

## **Proyecto 1: DISEÑO**

### **Base de Datos para SuperAndes: Diseño de base de datos**

**Maria Lucia Benavides Domínguez**

**Kevin Andrés Castillo Peña**

**Angela Jiménez Gonzáles**

#### **1. Elementos fundamentales:**

- a. Sucursal
- b. Ciudad
- c. Bodega
- d. Producto
- e. Categoría
- f. Proveedor
- g. ProductoBodega
- h. ProductoProveedor
- i. OrdenCompra
- j. DetalleOrdenCompra
- k. RecepcionProducto
- l. DetalleRecepcion
- m. NivelReorden
- n.

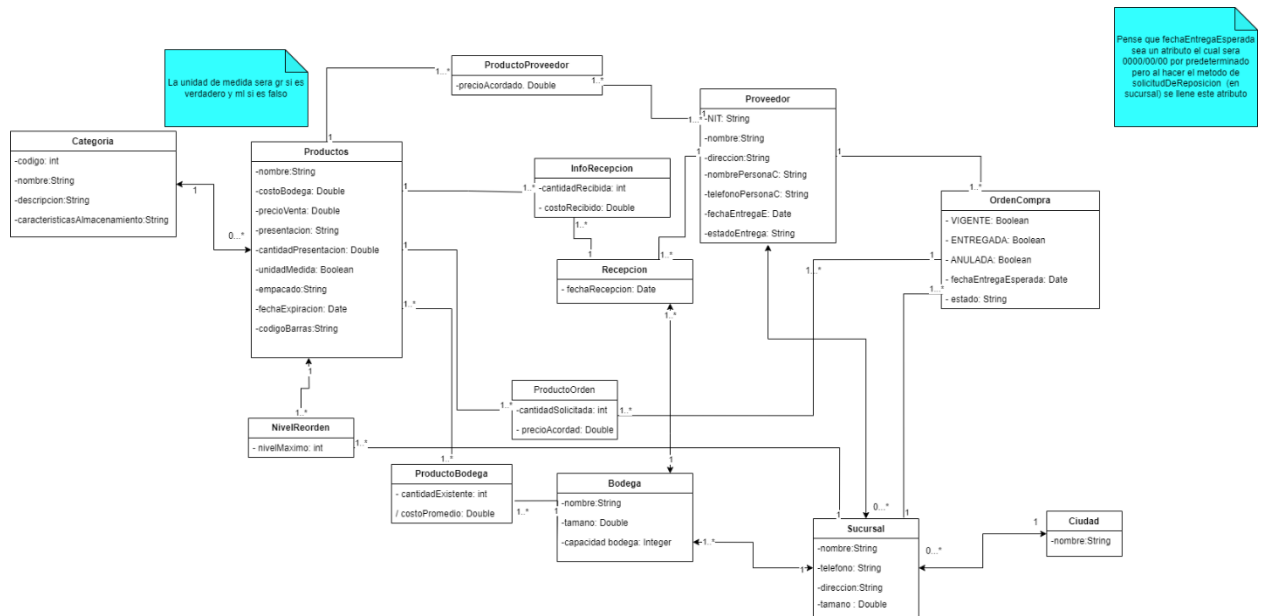
#### **2. Reglas de Negocio:**

1. Cada bodega debe tener un registro exacto de la cantidad y el costo promedio de los productos almacenados.
2. El costo promedio de un producto en una bodega debe actualizarse cada vez que se recibe una nueva entrega del producto.
3. Los productos deben clasificarse en categorías específicas que determinen sus condiciones de almacenamiento.
4. Los productos perecederos deben tener una fecha de expiración y deben gestionarse con prioridad en cuanto a rotación de inventario.
5. Cada sucursal debe mantener un nivel mínimo de reorden para cada producto. Si las existencias caen por debajo de este nivel, debe generarse una orden de compra automáticamente.
6. El nivel de reorden debe ser revisado periódicamente y ajustado según las tendencias de demanda.
7. La recepción de productos debe estar asociada a una orden de compra y una bodega específica.
8. Al recibir productos, se debe verificar que la cantidad y el costo unitario coincidan con lo especificado en la orden de compra.
9. Después de la recepción, la cantidad de producto y el costo promedio en la bodega deben actualizarse.

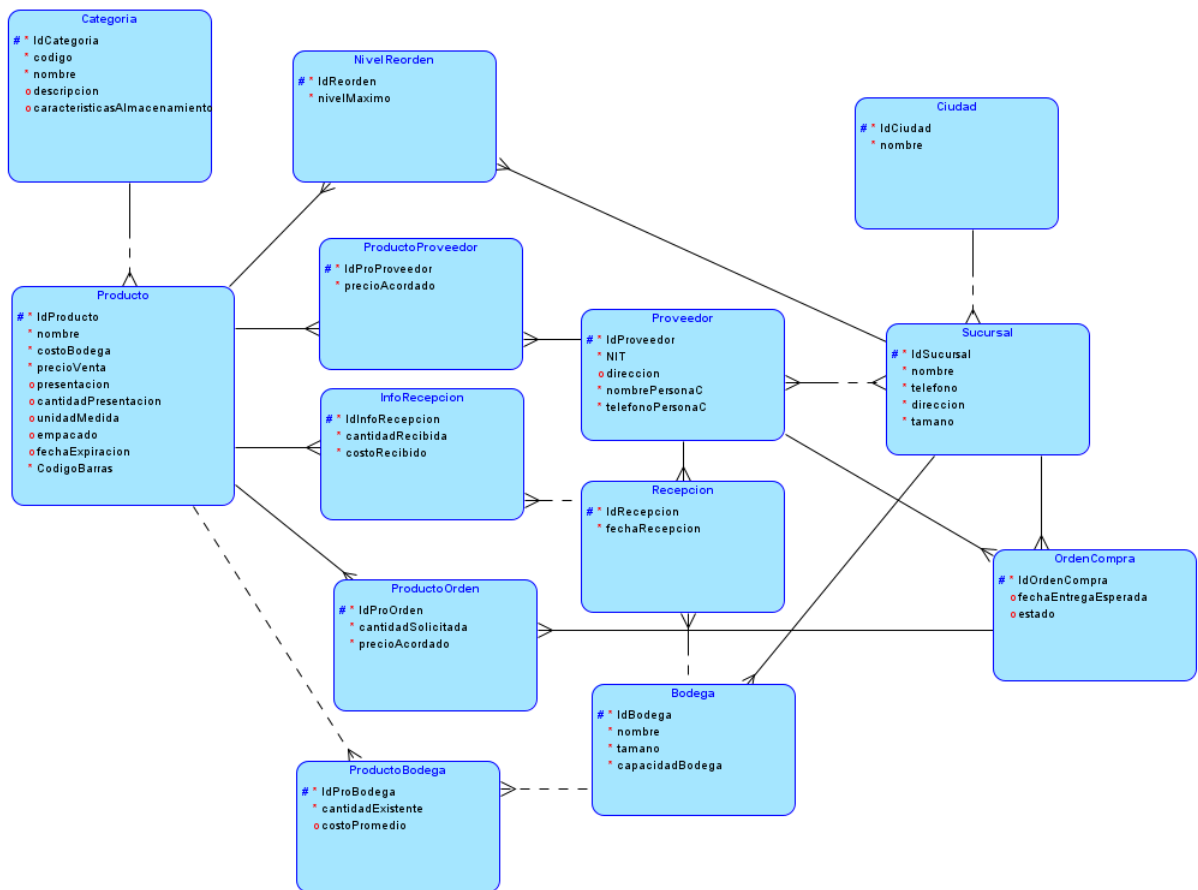
10. Una orden de compra puede estar en estado "vigente", "entregada" o "anulada". Solo las órdenes en estado "vigente" pueden recibir productos.
11. Una orden de compra pasa al estado "entregada" cuando todos los productos han sido recibidos.
12. Cada movimiento de productos debe ser rastreable desde la orden de compra, pasando por la recepción, hasta su ubicación final en la bodega.
13. La información sobre fechas de expiración debe ser utilizada para la rotación adecuada del inventario (primero en entrar, primero en salir).
14. Cada sucursal debe ser capaz de realizar y gestionar sus propias órdenes de compra, inventarios y recepciones de productos.
15. Las sucursales deben poder realizar traspasos de productos entre bodegas dentro de la misma sucursal o entre diferentes sucursales.
16. El nombres, códigos o títulos de las entidades Categoría, Producto, Sucursal, Bodega o ciudad nunca pueden tener un valor nulo
17. Los precios acordados nunca pueden tener valores nulos
18. Los precios de los productos nunca pueden tener valores nulos
19. Los productos dentro de las órdenes de compra nunca pueden ser nulos, al igual que las cantidades.

### 3. Análisis y modelo conceptual:

#### UML



#### Modelo E/R



#### 4. Niveles de Normalización del modelo:

- **Primera forma normal (1FN)**

- Todos los atributos de las entidades contienen valores atómicos. Es decir, cada atributo almacena un solo valor por tupla (registro) y no existen atributos multivalorados ni repetidos.
- Cada tabla tiene una clave primaria que identifica de manera única a cada tupla.
- No existen grupos repetitivos o conjuntos de atributos que puedan descomponerse en tablas adicionales.

- **Segunda forma normal (2FN)**

- Además de estar en 1FN, todas las tablas cumplen con la condición de que todos los atributos no clave (no primos) dependen completamente de la clave primaria completa.
- En las tablas que tienen una clave primaria compuesta, como *ProductoProveedor*, *ProductoBodega* y *ProductoOrden*, cada atributo no clave depende de la totalidad de la clave primaria, y no de una parte de ella.
  - Por ejemplo, en la tabla *ProductoProveedor*, el atributo *precioAcordado* depende tanto del *idProducto* como del *idProveedor*. Esto significa que no

existe una dependencia parcial de la clave compuesta. El *precioAcordado* siempre dependerá de estas llaves primarias.

Dentro de cada una de las tablas se cumple la relación descrita anteriormente

- **Tercera forma normal (3FN)**
  - Además de estar en 2FN, no existen dependencias transitivas entre atributos no clave. Es decir, no existe un atributo no clave que dependa de otro atributo no clave.
  - Todos los atributos no clave dependen únicamente de la clave primaria y no de otros atributos no clave.
  - Por ejemplo, en la tabla *Sucursal*, todos los atributos como *nombre*, *ciudad*, *dirección* y *teléfono* dependen directamente del *idSucursal*, que es la clave primaria. Estos no tienen ninguna dependencia que pretenda llegar a un dato por medio de otro.
- **Forma normal de Boyce-Cood**
  - Además de estar en 3FN, para cada dependencia funcional en el modelo, el determinante es una superclave. Es decir, si un atributo X determina un atributo Y, entonces X es una superclave (una clave candidata o parte de una clave candidata).
  - En las tablas con claves compuestas, como *ProductoProveedor*, *ProductoBodega* y *ProductoOrden*, las dependencias funcionales son tales que los determinantes (las combinaciones de las claves compuestas) son claves candidatas. No hay dependencias donde un atributo no clave determine otro atributo no clave.

## 5. Justificación de las tablas:

### Tabla: Producto

- **IdProducto:** Es el identificador de la tabla (PK). Es un valor asignado automáticamente por el sistema (SA). No puede ser nulo (NN) ni duplicado (ND).
- **Nombre:** Nombre del producto. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **CostoBodega:** Costo del producto en bodega. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **PrecioVenta:** Precio al que se venderá el producto. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **Presentación:** Descripción de la presentación del producto. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **CantidadPresentacion:** Cantidad en la presentación del producto. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **UnidadMedida:** Unidad de medida del producto. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **Empacado:** Características del empaque. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **FechaExpiracion:** Fecha en que expira el producto. No puede ser nulo (NN), y tiene un CK que verifica que la fecha sea posterior a la actual.
- **CodigoBarras:** Código de barras para la identificación. No puede ser nulo (NN), ni duplicado (ND), y es asignado por el usuario (UA).

- **IdCategoria:** Es una clave foránea (FK) que referencia a la tabla **Categoría**. No puede ser nulo (NN).

#### **Tabla: Categoría**

- **IdCategoria:** Es el identificador de la tabla (PK). Es un valor asignado por el sistema (SA), no puede ser nulo (NN) ni duplicado (ND).
- **Nombre:** Nombre de la categoría. No puede ser nulo (NN), ni duplicado (ND), y es asignado por el usuario (UA).
- **Descripción:** Descripción de la categoría. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **CaracterísticasAlmacenamiento:** Define las condiciones de almacenamiento. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **Perecedero:** Indica si los productos de esta categoría son perecederos. Es asignado por el usuario (UA).

#### **Tabla: Proveedor**

- **IdProveedor:** Es el identificador de la tabla (PK). Es un valor asignado por el sistema (SA), no puede ser nulo (NN).
- **NIT:** Número de Identificación Tributaria del proveedor. No puede ser nulo (NN), ni duplicado (ND), y es asignado por el usuario (UA).
- **Nombre:** Nombre del proveedor. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **Dirección:** Dirección del proveedor. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **NombrePersonaContacto:** Nombre de la persona de contacto. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **TeléfonoPersonaContacto:** Teléfono de la persona de contacto. No puede ser nulo (NN) y es asignado por el usuario (UA).

#### **Tabla: Bodega**

- **IdBodega:** Es el identificador de la tabla (PK). Es un valor asignado por el sistema (SA), no puede ser nulo (NN).
- **Nombre:** Nombre de la bodega. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **Tamaño:** Tamaño en metros cuadrados de la bodega. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **IdSucursal:** Es una clave foránea (FK) que referencia a la tabla **Sucursal**. No puede ser nulo (NN).

#### Tabla: Sucursal

- **IdSucursal:** Es el identificador de la tabla (**PK**). No puede ser nulo (**NN**).
- **Nombre:** Nombre de la sucursal. No puede ser nulo (**NN**) y es asignado por el usuario (**UA**).
- **Teléfono:** Teléfono de la sucursal. No puede ser nulo (**NN**) y es asignado por el usuario (**UA**).
- **Dirección:** Dirección de la sucursal. No puede ser nulo (**NN**) y es asignado por el usuario (**UA**).
- **Tamaño:** Tamaño de la sucursal en metros cuadrados. No puede ser nulo (**NN**) y es asignado por el usuario (**UA**).
- **IdCiudad:** Es una clave foránea (**FK**) que referencia a la tabla **Ciudad**. No puede ser nulo (**NN**).

#### Tabla: InfoRecepcion

- **IdInfoRecepcion:** Es el identificador de la tabla (**PK**). Es un valor asignado por el sistema (**SA**), no puede ser nulo (**NN**).
- **CantidadRecibida:** Cantidad de producto recibida. No puede ser nulo (**NN**) y tiene una restricción **CK** que asegura que sea mayor a 0.
- **CostoRecibido:** Costo de la recepción del producto. No puede ser nulo (**NN**) y es asignado por el usuario (**UA**).
- **IdRecepcion:** Es una clave foránea (**FK**) que referencia a la tabla **Recepcion**. No puede ser nulo (**NN**).
- **IdProducto:** Es una clave foránea (**FK**) que referencia a la tabla **Producto**. No puede ser nulo (**NN**).
- **Estado:** Estado de la recepción. No puede ser nulo (**NN**) y es asignado por el usuario (**UA**).

#### Tabla: Recepcion

- **IdRecepcion:** Es el identificador de la tabla (**PK**). Es un valor asignado por el sistema (**SA**), no puede ser nulo (**NN**).
- **FechaRecepcion:** Fecha de la recepción. No puede ser nulo (**NN**) y tiene una restricción **CK** para verificar que la fecha sea válida.
- **IdBodega:** Es una clave foránea (**FK**) que referencia a la tabla **Bodega**. No puede ser nulo (**NN**).
- **IdOrdenCompra:** Es una clave foránea (**FK**) que referencia a la tabla **OrdenCompra**. No puede ser nulo (**NN**).

#### Tabla: ProductoOrden

- **IdProOrden:** Es el identificador de la tabla (**PK**). No puede ser nulo (**NN**).

- **CantidadSolicitada:** Cantidad solicitada en la orden. No puede ser nulo (NN) y tiene una restricción CK para verificar que sea mayor a 0.
- **PrecioAcordado:** Precio acordado para la compra. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **IdOrdenCompra:** Es una clave foránea (FK) que referencia a la tabla **OrdenCompra**. No puede ser nulo (NN).
- **IdProducto:** Es una clave foránea (FK) que referencia a la tabla **Producto**. No puede ser nulo (NN).

#### Tabla: NivelReorden

- **IdNivelReorden:** Es el identificador de la tabla (PK). No puede ser nulo (NN).
- **NivelMaximo:** Nivel máximo de reorden. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **IdProducto:** Es una clave foránea (FK) que referencia a la tabla **Producto**. No puede ser nulo (NN).
- **IdSucursal:** Es una clave foránea (FK) que referencia a la tabla **Sucursal**. No puede ser nulo (NN).

#### Tabla: ProductoBodega

- **IdProductoBodega:** Es el identificador de la tabla (PK). No puede ser nulo (NN).
- **CantidadExistente:** Cantidad existente del producto en la bodega. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **CostoPromedio:** Costo promedio del producto en la bodega. No puede ser nulo (NN), es un dato derivado (DD).
- **IdProducto:** Es una clave foránea (FK) que referencia a la tabla **Producto**. No puede ser nulo (NN).
- **IdBodega:** Es una clave foránea (FK) que referencia a la tabla **Bodega**. No puede ser nulo (NN).

#### Tabla: Ciudad

- **IdCiudad:** Es el identificador de la tabla (PK). No puede ser nulo (NN) ni duplicado (ND).
- **Nombre:** Nombre de la ciudad. No puede ser nulo (NN), ni duplicado (ND), y es asignado por el usuario (UA).

#### Tabla: OrdenCompra

- **IdOrdenCompra:** Es el identificador de la tabla (PK). Es un valor asignado por el sistema (SA), no puede ser nulo (NN).

- **FechaEntrega:** Fecha de entrega de la orden. No puede ser nulo (NN), y tiene un **CK** que verifica que la fecha sea posterior a la actual.
- **Estado:** Estado de la orden de compra. No puede ser nulo (NN) y es asignado por el usuario (UA).
- **IdSucursal:** Es una clave foránea (FK) que referencia a la tabla **Sucursal**. No puede ser nulo (NN).
- **IdProveedor:** Es una clave foránea (FK) que referencia a la tabla **Proveedor**. No puede ser nulo (NN).

#### **Tabla: Cliente**

- **IdCliente:** Es el identificador de la tabla (PK). Es un valor asignado por el sistema (SA), no puede ser nulo (NN).
- **nombre:** El nombre no puede ser nulo (NN)
- **cedula:** Lacedula No puede ser nulo (NN) y es asignado por el usuario

#### **Tabla: Venta**

- **idCompra:** Es el identificador de la tabla (PK). Es un valor asignado por el sistema (SA), no puede ser nulo (NN).
- **Fecha :** Día de la realización de la compra
- **Cantidad:** numeor de productos comrpados
- **precioUnitario:** valor total de la venta

### **6. Escenarios de Prueba:**

#### **RF1 - Crear producto**

##### **Escenario cumple restricciones:**

**Descripción:** Crear un producto con un código de barras único.

Entrada:

IdProducto: "P001"

Nombre: "Papas Fritas"

CostoBodega: 4200

PrecioVenta: 9000

Presentación: "Paquetón"

CantidadPresentacion: 1000

UnidadMedida: "gr"

Empacado: "Bolsa"



FechaExpiracion: "2025-12-31"

CodigoBarras: "1234567890"

IdCategoria: "C001"

Resultado esperado: El producto se crea exitosamente porque todos los valores cumplen con las restricciones. El CodigoBarras es único, la FechaExpiracion es futura, y todos los campos obligatorios están completos.

#### **Escenario NO cumple restricciones:**

**Descripción:** Intentar crear un producto con un código de barras duplicado.

Entrada:

- IdProducto: "P002"
- Nombre: "Jugo de Naranja"
- CostoBodega: 2000
- PrecioVenta: 4500
- Presentación: "Botella"
- CantidadPresentacion: 500
- UnidadMedida: "ml"
- Empacado: "Botella Plástica"
- FechaExpiracion: "2025-12-31"
- CodigoBarras: "1234567890" (**Duplicado**)
- IdCategoria: "C002"

Resultado esperado: Falla la creación porque el CodigoBarras ya existe y debe ser único (ND).

#### **RF2 – Creación Categoría**

##### **Escenario Cumple Restricciones:**

**Descripción:** Crear una categoría con un nombre y código únicos.

Entrada:

- IdCategoria: "C001"
- Nombre: "Perecederos"
- Descripción: "Productos que tienen fecha de vencimiento."
- CaracterísticasAlmacenamiento: "Refrigeración"
- Perecedero: "Sí"

Resultado esperado: La categoría se crea exitosamente porque el Nombre es único y los demás valores cumplen con las restricciones.

##### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear una categoría con un nombre duplicado.

Entrada:

- IdCategoria: "C002"
- Nombre: "Perecederos" (Duplicado)
- Descripción: "Productos no perecederos."
- CaracterísticasAlmacenamiento: "Almacenamiento en seco"
- Perecedero: "No"

Resultado esperado: Falla la creación porque el Nombre ya existe y debe ser único (ND).

### **RF3 – Creación Proveedor**

#### **Escenario Cumple Restricciones:**

**Descripción:** Crear un proveedor con un NIT único.

Entrada:

- IdProveedor: "PR001"
- NIT: "900123456"
- Nombre: "Proveedor A SAS"
- Dirección: "Calle 123"
- NombrePersonaContacto: "Juan Pérez"
- TeléfonoPersonaContacto: "3001234567"

Resultado esperado: El proveedor se crea exitosamente porque el NIT es único y los demás valores

cumplen con las restricciones.

#### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear un proveedor con un NIT duplicado.

Entrada:

- IdProveedor: "PR002"
- NIT: "900123456" (Duplicado)
- Nombre: "Proveedor B CO"
- Dirección: "Carrera 456"
- NombrePersonaContacto: "Ana Gómez"
- TeléfonoPersonaContacto: "3012345678"

Resultado esperado: Falla la creación porque el NIT ya existe y debe ser único (ND).

### **RF4 – Creación Bodega**

#### **Escenario Cumple Restricciones:**

**Descripción:** Crear una bodega con nombre único en una sucursal.

Entrada:

- IdBodega: "B001"
- Nombre: "Bodega Central"
- Tamaño: 500
- IdSucursal: "S001"

Resultado esperado: La bodega se crea exitosamente porque el Nombre y la IdSucursal son únicos y los demás valores cumplen con las restricciones.

#### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear una bodega con el mismo nombre en la misma sucursal.

Entrada:

- IdBodega: "B002"
- Nombre: "Bodega Central" (Duplicado en la misma sucursal)
- Tamaño: 300
- IdSucursal: "S001"

Resultado esperado: Falla la creación porque el Nombre de la bodega ya existe en esa sucursal.

#### **RF5 – Creación Sucursal**

##### **Escenario Cumple Restricciones:**

**Descripción:** Crear una sucursal con un nombre único.

Entrada:

- IdSucursal: "S001"
- Nombre: "Sucursal Norte"
- Teléfono: "3201234567"
- Dirección: "Avenida 5"
- Tamaño: 1000
- IdCiudad: "C001"

Resultado esperado: La sucursal se crea exitosamente porque el Nombre es único y los demás valores cumplen con las restricciones.

##### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear una sucursal con un nombre duplicado en la misma ciudad.

Entrada:

- IdSucursal: "S002"
- Nombre: "Sucursal Norte" (Duplicado)
- Teléfono: "3212345678"
- Dirección: "Calle 10"
- Tamaño: 800
- IdCiudad: "C001"

Resultado esperado: Falla la creación porque el Nombre ya existe en esa ciudad.

## **RF6- Creación ProductoProveedor**

### **Escenario Cumple Restricciones:**

**Descripción:** Crear una relación entre un producto y un proveedor con claves únicas.

- Entrada:
- IdProducto: "P001"
- IdProveedor: "PR001"
- PrecioAcordado: 8500
- Tamaño: 500

Resultado esperado: La relación se crea exitosamente porque las claves combinadas IdProducto y IdProveedor son únicas y los demás valores cumplen con las restricciones.

### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear una relación con una combinación duplicada de producto y proveedor.

Entrada:

- IdProducto: "P001"
- IdProveedor: "PR001" (Duplicado)
- PrecioAcordado: 8000
- Tamaño: 400

Resultado esperado: Falla la creación porque la combinación de IdProducto y IdProveedor ya existe.

## **RF7- Creación InfoRecepcion**

### **Escenario Cumple Restricciones:**

**Descripción:** Crear un registro de recepción con cantidad válida.

Entrada:

- IdInfoRecepcion: "IR001"
- CantidadRecibida: 100
- CostoRecibido: 4200
- IdRecepcion: "R001"
- IdProducto: "P001"
- Estado: "Recibido"
- 

Resultado esperado: El registro de recepción se crea exitosamente porque la CantidadRecibida es mayor a 0 y los demás valores cumplen con las restricciones.

### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear un registro de recepción con cantidad inválida.

Entrada:

- IdInfoRecepcion: "IR002"
- CantidadRecibida: -50 (Inválido)

- CostoRecibido: 4200
- IdRecepcion: "R002"
- IdProducto: "P002"
- Estado: "Recibido"

Resultado esperado: Falla la creación porque la CantidadRecibida es negativa y viola la restricción CK[>0.0].

## **RF8 Creación Recepción**

### **Escenario Cumple Restricciones:**

**Descripción:** Crear una recepción con una fecha válida.

Entrada:

- IdRecepcion: "R001"
- FechaRecepcion: "2024-09-01"
- IdBodega: "B001"
- IdOrdenCompra: "OC001"
- 

Resultado esperado: La recepción se crea exitosamente porque la FechaRecepcion es válida y los demás valores cumplen con las restricciones.

### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear una recepción con una fecha inválida.

Entrada:

- IdRecepcion: "R002"
- FechaRecepcion: "2023-01-01" (Fecha pasada)
- IdBodega: "B002"
- IdOrdenCompra: "OC002"

Resultado esperado: Falla la creación porque la FechaRecepcion es una fecha pasada y viola la restricción CK que exige que la fecha sea posterior a la actual.

## **RF9 - Creación ProductoOrden**

### **Escenario Cumple Restricciones:**

**Descripción:** Crear un registro de producto en una orden de compra con cantidad válida.

Entrada:

- IdProOrden: "PO001"
- CantidadSolicitada: 100
- PrecioAcordado: 8500
- IdOrdenCompra: "OC001"
- IdProducto: "P001"

Resultado esperado: El registro se crea exitosamente porque la CantidadSolicitada es mayor que 0 y los demás valores cumplen con las restricciones.

#### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear un registro de producto en una orden de compra con cantidad inválida.

Entrada:

- IdProOrden: "PO002"
- CantidadSolicitada: 0 (Inválido)
- PrecioAcordado: 8500
- IdOrdenCompra: "OC002"
- IdProducto: "P002"

Resultado esperado: Falla la creación porque la CantidadSolicitada es 0, lo cual viola la restricción CK que requiere una cantidad mayor a 0.

#### **RF10- CreacionNivelReorden**

##### **Escenario Cumple Restricciones:**

**Descripción:** Crear un nivel de reorden válido para un producto en una sucursal.

Entrada:

- IdNivelReorden: "NR001"
- NivelMaximo: 50
- IdProducto: "P001"
- IdSucursal: "S001"

Resultado esperado: El nivel de reorden se crea exitosamente porque el NivelMaximo es válido y los demás valores cumplen con las restricciones.

##### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear un nivel de reorden con un nivel máximo inválido.

Entrada:

- IdNivelReorden: "NR002"
- NivelMaximo: -10 (Inválido)
- IdProducto: "P002"
- IdSucursal: "S002"

Resultado esperado: Falla la creación porque el NivelMaximo es negativo, lo cual es inválido según las restricciones.

#### **RF11 - ProductoBodega**

##### **Escenario Cumple Restricciones:**

**Descripción:** Crear un registro de producto en una bodega con datos válidos.

Entrada:

- IdProductoBodega: "PB001"
- CantidadExistente: 200
- CostoPromedio: 5000
- IdProducto: "P001"
- IdBodega: "B001"
- 

Resultado esperado: El registro se crea exitosamente porque todos los valores cumplen con las restricciones. El CostoPromedio es calculado correctamente y la CantidadExistente es válida.

#### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear un registro de producto en una bodega con un costo promedio inválido.

Entrada:

- IdProductoBodega: "PB002"
- CantidadExistente: 200
- CostoPromedio: -5000 (Inválido)
- IdProducto: "P002"
- IdBodega: "B002"

Resultado esperado: Falla la creación porque el CostoPromedio es negativo, lo cual viola la restricción de dato derivado (DD), ya que el costo promedio no puede ser negativo.

### **RF12 – Creacion Ciudad**

#### **Escenario Cumple Restricciones:**

**Descripción:** Crear una ciudad con nombre y código únicos.

Entrada:

- IdCiudad: "C001"
- Nombre: "Bogotá"

Resultado esperado: La ciudad se crea exitosamente porque el Nombre es único y los demás valores cumplen con las restricciones.

#### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear una ciudad con un nombre duplicado.

Entrada:

- IdCiudad: "C002"
- Nombre: "Bogotá" (Duplicado)

Resultado esperado: Falla la creación porque el Nombre ya existe y debe ser único (ND).

## **RF13 - OrdenCompra**

### **Escenario Cumple Restricciones:**

**Descripción:** Crear una orden de compra con datos válidos.

Entrada:

- IdOrdenCompra: "OC001"
- FechaEntrega: "2024-09-15"
- Estado: "Vigente"
- IdSucursal: "S001"
- IdProveedor: "PR001"

Resultado esperado: La orden de compra se crea exitosamente porque la FechaEntrega es futura y los demás valores cumplen con las restricciones.

### **Escenario No Cumple Restricciones:**

**Descripción:** Intentar crear una orden de compra con una fecha de entrega pasada.

Entrada:

- IdOrdenCompra: "OC002"
- FechaEntrega: "2023-08-01" (Inválido)
- Estado: "Vigente"
- IdSucursal: "S002"
- IdProveedor: "PR002"

Resultado esperado: Falla la creación porque la FechaEntrega es una fecha pasada y viola la restricción CK que exige una fecha futura.

## **7. Cambios y actualizaciones**

- Creación de cliente venta para correcto modelado del supermercado, así los productos pueden o no disminuir
- Se elimino producto proveedor ya que podría llegar a hacer innecesarios
- Se genero volumen Empaque y pesoEmpaque dentro de producto ya que era necesario para buen modelamiento del programa
- A NivelReorden se le asigno la capacidad de la bodega ya que se debía relacionar con el producto.
- Se cambio la forma del estado dentro del UML

## **8. UsuarioOracle:**

ISIS2304A01202420

Maria Lucia Benavides



## 9. Documentacion:

### Descripción de las clases:

- **Bodega:** Almacena información sobre los almacenes.
- **Categoria:** Clasifica los productos.
- **Ciudad:** Guarda información sobre las ciudades.
- **Cliente:** Contiene los datos de los clientes.
- **InfoRecepcion:** Probablemente relacionada con la recepción de productos o pedidos.
- **InfoRecepcionPK:** Clave primaria de la tabla InfoRecepcion.
- **NivelReOrden:** Define los niveles de reorden de los productos.
- **OrdenCompra:** Representa una orden de compra.
- **Producto:** Contiene la información de los productos.
- **ProductoBodega:** Relaciona productos con bodegas.
- **ProductoBodegaPK:** Clave primaria de la tabla ProductoBodega.
- **ProductoOrden:** Relaciona productos con órdenes de compra.
- **ProductoOrdenPK:** Clave primaria de la tabla ProductoOrden.
- **ProductoPK:** Clave primaria de la tabla Producto.
- **Proveedor:** Proveedores que otorgan los productos
- **Recepción:** objeto que define la recepción de un producto de un proveedor
- **Sucursal:** Todas las sucursales de super\_andes
- **Venta:** información de las ventas de cada producto

**Cada una de estas clases cuenta con un controller y un repository en donde se encuentran sus consultas CRUD.**

Para las consultas avanzadas tenemos:

*calcularIndiceOcupacion* dentro de sucursal para sacar el índice de ocupación dentro de cada bodega de cada producto

*listarProductosReorden* dentro de producto para obtener los productos que necesitan una renovación o una nueva orden al proveedor

*darProductosConBodega* Lista de productos disponibles en la bodega

*darSucursalesSegunProducto* dar las sucursales según el producto

## **ESCENARIOS DE PRUEBAS DE CONCURRENCIA:**

### **Escenario 1**

#### **Para el escenario 1 llegamos a las siguientes conclusiones:**

Se lanza la consulta `consultaIngresoProductoBodega_SERIALIZABLE`, que busca obtener los documentos de ingreso de productos en una bodega específica.

Esta operación es transaccional y se ejecuta en un nivel de aislamiento `SERIALIZABLE`, lo cual es el nivel más estricto. Este nivel de aislamiento garantiza que no habrá interferencias de otras transacciones y asegura la consistencia en los datos durante la ejecución de RFC6.

La consulta realiza una primera obtención de los datos (almacenados en `respuestaConsultaIngreso`), espera 30 segundos (`Thread.sleep(30000)`), y luego vuelve a realizar la consulta en el repositorio antes de retornar los datos.

#### **Ejecución Concurrente de RF10:**

Antes de que transcurran los 30 segundos de espera en RFC6, se ejecuta la operación `registroProductoBodega` (RF10), cuyo propósito es registrar el ingreso de productos en la bodega. Esta operación también es transaccional, con un nivel `READ COMMITTED`.

RF10 busca actualizar el estado del inventario en la bodega y marcar la orden de compra como "ENTREGADA".

#### **Paso a Paso en la Línea de Tiempo**

T=0 segundos:

RFC6 comienza a ejecutarse y realiza la primera consulta en el repositorio para obtener los documentos de ingreso de productos. Como está en modo `SERIALIZABLE`, se "bloquea" un conjunto de datos, evitando que otros procesos modifiquen el mismo conjunto durante la transacción.

T=15 segundos:

Se lanza RF10 mientras RFC6 aún está en espera. Debido a la serialización de RFC6, RF10 no puede modificar o registrar ningún ingreso de producto en la bodega hasta que RFC6 termine su ejecución.

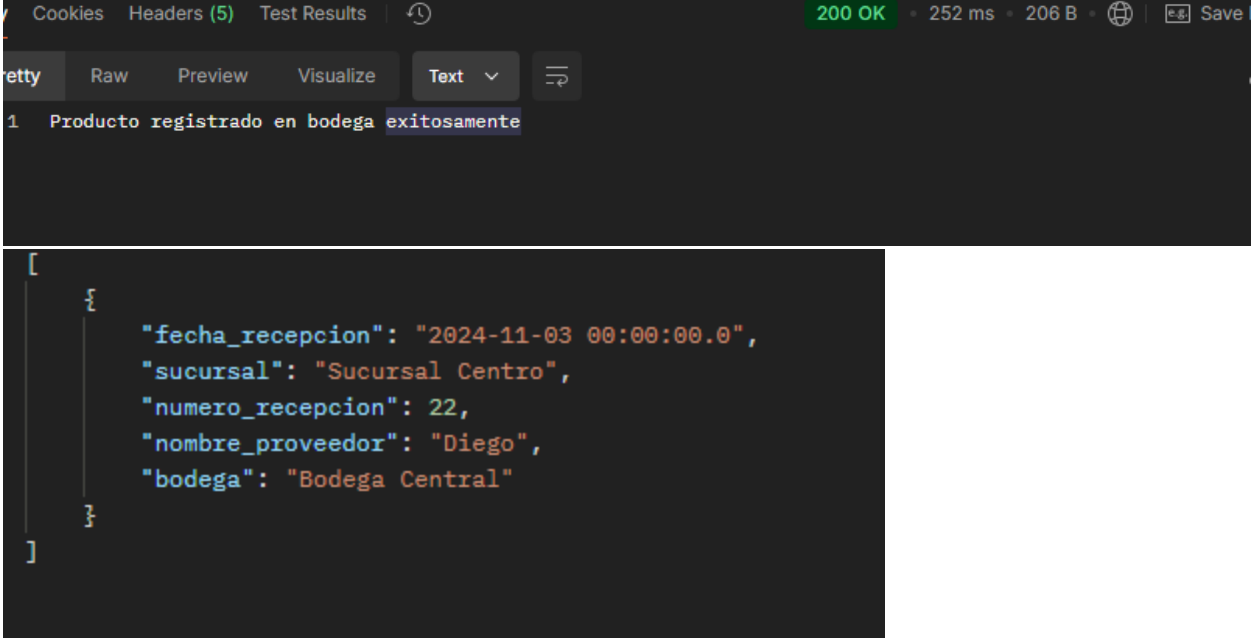
T=30 segundos:

RFC6 realiza la segunda consulta en el repositorio y finaliza, liberando el bloqueo serializable.

T=31 segundos:

RF10 puede ahora registrar los productos en la bodega y finalizar su proceso.

## Resultado y Observaciones



The screenshot shows a REST client interface with the following details:

- Top bar: Cookies, Headers (5), Test Results, and a refresh icon. On the right, it displays "200 OK", "252 ms", "206 B", and a "Save" button.
- Response tabs: Pretty (selected), Raw, Preview, Visualize, and Text.
- Status bar: "1 Producto registrado en bodega exitosamente".
- Response body (JSON):

```
[
  {
    "fecha_recepcion": "2024-11-03 00:00:00.0",
    "sucursal": "Sucursal Centro",
    "numero_recepcion": 22,
    "nombre_proveedor": "Diego",
    "bodega": "Bodega Central"
  }
]
```

### Descripción de lo sucedido:

El componente RF10 no pudo registrar el ingreso de productos en la bodega hasta que finalizó la ejecución de RFC6. Esto se debe a que el nivel de aislamiento SERIALIZABLE de RFC6 bloquea las modificaciones concurrentes de los datos que está leyendo.

Al estar en aislamiento serializable, RFC6 garantiza que los datos que leyó inicialmente permanecen inalterados hasta el final de la transacción. RF10, que requiere modificar estos datos, tiene que esperar a que RFC6 libere el bloqueo.

## Escenario 2

### 1. Pasos para la Ejecución Concurrente de RFC7 y RF10 a través de la Línea de Tiempo

- Paso 1: Se inició RFC7 con el nivel de aislamiento READ COMMITTED, lo cual comenzó una

consulta de documentos de ingreso de productos a bodega, incluyendo un temporizador de 30 segundos.

- Paso 2: Mientras RFC7 estaba en ejecución, se inició RF10 para registrar un nuevo ingreso

de producto en la bodega.

- Paso 3: RF10 finalizó rápidamente y confirmó la transacción de inserción, mientras RFC7

aún estaba en el proceso de espera.

- Paso 4: Al finalizar los 30 segundos, RFC7 completó su consulta y devolvió los documentos

de ingreso de los últimos 30 días.

## **2. Descripción de lo Sucedido**

- ¿RF10 Debió Esperar a RFC7?

No, RF10 no tuvo que esperar a que terminara RFC7. Dado que el nivel de aislamiento de RFC7 es READ COMMITTED, no bloquea la escritura concurrente de RF10. Esto significa

que RF10 pudo registrar el ingreso del producto de manera simultánea sin interferir con la

consulta en curso de RFC7.

## **3. Resultado Presentado por RFC7**

- ¿Apareció el Nuevo Documento de Ingreso?

Sí, el nuevo documento de ingreso creado por RF10 apareció en los resultados de la consulta de RFC7. Esto se debe a que RFC7, con el nivel de aislamiento READ

COMMITTED, puede ver las transacciones confirmadas durante su ejecución. Al completar

el temporizador de 30 segundos, RFC7 incluyó el nuevo documento de ingreso registrado por RF10 en sus resultados.

## **Conclusión**

El nivel de aislamiento READ COMMITTED permitió que la consulta de RFC7 incluyera el

documento de ingreso de productos realizado por RF10 de manera simultánea, ya que la transacción de RF10 se confirmó antes de que finalizara la ejecución de RFC7. Esto

demuestra que READ COMMITTED permite ver los datos confirmados sin bloquear las transacciones de escritura concurrentes.

SuperAndes / Recepcion / Consulta READ\_COMMITTED

GET

http://localhost:8080/proyecto/recepciones/consultaIngresoProductoBodega\_READ\_COMMITTED?bodega\_id=1&sucursal\_id=1

Send

Params • Authorization Headers (6) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	⋮ Bulk Edit
<input checked="" type="checkbox"/> bodega_id	1		
<input checked="" type="checkbox"/> sucursal_id	1		
Key	Value	Description	

Body Cookies Headers (5) Test Results

200 OK • 30.63 s • 313 B • Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "sucursal": "Sucursal Centro",
4     "fecha_recepcion": "2024-11-03 00:00:00.0",
5     "bodega": "Bodega Central",
6     "nombre_proveedor": "Diego",
7     "numero_recepcion": 22
8   }
9 ]
```

Postbot Ctrl Alt P

SuperAndes / Bodega / Registrar producto en bodega

GET

http://localhost:8080/proyecto/bodegas/registroProductoBodega?orden\_compra\_id=1&bodega\_id=1

Send

Params • Authorization Headers (6) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	⋮ Bulk Edit
<input checked="" type="checkbox"/> orden_compra_id	1		
<input checked="" type="checkbox"/> bodega_id	1		
Key	Value	Description	

Body Cookies Headers (5) Test Results

200 OK • 171 ms • 206 B • Save Response

Pretty Raw Preview Visualize Text

```
1 Producto registrado en bodega exitosamente
```