

UNIVERSIDAD DE BUENOS AIRES  
FACULTAD DE CIENCIAS EXACTAS Y NATURALES  
DEPARTAMENTO DE COMPUTACIÓN



**Bases de Datos**  
**Trabajo Práctico #2**  
Ciudad Gótica



---

## Introducción y Objetivos

El objetivo de este segundo trabajo práctico es que, a partir de un dominio del problema ya tratado, se desarrolle una base de datos NoSQL de tal manera que se pueda comparar las características específicas de este modelo de datos con respecto al relacional. Se utilizará en esta ocasión ArangoDB como motor de base de datos basada en documentos. En todos los casos los alumnos deberán asegurarse de contar con el software necesario para poder mostrar el trabajo práctico en las fechas y lugar de entrega.

Consignas: Al momento de la corrección se tendrán en cuenta tanto la correctitud de la solución como el uso de las herramientas disponibles. La entrega deberá constar, como mínimo, de la siguiente documentación:

- Carátula. Con tabla de contenidos, título del trabajo, fecha y nombre de los autores.
- Introducción y explicación del problema a resolver.
- Modelo Conceptual y Modelo de Interacción de Documentos
- Detalle de los supuestos asumidos para la resolución del problema.
- Diseño lógico correspondiente a la solución (JSONSchema)
- Diseño físico implementado en **ArangoDB**
- Código correspondiente a las consultas que se piden en el punto *Funcionalidades a Implementar*
- Conclusiones

Además, la base que se use para efectuar la demostración deberá contener datos de prueba cargados, de forma de poder evaluar el funcionamiento de las consultas incluidas en los requerimientos. No es necesario entregar una interfaz para ejecutar las consultas; las mismas podrán ser ejecutadas directamente desde la interfaz del motor de base de datos.

*Recomendamos revisar el avance del trabajo con el tutor asignado antes de la fecha de entrega.*

## 1 Enunciado del Problema

Se desea implementar el mismo espacio del problema que el TP1 en una base de datos basada en documentos. Para ello, deben tomar como entrada el modelo de entidad-relación desarrollado en el TP1.

Para implementar estas funcionalidades se decide usar **ArangoDB** como base de datos basada en documentos.

## 2 Funcionalidades a Implementar

- 2.1. Partir del modelo conceptual parcial tomado del DER del TP1
- 2.2. A partir del modelo conceptual realizar el diagrama de interacción de documentos y los JSON Schema correspondientes.
- 2.3. Implementar los tipos de documentos en **ArangoDB**.
- 2.4. Realizar las siguientes consultas:
  - (a) Dada una organización delictiva, el detalle de incidentes en que participaron las personas que componen dicha organización
  - (b) Los datos de los sumarios para un oficial dado.
  - (c) Los superheroes que participaron en algún incidente.
  - (d) Los seguimientos de un incidente en particular

## 3 DID - Ternarias y Agregaciones

En esta sección daremos algunos tips para modelar ternarias y agregaciones. Ante cualquier duda consultar con su tutor.

### 3.1. Modelar Ternarias

Supongamos el caso de la figura 1 donde tenemos una interrelación ternaria entre Alumno, Cuatrimestre y Materia. La cardinalidad en este caso es  $n:m:p$ . Para un alumno y un cuatrimestre hay muchas materias que cursa, un cuatrimestre y una materia tiene muchos alumnos, para una materia y un alumno puede haber muchos cuatrimestres donde la cursa. Obviamos los atributos para no complicar el diagrama. En principio tendríamos un tipo de documento por cada una de las entidades del DER.

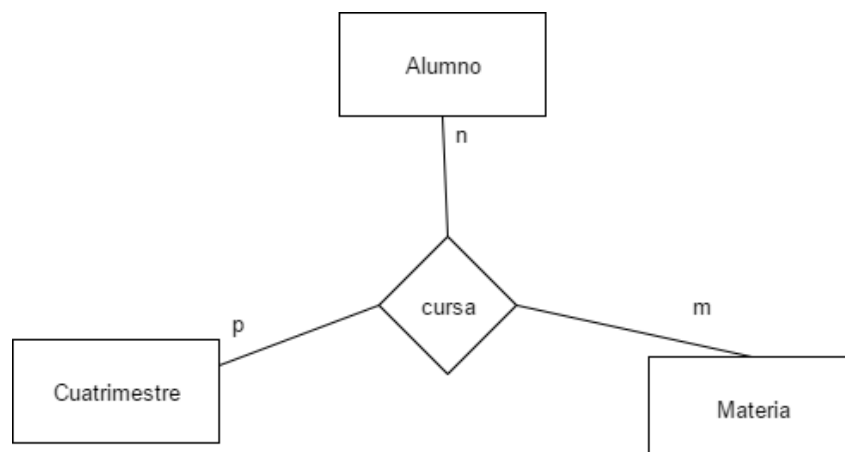


Fig. 1: DER Ternarias

La parte mas compleja es decidir como modelar la interrelación *cursa*. La decisión de cómo modelar dependerá, como siempre, de los patrones de consulta. El caso básico es el que se quiera generar un tipo de documento que simplemente guarde la información de la interrelación con los identificadores de cada una de las entidades involucradas. En ese caso basta un documento auxiliar cómo se ve en la figura 2.

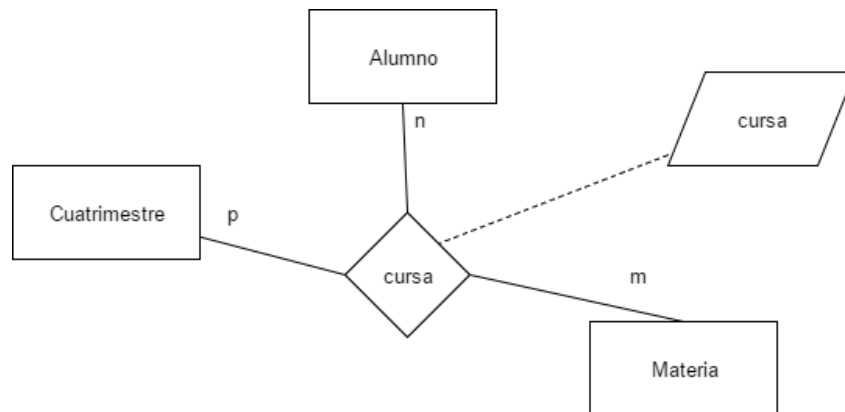


Fig. 2: DID Ternarias - 1

Si no se especifica otra cosa ese tipo de documento nuevo tendrá las claves de las entidades participantes de la ternaria.

Supongamos que además una consulta muy común es saber cuales son los alumnos anotados en una materia en un cuatrimestre, de hecho se quiere saber nombre, apellido y número de ellos para una materia y un cuatrimestre dado. En este caso necesitaríamos un documento que guarde los identificadores de cuatrimestre y materia más un vector con los datos que se necesitan de los alumnos.

El DID para modelar este caso se muestra en la figura 3. La semántica de este diagrama es de que aquellas entidades participantes de la interrelación que no están relacionadas mediante algún vínculo con el nuevo tipo de documento se agregan las claves. Lo que indica este diagrama es que por cada par cuatrimestre/materia se tiene un vector con el nombre, apellido y número de cada alumno

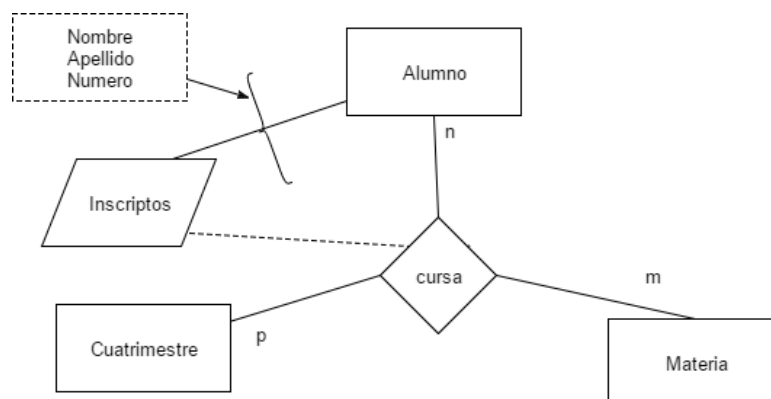


Fig. 3: DID Ternarias - Caso mas complejo

En caso de necesitarse además de la clave otros atributos de algunas de las entidades puede indicarse directamente en el diagrama. Supongamos por caso que quisiéramos el nombre de la materia y el número y año del cuatrimestre tendríamos lo que se ve en la figura 4

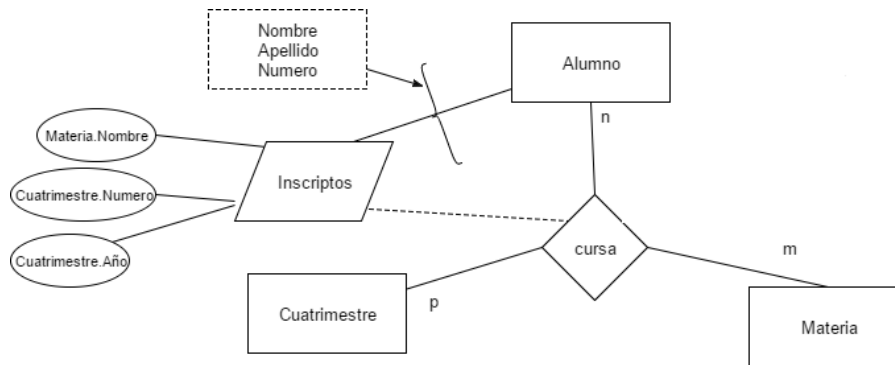


Fig. 4: DID Ternarias - Con atributos de las entidades además de las claves

Las combinaciones son variadas, se debe elegir la que mejor se adapte al dominio del problema.

### 3.2. Modelar Agregación

Supongamos la agregación de la figura 5. Tenemos en principio tres tipos de documentos: *Hotel*, *Servicio* y *Evaluación*. La interrelación *cuenta\_con* puede resolverse de muchas formas como ya vimos: embeber de un lado, embeber de ambos, embeber de un lado y referenciar de otro, referenciar de ambos, referencia de un solo lado, etc.

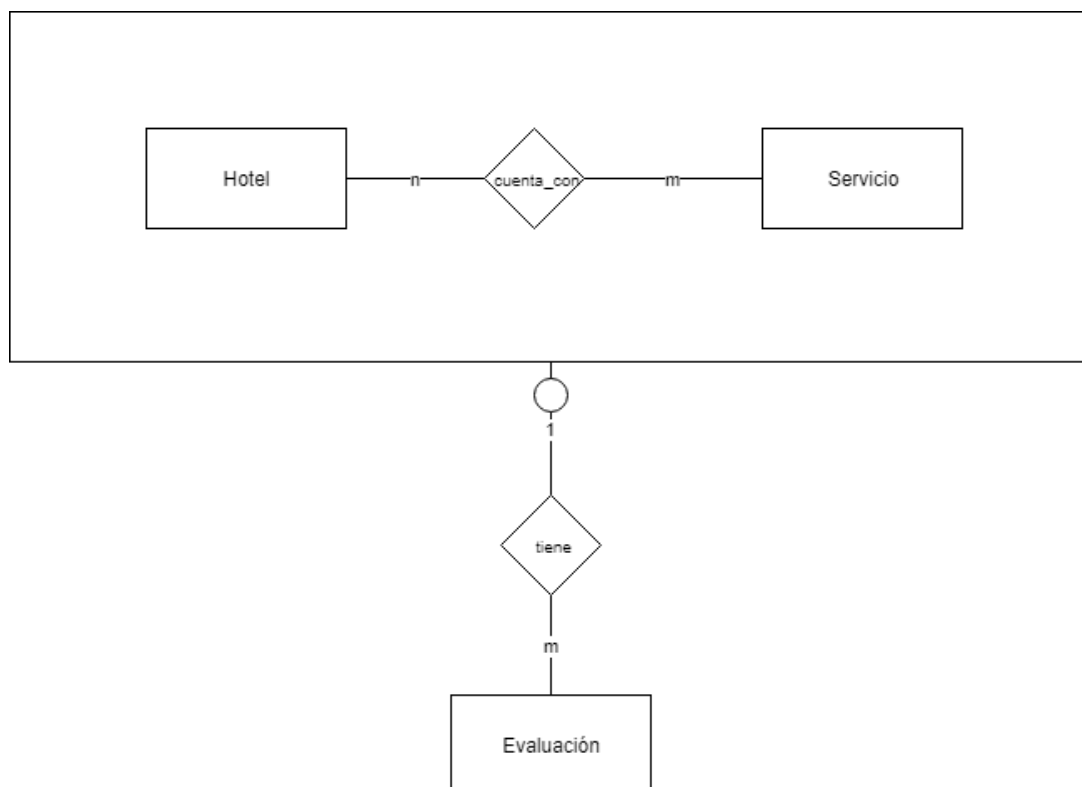


Fig. 5: DER Agregación

Lo que hay que decidir es cómo se resuelve la interrelación entre la agregación y la evaluación. La forma mas simple es es generar un tipo de documento con las claves de las dos relaciones Hotel y Servicio y embeber o referenciar a la entidad

evaluación. Ello se modela como en la figura 6. La agregación se transforma en un tipo de documento con las claves de Hotel y Entidad y una lista de evaluaciones asociadas. En ese ejemplo la interrelación  $m:n$  se resolvió referenciando de ambos lados. En caso de no haberse indicado nada en para interrelación  $m:n$  quedaría el documento con las dos claves y la lista de evaluaciones (eventualmente vacía) similar a si creáramos un documento auxiliar que en este caso sería la agregación.

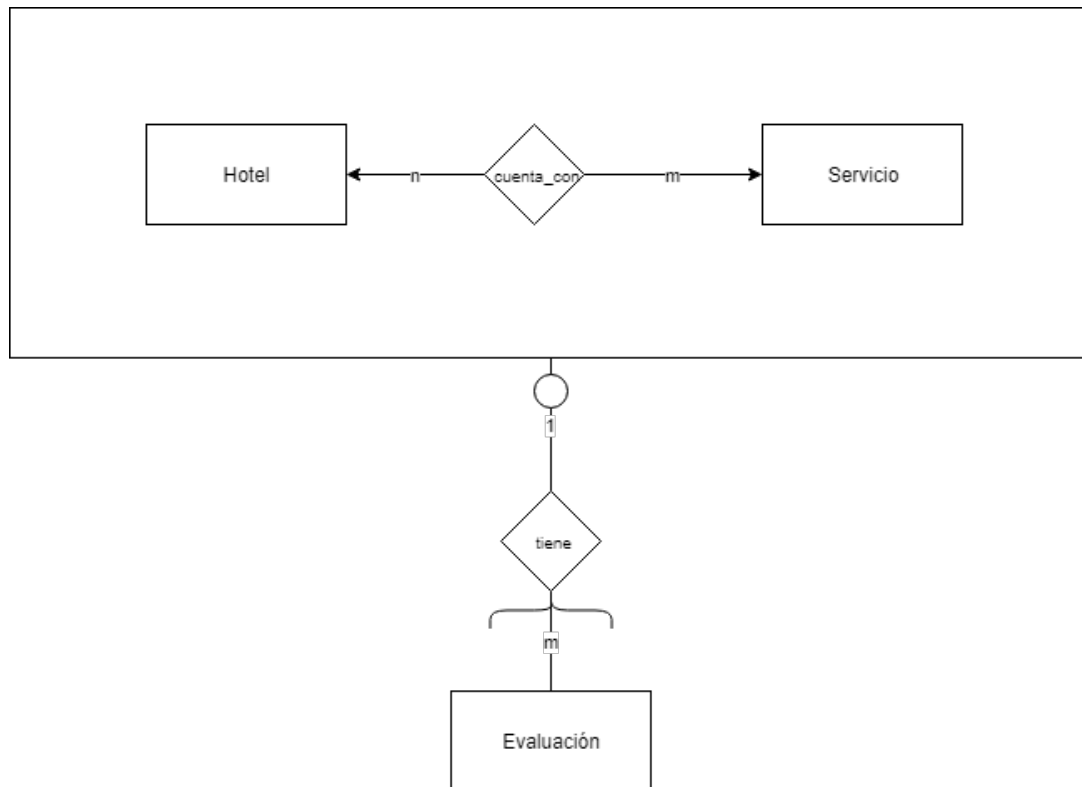


Fig. 6: DID Agregación - Modelo Básico

Supongamos ahora que la interrelación  $m:n$  entre Hotel y Servicio se resuelve embebiendo los servicios con su información en los Hoteles y referenciando Hoteles desde Servicio. En ese caso se puede decidir: agregar la información de las Evaluaciones en la lista de servicios de cada Hotel o dejar el modelo como en el caso anterior. Ambas alternativas se ven en las figuras 7 y 8

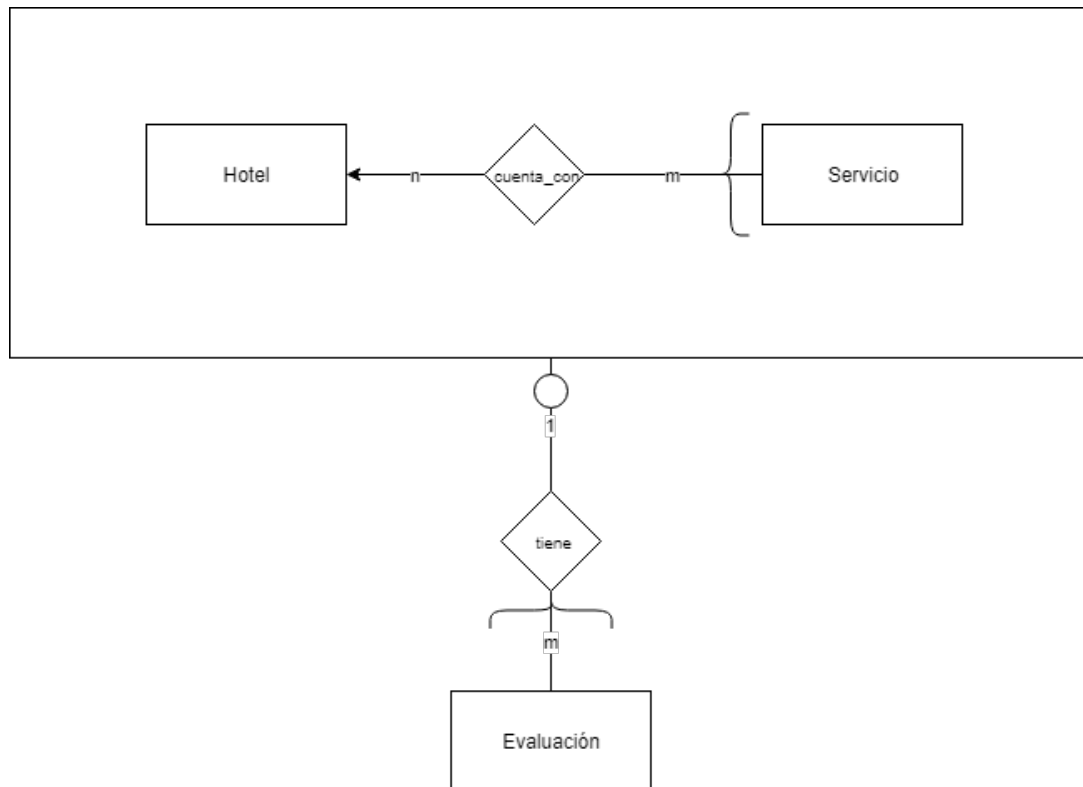


Fig. 7: DID Agregación - Solución Alternativa

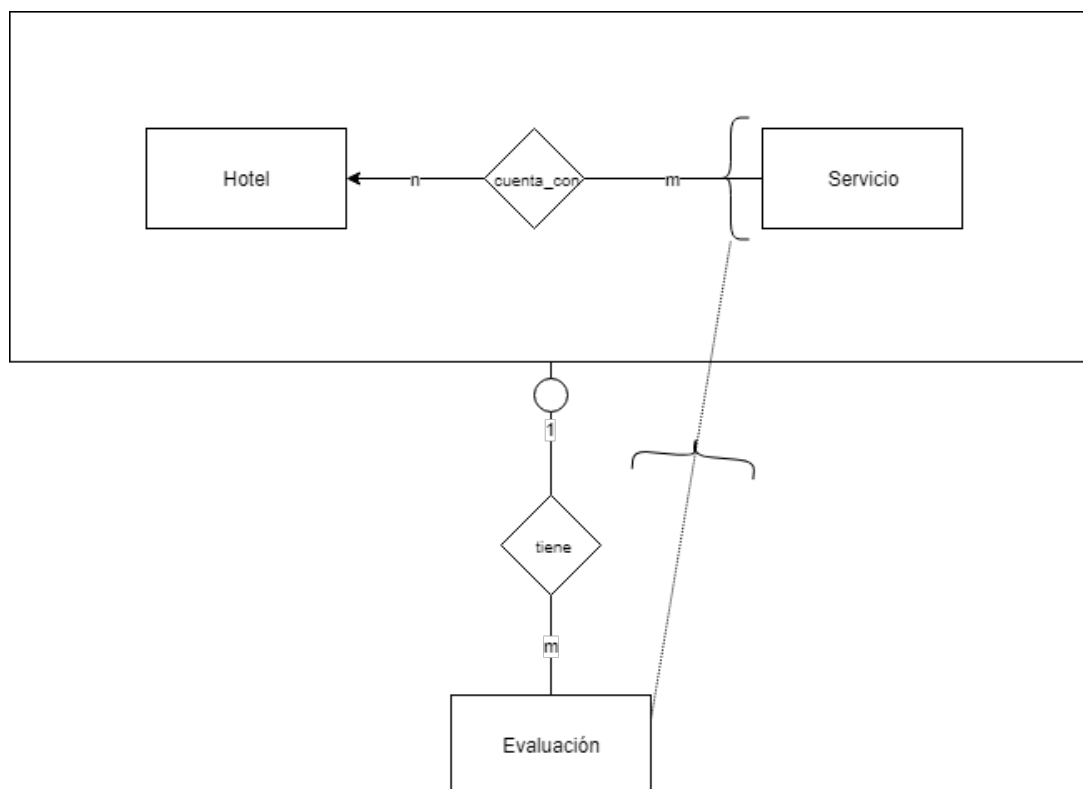


Fig. 8: DID Agregación - Otra Solución Alternativa