Adrià García Pons
1º ASIR: Bases de Datos no relacionales

Proyecto Mongo

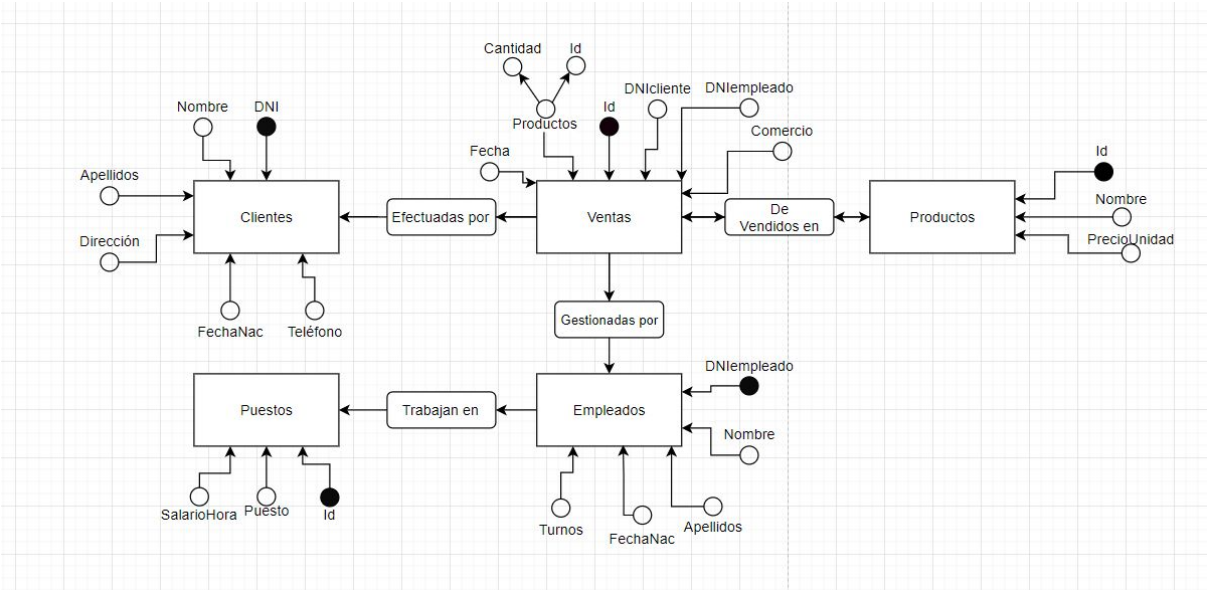
Índice:

1. Explicación General del Proyecto:

La base de datos creada para el proyecto está **diseñada para lo que podría ser un supermercado.**

Mantiene un registro sobre los **Clientes, Ventas, Empleados, Productos** y sobre los **Puestos** en los que trabajarán los empleados.

Diagrama de las relaciones de la base de datos:



Los **recuadros** equivalen a las distintas **colecciones** de nuestra base de datos.
Los **rectángulos** representan las **relaciones** entre estas colecciones.
Los **círculos** son los distintos **campos** que forman los distintos campos siendo los **círculos negros** los **campos _id**

2. Justificación sobre la elección de los distintos campos y elecciones:

Clientes:

- **Campo _Id:** DNI, al ser un elemento único que comparte toda persona que pueda llegar a ser Cliente es perfecto para ser un campo _Id por su naturaleza única.

- **Nombre, Apellidos, Dirección, FechaNac, Teléfono**: Estos campos por desgracia en mi proyecto en particular no se les saca demasiado provecho ya que su propósito en toda base de datos está ligado a una utilización de la base de datos de forma constante, profesional y como herramienta para gestionar la empresa. Estos campos son de tipo **Strings** a excepción de **Teléfono**.

Ventas:

- **Campo _Id:** En este caso a diferencia con la colección de Clientes este campo es un campo _Id artificial, es decir, no se forma a partir de una información ajena a la base de datos. Una alternativa para esto podría ser hacer un campo _Id mediante la Fecha y hora y el DNICliente por ejemplo, pero para una mayor comodidad a la hora de trabajar he optado por esta opción.

- **DNICliente, DNIEmpleado, Productos_Id:** Estos campos se corresponden a los campos clave de las colecciones Clientes, Empleados y Productos respectivamente. Nos permitirán crear las relaciones mediante los **$lookup**

- **Productos(Id,Cantidad)**, **Fecha, Comercio**: Estos campos propios de la colección ventas son de tipo Documento tipo anidado o Documento tipo objeto (**Productos**), de tipo fecha (**Fecha**), Int (**Cantidad**) y String (**Comercio**).

- **Productos:** El campo Productos lo he elaborado de una forma que creo que no hemos visto en clase y la cual me ha permitido relacionar directamente 2 campos en forma de Array, en este caso _Id producto y cantidad en la que se compra este producto.

```
Productos: [{ _id: 1, cantidad: 3 }, { _id: 3, cantidad: 4 }, { _id: 13, cantidad: 5 }],
```

Productos:

- **Campo _Id:** En este caso a diferencia con la colección de Clientes este campo es un campo _Id artificial.

- **Nombre, PrecioUnidad:** Estos campos son de tipo String (**Nombre**) e Int **(PrecioUnidad**), nos permitirán calcular el precio de nuestras ventas relacionandolos con el campo Productos de la colección Ventas.

Empleados:

- **Campo _Id:** DNI, al ser un elemento único que comparte toda persona que pueda llegar a ser Cliente es perfecto para ser un campo _Id por su naturaleza única.
- **Nombre, Apellidos, FechaNac**: Estos campos son de tipo String (**Nombre y Apellidos**), y fecha (**FechaNac**).
- **Turnos**: Este campo es una Array que contiene las horas en las que el trabajador ha trabajado. Existen 4 puestos en los que todo trabajador del Supermercado podrá trabajar a lo largo de la semana, según la posición que ocupen las horas dentro del Array significará que son las horas de trabajo correspondientes a un determinado puesto.

  Ej: Posición 1 → Puesto 1, Posición 3 → Puesto 3

  ```
  HorasSemanales: [5, 2, 1, 3] }
  ```

  En uno de los aggregate lo relacionaremos a los distintos puestos utilizando un **$unwind** con Índice.

Puestos:

- **Campo _Id:** El campo _Id es artificial y nos permitirá relacionarlo al índice que creemos mediante el operador **$unwind** en el campo **HorasSemanales** de la colección **Empleados.**
- **Puesto, SalarioHora:** Estos campos son de tipo String (**Puesto**) e Int (**SalarioHora**).

3. Explicación de los Aggregate.

El propósito del Aggregate en sí viene en el archivo aggregates.js

Primer Aggregate:

```
db.Ventas.aggregate([
    {
        $unwind:
        {
            path: "$Productos"
        }
    },
    {
        $lookup:
        {
            from: "Productos",
            localField: "Productos._id",
            foreignField: "_id",
            as: "Items"
        }
    },
    { $unwind: "$Items" },
    {
        $project:
        {
            _id: 1,
            Precio: { $multiply: ["$Productos.cantidad", "$Items.PrecioUnidad"] },
            DNICliente: 1,
            Comercio: 1,
            Productos: 1,
            Items: 1
        }
    },
    {
        $group:
        {
            _id: "$_id",
            PrecioBruto: { $sum: { $multiply: ["$Precio", 1] } },
            PrecioIva: { $sum: { $multiply: ["$Precio", 1.21] } },
        }
    },
    {
        $project:
        {
            "PrecioBruto":1,
            PrecioIva: { $round: ["$PrecioIva", 2] }
        }
    }
]).pretty();
```

Hace un primer filtrado mediante el **$match** utilizando el operador **$regex** filtrando todos los nombres de Clientes que comiencen por A.

```
$match: {
    Nombre: { $regex: "^A" }
}
```

Luego realiza un **$project** modificando los nombres de los campos que ya teníamos originalmente (Ya que el $match es una etapa que ni agrega ni elimina campos), vemos el operador **$year** el cual nos devuelve el año de la fecha que le pasemos.

```
$project: {
    _id: 0,
    Nombre: "$Nombre",
    DNI: "$_id",
    Residencia: "$Provincia",
    AñoNacimiento: { $year: "$FechaNac" }
}
```

Realiza un **$lookup** que relaciona nuestra colección local (**Clientes**) con **Ventas** mediante los campos **DNI** y **DNICliente** respectivamente.

La array resultante de la relación la denominaremos “Compras”

```
$lookup:
{
    from: "Ventas",
    localField: "DNI",
    foreignField: "DNICliente",
    as: "Compras"
}
```

Vuelve a realizar un **$project** esta vez para descartar todos los campos que no queremos (en su mayoría campos que han sido insertados al relacionar las colecciones)

```
$project:
{
    "Residencia": 0,
    "Compras._id": 0,
    "Compras.DNICliente": 0,
    "Compras.DNIempleado": 0,
    "Compras.Comercio": 0,
    "Compras.Fecha": 0,
    "Compras.Productos.Cantidad": 0
}
```

Volvemos a realizar un **$lookup** relacionando esta vez nuestra colección local (**Clientes**) con **Productos** mediante los campos **Compras.Productos._id** y **_id** respectivamente.
La array resultante de la relación la denominaremos “Producto”

```
$lookup:
{
    from: "Productos",
    localField: "Compras.Productos._id",
    foreignField: "_id",
    as: "Producto"
}
```

Resultado (No completo para evitar Spam de Capturas):


```
{
        "Nombre" : "Antonio",
        "DNI" : "3545453A",
        "AñoNacimiento" : 1970,
        "Compras" : [ ],
        "Producto" : [ ]
}
{
        "Nombre" : "Alfredo",
        "DNI" : "32425432M",
        "AñoNacimiento" : 1994,
        "Compras" : [
                {
                        "Productos" : [
                                {
                                        "_id" : 2,
                                        "cantidad" : 1
                                },
                                {
                                        "_id" : 10,
                                        "cantidad" : 3
                                },
                                {
                                        "_id" : 12,
                                        "cantidad" : 6
                                }
                        ]
                }
        ],
        "Producto" : [
                {
                        "_id" : 2,
                        "Nombre" : "Chorizo",
                        "PrecioUnidad" : 15
                },
                {
                        "_id" : 10,
                        "Nombre" : "Huevos",
                        "PrecioUnidad" : 10
                },
                {
                        "_id" : 12,
                        "Nombre" : "Pepsi",
                        "PrecioUnidad" : 5
                }
        ]
```

Segundo Aggregate:

```
db.Ventas.aggregate([
    {
        $unwind:
        {
            path: "$Productos"
        }
    },
    {
        $lookup:
        {
            from: "Productos",
            localField: "Productos._id",
            foreignField: "_id",
            as: "Items"
        }
    },
    { $unwind: "$Items" },
    {
        $project:
        {
            _id: 1,
            Precio: { $multiply: ["$Productos.cantidad", "$Items.PrecioUnidad"] },
            DNICliente: 1,
            Comercio: 1,
            Productos: 1,
            Items: 1
        }
    },
    {
        $group:
        {
            _id: "$_id",
            PrecioBruto: { $sum: { $multiply: ["$Precio", 1] } },
            PrecioIva: { $sum: { $multiply: ["$Precio", 1.21] } },
        }
    },
```

```
        $project:
        {
            "PrecioBruto":1,
            PrecioIva: { $round: ["$PrecioIva", 2] }
        }
    }
]).pretty();
```

Comenzamos con un **$unwind** el cual es un operador que crea un nuevo documento para cada valor de la array que introduzcamos en Path.

```
$unwind:
{
    path: "$Productos"
}
```

Realizamos un **$lookup** entre nuestra colección local (**Ventas**) y **Productos** mediante los campos **Productos._id** y **_id**.

El array resultante la denominaremos “Items”.

```
$lookup:
{
    from: "Productos",
    localField: "Productos._id",
    foreignField: "_id",
    as: "Items"
}
```

Realizamos un segundo **$unwind**, esta vez a “Items” para poder trabajar en la siguiente etapa con los campos que nos han entrado con el **$lookup**.

```
{ $unwind: "$Items" },
```

Tras esto, realizamos un **$project** para poder filtrar los campos que nos interesen y poder crear ya el campo **Precio** mediante el operador **$multiply** el cual nos permite multiplicar los campos **Productos.cantidad** y **Items.PrecioUnidad**.

```
$project:
{
    _id: 1,
    Precio: { $multiply: ["$Productos.cantidad", "$Items.PrecioUnidad"] },
    DNICliente: 1,
    Comercio: 1,
    Productos: 1,
    Items: 1
}
```

Realizamos un **$group** para ordenar todos nuestros documentos, ya que tras realizar **2 unwind** nuestros documentos poco tenían que ver con los originales.

En esta etapa es importante recordar que deberemos introducir los campos que queramos conservar ya que el resto los perderemos.

En este caso, mediante la función **$sum** sumaremos todos los campos repetidos del Array que demos como parámetro.

```
$group:
{
    _id: "$_id",
    PrecioBruto: { $sum: { $multiply: ["$Precio", 1] } },
    PrecioIva: { $sum: { $multiply: ["$Precio", 1.21] } },
}
```

Realizaremos un **$project** el cual nos permitirá redondear el campo **PrecioIva** mediante el operador **$round**, el cual es exclusivo de la etapa **$project** y es por eso que no lo he podido utilizar en el group.

Le pasamos al round como parámetro el campo y "2" el cual equivale al número de decimales a los que redondeará.

```
$project:
{
    "PrecioBruto":1,
    PrecioIva: { $round: ["$PrecioIva", 2] }
}
```

Finalmente, mediante un **$sort** ordenaremos todos los documentos según el campo _id

```
$sort: {
    _id:1
}
```

Resultado:

```
{ "_id" : 1, "PrecioBruto" : 120, "PrecioIva" : 145.2 }
{ "_id" : 2, "PrecioBruto" : 130, "PrecioIva" : 157.3 }
{ "_id" : 3, "PrecioBruto" : 185, "PrecioIva" : 223.85 }
{ "_id" : 4, "PrecioBruto" : 225, "PrecioIva" : 272.25 }
{ "_id" : 5, "PrecioBruto" : 75, "PrecioIva" : 90.75 }
{ "_id" : 6, "PrecioBruto" : 160, "PrecioIva" : 193.6 }
{ "_id" : 7, "PrecioBruto" : 225, "PrecioIva" : 272.25 }
{ "_id" : 8, "PrecioBruto" : 170, "PrecioIva" : 205.7 }
{ "_id" : 9, "PrecioBruto" : 59, "PrecioIva" : 71.39 }
{ "_id" : 10, "PrecioBruto" : 102, "PrecioIva" : 123.42 }
{ "_id" : 11, "PrecioBruto" : 40, "PrecioIva" : 48.4 }
{ "_id" : 12, "PrecioBruto" : 49, "PrecioIva" : 59.29 }
{ "_id" : 13, "PrecioBruto" : 69, "PrecioIva" : 83.49 }
{ "_id" : 14, "PrecioBruto" : 15, "PrecioIva" : 18.15 }
{ "_id" : 15, "PrecioBruto" : 6, "PrecioIva" : 7.26 }
>
```

Tercer Aggregate:

```
db.Ventas.aggregate([
    {
        $unwind: "$Productos"
    },
    {
        $group: {
            _id: {
                Mes: { $month: "$Fecha" }, Dia: { $dayOfMonth: "$Fecha" }, Año: { $year: "$Fecha" }
            },
            TotalVentaItems: { $sum: "$Productos.cantidad" },
            MaximoNumeroItems: { $max: "$Productos.cantidad" },
            MediaVentaItems: { $avg: "$Productos.cantidad" },
            IdItemMasvendido: { $max: "$Productos._id" }
        }
    },
    {
        $lookup: {
            from: "Productos",
            localField: "IdItemMasvendido",
            foreignField: "_id",
            as: "Item"
        }
    },
    {
        $project: {
            _id: 1,
            "NumeroVentas":1,
            "TotalVentaItems": 1,
            "MaximoNumeroItems": 1,
            MediaVentaItems: { $round: ["$MediaVentaItems", 2] },
            ItemMasVendido: "$Item.Nombre",
        }
    }
```

```
    {
        $sort: {
            _id: 1
        }
    },
])
```

Comenzamos realizando un **$unwind** al campo Productos para poder trabajar con los campos anidados ._Id, .cantidad.

```
{
    $unwind: "$Productos"
},
```

Realizamos un **$group** utilizando como campo **_id** el Mes, Día y Año obtenidos mediante los operadores de fechas $**month, $dayOfMonth y $year** respectivamente.

En los campos utilizaremos operadores como **$max** que de una serie de valores selecciona el más grande, **$avg** que realiza una Media.

```
$group: {
    _id: {
        Mes: { $month: "$Fecha" }, Dia: { $dayOfMonth: "$Fecha" }, Año: { $year: "$Fecha" }
    },
    TotalVentaItems: { $sum: "$Productos.cantidad" },
    MaximoNumeroItems: { $max: "$Productos.cantidad" },
    MediaVentaItems: { $avg: "$Productos.cantidad" },
    IdItemMasvendido: { $max: "$Productos._id" }
}
```

Ahora, relacionamos nuestra colección local (**Ventas**) con **Productos** mediante los campos **IdItemMasvendido** y **_id** respectivamente.

Denominaremos **"Item**" al Array resultante de la relación.

```
$lookup: {
    from: "Productos",
    localField: "IdItemMasvendido",
    foreignField: "_id",
    as: "Item"
}
```

Realizaremos un **$project** para crear nuevos campos y mantener los que queramos.

```
$project: {
    _id: 1,
    "NumeroVentas":1,
    "TotalVentaItems": 1,
    "MaximoNumeroItems": 1,
    MediaVentaItems: { $round: ["$MediaVentaItems", 2] },
    ItemMasVendido: "$Item.Nombre",
}
```

Finalmente realizaremos un **$sort** para ordenar los campos segun el Id

```
{
$sort: {
    _id: 1
}
```

Resultado:


```
{ "_id" : { "Mes" : 2, "Dia" : 25, "Año" : 2020 }, "TotalVentaItems" : 32, "MaximoNumeroItems" : 4, "MediaVentaItems" : 2.91, "ItemMasVendido" : [ "Palomitas" ] }
{ "_id" : { "Mes" : 2, "Dia" : 26, "Año" : 2020 }, "TotalVentaItems" : 20, "MaximoNumeroItems" : 5, "MediaVentaItems" : 3.33, "ItemMasVendido" : [ "Pipas" ] }
{ "_id" : { "Mes" : 2, "Dia" : 27, "Año" : 2020 }, "TotalVentaItems" : 18, "MaximoNumeroItems" : 5, "MediaVentaItems" : 3, "ItemMasVendido" : [ "Nutella" ] }
{ "_id" : { "Mes" : 2, "Dia" : 28, "Año" : 2020 }, "TotalVentaItems" : 35, "MaximoNumeroItems" : 6, "MediaVentaItems" : 3.18, "ItemMasVendido" : [ "Bizcocho" ] }
{ "_id" : { "Mes" : 3, "Dia" : 1, "Año" : 2020 }, "TotalVentaItems" : 13, "MaximoNumeroItems" : 4, "MediaVentaItems" : 3.25, "ItemMasVendido" : [ "Patatas" ] }
{ "_id" : { "Mes" : 3, "Dia" : 2, "Año" : 2020 }, "TotalVentaItems" : 12, "MaximoNumeroItems" : 5, "MediaVentaItems" : 4, "ItemMasVendido" : [ "Patatas" ] }
```

Adrià García Pons
1º ASIR: Bases de Datos no relacionales

Cuarto Aggregate:

(No cabe bien la captura, la he intentado poner pero queda fatal).

Iniciamos con un **$unwind** el cual actuará sobre el campo “HorasSemanales” de nuestra colección **Empleados**. Creará un índice llamado **“Indices”**.

```
$unwind: {
    path: "$HorasSemanales",
    includeArrayIndex: "Indices"
}
```

Realizaremos un **$lookup** relacionando nuestra colección local (**Empleados**) con **Puestos** mediante los campos **Índices** y **_id** respectivamente y llamaremos a la Array resultante “PuestoTrabajo”.

```
$lookup: {
    from: "Puestos",
    localField: "Indices",
    foreignField: "_id",
    as: "PuestoTrabajo"
}
```

Tras esto, mediante un **$project** crearemos una serie de campos y desecharemos los que no queramos.

Utilizaremos para la creación de los campos “Empleado” y “FechaNacimiento” el operador **$concat** el cual nos permite concatenar Strings en una sola.

(La línea de FechaNacimiento y las 2 de abajo en VisualStudio son una sola pero no cabe).

```
        $project: {
            Empleado: {$concat: ["$Nombre", " " ,"$Apellidos"]},
            DNIEmpleado: "$_id",
            HorasTrabajadas:"$HorasSemanales",
            "_id":0,

            FechaNacimiento: {$concat: ["Día ", {$toString:
            {$dayOfMonth: "$FechaNac"}}," del mes ",{$toString:{$month:
            "$FechaNac"}}," del año ",{$toString:{$year:"$FechaNac"}}]},

            SalarioHora:"$PuestoTrabajo.SalarioHora",
        }
```

Realizaremos un **$unwind** al campo “SalarioHora” permitiéndonos así iniciar nuestras operaciones

```
{$unwind: "$SalarioHora"},
```

Tras esto comenzaremos con el **$project** con el que crearemos el campo Salario y mantendremos el resto de campos que queramos.

```
$project: {
    "Empleado":1,
    "DNIEmpleado":1,
    "FechaNacimiento":1,
    Salario: {$avg: {$multiply: ["$HorasTrabajadas", "$SalarioHora"]}},
}
```

Para finalizar, realizaremos un **$group** en el que vemos por primera vez el operador **$first** el cual dentro de “x” elementos de un Array selecciona el primero que recibe (En este caso, como esos campos son de tipo String no podemos realizar un $avg y cómo estos campos no cambian para un mismo empleado podemos permitirnos usar **$first**).

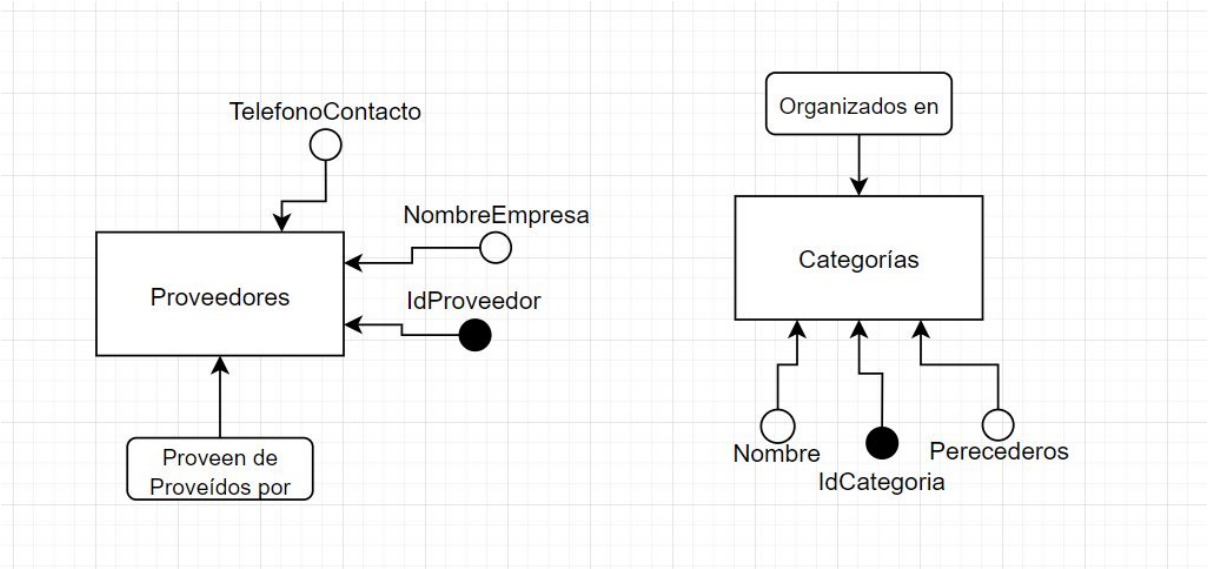
```
$group: {
    _id: "$Empleado",
    DNI: {$first: "$DNIEmpleado"},
    FechaNacimiento: {$first: "$FechaNacimiento"},
    Salario: {$avg: {$sum: ["$Salario"]}}
},
```

Resultado:

```
{ "_id" : "Marco Zapata", "DNI" : "92382398P", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 36.25 }
{ "_id" : "Jorge Serrano", "DNI" : "24393943N", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 21.25 }
{ "_id" : "Rafa Pons", "DNI" : "30402390S", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 25 }
{ "_id" : "Raul Perez", "DNI" : "94394320P", "FechaNacimiento" : "Día 20 del mes 3 del año 1975", "Salario" : 31.25 }
{ "_id" : "Aurelio Sanchez", "DNI" : "90123902A", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 28.75 }
{ "_id" : "Adrian Campos", "DNI" : "20310202B", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 17.5 }
{ "_id" : "Manuel Peña", "DNI" : "23020322S", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 30 }
{ "_id" : "Sofia Curro", "DNI" : "29321990A", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 12.5 }
{ "_id" : "Armando Garcia", "DNI" : "43939432S", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 7.5 }
{ "_id" : "Miguel Romero", "DNI" : "924032402J", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 33.75 }
{ "_id" : "Hector Montañez", "DNI" : "82439319H", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 23.75 }
{ "_id" : "Diego Perez", "DNI" : "34243243B", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 13.75 }
{ "_id" : "Estela Hidalgo", "DNI" : "93493499A", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 22.5 }
{ "_id" : "Ramon Chingueta", "DNI" : "39409030P", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 28.75 }
{ "_id" : "Elena Fuentes", "DNI" : "240394394J", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 23.75 }
{ "_id" : "Juanma Ugarte", "DNI" : "29203191P", "FechaNacimiento" : "Día 10 del mes 4 del año 1970", "Salario" : 35 }
>
```

4. Ideas para mejorar la base de datos para un futuro:

En un principio iba a crear 2 campos extras para la base de datos que creo que hubieran dado mucho juego para la hora de hacer Aggregates. Ya que finalmente no los agregué porque solo tendremos 15 minutos para exponer quería al menos dejarlos aquí plasmados.



Estos 2 campos **Proveedores** y **Categorías** estarían relacionados con **Productos** teniendo este 2 campos adicionales “IdProveedor” e “IdCategoría” para permitir la relación (Así como lo tiene el campo de Ventas).