

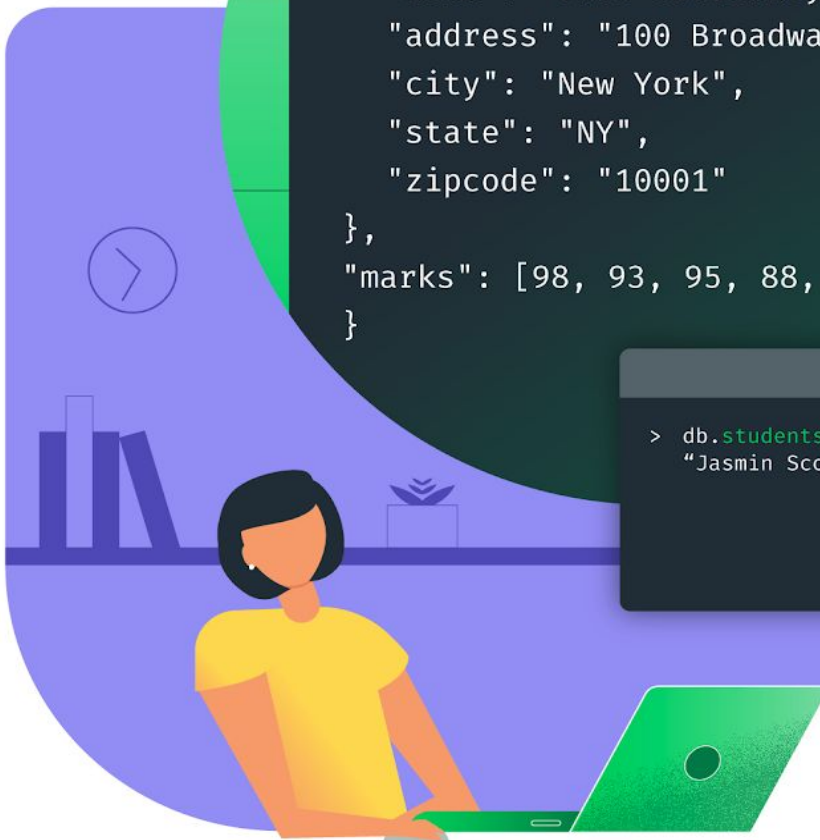
# Proyecto Trimestral 2 Evaluación.

1 DAM Subgrupo 1.



```
{
  "_id": 1,
  "student_name": "Jasmin Scott",
  "school": {
    "school_id": 226,
    "name": "Tech Secondary",
    "address": "100 Broadway St",
    "city": "New York",
    "state": "NY",
    "zipcode": "10001"
  },
  "marks": [98, 93, 95, 88, 100],
}
```

```
mongo
> db.students.find({"student_name":
  "Jasmin Scott"})
```



Adrián Pérez Cobo

# Índice

<b>Explicación de la funcionalidad de la bases de datos</b>	<b>3</b>
<b>Consultas</b>	<b>4</b>

## Explicación de la funcionalidad de la bases de datos

Las base de datos que tenemos tiene la información de una Empresa que crea diferentes aparatos electrónicos y luego los vende a diferentes tiendas grandes para que los vendan en sus tiendas. La primera colección guarda los datos de cada producto: Tenemos el código del producto, el nombre, el precio, el coste, la cantidad disponible en almacén y si la fabricación es manual o no.

```
{codigo: "01", nombre: 'Ordenadores', precio: 100, coste: 50, cantidadAlmacen: 1900, fabricacionmanual: false},
{codigo: "02", nombre: 'Portatiles', precio: 200, coste: 60, cantidadAlmacen: 2210, fabricacionmanual: true},
{codigo: "03", nombre: 'Movil', precio: 80, coste: 40, cantidadAlmacen: 2050, fabricacionmanual: false},
{codigo: "04", nombre: 'Tablet', precio: 85, coste: 50, cantidadAlmacen: 3200, fabricacionmanual: false}
```

y la segunda las ventas al por mayor de productos a tiendas:

El código de la venta, la fecha de la venta, el cliente(la tienda), y luego un campo detalle con los detalles de los productos comprados, con su código y su cantidad.

```
{codigo: '001', fecha: new Date("2019-12-20"), cliente: "Mediamark", detalle: {codigo: "04", cantidad:200}},
{codigo: '002', fecha: new Date("2019-12-30"), cliente: "Mediamark", detalle: {codigo: "03", cantidad:100}},
{codigo: '004', fecha: new Date("2019-12-31"), cliente: "ElCorteIngles", detalle: {codigo: "01", cantidad:300}},
{codigo: '005', fecha: new Date("2020-01-02"), cliente: "ElCorteIngles", detalle: {codigo: "02", cantidad:500}},
{codigo: '006', fecha: new Date("2020-01-05"), cliente: "Mediamark", detalle: {codigo: "04", cantidad:200}},
{codigo: '007', fecha: new Date("2020-01-10"), cliente: "Mediamark", detalle: {codigo: "02", cantidad:300}},
{codigo: '008', fecha: new Date("2020-01-22"), cliente: "ElCorteIngles", detalle: {codigo: "04", cantidad:200}},
{codigo: '009', fecha: new Date("2020-02-22"), cliente: "Worten", detalle: {codigo: "03", cantidad:500}},
{codigo: '010', fecha: new Date("2020-03-02"), cliente: "Fnac", detalle: {codigo: "04", cantidad:100}},
{codigo: '011', fecha: new Date("2020-03-02"), cliente: "Mediamark", detalle: {codigo: "01", cantidad:200}},
{codigo: '012', fecha: new Date("2020-02-22"), cliente: "ElCorteIngles", detalle: {codigo: "02", cantidad:250}},
{codigo: '013', fecha: new Date("2021-01-22"), cliente: "Fnac", detalle: {codigo: "03", cantidad:100}},
{codigo: '014', fecha: new Date("2021-01-22"), cliente: "Worten", detalle: {codigo: "03", cantidad:20}},
{codigo: '015', fecha: new Date("2021-01-22"), cliente: "Fnac", detalle: {codigo: "04", cantidad:200}},
{codigo: '016', fecha: new Date("2021-01-30"), cliente: "Worten", detalle: {codigo: "02", cantidad:50}},
```

## Consultas

### Consulta 1:

Para empezar queremos saber el detalle de las ventas, y mostrar también el precio del artículo y el total de las compras.

Con el lookup unimos la dos colecciones mediante el código de producto. Usamos set para darle el valor al nuevo campo "artículo".

Y hacemos un project para proyectar lo que queremos incluyendo los nuevos campos.

Entre ellos 3 campos de fecha, y un campo para calcular el precio del producto con la cantidad de ese producto comprado.

```
db.ventas.aggregate([
  {
    $lookup: {
      from: "almacen",
      localField: "detalle.codigo",
      foreignField: "codigo",
      as: "productos"
    }
  },
  {
    $set: {
      producto: {$arrayElemAt: ["$productos", 0]}
    }
  },
  {
    $project: {
      _id: 0,
      codigoCompra: "$codigo",
      dia: {$dayOfMonth: "$fecha"},
      mes: { $month: "$fecha" },
      ano: { $year: "$fecha" },
      cliente: 1,
      codigoProducto: "$detalle.codigo",
      Producto: "$producto.nombre",
      cantidadComprada: "$detalle.cantidad",
      precioProducto: "$producto.precio",
      totalCompra: {$multiply: ["$detalle.cantidad", "$producto.precio"]}
    }
  }
])
).pretty()
```

```
{
  "cliente" : "Worten",
  "codigoCompra" : "016",
  "dia" : 30,
  "mes" : 1,
  "ano" : 2021,
  "codigoProducto" : "02",
  "Producto" : "Portátiles",
  "cantidadComprada" : 50,
  "precioProducto" : 200,
  "totalCompra" : 10000
}
```

Ejemplo de salida.

## Consulta 2:

Para la segunda consulta quiero saber cuanto es el beneficio que saca por producto y por venta la compañía.

Para eso primero hacemos un \$unwind para separar el campo detalle, luego usamos un \$lookup para unir las dos colecciones y con el set y el \$arrayElemAt. Y en el project vamos a añadir un campo

```
db.ventas.aggregate([
  {
    $unwind: "$detalle"
  },
  {
    $lookup: {
      from: "almacen",
      localField: "detalle.codigo",
      foreignField: "codigo",
      as: "productos"
    }
  },
  {
    $set: {
      producto: {$arrayElemAt: ["$productos", 0]}
    }
  },
  {
    $project: {
      _id: 0,
      codigoCompra: "$codigo",
      dia: { $dayOfMonth: "$fecha" },
      mes: { $month: "$fecha" },
      ano: { $year: "$fecha" },
      cliente: 1,
      Producto: "$producto.nombre",
      cantidadComprada: "$detalle.cantidad",
      precioProducto: "$producto.precio",
      precioFabricacion: "$producto.coste",
      BeneficioIndividual: { $subtract: [ "$producto.precio", "$producto.coste" ] },
      BeneficioTotalCompra: {
        $multiply: [ {
          $subtract: [ "$producto.precio", "$producto.coste" ] }, "$detalle.cantidad"
        ]
      }
    }
  }
])
```

Ejemplo salida:

```
{
  "cliente" : "Worten",
  "codigoCompra" : "016",
  "dia" : 30,
  "mes" : 1,
  "ano" : 2021,
  "Producto" : "Portatiles",
  "cantidadComprada" : 50,
  "precioProducto" : 200,
  "precioFabricacion" : 60,
  "BeneficioIndividual" : 140,
  "BeneficioTotalCompra" : 7000
}
```

### Consulta 3:

Para la tercera consulta quiero sacar los datos de ventas de fecha después de 2019 y con el precio total de Compra superior a 50000. Mostrar datos generales de la venta y el total.

Para eso primero un match para solo dejar los que sean mayor que 2019.

Un unwind para detalle. Luego hacemos un lookup para unir las dos colecciones y un set.

Luego un group con un id con varios campos y un campo Compra que con un push

hacemos que el campo sea un array de varios campos. En el que se incluye el campo

PrecioTotal el cual luego con un match haremos que sólo salga los que su preciototal sea mayor que 50000.

```
db.ventas.aggregate([
  {
    $match: { fecha: { $gt: new Date("2019-12-31") } }},
  {
    $unwind: "$detalle"
  },
  {
    $lookup: {
      from: "almacen",
      localField: "detalle.codigo",
      foreignField: "codigo",
      as: "productos"
    }
  },
  {
    $set: {
      producto: {$arrayElemAt: ["$productos", 0]}
    }
  },
  {
    $group: {
      _id: {
        codigoCompra: "$codigo",
        dia: { $dayOfMonth: "$fecha" },
        mes: { $month: "$fecha" },
        ano: { $year: "$fecha" },
        Tienda: "$cliente"
      },
      Compra: {
        $push: {
          codigo: "$detalle.codigo",
          producto: "$producto.nombre",
          cantidad: "$detalle.cantidad",
          precio: "$producto.precio",
          PrecioTotal: {$multiply: ["$detalle.cantidad", "$producto.precio"]}
        }
      }
    }
  },
  {
    $match: {
      "Compra.PrecioTotal": {
        $gt: 50000
      }
    }
  }
]).pretty()
```

```
{
  "_id": {
    "codigoCompra": "005",
    "dia": 2,
    "mes": 1,
    "ano": 2020,
    "Tienda": "ElCorteIngles"
  },
  "Compra": [
    {
      "codigo": "02",
      "producto": "Portatiles",
      "cantidad": 500,
      "precio": 200,
      "PrecioTotal": 100000
    }
  ]
}
```

```
{
  "_id": {
    "codigoCompra": "005",
    "dia": 2,
    "mes": 1,
    "ano": 2020,
    "Tienda": "ElCorteIngles"
  },
  "Compra": [
    {
      "codigo": "02",
      "producto": "Portatiles",
      "cantidad": 500,
      "precio": 200,
      "PrecioTotal": 100000
    }
  ]
}
```

Ejemplo de Salida



#### Consulta 4:

Calcular el iva de las ventas de 2021.

Para eso primero un match para solo dejar los que sean mayor que 2020.

Un unwind para detalle. Luego hacemos un lookup para unir las dos colecciones y un set.

Luego un group con un id con varios campos y un campo Compra que con un push hacemos que el campo sea un array de varios campos. Aquí se le añade el campo del IVA.

```
db.ventas.aggregate([
  {
    $match: { fecha: { $gt: new Date("2020-12-31") } }},
  {
    $unwind: "$detalle"
  },
  {
    $lookup: {
      from: "almacen",
      localField: "detalle.codigo",
      foreignField: "codigo",
      as: "productos"
    }
  },
  {
    $set: {
      producto: {$arrayElemAt: ["$productos", 0]}
    }
  },
],
```

```
{
  $group: {
    _id: {
      codigoCompra: "$codigo",
      dia: { $dayOfMonth: "$fecha" },
      mes: { $month: "$fecha" },
      ano: { $year: "$fecha" },
      Tienda: "$cliente"
    },
    Compra: {
      $push: {
        codigo: "$detalle.codigo",
        producto: "$producto.nombre",
        cantidad: "$detalle.cantidad",
        precio: "$producto.precio",
        PrecioTotal: {$multiply: ["$detalle.cantidad", "$producto.precio"]},
        IVA: {
          $multiply: [ {
            $multiply: [ "$producto.precio", "$producto.coste" ] }, 0.12]
        }
      }
    }
  }
},
]).pretty()
```