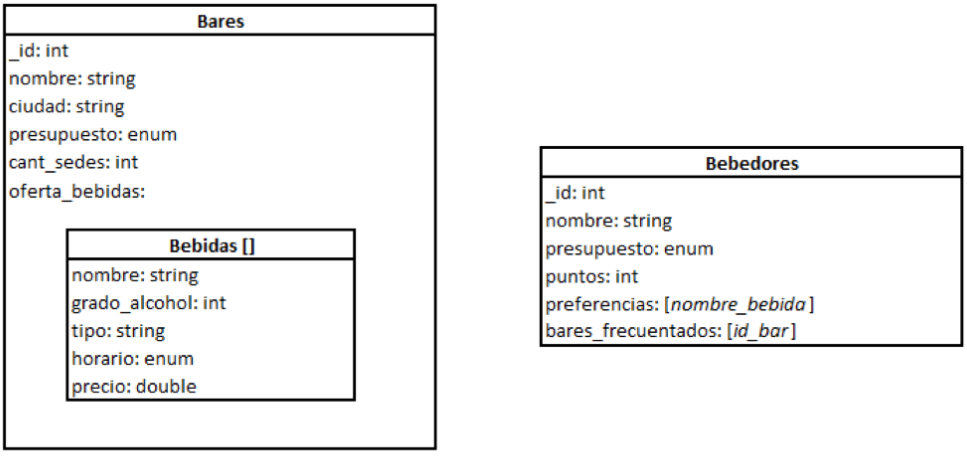
Tutorial: Cómo Ejecutar un Pipeline de Agregación en MongoDB para Obtener las Top 3 Bebidas Más Consumidas

En este tutorial, vamos a construir una consulta avanzada utilizando **MongoDB** para obtener las **Top 3 bebidas** que son más servidas en los bares, teniendo en cuenta solo aquellos bares **frecuentados por bebedores** y **bebidas preferidas por esos bebedores**. A lo largo del tutorial, utilizaremos ejemplos reales para que sea más fácil entender cómo funciona cada etapa del pipeline.

Aquí tienes el esquema de la base de datos con el que vamos a trabajar:



Lo que Retorna la Consulta:

La consulta devuelve las **Top 3 bebidas que son más consumidas**, pero solo se tienen en cuenta las siguientes condiciones:

1. **Bares frecuentados por los bebedores**.
2. **Bebidas preferidas por los bebedores**.

Ejemplo de Datos Iniciales

Para entender mejor cómo funciona la consulta, vamos a trabajar con estos datos ficticios en las colecciones de MongoDB:

Colección Bares

_id	nombre	ciudad	oferta_bebidas
1	Bar Uno	Bogotá	[{nombre: "Cerveza", precio: 5000}, {nombre: "Ron", precio: 15000}]
2	Bar Dos	Medellín	[{nombre: "Vino", precio: 12000}, {nombre: "Cerveza", precio: 4500}]
3	Bar Tres	Bogotá	[{nombre: "Tequila", precio: 13000}]

Colección Bebedores

_id	nombre	preferencias	bares_frecuentados
1	Juan	["Cerveza", "Ron"]	[1, 2]
2	Pedro	["Tequila", "Vino"]	[3, 1]
3	María	["Cerveza", "Vino"]	[2, 1]

Implementación del Código y Explicación con Ejemplos

Paso 1: Descomponer la Lista de Bares Frecuentados ($unwind)

- **¿Qué hace este paso?**: Este paso toma la lista de bares que un bebedor ha frecuentado y la descompone en varios documentos, uno por cada bar. Si un bebedor ha visitado varios bares, MongoDB creará un documento separado para cada bar.

- **¿En qué parte de la consulta estamos?**: Estamos comenzando a preparar los datos, tratando cada bar frecuentado por separado para analizarlos individualmente.

- **Entrada**: Un documento de bebedor con una lista de bares frecuentados.

- **Salida**: Varios documentos, cada uno con un solo bar frecuentado por el bebedor.

```
new Document(key:"$unwind", value:"$bares_frecuentados"),
```

Ejemplo de Ejecución:

En la colección bebedores, Juan frecuenta los bares 1 y 2, lo que da como resultado:

_id	nombre	preferencias	bares_frecuentados
1	Juan	["Cerveza", "Ron"]	1
1	Juan	["Cerveza", "Ron"]	2

El $unwind ha descompuesto la lista bares_frecuentados en documentos separados.

Paso 2: Unir la Colección bares con bebedores ($lookup)

- **¿Qué hace este paso?**: Este paso busca los detalles de cada bar que frecuenta el bebedor, utilizando la colección bares, y añade esa información al documento del bebedor.

- **¿En qué parte de la consulta estamos?**: Estamos ampliando la información de cada bar que frecuenta el bebedor, obteniendo los detalles de las bebidas que ese bar ofrece.

- **Entrada**: Un documento de bebedor con un ID de bar.

- **Salida**: Un documento de bebedor con los detalles del bar incluido.

```
new Document(key:"$lookup", new Document()
    .append(key:"from", value:"bares")
    .append(key:"localField", value:"bares_frecuentados")
    .append(key:"foreignField", value:"_id")
    .append(key:"as", value:"bar_detalle")
),
```

Ejemplo de Ejecución:

Ahora, unimos los bares frecuentados por Juan con los detalles de los bares:

_id	nombre	preferencias	bares_frecuentados	bar_detalle
1	Juan	["Cerveza", "Ron"]	1	{nombre: "Bar Uno", oferta_bebidas: [...] }
1	Juan	["Cerveza", "Ron"]	2	{nombre: "Bar Dos", oferta_bebidas: [...] }

Esto añade los detalles del bar al documento de cada bebedor.

Paso 3: Descomponer el Campo bar_detalle y oferta_bebidas

- **¿Qué hace este paso?**: Este paso descompone la lista de bebidas ofrecidas por cada bar. Si un bar ofrece múltiples bebidas, se crea un documento separado por cada bebida.

- **¿En qué parte de la consulta estamos?**: Estamos preparando cada bebida de los bares para poder compararla con las preferencias de los bebedores.

- **Entrada**: Un documento con los detalles del bar y su lista de bebidas.

- **Salida**: Varios documentos, uno por cada bebida que ofrece el bar.

```
new Document(key:"$unwind", value:"$bar_detalle"),
new Document(key:"$unwind", value:"$bar_detalle.oferta_bebidas"),
```

Ejemplo de Ejecución:

Ahora descomponemos la oferta de bebidas de cada bar:

_id	nombre	preferencias	bares_frecuentados	bar_detalle	oferta_bebidas
1	Juan	["Cerveza", "Ron"]	1	Bar Uno	{nombre: "Cerveza", ...}
1	Juan	["Cerveza", "Ron"]	1	Bar Uno	{nombre: "Ron", ...}
1	Juan	["Cerveza", "Ron"]	2	Bar Dos	{nombre: "Cerveza", ...}
1	Juan	["Cerveza", "Ron"]	2	Bar Dos	{nombre: "Vino", ...}

Cada bebida ofrecida por los bares ha sido descompuesta en un documento separado.

Paso 4: Descomponer las Preferencias de los Bebedores

- **¿Qué hace este paso?**: Este paso descompone la lista de preferencias de cada bebedor. Si un bebedor tiene varias bebidas en su lista de preferencias, MongoDB creará un documento separado por cada preferencia.

- **¿En qué parte de la consulta estamos?**: Estamos preparando las preferencias de los bebedores para compararlas con las bebidas que ofrecen los bares.

- **Entrada**: Un documento con una lista de preferencias.

- **Salida**: Varios documentos, uno por cada bebida en la lista de preferencias del bebedor.

```
new Document(key:"$unwind", value:"$preferencias"),
```

Ejemplo de Ejecución:

Ahora descomponemos las preferencias de bebidas de cada bebedor:

_id	nombre	preferencias	bares_frecuentados	bar_detalle	oferta_bebidas
1	Juan	"Cerveza"	1	Bar Uno	{nombre: "Cerveza", ...}
1	Juan	"Ron"	1	Bar Uno	{nombre: "Ron", ...}
1	Juan	"Cerveza"	2	Bar Dos	{nombre: "Cerveza", ...}
1	Juan	"Ron"	2	Bar Dos	{nombre: "Vino", ...}

Las preferencias de Juan también se han descompuesto en documentos individuales.

Paso 5: Filtrar las Bebidas que Coinciden con las Preferencias ($match)

- **¿Qué hace este paso?**: Este paso filtra los documentos para quedarse solo con las bebidas que el bar ofrece y que además están en la lista de preferencias del bebedor. Si la bebida ofrecida por el bar no está en la lista de preferencias del bebedor, ese documento se elimina.

- **¿En qué parte de la consulta estamos?**: Estamos comparando las bebidas ofrecidas por los bares con las preferencias de los bebedores. Este es el paso clave donde se cruzan los datos.

- **Entrada**: Un documento con una bebida del bar y una preferencia del bebedor.

- **Salida**: Solo se retienen los documentos donde el nombre de la bebida ofrecida coincide con el nombre de la preferencia. Si no coinciden, el documento se descarta.

```
new Document(key:"$match", new Document()
    .append(key:"$expr", new Document()
        .append(key:"$eq", List.of(e1:"$bar_detalle.oferta_bebidas.nombre", e2:"$preferencias"))
    )
),
```

Ejemplo de Ejecución:

Ahora filtramos solo las bebidas que coinciden con las preferencias de Juan:

_id	nombre	preferencias	bares_frecuentados	bar_detalle	oferta_bebidas
1	Juan	"Cerveza"	1	Bar Uno	{nombre: "Cerveza", ...}
1	Juan	"Ron"	1	Bar Uno	{nombre: "Ron", ...}
1	Juan	"Cerveza"	2	Bar Dos	{nombre: "Cerveza", ...}

Se eliminan las bebidas que no coinciden con las preferencias del bebedor, en este caso el "Vino" en el Bar Dos.

Paso 6: Agrupar por Bebida y Contar Cuántas Veces Fue Consumida ($group)

- **¿Qué hace este paso?**: Agrupa los documentos por el nombre de la bebida (es decir, por cada preferencia) y cuenta cuántas veces ha sido consumida por los bebedores. Cada vez que una bebida preferida coincide con una bebida ofrecida en un bar frecuentado, se suma 1 al total.

- **¿En qué parte de la consulta estamos?**: Estamos agrupando y contando cuántas veces ha sido consumida cada bebida que cumple con las condiciones.

- **Entrada**: Un documento con la bebida que coincide con las preferencias del bebedor.

- **Salida**: Un documento por cada bebida, con el total de veces que ha sido consumida.

```
new Document(key:"$group", new Document()
    .append(key:"_id", value:"$preferencias")
    .append(key:"total", new Document()
        .append(key:"$sum", value:1)
    )
),
```

Ejemplo de Ejecución:

Agrupamos por el nombre de la bebida preferida y contamos cuántas veces ha sido consumida:

_id	total
Cerveza	2
Ron	1

El resultado muestra que "Cerveza" fue consumida dos veces y "Ron" una vez.

Paso 7: Ordenar y Limitar el Resultado ($sort y $limit)

- **¿Qué hace este paso?**: Ordena los resultados de mayor a menor en función del total de veces que ha sido consumida cada bebida. Luego, limita los resultados a las 3 bebidas más consumidas.

- **¿En qué parte de la consulta estamos?**: Estamos finalizando la consulta, ordenando los resultados y limitando el número de bebidas a las Top 3 más consumidas.

- **Entrada**: Varios documentos con el nombre de la bebida y el total de veces consumida.

- **Salida**: Los 3 documentos que representan las bebidas más consumidas, ordenados de mayor a menor.

```
new Document(key:"$sort", new Document()
    .append(key:"total", -1)
),

new Document(key:"$limit", value:3)
```

Ejemplo de Ejecución:

Finalmente, ordenamos las bebidas por el número total de veces que han sido consumidas y limitamos el resultado a las 3 más consumidas:

_id	total
Cerveza	2
Ron	1

Este es el resultado final, con las **Top 3 bebidas** más consumidas en este caso (aunque en este ejemplo solo hay 2 bebidas consumidas).

Conclusión

Con este pipeline de agregación hemos logrado:

1. Filtrar solo los bares frecuentados por los bebedores.
2. Considerar solo las bebidas que son preferidas por los bebedores.
3. Devolver las **Top 3 bebidas** más consumidas entre las que cumplen estos criterios.