

Unidad 3 - Tarea 4 - Almacenamiento y Consultas de Datos en Big Data

Nombre del estudiante

Giovanny Alejandro Pardo

Grupo:

Big Data (202016911_27)

Tutora

Sandra Milena Patino Avella

Universidad Nacional Abierta y a Distancia-UNAD

Escuela de ciencias básicas, tecnológicas e ingeniería

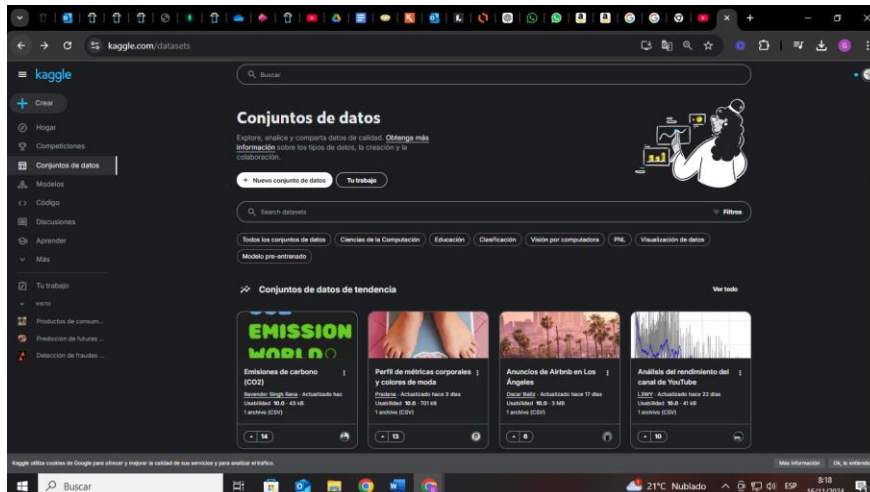
Ingeniería de sistemas

Palmira – noviembre 16 del 2024

Diseño de base de datos

Para el desarrollo de esta actividad nos dirigiremos a la página kaggle para descargar un dataset de información de productos para usarla en la base de datos MongoDB y realizar las respectivas consultas.

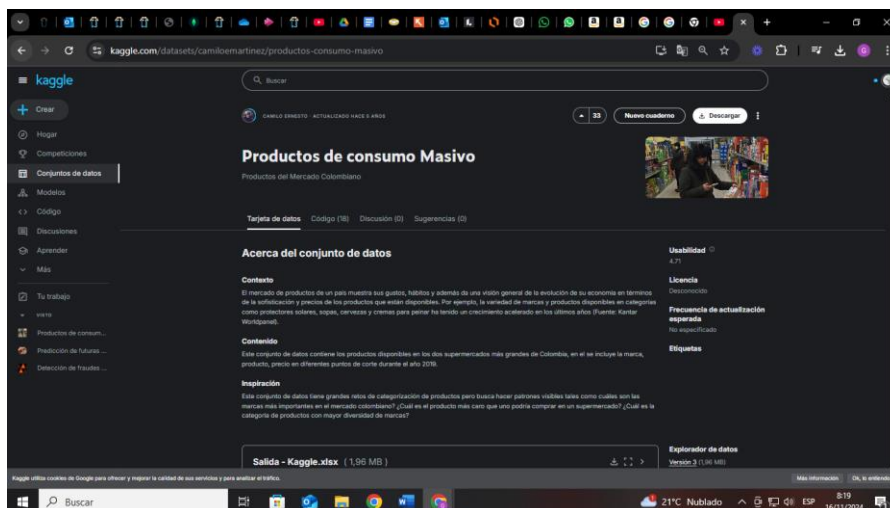
Imagen 01



Nota: En esta imagen se evidencia que estamos en la página oficial de kaggle.

Para nuestro caso usaremos un dataset de productos de consumo masivo el dataset cuenta con 25638 registros y 13 tipos de datos y se llama **output - Kaggle**.

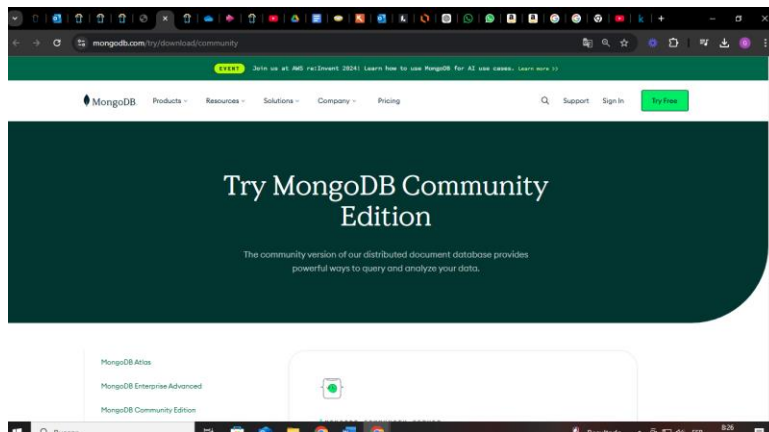
Imagen 02



Nota: En esta imagen se evidencia el dataset a usar.

Ahora con el dataset descargado procedemos a descargar la base de datos MongoDB he instalarla en nuestro equipo.

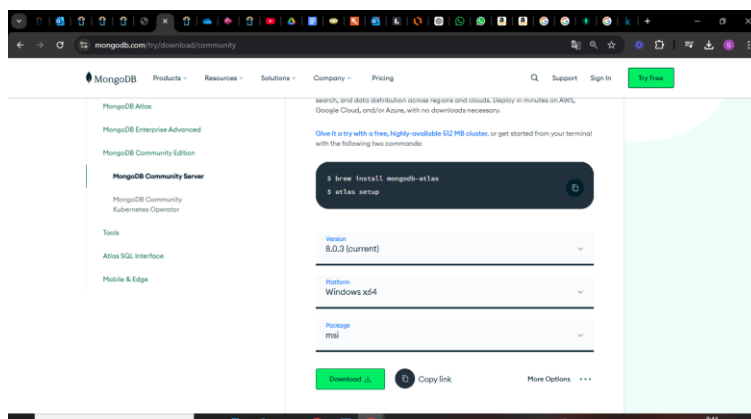
Imagen 03



Nota: En esta imagen se evidencia la página oficial de MongoDB.

Descargamos la versión más reciente en este caso es la 8.0.3 (current) para la plataforma Windows a 64bit.

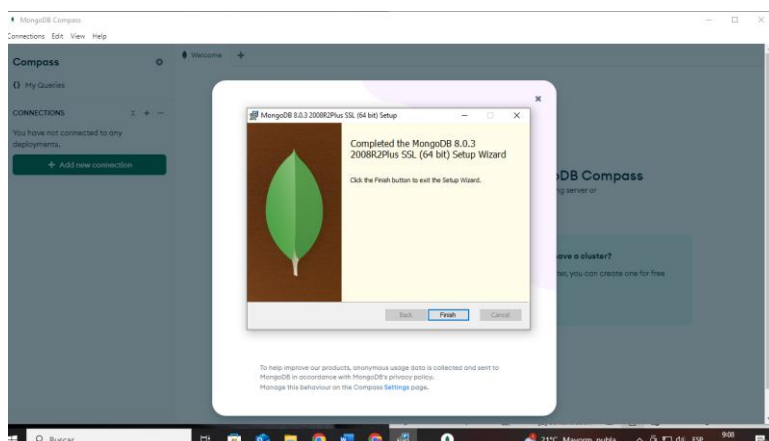
Imagen 04



Nota: En esta imagen se evidencia versión de MongoDB a descargar.

Ahora procedemos a descargar de la base de datos MongoDB Community Server

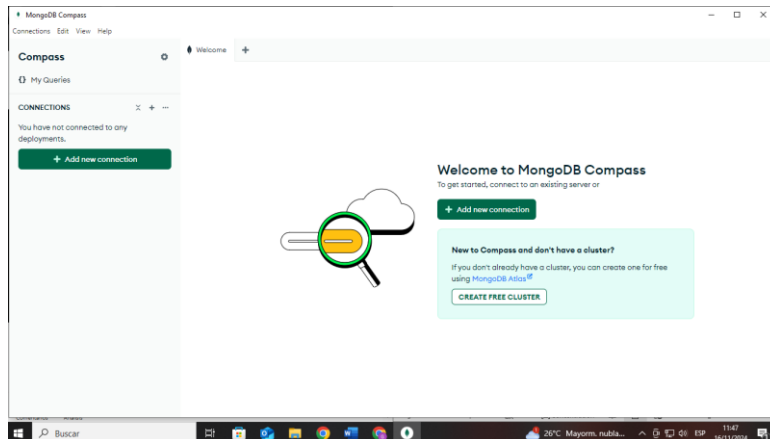
Imagen 05



Nota: En esta imagen se evidencia la instalación de MongoDB versión 8.0.2

Al culminar la instalación automáticamente se nos abre la interfaz de MongoDB

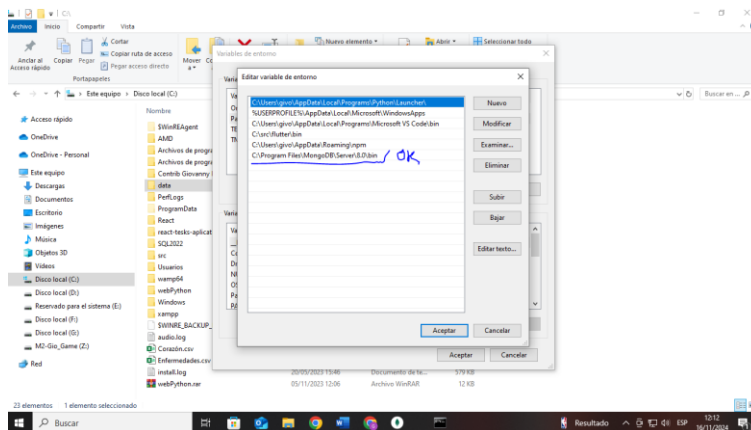
Imagen 06



Nota: En esta imagen se evidencia la interfaz de MongoDB.

Antes de iniciar se debe configurar la variable de entorno para agrega la ruta a la carpeta de Mongo y así poder usar la terminar cmd, luego para que no se presente un error en el servicio debemos crear en C: una carpeta de nombre **base** y dentro de ella otra carpeta de nombre **bd** y si volvemos a nuestra terminal de MongoDB se puede ver que ya temeos nuestra ruta url

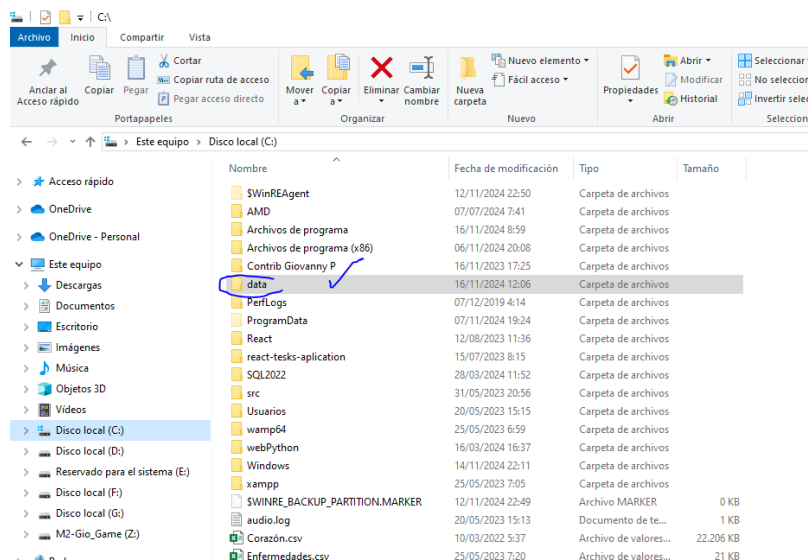
Imagen 07



Nota: En esta imagen se evidencia que se agrego la ruta a la variable de entorno.

Se crea las carpetas data y bd en nuestro disco C:

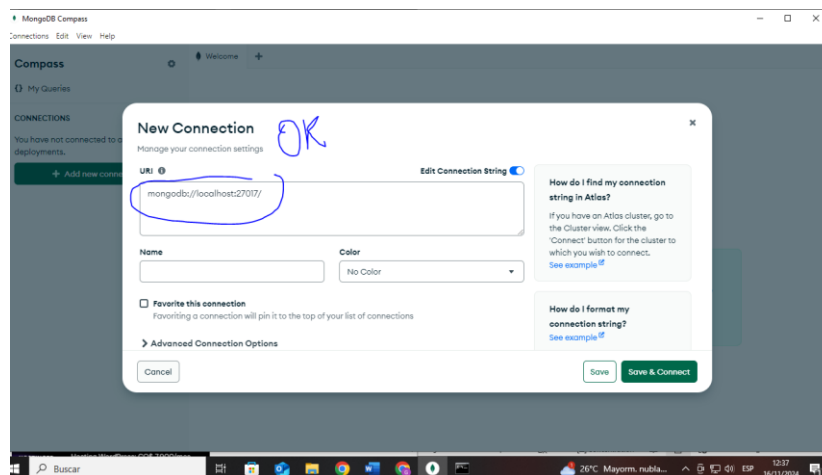
Imagen 08



Nota: En esta imagen se evidencia la creación de las carpetas data y db en la unidad C:

Luego verificamos el entorno de MongoDB para validar la ruta de conexión y solo le daríamos en conectar.

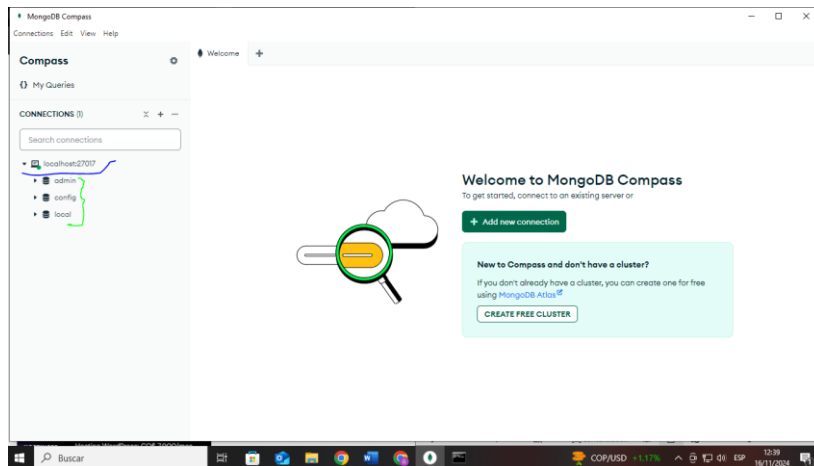
Imagen 09



Nota: En esta imagen se evidencia que tenemos la ruta de conexión.

Al dar guardar y abrir ya estaríamos conectado con la base de datos de MongoDB y se pueden apreciar las 3 bases de datos admin, local y confin.

Imagen 10

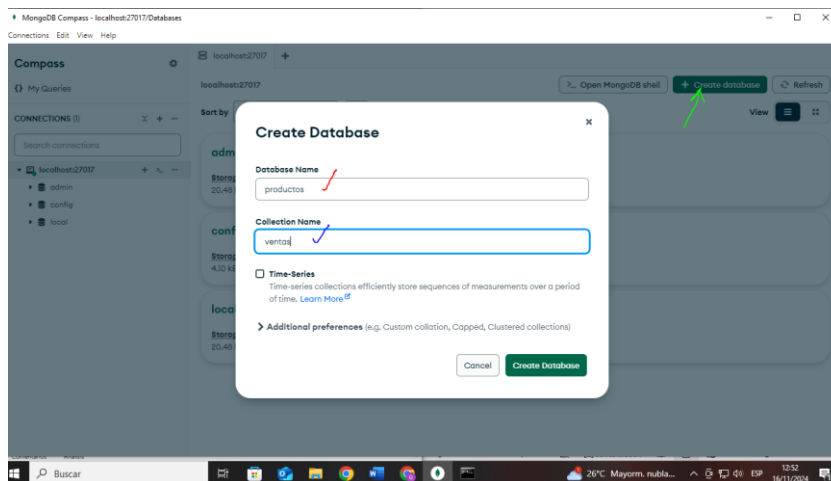


Nota: En esta imagen se la conexión a la base de datos de MongoDB.

Diseño de la base de datos en MongoDB.

Ahora procedemos a crear nuestra colección, para ello le damos al botón que dice **créate database** y le daremos a la base de datos el nombre de **Productos1** y a la colección la nombraremos **Ventas1**.

Imagen 11



Nota: En esta imagen se evidencia la creación de la base de datos y la colección.

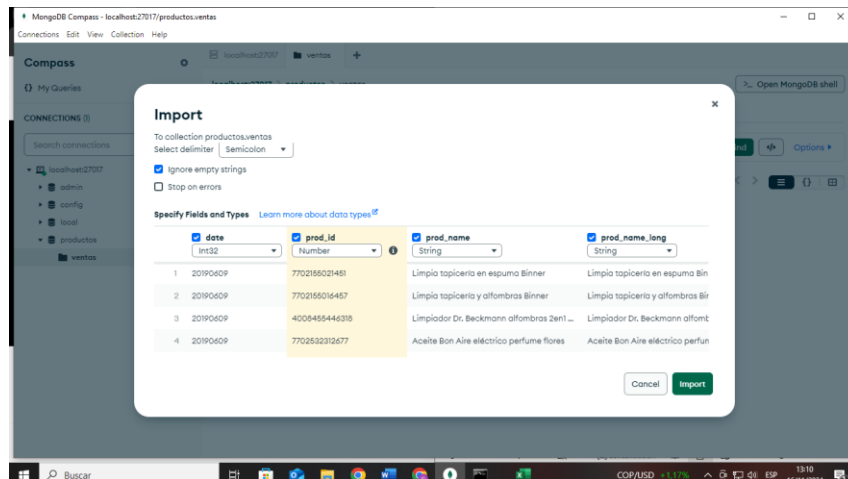
Implementación en MongoDB

- Crear una base de datos en MongoDB e insertar un conjunto de datos de prueba (al menos 100 documentos).

Ahora procedemos a cargar el dataset para insertar en la colección los documentos para ello le damos en el botón **add data** y luego seleccionamos la ruta donde se encuentra nuestro dataset.

Nota: Hay que tener presente que se debe cargar el dataset convertido a .csv, para ello solo es abrirlo en excel y guardarlo como .csv

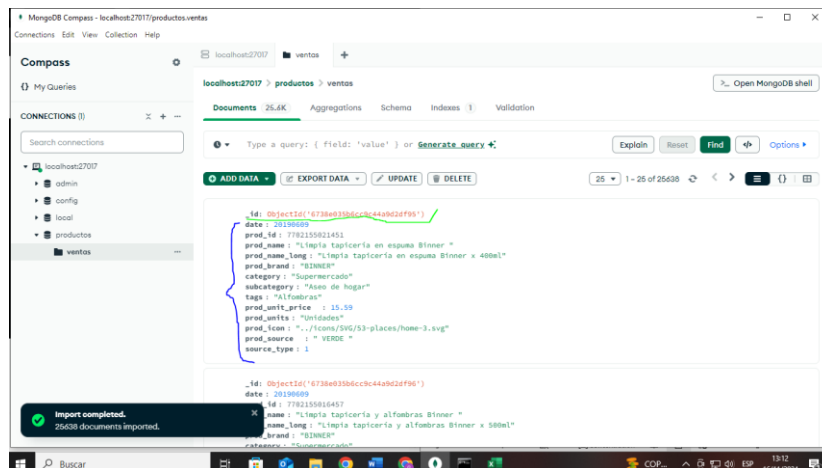
Imagen 12



Nota: En esta imagen se evidencia el cargue del dataset.

Luego de cargado el dataset se nos mostrara los documentos, hay que tener presente que MongoDB genera una ID para cada documento.

Imagen 13

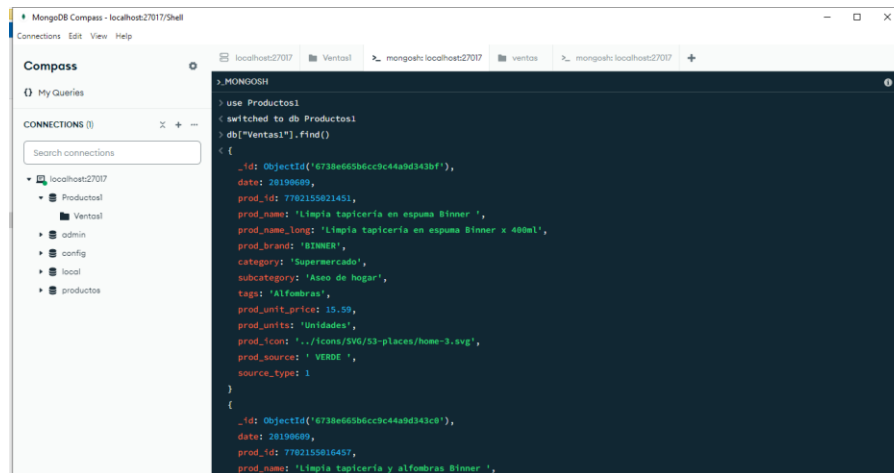


Nota: En esta imagen se evidencia que se cargo exitosamente los documentos en la colección ventas.

Verificamos los documentos con el siguiente comando.

```
Productos1>bd[“Ventas1”].find()
```

Imagen 14



Nota: En esta imagen se evidencia lo que hay en cada documento dentro de la colección.

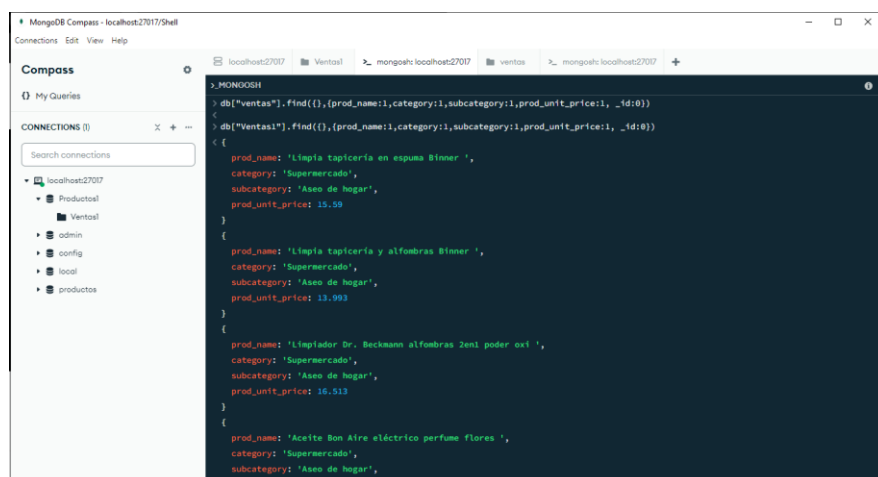
- Implementar consultas utilizando el lenguaje de consulta de MongoDB para realizar las siguientes operaciones:

Consultas en MongoDB con “find”

Ahora procederemos a realizar la consulta para ello usaremos la Shell de MongoDB y ingresaremos la siguiente línea de comando para consultas las columnas **prod_name**, **subcategory** y **prod_unit_price** de la siguiente forma.

Productos1>db["Ventas1"].find({}, {prod_name:1, subcategory:1, prod_unit_price:1, _id:0})

Imagen 14



Nota: En esta imagen se evidencia el resultado de la consulta, donde se nos muestra cada documento con las columnas seleccionadas en la consulta.

Con la consulta el resultado es que nos muestra los documentos donde tendremos el nombre del producto, la categoría, la subcategoría y el precio unitario.

{


```

prod_name: 'Limpia tapicería en espuma Binner ', // Nombre del producto
category: 'Supermercado', //Categoría del producto
subcategory: 'Aseo de hogar', //Subcategoría del producto
prod_unit_price: 15.59 //El precio de la unidad
}

```

Actualizar en MongoDB con “updateOne-updateMany”

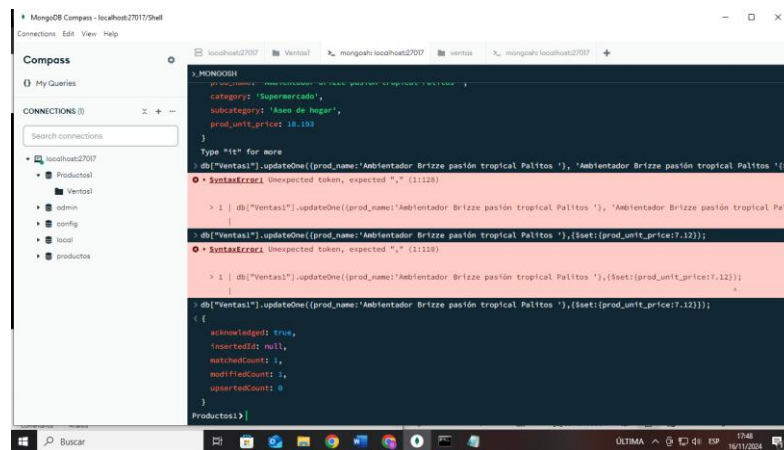
Cuando se realiza la actualización con el comando **updateOne** estaremos modificando solo un documento, mientras que cuando usamos el comando **updateMany** estaremos modificando muchos documentos, ahora procederemos a filtrar un producto y a este se le cambiara el precio unitario.

```

Producto1> db["Ventas1"].updateOne({prod_name:'Ambientador Brizze pasión tropical Palitos '},{ $set:{prod_unit_price:7.12}});

```

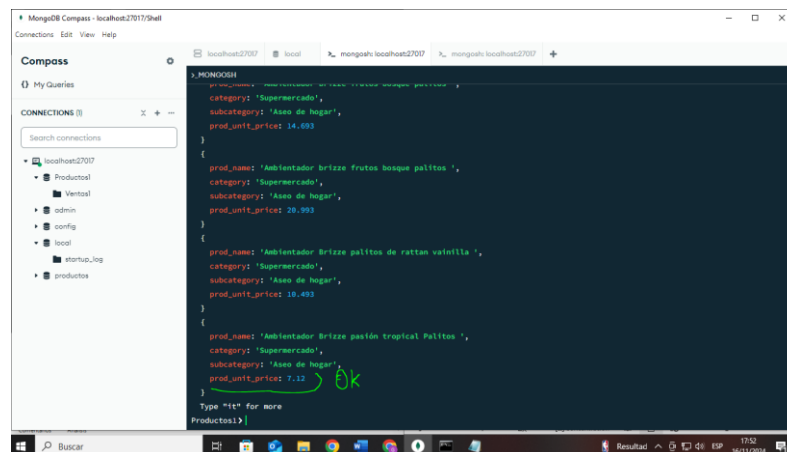
Imagen 15



Nota: En esta imagen se evidencia que se realiza la inserción de manera exitosa

Ahora realizamos la consulta para ver el cambio en el documento que cumplido con el filtro y la respectiva modificación de su precio unitario.

Imagen 16



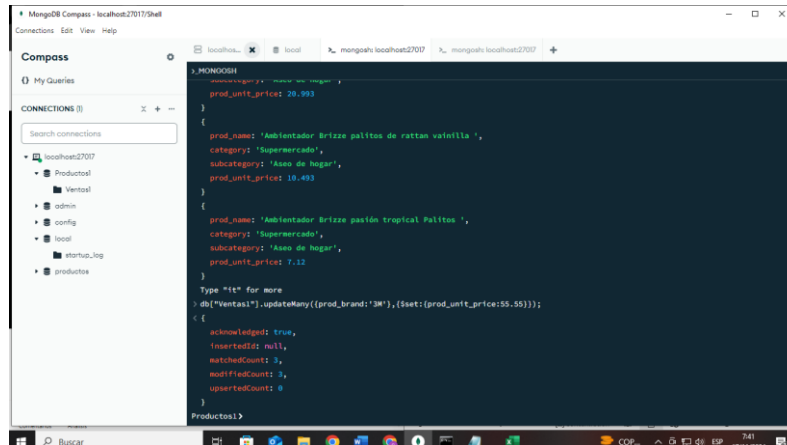
Nota: En esta imagen se evidencia que si se actualizo el documento.

Ahora usaremos el comando `updateMany` para actualizar varios documentos de la colección mediante la siguiente consulta.

```
Producto1>db["Ventas1"].updateMany({prod_brand:'3M'},{$set:{prod_unit_price:55.55}});
```

Con esta consulta se filtra por la marca “3M” y todos los documentos que cumplan se les modificara su precio unitario a 55.55

Imagen 17

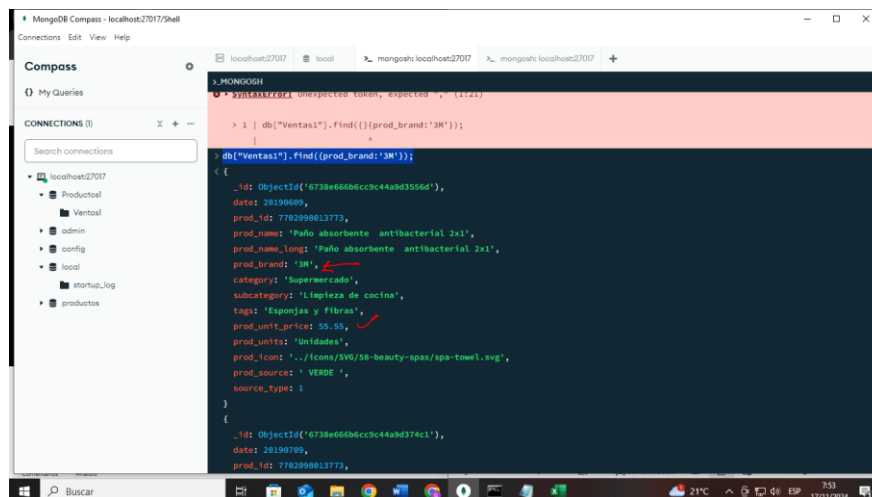


Nota: En esta imagen se evidencia que la operación se realiza a tres documentos satisfactoriamente

Ahora procedemos a consultar los documentos modificados.

```
Producto1> db["Ventas1"].find({prod_brand:'3M'});
```

Imagen 18



Nota: En esta imagen se evidencia que la actualización a los tres documentos se les modifico el precio unitario.

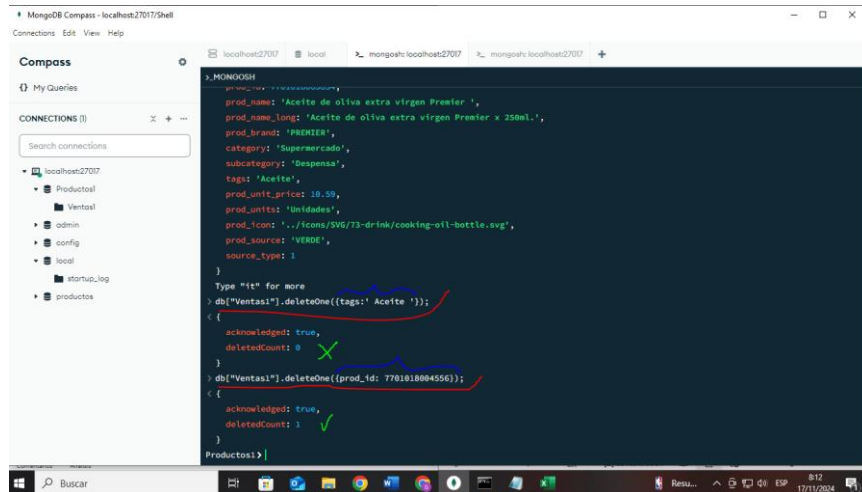
Consultas en MongoDB con “deleteOne y deleteMany”

Así como la actualización, la eliminación se puede realizar por un documento o por varios documentos, para estas consultas vamos a eliminar un documento que

cumpla la condición de filtrado e igual para la eliminación de varios documentos que cumplan con el filtrado.

```
Producto1> db["Ventas1"].deleteOne({prod_id: 7701018004556});
```

Imagen 19

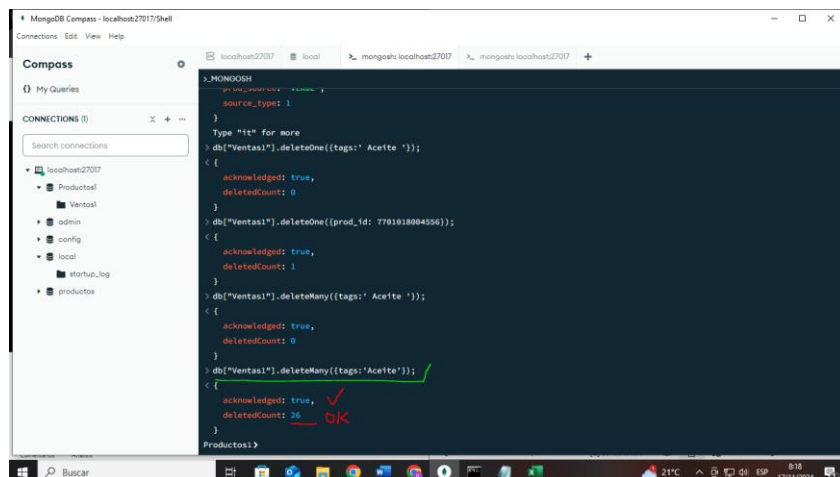


Nota: En esta imagen se evidencia que se realizaron dos eliminaciones de las cuales una tuvo éxito porque definido el filtro a un único documento, mientras con el filtro de “Aceite” no arrojo resultados porque eran muchos documentos

Para la eliminación de varios documentos usaremos el filtro de “Aceite” que en no funciona para eliminar un solo registro.

```
Producto1> db["Ventas1"].deleteMany({tags:' Aceite '});
```

Imagen 20



Nota: En esta imagen se evidencia la eliminación de 26 documentos que la familia de producto era “Aceite”

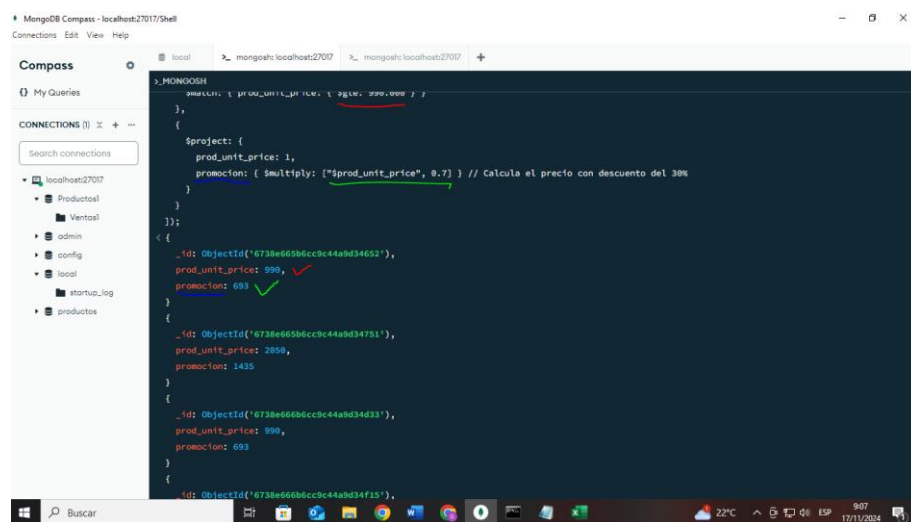
Consultas básicas de filtrado y operaciones

Consultas en MongoDB con “aggregate”

Ahora procederemos a buscar los productos que tenga un valor especificado y a estos productos les daremos una promoción de descuento del 30% se creara una nueva columna para agregar el valor con el descuento.

```
db.Ventas1.aggregate([  
  
  {  
  
    $match: { prod_unit_price: { $gte: 990.000 } }  
  
  },  
  
  {  
  
    $project: {  
  
      prod_unit_price: 1,  
  
      promocion: { $multiply: ["$prod_unit_price", 0.7] } // Calcula el precio con  
descuento del 30%  
  
    }  
  
  }  
  
]);
```

Imagen 21



Nota: En esta imagen se evidencia que se creo una nueva columna donde esta el valor del precio de promoción que equivale a 30% del valor unitario.

También podemos realizar la operación estableciendo un rango de valores usado la condicional \$and donde se buscarán dentro de los documentos los productos que tengan un valor mayor a 30 y menor a 50 y le calcule el valor de descuento de 70%.

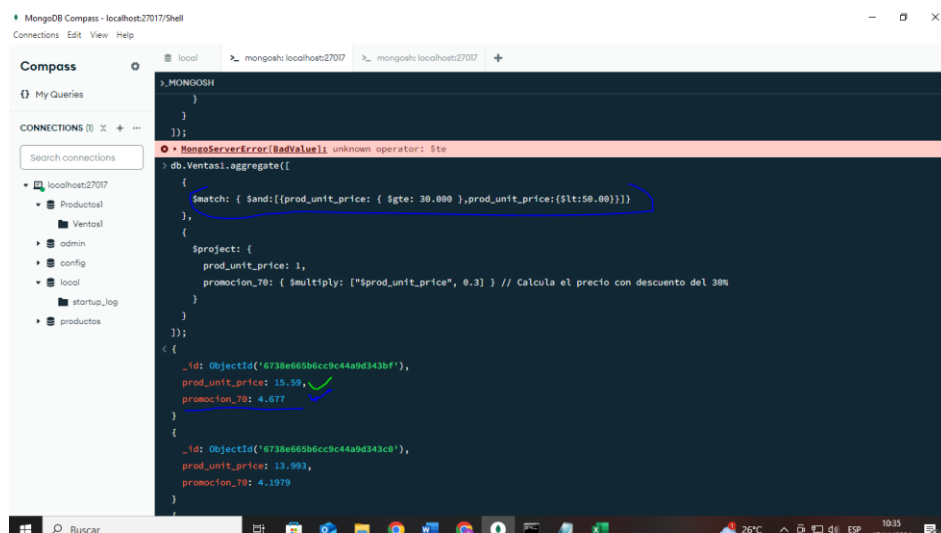
```
db.Ventas1.aggregate([  
  
  {  
  
    $match: [$and:{ { prod_unit_price: { $gte: 30 }, prod_unit_price: { $lt: 50 } } ]
```

```

},
{
  $project: {
    prod_unit_price: 1,
    promocion: { $multiply: ["$prod_unit_price", 0.7] } // Calcula el precio con
descuento del 30%
  }
}
});

```

Imagen 22



Nota: En esta imagen se evidencia el calculo de 70% de promoción para productos entre el rango de mayor de 30 y menor de 50 en una nueva columna.

Debemos tener en cuenta que existen otros operadores útiles como son:

- \$lt (menor que)
- \$gte (mayor o igual que)
- \$in (en un conjunto de valores)
- \$or (consulta con condiciones múltiples)

Consultas en MongoDB con “insertOne”

Ahora agregaremos un nuevo documento mediante la inserción en la colección.

```

db.Ventas1.insertOne({
  date:20241117,
  prod_id: 753159852456,
  prod_nane: "Laptop",
  peod_name_long:"Portatil Gamer Pro5000",
  prod_brand:"Electronica",

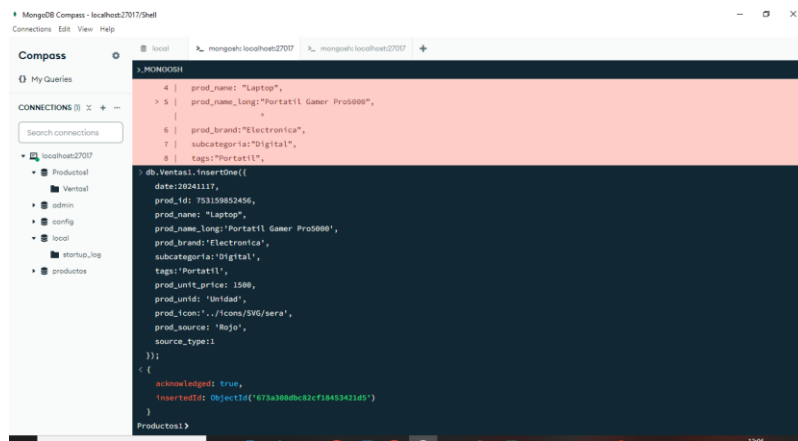
```

```

subcategoria:"Digital",
tags:"Portatil",
prod_unit_price: 1500,
prod_unid: "Unidad",
prod_icon:"../icons/SVG/sera",
prod_source: "Rojo",
source_type:1
});

```

Imagen 23



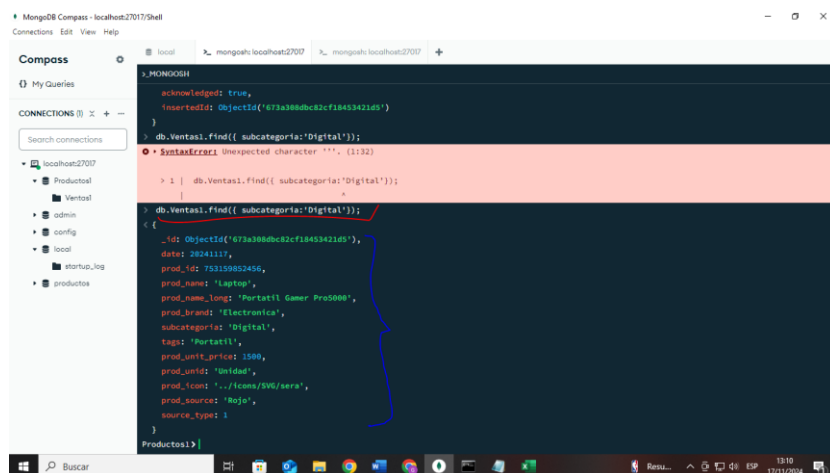
Nota: En esta imagen se evidencia que se realizó la inserción exitosamente.

Consultas en MongoDB con “find”

Ahora usaremos la consulta para mostrar el documento insertado.

```
Productos1> db.Ventas1.find({ subcategoria:'Digital'});
```

Imagen 24



Nota: En esta imagen se evidencia que existe el documento insertado anteriormente.

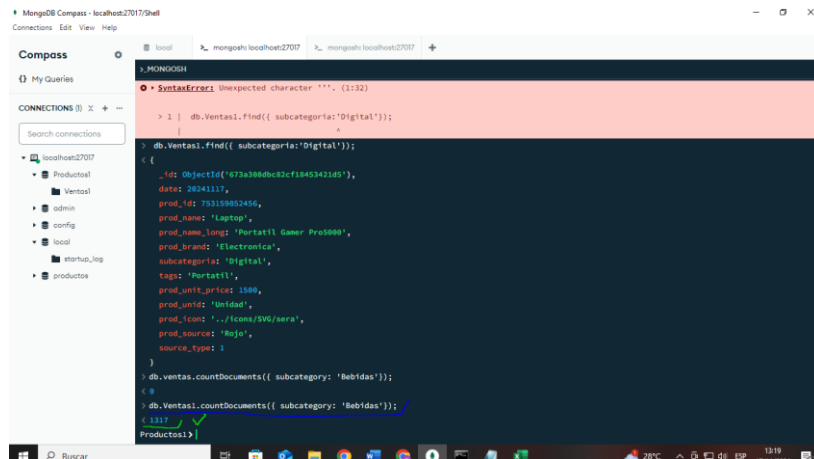
Consultas de Agregación para Calcular Estadísticas

Consultas en MongoDB con “countDocuments”

Ahora vamos a contar los productos en la categoría “Bebidas”

```
Productos1> db.ventas.countDocuments({ subcategory: 'Bebidas' });
```

Imagen 25



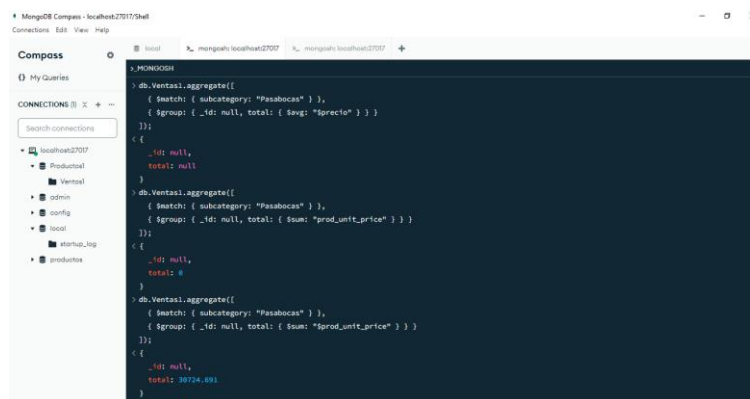
Nota: En esta imagen se evidencia que tenemos 1317 documentos con la subcategoría Bebidas.

Consultas en MongoDB con “sumar valores”

Ahora realizamos una consulta y agregaremos el total del precio de todos los documentos que cumplan con el filtro que la subcategoría sea igual a pasabocas.

```
db.Ventas1.aggregate([
  { $match: { subcategory: "Pasabocas" } },
  { $group: { _id: null, total: { $avg: "$prod_unit_price" } } }
]);
```

Imagen 26



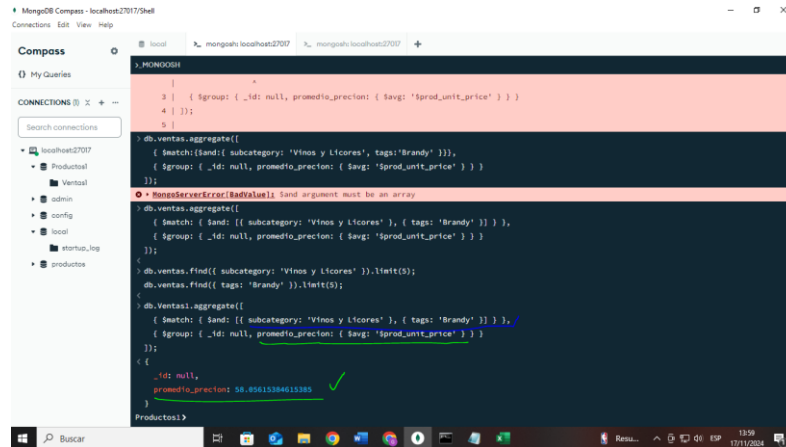
Nota: En esta imagen se evidencia que hay 30724.691 en la suma de los documentos que cumple con el filtro categoría igual a pasabocas.

Consultas en MongoDB con “promediar valores”

Para calcular el precio promedio de los productos con la subcategoría “Vinos y Licores”

```
db.Ventas1.aggregate([  
  
  { $match: { $and: [{ subcategory: 'Vinos y Licores' }, { tags: 'Brandy' }] } },  
  
  { $group: { _id: null, promedio_precio: { $avg: '$prod_unit_price' } } }  
  
]);
```

Imagen 27



Nota: En esta imagen se evidencia que el promedio de los documentos que cumple con el filtro subcategoría vino y licores y tipo brandy es de 58.05