



UTN

# MÉTODOS DE AGREGACIÓN

POR GABRIEL ALBERINI



# Agregación en MongoDB

La agregación en MongoDB se usa para procesar y transformar datos en una serie de etapas de manera flexible y dinámica, como el procesamiento en serie de un pipeline.

En MongoDB, un **pipeline** es una serie de etapas que se ejecutan en secuencia para procesar datos. Cada etapa toma los datos de entrada, realiza una operación y transfiere los resultados a la siguiente etapa.



# Etapas:

- **\$match**: Filtra documentos con base en ciertos criterios, similar a un WHERE en SQL.
- **\$group**: Agrupa documentos por un campo específico y aplica operaciones de agregación como sum, avg, max, min, push, y addToSet.
- **\$project**: Selecciona o excluye campos específicos en los documentos, útil para mostrar solo los campos necesarios.
- **\$sort**: Ordena los documentos por un campo en orden ascendente o descendente.
- **\$limit** y **\$skip**: Limita el número de documentos en el resultado o ignora un número específico.
- **\$lookup**: Une documentos de dos colecciones distintas, similar a un JOIN en SQL.
- **\$unwind**: Descompone un campo de arreglo en múltiples documentos individuales.

# Sintaxis:

```
[
{
  $match: { campo: criterio }
},
{
  $group: {
    _id: "$campoParaAgrupar",
    campoResultado: { operacion: "$campo" }
  }
},
{
  $project: {
    campo1: 1,
    campo2: 1,
    campoExcluido: 0
  }
},
{
  $sort: { campo: 1 } // 1 para ascendente, -1 para descendente
},
{
  $limit: numeroDeDocumentos
},
{
  $skip: numeroDeDocumentos
},
{
  $lookup: {
    from: "coleccionRelacionada",
    localField: "campoLocal",
    foreignField: "campoExtranjero",
    as: "nombreDelCampoUnido"
  }
},
{
  $unwind: "$campoArreglo"
}
```

]

# Estructura de query dinámica:

Imagina que tienes la siguiente tarea: multiplica el precio y el stock de cada producto y calcula la suma total.

La estructura del documento es:

```
{  
  _id: 5,  
  nombre: 'Mesa',  
  categoria: 'Muebles',  
  precio: 300,  
  stock: 100  
}
```

# Estructura de query dinámica:

La misma query se podría realizar de las siguientes maneras:

Opción 1:

```
db.productos.aggregate([
  { $project: { precioTotal: { $multiply: ["$precio", "$stock"] } } },
  { $group: { _id: null, precioTotalInventario: { $sum: "$precioTotal" } } },
  {$project: {_id: 0}}
])
```

Opción 2:

```
db.productos.aggregate([
  { $addFields: { precioTotal: { $multiply: ["$precio", "$stock"] } } },
  { $group: { _id: null, precioTotalInventario: { $sum: "$precioTotal" } } },
  { $project: { _id: 0, precioTotalInventario: 1 } }
])
```

### Opción 3:

```
db.productos.aggregate([
  {
    $group: {
      _id: null,
      precioTotalInventario: {
        $sum: { $multiply: ["$precio", "$stock"] }
      }
    }
  },
  { $project: { _id: 0, precioTotalInventario: 1 } }
])
```

# Persistencia de resultados:

**\$out:** Almacena el resultado en la colección nuevoInventario.

```
db.productos.aggregate([
  {
    $project: {
      precioTotal: { $multiply: ["$precio", "$stock"] }
    }
  },
  {
    $group: {
      _id: null,
      precioTotalInventario: { $sum: "$precioTotal" }
    }
  },
  {
    $project: {
      _id: 0,
      precioTotalInventario: 1
    }
  },
  {
    $addFields: {
      fechaRegistro: new Date()
    }
  },
  {
    $out: "nuevoInventario"
  }
])
```



# Persistencia de resultados:

**\$merge:** Agrega output en collección existente.

```
[
  {
    $merge: {
      into: "nuevoInventario",
    }
  }
]
```



# ¡MUCHAS GRACIAS!

<https://linkedin.com/in/gabriel-alberini>

