

Sistemas Transaccionales

Ximena López - 202312848

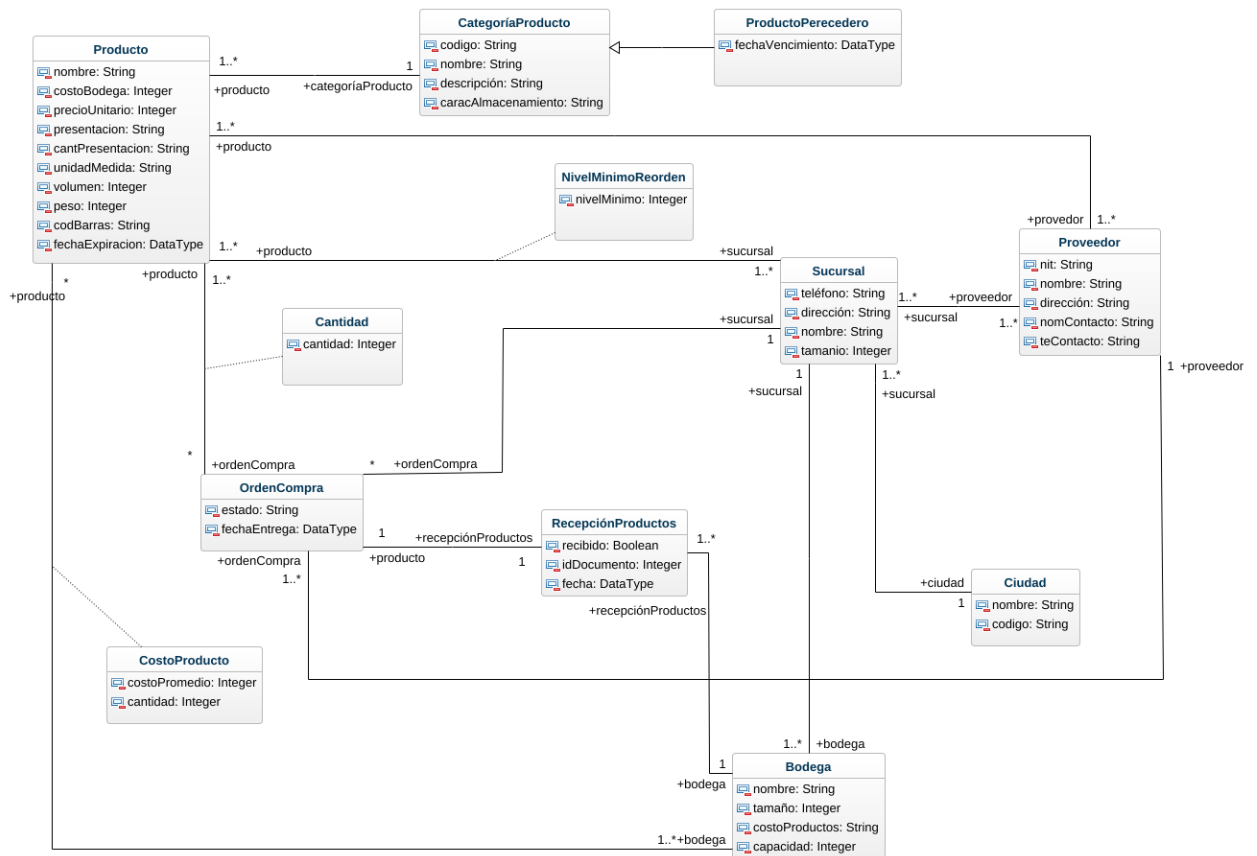
Santiago Pineda - 202023262

Sofia Losada - 20221008

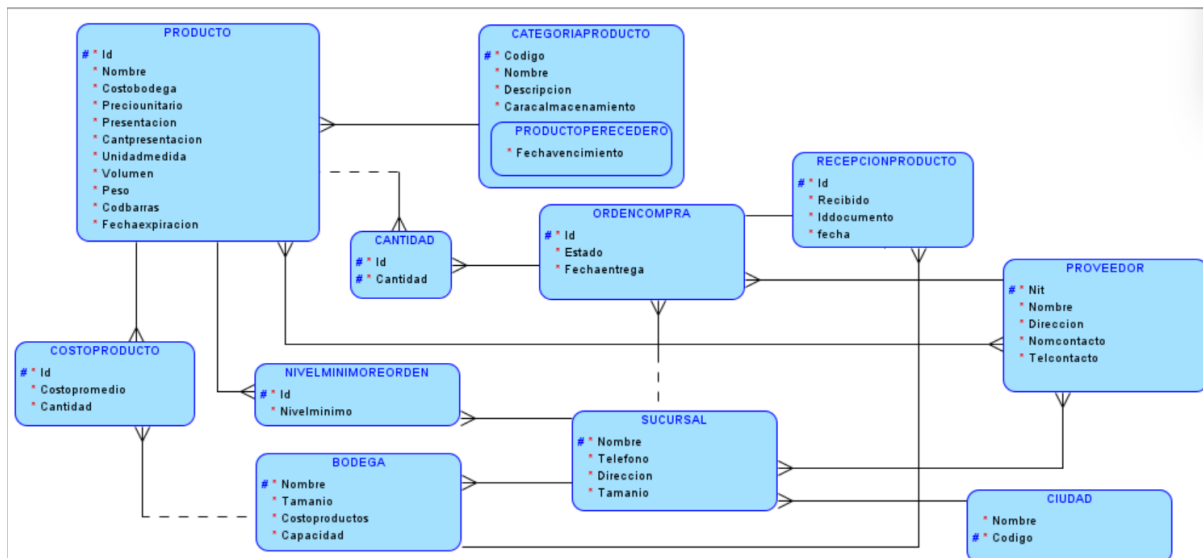
Proyecto 1 - Entrega 1/2

SuperAndes Diseño

El diseño de UML propuesto:



Modelo E/R propuesto:



El diseño de datos relacional correspondiente al UML presentado es el siguiente:

Sucursal					
id	nombre	direccion	telefono	tamano	id_ciudad
SA, PK	NN	NN	NN	NN	FKciudad.id
Suc-01	sucursal norte	Carrera 4	3222345931	500	1001

En la tabla de Sucursal, se incluyen los atributos correspondientes propuestos en el diagrama UML. Además, se establece la relación con la entidad Ciudad, dado que cada sucursal debe estar asociada a una ciudad.

Para el dominio de los atributos se seleccionó id como PK, el cual, es proporcionado por el sistema, debido a que no hay un atributo único y no nulo ya dado. Asimismo, en tanto a nombre, direccion, telefono y tamano, son atributos que cuentan con la restricción de no nulos porque son necesarios para representar esta clase.

Proveedor				
nit	nombre	direccion	nomContacto	telContacto
PK, UA	NN	NN	NN	NN
1302	Lay	calle 4#2-34	Juan	3112018888

La tabla de proveedor está compuesta por sus atributos presentados en el UML, con ninguna relación obligatoria única a representar en ella.

Para el dominio de los atributos se seleccionó nit como PK, el cual, es proporcionado por el usuario, debido a que es un atributo único y no nulo ya dado. Asimismo, en tanto

Para el dominio de los atributos se seleccionó id como PK, el cual, es proporcionado por

Bodega

Para el dominio de los atributos se seleccionó id como PK, el cual, es proporcionado por

Ciudad	
codigo	nombre
PK,UA	NN
1001	Bogotá

La tabla de ciudad está compuesta por sus atributos presentados en el UML, con ninguna relación obligatoria única a representar en ella.

Para la PK se escogió código porque es un valor único de cada ciudad. Además, código cuenta con un dominio más y es UA, puesto que, es asignada por el usuario. Asimismo, el atributo nombre no puede ser null, es decir NN.

OrdenCompra					
id	estado	fecha_entrega	cantidad	id_sucursal	id_proveedor
PK,SA	NN	NN	NN	FKsucursal.id,NN	FKproveedor.id,NN
Orden-01	Entregada	12/02/2020	4	Suc-01	1302

En la tabla de OrdenCompra, se incluyen los atributos correspondientes propuestos en el diagrama UML. Además, se establece la relación con la entidad Sucursal, dado que cada orden de compra debe estar asociada a una Sucursal. Asimismo, se representa la relación con proveedor, puesto que, cada orden de compra debe de ir para un proveedor en específico. Finalmente, para lo anterior nos basamos en el algoritmo modificado de Chen la PK donde el extremo univalor se incluye como atributo en la tabla del extremo multivalor.

Para el dominio de los atributos se seleccionó id como PK, el cual, es proporcionado por el sistema, debido a que no hay un atributo único y no nulo ya dado. Asimismo, en tanto a nombre, estado, fecha_entrega, cantidad, id_proveedor y id_sucursal son atributos que cuentan con la restricción de no nulos porque son necesarios para representar esta clase.

CategoriaProducto			
codigo	nombre	descripción	caracAlmacenamiento
PK,UA	NN	NN	NN
124	No-perecedero	No vencimiento	No frio

La tabla de categoriaProducto está compuesta por sus atributos presentados en el UML, con ninguna relación obligatoria única a representar en ella.

Para el dominio de los atributos se seleccionó código como PK, el cual, es proporcionado por el usuario, debido a que hay un atributo único y no nulo ya dado. Asimismo, en tanto a nombre, descripción y caracAlmacenamiento son atributos que cuentan con la restricción de no nulos porque son necesarios para representar esta clase.

ProductoPerecedero				
codigo	nombre	descripción	caracAlmacenamiento	fechaVencimiento
PK,UA	NN	NN	NN	NC, NN
123	Perecedero	Vencimiento	Frio	17/02/2026

La tabla de ProductoPerecedero está compuesta por sus atributos presentados en el UML, con ninguna relación obligatoria única a representar en ella. Sin embargo, esta la representación de herencia, donde esta tabla es la hija de la clase madre CategoriaProducto, por lo cual, los atributos de la entidad mayor se encuentran repetidos en esta.

Para el dominio de los atributos se seleccionó código como PK, el cual, es proporcionado por el usuario, debido a que hay un atributo único y no nulo ya dado. Asimismo, en tanto a nombre, descripción, caracAlmacenamiento y fechaVencimiento son atributos que cuentan con la restricción de no nulos porque son necesarios para representar esta clase. Además, fechaVencimiento es un valor estático en el tiempo.

NivelMinimoReorden		
id_producto	id_sucursal	nivelMinimo
PK, FKproducto.id	PK,F Ksucursal.id	NN,DD
Pro-01	1001	5

La tabla NivelMinimoReorden es una relación intermedia entre producto y sucursal, en la cual se representan sus atributos y la PK de producto y sucursal. Para esta relación se seleccionó una tabla adicional debido a que es de muchos a muchos. Además, ésta en una relación intermedia debido a que tiene dos atributos adicionales.

El dominio del atributo nivelMinimo es que no puede ser nulo y además es un dato derivado porque es calculado a partir de información de la relación.

ProveedorSucursal	
nit_proveedor	id_sucursal
PK, FKproveedor.nit	PK,FKsucursal.id
1302	Suc-01

La tabla ProveedorSucursal es una relación de muchos a muchos entre proveedor y sucursal, por lo cual basados en el algoritmo de Chen se decidió una tabla extra para representarlo.

CostoProductos			
id_producto	id_bodega	costoPromedio	capacidad
PK, FKproducto.id	PK, FKbodega.id	NN,DD	NN,DD
Pro-01	Bod-01	9000	1000

La tabla CostoProducto es una relación intermedia entre producto y bodega, en la cual se representan sus atributos y las PK de producto y bodega. Para esta relación se seleccionó una tabla adicional debido a que es de muchos a muchos. Además, ésta en una relación intermedia debido a que tiene un atributo adicional.

RecepcionProductos				
idDocumento	id_OrdenCompra	id_bodega	recibido	fecha
PK,UA	FKordenCompra.id	Fkbodega.id	NN	NN,NC
recepcion-01	Orden-01	Bod-01	True	17/02/2007

La tabla RecepcionProductos es una entidad con sus respectivos atributos, recibido y fecha, los cuales no pueden ser nulos y la fecha es un valor estático en el tiempo. En la tabla está el id de orden de compra porque es una relación de 1 a 1, en la que solo se registra en esta tabla. Además, se encuentra el id de bodega puesto que es una relación de 1 a muchos.

Cantidad		
id_OrdenCompra	id_Producto	cantidad
PK, FKOrdenCompra.id	PK,FKproduct0.id	NN
Orden-01	Pro-01	1

La tabla ProductoOrden es una relación intermedia de muchos a muchos entre producto y orden. Además, tiene un atributo de cantidad de productos por orden de compra, por lo cual basados en el algoritmo de Chen se decidió una tabla extra para representarlo.

Pruebas Unitarias Proyecto 1

Requerimiento funcional 1: Crear una ciudad

- Unicidad de tupas:

La entidad Ciudad se compone de dos atributos: Nombre y Código. Se definió como PK el código respectivo, por lo que, la creación de una ciudad en la base de datos no permite que existan códigos repetidos y con otros valores en los demás atributos. No es posible que existan dos códigos de ciudades iguales con nombres diferentes. De ser así se genera un error en la base de datos, como se muestra a continuación.

Ciudad	
codigo	nombre
PK,UA	NN
1001	Bogotá
1001	Manizales

- Integridad con FK: NA
- Integridad de restricciones de chequeo:

La entidad Ciudad tiene como PK el código de ciudad, lo que significa que es único y exclusivo para reconocer una ciudad específica. Además, Nombre no puede ser nulo o vacío, ya que no existe ninguna ciudad sin nombre y no se puede identificar. Por lo tanto, una tupla que no cumpla con las condiciones mencionadas genera error, como se muestra a continuación:

Ciudad	
codigo	nombre
PK,UA	NN
1001	Bogotá
1001	

Requerimiento funcional 2: Crear una sucursal

- Unicidad de tupas:

Basándonos en la tabla de Sucursal, con una sucursal ya creada que tiene como id Suc-01, con el nombre de sucursal norte, en la dirección carrera 4, teléfono 3222345931, un tamaño de 500 metros cuadrados y está en la ciudad con código 1001. Si después intentamos agregar otra tupla, con la misma PK, no nos lo permitirá, porque no puede duplicarse. Lo anterior, se puede ejemplificar con la siguiente tabla:

Sucursal					
id	nombre	direccion	telefono	tamano	id_ciudad
SA, PK	NN	NN	NN	NN	FKciudad.id
Suc-01	sucursal norte	Carrera 4	3222345931	500	1001
Suc-01	sucursal Sur	Calle 55	3112018882	100	1001

- Integridad con FK: Para realizar la prueba de integridad con llaves foráneas, se considera que las FK se usan para establecer relaciones entre entidades de manera correcta. La integridad asegura que cualquier valor de FK esté presente como PK en la tabla referenciada.

La entidad Sucursal tiene como FK id_ciudad. Esto se debe a que una sucursal debe estar ubicada en una ciudad. De esta manera, el valor de id_ciudad en la entidad Sucursal debe corresponder a algún valor de la entidad Ciudad. De no ser así, no habría integridad en los datos y la operación no se podría realizar, como se muestra el error a continuación.

Sucursal					
id	nombre	direccion	telefono	tamano	id_ciudad
SA, PK	NN	NN	NN	NN	FKciudad.id
Suc-01	sucursal norte	Carrera 4	3222345931	500	1001
Suc-01	sucursal Sur	Calle 55	3112018882	100	1206

En la tabla de Sucursal existe se crea una Sucursal en la ciudad cuyo id es: 1206, señalado con color verde.

Ciudad	
codigo	nombre
PK,UA	NN
1001	Bogotá
1004	Manizales

Sin embargo, en la entidad Ciudad el id: 1206 no existe, por lo que no hay integridad en los datos y se genera un error.

- Integridad de restricciones de chequeo:

Los atributos: Nombre, dirección, teléfono y tamaño de la entidad Sucursal, no pueden ser nulos ya que son aquellos que conforman la descripción y especificaciones de la entidad. De ser nulos, no sería posible tener una sucursal ni identificarla. El id es asignado por el sistema y representa la llave principal, pues es el atributo único que permite identificar fácilmente a una sucursal y sus características asociadas. Si el id se repite y el valor asociado a nombre fuera nulo, habría un error en el sistema, como se muestra a continuación.

Sucursal					
id	nombre	direccion	telefono	tamano	id_ciudad
SA, PK	NN	NN	NN	NN	FKciudad.id
Suc-01	sucursal norte	Carrera 4	3222345931	500	1001
Suc-01					

Requerimiento funcional 3: Crear y borrar una bodega

- Unicidad de tupas:

En la entidad bodega se tiene como PK asignada por el sistema el id, para que se pueda asociar las diferentes características específicas a una bodega. De igual manera, tiene como atributos no nulos nombre y tamaño y como FK asociada sucursal, pues cada bodega está asociada a una sucursal específica. En caso de dos tuplas estén asociadas al mismo id se genera un error, pues son bodegas diferentes o hay repetición innecesaria de datos. El error se muestra a continuación.

Bodega			
id	nombre	tamano	id_sucursal
PK, SA	NN	NN	NN, FKsucursal.id
Bod-01	Bodeguita	100	Suc-01
Bod-01	BodegaUno	210	Suc-06

- Integridad con FK:

En esta entidad se tiene como FK a id_sucursal, ya que cada bodega está asociada a una sucursal. Por lo tanto, el id de sucursal como FK debe corresponder a algún id en la entidad Sucursal. De no ser así, se genera un error en la base de datos como se muestra a continuación:

Esta es la entidad sucursal que solo contiene el id:Suc-01

Sucursal					
id	nombre	direccion	telefono	tamano	id_ciudad
SA, PK	NN	NN	NN	NN	FKciudad.id
Suc-01	sucursal norte	Carrera 4	3222345931	500	1001

Esta es la entidad bodega que contiene FK id:Suc-06

Bodega			
id	nombre	tamano	id_sucursal
PK, SA	NN	NN	NN, FKsucursal.id
Bod-01	Bodeguita	100	Suc-01
Bod-03	BodegaUno	210	Suc-06

Por lo tanto, se genera error.

- Integridad de restricciones de chequeo:

La entidad bodega tiene asignado por el sistema la PK de id, de esta manera todos los valores de los atributos están asociados a un respectivo id. El nombre, tamaño y FK id_sucursal, no pueden ser nulos ya que se genera un error en la creación de la entidad, pues no es posible que, en el momento de crear una bodega, no tenga nombre, capacidad asociada o una sucursal. De igual manera, no se puede borrar una bodega sin saber su nombre, id, capacidad y/o sucursal a la cual está asociada, pues no se podría encontrar en un principio. El error se ejemplifica muestra a continuación.

Bodega			
id	nombre	tamano	id_sucursal
PK, SA	NN	NN	NN, FKsucursal.id
Bod-01	Bodeguita	100	Suc-01
Bod-03	BodegaUno	210	

Requerimiento funcional 4: Crear y actualizar proveedores

- Unicidad de tupas:

En la entidad proveedor la PK determinada por el usuario es el nit, ya que representa el valor único asociado a un proveedor específico. Los atributos asociados son Nombre, dirección, nombre de contacto y teléfono de contacto. De esta manera, están asociados a un único nit. En caso de que dos tuplas estén asociadas a dos ids o códigos nit iguales, se genera un error en la base de datos, pues no es posible que dos proveedores con diferentes atributos tengan un nit igual, por lo que se creó de manera incorrecta. El error se muestra a continuación.

Proveedor				
nit	nombre	direccion	nomContacto	telContacto
PK, UA	NN	NN	NN	NN
1302	Lay	calle 4#2-34	Juan	3112018888
1302	Lay	calle 8#2-68	Pedro	3182018988

- Integridad con FK: Esta entidad no contiene FK, por lo que no es relevante realizar el caso de prueba.
- Integridad de restricciones de chequeo:

La entidad Proveedores tiene un id definida por el usuario, pues es una identificación externa al negocio. Los atributos no pueden ser nulos, pues no es posible que un proveedor no tenga nombre o dirección o medio de contacto, pues no sería posible identificarlo, que es el primer paso para actualizar el proveedor específico y se generaría un error al crearlo también.

Proveedor				
nit	nombre	direccion	nomContacto	telContacto
PK, UA	NN	NN	NN	NN
1302	Lay	calle 4#2-34	Juan	3112018888
1305	Lay	calle 4#2-34		3112018888

Requerimiento funcional 5: Crear, leer y actualizar un producto

- Unicidad de tupas:

La entidad producto tiene como PK el id del producto específico y como atributos asociados nombre, costo en bodega, precio unitario, presentación, cantidad de la presentación, unidad de medida, volumen, peso, código de barras, fecha de expiración y como FK id_categoria a la que pertenece. Así, se puede crear un producto concreto de manera correcta. Si dos o más productos tienen valores asociados a los atributos dados y el id igual, implica un error en la base de datos, pues no se pueden crear dos productos diferentes con un mismo identificador, por lo tanto se creó de manera incorrecta. El error se muestra a continuación.

Producto											
id	nombre	costoBodega	precioUnitario	presentacion	cantPresentacion	unidadMedida	volumen	peso	codBarras	fechaExpiracion	id_categoria
PK,SA	NN	NN	NN	NN	NN	NN	NN	NN	NN,NC	NN,NC	NN, FKcategoriaProducto.id
Pro-01	Papas fritas les fr	4200	9000	de 5 paquetes de 200 gr	1000	gr	150	15	60606060	12/12/2027	123
Pro-01	Teatoca	4200	9000	de 5 paquetes de 200 gr	1000	gr	150	15	60606060	12/12/2027	123

- Integridad con FK:

La entidad Producto tiene como FK id_categoria pues cada producto tiene unas características específicas que lo encasillan en una categoría. De esta manera, la integridad debe asegurar que el valor existente como FK en la entidad producto corresponda a algún valor id de la entidad categoría. De no ser así se genera un error, pues no se puede actualizar de manera correcta.

En la entidad cantidadProducto se encuentran los siguientes id.

CategoriaProducto			
codigo	nombre	descripcion	caracAlmacenamiento
PK,UA	NN,ND	NN	NN
124	No-perecedero	No vencimiento	No frio

Como FK el id_categoria es 127

Producto											
id	nombre	costoBodega	precioUnitario	presentacion	cantPresentacion	unidadMedida	volumen	peso	codBarras	fechaExpiracion	id_categoria
PK,SA	NN	NN	NN	NN	NN	NN	NN	NN	NN,NC	NN,NC	NN, FKcategoriaProducto.id
Pro-01	Papas fritas les fr	4200	9000	de 5 paquetes de 200 gr	1000	gr	150	15	60606060	12/12/2027	123
Pro-06	Papas fritas les fr	4200	9000	de 5 paquetes de 200 gr	1000	gr	150	15	60606060	12/12/2027	127

- Integridad de restricciones de chequeo:

En cuanto a las restricciones de chequeo los atributos asociados el nombre, codigoBodega, precioUnitario, presentación, cantPresentacion, unidadMedida, volumen, peso y la FK asociada no pueden ser nulos pues de ser así el producto no se podría crear de manera correcta. En cuanto a codBarras y fechaExpiración son atributos que no son cambiables en el tiempo, pues el código de barras es el identificador específico de un solo producto, el cual es único e independiente del resto de atributos mencionados previamente. En cuanto a la fecha de Expiración es un atributo que tampoco depende de los demás y no debería cambiar desde que es asignado. En caso de FK fuese nula se podría establecer la relación con la entidad categoría y el producto no se leería de manera correcta. El error se ve a continuación.

Producto											
id	nombre	costoBodega	precioUnitario	presentacion	cantPresentacion	unidadMedida	volumen	peso	codBarras	fechaExpiracion	id_categoria
PK,SA	NN	NN	NN	NN	NN	NN	NN	NN	NN,NC	NN,NC	NN, FKcategoriaProducto.id
Pro-01	Papas fritas les fr	4200	9000	de 5 paquetes de 200 gr	1000	gr	150	15	60606060	12/12/2027	

Requerimiento funcional 6: Crear, leer y actualizar una categoría de producto

- Unicidad de tupas:

La entidad categoriaProducto tiene como PK el respectivo código, nombre, descripción y características de almacenamiento. Para crear una nueva categoría se debe asignar un código específico y único para así poder acceder y leer los datos de los atributos asociados. De no ser así se genera un error, ya que una categoría cuyo id sea igual a la de otra, pero sus datos asociados a los atributos sean diferentes significa que dos categorías diferentes tienen los mismos productos y un producto no puede pertenecer a dos categorías diferentes. Esto genera el siguiente error:

CategoriaProducto			
codigo	nombre	descripción	caracAlmacenamiento
PK,UA	NN,ND	NN	NN
124	No-perecedero	No vencimiento	No frio
124	No-perecedero	No vencimiento	frio

Una categoría de producto diferente tiene el mismo id que otra, representa un error y no se crea correctamente la categoría.

- Integridad con FK:

La entidad no cuenta con FK por lo que no es relevante hacer el caso de prueba bajo estas condiciones.

- Integridad de restricciones de chequeo:

Como PK asignada por el usuario se utiliza el código, pues así se pueden leer correctamente los valores asociados a los atributos. Los atributos nombre, descripción y características de almacenamiento no pueden ser nulos, ya que no permite crear de manera correcta una categoría pues faltaría información, lo que a su vez no permite que se actualicen los datos de manera correcta. El error se ve a continuación:

CategoriaProducto			
codigo	nombre	descripción	caracAlmacenamiento
PK,UA	NN,ND	NN	NN
124	No-perecedero		No frio

Requerimiento funcional 7: Crear una orden de compra para una sucursal

- Unicidad de tupas:

La entidad Orden de compra tiene como PK el id de la respectiva orden, esto permite crear una orden de compra de manera que el id es único y permite acceder a los valores respectivos de la orden. Por esto, se genera un error cuando dos órdenes de compra diferentes tienen el mismo id. Ya que, para id hay una fecha de entrega, un estado y está asociado a una sucursal y un proveedor específico, incluso si los valores son los mismos. El error se muestra a continuación:

OrdenCompra				
id	estado	fecha_entrega	id_sucursal	id_proveedor
PK,SA	NN	NN	FKsucursal.id,NN	FKproveedor.id,NN
Orden-01	Entregada	12/2/2020	Suc-01	1302
Orden-01	Entregada	12/2/2020	Suc-01	1302

- Integridad con FK:

La entidad Orden de compra se relaciona con sucursal y proveedor, ya que cada sucursal emite una orden de compra y quien la resuelve es el o los proveedores, pues son quienes envían los productos solicitados. Ocurre un error cuando las FK en Orden de compra asociadas respectivamente a sucursal o proveedor, corresponden a los valores de las respectivas entidades. El error se muestra a continuación.

Entidad Proveedor:

Proveedor				
nit	nombre	direccion	nomContacto	telContacto
PK, UA	NN	NN	NN	NN
1302	Lay	calle 4#2-34	Juan	3112018888

Entidad Sucursal:

Sucursal					
id	nombre	direccion	telefono	tamano	id_ciudad
SA, PK	NN	NN	NN	NN	FKciudad.id
Suc-01	sucursal norte	Carrera 4	3222345931	500	1001

Error, pues los valores en id_sucursal y en id_proveedor no existen en las respectivas entidades.

OrdenCompra				
id	estado	fecha_entrega	id_sucursal	id_proveedor
PK,SA	NN	NN	FKsucursal.id,NN	FKproveedor.id,NN
Orden-01	Entregada	12/2/2020	Suc-01	1302
Orden-03	Entregada	12/2/2020	Suc-70	2029

- Integridad de restricciones de chequeo:

El sistema asigna el id de la orden en cuanto a las restricciones de chequeo. Para la creación de una orden ninguno de los atributos puede estar vacío ya que la información estaría incompleta y no se recibiría bien el pedido. El error se muestra a continuación:

OrdenCompra				
id	estado	fecha_entrega	id_sucursal	id_proveedor
PK,SA	NN	NN	FKsucursal.id,NN	FKproveedor.id,NN
Orden-01	Entregada	12/2/2020	Suc-01	1302
Orden-03	Entregada		Suc-70	2029

Normalización

Primera forma normal:

El modelo que se planteó para el diseño sigue la primera forma normal (1FN) debido a que cada celda de las tablas contiene valores atómicos (individuales), y no se encuentran grupos repetitivos o valores múltiples como listas o arreglos dentro de estas. Cabe resaltar que, para el caso de la relación de muchos a muchos entre Proveedor y Sucursal, ya que no se creó una relación intermedia entre las entidades, se ideó una tabla intermedia basada en el algoritmo de Chen, de lo contrario alguna tabla de las entidades no hubiera cumplido con la 1FN porque se hubieran necesitado múltiples valores en una columna.

ProveedorSucursal	
nit_proveedor	id_sucursal
PK, FKproveedor.nit	PK,FKsucursal.id
1302	Suc-01

Por lo tanto, la tabla está diseñada para que nit_proveedor y id_sucursal contengan un solo valor individual por celda para mostrar la relación entre las dos entidades, mientras se mantiene el 1FN. De igual forma, las demás tablas del modelo fueron creadas de manera que ninguna celda tenga valores no atómicos y el modelo quede dentro del 1FN.

Segunda forma normal:

Siguiendo con el proceso de normalización, en el caso de la segunda forma normal (2FN), aparte de que se cumpla la 1FN, es necesario que todos los atributos de la tabla dependan completamente de las llaves primarias y no solo de una parte. Así que, para el modelo se logró llegar al cumplimiento de estos criterios. De tal manera que, por ejemplo, para la tabla “CostoProductos” donde hay dos llaves primarias (PK) que son id_producto y id_bodega, se tienen dos atributos. El primero es costoPromedio, el cual si depende totalmente de las dos PK ya que solo con el id_producto no se puede saber a qué producto se relaciona este costo y de la misma forma, con el id_bodega, sabemos la bodega específica de la cual es el costo promedio para este producto, debido a que, para cada una, se tiene un costo promedio diferente. El segundo atributo es capacidad, con id_bodega, igualmente se tiene una cierta capacidad para cada producto en la

bodega por lo que es necesario el id_producto, y es necesario saber la bodega específica para esta cantidad.

CostoProductos			
id_producto	id_bodega	costoPromedio	capacidad
PK, FKproducto.id	PK, FKbodega.id	NN,DD	NN,DD
Pro-01	Bod-01	9000	1000

Otro caso es para la tabla de “Cantidad”, donde otra vez hay dos PK (id_OrdenCompra y id_Producto) y el atributo cantidad, que sería la cantidad de productos por orden de compra. Primero, sin la llave id_OrdenCompra no se puede saber a qué compra específica se refiere la cantidad y sin id_Producto es desconocida la orden de compra a la que está relacionada la misma.

Cantidad		
id_OrdenCompra	id_Producto	cantidad
PK, FKOrdenCompra.id	PK,FKproduct0.id	NN
Orden-01	Pro-01	1

Igualmente, las demás tablas tienen solo una PK y/o los atributos dependen totalmente de las llaves primarias, por lo tanto, el modelo si cumple con la 2FN.

Tercera forma normal:

En el caso de la tercera forma normal (3FN), es necesario que se cumpla la 2FN y que no haya dependencias transitivas entre los atributos no clave (atributo que no hace parte de la llave primaria) y la PK. Es decir que ningún atributo que no esté en la llave primaria debe depender de otro que tampoco lo esté. En general, las tablas del modelo propuesto si cumple con la 3FN y un buen ejemplo para mostrarlo es con la tabla de “CategoriaProducto”, en la cual hay una llave primaria código y tres atributos de la categoría del producto. En este caso, cada uno de estos atributos depende únicamente del Código (clave primaria), sin depender entre sí, lo que demuestra que la tabla está en 3FN. Es decir, con solo conocer el Código, se pueden determinar los demás atributos de la tabla sin necesidad de acceder a ninguna otra información adicional.

CategoriaProducto			
codigo	nombre	descripción	caracAlmacenamiento
PK,UA	NN	NN	NN
124	No-perecedero	No vencimiento	No frio

Esto igualmente se cumple para las demás tablas del modelo, ya sea porque tienen una PK que permite obtener cualquier atributo y/o no existen dependencias transitivas entre las llaves no clave.

Forma Normal de Boyce-Codd:

Para la forma normal de Boyce-Codd (BCNF), se necesita que el modelo este en 3FN y que, además todas las llaves determinantes (atributos que determinan otros atributos en una dependencia funcional) deben ser atributos clave (primos) o triviales con la llave. Lo que significa que es una versión más restrictiva que el 3FN, así que por lo general las tablas del modelo que tienen una única llave primaria (solo un atributo) y ya están en 3FN, pertenecen también a BCNF. Por lo que, quedaría verificar las tablas que tienen llaves primarias compuestas.

En primer lugar, está “CostoProductos”, que como se había establecido anteriormente los atributos que no hacen parte de la llave primaria dependen de esta, y véase que como ambos son necesarios para saber los atributos, realmente no tienen un elemento en común individualmente. O sea, de id_producto no es posible saber costo promedio ni capacidad, es necesario saber el id_bodega, y viceversa. De esta manera la tabla si entrara en BCNF.

CostoProductos			
id_producto	id_bodega	costoPromedio	capacidad
PK, FKproducto.id	PK, FKbodega.id	NN,DD	NN,DD
Pro-01	Bod-01	9000	1000

Este mismo caso se repite para las tablas ya mencionadas en los otros casos de normalización. Hay otros casos en donde las llaves primarias tienen acceso a un único atributo como en el caso de “NivelMininoReorden” y entraría en la misma circunstancia que la tabla anterior, dado que en cada caso ambas claves son necesarias para determinar los atributos no clave, no existen dependencias funcionales que violen BCNF. Así que, el modelo finalmente entraría en BCNF.

.