

María Paula Murillo

202224530

Mateo López

202220119

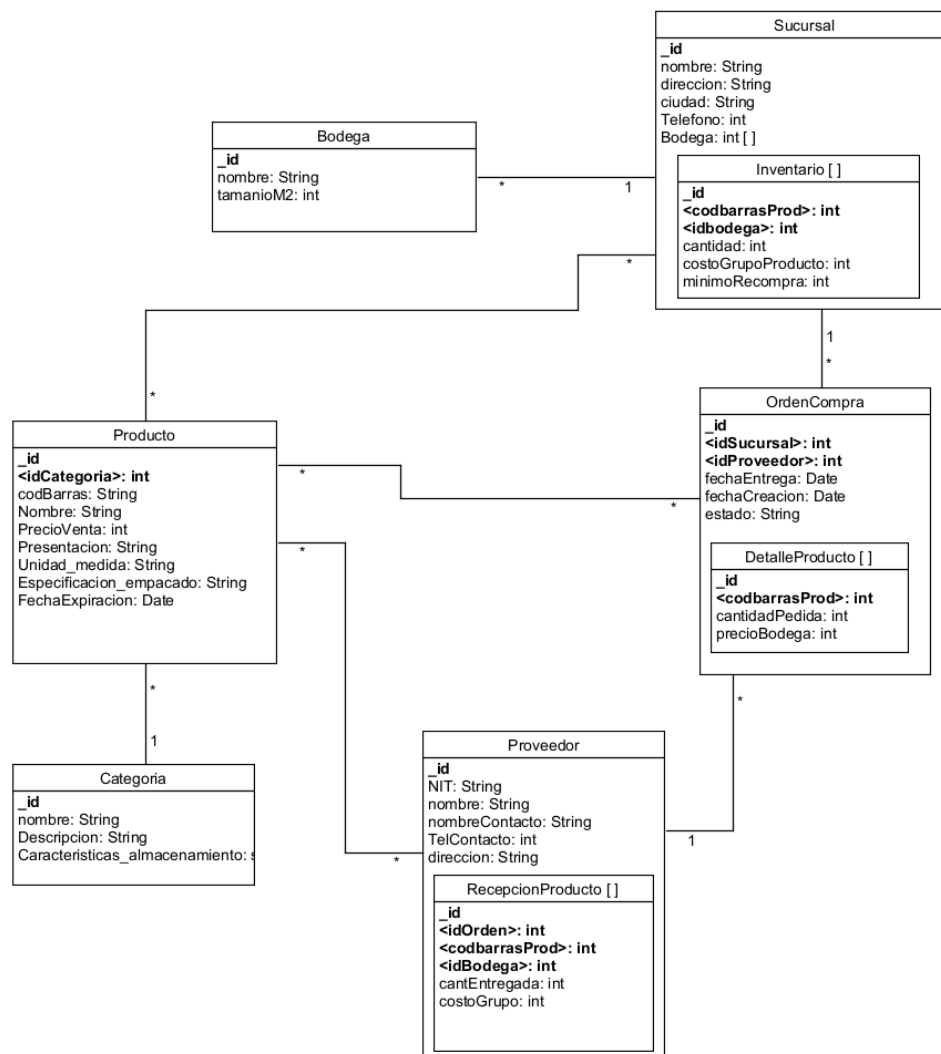
Adrián Arturo Suárez García

202123771

# Entrega 3

## Replanteamiento del modelo

Para el cambio a la nueva arquitectura de base de datos de MongoDB, se planteó un nuevo diagrama UML que considera las características de la base de datos no relacional utilizada:



# Diseño de la base de datos

## 1. Análisis de la carga de trabajo (workload). Para ello, presenten lo siguiente:

a. Identifiquen entidades y sus atributos

### Bodega

Atributo	Descripción
<b>_id</b>	Identificador de Mongo del documento
<b>Nombre</b>	Nombre de la bodega
<b>tamanoM2</b>	Tamaño de la bodega, medido en metros cuadrados.

### Producto

Atributo	Descripción
<b>_id</b>	Identificador de Mongo del documento
<b>IdCategoria</b>	Identificador de la categoría a la que pertenece.
<b>codBarras</b>	Identificador único del producto
<b>Nombre</b>	Nombre del producto
<b>PrecioVenta</b>	Precio por unidad de venta del producto
<b>Presentación</b>	Define la presentación del producto, puede especificar la cantidad de unidades contiene la unidad del producto (i.e. si se vende una caja de 6 unidades)
<b>Unidad_medida</b>	Especificación de si se mide por gramaje o mililitros
<b>Especificacion_enpacado</b>	Permite especificar el valor de volumen (si es ml) o masa (si es gr)
<b>fechaExpiracion</b>	Define la fecha de vencimiento en caso de tenerla

### Categoria

Atributo	Descripción
<b>_id</b>	Identificador de Mongo del documento
<b>nombre</b>	Nombre de la categoría
<b>Descripcion</b>	Descripción de los productos que pertenecen a esa categoría (restricciones o atributos compartidos)
<b>Características_almacenamiento</b>	Especifica si los productos que recaen en esta categoría necesitan un

	tratamiento especial al momento de ser empacados.
--	---

## Proveedor

Atributo	Descripción														
<b>_id</b>	Identificador de Mongo del documento														
<b>nombre</b>	Nombre de la empresa proveedora														
<b>NIT</b>	Número de identificación tributaria														
<b>nombreContacto</b>	Nombre de la persona de la empresa a la que se le puede contactar														
<b>TelContacto</b>	Información de contacto de la persona a contactar														
<b>Dirección</b>	Dirección de la sucursal del proveedor														
<b>RecepcionProducto</b>	<p>Lista de productos que han sido distribuidos por el proveedor. Se almacenan de la siguiente forma:</p> <table> <tr> <th>Atributo</th><th>Descripción</th></tr> <tr> <td><b>_id</b></td><td>Identificador de Mongo del documento</td></tr> <tr> <td><b>idOrden</b></td><td>Identificador de la orden que incluía este producto</td></tr> <tr> <td><b>codbarrasProd</b></td><td>Código de barras del producto ordenado</td></tr> <tr> <td><b>idBodega</b></td><td>Identificador de la bodega a la cual fue entregado el producto.</td></tr> <tr> <td><b>cantEntregada</b></td><td>Cantidad del producto que fue ordenado y entregado</td></tr> <tr> <td><b>costoGrupo</b></td><td>Costo de almacenamiento del grupo de productos</td></tr> </table>	Atributo	Descripción	<b>_id</b>	Identificador de Mongo del documento	<b>idOrden</b>	Identificador de la orden que incluía este producto	<b>codbarrasProd</b>	Código de barras del producto ordenado	<b>idBodega</b>	Identificador de la bodega a la cual fue entregado el producto.	<b>cantEntregada</b>	Cantidad del producto que fue ordenado y entregado	<b>costoGrupo</b>	Costo de almacenamiento del grupo de productos
Atributo	Descripción														
<b>_id</b>	Identificador de Mongo del documento														
<b>idOrden</b>	Identificador de la orden que incluía este producto														
<b>codbarrasProd</b>	Código de barras del producto ordenado														
<b>idBodega</b>	Identificador de la bodega a la cual fue entregado el producto.														
<b>cantEntregada</b>	Cantidad del producto que fue ordenado y entregado														
<b>costoGrupo</b>	Costo de almacenamiento del grupo de productos														

## OrdenCompra

Atributo	Descripción
<b>_id</b>	Identificador de Mongo del documento
<b>idSucursal</b>	Identificador de la sucursal a la que se le está haciendo el pedido
<b>idProveedor</b>	Identificador del proveedor que va a enviar los productos
<b>fechaEntrega</b>	Fecha en la que se espera recibir la orden

<b>fechaCreacion</b>	Fecha en la que se realizó la orden										
<b>Estado</b>	Estado de la orden, puede ser vigente, entregada o anulada										
<b>DetalleProducto</b>	<p>Lista de documentos que contienen la información de los productos ordenados, con la siguiente estructura:</p> <table> <tr> <th>Atributo</th><th>Descripción</th></tr> <tr> <td><b>_id</b></td><td>Identificador de Mongo del documento</td></tr> <tr> <td><b>codbarrasProd</b></td><td>Código de barras del producto que fue ordenado</td></tr> <tr> <td><b>cantidadPedida</b></td><td>Cantidad de unidades del producto ordenadas</td></tr> <tr> <td><b>precioBodega</b></td><td>Precio que le costará a la sucursal el producto</td></tr> </table>	Atributo	Descripción	<b>_id</b>	Identificador de Mongo del documento	<b>codbarrasProd</b>	Código de barras del producto que fue ordenado	<b>cantidadPedida</b>	Cantidad de unidades del producto ordenadas	<b>precioBodega</b>	Precio que le costará a la sucursal el producto
Atributo	Descripción										
<b>_id</b>	Identificador de Mongo del documento										
<b>codbarrasProd</b>	Código de barras del producto que fue ordenado										
<b>cantidadPedida</b>	Cantidad de unidades del producto ordenadas										
<b>precioBodega</b>	Precio que le costará a la sucursal el producto										

## Sucursal

Atributo	Descripción								
<b>_id</b>	Identificador de Mongo del documento								
<b>Nombre</b>	Nombre de la sucursal								
<b>Dirección</b>	Dirección de la sucursal								
<b>Ciudad</b>	Ciudad en la que se encuentra ubicada la sucursal								
<b>Teléfono</b>	Número de teléfono de la sucursal								
<b>Bodega</b>	Lista de identificadores de bodegas que almacenan productos de esa sucursal específica								
<b>Inventario</b>	<p>Lista de cada uno de los productos que están almacenados en las bodegas que proveen por esa sucursal, modelados de la siguiente forma:</p> <table> <tr> <th>Atributo</th><th>Descripción</th></tr> <tr> <td><b>_id</b></td><td>Identificador de Mongo del documento</td></tr> <tr> <td><b>codbarrasProd</b></td><td>Código de barras del producto almacenado</td></tr> <tr> <td><b>idBodega</b></td><td>Identificador de la bodega donde se encuentra almacenado</td></tr> </table>	Atributo	Descripción	<b>_id</b>	Identificador de Mongo del documento	<b>codbarrasProd</b>	Código de barras del producto almacenado	<b>idBodega</b>	Identificador de la bodega donde se encuentra almacenado
Atributo	Descripción								
<b>_id</b>	Identificador de Mongo del documento								
<b>codbarrasProd</b>	Código de barras del producto almacenado								
<b>idBodega</b>	Identificador de la bodega donde se encuentra almacenado								

	<b>Cantidad</b>	Cantidad de unidades de ese grupo del producto
	<b>costoGrupoProducto</b>	Costo de almacenamiento por unidad de ese grupo del producto
	<b>minimoRecompra</b>	Mínimo de unidades del producto que se requieren para solicitar una orden de compra

b. Cuantifiquen las entidades (cantidad de registros que tendría la BD para cada una de las entidades, pueden encontrar un aproximado en el enunciado).

Entidad	Cantidad de registros
<b>Sucursal</b>	150
<b>Inventario</b>	40,000 Asumiendo que en promedio cada producto tiene en promedio 2 registros, pueden ser en la misma sucursal y/o bodegas o totalmente diferentes
<b>Bodega</b>	900
<b>Proveedor</b>	10,000
<b>Producto</b>	20,000
<b>Categoría</b>	8
<b>Orden de Compra</b>	70,000
<b>Detalle de Producto</b>	210,000 Asumiendo que cada orden de compra tiene en promedio 3 productos
<b>Recepción de producto</b>	70,000

c. Análisis de las operaciones de lectura y escritura por entidad:

Entidad	Operaciones	Información necesaria	Tipo
<b>Sucursal</b>	Crear sucursales, editar la información del inventario	Detalles de bodegas y productos	Escritura
<b>Inventario</b>	Modificar y revisar estado de inventarios	Detalles de productos	Escritura + lectura
<b>Bodega</b>	Consultar las bodegas	Detalles de la bodega	Lectura

<b>Proveedor</b>	Insertar proveedores	Detalle de proveedores	Escritura
<b>Producto</b>	Insertar productos	Detalle de los productos, identificador de la sucursal	Escritura
<b>Categoría</b>	Revisar categorías	Información de categorías y de productos	Lectura
<b>Orden de Compra</b>	Generar órdenes de compra	Detalles de productos y proveedores	Escritura
<b>Recepción de producto</b>	Registrar recepciones de producto	Detalle de la orden de compra	Escritura

d. Cuantifiquen las operaciones de lectura y escritura para cada entidad. Para ello utilicen una tabla como la del ejemplo del anexo B.

Entidad	Operaciones	Información necesaria	Tipo	Tasa
<b>Sucursal</b>	Crear sucursales, editar la información del inventario	Detalles de bodegas y productos	Escritura	1/mes
<b>Inventario</b>	Modificar estado de inventarios	Detalles de productos	Escritura	7/min
<b>Inventario</b>	Revisar el estado de inventarios	Detalles de productos	Lectura	1.3/min
<b>Bodega</b>	Consultar las bodegas	Detalles de la bodega	Lectura	1/sem
<b>Proveedor</b>	Consultar proveedores	Detalle de proveedores	Lectura	0.13/min
<b>Producto</b>	Insertar productos	Detalle de los productos, identificador de la sucursal	Escritura	7/min
<b>Categoría</b>	Revisar categorías	Información de categorías y de productos	Lectura	1/min
<b>Orden de Compra</b>	Generar órdenes de compra	Detalles de productos y proveedores	Escritura	0.13/min
<b>Orden de compra</b>	Consultar ordenes de compra	Detalles de las órdenes, productos y proveedores	Lectura	0.7/min

<b>Recepción de producto</b>	Registrar recepciones de producto	Detalle de la orden de compra	Escritura	0.13/min
------------------------------	-----------------------------------	-------------------------------	-----------	----------

2. Describan las colecciones de datos y las relaciones entre ellas (NoSQL) que corresponden al modelo conceptual UML propuesto. Para ello, presenten lo siguiente:

- a. La lista de entidades con la descripción de cada una de ellas.

ENTIDAD	DESCRIPCIÓN
<b>Bodega</b>	Cada entidad Bodega representa una bodega de SuperAndes. Una sucursal puede estar asociada muchas bodegas y muchas bodegas pueden guardar muchos productos.
<b>Sucursal</b>	Cada entidad Sucursal representa una sucursal de SuperAndes. Una sucursal puede estar asociada a muchas bodegas y una sucursal puede tener muchas ordenes de compra. Además, cada sucursal puede tener múltiples inventarios asociados.
<b>OrdenCompra</b>	Cada entidad OrdenCompra representa una orden de compra que realiza una sucursal de SuperAndes a un Proveedor. Una sucursal puede tener múltiples ordenes de compra, una orden de compra puede incluir múltiples productos y un proveedor puede recibir muchas ordenes de compra. Además cada orden de compra puede tener múltiples DetalleProducto.
<b>Producto</b>	Cada entidad Producto representa un producto que vende SuperAndes. Una bodega puede almacenar múltiples productos, cada producto se clasifica dentro de una categoría y muchos productos pueden tener la misma categoría; un proveedor puede ofrecer muchos productos y un producto puede ser ofrecido por muchos proveedores; una orden de compra puede incluir muchos productos.

<b>Categoria</b>	Cada entidad Categoria representa una categoría en la que pueden ser clasificados los productos vendidos por SuperAndes. Cada producto se clasifica dentro de una categoría y muchos productos pueden tener la misma categoría
<b>Proveedor</b>	Cada entidad Proveedor representa un proveedor asociado a SuperAndes. Cada Proveedor puede tener muchas ordenes de compra; un proveedor puede ofrecer muchos productos y un producto puede ser ofrecido por muchos proveedores.
<b>Inventario</b>	Cada entidad Inventario hace referencia al el inventario o información útil sobre un producto de una sucursal de SuperAndes. Cada sucursal puede tener múltiples inventarios y cada inventario está asociado a un producto y una bodega donde está ese producto.
<b>DetalleProducto</b>	Cada entidad DetalleProducto hace referencia a la información relevante sobre un producto para hacer su orden de compra. Una orden de compra puede tener múltiples detalle producto y cada detalle producto está asociado a un producto.
<b>RecepcionProducto</b>	Cada entidad RecepcionProducto hace referencia a la información relevante para un proveedor sobre un producto de una orden. Cada proveedor puede tener múltiples recepción producto y cada recepción producto está asociada a una orden, un producto y una bodega.

b. Las relaciones entre entidades y su cardinalidad (uno a uno, uno a muchos, o muchos a muchos).

ENTIDADES	RELACIÓN
<b>Bodega - Sucursal</b>	Many to One
<b>Producto - Inventarios</b>	Many to Many
<b>Sucursal - OrdenCompra</b>	One to Many
<b>OrdenCompra - Proveedor</b>	Many to One
<b>Proveedor - Producto</b>	Many to Many
<b>Producto - Categoria</b>	Many to One



<b>Producto – OrdenCompra</b>	Many to Many
<b>Proveedor - Recepción producto</b>	One to Many
<b>Orden - DetalleProducto</b>	One to Many
<b>Sucursal - Inventario</b>	One to Many

c. El análisis de selección de esquema de asociación (referenciado o embebido) para cada relación entre entidades. Para ello use la tabla de análisis vista en clase, la cual se retoma en el anexo C, junto con los resultados del análisis de la carga de trabajo (workload), descrita antes.

**Bodega – Sucursal:** Se maneja como una referencia dado que ambas entidades no se deben archivar al tiempo y tienen cargas de trabajo distintas, pues, como indica el enunciado, la consulta sobre bodegas es muy recurrente, mientras que la consulta sobre sucursales no. Así, al separarlos, podemos hacer consultas y cambios sobre bodegas sin cambiar las sucursales.

**Producto-Inventario:** Se maneja referenciación para evitar la duplicación de datos y demás problemas asociados. Los inventarios contienen los datos para todas las bodegas, y en cada bodega pueden haber los mismos productos. Si tenemos 20000 productos y 150 sucursales, imaginemos tener el 50% de los productos en todas las bodegas, sería un desperdicio grandísimo de espacio tener 150 veces todos los productos. Además, siempre pueden haber cambios sobre productos en sucursales, algunos discontinuados y otros nuevos, entonces agregar o quitar a cada sucursal un producto es muy complejo, en cambio manejar la referencia es óptimo.

**Sucursal-OrdenCompra:** Se usa referenciación porque, si tenemos 70000 órdenes de compra, es muy complejo manejar todo en conjunto. Concretamente, la cardinalidad de la entidad hijo, o sea Sucursal, crece inmensamente, entonces esto dificultaría mucho manejar las sucursales dentro de las órdenes. Además, evidentemente, la replicación de datos sería inmensa, pues guardar las sucursales dentro de cada una de las 70000 órdenes simplemente no tiene sentido. En ese orden de ideas, el desperdicio de memoria sería absurdo, sin mencionar la dificultad de las consultas, pues revisar las 70000 órdenes para buscar los datos de una sucursal sería un despropósito. La referenciación facilita todo.

**OrdenCompra – Proveedor:** los proveedores pueden tener múltiples órdenes de compra, por lo cual es mejor referenciar, dado que al embeber tendríamos demasiados datos hijo. Las órdenes tienen varios datos y relaciones, además de una entidad embebida, por lo cuál insertar las órdenes sería una complicación por la cardinalidad y el tamaño de la entidad orden. Hacerlo al contrario sería un

despropósito, pues guardar proveedores dentro de las órdenes implicaría una inmensa replicación de datos de los proveedores, dando problemas para las búsquedas y el uso de memoria. Así, concluimos que es mejor referenciar.

Proveedor – Producto: dado que la relación es muchos a muchos, por temas de cardinalidad, duplicación de los datos, el tamaño de ambas relaciones, las otras relaciones que tienen cada entidad, y lo importante que son las consultas independientes, es mejor referenciar. Embeber una en la otra implica duplicar muchos datos y gastar memoria, sin importar cuál sea el hijo, además que son 2 entidades principales y cuyas consultas son muy importantes, por lo cual debemos poder consultarlas independientemente.

Producto - Categoría: dentro de cada producto está referenciada la categoría. No se puede embeber la categoría dentro de un producto por duplicación de datos, pues hay solo 8 categorías, si hay 20000 productos sería tener cada categoría 2500 veces, lo cual es una pérdida de memoria inconcebible. Se podría haber hecho al revés, embeber los productos en las categorías, pero dado que las consultas hacia productos son más importantes y frecuentes, mientras que las categorías no, tomamos la decisión de no embeber los productos. Por ejemplo, productos tiene muchas consultas dado que se buscan los productos en las bodegas y demás, entonces tener que pasar por la categoría cada vez que se busque un producto va en contra de los deseos de negocio. Concretamente, la carga de lecturas y escrituras de productos es demasiado grande para la carga de categorías, por lo cual es inviable embeberla.

Producto-OrdenCompra: es una relación muchos a muchos, específicamente manejada a partir de la clase embebida en OrdenCompra. Sabiendo que hay 70000 órdenes, y 20000 productos, por temas de duplicación y memoria no puede embeberse una en otra. Además, la cantidad de consultas respecto a las órdenes es muy grande, pero respecto a productos también, por lo cual no podemos dar prelación a ninguna, simplemente hay que referenciar. Se guarda la referencia desde las órdenes porque tiene más sentido, pues las órdenes necesitan productos según la lógica de negocio, mientras que los productos existen precisamente para poder ser usados (llamados) por otras entidades, como es el caso con las órdenes, los inventarios o los proveedores.

Proveedor-recepción producto: se embebió la recepción de un producto a un proveedor por simplicidad y porque, desde la perspectiva del negocio, van muy de la mano. Dentro del modelo la única utilidad de un proveedor es hacerle pedidos y saber que llegaron; ya se explicó la relación de los proveedores con órdenes, pero esas razones no aplican con aquí. Concretamente, no hay razones para separar las 2 clases, en cambio sí facilitan almacenar juntos los proveedores con los productos recibidos. No se embebe al revés porque los proveedores son más

importantes en términos de consultas y las relaciones con otras entidades que las recepciones de productos.

**OrdenCompra – DetalleProducto:** Detalle producto es una clase que se usa para guardar datos en la relación entre las órdenes y los productos. Los atributos que guarda no son propios de ninguna de las otras clases, solo de su relación, por lo cual se decidió embeberla, ya que no podíamos solo eliminar esta información entre relaciones. Ahora bien, se embebe en órdenes porque tiene más sentido que en productos, pues así podemos saber cuáles son los productos, cantidades y precios de una orden, mientras que, al revés, sería como tener un producto, y cuanto costó para cada orden individual; es mejor manejarlo por la orden. Concretamente, se embebe por simplicidad del modelo, porque las 2 cosas van juntas (una orden no puede no tener productos), las 2 se consultan al tiempo y por ende es más fácil guardarlas juntas. Desde DetalleProducto es que se hace la referenciación entre productos y órdenes.

**Sucursal – Inventario:** Hemos embebido los inventarios en las sucursales por temas de simplificación del modelo. Ya sabemos que una sucursal tiene muchas bodegas, pero también tiene muchos productos asociados, además, necesitamos registrar atributos propios de la relación entre bodegas y productos. Todo esto se simplifica unificándolo en una sucursal, pues ahí se manejan las relaciones referenciadas, explicadas previamente. De esta forma podemos manejar el inventario de las bodegas, sabiendo cuáles productos están dónde. También tiene sentido porque estas 2 clases van juntas, pues una sucursal no puede no tener un inventario, esa es la esencia del negocio, y al tiempo, un inventario debe estar en algún sitio, o sea, una sucursal.




d. Una descripción gráfica usando Json de cada relación entre entidades en donde presente un ejemplo de datos junto con el esquema de asociación usado (referenciado o embebido). En el anexo D se muestra un ejemplo de lo que se requiere.

**Producto-Categoría:** referenciado. Vemos que en cada producto (parte izquierda) está referenciado el id de cada categoría (parte derecha)

Entidades	Asociación	Ejemplo
-----------	------------	---------

Sucursal inventario	- Embebido	<pre> _id: 9 NOMBRE: "Sucursal 9" DIRECCION: "Calle 9 #123, Zona 9" CIUDAD: "Ciudad 9" TELEFONO: 1000000009 BODEGA: Array (3)   0: 10   1: 3   2: 5 INVENTARIOS: Array (10)   0: Object     _id: 1     CODIGO_BARRAS: 1     ID_BODEGA: 1     CANTIDAD_OCUPADA: 28     COSTO_GRUPO_PRODUCTO: 1801     MINIMO_RECOMPRA: 6     1: Object     2: Object     3: Object     4: Object     5: Object     6: Object     7: Object     8: Object     9: Object </pre> <p><i>Inventario</i> (green bracket on INVENTARIOS array)</p> <p><i>Sucursal</i> (blue bracket on Sucursal object)</p>
Inventario Producto	- Referenciado	<pre> _id: 9 NOMBRE: "Sucursal 9" DIRECCION: "Calle 9 #123, Zona 9" CIUDAD: "Ciudad 9" TELEFONO: 1000000009 BODEGA: Array (3)   0: 10   1: 3   2: 5 INVENTARIOS: Array (10)   0: Object     _id: 1     CODIGO_BARRAS: 1     ID_BODEGA: 1     CANTIDAD_OCUPADA: 28     COSTO_GRUPO_PRODUCTO: 1801     MINIMO_RECOMPRA: 6     1: Object     2: Object     3: Object     4: Object     5: Object     6: Object     7: Object     8: Object     9: Object </pre> <p><i>Referencia</i> (blue arrow from CODIGO_BARRAS: 1 to CODIGO_BARRAS: 1 in product object)</p> <pre> _id: 1 CODIGO_BARRAS: 1 NOMBRE: "Arroz" PRECIO_VENTA: 95 PRESENTACION: "Tarro" UNIDAD_MEDIDA: "g" ESP_EMPACADO: "S" FECHA_EXP: 2025-09-16T05:00:00.000+00:00 CATEGORIA: 4 </pre>
Inventario - Bodega	- Referenciado	<pre> _id: 9 NOMBRE: "Sucursal 9" DIRECCION: "Calle 9 #123, Zona 9" CIUDAD: "Ciudad 9" TELEFONO: 1000000009 BODEGA: Array (3)   0: 10   1: 3   2: 5 INVENTARIOS: Array (10)   0: Object     _id: 1     CODIGO_BARRAS: 1     ID_BODEGA: 1     CANTIDAD_OCUPADA: 28     COSTO_GRUPO_PRODUCTO: 1801     MINIMO_RECOMPRA: 6     1: Object     2: Object     3: Object     4: Object     5: Object     6: Object     7: Object     8: Object     9: Object </pre> <p><i>Referencia</i> (blue arrow from ID_BODEGA: 1 to ID: 1 in Bodega object)</p> <pre> _id: 1 NOMBRE: "Bodega Central" TAMANIO: 1000 </pre>
OrdenCompra DetalleProducto	- Embebido	<pre> _id: 1 IDSUCURSAL: 9 IDPROVEEDOR: 2 FECHAENTREGA: 2024-12-08T19:16:55.500+00:00 FECHACREACION: 2024-11-20T19:16:55.500+00:00 ESTADO: "Completada" DETALLEPRODUCTO: Array (10)   0: Object     _id: 1     CODIGO_BARRAS_PROD: 8     CANTIDAD_PEDIDA: 83     PRECIO_BODEGA: 212     1: Object     2: Object     3: Object     4: Object     5: Object     6: Object     7: Object     8: Object     9: Object </pre> <p><i>Detalle Producto</i> (green bracket on DETALLEPRODUCTO array)</p> <p><i>Orden Compra</i> (blue bracket on OrdenCompra object)</p>

DetalleProducto- Producto	Referenciado	<pre>_id: 1 IDSUBCURSAL: 9 IDPROVEEDOR: 2 FECHAENTREGA: 2024-12-08T19:16:55.500+00:00 FECHACREACION: 2024-11-20T19:16:55.500+00:00 ESTADO: "Completada" DETALLEPRODUCTO: Array (10)   0: Object     _id: 1     CODIGOBARRASPROD: 8     CANTIDAD_PEDIDA: 83     PRECIOBODEGA: 212   1: Object   2: Object   3: Object   4: Object   5: Object   6: Object   7: Object   8: Object   9: Object     _id: 8     CODBARRAS: 8     NOMBRE: "Azucar"     PRECIOVENTA: 92     PRESENTACION: "Caja"     UNIDAD_MEDIDA: "g"     ESP_EMPACADO: "No"     FECHA_EXP: 2025-01-25T01:59:29.469+00:00     CATEGORIA: 1</pre>
Proveedor – RecepcionProducto	Embebido	<pre>_id: 2 NIT: 2 NOMBRE: "Proveedor 2" NOMBRECONTACTO: "Contacto 2" TELCONTACTO: 1000000002 DIRECCION: "Avenida 2, Ciudad 2" RECEPCIONPRODUCTO: Array (10)   0: Object     _id: 1     IDPRODUCTO: 1     IDORDEN: 40     IDBODEGA: 3     CANTIDADENTREGADA: 109     COSTOGRUPO: 3291   1: Object   2: Object   3: Object   4: Object   5: Object   6: Object   7: Object   8: Object   9: Object</pre> <p>Recepción Producto</p> <p>Proveedor</p>
RecepcionProducto - OrdenProducto	Referenciado	<pre>_id: 2 NIT: 2 NOMBRE: "Proveedor 2" NOMBRECONTACTO: "Contacto 2" TELCONTACTO: 1000000002 DIRECCION: "Avenida 2, Ciudad 2" RECEPCIONPRODUCTO: Array (10)   0: Object     _id: 1     IDPRODUCTO: 1     IDORDEN: 40     IDBODEGA: 3     CANTIDADENTREGADA: 109     COSTOGRUPO: 3291   1: Object   2: Object   3: Object   4: Object   5: Object   6: Object   7: Object   8: Object   9: Object     _id: 40     IDSUBCURSAL: 6     IDPROVEEDOR: 3     FECHAENTREGA: 2024-12-08T19:16:55.500+00:00     FECHACREACION: 2024-11-20T19:16:55.500+00:00     ESTADO: "Completada"     DETALLEPRODUCTO: Array (10)</pre>
RecepcionProducto - Producto	Referenciado	<pre>_id: 2 NIT: 2 NOMBRE: "Proveedor 2" NOMBRECONTACTO: "Contacto 2" TELCONTACTO: 1000000002 DIRECCION: "Avenida 2, Ciudad 2" RECEPCIONPRODUCTO: Array (10)   0: Object     _id: 1     IDPRODUCTO: 1     IDORDEN: 40     IDBODEGA: 3     CANTIDADENTREGADA: 109     COSTOGRUPO: 3291   1: Object   2: Object   3: Object   4: Object   5: Object   6: Object   7: Object   8: Object   9: Object     _id: 1     CODBARRAS: 1     NOMBRE: "Arroz"     PRECIOVENTA: 95     PRESENTACION: "Tarro"     UNIDAD_MEDIDA: "g"     ESP_EMPACADO: "Si"     FECHA_EXP: 2025-09-16T05:00:00.000+00:00     CATEGORIA: 4</pre>

RecepcionProducto - Bodega	Referenciado	<pre>_id: 2 NIT: 2 NOMBRE: "Proveedor 2" NOMBRECONTACTO: "Contacto 2" TELCONTACTO: 1000000002 DIRECCION: "Avenida 2, Ciudad 2" RECEPCIONPRODUCTO: Array (10)   0: Object     _id: 1     IDPRODUCTO: 1     IDORDEN: 40     IDBODEGA: 3     CANTIDADENTREGADA: 109     COSTOGRUPO: 3291     1: Object     2: Object     3: Object     4: Object     5: Object     6: Object     7: Object     8: Object     9: Object</pre> 
OrdenCompra Sucursal	- Referenciado	<pre>_id: 1 IDSUCURSAL: 9 IDPROVEEDOR: 2 FECHAENTREGA: 2024-12-08T19:16:55.500+00:00 FECHACREACION: 2024-11-20T19:16:55.500+00:00 ESTADO: "Completada" DETALLEPRODUCTO: Array (10)   0: Object     _id: 1     CODIGOBARRASPROD: 8     CANTIDAD_PEDIDA: 83     PRECIOBODEGA: 212     1: Object     2: Object     3: Object     4: Object     5: Object     6: Object     7: Object     8: Object     9: Object     _id: 8     NOMBRE: "Sucursal 9"     DIRECCION: "Calle 9 #123, Zona 9"     CIUDAD: "Ciudad 9"     TELEFONO: 1000000009     BODEGA: Array (3)       0: 10       1: 9       2: 5     INVENTARIOS: Array (10)       0: Object         _id: 1         CODIGOBARRAS: 1         IDBODEGA: 1         CANTIDAD_OCUPADA: 28         COSTO_GRUPO_PRODUCTO: 1881         MENINO_RECOMPRA: 6         1: Object         2: Object         3: Object         4: Object         5: Object         6: Object         7: Object         8: Object         9: Object</pre> 
OrdenCompra Proveedor	- Referenciado	<pre>_id: 1 IDSUCURSAL: 9 IDPROVEEDOR: 2 FECHAENTREGA: 2024-12-08T19:16:55.500+00:00 FECHACREACION: 2024-11-20T19:16:55.500+00:00 ESTADO: "Completada" DETALLEPRODUCTO: Array (10)   0: Object     _id: 1     CODIGOBARRASPROD: 8     CANTIDAD_PEDIDA: 83     PRECIOBODEGA: 212     1: Object     2: Object     3: Object     4: Object     5: Object     6: Object     7: Object     8: Object     9: Object     NIT: 2     NOMBRE: "Proveedor 2"     NOMBRECONTACTO: "Contacto 2"     TELCONTACTO: 1000000002     DIRECCION: "Avenida 2, Ciudad 2"     RECEPCIONPRODUCTO: Array (10)       0: Object         _id: 1         IDPRODUCTO: 1         IDORDEN: 40         IDBODEGA: 3         CANTIDADENTREGADA: 109         COSTOGRUPO: 3291         1: Object         2: Object         3: Object         4: Object         5: Object         6: Object         7: Object         8: Object         9: Object</pre> 
Producto Categoria	- Referenciado	<pre>_id: 8 CODBARRAS: 8 NOMBRE: "Azucar" PRECIOVENTA: 92 PRESENTACION: "Caja" UNIDAD_MEDIDA: "g" ESP_EMPACADO: "No" FECHA_EXP: 2025-01-25T01:09:29.469+00:00 CATEGORIA: 1 _id: 1 NOMBRE: "perecedero" DESCRIPCION: "Con fecha de caducidad" CARACTERISTICAS: "Vence pronto"</pre> 

# Escenarios de Prueba

## 1. Escenarios para los RFs:

### a. RF3:

La primera foto es del front, con los datos para crear el proveedor. Vemos que al final la respuesta indica que se creó el proveedor exitosamente. La segunda foto es del back, mostrando el proveedor que acabamos de crear en la base de datos, demostrando que funciona el requerimiento.

Crear proveedor

13	
456	
Prueba	
Prueba	
234567	
Prueba	
35	10
7	6
500	222000

Agregar Recepción Producto a la lista

Agregar lista de recepción producto

Submit

"Proveedor creado exitosamente"

PROVEEDORES +

157.253.236.88:8087 > ISIS2304F16202420 > PROVEEDORES

Documents Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find

ADD DATA EXPORT DATA UPDATE DELETE 50 1 - 14 of 14

```
{
  "_id": 12,
  "NIT": 456,
  "NOMBRE": "Prueba",
  "DIRECCION": "Prueba",
  "NOMBRECONTACTO": "Prueba",
  "TELCONTACTO": 234567,
  "RECEPCIONPRODUCTO": Array (1)
  "_class": "un1andes.edu.co.proyecto.modelo.Proveedor"
}
```

```
1  _id: 13
2  NIT: 456
3  NOMBRE: "Prueba,"
4  DIRECCION: "Prueba,"
5  NOMBRECONTACTO: "Prueba,"
6  TELCONTACTO: 234567
7  RECEPCIONPRODUCTO: Array (1)
8    0: Object
9      _id: 35
10     IDPRODUCTO: 10
11     IDORDEN: 7
12     IDBODEGA: 6
13     CANTIDADENTREGADA: 500
14     COSTOGRUPO: 222000
15     _class: "un1andes.edu.co.proyecto.modelo.Proveedor,"
```

### b. RF6:

La primera foto es del front, con los datos para crear una orden. Vemos que al final la respuesta indica que se creó la orden exitosamente. La segunda foto es del back, mostrando la orden que acabamos de crear en la base de datos, demostrando que funciona el requerimiento.

Crear Orden para una sucursal

2024-12-24

vigente

34

456

17

Producto Código de barras

Producto Cantidad

Producto Precio en bodega

Producto ID

Agregar Producto a la lista

- 11, 12, 13, 14
- 15, 16, 17, 18

Agregar lista de productos

Submit

"Orden creada exitosamente"



PROVEEDORES

ORDENCOMPRA

+

157.253.236.88:8087

>

ISIS2304F16202420

>

ORDENCOMPRA

Documents 14

Aggregations

Schema

Indexes 1

Validation

🔍

Type a query: { field: 'value' } or [Generate query](#)

➕ ADD DATA

📄 EXPORT DATA

✎ UPDATE

🗑 DELETE

```

_id: 15
IDSUCURSAL : 34
IDPROVEEDOR : 456
FECHAENTREGA : 2024-12-24T05:00:00.000+00:00
FECHACREACION : 2024-12-01T05:00:00.000+00:00
ESTADO : "vigente"
▶ DETALLEPRODUCTO : Array (1)
  _class : "uniandes.edu.co.proyecto.modelo.Orden"

```

```

_id: 16
IDSUCURSAL : 34
IDPROVEEDOR : 456
FECHAENTREGA : 2024-12-24T05:00:00.000+00:00
FECHACREACION : 2024-12-01T05:00:00.000+00:00
ESTADO : "vigente"
▶ DETALLEPRODUCTO : Array (1)
  _class : "uniandes.edu.co.proyecto.modelo.Orden"

```

```

_id: 17
IDSUCURSAL : 34
IDPROVEEDOR : 456
FECHAENTREGA : 2024-12-24T05:00:00.000+00:00
FECHACREACION : 2024-12-01T05:00:00.000+00:00
ESTADO : "vigente"
▼ DETALLEPRODUCTO : Array (2)
  0: Object
    _id : 14
    CODIGOBARRASPROD : 11
    CANTIDAD_PEDIDA : 12
    PRECIOBODEGA : 13
  1: Object
    _id : 18
    CODIGOBARRASPROD : 15
    CANTIDAD_PEDIDA : 16
    PRECIOBODEGA : 17
  _class : "uniandes.edu.co.proyecto.modelo.Orden"

```

## 2. Escenarios para los RFCs:

### a. RFC1:

Prueba para sucursal:

La foto anexada a la izquierda es la prueba desde el front. A la derecha tenemos una foto de la sucursal con id 1. No se puede mostrar con fotos todos los productos de la sucursal, pues es demasiado largo, pero vemos que en la izquierda, al hacer la consulta, aparecen los productos con códigos de barra 1 y 2, mientras que en la foto de la izquierda vemos que los 2 primeros productos en el inventario de la sucursal 1 son aquellos con códigos de barra 1 y 2.

### Productos por sucursal

```
[
  {
    "codBarras": 1,
    "_id": 1,
    "CODBARRAS": 1,
    "NOMBRE": "Arroz",
    "PRECIOVENTA": 95,
    "PRESENTACION": "Tarro",
    "UNIDAD_MEDIDA": "g",
    "ESP_EMPACADO": "SI",
    "FECHA_EXP": "2025-09-16",
    "CATEGORIA": 4
  },
  {
    "codBarras": 2,
    "_id": 2,
    "CODBARRAS": 2,
    "NOMBRE": "Leche",
    "PRECIOVENTA": 120,
    "PRESENTACION": "Caja",
    "UNIDAD_MEDIDA": "g",
    "ESP_EMPACADO": "SI",
    "FECHA_EXP": "2025-04-04",
    "CATEGORIA": 1
  },
  {
    "codBarras": 3,
    "_id": 3,
    "CODBARRAS": 3,
    "NOMBRE": "Huevos",
    "PRECIOVENTA": 76,
    "PRESENTACION": "Botella",
    "UNIDAD_MEDIDA": "L",
    "ESP_EMPACADO": "SI",
    "FECHA_EXP": "2025-01-02",
    "CATEGORIA": 2
  },
  {
    "codBarras": 4,
    "_id": 4,
    "CODBARRAS": 4,
    "NOMBRE": "Pan",
    "PRECIOVENTA": 427,
    "PRESENTACION": "Caja",
    "UNIDAD_MEDIDA": "ml",
    "ESP_EMPACADO": "No",
    "FECHA_EXP": "2025-10-30",
    "CATEGORIA": 2
  }
]
```

\_id: 1

NOMBRE : "Sucursal 1"

DIRECCION : "Calle 1 #123, Zona 1"

CIUDAD : "Ciudad 1"

TELEFONO : 1000000001

► BODEGA : Array (4)

▼ INVENTARIOS : Array (14)

▼ 0: Object

\_id: 1

CODIDIGOBARRAS : 1

IDBODEGA : 3

CANTIDAD\_OCUPADA : 4

COSTO\_GRUPO\_PRODUCTO : 201

MINIMO\_RECOMPRA : 6

▼ 1: Object

\_id: 2

CODIDIGOBARRAS : 2

IDBODEGA : 4

CANTIDAD\_OCUPADA : 7

COSTO\_GRUPO\_PRODUCTO : 401

MTNTMO\_RECOMPRA : 11

### Prueba para rango de precios:

En la foto de la izquierda se observa la prueba en el front, estableciendo un precio para cada producto entre 95 y 110. Cuando se cambia el mínimo a 96, y el tope a 120, observamos que cambia el resultado, eliminando el producto con precio 95 y agregando otro con precio 120, mostrando que el filtro funciona.

### Productos por precio

```
[
  {
    "codBarras": 1,
    "_id": 1,
    "CODBARRAS": 1,
    "NOMBRE": "Arroz",
    "PRECIOVENTA": 95,
    "PRESENTACION": "Tarro",
    "UNIDAD_MEDIDA": "g",
    "ESP_EMPACADO": "SI",
    "FECHA_EXP": "2025-09-16",
    "CATEGORIA": 4
  },
  {
    "codBarras": 6,
    "_id": 6,
    "CODBARRAS": 6,
    "NOMBRE": "Queso",
    "PRECIOVENTA": 97,
    "PRESENTACION": "Caja",
    "UNIDAD_MEDIDA": "L",
    "ESP_EMPACADO": "SI",
    "FECHA_EXP": "2025-07-10",
    "CATEGORIA": 5
  }
]
```

## Productos por precio

```
[
  {
    "codBarras": 6,
    "_id": 6,
    "CODBARRAS": 6,
    "NOMBRE": "Queso",
    "PRECIOVENTA": 97,
    "PRESENTACION": "Caja",
    "UNIDAD_MEDIDA": "L",
    "ESP_EMPACADO": "Si",
    "FECHA_EXP": "2025-07-10",
    "CATEGORIA": 5
  },
  {
    "codBarras": 10,
    "_id": 10,
    "CODBARRAS": 10,
    "NOMBRE": "Te",
    "PRECIOVENTA": 120,
    "PRESENTACION": "Botella",
    "UNIDAD_MEDIDA": "Kg",
    "ESP_EMPACADO": "Si",
    "FECHA_EXP": "2025-02-03",
    "CATEGORIA": 4
  }
]
```

Prueba para fecha expiración mayor:

La primera imagen muestra el requerimiento para una fecha superior al 1 de diciembre del 2025. Se observa que solo hay un producto con la fecha de expiración mayor. Si ponemos un mes menos, vemos que esto cambia, agregando 2 productos cuya fecha de expiración es mayor a la indicada, demostrando que el filtro funciona.

## Productos por fecha

```
[
  {
    "codBarras": 5,
    "_id": 5,
    "CODBARRAS": 5,
    "NOMBRE": "Jamon",
    "PRECIOVENTA": 362,
    "PRESENTACION": "Lata",
    "UNIDAD_MEDIDA": "ml",
    "ESP_EMPACADO": "Si",
    "FECHA_EXP": "2025-11-03",
    "CATEGORIA": 4
  }
]
```

## Productos por fecha

```
[
  {
    "codBarras": 4,
    "_id": 4,
    "CODBARRAS": 4,
    "NOMBRE": "Pan",
    "PRECIOVENTA": 427,
    "PRESENTACION": "Caja",
    "UNIDAD_MEDIDA": "ml",
    "ESP_EMPACADO": "No",
    "FECHA_EXP": "2025-10-30",
    "CATEGORIA": 5
  },
  {
    "codBarras": 5,
    "_id": 5,
    "CODBARRAS": 5,
    "NOMBRE": "Jamon",
    "PRECIOVENTA": 362,
    "PRESENTACION": "Lata",
    "UNIDAD_MEDIDA": "ml",
    "ESP_EMPACADO": "Si",
    "FECHA_EXP": "2025-11-03",
    "CATEGORIA": 4
  },
  {
    "codBarras": 7,
    "_id": 7,
    "CODBARRAS": 7,
    "NOMBRE": "Mantequilla",
    "PRECIOVENTA": 157,
    "PRESENTACION": "Caja",
    "UNIDAD_MEDIDA": "Unidad",
    "ESP_EMPACADO": "Si",
    "FECHA_EXP": "2025-10-07",
    "CATEGORIA": 2
  }
]
```

Prueba para fecha expiración menor:

La primera imagen muestra el requerimiento para una fecha inferior al 2 de enero del 2025. Se observa que solo hay un producto con la fecha de expiración menor. Si ponemos 8 días más, vemos que esto cambia, agregando 1 producto cuya fecha de expiración es menor a la indicada, demostrando que el filtro funciona.

### Productos por fecha

```
[
  {
    "codBarras": 12,
    "_id": 12,
    "CODBARRAS": 12,
    "NOMBRE": "Aceite",
    "PRECIOVENTA": 132,
    "PRESENTACION": "Lata",
    "UNIDAD_MEDIDA": "Kg",
    "ESP_EMPACADO": "No",
    "FECHA_EXP": "2024-12-23",
    "CATEGORIA": 3
  }
]
```

### Productos por fecha

```
[
  {
    "codBarras": 3,
    "_id": 3,
    "CODBARRAS": 3,
    "NOMBRE": "Huevos",
    "PRECIOVENTA": 76,
    "PRESENTACION": "Botella",
    "UNIDAD_MEDIDA": "L",
    "ESP_EMPACADO": "Si",
    "FECHA_EXP": "2025-01-02",
    "CATEGORIA": 2
  },
  {
    "codBarras": 12,
    "_id": 12,
    "CODBARRAS": 12,
    "NOMBRE": "Aceite",
    "PRECIOVENTA": 132,
    "PRESENTACION": "Lata",
    "UNIDAD_MEDIDA": "Kg",
    "ESP_EMPACADO": "No",
    "FECHA_EXP": "2024-12-23",
    "CATEGORIA": 3
  }
]
```

### Prueba para categoría:

En la primera foto observamos la consulta para los productos de la categoría con id 5, donde se puede observar el resultado con toda la información de los productos, y es evidente que el atributo de categoría es de 5. En la segunda foto tenemos la misma consulta, pero para la categoría de id 4, y observamos el resultado con diferentes productos, todos con su atributo de categoría igual a 4.

Productos por categoría

5

Submit

```
[
  {
    "codBarras": 4,
    "_id": 4,
    "CODBARRAS": 4,
    "NOMBRE": "Pan",
    "PRECIOVENTA": 427,
    "PRESENTACION": "Caja",
    "UNIDAD_MEDIDA": "ml",
    "ESP_EMPACADO": "No",
    "FECHA_EXP": "2025-10-30",
    "CATEGORIA": 5
  },
  {
    "codBarras": 6,
    "_id": 6,
    "CODBARRAS": 6,
    "NOMBRE": "Queso",
    "PRECIOVENTA": 97,
    "PRESENTACION": "Caja",
    "UNIDAD_MEDIDA": "L",
    "ESP_EMPACADO": "Si",
    "FECHA_EXP": "2025-07-10",
    "CATEGORIA": 5
  },
  {
    "codBarras": 22,
    "_id": 22,
    "CODBARRAS": 22,
    "NOMBRE": "Helado",
    "PRECIOVENTA": 289,
    "PRESENTACION": "Caja",
    "UNIDAD_MEDIDA": "Kg",
    "ESP_EMPACADO": "No",
    "FECHA_EXP": "2025-05-15",
    "CATEGORIA": 5
  }
]
```

Productos por categoría

3

Submit

```
[
  {
    "codBarras": 12,
    "_id": 12,
    "CODBARRAS": 12,
    "NOMBRE": "Aceite",
    "PRECIOVENTA": 132,
    "PRESENTACION": "Lata",
    "UNIDAD_MEDIDA": "Kg",
    "ESP_EMPACADO": "No",
    "FECHA_EXP": "2024-12-23",
    "CATEGORIA": 3
  },
  {
    "codBarras": 13,
    "_id": 13,
    "CODBARRAS": 13,
    "NOMBRE": "Galletas",
    "PRECIOVENTA": 395,
    "PRESENTACION": "Botella",
    "UNIDAD_MEDIDA": "ml",
    "ESP_EMPACADO": "No",
    "FECHA_EXP": "2025-02-25",
    "CATEGORIA": 3
  },
  {
    "codBarras": 16,
    "_id": 16,
    "CODBARRAS": 16,
    "NOMBRE": "Refresco",
    "PRECIOVENTA": 24,
    "PRESENTACION": "Tarro",
    "UNIDAD_MEDIDA": "L",
    "ESP_EMPACADO": "No",
    "FECHA_EXP": "2025-06-05",
    "CATEGORIA": 3
  },
  {
    "codBarras": 21,
    "_id": 21,
    "CODBARRAS": 21,
    "NOMBRE": "Yogurt",
    "PRECIOVENTA": 404,
    "PRESENTACION": "Botella",
    "UNIDAD_MEDIDA": "Unidad",
    "ESP_EMPACADO": "No",
    "FECHA_EXP": "2025-06-12",
    "CATEGORIA": 3
  }
]
```

b. RFC2:

En la base de datos, si se observa la sucursal 1, en el atributo Inventario se encuentran varios productos, pero para el escenario de prueba se hará énfasis en el producto con código de barras 4 que está en la sucursal con identificador 6 (llamada 'Bodega Secundaria'). Este es el producto de Pan, y tiene 2 entradas en esta sucursal:

```
▼ 3: Object
  _id : 4
  CODIDIGOBARRAS : 4
  IDBODEGA : 6
  CANTIDAD_OCUPADA : 13
  COSTO_GRUPO_PRODUCTO : 801
  MINIMO_RECOMPRA : 21

▼ 10: Object
  _id : 11
  CODIDIGOBARRAS : 4
  IDBODEGA : 6
  CANTIDAD_OCUPADA : 6
  COSTO_GRUPO_PRODUCTO : 1500
  MINIMO_RECOMPRA : 21
```

Cuando se consulta el estado de los productos del inventario de la sucursal 1, aparece lo siguiente:

```
{
  "_id": {
    "CODIDIGOBARRAS": 4,
    "IDBODEGA": 6
  },
  "nombre_producto": "Pan",
  "totalCantidadOcupada": 19,
  "avgCostoGrupoProducto": 1150.5,
  "minimoRecompra": 21
},
```

Se puede evidenciar que aparece únicamente esta entrada para la combinación codbarras = 4 y idBodega = 6, se han sumado las cantidades de los 2 grupos del mismo producto, y se promedió el costo de almacenamiento.