

GESTIÓN DE UNA ZAPATERÍA

Iñaki Melguizo Marcos

David Armenteros Soto

César Muñoz Reinoso

DESCRIPCIÓN DEL SISTEMA

Un propietario de una zapatería de Granada nos ha solicitado un programa para gestionar las compras, ventas de la zapatería así como la información asociada a los clientes y proveedores.

La tienda diariamente comprará y venderá productos (a los proveedores y clientes respectivamente) sobre los que estará obligado a tener un registro para el correcto funcionamiento de la zapatería. También los proveedores lanzarán promociones dirigidas al gerente de la zapatería y este último tendrá la opción de liquidar productos según el tiempo que lleven los productos en la tienda.

En la tienda, el gerente venderá sus productos a los clientes, a la vez que compra a distintos proveedores (marcas). El gerente podrá crear un registro para cada cliente con la finalidad de gestionar las ventas de cada uno. Dichos proveedores ofrecen sus productos y en ocasiones lanzarán ofertas con el fin de dar salida a más zapatillas a un precio más bajo. También tendrá la opción de aceptar la promoción siempre y cuando esté dentro del plazo señalado por el proveedor, realizando automáticamente el pedido del mismo.

Para realizar las distintas funcionalidades dividiremos el sistema en los siguientes subsistemas:

- **Ciente:** Se encarga de gestionar los clientes registrados en la tienda. Se guardará información personal del cliente como su nombre, teléfono, y correo

electrónico. Además se le permitirá al cliente modificar sus datos personales, eliminar su ficha personal de cliente de la zapatería. El gerente podrá acceder a dicha información introduciendo el correo electrónico del cliente. Cada cliente tiene un valor único con el que se identifica en las ventas que realice.

- **Ventas:** Se encarga de gestionar las distintas ventas realizadas por la tienda. Al efectuarse una venta, se guardará información detallada para el correcto funcionamiento de la tienda. El cliente podrá interactuar con el sistema para realizar reservas de los productos que no estén en stock en ese momento y cancelar reservas que ya no le interesan solo en el caso de que no hayan sido vendidas todavía. Se realiza una distinción a la hora de vender productos directamente o productos que han sido reservados previamente. Se pueden consultar las ventas realizadas a un cliente concreto con su correspondiente identificador.

El sistema lleva un control del stock de cada uno de los productos de tal forma que cuando esté por debajo de un determinado número se hará una recomendación de comprar para reponer el género.

- **Pedidos:** Se encarga de gestionar las distintas compras que realiza la tienda a los proveedores. El gerente será el encargado de realizar el pedido de un producto, confirmar la llegada de dicho producto o devolver el producto si así se requiere en el caso de que aún no haya sido recibido en la tienda. Se podrá realizar un seguimiento del pedido mostrando el estado del mismo (en tránsito, recibido o devuelto).
- **Proveedores (Marcas):** Se encarga de gestionar las marcas/proveedores de los distintos productos de la zapatería, para poder liquidar productos, promocionar productos y calificar proveedores dependiendo de las ventas de sus productos a los clientes.

SUBSISTEMA DE CLIENTES

Requisitos Funcionales y Requisitos de datos

RF1.1: Registrar cliente: Crear una ficha nueva de cliente previamente no registrado. El cliente proporciona al sistema un correo electrónico, nombre y teléfono. Dicho correo no puede coincidir con el correo de otro cliente. El sistema asigna un identificador distinto a cada cliente.

- Entrada:
 - Agente externo: Cliente
 - Acción: Solicitud de registro al sistema
 - Requisito de datos de entrada: RDE 1.1.1

- **RDE1.1.1 Datos personales del cliente** proporcionados por el propio cliente y el responsable de la tienda cuando se realiza una ficha, se compone de:
 - Nombre: Cadena de hasta 40 caracteres.
 - Teléfono: Cadena de hasta 9 caracteres numéricos.
 - Correo electrónico: cadena de hasta 40 caracteres alfanuméricos con el formato “caracteres@caracteres.caracteres”

- BD: Requisito de datos de escritura RDW1.1.1 RDW1.1.2

- **RDW1.1.1 Datos personales del cliente ya registrado** proporcionados por el propio cliente y el responsable de la tienda cuando se realiza una ficha, se compone de:
 - Nombre: Cadena de hasta 40 caracteres.
 - Teléfono: Cadena de hasta 9 caracteres numéricos.
 - Correo electrónico: cadena de hasta 40 caracteres alfanuméricos con el formato “caracteres@caracteres.caracteres”

- **RDW1.1.2 Identificador del cliente para registro**, identificador único creado por el sistema una vez se registra el cliente
 - Número identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)

- Salida:
 - Agente externo: Cliente
 - Acción: Confirmación de registro
 - Requisito de datos de salida: ninguno

RF1.2: Modificar correo del cliente: Modificar correo electrónico de un cliente ya registrado. Se le introduce al sistema el identificador del cliente cuyo correo queremos modificar y el nuevo correo que se le asignará a ese cliente.

- Entrada:
 - Agente externo: Cliente
 - Acción: Solicitud de modificación
 - Requisito de datos de entrada: RDE1.2.1

- **RDE1.2.1 Datos a modificar del cliente** proporcionados por el propio cliente y el responsable de la tienda cuando se realiza una ficha se compone de:
 - Número identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Correo electrónico: cadena de hasta 40 caracteres alfanuméricos con el formato “caracteres@caracteres.caracteres”

- BD: Requisito de datos de escritura RDW1.2.1

- **RDW1.2.1 Datos a modificar del cliente** proporcionados por el propio cliente y el responsable de la tienda cuando se realiza una ficha se compone de:
 - Correo electrónico: cadena de hasta 40 caracteres alfanuméricos con el formato “caracteres@caracteres.caracteres”

- Salida:
 - Agente externo: Cliente
 - Acción: Confirmación de modificación
 - Requisito de datos de salida: ninguno

RF1.3: Eliminar cliente: El sistema elimina la ficha del cliente cuyo identificador se le indica en la entrada.

- Entrada:
 - Agente externo: Cliente
 - Acción: Solicitud de eliminación
 - Requisito de datos de entrada: RDE1.3.1

- **RDE1.3.1 Identificador del cliente para eliminarlo**, identificador único creado por el sistema una vez se registra el cliente
 - Número identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)

- BD: Requisito de datos de escritura RDW1.3.1
Requisito de datos de lectura RDR 1.3.1

- **RDR1.3.1 Identificador del cliente para eliminarlo**, identificador único creado por el sistema una vez se registra el cliente
 - Número identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)

- **RDW1.3.1 Datos personales del cliente: identificador, correo electrónico, nombre y teléfono para eliminarlo**,
 - Nombre: Cadena de hasta 40 caracteres.
 - Teléfono: Cadena de hasta 9 caracteres numéricos.
 - Número identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Correo electrónico: cadena de hasta 40 caracteres alfanuméricos con el formato "caracteres@caracteres.caracteres"

- Salida:
 - Agente externo: Cliente
 - Acción: Confirmación de eliminación
 - Requisito de datos de salida: ninguno

RF 1.4: Mostrar teléfono, nombre e identificador del cliente: El sistema muestra al gerente el nombre, teléfono e identificador de un cliente determinado recibiendo el correo electrónico como entrada. Este identificador le servirá al gerente para asociar las ventas que realice este cliente con ese identificador.

- Entrada:
 - Agente externo: Gerente
 - Acción: Solicitar información
 - Requisito de datos de entrada: RDE1.4.1

- **RDE1.4.1 Datos de búsqueda del cliente** proporcionados por el propio cliente y el responsable de la tienda cuando se realiza una ficha se compone de:
 - Correo electrónico: cadena de hasta 40 caracteres alfanuméricos con el formato “caracteres@caracteres.caracteres”

- BD: Requisito de datos de lectura RDR1.4.1

- **RDR1.4.1 Datos personales del cliente para mostrar** proporcionados por el propio cliente y el responsable de la tienda cuando se realiza una ficha, se compone de:
 - Número identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Nombre: Cadena de hasta 40 caracteres.
 - Teléfono: Cadena de hasta 20 caracteres.
 - Correo electrónico: cadena de hasta 40 caracteres alfanuméricos con el formato “caracteres@caracteres.caracteres”

- Salida:
 - Agente externo: Gerente
 - Acción: Confirmación de resultado
 - Requisito de datos de salida: RDS 1.4.1

- **RDS1.4.1 Datos personales del cliente para mostrar** proporcionados por el propio cliente y el responsable de la tienda cuando se realiza una ficha, se compone de:
 - Número identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Nombre: Cadena de hasta 40 caracteres.
 - Teléfono: Cadena de hasta 20 caracteres.

Restricciones Semánticas

RS1.1 El correo electrónico debe de ser único para cada cliente.

- RF: 1.1
- RD: RDE1.1.1

SUBSISTEMA DE VENTAS

Requisitos Funcionales

RF2.1: Venta directa de un producto: Se realiza una venta de un producto sin previa reserva cuando tiene stock en la tienda. Cada venta se describe por un identificador de producto, identificador de cliente y por una cantidad de producto. El sistema permite vender varias unidades de un mismo producto a un cliente. Se consulta el stock de ese producto tras la venta y se recomienda pedirlo si el stock es escaso.

- Entrada:
 - Agente externo: Gerente
 - Acción: Vender producto sin reserva
 - Requisito de datos de entrada: RDE2.1.1

- **RDE2.1.1 Identificador del producto y del cliente**, asociado a un producto y a un cliente respectivamente.
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad de producto (número entero)

- BD: Requisito de datos de escritura RDW2.1.1 RDW2.1.2 RDW2.1.3
:Requisito de datos de lectura RDR2.1.1 RDR 2.1.2

- **RDR 2.1.1 Producto y cantidad en stock**, habría que ver si el producto está en stock, se describe por:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Número de pares en stock (cadena numérica de dos enteros)

- **RDR 2.1.2 Identificador del cliente**, se describe por:
 - Identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)

- ❑ **RDW2.1.1 Datos de Venta**, se describe por:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)

- ❑ **RDW2.1.2 Datos de Venta**, se describe por:
 - Identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)

- ❑ **RDW2.1.3 Datos de Venta**, se describe por:
 - Cantidad de unidades del producto (Número entero positivo)

- Salida:
 - Agente externo: Gerente
 - Acción: Confirmación de venta directa
 - Requisito de datos de salida: ninguno

RF2.2: Consultar ventas por cliente: Muestra las ventas según el identificador del cliente. De esta forma se puede consultar qué producto y cantidad del mismo fue la que se vendió a dicho cliente.

- Entrada:
 - Agente externo: Gerente
 - Acción: Consultar ventas
 - Requisito de datos de entrada: RDE 2.2.1

- ❑ **RDE2.2.1 Identificador del cliente**, identificador único creado por el sistema una vez se registra el cliente
 - Identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)

- BD: Requisito de datos de lectura RDR2.2.1

- ❑ **RDR2.2.1 Venta**, se describe por:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)

- Identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)
- Cantidad de unidades del producto (Número entero)

- Salida:
 - Agente externo: Gerente
 - Acción: Confirmación de resultado
 - Requisito de datos de salida: RDS2.2.1

❑ **RDS2.2.1** Listado de registros, cada uno de ellos con los datos de **RDR2.2.1**

RF 2.3: Venta producto reservado, se realiza una venta con previa reserva. Esto se puede realizar cuando no hay stock en la tienda. Se venderá cuando haya una reserva previa realizada por el cliente. Una vez que se tiene el identificador de la reserva, para poder vender el producto se tiene que cambiar el estado de la reserva para indicar al sistema que ya ha sido vendida.

- Entrada:
 - Agente externo: Gerente
 - Acción: Vender producto reservado
 - Requisito de datos de entrada: RDE2.3.1

❑ **RDE2.3.1 Identificador de reserva**, identificador único creado por el sistema una vez se registra la reserva

- Número identificador de la reserva (cadena alfanumérica de hasta 10 caracteres no vacía)

- BD: Requisito de datos de escritura RDW2.3.1 RDW2.3.2
 Requisito de datos de lectura RDR2.3.1 RDR2.3.2

❑ **RDW2.3.1 Venta**, se describe por:

- Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
- Identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)
- Cantidad de unidades del producto (Flotante con dos decimales)

- ❑ **RDW2.3.2 Estado de reserva**, se describe por:
 - Estado de la reserva (cadena alfanumérica de hasta 10 caracteres no vacía que puede tomar los valores {"completada", "pendiente", "cancelada"})

- ❑ **RDR 2.3.1 Producto y cantidad en stock**, habría que ver si el producto reservado está ya en stock, se describe por:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Número de pares en stock (cadena numérica de dos enteros)

- ❑ **RDR 2.3.2 Identificador de reserva**, único identificador creado por el sistema una vez se registra la reserva.
 - Identificador de la reserva (cadena alfanumérica de hasta 10 caracteres no vacía)

- Salida:
 - Agente externo: Gerente
 - Acción: Confirmación de venta por reserva
 - Requisito de datos de salida: ninguno

RF2.4: Realizar reserva: Se realiza una reserva de un producto.

- Entrada:
 - Agente externo: Cliente
 - Acción: Realizar reserva
 - Requisito de datos de entrada: RDE2.4.1

- ❑ **RDE2.4.1 Datos de la reserva**, identificadores asociados a cliente y producto y la cantidad del mismo.
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad de unidades del producto (Número entero)

- BD: Requisito de datos de escritura RDW2.4.1 RDW2.4.2
Requisito de datos de lectura RDR2.4.1 RDR2.4.2
- ❑ **RDW2.4.1 Identificador de reserva**, identificador único creado por el sistema una vez se registra la reserva.
 - Identificador de la reserva (cadena alfanumérica de hasta 10 caracteres no vacía)
- ❑ **RDW2.4.2 Estado de la reserva**, indica el estado en el que se encuentra la reserva .
 - Estado de la reserva (cadena alfanumérica de hasta 10 caracteres no vacía que puede tomar los valores {"completada", "pendiente", "cancelada"})
- ❑ **RDR2.4.1 Identificador del producto**, se describe por:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad de Stock
- ❑ **RDR 2.4.2 Identificador del cliente**, se describe por:
 - Identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)
- Salida:
 - Agente externo: Cliente
 - Acción: Confirmación de reserva
 - Requisito de datos de salida: RDS2.4.1
- ❑ **RDS2.4.1 Identificador de reserva**, identificador único creado por el sistema una vez se registra la reserva
 - Identificador de la reserva (cadena alfanumérica de hasta 10 caracteres no vacía)

RF2.5: Cancelar reserva: Se cancela una reserva previamente realizada.

- Entrada:
 - Agente externo: Cliente
 - Acción: Cancelación de reserva
 - Requisito de datos de entrada: RDE2.5.1

- ❑ **RDE2.5.1 Identificador de reserva**, identificador único creado por el sistema una vez se registra la reserva
 - Número identificador de la reserva (cadena alfanumérica de hasta 10 caracteres no vacía)

- BD: Requisito de datos de escritura RDW2.5.1 RSW2.5.2
Requisito de datos de lectura RDR2.5.1

- ❑ **RDR2.5.1 Identificador de reserva**, identificador único creado por el sistema una vez se registra la reserva.
 - Identificador de la reserva (cadena alfanumérica de hasta 10 caracteres no vacía)

- ❑ **RDW2.5.1 Identificador de reserva**, identificador único creado por el sistema una vez se registra la reserva.
 - Identificador de la reserva (cadena alfanumérica de hasta 10 caracteres no vacía)

- ❑ **RDW2.5.2 Estado de la reserva**, indica el estado en el que se encuentra la reserva.
 - Estado de la reserva (cadena alfanumérica de hasta 10 caracteres no vacía que puede tomar los valores {"completada", "pendiente", "cancelada"})

- Salida:
 - Agente externo: Cliente
 - Acción: Confirmación de cancelación de reserva

- Requisito de datos de salida: ninguno

Restricciones Semánticas

RS2.1 Realizar una reserva, solo es posible cuando el producto no está en stock

- RF: 2.4
- RD: RDE2.4.1

RS2.2 Cancelación de reserva, no se puede cancelar una reserva cuando ya está vendida

- RF: 2.5
- RD: RDE2.5.1

SUBSISTEMA DE PEDIDOS

Requisitos Funcionales

RF3.1 : Hacer un pedido: comprar un producto a un proveedor.

- Entrada:
 - Agente externo: Gerente
 - Acción: Realizar un pedido
 - Requisito de datos de entrada: RDE3.1.1

- **RDE3.1.1 Datos del pedido** , identificador único asociado a un producto y a un proveedor respectivamente
 - Número identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad de unidades del producto (Número entero)

- BD: Requisito de datos de escritura RDW3.1.1 RDW3.1.2
Requisito de datos de lectura RDR3.1.1

- **RDW3.1.1 Identificador del pedido**, identificador único asociado a un pedido
 - Número identificador del pedido (cadena alfanumérica de hasta 10 caracteres no vacíos.
 - Fecha

- **RDW3.1.2 Estado del pedido**, indica el estado en el que se encuentra el pedido.
 - Estado del pedido (cadena alfanumérica que solo puede tomar estos valores {"recibido", "en tránsito", "devuelto"})

- **RDR3.1.1 Identificador del producto** , identificador único asociado a un producto
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)

- Salida:
 - Agente externo: Gerente
 - Acción: Confirmación pedido realizado
 - Requisito de datos de salida: ninguno

RF3.2: Confirmar pedido: Dar a conocer al sistema que el pedido ha llegado. Se comprueba que hay una compra que tiene como identificador la entrada y se modifica su estado de “en tránsito” a “recibido”

- Entrada:
 - Agente externo: Gerente
 - Acción: Confirmar un pedido recibido
 - Requisito de datos de entrada: RDE3.2.1

- **RDE3.2.1 Identificador del pedido**, identificador único asociado a un pedido
 - Número identificador del pedido (cadena alfanumérica de hasta 10 caracteres no vacía)

- BD: Requisito de datos de escritura RDW3.2.1
Requisito de datos de escritura RDR3.2.1

- **RDR3.2.1 Identificador del pedido** , identificador único asociado a un pedido
 - Número identificador del pedido (cadena alfanumérica de hasta 10 caracteres no vacía)

- **RDW3.2.1 Estado del pedido**, indica el estado en el que se encuentra el pedido.
 - Estado del pedido (cadena alfanumérica que solo puede tomar estos valores {"recibido", "en tránsito", "devuelto"})

- Salida:
 - Agente externo: Gerente
 - Acción: Confirmación de pedido recibido
 - Requisito de datos de salida: ninguno

RF3.3: Ver estado pedido: Ver la información respecto a un pedido.

- Entrada:
 - Agente externo: Gerente
 - Acción: Ver estado de un pedido
 - Requisito de datos de entrada: RDE3.3.1

- ❑ **RDE3.3.1 Identificador del pedido**, identificador único asociado a un pedido
 - Número identificador del pedido (cadena alfanumérica de hasta 10 caracteres no vacía)
- BD: Requisito de datos de lectura RDR3.3.1
- ❑ **RDR3.3.1 Pedido**, se describe por:
 - Identificador del pedido (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Estado del pedido
- Salida:
 - Agente externo: Gerente
 - Acción: Mostrar el estado de pedido
 - Requisito de datos de salida: RDS3.3.1
- ❑ **RDS3.3.1 Estado del pedido**, se describe por:
 - Estado del pedido (cadena alfanumérica que solo puede tomar estos valores {"recibido", "en tránsito", "devuelto"})

RF3.4: Devolver pedido: Devolución de pedido mientras esté en tránsito. Se cambia el estado del pedido determinado por el identificador de entrada y se cambia su estado de "en tránsito" a "devuelto"

- Entrada:
 - Agente externo: Gerente
 - Acción: Solicitud de devolución de un pedido
 - Requisito de datos de entrada: RDE3.4.1
- ❑ **RDE3.4.1 Identificador del pedido**, identificador único asociado a un pedido
 - Número identificador del pedido (cadena alfanumérica de hasta 10 caracteres no vacía)
- BD: Requisito de datos de escritura RDW3.4.1
Requisito de datos de lectura RDR3.4.1
- ❑ **RDR3.4.1 Identificador del pedido**, identificador único asociado a un pedido

- Número identificador del pedido (cadena alfanumérica de hasta 10 caracteres no vacía)

❑ **RDW3.4.1 Estado del pedido**, se describe por:

- Estado del pedido (cadena alfanumérica que solo puede tomar estos valores {"recibido", "en tránsito", "devuelto"})
- Salida:
 - Agente externo: Gerente
 - Acción: Mostrar el estado de pedido
 - Requisito de datos de salida: ninguno

Requisitos Semánticos

RS3.1 No se puede devolver un pedido ya recibido o devuelto

- RF: 3.4
- RD: RDE 3.4.1

RS3.2: No se puede confirmar un pedido ya recibido o que haya sido devuelto

- RF: 3.2
- RD: RDE 3.2.1

SUBSISTEMA DE PROVEEDORES (MARCAS)

Requisitos Funcionales

RF4.1: Promoción de un nuevo producto: El proveedor quiere promocionar un nuevo producto entre el proveedor y gerente, así que lanzará una oferta. Dicha oferta del nuevo producto consistirá en que el gerente, antes de la fecha de vencimiento de ésta, podrá comprar una cantidad de ese producto superior a la que marca la oferta por el precio por unidad que la oferta indica. Dicha promoción le llegará al gerente cuando el proveedor la oferta.

- Entrada:
 - Agente externo: Proveedor
 - Acción: Lanzar una oferta
 - Requisito de datos de entrada: RDE4.1.1

- **RDE4.1.1 Oferta**, se describe por:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Identificador del proveedor (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad mínima de unidades del producto (Flotante con dos decimales)
 - Precio por unidad del producto (Flotante con dos decimales)
 - Tiempo de expiración (Tiempo en semanas)

- BD: Requisito de datos de escritura RDW4.1.1 RDW 4.1.2
Requisito de datos de lectura RDR4.1.1

- **RDR4.1.1 Identificador del proveedor** , identificador único asociado a un proveedor
 - Identificador del proveedor (cadena alfanumérica de hasta 10 caracteres no vacía)

- **RDW4.1.1 Oferta**, se describe por:
 - Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Identificador del proveedor (cadena alfanumérica de hasta 10 caracteres no vacía)
 - Cantidad mínima de unidades del producto (Flotante con dos decimales)

- Precio por unidad del producto (Flotante con dos decimales)
- Tiempo de expiración (Tiempo en semanas)

❑ **RDW4.1.2 Identificador del producto** , identificador único asociado a un producto

- Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)

● Salida:

- Agente externo: Gerente
- Acción: Notificación de la oferta
- Requisito de datos de salida: ninguno

RF4.2: Liquidación por stock: Se quiere liquidar una serie de productos según su tiempo en tienda, que es la entrada que recibe el sistema para seleccionar los productos a los que se les hará una reducción de su precio

● Entrada:

- Agente externo: Gerente
- Acción: Lanzar una oferta de un producto
- Requisito de datos de entrada: RDE4.2.1

❑ **RDE4.2.1 Tiempo en stock**

- Número de meses (cadena numérica de 2 dígitos)
- Porcentaje de descuento (Número entero de 0-100)

- BD: Requisito de datos de lectura RDR4.2.1
Requisito de datos de lectura RDW4.2.1

❑ **RDR4.2.1 Fecha**, se describe por:

- Fecha en la que un producto actualmente en stock llegó a la tienda

❑ **RDW4.2.1 Listado de registros de Productos** , se describe por:

- Precio de los productos cuya fecha de llegada es anterior a la de liquidación se reduce lo que el porcentaje de descuento indica

- Salida:
 - Agente externo: Gerente
 - Acción: Liquidar un producto
 - Requisito de datos de salida: ninguno

RF4.3: Calificar Proveedor: Se le puntúa al proveedor dependiendo de los productos de dicho proveedor que han sido vendidos a los clientes y los productos vendidos de dicho proveedor al gerente de la zapatería

- Entrada:
 - Agente externo: Gerente
 - Acción: Valorar al proveedor
 - Requisito de datos de entrada: RDE4.3.1

☐ **RDE4.3.1 Proveedor**, al que se quiere valorar:

- Identificador del proveedor (cadena alfanumérica de hasta 10 caracteres no vacía)

- BD: Requisito de datos de lectura RDR4.3.1 RDR4.3.2 RDR4.3.3

☐ **RDR4.3.1 Listado de registros de Pedidos**, se describe por:

- Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
- Cantidad de unidades del producto (Flotante con dos decimales)
- Precio por unidad del producto (Flotante con dos decimales)
- Estado del pedido (cadena alfanumérica que solo puede tomar estos valores {"recibido", "en tránsito", "devuelto"})

☐ **RDR4.3.2 Listado de registros de Ventas**, se describe por:

- Identificador del producto (cadena alfanumérica de hasta 10 caracteres no vacía)
- Identificador del cliente (cadena alfanumérica de hasta 10 caracteres no vacía)
- Cantidad de unidades del producto (Flotante con dos decimales)
- Precio por unidad del producto (Flotante con dos decimales)

☐ **RDR4.3.3 Proveedor**, al que se quiere valorar:

- Identificador del proveedor (cadena alfanumérica de hasta 10 caracteres no vacía)

- Salida:
 - Agente externo: Gerente

- Acción: Liquidar un producto
- Requisito de datos de salida: RDS4.3.1

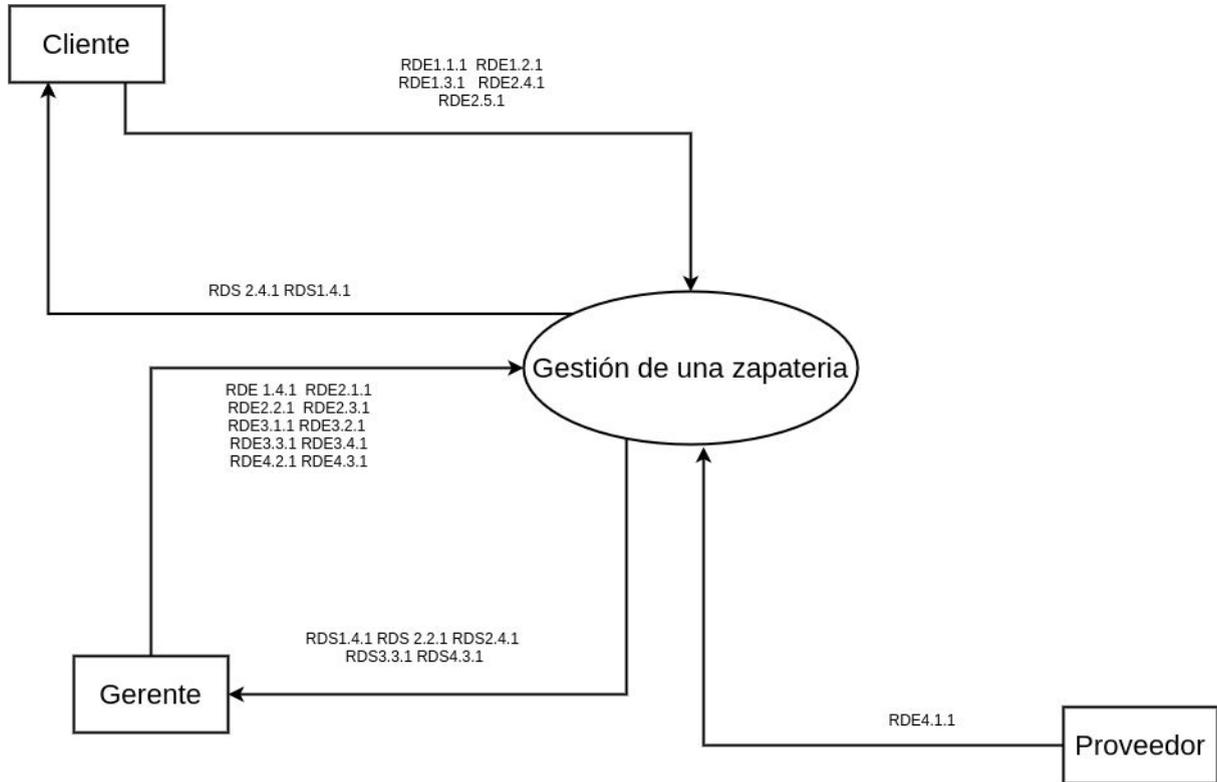
- **RDS4.3.1 Calificación**, se describe por:
 - Valoración (Entero de 0 a 10)

Restricción Semántica

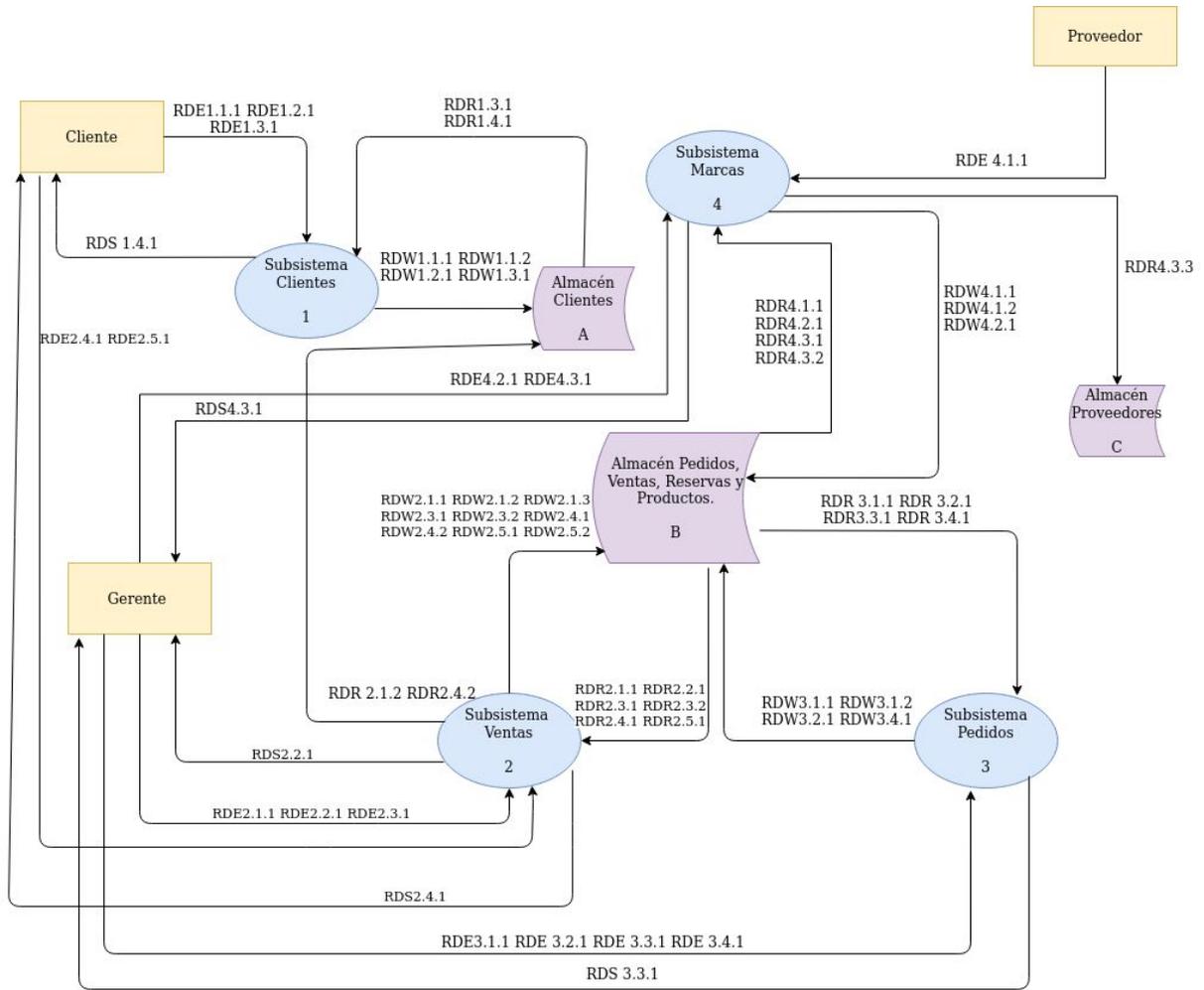
RS4.1 Cuando promocionamos un nuevo producto, el sistema añadirá previamente ese producto, para que la oferta tenga sentido

- RF: 4.1
- RD: RDE 4.1.1

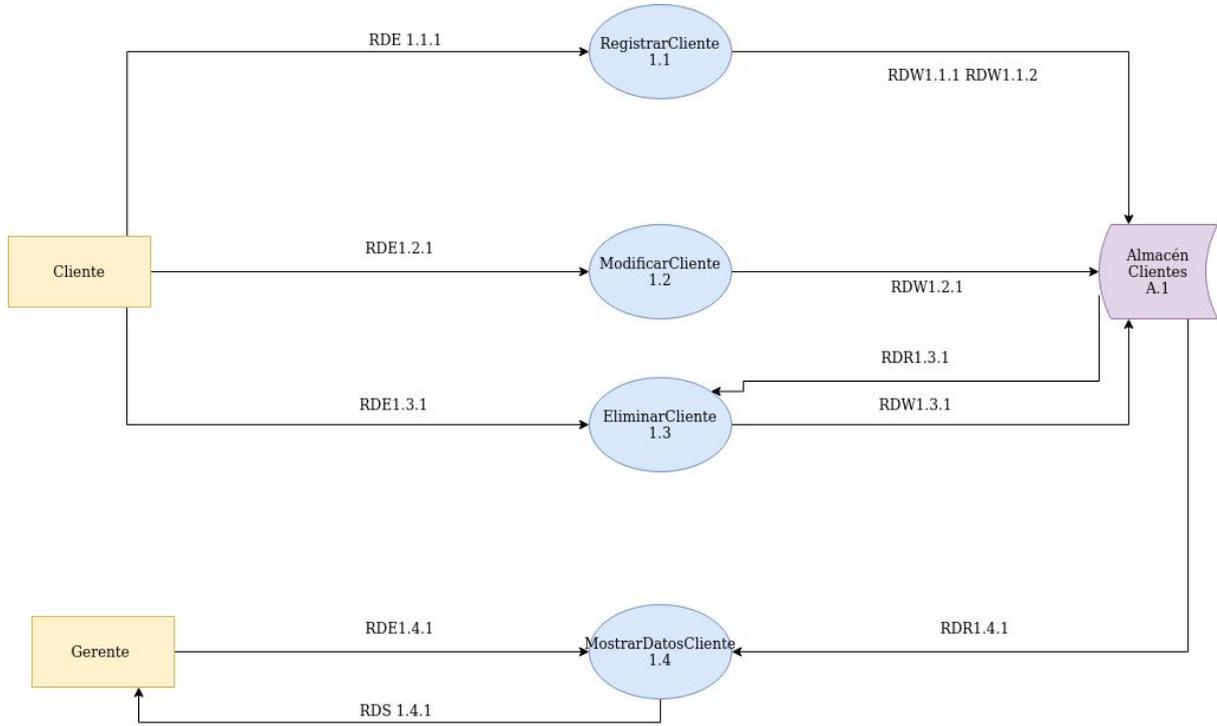
ESQUEMA DE CAJA NEGRA



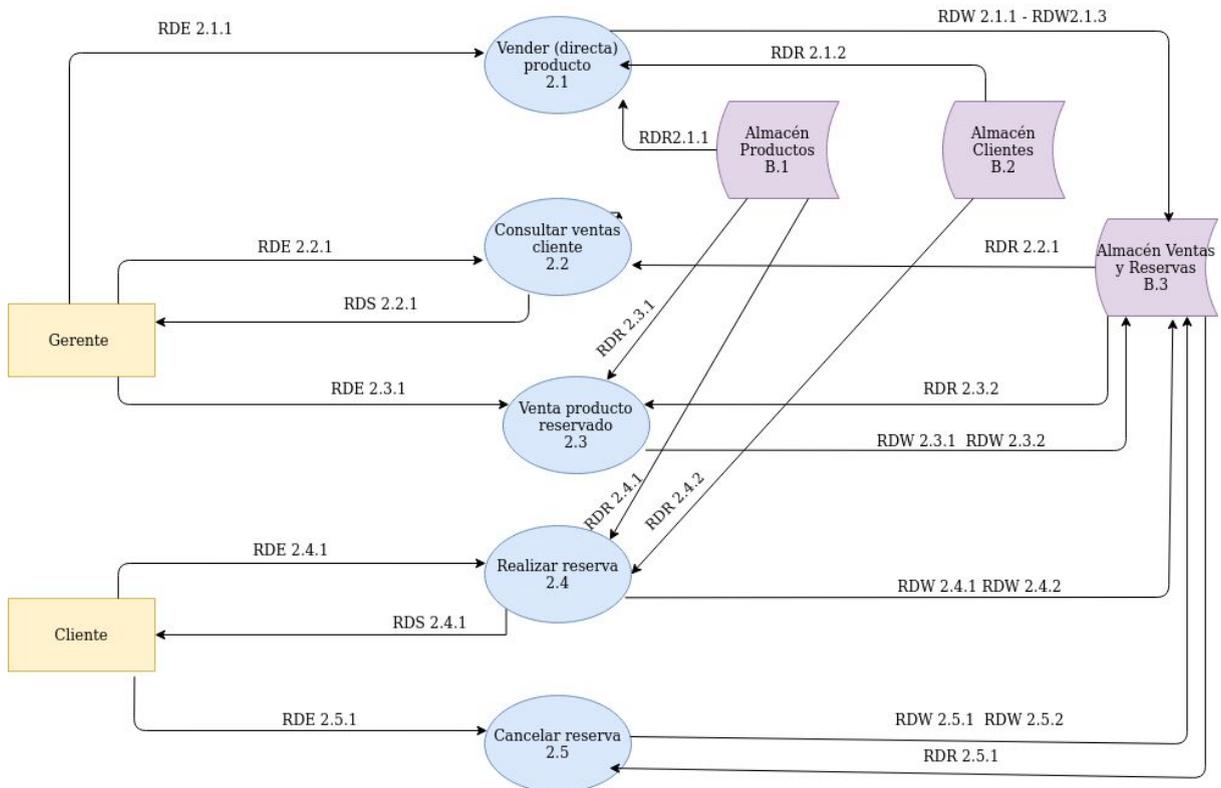
DFD0: ESQUEMA ARMAZÓN



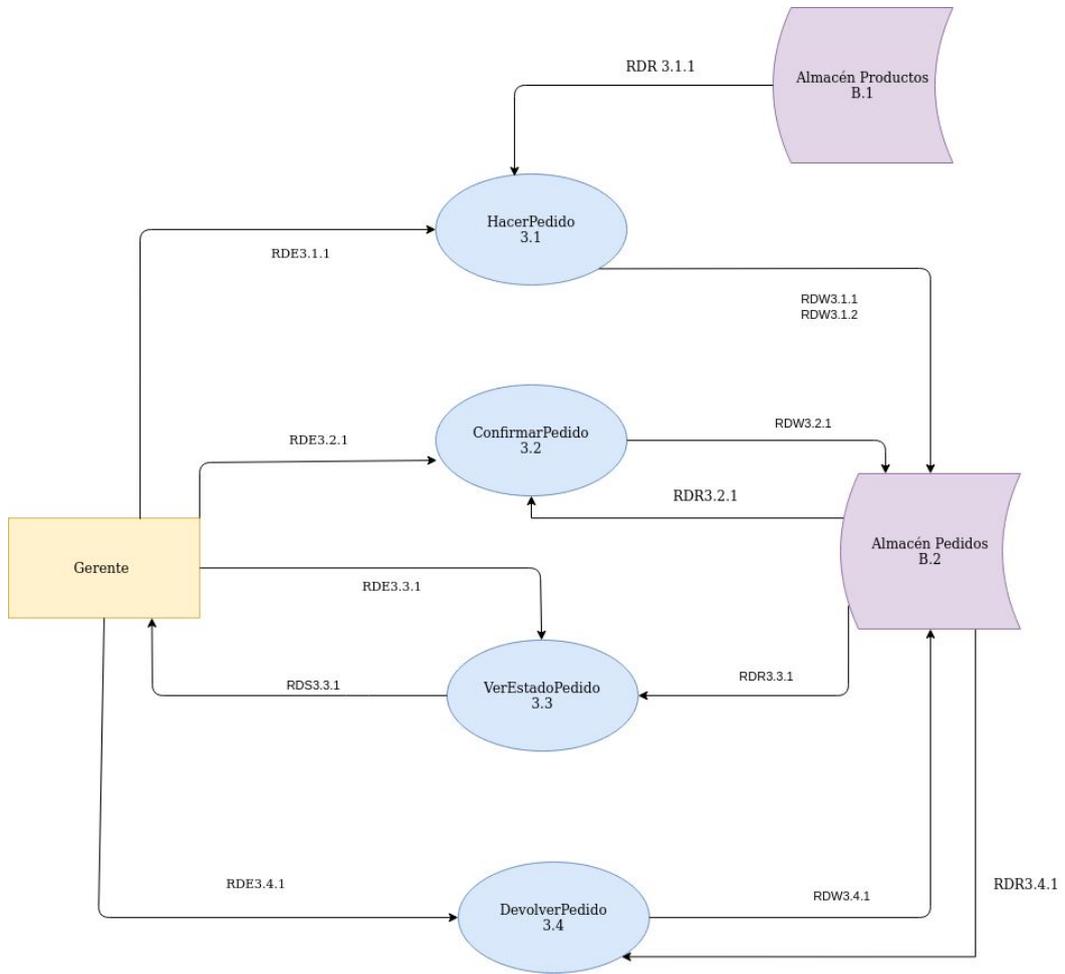
DFD1: ESQUEMA SUBSISTEMA 1



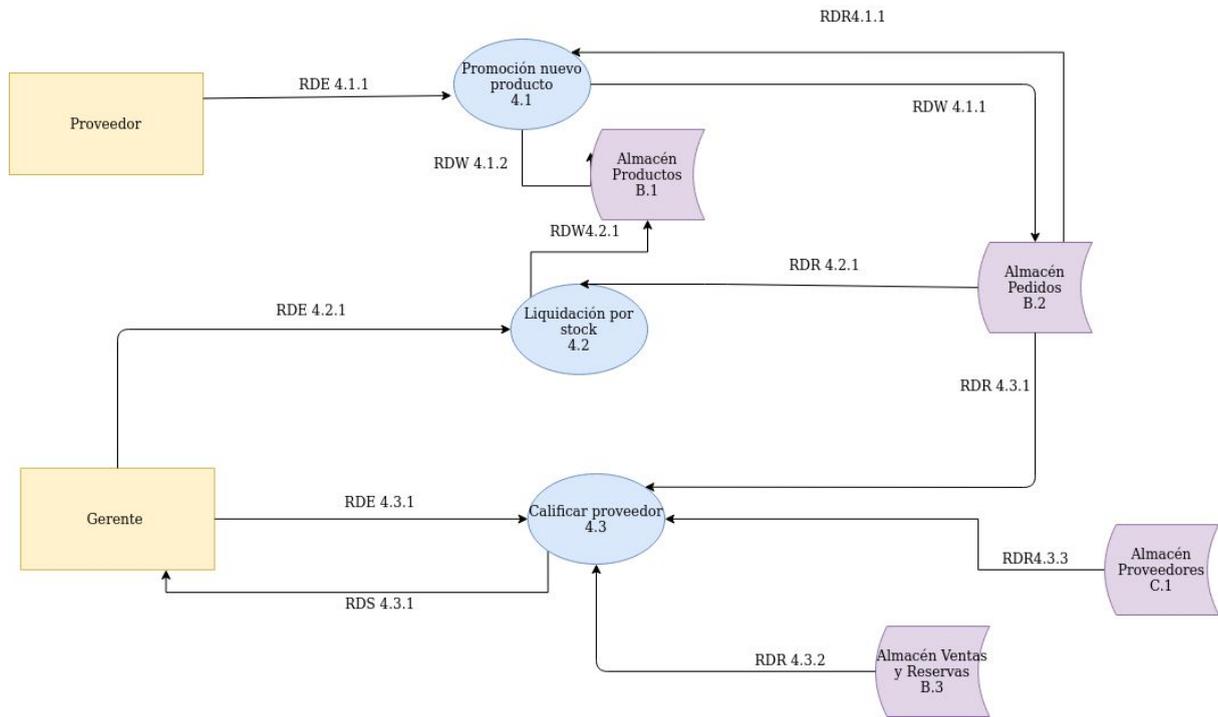
DFD1: ESQUEMA SUBSISTEMA 2



DFD1: ESQUEMA SUBSISTEMA 3



DFD1: ESQUEMA SUBSISTEMA 4



MODIFICACIONES PRÁCTICA 1

SUBSISTEMA DE CLIENTES

- Eliminado RDR1.2.1 (identificador del cliente) en el requisito funcional RF1.2 Modificar correo del cliente.

SUBSISTEMA DE VENTAS

- La recomendación de compra no es una restricción semántica, forma parte de la descripción del requisito funcional venta directa.
- El identificador de la reserva no es una restricción semántica.

SUBSISTEMA DE PROVEEDORES (MARCAS)

- Se añade el porcentaje de descuento al requisito funcional RF4.2 Liquidación por stock como requisito de datos de entrada.
- El precio de la oferta no es una restricción semántica

SUBSISTEMA DE PEDIDOS

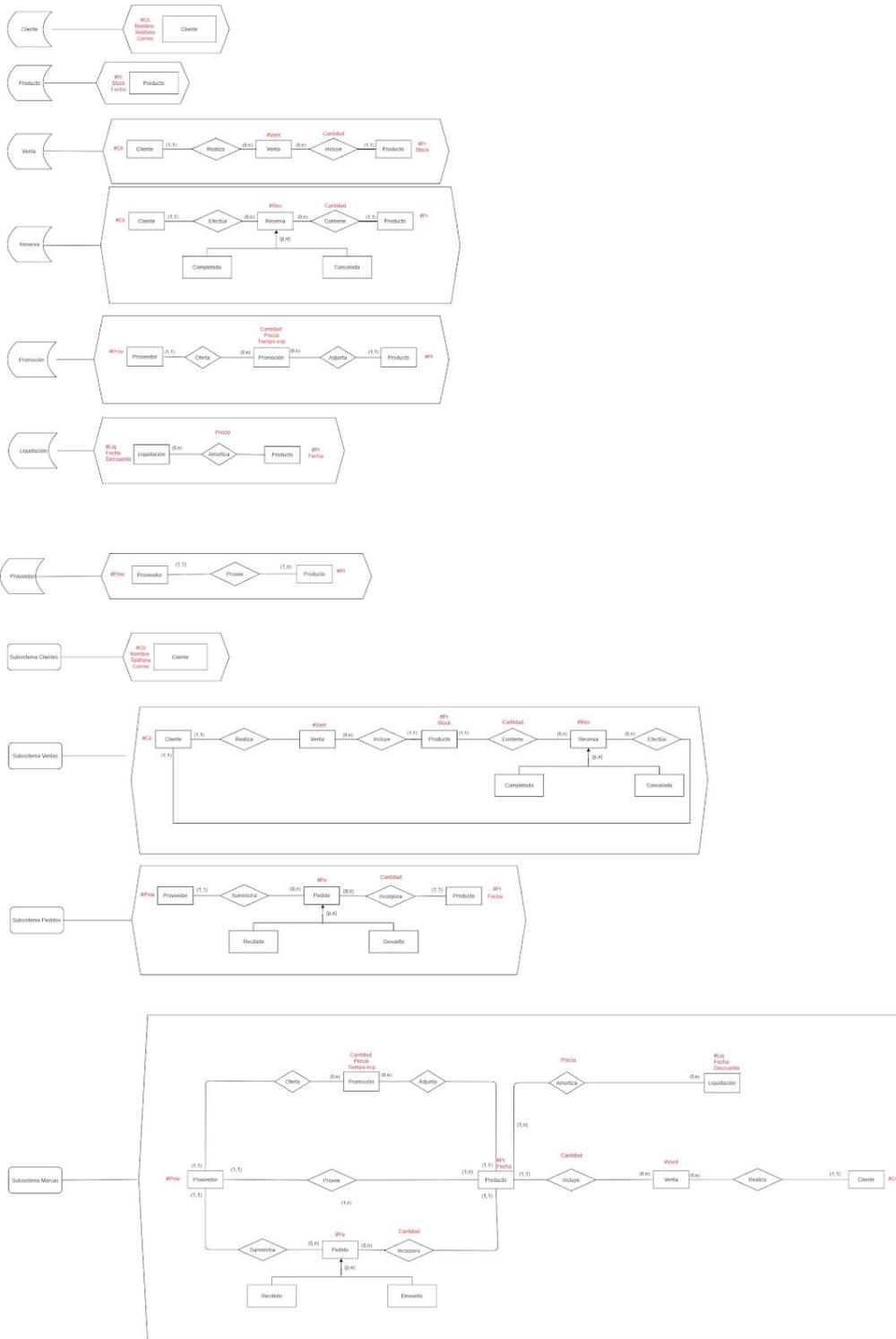
- Eliminado RDR3.1.2 (Identificador del proveedor) en el requisito funcional RF3.1 Hacer un pedido, porque el producto tiene asociado un único proveedor.

DFDs

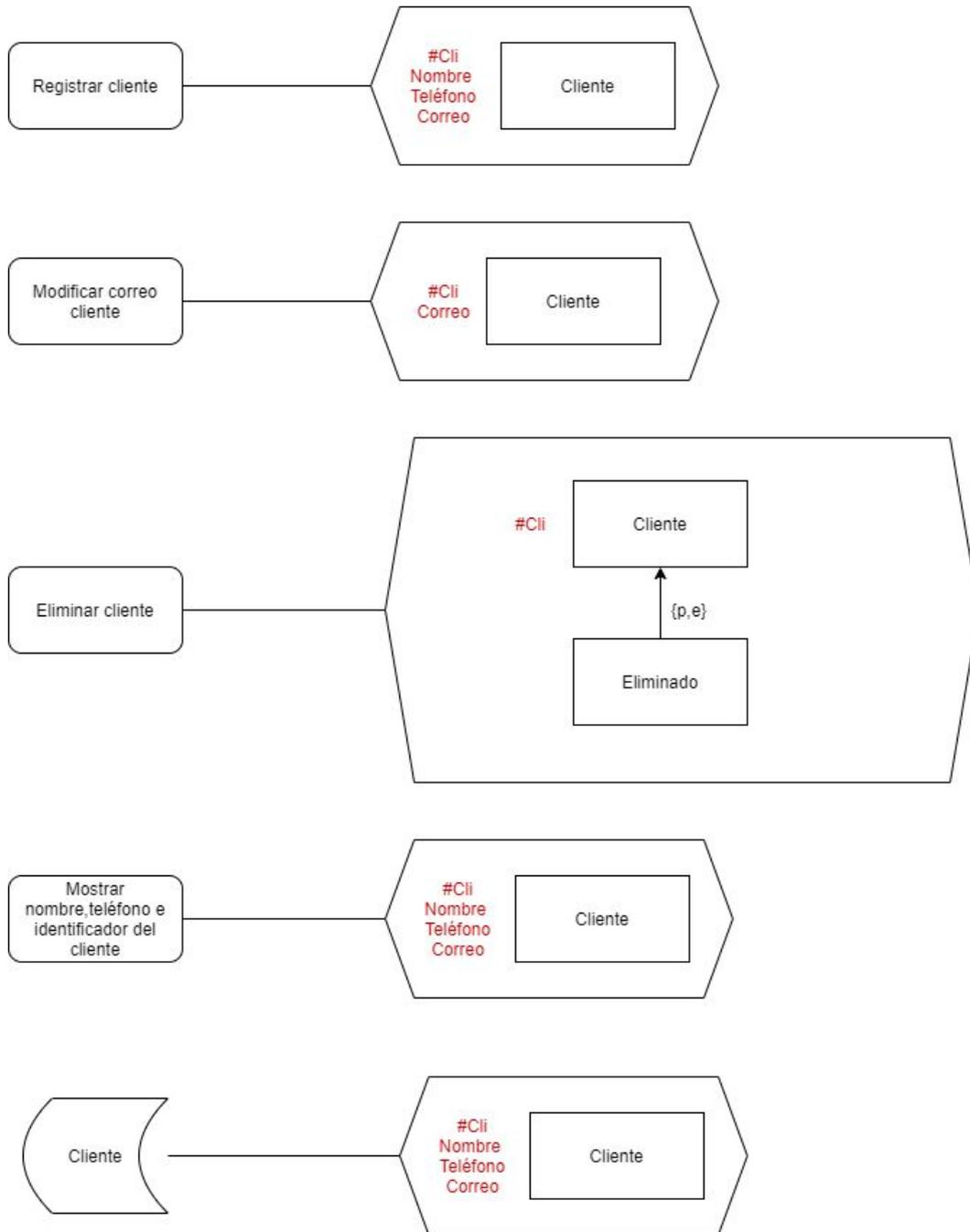
- Cambio de nomenclatura

PRÁCTICA 2

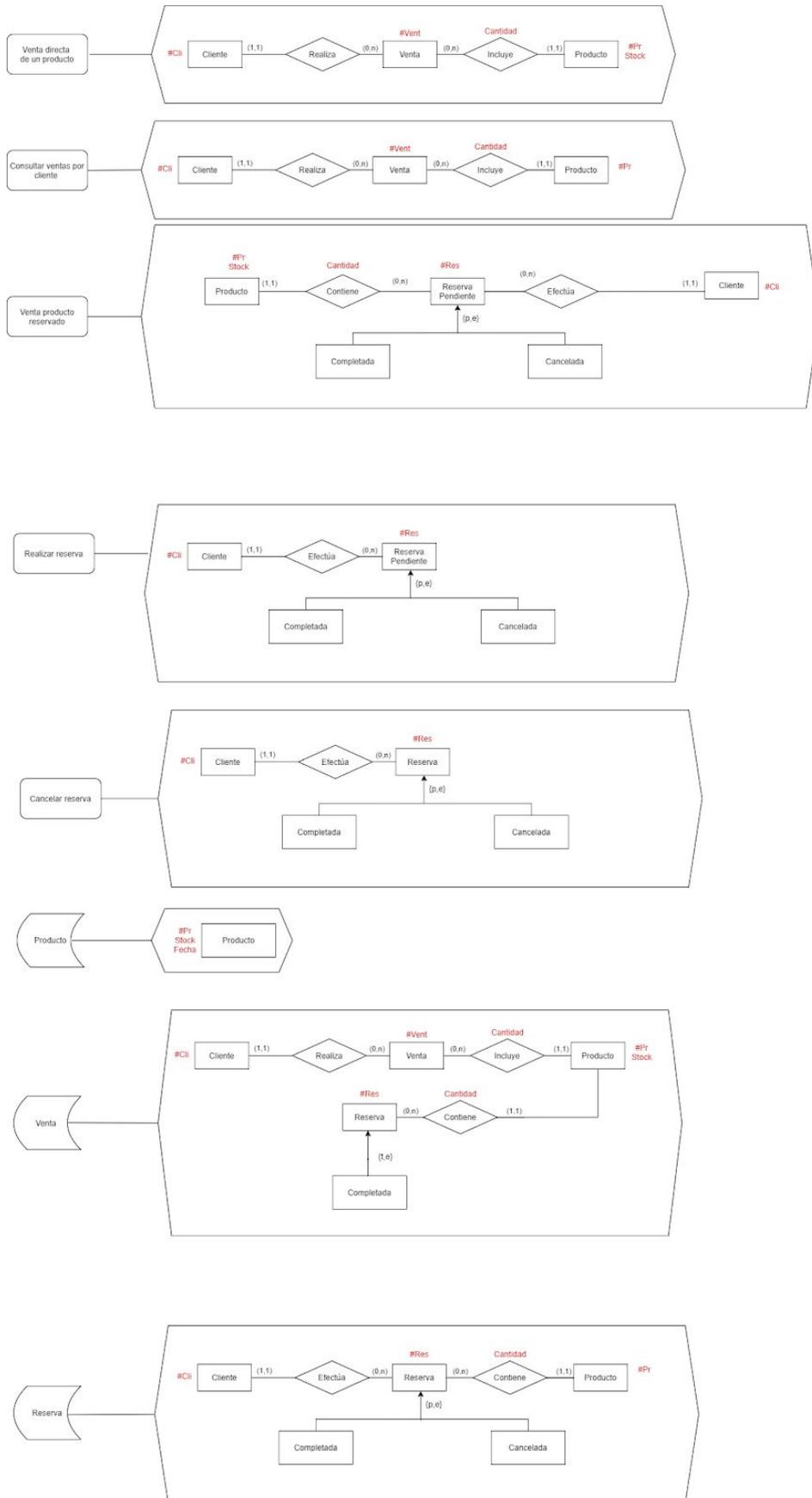
Esquema externo DFD0:



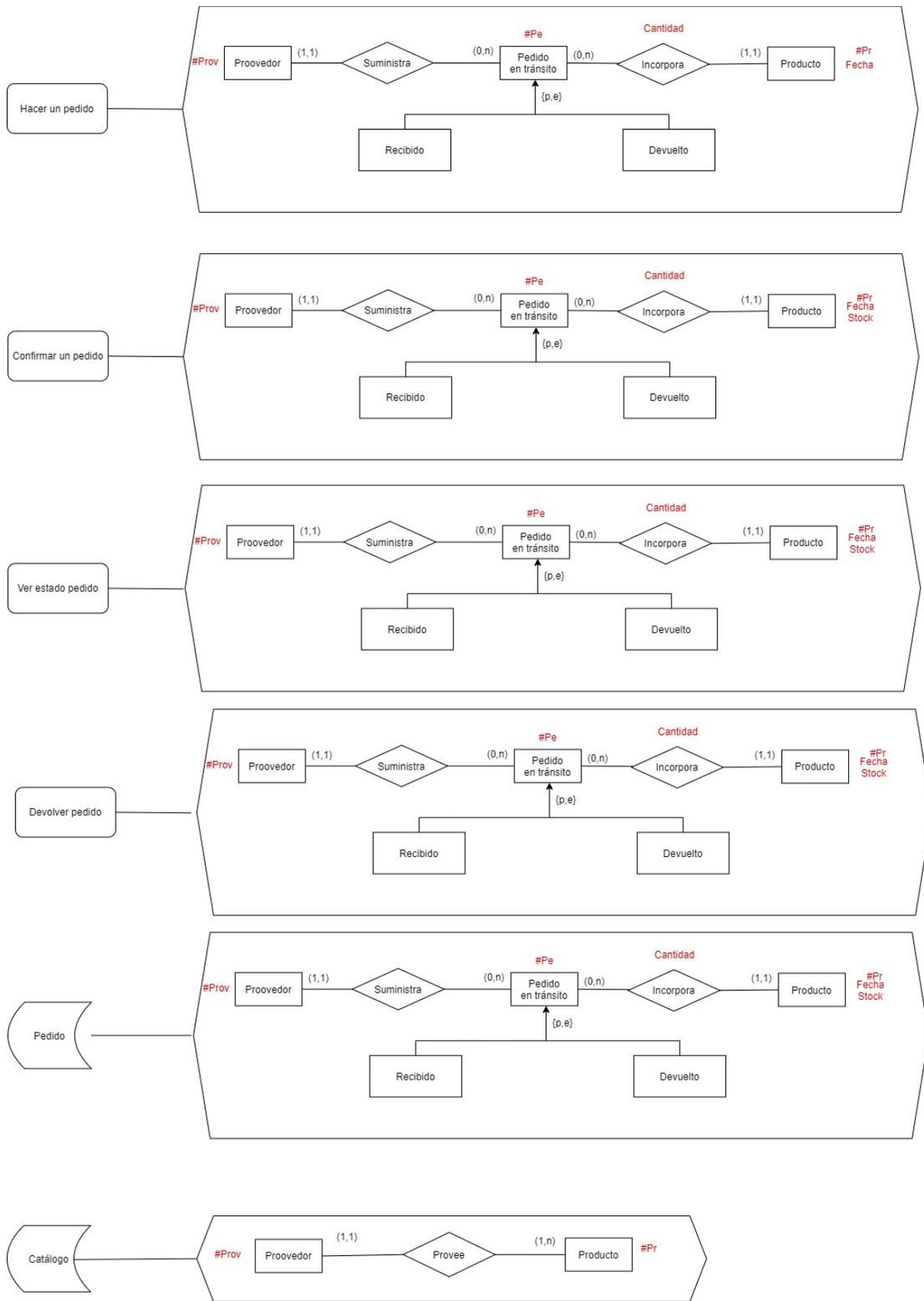
Esquema externo Subsistema Clientes:



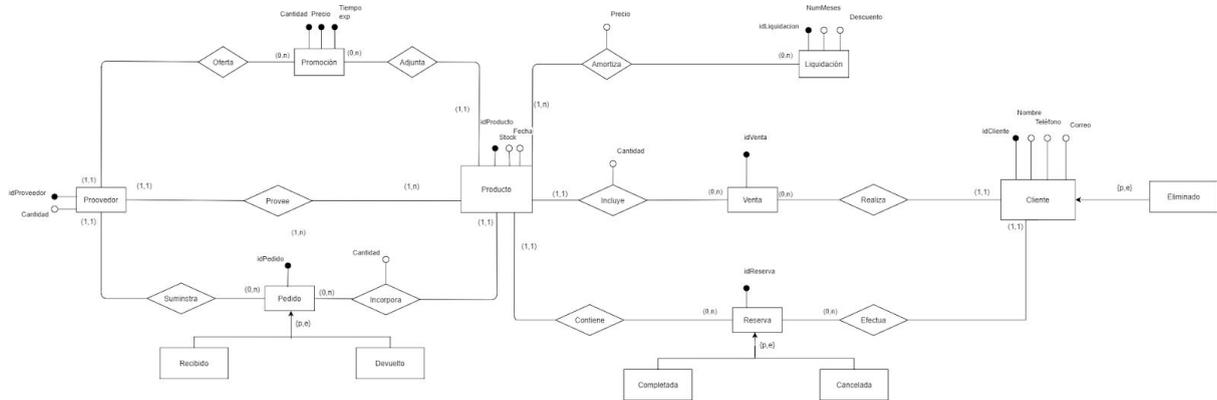
Esquema externo Subsistema Ventas:



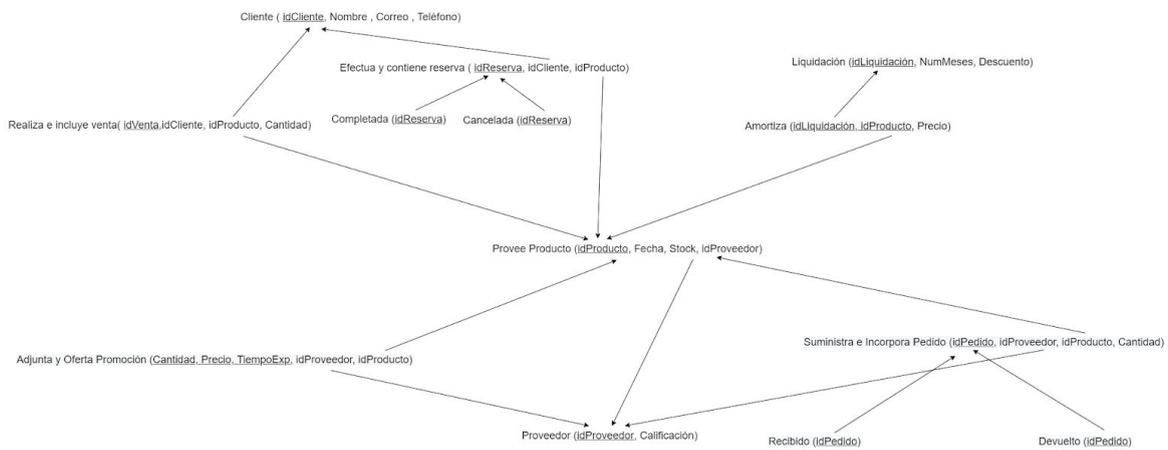
Esquema externo Subsistema Pedidos:



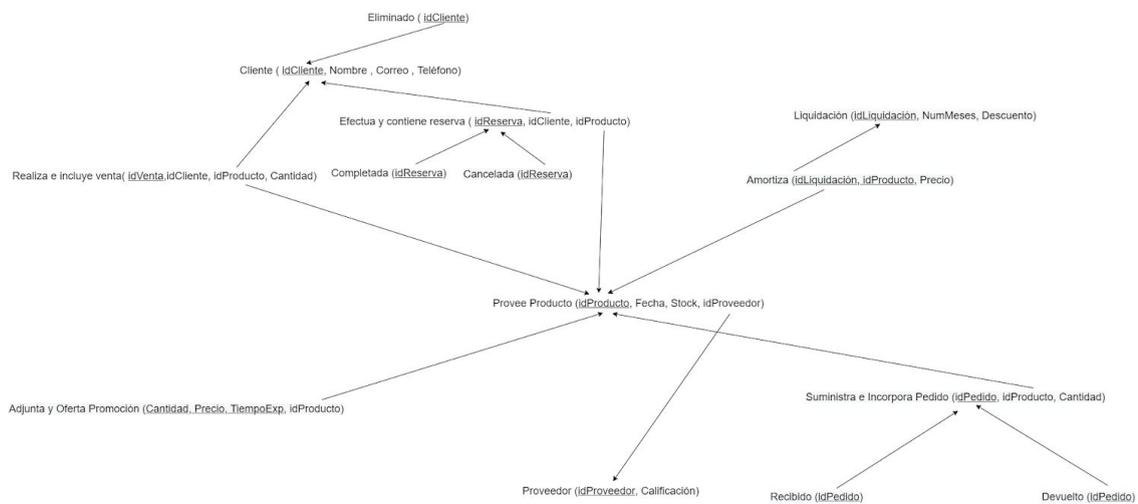
Esquema externo E/R:



Paso a Tablas:



Paso a Tablas Normalizado:



DEPENDENCIAS FUNCIONALES, PROCESO DE NORMALIZACIÓN Y PASO A TABLAS

- **Cliente** (#idCliente, Nombre, Teléfono, Correo)
#idCliente -> Nombre, Teléfono, Correo
- **Eliminado**(#idCliente)
- **Realiza e incluye venta**(#idVenta, idCliente, idProducto)
#idVenta -> idCliente, idProducto
- **Efectua y contiene reserva** (#idReserva, idCliente, idProducto)
#idReserva -> idCliente, idProducto
- **Cancelada** (#idReserva)
- **Completada** (#idReserva)
- **Pendiente** (#idReserva)

- **Liquidación** (#idLiquidación, NumMeses, Descuento)
#idLiquidación -> NumMeses, Descuento
- **Amortiza** (idLiquidación, idProducto, Precio)
{idLiquidación, idProducto} -> Precio

- **Provee Producto** (#idProducto, Fecha, Stock, idProveedor)

#idProducto -> Fecha, Stock, idProveedor

- **Adjunta y oferta promoción** (Cantidad, Precio, TiempoExp, idProveedor, idProducto)

{Cantidad, Precio, TiempoExp} -> idProveedor, idProducto

idProducto -> idProveedor

Vemos que la relación no está en FNBC ya que existe una dependencia funcional donde el determinante no es clave candidata (idProducto). Al normalizar nos salen 2 relaciones:

R1(Cantidad, Precio, TiempoExp, idProducto)

{Cantidad, Precio, TiempoExp} -> idProducto

R2(#idProducto, idProveedor)

#idProducto-> idProveedor

Pero la relación R2 ya está incluida en "Provee Producto" por lo que sólo haría falta sustituir Adjunta y oferta promoción por R1

- **Suministra e incorpora Pedido** (#idPedido, idProveedor, idProducto)

#idPedido -> idProveedor, idProducto

idProducto -> idProveedor

Vemos que la relación no está en FNBC ya que existe una dependencia funcional donde el determinante no es clave candidata (idProducto). Al normalizar nos salen 2 relaciones:

R1'(#idPedido, idProducto)

#idPedido -> idProducto

R2'(#idProducto, idProveedor)

#idProducto -> idProveedor

Pero la relación R2' ya está incluida en "Provee Producto" por lo que sólo haría falta sustituir Suministra e incorpora Pedido por R1'

- **Recibido** (#idPedido)
 - **En Tránsito** (#idPedido)
 - **Devuelto** (#idPedido)
-
- **Proveedor** (#idProveedor, Calificación)
#idProveedor-> Calificación

En el paso a tabla de la versión normalizada hemos utilizado los nombres:

- **Adjunta y oferta promoción** para R1
- **Suministra e incorpora Pedido** para R2

SENTENCIAS SQL

CREATE TABLE Proveedor(
idProveedor int Primary Key not null,
Calificacion INT NULL);

CREATE TABLE Producto(
idProducto INT PRIMARY KEY NOT NULL,
Fecha Date,
Stock INT,
idProveedor INT NOT NULL,
Precio FLOAT,
FOREIGN KEY(idProveedor) REFERENCES Proveedor
(idProveedor)
);

CREATE TABLE Cliente(
idCliente INT PRIMARY KEY not null ,
Nombre VARCHAR2(25),
Correo VARCHAR2(25) UNIQUE,
Telefono INT
);

CREATE TABLE Venta(

idVenta INT PRIMARY KEY not null,

idCliente INT NOT NULL,

idProducto INT NOT NULL,

Cantidad INT NOT NULL,

(idCliente),
FOREIGN KEY(idCliente) REFERENCES Cliente

(idProducto)
FOREIGN KEY(idProducto) REFERENCES Producto

);

CREATE TABLE Reserva(

idReserva INT PRIMARY KEY not null,

idCliente INT NOT NULL,

idProducto INT NOT NULL,

Cantidad INT NOT NULL,

(idCliente),
FOREIGN KEY(idCliente) REFERENCES Cliente

(idProducto)
FOREIGN KEY(idProducto) REFERENCES Producto

);

CREATE TABLE Completada(

idReserva INT PRIMARY KEY not null,

```
FOREIGN KEY(idReserva) REFERENCES Reserva
(idReserva)
);
```

```
CREATE TABLE Cancelada(
    idReserva INT PRIMARY KEY not null,
    FOREIGN KEY(idReserva) REFERENCES Reserva
(idReserva)
);
```

```
CREATE TABLE Liquidacion(
    idLiquidacion INT PRIMARY KEY NOT NULL,
    NumMeses INT NOT NULL,
    Descuento INT NOT NULL
);
```

```
CREATE TABLE Amortiza(
    idLiquidacion INT NOT NULL,
    idProducto INT NOT NULL,
    Precio FLOAT NOT NULL,
    PRIMARY KEY (idLiquidacion, IdProducto),
    FOREIGN KEY(idLiquidacion) REFERENCES
Liquidacion (idLiquidacion),
```

```
FOREIGN KEY(idProducto) REFERENCES Producto
(idProducto)
);
```

```
CREATE TABLE Promocion(
    Cantidad INT NOT NULL,
    Precio FLOAT NOT NULL,
    TiempoExp INT NOT NULL,
    idProducto INT NOT NULL,
    PRIMARY KEY (Cantidad, Precio, TiempoExp),
    FOREIGN KEY(idProducto) REFERENCES Producto
(idProducto)
);
```

```
CREATE TABLE Pedido(
    idPedido INT PRIMARY KEY NOT NULL,
    idProducto INT NOT NULL,
    Cantidad INT NOT NULL,
    FOREIGN KEY(idProducto) REFERENCES Producto
(idProducto)
);
```

```
CREATE TABLE Recibido(
```

```
idPedido INT PRIMARY KEY NOT NULL,  
FOREIGN KEY(idPedido) REFERENCES Pedido  
(idPedido)  
);
```

```
CREATE TABLE Devuelto(  
idPedido INT PRIMARY KEY NOT NULL,  
FOREIGN KEY(idPedido) REFERENCES Pedido  
(idPedido)  
);
```

```
CREATE TABLE Eliminado(  
idCliente INT PRIMARY KEY NOT NULL,  
FOREIGN KEY(idCliente) REFERENCES Cliente  
(idCliente)  
);
```

INSERTAR TUPLAS

- En primer lugar insertamos los productos y los proveedores:

```
INSERT INTO Proveedor (idProveedor) VALUES (1);
```

```
INSERT INTO Proveedor (idProveedor) VALUES (2);
```

```
INSERT INTO Proveedor (idProveedor) VALUES (3);
```

```
INSERT INTO Proveedor (idProveedor) VALUES (4);
```

```
INSERT INTO Proveedor (idProveedor) VALUES (5);
```

```
INSERT INTO Proveedor (idProveedor) VALUES (6);
```

```
INSERT INTO Proveedor (idProveedor) VALUES (7);
```

```
INSERT INTO Proveedor (idProveedor) VALUES (8);
```

```
INSERT INTO Proveedor (idProveedor) VALUES (9);
```

```
INSERT INTO Proveedor (idProveedor) VALUES (10);
```

```
INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (1,TO_DATE('2021/01/01',  
'yyyy/mm/dd'),3,1);
```

```
INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (2,TO_DATE('2021/01/01',  
'yyyy/mm/dd'),5,2);
```

```
INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (3,TO_DATE('2021/01/01',  
'yyyy/mm/dd'),7,6);
```

```
INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (4,TO_DATE('2021/01/01',  
'yyyy/mm/dd'),9,4);
```

```
INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (5,TO_DATE('2021/01/01',  
'yyyy/mm/dd'),11,8);
```

```
INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (6,TO_DATE('2021/01/01',
'yyyy/mm/dd'),13,7);

INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (7,TO_DATE('2021/01/01',
'yyyy/mm/dd'),15,9);

INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (8,TO_DATE('2021/01/01',
'yyyy/mm/dd'),17,1);

INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (9,TO_DATE('2021/01/01',
'yyyy/mm/dd'),21,5);

INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (10,TO_DATE('2021/01/01',
'yyyy/mm/dd'),25,2);

INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (11,TO_DATE('2008/05/08',
'yyyy/mm/dd'),25,2);

INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (12,TO_DATE('2005/06/05',
'yyyy/mm/dd'),25,6);

INSERT INTO Producto (idProducto,Fecha,Stock,idProveedor) VALUES (13,TO_DATE('2019/05/12',
'yyyy/mm/dd'),25,8);
```

MODIFICACIONES PRÁCTICA 2

- Inserción de tuplas y creación de tablas. Hemos cambiado la inserción de productos, proveedores y eliminado la inserción de ventas y clientes (ya que eso lo hace el sistema). En cuanto a la creación de tablas, había algunos errores.
- Normalización y paso a tablas:
 - Hemos creado una relación nueva llamada Eliminado, la hemos normalizado y paso a tablas.
- Esquemas externos y esquema E/R:
 - Modificación de la especialización de Cliente llamada eliminado.
 - Modificación de la especialización de Pedido, que antes era total y exclusiva por parcial y exclusiva (hemos quitado la especialización “en tránsito” ya que todo pedido tiene que pasar por ese estado)
 - Modificación de la especialización de Reserva, que antes era total y exclusiva por parcial y exclusiva (hemos quitado la especialización “pendiente ya que toda reserva tiene que pasar por ese estado)
 - En el esquema externo del subsistema Ventas hemos añadido reserva a la entidad Venta (para contemplar las ventas por reserva)
 - En el esquema externo del subsistema Proveedores hemos añadido reserva a calificar proveedor.

PRÁCTICA 3

DESCRIPCIÓN DE LAS TRANSACCIONES UTILIZADAS:

Para el uso de transacciones en lenguaje Java hemos utilizado la sentencia:

```
“jdbcConnection.setAutoCommit(false);”
```

para deshabilitar los commit automáticos que se realizaban después de la ejecución de una sentencia PL/SQL.

Hemos creado **una transacción por cada submenú** del sistema de manera que tras realizar una serie de operaciones del subsistema tengamos la opción de volver atrás sin guardar cambios, es decir, se ejecuta un rollback al inicio del submenú, saliendo de él.

Al inicio del submenú, se ejecuta

```
"SET TRANSACTION NAME 'transaccion' "
```

para crear una transacción.

CÓDIGO DE LOS DISPARADORES

1.- Subsistema clientes

```
create or replace TRIGGER ELIMINARVENTA
```

```
BEFORE INSERT ON Eliminado
```

```
FOR EACH ROW
```

```
DECLARE
```

```
    cuantos int;
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO cuantos FROM Venta WHERE idCliente= :new.idCliente;
```

```
    IF(cuantos!= 0) THEN
```

```
        RAISE_APPLICATION_ERROR(-20015,'No se puede eliminar el cliente ya que  
ha hecho una compra');
```

```
    END IF;
```

```
END;
```

```
/
```

2.- Subsistema Ventas

create or replace TRIGGER COMPROBARSTOCKRESERVA

AFTER INSERT ON Completada

FOR EACH ROW

DECLARE

cantidadpro INT;

cantidadstock INT;

idpro INT;

BEGIN

SELECT Cantidad INTO cantidadpro FROM Reserva WHERE idReserva = :new.idReserva;

SELECT idProducto INTO idpro FROM Reserva WHERE idReserva = :new.idReserva;

SELECT Stock INTO cantidadstock FROM Producto WHERE idProducto= idpro;

IF(cantidadstock>=cantidadpro) THEN

UPDATE Producto set Stock = (cantidadstock - cantidadpro) WHERE idProducto= idpro;

ELSE

RAISE_APPLICATION_ERROR(-20005,'Stock insuficiente');

END IF;

END;

/

create or replace TRIGGER COMPROBARSTOCKVENTA

AFTER INSERT ON Venta

FOR EACH ROW

DECLARE

cantidadpro INT;

BEGIN

SELECT Stock INTO cantidadpro FROM Producto WHERE idProducto=
:new.idProducto;

IF(cantidadpro>=:new.Cantidad) THEN

UPDATE Producto set Stock = (cantidadpro -:new.Cantidad) WHERE idProducto=
:new.idProducto;

ELSE

RAISE_APPLICATION_ERROR(-20014,'Stock insuficiente');

END IF;

END;

/

3.- Subsistema Pedidos

create or replace TRIGGER COMPROBARPEDIDO

BEFORE INSERT ON Devuelto

FOR EACH ROW

DECLARE

 cuantos int;

BEGIN

 SELECT COUNT(*) INTO cuantos FROM Recibido WHERE idPedido=
:new.idPedido;

 IF(cuantos!=0) THEN

 RAISE_APPLICATION_ERROR(-20010,'El pedido no se puede devolver porque
ya ha sido recibido');

 END IF;

END;

/

4.- Subsistema Proveedores

create or replace TRIGGER COMPROBARLIQUIDACION

BEFORE INSERT ON Amortiza

FOR EACH ROW

DECLARE

 cuantos int;

BEGIN

 SELECT COUNT(*) INTO cuantos FROM Amortiza WHERE idProducto=
:new.idProducto;

 IF(cuantos!=0) THEN

 RAISE_APPLICATION_ERROR(-20011,'El producto no se puede liquidar de
nuevo');

 END IF;

END;

/

BREVE MOTIVACIÓN DE LA ELECCIÓN DEL SOFTWARE:

Para la implementación del sistema de información hemos utilizado el lenguaje de propósito general Java.

Como conexión al SGBD hemos utilizado JDBC (ojdbc10.jar) que nos permite la ejecución de operaciones sobre la base de datos así como una gestión de manejadores. Nos ofrece las siguientes clases: DriverManager, para cargar un driver; Connection, para establecer conexiones; PreparedStatement y Statement, para ejecutar sentencias PL/SQL; y ResultSet para almacenar el resultado de la orden PL/SQL ejecutada.

Hemos utilizado JDeveloper para visualizar los cambios que se van realizando en la base de datos.