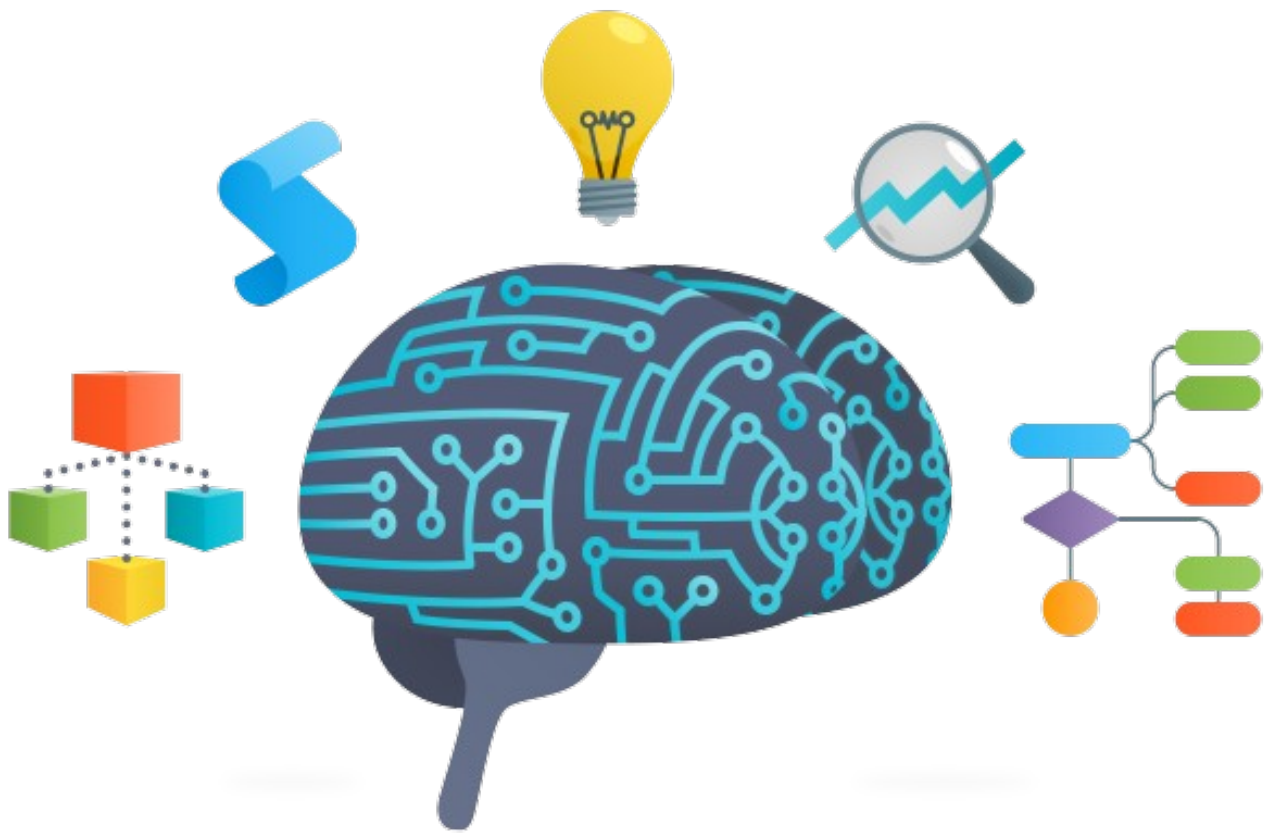


# PROYECTO VENTAS

## GBD



**Nombre:** Diego José Fernández Augusto  
**Curso:** 1º ASIR  
**Módulo:** Gestión de Base de Datos

-En este nuevo proyecto nos requería una formación básica sobre bases de datos dada por el anterior proyecto, lo cual ha ayudado mucho a la creación de esta nueva base de datos con sus consultas.

-Dentro del proyecto nos encontraremos nuevos operadores del comando `.aggregate()`, que vamos a explicar su función en este .pdf, además de haber utilizado los que conocíamos ya del `.find()` y varias aportaciones personales.

```
db.drogueria.aggregate([{\n  $group:{_id: "$articulo", cantidadDeArticulo: {$sum: "$unidades"},\n  Cantidad_de_categorias:{$sum:1}}\n}]).pretty()
```

→ En esta primera consulta he utilizado los operadores:

-**\$group**: Agrupa los documentos de entrada por la `_id` expresión especificada y para cada agrupación distinta, genera un documento. El `_id` campo de cada documento de salida contiene el grupo único por valor. Los documentos de salida también pueden contener campos calculados que contienen los valores de alguna expresión de acumulador.

-**\$sum**: Devuelve una suma de valores numéricos. Ignora los valores no numéricos.

```
db.drogueria.aggregate([{\n  $group:{_id:"$articulo", cantidadDinero:{$sum:{$multiply:["$precioBase", "$unidades"]}}}\n}]).pretty()
```

→ En la segunda consulta he utilizado:

-**\$multiply**: Multiplica números y devuelve el resultado. Pase los argumentos a `$multiply` en una matriz.

```
db.drogueria.aggregate([\n  {\n    $group:{\n      _id: "$articulo",\n      dineroRecaudado:{$sum:{$multiply:["$precioBase", "$unidades"]}},\n      dineroGastado:{$sum:{$multiply:["$precioUnidad", "$unidades"]}}\n    },\n    {\n      $project:{\n        beneficios:{$subtract:["$dineroRecaudado", "$dineroGastado"] }\n      }\n    }\n  }\n] ).pretty()
```

→ En la siguiente consulta aparece:

-**\$subtract**: Resta los valores especificados dentro de la matriz y devuelve un nuevo campo con el resultado.

```
db.drogueria.aggregate( [
  { $match: { articulo: "Lejía" } },
  { $group: { _id: "$articulo", cantidadUnidades: { $avg: "$unidades" },
  } }
] ).pretty()
```

→ En esta consulta he utilizado:

-**\$match:** Filtra los documentos para pasar solo los documentos que coinciden con las condiciones especificadas a la siguiente etapa de canalización.

-**\$avg:** Devuelve el valor medio de los valores numéricos, ignora los valores no numéricos.

```
db.drogueria.aggregate([
  {$group: { _id: "$articulo", cantidadUnidades: { $sum: "$unidades" } }},
  {$project: {unidadesGrandesEmpresas: {$divide: ["$cantidadUnidades", 5]}}}
]).pretty()
```

→ En la siguiente consulta he tratado con:

-**\$project:** Pasa los documentos con los campos solicitados a la siguiente etapa del proceso. Los campos especificados pueden ser campos existentes de los documentos de entrada o campos recién calculados.

-**\$divide:** Divide un número por otro y devuelve el resultado. Pase los argumentos a \$divide en una matriz.

```
db.drogueria.aggregate([
  {
    $group: { _id: "$articulo",
      totalDinero: { $max: { $multiply: ["$precioBase", "$unidades"] } },
      maximaCantidad: { $max: "$unidades" }
    }
  }
])
```

→ En una de las ultimas consulta de los operadores utilizados en clase he utilizado:

-**\$max:** Devuelve el valor máximo, compara tanto el valor como el tipo, utilizando el orden de comparación BSON especificado para valores de diferentes tipos.

```

db.drogueria.aggregate(
[
  { $match: { $expr: { $gt: [{ $year: "$fechaRepuesto" }, 2021] } } },
  {
    $group:
    {
      _id: { $year: "$fechaRepuesto" },
      unidadesRepuestas: { $sum: { $multiply: [1.5, "$unidades"] } }
    }
  }
]
).pretty()

```

- En esta ultima consulta de los operadores dados en clase he utilizado:
- \$expr**: Permite el uso de expresiones de agregación dentro del lenguaje de consulta.
  - \$gt**: selecciona aquellos documentos donde el valor del campo que es mayor que el valor especificado
  - \$year**: Devuelve la parte del año de una fecha.

## NUEVOS OPERADORES

```

db.drogueria.aggregate([
  $addFields:{mesesOferta:["febrero","marzo","septiembre","diciembre"]}
]).pretty()

```

- Este es uno de los nuevos operadores que he integrado en el proyecto:
- \$addFields**: Agrega nuevos campos a los documentos, documentos de salida que contienen todos los campos existentes de los documentos de entrada y los campos recién agregados.

```

db.drogueria.aggregate([
  $project:{
    articulo:1,
    masde3500Unidades:{$gt:["$unidades",3500]},
    _id:0
  }
])

```

- Este es el operador ya visto con el .find() pero con una función distinta:
- \$gt**: compara 2 valores y devuelve el resultado en forma de boolean.