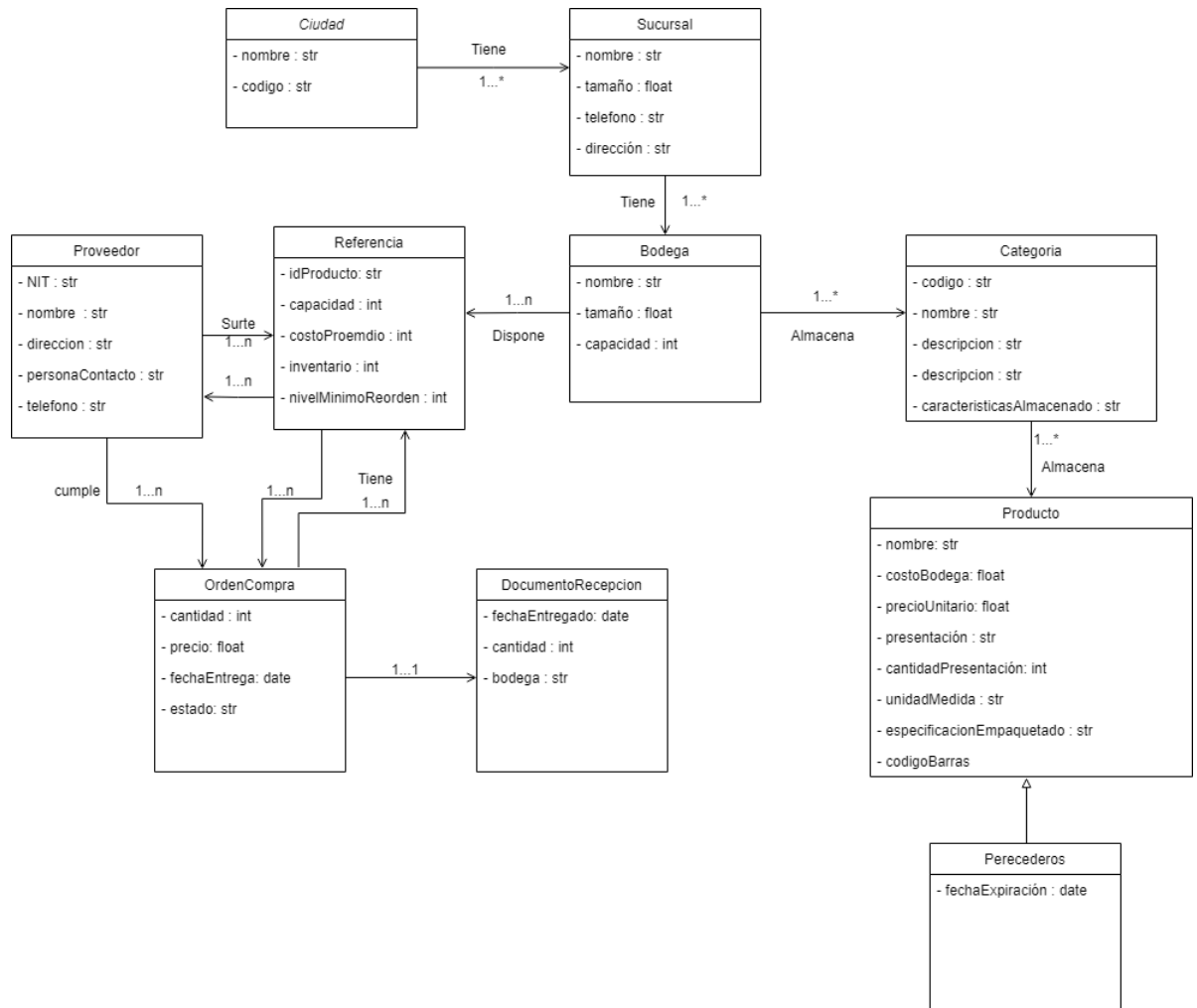


Entrega 1 - Diseño modelo relacional y modelo relacional actualizado

ENTREGA 1:

1. Modelo conceptual UML



El diagrama UML representa un sistema de gestión de inventarios enfocado en la administración de sucursales, bodegas, proveedores, productos y órdenes de compra. Este modelo incluye varias entidades clave, como Ciudad, Sucursal, Bodega, Proveedor, Referencia, OrdenCompra, DocumentoRecepcion, Categoria, Producto, y Perecederos, y sus relaciones entre sí.

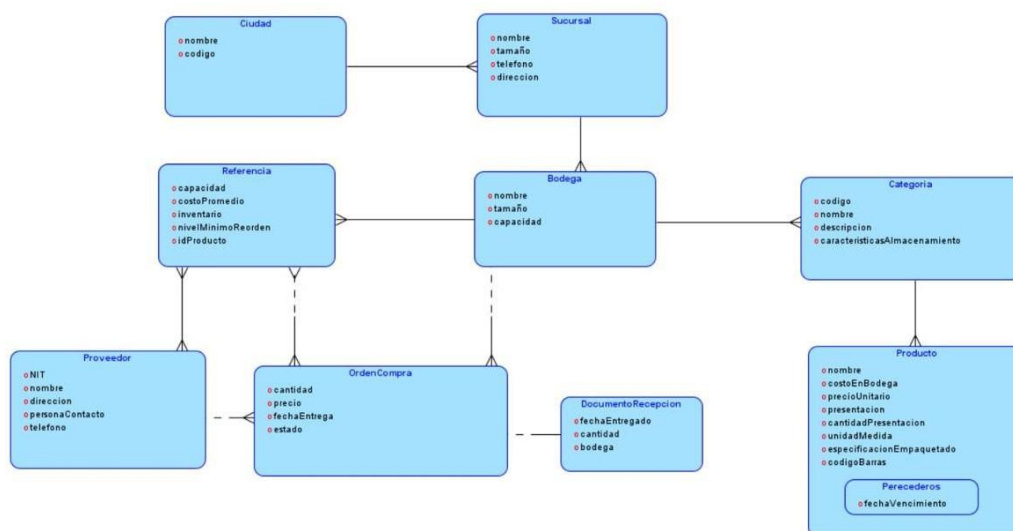
En el sistema, una Ciudad contiene múltiples Sucursales, cada una de las cuales puede tener una o varias Bodegas. Las Bodegas almacenan productos organizados en categorías específicas, y están vinculadas a referencias de productos (Referencia) que indican detalles como la capacidad, el costo promedio, el inventario disponible y el nivel mínimo de reorden. Las Referencias son suministradas por Proveedores, que tienen su propia identificación y detalles de contacto.

Las Órdenes de Compra son generadas para adquirir productos y están asociadas a Referencias específicas, indicando la cantidad, el precio y la fecha de entrega. Estas órdenes se registran y son verificadas por un Documento de Recepción que confirma la cantidad entregada y la bodega donde se almacenaron los productos.

Las Categorías agrupan diferentes tipos de productos, que tienen atributos como nombre, costo en bodega, precio unitario, presentación, unidad de medida, especificación de empaquetado y código de barras. Además, existe una subclase de productos denominada Perecederos, que incluye un atributo adicional para registrar la fecha de expiración, lo que es crítico para la gestión de productos con una vida útil limitada.

En resumen, este modelo UML describe de manera detallada cómo se relacionan las distintas entidades dentro de un sistema de gestión de inventarios, facilitando la organización y control de productos, la coordinación con proveedores, y el manejo eficiente de las existencias en múltiples sucursales y bodegas.

2. Modelo conceptual en E/R



Este diagrama E/R (Entidad-Relación) representa un sistema de gestión de inventarios donde se destacan las entidades clave y sus relaciones, poniendo especial énfasis en la obligatoriedad y cardinalidad de estas relaciones.

En el modelo, cada Ciudad está relacionada obligatoriamente con una o más Sucursales, ya que una ciudad siempre debe tener al menos una sucursal. A su vez, cada Sucursal debe tener al menos una Bodega, lo que indica que no puede existir una sucursal sin una bodega asociada.

La entidad Referencia está obligatoriamente asociada con una Bodega y un Proveedor. Esto significa que, para cada referencia de producto, debe haber un proveedor específico que lo suministre y una bodega que lo almacene. Las Referencias también están relacionadas obligatoriamente con Órdenes de Compra, indicando que una referencia siempre debe estar presente en al menos una orden de compra.

Las Órdenes de Compra se vinculan obligatoriamente a Documentos de Recepción, lo que asegura que cada orden debe ser recibida y registrada en un documento. Esto resalta la importancia de la trazabilidad de las compras en el sistema.

La entidad Categoría se relaciona obligatoriamente con Producto, lo que implica que cada producto debe pertenecer a una categoría específica, garantizando una correcta clasificación dentro del sistema. Los productos perecederos se destacan como una subcategoría de Producto, donde la fechaVencimiento es obligatoria para todos los productos que pertenecen a esta subcategoría, asegurando un control estricto sobre los productos con una vida útil limitada.

Este modelo subraya la obligatoriedad de las relaciones en un sistema de gestión de inventarios, asegurando que cada entidad clave esté adecuadamente vinculada y que los datos esenciales no se queden sin registrar, promoviendo así la integridad y consistencia del sistema.

3. Modelo relacional (sin normalizar):

A continuación, se presentarán las tablas del modelo relacional sin normalizar:

- Ciudad

Ciudad

Código ciudad	Nombre
PK, NN, ND, NC, CK	NC, NN

La entidad Ciudad posee dos atributos: CódigoCiudad y Nombre. El atributo CódigoCiudad tiene las siguientes restricciones: PK (clave primaria), NN (no nulo), ND (no duplicado), NC (no cambiante), y CK (chequeo). El atributo Nombre tiene las restricciones NC (no cambiante) y NN (no nulo).

- Sucursal

Sucursal

Nombre	Tamaño	Teléfono	Dirección	CódigoCiudad
PK	NN	ND	NN	FK, NN, NC, CK, ND

La entidad sucursal posee cuatro atributos iniciales con sus respectivas restricciones: Nombre (PK, llave primaria, Tamaño (NN, not null), Teléfono (ND, not duplicate) y Dirección (NN, not null). Además, se relaciona con la entidad ciudad por lo que comparte el atributo Código ciudad (FK, Foreign key; NN, not null; ND, not duplicate; NC, Valores estadísticos en el tiempo; CK, check) que indica una relación uno a muchos con la entidad Ciudad.

- Referencia

Referencia

ID producto	Inventario	costoPromedio	Capacidad	nivelMinimoReorden	NombreBodega
PK, NN, SA	NN, CK	NN	NN, DD	NN, DD	FK, NN, NC, ND

La entidad Referencia cuenta con cinco atributos principales: IdProducto, Inventario, CostoPromedio, Capacidad y NivelMinimoReorden. El atributo IdProducto se define como clave primaria y tiene restricciones de no nulo y System Assigned. El atributo Inventario está sujeto a restricciones de no nulo y chequeo. El atributo CostoPromedio también tiene la restricción de no nulo. Tanto Capacidad como NivelMinimoReorden tienen las restricciones de no nulo y dato dependiente. Adicionalmente, la entidad incluye el atributo NombreBodega, el cual establece una relación de uno a muchos con la entidad Bodega.

- Bodega

Bodega

Nombre	Tamaño	Capacidad	NombreSucursal
PK	NN	NN	FK, NN, NC, CK, ND

La entidad Bodega posee tres atributos iniciales con sus respectivas restricciones: Nombre (PK, llave primaria, Tamaño (NN, not null) y Capacidad (NN, not null). Además, se relaciona con la entidad sucursal por lo que comparte el atributo NombreSucursal (FK, Foreign key; NN, not null; ND, not duplicate; NC, Valores estadísticos en el tiempo; CK, check) que indica una relación uno a muchos con la entidad sucursal.

- Categoría

Categoría

Codigo categoria	Nombre	Descripción	Características almacenamiento	NombreBodega
PK	NN	NN	NN	FK, NN, ND

La entidad Categoría posee cuatro atributos iniciales con sus respectivas restricciones: Código categoría (PK, llave primaria), Nombre (NN, not null), Descripción (NN, not null) y Características almacenamiento (NN, not null). Además, se relaciona con la entidad Bodega por lo que comparte el atributo NombreBodega (FK, Foreign key; NN, not null; ND, not duplicate) que indica una relación uno a muchos con la entidad bodega.

- Producto

Producto								
Nombre	costoEnBodega	PrecioUnitario	CantidadPresentación	UnidadMedida	EspecificaciónEmpaquetado	CódigoBarras	Presentación	CódigoCategoría
PK	NN	NN	NN, NC	NN, NC	NN, NC	NN, NC, ND	NN, ND, NC	FK, NN, ND

La entidad Producto posee ocho atributos iniciales con sus respectivas restricciones: Nombre (PK, llave primaria), costoEnBodega (NN, not null), PrecioUnitario (NN, not null), CantidadPresentación (NN, not null; NC, valores estadísticos en el tiempo), EspecificaciónEmpaquetado (NN, not null; NC, valores estadísticos en el tiempo), CódigoBarras (NN, not null; NC, valores estadísticos en el tiempo; ND, not duplicate), Presentación (NN, not null; NC, valores estadísticos en el tiempo; ND, not duplicate). Además, se relaciona con la entidad sucursal por lo que comparte el atributo CódigoCategoría (FK, Foreign key; NN, not null; ND, not duplicate) que indica una relación uno a muchos con la entidad categoría.

- Producto perecedero

Producto Perecedero									
Nombre	costoEnBodega	PrecioUnitario	CantidadPresentación	UnidadMedida	EspecificaciónEmpaquetado	CódigoBarras	Presentación	FechaVencimiento	CódigoCategoría
PK, FK, NN,	NN	NN	NN, NC	NN, NC	NN, NC	NN, NC, ND	NN	NN, NC	FK, NN, ND

La entidad Producto perecedero posee ocho atributos iniciales con sus respectivas restricciones: Nombre (PK, llave primaria), costoEnBodega (NN, not null), PrecioUnitario (NN, not null), CantidadPresentación (NN, not null; NC, valores estadísticos en el tiempo), EspecificaciónEmpaquetado (NN, not null; NC, valores estadísticos en el tiempo), CódigoBarras (NN, not null; NC, valores estadísticos en el tiempo; ND, not duplicate), Presentación (NN, not null; NC, valores estadísticos en el tiempo; ND, not duplicate) y FechaVencimiento (NN, not null; NC, valores estadísticos en el tiempo). Además, se relaciona con la entidad sucursal por lo que comparte el atributo CódigoCategoría (FK, Foreign key; NN, not null; ND, not duplicate) que indica una relación uno a muchos con la entidad categoría.

- Proveedor

Proveedor			
NIT	Nombre	Dirección	personaContacto
PK	NN	NN, DD	NN, DD

La entidad Proveedor posee cuatro atributos iniciales con sus respectivas restricciones: NIT (PK, llave primaria), Nombre (NN, not null), Dirección (NN, not null; DD, dato derivado), PersonaContacto (NN, not null; DD, dato derivado), Teléfono (NN, not null; ND, not duplicate). Además, se relaciona con la entidad Referencia por lo que indica una relación muchos a muchos y también se relaciona con la entidad OrdenCompra por lo que indica una relación uno a muchos. Dicho lo anterior fue necesario crear una tabla adicional para vincular los atributos de las relaciones que serán explicadas a continuación.

- OrdenCompra

OrdenCompra

ID compra	Estado	Precio	fechaEntrega	Cantidad	NIT-Proveedor	NombreBodega
PK, NN, UA, ND	NN, CK	NN,DD	NN		FK, NN, ND	FK, NN, NC, ND

La entidad OrdenCompra posee cinco atributos iniciales con sus respectivas restricciones: ID compra (PK, llave primaria; NN, not null; UA, user assigned; ND, not duplicate), Estado (NN, not null; CK, check), Precio (NN, not null; DD, dato derivado), FechaEntrega (NN, not null) y Cantidad. Además, se relaciona con la entidad Proveedor por lo que comparte el atributo NIT proveedor (FK, Foreign key; NN, not null; ND, not duplicate) que indica una relación uno a muchos con la entidad categoría y con la entidad NombreBodega (FK, Foreign key; NN, not null; ND, not duplicate; NC, valores estadísticos en el tiempo) que indica una relación uno a muchos con la entidad bodega.

- Documento recepción

DocumentoRecepción

fechaEntrega	Cantidad	ID compra
PK		FK, NN, ND, UA

La entidad Documento recepción posee dos atributos iniciales con sus respectivas restricciones: fechaEntrega (PK, llave primaria) y Cantidad. Además, se relaciona con la entidad OrdenCompra por lo que comparte el atributo ID compra (FK, Foreign key; NN, not null; ND, not duplicate; UA, user assigned) que indica una relación uno a uno con la entidad.

- Relación proveedor referencia

Proveedor-Referencia

NIT	ID producto
PK, FK, NN	PK, FK, NN, SA

La entidad Proveedor-Referencia posee dos atributos iniciales con sus respectivas restricciones: NIT (PK, llave primaria; FK, Foreign key; NN, not null) y ID producto(PK, llave primaria; FK, Foreign key; NN, not null; SA, System Assigned). Esta tabla representa la relación entre proveedor y referencia

- Relación referencia – orden de compra

Referencia-Orden Compra

ID producto	ID compra
PK, FK, NN, SA	PK, FK, NN, UA, ND

La entidad Referencia-OrdenCompra posee dos atributos iniciales con sus respectivas restricciones: ID producto (PK, llave primaria; FK, Foreign key; NN, not null; SA, System Assigned) y ID compra (PK, llave primaria; FK, Foreign key; NN, not null; UA, User Assigned; ND, Valores estáticos en el tiempo). Esta tabla representa la relación entre referencia y orden de compra

4. Normalización Modelo

Con el fin de mejorar la calidad del modelo relacional, se realizó una normalización hasta la etapa BCNF para llegar a esto se realizaron los siguientes cambios:

- Primero se hizo un cambio de PK en las entidades Producto y ProductoPerecedero esto con el fin de que todos los atributos dependieran de la llave principal. La nueva llave principal es CodigoBarras.

Producto											1NF
CodigoBarras	costoEntodega	PrecioUnitario	CantidadPresentación	UnidadMedida	EspecificaciónEmpaquetado	Nombre	Presentación	CodigoCategoria			2NF
PK, NN, ND, NC	NN	NN	NN, NC	NN, NC	NN, NC	NN, NC	NN, ND, NC	FK, NN, ND			3NF
											BCNF

Producto Perecedero											1NF
CodigoBarras	costoEntodega	PrecioUnitario	CantidadPresentación	UnidadMedida	EspecificaciónEmpaquetado	Nombre	Presentación	FechaVencimiento	CodigoCategoria		2NF
PK, NN, ND, NC, FK	NN	NN	NN, NC	NN, NC	NN, NC	NN, NC	NN	NN, NC	FK, NN, ND		3NF
											BCNF

- Además, anteriormente en la entidad DocumentoRecepción la PK era numeroRadicado, sin embargo, esta llave no cumplía con los criterios de normalización requeridos. Ahora, se establecio una nueva llave numeroRadicado de tal manera que el resto de los atributos dependan de esta.

DocumentoRecepción

numeroRadicado	fechaEntrega	Cantidad	ID compra
PK	NN	NN	FK, NN, ND, UA

5. Escenarios de prueba:

Casos
prueba

Entidad	Descripción del caso	Prueba	Resultado esperado	Notas
Ciudad	Insertar un registro de Ciudad correcto.	Código ciudad: 1001, Nombre: Bogotá.	El registro se inserta exitosamente.	Todo perfecto
Ciudad	Insertar un registro de Ciudad con un código existente.	Código ciudad: 1001 (ya existe), Nombre: Medellín	Falla la inserción debido a la restricción de clave primaria.	La llave primaria debe ser única
Sucursal	Insertar un registro de Sucursal correcto.	Nombre: Sucursal Centro, Tamaño: 500 m², Teléfono: 1234567, Dirección: Calle 123, CódigoCiudad: 1001.	El registro se inserta exitosamente.	Campos correctos
Sucursal	Insertar una sucursal con un código de ciudad no existente.	Nombre: Sucursal Norte, Tamaño: 400 m², Teléfono: 7654321, Dirección: Calle 456, CódigoCiudad: 9999 (no existe).	Falla la inserción debido a la restricción de clave foránea.	El código debe existir
Referencia	Insertar un registro de Referencia correcto.	ID producto: 1234, Inventario: 50, CostoPromedio: 100, Capacidad: 100, NivelMínimoReorden: 10, NombreBodega: Central.	El registro se inserta exitosamente.	Los campos deberían coincidir.
Referencia	Insertar una referencia sin el nombre de la bodega.	ID producto: 1235, Inventario: 30, CostoPromedio: 80, Capacidad: 50, NivelMínimoReorden: 5, NombreBodega: NULL.	Falla la inserción debido a la restricción de clave foránea.	No pueden haber campos nulos, siempre debe haber información
Bodega	Insertar un registro correcto	Nombre: Bodega Principal, Tamaño: 1000 m², Capacidad: 5000, NombreSucursal: Sucursal Centro.	El registro se inserta exitosamente.	Que los campos sean medianamente coherentes
Bodega	Insertar una bodega con un tamaño negativo	Nombre: Bodega Secundaria, Tamaño: -200 m², Capacidad: 3000, NombreSucursal: Sucursal Norte.	Un campo tiene un registro	Tamaños positivos siempre

	oseaaaa erroneo		incoherente, falla por eso	
Categorí a	Insertar un registro de Categoría correcto.	CódigoCategoría: CAT001, Nombre: Electrónica, Descripción: Dispositivos electrónicos, CaracterísticasAlmacenamiento: Temperatura controlada, NombreBodega: Central.	Registro correcto	Bodega asociada correcta?
Categorí a	Insertar una categoría con un código duplicado.	CódigoCategoría: CAT001 (ya existe), Nombre: Muebles, Descripción: Mobiliario, CaracterísticasAlmacenamiento: Sin especificar, NombreBodega: Norte.	Falla la inserción debido a la restricción de clave primaria.	Codigo repetido, no puede existir
Product o	Insertar un registro de Producto correcto.	CodigoBarras: 0011223344, CostoEnBodega: 50, PrecioUnitario: 100, CantidadPresentación: 10, UnidadMedida: Unidad, EspecificaciónEmpaquetado: Caja, Nombre: Televisor, Presentación: Empaque individual, CodigoCategoría: CAT001.	Registro correcto	Campos completos
Product o	Codigo barras duplicado	CodigoBarras: 0011223344 (ya existe), CostoEnBodega: 60, PrecioUnitario: 120, CantidadPresentación: 20, UnidadMedida: Unidad, EspecificaciónEmpaquetado: Caja, Nombre: Laptop, Presentación: Empaque individual, CodigoCategoría: CAT002.	No hay una clave única, registro inválido	Tiene que haber siempre una llave única
Product o percedero	Insertar un registro de Producto Perecedero correcto.	CodigoBarras: 1122334455, CostoEnBodega: 20, PrecioUnitario: 40, CantidadPresentación: 5, UnidadMedida: Litro, EspecificaciónEmpaquetado: Botella, Nombre: Leche, Presentación: Botella, FechaVencimiento: 2024-10-01, CodigoCategoría: CAT003.	registro correcto	No hay fallas
Product o percedero	Insertar un producto percedero con una fecha de vencimiento pasada.	CodigoBarras: 1122334456, CostoEnBodega: 15, PrecioUnitario: 30, CantidadPresentación: 5, UnidadMedida: Litro, EspecificaciónEmpaquetado: Botella, Nombre: Yogurt, Presentación: Botella, FechaVencimiento: 2023-01-01, CodigoCategoría: CAT003.	El resultado presenta fallas	Hay un problema de inserción de datos, algo pasó
Proveed or	Hacer un registro de orden correcto	ID compra: 10001, Estado: Pendiente, Precio: 1500, FechaEntrega: 2024-09-15, Cantidad: 50, NIT-Proveedor: 8901234567, NombreBodega: Central.	Registro correcto	NIT correcto
Proveed or	Insretar registro con un NIT erroneo	ID compra: 10002, Estado: Pendiente, Precio: 2000, FechaEntrega: 2024-10-01, Cantidad: 60, NIT-Proveedor: 9999999999, NombreBodega: Norte.	Fallo, nit incorrecto o inexistente	Fallas en inserción de datos

Orden compra	Registro correcto	ID producto: 1234, ID compra: 10001.	Registro insertado correctamente	Existencias correctas
Orden compra	Insertar orden compra inexistente	ID producto: 1234, ID compra: 9999 (no existe).	Fallo en llaves foraneas	error de existencia

Dentro de los casos de prueba podemos notar que si se realiza un cambio importante o exponencial dentro de los datos, llevará a un fallo directo para realizar un registro correcto. Cómo podemos evidenciar, se debe a esto:

CIUDAD

- **Registro Correcto:** Inserta una ciudad con código 1001 y nombre Bogotá. Resultado esperado: éxito.
- **Código Existente:** Intento de insertar ciudad con código ya existente. Resultado esperado: error por clave primaria duplicada.

SUCURSAL

- **Registro Correcto:** Inserta una sucursal con código de ciudad existente. Resultado esperado: éxito.
- **Código Ciudad No Existente:** Inserta sucursal con código de ciudad no registrado. Resultado esperado: error por clave foránea.

REFERENCIA

- **Registro Correcto:** Inserta una referencia con todos los campos válidos. Resultado esperado: éxito.
- **Sin Nombre de Bodega:** Intento de insertar referencia sin nombre de bodega. Resultado esperado: error por campo nulo.

BODEGA

- **Registro Correcto:** Inserta una bodega con datos válidos. Resultado esperado: éxito.
- **Tamaño Negativo:** Inserta una bodega con tamaño negativo. Resultado esperado: error por datos incoherentes.

CATEGORÍA

- **Registro Correcto:** Inserta una categoría con todos los campos válidos. Resultado esperado: éxito.

- **Código Duplicado:** Intento de insertar categoría con código ya existente. Resultado esperado: error por clave primaria duplicada.

PRODUCTO

- **Registro Correcto:** Inserta un producto con todos los campos válidos. Resultado esperado: éxito.
- **Código Barras Duplicado:** Intento de insertar producto con código de barras duplicado. Resultado esperado: error por clave única duplicada.

PRODUCTO PERECEDERO

- **Producto Perecedero Correcto:** Inserta un producto perecedero con todos los datos válidos. Resultado esperado: éxito.
- **Fecha de Vencimiento Pasada:** Inserta un producto perecedero con fecha de vencimiento pasada. Resultado esperado: error por datos inválidos.

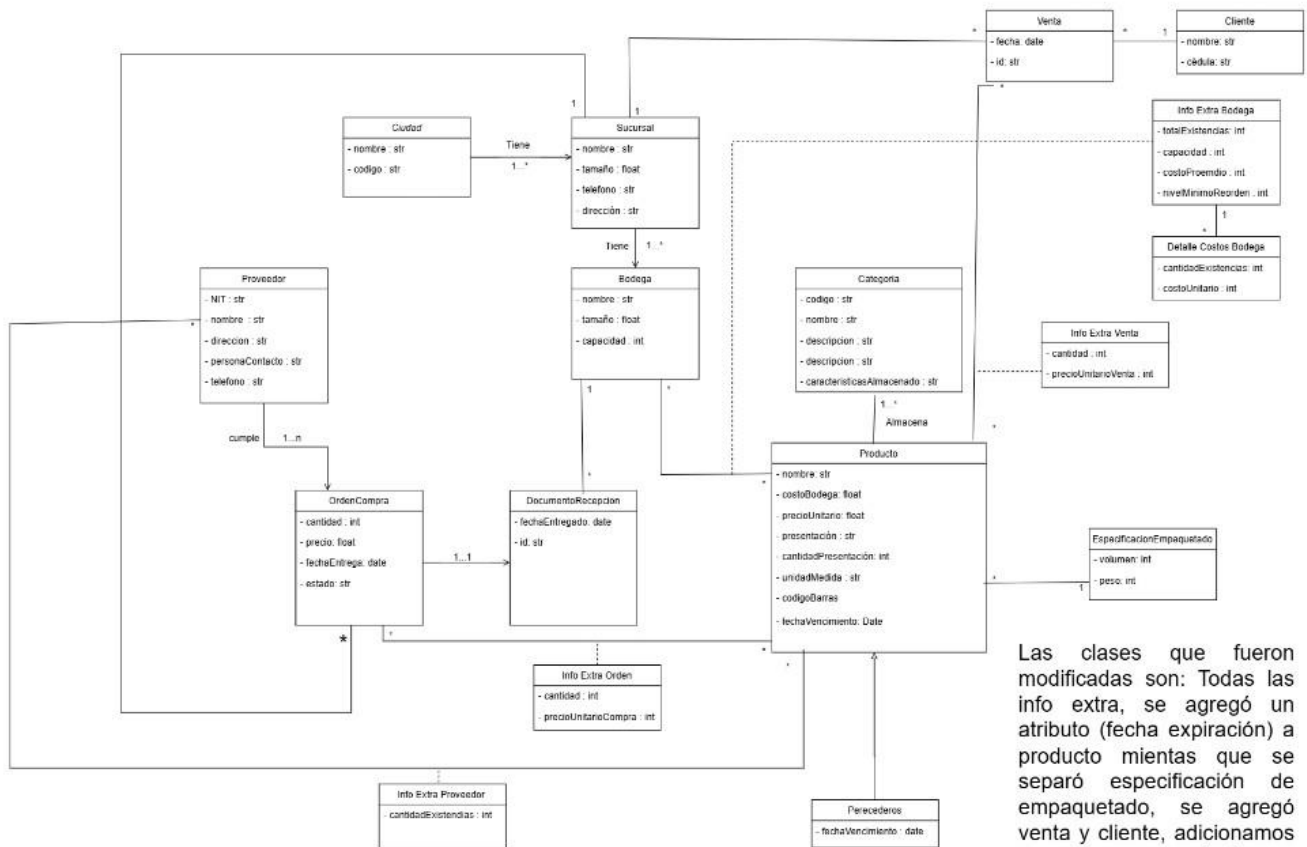
PROVEEDOR

- **Registro Correcto:** Inserta un registro de compra con NIT válido. Resultado esperado: éxito.
- **NIT Erróneo:** Intento de insertar registro de compra con NIT inválido. Resultado esperado: error por NIT incorrecto.

ORDEN COMPRA

- **Registro Correcto:** Inserta una orden de compra con IDs válidos. Resultado esperado: éxito.
- **Orden Compra Inexistente:** Intento de insertar orden de compra con ID inexistente. Resultado esperado: error por clave foránea.

ENTREGA 2:



Las clases que fueron modificadas son: Todas las info extra, se agregó un atributo (fecha expiración) a producto mientras que se separó especificación de empaquetado, se agregó venta y cliente, adicionamos nuevas relaciones, retiramos referencias y corregimos Documento Recepcion.

El diagrama UML representa un sistema de gestión de inventarios enfocado en la administración de sucursales, bodegas, proveedores, productos y órdenes de compra. Este modelo incluye varias entidades clave, como Ciudad, Sucursal, Bodega, Proveedor, OrdenCompra, DocumentoRecepcion, Categoria, Producto y Perecederos, y sus relaciones entre sí.

En el sistema, una Ciudad contiene múltiples Sucursales, cada una de las cuales puede tener una o varias Bodegas. Las Bodegas almacenan Productos organizados en Categorías específicas, las cuales son esenciales para clasificar los productos y facilitar su gestión. Los productos dentro de las bodegas están vinculados a Referencias que indican detalles como la capacidad, el costo promedio, el inventario disponible y el nivel mínimo de reorden. Estas referencias son suministradas por Proveedores, quienes tienen su propia identificación (NIT) y detalles de contacto.

Las Órdenes de Compra se generan para adquirir productos y están asociadas a Referencias específicas, indicando la cantidad, el precio y la fecha de entrega esperada. La recepción de estos productos se documenta mediante un Documento de Recepción, que confirma la

La entidad Producto está relacionada obligatoriamente con una Categoría, asegurando que cada producto pertenezca a una categoría específica. Los productos perecederos, como

subcategoría de producto, requieren un atributo adicional: la Fecha de Vencimiento, lo que es crítico para la gestión de productos con una vida útil limitada.

Cada Referencia está obligatoriamente vinculada a una Bodega y a un Proveedor, indicando que para cada referencia de producto debe existir un proveedor que lo suministre y una bodega que lo almacene. Además, las Órdenes de Compra están asociadas a referencias específicas, asegurando que cada referencia esté presente en al menos una orden de compra. Estas órdenes de compra son verificadas a través de un Documento de Recepción, que confirma la cantidad entregada y la bodega donde se almacenaron los productos.

Las Ventas se asocian con Clientes y sucursales, lo que garantiza un seguimiento eficaz de las transacciones. La información adicional sobre existencias y costos se gestiona a través de InfoExtraBodega y DetalleCostosBodega, proporcionando un control riguroso sobre los costos y existencias en cada bodega.

Ciudad

Ciudad				1NF
CiudadID	Nombre	Codigo		2NF
PK, NN, ND, NC, CK	NC, NN	NC, NN		3NF
				BCNF

- **Atributos:**
 - **CiudadID:** PK, NN, ND, NC (Clave primaria, No nulo, No duplicado, Clave natural).
 - **Nombre:** NC, NN (No duplicado, No nulo).
 - **Codigo:** NC, NN (No duplicado, No nulo).
- **BCNF:** Cada atributo depende completamente de la clave primaria (CiudadID). No hay dependencias funcionales parciales ni transitorias, ya que el Nombre y el Código no dependen de ninguna otra columna.

2. Sucursal

Sucursal							1NF
SucursalID	Nombre	Tamaño	Teléfono	Dirección	CiudadID		2NF
PK, NN, ND, NC, CK	NN	NN	ND	NN	FK, NN, ND, NC, CK		3NF
							BCNF

- **Atributos:**
 - **SucursalID:** PK, NN, ND, NC.
 - **Nombre:** NN.

- **Tamaño:** NN.
- **Teléfono:** ND.
- **Dirección:** NN.
- **CiudadID:** FK, NN, ND, NC, CK.
- **BCNF:** SucursalID es la clave primaria y todos los otros atributos dependen directamente de ella. No hay dependencias que involucren un atributo no clave que determine otro atributo no clave.

3. Info Extra Orden

Info Extra Orden					1NF
Cantidad	CostoUnitario	ProductoID	OrdenCompraID		2NF
NN, CK	NN	FK, NN, ND, NC	FK, NN, ND, UA		3NF
					BCNF

- **Atributos:**
 - **Cantidad:** NN, CK.
 - **CostoUnitario:** NN.
 - **ProductoID:** FK, NN, ND, NC.
 - **OrdenCompraID:** FK, NN, ND, UA.
- **BCNF:** La combinación de ProductoID y OrdenCompraID determina la cantidad y el costo unitario. No hay atributos que dependan de otros atributos no clave, cumpliendo con la condición de BCNF.

4. Bodega

Bodega						1NF
BodegaID	Nombre	Tamaño	Capacidad	SucursalID		2NF
PK, NN, SA	NN	NN	NN	FK, NN, NC, CK, ND		3NF
						BCNF

- **Atributos:**
 - **BodegaID:** PK, NN, SA.
 - **Nombre:** NN.
 - **Tamaño:** NN.
 - **Capacidad:** NN.
 - **SucursalID:** FK, NN, NC, CK, ND.
- **BCNF:** Todos los atributos dependen de BodegaID. La relación con SucursalID asegura que la bodega está vinculada a una sucursal específica, sin dependencias funcionales transitorias.

5. Categoría

Categoría						
CategorialID	Nombre	Descripción	Características almacenamiento	Código		1NF
PK, NN, SA	NN	NN	NN	NN		2NF
						3NF
						BCNF

- **Atributos:**
 - **CategorialID:** PK, NN, SA.
 - **Nombre:** NN.
 - **Descripción:** NN.
 - **Características almacenamiento:** NN.
 - **Código:** NN.
- **BCNF:** Cada atributo es funcionalmente dependiente de CategorialID, sin depender de otros atributos que no sean clave.

6. Producto

Producto												
ProductoID	CostoEnBodega	PrecioUnitario	CantidadPresentación	UnidadMedida	EspecificaciónEmpaquetado	Nombre	Presentación	FechaVencimiento	CódigoBarras	CategorialID	EspecificaciónEmpaquetadoID	
PK, NN, ND, NC	NN	NN	NN, NC	NN, NC	NN, NC	NN, NC	NN, ND, NC	NN, NC	NN, ND, NC	FK, NN, ND	FK, NN, SA	

- **Atributos:**
 - **ProductoID:** PK, NN, ND, NC.
 - **CostoEnBodega:** NN.
 - **PrecioUnitario:** NN.
 - **CantidadPresentación:** NN, NC.
 - **UnidadMedida:** NN, NC.
 - **EspecificaciónEmpaquetado:** NN, NC.
 - **Nombre:** NN, NC.
 - **Presentación:** NN, NC.
 - **FechaVencimiento:** NN, ND, NC.
 - **CódigoBarras:** NN, NC.
 - **CategorialID:** FK, NN, ND.
 - **EspecificaciónEmpaquetadoID:** FK, NN, SA.
- **BCNF:** ProductoID es clave y todos los atributos dependen de él. No hay dependencias de atributos no clave.

6. Producto Perecedero

Producto Perecedero

ProductoID	Date
PK, FK, NN,ND,NC	NN,ND,NC

	1NF
	2NF
	3NF
	BCNF

- **Atributos:**
 - **ProductoID:** PK, FK, NN, ND, NC (Clave primaria, Clave foránea, No nulo, No duplicado, Clave natural).
 - **Date:** NN, ND, NC) (No nulo, No duplicado, Clave Natural).
- **BCNF:** ProductoID es clave primaria y foránea, dependiente de la clave de producto.

7. Proveedor

Proveedor

ProveedorID	Nombre	Dirección	personaContacto	telefonoContacto	NIT
PK, NN, ND, CK	NN	NN, DD	NN, DD	NN, DD	NN, DD

	1NF
	2NF
	3NF
	BCNF

- **Atributos:**
 - **ProveedorID:** PK, NN, ND, CK.
 - **Nombre:** NN.
 - **Dirección:** NN.
 - **PersonaContacto:** DD.
 - **TeléfonoContacto:** DD.
 - **NIT:** DD.
- **BCNF:** ProveedorID determina la identidad de cada proveedor. Todos los demás atributos dependen directamente de esta clave.

8. OrdenCompra

OrdenCompra

OrdenCompraID	Estado	Precio	fechaEntrega	FechaCreacion	ProveedorID	SucursalID
PK, NN, UA, ND	NN, CK	NN, DD	NN	NN, DD	FK, NN, ND	FK, NN, NC, ND

	1NF
	2NF
	3NF
	BCNF

- **Atributos:**
 - **OrdenCompraID:** PK, NN, UA, ND.
 - **Estado:** NN, CK.
 - **Precio:** NN, DD.
 - **FechaEntrega:** NN.
 - **FechaCreacion:** NN, DD.
 - **ProveedorID:** FK, NN, ND.
 - **SucursalID:** FK, NN, NC, ND.

- **BCNF:** OrdenCompraID es la clave primaria, y todos los demás atributos dependen directamente de él, sin dependencias entre atributos no clave.

G. DocumentoRecepción

DocumentoRecepción					
fechaEntrega	ID	OrdenCompraID	BodegaID		1NF
NN	NN	FK, NN, ND, UA	FK, NN, ND		2NF
					3NF
					BCNF

- **Atributos:**
 - **FechaEntrega:** NN.
 - **ID:** NN.
 - **OrdenCompraID:** FK, NN, ND, UA.
 - **BodegaID:** FK, NN, ND.
- **BCNF:** Cada documento de recepción se identifica por su ID, y los demás atributos dependen de él, sin atributos no clave que influyan en la determinación.

InfoExtraProveedor

InfoExtraProveedor	
CantidadExistencias	OrdenCompraID
PK, NN, SA	FK, NN, ND, UA

	1NF
	2NF
	3NF
	BCNF

- **Atributos:**
 - **CantidadExistencias:** PK, NN, SA (Clave primaria, No nulo, Sin atributos adicionales).
 - **OrdenCompraID:** FK, NN, ND, UA (Foreign Key, no nulo, no duplicado, user assigned)
- **BCNF:** Cada cantidad es única y depende de la clave de proveedor.

10. Venta

Venta				
IdVenta	fecha	ID	ClienteID	SucursalID
PK, NN, SA	NN	NN	FK, NN, SA, UA	FK, NN, NC, ND

	1NF
	2NF
	3NF
	BCNF

- **Atributos:**
 - **IdVenta:** PK, NN, SA.
 - **Fecha:** NN.
 - **ID:** NN.
 - **CienteID:** FK, NN, SA, UA.
 - **SucursalID:** FK, NN, NC, ND.
- **BCNF:** IdVenta es la clave única, y todos los atributos dependen de ella, sin dependencias transitorias.

11. Cliente

Cliente					1NF
CienteID	Nombre	Telefono	Cedula		2NF
PK, NN, SA	NN	NN	NN		3NF
					BCNF

- **Atributos:**
 - **CienteID:** PK, NN, SA.
 - **Nombre:** NN.
 - **Teléfono:** NN.
 - **Cédula:** NN.
- **BCNF:** CienteID es la clave única, y no hay atributos que dependan de otros no clave.

12. InfoExtraBodega

InfoExtraBodega							
InfoExtraBodega	TotalExistencias	Capacidad	CostoPromedio	NivelMinimoReorden	CapacidadAlmacenamiento	ProductoID	BodegaID
PK, NN, SA	NN	NN	NN	NN	NN	FK, NN, SA	FK, NN, SA

- **Atributos:**
 - **TotalExistencias:** NN.
 - **Capacidad:** NN.
 - **CostoPromedio:** NN.
 - **NivelMinimoReorden:** NN.
 - **CapacidadAlmacenamiento:** NN.
 - **ProductoID:** FK, NN, SA.
 - **BodegaID:** FK, NN, SA.
- **BCNF:** Los atributos dependen de las claves foráneas, y no hay atributos no clave que afecten.

DetalleCostosBodega

DetalleCostosBodega		
CantidadExistencias	CostoUnitario	InfoExtraBodega
NN	NN	FK, NN, SA

	1NF
	2NF
	3NF
	BCNF

- **Atributos:**
 - **CantidadExistencias:** NN (No nulo).
 - **CostoUnitario:** NN(No nulo)
 - **InfoExtraBodega:** FK, NN, SA (Foreing Key, no nulo, system Assigned)
- **BCNF:** Este atributo depende de la relación entre bodega y productos.

13. EspecificaciónEmpaquetado

EspecificaciónEmpaquetado		
EspecificaciónEmpaquetadoID	Volumen	Peso
PK, NN, SA	NN, CK	NN

	1NF
	2NF
	3NF
	BCNF

- **Atributos:**
 - **EspecificaciónEmpaquetadoID:** PK, NN, SA.
 - **Volumen:** NN.
 - **Peso:** NN.
- **BCNF:** Cada especificación está determinada por su ID y no depende de otros atributos no clave.

ESCENARIOS DE PRUEBA ESENCIALES:

Ciudad

Entidad	Descripción caso	Prueba	Resultado esperado	Notas
Ciudad	Actualizar un registro de Ciudad correcto.	Código ciudad: 1001, Nombre: Bogotá (cambiar a Bogotá D.C.)	Código ciudad: 1001	Actualización correcta.

	Eliminar un registro de Ciudad existente.	Código ciudad: 1001	El registro se elimina exitosamente.	Confirmar que ya no está en la base de datos.
	Intentar eliminar un registro de Ciudad que tiene sucursales asociadas.	Código ciudad: 1002	Falla la eliminación debido a la restricción de clave foránea.	No se puede eliminar, aún hay referencias.

Sucursal

Entidad	Descripción caso	Prueba	Resultado esperado	Notas
Sucursal	Actualizar un registro de Sucursal correcto	Nombre: Sucursal Centro (cambiar a Sucursal Centro	El registro se actualiza exitosamente.	Verificar que el nombre ha cambiado.
Sucursal	Eliminar una Sucursal que no tiene referencias.	Nombre: Sucursal Norte	El registro se elimina exitosamente.	Confirmar que ya no está en la base de datos.
Sucursal	Intentar eliminar una Sucursal que tiene productos asociados.	Nombre: Sucursal Centro	Falla la eliminación debido a la restricción de clave foránea.	No se puede eliminar, aún hay productos asociados.

Referencia

Entidad	Descripción caso	Prueba	Resultado esperado	Notas
Referencia	Actualizar un registro de Referencia correcto.	ID producto: 1234, Inventario: 60 (cambiar a 70)	El registro se actualiza exitosamente.	Confirmar que el inventario ha cambiado.
Referencia	Eliminar un registro de Referencia existente.	ID producto: 1234	El registro se elimina exitosamente.	Verificar que ya no está en la base de datos.
Referencia	Intentar eliminar una referencia que tiene productos asociados.	ID producto: 1235	Falla la eliminación debido a la restricción de clave foránea.	No se puede eliminar, hay productos asociados.

Bodega

Entidad	Descripción del caso	Prueba	Resultado esperado	Notas
Bodega	Actualizar un registro de Bodega correcto.	Nombre: Bodega Principal (cambiar a Bodega Central)	El registro se actualiza exitosamente.	Verificar que el nombre ha cambiado.
Bodega	Eliminar un registro de Bodega existente.	Nombre: Bodega Principal	El registro se elimina exitosamente.	Confirmar que ya no está en la base de datos.
Bodega	Intentar eliminar una Bodega que tiene referencias asociadas.	Nombre: Bodega Secundaria	Falla la eliminación debido a la restricción de clave foránea.	No se puede eliminar, hay referencias asociadas.

Categoría

Entidad	Descripción del caso	Prueba	Resultado esperado	Notas
Categoría	Actualizar un registro de Categoría correcto.	CódigoCategoría: CAT001, Nombre: Electrónica (cambiar a Electrónica Avanzada)	El registro se actualiza exitosamente.	Verificar que el nombre ha cambiado.
Categoría	Eliminar un registro de Categoría existente.	CódigoCategoría: CAT001	El registro se elimina exitosamente.	Confirmar que ya no está en la base de datos.
Categoría	Intentar eliminar una Categoría que tiene productos asociados.	CódigoCategoría: CAT002	Falla la eliminación debido a la restricción de clave foránea.	No se puede eliminar, hay productos asociados.

Producto

Entidad	Descripción del caso	Prueba	Resultado esperado	Notas
Producto	Actualizar un registro de Producto correcto.	CodigoBarras: 0011223344, CostoEnBodega: 50 (cambiar a 55)	El registro se actualiza exitosamente.	Confirmar que el costo ha cambiado.
Producto	Eliminar un registro de Producto existente.	CodigoBarras: 0011223344	El registro se elimina exitosamente.	Verificar que ya no está en la base de datos.
Producto	Intentar eliminar un Producto que tiene ventas asociadas.	CodigoBarras: 0011223345	Falla la eliminación debido a la restricción de clave foránea.	No se puede eliminar, hay ventas asociadas.

Producto Perecedero

Entidad	Descripción del caso	Prueba	Resultado esperado	Notas
Producto Perecedero	Actualizar un registro de Producto Perecedero correcto.	CodigoBarras: 1122334455, CostoEnBodega: 20 (cambiar a 25)	El registro se actualiza exitosamente.	Confirmar que el costo ha cambiado.
Producto Perecedero	Eliminar un registro de Producto Perecedero existente.	CodigoBarras: 1122334455	El registro se elimina exitosamente.	Verificar que ya no está en la base de datos.
Producto Perecedero	Intentar eliminar un Producto Perecedero que tiene ventas asociadas.	CodigoBarras: 1122334456	Falla la eliminación debido a la restricción de clave foránea.	No se puede eliminar, hay ventas asociadas.

Proveedor

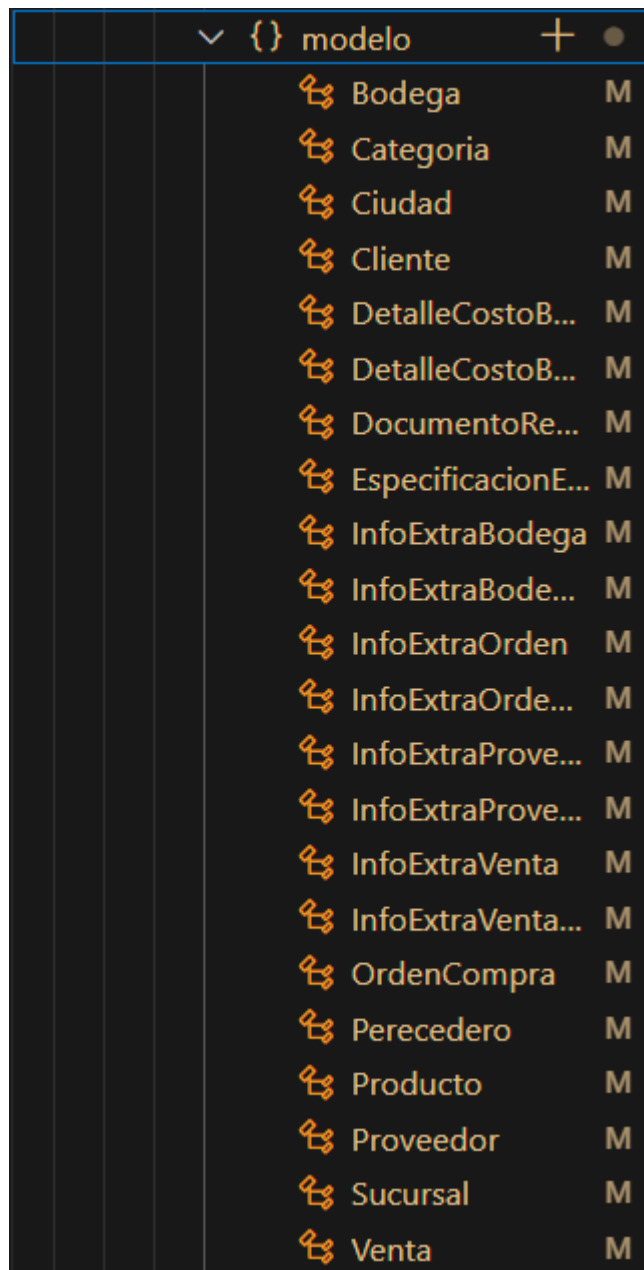
Entidad	Descripción del caso	Prueba	Resultado esperado	Notas
Proveedor	Actualizar un registro de Proveedor correcto.	NIT-Proveedor: 8901234567 (cambiar a 8901234568)	El registro se actualiza exitosamente.	Confirmar que el NIT ha cambiado.
Proveedor	Eliminar un registro de Proveedor existente.	NIT-Proveedor: 8901234567	El registro se elimina exitosamente.	Verificar que ya no está en la base de datos.
Proveedor	Intentar eliminar un Proveedor que tiene órdenes asociadas.	NIT-Proveedor: 8901234568	Falla la eliminación debido a la restricción de clave foránea.	No se puede eliminar, hay órdenes asociadas.

Orden Compra

Entidad	Descripción del caso	Prueba	Resultado esperado	Notas
Orden Compra	Actualizar un registro de Orden Compra correcto.	ID compra: 10001, Estado: Pendiente (cambiar a Completada)	El registro se actualiza exitosamente.	Confirmar que el estado ha cambiado.
Orden Compra	Eliminar un registro de Orden Compra existente.	ID compra: 10001	El registro se elimina exitosamente.	Verificar que ya no está en la base de datos.
Orden Compra	Intentar eliminar una Orden Compra que tiene productos asociados.	ID compra: 10002	Falla la eliminación debido a la restricción de clave foránea.	No se puede eliminar, hay productos asociados.

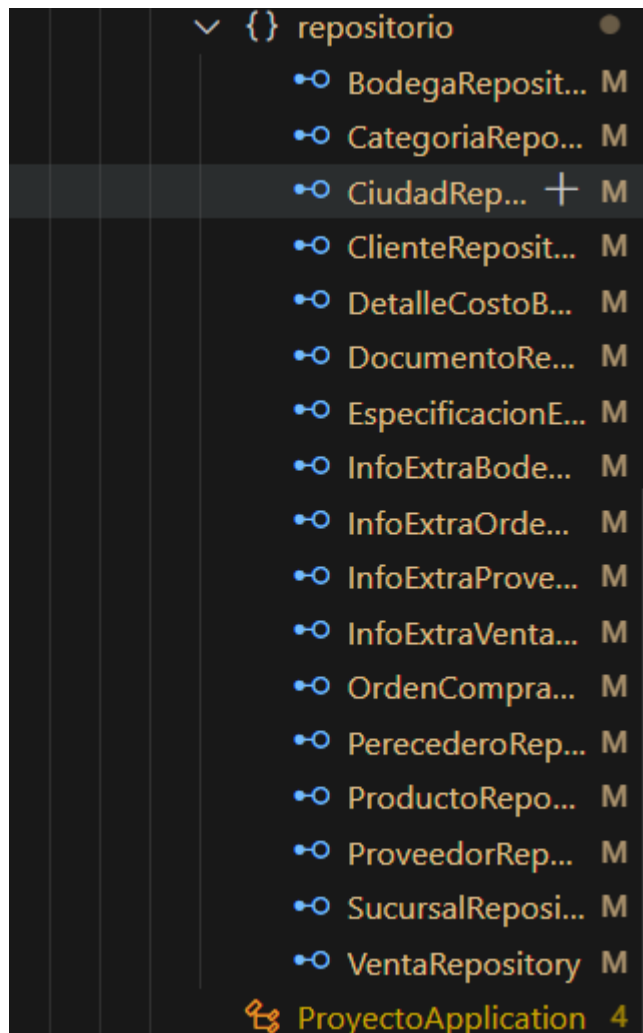
Documentación de las clases de Java Spring y de la arquitectura de la aplicación:

APLICACIÓN VISUAL DOCUMENTACIÓN:

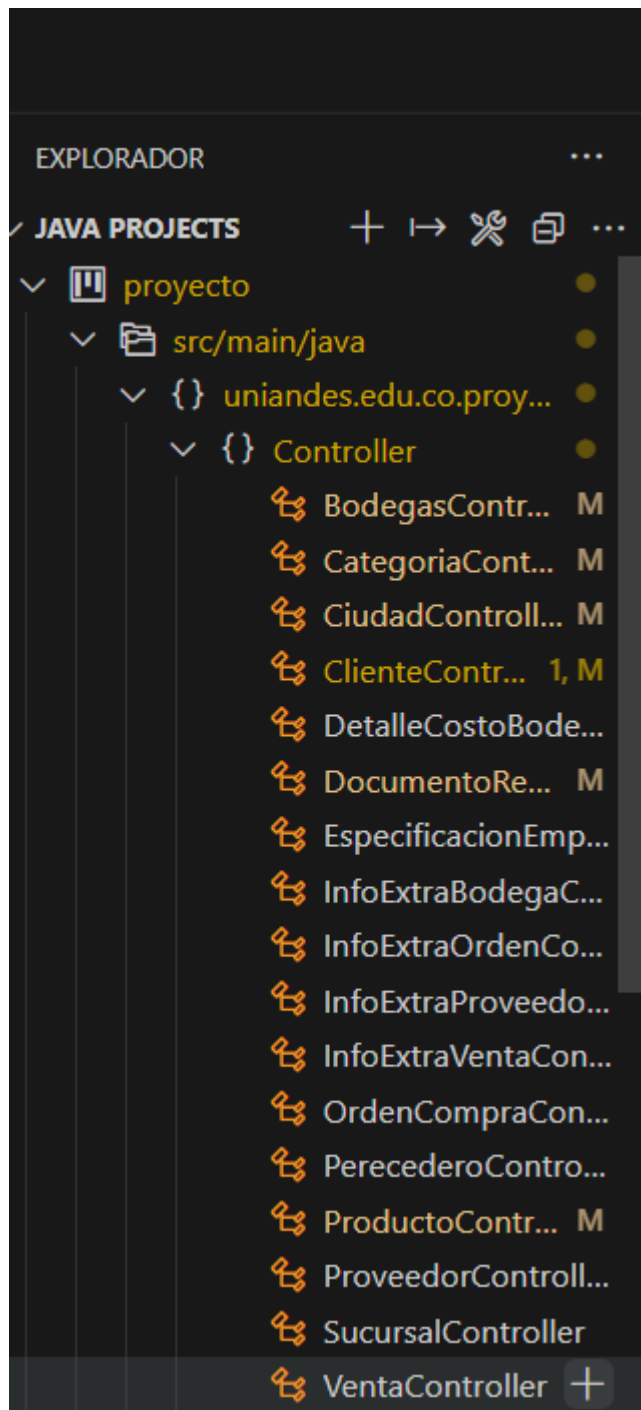


▼ {}	modelo	+	●
	Bodega	M	
	Categoria	M	
	Ciudad	M	
	Cliente	M	
	DetalleCostoB...	M	
	DetalleCostoB...	M	
	DocumentoRe...	M	
	EspecificacionE...	M	
	InfoExtraBodega	M	
	InfoExtraBode...	M	
	InfoExtraOrden	M	
	InfoExtraOrde...	M	
	InfoExtraProve...	M	
	InfoExtraProve...	M	
	InfoExtraVenta	M	
	InfoExtraVenta...	M	
	OrdenCompra	M	
	Perecedero	M	
	Producto	M	
	Proveedor	M	
	Sucursal	M	
	Venta	M	

Todos los modelos de las clases representan entidades fundamentales dentro de un sistema de gestión de inventarios y operaciones comerciales en una base de datos. Cada clase está diseñada para mapear a una tabla específica en la base de datos, permitiendo la persistencia de datos y facilitando operaciones de creación, lectura, actualización y eliminación (CRUD). La clase Bodega, por ejemplo, almacena información sobre las distintas ubicaciones de almacenamiento, incluyendo atributos como el nombre, tamaño y capacidad, mientras que la clase Categoria organiza los productos en diferentes categorías con descripciones y características de almacenamiento. Este enfoque orientado a objetos promueve una estructura clara y coherente, donde las relaciones entre las diferentes entidades, como Sucursal, Producto, Venta y otras, pueden definirse mediante asociaciones, como las relaciones de uno a muchos o muchos a uno.

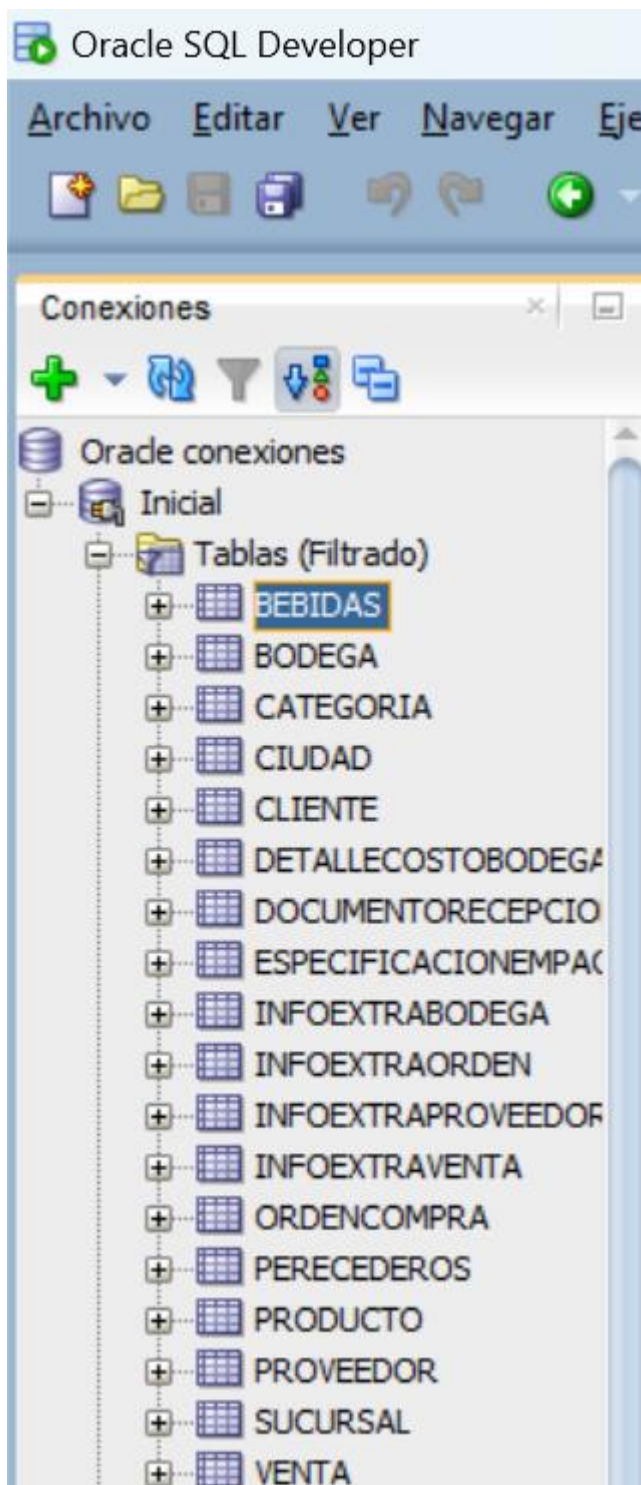


Los repositorios definidos para los modelos en la base de datos proporcionan una interfaz para realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre cada entidad. Cada repositorio, como BodegaRepository, CategoriaRepository, CiudadRepository, ClienteRepository y DetalleCostoBodegaRepository, utiliza Spring Data JPA para facilitar la interacción con la base de datos mediante consultas nativas y métodos personalizados. Estos repositorios permiten obtener colecciones de entidades, buscar entidades específicas por su identificador, insertar nuevas entradas y actualizar o eliminar registros existentes. Además, se utilizan anotaciones como @Modifying y @Transactional para asegurar que las operaciones de modificación de datos se realicen de manera efectiva dentro de transacciones, garantizando la integridad de la base de datos. En conjunto, estos repositorios ofrecen un acceso estructurado y eficiente a la información almacenada, facilitando el desarrollo de la lógica de negocio relacionada con las distintas entidades del sistema.

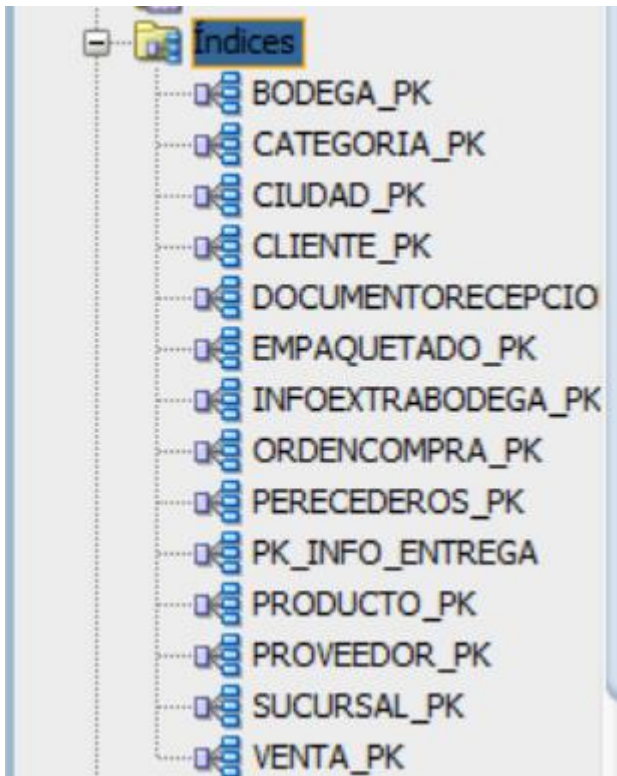


Los controladores son responsables de manejar las solicitudes HTTP relacionadas con las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) para las entidades respectivas generando respuestas coherentes a problemas puntuales. Cada controlador se conecta a su correspondiente repositorio, que actúa como una capa de acceso a datos, facilitando la interacción con la base de datos. A través de sus métodos correspondientes se pueden recuperar colecciones de estas entidades. Los métodos insertar, actualizar y eliminar permiten gestionar las entidades de forma persistente, asegurando que se puedan crear nuevas entradas, modificar existentes y eliminar registros según su ID. Además, el manejo de excepciones dentro de los controladores garantiza que, en caso de errores, se proporcionen respuestas adecuadas, lo que mejora la robustez del sistema y la experiencia del usuario. Con esta arquitectura, se puede gestionar de manera eficiente el ciclo de vida de las entidades dentro de una base de datos, permitiendo mantener la integridad y disponibilidad de la información.

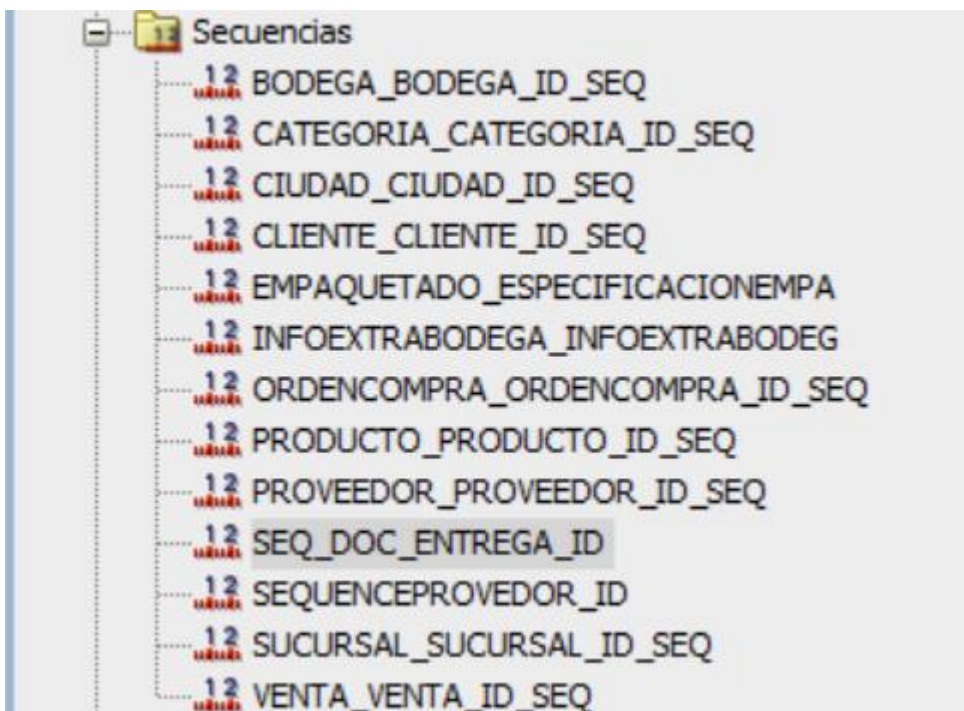
ORACLE DOCUMENTACION:



Dentro de las propiedades de cada tabla, se encuentran los atributos correspondientes a cada entidad y una tabla representativa de la entidad, más allá de esto, estas tablas resumen las transferencias de información entre métodos de cada entidad y sus relacionamientos directos a la hora de requerirlos.



Dentro de estos índices se encuentran las llaves primarias de cada entidad, basándonos en esto es posible generar ejemplos y relacionamientos entre las diferentes tablas de datos generadas, más allá de esto, es posible tener en consideración los datos mostrados en las tablas de los índices como una guía a futuros ejemplos.



Dentro de estas secuencias, se encuentran unos ejemplos básicos de lo que sería la creación de datos para cada ejemplo de ID, es una secuencia de datos organizados directamente a la función de especificar cambios dentro de las tablas de datos y un breve ciclo de información.

TRAZOS DE INFORMACIÓN Y METAS LOGRABLES:

The screenshot shows the Postman interface with the 'Scripts' tab selected. The 'Post-res' section contains a script that tests the status code of the response. The 'Test Results' tab shows a single result: 'PASS Status code is 201'. The status '201 Created' is highlighted in green, indicating a successful creation.

```
1 pm.test("Status code is 201", function () {
2   pm.response.to.have.status(201);
3 });
```

201 Created • 698 ms • 198 B • [Globe Icon] [e.g.] [More Icon]

Perform API tests faster with templates for integration testing, regression testing, and more. [View Templates](#)

All Passed Skipped Failed [Refresh Icon]

PASS Status code is 201

Este código rectifica la creación de un proveedor dentro del sistema esperado

The screenshot shows the Postman interface with the 'Body' tab selected. The response status is '200 OK'. The response body is displayed in JSON format, showing details for a product category.

Params Auth Headers (7) Body Scripts Settings [Cookie](#)

none ▾

This request does not have a body

Body ▾ 200 OK • 324 ms • 373 B • [Globe Icon] [e.g.] [More Icon]

Pretty Raw Preview Visualize JSON ▾ [Copy Icon] [Search Icon]

```
1 {
2   "id": 14,
3   "nombre": "Productos Congelados",
4   "descripcion": "Alimentos que se almacenan a bajas temperaturas para
5     evitar su deterioro.",
6   "caracteristicasAlmacenamiento": "Mantener en congeladores a -18 °C o
7     menos."
8 }
```

Esta prueba realiza la lectura de una categoría con atributos específicos para una mejor especificación

POST localhost:8080/categorias/new/save Send

Params Auth Headers (9) Body JSON Scripts Settings Cookies

raw Beautify

```
1 {
2   "nombre": "Prodcutos Tecnologicos",
3   "descripcion" : "Productos de tecnologia moderna",
4   "Caracteristicas almacenamiento" : "Productos fragiles"
5 }
6 }
```

Body 201 Created • 712 ms • 198 B • e.g. ...

Pretty Raw Preview Visualize Text 1 Categoria creada exitosamente

Este código genera una categoría y crea su historial para el servicio de la base de datos

PUT localhost:8080/proveedores/41/edit/save Send

Params Auth Headers (9) Body JSON Scripts Settings Cookies

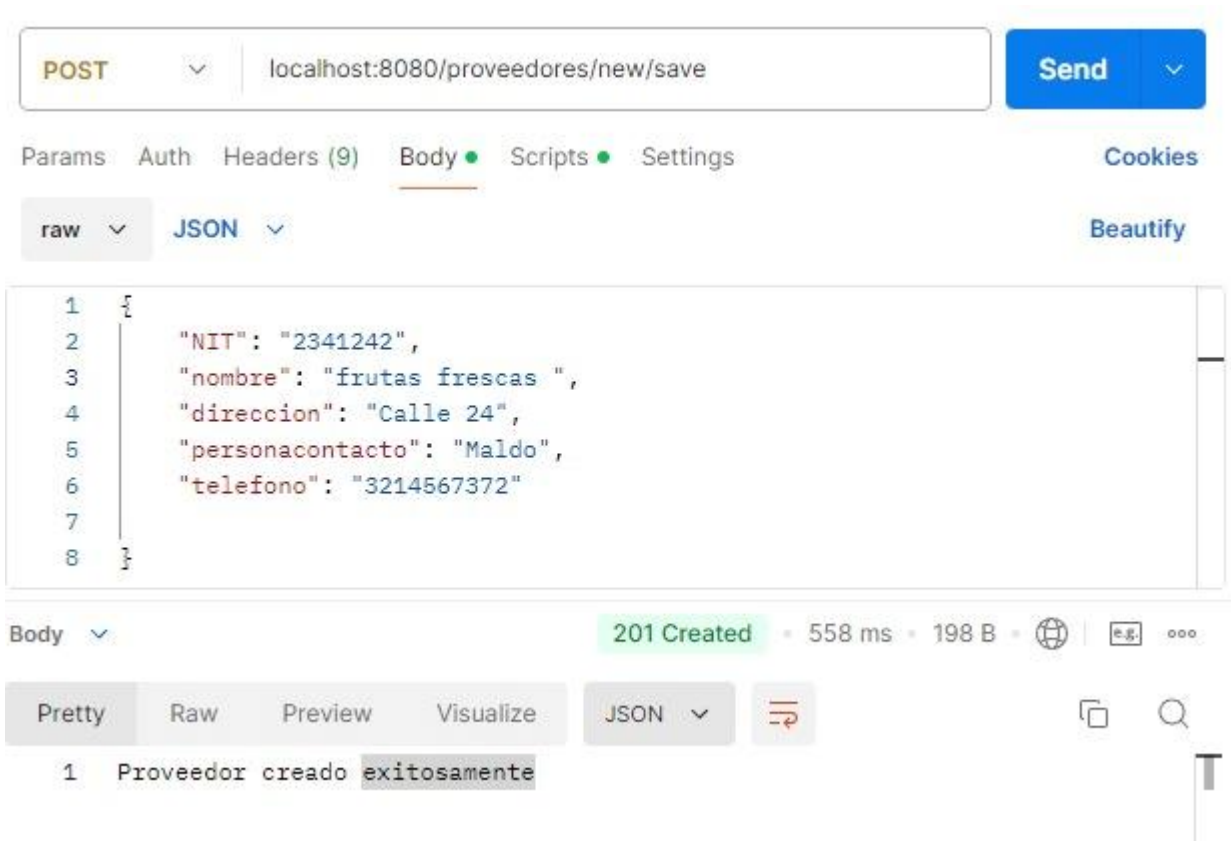
raw Beautify

```
1 {
2   "id": 41,
3   "NIT": "548",
4   "nombre" : "Sebastian",
5   "direccion" : "Cr 100 # 99A-98",
6   "personaContacto" : "camilo",
7   "telefono" : "3254564"
8 }
```

Body 200 OK • 644 ms • 199 B • e.g. ...

Pretty Raw Preview Visualize Text 1 Proveedor actualizado correctamente

Esta prueba generaliza la actualización de un proveedor



Esta parcela de código realiza la creación de un proveedor para su uso en el sistema

CONCLUSIÓN EN BASE A LOS PANTALLAZOS:

La mayor parte de los requerimientos estándar fueron logrados, la mayor cantidad de pruebas Postman y en Oracle fueron posibles. La base de datos tiene una conexión fuerte con lo que sería el código principal dentro de los controladores y los repositorios dentro de los cuáles fue posible enlazarlos con la base de Oracle y en Postman para poder realizar las pruebas correspondientes. Lo que no fue lograble dentro de este proyecto fue generar detalles específicos dentro de las pruebas y el funcionamiento correspondiente de la aplicación, sin embargo, no son problemas de gravedad a la larga, puesto que solo corresponden a problemas estéticos y superficiales. Más allá de esto, las metas principales del proyecto fueron completadas, en realidad es posible asegurar que el programa cumple con las ideas necesarias.

▪ Nombre del usuario en Oracle en donde se encuentran las tablas del modelo físico del proyecto con el fin de comprobar que están pobladas y que durante la ejecución de las pruebas en Postman se modifican los datos de manera consistente.

CREDENCIALES USUARIO ORACLE CON FUNCIONAMIENTOS ACTUALIZADOS:

jdbc:oracle:thin:@fn4.oracle.virtual.uniandes.edu.co:1521/PROD
username=ISIS2304A27202420
password=SrFSChbQxm