Proyecto Individual

Especificación de Requisitos de Software

IEEE Std. 830-1998

Proyecto Individual

Autor: Luis Jaramillo Pulido

ÍNDICE

1. Introducción

1.1. Propósito

El presente documento de Especificación de Requisitos Software tiene como propósito definir de manera clara y precisa las funcionalidades y restricciones que tendrá la aplicación que se desea construir.

1.2. Alcance

El documento cubrirá la especificación de clientes, productos y ventas, que consiste en la implementación de un software de gestión administrativa. Se detallará cualquier transacción derivada de la gestión de cualquiera de los módulos.

Los datos estarán disponibles a través de la propia interfaz, pudiendo en cualquier momento mostrar la información que se precise. Dicha interfaz, será sencilla y de fácil manejo.

1.3. Definiciones, Acrónimos y Abreviaturas

SRS: Especificación de Requisitos Software (del inglés Software Requirements Specification); Este mismo documento es una SRS.

IS: Ingeniería del Software; el conjunto de buenas prácticas y hábitos en el desarrollo de productos software que permite garantizar la calidad y mantenibilidad del trabajo.

IEEE: Institute of Electrical and Electronics Engineers y, por metonimia, el conjunto de estándares que dicha organización redacta; Se trata de una de las referencias más importantes en la práctica de la IS.

BD: Base de datos; El sistema que gestionará la información que manejará la aplicación.
SQA: Software Quality Assurance; el conjunto de prácticas de IS centradas en la supervisión del trabajo de diseño e implementación. Suele ser llevado a cabo por un equipo independiente que no estará en este proyecto por falta de personal.

IU: Interfaz de Usuario; se trata de la parte visual de la aplicación con la que interactúa el usuario final.

2. Descripción General

2.1. Perspectiva del Producto

Se pretende implementar una aplicación, que gestione administrativamente la información relativa a sus clientes, productos y ventas en los que trabajan.

2.2. Funciones

Para cada módulo se van a implementar las siguientes funciones:

· Clientes

Habrá un subsistema que se encargará de la gestión de los clientes, así como su alta y baja del sistema. Se les identificará por el DNI y habrá dos tipos de empleados: socio y estándar.

Casos de uso:

- Alta cliente.
- Baja cliente
- Modificar cliente
- Consultar Cliente
- Listar Cliente

· Productos

El subsistema de productos permite tener un numeroso catálogo de elementos con sencillez a la hora de hacer pedidos desde Ventas. Cada producto tendrá como elementos su nombre, el precio, la categoría a la que pertenece y el stock.

Casos de uso:

- Alta producto.
- Baja producto.
- Modificar Producto
- Consultar departamento.
- Listar productos.

· Ventas

Un subsistema que se encarga de los pedidos que hacen los clientes al comprar los productos.

Casos de uso:

- Crear Venta.
- Listar Venta.
- Mostrar el nombre del cliente de una venta
- Listar productos de una venta mayores a un precio dado
- Agregar Producto al carrito
- Quitar Producto del carrito

2.3. Características de los Usuarios

El único usuario será el **administrador (Usuario)**. El cual tendrá todos los privilegios sobre el software y gestionará las características del mismo.

2.4. Restricciones

-La Arquitectura será Multicapa.

-La aplicación será de escritorio.

-Se usará el software IBM RSA para el desarrollo de diagramas de Casos de Uso, de actividad, de secuencia, de clase.

-El código será escrito en JAVA y se usaran patrones de diseño previamente seleccionados.

-Para la gestión de la base de datos se usará Sqlite.

2.5. Suposiciones y Dependencias

Se considera que la suposición de que estos requisitos son estables, no se producirá ningún cambio a petición de este, no se añadirán nuevas funcionalidades.

3. Requisitos Específicos

3.1. Interfaces Externos

· Interfaz Gráfica

La interfaz gráfica con la que el usuario final interactúa deberá ser intuitiva de manera que, sin un manual de uso, el usuario identifique rápidamente los componentes y las secciones del sistema.

Las interfaces de usuario están relacionadas con los módulos descritos en la SRS. Para modificar los datos se deberá realizar por medio del teclado y el Mouse (ratón).

La interfaz estará formada por dos niveles de pantallas. El primer nivel contendrá una sola ventana de empleados donde aparecerá todas las posibles operaciones a realizar. El segundo nivel lo formarán la ventana de departamentos que contendrá las operaciones relacionadas con este módulo.

La interfaz gráfica seguirá el mismo tratamiento que la de usuario, de forma que esta ayude al usuario a manejar la aplicación sin necesidad de ningún tipo de ayuda externa como, por ejemplo, un manual de instrucciones.

El sistema se basará en un control de los empleados agrupados en departamentos.

Las siguientes pantallas son diferentes *mock up* (prototipo de pantalla) iniciales de la interfaz de usuario que tendrá la aplicación final.

· Interfaz hardware

Será necesario disponer de equipos de cómputo en un buen estado con las siguientes características:

- Adaptadores de red
- Teclado
- Ratón
- Impresora
- Memoria mínima interna de 1Gb

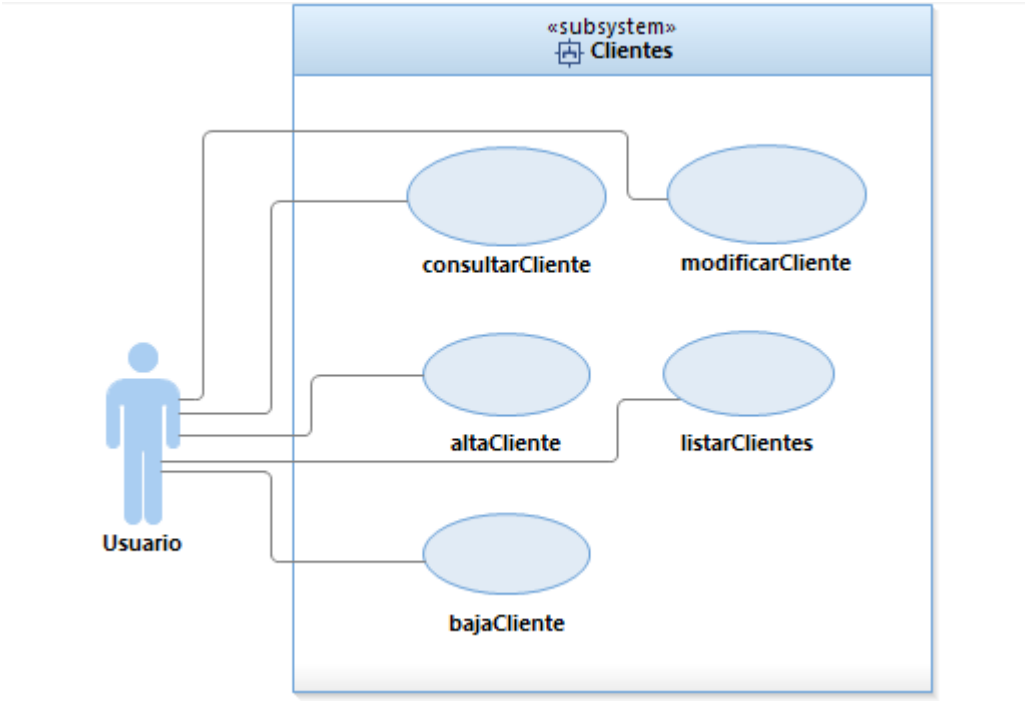
· Interfaz software

- Sistema operativo: Windows 7 o superior, distribución GNU/Linux compatible con

Java 8 SE y MAC OS X Mountain Lion y superiores.
- Aplicaciones adicionales: JAVA 8 SE.

3.2. Funciones

3.2.1 Clientes



Caso de uso: Alta Cliente

Identificador: CLI-1
Objetivo en Contexto: Dar de alta a un nuevo cliente con todos los datos recibidos por la entrada.
Actor principal: Administrador.
Actores secundarios: Base de Datos
Entrada Datos: Nombre, Apellidos, DNI, Dirección.
Éxito: Crea el cliente, guardando todos sus campos en la BBDD.
Fallo: No se crea el cliente.
Salida Datos: Se le adjudica un ID al cliente

Flujo principal

1. El usuario selecciona pestaña Cliente.
2. El usuario introduce todos los campos de los datos de entrada.
3. El usuario pulsa en alta cliente.
4. El sistema comprueba que los datos son correctos.
5. El sistema comprueba que ni el DNI o el nombre y apellidos ya estén la BBDD.
6. El sistema introduce al cliente en la BBDD, asignándole un ID.
7. Se pone el atributo "Activo" a true.

Flujos secundarios

4.a) Si algún dato no es correcto, se muestra un mensaje con el error (*Pantalla: Mensaje de Error datos).*

5.a) Si el cliente ya existe, porque bien su DNI o nombre y apellidos ya están registrados, se muestra un mensaje de error .

Caso de uso: Baja Cliente

Identificador: CLI-2
Objetivo en Contexto: Elimina todos los datos del cliente seleccionado, que puede ser realizada por ID de cliente o DNI.
Actor principal: Administrador.
Actores secundarios: BBDD **Entrada Datos:** ID o DNI.
Éxito: Da de baja al cliente.
Fallo: No se da de baja al cliente.
Salida Datos: Muestra los datos.

Flujo principal

1. El usuario introduce el ID del cliente a eliminar.
2. El usuario pulsa en eliminar cliente.
3. El sistema comprueba si existe el ID del cliente.
4. El sistema comprueba si el cliente está activo comprobando el atributo "Activo".
5. Se pone el atributo "Activo" a false.

Flujos secundarios

3.a) El cliente a eliminar no existe, porque bien el ID introducido no está en la BBDD y se muestra un mensaje de error.

4.a) El cliente está ya inactivo (dado de baja). Se muestra un mensaje de error y termina la ejecución.

Caso de uso: Modificar Cliente

Identificador: CLI-3
Objetivo en Contexto: Modificar todos o algún dato de un cliente seleccionado, la selección del cliente puede ser realizada por ID o DNI
Actor principal: Administrador.
Actores secundarios: Base de Datos.
Entrada Datos: ID o DNI.
Éxito: Modifica los datos del cliente.
Fallo: No se hace ninguna modificación.
Salida Datos: Muestra los datos del cliente con los cambios.

Flujo principal

1. El usuario introduce el ID del cliente a modificar.
2. El usuario pulsa en modificar cliente.
3. El sistema comprueba que los datos a modificar son correctos.
4. El sistema modifica los datos del cliente.

Flujos secundarios

2a) El cliente consultado no existe, porque bien el DNI o ID introducido no está en la BBDD, se muestra un mensaje de error.
3.a) Si algún dato no es correcto, se muestra un mensaje con el error *(Pantalla: Mensaje de Error datos).*

Caso de uso: Consultar Cliente

Identificador: CLI-4
Objetivo en Contexto: Muestra todos los datos del cliente seleccionado para realizar la consulta, que puede ser realizada por ID o DNI.
Actor principal: Administrador.
Actores secundarios: Base de Datos.
Entrada Datos: ID o DNI
Éxito: Muestra todos los datos del cliente.
Fallo: No se muestra nada.
Salida Datos: Muestra los datos.

Flujo principal

1. El usuario introduce el ID o el DNI del cliente a consultar.
2. El usuario pulsa en Consultar Cliente.
3. El sistema comprueba que los datos son correctos.
4. El sistema muestra el cliente consultado.

Flujos secundarios

3.a) Si algún dato no es correcto, se muestra un mensaje con el error *(Pantalla: Mensaje de Error datos).*

4.a) El cliente consultado no existe, porque bien el DNI o ID introducido no está en la BBDD, se muestra un mensaje de error .

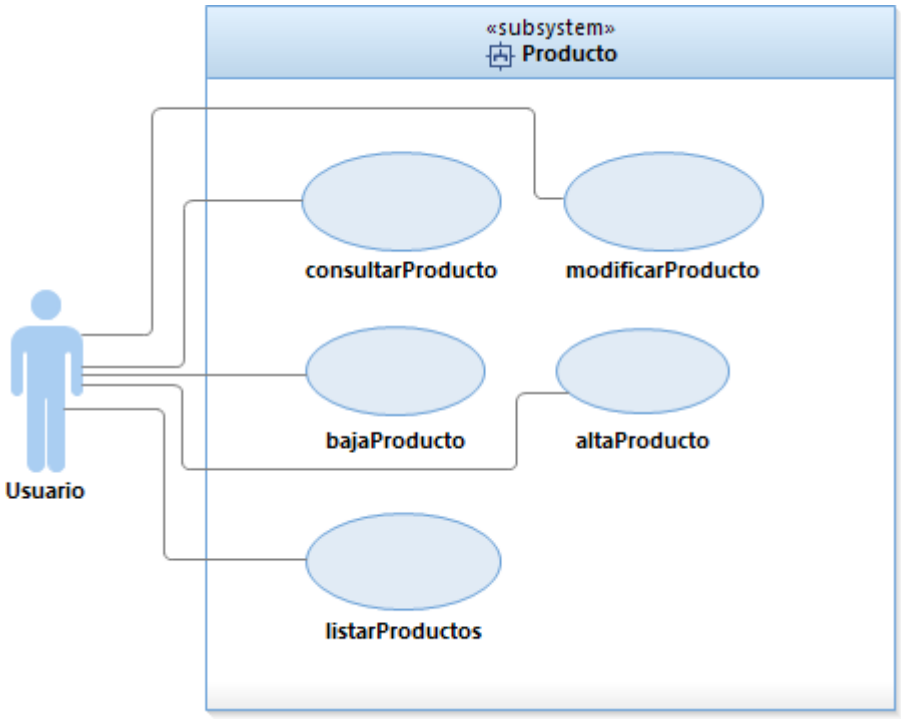
Caso de uso: Listar Clientes

Identificador: CLI-5
Objetivo en Contexto: Muestra todos los clientes guardados en la BBDD.
Actor principal: Administrador.
Actores secundarios: Base de Datos.
Entrada Datos: Nada.
Éxito: Muestra todos los datos de todos los clientes.
Fallo: No se muestra nada.
Salida Datos: Lista con todos los datos de todos los clientes.

Flujo principal

1. El usuario pulsa en listar clientes.
2. El sistema muestra todos los datos de todos los clientes.

3.2.2 Productos



Caso de uso: Alta Producto

Identificador: PRO-1
Objetivo en Contexto: Agregar un producto a nuestro almacén de productos mediante los datos de entrada.
Actor principal: Administrador.
Actores secundarios: Base de Datos.
Entrada Datos: Nombre del producto, precio del producto, categoría del producto.
Éxito: Se añade el producto a la BBDD.
Fallo: No se añade el producto a la BBDD y se muestra un mensaje de error por pantalla.
Salida Datos: muestra el producto dado de alta

Flujo principal:

1. El administrador introduce todos los campos de los datos de entrada.
2. El administrador pulsa alta producto.
3. El sistema comprueba que los datos son correctos.
4. El sistema comprueba el nombre de producto y la plataforma para que no exista previamente en la BBDD.
5. El sistema introduce el producto en la BBDD, asignándole un ID y un atributo "Activo".

Flujos secundarios:

3.a) Si algún dato no es correcto, se muestra un mensaje con el error *(Pantalla: Mensaje de Error datos).*
4.a) Si el producto ya existe, porque ya existe un producto con el mismo nombre y plataforma, se muestra un mensaje de error.

Caso de uso: Baja Producto

Identificador: PRO-2
Objetivo en Contexto: Dado un producto, lo buscamos en la BBDD y lo eliminamos.
Actor principal: Administrador
Actores secundarios: Base de Datos
Entrada Datos: ID del producto.
Éxito: Se da de baja el producto y se actualiza la BBDD.
Fallo: No se da de baja el producto y se muestra un mensaje de error por pantalla.
Salida Datos: Muestra la lista de productos activos.

Flujo principal:

1. El administrador introduce el ID del producto a eliminar.
2. El administrador pulsa en Baja producto.
3. El sistema comprueba que el ID existe en la BBDD.
4. El sistema comprueba que el producto esté activo mediante el atributo "Activo".
5. El sistema da de baja el producto de la BBDD poniendo el atributo "Activo" a false y la actualiza.

Flujos secundarios:

3.a) Si el ID no se encuentra en la BBDD, se muestra un mensaje con el error *(Pantalla: Mensaje de Error datos).*
4.a) El producto ya está inactivo (dado de baja), se muestra un mensaje por pantalla con el estado del producto y acaba la ejecución.

Caso de uso: modificar Producto

Identificador: PRO-3
Objetivo en Contexto: Modificar todos o algún dato de un producto seleccionado, la selección del producto se realiza por ID.
Actor principal: Administrador.
Actores secundarios: Base de Datos.
Entrada Datos: ID del producto.
Éxito: Modifica los datos del producto.
Fallo: No se hace ninguna modificación.
Salida Datos: Muestra los datos del producto con los cambios.

Flujo principal:

1. El administrador introduce el ID del producto a modificar.
2. El administrador pulsa en modificar producto.
3. El administrador introduce los datos a modificar.
4. El sistema comprueba que los datos a modificar son correctos.
5. El sistema modifica los datos del producto.

Flujos secundarios:

2.a) El producto que se quiere modificar no existe y se muestra un mensaje de error.

3.a) Si algún dato no es correcto, se muestra un mensaje con el error *(Pantalla: Mensaje de Error datos).*

Caso de uso: Consultar Producto

Identificador: PRO-4
Objetivo en Contexto: Muestra todos los datos del producto seleccionado por ID.
Actor principal: Administrador.
Actores secundarios: Base de Datos.
Entrada Datos: ID.
Éxito: Muestra todos los datos del producto.
Fallo: Se muestra un mensaje de error.
Salida Datos: Muestra los datos del producto.

Flujo principal:

1. El administrador introduce el ID del producto a consultar.
2. El administrador pulsa en consultar producto.
3. El sistema comprueba que el ID existe en la BBDD.
4. El sistema muestra el producto consultado.

Flujos secundarios:

3.a) Si el ID no existe en la BBDD, se muestra un mensaje con el error *(Pantalla: Mensaje de Error datos).*

Caso de uso: Listar productos

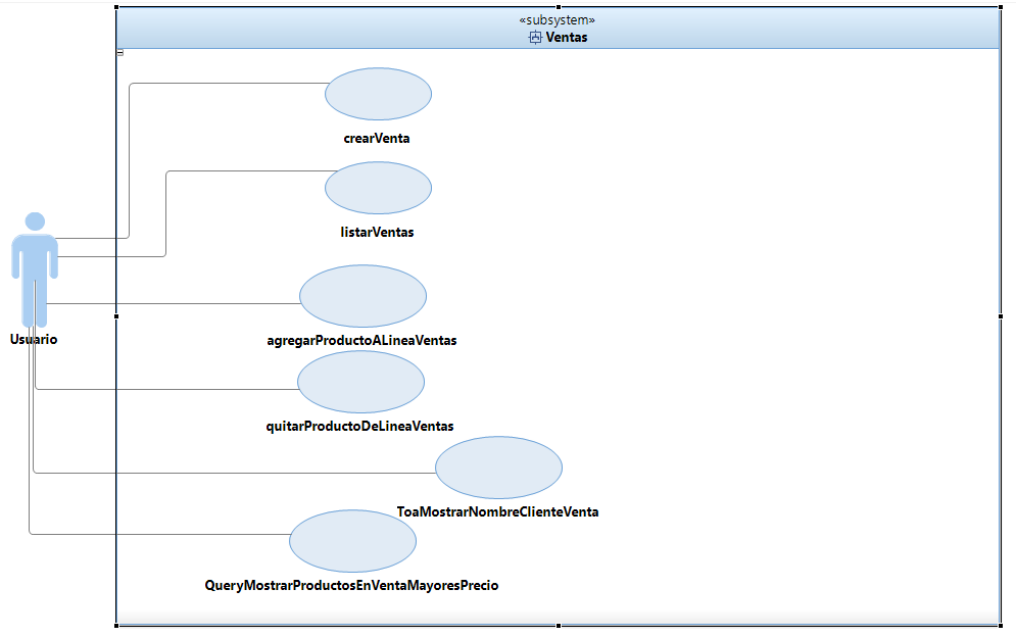
Identificador: PRO-5
Objetivo en Contexto: Muestra todos los productos guardados en la BBDD.
Actor principal: Administrador.
Actores secundarios: Base de Datos.
Entrada Datos: Nada.
Éxito: Muestra todos los datos de todos los productos.
Fallo:
Salida Datos: Lista con todos los datos de todos los productos.

Flujo principal:

1. El sistema muestra todos los datos de todos los productos.

.

3.2.3 Ventas


«subsystem»
Ventas
crearVenta
listarVentas
agregarProductoALineaVentas
quitarProductoDeLineaVentas
ToaMostrarNombreClienteVenta
QueryMostrarProductosEnVentaMayoresPrecio
Usuario

Caso de uso: Crear Venta

Identificador: VEN-1
Objetivo en Contexto: Se crea una venta y se asocia el cliente a la línea de ventas.
Actor principal: Administrador.
Actores Secundarios: Base de Datos.
Entrada Datos: id del Cliente, y carrito cuyos elementos son Linea de Venta
Éxito: Se ha creado una venta y se ha asociado el cliente a ventas.
Fallo: Mensaje de error notificando que cliente no existe / o no hay stock de algunos de los productos del carrito y por lo tanto no se ha creado la venta.
Salida Datos: Muestra la venta, el id de la venta asociado al id del cliente y el precio total de la venta.

Flujo Principal

1. El usuario introduce los productos al carrito y el id del cliente.
2. El usuario pulsa en crear venta.
3. El sistema comprueba que el cliente exista.
4. El sistema comprueba que los productos del carrito tienen stock
5. Si todos los datos son correctos, se procede a crear venta.

Flujos Secundarios

3.a) No existe cliente, por lo que no se puede crear la venta, saldrá un mensaje indicando “cliente no existe y no es posible crear la venta”

4.a) No hay stock de un producto de la cesta, el sistema informara que no se ha podido crear venta por falta de stock .

Caso de uso: Listar Venta

Identificador: VEN-2
Objetivo en Contexto: Muestra todos los productos añadidos en la línea de ventas (Carrito) guardados en la BBDD.
Actor principal: Empleado de departamento de Ventas.
Actores secundarios: Base de Datos.
Entrada Datos: Nada.
Éxito: Muestra todos los productos de la línea de ventas.
Salida Datos: Lista los clientes asociado a id de venta

Flujo principal:

1. El sistema muestra todas las ventas

Caso de uso: Mostrar Nombre Cliente Venta

Identificador: VEN-3
Objetivo en Contexto: muestra el nombre del cliente de una venta
Actor principal: Administrador.
Actores Secundarios: Base de Datos.
Entrada Datos: id de la venta.
Éxito: muestra por pantalla el nombre del cliente de una venta.
Fallo: Mensaje de error notificando que la venta no existe.
Salida Datos: Muestra el id de la venta, asociado al id del cliente y el nombre del cliente.

Flujo Principal

1. El usuario introduce el id de la venta
2. El usuario pulsa en mostrar nombre cliente venta.
3. El sistema comprueba que la venta exista.
4. El sistema muestra el id de la venta y el nombre del cliente.

Flujos Secundarios

3.a) No existe la venta, saldrá un mensaje indicando “venta no existe”

Caso de uso: Mostrar Productos de una Venta Mayores a un precio

Identificador: VEN-4
Objetivo en Contexto: muestra una lista de productos de una venta que sean mayores a un precio dado por el usuario.
Actor principal: Administrador.
Actores Secundarios: Base de Datos.
Entrada Datos: id de la venta.
Éxito: muestra por pantalla la lista de productos de una venta que sean mayores a un precio dado.
Fallo: Mensaje de error notificando que la venta no existe.
Salida Datos: Muestra el id de los productos, nombre del producto, precio del producto y el stock.

Flujo Principal

1. El usuario introduce el id de la venta y un valor de un precio.
2. El usuario pulsa en mostrar productos de una venta mayores a un precio.
3. El sistema comprueba que la venta exista.
4. El sistema muestra el id de los productos, nombre del producto, precio del producto y el stock.

Flujos Secundarios

3.a) No existe la venta, saldrá un mensaje indicando “venta no existe”

Caso de uso: Agregar Producto a Línea de Ventas

Identificador: VEN-5
Objetivo en Contexto: Agrega productos a línea de ventas.
Actor principal: Clientes.
Actores Secundarios: Base de Datos
Entrada Datos: id producto, cantidad.
Éxito: Se añade producto a línea de ventas.
Fallo: Mensaje de error notificando que el id del producto que se quiere agregar a la línea de venta no existe.
Salida Datos: Muestra los productos agregados en línea de ventas.

Flujo Principal

1. El usuario introduce el id del producto
2. El usuario pulsa en cargar producto
3. El usuario pulsa agregar producto
4. Si todos los datos son correctos, se añade producto a línea de ventas.

Flujos Secundarios

2.a) No existe el producto, saldrá un mensaje de error “no existe el producto”

Caso de uso: Quitar Producto a Línea de Ventas

Identificador: VEN-6
Objetivo en Contexto: Se elimina producto de línea de ventas.
Actor principal: Administrador
Actores secundarios: Base de Datos
Entrada Datos: ID producto**.**
Éxito: Se elimina el producto de la línea de ventas.
Fallo: Mensaje de error notificando que el id del producto que se quiere quitar a la línea de venta no existe.
Salida Datos: Muestra línea de ventas sin el producto.

Flujo principal:

1. El usuario introduce el id del producto.
2. El usuario pulsa cargar el producto
3. El usuario pulsa en quitar producto
4. El sistema comprueba que el ID del producto este asociado a línea de ventas
5. El sistema elimina el producto de línea de ventas.

Flujos secundarios:

2.a) No existe el producto, saldrá un mensaje de error “no existe el producto”

3.3. Requisitos de Rendimiento

Solo podrá haber un tipo de usuario conectado al mismo tiempo al software.
Para el correcto funcionamiento de este software, tenemos que garantizar que la ejecución de consultas no afecte negativamente al rendimiento, pues la BBDD tiene que estar siempre operativa, las transacciones con la BBDD se harán de una en una para evitar posibles conflictos, usando el patrón de transacciones para mantener una orden en la ejecución que no afecte al rendimiento.

3.4. Restricciones de Diseño

El modelo UML 2.x de la aplicación, será en formato IBM Rational Software Architect 9.5.
- El lenguaje de implementación será Java.
- La aplicación será de escritorio
- La persistencia de los datos debe hacerse en formato relacional
- La arquitectura de la aplicación será multicapa.
- Deben aplicarse los siguientes patrones obligatoriamente en la versión de la aplicación:
o Service to worker
o Transfer object
o Transfer Object Assembler
o Application Service
o Data Access Object
o Query

Parte DAO

Modelo de Negocio:

3.5. Atributos del Sistema

Para cumplir con el estándar que se está siguiendo en la creación del programa a través de este documento, el sistema tiene que cumplir las siguientes características:

- **Fiabilidad:** Probabilidad de que un sistema, aparato o dispositivo cumpla una determinada función bajo ciertas condiciones durante un tiempo determinado.

- **Usabilidad:** Cualidad del programa informático que son sencillos de usar porque facilitan la lectura de los textos, descargan rápidamente la información y presentan funciones y menús sencillos, por lo que el usuario encuentra satisfechas sus consultas y cómodo su uso.

- **Accesibilidad:** Posibilidad de acceder a cierta cosa o facilidad para hacerlo.

- **Mantenibilidad:** La **mantenibilidad** es la propiedad de un sistema que representa la cantidad de esfuerzo requerida para conservar su funcionamiento normal o para restituirlo una vez se ha presentado un evento de falla.

- **Seguridad:** Ausencia de riesgo.

- **Extensibilidad:** Factor de calidad del software que consiste en la facilidad de adaptación del software a nuevos requisitos o cambios en la especificación.