MODULO 04 - EJERCICIO 05-A

ALEXIS YURI M.

Diseña una estructura de colección para una tienda online que permita guardar información de productos, incluyendo variantes (talla, color), reviews y disponibilidad por sucursal.

1. Definición de campos.

Se diseñará un documento del producto con campos que incluyan las distintas variantes del producto, las reseñas y el inventario disponible en las sucursales, de esta forma, incluyendo mucha información se busca minimizar las consultas y optimizar el rendimiento

2. Estructura del documento.

A continuación, se presenta la estructura JSON propuesta con objetos y arrays anidados:

{

"\_id": ObjectId("5f9c3b8b9d3b9e2c6c8e3e4a"),

"nombre": "Zapatillas deportivas",

"descripcion": "Zapatillas ligeras, cómodas y resistentes, ideales para running.",

"marca": "DeportivaX",

"categoria": "Calzado",

"precio\_base": 79.99,

"sku": "DEPX-R-2023",

"variantes": [

{

"color": "Negro",

"talla": "42",

"inventario": [

{

"sucursal\_id": "suc\_A",

"stock": 15

},

{

"sucursal\_id": "suc\_B",

"stock": 5

}

]

},

{

"color": "Blanco",

"talla": "42",

"inventario": [

{

"sucursal\_id": "suc\_A",

"stock": 8

}

]

},

{

"color": "Negro",

"talla": "43",

"inventario": [

{

"sucursal\_id": "suc\_C",

"stock": 20

}

]

}

],

"reviews": [

{

"usuario\_id": "usr\_123",

"rating": 5,

"comentario": "¡Excelente producto, muy cómodas!",

"fecha": "2024-07-25"

},

{

"usuario\_id": "usr\_456",

"rating": 4,

"comentario": "Buena calidad, pero la talla es un poco ajustada.",

"fecha": "2024-07-28"

}

],

"disponible": true,

"fecha\_creacion": "2024-07-01T10:00:00Z"

}

3. Justificación de la estructura.

La estructura de documento propuesta utiliza un enfoque de modelo de datos embebido (o de subdocumentos). Esta elección se justifica por las siguientes razones:

- Minimización de consultas (Embedded data): Los datos de variantes (variantes), inventario (inventario) y reseñas (reviews) se embeben dentro del documento del producto. Esto permite que una sola consulta a la base de datos obtenga toda la información relevante de un producto, eliminando la necesidad de múltiples joins que serían necesarios en una base de datos relacional. Por ejemplo, al mostrar la página de un producto, la aplicación solo necesita leer un documento para acceder a sus variantes, stock por sucursal y reseñas.

- Agrupación de datos relacionados: La información que se accede en conjunto se mantiene unida. Las variantes de un producto, como talla y color, no tienen sentido si se separan del producto al que pertenecen. Lo mismo ocurre con las reseñas y el inventario.

- Manejo de colecciones semirregular (Arrays): Los campos como variantes y reviews se modelan como arrays. Esto es ideal porque el número de variantes y reseñas por producto puede varia, y al ser un array una estructura de datos flexible permite añadir o quitar elementos fácilmente.

- Datos planos: La información básica del producto, como nombre, precio\_base, y categoría, se mantiene plana en la parte superior del documento. Estos datos son comunes y se usan frecuentemente para búsquedas o listados.

4. Ejemplo en formato JSON.

El ejemplo JSON se muestra en el punto 2. Se puede observar cómo los datos de las variantes y las reviews están anidados dentro del documento principal del producto, demostrando un diseño flexible y escalable para una tienda de e-commerce.