

## 1) Actualización del UML

Anteriormente, el mayor inconveniente del diseño era el uso inadecuado de entidades que no respetaban la cardinalidad expuesta por “SuperAlpes” y omitían las consecuencias del uso de información como cantidades y precios que dependen del producto y la compra realizada. Sin embargo, el uso de clases intermedias permite que la información solo dependa de las dos clases unidas a ella, y no se vea afectada por cambios, en el caso de la cantidad de productos en Bodega, la clase “InfoExtraBodega” permite evidenciar la cantidad de un producto sin que esta se vea modificada por la cantidad de otro producto.

Los detalles de cada actualización se muestran a continuación

- Clase “InfoExtraOrden”: Permite registrar la cantidad de producto especificado en una orden de compra y el costoUnitario del producto dentro de la orden. Esta clase no es obligatoria por lo que se representa con una línea segmentada.
- Clase Venta: Presenta un id y una fecha de creación. Permite evidenciar la información de una compra realizada por un cliente en una sucursal específica, es por esto que está conectada a “Cliente”, “Sucursal” y “Producto”.
- Clase “InfoExtraVenta”: Sus atributos son la cantidad de producto seleccionado por en la venta y el precioUnitario del mismo. Esta es una clase intermedia entre “Producto” y “Venta”.
- Clase “InfoExtraBodega”: Clase intermedia entre “Bodega” y “Producto”, sus atributos son el total de existencias de un producto en específico dentro de la bodega, el costo promedio del mismo, la capacidad de almacenamiento de la bodega y el nivel mínimo del anterior mencionado.
- Clase detalleCostosBodega: Esta clase está conectada a “InfoExtraBodega2 con una cardinalidad de muchos a uno (\* - 1) que permite evidenciar el costo unitario de la bodega y la cantidad de existencias.
- Clase “productoPerecedero”: Es una clase que hereda de producto e identifica a una clase de producto manejado por “SuperAlpes”, su único atributo independiente es la fecha de vencimiento.
- Clase “especificaConEmpacado”: Clase conectada a “Producto” con cardinalidad 1 a muchos (1 - \*) que presenta la información del volumen y el peso de cada producto.
- Clase “infoExtraProveedor”: Clase intermedia entre “Proveedor” y “Producto” que tiene la cantidad de existencias de un producto manejada por el proveedor.

## 2) Actualización del modelo relacional

La actualización del modelo relacional se encuentra en el documento Excel, donde se incluyen las restricciones de cada atributo, así como las llaves primarias y foráneas de cada relación.

Es importante destacar que, para relacionar correctamente las clases intermedias en el modelo UML, las llaves primarias de estas clases son tuplas formadas por las llaves primarias de las relaciones a las que están conectadas. A continuación, se presenta un ejemplo:

La clase "InfoExtraProveedor", como se explicó previamente en el modelo UML, es una clase intermedia entre "Producto" y "Proveedor". La llave primaria de la relación "Producto" es su código de barras, mientras que la llave primaria de "Proveedor" es su NIT. Por lo tanto, la superllave de "InfoExtraProveedor" está compuesta por (NIT, códigoBarras), es decir, el conjunto de las llaves primarias de "Producto" y "Proveedor".

#### INFOEXTRAPROVEEDOR

cantidadExistencias (INTEGER)	NIT(INTEGER)	codigoBarras(VARCHAR)
NN, SA	PK, FK(Proveedor, NIT), NN, UA	PK, FK(Producto, codigoBarras), NN, ND
1	3	98765432109

Las otras relaciones y sus atributos no se modificaron. Sin embargo, aquí se presenta un pequeño resumen de algunas de las tablas desarrolladas, es decir, aquellas que no son intermedias ya que el funcionamiento de esas se explicó anteriormente. Con el objetivo de poder entender a grandes rasgos el desarrollo de este modelo, para más detalles dirigirse al documento Excel.:

- PRODUCTO PERECEDERO: Su llave primaria es el código de barras que hereda de "Producto".
- CIUDAD: Su llave primaria es el código de la ciudad asignado por el sistema, además el nombre de esta no puede ser nulo.
- SUCURSAL: La llave primaria es idSucursal el cual no puede ser nulo y no puede ser duplicado. Presenta una llave foránea que hace referencia al código de la ciudad.
- PRODUCTO: La llave primaria es su código de barras asignado por el sistema y presenta una llave foránea que es el idEmpacado proveniente de la tabla EspecificacionEmpacado.
- CATEGORIA: La PK es el código de la categoría el cual debe ser mayor a 0, y tiene una llave foránea proveniente de "Producto".
- BODEGA: La PK de esta relación es el id de la bodega el cual es de tipo "Integer" y no puede ser duplicado. Presenta una llave foránea que es el id de la Sucursal.
- RECEPCION PRODUCTOS: La super llave se compone del id de la Orden la cual hace referencia a una llave foránea proveniente de "Orden Compra" y del id de la bodega.
- PROVEEDOR: Su llave primaria es el NIT el cual es un valor que no puede ser nulo ni duplicado.

### 3) Normalización

- **Primera Forma Normal (1NF):** Cada atributo debe contener valores atómicos, es decir, no debe haber listas ni conjuntos de valores en una sola celda.

Se evidencia que este requisito se cumpla ya que cada atributo usado en el modelo es atómico, es decir, no se observan multivalores en ninguna celda.

Por lo tanto, el modelo cumple con la 1NF.

- **Segunda Forma Normal (2NF):** Debe estar en 1NF y todos los atributos no clave deben depender completamente de toda la clave primaria. No debe haber dependencias parciales.

En tablas como InfoExtraProveedor y demás tablas que caracterizan entidades intermedias, se evidencia que los atributos dependen en la totalidad de la SuperLlave especificada.

En conclusión, el modelo cumple con la 2NF.

- **Tercera Forma Normal (3NF):** Debe estar en 2NF y no debe haber dependencias transitivas entre los atributos no clave y la clave primaria.

En tablas como Producto, Venta y OrdenDeCompra, se evidencia que no existen dependencias transitivas; todos los atributos no clave dependen directamente de la clave primaria sin pasar por otro atributo no clave. Por consiguiente, el modelo cumple con la 3NF.

- **Forma Normal de Boyce-Codd (BCNF):** Debe estar en 3NF y, para cada dependencia funcional de la forma  $X \rightarrow Y$ , X debe ser una superclave.

Al estar todo el modelo en 3NF podemos afirmar que ya se cumple este requisito. Por otro lado, aquellas tablas que presentan una llave primaria de solo un atributo permiten identificar que el modelo esta en BCNF por sus llaves simples.

Sin embargo, respecto a las llaves compuestas de las tablas intermedias hay que analizar que cada llave sea superclave. En este caso, se puede afirmar que a partir de cada integrante de la superllave se puede llegar a los otros atributos. Por lo cual las tablas en las cuales las llaves primarias son compuestas también cumplen con los requisitos.

En conclusión, el modelo cumple con BCNF.