

FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS
SISTEMAS TRANSACCIONALES

Entrega 1 - Diseño

Juan Miguel Delgado – 202314903

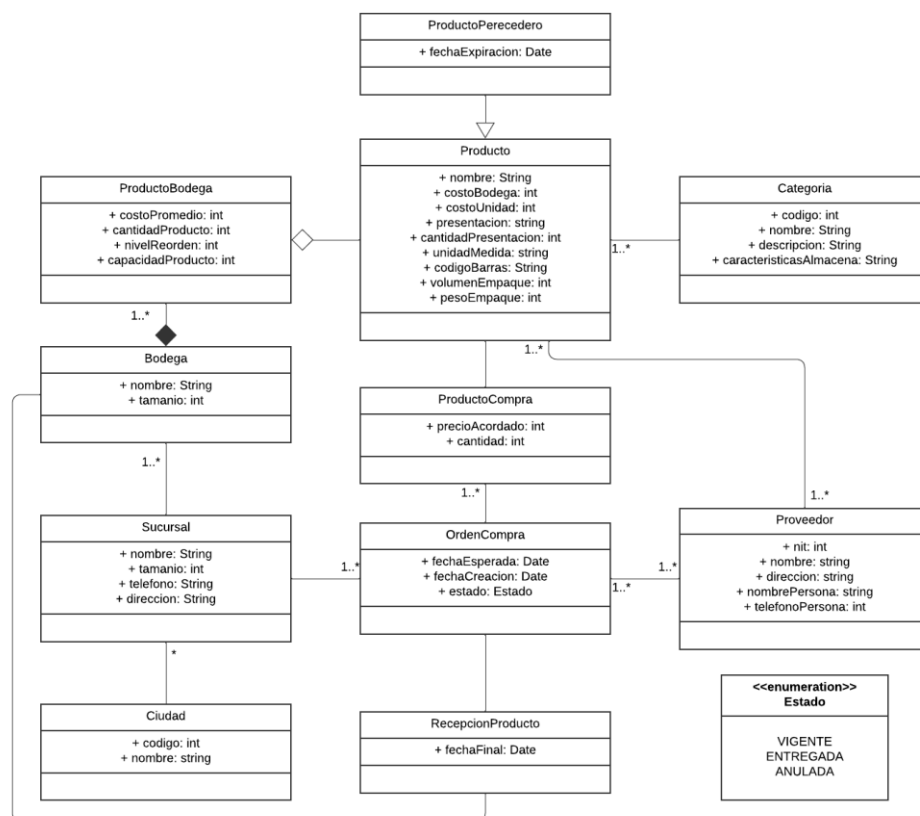
Johan Camilo Suarez Sinisterra - 202214091

Kalia Angelica Gonzalez Jimenez- 202226917

1. Análisis y Modelo Conceptual

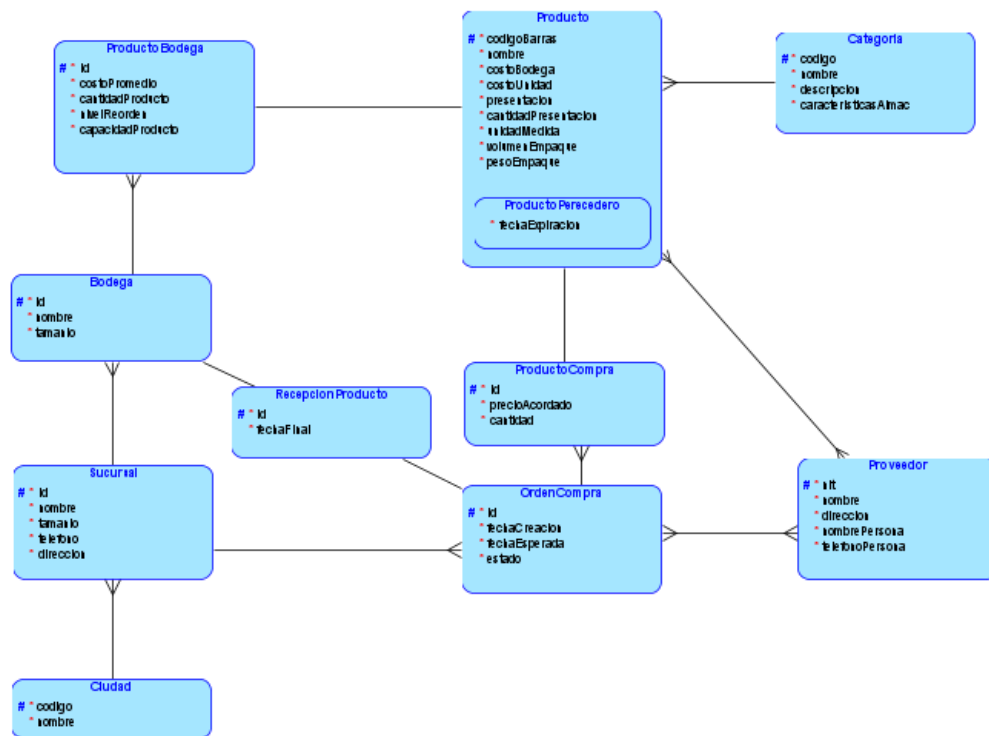
a. Modelo conceptual en UML

A continuación, se presenta el modelo conceptual que muestra y abstrae el negocio que lleva a cabo SuperAndes.



b. Modelo Entidad/Relación

A continuación, se presenta el modelo Entidad/Relación que se obtiene a traducir el modelo conceptual UML. En este prototipo, se especifican los atributos que serán usados como llaves primarias para el modelo relacional.



c. Modelo de datos relacional

El modelo relacional se obtiene a partir de la traducción del modelo conceptual. Este, debido a su tamaño, se encuentra en el repositorio de GitHub.

2. Diseño de la Base de Datos

Haciendo el uso del algoritmo de Chen, se puede traducir el modelo conceptual UML propuesto a un modelo relacional que abstrae y represente la información de manera detallada y precisa. En primer lugar, se pueden identificar las tablas bases las cuales representan las 10 clases principales que componen el modelo conceptual: Producto, Bodega, Categoría, Proveedor, Sucursal, Ciudad, OrdenCompra, ProductoBodega, ProductoCompra y RecepcionProducto. Entonces, por cada clase se crea una tabla con los mismos atributos expuestos en el UML.

Hasta aquí, se puede identificar la llave primaria (PK) de todas las relaciones que representan clases que no heredan de otras clases: Para Producto su PK será el codigoBarras, donde esta adicionalmente va a estar asignada por el sistema (SA). Para las clases Bodega, Categoría, Sucursal, OrdenCompra, ProductoBodega, ProductoCompra y RecepcionProducto la llave primaria será el id (PK) donde este nuevamente será asignado por el sistema (SA). Para Ciudad, su primary key es el código que esta tiene y, como en las anteriores clases, el sistema será encargado de asignarle dicha llave. Por último, la clase Proveedor tiene encargado al nit para ser su PK y que esta adicionalmente debe cumplir con que el usuario le asigne la llave (UA).

Por otro lado, nos fijamos que la clase ProductoPerecedero hereda de la clase Producto, en este caso la llave primaria de la clase hija será la misma que tiene la clase padre, es decir codigoBarras.

Producto - ProductoPerecedero: heredaProducto

heredaProducto	
codigoBarras	fechaExpiracion
PK,FKProducto.codigoBa	NN,CK
12341113	15/09/2025

De este modo, la llave primaria de la subclase también es una llave foránea (FK) que hace referencia a la tabla padre. Para los demás atributos que no son llaves primarias, solo se les pone las características de NN (Not Null) y CK (Check).

Ahora, se prosigue a modelar las asociaciones entre clases:

- **Bodega - ProductoBodega: bodAsignada (1 a 1):** Para esta relación una bodega tiene asignado el detalle de un producto cuyo destino será esa misma bodega. Para representar esta asociación, en la tabla de ProductoBodega se crea una nueva columna cuyo atributo es la bodega asignada del producto (bodAsignada) donde la conexión se realiza con la llave foránea (Bodega.id) que es la PK de la clase Bodega.

Bodega - ProductoBodega: bodAsignada

bodAsignada					
id	costoPromedio	cantidadProducto	nivelReorden	capacidadProducto	bodAsignada
PK,SA	NN	NN,CK	NN,CK	NN,CK	NN,FKBodega.id
PB_001	9800	24	20	40	BOD_01
PB_002	19700	36	10	50	BOD_01
PB_003	8100	127	40	250	BOD_01

- **Sucursal - Bodega: sucAsignada (1...*):** Para esta relación una sucursal tiene una o varias bodegas asociadas. Para esta relación, en nuestra tabla Bodega se le agrega una nueva columna denominada sucAsignada que hace referencia a la sucursal que está ligada a dicha bodega. Para realizar la conexión, se tiene como llave foránea a la primary key de la clase Sucursal que es el id (Sucursal.id) .

Sucursal - Bodega: sucAsignada

sucAsignada				
id	nombre	tamano	capacidad	sucAsignada
PK,SA	NN	NN,CK	NN,CK	NN,FKSucursal.id
BOD_01	Bodega Central	100	400	402

- **Ciudad - Sucursal: ciuAsignada (1...*):** Para esta relación, una ciudad puede tener varias sucursales, y varias sucursales pueden estar en una ciudad. Para representar esta conexión, en la tabla de Sucursal se le agrega una nueva columna que representa la ciudad donde se encuentra dicha sucursal (ciuAsignada) y cuya llave foránea que va a crear la relación es el código de la ciudad (Ciudad.codigo).

Ciudad - Sucursal: ciuAsignada

ciuAsignada					
id	nombre	tamaño	telefono	direccion	ciuAsignada
PK,SA	NN	NN	NN,ND	NN,ND	NN,FKCiudad.codigo
S_001	Sucursal Norte	100	5556789	Av. Siempreviva 742	C_001
S_002	Sucursal Sur	150	5559876	Calle Principal 123	C_001

- **ProductoBodega - Producto: prodBodega(1 a 1):** Para esta relación, el detalle de un producto en bodega tiene si o si solo un producto asociado. Para representar esta relación, se tiene el atributo prodBodega que representa dicho producto que será guardado. Y para realizar la conexión, se tiene que la llave foránea (FK) es el código de barras del producto (Producto.codigoBarras).

ProductoBodega - Producto: prodBodega

prodBodega					
id	costoPromedio	cantidadProducto	nivelReorden	capacidadProducto	prodBodega
PK,SA	NN	NN,CK	NN,CK	NN,CK	ND,FKProducto.codigoBarras
PB_001	9800	24	20	40	12341111
PB_002	19700	36	10	50	12341112
PB_003	8100	127	40	250	12341113

- **Categoría- Producto: catAsignada (1...*):** Para esta relación, una categoría tiene uno o varios productos asociados. Para representar esta relación, se tiene una nueva columna en la clase Producto donde indica el tipo de categoría (catAsignada) y cuya llave foránea que nos va a conectar con la tabla de Categoría es el id (Categoría.id).

Producto - Categoría: catAsignada

catAsignada									
codigoBarras	nombre	costoBodega	costoUnidad	presentacion	cantidadPresentacion	unidadMedida	volumenEmpaque	pesoEmpaque	catAsignada
PK,SA	NN	NN,CK	NN,CK	NN	NN,CK	NN	NN,CK	NN,CK	NN,FKCategoría.id
12341111	Producto1	10.000	12.000	Caja	10	kg	10	125	CAT_001
12341112	Producto2	15.000	18.000	Bolsa	5	lt	20	235	CAT_002
12341113	Producto3	5.000	8.000	Paquete	20	gr	7	75	CAT_001
12341114	Producto4	8.000	9.500	Caja	12	kg	15	90	CAT_001
12341115	Producto5	20.000	25.000	Saco	50	kg	50	600	CAT_003

- **ProductoCompra - Producto: prodCompra(1 a 1):** Para esta relación, el detalle de un producto de una orden compra está asociado a un producto. Para esto en la tabla de ProductoCompra se le agrega una nueva columna prodCompra que representa esta relación (prodCompra). Adicionalmente, la llave foránea será el código de barras del producto al cual se le quiere acordar el precio (Producto.codigoBarras).

ProductoCompra - Producto: prodCompra

prodCompra			
id	precioAcordado	cantidad	prodCompra
PK,SA	NN,CK	NN	ND,FKProducto.codigoBarras
PC_001	8500	30	12341111
PC_002	17000	80	12341115

- **OrdenCompra - ProductoCompra : ordAsignada(1...*):** Para esta relación, una orden de compra tiene uno o varios detalles de productos asociados para poder tener características de cada producto que se quiere comprar. Para esto en la tabla de ProductoCompra se le agrega una nueva columna ordCompra que representa la orden de compra a la que esta asignado el detalle del producto. Adicionalmente, la llave foránea será el id del orden de compra (OrdenCompra.id).

OrdenCompra - ProductoCompra: ordAsignada

ordAsignada			
id	precioAcordado	cantidad	ordAsignada
PK,SA	NN,CK	NN	NN,FKOrdenCompra.id
PC_001	8500	30	OC_001
PC_002	17000	80	OC_001

- **Sucursal - OrdenCompra: sucAsignada(1...*)**: Para esta relación, una sucursal puede hacer una o varias órdenes de compra. Para esto en la tabla de OrdenCompra se le agrega una nueva columna que representa esta relación (sucAsignada). Adicionalmente, la llave foránea será el id de la sucursal donde se realizará las diferentes órdenes de compra (Sucursal.id).

OrdenCompra - Sucursal: sucAsignada

sucAsignada				
id	fechaEsperada	fechaEsperada	estado	sucAsignada
PK,SA	NN,CK,SA	NN,CK	NN	NN,FK,sucursalid
OC_001	1/11/2024	1/12/2024	VIGENTE	S_001
OC_002	15/10/2024	15/11/2024	VIGENTE	S_001

- **Proveedor - OrdenCompra: asignadoEnOrden(*...*)**: Para esta relación, uno o varios proveedores están asignados a una o varias órdenes de compra hechas por una sucursal, queriendo decir que tenemos una multiplicidad de muchos a muchos. Para esto se crea una tercera tabla cuya llave primaria será el id del orden de compra (OrdenCompra.id) y la llave foránea que me va a conectar con la tabla de Proveedor será el nit de este (Proveedor.nit). Esta FK al mismo tiempo será una PK.

Proveedor - OrdenCompra: asignadoEnOrden

asignadoEnOrden	
id	nitProveedor
PK,FKOrdenCompra.id	PK,FKProveedor.nit
OC_001	123456789
OC_002	123456780
OC_001	123456780
OC_002	123456789

- **Proveedor - Producto: vendeProducto(*...*)**: Para esta relación, varios proveedores (Proveedor) pueden vender varios productos (Producto), queriendo decir que tenemos una multiplicidad de muchos a muchos. Para esto se crea una tercera tabla cuya llave primaria será el nit del proveedor (Proveedor.nit) y la llave foránea que me va a conectar con la tabla del Producto será el código de barras de este (Producto.codigoBarras). Esta FK al mismo tiempo será una PK.

Proveedor - Producto: vendeProducto

vendeProducto	
nit	codBarrasProducto
PK,FKProveedor.nit	PK,FKProducto.codigoBarras
123456789	12341111
123456780	12341111
123456780	12341112
123456789	12341112

3. Nivel de Normalización del modelo:

Para determinar correctamente el nivel de normalización del modelo relacional propuesto resulta necesario conocer las dependencias funcionales asociadas a cada relación y, ya con esta información, evaluar cada relación para determinar el grado de normalización que tiene actualmente.

a. **Relación Producto**: La definición de relación y de sus dependencias funcionales es la siguiente:

Producto {codigoBarra, nombre, costoBodega, costoUnidad, presentación, cantidadPresentacion, unidadMedia, volumenEmpaque, pesoEmpaque}

Dependencias:

-codigoBarra -> nombre, costoBodega, costoUnidad, presentación, cantidadPresentacion, unidadMedia, volumenEmpaque, pesoEmpaque.

Llave candidata:

Luego de analizar los atributos de la clase, se llega a la siguiente llave:

-{codigoBarra}

- Primera Forma Normal (1NF)

La relación se encuentra en primera forma normal ya que esta no cuenta con atributos multivariados. Es decir que contengan listas o algún tipo de estructura similar.

- Segunda Forma Normal (2NF)

La relación como ya se encuentra en 1NF, prosigue a ser evaluada para mirar si no se tienen dependencias parciales desde el atributo primo (codigoBarra). Analizando la clase, el código de barras básicamente es el encargado de guardar la información detallada de cada producto. Entonces, al saber cuál es este código, puedo saber su precio, nombre, presentación y demás atributos. Por lo tanto, se puede afirmar que la relación está en 2NF.

- Tercera Forma Normal (3NF)

Como la relación cumple con las dos reglas anteriores, se prosigue a analizar si está en la regla 3NF. En esta se evalúa que no exista alguna dependencia transitiva. Con la llave primaria, un atributo NO determina otro primo. Analizando todos los atributos junto con la llave primaria, no se ve otra dependencia o relación que puedan existir entre los demás atributos ya que estos son solo características del producto, mas no son los que guardan la información de él.

- Forma Normal Boyce-Codd:

La relación está en 3NF, y adicionalmente las llaves son simples y no dependen de otra llave, se puede decir que se encuentra en FNBC.

b. Relación Bodega: La definición de relación y de sus dependencias funcionales es la siguiente:

Bodega{id,nombre,tamano,capacidad}

Dependencias:

-id -> nombre, tamano, capacidad

-nombre -> id, tamaño, capacidad.

Llaves candidatas:

Luego de analizar los atributos de la clase, se llega a las siguientes llaves:

-{id}

-{nombre}

- Primera Forma Normal (1NF)

La relación se encuentra en primera forma normal ya que esta no cuenta con atributos multivariados. Es decir que contengan listas o algún tipo de estructura similar.

- Segunda Forma Normal (2NF)

La relación como ya se encuentra en 1NF, prosigue a ser evaluada para mirar si no se tienen dependencias parciales desde los atributos primos (id y nombre). Analizando la clase, una bodega tiene un identificador único (conocido como id), donde contiene la información detallada de la bodega. A la vez, posee un nombre único que nos podría dar información de los demás atributos. Por lo tanto, se puede afirmar que la relación está en 2NF.

- Tercera Forma Normal (3NF)

Como la relación cumple con las dos reglas anteriores, se prosigue a analizar si está en la regla 3NF. En esta se evalúa que no exista alguna dependencia transitiva. Con la llave primaria, un atributo NO determina otro primo. Analizando todos los atributos junto con las llaves primarias, no se ve otra dependencia o relación que puedan existir entre los demás atributos ya que estos son solo características de la bodega, mas no son los que guardan la información de ella. Por ejemplo, no se puede determinar el tamaño de la bodega con una llave diferente a la primaria (id o nombre).

- Forma Normal Boyce-Codd:

La relación está en 3NF, y adicionalmente las llaves son simples y no dependen de otra llave, se puede decir que se encuentra en FNBC.

c. **Relación Categoría:** La definición de relación y de sus dependencias funcionales es la siguiente:

Categoría {id, nombre, descripción, característicasAlmac}

Dependencias:

-id -> nombre, descripción, característicasAlmac

Llave candidata:

Luego de analizar los atributos de la clase, se llega a la siguiente llave:

-{id}

- Primera Forma Normal (1NF)

La relación se encuentra en primera forma normal ya que esta no cuenta con atributos multivariados. Es decir que contengan listas o algún tipo de estructura similar.

- Segunda Forma Normal (2NF)

La relación como ya se encuentra en 1NF, prosigue a ser evaluada para mirar si no se tienen dependencias parciales desde el atributo primo (id). Analizando la clase, una categoría al tener un id que los identifique y distingue de las demás categorías a pesar de tener mismo nombre, muestra la dependencia de la otra información que se necesita saber. Por lo tanto, se puede afirmar que la relación está en 2NF.

- Tercera Forma Normal (3NF)

Como la relación cumple con las dos reglas anteriores, se prosigue a analizar si está en la regla 3NF. En esta se evalúa que no exista alguna dependencia transitiva. Con la llave primaria, un atributo NO determina otro primo. Analizando todos los atributos junto con la llave primaria, no se ve otra dependencia o relación que puedan existir entre los demás atributos ya que estos son solo características de la categoría, mas no son los que guardan la información de ella. Por ejemplo, no se puede determinar una descripción de una categoría sin saber su id.

- Forma Normal Boyce-Codd

La relación está en 3NF, y adicionalmente las llaves son simples y no dependen de otra llave, se puede decir que se encuentra en FNBC.

d. **Relación Proveedor:** La definición de relación y de sus dependencias funcionales es la siguiente:

Proveedor {nit, nombre,direccion,nombrePersona,telefonoPersona }

Llave candidata: Luego de analizar los atributos de la clase, se llega a la siguiente llave:

-{nit}

- Primera Forma Normal (1NF)

La relación se encuentra en primera forma normal ya que esta no cuenta con atributos multivariados. Es decir que contengan listas o algún tipo de estructura similar.

- Segunda Forma Normal (2NF)

La relación como ya se encuentra en 1NF, prosigue a ser evaluada para mirar si no se tienen dependencias parciales desde el atributo primo (nit). Analizando la clase, un proveedor al tener un nit que lo identifique y distingue de los demás proveedores, a pesar de tener mismo nombre, muestra la dependencia de la otra información que se necesita saber. Por lo tanto, se puede afirmar que la relación está en 2NF.

- Tercera Forma Normal (3NF)

Como la relación cumple con las dos reglas anteriores, se prosigue a analizar si está en la regla 3NF. En esta se evalúa que no exista alguna dependencia transitiva. Con la llave primaria, un atributo NO determina otro primo. Analizando todos los atributos junto con la llave primaria, no se ve otra dependencia o relación que puedan existir entre los demás atributos ya que estos son solo características de la categoría, mas no son los que guardan la información de ella. Por ejemplo, no se puede determinar el nombre del proveedor sin su nit.

- Forma Normal Boyce-Codd:

La relación está en 3NF, y adicionalmente las llaves son simples y no dependen de otra llave, se puede decir que se encuentra en FNBC.

- e. **Relación Sucursal:** La definición de relación y de sus dependencias funcionales es la siguiente:

Sucursal {id, nombre, tamaño, teléfono, dirección}

Llave candidata: Luego de analizar los atributos de la clase, se llega a la siguiente llave:

-{id}

- Primera Forma Normal (1NF)

La relación se encuentra en primera forma normal ya que esta no cuenta con atributos multivariados. Es decir que contengan listas o algún tipo de estructura similar.

- Segunda Forma Normal (2NF)

La relación como ya se encuentra en 1NF, prosigue a ser evaluada para mirar si no se tienen dependencias parciales desde el atributo primo (id). Analizando la clase, una sucursal al tener un id que lo identifique y distingue de las demás sucursales, a pesar de tener mismo nombre, muestra la dependencia de la otra información que se necesita saber. Por lo tanto, se puede afirmar que la relación está en 2NF.

- Tercera Forma Normal (3NF)

Como la relación cumple con las dos reglas anteriores, se prosigue a analizar si está en la regla 3NF. En esta se evalúa que no exista alguna dependencia transitiva. Con la llave primaria, un atributo NO determina otro primo. Analizando todos los atributos junto con la llave primaria, no se ve otra dependencia o relación que

puedan existir entre los demás atributos ya que estos son solo características de la categoría, mas no son los que guardan la información de ella. Por ejemplo, no se puede determinar el nombre de la sucursal sin su id. Pueden que haya sucursales con mismo nombre, pero id diferente.

- Forma Normal Boyce-Codd

La relación está en 3NF, y adicionalmente las llaves son simples y no dependen de otra llave, se puede decir que se encuentra en FNBC.

- f. **Relación Ciudad:** La definición de relación y de sus dependencias funcionales es la siguiente:

Ciudad {código, nombre}

Llave candidata: Luego de analizar los atributos de la clase, se llega a la siguiente llave:

-{código}

- Primera Forma Normal (1NF)

La relación se encuentra en primera forma normal ya que esta no cuenta con atributos multivariados. Es decir que contengan listas o algún tipo de estructura similar.

- Segunda Forma Normal (2NF)

La relación como ya se encuentra en 1NF, prosigue a ser evaluada para mirar si no se tienen dependencias parciales desde el atributo primo (código). Analizando la clase, una ciudad al tener un código que lo identifique y distinga de las demás ciudades, muestra la dependencia de la otra información que se necesita saber. Por lo tanto, se puede afirmar que la relación está en 2NF.

- Tercera Forma Normal (3NF)

Como la relación cumple con las dos reglas anteriores, se prosigue a analizar si está en la regla 3NF. En esta se evalúa que no exista alguna dependencia transitiva. Con la llave primaria, un atributo NO determina otro primo. Analizando todos los atributos junto con la llave primaria, no se ve otra dependencia o relación que puedan existir entre los demás atributos ya que estos son solo características de la categoría, mas no son los que guardan la información de ella. Por ejemplo, el código puede diferenciar una ciudad de otra.

- Forma Normal Boyce-Codd

La relación está en 3NF, y adicionalmente las llaves son simples y no dependen de otra llave, se puede decir que se encuentra en FNBC.

- g. **Relación OrdenCompra:** La definición de relación y de sus dependencias funcionales es la siguiente:

OrdenCompra {id, fechaEsperada, estado}

Llave candidata: Luego de analizar los atributos de la clase, se llega a la siguiente llave:

-{id}

- Primera Forma Normal (1NF)

La relación se encuentra en primera forma normal ya que esta no cuenta con atributos multivariados. Es decir que contengan listas o algún tipo de estructura similar.

- Segunda Forma Normal (2NF)

La relación como ya se encuentra en 1NF, prosigue a ser evaluada para mirar si no se tienen dependencias parciales desde el atributo primo (id). Analizando la clase, una orden de compra al tener un id que lo identifique y distinga de los demás órdenes de compra, muestra la dependencia de la otra información que se necesita saber. Por ejemplo, la fecha esperada está determinada por cual fue la orden de compra ya que en esa fecha se realiza varias órdenes de compra, Por lo tanto, se puede afirmar que la relación está en 2NF.

- Tercera Forma Normal (3NF)

Como la relación cumple con las dos reglas anteriores, se prosigue a analizar si está en la regla 3NF. En esta se evalúa que no exista alguna dependencia transitiva. Con la llave primaria, un atributo NO determina otro primo. Analizando todos los atributos junto con la llave primaria, no se ve otra dependencia o relación que puedan existir entre los demás atributos ya que estos son solo características de la categoría, mas no son los que guardan la información de ella. Por ejemplo, no se puede determinar el estado de una orden de compra sin saber cuál es, que este lo determina el id.

- Forma Normal Boyce-Codd:

La relación está en 3NF, y adicionalmente las llaves son simples y no dependen de otra llave, se puede decir que se encuentra en FNBC.

- h. **Relación ProductoBodega:** La definición de relación y de sus dependencias funcionales es la siguiente:

ProductoBodega {id, costoPromedio, cantidadProducto, nivelReorden, capacidadProducto}

Llave candidata: Luego de analizar los atributos de la clase, se llega a la siguiente llave:

-{id}

- Primera Forma Normal (1NF)

La relación se encuentra en primera forma normal ya que esta no cuenta con atributos multivariados. Es decir que contengan listas o algún tipo de estructura similar.

- Segunda Forma Normal (2NF)

La relación como ya se encuentra en 1NF, prosigue a ser evaluada para mirar si no se tienen dependencias parciales desde el atributo primo (id). Analizando la clase, un producto de bodega al tener un id que lo identifique y distinga de los demás productos de bodegas, muestra la dependencia de la otra información que se necesita saber. Por ejemplo, para saber el costo, cantidad y capacidad de los productos que tiene una bodega x, se necesita saber su id para identificarlos uno por uno. Por lo tanto, se puede afirmar que la relación está en 2NF.

- Tercera Forma Normal (3NF)

Como la relación cumple con las dos reglas anteriores, se prosigue a analizar si está en la regla 3NF. En esta se evalúa que no exista alguna dependencia transitiva. Con la llave primaria, un atributo NO determina otro primo. Analizando todos los atributos junto con la llave primaria, no se ve otra dependencia o relación que puedan existir entre los demás atributos ya que estos son solo características de la categoría, mas no son los que guardan la información de ella. Por ejemplo, no se puede determinar el costo promedio del producto con una llave diferente al id.

- Forma Normal Boyce-Codd

La relación está en 3NF, y adicionalmente las llaves son simples y no dependen de otra llave, se puede decir que se encuentra en FNBC

- i. **ProductoCompra:** La definición de relación y de sus dependencias funcionales es la siguiente:

ProductoCompra {id, precioAcordado, cantidad}

Llave candidata: Luego de analizar los atributos de la clase, se llega a la siguiente llave:

-{id}

- Primera Forma Normal (1NF)

La relación se encuentra en primera forma normal ya que esta no cuenta con atributos multivariados. Es decir que contengan listas o algún tipo de estructura similar.

- Segunda Forma Normal (2NF)

La relación como ya se encuentra en 1NF, prosigue a ser evaluada para mirar si no se tienen dependencias parciales desde el atributo primo (id). Analizando la clase, un producto de compra al tener un id que lo identifique y distinga de los demás productos de compra, muestra la dependencia de la otra información que se necesita saber. Por ejemplo, para saber la cantidad y el precio acordado de los productos que tiene una compra x, se necesita saber su id para identificarlos uno por uno. Por lo tanto, se puede afirmar que la relación está en 2NF.

- Tercera Forma Normal (3NF)

Como la relación cumple con las dos reglas anteriores, se prosigue a analizar si está en la regla 3NF. En esta se evalúa que no exista alguna dependencia transitiva. Con la llave primaria, un atributo NO determina otro primo. Analizando todos los atributos junto con la llave primaria, no se ve otra dependencia o relación que puedan existir entre los demás atributos ya que estos son solo características de la categoría, mas no son los que guardan la información de ella. Por ejemplo, no se puede determinar el precio acordado del producto con una llave diferente al id.

- Forma Normal Boyce-Codd

La relación está en 3NF, y adicionalmente las llaves son simples y no dependen de otra llave, se puede decir que se encuentra en FNBC.

- i. **Relación RecepcionProducto :** La definición de relación y de sus dependencias funcionales es la siguiente:

RecepcionProducto{id, fechaFinal}

Llave candidata: Luego de analizar los atributos de la clase, se llega a la siguiente llave:

-{id}

- Primera Forma Normal (1NF)

La relación se encuentra en primera forma normal ya que esta no cuenta con atributos multivariados. Es decir que contengan listas o algún tipo de estructura similar.

- Segunda Forma Normal (2NF)

La relación como ya se encuentra en 1NF, prosigue a ser evaluada para mirar si no se tienen dependencias parciales desde el atributo primo (id). Analizando la clase, una recepción de producto al tener un id que lo identifique y distinga de las demás recepciones de producto, muestra la dependencia de la otra información que se necesita saber. Por ejemplo, para saber la fecha esperada de una recepción de compra x, se necesita saber su id para identificar la recepción y que sea la correcta. Por lo tanto, se puede afirmar que la relación está en 2NF.

- Tercera Forma Normal (3NF)

Como la relación cumple con las dos reglas anteriores, se prosigue a analizar si está en la regla 3NF. En esta se evalúa que no exista alguna dependencia transitiva. Con

la llave primaria, un atributo NO determina otro primo. Analizando todos los atributos junto con la llave primaria, no se ve otra dependencia o relación que puedan existir entre los demás atributos ya que estos son solo características de la categoría, mas no son los que guardan la información de ella. Por ejemplo, no se puede determinar la fecha final de una recepción con una llave diferente al id.

- Forma Normal Boyce-Codd

La relación está en 3NF, y adicionalmente las llaves son simples y no dependen de otra llave, se puede decir que se encuentra en FNBC.

4. Casos de prueba

En esta sección detallaremos acerca de las restricciones que tiene la base de datos, y detallaremos los proceso para los requerimientos funcionales. Aclaremos que solo hablaremos de las restricciones que tiene la base de datos, pues los ejemplos de tuplas que serían aceptadas vienen en el archivo Excel, donde están las tablas con las relaciones, entidades, y ejemplos de tuplas.

- **Requerimientos Funcionales**

- **RF1 – Crear una ciudad**

En este caso no debe de haber ciudades repetidas, adicionalmente se agrega un atributo **Check**, donde se revisa que la ciudad sea de Colombia. De esa manera aseguramos que no se va a entrar cualquier String que no haga sentido con que sea una ciudad, luego el sistema le asigna un código a cada ciudad.

- **RF2 – Crear una sucursal**

Para poder crear una sucursal, es necesario dar un nombre, un tamaño (CK), teléfono (ND, y dirección (ND). Además, se debe especificar un código de la ciudad a la que la sucursal se asigna.

No se aceptarían tuplas donde hallan teléfonos o direcciones repetidas para varias sucursales en la base de datos. Adicionalmente el tamaño se debe de revisar, pues queremos evitar que hallan entradas con datos no deseados como: Números negativos, 0, o números irracionales

Si los datos son correctos se acepta la tupla y el sistema asigna un id.

- **RF3 – Crear y Borrar Bodega**

Para crear una Bodega necesitamos, su nombre, tamaño (CK) y capacidad (CK) y el id de la sucursal a la cual esta asignada.

Igual que en otros casos, el tamaño y la capacidad se deben de revisar, para evitar números no deseados como los nombrados anteriormente.

Finalmente, si todo esta correcto el sistema le asigna un ID.

Para borrarla será necesario el id tanto como de la bodega. Con esta eliminación se espera que se elimine tanto la tupla de la bodega relacionada al id requerido, como el detalle de los productos que se encuentran en la bodega (ProductoBodega). Esto ya que no puede existir instancias de ProductoBodega que estén asignados a una Bodega que ya no existirá más.

- **RF4 – Crear y actualizar proveedores**

Para crear un proveedor necesitas el NIT, nombre (ND), dirección, el nombre del representante y el teléfono del representante (ND), en caso de que la tupla pase las restricciones se guardara y se asignara como id el NIT del proveedor

Para actualizar el proveedor necesitas su NIT (PK) y hay que especificar que atributo necesitas modificar para finalmente realizar el remplazo. Si no existe en NIT o no existe el atributo que desea modificar, no podrá ser modificado.

- **RF5 – Crear y leer una categoría de producto**

Para crear una categoría de producto necesitas, su nombre (ND), descripción, y características, si el nombre no es duplicado el sistema le asigna un ID. Para poder leer una categoría necesitas un ID para poder identificar la categoría.

- **RF6 – Crear, leer y actualizar un producto**

Para poder crear un producto necesitas el nombre y el costo de bodega, la cantidad en la prestación, el volumen y el peso, para estos el sistema hará unas revisiones, pues queremos evitar números como negativos y 0, también se debe especificar la presentación, la unidad de medida. Finalmente, si la tupla paso las revisiones, el sistema le asignara un ID.

Para leer el producto solo necesitaras del respectivo ID del mismo para poder localizarlo.

Para poder actualizar un producto necesitas el ID y que atributo desea modificar, y el valor.

- **RF7, RF8 Y RF9 - Crear, actualizar (anulada) y mostrar las órdenes de compra**

Para crear la orden de compra será necesaria la fecha esperada, el estado de la orden, adicionalmente el ID de las siguientes relaciones, sucursal, proveedor.

Adicionalmente los IDs de los productos que se desean comprar, la cantidad y el respectivo precio.

Si el producto no existe no se permitirá comprar, si la fecha esperada no cumple con los requisitos (Igual o mayor a la fecha actual y un rango máximo) no se permitirá la creación.

Si la tupla pasa los respectivos checks, el sistema le asignara un ID y adicionara otro atributo que es la fecha actual.

Para poder actualizar la orden de compra es necesario el ID de la orden de compra y el nuevo estado que se desea actualizar. Si no existe el ID no se actualizará. Cabe aclarar que mínimo debe haber un producto.