María Prados

Proyecto 2 Trimestre

Para este nuevo proyecto hemos usado como colección base una combinación de las dos colecciones hechas en los trabajos anteriores, con una estructura más complicada que luego nos permita dividirla en varias colecciones.

La colección principal esta formada por 10 entradas,
Las cuales tienen un total de 19 campos base, contando con un solo producto.
De estos 19, aproximadamente 5 son tipo String, 5 de tipo numérico, 1 de tipo fecha, 4 de tipo booleanos y varios arrays de documentos.

Estructura de la colección.

```
db.pedidospizza.drop();
db.pedidospizza.insertMany([
    {
        id:01,
        Venta:[
            {
                Producto:"Pizza Jamonera",
                Tamaño: "Mediana",
                PrecioVenta: 10.90,
                PrecioCoste:3.50,
                Unidades:2,
                Extras: {
                        extra_queso: true,
                        masa_fina: false,
                        salsa_texmex: false,
                        bordes_rellenos: true
                    }
            },
            {
                Producto: "Cerveza",
                Tamaño: "33cl",
                PrecioVenta: 2,
                PrecioCoste: 0.50,
                Unidades: 1,
                Extras: {
                    sin_alcohol: false
                }
            },
            {
                Producto: "Alitas de pollo",
                Tamaño: "4u",
                PrecioVenta: 3.50,
                PrecioCoste: 1.60,
                Unidades: 1,
                Extras: {
                        salsa: "barbacoa"
```

```
                }
            }
        ],

        IVA: 0.21,
        Cliente:[
                {
                    Nombre:"Antonio"
                },
                {
                    Tlf: "584123958"
                }
        ],
        Vendedor: "Luis Santos",
        FechaVenta: new Date("2022-01-10"),
        FormaEntrega: "Recoger"
    },
```

Querys

Con esta colección principal he realizado varias búsquedas, filtrados y operaciones, para demostrar que aunque tiene una estructura compleja (trabaja dentro del array).
Es completamente funcional.

Filtrados simples con el operador $match

```
db.pedidospizza.aggregate([
    {$match: {Vendedor: "Estrella"}}
])
```

```
db.pedidospizza.aggregate([
    {$match: {FormaEntrega: "Local"}}
])
```

```
db.pedidospizza.aggregate([
    {$match: {FechaVenta: {"$gt": new Date("2022-01-31"), "$lt": new
Date("2022-02-28")}}}
])
```

Sacar datos del array "venta" con el operador $unwind y agrupar con $group.

```
db.pedidospizza.aggregate([

    {$unwind: "$Venta"},
    {$group: {_id:"$Venta.Producto"}}])
```

Operaciones matemáticas

```
/*Precio Medio por articulo en un pedido (por unidad)*/
db.pedidospizza.aggregate([

            {$unwind: "$Venta"},
            {$group: {_id:"$id",
            TotalxUnidadProducto: {$sum: "$Venta.PrecioVenta"},
            PrecioMedioArticulo: {$avg: "$Venta.PrecioVenta"}}}])
```

```
/*Calcula el precio total sin iva, del pedido 5*/
db.pedidospizza.aggregate([
        {$unwind: "$Venta"},
        {$match: {"id":05}},
        {$group: {_id: "$Producto",
        TotalxPedido: {$sum: {$multiply: ["$Venta.PrecioVenta",
"$Venta.Unidades"]}}}}])
```

División de las colecciones

He dividido mi colección principal en 2, una por clientes y otra por productos.
Utilizando el operado **$unwind** para extraer los datos del array y el operador **$out** para darle salida.

```
/*Separar colecciones*/

db.pedidospizza.aggregate([
    {$unwind:"$venta"},
    {$project:{"id":1,"venta":1}},
    {$out:{db:"test",coll:"productospizza"}}
])
```

```
db.pedidospizza.aggregate([
    {$unwind: "$Cliente"},
    {$project: {_id:0, "Cliente": 1, "Vendedor": 1, "FechaVenta": 1,
"FormaEntrega": 1}},
    {$out:{db:"test", coll:"clientespizza"}}
])
```

A pesar de haber usado el mismo comando para las dos el resultado del $unwind ha sido diferente.

Así es como se muestran las colecciones en mongo compass.

La colección productos, nos ha mantenido el array creando dentro campos tipo objeto, por lo tanto para operar debemos seguir usando el $unwind:

Sin embargo la coleccion clientes si que nos ha quitado el array, pero nos ha creado un insert diferente para cada campo del array, (nombre y tlf), de forma que hemos perdido los telefonos de nuestros clientes, Al estar separados en diferente _id, aunque siguen conservando vendedor y fecha.

Operador **$lookup**

Une dos colecciones llamando a un campo desde otra colección.
En mi caso no puede ser, porque ese campo id no llama a los articulos correspondientes, ya que la coleccion clientes no tiene ningun campo en comun con la coleccion productos.

```
   db.clientespizza.aggregate([
                {
                    $lookup: {
                    from: 'productospizza',
                    localField: 'id',
                    foreignField: 'id',
                    as: 'venta'
                   }
                },
                {
                    $addFields: {
                    Preciototal: {$multiply: ["$PrecioVenta",
"$unidades"]},
                         {$arrayElemAt: ["$venta", 0]}}}])
```