

PROYECTO 1.1 — SISTEMAS TRANSACCIONALES

1) Entidades de negocio y reglas de negocio.

Las entidades de negocio son las sucursales, los proveedores, las bodegas, los productos y las categorías a las que dichos productos pertenecen.

Las reglas de negocio identificadas son:

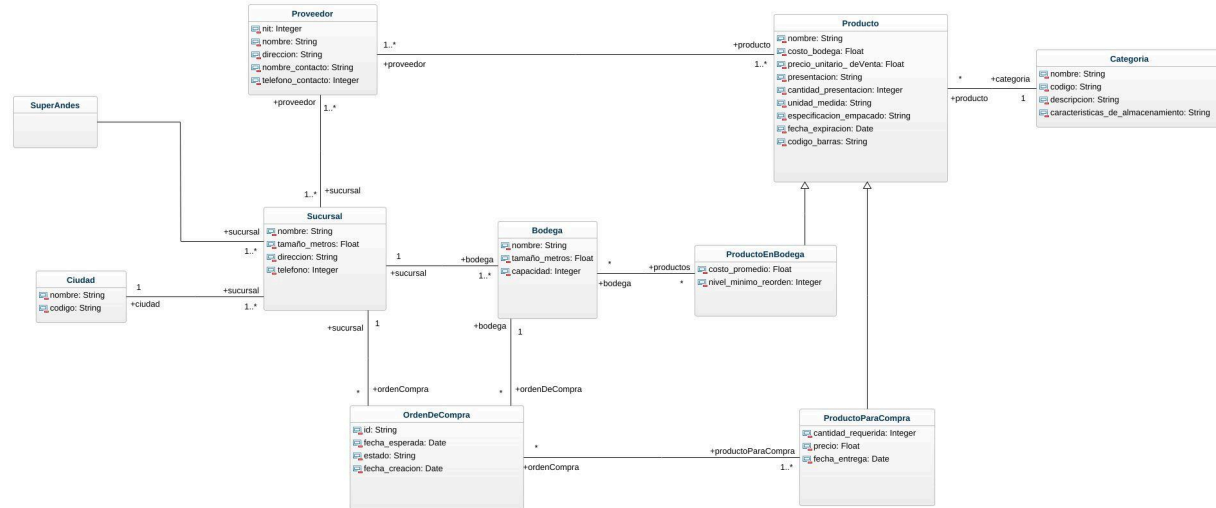
- Puede haber múltiples sucursales por ciudad, y cada sucursal tiene una bodega propia.
- Si se alcanza el nivel de recompra se pone una orden de compra con el proveedor.
- El costo promedio por producto se recalcula cada vez que se surte con nuevas unidades del producto.

2) Análisis y modelo conceptual

a) Modelo conceptual UML de SuperAndes

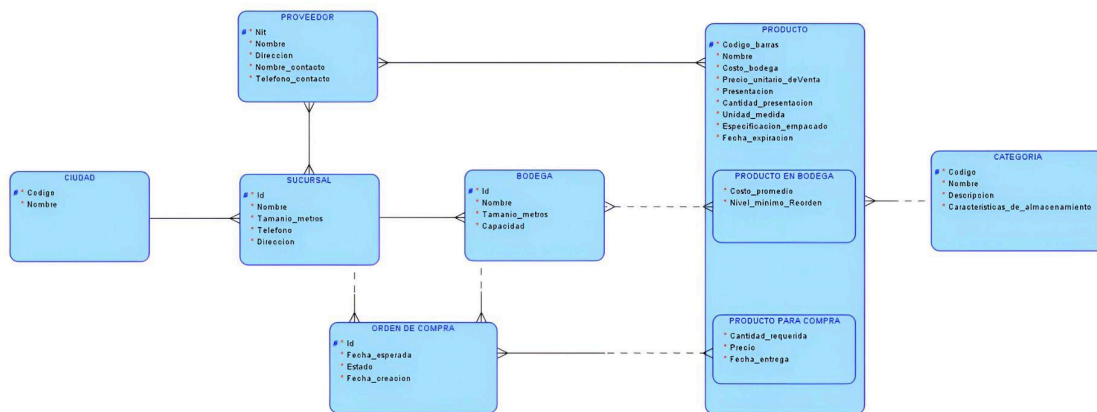
Dentro de nuestro diseño determinamos que en lugar de hacer una clase para recepción de compra, optamos por dejar la orden de compra con un estado que puede ser “recibido” para emular una recepción de compra. De esta manera reducimos el número de relaciones finales manteniendo genericidad en el código.

Por otro lado, somos conscientes de que la orden de compra no tiene un proveedor asignado directamente, sin embargo se puede encontrar por medio de las conexiones producto y posteriormente proveedor.



Modelo UML - SuperAndes. Para ver con más detalle se encontrará en la carpeta Modelos en GitHub.

b) Modelo conceptual E/R en DataModeler



Modelo conceptual E/R - Para ver con más detalle: Carpeta Modelos en GitHub.

3) Diseño de la base de datos

c) Modelo de datos relacional correspondiente al modelo conceptual UML propuesto

Modelo de datos relacional

Justificación con algoritmo de Chen

Para transformar el modelo UML al modelo de datos relacional nos basamos en el algoritmo de Chen, que se compone de dos pasos:

1. Identificar tablas básicas

Aquí se tiene en cuenta que toda clase independiente es candidata a ser una relación, es decir, una tabla. Además, se deben elegir los atributos y determinar la PK de cada una de las tablas.

Por lo anterior, es que cada una de las clases presentes en el UML se convirtieron en una tabla, excepto por la clase SuperAndes, ya que al ser un nodo raíz esta no cuenta con atributos propios, por lo que no es necesario que esté presente en el modelo relacional.

Las relaciones presentes en el modelo relacional son: Bodega, Sucursal, Proveedor, Producto, Orden de compra, Ciudad, Categoría, Producto en Bodega y Producto Para Compra.

La elección de la Primary Key para cada una de las tablas está basada en cuál atributo permite reconocer a cualquier elemento perteneciente a esa tabla. Por lo tanto, en relaciones como Sucursal, Bodega y Orden de compra sus PK son sus respectivos id's; en estos casos son únicos e irrepetibles.

Por otro lado, entidades como Ciudad y Categoría tienen como PK un código, debido a que, ninguno de sus atributos permite que sea totalmente diferenciado de los demás. En cuanto a Proveedor, su PK es el NIT, debido a que no existen duplicados de un mismo NIT. Finalmente, con las relaciones Producto, Producto Para Compra y Producto en Bodega, todos cuentan con la misma Primary Key, pues estos últimos dos mencionados son hijos de la clase Producto, la cuál se identifica con su Código de Barras.

2. Asociaciones

2.1. Identificar asociaciones

Este paso tiene como objetivo identificar los roles, la cardinalidad, la obligatoriedad y la direccionalidad entre dos relaciones o tablas, además de definir si hay dependencias y cuáles clases de asociación están presentes en el modelo.

Para cumplir con los objetivos anteriormente planteados, se analizará cada una de las asociaciones propuestas en el modelo UML.

- **Ciudad - Sucursal**

Esta asociación es de 1 a muchos, ya que una sucursal solo puede estar en una ciudad, mientras que en una ciudad puede haber muchas sucursales. Además, es obligatorio por ambas partes porque si una sucursal no tuviera una ciudad en la cuál está presente significa que no existe y si una ciudad no tiene sucursales, no debería esta presente en la base de datos. Esta asociación es bidireccional, pues tanto la ciudad tiene como atributos las sucursales y la sucursal tiene como atributo la ciudad. Por otro lado, en esta asociación no hay dependencias presentes.

Cardinalidad: 1 a muchos.

Obligatoriedad: Obligatorio por ambas entidades.

Asociación: Asociación bidireccional.

- **Sucursal - Bodega**

La cardinalidad en esta asociación es de 1 a muchos debido a que, una sucursal puede tener de una a muchas bodegas, mientras que una bodega, solo puede pertenecer a una sucursal. Con respecto a la obligatoriedad, es obligatorio por ambas partes, ya que si existe una bodega significa que tiene una sucursal asociada a la cuál pertenece y si existen sucursales deben tener por lo menos una bodega en la cuál almacenar los productos. Esta asociación es bidireccional, puesto que la bodega tiene como atributo una sucursal y las sucursales tienen como atributos las bodegas que poseen. Además, en esta asociación no hay dependencias presentes.

Cardinalidad: 1 a muchos.

Obligatoriedad: Obligatorio por ambas entidades.

Asociación: Asociación bidireccional.

- **Sucursal - Proveedor**

En esta asociación hay una cardinalidad presente de muchos a muchos dado que una sucursal puede tener muchos proveedores y así mismo un proveedor puede distribuir sus productos a muchas sucursales. Además, es obligatoria por ambas entidades, puesto que un proveedor para estar presente en la base de datos debe por lo menos surtir en una sucursal y una sucursal para poder estar en funcionamiento y tener productos debe contar con al menos un proveedor. Esta asociación es bidireccional, ya que ambas entidades tienen como atributo a la otra. Por otro lado, no hay dependencias en esta asociación, porque ninguna entidad depende de la otra para existir.

Cardinalidad: Muchos a muchos.

Obligatoriedad: Obligatorio por ambas entidades.

Asociación: Asociación bidireccional.

- **Sucursal - Orden de compra**

La cardinalidad presente es de 1 a muchos porque una orden de compra solo puede ser hecha por una única sucursal (puede que dos o más sucursales pidan los mismos productos a la misma hora para la misma fecha, sin embargo son órdenes de compra diferentes), mientras que una sucursal puede hacer múltiples órdenes de compra para poder abastecerse. Esta asociación es obligatoria de parte de la entidad Orden de Compra, ya que para que una orden existiese debe haber sido solicitada por una sucursal, mientras que puede que exista una Sucursal nueva y por lo tanto todavía tiene productos disponibles y no ha tenido la necesidad de hacer una orden de compra. Por otro lado, no hay dependencias y es una asociación bidireccional, debido a que ambas entidades tienen como atributo a la otra.

Cardinalidad: 1 a muchos.

Obligatoriedad: Obligatorio de parte de la entidad Orden de compra, no obligatorio de parte de la entidad Sucursal.

Asociación: Asociación bidireccional.

- **Orden de compra - Bodega**

La cardinalidad de esta asociación es de 1 a muchos, pues lo que se solicita en una orden de compra es entregado a una única bodega, mientras que una bodega puede tener múltiples ingresos de productos realizados a través de una orden de compra. Por otro lado, hay

obligatoriedad por parte de la entidad Orden de Compra, puesto que si se crea una orden el objetivo es que llegue a una bodega, sin embargo no se presenta obligatoriedad por parte de Bodega, ya que, puede que existan bodegas nuevas que nunca hayan sido receptoras de productos solicitados a través de una orden. Además, no presenta dependencia y es una asociación bidireccional.

Se hace necesario resaltar que esta asociación debe existir, pues a pesar de que la orden de compra esté relacionada con la sucursal y la sucursal con la bodega, una sucursal puede tener múltiples bodegas, por lo cuál no se sabría a cuál hacer la entrega de la orden. Por tanto, al crear esta asociación se deja claro cuál bodega recibirá los productos.

Cardinalidad: 1 a muchos.

Obligatoriedad: Obligatorio de parte de la entidad Orden de compra, no obligatorio de parte de la entidad Bodega.

Asociación: Asociación bidireccional.

- **Producto - Proveedor**

Esta asociación es de muchos a muchos, debido a que un producto puede tener varios proveedores y un proveedor puede distribuir varios productos. Así mismo, es una asociación obligatoria por ambos lados, ya que un producto debe tener al menos un proveedor y un proveedor debe al menos tener un producto que ofrecer, porque sino no estaría presente en la lista de proveedores. Por otro lado, no existe una relación de dependencia entre estas entidades pero si hay una asociación bidireccional.

Cardinalidad: Muchos a muchos.

Obligatoriedad: Obligatorio por ambas partes.

Asociación: Asociación bidireccional.

- **Producto - Categoría**

Esta asociación es de 1 a muchos, una categoría puede tener muchos productos que pertenezca a ella, mientras que un producto debe pertenecer únicamente a una sola categoría. Por otro lado, esta asociación es obligatoria de parte de la entidad Producto, porque un producto si o si debe tener una categoría, mientras que no es obligatorio por parte de la entidad Categoría, ya que es posible que existan categorías pero que en ese momento ningún producto pertenezca a ellas.

No hay dependencia entre las entidades pero si hay una asociación bidireccional.

Cardinalidad: 1 a muchos.

Obligatoriedad: Obligatorio por parte de Producto pero no por parte de Categoría.

Asociación: Asociación bidireccional.

- **ProductoParaCompra - OrdenDeCompra**

Esta asociación es de muchos a muchos, una Orden de Compra puede contar con muchos productos ordenados y un producto puede ser solicitado a través de múltiples órdenes de compra. Por otro lado, la asociación es obligatoria de parte de la Orden, ya que si se crea una orden es con el fin de obtener uno o más productos, sin embargo no es obligatorio de parte del Producto pues pueden existir productos que nunca hayan sido solicitados. No hay dependencia entre las entidades pero si hay una asociación bidireccional.

Cardinalidad: Muchos a muchos.

Obligatoriedad: Obligatorio por parte de Orden de Compra pero no por parte de Producto.

Asociación: Asociación bidireccional.

- **ProductoEnBodega - Bodega**

Esta asociación es de muchos a muchos, porque un producto puede estar en varias bodegas y una bodega puede tener múltiples productos. Esta asociación no es obligatoria por ninguna entidad, dado que pueden existir productos que no se encuentren en ninguna bodega y puede existir una bodega nueva que aún no tenga productos. No hay dependencia entre las entidades pero si hay una asociación bidireccional.

Cardinalidad: Muchos a muchos.

Obligatoriedad: No obligatorio por ambas entidades.

Asociación: Asociación bidireccional.

Herencias

- **Producto - ProductoParaCompra**

ProductoParaCompra hereda de la entidad Producto, esta entidad se creó con el fin de que exista un producto con los mismos atributos que cualquier producto podría tener pero que también tenga otros atributos que le permitan relacionarse a la entidad Orden de Compra.

- **Producto - ProductoEnBodega**

ProductoEnBodega hereda de la entidad Producto, esta entidad se creó con el fin de que exista un producto con los mismos atributos que cualquier producto podría tener pero que también tenga otros atributos que le permitan relacionarse a la entidad Bodega.

2.2 Conversión de asociaciones

- **Ciudad - Sucursal**

Al ser esta una relación de 1 a muchos se coloca la PK de Ciudad como atributo en la tabla de Sucursal porque Ciudad es el extremo univalue y así se asegura que no existan datos repetidos y la información se encuentra organizada de una manera que no haya espacios vacíos en las tablas.

- **Sucursal - Bodega**

Como esta asociación es de 1 a muchos se coloca la PK del extremo univalue como atributo en la tabla del extremo multivalue, es decir, se coloca la PK de Sucursal como un atributo en la tabla de Bodega.

- **Sucursal - Proveedor**

Esta es una asociación de muchos a muchos, por lo cual para mostrar la relación en el modelo relacional, se crea una nueva tabla que represente dicha asociación. En este caso se creó una nueva tabla llamada ProveedorDeSucursal la cual tiene como atributos la PK de Sucursal y la PK del Proveedor.

- **Sucursal - Orden de compra**

La asociación entre estas entidades es de 1 a muchos, por lo cual al ser Sucursal el extremo univalue se coloca su PK como atributo en la tabla de Orden de Compra, que es el extremo multivalue.

- **Orden de Compra - Bodega**

La asociación presente es de 1 a muchos, por lo cuál al ser Bodega el extremo univalor se coloca su PK como atributo en la tabla de Orden de Compra, que es el extremo multivalor.

- **Producto - Proveedor**

La asociación entre estas entidades es de muchos a muchos, por lo cuál se generó una nueva tabla llamada ProveedorDeProducto que tiene como atributos la PK de Proveedor y la PK del Producto relacionado.

- **Producto - Categoría**

Esta asociación es de 1 a muchos, por lo cuál en la tabla de de Producto se añade como atributo la PK de categoría, que es el extremo univalor. Además, se añade la categoría al producto porque para Producto la asociación es obligatoria de parte de Producto pero no de Categoría.

- **ProductoParaCompra - OrdenDeCompra**

Esta asociación es de muchos a muchos, por lo cuál se creó una nueva tabla llamada ProductoSolicitado que tiene como atributos las PK's de ProductoParaCompra y OrdenDeCompra.

- **ProductoEnBodega - Bodega**

Esta asociación es de muchos a muchos, así que para su conversión se creó una nueva tabla llamada ProductoAlmacenado que tuviera como atributos las PK's de ProductoEnBodega y Bodega.

Asociaciones tácitas o indirectas

Este apartado está dedicado a explicar aquellas asociaciones que existen en el contexto pero que no están directamente plasmadas en los modelos pues se presentan a través de una conexión indirecta.

- **Sucursal - Producto**

Según el contexto planteado una sucursal debería tener una asociación con productos, sin embargo esta relación existe a través de la entidad Bodega, debido a que una Sucursal tiene bodegas y esas bodegas contienen los productos pertenecientes a las sucursales.

- **Bodega - Proveedor**

Esta asociación indirecta se hace por medio de la entidad Producto por que una bodega tiene productos y estos tienen proveedores, así que para

conocer quiénes son los proveedores de una bodega se puede saber mediante los productos presentes en esa bodega. Así mismo esta asociación también existe a través de la entidad Sucursal, dado que una bodega pertenece a una Sucursal y una Sucursal tiene proveedores.

- **Orden de compra - Proveedor**

Para que una orden de compra se pueda llevar a cabo debe tener proveedores, sin embargo una orden de compra tiene productos y cada uno de estos tiene sus proveedores, por lo cuál si se necesita saber quiénes son los proveedores que cumplirán con la orden asignada se puede encontrar gracias a los productos solicitados.

d) Nivel de normalización del modelo

El nivel de normalización de todas las relaciones del modelo es BCNF. Esto se comprobó con cada una de las relaciones. A continuación se muestran las condiciones que se revisaron individualmente:

1NF	✓
Cada celda contiene un solo valor	✓
Los nombres de las columnas son unicos	✓
2NF	✓
Cumple con 1NF	✓
Cada atributo depende de toda la llave	✓
No hay dependencias parciales	✓
3NF	✓
Cumple con 2NF	✓
No hay dependencias transitivas	✓
BCNF	✓
Cumple con 3NF	✓
Las llaves son simples	✓

Cada relación tiene nombres de atributo únicos y los valores de las celdas son atómicos por lo que tienen un solo valor. Con esto se aseguro que pasara los criterios para 1NF. Además, todas las tablas propuestas tienen solo una llave primaria por lo que no se generan dependencias parciales. Además, cuidamos que los atributos dependieran de la

llave y de lo contrario revisamos si era un tema de falta de coherencia en el modelo. Al hacer esto aseguramos 2NF, después revisamos todas las conexiones y aseguramos que no había dependencias transitivas. Por último, se alcanzó el estatus de BCNF, pues como ya se mencionó antes todas las relaciones tienen PK única (simple).

4) Escenarios de Prueba

RF1 – Crear una ciudad

ID de la prueba	RF01
Descripción	Verificar que se puede crear una ciudad con nombre y código único
Restricciones	La ciudad no puede existir anteriormente en la base de datos y su código debe ser unitario
Pasos	Ejecutar la operación de creación de la ciudad con nombre “Bogotá” y código único “C601”
Resultado esperado	Se integra la ciudad a la base de datos con su identificador: “Código”
Caso negativo	No se puede crear la nueva ciudad debido a que existe otra ciudad con el mismo código y nombre (Caso repetido)

RF2 – Crear una sucursal

ID de la prueba	RF02
Descripción	Verificar que se puede crear una sucursal en una ciudad existente.
Restricciones	La sucursal integrada solo debe de tener una ciudad asociada existente junto con su nombre, tamaño, dirección y teléfono

Pasos	Ejecutar la operación de creación de la sucursal con nombre: “Sede Colina”, Tamaño: “1500”, Dirección: Cra. 58d #146-51 y teléfono: “326 5177 145”
Resultado esperado	Se integra la sucursal a la base de datos con todos sus datos sin ser NN y su PK: “Nombre”
Caso Negativo	No se crea la Sucursal ya que está asociada en 2 ciudades: “Bogotá”, “Bucaramanga”. (Solo puede estar asociada a una ciudad)

RF3 – Crear y borrar una bodega

ID de la prueba	RF03
Descripción	Verificar que se puede crear y borrar una bodega asociada a una sucursal existente
Restricciones	La bodega solo debe de tener una sucursal asociada debe existir en la base de datos
Pasos	<p>Ejecutar la creación de la bodega asociada a “Sede colina” junto con sus datos.</p> <p>Eliminar la bodega requerida junto con el dato de su sucursal asociada</p>
Resultado esperado	Se integra o elimina la bodega junto con todos sus datos actualizando los datos de la sucursal asociada y su PK: “nombre”
Caso negativo	La bodega está asociada a 2 sucursales por lo que no se puede crear (Sólo puede estar asociada a una sucursal), la bodega que se quiere eliminar no existe.

RF4 – Crear y actualizar proveedores

ID de la prueba	RF04
Descripción	Verificar que se puede crear y actualizar la información de un proveedor en una sucursal existente.
Restricciones	El proveedor debe tener al menos una bodega o sucursal asociada y su identificador: "NIT"
Pasos	Ejecutar la creación del proveedor con toda la información y su sucursal asociada. Al momento de actualizar la información se debe tener en cuenta su NIT y si se cambia también se tiene que actualizar en "ProveedorSucursal"
Resultado esperado	Se integra o actualiza el proveedor a la base de datos con todos sus datos y su PK: "NIT"
Resultado negativo	El proveedor no tiene una sucursal o bodega asociada (debe tener por lo menos una), El proveedor no se puede crear o actualizar por NIT repetido (El NIT es clave identificadora única)

RF5 – Crear y leer una categoría de producto

ID de la prueba	RF05
Descripción	Verificar que se puede crear una categoría de producto y leer su información a partir de su código
Restricciones	Cada categoría debe tener un código único siendo este su PK
Pasos	Crear la categoría "Perecedero" junto con su información dada y su código: "CA789".

	Al leer la categoría deberá hacerse por su nombre o código siendo este último el más pertinente.
Resultado esperado	Se crea la categoría y al leerse deberá filtrarse por el código o nombre y se podrá recorrer toda la información de este
Resultado Negativo	La categoría no se puede crear porque su código identificador no es único. No se puede leer la categoría debido a que no existe (Categoría/producto inexistente)

RF6 – Crear. Leer y actualizar un producto

ID de la prueba	RF06
Descripción	Verificar que se puede crear, leer y actualizar la información de un producto en la base de datos
Restricciones	El producto debe tener asociado una categoría existente en la bases de datos y un código de barras el cual es su identificador
Pasos	Crear un producto junto con su información, Codigos_barras y categoría. Entrar a la base de datos con el código del producto para leer toda la información y actualizar
Resultado esperado	Se crea el producto y al leerse deberá filtrarse por el código y se podrá leer y actualizar la información
Resultado Negativo	El producto no puede ser creado debido a que no tiene un código de barras único ni categoría existente (Identificadores)

RF7 – Crear una orden de compra para una sucursal

ID de la prueba	RF07
Descripción	Verificar que se puede crear una orden de compra para una sucursal
Restricciones	La orden de compra solo puede estar relacionada a una sucursal(Bodega), la fecha de creación de la orden no puede ser de una fecha pasada al igual que la fecha de entrega.
Pasos	Crear la orden de compra con toda la información requerida para esta con el detalle de los productos, su código y finalmente el estado de la compra
Resultado esperado	Se crea la orden con toda la información y detalle de los productos y su estado “Vigente”
Resultado Negativo	Orden de compra no válida por incoherencias en la fecha de creación (Fecha pasada). Orden de compra no válida por tener más de una sucursal asociada (Sólo puede tener 1).

RF8 – Actualizar una orden de compra cambiando su estado a anulada

ID de la prueba	RF08
Descripción	Verificar la actualización del estado de una orden de compra de “Vigente” a “Anulada”
Restricciones	La orden de compra debe estar “Vigente”, una orden “Entregada” no puede ser anulada

Pasos	En base al “ID” de la orden de compra se busca en la base de datos para actualizar su estado y guardarse en esta
Resultado esperado	Se cambia el estado de la orden de compra dejándola como “Anulada”
Resultado negativo	Orden inexistente, el ID de la orden de compra no existe. Orden Entregada, no se puede anular una orden de entrega ya finalizada

RF9 – Mostrar todas las órdenes de compra

ID de la prueba	RF09
Descripción	Verificar que se puede listar todas las órdenes de compra de la empresa
Restricciones	Las órdenes deben estar vigentes
Pasos	Ingresar a la sección de ordenes de compra de la base de datos y listados con toda su información siendo su PK:”ID”
Resultado esperado	Se entrega una lista de todas las órdenes de compra vigentes de la empresa junto con su información y detalles de la sucursal relacionada
Resultado Negativo	Caso vacío, no existen órdenes de compra actuales por lo que se entrega una lista vacía.

RFC1 - Mostrar el índice de ocupación de cada una de las bodegas de la sucursal

ID de la prueba	RFC01
Descripción	Calcular y mostrar el índice de ocupación (volumen ocupado / capacidad total) de cada bodega para un conjunto de productos específicos en una sucursal.
Restricciones	Debe existir al menos una sucursal en la base de datos con una o más bodegas.
Pasos	Ingresa a la sucursal la lista de productos para calcular el porcentaje de ocupación de estos
Resultado esperado	Se muestra el porcentaje de ocupación de cada bodega para los productos seleccionados, calculado como $(\text{volumen ocupado} / \text{capacidad de la bodega}) * 100$
Resultado Negativo	Entrega vacía, No existen bodegas en la base de datos de SuperAndes por lo que se entrega vacío

RFC2 - Mostrar el índice de ocupación de cada una de las bodegas de la sucursal

ID de la prueba	RFC02
Descripción	Mostrar productos que cumplen las características dadas por el usuario como rango de precio, fecha de vencimiento, disponibilidad en una sucursal específica, o pertenencia a una categoría.
Restricciones	Deben existir productos en la base de datos con diferentes precios, fechas de vencimiento, y categorías asociadas.
Pasos	Cuando el usuario ingresa los criterios de búsqueda, se ejecuta la consulta para obtener

	los productos que cumplen con las restricciones dadas identificándose con su código de barras.
Resultado esperado	Se muestran todos los productos que cumplen con las características especificadas, mostrando toda la información del producto.
Resultado Negativo	Resultado vacío, Se entrega una lista vacía debido a que no existen productos en la base de datos

RFC3 - Mostrar el índice de ocupación de cada una de las bodegas de la sucursal

ID de la prueba	RFC03
Descripción	Mostrar la lista de productos disponibles en una bodega específica de una sucursal, con detalles de cantidad actual, cantidad mínima requerida, y costo promedio.
Restricciones	Debe existir productos almacenados en la bodega mencionada con su cantidad actual y mínima establecida.
Pasos	Cuando el usuario ingresa la sucursal y bodega a buscar se genera la consulta para obtener el inventario de productos en la bodega seleccionada.
Resultado esperado	Se muestra la lista de productos con la cantidad actual, cantidad mínima requerida, y costo promedio en la bodega especificada.
Resultado Negativo	Entrega vacía, no existen productos registrados en la base de datos por lo que la bodega no tiene índice de ocupación

RFC4 - Mostrar las sucursales en las que hay disponibilidad de un producto

ID de la prueba	RFC04
Descripción	Mostrar la lista de productos disponibles donde un producto específico está disponible en alguna de sus bodegas
Restricciones	El producto debe estar registrado en la base de datos y disponible en al menos una bodega.
Pasos	Cuando el usuario ingresa el identificador del producto se ejecuta la consulta para listar las sucursales con disponibilidad del producto.
Resultado esperado	Se muestra una lista de todas las sucursales donde el producto está disponible en alguna de sus bodegas
Resultado Negativo	Entrega vacía, No existe el producto en ninguna sucursal registrada en la base de datos por lo que se entrega vacío

RFC5 - Mostrar los productos que requieren de una orden de compra

ID de la prueba	RFC05
Descripción	Mostrar la lista de todos los productos para los que se debe generar una nueva orden de compra debido a que sus cantidades están por debajo del nivel mínimo de reorden en alguna bodega.
Restricciones	Deben existir productos con cantidades por debajo del nivel mínimo en al menos una bodega.
Pasos	Cuando se ejecuta la consulta el programa debe listar los productos con su nombre,

Juan Esteban Ríos González — 202221404

Alejandra Fruto Osorio — 202311287

Santiago Quevedo — 202312920

	identificador, bodega, sucursales asociadas, posibles proveedores, y cantidad actual.
Resultado esperado	El programa muestra una lista de todos los productos que necesitan reabastecerse incluyendo los detalles del producto
Resultado Negativo	No existen productos con necesidad de orden de compra en la base de datos por lo que se entrega una lista vacía.