**Documentación de Requisitos Funcionales (Entrega 2)**

**Sebastian Quevedo – 202223735**

**Daniel Stiven Roa Uribe - 202215803**

**RF1 - Crear una Ciudad**

Se implementó la funcionalidad para crear una ciudad en el sistema. Esta operación incluye el registro del nombre y código de la ciudad. Se probó exitosamente utilizando Postman, realizando una petición en la interfaz hecha con thymeleaf. La respuesta confirma la creación exitosa de la ciudad con un código de estado 201 Created.

**RF2 - Crear una Sucursal**

Se desarrolló la función para crear una sucursal asociada a una ciudad existente. La información básica requerida incluye el nombre de la sucursal y su relación con la ciudad. Se validó el correcto funcionamiento de esta operación en la interfaz hecha con thymeleaf con una petición. Se obtuvo una respuesta 201 Created, asegurando que la sucursal fue creada correctamente.

**RF3 - Crear y Borrar una Bodega**

La implementación permite la creación y eliminación de bodegas, cada una asociada a una sucursal específica. Para la creación, se hace una solicitud POST a http://localhost:8080/superandes/api/bodegas, que requiere el nombre de la bodega y la sucursal asociada. Se probó en Postman, obteniendo una respuesta 201 Created. La eliminación de una bodega se realiza mediante una petición DELETE a http://localhost:8080/superandes/api/bodegas/{id}, confirmando la operación con un código 204 No Content.

**RF4 - Crear y Actualizar Proveedores**

Se habilitó la funcionalidad para crear y actualizar proveedores, capturando información esencial como nombre, dirección y teléfono. La creación se verifica con una solicitud POST a http://localhost:8080/superandes/api/proveedores, resultando en una respuesta 201 Created. La actualización se prueba con una petición PUT a http://localhost:8080/superandes/api/proveedores/{id}, confirmando los cambios con un 200 OK.

**RF5 - Crear y Leer una Categoría de Producto**

Se desarrolló la capacidad para crear y leer categorías de productos. La creación se prueba con una solicitud POST a http://localhost:8080/superandes/api/categorias, que recibe el nombre y código de la categoría, resultando en una respuesta 201 Created. Para la lectura, se hace una petición GET a http://localhost:8080/superandes/api/categorias/{id}, confirmando que se devuelve la información correcta con un 200 OK.

**RF6 - Crear, Leer y Actualizar un Producto**

La funcionalidad para gestionar productos incluye operaciones de creación, lectura y actualización. Al crear un producto, se realiza una solicitud POST a http://localhost:8080/superandes/api/productos, asociándolo a una categoría existente. Se probó en Postman, obteniendo un 201 Created. La lectura se valida con una solicitud GET a [http://localhost:8080/superandes/api/productos/{id}](http://localhost:8080/superandes/api/productos/%7bid%7d) y la actualización se prueba con una petición PUT a /productos/{codigo}, con respuesta 200 OK al confirmar los cambios.

**RF7 - Crear una Orden de Compra para una Sucursal**

Se implementó la creación de órdenes de compra, capturando toda la información requerida, incluyendo la fecha de creación, sucursal, proveedor y fecha esperada de entrega. Esta operación se prueba con una solicitud POST a http://localhost:8080/superandes/api/ordenes\_compra, que devuelve 201 Created. La estructura de la orden se valida en la respuesta, confirmando que todos los datos están correctamente registrados.

**RF8 - Actualizar una Orden de Compra Cambiando su Estado a Anulada**

La funcionalidad permite anular una orden de compra existente mediante la actualización de su estado. Se verifica esta operación con una solicitud PUT a http://localhost:8080/superandes/api/ordenes\_compra/{id}, obteniendo un 200 OK al confirmar que el estado se ha actualizado a "anulada".

**RF9 - Mostrar Todas las Órdenes de Compra**

Se desarrolló la funcionalidad para listar todas las órdenes de compra, mostrando información detallada y su estado actual. Esta operación se prueba con una solicitud GET a /ordenes\_compra, que devuelve una lista de órdenes de compra con un código de estado 200 OK.

**RF10 - Registrar Ingreso de Productos a la Bodega**

Se implementó la funcionalidad para registrar el ingreso de productos a una bodega en el sistema. Este ingreso utiliza un documento que se basa en una orden de compra existente y actualiza la información de inventario de la bodega, así como el costo promedio de cada producto. La creación del ingreso se realiza mediante una solicitud POST a /api/ingresoProductos/crear, pasando como parámetros el idOrdenCompra y el idBodega. Al completarse la operación, el estado de la orden de compra se actualiza a "ENTREGADA" si todos los pasos son exitosos. Se probó en Postman, obteniendo una respuesta 201 Created.

**RFC1 – Mostrar porcentaje ocupación de un producto**

Este requerimiento de consulta se encarga de contar todos los productos que existen bajo el mismo nombre, esto con la finalidad de comparar la cantidad de existencias contra el espacio disponible de cada bodega de una sucursal especificada. Como resultado, se puede ver el nombre de cada bodega con el porcentaje de ocupación que tiene el objeto especificado. El requerimiento tiene en cuenta que es posible que no se encuentren los productos o las bodegas, anulando la división para que no de un dato indeterminado.

**RFC2 – Mostrar productos que cumplen con cierta característica**

El método se encarga de recibir los parámetros de: precioMayor, precioMenor, fechaVencimiento, sucursalID y categoriaID. Esto con el fin de dar la indicación al query de mostrar la información de los productos que cumplen con esas características.

**RFC3 – Mostrar productos disponibles en bodega y sucursal**

Como entradas, se requieren las id de la sucursal y la bodega la cual se desea conocer el inventario de productos disponibles. El método se encarga de contar la cantidad de veces que aparecen los productos para poder agruparlos por su nombre y dar una cuenta de existencias. Por otro lado, este también se encarga de realizar las sumas y divisiones pertinentes para hallar el costo promedio y la cantidad mínima antes de pedir una nueva orden.

**RFC4 – Mostrar productos disponibles en bodega y sucursal**

Como entradas, se requieren las id de la sucursal y la bodega la cual se desea conocer el inventario de productos disponibles. El método se encarga de contar la cantidad de veces que aparecen los productos para poder agruparlos por su nombre y dar una cuenta de existencias. Por otro lado, este también se encarga de realizar las sumas y divisiones pertinentes para hallar el costo promedio y la cantidad mínima antes de pedir una nueva orden.

RFC5 - Mostrar Productos que Requieren una Orden de Compra

Este requerimiento permite obtener una lista de productos que han alcanzado un nivel de existencias bajo en la bodega y que, por lo tanto, requieren una nueva orden de compra. La consulta verifica el inventario de cada bodega y compara la cantidad de productos disponibles contra el nivel mínimo de reorden establecido. Los resultados muestran el nombre e identificador del producto, la bodega y sucursal donde se requiere la orden, los proveedores previos y la cantidad actual. La consulta se realiza con una solicitud GET a /productos/reorden, devolviendo un 200 OK y la lista de productos que necesitan reabastecimiento. Se probó en Postman, confirmando el funcionamiento esperado

RFC6 - Consulta de Documentos de Ingreso de Productos a Bodega (Nivel de Aislamiento SERIALIZABLE)

Esta consulta permite obtener los documentos de ingreso de productos de los últimos 30 días en una bodega específica, asociada a una sucursal. Implementa un nivel de aislamiento SERIALIZABLE, lo cual bloquea otras operaciones que puedan afectar los datos consultados hasta que esta transacción termine. La consulta se realiza con una solicitud GET a /serializable/{idSucursal}/{idBodega}, con un temporizador de 30 segundos para simular concurrencia. La operación devuelve un listado de documentos de ingreso de productos junto con los datos del proveedor y de la sucursal. Se probó en Postman y obtuvo un 200 OK con los resultados esperados.

RFC7 - Consulta de Documentos de Ingreso de Productos a Bodega (Nivel de Aislamiento READ\_COMMITTED)

Este requerimiento permite obtener los documentos de ingreso de productos de una bodega específica en los últimos 30 días, utilizando un nivel de aislamiento READ\_COMMITTED. Este nivel permite que otras operaciones puedan modificar los datos durante la consulta, reflejando los cambios en tiempo real. La consulta se realiza con una solicitud GET a /read-committed/{idSucursal}/{idBodega} y un temporizador de 30 segundos para pruebas de concurrencia. Devuelve un listado de documentos con información sobre el proveedor, la sucursal, y la bodega. Se probó en Postman, obteniendo un 200 OK y resultados consistentes en cada consulta.

Prueba de concurrencia 1:

Pasos para la Ejecución Concurrente de RFC6 y RF10:

* Paso 1: Ejecutar RFC6 mediante una solicitud GET a /serializable/{idSucursal}/{idBodega}. Esta consulta utiliza el nivel de aislamiento SERIALIZABLE y un temporizador de 30 segundos para asegurar que la consulta mantenga un bloqueo que impida cambios concurrentes en los datos.
* Paso 2: Dentro de los 30 segundos de ejecución de RFC6, ejecutar RF10 mediante una solicitud POST a /api/ingresoProductos/crear, indicando idOrdenCompra e idBodega. Esta operación intenta registrar un ingreso de producto mientras RFC6 está en curso.

Descripción de lo Sucedido:

* Al ejecutar RF10 mientras RFC6 se encuentra en proceso, el sistema debería de imponer un bloqueo debido al nivel de aislamiento SERIALIZABLE. Dando como resultado que el RF10 deberia esperar a que RFC6 complete su consulta antes de proceder con el registro de ingreso de productos. Extrañamente, postman deja hacer los dos procesos a pesar de especificar el nivel de asilamiento.

Resultado Presentado por RFC6:

* La consulta de RFC6 finaliza incluyendo el ingreso de producto que registro el RF10.

Prueba de concurrencia 2:

Pasos para la Ejecución Concurrente de RFC7 y RF10:

* Paso 1: Ejecutar RFC7 mediante una solicitud GET a /read-committed/{idSucursal}/{idBodega}. Esta consulta emplea el nivel de aislamiento READ\_COMMITTED y mantiene un temporizador de 30 segundos, permitiendo que otras operaciones de modificación puedan interactuar con los datos durante su ejecución.
* Paso 2: Dentro de los 30 segundos de ejecución de RFC7, lanzar RF10 mediante una solicitud POST a /api/ingresoProductos/crear, proporcionando idOrdenCompra e idBodega. Esta operación intenta registrar un ingreso de producto mientras RFC7 está en proceso.

Descripción de lo Sucedido:

* En este caso, RF10 puede ejecutarse y registrar el ingreso de productos sin esperar a que RFC7 termine, debido a que el nivel de aislamiento READ\_COMMITTED permite modificaciones concurrentes. Esto permite que RFC7 lea cambios que ocurran en tiempo real.

Resultado Presentado por RFC7:

* La consulta de RFC7 puede mostrar el documento de ingreso registrado por RF10, dependiendo del momento exacto en que se realice la lectura. Dado que READ\_COMMITTED permite lecturas no repetibles, es posible que el resultado de RFC7 incluya el nuevo ingreso, mostrando así la actualización en tiempo real.