**Informe Técnico**

**Análisis de Datos de Ventas utilizando MongoDB y Python**

**1. Introducción**

En este proyecto, se desarrolló una solución práctica para analizar datos de ventas utilizando MongoDB y Python. Se seleccionó la temática de "Análisis de Ventas" debido a su relevancia en la toma de decisiones empresariales.

El objetivo principal fue diseñar una base de datos, poblarla con información realista y desarrollar consultas avanzadas mediante pipelines de agregación, permitiendo obtener información valiosa como ingresos por producto, ventas por categoría, y el producto más vendido.

**2. Diseño de la Base de Datos**

La base de datos consta de dos colecciones principales:

* **Productos**: Contiene información sobre los productos en venta, como su nombre, categoría y precio.
* **Ventas**: Registra las transacciones realizadas, asociando cada venta con un producto mediante su producto\_id.

**Estructura de las colecciones:**

**Colección**productos**:**

| **Campo** | **Tipo** | **Descripción** |
| --- | --- | --- |
| \_id | Integer | Identificador único del producto. |
| nombre | String | Nombre del producto. |
| categoria | String | Categoría a la que pertenece el producto. |
| precio | Decimal | Precio del producto. |

**Colección**ventas**:**

| **Campo** | **Tipo** | **Descripción** |
| --- | --- | --- |
| \_id | Integer | Identificador único de la venta. |
| producto\_id | Integer | Relación con el \_id de la colección productos. |
| cantidad | Integer | Cantidad de productos vendidos. |
| fecha | Date | Fecha de la transacción. |

Esta estructura garantiza la integridad y la facilidad de realizar operaciones complejas mediante $lookup.

**3. Desarrollo del Trabajo**

**Conexión a MongoDB**

Se utilizó la librería pymongo para conectar Python con MongoDB. Se crearon funciones para poblar la base de datos con datos realistas y ejecutar consultas con pipelines de agregación.

**Consultas Realizadas**

1. **Consulta 1: Total de ingresos por producto**
   * **Propósito**: Calcular el ingreso generado por cada producto, multiplicando su precio por la cantidad vendida.
   * **Pipeline**:

pipeline = [

{"$lookup": {"from": "productos", "localField": "producto\_id", "foreignField": "\_id", "as": "producto"}},

{"$unwind": "$producto"},

{"$group": {"\_id": "$producto.nombre", "total\_ingresos": {"$sum": {"$multiply": ["$cantidad", "$producto.precio"]}}}},

{"$sort": {"total\_ingresos": -1}}

]

* + **Resultado**:
    - Laptop: $2,400
    - Auriculares: $250
    - Zapatillas: $240

1. **Consulta 2: Ventas totales por categoría**
   * **Propósito**: Determinar el número total de productos vendidos por categoría.
   * **Pipeline**:

pipeline = [

{"$lookup": {"from": "productos", "localField": "producto\_id", "foreignField": "\_id", "as": "producto"}},

{"$unwind": "$producto"},

{"$group": {"\_id": "$producto.categoria", "total\_ventas": {"$sum": "$cantidad"}}},

{"$sort": {"total\_ventas": -1}}

]

* + **Resultado**:
    - Electrónica: 8
    - Ropa: 5
    - Calzado: 4

1. **Consulta 3: Producto más vendido**
   * **Propósito**: Identificar el producto con mayor cantidad de ventas.
   * **Pipeline**:

pipeline = [

{"$lookup": {"from": "productos", "localField": "producto\_id", "foreignField": "\_id", "as": "producto"}},

{"$unwind": "$producto"},

{"$group": {"\_id": "$producto.nombre", "total\_vendido": {"$sum": "$cantidad"}}},

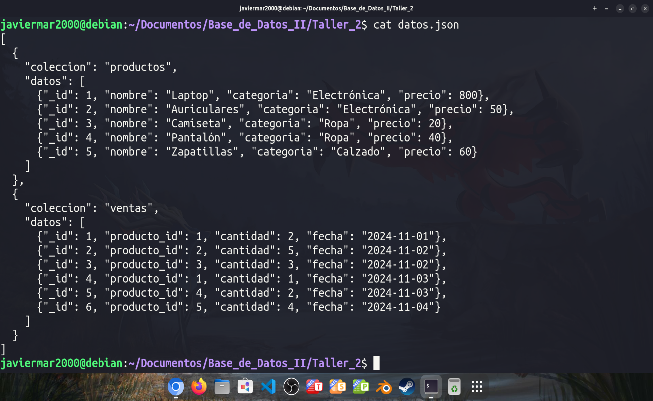
{"$sort": {"total\_vendido": -1}},

{"$limit": 1}

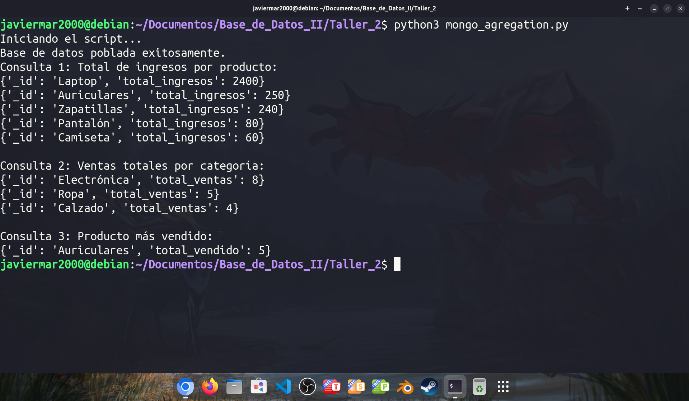
]

* + **Resultado**: Auriculares (5 unidades).

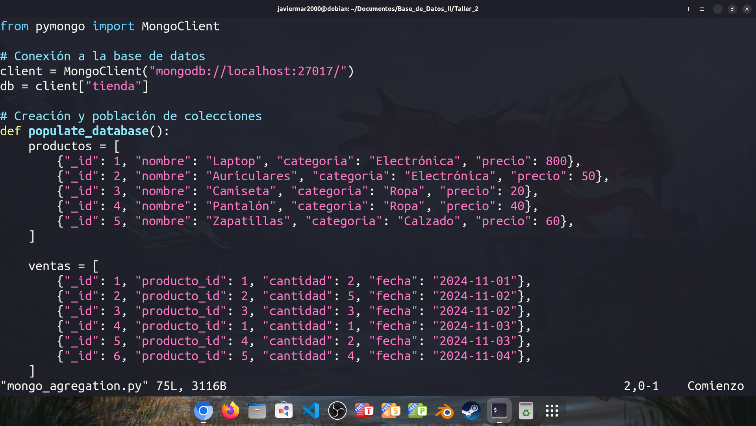
**Capturas de Pantalla:**

****

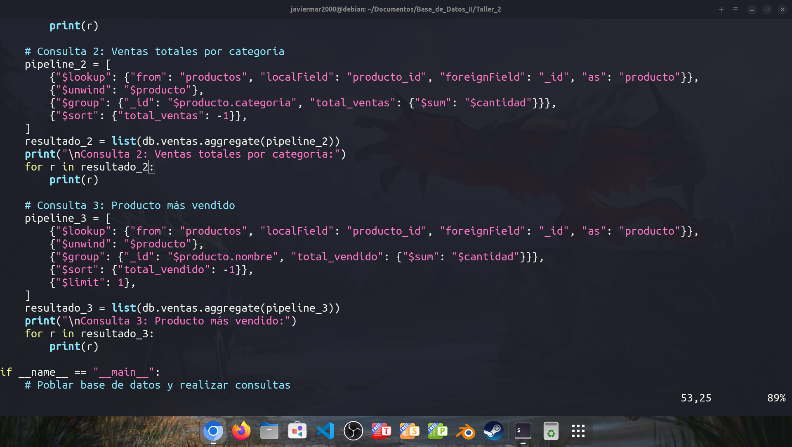
*Captura 1: “Estructura de los datos iniciales en formato JSON, organizados en las colecciones productos y ventas, listos para ser importados a MongoDB.”*

****

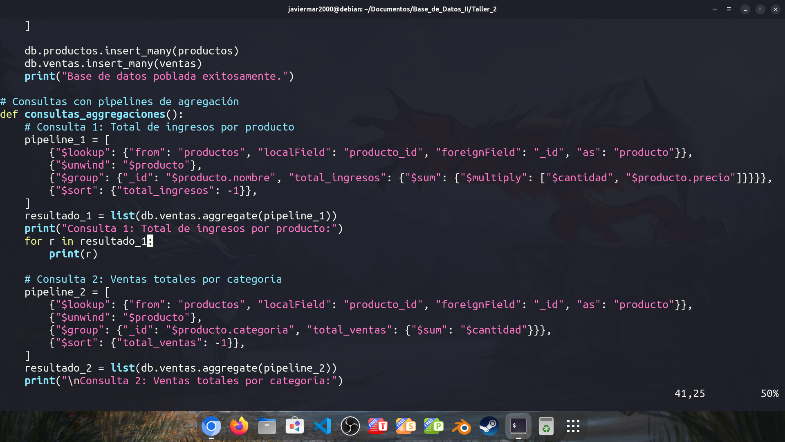
*Captura 2: "Resultados obtenidos al ejecutar el script mongo\_agregation.py, mostrando el total de ingresos por producto, ventas por categoría y el producto más vendido."*

****

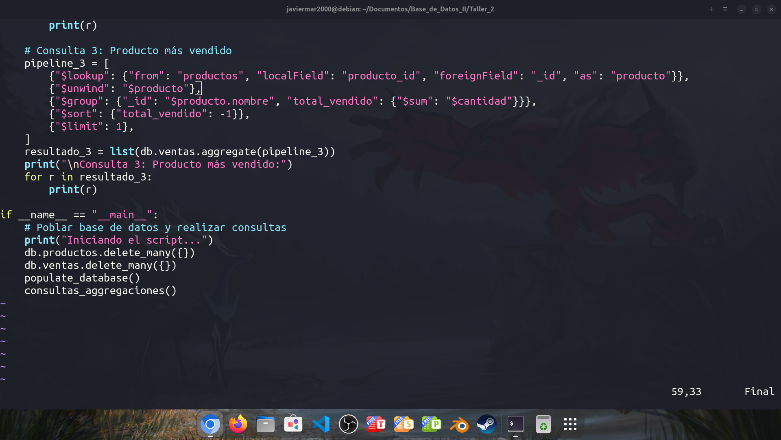
*Captura 3: "Fragmento del script mongo\_agregation.py que conecta Python con MongoDB y pobla la base de datos con las colecciones productos y ventas."*

****

*Captura 4: "Definición del pipeline para calcular el total de ingresos por producto, utilizando operadores avanzados como $lookup, $unwind, $group y $sort para realizar operaciones de agregación sobre las colecciones productos y ventas."*

****

*Captura 5: "Definición del pipeline para calcular las ventas totales por categoría, aprovechando operadores como $lookup, $unwind, $group y $sort para agrupar los datos por categorías y sumar las cantidades vendidas."*

****

*Captura 6: "Definición del pipeline para identificar el producto más vendido, utilizando operadores como $lookup, $unwind, $group, $sort y $limit para encontrar el producto con mayor cantidad de ventas."*

**4. Resultados y Análisis**

Los resultados obtenidos a partir de las consultas muestran tendencias importantes:

* El producto más vendido son los auriculares, posiblemente debido a su precio accesible.
* La categoría "Electrónica" genera más ingresos y tiene mayor número de ventas.
* Productos con precios elevados como la Laptop contribuyen significativamente a los ingresos totales.

Este análisis podría ser utilizado para optimizar el inventario, priorizar promociones y maximizar las ganancias.

**5. Conclusiones**

Este proyecto permitió explorar la integración de Python con MongoDB y desarrollar habilidades en el uso de pipelines de agregación. Los principales aprendizajes incluyen:

* La eficacia de $lookup para unir colecciones y enriquecer datos.
* La importancia de las operaciones de $group y $project para realizar cálculos avanzados.
* Cómo un diseño adecuado de la base de datos facilita el análisis de datos.

**Desafíos superados**:

* Configuración inicial de MongoDB.
* Ajuste de pipelines para obtener resultados precisos.

**Futuras mejoras**:

* Incorporar visualizaciones interactivas para facilitar la interpretación de los datos.
* Ampliar las consultas para incluir análisis de tendencias temporales.

**6. Anexos**

**Código Python Completo:**

****

**Archivo JSON:**

