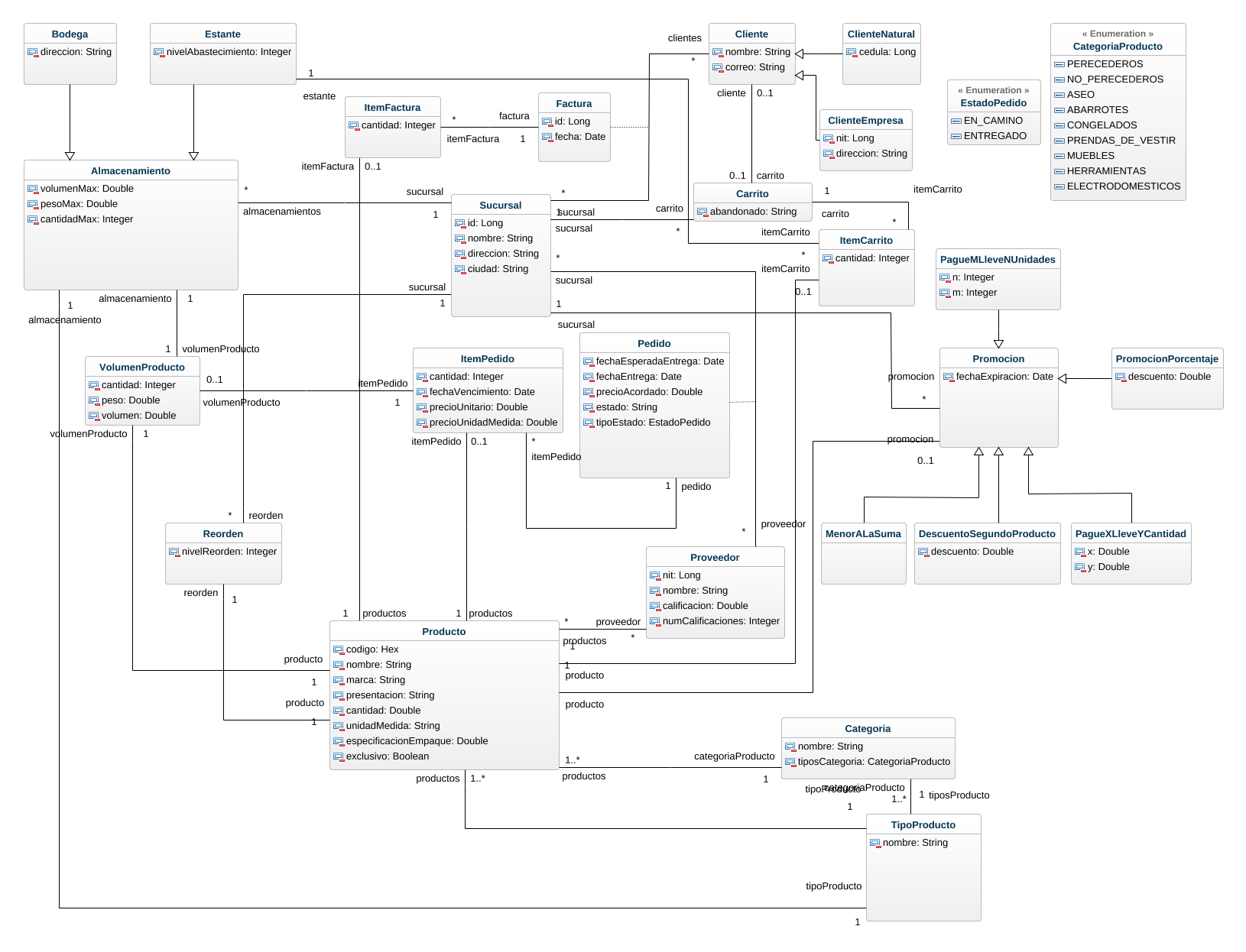
# ITERACIÓN 2

# ANÁLISIS

* **Modelo conceptual**



Los cambios que se realizaron al modelo fueron los siguientes:

* Se creó la clase Carrito para manejar los carritos de una sucursal
* Se creó la clase ItemCarrito para manejar todos productos que se agregan aun carrito
* Se modificaron las relaciones de la clase VolumenProducto porque estaba siendo usada para muchas cosas, para solucionar esto se creó la clase ItemPedido, donde ahora se maneja la cantidad de productos que se piden al proveedor, la fecha de vencimiento de estos, y los precios unitarios de compra y venta de los productos. Asimismo, se creó la clase ItemFactura donde se manejan las n unidades vendidas de un producto.
* Se creó una asociación entre las clases Carrito y Estante para poder saber de cuál estante se sacaron los productos que se agregan al carrito.
* En la clase cliente se agregó el atributo contraseña para poder garantizar el requerimiento no funcional de privacidad.

# DISEÑO DE LA APLICACIÓN

1. El impacto que genera la modificación del modelo conceptual es que es necesario añadir nuevas tablas al modelo relacional y también nuevos atributos en algunas tablas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Carrito** | | | | |
| id | idSucursal | idCliente | estado | idEstante |
| PK | PK, FK(Sucursal.id) | FK(Cliente.id) | NN | FK(Estante.id) |

|  |  |  |  |
| --- | --- | --- | --- |
| **ItemCarrito** | | | |
| id | idCarrito | idProducto | cantidad |
| PK | PK, FK(Carrito.id) | PK, FK(Producto.id) | NN |

|  |  |  |  |
| --- | --- | --- | --- |
| **ItemPedido** | | | |
| id | idPedido | cantidad | fechaVencimiento |
| PK | FK(Pedido.id), NN | NN |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ItemFactura** | | | |
| id | idFactura | idProducto | cantidad |
| PK | FK(Factura.id), NN | FK(Producto.id) | NN |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **VolumenProducto** | | | | |
| Id | cantidad | peso | volumen | idProducto |
| PK | NN, CK(cantidad>0) | CK(peso > 0) | CK(volumen > 0) | FK(Producto.id), NN |

1. Listado de tablas

A close up of text on a white background

Description generated with very high confidence

Tomado del archivo ListadoTablas ubicado en la carpeta docs.

1. Lógica de los nuevos requerimientos.

Precondición: El cliente ha iniciado sesión satisfactoriamente.

* RF12 Solicitar carrito de compras:

1. El cliente selecciona la opción de solicitar carrito
2. Ingresa el id de la sucursal
3. La aplicación crea una nueva tupla en la tabla Carrito, asociándolo con el cliente y la sucursal.
4. El cliente recibe el id del carrito que se creó.
5. En caso de cualquier error que pudiera ocurrir, se realiza rollback.

* RF13 Adicionar un producto al carrito de compras.

1. El cliente selecciona la opción añadir producto y el id de su carrito.
2. Superandes verifica que el carrito sea del cliente para proceder.
3. El cliente ingresa el id del producto que va a agregar, las unidades que quiere y el id del estante del que obtuvo los productos.
4. Superandes resta las n unidades que obtuvo el cliente al estante y las suma al ítem carrito que se creó.

* RF14 Devolver un producto del carrito de compras.

1. El cliente ingresa el id del producto que desea devolver y las unidades que desea devolver.
2. Superandes aumenta en n la cantidad de productos del estante del que el cliente tomó los n productos.
3. Si el cliente decide devolver todos los productos, se elimina la tupla itemCarrito, que representa las n unidades del producto que elgió el cliente.

* RF15 Pagar la compra.

1. Se genera una factura con items factura (revisar si tiene promoción)
2. Se verifica que los estantes estén por encima de su nivel de abastecimiento, sino se trae de bodega.
3. Si se trae de bodega se verifica el nivel de reorden del producto en esa sucursal contra la suma (de las cantidades) de todos los volumenes producto (de ese producto) de las bodegas de esa sucursal.
4. Si es menor, se genera un pedido (o varios), el cual puede contener muchos productos.

* RF17 Abandonar carrito de compras.
  + 1. Cuando el cliente sale de su sesión en SuperAndes sin haber pagado los productos que llevaba en su carrito se procede a realizar el requerimiento RF17.
* RF16 Recolectar carritos abandonados

1. Esto ocurre de dos maneras:

* Si el cliente sale de su sesión, se le pide el id del carrito que tenía asociado, se devuelven a los estantes los productos que tenía y se elimina el carrito.
* Si selecciona la opción de abandonar carrito se le pide el id del carrito que tenía asociado, se devuelven a los estantes los productos que tenía y se elimina el carrito.

En cada uno de los requerimientos anteriores si no se efectúa la transacción completa se hace rollBack para garantizar la atomicidad de la transacción.

Cada requerimiento tiene una implementación en la capa de lógica respetando las reglas del negocio para garantizar la consistencia de la base de datos.

Para ciertas transacciones se usa el nivel de aislamiento Serializable y para las demás usamos el nivel Read Committed.

Gracias al sistema de recuperación ante fallas de Oracle, garantizamos la durabilidad y el estado coherente de la base de datos.