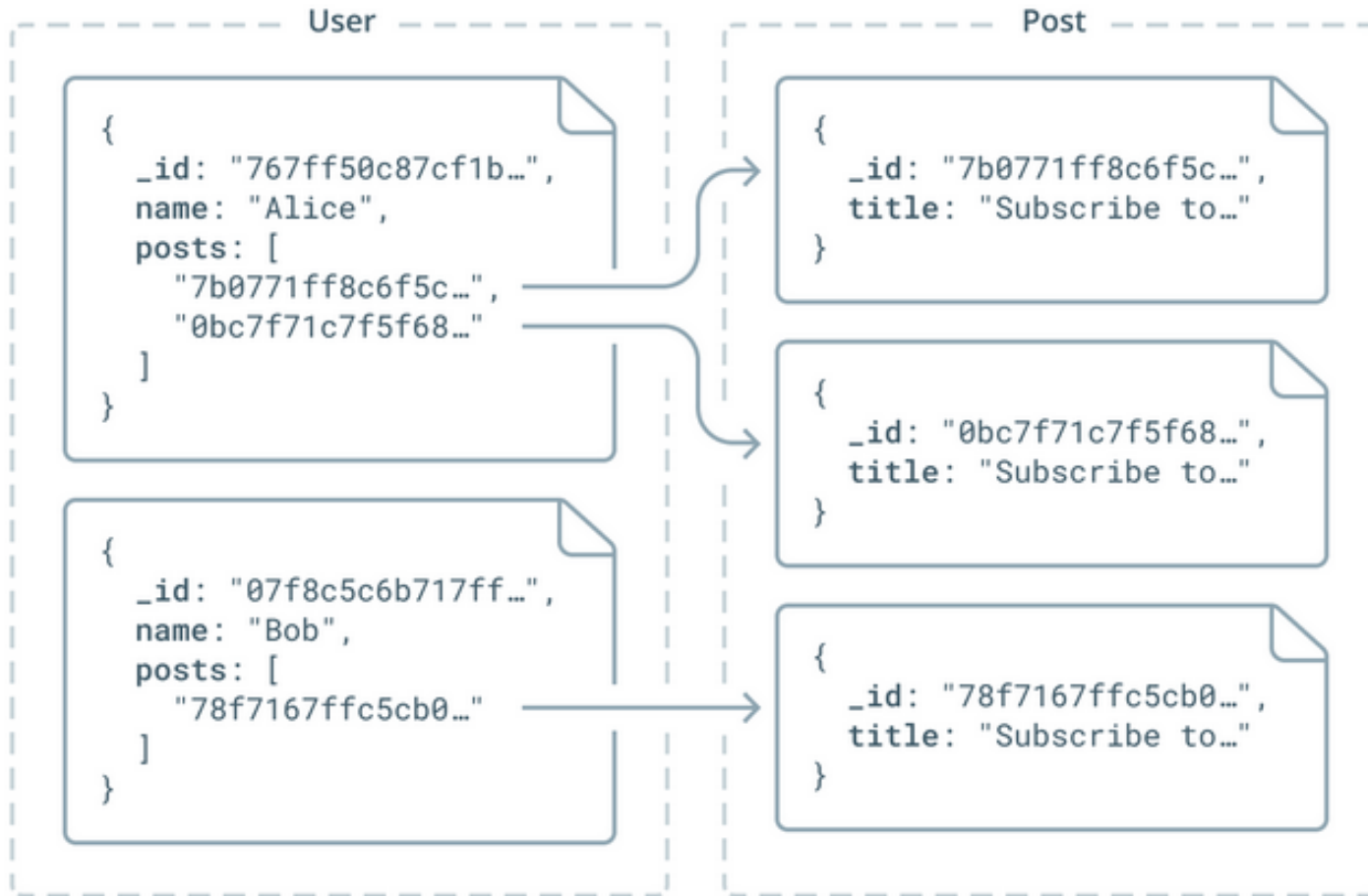
En “User” Arreglo enlaces de “Post”



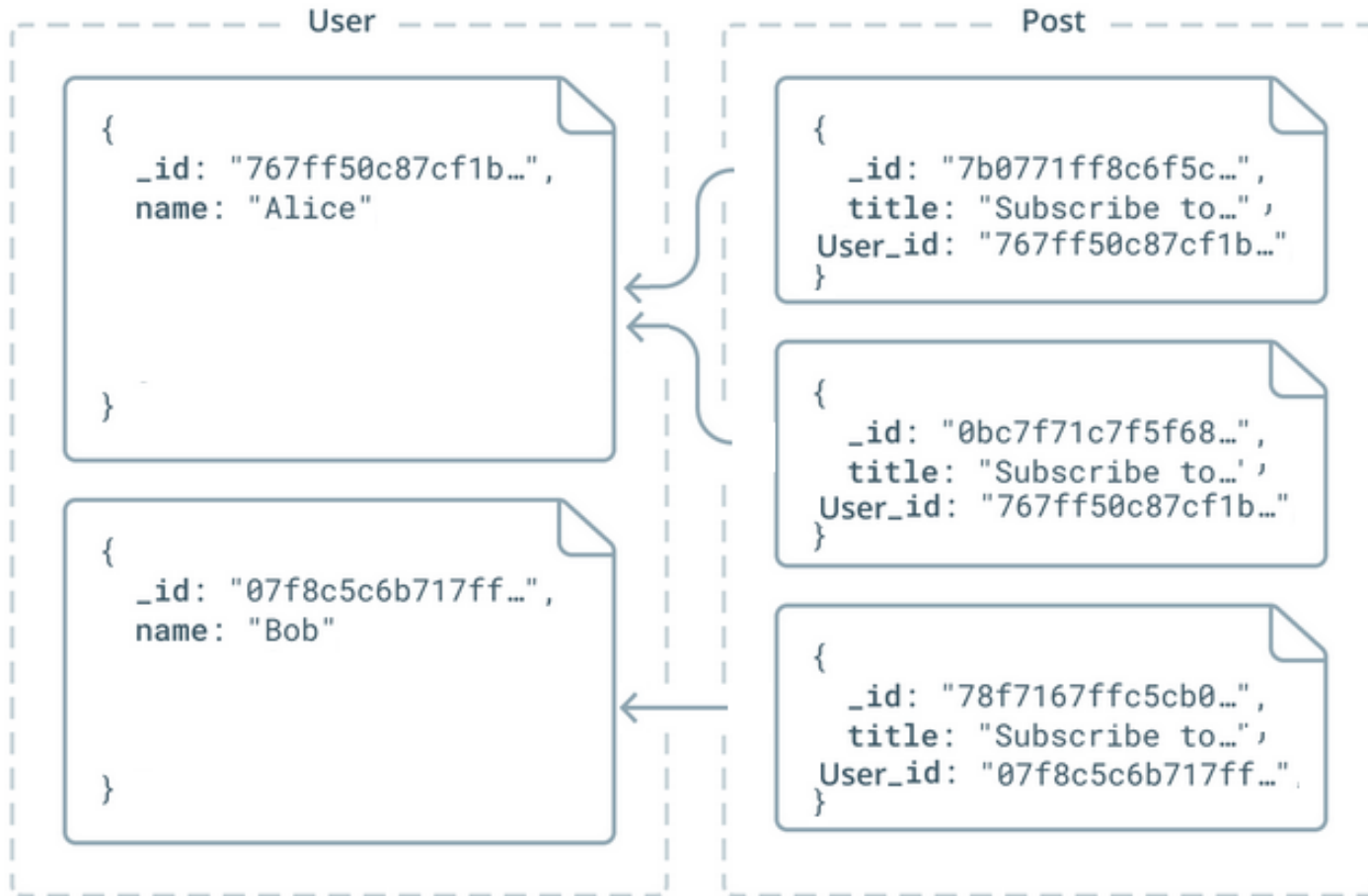
En “Post” enlace a “User”



# MODELO POR REFERENCIA (1)



# MODELO POR REFERENCIA (2)



# MODELO POR REFERENCIA (1)

```
type User {  
  id: ID! @id  
  name: String!  
  posts: [Post!]! @relation(link: INLINE)  
}  
  
type Post {  
  id: ID! @id  
  title: String!  
  author: User!  
}
```

# FIN

[sergalpe@gmail.com](mailto:sergalpe@gmail.com)

