eCommerce Big Data.

Contenido

[Recursos útiles 1](#_Toc203979268)

[Objetivo de la práctica 2](#_Toc203979269)

[Funcionalidades por tipo de base de datos / tecnología 2](#_Toc203979270)

[Gestión del Catálogo y Stock – MongoDB 2](#_Toc203979271)

[Indexación y Búsqueda Semántica – Elasticsearch 2](#_Toc203979272)

[Relaciones Cliente-Producto y Recomendaciones – Neo4j 3](#_Toc203979273)

[Simulaciones 3](#_Toc203979274)

[Simulación de Compras 3](#_Toc203979275)

[Simulación de Búsquedas 3](#_Toc203979276)

[Visualización en Kibana 3](#_Toc203979277)

# Recursos útiles

Recuerde el directorio de recursos compartidos disponible en:

<https://nascorformacion0-my.sharepoint.com/:f:/g/personal/juan_pinuela_docente_nascorformacion_com/EtGSBITnqFpJp6-XpG6REOQBFycP_G9zgcB89jxPymjV2Q?e=IBIEER>

# Objetivo de la práctica

El presente proyecto integrador tiene como objetivo desarrollar un sistema completo de gestión y análisis para un mini eCommerce inteligente, integrando simultáneamente MongoDB, Elasticsearch y Neo4j. El alumno deberá implementar un flujo operativo y analítico completo que combine la gestión de datos, la búsqueda avanzada y el análisis de relaciones.

En cada apartado se mencionan los pasos y código clave de cada grupo de funcionalidades.

# Funcionalidades por tipo de base de datos / tecnología

## Gestión del Catálogo y Stock – MongoDB

* Almacena el catálogo de productos en MongoDB.
* Permite consultar el catálogo disponible.
* Decrementa automáticamente el stock tras cada compra.
* Guarda un registro histórico de compras en la colección 'compras'.

**Código clave:**

# Decrementar stock en MongoDB tras una compra  
col\_productos.update\_one(  
 {"nombre": prod["nombre"]},  
 {"$inc": {"stock": -1}}  
)  
  
# Registrar la compra en MongoDB  
db\_mongo["compras"].insert\_one({  
 "cliente": cliente["email"],  
 "producto": prod["nombre"],  
 "precio": prod["precio"],  
 "fecha": datetime.now().isoformat()  
})

## Indexación y Búsqueda Semántica – Elasticsearch

* Indexa el catálogo desde MongoDB.
* Realiza búsquedas semánticas sobre los productos.
* Registra cada búsqueda en el índice 'logs\_busquedas'.
* Exporta ventas a Elasticsearch para visualización.

**Código clave:**

def log\_busqueda(termino, resultados):  
 es.index(index="logs\_busquedas", document={  
 "termino\_busqueda": termino,  
 "resultados\_encontrados": len(resultados),  
 "fecha": datetime.now().isoformat()  
 })

## Relaciones Cliente-Producto y Recomendaciones – Neo4j

* Registra cada compra como relación en Neo4j.
* Genera recomendaciones automáticas.

**Código clave:**

tx.run("MATCH (c:Cliente {email: $email}), (p:Producto {nombre: $producto}) "  
 "MERGE (c)-[:COMPRA {fecha: date($fecha)}]->(p)",  
 email=cliente["email"], producto=prod["nombre"],  
 fecha=datetime.now().strftime("%Y-%m-%d"))

Para que el proyecto tenga verdadero valor práctico y visualización efectiva en Kibana, necesitas generar suficiente actividad simulada en el sistema, especialmente:

# Simulaciones

Para que el proyecto tenga verdadero valor práctico y visualización efectiva en Kibana, necesitas generar suficiente actividad simulada en el sistema, especialmente:

## Simulación de Compras

* Varias compras simuladas por cliente.
* Registrar cada compra en:
* MongoDB (compras + decremento de stock).
* Neo4j (relación :COMPRA).
* Elasticsearch (índice ventas para visualización).

## Simulación de Búsquedas

* Ejecutar muchas búsquedas aleatorias (por términos relevantes).
* Registrar cada búsqueda como log en el índice logs\_busquedas.
* Simulación de Recomendaciones (opcional pero recomendable)
* Generar recomendaciones en base a las compras simuladas.

# Visualización en Kibana

El alumno deberá crear un dashboard en Kibana que visualice:  
- Términos más buscados.  
- Productos más vendidos.  
- Evolución de ventas.