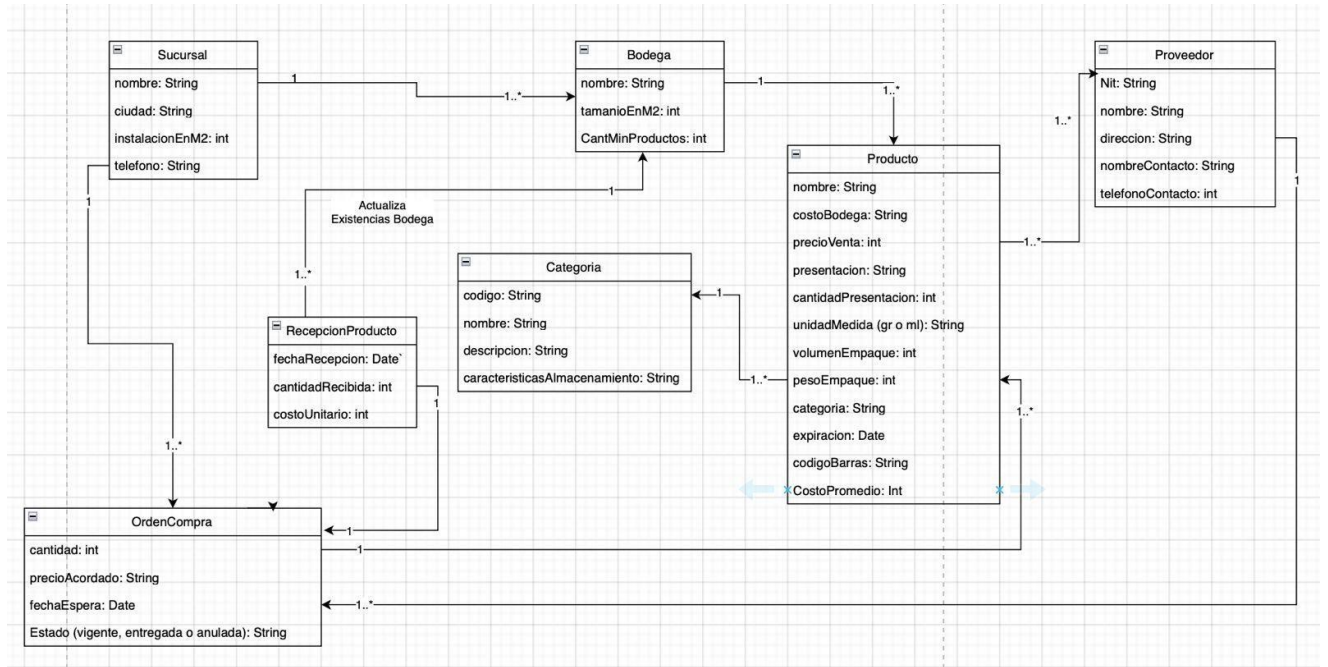
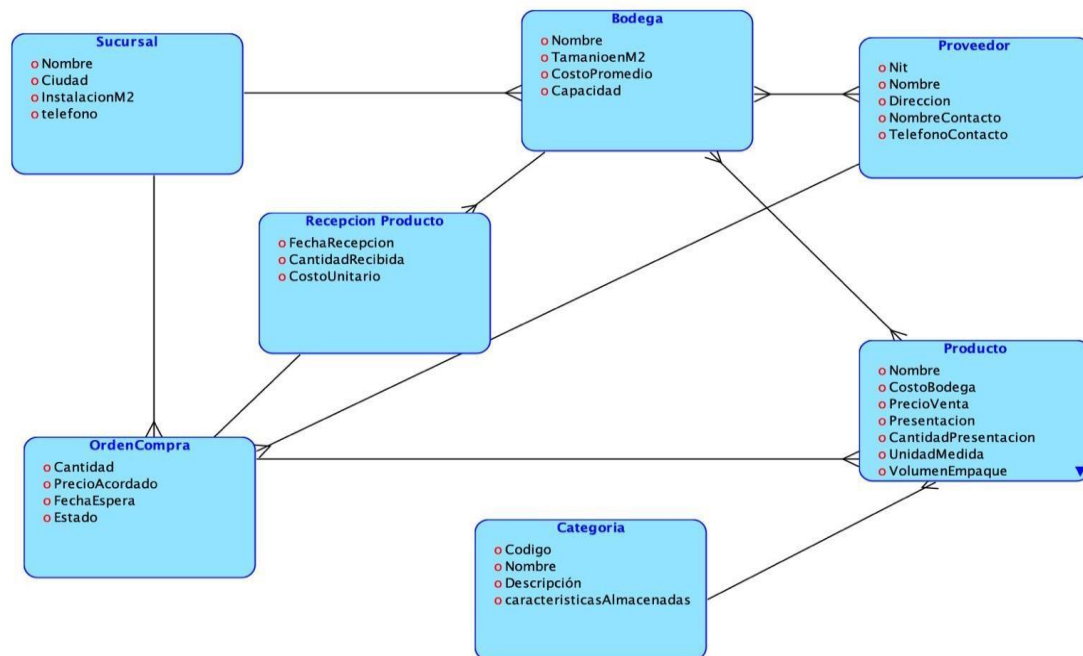


Proyecto 1 – Sistrans

UML:



Modelo E/R



Proceso de transformación de UML a relacional

Sucursal				
id_sucursal	nombre	Ciudad	instalacionEnM2	telefono
PK, NN, SA	NN, CK	NN	NN	NN

Bodega					
id_bodega	Nombre	tamanoEnM2	costoPromedio	capacidad	id_sucursal
PK, SA, NN	NN, CK	NN	N	NN	FKsucursalid_sucursal, NN, ND

Producto											
id_producto	nombre	costoBodega	precioVenta	presentacion	cantidadPresentacion	unidadMedida	volumenEmpaque	pesoEmpaque	expiracion	codigoBarras	id_categoria
PK, SA, NN	NN, CK	NN	NN	NN	NN	NN	NN	NN	NN	NN, ND, CK	FKcategoriaid_categoria, NN, ND

Categoria				
id_categoria	codigo	nombre	descripcion	caracteristicasAlmacenamiento
PK, SA, NN	NN, CK	NN	NN	NN

Proveedor					
id_proveedor	nif	nombre	direccion	nombreContacto	telefonoContacto
PK, SA, NN	NN, ND, CK	NN	NN	NN	NN

OrdenCompra							
id_compra	cantidad	precioAcordado	fechaEspera	estado	id_proveedor	id_producto	id_sucursal
PK, SA, NN, ND	NN	NN, ND	NN	NN	FKproveedorid_proveedor, NN, ND	FKproducto.id_producto, NN, ND	FKsucursalid_producto, NN, ND

RecepcionProducto						
id_recepcion	fechaRecepcion	cantidadRecebida	costoUnitario	id_bodega	id_producto	id_bodega
PK, SA, NN	NN	NN	NN	FKbodegaid_bodega, NN, ND	FKproducto.id_producto, NN, ND	FKbodega.id_bodega, NN, ND

Orden_Producto	
id_producto	id_ordenCompra
FKproducto.id_producto, NN	FKordenCompra.id_compra, NN

Producto_proveedor	
id_producto	id_proveedor
FKproducto.id_producto, NN	FKproveedor.id_proveedor, NN

1. Identificación de Entidades: Entidad:

Sucursal

- En el modelo UML, una sucursal es una entidad que representa una unidad de operación en diferentes ciudades. En el modelo relacional, los atributos de la entidad son nombre, ciudad, Instalación en m2 y teléfono, para todos se tiene de restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Específicamente ID_SUCURSAL que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor. **Entidad: Bodega**
- En el modelo UML, una bodega es una entidad que representa el lugar donde se almacenan los productos antes de ser enviados a su respectiva sucursal. Tiene atributos como nombre, tamaño en m2, costo promedio, capacidad, y el id de la sucursal a la que está asociada, para todos se tiene como restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Específicamente ID_BODEGA que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor.

Entidad: Producto

- En el modelo UML, un producto es una entidad que representa todos los artículos disponibles en el supermercado. Tiene atributos como nombre, costo en bodega, precio de venta, presentación, cantidad, unidad de medida, peso empaque, expiración y código de barras y el id de la categoría a la que está asociada, para todos se tiene como restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Además en código de barras se tiene de restricción ND la cual nos indica que los valores dentro de esta columna no pueden ser repetidos ya que son únicos al producto que están representando. Específicamente ID_PRODUCTO que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor. **Entidad: Categoría**

- En el modelo UML, la categoría es la entidad encargada de clasificar los productos según su tipo (perecederos, no perecederos) . Tiene atributos como código, nombre, descripción y características de almacenamiento, para todos se tiene de restricción

NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Específicamente ID_CATEGORIA que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor. **Entidad:**

Proveedor

- En el modelo UML, un proveedor es una entidad que representa las compañías que suministran los productos a las sucursales. Tiene atributos como NIT, nombre, dirección, nombre del contacto y teléfono de contacto., para todos se tiene de restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Además, para el atributo NIT se restringe con ND (not duplicate) ya que cada código de nit es único para cada empresa. Específicamente ID_PROVEEDOR que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor. **Entidad: OrdenCompra**
- En el modelo UML, una orden de compra es una entidad que registra la adquisición de productos de los proveedores. Tiene atributos como cantidad, precio acordado, fecha de espera, estado, id del proveedor a quien le corresponde la orden de compra e id del producto que se va a adquirir, para todos se tiene de restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Además, para el atributo precio acordado se restringe con ND (not duplicate) ya que un precio de una orden de compra no se puede repetir. Específicamente ID_ORDENCOMPRA que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor.

Entidad: RecepciónProducto

- Esta entidad registra la recepción de productos en una bodega específica. Tiene atributos como fecha de recepción, cantidad recibida, costo unitario, id de la bodega donde pertenece e id del producto, para todos se tiene como restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Específicamente ID_RECEPCIÓNPRODUCTO que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor.

Entidad: Orden_Producto

- Esta entidad registra la relación que tienen el producto y la orden de compra. Los atributos que tienen son las llaves foráneas que llaman a sus ids, es decir el id de producto y el id de orden de compra. Para estas dos llaves se tiene como restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío.

Entidad: Producto_proveedor

2. Relaciones:

Relación: Sucursal-Bodega

- **Descripción:** Una sucursal tiene una o varias bodegas, lo que es una relación 1 a muchos. Ya que la sucursal puede pedir nuevo inventario de varias bodegas si hay un imprevisto con alguna de ellas.
- **Transformación:** Esta relación se maneja a través de la clave foránea id_sucursal en la tabla Bodega, que referencia la tabla Sucursal.

Relación: Bodega-Producto

- **Descripción:** Una bodega puede almacenar varios productos y uno en varias bodegas, una relación N. No tendría sentido que una bodega pueda almacenar solo uno de ellos puede almacenar diferentes si tiene la capacidad necesaria para hacerlo.
- **Transformación:** Esta relación podría requerir una tabla intermedia para manejar la relación muchos a muchos. En este caso se maneja mediante la inclusión de id_bodega y id_producto en la tabla RecepciónProducto.

Relación: Producto-Categoría

- **Descripción:** Un producto pertenece a una categoría y una categoría puede tener múltiples productos. Un producto obligatoriamente debe pertenecer a una categoría y múltiples productos pueden pertenecer a múltiples categorías como por ejemplo “refrigerados” que son productos que pueden ser almacenados juntos en una bodega refrigerada

- **Transformación:** Se maneja a través de la clave foránea id_categoria en la tabla Producto, que referencia la tabla Categoría.

Relación: Proveedor-Producto

- **Descripción:** Un proveedor puede suministrar uno o varios productos y uno o varios proveedores pueden suministrarlo (ambos 1..*). Es lógico que varios proveedores puedan adquirir un producto si se tienen inconvenientes con alguno de estos. Por otra parte, un proveedor puede proveer múltiples productos pues no hay una razón para restringir al proveedor a suministrar únicamente un producto
- **Transformación:** Esta relación se maneja en la tabla OrdenCompra donde id_proveedor y id_producto se relacionan con las tablas correspondientes.

Relación: Orden de Compra-Recepción de Producto

- **Descripción:** Es verdad que una orden de compra puede incluir múltiples productos, pero una recepción de producto se asocia a una única orden de compra. Al recibir un grupo de productos estos van a estos asociados a solo una orden de compra. Al recibir los productos no se pueden combinar productos de otras órdenes de compra
- **Transformación:** Esta relación se maneja mediante las claves foráneas en las tablas correspondientes (id_ordenCompra en RecepciónProducto).

Relación: Sucursal-Orden de compra

- **Descripción:** una Orden de compra debe estar dirigida a 1 sucursal. Por otro lado, una sucursal puede tener varias órdenes de compra. Se pueden haber realizado varias órdenes de compra por diferentes sucursales, una sucursal puede tener 1 o muchas órdenes de compra pues estas pueden estar asociadas a diferentes proveedores.
- **Transformación:** Esta relación se maneja mediante las llaves foráneas en la tabla orden Compra referenciando el id sucursal).

Relación: Bodega-Recepción Producto

- **Descripción:** una bodega puede tener varias recepciones de productos y una recepción de productos está asociada a una bodega. Las recepciones de productos se hacen por bodega. Una bodega puede múltiples pedidos de otra manera cuando se acaben no se podría reabastecer
- **Transformación:** Esta relación se maneja mediante las llaves foráneas en la tabla recepcionProducto referenciando el id bodega).

Relación: Orden de Compra-Producto

- **Descripción:** Una orden de compra puede tener múltiples productos, sin embargo, y un producto puede aparecer en diferentes órdenes de compra siempre y cuando sean diferentes unidades (si es un televisor que ese mismo televisor no aparezca en otra

orden de compra, sino que sea otra unidad del mismo modelo de televisor). Un almacén puede pedir múltiples productos en una misma orden de compra. Y un producto puede ser pedido por múltiples órdenes de compra.

- **Transformación:** Esta relación se maneja mediante una tercera tabla que tiene como atributos una llave foránea que referencie el id del producto y una llave foránea que represente el id de la orden de compra.

Relación: Orden de Compra-Proveedor

- **Descripción:** Una orden de compra solo está asociada a un proveedor mientras que un proveedor puede tener múltiples órdenes de compra. Si una orden de compra estuviera asociado a múltiples proveedores tocaría especificar que productos vienen de que proveedor y a que orden de compra pertenecen complicando las tablas. Y un proveedor no tiene razón para estar restringido a tener un solo orden de compra.
- **Transformación:** Esta relación se maneja mediante las llaves foráneas en la tabla OrdenCompra referenciando el id_proveedor).

3. Proceso de Normalización:

- **Primera Forma Normal (1FN):** Cada tabla contiene solo valores atómicos; es decir, no hay conjuntos o listas de valores en una sola celda.
- **Segunda Forma Normal (2FN):** Todas las tablas están en 1FN, y todos los atributos que no son clave dependen completamente de la clave primaria.
- **Tercera Forma Normal (3FN):** Todas las tablas están en 2FN, y no hay dependencia transitiva de los atributos que no son clave primaria respecto a la clave primaria.

Para la tabla sucursal, todos los atributos contienen valores atómicos: nombre, ciudad, instalacionEnM2, telefono, es decir que cumple con 1NF. Ahora bien, todos los atributos dependen completamente de la llave principal: id_sucursal por lo que cumple con 2NF. Asimismo no existen dependencias transitivas y por tanto cumple con 3NF. Por último, la llave primaria es la única superllave (cumple con BCNF).

Para la tabla bodega, todos los atributos contienen valores atómicos: nombre, tamanoEnM2, costoPromedio, capacidad, lo que indica que cumple con 1NF. Asimismo, todos estos atributos dependen netamente de la llave principal: id_bodega y cumple con 2NF. No existen dependencias transitivas (3NF), y la llave primaria es la única superllave (BCNF).

Para la tabla producto, los atributos contienen valores atómicos: nombre, costoBodega, precioVenta, presentación, cantidadPresentacion, unidadMedida, volumenEmpaque, pesoEmpaque, expiracion, codigoBarras (1NF). De igual manera, todos los atributos

dependen directamente de la llave principal: id_producto (2NF). También no se reflejan dependencias transitivas (3NF) y además la llave primaria es superllave (BCNF).

Para la tabla categoría, todos los atributos contienen valores atómicos: código, nombre, descripción, característicasAlmacenamiento, es decir que cumple con 1NF. Igualmente, todos los atributos dependen de la llave principal: id_categoria y por tanto respeta la segunda forma normal. No existen dependencias transitivas. Y la llave primaria es la única superllave (BCNF).

Para la tabla proveedor, todos los atributos contienen valores atómicos: nit, nombre, dirección, nombreContacto, telefonoContacto, es decir que cumple con 1NF. Ahora bien, todos los atributos dependen completamente de la llave principal: id_proveedor por lo que cumple con 2NF. Asimismo no existen dependencias transitivas y por tanto cumple con 3NF. Por último, la llave primaria es la única superllave (cumple con BCNF).

Para la tabla ordenCompra, todos los atributos contienen valores atómicos: cantidad, precioAcordado, fechaEspera, estado, es decir que cumple con 1NF. Ahora bien, todos los atributos dependen completamente de la llave principal: id_compra por lo que cumple con 2NF. Asimismo no existen dependencias transitivas y por tanto cumple con 3NF. Por último, la llave primaria es la única superllave (cumple con BCNF).

Para la tabla RecepcionProducto, todos los atributos contienen valores atómicos: fechaRecepcion, cantidadRecibida, costoUnitario, es decir que cumple con 1NF. Ahora bien, todos los atributos dependen completamente de la llave principal: id_recepcion por lo que cumple con 2NF. Asimismo no existen dependencias transitivas y por tanto cumple con 3NF. Por último, la llave primaria es la única superllave (cumple con BCNF).

Para la tabla orden_Producto, todos los atributos contienen valores atómicos: id_Producto, id_ordenCompra, es decir que cumple con 1NF. Asimismo, todos estos atributos dependen netamente de la llave principal compuesta por: id_producto e id_ordenCompra y cumple con 2NF. No existen dependencias transitivas (3NF), y la llave primaria compuesta es la única superllave (BCNF)

Y por último, para la tabla producto_proveedor, todos los atributos contienen valores atómicos: id_Producto, id_proveedor, es decir que cumple con 1NF. Asimismo, todos estos atributos dependen netamente de la llave principal compuesta por: id_producto e id_proveedory cumple con 2NF. No existen dependencias transitivas (3NF), y la llave primaria compuesta es la única superllave (BCNF)

4. Escenarios de pruebas:

RF1 - Crear una Ciudad

1.Prueba de Creación Exitosa:

Acción: Insertar una ciudad con un nombre y código únicos, por ejemplo, "Medellín" con el código "MED".

Resultado Esperado: La ciudad se registra correctamente en la base de datos.

2.Prueba de Unicidad:

Acción: Intentar insertar otra ciudad con el mismo código "MED".

Resultado Esperado: El sistema debe rechazar la inserción y manejar el error de duplicado adecuadamente.

RF2 - Crear una Sucursal

1. Prueba de Creación Exitosa:

- **Acción:** Insertar una nueva sucursal en "Medellín" con nombre "Sucursal Centro", dirección "Av. Principal 123", teléfono "123456789".
- **Resultado Esperado:** La sucursal se registra correctamente en la base de datos asociada a la ciudad de "Medellín".

2. Prueba de Asociación:

- **Acción:** Intentar crear una sucursal asociada a una ciudad que no existe, por ejemplo, "Sincelejo".
- **Resultado Esperado:** El sistema debe manejar el error de ciudad inexistente adecuadamente.

RF3 - Crear y Borrar una Bodega

1. Prueba de Creación Exitosa:

- **Acción:** Insertar una nueva bodega llamada "Bodega 1" con tamaño de 100 m² asociada a la sucursal "Sucursal Centro".
- **Resultado Esperado:** La bodega se registra correctamente y está asociada a la sucursal especificada.

2. Prueba de Borrado Exitoso:

- **Acción:** Eliminar la bodega "Bodega 1".

- **Resultado Esperado:** La bodega se elimina correctamente, incluyendo todos los datos relacionados, y no quedan registros huérfanos.

3. Prueba de Integridad:

- **Acción:** Intentar eliminar una bodega que tiene productos almacenados.
- **Resultado Esperado:** El sistema debe manejar el error adecuadamente y no permitir la eliminación si hay productos asociados.

RF4 - Crear y Actualizar Proveedores

1. Prueba de Creación Exitosa:

- **Acción:** Insertar un nuevo proveedor con NIT "123456789", nombre "Proveedor ABC", dirección "Calle 45", contacto "Juan Pérez", teléfono "123456789".
- **Resultado Esperado:** El proveedor se registra correctamente en la base de datos.

2. Prueba de Actualización Exitosa:

- **Acción:** Modificar el contacto del proveedor "Proveedor ABC" a "Ana López".
- **Resultado Esperado:** La información del proveedor se actualiza correctamente en la base de datos.

3. Prueba de Validación:

- **Acción:** Intentar crear o actualizar un proveedor con un NIT duplicado o formato de teléfono incorrecto.
- **Resultado Esperado:** El sistema debe manejar los errores adecuadamente y rechazar la inserción o actualización.

RF5 - Crear y Leer una Categoría de Producto

1. Prueba de Creación Exitosa:

- **Acción:** Insertar una nueva categoría de producto llamada "Electrónicos" con código "ELEC" y descripción "Categoría para productos electrónicos".
- **Resultado Esperado:** La categoría se registra correctamente en la base de datos.

2. Prueba de Lectura:

- **Acción:** Consultar la categoría con el código "ELEC".
- **Resultado Esperado:** Se obtiene la información correcta de la categoría, incluyendo el nombre y descripción.

RF6 - Crear, Leer y Actualizar un Producto

1. Prueba de Creación Exitosa:

- **Acción:** Insertar un nuevo producto "Papas fritas Les frites" con todos los detalles especificados y asociarlo a la categoría "Perecedero".
- **Resultado Esperado:** El producto se registra correctamente y está asociado a la categoría correcta.

2. Prueba de Lectura:

- **Acción:** Consultar el producto "Papas fritas Les frites" por su código de barras/código.
- **Resultado Esperado:** Se obtiene la información completa del producto, incluyendo su categoría.

3. Prueba de Actualización Exitosa:

- **Acción:** Actualizar el precio unitario del producto "Papas fritas Les frites".
- **Resultado Esperado:** El precio se actualiza correctamente en la base de datos.

4. Prueba de Asociación:

- **Acción:** Intentar crear un producto asociado a una categoría inexistente.
- **Resultado Esperado:** El sistema debe manejar el error adecuadamente.

RF7 - Crear una Orden de Compra para una Sucursal

1. Prueba de Creación Exitosa:

- **Acción:** Insertar una nueva orden de compra con todos los detalles (fecha, sucursal, proveedor, productos, cantidades, precios) y verificar que quede en estado "vigente".
- **Resultado Esperado:** La orden de compra se registra correctamente y su estado es "vigente".

2. Prueba de Validación de Datos:

- **Acción:** Intentar crear una orden de compra con datos inválidos, como una fecha de entrega pasada.
- **Resultado Esperado:** El sistema debe manejar el error adecuadamente.

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

1. Prueba de Actualización Exitosa:

- **Acción:** Cambiar el estado de una orden de compra de "vigente" a "anulada".
- **Resultado Esperado:** La orden de compra cambia correctamente a "anulada".

2. Prueba de Restricciones:

- **Acción:** Intentar anular una orden de compra que ya está en estado "entregada".
- **Resultado Esperado:** El sistema debe rechazar la anulación y manejar el error adecuadamente.

RF9 - Mostrar Todas las Órdenes de Compra

1. Prueba de Consulta:

- **Acción:** Listar todas las órdenes de compra.
- **Resultado Esperado:** Se muestran correctamente todas las órdenes de compra con la información completa y su estado actual.

2. Prueba de Filtros:

- **Acción:** Aplicar filtros para consultar órdenes de compra por estado o sucursal.
- **Resultado Esperado:** Los resultados de la consulta se ajustan a los filtros aplicados.