

1ºDAM

Proyecto 2º trimestre - Aggregations



Índice del manual

1.) Estructura de la base de datos..... 2

 1.1) Trabajadores 2

 1.2) Secciones..... 2

 1.3) Muebles..... 3

2.) Operadores usados: 4

 \$group: 4

 \$match: 4

 \$project: 4

 \$lookup: 4

 \$cond:..... 4

 \$sort: 4

 \$limit: 4

 \$unwind: 4

 \$arrayElemAt:..... 4

 \$addFields: 4

 \$push..... 4

1.) Estructura de la base de datos

Para esta base de datos he creado el concepto de "Ikeo", una empresa diseñadora de muebles que lleva a cabo sus propios muebles con sus respectivas ventas.

Dentro de esta empresa hay 5 secciones en las que trabajan distintos trabajadores; y estas secciones se encargan de diseñar muebles para áreas específicas de una vivienda.

Para esta base de datos he creado 3 colecciones:

1.1) Trabajadores

La colección trabajadores tiene la siguiente estructura:

```
{
  _id: "91653725J",
  nombre: "Hernan",
  apellidos: "Avila Roman",
  seccion: "cocina",
  añosAntigüedad: 3,
  sanciones: 0
}
```

_id: Es un campo alfanumérico que indica el DNI del trabajador. Al ser un atributo único, lo cojo de ID.

nombre: Es un campo alfanumérico que indica el nombre del trabajador.

apellidos: Es un campo alfanumérico que indica los apellidos del trabajador.

seccion: Es un campo alfanumérico que indica la sección a la que pertenece el trabajador.

añosAntigüedad: Es un campo numérico que indica los años que lleva en la empresa el trabajador.

sanciones: Es un campo numérico que indica cuantas sanciones lleva acumuladas el trabajador.

1.2) Secciones

```
{
  _id: "cocina",
  sueldoMensual: 1100,
  muebles: ["KNOXHULT", "TAGGIG", "HUTTEN", "IVAR", "TORNVIKEN"]
}
```

_id: Es un campo alfanumérico que indica el nombre de la sección.

sueldoMensual: es un campo numérico que indica cual es el salario mensual de los trabajadores pertenecientes al departamento.

muebles: es un Array de valores alfanuméricos que indica cuáles son los muebles creados por el departamento.

1.3) Muebles

```
{
  _id: "KNOXHULT",
  categoria: "Armario",
  precioBase: 80,
  fechaSalida: new Date("2021-01-12"),
  unidadesVendidas: 234,
  propiedades: {
    disponible:true,
    color: "blanco"
  }
}
```

_id: Es un campo alfanumérico que indica el nombre del mueble.

categoria: Es un campo alfanumérico que indica que tipo de mueble es.

precioBase: Es un campo numérico que indica el precio del producto.

fehcaSalida: Es un campo de tipo fecha que indica cuando se lanzó el producto.

unidadesVendidas: Es un campo numérico que indica cuantas unidades del producto se han vendido en total.

disponible: Es un campo booleano que indica si hay stock del producto en los almacenes.

propiedades: Es un documento con los campos disponibles y color.

2.) Operadores usados:

\$group: <https://docs.mongodb.com/manual/reference/operator/aggregation/group/>

Agrupar los documentos que le indiquemos por el que elijamos como `_id`, y por cada agrupación distinta, crea un documento.

\$match: <https://docs.mongodb.com/manual/reference/operator/aggregation/match/>

Filtrar los documentos para dejar pasar solo los documentos que coincidan con las condiciones especificadas.

\$project: <https://docs.mongodb.com/manual/reference/operator/aggregation/project/>

Pasa los documentos con los campos que le digamos a la siguiente etapa. Los campos especificados pueden ser campos existentes del documento o campos recién calculados en esta misma etapa.

\$lookup: <https://docs.mongodb.com/manual/reference/operator/aggregation/lookup/>

Permite unir 2 colecciones. Se elige una clave primaria y una clave foránea, y crea un campo nuevo en la colección base cuyo valor es un Array formado por los datos de la otra colección.

\$cond: <https://docs.mongodb.com/manual/reference/operator/aggregation/cond/>

Actúa como si fuese un condicional, un "if". Se le da una condición, si la cumple le podemos indicar que haga X cosa, y en caso de que no, que haga otra X cosa.

\$sort: <https://docs.mongodb.com/manual/reference/operator/aggregation/sort/>

Ordena los documentos que le digamos de manera ascendente o descendente.

\$limit: <https://docs.mongodb.com/manual/reference/operator/aggregation/limit/>

Limita los resultados de una consulta. Si lo usamos después de un \$sort, es una gran opción para quedarse con los mayores o los menores resultados.

\$unwind: <https://docs.mongodb.com/manual/reference/operator/aggregation/unwind/>

Rompe un Array de documentos para generar un documento para cada elemento. Perfecto para tratar con los documentos que nos devuelve el \$unwind.

\$arrayElemAt:

<https://docs.mongodb.com/manual/reference/operator/aggregation/arrayElemAt/>

Devuelve el elemento de un array de la posición que le indiquemos. Perfecto para cuando el \$lookup nos devuelve solo 1 array de documentos. Esta situación se da cuando por ejemplo, un mueble solo puede pertenecer a una sección. Si un mueble pudiese pertenecer a varias secciones, se usaría el \$unwind.

\$addFields: <https://docs.mongodb.com/manual/reference/operator/aggregation/addFields/>

Agrega nuevos campos a los documentos.

\$push: <https://docs.mongodb.com/manual/reference/operator/update/push/>

Agrega los valores que indiquemos a un Array.